DOT/FAA/AR-08/31

Air Traffic Organization
Operations Planning
Office of Aviation Research
and Development
Washington, DC 20591

Networked Local Area Networks in Aircraft: Safety, Security, and Certification Issues, and Initial Acceptance Criteria (Phases 1 and 2)

November 2008

Final Report

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof. The United States Government does not endorse products or manufacturers. Trade or manufacturer's names appear herein solely because they are considered essential to the objective of this report. This document does not constitute FAA certification policy. Consult your local FAA aircraft certification office as to its use.

This report is available at the Federal Aviation Administration William J. Hughes Technical Center's Full-Text Technical Reports page: actlibrary.tc.faa.gov in Adobe Acrobat portable document format (PDF).

Technical Report Documentation Page

1. Report No.

DOT/FAA/AR-08/31

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle

NETWORKED LOCAL AREA NETWORKS IN AIRCRAFT: SAFETY, SECURITY, AND CERTIFICATION ISSUES AND INITIAL ACCEPTANCE CRITERIA (PHASES 1 AND 2)

5. Report Date

November 2008

6. Performing Organization Code

7. Author(s)

Eric Fleischman, Randall E. Smith, and Nick Multari

8. Performing Organization Report No.

9. Performing Organization Name and Address

The Boeing Company
P.O. Box 3707, MC 7L-49
Seattle, WA 98124-2207

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

DTFACT-05-C-00003

12. Sponsoring Agency Name and Address

U.S. Department of Transportation
Federal Aviation Administration
Air Traffic Organization Operations Planning
Office of Aviation Research and Development
Washington, DC 20591

13. Type of Report and Period Covered

Final Report

December 2004-December 2006

14. Sponsoring Agency Code

AIR-120

15. Supplementary Notes

The Federal Aviation Administration Airport and Aircraft Safety R&D Division Technical Monitor was Charles Kilgore.

16. Abstract

This report presents the results of the Federal Aviation Administration (FAA) local area network (LAN) research effort addressing potential safety impacts introduced by LANs in aircraft. Interconnecting previously isolated components on aircraft increases the complexity of unintended interactions between components and provides potential new access points that could be exploited to cause harm. This report addresses the potential security vulnerabilities introduced by networking LANs, the safety affects of security failures, and a process for designing and certifying LANs on aircraft to ensure the safety of these new aircraft systems.

This report extends the current FAA safety assurance processes into airborne networked environments by leveraging the Biba Integrity Model. It builds upon existing FAA studies that articulate mechanisms to integrate RTCA/DO-178B and common criteria processes for the National Airspace System. This approach creates a safety-oriented airborne networked architecture that is built upon existing DO-178B and Aerospace Recommended Practice 4754 safety mechanisms. This produces results that are a direct analog to existing U.S. Department of Defense policies and processes.

17. Key Words

Local area network, Network, Aircraft safety, Aircraft security

18. Distribution Statement

This document is available to the U.S. public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
204

22. Price

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AC	Advisory Circulars
ACARS	Aircraft Communications and Reporting System
AFDX	Avionics Full-Duplex Switched [Ethernet]
AJ	Anti-jamming
ARP	Aerospace Recommended Practice
AS	Autonomous system
ASBR	AS boundary router (alternatively: AS Border Router)
ATN	Aeronautical Telecommunications Network
BGP	Border Gateway Protocol
CA	Certificate Authority
CC	Common criteria
CE	Customer edge
CERT	Computer Emergency Response Team
CFDIU	Central fault display information unit
CIDR	Classless interdomain routing
CIM	Common information model
CIO	Chief Information Officer
CLNP	Connectionless network protocol
COMSEC	Communications security
CONOPS	Concept of Operations
CONS	Connection-oriented network service
COPS	Common Open Policy Service
COTS	Commercial off-the-shelf
CPU	Central processing unit
CRC	Cyclic redundancy check
DAA	Designated approving authority
DARPA	Defense Advanced Research Projects Agency
DEN	Directory-enabled networking
DHCP	Dynamic host configuration protocol
DMTF	Distributed Management Task Force
DNS	Domain Name System
DoD	Department of Defense
DoDI	Department of Defense Instruction
DoS	Denial of service
DSS	Digital Signature Standard
EAL	Evaluation assurance level
EFB	Electronic flight bag
EGP	Exterior gateway protocol
ESP	Encapsulating security payload
FAA	Federal Aviation Administration
FIPS	Federal Information Processing Standard
FTP	File transfer protocol
GIG	Global information grid
HAG	High-assurance guard

HMAC	Hashed Message Authentication Code
HTTP	Hypertext transfer protocol
IA	Information assurance
IATF	Information Assurance Technical Framework
ICMP	Internet control message protocol
IDS	Intrusion detection system
IETF	Internet Engineering Task Force
IGP	Interior gateway protocol
IMA	Integrated modular avionics
IP	Internet protocol
IPsec	Internet protocol security
IPSP	Internet protocol security policy
ISMS	Integrated security model for SNMP
ISP	Internet service providers
IS-IS	Intermediate system to intermediate system
IT	Information technology
L2VPN	Layer 2 virtual private network
L3VPN	Layer 3 virtual private network
LAN	Local area network
LDAPv3	Lightweight directory access protocol version 3
LLC	Limited Liability Corporation
LPI/LPD	Low probability of intercept/low probability of detection
MAC	Mission assurance category
MANET	Mobile ad hoc networking
MIB	Management information base
MIP	Mobile Internet protocol
MOSPF	Multicast open shortest path first (protocol)
MPLS	Multiprotocol label switching
MSLS	Multiple single levels of security
NAS	National airspace system
NAT	Network address translator
NEMO	Network mobility
NFS	Network file system
NIDS	Network intrusion detection system
NIST	National Institute of Standards and Technology
NMS	Network management system
NSA	National Security Agency
NTP	Network time protocol
OMT	Onboard maintenance terminal
OS	Operating system
OSI	Open system interconnect
OSPF	Open shortest path first (protocol)

PBN	Policy-based networking
PC	Personal computer
PE	Provider edge
PEP	Policy enforcement point
PFS	Perfect forward secrecy
PIB	Policy information base
PIM-DM	Protocol-independent multicast-dense mode
PIM-SM	Protocol-independent multicast-sparse mode
PKI	Public key infrastructure
PPS	Ports protocols and services
QoS	Quality of service
RAM	Random access memory
RFC	Request for comment (i.e., Publications of the IETF)
RPC	Remote procedure call
RSA	Rivest Shamir Addleman
RTP	Real-time protocol
SA	Security association
SATS	Small aircraft transportation system
SBU	Sensitive but unclassified
SLA	Service level agreement
SMTP	Simple mail transfer protocol
SNMP	Simple network management protocol
SPD	Security policy database
SSE	System security engineering
SSH	Secure shell (protocol)
SWAP	Size, weight, and power
SYN	Synchronous (bit)
TCP	Transmission control protocol
TCSEC	Trusted Computer System Evaluation Criteria
TFTP	Trivial file transfer protocol
TLL	Time to live
TLS	Transport layer security
TP	Transport layer protocol
TSIG	Secret key transaction authentication for DNS
UDP	User datagram protocol
V	Version
VPN	Virtual private network
WAN	Wide area network

EXECUTIVE SUMMARY

This is the final report of a 2-year Federal Aviation Administration (FAA)-funded study to address security and safety issues associated with networked airborne local area networks. This study consisted of two phases. Phase 1 investigated the methodologies for identifying and mitigating potential security risks of onboard networks that could impact safety. Phase 2 investigated techniques for mitigating security risks in the certification environment.

Current FAA safety assurance processes for airborne systems are based on Aerospace Recommended Practice (ARP) 4754, ARP 4761, and Advisory Circulars (AC) (e.g., AC 25.1309-1A and AC 23.1309-1C). FAA software assurance is based on compliance with DO-178B that guides software development processes. Complex electronic hardware design assurance is based on RTCA/DO-254. ARP 4754 extends the DO-178B software assurance process to address the additional safety issues that arise when software is embedded into highly integrated or complex airborne system relationships. Connecting airborne software within network systems represents an extension of the ARP 4754 environment to networked items that share limited common functional relationships with each other. This is because networks connect entities or components of a system into a common networked system regardless of the original functional intent of the system design (e.g., multiple aircraft domains can be connected by a common network system).

Networks are inherently hostile environments because every network user, including both devices (and their software) and humans, is a potential threat to that environment. Networked entities form a fate-sharing relationship with each other because any compromised network entity can theoretically be used to attack other networked entities or their shared network environment. Networked environments and the entities that comprise them need to be protected from three specific classes of threat agents: (1) the corrupted or careless insider, (2) the hostile outsider, and (3) client-side attacks. Because of these dangers, ARP 4754 needs to be extended for networked environments by ensuring network security protection and function/component availability and integrity. This, in turn, implies the need to strategically deploy information assurance security controls within network airborne systems.

Safety and security have therefore become intertwined concepts within networked airborne environments. Security engineering addresses the potential for failure of security controls caused by malicious actions or other means. Safety analysis focuses on the effects of failure modes. The two concepts (safety and security) are therefore directly related through failure effects. A shortcoming of either a safety process or a security process may cause a failure in a respective system safety or security mechanism, with possible safety consequences to the aircraft, depending on the specific consequence of that failure.

Previous studies have sought to address airborne safety and security by correlating DO-178B safety processes with common criteria security processes. This correlation produces necessary but inadequate results. It is inadequate because it lacks mathematical rigor and, therefore, produces ad hoc conclusions. The results are ad hoc because, even when safety and security are correlated, they are nevertheless distinct concepts from each other, addressing very different concerns.

This report states that the primary issue impacting network airborne system safety is how to extend existing ARP 4574, ARP 4761, DO-178B, and DO-254 assurance guidance processes into networked systems and environments in a mathematically viable manner. This study recommends that these processes can be extended into arbitrarily vast network environments in a mathematically viable manner by using the Biba Integrity Model framework. This report maps current DO-178B and ARP 4754 processes into the Biba Integrity Model framework using well-established system security engineering processes to define airborne safety requirements. It applies best current information assurance techniques upon those airborne safety requirements to create a generic airborne network architecture.

Since the Biba Integrity Model is an integrity framework, it carries within itself a natural mechanism for relating safety and security concepts in terms of their respective integrity attributes. Nevertheless, this study recommends that the model be implemented solely within the context of existing FAA safety processes. This results in airborne network systems being organized into networks that operate at specific safety levels (the DO-178B software levels).

There are fortuitous secondary effects from using the Biba Integrity Model to extend current FAA processes into networked environments that stem from it being the direct analog of the Bell-LaPadula Confidentiality Model. The Bell-LaPadula Confidentiality Model forms the framework for confidentiality within U.S. Department of Defense (DoD) information processing. Consequently, the application of the Biba Integrity Model to airborne system assurance processes results in an airborne network architecture that remarkably resembles the emerging DoD network architecture (the global information grid (GIG)), despite their very different underlying goals. Consequently, the generic airborne network architecture identified by this study greatly resembles the DoD's GIG architecture. While military technologies could be used to implement the airborne network architecture, this study recommends the use of civilian Internet protocols deployed as a virtual private network. In addition, the similarities between the Biba Integrity Model and the Bell-LaPadula Confidentiality Models may result in increased synergies between the DoD and FAA certification processes.

Deploying airborne systems into networked environments means that the FAA system safety assessment (ARP 4761), system development (ARP 4754), software assurance (DO-178B), and complex electronic hardware assurance (DO-254) processes need to be extended to address and mitigate network threats. For example, although security is primarily a systems concept involving system issues (e.g., ARP 4754), the Biba Integrity Model relies upon the networked items having integrity attributes that function at a known assurance level (i.e., specific DO-178B software levels). This means that the processes for developing those items for network environments should be extended to address network attack risks. The concept of high-assurance software in networked environments should therefore mean that items and systems will behave in the same manner before, during, and after network attacks; i.e., be immune to potential network-based threats. Exploits in network environments leverage latent software blemishes so that software items are subject to misbehavior, corruption, or compromise, possibly including use as a launching pad to attack other systems and items. Current DO-178B processes do not currently include mechanisms to identify and fix well-known network attack vectors. This study identifies specific additional tests to perform that function. Unfortunately, software testing alone cannot result in high-assurance software. This is because tests only identify the flaws for which the tests are designed to identify—they cannot warrantee the absence of other flaws that were not addressed by the test suite. There is no existing security theory or process that can be leveraged to produce guaranteed high-assurance results for networked environments. This is a very significant certification issue. Until a solution for this problem is found, this study recommends that the FAA ensure that high-assurance software complies with formal models and undergoes a rigorous line-by-line code inspection to demonstrate a lack of bugs that can be hostilely attacked.

1. INTRODUCTION.

This is the final report of a 2-year Federal Aviation Administration (FAA)-funded study to address security and safety issues associated with networked airborne local area networks (LAN). This study consisted of two phases. Phase 1 investigated the methodologies for identifying and mitigating potential security risks of onboard networks that could impact safety. Phase 2 investigated techniques for mitigating security risks in the certification environment.

Individual systems onboard aircraft are designed to meet specific operational, functional, and physical requirements. The safety requirements of a flight-critical avionics system differ from a cabin management system or a passenger in-flight Internet service. If these systems become interconnected, incompatibilities in design assumptions, administrative policies, user interaction, and data security considerations increase their exposure to risk. Actions taken in the context of an open passenger network (whether originating onboard an airplane or from some remote site via a network connection) must be prevented from introducing flight-safety risks to flight-critical systems.

Airborne systems are designed, built, and approved in accordance with airworthiness requirements. Current FAA safety assurance processes for airborne systems are based upon Aerospace Recommended Practice (ARP) 4754 [1], ARP 4761 [2], and Advisory Circulars (AC); e.g., AC 23.1309-1C [3] and AC 25.1309-1A [4]. FAA software assurance is based on compliance with RTCA/DO-178B [5] that guides software development processes. Complex electronic hardware design assurance is based on RTCA/DO-254 [6]. The primary FAA certification standards are the respective regulations, FAA policy, and the ACs.

This study addresses how to extend current FAA processes and certification environment to include networked airborne LANs in a mathematically viable manner. Because of the extensive scope of the current FAA policies and processes, this report addresses this larger issue by specifically explaining how to extend the software assurance subset. Other aspects of FAA policy and processes can be extended into networked environments in a parallel manner (i.e., by leveraging a security model framework, see section 6.2).

DO-178B is one means to secure approval of airborne software. The system safety assessment processes (ARP 4754 and ARP 4761) determine failure conditions of the system and define safety-related requirements as input to the software life cycle processes. DO-178B identifies the software level of a software item based on the potential contribution of the software to failure conditions for the entity in question. Software level refers to the worst-case result of a failure of that software upon aircraft safety in terms of one of five possible failure condition categories. The failure condition categories range from failure conditions that would prevent the aircraft's continued safe flight and landing (catastrophic) to failure conditions that do not affect the operational capability of the aircraft nor increase crew's workload (no effect). Higher software levels, and greater assurance protections, are provided to entities that would have higher safety consequences should they fail. DO-178B also addresses some software design and certification considerations for user-modifiable software, option-selectable software, commercial off-the-shelf (COTS) software, and other software-related issues.

Similar guidance is also provided for airborne electronic hardware (e.g., DO-254). The principals that pertain to networking airborne systems that are addressed in this report directly pertain to all networked system entities, both software and hardware. This report's resulting exemplar network architecture (see section 8.3) is therefore agnostic concerning whether the airborne networked entities are hardware, software, or a combination of both.

Historically, many software entities onboard aircraft are embedded in systems performing specialized mission-related functions. COTS-generic computing devices are increasingly being deployed within the National Airspace System (NAS), and they are occasionally deployed within aircraft as well.[1] John Knight [7] identifies two primary reasons for the increased use of general-purpose computing devices and hardware (e.g., microprocessors, random access memory (RAM), and memory management unit):

1. For enhanced avionics functionality. Specifically, entirely new concepts become possible with the introduction of digital systems. These include modern autopilot technology with greater functionality and flexibility than was possible with historic analog systems, modern full authority digital engine controls, envelope protection systems, and flight deck automation ("glass cockpit").

2. For enhanced safety. Examples include aircraft condition analysis and management systems, structural health monitoring, and automatic alerts of potential runway incursions.

DO-178B and supporting advisory circulars address all phases of the software development process for all software types to ensure that airborne systems are safe. This includes electronic flight bag computing devices [8]. These cumulative systems, processes, and guidelines were historically targeted for aircraft environments whose internal networks are specialized to accomplish specific mission functionality. Aircraft devices are historically connected via specialized and sometimes proprietary data buses to other devices with which they perform associated functions. The nature of these connections is tailored to support their specific communication requirements (e.g., deterministic real time, asynchronous, etc.). Air-to-ground communication similarly occurs via special-purpose communication systems using special-purpose data communication protocols.

Visionaries anticipate forces that could motivate future airborne system designs to replace today's diverse data bus systems within aircraft, including many of their current constraints (e.g., access point limitations, proprietary protocols, labeling, and mitigations such as cyclic redundancy checks), with airplane-appropriate LAN technologies that support standard Internet

[1] COTS software components are very rare critical airborne systems and are discouraged from being deployed in those environments because they cannot comply with DO-178B and other software policy in general.

protocol (IP)-based communications. For example, reference 9 concluded that Ethernet-based LANs could be appropriate to serve as aviation data buses if they use

> "a switched Ethernet topology along with traffic regulation, bandwidth restriction (guarantee and control of bandwidth allocation), and call admission control."

Coupled with the linkage of aircraft systems via a common network system is a growing perception of the desirability to improve and enhance air-to-ground and air-to-air communication systems and processes as well as to more closely integrate airborne systems with NAS systems. For example:

- Integrating multiple data bus systems into onboard LAN(s) is expected to reduce aircraft size, weight, and power (SWAP) overheads, thereby improving aircraft flight performance parameters.

- Next generation aircraft display systems may want to combine map and air traffic data, terrain information, weather radar returns, information on man-made obstacles, and imagery on the airport environment. This would require fusing data from sources that are not currently associated together. It would also necessitate the support of high-bandwidth data communications internally within the aircraft, as well as air-to-ground and within the NAS.

- National Aeronautics and Space Administration Small Aircraft Transportation System (SATS) is investigating mechanisms that would enable small aircraft to fly to and from the 5400 small airports that are not currently being used for reliable public transportation. "A key to implanting SATS is a robust and extremely reliable automated communications system. The system must be capable of passing large amounts of data between aircraft and ground systems as well as between neighboring aircraft in a reliable manner" [10].

- George Donohue, former FAA Associate Administrator of Research and Acquisition, has expressed concerns that the United States'

  > "air transportation network is seriously overloaded in the major cities that support airline hub operations. … This … is leading to a gradual decrease in the US air transportation system safety. … There is a growing consensus over the last 3 years that the capacity of the US National Airspace System is finite and currently approaching critical saturation limits. … Without new technology and operational procedures, we cannot increase capacity without decreasing the systems safety. … Without increased capacity, the increased cost of air transportation will effectively suppress demand (for new aircraft, domestic tourism, international travel, etc.) and have a profound effect on the nation's culture and economy. … System maximum capacity is very sensitive to aircraft final approach spacing. Decreasing aircraft separation in the final approach to a runway from an average of 4 nautical miles between aircraft to 3 nautical miles

> would increase this capacity in the USA [from the current 30 million operations per year] to over 40 million operations per year. … [To accomplish this,] all commercial aircraft will need to have double to triple redundant, collision detection and avoidance systems on the aircraft with professionally trained pilots providing safe aircraft separation. The national air traffic control system should be distributed between ground and airborne systems in such a way that it will be almost immune to single point failures…" [11].

- Arguments that the air traffic management system should become network centric to ultimately achieve the NAS goals. Dennis Buede, John Farr, Robert Powell, and Dinesh Verma define a network centric-system as:

    - "A network of knowledgeable nodes shares a common operating picture and cooperates in a shared common environment.

    - Functional nodes reside in the cognitive, physical, and information domains and communicate with each other and between domains.

    - The heart of the system is the network. Knowledgeable nodes may act autonomously (self-synchronization) with or without a central command and control facility. The US Federal Aviation Administration (FAA) refers to the National Airspace System (NAS), which is made up of more than 18,300 airports, 21 air route traffic control centers (ARTCC), 197 terminal radar approach control (TRACON) facilities, over 460 airport traffic control towers (ATCT), 75 flight service stations, and approximately 4,500 air navigation facilities. The airlines and government employ more than 616,000 active pilots operating over 280,000 commercial, regional, general aviation, and military aircraft. …

      … The current improvements to the NAS focus on safety, accessibility, flexibility, predictability, capacity, efficiency, and security" [12].

- Evolving airborne software systems to similarly support network centric operations promises enhanced, automated aircraft system update procedures and maintenance processes that are not possible with today's federated systems.

The proposed transformation of today's aircraft and airspace system can be compared to the rapid increase in interoperability seen in commercial and military systems. These previous revolutions similarly capitalized on explosive increases in computing power and interconnectivity, as well as rapidly evolving protocols and services. This rapid rate of change, combined with a lack of discipline among commercial vendors who prized functionality over security, unfortunately has resulted in a landscape where secure interconnected systems rarely (if ever) exist.

The anticipated deployment of networked LANs in aircraft is, therefore, not an isolated event. It is a constituent element within the larger network centric evolution of society. The implications of linking aircraft-resident systems over a common data bus needs to be considered within the larger context of the network-centric evolution of air-to-air, air-to-ground, and ground-to-ground communications within the airspace as a whole.

However, with these advantages come risks associated with increased exposure of previously isolated components. Aircraft vendors, operators, and regulators need to understand the impact that interconnected systems may have upon flight safety. Design, test, validation, and verification techniques should consider the impact of unanticipated interactions between previously isolated systems. In addition, the effects of intentional failures caused by malicious software or persons need to be considered. Existing evaluation techniques, where individual systems have been evaluated in isolation, should be updated to address safety concerns introduced by future interconnected systems.

FAA Order 1370.82 "Information Systems Security Program" requires "the FAA must ensure that all information systems are protected from threats to integrity, availability, and confidentiality" [13]. Section 4.1 of this report explains that networks potentially expose software to larger populations of attack threats. As John Knight explains, "unless a system is entirely self contained, any external digital interface represents an opportunity for an adversary to attack the system" [7]. Section 4.4 explains that COTS computing devices, when deployed within networked environments, have an indeterminate number of latent security vulnerabilities that can be attacked and potentially exploited. COTS systems, therefore, have very questionable assurance characteristics in networked environments. Even though aircraft may not deploy COTS software within their airborne LANs, they nevertheless can benefit from the extensive experience gained to date from deploying COTS systems within networks and they may communicate with ground-based networks that widely deploy COTS systems. Airborne software and devices, unless they have been specifically assured for use in networked environments, may or may not manifest similar problems, depending on the number and type of bugs present in networked airborne software. This is because latent security vulnerabilities, when combined with the increased exposure of networked systems, can result in security problems that have direct safety implications. Vulnerabilities include:

- Modification or replacement of authentic aviation software by an alternative variant introduced by an attacker. For example, if an attacker could thwart onboard security procedures to download corrupted software of their own choosing, then a safety hazard can arise if that corrupted software, for example, causes the pilots—and the navigation systems they rely upon—to believe that their current altitude is 2000 feet higher than it actually is.

- Attacks to network system elements that either hinder correct software operation or else modify the reported results of correct software operation. For example, if an attacker takes control of an onboard device and uses it to continuously flood the onboard network with spurious transmissions, a safety hazard may arise should that denial of service attack on the network actually succeed in disrupting latency-sensitive real-time transmissions between distributed avionics components and, by so doing, induce incorrect computation results that affect critical onboard systems.

Protections against these types of safety threats are primarily accomplished by means of deploying effective information assurance (IA) security controls. Safety and security have, therefore, become intertwined concepts within networked airborne environments. Security engineering addresses the potential for failure of security controls caused by malicious actions or other means. Safety analysis focuses on the affects of failure modes. The two concepts (safety and security) are, therefore, directly related through failure effects. A shortcoming of either a safety process or a security process may cause a failure in a respective safety or security mechanism, with possible safety consequences to the aircraft, depending on the specific consequence of that failure.

This study deals with the security and safety issues and acceptance criteria of networked LANs in aircraft. In view of the discussion above, this study cannot merely evaluate the safety and security implications of introducing LANs to aircraft in an isolated manner. Rather, the system-wide consequences associated with these changes must be evaluated. At a minimum, this comprises two distinct but related concepts:

- The security and safety issues, including acceptance criteria, of linking aircraft-resident systems upon a common network infrastructure. This common data bus causes those systems to no longer be physically isolated from each other. Rather, it links them into a common communications system. A direct consequence of this change is that formerly isolated onboard software systems have become theoretically accessible via a common onboard communications system. This common onboard communication system is also connected to the aircraft's air-to-ground communication system.

- As described in the bullets shown on pages 3 and 4, anticipated network-centric changes to the NAS itself results in NAS systems being increasingly redefined into network-centric systems. This includes the possibility of aircraft systems being accessible from expanding regions of the NAS. Should the NAS itself become connected to public data communications systems, such as the Internet, then this would result in the theoretical possibility of aircraft being accessible from systems outside of the NAS. Specifically, should the NAS become connected to the Internet, the theoretical possibility exists of accessing aircraft from any location worldwide.

By combining these two concepts, the net result is the theoretical possibility that specific aircraft-resident systems may become accessible from any location worldwide. This possibility has potentially severe safety assurance implications. However, similar risks result even when no entities using aircraft LANs are involved in communications with entities outside of the aircraft. This is because attackers do not need to access aircraft-resident systems to launch electronic attacks upon aircraft—similar affects can be achieved solely by attacking ground-resident NAS elements. For example,

> "One of the most frightening images of cyber terrorism is a scenario in which terrorists take over the air traffic control system and cause an aircraft to crash or two planes to collide in flight.

> Industry and governments are extremely sensitive even to the appearance of threats and vulnerabilities. A collapse of public confidence in civil aviation safety, and a failure to manage public expectation, may have serious and widespread economic and social consequences." [14]

These affects may be similar to the events following the terrorist attacks on September 11, 2001. Ken Birman has noted that parallel network-centric evolutions are widespread throughout our society. He warns that

> "We're poised to put air-traffic control, banking, military command-and-control, electronic medical records, and other vital systems into the hands of a profoundly insecure, untrustworthy platform cobbled together from complex legacy software components." [15]

The potential scale of harm from a successful electronic attack against elements within a network-centric NAS system, including network-connected aircraft, is huge. Adversaries may be anonymous and range from individual recreational hackers to well-financed criminal enterprises to well-organized, state-led initiatives. To the extent that the NAS builds upon COTS technologies, the technology employed for electronic attacks will be simple, cheap, and widely available. Reliance upon advances in COTS technology creates an extremely fluid threat environment as historic security vulnerabilities are addressed and new vulnerabilities discovered [16]. Fortunately, lessons from industry (civilian) and military security processes and experience can be applied to the aviation industry.

This report, therefore, examines the safety and security issues introduced by networked LANs on aircraft. It seeks to adapt industry's best system security engineering (SSE) practices to identify safety risks caused by aircraft networks. It identifies potential security threats and assesses evaluation criteria. It leverages best current industry and military practices. It proposes specific extensions to ARP 4751 and DO-178B processes to address network security threats and certification issues that arise from networking airborne systems.

2. OBJECTIVES AND APPROACH.

The purpose of this report is to document evaluation criteria that can be used by certification authorities and industry to ensure that onboard networks will not negatively impact aircraft safety. The results will be used by the FAA as input for development of policy, guidance, and regulations.

This work is divided into two phases. The first phase focused on the potential security risks of onboard networks that affect safety and explored issues and solutions to critical questions raised by the aviation industry as manufacturers consider using LANs in aircraft. Initial acceptance criteria for certifying aircraft that use LANs is provided to help evaluators understand the safety and security issues and specific evidence needed to show that proposed designs and countermeasures are sufficient to ensure safety of flight.

This document also includes the study's phase 2 effort. The phase 2 effort, refines the phase 1 results to create specific recommendations to certifiers for the evaluation of the security and safety posture of airplane architectures, which include both onboard and offboard networking capabilities. Phase 2 also attempts to complete the acceptance criteria started in phase 1. It provides guidelines to help evaluators understand the safety and security issues with networks on airplanes and recommends specific evidence needed to ensure that proposed designs and countermeasures are sufficient to ensure safety of flight.

This report seeks to answer questions raised by the potential use of LANs on aircraft, including:

- Are current regulations adequate to address networked airborne security concerns?
- How does security assurance fit into the overall certification process, including ties to safety assessment?
- What should a network security assurance process contain to enable onboard networks to meet Title 14 Code of Federal Aviation Part XX.1309 [Where XX refers to the particular CFR Part (Parts 23, 25, 27, 29, or 33)]?
- How will continued airworthiness be addressed for onboard networks and how will regular maintenance be performed in the certification environment?
- How can it be ensured that the systems connected to the onboard network cannot negatively impact safety?
- What should the process be for updating security protection software?
- How can security breaches be handled?

Consequently, this project

- investigates safety and security issues introduced by using LANs on aircraft
- investigates the potential security risks of an onboard network that could impact safety
- investigates the means for mitigating the security risks in the certification environment
- provides recommendations for assessing safety effects caused by potential security failures
- provides recommendations for certification of LANs on aircraft

2.1 NOTIONAL NETWORKED AIRCRAFT ARCHITECTURE.

Networking aircraft with the NAS is symptomatic of larger societal changes that are arising from the emergence of the worldwide Internet network of networks. The Internet has had a profound impact upon many aspects of modern life. Many businesses have redefined their relationships with other businesses, increasingly basing them upon Internet-oriented electronic commerce technologies. The public has also embraced the Internet, as witnessed by the growing ubiquity of Internet services such as the worldwide web, instant messaging, and electronic mail within popular culture. Perhaps because of this, many aircraft manufacturers are planning to install onboard networks enabling passenger access to the Internet. However, if an aircraft manufacturer opts to have an onboard network that is available to both passengers and avionics equipment (i.e., a shared LAN), aircraft safety and security concerns arise. As previously mentioned in the introduction, parallel, evolutionary changes to the NAS increasingly are being proposed that rely upon greater integration between air-based and ground-based airspace systems. For example, the operational integration of aircraft with the NAS' communication and logistics infrastructure promises dramatic improvements in operational efficiency. Thus, a variety of motivations are increasing the connectivity of aircraft systems to air-based and ground-based network infrastructures.

Current commercial aircraft systems and networks can be grouped in three major categories: closed, private, and public. The closed networks are representative of safety-critical avionics systems; private systems represent airline operational systems, cabin management systems, etc; open systems are represented by public Internet services offered to passengers.

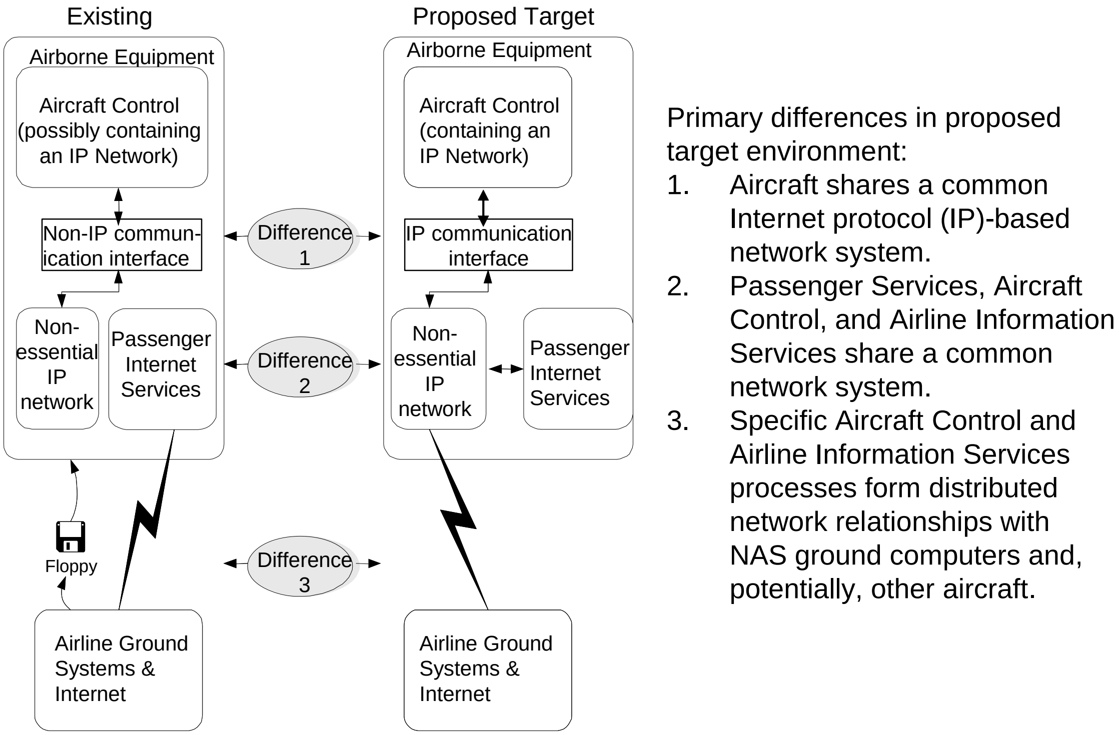
Several changes have been proposed for next generation of aircraft due to the use of local area networking technologies. In response to these proposed changes, some projects have been initiated in which the common onboard network is designed with partitioning protections. The assured robustness of the proposed partitioning is a concern, from both a security and safety perspective. Previous avionics systems have had their own data bus and have not been accessible by nonavionics systems. Security has historically been enforced by a total lack of access (i.e., an air gap) between systems. However, as this paradigm changes to support common networked systems, the safety and security aspects of the onboard network must be addressed by identifying the resulting risks and establishing appropriate controls to mitigate those risks.

Several notional views of the current and future aircraft networked systems have been formulated. Figure 1 shows one of those views. In figure 1, the current (existing) architecture is shown on the left side, a logical picture of the proposed target architecture is in the middle, and a list of changes to achieve the target (future) architecture is enumerated on the right. As previously mentioned, a key feature of the existing architecture is its air gap between airborne functions and passenger Internet services. Thus, there is no way a passenger or entities within the worldwide Internet infrastructure can currently access airborne functions. This proposed target architecture alternative, by contrast, has made three very significant modifications to the current airplane design:

- It has created a network within the aircraft itself that supports communications using the IP protocol family (see section 4.5). This implies that aircraft control entities logically share a common network infrastructure that is also connected to the nonessential IP network. Consequently, aircraft control entities are now theoretically connected to nonessential network entities.

- The passenger Internet services have now been connected to the nonessential IP network. This means that all entities within the aircraft will be theoretically connected within the same network system.

- The nonessential worldwide IP network is now connected to NAS and airline ground systems and the Internet. Specifically, aircraft control elements are shown to be in the same network system that includes the NAS entities with which they may need to communicate as well as more than one billion people worldwide with Internet connectivity today.

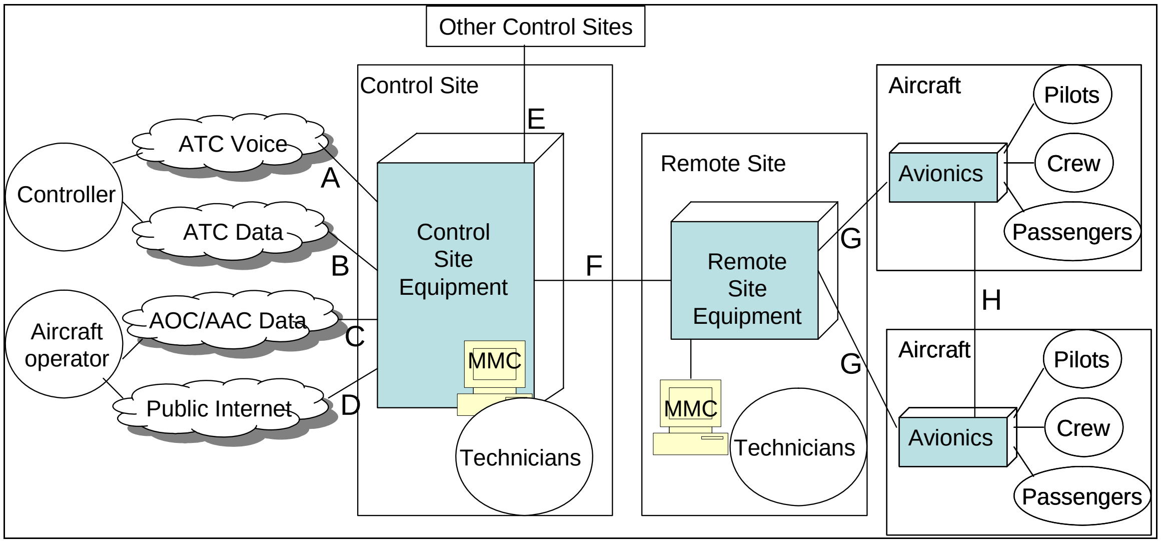

Existing
Proposed Target
Airborne Equipment
Aircraft Control (possibly containing an IP Network)
Non-IP communication interface
Non-essential IP network
Passenger Internet Services
Floppy
Airline Ground Systems & Internet
Difference 1
Difference 2
Difference 3
Airborne Equipment
Aircraft Control (containing an IP Network)
IP communication interface
Non-essential IP network
Passenger Internet Services
Airline Ground Systems & Internet

Primary differences in proposed target environment:

1. Aircraft shares a common Internet protocol (IP)-based network system.
2. Passenger Services, Aircraft Control, and Airline Information Services share a common network system.
3. Specific Aircraft Control and Airline Information Services processes form distributed network relationships with NAS ground computers and, potentially, other aircraft.

Figure 1. Notional Networked Aircraft Architecture

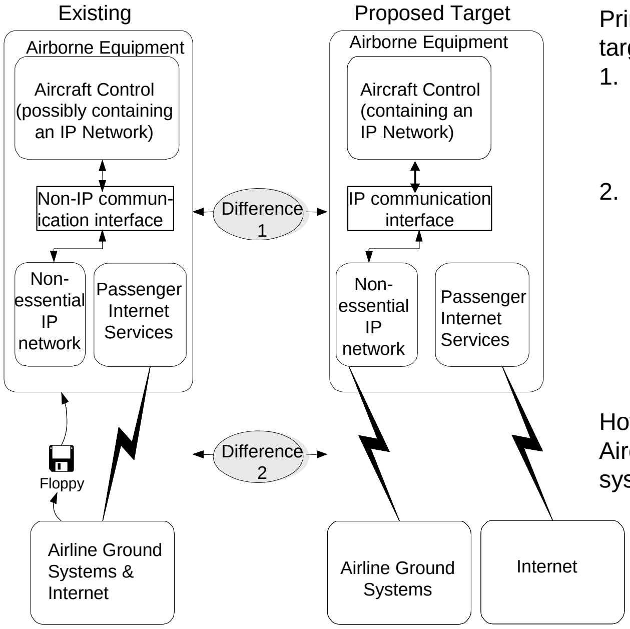
The FAA (ACB-250) community has provided a generic future communication system physical architecture proposal taken directly from reference 17 and is shown in figure 2, which provides greater detail about the network links of the figure 1 target alternative.



AAC = Airline Administration Communication
AOC = Airline Operational Communication
ATC = Air Traffic Control
MMC = Maintenance, monitor, and control

Figure 2. Generic Future Communication System Physical Architecture [17]

Advocates have identified undesirable security implications with the approach shown in figure 1, related to potentially exposing avionics systems to passenger devices and systems. These advocates argue that the advantages achieved by removing the historic security air gap between avionics and passenger systems cannot justify the increased risk to avionic systems posed by that connectivity. Consequently, they have identified an alternative target architecture, which is shown in figure 3.


Existing
Airborne Equipment
Aircraft Control (possibly containing an IP Network)
Non-IP communication interface
Non-essential IP network
Passenger Internet Services
Floppy
Airline Ground Systems & Internet
Difference 1
Difference 2
Proposed Target
Airborne Equipment
Aircraft Control (containing an IP Network)
IP communication interface
Non-essential IP network
Passenger Internet Services
Airline Ground Systems
Internet

Primary differences in proposed target environment:

1. Aircraft Control, and Airline Information Services share a common network system.
2. Specific Aircraft Control and Airline Information Services processes form distributed network relationships with NAS ground computers by using an IP-based air-to-ground link.

However, the air gap between the Aircraft passengers and the avionics systems remains in tact.

Figure 3. Alternative Notional Aircraft Architecture

While there are strong motivations (see section 1) arguing for the migration indicated in either figure 1 or figure 3 to take place, there are security and safety reasons for why the current airborne architecture should not change. Specifically, the existing system is safer and more secure than either of the target systems. Section 4 explains some of the reasons why the proposed changes that form the target designs result in significantly more risk to aircraft. A summary of these reasons include the following:

- The larger the networked community of devices, the larger the potential number of threats to the entities within those networks due to (1) direct or indirect relationships between the networked entities themselves and (2) the increased possibility of hostile attackers being present within the system.

- Due to the emergence of client-side attacks and other threats, the (human) end users of networked resources are now an important part of that network's total security defense posture. Aircraft have limited control over the computer and network behavior of their Internet-connected passengers. NAS employees, as well as aircraft crew members, must be trained to ensure that their own computer behavior does not inadvertently enable attacks against the aircraft. This has direct security implications to both the airborne systems themselves as well as to those networks with which the aircraft connects.

- Entities within networks that are directly or indirectly connected to the Internet may possibly be accessible by attackers located elsewhere in the Internet, despite the presence of intervening security firewalls. This implies that more than one billion people may (theoretically) potentially have access to aircraft.

- The Internet has experienced many well-documented instances of hostile attacks affecting the integrity of computers, networked systems, and the data and services they support.

- COTS computer systems have an indeterminate number of latent bugs that can be attacked.

- COTS computer systems cannot be adequately secured within large network environments, in general, because their security controls cannot be trusted to perform as intended when attacked. They represent possible footholds that attackers can compromise and use as a base from which to attack other networked entities.

- Security vulnerabilities, including those stemming from deploying COTS devices or systems, can theoretically be mitigated. However, the viability of those mitigation approaches are themselves suspect to the extent that they rely upon COTS systems for their implementation. This is because COTS software and systems—and the functionality they support—are not trustworthy when attacked.

- The security viability of current networked systems is partially a direct function of the network and system management (including configuration management) expertise of its administrative personnel.
- The protocols of the IP family can be secured, but their cumulative underlying key management system is ad hoc and complex—with direct configuration and management implications.

- The IP family's network management system, including its underlying simple network management protocol version 3 (SNMPv3) protocol, has questionable security viability when used in network environments that have large numbers of devices built by many different vendors.

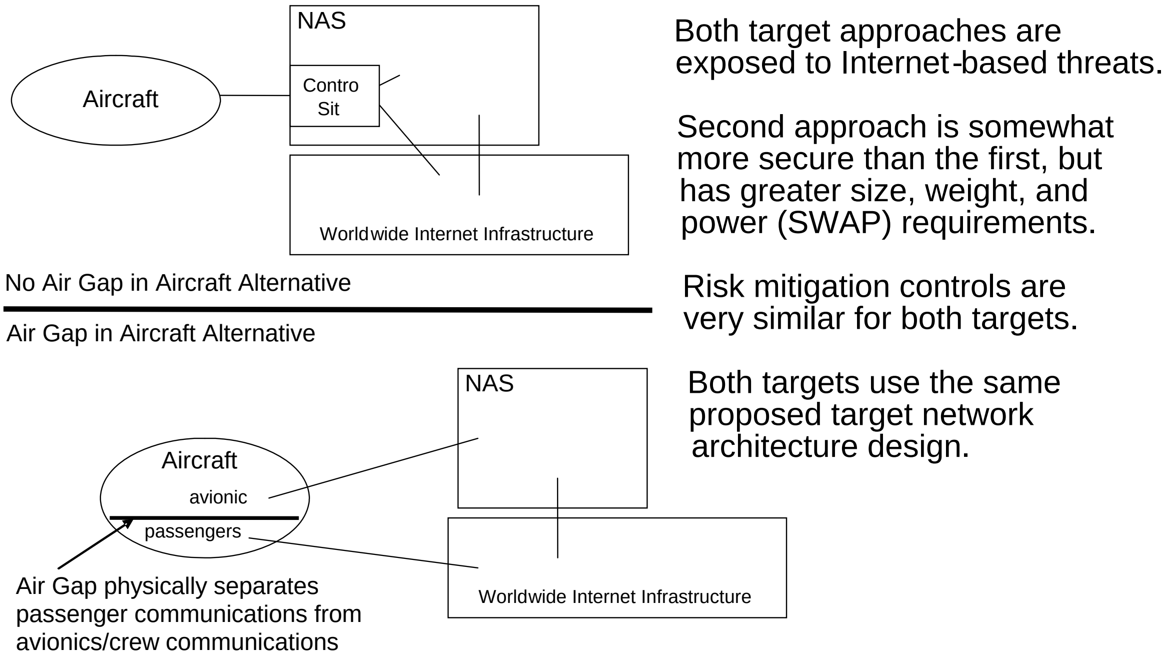
- Whenever different security administrations or technologies are joined together in a cooperative manner (e.g., aircraft and ground systems), it is important and challenging to define the interfaces between the systems in such a way that a diminished security posture for the combined system as a whole does not result.

This report describes assurance mechanisms to mitigate these threats. However, the mitigation system has one key missing element (see section 7.2). Until that element has been successfully addressed, no networked system can currently be guaranteed to be as safe or as secure as the currently deployed non-networked airborne systems.

2.2 WHY BOTH TARGET ALTERNATIVES HAVE SIMILAR SECURITY POSTURES.

It was mentioned that the architecture in figure 3 is more secure than the architecture in figure 1. However, both alternatives have similar security postures, such that the same network solution, which is described in section 8.3, addresses the security and safety requirements for both target alternatives.

Figure 4 shows that both target alternatives similarly expose onboard aircraft systems to possible attacks from the worldwide Internet infrastructure for the reasons explained in section 4.1. While the air gap between passenger and avionics equipment of figure 3 (see bottom of figure 4) protects avionics systems from being directly attacked intra-aircraft from the passenger network, they are still theoretically exposed to remote passenger or Internet attack via the NAS.


NAS
Aircraft
Contro
Sit
Worldwide Internet Infrastructure
No Air Gap in Aircraft Alternative
Air Gap in Aircraft Alternative
NAS
Aircraft
avionic
passengers
Worldwide Internet Infrastructure
Air Gap physically separates passenger communications from avionics/crew communications

Both target approaches are exposed to Internet-based threats.

Second approach is somewhat more secure than the first, but has greater size, weight, and power (SWAP) requirements.

Risk mitigation controls are very similar for both targets.

Both targets use the same proposed target network architecture design.

Figure 4. Both Target Architectures Have Similar Security Profiles

Consequently, the primary advantage of the target approach shown in figure 3 versus the target approach shown in figure 1 is that the figure 3 approach enables the port 80 (i.e., hypertext transfer protocol (HTTP)) overt channel to be closed within the aircraft's perimeter defense firewall (see section 8.3.5), thereby eliminating a vulnerability by which firewall protections can be circumvented. There are also two helpful secondary affects of the figure 3 approach:

- The packet filter design (see section 8.3.4) is simplified. Passenger communications of the figure 3 approach do not traverse avionics networks. Consequently, the avionics network of that approach does not require that the packet filter system protect it by enforcing quality of service (QoS) provisions upon passenger communications to ensure that those communications do not consume too much avionics LAN capacity. Similarly, the packet filter would no longer need to ensure that passengers cannot address the encapsulation gateways (see section 8.3.3) or the cockpit (pilot) network since there would be no connectivity to those systems. However, the figure 3 approach still requires that the packet filter be retained to ensure that the noncockpit crew network cannot send packets to the encapsulating gateways, unless those crew systems could be provided with physical security guarantees that they are never accessible to passengers.

- The high-assurance LAN (see section 8.3.7) is similarly simplified because it no longer must be deployed in a manner to ensure that the passenger network can only access other avionics systems by means of the packet filter. Rather, different physical LAN systems must be used by the passenger system and the rest of the aircraft.

Consequently, the figure 3 approach does not eliminate the need to deploy a packet filter within the aircraft, but it does simplify what that packet filter system does. However, the figure 3 alternative requires that parallel (i.e., distinct) sets of onboard networks and wireless external communications systems be created, one for passengers and one for the other aircraft systems. The figure 3 approach, therefore, has a higher SWAP overhead than the figure 1 approach, by requiring parallel internal (onboard LAN) and external network (radio, satellite, etc.) connectivity, without significantly improving the security profile for the aircraft itself.

3. EXTENDING THE CURRENT FAA CERTIFICATION ENVIRONMENT.

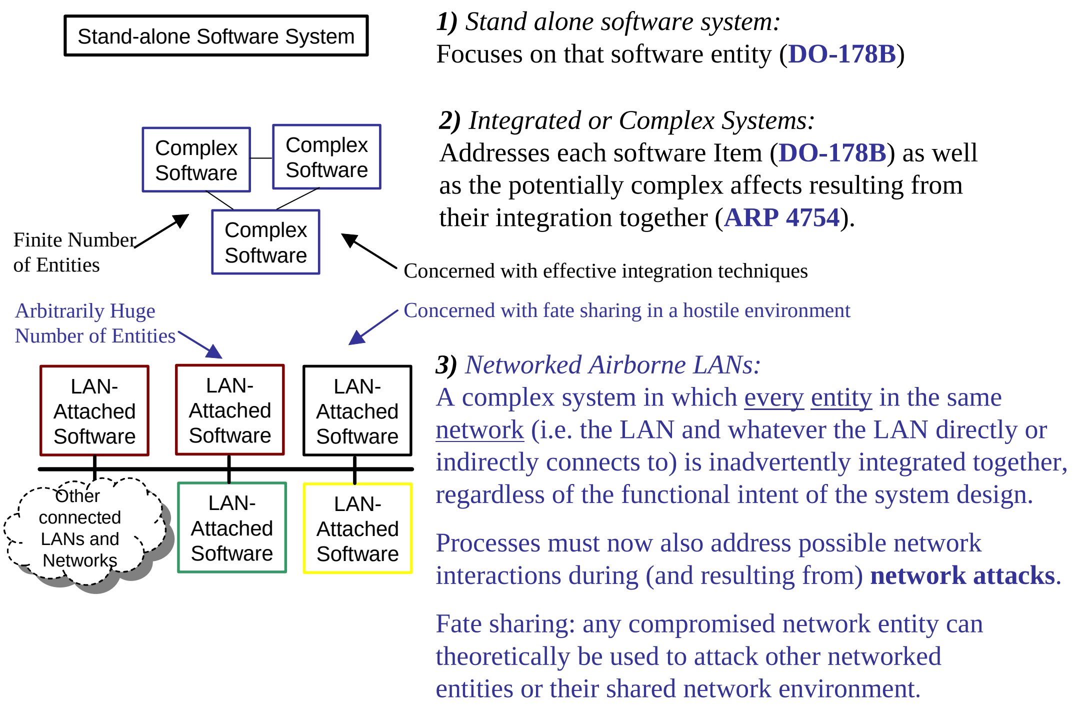
The purpose of this section is to provide an orientation for how this study recommends that the certification environment be extended to handle networked airborne LANs. The specific theory, rationale, and process underlying this study's recommendations are explained in subsequent sections (i.e., sections 6, 7, and 8).

It was previously mentioned that current FAA and civil aviation safety assurance processes for airborne systems are based on ARP 4754, ARP 4761, and the ACs. FAA software assurance is based on compliance with DO-178B, and complex electronic hardware design assurance is based on DO-254. The primary FAA certification standards are the respective regulations, FAA policy, and the ACs. This study addresses how to extend these processes and certification environment to include networked airborne LANs in a mathematically viable manner. Because of the large scope of the current FAA policies and processes, this report addresses this larger task by explaining how to specifically extend its airborne software assurance subset.

Figure 5 shows a simplified and abstracted view of the current FAA software assurance approval process. It shows that airborne software is currently developed and approved primarily according to the guidance and processes described within DO-178B.[2] When individual software items are combined into integrated or complex systems, then additional safety considerations apply, which are documented in ARP 4754. These considerations address integration issues and system vulnerabilities that may arise from system dependencies. ARP 4754 refers to each element within that system as being an item. This same terminology is adopted by this study.

DO-178B builds upon system design concepts such as the AC 25.1309-1A fail safe design concepts, one of which is integrity. Both DO-178B and ARP 4754 (i.e., section 2.2.2 of DO-178B, where it is called the "software level definitions," and Table 3 of ARP 4754) rely upon the same five failure condition categories. Indeed, the same failure condition categories are consistently used within other civil aviation documents as well (e.g., Table 2-1 of DO-254 and Table 1 of ARP 4761). Different development processes are applied to items classified in different failure condition categories so that items classified in the more severe safety failure conditions are developed by more extensive processes that produce higher assurance results. For software items, this is reflected in the DO-178B software level definitions. For this reason, this report refers to DO-178B software levels as reflecting safety assurance levels.

[2] There are other applicable policies and guidance in addition to DO-178B that can also be applied. Please recall that this figure is a simplified abstraction.


Stand-alone Software System
Complex Software
Complex Software
Complex Software
Finite Number of Entities
Concerned with effective integration techniques
Arbitrarily Huge Number of Entities
Concerned with fate sharing in a hostile environment
LAN-Attached Software
LAN-Attached Software
LAN-Attached Software
Other connected LANs and Networks
LAN-Attached Software
LAN-Attached Software

1) *Stand alone software system:*
Focuses on that software entity (**DO-178B**)

2) *Integrated or Complex Systems:*
Addresses each software Item (**DO-178B**) as well as the potentially complex affects resulting from their integration together (**ARP 4754**).

3) *Networked Airborne LANs:*
A complex system in which every entity in the same network (i.e. the LAN and whatever the LAN directly or indirectly connects to) is inadvertently integrated together, regardless of the functional intent of the system design.

Processes must now also address possible network interactions during (and resulting from) **network attacks**.

Fate sharing: any compromised network entity can theoretically be used to attack other networked entities or their shared network environment.

Figure 5. Three Different Software Certification Environments

ARP 4754 is directly concerned with architectural considerations that pertain to highly integrated or complex airborne systems:

> "System architectural features, such as redundancy, monitoring, or partitioning, may be used to eliminate or contain the degree to which an item contributes to a specific failure condition. System architecture may reduce the complexity of the various items and their interfaces and thereby allow simplification or reduction of the necessary assurance activity. If architectural means are employed in a manner that permits a lower assurance level for an item within the architecture, substantiation of that architecture design should be carried out at the assurance level appropriate to the top-level hazard. …
>
> It should be noted that architectural dissimilarity impacts both integrity and availability. Since an increase in integrity may be associated with a reduction in availability, and vice-versa, the specific application should be analyzed from both perspectives to ensure its suitability." (Quoted from Section 5.4.1, pages 25 and 26 of reference 1.)

Because ARP 4754 addresses possible system vulnerabilities that derived from creating functional system relationships between items, to a certain degree, it can be characterized as being directly concerned with effective integration techniques between those system items. It, therefore, presumes that the regulator can correctly identify the items that comprise a system as well as their mutual relationships together.

Aircraft network security is a systems issue. System development (ARP 4754), in conjunction with the system safety assessment process (ARP 4761), is responsible for defining network accesses, vulnerabilities, detection, and protection requirements. Some of the vulnerabilities will be mitigated by limiting and controlling access by using hardware and software capabilities. Some identified vulnerabilities will be mitigated by monitoring and detection capabilities. The security protection should be defined by the system and then by appropriate system requirements allocated to hardware, software, and hybrids. This study assumes that best current IA practice will be followed, including deployment of traditional IA security controls when appropriate. After implementation, these protections, mitigations, and monitoring will also likely be verified and validated at the system level, as well. Consequently, aircraft network security is an ARP 4754 issue.

However, approving networked systems in some ways should be recognized as being a significant extension to ARP 4754. Networked systems differ from the current ARP 4754 environment in several significant ways. Networked elements are systems that include all of the networks and their constituent elements and users to which the network is directly or indirectly attached. Networks are therefore arbitrarily huge, and the many interrelationships of the system items are often too subtle to discern. Networks are inherently complex systems in which every item in the network is inadvertently integrated, regardless of whether those items share any common functional goal. Approval of networked entities must now also address possible network interactions that occur during, and result from, network attacks. The various networked elements potentially have a fate sharing relationship with each other, because any compromised network entity theoretically can be used to attack other networked items or their shared network environment.

Therefore, networked airborne LAN environments are inherently “highly integrated or complex aircraft systems,” with attributes that extend the complex relationships for which ARP 4754 was created. Section 4 and appendix A will introduce some of the risks that characterize networked systems and underlie the following observations:

- In networked environments, ARP 4754 needs to be extended to consider each item within the LAN to be integrated, even if that item has no functional relationship with anything else. For example,

    - If the LAN experiences a successful denial of service (DoS) attack, then each networked item in that LAN may potentially be unable to fulfill its function. Therefore, ARP 4754 must be extended in networked environments to ensure availability.

    - If an item in the LAN becomes hostilely compromised by an attacker, then it potentially can be used by that attacker to attack the network itself or other items on the LAN. Therefore, ARP 4754 must be extended in networked environments to address LAN and item integrity. To ensure LAN and item integrity, ARP 4754 needs to be extended to require verifiably secure software installation procedures as well as mechanisms to ensure the continued integrity of deployed items and systems.

- If airborne LANs are connected into networks, then the cumulative network system has similarly become integrated and existing safety processes need to become extended to each system and item within that larger networked system if they are to remain viable, even if any component element within the larger system never itself becomes airborne.

- If the network has both device and human users, then ARP 4754 must also become extended to also pertain to humans. Every human or device with access to that network is a potential threat to that network and may potentially initiate attacks against the network itself, the LANs, or subnetworks that comprise that network, of the items located within that network. If the network is directly or indirectly connected to the Internet, then there are theoretically more than one billion humans with potential access to that airborne LAN—despite the presence of intermediate firewalls. This means that mechanisms are needed within networked systems so that human behavior cannot deprecate historic DO-178B and ARP 4754 safety assurances.

This study identifies mechanisms by which ARP 4754 can be extended to address networking requirements by strategically introducing integrity and availability security controls. It does this by building upon a mathematically based security model that can extend ARP 4754 concepts into arbitrarily vast and complex network systems. This study uses proven SSE processes to combine these concepts and controls into the exemplar network architecture (see section 8).

This study is also similarly concerned with extending DO-178B so that highly assured software items within networked environments can be developed and assured to mitigate known network risks. The concept of highly assured software in networked environments explicitly means that the software can be trusted to behave in the same fashion before, during, and after attacks—something that current DO-178B processes cannot ensure because they do not explicitly address network attack threats. Consequently, current DO-178B software in networked environments may behave in an indeterminate manner during or after attacks if latent bugs within the software itself are successfully attacked by exploits that violate its integrity. Such software is a potential threat to its deployment environment. It is potentially subject to misbehavior, corruption, or compromise, potentially including becoming used as a launching pad to attack other systems and items. This issue is addressed in section 7.2.

This study does not provide suggestions for the process by which the FAA or the worldwide civil aviation community may choose to respond to these recommendations. It may be that the relevant experts may conclude that the revision of ARP 4754 and DO-178B that this study presumes is an unacceptably long term or unreliable approach for ensuring that network security concerns are adequately addressed in a timely manner. If so, then it is conceivable that the FAA may decide to produce an airborne network system security assurance policy in the near term. If this is done, then if SAE and RTCA/EUROCAE subsequently decide to address this issue in the future, then that initial policy could be revised to reflect their proposal.

4. NETWORK RISKS.

This section specifically discusses network security issues that are directly relevant to airborne safety and security within networked environments. Two papers authored by Daniel Mehan [18 and 19], the former FAA Chief Information Officer (CIO), provide an important context for the material discussed in this section. These papers describe the cumulative processes needed to create a secure and safe NAS environment. These processes are partially summarized in figure 6. The majority of the topics covered in this section directly focus on network system aspects of "Cyber Hardening of System and Network Elements," shown in figure 6, of a total protection system. Many other aspects of NAS safety and security are not addressed within this section and are only briefly addressed by this study.

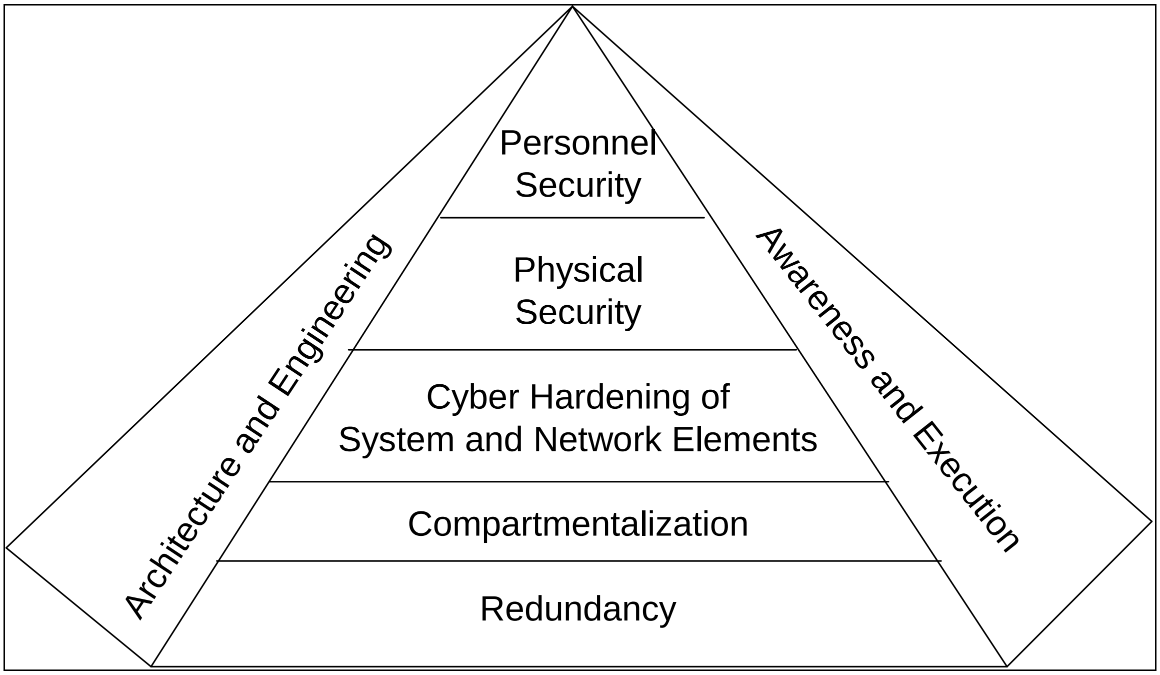


Figure 6. The FAA Five Layers of System Protection [18]

Reference 20 also provides very helpful concepts and background information for the aircraft network safety and security topics presented in this section and appendix A.

It is important to recognize that network threats are always evolving. An indeterminate number of network threats exist. These network threats can mutate quite rapidly. For example, the following is an example of a current network threat that did not exist the year before this study was written.

> "File encrypting Trojans are becoming so complex that security companies could soon be powerless to reverse their effects, a new report from Kaspersky Lab said.

> The report notes the rapid evolution of the public key encryption used by one family of crypto malware, Gpcode, which went from using 56-bit to 660-bit RSA [encryption key] in a matter of weeks.
>
> Commonly termed 'ransomware', Trojans that encrypt data files on a user's [personal computer] PC before demanding a payment in return for supplying the key to unlock the files, have come from nowhere in recent months to become a measurable problem.
>
> At the time of its discovery in June [2006] Gpcode.ag – which used a formidable 660-bit key – Kaspersky described the process required to decrypt such a key as equivalent to setting a 2.2 GHz PC to work for thirty years." (Quoted from reference 21.)

Defenses that require a response to each new threat instance as it appears are both expensive and of questionable efficacy. The best mechanism to effectively constrain the impact of these threats over time is to create a very solid safety and security foundation for both the NAS and aircraft.

Figures 7 and 8 were created by David Robinson [22] to indicate the possible threat affects that can occur when networked aircraft becomes successfully attacked. These types of threats cannot occur for aircraft whose aviation systems are not attached to network systems.

> "These new generation aircraft [e.g., B787, A370, A350, BY-1] will include a new aircraft data network design which will introduce new cyber security vulnerabilities to the aircraft." [22]

Unless properly mitigated, any networked aviation system is potentially subject to the following range of impacts should they be successfully attacked.

General Threat Identifiers	Aircraft Data Network Threats	Aviation Infrastructure Mission and Operational Impact
FAILURE	Safe state of the aircraft system could be compromised in the event of a security penetration	Access to the flight controls by unauthorized individuals affecting safety
DENIAL	Aircraft system resources exhausted due to denial of service attack, system error, malicious actions	Critical services disrupted by system overload or traffic jamming
Access Control	Individual other than an authorized user may gain access to the aircraft system via phantom controller, masquerade or spoofing system error or an attack for malicious purposes.	Unauthorized Access
Passive Attack	Snooping or eavesdropping compromising security (misdirection). Design Flaws may lead to back door access.	Unauthorized corruption or destruction of data causing unsafe flight conditions.

Figure 7. Network Threat Mission and Operational Impact [22]

Threat Targets	Threat Effects
Aircraft Operation	Serious degradation or loss of mission capability, airline is not able to perform its primary function
Assets	Major damage to airline assets
Financial	Major financial loss
Human	Serious or catastrophic physical harm to individuals
Public Perception	Total loss of confidence in air traffic by passengers, disclosure of security information

Figure 8. Airborne Network Threat Targets [22]

The subsequent sections describe technical mechanisms to mitigate these risks.

4.1 DIFFERENT UNIVERSES: STAND-ALONE VERSUS NETWORKED.

It is commonly recognized that the safety and security assurance properties of stand-alone systems are much more easily ascertained than the assurance of systems within networked environments. This difference is primarily due to the fact that the assurance of stand-alone entities is a function of the inherent design of that system itself. These include the repertoire of issues currently considered by DO-178B such as hardware and software design, input-output, direct memory access, interrupt and interrupt processing, design and development process controls, operating system (OS) issues, and security modes. The assurance of networked systems, by contrast, is a function of not only that system's own internal design and processes, but also the implications of the effects to its design and operation caused by the other elements within the total system as a whole. As Joel Knight has observed:

> "Unless a system is entirely self contained, any external digital interface represents an opportunity for an adversary to attack the system. It is not necessary for an adversary to have physical access. Of necessity many systems will communicate by radio, and digital radio links present significant opportunities for unauthorized access" [7].

A great many issues partially determine the susceptibility of any networked item to possible attacks:

- To what extent is an entity manageable? If it is manageable, how secure is the identity, authentication, authorization, and access control processes imposed upon administrative personnel and processes (e.g., separation of duties with least privilege)?

- To what extent is the entity configurable? If it is configurable, what controls ensure that it is configured correctly?

- How confident (e.g., assurance level) is the designer, certifier, and designated approving authority (DAA) that there is a total absence of latent software bugs that can be attacked by hostile attackers to create safety threatening affects?

- How impervious is the implementation to attacks originating from other devices, including how dependent is the implementation upon inherent network availability or security attributes?

- How dependent is the entity upon other distributed components? Can their misbehavior result in safety threatening scenarios?

- What is the relative security and integrity assurance of the data communications protocols and underlying network media (including networking devices) used within that network infrastructure?

The potential interaction of these networked elements is complex. The possible complexity of these interactions is a partial function of the number of elements within the total system and the number of possible interaction mechanisms. Some possible interactions can be unintended and subtle.

For example, a system can be assured as being safe in a controlled environment using DO-178B processes. However, is this assurance still viable if that system is transplanted into a highly networked environment where unforeseen processes may try to influence it in unanticipated ways? Will a real-time system perform adequately in environments where it is continually being accessed by a rogue process? Even if the rogue process fails all authentication and authorization attempts, can it still consume enough central processing unit (CPU) or network capacity resources so that the required real-time interactions of that system with its legitimate peers are detrimentally impacted?

A basic attribute of network environments is that risks to elements within that system increase in direct relationship to the network's population size. The larger the community of networked devices, the greater the possibility that at least one of those devices has been constructed with latent bugs that can be leveraged to compromise that device to directly or indirectly attack other parts of the system. Also, the larger the community of humans that can access elements within the total network system, the greater the possibility that at least one of those humans will exploit bugs either intentionally (maliciously) or accidentally. Hostile electronic attacks may be conducted by both the corrupted insider (e.g., insider threat) as well as by unauthorized personnel who have leveraged system or process blemishes to gain unauthorized (remote) entry into the system. It can also occur by means of accidental mistakes made by authorized personnel.

Widely used COTS network equipment, such as Internet technologies, is more easily assembled into large network systems than less popular communications technologies. For example, the Aeronautical Telecommunications Network (ATN), which is used for air traffic management systems today, is built using open system interconnect (OSI) protocols. OSI protocols are rarely deployed today except within specialized niche environments. Because of this, it is comparatively difficult to link ATN systems with other networks to create large network communities. IP systems, by contrast, are ubiquitously deployed today. Because of this, it is comparatively easy to link together IP-based systems with other networks to create large network environments. A key point to recognize is that just because an IP-based system is not connected to a large network environment today, does not mean that it cannot easily be connected into a large networked environment tomorrow, perhaps inadvertently. For example, inadvertent exposure of allegedly stand-alone (i.e., physically isolated via an air gap) IP networks to remote Internet-based attacks have occurred many times in real life by means of inadequately secured modems located within those allegedly isolated networks.

Widely deployed public networks have larger populations of users than small private networks. The more people within the networking community, the greater the probability that one or more of them may pose an attack risk to the elements within the system. The larger the cumulative number of users within any aspect of the network, the greater the possibility is that a user may purposefully or accidentally exploit those weaknesses in a detrimental manner.

The inclusion of the words "aspect of the network" in the previous sentence is a reference to a technical point that is partially explained within appendix A. That point is that in large network-of-network systems, such as the worldwide Internet, network access control defenses are established between discrete network administrative entities by means of security firewalls [23]. Firewall technologies have significantly improved over time. Unfortunately, so has the sophistication of attacks against them. A class of exploits[3] exist that may possibly circumvent the access control protections of firewall systems. Should these attacks succeed, then those attackers could access network systems where they are not authorized.

[3] e.g., fragmentation attacks, time-based attacks, HTTP-based (Port 80) attacks, and other emerging exploits.

Specifically, most networks implement firewall policies that permit remote access into the autonomous systems (AS) they protect through a port 80 (i.e., HTTP) overt channel. Consequently, many sophisticated attacks explicitly leverage the policy weakness that enables this overt channel to penetrate firewall systems. Only a small percentage of currently deployed networks today have closed port 80. Even when administrative policy permits this vulnerability to be closed, the efficacy of correctly configured firewalls using the very best technology can be circumvented by client-side attacks (see section 4.2) or improper configuration of other system elements (e.g., modems). Also, firewalls that are deployed in SWAP-constrained environments (e.g., aircraft) are often susceptible to a range of modern attacks (e.g., fragmentation attacks, time-based attacks) because they may not contain the necessary resources (e.g., CPU or RAM) to handle those attack vectors. Consequently, firewall protections can potentially be circumvented. Firewalls, therefore, need to be part of a larger defense-in-depth system (see section 5.1), which needs to provide redundant protections (e.g., virtual private networks (VPN), see section 5.6)) to supplement the firewall in case its protections are circumvented.

In view of this potential danger, the number of people that can access a network should not be equated to the number of people that are authorized to access that network. Rather, it should be considered to be the total number of people that can access any part of the larger network system in which that network is a part. This explicitly includes users that are solely authorized to access another network to which one's own network is only indirectly connected. Consequently, if airplanes are even indirectly connected to the Internet, then theoretically, there are over one billion people that can potentially access entities within that airplane.

4.2 INTERNAL, EXTERNAL, AND CLIENT-SIDE ATTACKS.

Because networked systems traditionally use perimeter defense mechanisms (security firewalls) to limit access to internal network resources, a distinction has been created between insiders and outsiders. An insider is an individual who is authenticated and authorized to use internal network resources regardless of whether or not they are physically located geographically in the same location as the networked resource. Outsiders are not authorized to have such access.

A large percentage of security controls have historically been centered on repelling security attacks from outsiders. This reflects the fact that insiders usually undergo scrutiny to obtain their authorizations. However, higher assurance environments need to consider the possible threats stemming from corrupted insiders (i.e., the insider threat). These environments need to deploy controls so that the activities of all authorized users inside the network are restricted in terms of separation of duties with least privilege.

Unfortunately, an entirely new class of attack, the client-side attack, has become increasingly popular and dangerous. Client-side attacks include inadvertent exposure to hostile e-mail attachments or accesses to malicious web pages containing executables or scripts that allow arbitrary code to run. In both cases, the attacker leverages latent security vulnerabilities within the user's web browser or e-mail client.

> "With the rise of client-side attacks, a flaw emerges in the old [security] model; despite avoiding a direct connection to the outside, users might still be attacked by the very services that they've requested." [24]
>
> "A new attack vector has been created in which users are transformed into a platform to attack internal resources without their consent or even their awareness. Users are no longer passive participants in the security model; they've become the very service by which entrance is gained into the protected interior of the network." [16]

There are many published examples of successful client-side attacks, including the following:

> "The Oregon Department of Revenue has been contacting some 2,300 taxpayers this week to notify them that their names, addresses or Social Security numbers may have been stolen by a Trojan horse program downloaded accidentally by a former worker who was surfing pornographic sites while at work in January [2006].
>
> An investigation by agency security personnel and the Oregon State Police found that the malicious program was designed to capture keystrokes on the former employee's computer … The employee was an entry-level worker who was assigned to enter taxpayer name and address changes, as well as some social security numbers. 'We know that the information that the Trojan gathered up was transmitted outside of the agency' to an unrelated Web site. The incident is still under investigation." [25]

Therefore, attacks against networked entities may occur from outsiders, from corrupted insiders, as well as from client-side attacks (see figure 9). The effect of outsider attacks is to emphasize perimeter defense protections (e.g., firewalls, VPNs). The effect of corrupted insiders is that network security is no longer primarily a function of establishing adequate perimeter defense controls; it now must also include viable access control within the network itself. The effect of client-side attacks is that network security is no longer solely a function of the total control protections established on devices within the network. It is now also reliant upon the appropriate activities of every human using those network resources. While filtering services located at the perimeter, defense firewalls can and do combat client-side attacks; however, new attacks are continually being devised that perimeter defense filtering systems must be updated to identify and eliminate. Consequently, there is often a vulnerability window between when a new attack type has been devised and when the protections against that new attack have been deployed. For this reason, defense against client-side attacks heavily relies upon end-user education—and can be circumvented by end-user mistakes.

- **Corrupted or Careless Insider**
  - Are authorized to access the network
  - E.g., NAS personnel, aircraft personnel or passengers, local systems
- **Hostile Outsider**
  - Are not authorized to access the network
  - May be located on “the Internet”
- **Client-side Attacks**
  - Malicious software lurking in “neutral” environments (e.g., email, web sites, other)
  - The historic distinction between “data” and “code” is vanishing
  - NAS personnel, aircraft personnel, and aircraft passengers may be duped into inadvertently executing, and thereby introducing, malicious software into the network
  - Network users therefore have become an integral element of a network’s security defenses

Figure 9. Threat Agents in a Networked Environment

This topic will resume in section 4.4 when the implications of mixing embedded systems and generic OSs within the same network will be discussed. Before that discussion can occur, however, it is first necessary to discuss the vulnerabilities that exist within a networked environment.

4.3 COMMERICIAL OFF-THE-SHELF VULNERABILITIES IN A NETWORKED ENVIRONMENT.

While this section specifically addresses well-known COTS vulnerabilities in networked environments, similar problems may or may not exist within embedded avionics systems, depending upon whether latent bugs exist within those systems that can be exploited by network attacks.

Lance Spitzner has gathered together the following statistics, which provide partial evidence that the worldwide Internet infrastructure is a very dangerous place:

- “At the end of the year 2000, the life expectancy of a default installation of Red Hat 6, a commonly used version of Linux [a computer OS], was less than 72 hours.

- One of the fastest recorded times a honeypot [i.e., a device deployed in order to study the behavior of electronic attackers] was compromised was 15 minutes. This means that within 15 minutes of being connected to the Internet, the system was found, probed, attacked, and successfully exploited by an attacker. The record for capturing a worm was under 90 seconds.

- During an 11-month period (April 2000-March 2001), there was a 100 percent increase in unique scans and an almost 900 percent increase in Intrusion Detection Alerts, based on Snort [an Intrusion Detection utility].

- In the beginning of 2002, a home network was scanned on average by 31 different systems a day." [26]

This list can be supplemented by many other data points including:

- "The most virulent [computer] virus to date infected several million machines in about 20 minutes…." [15]

- "When we put this [honeypot] machine online it was, on average, hit by a potential security assault every 15 minutes. None of these attacks were solicited, merely putting the machine online was enough to attract them. The fastest an attack struck was mere seconds and it was never longer than 15 minutes before the honeypot logged an attempt to subvert it. …

- At least once an hour, on average, the BBC honeypot was hit by an attack that could leave an unprotected machine unusable or turn it into a platform for attacking other PCs. …

- By using carefully crafted packets of data, attackers hope to make the PC run commands that hand control of it to someone else. Via this route many malicious hackers recruit machines for use in what is known as a botnet. This is simply a large number of hijacked machines under the remote control of a malicious hacker." [27]

- "IronPort recently published a report showing that Trojan horses and system monitors – two of the most serious types of malware – infect one out of every 14 corporate PCs. That means that in an organization of 1,000 desktop PCs, there is an average of 70 computers that represent a major security risk. … Dwarfing Trojans and system monitors are less serious types of malware, such as adware and tracking cookies, which infect 48% and 77% of PCs, respectively." [28]

- "The number of new [COTS] software security vulnerabilities identified by security experts, hackers and others during the first eight months of this year [2006] has already exceeded the total recorded for all of 2005, according to Internet Security Systems.

  Vulnerabilities through September have reached 5,300, leaping past the 5,195 discovered for all of 2005, says Gunter Ollmann, director of the X-Force research group at ISS. 'Eight hundred seventy-one were found to affect Microsoft operating systems, while 701 vulnerabilities were only found to affect Unix operating system,' Ollmann says. But many vulnerabilities cross platform

> boundaries to affect them all, including Linux. About 3,219 vulnerabilities fall into that realm, Ollmann notes.
>
> ISS ranks vulnerabilities as critical, high, medium and low. Of the 5,300 vulnerabilities recorded for 2006 so far, 0.4 percent were deemed critical (could be used to form a prolific automated worm); 16.6 percent were deemed high (could be exploited to gain control of the host running the software); 63 percent were medium (could be used to access files or escalate privileges); and 20 percent were low (vulnerabilities that leak information or would allow a denial-of-service attack). …
>
> 'Of the 5,300 vulnerabilities …, 87.6 percent could be exploited remotely; 10.8 percent could be exploited from the local host only; and 1.6 percent could be exploited remotely and local.'" [29]

The Computer Emergency Response Team[4] (CERT) coordination center keeps a monotonically increasing list of reported Internet-related security incidents dating from 1988 to 2003 inclusive.[5] These statistics show that there was more than a 100 percent increase in reported security incidents in 2001, increasing from 21,756 in 2000 to 52,658 in 2001. The most recent incidents were publicly disclosed in 2003, which had 137,529 different reported security incidents. As the CERT notes, "an incident may involve one site or hundreds (or even thousands) of sites. Also, some incidents may involve ongoing activity for long periods of time." [30] The CERT ceased reporting the number of security incidents after 2003 because: "Given the widespread use of automated attack tools, attacks against Internet-connected systems have become so commonplace that counts of the numbers of incidents reported provide little information with regard to assessing the scope and impact of attacks. Therefore, as of 2004, we will no longer publish the number of incidents reported." [30]

An example of an undisclosed incident occurring since 2003 is the following:

> "Chinese hackers launched a major attack on the U.K. Parliament earlier this month, the government's e-mail filtering company, MessageLabs Ltd., has confirmed.
>
> The attack, which occurred on Jan. 2 [2006], attempted to exploit the Windows Metafile (WMF) vulnerability to hijack the PCs of more than 70 named individuals, including researchers, secretaries and members of Parliament (MP) themselves.
>
> E-mails with an attachment that contained the WMF-exploiting Setabortproc Trojan horse were sent to staffers. Anyone opening this attachment would have enabled attackers to browse files, and possibly install a key logging program to

[4] CERT; see http://www.cert.org
[5] See http://www.cert.org/stats/cert_stats.html

> attempt the theft of passwords. None of the e-mails got through to the intended targets, MessageLabs said, but the U.K. authorities were alerted." [31]

Network attacks range in severity and purpose, which include:

- Learning about the target environment to discern which entity to attack, using which attack tools (see appendix A, section A.1). This is known as fingerprinting and consists of network reconnaissance, mapping, and target acquisition activities.

- Attempting to compromise (i.e., takeover) one or more devices within the target network (see appendix A, section A.2). Once a device has been successfully cracked (i.e., hostilely taken over by an attacker), then the attacker can leverage that device to attack other entities within the network.

- Attempting to attack the network distribution system itself (see appendix A, section A.3). This is often accomplished by availability attacks such as DoS attacks.

- Attempting to attack the data that traverses the network (see appendix A, section A.4). This consists of integrity and confidentiality attacks.

All entities within a network are potentially subject to electronic attack. Entities include the devices and software present within the network, the (physical) communications links, and the communications protocols used within the network. Figure 10 shows a network deployment example. The figure shows that there are three types of devices that can be present within an IP network:

- Hosts (e.g., computers, which are known in OSI terminology as end-systems) are the source and/or sink of end-user communications.

- Routers (known in OSI terminology as the network layer intermediate system element) perform IP forwarding of communications between network elements.[6]

- Middleboxes are defined by Request for Comment (RFC) 3234 as "any intermediary box performing functions apart from [the] normal, standard functions of an IP router on the data path between the source host and destination host." Figure 10 shows three different examples of middleboxes:

  - Network Address Translator (NAT)—a device that dynamically assigns a globally unique IP address (without the hosts' knowledge) to hosts that do not have one.

  - Protocol Translation Gateway—a device that translates communication protocols between dissimilar protocol systems (e.g., mapping between IP and OSI (e.g., ATN) networks).

[6] See the IP Topology Hierarchy section below (section 5.3) for a description of network hierarchy elements.

- Firewall—a device or series of devices that provide security perimeter defense (access control) protections to networks.

Note: Internet Engineering Task Force (IETF) Request for Comment (RFC) documents are not included in Section 12, References, of this report because of their electronic availability. All IETF RFCs are found at http://www.ietf.org/rfc/rfc####.txt, where #### is their RFC number. For example, the complete text for RFC 3234 in the previous paragraph is found by inserting "3234" into the above URL template to form http://www.ietf.org/rfc/rfc3234.txt. A current list of IETF RFCs is kept at http://www.ietf.org/iesg/1rfc_index.txt. The list of currently active IETF working groups is found at http://www.ietf.org/html.charters/wg-dir.html and current Internet draft (I-D) documents are found at http://www.ietf.org/ID.html.

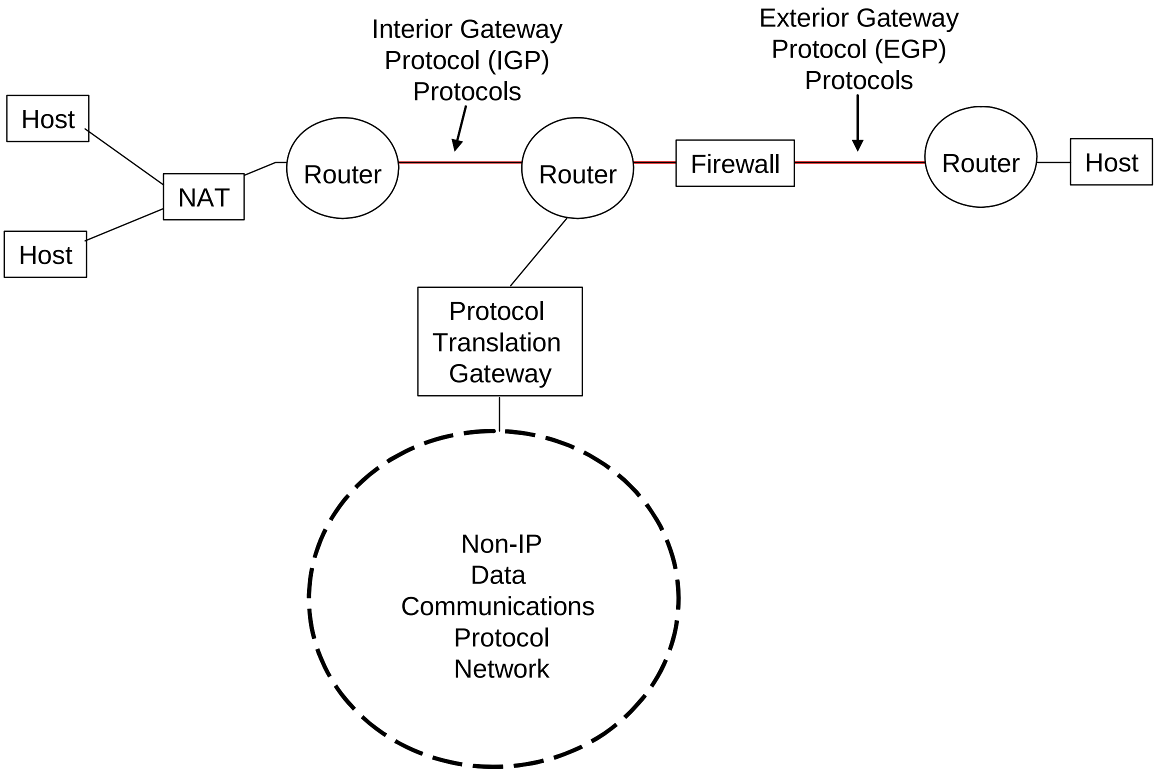


Figure 10. A Sample Deployment

All three of these device types are subject to attack. The effects of a successful attack vary depending on the role of the compromised device (i.e., host, router, or middlebox).

In addition, the communications protocols exchanged between devices may be attacked, either as a mechanism to attack a specific device or else to attack the network system itself. Each of the device types leverage protocol systems to communicate together. Of these systems, the protocols used between hosts and the protocols used between routers are the best known.

IP networks are organized in terms of ASs, which are the unit of policy (e.g., security policy, QoS policy) within IP networks (see section 5.3). The router-to-router protocols of IP networks are subdivided into two distinct systems:

- An interior gateway protocol (IGP) is used between routers within a common AS. Example IGP protocols in IP systems include OSPF (see RFC 2328) and IS-IS (see RFC 1195).
- An exterior gateway protocol (EGP) is used between routers located in different ASs from each other. The prevalent IP EGP is the border gateway protocol (BGP, see RFC 1771).

Both of these router protocol systems are subject to attack. Attacks against routing protocols are a subset of the possible attacks against the network system itself.

Appendix A contains technical details about historic attack mechanisms and tools to identify and exploit latent bugs within COTS computing and network systems [32-39]. These mechanisms are not fully explained for nonsecurity personnel—a complete explanation of those details is outside of the scope of this document. Rather, those details are described in the appendix to provide partial evidence of the fact that the vast majority of modern computing equipment deployed within IP networks today cannot be trusted to be secure in general. Their security provisions, including their trusted paths and security controls, have repeatedly been demonstrated to not be viable when attacked. This point is discussed in section 4.4. However, to prepare the reader for that discussion, it is necessary to alert the reader about the myriad of vulnerabilities that are currently latent in today's COTS devices, vulnerabilities that can be exploited by (remote) attackers to attack the security, and possibly the safety, of networked systems. Readers who are unfamiliar with these vulnerabilities are encouraged to read appendix A before proceeding.

4.4 MIXING EMBEDDED SYSTEMS AND GENERIC OPERATING SYSTEMS.

Embedded systems can be successfully designed for high-assurance environments. For example, DO-178B defines processes so that Level A systems can have a very high degree of safety assurance. What is unknown, however, is whether these same systems will maintain their high level of assurance should they be deployed in a network environment for which they were not originally designed or approved. Embedded systems can be potentially stressed by localized attacks in ways that were not anticipated by their developers or certifiers, potentially creating unexpected results. For example, a latency-sensitive real-time application that is deployed within a networked environment should be evaluated with a view towards the effects that may occur should its supporting network experience an availability attack (see appendix A, section A.3). Therefore, each of the items that will be deployed within a generic network environment need to be evaluated for the entire gamut of network threats discussed in appendix A.

However, COTS (both generic and special-purpose) OSs, including those used for electronic flight bags (EFB), are inherently nonsecurable at this current time when deployed within network environments. A variety of reasons contribute to this, including:

> "… designing a 'truly' secure system (i.e., defending from all credible threats) is too expensive. In practice, limited development resources force compromises. Currently, these compromises are made on an ad-hoc basis, mostly as an afterthought. …
>
> Very often, security is an afterthought. This typically means that policy enforcement mechanisms have to be shoehorned into a pre-existing design. This leads to serious (sometimes impossible) design challenges for the enforcement mechanism and the rest of the system." [40]

Regardless of the cause, the security of COTS devices has repeatedly been shown to not maintain viability when attacked in networked environments. Even though specific bugs continue to be identified and fixed, the security profile of COTS devices has not improved due to the indeterminate number of latent vulnerabilities still remaining.

> "IP implementations have been tested for at least twenty years by thousands of computer professionals in many different environments and there are still vulnerabilities being discovered almost monthly." (Quoted from page 3-5 of reference 41.)

The National Security Agency (NSA) paper, "The Inevitability of Failure: The Flawed Assumptions of Security in Modern Computing Environments" [32], provides an analysis of why current COTS devices will continue to have ineffective security. The paper states the importance that

> "… assurance evidence must be provided to demonstrate that the features meet the desired system security properties and to demonstrate that the features are implemented correctly." [32]

It emphasizes the importance of implementing mandatory security policies implemented by means of nondiscretionary controls within OSs to enforce

- an access control policy,
- an authentication usage policy, and
- a cryptographic usage policy

These key policy systems are not rigorously supported by COTS OSs today.

> "To reduce the dependency on trusted applications, the mandatory security mechanisms of an operating system should be designed to support the principle of least privilege. … [A] confinement property is critical to controlling data flows in support of a system security policy. … A trusted path is a mechanism by which a user may directly interact with trusted software, which can only be activated by

> either the user or the trusted software and may not be imitated by other software. … This section argues that without operating system support for mandatory security and trusted path, application-space mechanisms for access control and cryptography cannot be implemented securely." (Quoted from Section 2 of reference 32.)
>
> "A secure operating system is an important and necessary piece to the total system security puzzle, but it is not the only piece. A highly secure operating system would be insufficient without application-specific security built upon it. Certain problems are actually better addressed by security implemented above the operating system. One such example is an electronic commerce system that requires a digital signature on each transaction." (Quoted from Section 5 of reference 32.)

Additionally, although not mentioned in the NSA paper, a secure system also needs to leverage secured communications protocols (see section 4.5).

Modern COTS OSs lack the controls that permit them to be secured in a high-assurance manner. Because of this, the applications that they host do not provide the provisions to permit them to have high-assurance properties either. For example, their access control and cryptographic functions cannot be implemented in a demonstrably secure manner today. These effects escalate, impacting the effectiveness of their data communications protocols and interdevice relationships, cumulatively potentially affecting the many devices populating the networked environment.

Another factor directly affecting the viability of COTS security in networked environments is the very high reliance that COTS devices have upon correct configuration and management practice. COTS devices usually have many possible configuration settings that must be properly set in a coordinated manner with the settings of other devices within the networked system if the cumulative protections of that networked system can be effective. The relative competency of system administrators and network administrators to correctly configure these devices is, therefore, an essential issue affecting the security of these systems. Because network security currently has such high operational reliance, it is not possible to certify the vast majority of COTS-based network environments today except at the lowest assurance levels.

While these observations about the security vulnerabilities of COTS devices in networked systems are sobering, it is important to recognize that these issues are not localized to avionics systems but rather are universally common to both industry and government worldwide.

Network systems are potentially vast collections of entities directly or indirectly cooperating together. The relative security profile of networked COTS devices is based upon each of the following dependencies working correctly and in harmony:

- Potentially complex device settings effectively coordinated among the devices network-wide. For COTS system elements, this traditionally equates to a high dependence upon the competency of system and network administrative personnel to correctly configure and manage networked devices over time.

- The dubious viability of discrete security subsystems within each device to withstand attacks.

- Dependence upon the users of the system behaving correctly.

Security systems with these interdependencies have numerous possible vulnerabilities that attackers try to identify and exploit. Current information technology (IT) security practices define mechanisms to defend these systems. These practices are as much of an art as a science. For this reason, IT security explicitly expects its systems to fail. This is why a core IT security tenet is to design defense-in-depth systems, implemented with full life cycle controls so that the total system may itself hopefully remain viable in the presence of security failure (see section 5.1).

Systems naturally evolve over time to reflect evolving policy, administrative competency, and technology changes. Exploits also mutate and evolve as well, taking advantage of available opportunities.

> “Models and assumptions used to develop security solutions must be grounded in real-world data and account for the possibility of failure due to unexpected behavior, both human and technological. … Any design will fail at some point. However, if you design for the inevitability of failure in mind, when it happens you’ll at least have a chance to find out about it. The key is designing systems that are able to fail gracefully. Determining that there is a problem when it happens is the best option for minimizing damage, besides preventing it outright. Solutions must be designed to make a great deal of noise when they fail or misbehave. Most systems end up doing something unexpected. When they do, you’ll want to know about it.” [16]

One of the more difficult policy issues currently confronting both the NSA (for certifying Department of Defense (DoD) systems) and the FAA (for approving networked aircraft systems) is: How can systems be certified at even moderate assurance levels whose protections have dependence upon subsequent human activity? For example, extensive operational evidence demonstrates that even the most security conscious environments have been accidentally misconfigured. Consequently, if human activity becomes an integral part of the network security posture, certification authorities have only a few choices:

- They could redefine the meaning of the concept of certification, significantly lessoning its assurance value.

- They could put so many restrictions upon specific certified systems that they are essentially nondeployable.

- They could extend the certification process to address the myriad of additional threats to devices that exist in networked environments. This is the approach presumed by this study.

However, the previous paragraph begs an even more fundamental question: Can Internet protocol-based network systems be certified for high-assurance deployments? That is, most IP implementations have a large number of possible configuration settings. If all the devices in IP network X are certified at a certain assurance level or above, does that mean that the network system itself also operates at that level? The NSA has previously observed this problem during the Rainbow series. Specifically, they had the Orange book [42] and then found that a secure collection of computers is not necessarily secure when networked. This resulted in the creation of the Red book [43]. However, the issue being discussed here is not primarily concerned with limitations of the Red book, or the resulting evolution to the common criteria (CC) [44-46], but the fact that security concepts are extended into networked environments by means of mathematically based security models, and that these models have no provisions for addressing client-side-attack or configuration-based uncertainties. The latter becomes relevant because the vast majority of IP devices today can be configured in many different ways. For this reason, this report states that an attribute of high-assurance implementations is that they cannot be misconfigured.

In conclusion, COTS devices, when deployed within large networked environments, are inherently nonsecure in general. These inherent risks can theoretically be mitigated by appropriate IA security practices. FAA studies, such as reference 47, have discussed possible mitigation approaches to address COTS vulnerabilities and encourages the mitigation of COTS vulnerabilities via mechanisms as those discussed in reference 47 and section 5. However, it simultaneously warns that the viability of these mitigation approaches are suspect to the extent that they rely upon COTS software and systems for their implementation. This is because COTS software and systems are not trustworthy, in general, when attacked. It is also because the efficacy of COTS software and systems are highly reliant upon (human) administrative oversight.

4.5 INTERNET PROTOCOL FAMILY SECURITY.

The IETF[7] has defined a series of protocols associated with IP, which is known as the IP family (also known as the transmission control protocol (TCP)/IP family). Table 1 describes an important subset of these IETF protocols. The table summarizes their security features and key management configurations. It contains many details that are outside of the scope of this document. These details are included within this table to provide evidence for the following generic observations.

[7] Internet Engineering Task Force (IETF); see http://www.ietf.org

Table 1. Internet Engineering Task Force Protocol Security Features and Key Management Configuration

Protocol	Security Features	Security Algorithm	Keys	Key Store in Linux
OSPFv2 (IPv4)—RFC 1583 OSPFv3 (IPv6)—RFC 2740 OSPF is an Interior (IGP)	Authentication, Integrity	Password plus MD5[8] (HMAC; see RFC 2085)	DES	The DES key used for the MD5 algorithm is specified on the command line when first invoking the OSPF daemon.
BGPv4 (IPv4)—RFC 1771 MBGP (IPv6)—RFC 2283 BGP is an EGP Protocol	Authentication, Integrity	Password plus MD5 HMAC (see RFC 2085)	Symmetric key whose printed ASCII value is 80 bytes or less (traditionally uses DES)	Linux implementations currently only support the BGP communities attributes that are configured during the BGP process invocation on a per-interface basis.
MOSPF—RFC 1584 Multicast OSPF is a multicast routing protocol	Authentication, Integrity	Password plus MD5 HMAC (see RFC 2085)	DES	The DES key used for the MD5 algorithm is specified on the command line when first invoking the MOSPF daemon.
PIM-SM—RFC 2362 PIM-DM—RFC 3973 Protocol Independent Multicast is a multicast routing protocol	Authentication, Integrity	Secured by using IPsec below		Uses IPsec

[8] MD5 is a message digest algorithm that was developed by Ronald Rivest in 1991. MD5 takes a message of an arbitrary length and generates a 128-bit message digest. In MD5, the message is processed in 512-bit blocks in four distinct rounds.

Table 1. Internet Engineering Task Force Protocol Security Features and Key Management Configuration (Continued)

Protocol	Security Features	Security Algorithm	Keys	Key Store in Linux
LDAPv3—RFC 2829 Lightweight Directory Access Protocol	Authentication, Integrity. Privacy	Simple Authentication and Security Layer (see RFC 2222) uses PKI; optionally TLS (see below) PKI uses	Kerberos or PKI	Client's PKI identity is registered (or passed) to the LDAPv3 daemon before or during client accesses.
HTTPv1.1—RFC 2616 Hypertext Transfer Protocol—primary protocol used for web accesses	Authentication, Integrity, Privacy	Secured by using TLS below	PKI	• Network Manager clients register their PKI certificate(s) to their web browser (e.g., Netscape Navigator or Microsoft® Internet Explorer) • A Web Server is configured with the appropriate PKI Server Certificate.
DNS—RFC 2535 Domain Name System provides IP address-to-name bindings. Also performs some directory services.	Authentication, Integrity	HMAC-MD5 (see RFC 2085) as used by the Secret Key Transaction Authentication for DNS (TSIG; see RFC 3645) mechanism (IETF is currently enhancing DNS Security)	Symmetric key obtained from a BIND utility	It is secured by pair-wise configuration of the same secret key between each DNS server pair that communicates together. Key assignments are configured using the *key* DNS statement in conjunction with the *keys* DNS substatement.

Table 1. Internet Engineering Task Force Protocol Security Features and Key Management Configuration (Continued)

Protocol	Security Features	Security Algorithm	Keys	Key Store in Linux
DHCP—RFC 2131 Dynamic Host Configuration Protocol is a mechanism for computers to receive dynamic IP address assignments.	Authentication, Integrity	HMAC-MD5 (see RFC 2085) as used by the TSIG mechanism (see RFC 3645)	Symmetric key obtained from a BIND utility	However, DNS TSIG is configured via same mechanism as for DNS. Managers may also secure DNS by the allow-update or update-policy substatements (within DNS RR) to provide access control to specific DHCP servers only.
SNMPv3—RFC 3414 Simple Network Management Protocol—see discussion in section 4.6 below.	Authentication, Integrity, Privacy	HMAC-MD5 (see RFC 2085) or HMAC-SHA-1 (see RFC 4231) (IETF is currently enhancing SNMP security)	Symmetric Key	Pair-wise assignment of two symmetric keys between each SNMP agent and each network administrator. This can be constructed from the user's password via the mechanism described in RFC 3414 or else distributed by an out-of-band method.
COPS—RFC 2748 Common Open Policy Service	Authentication, Integrity, Replay Protection	HMAC Optional: IPsec or TLS	Symmetric Key	

Table 1. Internet Engineering Task Force Protocol Security Features and Key Management Configuration (Continued)

Protocol	Security Features	Security Algorithm	Keys	Key Store in Linux
SSHv2—RFC 4251 The Secure Shell is a secure replacement for the ARPA Telnet, FTP, and TFTP services	Authentication, Privacy	Negotiated	Rivest Shamir Addleman (RSA) asymmetric key pair	RSA public key can be extracted from the user's PKI Identity Certificate and be stored within the ~/.ssh/authorized_keys file on a Linux system.
RTP—RFC 3550 Real Time Protocol for voice, video, and other real-time applications	Confidentiality	Payload encryption	DES key exchange occurs out-of-band	
RSVP—RFC 2747 Resource ReServation Protocol is associated with network policy- and reservation systems	Authentication, Integrity, Replay Protection	HMAC-MD5 is default but other stronger approaches (e.g., HMAC-SH1) are supported	Symmetric key	RFC 2747 explicitly did not define a key management approach. Therefore, every RSVP implementation probably has a unique mechanism for storing and distributing keys.

Table 1. Internet Engineering Task Force Protocol Security Features and Key Management Configuration (Continued)

Protocol	Security Features	Security Algorithm	Keys	Key Store in Linux
IPsec—RFC 4301 Internet Protocol Security	Authentication, Integrity, Privacy, Replay Protection	HMAC signed with Symmetric Keys. DES in cipher block chaining mode is the default but other algorithms/ approaches may be negotiated (e.g., by the Oakley variant of the Diffie-Hellman algorithm)	Two alternatives for configuring IPsec keys: • Manual key management requires the preplacement of Symmetric Keys • Automated key management requires an Asymmetric key to serve as a basis for creating (on demand) and distributing symmetric keys via the ISAKMP (see RFC 4306).	The Linux FreeS/WAN implementation permits automated key management through generating (and configuring) an RSA asymmetric key via the IPsec_RSASIGKEY utility. Alternatively, symmetric keys can be manually pre-placed within IPsec's databases on a security association (SA)-unique or common basis. The FreeS/WAN implementation also supports the use of PKI to function as a seed key value.

Table 1. Internet Engineering Task Force Protocol Security Features and Key Management Configuration (Continued)

Protocol	Security Features	Security Algorithm	Keys	Key Store in Linux
TLS—RFC 2246 Transport Layer Security Note: TLS is the standardization of Netscape's Secure Socket Layer Protocol version 3.	Authentication, Integrity, Privacy	Configured with an asymmetric key so that the protocol internally can compute secret keys for HMAC and privacy. Optional X.509v3 compliant digital certificates (e.g., PKI) for client/server authentication	Asymmetric key (e.g., RSA, DSS) or else PKI; TLS-record protocol uses symmetric keys for authentication and privacy: HMAC-MD5, HMAC-SHA1 TLS-handshake protocol uses asymmetric keys (e.g., Diffie-Hellman, RSA, Fortessa) as a basis for exchanging symmetric keys used by the TLS-record protocol	The PKI Server Certificate.
NTP—RFC 1305 Network Time Protocol	Integrity, Limited authentication	DES signing of a 64-bit packet checksum	DES cipher-block chaining	DES keys with associated Key Identifier stored within the NTP application

DHCP = Dynamic host configuration protocol
COPS = Common open policy service
TSIG = Secret key transaction authentication for DNS
HMAC = Hashed message authorization code
PIM-DM = Protocol-independent multicast-dense mode
PIM-SM = Protocol-independent multicast-sparse mode
ISAKMP = Internet Security Association and key management protocol
MBGP = Multiprotocol extensions to border Gateway Protocol Version 4
LDAPv3 = Lightweight directory access protocol version 3
BIND = Berkeley Internet name domain
DNS = Domain Name System
DSS = Digital Signature Standard
DES = Data encryption standard
V = Version

The IETF has been defining the protocols of the Internet protocol family for decades. The early Advanced Research Projects Agency (ARPA) net protocols (i.e., IP, TCP, user datagram protocol (UDP), and the ARPA services) were defined during the 1970s when the Internet was a trusted environment. These protocols either had very weak security (ARPA services) or no security at all (IP, UDP, TCP). As the Internet grew and evolved into an untrusted environment, the security provisions of the IETF's protocols improved. Security enhancements (i.e., Internet protocol security (IPsec) for IP, transport layer security (TLS) for TCP) and protocol replacement Secure Shell (SSH version (v)2 replaces the file transfer protocol (FTP), trivial file transfer protocol (TFTP), and Telnet ARPA services) were devised so that most of the original protocols could be secured. The security provisions of the newer IETF protocols reflect the security knowledge of the era when the protocol was designed. Certain protocols, therefore, were designed with what proved over time to have security limitations that thwarted their ability to evolve as best current practice network security evolved. Other protocols do not have these limitations and thus are able to use Federal Information Processing Standard (FIPS)-compliant encryption algorithms and keying material.

In all cases, the security provisions of IETF protocols are optional. Secured protocol deployments are unable to interoperate with unsecured protocol deployments. Originally, few, if any, deployments deployed IETF protocols with their security features turned on. More deployments have been configuring their systems to use these security features since network attacks have become increasingly common.

An attribute defining the IETF work in general is that they did not design their protocols in terms of a common top-down systems perspective. They were designed in a piecemeal fashion to resolve specific technology needs as they were identified over time. Until recently, lessons learned from the development of one protocol were incompletely applied to the development of other protocols. The lessons learned were not completely applied because the working group developing that protocol was composed of specialists for that particular technology, who may or may not be aware of how similar problems were addressed by other IETF working groups. Also until recently, the security provisions of protocols were designed in isolation, usually without reference to the security provisions used by other IETF protocols. As of mid 2006, the IETF has yet to begin trying to orchestrate the key management requirements of the various protocols that populate the IP family. As a result, the cumulative key management requirements for the IP family are varied and extraordinarily complex, with most protocols approaching key management in a unique and idiosyncratic manner. Worse, different implementations of the same protocol on different platforms usually have devised key management mechanisms that are unique to that implementation only. Thus, a very large diversity of key management approaches currently exist across the COTS Internet products. However, a few general patterns can be abstracted. These patterns are:

- Router-to-router protocols (e.g., open shortest path first (OSPF), BGP, and multicast open shortest path first (MOSPF)) generally need to be configured with identical passwords and symmetric keys for their communicating interfaces. The specific mechanism for accomplishing this varies widely between differing implementations of the same protocol. Although these protocols have similar algorithms, they are implemented differently on each protocol. For example, although OSPF and BGP use common password and symmetric keys, on OSPF, this is done on an area basis, while on BGP, it is done on a per interface basis. Please note from table 1 that the Linux implementations of BGP do not support MD5 authentication as of 2005.

- Lightweight directory access protocol (LDAP), HTTP, SSH, TLS and, optionally, IPsec rely upon asymmetric cryptography. However, the specific mechanism for doing this varies widely between these protocols. LDAP and TLS, for example, natively use X.509v3 conformant public key infrastructure (PKI) certificates. HTTP uses the underlying provisions provided by TLS. TLS can function without the use of asymmetric keys, but they are required if mutual authentication is supported. In the latter case, the server must provide a PKI Server Certificate and the Client a PKI Identity Certificate. IPsec only uses asymmetric keys for automated key management. The manual key management alternative, by contrast, solely uses preplaced symmetric keys. On Linux systems, SSH can be directly configured by running an internal Rivest Shamir Addleman (RSA) algorithm within their daemon to create their asymmetric keys.

- Other approaches require that unique symmetric key instances be distributed between each client-server pairing. This is the case for Domain Name System (DNS), dynamic host configuration protocol (DHCP), network time protocol (NTP) and real-time protocol (RTP). These symmetric keys must have been established at configuration time since these protocols lack a mechanism to dynamically distribute these keys. Simple network management protocol (SNMP) also requires unique symmetric key pairings between network administrators and SNMP agents; however, these keys may be constructed from the network administrator's password. The key point is that a single SNMP agent, DNS, DHCP, or RTP daemon within any given device has a large number of unique secret key values that are used on a per-protocol basis that it must maintain and associate with the appropriate remote peer. This represents substantial local key management complexity that is often implemented in a manner that is difficult to subject to administrative oversight.

4.6 NETWORK MANAGEMENT—NETWORK SECURITY CONCERN.

Network management is an inherently important and a difficult task. The difficulty of the task becomes increasingly untenable the greater the size and diversity of the deployed network devices being managed. This difficulty arises from subtle and not-so-subtle differences between various implementations of the management protocol and the management schemas (i.e., management information and variables) supported by the various devices. For example, in 2001, one of the authors of this document investigated industry support for the Distributed Management Task Force's (DMTF)[9] management schemas to examine their applicability to create policy-based network management constructs. He learned that although the vast majority of vendors claimed compliance with DMTF standards, upon closer inspection, it became apparent that they were supporting different (noninteroperable) versions of the schemas from each other. Most of the vendors had also introduced unique extensions to the schemas, and some of them had substituted constructs of their own invention for elements within the standard schemas. The net result was that a common management approach became increasingly untenable the more the deployment included different vendors products. A similar observation can be made concerning multivendor support for the SNMP's management information base

[9] DMTF; see http://www.dmtf.org/home

(MIB) information. Although the IETF has defined a great many standard MIB definitions, vendors often implement these MIBs in idiosyncratic and nonstandard ways. The net result is that the greater the diversity of deployed devices within a deployment, the harder it is to identify common MIB subsets that can be used to manage devices in a consistent way—and the less useful the total management system becomes. To correct this, vendors have built management systems[10] that operate at a higher level of abstraction. Such approaches overcome these limitations for the products that they support by creating localized clients that address the administrative differences between vendor products and present these differences in a regularized (abstracted) manner. Unfortunately, due to the cost of creating the clients, only the more commonly deployed systems are supported by these systems in general. Consequently, there is no single management system today that universally supports all IP products.

IP networks are historically managed using the IETF's SNMP. The first two versions of SNMP (SNMPv1, SNMPv2) do not have security provisions. While SNMPv3 does have well-defined security capabilities, helpful functions for enabling SNMP key management within multivendor environments are optional and, therefore, irregularly supported by the various vendors. The net result is that SNMPv3 key management is questionable when deployed in large, multivendor environments, debatably making SNMPv3 among the least secure of the major IETF-defined protocols for those environments. It is also possible that SNMPv3 in large, multivendor environments may be among the more vulnerable elements to attack within those deployments—a distinctly undesirable situation for a protocol that is used to remotely configure and manage network devices.

The following sections (4.6.1 to 4.6.5) identify historic weaknesses in SNMPv3 security. These sections are listed in order of increasing importance. The IETF is currently in the process of enhancing SNMPv3 security within the Integrated Security Model for SNMP (ISMS) working group to correct many of these problems.[11]

4.6.1 The SNMP has no Provisions for Two-Factored Authentication.

Many deployments require their system and network administrators to undergo two factored authentications to increase the difficulty of hostile attackers successfully impersonating these important functions. SNMPv3 has no provisions to authenticate based on PKI, password, biometrics, or almost anything else. SNMPv3 authentication is solely based on the user's symmetric authentication key. Therefore, the protocol has no provisions for supporting two factored authentication.

4.6.2 The SNMP Symmetric Keys may be Assembled From Passwords.

The symmetric keys that are used to authenticate and provide privacy for SNMP communications may be independently established for each user or they may be algorithmically constructed from the user's password. Although the former technique results in significantly better security, the latter is frequently used because it is the only commonly deployed

[10] e.g., HP OpenView; see http://www.managementsoftware.hp.com/
[11] See http://www.ietf.org/html.charters/isms-charter.html

mechanism to overcome the key distribution problem in multivendor environments. Many also use it because it is the simpler approach. Password-derived symmetric keys are no more secure than the passwords they are derived from.

4.6.3 The SNMP Key Updates do not Provide for Perfect Forward Secrecy.

SNMPv3's key update capability does not provide for perfect forward secrecy (PFS). PFS is defined in section 3.3 of RFC 2409 as:

> "When used in the memo Perfect Forward Secrecy (PFS) refers to the notion that compromise of a single key will permit access to only data protected by a single key. For PFS to exist the key used to protect transmission of data MUST NOT be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material MUST NOT be used to derive any more keys."

Specifically, in SNMPv3, replacement keys can be used to derive previous keys. As a result, if an attacker recovers a SNMPv3 authentication or privacy key, then he can decrypt all (recorded) traffic in the past even from previous key sets—assuming that he also captured the key change operation packets.

4.6.4 The SNMP Symmetric Key Distribution Problems.

Key distribution is a very serious implementation problem for symmetric key-based systems like SNMPv3. This problem has not been addressed by the IETF in general, and certainly has not been addressed within the IETF's SNMPv3 working group, in particular. Thus, there are no common or standard mechanisms used by SNMP implementations to perform initial symmetric key distribution. That is why so many systems rely upon password-based distributions. Solutions do exist within the IETF milieu for securely exchanging symmetric keys. However, none of them are standards, nor are they commonly deployed by SNMP products. For example, Kerberos has provided a mechanism for distributing symmetric keys, and RFC 2786 provides a Diffie-Hellman-like key exchange mechanism that is available for SNMP systems to use. However, the latter is an experimental RFC (i.e., it is not a standard SNMP mechanism) that is only implemented in a subset of SNMP products.

It is well known that many otherwise perfectly good security systems have been rendered ineffectual through improper or inadequate key distribution implementations. In regard to a deployment's use of SNMPv3, it is obvious that both the SNMP agent and the administrator's management system need to share a consistent, coherent, and interoperable approach to securely distribute these keys if the symmetric keys are to be securely distributed between them. Unfortunately, because this essential requirement is implemented on an ad hoc and idiosyncratic manner by different SNMP implementations, there is a strong basis for questioning whether these keys are being securely conveyed in multivendor environments, unless the deployment itself has used its own resources to create an out-of-band mechanism to do so.

4.6.5 The SNMP Currently Lacks Demonstrably Viable Session Keys.

Frequently changing privacy keys are very important for reducing the amount of ciphertext that is available for cryptanalysis. Because SNMP is built upon the UDP protocol rather than TCP, sessions are an abstraction and viable session keys are a challenge to create due to unpredictable request/response relationships. For this reason, the SNMPv3 privacy keys do not operate as session keys. Rather, they frequently have indefinitely long lifetimes, thereby permitting the accumulation of a substantial body of ciphertext over time. The attacker's ability to gather large amounts of ciphertext may potentially assist in the breaking of these keys.

4.7 MIXING DIFFERENT COMMUNICATION PROTOCOL SYSTEMS.

The NAS currently supports protocols belonging to several different protocol families (e.g., the ATN uses the OSI protocol). Private industry and governments repeatedly confront network and security issues related to the translation mechanisms used to get different protocol families to interoperate together. This section discusses relevant issues that need to be considered whenever diverse protocol systems are integrated together.

Protocol families is a generic term to refer to distinct protocol systems. The word families reflects the fact that some protocol systems are comprised of multiple orchestrated protocols that cooperate together to form a common system. Figure 11 identifies the two best known examples of protocol systems: OSI and TCP/IP.

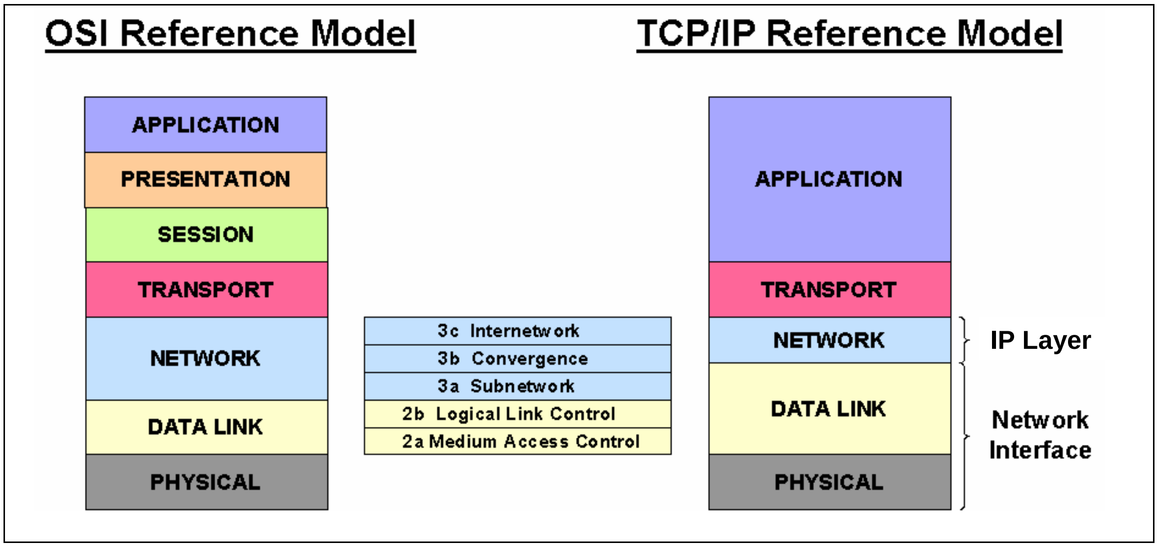


Figure 11. Comparison Between the OSI and TCP/IP Protocol Stacks

Deployments that need to support multiple protocol families usually identify a target protocol system and target infrastructure for the deployment to standardize upon. That target system becomes the network core and the other protocol systems "hang off" of that core. Almost without exception, the target protocol is the IP family, which is also known as TCP/IP. The nontarget protocol deployments are generically referred to as being legacy systems. To communicate with IP systems, legacy protocol systems either must be gatewayed into the core target system via protocol translators or converted over time to become part of the target system. For example, Bob Stephens [48] describes a possible approach where the NAS' ATN protocol and infrastructure can be modified to replace their current OSI connectionless network protocol (CLNP) protocol by IPv6,[12] thereby making ATN become an IP protocol system.[13]

ATN is currently an OSI protocol system. The OSI Reference Model defines the high-level unifying networking constructs that guided the creation of the OSI protocols. The OSI protocol deployments are divided into two major protocol variants:

- One major variant uses a connection-oriented network service (CONS) layer protocol and a less complicated transport layer protocol (TP) variant (i.e., CONS/TP0). This variant is patterned after historic X.25 protocol systems.

- The other major OSI variant that existed in the early 1990s uses a CLNP and a more-complicated TP variant (i.e., CLNP/TP4). This system is closely patterned after TCP/IP, whose definition preceded it by about a decade. This is the OSI variant that was selected for ATN.

The right-hand side of figure 11 shows the TCP/IP family protocol stack. The middle material between the two stack charts in figure 11 is an elaboration of the distinct sublayers defined by OSI for their data link and network layers. One should observe that although the IP Family's IP Layer is often referred to as being TCP/IP's network layer, it primarily corresponds to OSI Layer 3c (i.e., OSI's internetwork sublayer of the network layer).

This nit is being explicitly mentioned to introduce a very important point: Even though the OSI CLNP/TP4 variant was closely patterned after TCP/IP, the two protocol systems differ from each other in numerous significant ways. These differences ensure that their protocol deployments cannot naturally communicate together. For example, the state machines that underlie their various protocols are sufficiently different from each other to cause opportunities for the translation gateway devices connecting the two systems to potentially experience problems.

[12] The IP has two major variants: IPv4 (version 4) is the historic variant that currently populates the majority of the worldwide Internet infrastructure today. IPv6 (version 6) improves upon IPv4's scaling properties and is gradually replacing IPv4. The IETF has created several transition technologies to ease the migration from IPv4 to IPv6 and to enable networks to simultaneously support both protocols—a topic that is outside of the scope of this document.

[13] There is a significant "bug" in reference 48: The presentation contains stack charts that show both OSI's CLNP and TCP/IP's protocols being hosted over the data link limited liability corporation (LLC) protocol (i.e., OSI Layer shown in reference 48 could not interoperate with existing IP systems if IP is conveyed over an LLC protocol as 2b—See figure 11). This representation is true for OSI but it is false for IP. Specifically, the IP stacks shown (i.e., IP interoperability requires that a LLC protocol not be present).

Consequently, protocol translation gateways have repeatedly been shown to have high operational overhead, often requiring significant administrative oversight even when translating between two such similar systems. Most other legacy systems are significantly different from TCP/IP than CLNP/TP4, which increases the difficulty of doing protocol translation between them. This often results in the need for increased administrative oversight of those translation gateways for them to remain operationally viable over time.

4.7.1 Significant Semantic Differences to Allegedly Similar Concepts.

Figure 10 introduced the concept that different protocol families can be connected into the same network system by leveraging protocol translation gateways. This section discusses protocol translation gateways themselves. The key point of this section is that even when protocol families have similar concepts, there are often subtle but important differences between the semantics of those concepts within the protocol family milieu in which they operate that obfuscate translations between these protocol systems. For example, it was previously explained how the OSI CLNP/TP4 variant that the ATN network uses is based upon TCP/IP. Despite the great deal of similarities between these two approaches, section 4.7 stated that the protocol systems are not mutually interoperable nor do they have identical underlying protocol state machines, which causes some translation difficulties for the protocol translation gateway.

4.7.2 Integrating Dissimilar Protocol Families.

As was mentioned in section 4.7, deployments that seek to integrate multiple different communication families into an interoperable network traditionally accomplish this via common mechanisms. This section discusses these generic mechanisms in terms of a notional airborne infrastructure that includes both military and civilian aviation protocol systems.

Figure 12 shows that environments that do not try to connect their dissimilar network families into a common structure have created a deployment characterized by “islands of communications.” For example, military Link16 communicating systems are only natively able to communicate with other Link16 devices, but not with HAVE QUICK or Link4 devices. These islands of communications are traditionally bridged by protocol translator gateway devices (depicted as G/W in figure 12). As was previously mentioned, protocol translator devices are operationally expensive to deploy and require an unusual amount of administrative attention. They also create single points of failure, communications bottlenecks, and deprecate (e.g., added extra latency and processing needs) the total system performance. Nevertheless, these devices are an essential part of bridging islands of communications, at least initially, for environments that (1) have network-centric operations requirements, (2) have requirements for humans or applications to communicate between (or across) these islands, or (3) have collaborative requirements to communicate with entities beyond the communications reach of legacy protocol systems.

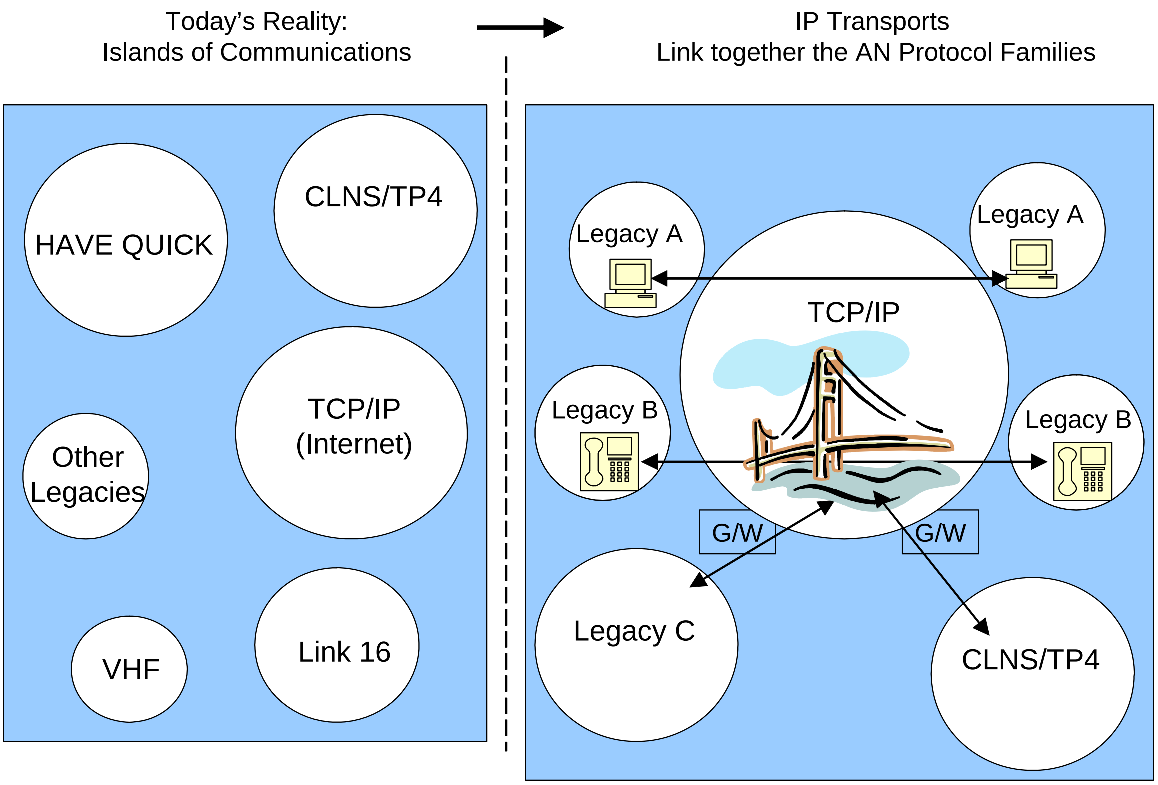


Figure 12. Transforming Islands of Communication Into a Single Logical Network Infrastructure

However, protocol translation gateways need not be the sole approach to bridge between protocol islands. Other approaches may include:

- Gradually migrating legacy applications and protocol systems to the TCP/IP family.
- Conveying (when appropriate) legacy communications over IP network transports.

Both of these approaches have been widely used within industry as mechanisms to replace protocol translators. For example, during the early 1990s, The Boeing Company internally migrated from 17 distinct protocol families to a single, corporate-wide enterprise network running IP. Examples of specific migration approaches that were used by Boeing to internally create its enterprise-wide IP network include:

- Networked Basic Input Output System was converted to run over TCP/IP via RFC 1001 and RFC 1002.

- OSI applications (e.g., X.400, X.500) were converted to run over TCP/IP via RFC 1006.

- Novell and Apple provided TCP/IP replacements to IPX/SPX (Netware) and AppleTalk in response to customer demand.

- IBM’s SNA evolved many mechanisms to be conveyed over TCP/IP transports (e.g., TN3270, 80d5 Ethernet, Multi-protocol Transport Networking, and the IBM AnyNet product line).

- Xerox XNS was gradually replaced by TCP/IP-based systems.

Figure 13 shows the protocol stack when TCP/IP family protocols are used to provide near-ubiquitous end-to-end communications to legacy environments (e.g., RFC 1006). Because the IP protocol can be conveyed over an extensive array of different media types, including many existing legacy systems (ultra high frequency (UHF), very high frequency (VHF), etc.), this approach directly leverages previous investments.

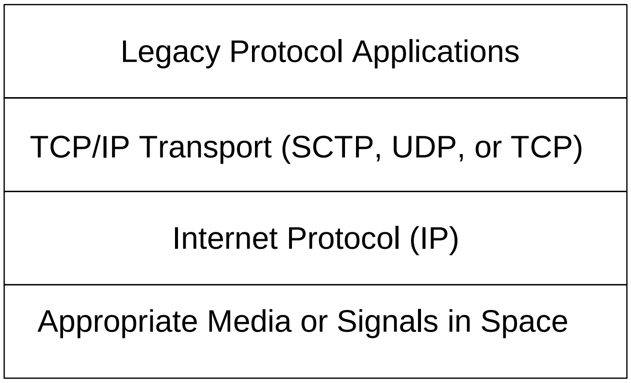

Legacy Protocol Applications
TCP/IP Transport (SCTP, UDP, or TCP)
Internet Protocol (IP)
Appropriate Media or Signals in Space

Figure 13. Internet Protocol Stack to Convey Legacy Protocols

Despite these benefits, IP-based communications may not be able to satisfy all legacy application requirements. Specifically, applications with extreme latency or jitter sensitivity may not be able to migrate to TCP/IP family transports despite the QoS improvements of IP systems. Systems that cannot evolve to use IP can integrate within the larger system as leaf nodes or edge subnets via protocol translation gateways to the IP infrastructure as is shown in figure 12.

4.8 IDENTITY PROBLEM.

IP has two major variants: IPv4 is the historic version of IP that currently populates the majority of the worldwide Internet infrastructure today. IPv6 improves upon IPv4’s scaling properties and is gradually replacing IPv4 worldwide. IP deployments may simultaneously support both IPv4 and IPv6.

The value of a specific IPv4 address is determined by the IP network topology location of its network interface in general. A multihomed IPv4 device, therefore, will have as many different IPv4 addresses as it has network interfaces, with one unique IPv4 address per network interface. This is because each network interface is located in a different network location within the IP routing topology. Specifically, the IP address value indicates the specific subnetwork to which that interface attaches, as well as the grouping of that interface within the other aggregations of the IP topology hierarchy.

Simultaneously, IP addresses are also used to identify application layer entities located within the device that hosts them. Therefore, IP addresses are semantically overloaded by simultaneously indicating two different semantic notions: routing topology location and device identity. The overloading of these very different semantic notions into the same address value results in what is known as the "IP Identity Problem." The identity problem may become manifested whenever a device physically moves within the routing topology (e.g., when aircraft move relative to ground-based infrastructures). Mobility can cause a conflict between the two semantic notions; because the moving entity has changed its network location, it is normally expected to readdress its network interfaces to reflect their new topological location. But if that is done, how can entities remote to that device authoritatively know that the device previously identified as having IP address X is the same device that now has IP address Y?

IPv6 addresses differ from IPv4 addresses in that each IPv6 network interface may simultaneously have multiple different IPv6 addresses, each with a potentially different network topology significance. IPv6 systems also support assigning unique IPv6 addresses to each application within that device. Consequently, IPv6 devices can support logical networks internal to that device itself, with each application supported by that device potentially having its own IPv6 address. By contrast, IPv4 systems are limited to referring to their applications solely via the port address field within the transport layer's protocol header (e.g., UDP, TCP, stream control transmission protocol).

Both IPv4 and IPv6 similarly share the IP identity problem, though its affects somewhat differ between the two protocol systems. Mechanisms to mitigate the IP identity problem are outside of the scope of this study.

The point of this discussion is that the worldwide civil aviation network infrastructure needs to devise a common mechanism by which the identity of networked elements is established. This means defining a common aeronautical solution for the IP identity problem for aircraft. If this is not done, then serious security vulnerabilities can arise whenever aircraft transition between system elements having dissimilar identity approaches.

4.9 INTEGRATED OR COOPERATING SYSTEM OF SYSTEMS.

The previous section discussed some of the issues related to creating a network infrastructure that links together two or more different protocol families. Section 4.8 mentioned the fact that TCP/IP systems have a weakness that is known as "The Identity Problem" that OSI systems do not share. The purpose of this section is to mention that the defense-in-depth provisions (see section 5.1) that are used to protect infrastructures rely on a coherent mechanism within that infrastructure for handling identity, authentication, and authorization. Should any of these elements not be handled in a consistent manner, then the infrastructure is subject to vulnerabilities that attackers can leverage to damage that infrastructure and potentially harm its safety attributes.

Each protocol family has its own mechanism for establishing identity. Protocol gateway translators will need to map between these different systems to successfully enable communications between dissimilar systems. The identity mechanism of each protocol system, and the mapping between them, must be assured to be consistent, complete, and definitive.

A great many different authentication and authorization systems exist. Should an infrastructure deploy multiple systems, then each alternative system and the mapping between them need to be assured to be consistent, complete, and definitive. Without such assurance, a possibility exists that flaws in these key foundational elements may exist, which can be hostilely leveraged by attackers. For this reason, the entire worldwide aeronautical infrastructure needs to define complementary authentication systems, preferably using a single, common authentication technology. It is helpful if they also use common authorization approaches, and the authorization system can be integrated into a consistent and coherent network management solution.

Assuring identity, authentication, authorization, and access control systems is much more of an art than a science. The task is simplified if a single technology for each system (identity, authentication, authorization, and access control) is deployed systemwide. For example, PKI has been proposed to become a common integrated authentication system for aeronautical systems [82]. PKI can be used for networks supporting many different network protocols. PKI is also used within the DoD (i.e., DoD PKI) to serve as the authentication system used by the military, including military aircraft. Regardless, a common technology should be identified as a common approach to standardize upon.

However, it is not always possible to ubiquitously deploy only a single technology solution system-wide. Because of this, some have proposed various mechanisms' cooperating systems that can be used to devise common policies that are

> "… expressed simply and in high level terms, but refined in many dimensions to map to specific infrastructures, organizational or individual needs and world events." [49]

Regardless of the specific mechanism used, whenever different security administrations or technologies are joined together in a cooperative manner (e.g., aircraft and ground systems), it is important and challenging to define the interfaces between the systems in such a way that the security posture for the combined system as a whole is not diminished.

5. NETWORK SECURITY DEFENSES.

This section discusses traditional mechanisms to try to mitigate those risks. However, it also contains sections that introduce specific concepts and technologies that provide important background information needed to understand important elements within the subsequent sections of this document.

5.1 DEFENSE-IN-DEPTH.

Networks traditionally attempt to mitigate the risks mentioned in section 4 and in appendix A, and, indeed, any possible network risk, by strategically deploying security controls in a defense-in-depth manner. Defense-in-depth means that redundant protection systems are deployed so that if one or more protection systems are defeated by an attacker, the deployment is still protected by the remaining viable systems.

The NSA's Information Assurance Technical Framework (IATF) [50] identifies the best current practice for securing network and information systems. This approach provides defense-in-depth protections at strategic locations within a network deployment. Each of these strategic locations needs to have their own set(s) of security controls. These strategic defense locations include:

- Defend the network perimeter (i.e., the AS).
- Defend the enclave boundaries (e.g., communities of interest within the AS).
- Defend each computing device.
- Defend each application.

Figures 14 and 15 show the defense and in-depth provisions at each strategic defense location. These provisions cumulatively form overlapping protection systems such that protection still exists even if an entire system fails. Specifically, applications are partially protected by OS protections. OS protections are partially protected by enclave protections. Enclave protections are partially protected by network defenses.

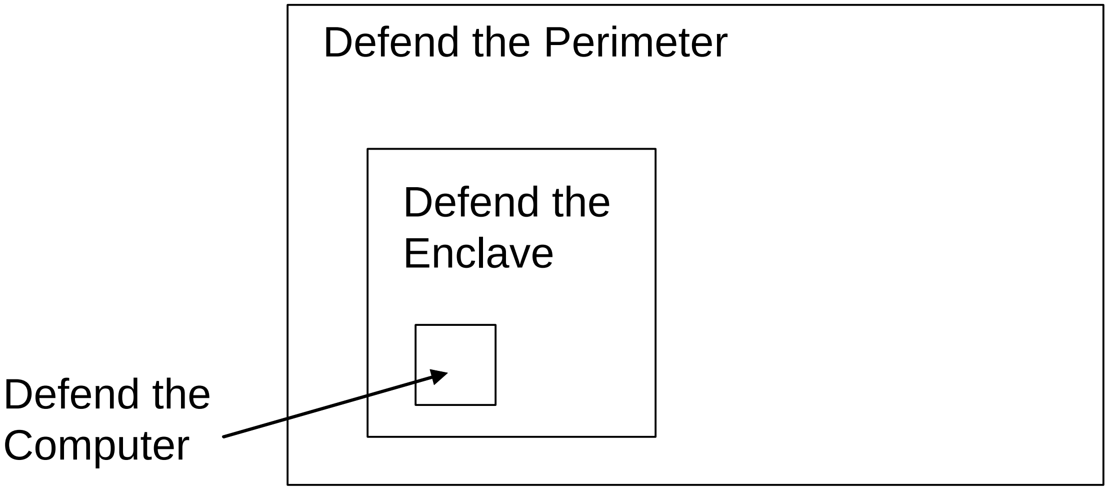


Figure 14. Overlapping Defense-in-Depth IA Systems

Defense-in-depth specifically means that redundant controls at each strategic defense location form a constituent part of the system design. For example, firewalls traditionally comprise part of a network's perimeter defense protections. However, as section 4.1 has already explained, there are three well-known attack vectors by which firewall protections can be defeated. For this reason, additional protections (e.g., VPNs, which can also function as enclave protections) are needed at the perimeter defense to maintain network integrity if the firewall protections are defeated.

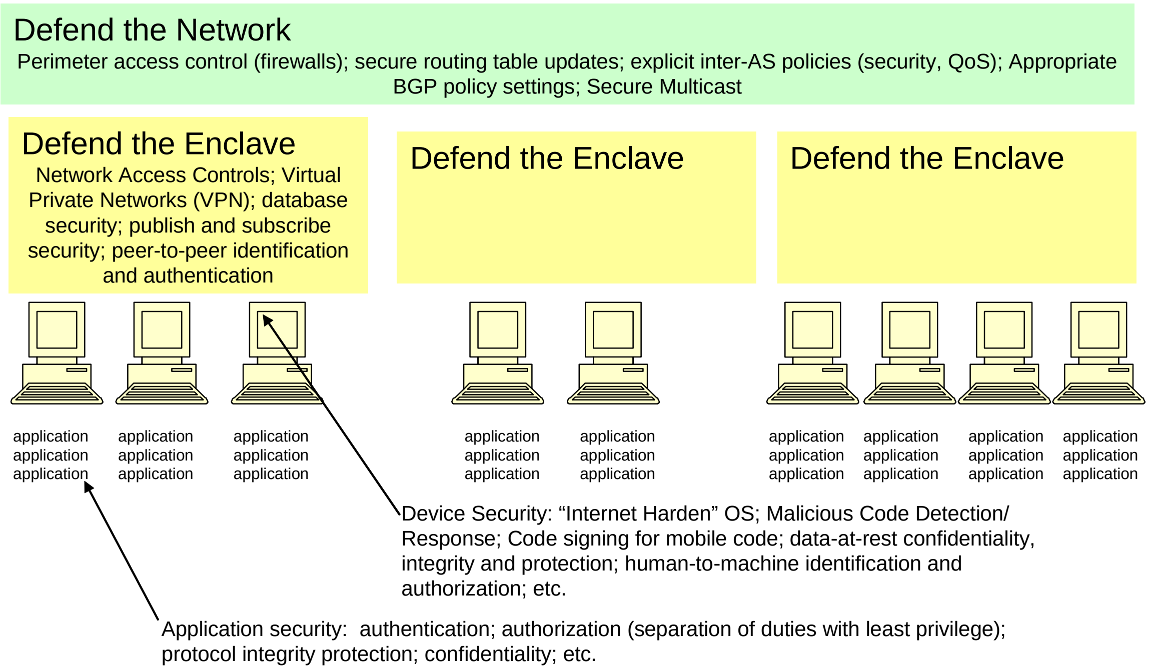

Defend the Network
Perimeter access control (firewalls); secure routing table updates; explicit inter-AS policies (security, QoS); Appropriate BGP policy settings; Secure Multicast
Defend the Enclave
Network Access Controls; Virtual Private Networks (VPN); database security; publish and subscribe security; peer-to-peer identification and authentication
Defend the Enclave
Defend the Enclave
application application application
application application application
application application application
application application application
application application application
application application application
application application application
application application application
application application application
Device Security: "Internet Harden" OS; Malicious Code Detection/ Response; Code signing for mobile code; data-at-rest confidentiality, integrity and protection; human-to-machine identification and authorization; etc.
Application security: authentication; authorization (separation of duties with least privilege); protocol integrity protection; confidentiality; etc.

Figure 15. Sample Defense-in-Depth Technologies

Each of these protection systems should preferentially support all elements of the control life cycle, which is shown in figure 16. Control life cycle defenses contain the following basic elements:

- Protection: security controls that provide protections to thwart possible attacks.

- Detection: security controls that detect, log, and report the existence of successful exploits that somehow overcame the protection system.

- Reaction/Neutralization: security controls that seek to neutralize any possible damage from successful exploits.

- Recovery/Reconstitution: controls that enable the entity to be reconstituted or recovered should successful exploits damage the entity beyond the capability of the neutralization controls to correct. The recovery and reconstitution often is integrated with system or network management processes.

The exemplar network architecture recommended by this study in (see section 8.3) heavily relies upon defense-in-depth concepts to defend against the network risks discussed in section 4 and appendix A.

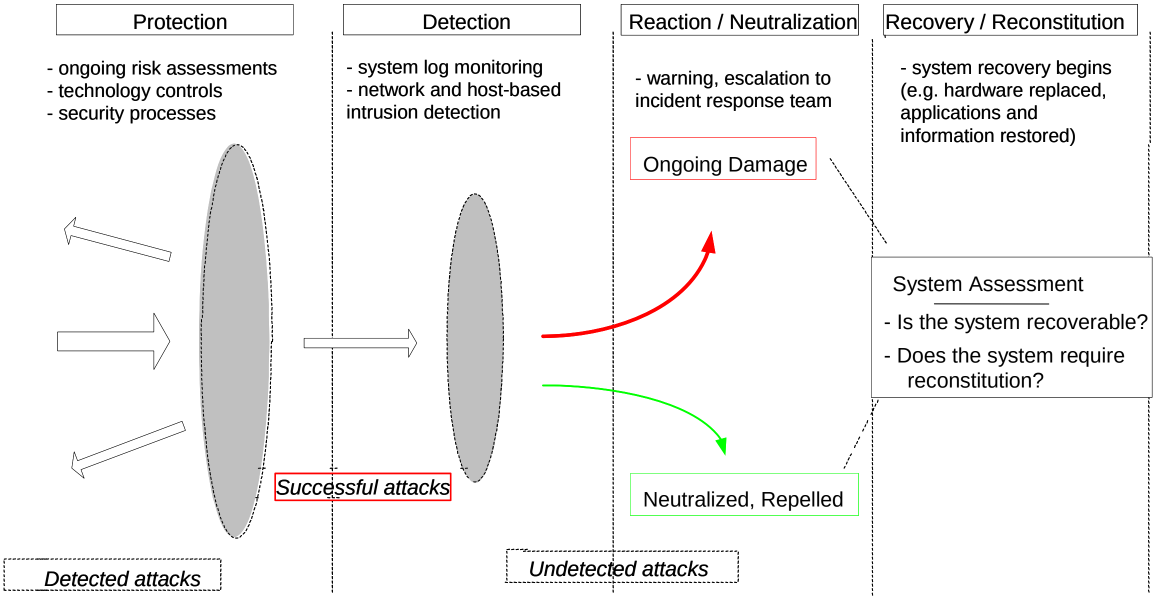

Protection
- ongoing risk assessments
- technology controls
- security processes
Detection
- system log monitoring
- network and host-based intrusion detection
Reaction / Neutralization
- warning, escalation to incident response team
Recovery / Reconstitution
- system recovery begins (e.g. hardware replaced, applications and information restored)
Ongoing Damage
System Assessment
- Is the system recoverable?
- Does the system require reconstitution?
Successful attacks
Neutralized, Repelled
Detected attacks
Undetected attacks

Figure 16. Control Life Cycle

5.2 DEPARTMENT OF DEFENSE NETWORKING.

The U.S. DoD is currently creating their next-generation network that has similar issues as the aircraft and NAS integration targets being addressed in this report. Section 6.3 will compare the DoD confidentiality classifications with the DO-178B software level safety classifications. Section 7 will propose extending the DO-178B and ARP 4754 safety concepts into networked environments by using the Biba Integrity Model [51 and 52]. The Biba Integrity Model is a direct analog of the Bell-LaPadula Confidentiality Model, which is used by the DoD to extend their confidentiality classifications into large system deployments such as networks. These proposed changes result in the safety extension approach that is recommended by this study for civilian aircraft, directly resembling the DoD global information grid (GIG) infrastructure, which is targeted for military aircraft. Because of this, this section provides a terse overview of how the DoD is designing their GIG.

The GIG seeks to empower the DoD’s network centric operations and network centric warfare vision. The GIG comprises the DoD’s internal network of networks, which is similar in concept to the worldwide civilian Internet infrastructure.

The architecture of the GIG is strongly influenced by DoD communications security (COMSEC) policies. The Bell-LaPadula Confidentiality Model forms the framework for confidentiality within U.S. DoD information processing, including the DoD’s COMSEC policy. This model creates a multilevel security system by means of mandatory access controls, labeling data at a specific classification level, and providing users’ clearances to a specific classification level. The controls ensure that users cannot read information classified at a security level higher than their own classification level, nor write information to a lower classification level, except via the controlled intervention by a trusted subject (e.g., a high-assurance guard (HAG)).

This framework is realized within military communications by creating networks, each operating at a specific classification level. These networks may operate as multiple single levels of security (MSLS) systems. Alternatively, they can operate as System High networks supporting all classifications at a given classification level or below. Networks operating at different classification levels are orthogonal to each other. For example, they are addressed, by definition, from address and naming spaces that are distinct (i.e., totally unrelated) to the address and naming spaces used by networks at all different classification levels.

In general, networks operating at one classification level have no idea of the existence of networks operating at a different classification level. There are two exceptions to this rule:

1. HAGs provide a controlled mechanism for some select communications to cross between networks operating at different classification levels (information downgrading and information upgrading). This includes appropriately mapping addresses between the dissimilar address spaces of the two networks. HAGs can "translate" between networks operating at different classification levels.

2. Military COMSEC provides a mechanism to encapsulate and encrypt data packets so that they can be conveyed over networks operating at a different classification level (see figure 17).

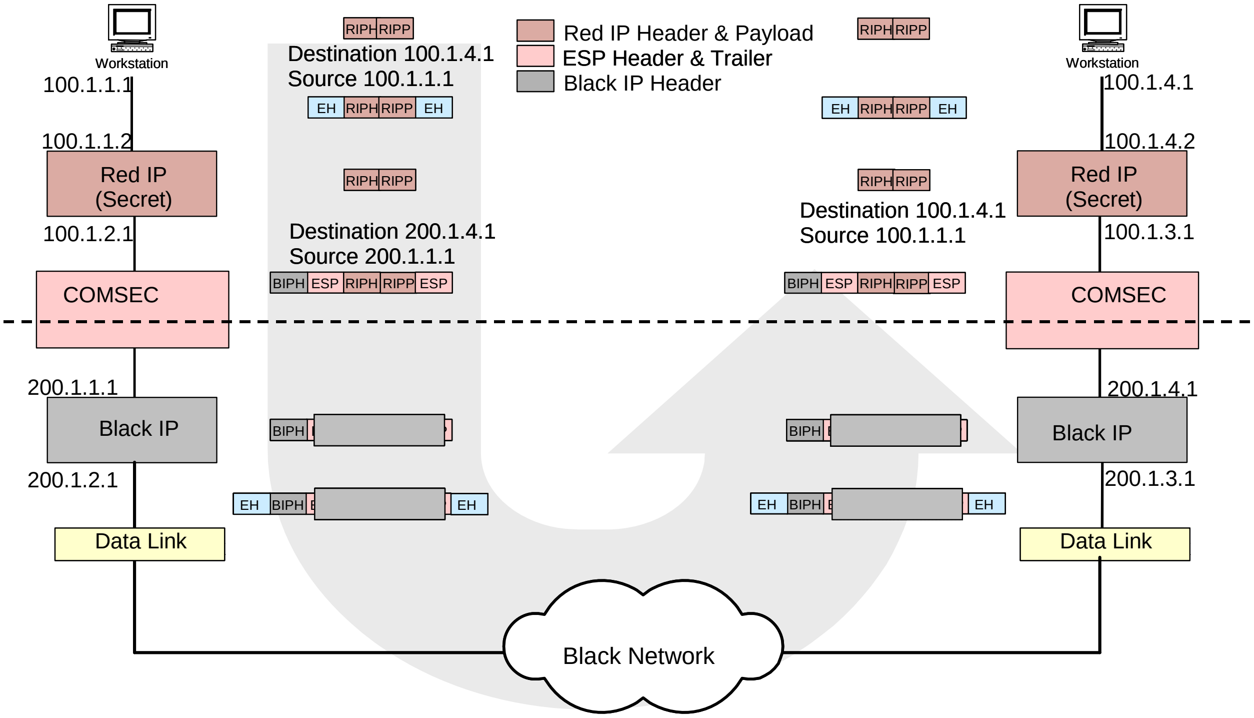


Figure 17. The DoD COMSEC End-to-End Packet Flow (IPV4 Example)

Current DoD COMSEC leverages the IETF's IPsec standard, whose architecture is defined by RFC 4301. Specifically, it is based upon IPsec's encapsulating security payload (ESP) (i.e., RFC 4303) operating in tunnel mode. Tunnel mode refers to a packet from one network being encrypted and then encapsulated with a packet header of a conveying network. That packet is then being tunneled across that conveying network to a point where the encapsulating/encryption process is reversed and the packet is re-introduced into a network operating at the original classification level. Many people refer to the conveying network as being BLACK, i.e., that they are often unclassified networks, and the conveyed network as being RED, which means that they may be classified at any specific classification level. Regardless, RED network packets are the original plain text packets, and BLACK packets are the cipher text (i.e., encapsulated and encrypted) packets. (Note: because the RED (customer) packets are encapsulated into that conveying BLACK network, the BLACK network itself is referred to as cipher text, even though the native non-VPN communications within that network are also normal plain text packets.) RED packets have only one IP layer header and operate in the normal manner, but BLACK packets have two IP layer headers: the original IP layer header that was used by the original end user and the encapsulated IP layer header that is used by the conveying network.

Figure 18 represents a possible mechanism for operating military aircraft within the DoD's GIG infrastructure. The figure has two distinct elements: (a) represents possible physical network systems and (b) shows how these systems logically work together.

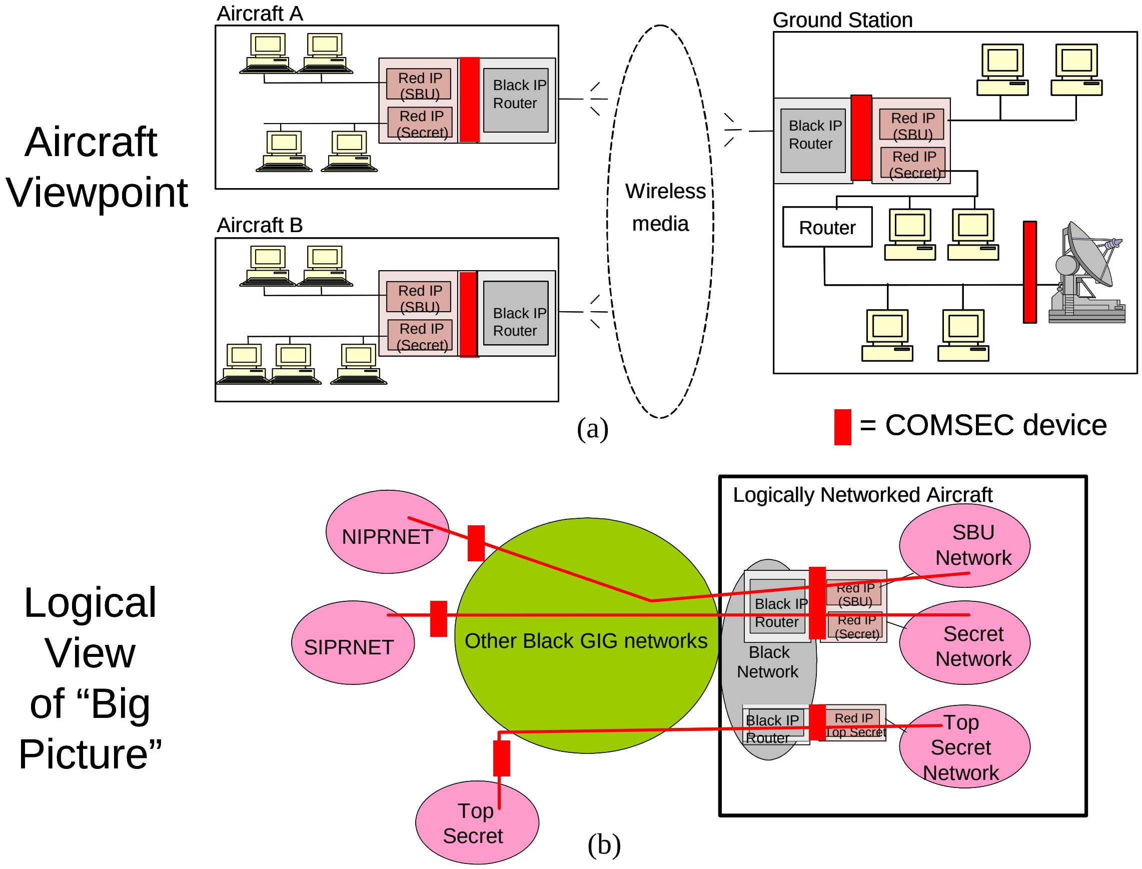


Figure 18. Representation of how Aircraft may Function Within the GIG

Figure 18(a) shows that the aircraft may internally support computing devices that operate at a specific classification level. These computing devices may be connected via onboard networks (LANs) that operate at a specific classification level. In those cases where aircraft internally support computing devices that function at different classification levels, they deploy distinct networks, each operating at that classification level. Alternatively, the devices can be connected in highly controlled ways via HAGs. These onboard computing devices and networks, are RED networks. Aircraft communicate together, and to ground stations, via wireless media that operate at an unclassified level. The onboard networks undergo COMSEC encryption and encapsulation into BLACK IP network headers to be conveyed across the wireless unclassified network. Thus, two distinct network systems exist: RED networks support end users and computer applications that are used by onboard communications. BLACK networks support the air-to-ground and air-to-air conveyance of that information.

If aircraft flying a common mission together establish RED network connectivity between themselves across BLACK air-to-air communications, then that mission logically functions as shown in figure 18(b). Specifically, different RED LAN segments within aircraft can become linked together to form common RED network systems, each operating at a specific classification level (e.g., sensitive but unclassified (SBU), secret, or top secret). Each of these RED systems can also communicate with equivalent remote computer applications or personnel at the same classification that are located in the same or different theaters of operation. For example, the figure 18(b) shows a mission that contains communicating elements (e.g., personnel or applications) that operate at three different classification levels: SBU, secret, and top secret. Each of these entities are shown as communicating with entities located within ground networks operating at their same classification level (e.g., the nonclassified Internet Protocol Router Network is an SBU network, and the Secret Internet Protocol Router Network is a secret network).

5.3 INTERNET PROTOCOL TOPOLOGY HIERARCHY AND POLICY SYSTEMS.

The IP natively supports a topology hierarchy comprised of increasing aggregations of networking elements (see figure 19). The figure shows that the IP assumes that the network interfaces with devices that are grouped into subnetworks, which are grouped into larger aggregations, depending on the scaling needs of the deployment. If the deployment has modest scaling needs, then subnetworks are grouped into an AS. If the deployment has larger scaling requirements, then subnetworks can be grouped into areas, which are grouped into an AS. A centerpiece of this hierarchy is the AS, which is the unit of routing policy within the IP topology hierarchy. IP’s standard (IGP, i.e., OSPF, intermediate system to intermediate system (IS-IS)) internally support up to two layers of hierarchy. When both layers of internal hierarchy are supported, then aggregations of subnetworks into areas occur, otherwise the IGP is deployed with a single layer of hierarchy, such that subnetworks are grouped into an AS. Therefore, IP’s IGP dynamically groups subnetworks or areas into ASs. IP’s EGP is the BGP, which is used to group ASs into internets (also known as “network-of-networks”).

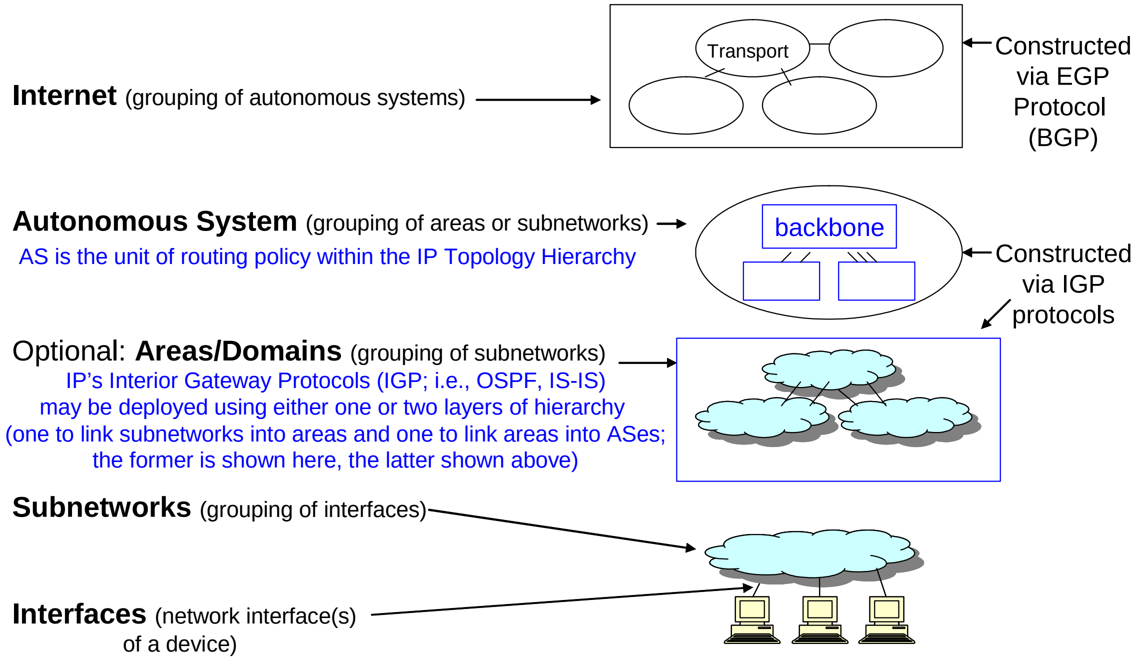

Internet (grouping of autonomous systems)
Transport
Constructed via EGP Protocol (BGP)
Autonomous System (grouping of areas or subnetworks)
AS is the unit of routing policy within the IP Topology Hierarchy
backbone
Constructed via IGP protocols
Optional: Areas/Domains (grouping of subnetworks)
IP's Interior Gateway Protocols (IGP; i.e., OSPF, IS-IS)
may be deployed using either one or two layers of hierarchy
(one to link subnetworks into areas and one to link areas into ASes;
the former is shown here, the latter shown above)
Subnetworks (grouping of interfaces)
Interfaces (network interface(s) of a device)

Figure 19. Internet Protocol Topology Hierarchy

As shown in the figure, each of these increasingly aggregated constructs is hierarchically constructed (e.g., a backbone or transport infrastructure connects leaf entities into a whole). This indirectly reflects a generic principal that network infrastructures have enhanced scalability and performance properties if they are organized hierarchically (e.g., references 53-58 discuss that principal as it applies to wireless networks). However, limiting deployments to purely hierarchical constructs has proven to be operationally confining for some network deployments, causing a less purely hierarchical provision to also be supported in a limited manner. For example, OSPF's not-so-stubby area permits a specific nonbackbone area to support BGP connections to another AS rather than the normal hierarchical case where only the backbone area can support such connections.

The AS is the unit of routing policy (e.g., security, QoS) within the IP topology hierarchy. This observation reflects the fact that an AS is a single administrative domain. For example, a corporation's network is grouped into an AS and relates to other corporations via the Internet's network-of-networks Internet infrastructure. In addition to providing routing information about the larger network-of-networks through their pairwise BGP connections, the connected ASs also establish formal relationships between each other where they specify how QoS, security, and packet data flow will be handled between each other's domains.

5.4 MECHANISMS TO CONNECT AIRCRAFT TO NETWORKS.

At least three very different models have been proposed for connecting aircraft to IP networks. Each of these models carries different assumptions and requirements.

- Network mobility (NEMO): The aircraft consists of a network (operating at a specific level of the IP topology hierarchy) that moves in reference to a largely stable infrastructure.

- Node mobility: The aircraft itself is a mobile node within a larger network system. There are two very different IP technologies that may be applied to this model:

  - Mobile Internet protocol (MIP)
  - Mobile ad hoc networking (MANET)

- Multilevel systems. For example, military COMSEC system (see section 5.2) views the aircraft as participating in two different network systems: the BLACK air-to-ground and/or air-to-air network system and the RED application/human to application/human network.

Combinations of the models are possible. For example, this study recommends that aircraft be defined as mobile ASs that have embedded VPN enclave partitions, thus creating a multilevel system. Specifically, aircraft communicate within the BLACK network, which defines the cumulative air-to-air, air-to-ground, and ground-to-ground network relationships. They operate as a mobile AS, and RED network enclave partitions, implemented by VPNs, operate as secure partitions within larger aeronautical network system.

5.4.1 Aircraft and Network Mobility.

The NEMO algorithm views on-aircraft networks as being mobile networks that change their point of attachment to a larger IP network infrastructure, affecting its reachability in the larger network topology. The approach assumes that the mobile network moves across the larger, comparatively stable IP network infrastructure. The IETF approach assumes that NEMO networks move between Internet attachment points (e.g., between different Internet service providers (ISP)). Of course, attachments are possible at other layers of the IP topology hierarchy. The IETF also approaches NEMO by leveraging mobile IP (MIP, see section 5.4.2) concepts. Other underlying algorithms are also possible.

This study recommends (see section 5.5) that the aircraft should be seen as being a mobile AS that moves in reference to other ASs within the larger aeronautical system. In this approach, each individual networked entity within the aircraft is IP addressed, and the network topology changes that occur as the aircraft moves are handled by the BGP protocol that links the aircraft to other ASs. IP addressing issues may arise with this model, depending on whether the aircraft’s IP addresses are associated with a specific service provider (e.g., classless interdomain routing (CIDR) addresses, see RFC 1517) or not (see section 5.5).

5.4.2 Aircraft as a Node (MIP and MANET).

Aircraft can appear as a single mobile node within an AS. This approach is most natural if only a single onboard computing device is remotely visible. However, if multiple onboard computers are visible outside of the aircraft, then the various onboard computers would need to be accessed via that same IP address. Specifically, the node at that address would act as a proxy (see RFC 3234) for the other processors on that aircraft. Because aircraft move in relationship with stable (ground or satellite) network environments, the aircraft will need to be treated as a mobile IP node. IP currently has two different mechanisms for doing this:

- The subnetwork that the aircraft's mobile node connects to can be organized using MANET[14] protocols. MANET protocols self-configure, creating their own network infrastructure in an ad hoc manner as their constituent wireless nodes move to provide routing services among themselves. The system may include one or more dual-homed nodes that contain a wireless interface and an interface connected to wired stable networks.

- The mobile node connects within IP networks using MIP.[15] This approach enables a mobile node to retain its permanent home IP address as it moves around the Internet. A home agent, located on the same subnet as the mobile node's permanent home address, intercepts packets sent to the mobile node's home address and forwards them to the mobile node's current address. This forwarding impacts the efficiency of the communications by adding latency and increasing transmission overhead.

5.4.3 Multilevel Network Systems (RED-BLACK, VPN).

Section 5.2 described the U.S. DoD networking approach at a high level of abstraction. That section described how DoD systems can leverage COMSEC protections so that networks operating at a given classification level can securely use the network transport services of networks operating at a different classification level. Specifically, it described how air-to-ground and air-to-air communications can be a network system operating at a different classification layer than onboard networks.

Civilian networks can also create multilevel network systems by using VPN technologies (see section 5.6). In common civilian use, VPNs provide a mechanism that permits an end-user's networks (e.g., a corporation's AS) to use network resources that are physically controlled by a different IP administrative domain (e.g., an ISP) in such a manner so that the conveying network appears to be an opaque link within the user's network (e.g., the corporation's AS). This approach is directly parallel to the DoD networks (i.e., the end-user's networks are RED and the ISP's are BLACK) and can be implemented by a number of technologies, including those used by the DoD.

These multilevel network systems can define controlled RED network partition enclaves within public (BLACK) network environments. These controlled networks are protected network enclave environments having user populations that are restricted to that enclave only. They, therefore, constitute significantly reduced "networked threat" environments by mitigating the network threats mentioned in section 4.1 (i.e., the threat that any user connected to any network

[14] MANET; see http://www.ietf.org/html.charters/manet-charter.html
[15] MIP; see http://www.ietf.org/html.charters/mip4-charter.html for IPv4 and http://www.ietf.org/html.charters/next-charter.html

that is indirectly connected to one's network is theoretically able to access one's network). This is in direct contrast with all approaches, which create structures that logically belong to the same larger network system. Unless mitigated by network partitions (see sections 5.4.1 and 5.4.2), the approaches operate in network systems that are logically connected together. The risks are described in section 4.1. By contrast, multilevel networks create protected network systems. Specifically, RED users cannot access BLACK network resources or vice-versa. By so doing, the users that comprise a given network within the multilevel network system are solely the users within that specific network system. Thus, they have a controlled network population within a controlled network system. By contrast, the users that comprise a single level network system are the cumulative users that can access any network within that system. In the case of the Internet, that would be more than a billion people.

5.5 AIRPLANE ROUTING AND AUTONOMOUS SYSTEMS.

The AS defines the administrative boundaries of IP systems (see section 5.3). Entities within an AS share common network policies (e.g., QoS, security). They also share common network administrative systems. While military aircraft often belong within a common AS with the other military aircraft with which they are associated (e.g., a squadron), and possibly with the military ground stations that support them, civilian aircraft usually belong to a different AS than the ground systems that support them. This is because civilian aircraft are usually either privately owned or owned by a corporation. In either case, the aircraft owners usually do not belong to the same corporation or agency as the ground stations that support them. While aircraft within the same corporate fleet may be organized into a common AS with other aircraft from that same fleet, this is not done in general because it would cause their intrafleet communications to be significantly different than their interfleet communications. Creating such dissimilar air-to-air relationships adds needless complexity to the entire system and may cause significant problems if not done correctly.

The previous paragraph should be readily apparent when aircraft are considered in terms of the IP networking concepts presented in section 5.3. Unfortunately, these IP topology hierarchy relationships permeate airborne network communications in subtle ways. The purpose of this section is to explain the pervasive nature of these concepts upon airborne networking and, by so doing, indicate some of the inherent technical challenges with designing viable airborne network systems (e.g., section 8).

The majority of this section is concerned with the routing implications of each airplane being its own AS. However, there are also IP addressing issues that derive from that association. With the advent of CIDR addressing, IP routing systems have increasingly relied on address aggregation to enhance scalability. CIDR has changed IP address semantics by embedding Internet topology information into the address prefix. This information identifies the specific ISP, which that entity uses to connect to the Internet. By so doing, address aggregation is enhanced for the BGP peering relationships between ASs, significantly improving Internet scalability. A side affect of this is that the IP addresses that airplanes adopt contain implicit IP network topology semantics, directly associating that airplane with a specific ISP. This may not be an issue if the worldwide airspace functions as a single ISP. However, a more likely scenario is that the airspace will be segregated into identifiable nationally or regionally controlled deployments. Regional flights that are localized within one of these boundaries would not be affected by this coupling. However, issues occur when aircraft cross between regions during flight since the airplane's original addresses were associated with their departure ISP. If they maintain those addresses during flight, they will reduce the aggregation and scaling and increase the overhead for the new ISP. There have been many proposed solutions to this problem.

- Re-addressing the airplane to the new ISP's address space.
- Assigning multiple IPv6 addresses to every airplane node, each associated with a different ISP.
- Assigning the airplane's IP addresses from private address spaces and then using a NAT to switch between ISPs.
- Use of provider independent IP addresses within aircraft. Note: Blocks of the IP address space are not associated with any ISP. Some of the largest corporations and entities (governments) intend to use these addresses so that they would not have any dependencies upon an ISP.

This study does not seek to suggest a specific solution to this problem. Rather, it emphasizes that IP addressing is a very significant architectural issue that directly affects connecting aircraft to IP networks. Specifically, both aircraft and the NAS need to operate within a consistent worldwide airborne IP addressing context if civilian aircraft are to cleanly communicate using IP networks.

Another significant issue is that the protocols within the IP family were designed for stable network environments having near 100% network availability.[16] Until recently, IP connectivity was primarily accomplished by means of wired media. Wireless media was primarily restricted to environments that were heavily engineered to operate within tight constraints that resembled wire line media environments (from the perspective of the IPs they supported, e.g., wireless LANs, cellular networks). As IP is beginning to be deployed within highly mobile wireless environments (e.g., MANET networks), IPs are encountering environments that significantly differ from their design assumptions. Specifically, the combination of high mobility with wireless media often result in high-signal intermittence rates, and correspondingly diminished network availability rates, for the communicating systems. This signal intermittence may be caused by signal interference from foliage, landforms, buildings, weather, particulate matter (e.g., sandstorms), hostile jamming, signal attenuation, and other factors such aircraft pitch, roll, and yaw introducing signal blockage due to relative antenna placement. IPs in general, and IP routing protocols in particular (both IGP and EGP), react to signal intermittence within their underlying media by greatly exacerbated protocol overheads. These overheads manifest themselves for IP routing protocols both in terms of network capacity consumption as well as in lengthened convergence times. IP routing protocols fail at certain signal intermittence rates.

[16] Network availability means that the network services are present and accessible. The concept of availability is distinct from the concept of reliability. For example, a network can be available (i.e., be present) but unreliable (e.g., packets can arrive with jitter, arrive in the incorrect order, or be lost).

Protocol failure manifests itself in terms of route oscillations, routing loops, starvation (i.e., data traffic destined for a network or host is forwarded to a part of the network that cannot deliver it), network segmentation, and increased packet latencies (delays).

The remainder of this section discusses BGP routing issues that derive from airplanes being in different ASs than other airplane or ground systems. This discussion presumes that each airplane will comprise its own AS. Because of this, aircraft will need to leverage the BGP protocol to remain connected to other air or ground entities. Readers not actively interested in BGP issues are encouraged to skip the remainder of this section.

A growing body of research currently identifies mechanisms (e.g., cross-layer feedback [59-61]) to improve lower layer and IGP routing performance in highly mobile wireless IP environments. However, EGP routing within such environments has only recently begun to be studied [62]. Because it is anticipated that civilian aircraft will operate in different ASs than the ground or aircraft entities with which they communicate, the wireless links between aircraft and ground stations will need to support EGP routing. Inter-AS connectivity almost always occurs within IP environments today using BGP.

Because BGP links two ASs together and because the AS is the unit of routing policy within the IP topology hierarchy (e.g., each AS has its own security and administrative requirements), BGP is designed to handle policy issues. Correctly reflecting these policies potentially complicates the configuration of the BGP connections, because they often reflect formal, legal contractual relationships established between those two organizations (i.e., corporations, governments). Specifically, BGP connections need to be well engineered and anticipated in advance [63] (i.e., it is not a reactive protocol) so that the specific configurations for each pairwise connection can be correctly orchestrated by both communicating peers.

BGP has the undesirable characteristic that a small routing change is propagated globally, delaying routing convergence system-wide (see [64-66]) in the resulting network-of-networks. Mobility and movement may cause signal intermittencies, attenuation, and loss on the BGP connections that link ASs together, potentially causing system instability. While BGP is slow to detect changes and restore routing, shortening the BGP timers improves upon invalid and missing routes but creates much higher protocol traffic overhead and possible protocol instability.

Because BGP was designed to be deployed within wired network environments, it exhibits a certain amount of brittleness when deployed across wireless links. Specifically, BGP was designed to support very stable interdomain connection environments. These assumptions may become challenged in environments where ASs move in relationship with each other. There are three issues that are particularly significant:

- Signal intermittence events may exceed the BGP timer values. When a BGP peer fails to receive “KeepAlive” messages from its neighbor, it will expire routes that use the neighbor as a next hop after "HoldTime" seconds.[17] If the timer values are increased to reduce the number of these time outs, then the responsiveness of the protocol is also reduced, including the time interval it takes for the remote peer to discover that the connection has been broken and, therefore, stop needlessly wasting wireless bandwidth by sending nondeliverable packets across that link.

- BGP can only establish well-known, pairwise connections (i.e., it cannot support meshes) and lacks a peer discovery mechanism. Therefore, as ASs move in relationship with each other, the possibility exists that the communicating peers will move out of range of each other. If this happens, then the BGP connection is dropped, even if other routers within the peer AS are still within transmission range of the aircraft. This connectivity brittleness is a primary difficulty of using BGP in mobile environments.

- Since BGP does not have a peer discovery capability, the AS boundary routers (ASBR) that host BGP communications need to be configured to connect to other ASBRs within their (remote) peer ASs where connectivity is anticipated to be needed during flight planning. Once such connectivity has been anticipated (i.e., the ASBRs for all ASs within the flight plan need to be correctly configured to enable each pairwise connectivity relationship), these connections can either be turned on in advance or turned on via a coordinated out-of-band mechanism during flight. The later alternative runs the risk of undergoing the loss of connectivity while the previous AS connections are torn down and the new AS connections established. If the aircraft is moving slowly enough, and the ground systems are positioned closely enough, it may be possible to accomplish this transaction while the aircraft is in range of both ground system locations, thereby avoiding loss of communications. However, a key point to recognize is that active BGP connections (i.e., unless the BGP connections on both sides are turned off) continue to attempt to connect with their peers even when they are physically out of range of each other, thereby needlessly wasting wireless network capacity.

The second and third issues can be theoretically mitigated by establishing BGP relationships between ASs across satellite links. As long as each BGP peer remains within the satellite's beam, the entity is not moving from the satellite's perspective. Since satellite beams can be geographically quite large, this may be an attractive solution for airborne environments. However, the benefit is reduced if the aircraft or ground station is near the edge of a beam, if geographical movement exceeds the beam's diameter in unforeseen ways, if the cumulative user capacity exceeds the cumulative satellite capacity of that geographic region, or if the satellite becomes unavailable. There is also the issue of mitigating adverse IP and TCP reactions to geo-stationary satellite latencies. For example, BGP itself runs over TCP transports. It is possible that other air-to-ground or air-to-air communications also run over TCP transports as well. Unfortunately, TCP treats latency as being network congestion. Thus, TCP inappropriately backs off its transmission rate for their sessions in response to geo-synchronous latency, reducing the efficiency of those links, unless mitigation techniques have been introduced into the system to address this issue.

[17] The RFC 1771-recommended BGP timer values are 120 seconds for ConnectRetry, 90 seconds for HoldTime, and 30 seconds for KeepAlive.

5.6 VIRTUAL PRIVATE NETWORKS ENABLE NETWORK PARTITIONING.

VPNs are a well established mechanism to partition network systems and to mitigate the types of risks previously mentioned in sections 4.1 and 4.2. Partitioning is an important mechanism by which the complexity of integrated systems can be reduced to improve the quality of the analysis and to mitigate failure conditions. For example, ARP 4754 states:

> "System architectural features, such as redundancy, monitoring, or partitioning, may be used to eliminate or contain the degree to which an item contributes to a specific failure condition. System architecture may reduce the complexity of the various items and their interfaces and thereby allow simplification or reduction of the necessary assurance activity. If architectural means are employed in a manner that permits a lower assurance level for an item within the architecture, substantiation of that architecture design should be carried out at the assurance level appropriate to the top-level hazard. …
>
> It should be noted that architectural dissimilarity impacts both integrity and availability. Since an increase in integrity may be associated with a reduction in availability, and vice-versa, the specific application should be analyzed from both perspectives to ensure its suitability. …
>
> Partitioning is a design technique for providing isolation to contain and/or isolate faults and to potentially reduce the effort necessary for the system verification process." (Quoted from sections 5.4.1 and 5.4.1.1 of reference 1.)

Partitioning provides isolation, independence, and protection for functions that are either highly critical (availability and integrity) or require protection (isolation, independence) to meet system availability and integrity requirements. VPNs create actual network partitions in full conformance to ARP 4754 section 5.4.1. VPN technologies appear to the network end-user to function as a private network except that private network technology is not being used.

According to RFC 4110, a VPN

> "refers to a set of communicating sites, where (a) communication between sites outside of the set and sites inside the set is restricted, but (b) communication between sites in the VPN takes place over a network infrastructure that is also used by sites that are not in the VPN. The fact that the network infrastructure is shared by multiple VPNs (and possibly also by non-VPN traffic) is what distinguishes a VPN from a private network." RFC 4110

Figure 20 shows that VPN networks are created by means of distinct interface points established between the network entity that provide a shared network service provider functionality to the distributed customer sites that the service provider is supporting. This study refers to the partitioned networks created by VPNs as being network enclaves.

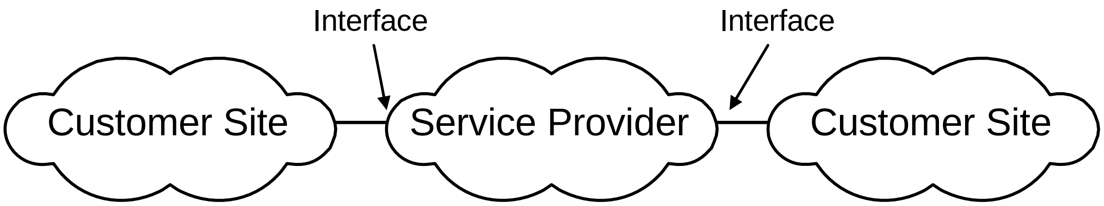


Figure 20. Interfaces Between Customer and Service Provider Networks

VPNs are examples of a multilevel network system (see section 5.4.3) where hosts within the service provider network cannot view, access, or know about hosts within the customer's networks, and vice versa. It is called virtual because the service provider forwards the customer's packets across its own network infrastructure in a manner that appears to the customer as if the service provider's network were the customer's own private network. The service provider can transparently provide VPN services to multiple different customers over that same physical infrastructure with each VPN being securely partitioned from the other. Each customer is provided a high degree of confidentiality and integrity protections from the VPN service, which protect their users from other VPN users of the same physical network infrastructure. This protection is accomplished either by data link layer protocol separations (see first bullet below) or else via tunneling (i.e., protocol stack encapsulations, explained in the second bullet below, which is the approach recommended by this study).[18] These inherent confidentiality and integrity provisions can be further strengthened by using IPsec security (see RFC 4301) in tunnel mode for network layer VPNs.

The IETF has defined two distinct types of VPNs:

- A Layer 2 Virtual Private Network (L2VPN) provides a VPN logically occurring at the customer's data link layer by using the service provider's physical network infrastructure operating at the data link layer. In L2VPN[19], a network provider offers the customer access to a VPN via a data link layer service interface (see figure 20). Consequently, the

[18] The mechanism by which physical network partitioning is accomplished differs in terms of the specific protocol layer at which the partitioning controls occur. The approach recommended by this study does the partitioning at the network layer (layer 3). The specific partitioning mechanism recommended by this study relies upon the controlled insertion (encapsulation) of a redundant IP packet header specific for the service provider network (i.e., the non-VPN enclave parts of the aircraft's network) within the protocol stack of the customer's (i.e., network enclave) packets (see figure 21) while they are conveyed across the network service provider's network. This encapsulation occurs at the interface point shown in figures 20 and 22. The encapsulated packets are conveyed across the network service provider's network by means of the encapsulated IP header (i.e., the service provider's IP header that was inserted into the protocol stack). The original IP packet header of the customer's packet, together with the entire contents of that original packet, is not visible to either the network service provider or to other VPNs supported by that service provider because they only can see the interface-inserted IP header. Additional assurance is provided by the fact that IP addressing of the original IP header comes from the IP address space of the (customer) network enclave, while the IP addressing of the redundant (encapsulated) IP header comes from the service provider's IP address space. The approach recommended by this study also has a third assurance mechanism: the customer's entire original IP stack is encrypted using FIPS-compliant encryption technology so that all network enclave packet information is in cipher text form while traversing the service provider's network. These provisions ensure total network partition between the various VPNs themselves as well as from the conveying service provider network.

[19] See http://www.ietf.org/html.charters/l2vpn-charter.html

VPN that is provided to the customer only appears to the customer to be a subnetwork (e.g., point-to-point wide area network (WAN) link, multipoint LAN) within the customer's own network. L2VPNs can be created by physically leveraging deployments of the service provider's asynchronous transfer mode, frame relay, Ethernet encapsulation in IP, or multiprotocol label switching (MPLS, see RFC 2031) networks.

- A Layer 3 Virtual Private Network (L3VPN) provides VPNs at the network layer (i.e., IP layer). In L3VPNs[20], a network provider offers the customer a private network infrastructure via an IP layer service interface (see figure 20). Consequently, the VPN that the service provider provides for the customer may be any IP topology hierarchy entity (e.g., subnetwork, area, AS, or network of networks). L3VPN networks that are designed for heightened security use IPSec's (see RFC 4301) ESP (see RFC 4305) in tunnel mode (e.g., see figure 21). Other technologies, in addition to IPsec, can be used to create other types of L3VPNs: BGP/MPLS, see RFC 2547 and reference 67), layer two tunneling protocol (see RFC 2661), IP/IP (see RFC 2003), and generic routing encapsulation (see RFC 2784).

Figure 21 shows a Layer 3 VPN example. This specific example is an IPv4 network that is using IPsec in tunnel mode to create the VPN. Note that this figure is essentially the same as figure 17, which showed how DoD COMSEC works in the DoD's GIG network.

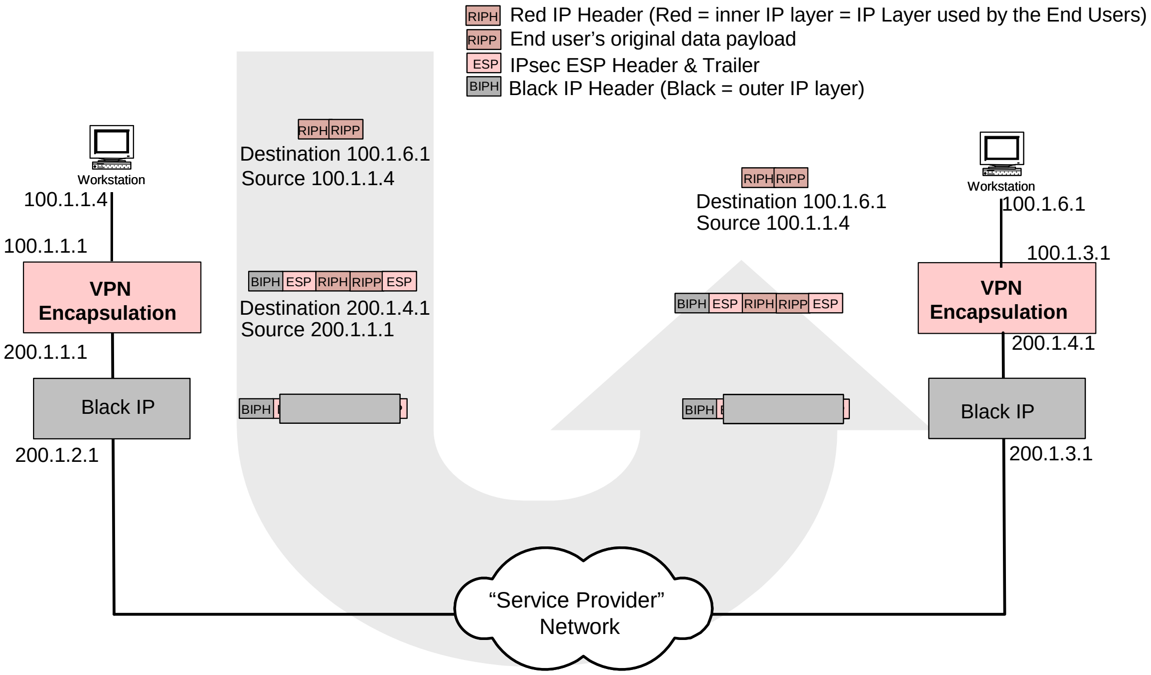


Figure 21. Example of VPN Encapsulation Using IPsec

Service providers provide L3VPN services by encapsulating an extra IP layer to the customer's IP layer protocol stack (see figure 22).

[20] See http://www.ietf.org/html.charters/l3vpn-charter.html

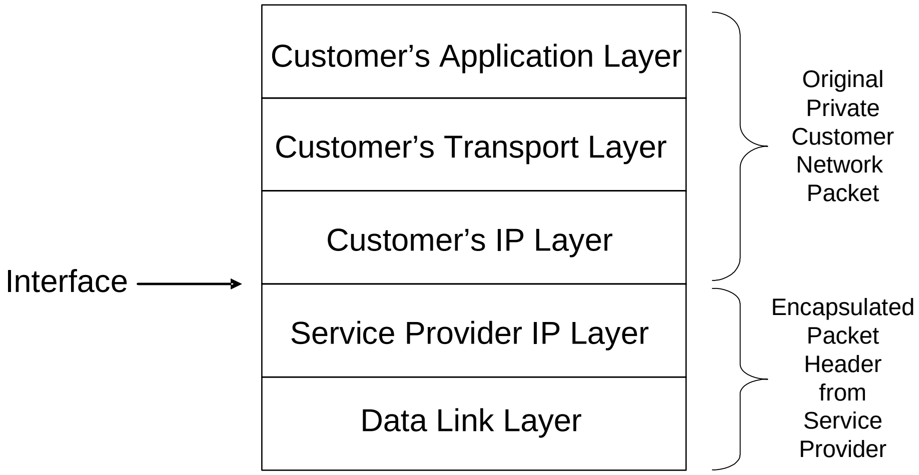

Customer's Application Layer
Customer's Transport Layer
Customer's IP Layer
Service Provider IP Layer
Data Link Layer
Interface
Original Private Customer Network Packet
Encapsulated Packet Header from Service Provider

Figure 22. Customer's L3VPN Protocol Stack Shown Within the Network Service Provider's Network

Specifically, the service provider provides an interface above its own IP layer for the customer to use to access the service provider's network. Figure 22 shows a common L3VPN protocol stack example where two IP layer protocols exist: one for the virtual network (i.e., the underlying service provider network) and one for the customer's own IP. Because the service provider's IP layer is an encapsulating redundant IP instance, it ensures that end-systems within the two network systems cannot communicate or be aware of each other (i.e., end-systems have only one IP layer protocol, not two). In this manner, the customer uses the service provider's network without being aware of other traffic using that same network, because the network traffic within the service provider's network occurs at the encapsulating IP layer, which the customer cannot see. It is similarly unable to access any devices directly attached to that network, nor can those devices access the customer's network because they only support a single IP layer and cannot see an (encapsulated) two IP layer protocol stack. It should be explicitly noted that the virtual part of the VPN occurs because of the abstraction that the service provider's network is solely supporting the customer: The other customers using that network infrastructure are not aware of each other's existence. L3VPNs are, therefore, an instance of multilevel network systems (see section 5.4.3).

RFC 4110, RFC 4111, and RFC 4176 provide architectural guidance for the creation of L3VPN network deployments. L3VPNs are an instance of multilevel network systems (see section 5.4.3).

5.7 SECURITY ZONES AND POLICY-BASED NETWORKING.

Policy-based networking (PBN) is a mechanism to create adaptive networking systems that continuously tune the network to achieve goals established by the network administrator. For example, it promises enterprises the ability to define business rules that can be translated into security or QoS policies that configure and control the network and its services as they evolve over time. While the approach sounds directly relevant to the topic of this study, this section explains why that is not the case. The final conclusion is that all open PBN alternatives have imploded due to the complexity of policy semantics causing prohibitive schema complexity in multivendor environments. Because of this, the only surviving PBN systems are vendor proprietary systems that are limited to a specific vendor's product lines. The remainder of this section can be skipped for readers who are not interested in this topic because it does not relate to the exemplar network architecture recommended by this study in any way.

Figure 23 shows a PBN framework that generically applies to many of the historic PBN approaches.

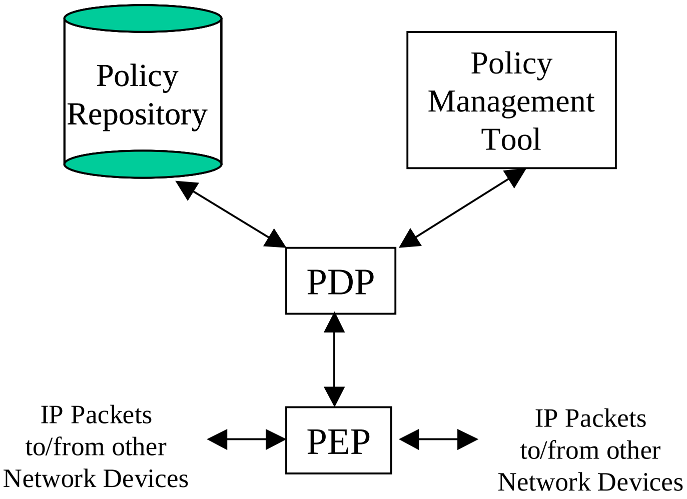

Policy Repository
Policy Management Tool
PDP
PEP
IP Packets to/from other Network Devices
IP Packets to/from other Network Devices

Figure 23. Historic PBN Framework

This figure is comprised of the following entities:

- A policy management tool that modifies the data found within the policy repository to articulate the current policies of the current deployment (e.g., a policy language response to environmental triggers). Policy is a set of rules that are used to manage and control the changing or maintaining the state of one or more managed objects. A policy rule is made up of four items: (1) metadata and semantics that define the behavior of the policy, (2) one or more events that trigger the policy, (3) a condition clause, and (4) an action clause.

- There are two distinct functions within PBN systems:

  - Policy Distribution Point is the mechanism for pushing policies and configuration data to configure a policy enforcement point (PEP).

  - Policy Decision Point is the functionality used by a PEP to enquire what it should do in specific situations. In this latter use case, the policy decision point instructs the PEP as to the proper action it should perform to enact the policies established for that AS.

- The PEP is the entity that actually enacts policy (i.e., a device).

Although the vast majority of PBN systems conformed to the architecture shown in figure 23, PBN approaches are historically divided into several distinct factions. Most of these alternatives are facing dwindling support today due to the complexity of their underlying policy systems. The more widely known PBN approaches include:

- The Distributed Management Task Force's[21] common information model (CIM) and directory-enabled networking (DEN) work. The DMTF CIM model is widely supported by most NMSs. However, many vendors have also tried to use CIM and DEN to enable PBN. Those latter attempts have not succeeded due to schema complexities.

- The IETF's[22] former Resource Allocation Protocol working group previously defined the COPS (see RFC 2748 and RFC 2749) protocols as well as an alternative approach to CIM/DEN for specifying policy and device configurations. This latter approach was defined by RFC 3159, "Structure of Policy Provisioning Information (SPPI)." This work leveraged the existing SNMP MIB work that is widely used today within IP-oriented NMSs to create a parallel structure to the MIB for conveying policy and configuration data, the policy information base (PIB). This latter concept was quite popular within the IETF for many years, influencing many other IETF working groups, including IP security policy (IPSP) and the Differentiated Services QoS working groups. Unfortunately, the various IETF PIBs were poorly coordinated together. Many of them used different schemas to do similar things, and the complexity of many of these systems was significant. For these reasons, this work has also lost its former popularity and mind share.

- The TeleManagement Forum's[23] shared information and data work, including their next generation DEN work, which is unrelated to the DMTF DEN work.

Although the construction of large multivendor policy-based systems had achieved a significant amount of mind share at one time, actually trying to build PBN systems using the figure 23 model consistently demonstrated how difficult and challenging the articulation of policy itself turns out to be [68].

Because of the sheer complexity associated with policy articulation (e.g., RFC 3060, RFC 3084, RFC 3159, RFC 3317, RFC 3318, RFC 3460, RFC 3571, and RFC 3585), multivendor PBN attempts to date have ultimately imploded. For this reason, this study recommends that airborne or NAS systems should not be designed using technologies that require significant policy complexity or a high degree of policy coordination between networked elements.

By contrast, Steve Bellovin wrote a report in 1999, "Distributed Firewalls" [69], which described a mechanism to build policy-based networks by leveraging the IPsec protocol (see RFC 4301). IPsec is a protocol that is implemented natively by IP devices. This approach addressed most of

[21] DMTF; http://www.dmtf.org
[22] IETF; http://www.ietf.org
[23] TMF; http://www.tmforum.org

the problems that occurred with the more ambitious PBN approaches. It has been used to create distributed firewall systems [70], including the construction of discrete security zones within the network infrastructure (i.e., elements of a network deployment with heightened or specialized security requirements different than the rest of the deployment). This remains a promising approach for implementing PBN systems.

The IETF (e.g., its former IPSP working group) has assembled several tools that can be optionally leveraged to create PBN systems using IPsec.

- RFC 3586 describes the problem space and solution requirements for developing an IPSP configuration and management framework.

- RFC 2704 describes the KeyNote policy language that can optionally be used to construct PBN systems. The KeyNote implementation functions as a compliance engine and is based on role-based access control techniques as encoded within PKI attribute certificates.

- Use of IPsec's ESP (see RFC 4305) in Transport Mode to provide confidentiality, data origin authentication, antireplay attack protection, and data integrity services to enhance network security between communicating devices (e.g., hosts-to-hosts, routers-to-routers) at a specific integrity level.

The Defense Agency Research Projects Agency (DARPA) Strong Man work originally experimented with integrating KeyNote with IPsec's Internet Key Exchange (see RFC 4306) protocol to create a very fine-grained authentication and access control infrastructure at the network layer [70]. These communications are secured by using IPsec in Transport Mode between communicating devices. A public implementation of this approach is freely available and is built into the Open BSD[24] Unix OS.[25] This approach creates a tight knit PBN system that has not been widely deployed to date.

However, IP deployments have been enhancing their network communication security by increasingly using native (unmodified) IPsec communications between their devices. DoD network systems (see section 5.2) and VPNs (see section 5.6) use IPsec's ESP in Tunnel Mode to create secured multilevel network systems. This creates controlled and protected network enclaves that have significantly reduced user populations within reduced networked-threat environments. Deployments are also increasingly using IPsec's ESP in Transport Mode within a common network enclave to create higher assurance communications within that network. Through systematically using native (unmodified) IPsec capabilities, these deployments are creating network environments with significantly improved network security today.

[24] See http://www.OpenBSD.org

[25] Specifically, most of this functionality is built into isakmpd (/usr/src/sbin/isakmpd) within the OpenBSD operating system (see ftp://ftp.openbsd.org/pub/OpenBSD/src/sbin/isakmpd/).

6. RELATING SAFETY AND SECURITY FOR CERTIFICATION.

Daniel Mehan, the former CIO of the FAA, wrote:

> "For FAA, information systems security extends beyond the computer environment to the security of airspace and the national airspace system. The structural, operating, and procedural foundations of information systems security provide the mechanisms for achieving FAA's safety, security, and efficiency goals." [18]

Airborne system safety as it relates to software is safeguarded by DO-178B [5] procedures, processes, and guidance. By contrast, traditional IT security is evaluated in terms of CC mechanisms [71, 44, 45, and 46]. The DO-178B processes are primarily focused on safety. The CC processes are primarily focused on security. Carol Taylor, Jim Alves-Foss, and Bob Rinker contrast the two approaches as follows [72 and 73]:

> "DO-178B is intended to certify that software used in aircraft is developed with "best known" practices and does not contribute to aircraft safety hazards. Software is not ever certified as a standalone component but only as a part of aircraft or engine type. Emphasis in DO-178B is in outlining general policies and procedures to produce safe software in terms of airworthiness requirements and to produce documentation to substantiate that the development requirements have been met. Thus, language and content is high-level and abstract leaving a lot of compliance decisions up to the developer.
>
> The Common Criteria (CC) is intended to specify security requirements that a system, hardware or software must satisfy in order to achieve a specific level of assurance. The CC only deals with security functionality of systems and does not address overall development issues except where they affect security. The CC is a much more detailed document in terms of specifying how compliance is achieved for an intended product. Each component of each assurance class has specific action elements and evidence of compliance for both developers and evaluators." [72]
>
> "While the purposes for each certification are clearly different, many of the requirements and procedures are aimed at insuring that the final design and implementation meets certain quality standards. In many cases, these standards are similar, and by modifying or adding to the current procedures in each case, a single common process can be developed which will satisfy both certifications. Since each certification process can potentially be quite expensive, a common process should result in significant cost savings for those systems that must meet both standards." [73]

As Dr. Mehan has repeatedly observed [18 and 19], safety and security are related concerns within the NAS. The U.S. federal government has subsequently studied mechanisms by which safety and security can be combined into a common, integrated process [13, 40, 41, and 72-77] with common certification results. It has specifically studied mechanisms for integrating CC security evaluations and DO-178B safety processes, including:

- Common Certification of Airborne Software Systems [72]
- Dual Certification for Software [73]
- Common Security Testing and Evaluation [41]
- Integrated Capability Maturity Models [76]

As a result, a growing body of work exists to guide the government and industry for how to create processes to design, test, evaluate, and certify airborne and NAS system elements for safety and security.

However, the optimum mechanism by which to relate safety and security in airborne systems has remained elusive. Resolving this issue forms one of the primary goals of this study. This issue is directly addressed in this section. This study has significantly diverged from previous studies by concluding that the primary issue, as it relates to network airborne safety, is not how to correlate DO-178B safety and CC security concepts and processes, as was presumed by previous studies, because such comparisons produce ad hoc results. They are ad hoc because while safety and security have become intertwined concerns in airborne environments, they are nevertheless distinct concepts from each other. Rather, this report states that the primary issue impacting network airborne safety is how to extend existing safety policies for airborne system, hardware, and software into networked environments in a mathematically viable manner. This section recommends that this can be accomplished by using the Biba Integrity Model. This approach preserves current safety assurance processes and extends them into networked environments. Section 6.2 begins the explanation of the relevant issues. However, before that can be done, section 6.1 will discuss the derived security requirements of networked safety environments.

6.1 SECURITY REQUIREMENTS OF AIRBORNE NETWORKED ENVIRONMENTS.

The information presented in this section has previously been discussed in many studies. Readers interested in additional information about these concepts are encouraged to read references 9, 17, 20, and 78-80.

Section 4 and appendix A mention a great many specific security risks that can occur within networked environments. Due to the target-rich nature of this situation, it is not possible to enumerate all possible security risks that may conceivably occur within airborne network environments. This section will consider the security requirements at a high-level of abstraction in terms of traditional IA concepts (see glossary). It is important to reiterate that the primary requirement of all airborne environments, including networked environments, is safety. The security requirements articulated in this section are derived from the need to mitigate the known security threats that occur in networked environments so that these security threats will not create software failure states that could impact safety. These security requirements presume traditional IA best current practices that were previously described in section 5.1.

6.1.1 Integrity.

As section 4.3 indicated, there are three different objects within networked airborne environments whose integrity particularly needs to be preserved:

- Integrity of the communications protocols that traverse the network (e.g., controls are needed so that modified packets can be recognized as having been modified). This can be ensured by only using secured versions of IP family protocols (see section 4.5). Device and user communications can be secured using IPsec in transport mode (see RFC 4301 and RFC 4303).

- Integrity of the security controls of a device that is used for the defense-in-depth security protections of that distributed system. This traditionally pertains to OS controls, but also includes security applications (e.g., network intrusion detection system (NIDS), firewalls). This will rely upon the IA provisions previously discussed in section 5.1.

- Integrity of the applications that support airborne operations. Specifically, airborne and NAS systems shall not be removed (e.g., turned-off), modified, or replaced by unauthorized personnel or processes. These provisions rely upon the viability of the availability and authentication provisions (see sections 6.1.2 and 6.1.3) deployed within the infrastructure.

Safety-critical systems are currently designed to survive in the presence of bad data. It must be assured that components used for safety-critical applications protect themselves from bad data.

Software parts present a challenge for verifying the integrity of the delivered component, especially if it is delivered electronically over a public network where tampering could occur. Airborne systems need to ensure that effective process controls are placed on electronic software so that they are appropriately signed by authorized entities, properly stored, securely downloaded, and that only authenticated software versions are actually deployed in NAS or airborne environments. Software parts are traditionally secured within the U.S. federal government and industry by establishing security engineering processes that leverage the U.S. federal Digital Signature Standard (DSS) (FIPS 186) [81]. FIPS 186 itself leverages PKI technology and infrastructures.

Software code signing is the application of the FIPS 186 to software executable code. Figure 24 shows a process by which code is signed.[26] Figure 25 shows the process by which received code that has been signed is verified. Code signing is a mechanism to establish the authenticity and integrity for software executable content. The signature provides authenticity by assuring users (recipients) as to where the code came from—who really signed it. If the certificate originated from a trusted third-party certificate authority (CA), then the certificate embedded in the digital

[26] FIPS 186 uses terms synonymous to those used within figures 24 and 25. FIPS 186 refers to the hash algorithm as being the secure hash algorithm. It also refers to the one-way hash as being a message digest. FIPS 186 does not require the signer’s PKI certificate to be inserted into the signed code, although that is the usual manner in which it is done in actual practice. (Note: the signer’s certificate includes the signer’s public key.) Rather, FIPS 186 only requires that the public key be available for verification, without specifying how it is made available.

signature as part of the code-signing process provides the assurance that the CA has certified that the signer of the code is who they claim to be. Integrity occurs by using a signed hash function that authoritatively indicates whether or not the resulting code has been tampered with since it was signed.

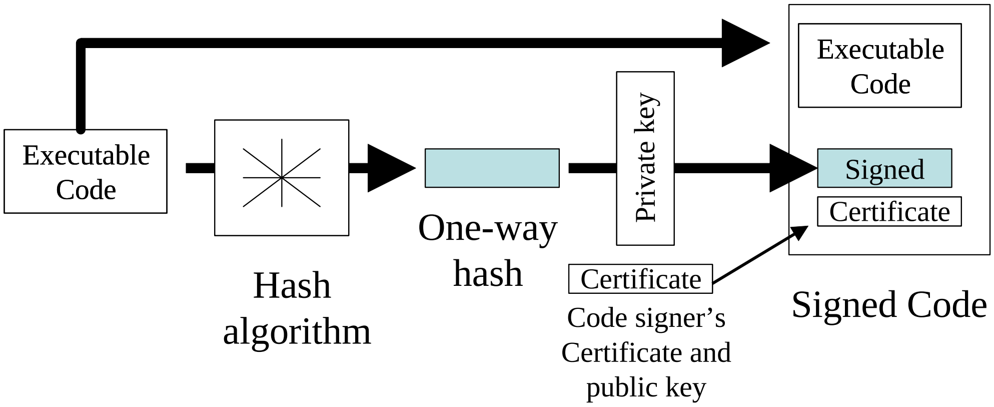


Figure 24. Code- and Document-Signing Process

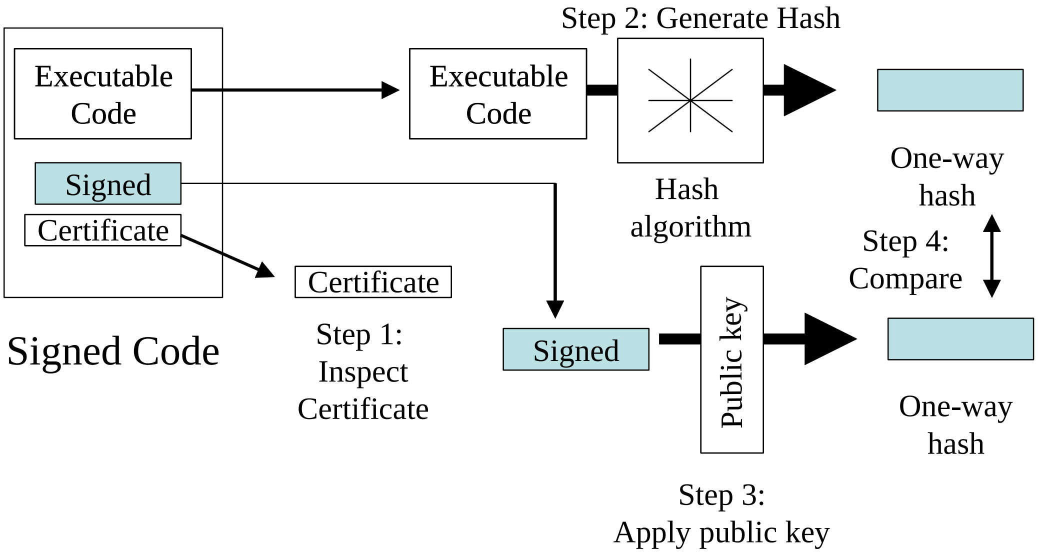


Figure 25. Code- and Document-Signing Verification Process

A document may also be signed and verified. In all cases, what is assured by code and document signing is the authorship, including the verification that third parties have not subsequently modified the code (or document). In no case does the user receive any assurance that the code itself is safe to run or actually does what it claims. Thus, the actual value of code signing remains a function of the reliability and integrity of the individual that signed that software and the processes that support software development and ongoing life cycle support. Code signing, therefore, is solely a mechanism for a software creator to assert the authorship of the product and validate that others have not modified it. It does not provide the end-user with any claim as to the code's quality, intent, or safety.

As mentioned in section 4.4, the higher assurance integrity entities (e.g., higher software levels) should not rely upon (human) administrative activity. Specifically, it should not be possible to misconfigure or mismanage high-assurance devices (including software) or systems.

6.1.2 Availability.

Availability issues directly impact the same three entities that were previously described for integrity:

- Adequate availability (or spare capacity) is needed for the physical network media that conveys data communications packets. Network availability can be attacked by causing the intermediate systems that forward packets to not function correctly or else by saturating the network so that entities that need to use it cannot do so. The latter is called a DoS attack, which leverages the fact that network capacity is a finite resource. The first threat can be reduced by deploying intermediate systems that cannot be misconfigured (i.e., are high assurance). DoS exploits can be reduced by ensuring that the capacity of the network exceeds the cumulative network use, either by rate limiting the devices that connect to the network or else by implementing other QoS techniques. If QoS is used, higher software levels applications (e.g., Level A and Level B) should preferentially leverage technology that is implemented so that it cannot be misconfigured.

- Availability of the security controls should be assured for a device that is used within a distributed system's defense-in-depth security protections. This requirement can be met by ensuring that defense-in-depth and control life cycle principals mentioned in section 5.1 are followed. Key system resources should also either have redundancies or else have fail-safe protections.

- Availability should be assured for the applications that support airborne operations. These devices need to be designed to be impervious to bad data. They also need to be designed to withstand repeated and prolonged attempted accesses by rogue processes or systems (e.g., DoS attacks).

Availability is traditionally addressed by using either real-time systems where information flows are predetermined and systems are preconfigured to guarantee delivery of critical information, or by QoS network capabilities. For safety systems, this property must be designed in mechanisms that must be provided to segregate non-real-time systems from real-time systems and techniques to assess interactions between systems that must be employed, in accordance with the rules that are articulated in section 8.2.

6.1.3 Authentication.

Authentication directly impacts the following entities:

- Communications protocols should be deployed with their security turned on (see section 4.5). This means that routing protocols should be configured to use the appropriate password and HMAC for that deployment. The password needs to be unique for that system and protected via best current practices password protection mechanisms. Mutual authentication should be used whenever possible. This implies that human users and devices should both be assigned an appropriate identity by the authentication system used by the deployment (e.g., Kerberos, PKI, [82]). This, in turn, implies that the best common practice for that authentication system should be followed.

- Devices (both end system and intermediate system) and software with higher safety requirements should be designed so that they cannot be misconfigured, including their naming (if any) and addressing assignments, if possible. Devices and applications with more modest safety requirements need to ensure that their administrators are authenticated and that administrative authorizations (including access control) are in accordance with the separation of duties with least privilege principals.

- Applications should ensure that their users (both processes and humans) are authenticated and, if applicable, their access control limited by separation of duties with least privilege. Authentication of human users should preferentially require two factored authentication (e.g., password plus PKI identity).

The ultimate goal of airborne security controls (including the authentication system) is to prevent safety failures. Physical techniques, along with policies and procedures, should be considered where practical. Remote access to safety-critical components should be minimized; however, where they are justified, authentication must be required.

Authentication of airborne entities would be materially strengthened if the airborne authentication system were a constituent part of the same integrated authentication infrastructure serving both airborne and NAS systems. A number of candidate technologies could serve as the basis for such an authentication infrastructure. The requirements of such an infrastructure are that a common identity system needs to be created system-wide for humans and devices that populate the total system. These identities need to be authenticated by means of a common technology infrastructure in accordance with best IA practices. The authentication system may or may not also be associated with authorization or access control. Well-known candidates for authentication systems include PKI (see RFC 3280, RFC 4210, and RFC 3494), Kerberos (see RFC 4120); Remote Authentication Dial In User Service (see RFC 2138 and RFC 3580); and Authentication, Authorization, and Accounting (see RFC 3127 and RFC 3539) including Diameter (see RFC 3588 and RFC 4005). References 79 and 82 describe a PKI-based authentication system for the ATN. A choice of PKI to become an avionics authentication infrastructure correlates well with the extensive DoD PKI infrastructure that is currently being built by the DoD to support PKI within DoD systems.

6.1.4 Confidentiality.

Confidentiality is generally not relevant for safety (see appendix B for a discussion to the contrary). While there are some scenarios where the real-time location of an airplane might become known to an adversary and conceivably put the plane in jeopardy, this threat has not become widely accepted within the FAA. The flight paths of commercial airplanes are already known, and the real-time information would have a short lifespan for an attacker. In this example, old data is of little value to the attacker in general.

6.1.5 Nonrepudiation.

With regards to digital security, nonrepudiation means that it can be verified that the sender and the recipient were, in fact, the parties who claimed to send or receive the message, respectively. Nonrepudiation of origin proves that data has been sent, and nonrepudiation of delivery proves it has been received. Digital transactions are potentially subject to fraud, such as when computer systems are broken into or infected with Trojan horses or viruses. Participants can potentially claim such fraud to attempt to repudiate a transaction. To counteract this, the underlying processes need to be demonstrably sound so that such claims would not have credence. Logging of significant events is needed to create accountability. Log files should be protected from being modified or deleted.

Nonrepudiation should be a required security attribute for all electronic parts distribution systems (e.g., software distribution). All electronic parts need to be signed in accordance with the U.S. Federal DSS [81] in accordance with an FAA-approved electronic distribution system. The source and integrity assurance of an electronic part is a critical element of verifying its authenticity prior to installation. This signature needs to be checked and verified at the deployment site before any electronic part can be deployed. The checks verify that the software has not been modified subsequent to being signed. The identity of the signer needs to be authenticated and authorized previous to deployment.

In addition, whenever administrators (both device and human) interact with aviation equipment or administer devices within aircraft, a log of their activity should be kept for analysis, accountability, and administrative purposes (e.g., fault investigation). The log file needs to record the specific identity of the human responsible, the time, actions performed, as well as optionally the location from which the access occurred. This log needs to be protected from subsequent modification or deletion. If network or host intrusion detection systems (IDS) are deployed, these log files should be available for those systems to read.

6.2 EXTENDING FAA ORDERS, GUIDANCE, AND PROCESSES INTO VAST NETWORK SYSTEMS.

Different communities use different terms to refer to the same or similar concepts. For example, it was previously mentioned that current FAA safety assurance processes for airborne systems are based on ARP 4754, ARP 4761, and ACs (e.g., AC 25.1309-1A and AC 23.1309-1C); software assurance is based on DO-178B; and complex electronic hardware design assurance is based on DO-254. These references reflect common FAA parlance that speaks about the laws, orders, guidance, and processes, which govern the civil aviation community by using those terms. However, in the parlance of the security community, laws, orders, guidance, and processes are referred to as being policy. Consequently, ARP 4754, ARP 4761, DO-178B, DO-252, and the ACs are referred to as being FAA safety policies. This point is mentioned because the following quotation is taken from the security community.

> "An important concept in the design and analysis of secure systems is the security model, because it incorporates the security policy that should be enforced in the system. A model is a symbolic representation of policy. It maps the desires of the policy makers into a set of rules that are to be followed by a computer system. … A security model maps the abstract goals of the policy to information system terms by specifying explicit data structures and the techniques necessary to enforce the security policy. A security model is usually represented in mathematics and analytical ideas, which is then mapped to system specifications, and then developed by programmers through programming code. … Formal security models, such as Bell-LaPadula are used to provide high assurance in security... A security policy outlines goals with no idea of how they would be accomplished and a model is a framework that gives the policy form and solves security problems for particular situations." (Quoted from reference 83 pages 239-240.)

It is important that the civil aviation community understand the intended meaning of this quotation (i.e., that differences in terminology not cause misunderstanding).

Therefore, using security community terminology, ARP 4574 and DO-178B reflect FAA policy for airborne software. Other entities (e.g., the DoD) have articulated other policy systems. Security models exist to provide a mathematical foundation by which well-defined policy systems (such as the DoD or the FAA) can be extended into arbitrarily complex and vast networked environments and still retain their original policy viability in a mathematically demonstrable manner. The goal of this section is to explain the technical foundation for this study's recommendation for how to extend the current certification safety processes (e.g., ARP 4574 and DO-178B safety policy) into arbitrarily large networked system environments by means of the Biba Integrity Model.[27]

The Bell-LaPadula Confidentiality Model [84] was developed to formalize the U.S. DoD's multilevel security policy. It forms the framework for confidentiality within the Federal government's information processing, including the DoD's COMSEC policy. This model creates a multilevel security policy system by means of mandatory access controls that label data at a specific classification level, and provide users clearances to a specific classification level. The controls ensure that users cannot read information classified at a security level higher than

[27] This quotation consistently refers to security policy. This is because the context from which this quotation was taken was talking about security policy. The system (i.e., policy vis-à-vis security model) is not dependent upon whether the operative policy is a security or a safety policy. Rather, the operative concept is that it is a well defined policy within the security domain. As was previously stated, airborne safety is within the security domain whenever it pertains to networked environments.

their own classification level nor can they write information to a lower classification level, except via the controlled intervention by a trusted subject (e.g., HAG).

The Bell-LaPadula framework is realized within military communications by creating networks, each operating at a specific classification level. These networks can operate as MSLS (see section 5.2) systems[28] or as DoD networks operating at system high, where the network is classified at the highest classification level of the data it conveys. For example, a system-high secret network could transmit secret information as well as information classified below the secret level (e.g., SBU information and unclassified information), but not information at a higher classification level than secret.

DoD networks operating at different classification levels are orthogonal to each other. For example, they are addressed, by definition, from address and naming spaces that pertain to their classification level. This results into network systems having distinct (i.e., unrelated) IP addresses and naming spaces than networks operating at other classification levels in general.

> "The Bell-LaPadula model is built on the state machine concept. This concept defines a set of allowable states (A_i) in a system. The transition from one state to another upon receipt of an input(s) (X_j) is defined by transition functions (f_k). The objective of this model is to ensure that the initial state is secure and that the transitions always result in a secure state.
>
> The Bell-LaPadula Confidentiality Model defines a secure state through three multilevel properties. The first two properties implement mandatory access control, and the third one permits discretionary access control. These properties are defined as follows:
>
> 1. *The Simple Security Property (ss Property).* States that reading of information by a subject at a lower sensitivity level from an object at a higher sensitivity level is not permitted (no read up).
>
> 2. *The * (star) Security Property,* also known as the confinement property. States that writing information by a subject at a higher level of sensitivity to an object at a lower level of sensitivity is not permitted (no write down).
>
> 3. *The Discretionary Security Property*. Uses an access matrix to specify discretionary access control." (Quoted from page 202 of reference 85.)

The Bell-LaPadula Confidentiality Model, therefore, creates access control protections between entities at different sensitivity levels. These sensitivity levels are the DoD classification levels (see section 6.3). A weakness of the Bell-LaPadula Confidentiality Model is that it only deals with confidentiality of classified material. It does not address integrity or availability—the key

[28] Other possibilities also exist, including multiple levels of security and multiple independent levels of security. However, the goal of this paragraph is to contrast MSLS with system high because that contrast is relevant to subsequent airborne network policy issues discussed in section 8.2.

security issues that potentially impact safety. Figure 26 displays and compares how the Bell-LaPadula Confidentiality and Biba Integrity Models operate.

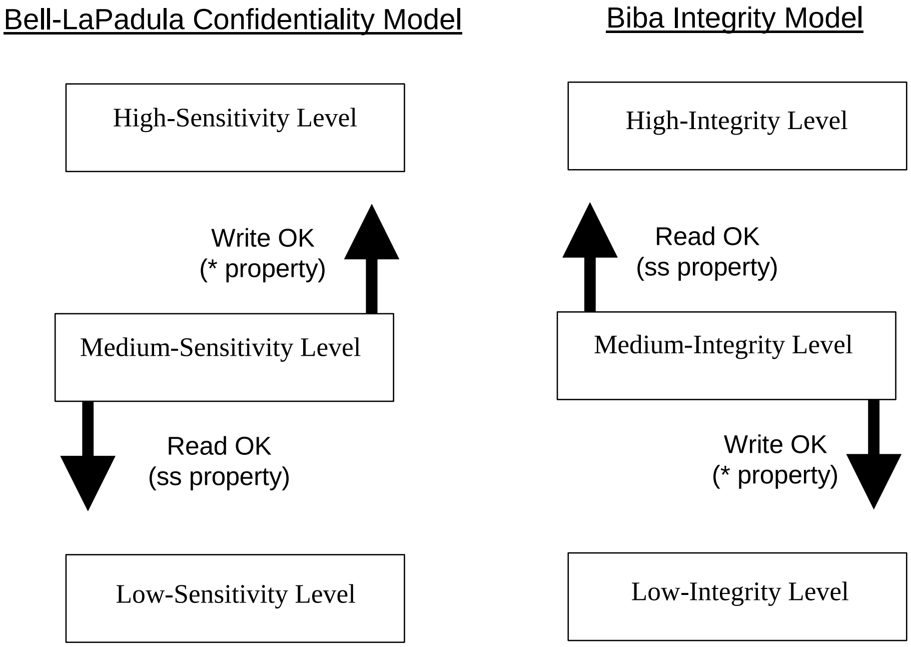


Figure 26. Bell-LaPadula Confidentiality and Biba Integrity Models Compared

The Biba Integrity Model was created as a direct analog to the Bell-LaPadula Confidentiality Model to address integrity issues. Specifically, integrity is usually characterized as comprising the following three goals (taken from page 204 of reference 85):

- The data or system is protected from modification by unauthorized users or processes.
- The data or system is protected from unauthorized modification by authorized users or processes.
- The data or system is internally and externally consistent. For example, the data held in a database must balance internally and must accurately correspond to the external, real-world situation that it represents.

These integrity issues directly correspond to the safety policy concerns that DO-178B and ARP 4754 address.

The Biba Integrity Model shares the same concepts as the Bell-LaPadula Confidentiality Model, except that its mandatory policies are the inverse of each other (see figure 26). The Biba Integrity Model is Lattice-based and uses a lattice structure that represents a set of integrity classes and an ordered relationship among those classes such as the DO-178B levels of safety (see section 6.3). The Biba simple integrity axiom (ss) requires that a subject at one level of integrity is not permitted to observe (read) an object at a lower level of integrity (no read down). The Biba * (star) Integrity axiom requires that an object at one level of integrity is not permitted to modify (write to) an object of a higher level of integrity (no write up), thereby preserving the higher level of integrity. As with the Bell-LaPadula Confidentiality Model, a subject at one level of integrity cannot invoke a subject at a higher level of integrity.

Also similar to the Bell-LaPadula Confidentiality Model, the Biba Integrity Model has provisions for HAGs, which enable highly controlled functions to occur that would have otherwise been prohibited by the model. HAGs are trusted subjects that operate in a highly controlled and highly localized manner. However, in the Biba case, the HAG is concerned with integrity issues that permit a highly trusted integrity environment to safely receive communication from a less trusted one in a highly controlled way. For example, a HAG might be inserted into the network to support a Level C software system that needs to communicate with a Level A software system.

6.3 COMPARING CIVILIAN AIRCRAFT SAFETY AND FEDERAL GOVERNMENT SECURITY LEVELS.

6.3.1 Civil Aircraft Software Levels.

The civilian aircraft industry’s DO-178B software levels are:

> “based upon the contribution of software to potential failure conditions as determined by the system safety assessment process. The software level implies that the level of effort required to show compliance with certification requirements varies with the failure condition category.” (Quoted from Section 2.2.2 of reference 5.)

DO-178B defines the following specific failure condition categories.

> “The categories are:

a. Catastrophic: Failure conditions which would prevent continued safe flight and landing.

b. Hazardous/Severe-Major: Failure conditions which would reduce the capability of the aircraft of the ability of the crew to cope with adverse operating conditions to the extent that there would be:

(1) a large reduction in safety margins or functional capabilities,

(2) physical distress or higher workload such that the flight crew could not be relied on to perform their tasks accurately or completely, or

(3) adverse effects on occupants including serious or potentially fatal injuries to a small number of those occupants.

c. Major: Failure conditions which would reduce the capability of the aircraft or the ability of the crew to cope with adverse operating conditions to the extent that there would be, for example, a significant reduction in safety margins or functional capabilities, a significant increase in crew workload or in conditions impairing crew efficiency, or discomfort to occupants, possibly including injuries.

d. Minor: Failure conditions which would not significantly reduce aircraft safety, and which would involve crew actions that are well within their capabilities. Minor failure conditions may include, for example, a slight reduction in safety margins or functional capabilities, a slight increase in crew workload, such as, routine flight plan changes, or some inconvenience to occupants.

e. No Effect: Failure conditions which do not affect the operational capability of the aircraft or increase crew workload." (Quoted from Section 2.2.1 of reference 5.)

In addition to the safety definitions, all software involved in civil aircraft systems is also assigned a level, depending upon the software causing or contributing to potential failure conditions as determined by the system safety assessment process (e.g., ARP 4754 and ARP 4761).

> "The software level implies that the level of effort required to show compliance with certification requirements varies with the failure condition category. The software level definitions are:
>
> a. Level A: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft.
>
> b. Level B: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a hazardous/severe-major failure condition for the aircraft.
>
> c. Level C: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a major failure condition for the aircraft.
>
> d. Level D: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function resulting in a minor failure condition for the aircraft.
>
> e. Level E: Software whose anomalous behavior, as shown by the system safety assessment process, would cause or contribute to a failure of system function with no effect on aircraft operational capability or pilot workload. Once

> software has been confirmed as level E by the certification authority, no further guidelines of this document apply." (Quoted from Section 2.2.2 of reference 5.)

6.3.2 Federal Government Security Classifications.

The Federal Government security classification system is codified in Executive Order 12958, "Classified Nation Security Information" [86], and Executive Order 13292 [87], "Further Amendment to Executive Order 12958." Software and data are protected based upon their degree of sensitivity as measured by how much damage the release of information could cause to national security. The Executive Order defines the following levels of security and their impact on national security.

"Information may be classified at one of the following three levels:

a. Top Secret shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause exceptionally grave damage to the national security that the original classification authority is able to identify or describe. It merits the highest level of protection.

b. Secret shall be applied to information, the unauthorized disclosure of which reasonable could be expected to cause serious damage to the national security that the original classification authority is able to identify or describe.

c. Confidential shall be applied to information, the unauthorized disclosure of which reasonably could be expected to cause damage to the national security that the original classification authority is able to identify or describe." (Quoted from Section 1.3 of reference 87.

The Executive Order also specifies that the information systems containing this information have controls that prevent access by unauthorized persons and ensure the integrity of the information.

In addition to the Executive Order, the Computer Security Act of 1987 (PL-100-235) [88] established requirements for protection of certain information on Federal Government computer systems. It also defined an addition information classification, SBU.

> "Any information the loss, misuse, or unauthorized access to or modification of which could adversely affect the national interest or the conduct of Federal programs or the privacy to which individuals are entitled under [the Privacy Act of 1974] but which has not been specifically authorized under criteria established by an Executive Order or an Act of Congress to be kept secret in the interest of national defense or foreign policy." [88]

As a result, any information that violates privacy of an individual or that is controlled from export to foreign nations can fall into the SBU category.

The corporate world also has classification systems. Each company has its own rules for the protection of information depending upon its sensitivity level to intellectual property, business strategy, and other similar areas. In many respects, the corporate classification system follows the federal government system. As an example, there might be three levels of information control within a corporation where the highest is limited and cannot be released outside of the company. Proprietary information can be released but only to those individuals and companies bound by a signed nondisclosure agreement. Finally, any information not covered by the above can be released outside the company within the restrictions placed by the federal government. In addition, International Traffic in Arms Regulations [89] controlled information, although unclassified and nonproprietary, must still be controlled to prevent disclosure to foreign nationals unless an appropriate export license has been obtained.

6.3.3 Comparison of the Two Policy Systems.

It is clear the FAA and civil aviation are concerned about airplane safety and so they define airborne software in terms of the possible safety affects of software failure conditions. The Federal Government, which includes the DoD, is concerned about protection of sensitive information and programs. It defines its software systems in terms of the impact of that software upon the protection of sensitive information and programs. Although the focus on what is being protected against is entirely different between these two policy systems, the intent of the protection mechanisms are similar. Both enforce restrictions on how software operates within its system context. Both are also concerned with the impact of protection mechanisms and the consequences of possible failure affects. Both define their assurance system in terms of the worst-case affects of failure conditions. Coincidentally, both assurance systems are also remarkably similar to each other when viewed at a high level of abstraction, as shown in table 2.

Table 2. Comparison of Safety Levels to Security Classifications

Safety	Security
Level A (catastrophic condition)	Top Secret (exceptionally grave damage)
Level B (hazardous/severe-major condition)	Secret (serious damage)
Level C (major condition)	Confidential (damage)
Level D (minor condition)	Sensitive but Unclassified (could adversely affect)
Level E (no-effect condition)	Unclassified (no effect)

Therefore, although the civil aviation and federal government systems are distinct systems from each other and are oriented around very different issues, they nevertheless share important attributes. Additional similarities and differences between the two systems include the following:

- Only the security side is concerned with confidentiality issues—this issue is briefly discussed in section 6.1.4.

- Both safety and security are concerned with integrity issues. Once the programs and data are certified to be correct and operating correctly, any unauthorized changes could result in anomalous behavior. If a software item is evaluated to be at Level E, this unauthorized modification may only be a nuisance at worse. However, as analogous to highly sensitive federal government information, an unauthorized modification to a Level A- or B-rated software may have serious or disastrous results.

- Both safety and security are concerned with availability. If flight-critical software on an aircraft is not available when needed, catastrophic results can occur. Likewise, if highly critical and time-sensitive information owned by the federal government is not available, the latest information may not be available during mission planning, potentially resulting in loss of life.

- Both safety and security should be concerned with authentication and authorization. Without knowledge of who is attempting to access the software or data, modifications could be made by unauthorized personnel. If malicious, the unauthorized changes could, potentially, cause catastrophic results.

- Nonrepudiation is predominately in the security domain. From a security point of view, nonrepudiation provides the capability to ensure that any actions cannot be later denied (e.g., ensures the validity of audit information).

In conclusion, safety and security, although they have some differences in protection requirements, also have many similar requirements. The levels defining the criticality of the software and data in both domains have parallels that can help in determining the safety of onboard networks.

6.4 BIBA INTEGRITY MODEL AND BELL-LAPADULA CONFIDENTIALITY MODEL ARE DIRECT ANALOGS.

If the FAA were to adopt the Biba Integrity Model for ensuring the safety of networked airborne and NAS systems in accordance with existing DO-178B and ARP 4754 safety policies, then the resulting system could look very much like the current DoD system (see section 5.2). This similarity is directly due to the Bell-LaPadula Confidentiality and Biba Integrity Models being a direct analog of each other, built upon the same state machine concepts. The prime differences would be:

- The FAA system is based upon FAA safety policies, while the DoD system is based upon DoD confidentiality policies.

- The mandatory policies of the Biba Integrity Model are the direct inverse of the mandatory policies of the Bell-LaPadula Model (see figure 26).

The affects of the two models are directly parallel. However, the fact that both the resulting FAA and DoD systems contain a five-level assurance system is not an artifact of either the Bell-LaPadula Confidentiality or the Biba Integrity Models. It rather reflects the coincidence that both the DoD and FAA policy systems proscribe five distinct classification levels (see section 6.3.3).

Both models partition networked items into distinct network systems that operate at a specific assurance level. In the civil aviation system, this level is in accordance with DO-178B and ARP 4754 policy. In the DoD system, it is in regard to confidentiality levels articulated by federal law. Regardless, distinct networked systems operating at known levels are created.

- DO-178B systems using the Biba Integrity Model can also be deployed in terms of system-high network groupings, just like DoD systems can. However, it differs from DoD systems in that the system high for the Biba Integrity Model is in terms of the lowest integrity classification for that common grouping (i.e., it is actually a system low, since the mandatory policies of the Biba Integrity Model are the inverse of the Bell-LaPadula Confidentiality Model).

- DO-178B systems using the Biba Integrity Model can also be partitioned into MSLS systems, each operating at a specific safety classification only, in a parallel fashion to DoD systems.

- Network partitioning in terms of the Biba Integrity Model is recommended to occur by means of civilian VPN technologies (see section 5.6) although the military COMSEC equivalent could be used. Specifically, this study recommends that Biba Integrity Model partitioning is accomplished by IPsec’s ESP in tunnel mode (RFC 4301 defines IPsec in tunnel mode, and RFC 4303 defines the ESP protocol). This bullet is very important to the extent that the following paragraphs provide further elaborations.

Section 5.4.1.1 of ARP 4754 discusses mechanisms to partition highly integrated or complex aircraft systems. Both the Bell-LaPadula Confidentiality and Biba Integrity Models explicitly rely upon similar partitioning techniques. In IP environments, VPNs permit the creation of a networked system that operates at a specific assurance level within a larger context of a total system that cumulatively operates at many different assurance levels. VPNs specifically enable associated networked items to become partitioned to operate at a trusted specific assurance level that is potentially a different assurance level than the underlying physical network itself (e.g., the LAN), as well as different from the other VPNs (and their networked items), which are also similarly supported by that same physical network.

The current U.S. DoD networking environment was described in section 5.2. Figure 17 showed how DoD COMSEC is currently also based upon IPsec’s ESP in tunnel mode. Section 5.6 described how the highly secured civilian VPN alternatives can similarly be based on ESP in tunnel mode. When DO-178B and ARP 4754 safety policies are organized according to the Biba Integrity Model, these same DoD COMSEC and industry VPN concepts can be applied to airborne and NAS safety deployments. Figure 18 shows those security concepts (i.e., DoD Classifications in a Bell-LaPadula Confidentiality Model and its resulting COMSEC articulations) in a DoD environment having aircraft. Figure 27 shows those same concepts applied to DO-178B safety classifications using the Biba Integrity Model. Specifically, this figure shows the Biba Integrity Model elements applied as MSLS networks.

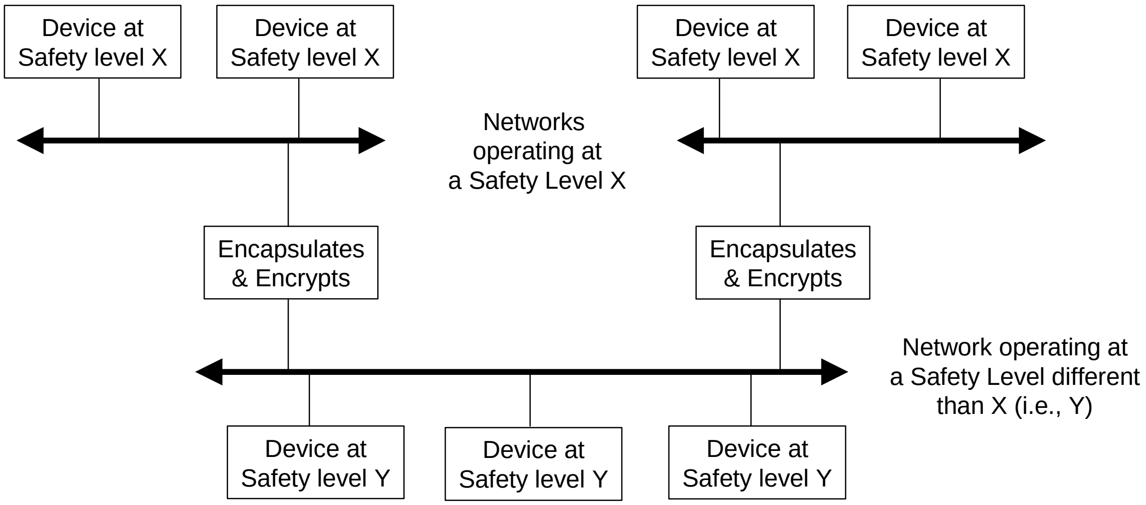


Figure 27. DO-178B Classifications Using Biba Integrity Model

Figure 27 shows devices operating at safety classification X (e.g., either level A, B, C, D, or E). These devices operate within a network (e.g., a VPN) functioning at that specific safety classification level. Network partitioning in terms of safety classifications may implicitly involve data categorization to the extent that data is directly related to safety distinctions. Figure 27 shows that those networks operating at the same safety level may be discontinuous. For example, the items located at the top left need to communicate with the items located at the top right, and vice versa. These discontinuous network segments can be joined by a different network system operating at a different safety level through encrypting the original packets and encapsulating them into the protocol headers of the lower network system (see figure 17). The top networks in figure 27 are the customer site networks mentioned in figure 20. It is a RED (plain text) network. The bottom (linking) network is the service provider network mentioned in figure 20. It is a BLACK (cipher text) network—although, as a point of fact, it almost certainly conveys plain text packets that are operational at its own classification level. The encapsulation and encryption is performed in accordance with IPsec’s ESP in tunnel mode, which is the “encapsulates and encrypts” function shown within figure 27. That function is also the “interface” described in figure 20. The stack chart of the packets from the top network system (operating at safety level X) appears as is shown in figure 22, when they are conveyed over the bottom network system of figure 27 (operating at safety level Y). Consequently, one can see this approach corresponds to both DoD COMSEC and industry VPNs.

VPN encryption should use FIPS compliant encryption algorithms. Protocol encapsulation ensures that these are logically distinct network systems that are unable to address or interwork with different logical network systems operating at different safety levels except at the encapsulation and encryption interface. However, since each interface is specialized to only one VPN instance (i.e., it physically cannot support multiple RED VPN systems), confusion between VPNs cannot occur. This is true regardless of whether or not these networks have physically distinct media systems. Specifically, figure 27 can be interpreted as showing interconnected networks having three distinct physical media instances (top left, top right, bottom), with the top two physical media systems operating at the same safety level that is a different safety level than the bottom network system. However, figure 27 can also be interpreted as showing a network having the same ubiquitous physical media subdivided into logically different network elements. In the latter case, the top left, top right, and bottom all use the same physical media. In this case, different logical network systems, each having effective network security and isolation through protocol encapsulation, have been created from the same physical system. This latter observation is directly applicable to aircraft systems sharing a common LAN system. That is, COMSEC and VPN techniques permit the creation of partitioned network systems even when sharing a common physical network.

This study recommends using the Biba Integrity Model to extend current FAA policies into arbitrarily complex networked environments because it is a formal model on the par with the DoD's Bell-LaPadula Confidentiality Model and also because it creates structures that are the direct analog of the Bell-LaPadula Confidentiality Model. Other security models are also available, including other integrity models (e.g., the Clark-Wilson Integrity Model). Similarly, the FAA could invent a security model of its own, including performing the necessary mathematical proofs. Any of these are valid alternatives for the FAA to consider. What is not a valid alternative is to attempt to extend ARP 4754 into networked environments without using a viable formal mathematical model (e.g., a security model) of some sort. Any such extension would necessarily be ad hoc and produce results that cannot be trusted to be safe.

6.5 RELATING SAFETY CLASSIFICATION LEVELS TO THE CC.

The exemplar network architecture described in section 8.3 relies upon security controls (e.g., firewall, packet filter, ASBR, VPN encapsulation gateways, and HAGs) to provide security protections for the networked system so that the resulting system is assured to operate at a specific safety level. Section 6.4 explained that airborne networks operate at specific safety levels as defined by FAA policy (e.g., DO-178B and ARP 4574) and enforced by the Biba Integrity Model. Therefore, for certification purposes, the integrity of these security controls must be mapped to the appropriate DO-178B safety level. This implies that these security controls can be evaluated in terms of specific DO-178B safety level assurances for the Biba Integrity Model provisions to be viable. This section discusses this issue.

The FAA has sponsored a growing body of work evaluating common security and safety processes and systems [41 and 72, 73, and 76]. This issue directly impacts aircraft that need to be dual certified by both the FAA (for safety) and DoD (e.g., the U.S. Air Force; for security). However, this issue is also of a more generic interest. For example, the DoD, in addition to defining their information systems in accordance with security (i.e., confidentiality in particular) constructs, is also concerned with safety issues, which are defined in terms of MIL-STD-882D [90]. MIL-STD-882D shares many similarities with existing civil aviation concepts including a similar safety five-level classification system.

Although safety and security are very distinct concepts, they share some common attributes that permit them to be compared (and equated) in several different ways. For example, the FAA and the DoD have created comparable certification environments having similar concepts of assurance. Both safety and security also have similar integrity attributes that may be leveraged in a Biba Integrity Model environment to provide a mechanism to relate otherwise dissimilar safety and security concepts. Both approaches will be considered in this section.

Department of Defense Instruction (DoDI) 8500.2 Enclosure 4 [91] provides specific guidance to DoD systems on how to identify specific CC (security) protection profiles. While there are many details associated with this process, the issues examined in DoDI 8500.2 Enclosure 4 are particularly relevant for FAA consideration. This is because while the DoD itself is primarily oriented upon confidentiality issues, which have little or no safety consequence, Enclosure 4 focuses on availability and integrity, which are the security concepts that are the most centrally relevant to airborne safety in networked environments (see section 6.1). For example, "the FAA often considers data integrity and availability among the most important" security services (quoted from page 1 of reference 20). The following are direct quotations from DoDI 8500.2 Enclosure 4:

> "The IA Controls provided in enclosure 4 of this Instruction are distinguished from Common Criteria security functional requirements in that they apply to the definition, configuration, operation, interconnection, and disposal of DoD information systems. They form a management framework for the allocation, monitoring, and regulation of IA resources that is consistent with Federal guidance provided in OMB A-130 [see [92]]. In contrast, Common Criteria security functional requirements apply only to IA & IA-enabled IT products that are incorporated into DoD information systems. They form an engineering language and method for specifying the security features of individual IT products, and for evaluating the security features of those products in a common way that can be accepted by all." (Quoted from E3.4.3 of reference 91.)
>
> "This enclosure [i.e., Enclosure 4 within [91]] establishes a baseline level of information assurance for all DoD information systems through the assignment of specific IA Controls to each system. Assignment is made according to mission assurance category and confidentiality level. Mission assurance category (MAC) I systems require high integrity and high availability, MAC II systems require high integrity and medium availability, and MAC III systems require basic integrity and availability. Confidentiality levels are determined by whether the system processes classified, sensitive, or public information. Mission assurance categories and confidentiality levels are independent, that is a MAC I system may process public information and a MAC III system may process classified information. The nine combinations of mission assurance category and confidentiality level establish nine baseline IA levels that may coexist within the GIG. See Table E4.T2. These baseline levels are achieved by applying the specified set of IA Controls in a comprehensive IA program that includes acquisition, proper security engineering, connection management, and IA administration as described in enclosure 3 of this Instruction." (Quoted from E4.1.1 of reference 91.)

The DoDI 8500.2 Enclosure 4 MAC is defined by the intersection of integrity and availability (the MAC level) and DoD security classifications (the confidentiality attribute for each MAC level). This pairing potentially provides a framework for considering FAA and the CC processes and concepts in an integrated manner. Specifically, it is conceivable that the modest FAA confidentiality requirements (if any) roughly equate to the DoD public (i.e., basic) confidentiality level, such that the DO-178B software levels can be mapped into the public variant of the three different MAC levels to identify IA (i.e., security) requirements for FAA systems. Of course, since DoDI 8500.2 is a DoD document, this association is in terms of DoD processes, and not FAA processes. However, it does provide a possible intersection that may be relevant for increased synergy between the DoD and FAA.

Therefore, DoDI 8500.2 may provide a starting point for potentially integrating airborne network safety and security concepts into a common federal system by leveraging established DoD processes that comply with federal law. Nevertheless, to pursue this, the FAA needs to study and verify whether the three MAC levels identified by DoDI 8500.2 provide adequate granularity for the NAS and airborne system requirements. If they do, then the FAA could potentially directly leverage current DoD processes, if appropriate, perhaps creating a government-wide integrated safety and security engineering system.

Regardless, this study concludes that this issue needs further study to be useful. Consequently, at this time, it does not provide the assurances needed to underlie the exemplar airborne network architecture. Therefore, this study will tentatively relate safety and security issues in terms of the relative assurances provided by their respective certification processes.

The CC has provided seven predefined security assurance packages, on a rising scale of assurance, which are known as evaluation assurance levels (EAL). EALs provide groupings of assurance components that are intended to be generally applicable. The seven EALs are as follows:

- EAL 1—Functionally Tested
- EAL 2—Structurally Tested
- EAL 3—Methodically Tested and Checked
- EAL 4—Methodically Designed, Tested, and Reviewed
- EAL 5—Semiformally Designed and Tested
- EAL 6—Semiformally Verified Design and Tested
- EAL 7—Formally Verified Design and Tested

EAL 1, therefore, is the entry level classification of the system. EAL 1 through EAL 4 (inclusive) are expected to be generic commercial products. EAL 5 through EAL 7 (inclusive) are considered to be high-assurance products.

Carol Taylor, Jim Alves-Foss, and Bob Rinker of the University of Idaho have studied the issue of dual software certification [93] for CC and DO-178B. Figure 28 is copied from their study and shows a high-level gap analysis between the CC classes and the DO-178B processes. Their study provided a fairly detailed analysis of the differences. Their study suggested that security functionality certified at EAL 5 can be directly compared with DO-178B Level A.

Common Criteria Classes	DO-178B Processes
ACM—Configuration Management	Software Configuration Management
ADO—Deliver and Operation	<no correspondence>
ADV—Development Software	Software Development Process
AGD—Guidance Documents	<no correspondence>
ALC—Life Cycle Support	Software Planning Process
ATE—Tests Software	Verification Process
AVA—Vulnerability Assessment	<no correspondence>
<no correspondence>	Software Quality Assurance

Figure 28. Gap Analysis in the Alves-Foss, et al. Study [93]

The study recommends the basis for equivalency between the integrity of security controls and DO-178B safety levels should be confirmed by further study. However, in the interim, the FAA can leverage the University of Idaho results to temporarily equate the assurance of security systems certified at the CC's EAL 5 with airborne software certified at DO-178B Level A. This means that security controls deployed on aircraft that support DO-178B Level A software currently must be certified at CC EAL 5 or higher.[29]

7. EXTENDING FAA CERTIFICATION TO AIRBORNE NETWORKS.

The previous sections discussed the issues that underlie how FAA certification assurance could be extended to airborne network environments. The fundamental certification issue is that when airborne software becomes deployed in a network environment, the risks and dangers of the network environment need to be mitigated. Airborne network environments are inherently different than historic ARP 4754 environments for the reasons that were previously introduced in section 3. Section 6 discussed the foundational certification issues associated with formally extending DO-178B and ARP 4754 policies by means of the Biba Integrity Model into airborne network environments. The purpose of this section is to provide greater details as to how specifically ARP 4754 (section 7.1) and DO-178B (section 7.2) processes should be extended to handle airborne network deployments.

A presupposition of this study is that all airborne entities that are currently assured to DO-178B criteria or ARP 4754 guidance will need to become re-evaluated if hosted within a networked airborne environment. Unless these entities are re-evaluated in the context of the networked environment, their security provisions and the safety of the resulting system would be indeterminate.

[29] This section concludes that until more definitive studies are conducted, security controls that support Level A software should be certified at CC EAL 5 or higher. Please note that this is regarding security controls, not airborne software. Specifically, this study recommends that airborne software should continue to be ensured by using FAA processes rather than in terms of CC concepts. Please note that EAL 5 is the lowest of the CC's high assurance levels. Few COTS products in the general case are currently certified at EAL 5 or above. While this should not be problematic for firewalls or HAGs (other than the fact there are few if any Biba Integrity Model HAG products today), it may be problematic for routers.

7.1 EXTENDING ARP 4754 INTO NETWORKED ENVIRONMENTS.

The primary differences between networked airborne environments and the highly integrated or complex aircraft systems for which ARP 4754 was designed is:

- The inadvertent integration between all networked entities, including possibly subtle fate sharing relationships.

- Networks are inherently hostile environments where any software bug may be attacked and potentially leveraged to corrupt or compromise that item. If compromised, the item may potentially be used to attack other networked items or their common network environment.

There are two primary changes that are needed to extend ARP 4754 to address the challenges that occur within networked environments:

- ARP 4754 itself needs to become enhanced by the application of a security model so that the current ARP 4754 concepts could be assured to be extended in a mathematically viable manner into networked environments. This study recommends that ARP 4754 become extended by leveraging the Biba Integrity Model.

- Strategic security controls need to become introduced into an ARP 4754 deployment to provide IA protections that mitigate or reduce the efficacy of networked attacks, including restricting access to unauthorized humans and devices. As previously stated, these IA controls need to comply with best common IA practice, which is defined by the NSA's IATF [50]. These controls need to be implemented in accordance with defense-in-depth practices, which were discussed in section 5.1. Section 8.2 will apply best current SSE practices to the combination of current FAA safety assurance policies and Biba Integrity Model concepts to define the rules and relationships that underlie this study's recommended exemplar airborne network architecture, which is presented in section 8.3. Section 8.3, therefore, will discuss each of a minimal subset of security controls that are needed in airborne networked environments, including their recommended configurations to achieve a minimal set of defense-in-depth protections.

These two primary changes produce at least two secondary effects, which are also a component part of extending safety policy for networked environments. The first of these secondary effects is the need to require viable software life cycle integrity protections as an ARP 4754 system requirement. There are two constituent aspects for creating software integrity:

- Loading software onto aircraft needs to occur within a secure FAA-approved software download system. (Please see FAA Order 8110.49 chapters of field-loadable software.) This system needs to ensure that only the correct versions of the correct software are loaded into aircraft. This implies that a reliable mechanism of creating software and software updates is defined that includes a mechanism to securely store software within an authoritative ground-based software storage facility. Assured software versioning mechanisms and processes need to be established that provide nonrepudiation assurances (see section 9.10). A secure mechanism to associate software versions with appropriate target devices within aircraft also needs to be established that has viable integrity and nonrepudiation attributes. The software that is stored within the authoritative storage facility needs to be digitally signed in accordance with the U.S. Federal DSS (FIPS Publication 186) by an individual authorized to sign aircraft software. The secure software download system also includes provisions to ensure that mandatory onboard aircraft procedures verify that the received software has been signed by an authorized individual and that the software has not been modified subsequent to signing (i.e., software integrity and authorization protections) as a prerequisite for deploying the software within aircraft.

- Software, after it has been securely installed upon aircraft, must undergo frequent (e.g., potentially several times an hour) integrity verification checks to verify that the currently installed software is what it purports to be and that it has not been clandestinely replaced by a Trojan horse or other unauthorized software variant. There are a number of mechanisms by which such tests may be accomplished, including Tripwire mechanisms [94]. It is important that the onboard integrity verification procedures themselves be designed to be as impervious as possible to compromise from network attacks.

The second secondary effect is to supplement the current certification policy by introducing a wide range of penetration tests upon the actual completed airborne system. These tests should systematically address the capabilities of the network airborne deployment system under evaluation, which includes its security controls, to withstand the range of attack vectors that are described in appendix A. These tests will hopefully identify latent vulnerabilities within the proposed networked system itself that need to be fixed as a condition for becoming approved. While such testing cannot provide assurance guarantees, it can identify specific areas needing additional attention.

> “Operational system security testing should be integrated into an organization’s security program. The primary reason for testing an operational system is to identify potential vulnerabilities and repair them prior to going operational. The following types of testing are described: network mapping, vulnerability scanning, penetration testing, password cracking, log review, integrity and configuration checkers, malicious code detection, and modem security. … Attacks, countermeasures, and test tools tend to change rapidly and often dramatically. Current information should always be sought.” [41]

A related topic is that the worldwide civil aviation community needs to identify common solutions for identity (section 4.8), IP addressing (sections 5.3 and 5.4), naming,[30] routing (section 5.5), protocol security (section 4.5), and authentication (section 4.9) subsystems. These common approaches need to be realized by consistent technology and configuration choices that produce a coherent worldwide civil aviation network infrastructure. These important technical

[30] Because airborne naming issues are common to naming issues present elsewhere in the Internet, this study did not specifically discuss naming. Readers who are unfamiliar with Internet naming are encouraged to learn about the DNS protocol (see RFC 2535).

issues need to be agreed upon by the aeronautical community before airborne avionics systems become networked to other aircraft or ground systems. This is because the safety of networked airborne LAN systems is potentially affected by the quality and integrity of the network system that is created by the worldwide civil aviation community. It is risky to permit networked airborne LAN systems to be created before the worldwide civil aviation community has decided on a common approach to address these key subsystems. Aircraft need to handle identity, IP addressing, naming, routing, protocol security, and authentication in a consistent manner with each other and with civil aviation ground systems if aircraft and NAS systems are to be networked together. The interfaces of both airborne and ground systems, therefore, need to be carefully articulated and designed if potentially significant security problems are to be avoided.

7.2 EXTENDING DO-178B INTO NETWORKED ENVIRONMENTS.

The system should identify the security and, thereby, the safety-related requirements for software. Software and system verification should ensure that they were correctly and completely implemented. The primary difference of extending software assurance processes into networked environments is to try to ensure that software vulnerabilities that can be attacked in networked environments do not exist. Latent bugs in software can be located in either the operating system, the application, or both. Of the five respondents to the FAA LAN survey (see appendix B) who identified which operating system hosted their airborne application, three did not use any OS at all, one used a COTS operating system, and one used a high-assurance OS. While any latent software bug is a potential avenue of attack, not all software bugs have equal exploitative potential. The vulnerabilities that exist within applications that are not built upon an OS are a function of that specific application environment itself and the ability of the attacker to compromise or modify that environment. By contrast, root kits are available on the Internet for exploiting generic COTS OSs (e.g., Windows®, Apple Mac OS, Unix, etc.). These root kits often contain script-based attacks against the commonly known vulnerabilities of those systems with the goal to compromise the OS, deploy Trojan horses (for continued control), erase log files, and to launch attacks on other entities. Section 4.3 discussed the dangers associated with using COTS OSs. For these reasons, COTS OSs should not be deployed within high-assurance environments except via a HAG. By contrast, high-assurance OSs are an excellent choice for high-assurance airborne network environments. If a high-assurance OS contains any vulnerabilities at all, those vulnerabilities are esoteric.

The DO-178B processes used to develop software targeted for networked airborne deployments need to be extended to explicitly reduce or eliminate the number of software vulnerabilities that can be leveraged by network-based attacks. However, as Ghosh, O’Connor, and McGraw have observed, processes alone cannot guarantee the creation of high-quality software:

> “Process maturity models and formally verified protocols play a necessary and important role in developing secure systems. It is important to note, however, that even the most rigorous processes can produce poor quality software. Likewise, even the most rigorously and formally analyzed protocol specification can be poorly implemented. In practice, market pressures tend to dominate the engineering and development of software, often at the expense of formal verification and even testing activities. … The result is a software product

> employed in security-critical applications … whose behavioral attributes in relationship to security are largely unknown.” [95]

Despite this, a variety of previous studies have proposed process extensions (e.g., references 40, 41, 73-75, and 95-97) using automated testing mechanisms at various stages of the development process to identify security vulnerabilities within software targeted for network environments.

This study concurs with those studies that development processes need to include tests that examine the actual implemented product to verify that its development processes did indeed produce the expected results. Various mechanisms to improve the current process have been proposed including:

- Using of model checkers on abstractions derived automatically from source code [40].

- Software fault injection into software to force anomalous program states during software execution and observing their corresponding effects on system security [95].

- Since a certain class of exploits relies upon buffer overflow vulnerabilities, various studies [96] have also recommended specific development mechanisms and tools for reducing that vulnerability during software development. Each of these approaches has a certain amount of overhead that may or may not be acceptable given specific implementation requirements. Regardless, these ideas nevertheless point out the desirability of understanding the root cause of the specific vulnerability and taking steps to correct it.

However, while these additional tests are helpful, they cannot ensure that the resulting software is of a high quality. Tests only identify the presence of specific problems. Software testing alone cannot guarantee the absence of flaws that were not addressed by the test suite. Creating test suites to address all of the possible flaws that may exist in airborne software is an unachievable goal due to the myriad of potential problems that may arise. The goals of software testing should be solely viewed as

> “The approach described in this paper does not purport to find the needle in the haystack, but rather to reduce the size of the haystack significantly …” [95]

There is no existing security theory or process that can extend testing systems to produce guaranteed high-assurance results for networked environments. This is a significant certification issue. Until this key missing certification element has been fixed, no networked system can currently be guaranteed to be as safe as nonnetworked airborne systems. Fortunately, this problem is partially mitigated by having code inspection be a constituent part of the certification process for higher assurance software (e.g., see DO-178B, Section 6.3.4 and Table A-5).

In conclusion, this study recommends that the FAA study the viability of enhancing current DO-178B processes with the specific process extensions and tests suggested by previous studies [40, 41, 73-75, and 95-98].

This study also recommends that the existing DO-178B assurance processes be applied very rigorously for higher assurance software (i.e., Level A and Level B software) in networked environments. The approval process should include the following three specific tests:

- A series of penetration tests should be performed upon the completed software item. Specifically, the software (including its OS, if any) needs to be subjected to a range of network attacks described in appendix A. Any problems identified from these attacks need to be fixed.

- The software should be examined under evaluation to verify that its internal construction complies with formal models of software construction, such as being modular and layered in terms of a structured presentation within the implementation itself.

- A rigorous line-by-line code inspection of the software should be conducted to demonstrate a lack of bugs that can be hostilely attacked. This implies that the approver has an excellent understanding of how software bugs can be exploited by network attacks and that the approver stringently examines that code base to identify and fix those problems.

Software items that do not undergo, or cannot pass, these three additional tests cannot be stated to be high assurance when deployed in network environments. Therefore, like any other non-high-assurance entity, they can only be deployed within high-assurance environments by means of an intervening HAG.

This study recommends very stringent application of existing software certification processes for high-assurance software in networked environments. The line-by-line code inspection requirement for high-assurance software certification should ensure that high-assurance software code bases explicitly use formal software techniques and are comparatively small in size (in terms of number of lines of code). The indeterminate number of bugs that are latently present in large code bases represent unaddressed attack vulnerabilities in networked environments. Current software development methods cannot be trusted to produce high-assurance results unless those results are supplemented with extensive scrutiny. The larger the code base, the more questionable the quality of the scrutiny. This means that software developers need to actively consider how to create high-assurance software for network environments so that the resulting software can be assured to be as bug-free as possible. Until a theoretical solution is devised that produces guaranteed, high-assurance, bug-free results, high-assurance software needs to undergo a very thorough (formal) line-by-line code inspection. A possible alternative is for the software developer to assemble high-assurance software into modules. The integration of these modules face the same types of integration issues that are addressed in ARP 4754, but this may potentially result in an approval approach in which only a select subset of the total software corpus will require a formal line-by-line code inspection.

8. CANDIDATE SAFETY AND SECURITY NETWORK SOLUTION.

The candidate safety and security network solution, which is presented in section 8.3, naturally follows from the material that has been presented to date. The final remaining explanatory concept, which is needed to create the exemplar architecture itself, is to discuss best practice SSE. Section 8.1 presents this remaining explanatory topic. Section 8.2 then applies the SSE to the combination of current FAA safety policies and Biba Integrity Model concepts that were explained in sections 3, 6, and 7 to address the network risks that were presented in section 4 and appendix A. This application defines the rules and relationships that underlie this study's recommended exemplar airborne network architecture. Section 8.3 presents the resulting airborne network architecture that directly derives from these rules and relationships. Section 8.3 architecture defines an exemplar environment needed by airborne networks to implement FAA policies extended by the Biba Integrity Model. That section includes the recommended configurations of the security controls to achieve a minimal set of defense in depth protections. A given deployment may choose to implement additional controls in addition to those described in section 8.3, because this design is a minimal subset needed to fulfill the criteria.

8.1 SYSTEM SECURITY ENGINEERING METHODOLOGY.

SSE defines the process for integrating computer security concepts and technologies into coherent system architectures, as shown in figure 29. To achieve maximum benefit from the SSE process, it should permeate the entire life cycle of a system, from birth to death. The SSE process helps to ensure that all decisions are consistent with the overall system design and purposes. This process also avoids the bolted-on phenomenon that has proven over time to be ineffective. Only by being developed as an integral part of the systems in which they operate can subsystem elements successfully counter serious threats and reduce vulnerabilities.

Security is the result of a complex interaction between multiple elements. As a result, one critical component of the SSE process is to understand the operational environment. This is accomplished by examining the actual operational environment to identify high-value assets, determining the threats to those assets, understanding their vulnerabilities, and selecting the proper countermeasures to protect the high-value asset. This process also provides an accrediting officer with the information needed to determine whether the residual risk is acceptable.

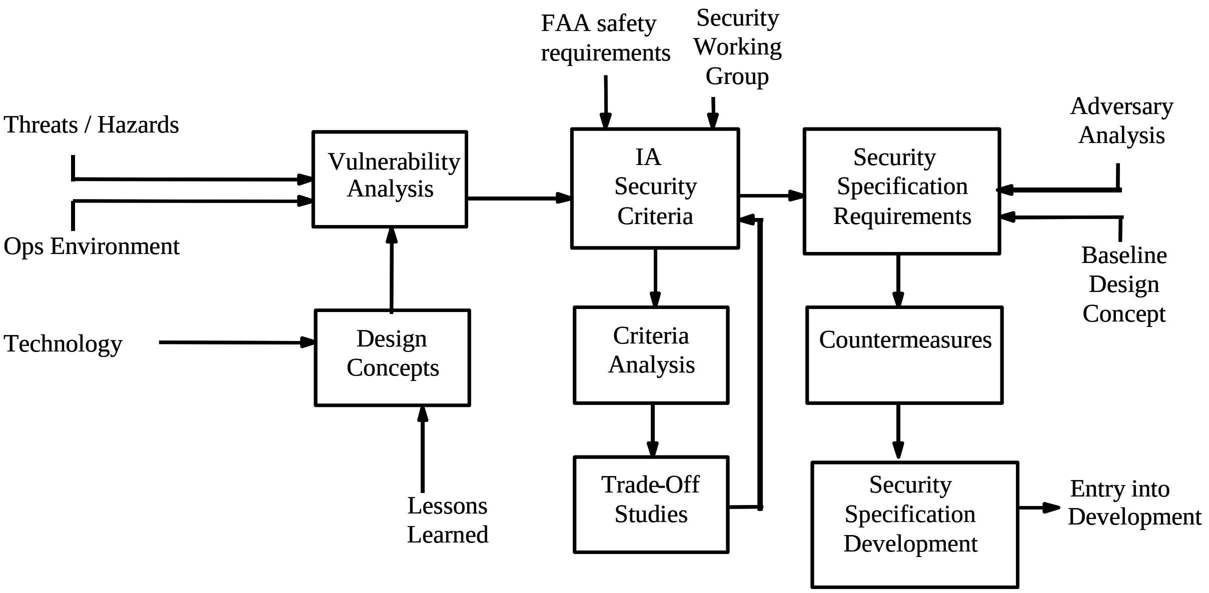


Figure 29. Security Engineering Process

The Systems and Software Consortium[31] has developed well-accepted SSE processes. Their generic approach can be summarized by the following steps.

1. Determine the security policies. This is a high-level definition of what is allowed and what is forbidden within the system. The policies provide the basis for determining the security requirements that will be developed and implemented. Without good security policies, one cannot determine the high-value assets and data that must be protected.

2. Determine and specify the security requirements. In this step, requirements for the protection of assets and data are determined using the security policies as a guide. It is essential that only requirements be specified, not solutions or constraints. Therefore, the requirements must be stated in technology neutral terms. In addition, the requirements must be practical and testable to permit eventual verification that the requirements have been satisfied by the final system. Finally, the security requirements should not be open to interpretation. This is accomplished in high-assurance systems by specifying the security design via mathematical formalisms. However, this is rare. In most cases, English is used to specify the requirements. Care must be taken to avoid ambiguity of meaning.

3. Establish a security engineering plan. This plan should include items critical to the design and implementation of the security protection mechanisms. Such items include the security requirements, constraints, and decisions already made. It should be used to help allocate the resources needed to properly complete the project while simultaneously establishing realistic expectations.

[31] See http://www.software.org

4. Learn from past mistakes. Poor development practices typically result in security vulnerabilities. By examining these past development practices and identifying those that improve or hinder system security, valuable lessons can be obtained and future implementations improved.

5. Document the operational environment. This is typically done in a document called the Concept of Operations (CONOPS). It describes the environment in which the system will operate, the roles and responsibilities of the major players, how the system is designed to normally operate, and potential contingency modes of operation. The security environment can be included in the CONOPS as a separate section or included in its own document (a Security CONOPS.) Elements of this security CONOPS should include a reiteration of the security requirements, the process used to select all countermeasures, how defense-in-depth is implemented, how the security mechanisms will operate, including user impacts, the effectiveness of the implemented countermeasures, how misuse is prevented or detected, the response mechanisms to a misuse incident, and the recovery process, if needed.

6. Perform a risk analysis. The risk analysis examines the operational environment to determine high-value assets, the threats to these assets, their vulnerabilities, countermeasures needed to reduce the risk for each threat/vulnerability pairing, and validation of the cost effectiveness of each countermeasure. For airborne environments, this approach differs from the traditional security engineering process by including safety as a key factor. It also differs from traditional safety analysis by considering the possible effects of malicious actions. In more detail, the first step is to determine the high-value assets to assist in focusing where the limited security dollars should be spent. In placing a value on each asset, the cost effectiveness of the selected countermeasures can later be determined. Once the assets are determined, each threat, which is asset- and environment-dependent, must be ascertained. In conjunction with this, the vulnerabilities of these assets must also be determined. Once the threats and vulnerabilities are determined, each threat is matched with the appropriate vulnerability. Any vulnerability without a threat or vice versa can be ignored. Otherwise, countermeasures are selected to reduce the threat and the cost of the countermeasures determined. A tradeoff is then performed between threat and vulnerability matches, countermeasure costs, and protected asset value.

7. Design the security architecture using the above information. The above risk analysis will identify the areas requiring protection and the cost-effective countermeasures requiring implementation. The security design should be consistent with accepted best practices. One such best practice is the concept of defense-in-depth (discussed in section 3.5). This concept uses the medieval castle as its model. Multiple layers of defense are implemented so that when one layer is successfully penetrated, other layers of protection still exist. While it is widely accepted that no security mechanism is foolproof, an architecture implementing the defense-in-depth concept should sufficiently delay the attacker to allow for the detection of the attack and to implement an appropriate response. This assumes that full control life cycles have been implemented to enable attack detection and response. Other best practices include least privilege, object reuse,

separation of roles, need-to-know, secure failure and recovery, input validation, and training plans.

8. Develop the system. In this step, the design is tested and technologies are selected for implementation. In most cases, this includes the use of COTS systems and applications software. However, COTS products with a large installed base are attractive targets for attackers. As a result, all COTS products should be identified and their suitability for implementation within specific NAS subsystems determined during risk analysis. Another potential security concern is the outsourcing of software development. The problem that must be considered is the potential for the introduction of malicious software into the developed and delivered product. Steps such as security vetting of the development company, verifying the company's development practices (capability maturity models or ISO certified), and issues such as ownership should be considered. Next, the developed system should include auditing capabilities and, optionally, automated alerts to administrative personnel. Only by examining the audits, can misuse actions be traced to the offending user or program. As a result, these audits should be organized by individual user, and they should record all user or software interaction with protected data. Other elements of concern during the development process include the software languages used (some are inherently insecure), constructs used, how errors are handled, the use of cryptography and digital signatures and their implementation, the access control mechanisms selected and implemented, and the proper implementation of all countermeasures.

9. Test the developed system. In this step, the implemented security countermeasures are verified. Testing can be as simple as a visual verification or as complex as a full mathematical proof of correctness. Most testing falls in between the two, relying upon use and misuse cases to verify correctness. These cases ensure the system properly protects the high-value assets from malicious insiders and outsiders. The approach taken is typically documented in a test plan that includes the use and misuse cases. The result of the testing phase is a report of the tests performed and the verification that all security functionality has been exercised according to the plan.

10. Operations. Such issues still relevant to the security systems engineering process include processes for software updates. During the operation of the system, security mechanisms must be patched and updated. This process should be planned prior to operations.

8.2 APPLYING THE SSE METHODOLOGIES TO AIRBORNE NETWORKS.

Following the SSE process is intended to produce a best current practice security design for a specific deployment in terms of the specific requirements and needs of that deployment. SSE was not devised to create generic security designs for generic deployments. This study leverages SSE to benefit from best current practices rather than invent a novel approach with unproven results. This application of SSE solely addresses the articulation of current FAA safety policy (e.g., DO-178B and ARP 4754) in terms of the Biba Integrity Model framework. It does not address the very important issues and requirements that specific deployments have that extend beyond this foundational policy framework. For this reason, this study views its resulting exemplar airborne network architecture (see section 8.3) to only be a minimal airborne network architectural subset that needs to be built upon to satisfy the actual safety and security requirements of specific NAS and airborne deployments.

The initial steps of the SSE process will be discussed in this section to examine the safety requirements of a generic networked airborne system environment. As previously observed, networked environments have both safety and security requirements. Although the SSE processes were originally intended to address security needs only, this section extends them to existing FAA (i.e., DO-178B and ARP 4754) safety policies applied within a Biba Integrity Model context. As explained in section 7, this policy foundation also leverages best current IA practices as articulated by the IATF, most notably, its defense-in-depth (see section 5.1) provisions.

The first step in the SSE process is to determine the security policies of a deployment. The security policies are the current DO-178B and ARP 4754 safety processes mapped in terms of the Biba Integrity Model framework.

The second step in the SSE process is to determine the security requirements that are derived from the security policies. Because this study uses existing FAA safety processes mapped to the Biba Integrity Model framework (i.e., step 1 of the SSE process), the result of this step produces the following set of safety requirements:

- Requirement 1: Networked entities that are classified at a software level that has potential safety repercussions to aircraft operation (i.e., Level A through Level D) shall be partitioned from the larger network environment and combined into a network enclave that functions at that specific software safety level with other entities classified at the same safety level (see figures 11 and 14). Networks or items at a different safety level from each other shall not be able to communicate together (see Requirements 6 and 8 for two specific exceptions to this general requirement). For example, Level B systems or software shall not be combined into the same partitioned network enclave with Level C systems or software.

- Requirement 2: Because Level E software systems have no safety repercussions to the aircraft, they do not need be partitioned (i.e., formed into common network enclaves). (Note: the FAA may want to study whether Level D software should be treated as a Requirement 1 or a Requirement 2 entity. Because this study did not know the most appropriate way to treat Level D entities, it is tentatively classifying them as Requirement 1 systems.)

- Requirement 3: Physical network media and devices that operate at the physical or data link layer of the OSI Reference Model (i.e., data link layer and below), deployed within aircraft, must be assured at the same software (safety) level as the highest software level entity that they support. For example, if entities operating at software Level A are conveyed within a physical airborne network, then the media, switches, and bridges that create that physical network system that transport Level A packets must also be assured at software Level A.

- Requirement 4: Entities that are located outside of aircraft (e.g., ground-based, space-based (e.g., satellite), other aircraft)) that directly or indirectly communicate with elements within the airborne system at Level A through Level D (i.e., Requirement 1 systems) must belong to the same distributed network enclave partition as the airborne software or system with which they are communicating (see figures 27 and 30). These entities therefore need to either have been certified and accredited at that software level or else be connected to that software level (VPN) network via a Biba Integrity Model HAG (see Requirement 8).

- Requirement 5: The physical network system elements that connect the airborne network elements with other entities located outside of that aircraft (see Requirement 4), need to comply with the same requirements that pertain to aircraft physical network systems (i.e., Requirement 3).

- Requirement 6: If a software system (e.g., a combination of software entities) primarily or exclusively communicates in a tight relationship within their select group and the group is comprised of entities at different software levels, then that tight-knit, cross-level community can be combined into a partitioned network enclave together (e.g., integrated modular avionics systems). That localized enclave operates in a system-high manner. There needs to be a special extenuating process or policy established within that enclave to enable a system-high situation to exist, since it represents an exception to the most direct application of the Biba Integrity Model, which naturally results in MSLS partitioned networks (i.e., see Requirement 1). System high networks are classified at the software level of the lowest classification level entity within that grouping and are distinct network enclave partitions from MSLS partitioned enclaves (i.e., Requirement 1 systems).

- Requirement 7: It needs to be noted within the assurance process whenever a system or software entity has safety-related network connectivity requirements or dependencies with any other system or software entities. Specifically, it should be noted if entities have real-time, latency-sensitive, or high-availability connectivity requirements with specific other entities. If the network enclave that supports those entities cannot be assured to satisfy those network connectivity requirements, then those elements can be supported via a dedicated data bus (or LAN) that solely exists to meet that connectivity requirement.[32] If a dedicated physical data bus needs to communicate with other LANs or data buses, then the dedicated physical data bus is linked to that other physical network via a router (i.e., a relay device operating at the network (i.e., IP) layer only).

[32] The reason for the dedicated data bus (or LAN) is to ensure that the special network requirements of those devices will be met. It is of course preferable if their requirements can be met in the normal manner (e.g., via a common high-assurance LAN). However, this requirement exists to say that it is okay to provide special data bus connectivity for certain devices having requirements that absolutely require dedicated physical data buses or LANs.

- Requirement 8: Biba Integrity Model HAGs may be strategically positioned, on an as-needed only basis, to safely join together entities classified at different software levels. The HAG is specifically designed to address the issues that otherwise would hinder a less trusted integrity entity to safely communicate with a more highly trusted one in accordance with Biba Integrity Model precepts. The HAG device is a middlebox that is inserted between the communicating entities or networks to provide the controls (e.g., availability and integrity) necessary to ensure safety between the communicating entities. The HAG is a highly trusted device. It, therefore, needs to be certified at both the highest software level of the specific entities it is connecting (for safety) and also at EAL 5 or above (for security).

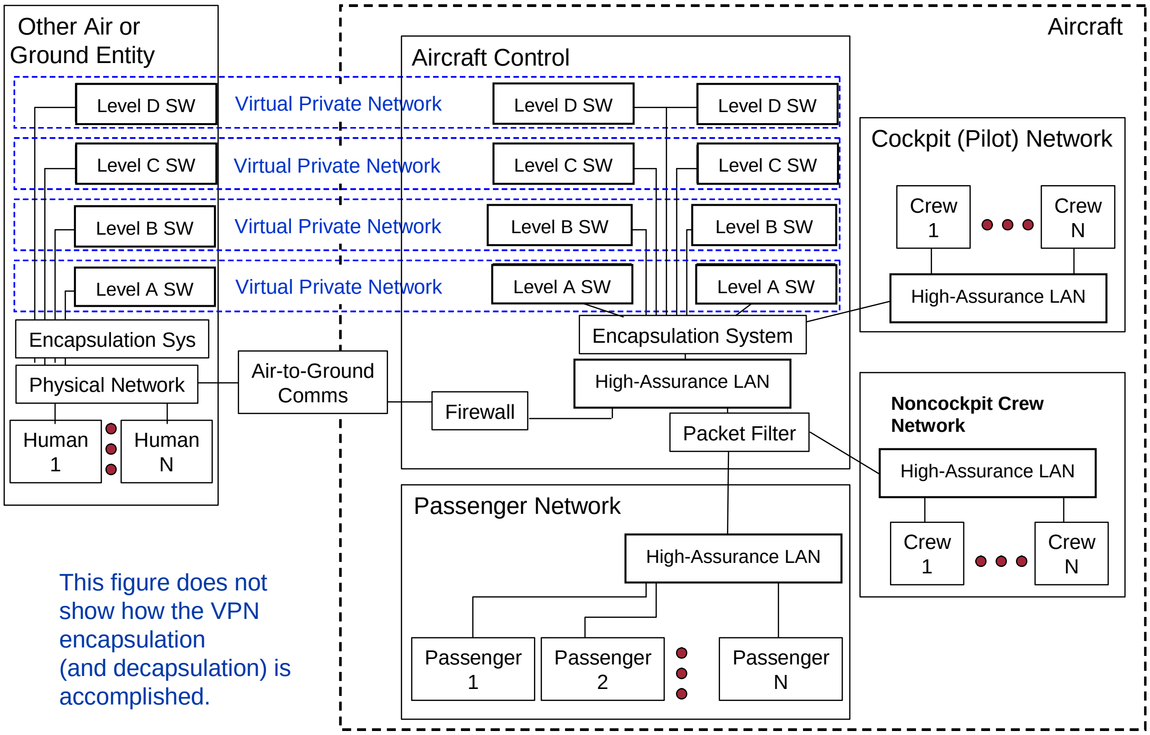
These requirements require that a system or software entity be classified at a specific software level and only communicate with entities classified at that same level via a VPN network also certified at the same level, in general.

Step 3 of the SSE process is to determine a security engineering plan. The security engineering plan used for networked airborne systems shall comply with the extended DO-178B and ARP 4754 concepts explained in section 7.

The next steps (steps 4 and 5) of the SSE process are specific to a given deployment. These steps need to be followed to extend the generic architecture identified by this study into a specific deployment environment. In step 6, a risk analysis for that deployment is performed. Section 6.1 presented the result of a risk analysis for generic networked airborne environment. With the previous steps as background, the SSE process in step 7 then creates a security architecture. This architecture applies best current IA practice (i.e., IATF) to the resulting generic system. The resulting security architecture for a generic airborne network environment is presented in section 8.3.

8.3 EXEMPLAR AIRBORNE NETWORK ARCHITECTURE SOLUTION.

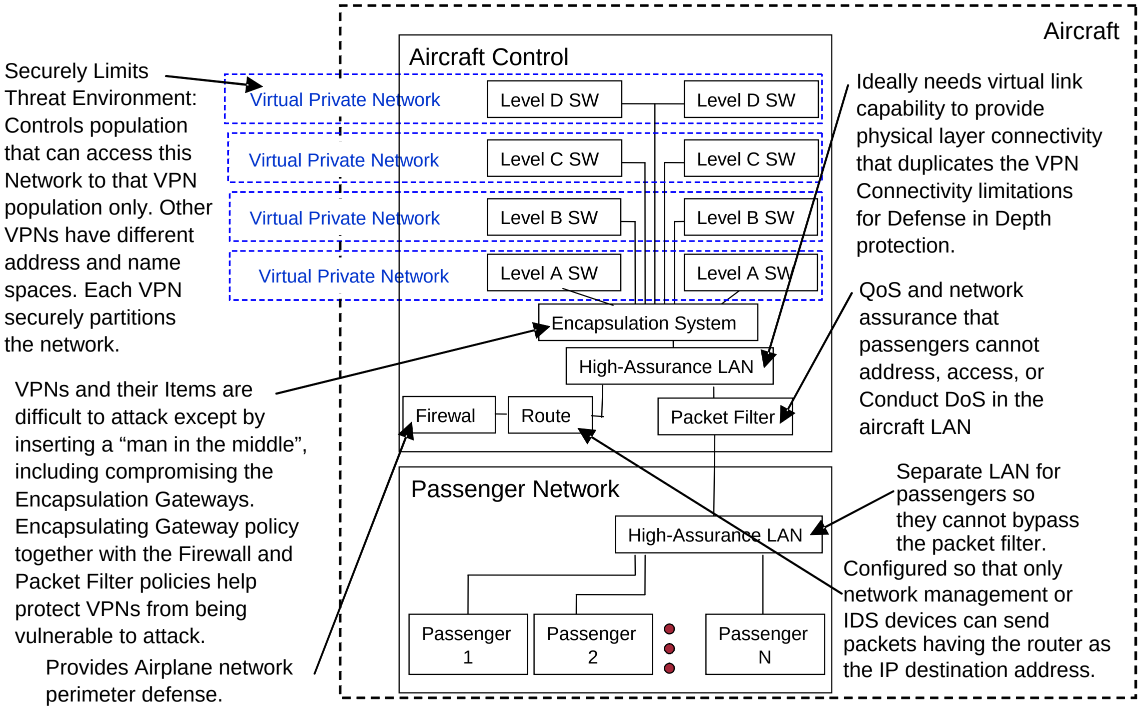
Figure 30 shows a high-level view of a generic network design that this study recommends for airborne networked environments. This design was constructed by following the SSE processes (see section 8.2) for the extended DO-1789B and ARP 4754 processes described in section 7. Specifically, this section provides the generic airborne security architecture defined by SSE step 7.



SW = Software

Figure 30. Secure Generic Airborne Network Design (High-Level View)

Figure 31 shows how the recommended architecture addresses many of the network risks that were previously discussed in section 4.



Securely Limits Threat Environment: Controls population that can access this Network to that VPN population only. Other VPNs have different address and name spaces. Each VPN securely partitions the network.

VPNs and their Items are difficult to attack except by inserting a “man in the middle”, including compromising the Encapsulation Gateways. Encapsulating Gateway policy together with the Firewall and Packet Filter policies help protect VPNs from being vulnerable to attack.

Provides Airplane network perimeter defense.

Ideally needs virtual link capability to provide physical layer connectivity that duplicates the VPN Connectivity limitations for Defense in Depth protection.

QoS and network assurance that passengers cannot address, access, or Conduct DoS in the aircraft LAN

Separate LAN for passengers so they cannot bypass the packet filter.

Configured so that only network management or IDS devices can send packets having the router as the IP destination address.

SW = Software

Figure 31. How Design Addresses Network Risks

Figure 32 shows how these threats are addressed in a defense-in-depth manner.

Larger the network, the larger the number of threats—Indirect Internet connectivity means 1B+ potential human users	• VPN for network partitioning • Firewall for network perimeter defense • IPsec required for protocol security
End users are now part of security framework	• VPN for network partitioning • Packet filter keeps passengers from accessing inappropriate Items and LANs
Availability of Airborne LAN	• Firewall and packet filter to control access • QoS policies ensure support for VPN traffic
Integrity of computers, networks, applications, and data	• VPN for networking partitioning • Firewall and packet filter for LAN defense • IPsec for secure protocol interactions • Secure software download and integrity checks
COTS device security questionable (e.g., routers, PCs) and subject to compromise	• IATF defense-in-depth security controls • Increase CC assurance when relied upon • Only attached to VPN via HAG
Complex internet protocol family security	Use available IETF protocols' security Alternatives and IPsec whenever possible
SNMPv3 security issues	• Always use IPsec with SNMPv3 • Once improved SNMPv3 alternative (i.e., ISMS) available, preferentially use it.

Figure 32. How Design Addresses Network Threats

Because all communications between aircraft and other aircraft or ground stations occur across AS boundaries (see section 5.3), aircraft networks form BGP relationships with their peer ASs on the ground or in the air. The aircraft's ASBR is not shown in figure 30, but it is physically located between the airplane's high-assurance LAN and the air-to-ground communications within the figure. That ASBR links the airplane's network to other ASs (air- or ground-based).

The following sections each describe a specific security control that is identified within figure 30. Please note that the configurations described in these sections will produce the defense-in-depth results shown in figure 32.

8.3.1 The VPN Encapsulation Method.

The VPN encapsulation is accomplished by using IPsec's ESP in tunnel mode in accordance with reference 99. The encapsulating gateways that perform the tunnel mode service may theoretically be end-systems, routers, or middleboxes. However, because the items located within the VPN needs to be managed by means of the agency of the encapsulating gateway (see section 8.4), this architecture presumes that the encapsulating gateways will preferentially be middleboxes. If they are middleboxes, then it is very important that they not decrement the time-to-live (TTL) field in the IP header of the encapsulated (RED) packets of the forwarded packets so that they remain transparent to the packet flow. (Note: if they are end-systems, they similarly will not decrement the TTL. However, if they are routers, then they will need to decrement the TTL because that is normal router behavior.)

The selected VPN approach for this architecture uses IPsec in tunnel mode. It was designed by the L3VPN working group of the IETF [100]. This VPN design is entitled "Architecture for the Use of PE-PE IPsec Tunnels in BGP/MPLS IP VPNs" [99]. (Note: at the time of this writing, reference 99 has passed the IETF L3VPN working group's last call and is currently in the RFC editor's queue to be issued as an Informational RFC.) This is the secured IPsec variant to the L3VPN's VPN design approach, which is "BGP/MPLS IP Virtual Private Networks" that was defined in RFC 4364. RFC 4364 is an IETF Proposed Standard protocol.

The high-level architectural view of figure 30 does not show the encapsulation method recommended by this architecture. The encapsulation method detail is shown in figure 33. Section 5.6 introduced the concept of VPN and section 5.2 described the current DoD COMSEC approach using a type of IPsec VPN. The particular VPN variant selected for this design was chosen because of its scalability, minimal latency, and high security properties. However, other VPN alternatives also exist (e.g., Intra-Site Automatic Tunnel Addressing Protocol (see RFC 4214); IP with virtual link extension [101]; Teredo (see RFC 4380); and the bump-in-the-wire security gateway of RFC 4301).

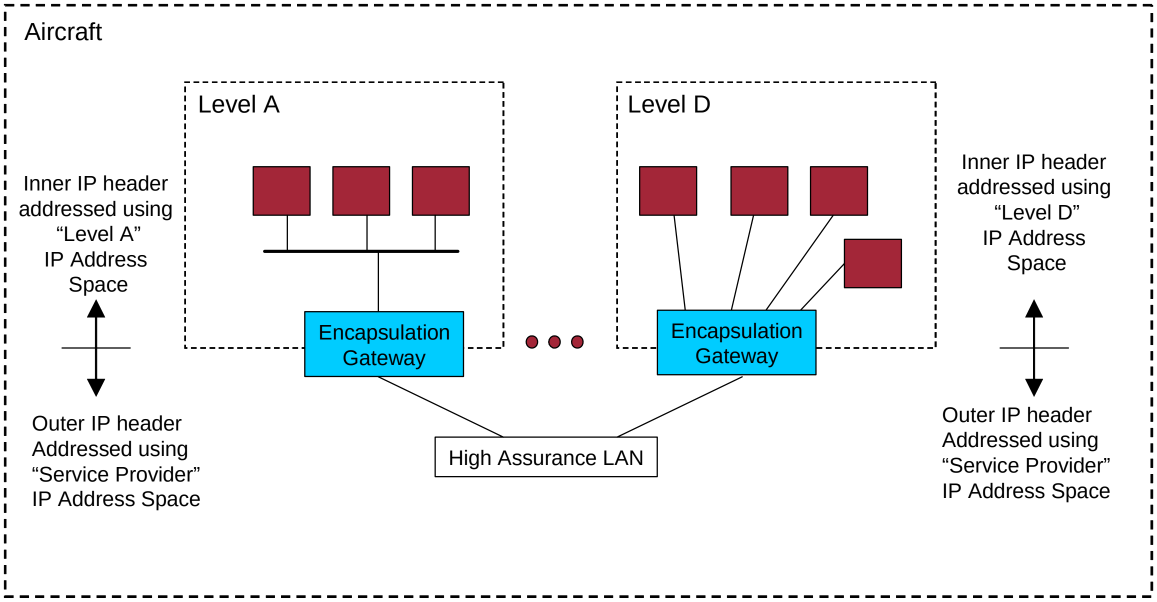

Aircraft
Level A
Level D
Inner IP header addressed using "Level A" IP Address Space
Outer IP header Addressed using "Service Provider" IP Address Space
Encapsulation Gateway
Encapsulation Gateway
Inner IP header addressed using "Level D" IP Address Space
Outer IP header Addressed using "Service Provider" IP Address Space
High Assurance LAN

Figure 33. Close-Up Detail of How Encapsulation is Accomplished

Figure 34 shows the current architecture that underlies this design. This figure, which is a copy of Figure 1.1 from RFC 4110, shows that an ISP provides a provider edge (PE) interface to their network services. The fact that these network services are physically conveyed via a VPN through the service provider's network infrastructure is not necessarily known to their customers, who interface to the PE interface device via their own Customer Edge (CE) device.

Both the PE and CE devices are normally either IP routers or label switching routers (i.e., the latter supports MPLS, and the former supports traditional IP IGP and EGP routing). The labels r3, r4, r5, and r6 in figure 34 represent IP routers that are internal to the customer site. The IPsec variant [99] of RFC 4364 that is used in this architecture is described as follows:

> "In BGP/MPLS IP Virtual Private Networks (VPNs), VPN data packets traveling from one Provider Edge (PE) router to another generally carry two MPLS labels, an "inner" label that corresponds to a VPN-specific route, and an "outer" label that corresponds to a Label Switched Path (LSP) between PE routers. In some circumstances, it is desirable to support the same type of VPN architecture, but using an IPsec Security Association in place of that LSP. The "outer" MPLS label would thus be replaced by an IP/IPsec header. This enables the VPN packets to be carried securely over non-MPLS networks, using standard IPsec authentication and/or encryption functions to protect them." [99]

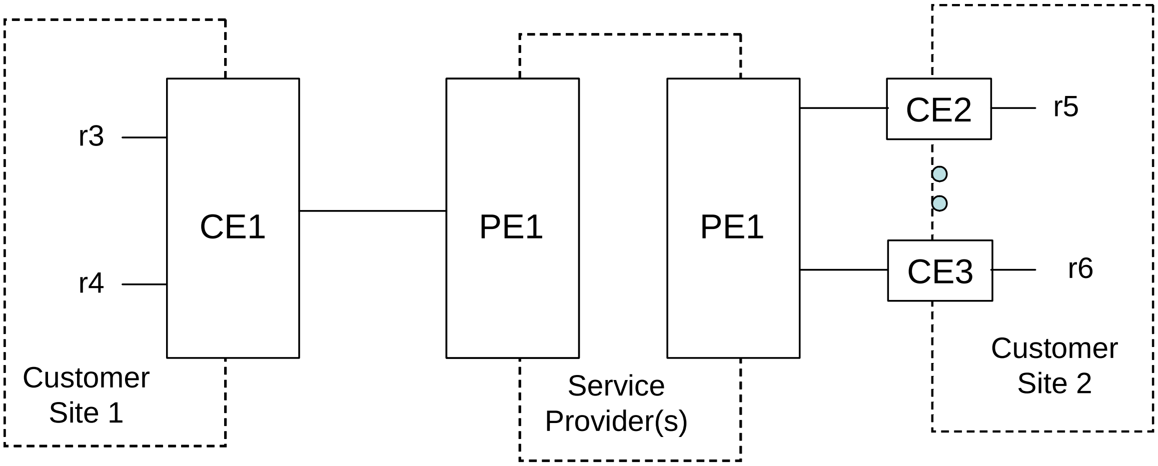


Figure 34. The VPN Interconnecting Two Sites (copy of Figure 1.1 of RFC 4110)

The specific implementation proposed in this report's design has defined an encapsulation gateway middlebox (RFC 3234) that performs the functions of both the CE and PE interfaces of figure 34 for each specific software level VPN community. For example, if an airplane has four different software level communities, then there will be four distinct encapsulating gateway devices on that airplane, one for each software level community. The encapsulation gateway, therefore, operates exactly like the COMSEC device in figure 17 and the interface in figures 20 and 22. The VPN technology recommended by this study uses ubiquitously available IPsec technology. However, VPN scalability itself is achieved by adopting proven IETF L3VPN BGP/MPLS techniques. The IPsec variant of BGP/MPLS, which this study recommends, is not as widely deployed today as its BGP/MPLS parent technology. The deployments that do exist primarily (probably exclusively) implement the IPsec approach via routers. There are two reasons this study recommends developing an encapsulation gateway middlebox rather than using the traditional dual router implementation of reference 99 currently deployed in the Internet today:

- To reduce the SWAP footprint of the encapsulation upon aircraft.

- To enable network management deployments where an entire airplane (e.g., multiple enclaves) can be managed from a single network management system (see section 8.4).

It is probable that no middlebox implementation of this technology existed at the time this study was written. Creating a middlebox variant of this technology, therefore, represents a recommended development activity. Special care should be taken in the security design of its network management support capability (see section 8.4).

Figure 30 shows that the encapsulating gateway that service Level A software networks on the airplane communicates with its peer encapsulation gateways servicing Level A networks on another airplane or on the ground via IPsec's ESP in tunnel mode communications. Entities within the Level A networks use normal IP communications between themselves (i.e., plain text). From their perspective, they are using COTS IPs just like any other IP device would. They are unaware that any network exists outside of their own Level A enclave. They are also unaware that their enclave is using network services provided outside of their enclave (e.g., the network between the encapsulation gateways that service their enclave). VPN encryption and encapsulation is performed by their local encapsulation gateway so that no entity or network outside of their network enclave sees intraenclave communication except in its encrypted and encapsulated form. For example, from the point of view of the firewall in figure 30, communications from a Level A device on the airplane to a Level A device off of the airplane is merely an IP communication between two different encapsulation gateways (i.e., no entity outside of the VPN-protected enclave itself knows about the enclave).

Therefore, the Level A VPN enclave entities have no knowledge about any entity outside of their own enclave community. The same is true for the Level B VPN enclave, the Level C VPN enclave, and so on—each VPN enclave only knows about itself. No entity outside of that enclave knows about entities inside a different enclave. Therefore, the enclave population is narrowly restricted to the members of the enclave only. Effective network partitioning has occurred. Even in the worst-case scenario where all firewalls in the entire NAS and on every airplane have become compromised and the airplanes are connected to the worldwide Internet, the enclave population remains restricted to the enclave membership only. Airplane passengers cannot communicate with devices inside an enclave (indeed, they do not know they exist) nor can any other entity outside of the enclave do so. Therefore, the risks articulated in section 4.1 have been mitigated. If there is no human presence in an enclave (i.e., if the enclave is solely populated by devices), then the risks articulated in section 4.2 have also been mitigated for that enclave. If both of these are the case, then the concerns mentioned in sections 4.3 and 4.4 persist at a diminished risk level because the threat agents that can directly attack VPN entities are now restricted to the device population of the VPN itself. (Note: defense-in-depth protections (e.g., QoS) are still needed to ensure that the LAN supporting the VPN is not attacked, which, if successful, could potentially result in DoS to the VPN.) Nevertheless, COTS devices are not deployed in higher software level networks (except via HAGs) for defense-in-depth reasons (see section 8.4).

However, VPNs are distinct partitioned networks within the larger network system. The VPNs are unaware of the existence of anything outside of their VPN. Each of the VPNs shown within figure 30 is isolated, unaware of the existence of other entities outside of their own VPN. Other entities cannot communicate with them and they cannot communicate with other entities—nor can they know about each other in the general case (see section 8.4). The reason this approach leverages reference 99 is that it provides for the VPNs themselves to internally grow to become as arbitrarily large and complex as they need to be in a secure and scalable manner.

Figure 33 shows two additional points that have not yet been discussed. The first is that the devices within the enclaves are shown in two different network configurations. In the Level A network example on the left, they are shown as using a common, private physical LAN among themselves (alternatively, a switch or hub could have been shown). Second, in the Level D network example, which is on the right side of the figure, they are shown connected via multihomed interfaces of the encapsulating gateway. The right-hand approach requires the encapsulating gateway to perform relaying functions within the LAN itself. The left-hand approach offloads that responsibility from the gateway and also enables support for devices with real-time or latency-sensitive requirements (e.g., see Requirement 7 in section 8.2).

By performing both the PE and CE functions of figure 34, the encapsulating gateway straddles two different worlds. Its IP interface to the enclave is addressed in accordance with the IP addressing policy of that enclave (see figure 33). Its IP interface to the high-assurance LAN is addressed in accordance with the IP addressing policy of that airplane. If the VPN enclave and the airplane are addressed from the same IP address space, then that fact is not known to either the enclave or the airplane. Specifically, the IP address space of each VPN enclave is orthogonal to the other enclaves and to the airplane. No collision occurs if entities within two different enclaves (or an enclave and the non-VPN parts of an airplane) have identical IP addresses. The only requirement is that the nonenclave entities within the airplane need to be addressed from the same IP address space as is used by the NAS and that each entity within a VPN enclave be addressed in a manner that is consistent for that specific enclave.

Figure 30 shows that pilot and crew networks are not part of VPN encapsulated enclaves. If the pilot or crew members need to communicate with entities within an enclave, the device accessed by the pilot or crew for that communication should be solely attached to that enclave.[33] Alternatively, a HAG could be inserted directly between the enclave (or device) that the pilot or crew needs to communicate with, and the pilot's (or crew's) computer.[34]

The mechanism by which VPN partitioning physically is accomplished differs in terms of the specific protocol layer at which the partitioning controls occur. The approach recommended by this study does the partitioning at the network layer (layer 3). The recommended partitioning

[33] Requirement 1 (see section 8.2) requires that enclave-attached entities must never be dual-homed between the enclave and anything else except via the agency of a HAG (see Requirement 8).

[34] Only encapsulation gateways and HAGs are permitted to be dual-homed between VPN enclaves and the airplane's network.

mechanism relies upon the controlled insertion (encapsulation) of a redundant IP packet header specific for the service provider within the protocol stack of the customer network packets (see figure 21) while they are conveyed across the service provider's network. The insertion of the redundant (encapsulated) IP header is accomplished by means of a specific encapsulation mechanism for that VPN connection (see figure 34). The encapsulated packets are then conveyed across the service provider's network by means of the encapsulated IP header (i.e., the service provider's IP header that was inserted into the protocol stack). Each of the customer packets conveyed by the VPN has their own IP header for their own customer network, which is not visible to either the service provider or other VPNs supported by that service provider because they only see the service provider-inserted IP header. Additional assurance is provided by the fact that the addressing within the VPN is a function of the specific network (i.e., IP addressing of the redundant IP header is from the address spaced used by the service provider's network; IP addressing of the customer's original IP header is from the address space used by the customer's network), which may or may not be from the same IP address space. The approach recommended by this study also has a third assurance mechanism: the customer's entire original IP protocol stack is encrypted when the encapsulation takes place so that all customer information is in cipher text form while traversing the service provider's network. These provisions ensure separation between the various VPNs themselves as well as from the conveying service provider network

Because the network management approach suggested in section 8.4 could possibly (depending on how it is implemented) introduce security vulnerabilities that otherwise could not exist within VPN systems, VPNs should be deployed with the following defense-in-depth [50] security protections:

- Firewall (and, if in a nonair gap target environment, the packet filter as well) should be configured to discard any non-IPsec packets addressed to airborne encapsulating gateways.

- The encapsulating gateway should also be configured to discard any packet sent to it that does not use the IPsec's ESP. It decapsulates and decrypts any received tunnel mode packets and forwards them to the VPN. Received transport mode packets are those communications to the encapsulating gateway itself. All transport mode packets must be successfully authenticated by the encapsulating gateway or else be discarded.

- QoS provisions to ensure that the VPN is provided adequate network capacity (e.g., to avoid DoS) are also needed to ensure the viability of VPN partitioning.

An integral part of this study's recommendation is that VPN enclaves should be created to protect safety-relevant airborne assets from network risks and to enable controlled, safe, and secure communications between air-to-air and air-to-ground entities. This means that ground entities that communicate with safety-relevant airborne systems also need to be arranged into appropriate VPN enclaves to communicate with those airborne enclaves. This further means that their networks are defined according to the same requirements (see section 8.2) as airborne systems so that their communications could mitigate the risks identified in section 4 and appendix A. This parallelism means that ground systems would also need to address the same network management issues (see section 8.4).

The exemplar network architecture recommended by this study, therefore, presumes that if airborne VPN enclaves are connected to other airborne VPN enclaves or to ground VPN enclaves at the same software (safety) level, then those linked VPN enclaves form a common distributed VPN network enclave together that jointly operates at that specific safety level. The specific VPN technology identified by this study was chosen because it is expected to be able to scale to whatever VPN network size is required to support a worldwide deployment. It is important to recognize that this connectivity means that the worldwide aeronautical network consists of both the nonenclave worldwide aeronautical network as well as the various worldwide VPN network enclaves, with each of the latter operating at a specific safety level. It therefore comprises partitioned network enclaves located within a larger civil aviation network whole.

This relationship creates explicit policy issues that the worldwide civil aviation community will need to address in a coherent way. Specifically, what is the trust model between civil aviation regions? Will the trust model for the regions' Level A software networks be the same as for their Level C software networks? What is the trust model between aircraft and ground entities? If air-to-air communications occur, what is the trust model between aircraft belonging to different airlines? Will the Level A VPN components of the NAS completely trust European Level A VPN components and vice versa, or will they establish distinct policies and service level agreement (SLA) mappings between their components? What security protections (e.g., firewalls) will be inserted to protect the rest of the VPN elements at that safety level from a contamination that occurred within a specific region? How will aircraft that travel between regions maintain their connectivity in a seamless, safe, and secure manner? If air-to-air applications and systems are created, what mechanisms (e.g., firewalls) will protect the VPN at a given safety level in one airplane from (perhaps undiagnosed) misbehaviors occurring in the VPN at that same safety level in a different airplane? What policy systems will govern the interrelationship between aircraft and ground entities? Will SLAs be required?

For any airborne network architecture to be viable in real-life deployments, common worldwide design choices need to be agreed upon to decide how identity, IP addressing, naming, routing, and authentication will be handled systemwide. These common definitions and their associated infrastructure should be shared by both air and ground systems within the worldwide civil aviation network deployment if the resulting airborne network is to operate seamlessly between regions.

8.3.2 Physical Security.

Specific physical security requirements are embedded within the figure 30 design. Those requirements are that aircraft control and the cockpit (pilot) networks or their devices must not be physically accessible by aircraft passengers. If there is any possibility of passengers physically accessing the cockpit (pilot) network, then the high-assurance LAN within the cockpit must be connected to the aircraft control network via the packet filter. Otherwise, the high-assurance LAN in the cockpit can use the same physical high-assurance LAN as aircraft control.

HAGs are high-assurance devices that need to be physically protected from areas that are accessible by passengers.

The noncockpit crew network devices should also not be accessible by passengers in general, but the design could accommodate situations in which passengers are not always physically excluded from the area where those devices are located. If physical separation is not possible, crew members must be very careful to not leave open applications running in situations when the crew member is not present (i.e., situations where passengers may access applications that have been opened with crew member authentications).

8.3.3 Encapsulation Gateways.

Encapsulation gateways support IPsec in accordance with reference 99 (see section 8.3.1). The encapsulation gateways must be configured so that all packets sent to their nonenclave IP interfaces must be dropped unless they use the IPsec's ESP. Encapsulation gateways communicate together using the ESP in tunnel mode. Network managers or IDS devices communicate with encapsulation gateways via the ESP in transport mode. Because of the authentication provisions contained within the ESP, encapsulation gateways should be configured so that they only accept communications from outside of the VPN enclave they support from three types of devices only: other encapsulation gateways, network managers, or IDS devices. They should be configured so that they ignore (e.g., drop) all non-IPsec packets coming from outside of the VPN. Packets sent to the VPN that they support must be IPsec in tunnel mode. The encapsulating gateway does not put any restriction upon packets sent within the VPN that it forwards. However, all packets addressed to the encapsulating gateway itself (from either outside of the VPN or within the VPN regardless) must be sent in IPsec or else they will be ignored (i.e., dropped).

Because VPN gateways only link together distributed VPN elements that operate at the same software level, their IPsec's security policy database (SPD) entries need to be configured to only permit IPsec security associations (SA) to be established with other encapsulating gateways operating at the same software level. Their SPD should be configured to prohibit any SAs to be created with any encapsulating gateway that services a different software level. The only exception is if a HAG exists on the plain text network (i.e., if the HAG is in place, then the two encapsulating gateways can be configured to establish SAs with each other). Encapsulating gateways should not be configured to permit SAs to become established between MSLS and system-high networks, regardless of whether they are operating at the same software level or not.

Encapsulating gateways may also need to support network management relaying, depending on how a given implementation has configured its network management system. The relevant issue is that because each VPN system can only know about its own VPN, and the internals of that system are hidden from all entities not in that VPN, then there is no natural way for a single network management system to manage an airplane's entire network environment if that airplane supports multiple VPNs. The most obvious management approach is to have a single management system per VPN; but that alternative means that multiple disjoint network management systems will exist, none of which have coordinated oversight over the aircrafts' entire airborne network environment. Network management devices will not be able to physically view, or even to have knowledge of the physical existence of, any device within a VPN unless the management station itself is within that same VPN. Therefore, the encapsulation gateways may optionally support provisions to provide visibility of VPN-resident systems that they support to network managers located outside of their VPN (e.g., on the common aircraft LAN) so that a single aircraft network manager can potentially manage all of the devices within that aircraft. (Note: because highly assured devices cannot be misconfigured, similarly highly assured devices may not need to be managed either. If this is the case, then the encapsulating gateways primarily serve to forward status and logging information to the network management system, including reports of the ongoing software integrity checks.) If this provision is supported, then strong authentication and authorization protections need to be in place to ensure that only that management station can manage those devices. Specifically, the system needs to be designed to prohibit spoofing, or man-in-the-middle vulnerabilities, between the network manager and the encapsulation gateways by requiring authenticated communications having strong integrity protections (i.e., required use of the IPsec’s ESP in transport mode between the manager and encapsulating gateway).

8.3.4 Packet Filter.

The packet filter in the aircraft control must be configured such that noncockpit crew network cannot address any encapsulation gateway. If the aircraft is using the figure 1 target architecture (i.e., no air gap between the passenger and avionics systems), then the packet filter needs to additionally provide the following services:

- No device within the passenger network can access the noncockpit crew network or the cockpit pilot network. (Note: If the network is configured so that devices in the cockpit or noncockpit crew network can access entities within the passenger network (e.g., for network debugging and management), then the filter definitions would probably need to combine transport layer connections originating from the passenger network with IP addresses in the cockpit and noncockpit networks rather than solely in terms of IP address filtering alone. If airlines restrict network management oversight to solely use TCP transports (which is what the IETF’s ISMS update to SNMPv3 will probably require), then the restriction could possibly be defined at the packet filter in terms of the direction of the TCP synchronous (SYN) and require that all user datagram protocol and other transports be blocked to those addresses.)

- No device within the passenger network can send packets to any encapsulation gateways (located within aircraft control).

- The packet filter, or a device closely associated with the packet filter comprising a common system (e.g., QoS middlebox), rate-limits communications from the passenger network to ensure that passenger communications cannot exceed a certain threshold rate. This provision attempts to ensure that passengers alone cannot cause a denial of service attack on the aircraft control’s high-assurance LAN by consuming a disproportionate share of its capacity.

8.3.5 Firewall.

The firewall needs to be configured as exclusively as possible. Because of the presence of passengers in the network in the figure 1 target, the HTTP overt channel vulnerability (see section 4.1 and appendix A.1) unfortunately cannot be fully mitigated, unlike the figure 3 target alternative. However, if aircraft design restricts pilot and crew communications such that they never use HTTP, then the firewall can be configured so that HTTP traffic (i.e., both Port 80 and Port 443) is filtered by the firewall whenever the packet's destination address is a nonpassenger device. Such a rule would provide aircraft devices helpful protection in figure 1 environments. Even if the pilot and crew were only permitted to use secure HTTP (i.e., Port 443), then at least the more dangerous Port 80 transmissions could be filtered. In any case, the firewall needs to be configured with the following considerations.

- All fingerprinting attempts (see appendix A.1) originating from outside of the aircraft to any entity within the aircraft will fail (except for those that occur through the HTTP overt channel for figure 1 environments).

- All communications to encapsulation gateways from outside of an airplane are blocked by the firewall unless they use IPsec's ESP. (Note: both the firewall and the encapsulation gateways themselves need to redundantly enforce this same rule for defense-in-depth reasons.)

- The firewall should also drop all packets originating from outside of the aircraft to IP destination addresses that are not deployed within the aircraft LAN. The firewall does not have visibility into VPNs since it only sees their encapsulating packet headers, which are solely addressed to encapsulation gateways.

It is desirable that an NIDS be associated with the firewall system if SWAP considerations permit and that the NIDS be configured to recognize attack footprints and to optionally send alerts to designated crew members or ground systems alerting them when certain types of attacks occur.

8.3.6 The ASBR Router.

In forthcoming air-to-ground digital communications systems, such as an IP variant of the ATN, internal aircraft routing will need to be associated with routing elements outside of the airplane for air-to-air and air-to-ground communications to occur. The specific mechanisms by which this will occur will be an integral part of the communication system itself. Since this network system has not yet been defined at the time of this writing, this report will speak of this relationship as if it were to occur between AS. In such a system, the aircraft would use an ASBR within the aircraft to communicate to external networking elements.

The ASBR, which is not shown in figure 30, must be present on the airplane to provide BGP connectivity with the remote air and ground networks with which the airplane is communicating. The airplane's ASBR should be configured such that all packets that are sent with an ASBR's network interface as the IP destination address should be dropped unless they use IPsec in transport mode and come from a network management station or IDS device that is local to that airplane.

8.3.7 High-Assurance LAN.

The high-assurance LAN should consider the restrictions and provisions specified by the "Safety and Certification Approaches for Ethernet-Based Aviation Databuses" document [9]. The virtual link capability that is available within avionics full-duplex switched (AFDX) [102-104] deterministic Ethernet makes that technology an attractive alternative to serve as the high-assurance LAN. The high-assurance LAN should be configured, if possible, to provide physical layer connectivity that duplicates the VPN enclave configurations as a defense-in-depth provision. This means that enclaves would be defined and protected by two complementary controls: the physical (OSI physical layer) connectivity restrictions by the high-assurance LAN and the protocol restrictions at the IP Layer enforced by VPN encapsulation and encryption.

The SWAP footprint of the airborne LAN system could be theoretically reduced by logically creating the multiple instances of high-assurance LANs shown in figure 30. Specifically, the many high-assurance LAN entities within figure 30 may actually be two physical LANs, with the remainder logically created by means of AFDX virtual links. However, the entire LAN system should not be limited to a single physical LAN because the passenger network needs to be a distinct physical LAN entity from all other LANs on the airplane. This latter requirement exists so that there could be no possibility to misconfigure the network and bypass the packet filter controls that need to be applied to passenger services in figure 1 deployments.

8.3.8 Quality of Service.

It is desirable for the virtual links to support QoS rate control semantics. This may be accomplished at the physical layer through explicit rate controls or, more probably, at the network layer (i.e., IP Layer) through deploying differentiated service QoS (see RFC 2474). However it is accomplished, the communications within the safety enclaves must be ensured to have the capacity that they need to perform their function. If the total actual network use across the aircraft control's high-assurance LAN exceeds the physical capacity of that LAN, then the difference needs to come from dropping the passengers' packets to ensure that aircraft systems have adequate network capacity. The rate controls associated with the packet filter cannot ensure that this happens alone because of the possibility of denial of service attacks originating from other sources (e.g., ground, other aircraft). While the firewall will drop packets targeted inappropriately, it will permit packets targeted to passengers to pass through. Thus, an internal QoS system is also needed to rate-limit external traffic going to passengers in aircraft that may be deployed (see figure 1).

8.3.9 Air-to-Ground and Air-to-Air Communications.

Air-to-ground COMSEC should ensure that the signals in space used for wireless communication are encrypted at the OSI reference model's physical layer. This would provide protection from eavesdropping by nonauthorized entities and discourage attacks that inject false communications into the data stream. However, these links will remain potentially vulnerable to availability attacks caused by hostile jamming, unless mitigation techniques such as antijamming

(AJ) or low probability of intercept/low probability of detection (LPI/LPD) waveforms are used. This study recommends further research using AJ waveforms for air-to-ground communications.

8.4 NETWORK MANAGEMENT EXTENSIONS.

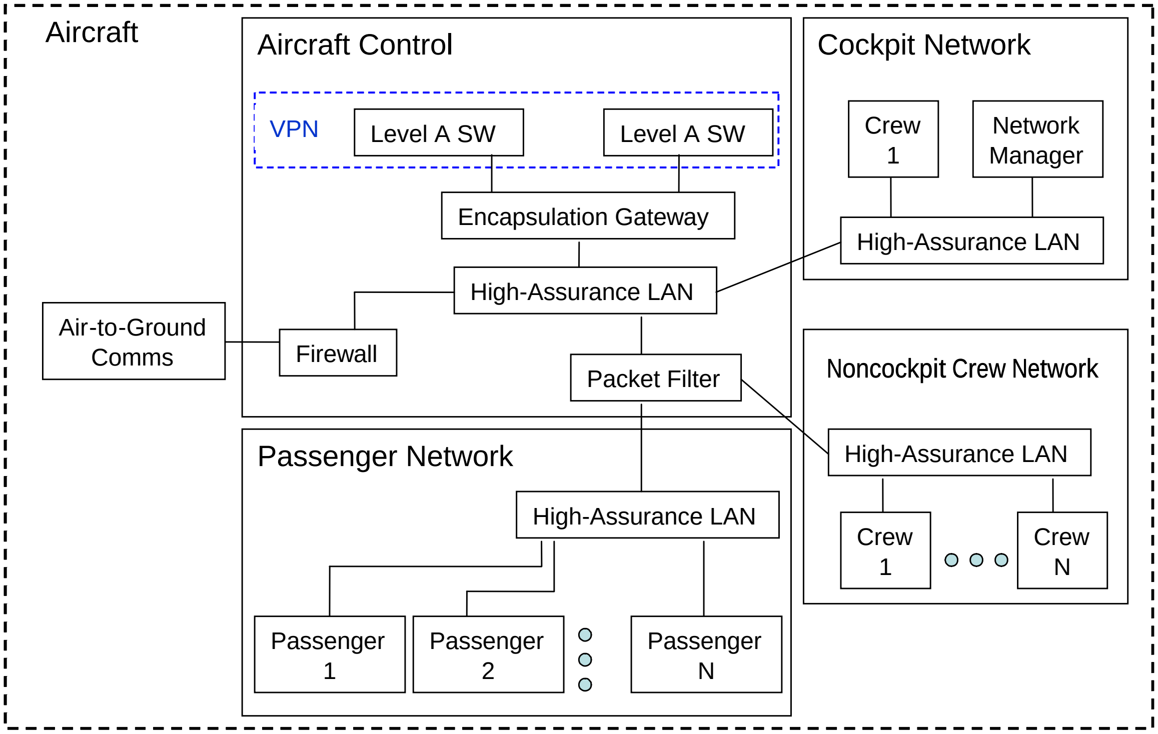
Network management is a very significant network design issue that was previously discussed in section 4.6. A basic network management tenet is that from a single management station the authorized manager should be able to: learn the current state of the total network system, and perform the appropriate management functions. This tenet is challenged by the network partitions that occur by deploying VPNs. Because it is unlikely that crew members will have the sophisticated training needed to perform traditional network management functions, the network designers need to consider just how network management should be performed. This is a very important issue that is directly related to the underlying concept of operations for aircraft. Relevant issues include the following:

- Will many management functions become automated so that human managers will be presented with a high level of abstraction? If so, then the education requirements for the crew could be reduced, but what would happen should successful attacks occur against the automated management systems themselves (e.g., how will those successful exploits be discovered and handled)?

- Will the network management of airborne aircraft actually occur from the ground? If so, then what would happen should the integrity of those management systems become compromised or air-ground connectivity be lost? In such a situation, will pilots have an override control capability? If so, how will the pilots discern that the integrity of the management system is in doubt?

Because these issues are directly related to airline, manufacturer, and FAA the concept of operations, this study has not provided a well developed network management recommendation. Nevertheless, these issues need to be competently addressed and a viable network management system needs be designed if airborne LAN systems are to be safely networked.

Figure 35 shows an example airborne network that has chosen to locate a network management station in the aircraft’s cockpit network. As previously discussed, this design would enable the network manager to potentially manage all of the devices within the network except for those that are physically located in a VPN. It could not manage devices within VPNs because it cannot “see” them (or even know about them) because they operate on a different IP protocol stack (i.e., an encapsulated one) than that used by the rest of the airplane.



SW = Software

Figure 35. Sample Airborne Network Management

If the entities within a VPN are to be managed, then they need to be managed by a network management station that also resides within that same VPN. However, if this is done, then the airplane will have multiple network manager systems, one for the unencapsulated network and one for each managed VPN. This would create a fragmented management view of the total network, which would greatly increase the difficulty of effectively managing that airplane.

Because of this, this study recommends that the VPN encapsulation be established by means of an encapsulation gateway middlebox, rather than the traditional dual PE and CE router approach (see figure 34), so that the aeronautical community would have the alternative of optionally building integrated VPN management capabilities into the encapsulation gateway itself.

As figure 33 shows, the encapsulation gateways have two faces: one to the unencapsulated airborne network and one to the encapsulated VPN community that they serve. In traditional VPN practice, there is no mechanism for these two networks to be linked, which is why VPN technology qualifies as being a viable ARP 4754 partition design for networked systems. However, if the aeronautical community decides to implement the IPsec VPN [99] technology by means of encapsulation gateway middleboxes recommended by this study, then the aeronautical community needs to determine if and how VPN management adjunct capabilities are defined within the encapsulation gateway design.

Such a design needs to carefully preserve the safety and security integrity protections that are provided by VPN technologies while simultaneously meeting the actual network management requirements. This is a very serious issue. The following discussion is a sample of the type of design decisions that need to be determined if encapsulation gateways are to effectively support VPN network management.

This study has stated that high-assurance devices cannot be misconfigured. For this reason, devices in Levels A and B VPNs may have comparatively diminished management requirements than other airborne devices. The stakeholders need to determine what that actually means. Does it mean that the primary management requirement of these devices will be to report their current status, explicitly including the results of the current (Tripwire-like) software integrity reports? Will different variants of encapsulation gateways be defined, with some variants supporting extensive configuration and management functions (e.g., for lower-software assurance VPNs) and others primarily supporting status reports (for higher-assurance VPNs)? Will the encapsulating gateways solely function to forward (pass through) traditional SNMP management communications between network managers and management agents that reside on the devices within the VPNs? Alternatively, will the management agent actually be located within the encapsulating gateway itself such that the agent within the gateway translates SNMP communications to and from standard network managers into actual management tasks performed upon the devices located within the VPN that it supports? Many other management approaches are possible, but it is desirable to find a consistent approach that is supported by the aeronautical community in which the interfaces and management schemas supported by the VPN encapsulation gateways are common and consistent worldwide.

From a security perspective, it is important that the encapsulation gateway be configured to drop all packets addressed to itself that do not use IPsec’s ESP in transport mode. Thus, the network manager will send management queries (or commands) to a specific encapsulation gateway and the encapsulation gateway will eventually report back to the network manager, with all communications occurring via ESP in transport mode. Both the encapsulation gateway and the network manager must authenticate each others’ communications. Approaches to authorize network managers also need to be carefully considered, with separation of duties with least privilege being recommended by this study. The encapsulation gateways will need to be certified as high-assurance security items (i.e., EAL 5 or higher).

Because network managers located on unencapsulated networks natively do not know about VPN entities, it is possible to preconfigure a network manager with information associating VPN devices with a specific encapsulation gateway. Alternatively, the encapsulation gateway could be queried—or pass through such queries directly to the VPN devices—concerning entities within that VPN, possibly providing information about their software identity, current software version, current status, and configuration (if appropriate).

Network management also contains software development implications. If software items are to be managed, then the management schemas by which the software is managed need to be devised in accordance with the network management system used on that aircraft. This requires coordination and advanced knowledge of the specific management protocol that will be used, the mechanisms by which that protocol will be secured, the desired format for the management schema, and a common approach for schema definition.

9. ANSWERS TO THE PHASE 1 QUESTIONS.

This section discusses several aviation safety concerns identified during the original FAA Screening Information Request for this study. These specific questions were a starting point for the work performed in phase 1 of this effort. The exemplar architecture, which was presented in section 8.3, describes the generic airborne network environment that identifies how these specific questions should be answered.

9.1 CONNECTION OF MULTIPLE DOMAINS.

Flight safety domains can be compared to security classification domains (see section 6.3), in that ratings are established based on the damage that a failure (or compromise) could cause. In addition, the assurance ratings of those systems are commensurate with the failure risk.

This report's architecture relies upon the Biba Integrity Model to segregate entities that have been classified according to DO-178B Section 2.2.2 software levels. These partitions are accomplished using the IPsec VPN variant. Virtual network enclaves are created for all networked entities that may possibly cause aircraft failure conditions. Specifically, Level A virtual network enclaves are created, as are Levels B, C, and D VPN enclaves. Because the possible failure of Level E entities does not result in aircraft failure risk, Level E entities are not similarly segregated into VPN enclaves.

These VPN enclaves materially reduce the security risks for networked devices, directly mitigating many or most of the threats identified in section 4. Specifically, this approach mitigates all identified threats for the higher-assurance enclaves. Nevertheless, each enclave, as well as the total network system, must be further protected by adopting IATF [50] defense-in-depth security provisions with full control life cycle protections (see section 5.1) for the reasons explained in section 7.

Even though each enclave shares a common underlying network infrastructure, entities in different enclaves are not physically unable to route to entities in other enclaves, nor are any entities within any enclave able to route to nonenclave (Level E) entities, or vice versa. This is due to inherent routing provisions within the VPN design.

HAGs can be deployed to provide localized, highly controlled, high-assurance connections between specific devices or specific enclave subgroups that are classified at different safety classification levels. This is the sole provision permitted by the Biba Integrity Model for entities in different enclaves to communicate together.

9.2 INTEGRATED MODULAR AVIONICS IMPLEMENTATION.

Integrated modular avionics (IMA) describe a distributed real-time computer network aboard aircraft. This network consists of a number of computing modules capable of supporting numerous applications operating at differing safety criticality levels.

Section 8.2 specifies the safety requirements derived from the use of the Biba Integrity Model. Four of these requirements are directly applicable to IMA requirements:

- Requirement 1 ensures that current FAA assurance provisions are maintained within networked environments.

- Requirement 6 enables software entities operating at different software levels but having tight-knit operating relationships to form a common system-high VPN together. That VPN is viewed as operating at the same software level as the software entity with the lowest software level in the VPN.

- Requirement 7 ensures that provisions exist to support networked entities needing QoS guarantees from their underlying VPN to support real-time, latency sensitivity, or guaranteed availability requirements. This is accomplished by deploying a dedicated physical network (e.g., LAN) to connect these entities.

- Requirement 8 provides a mechanism (i.e., HAGs) where entities or subenclave groupings can communicate with other entities or subenclave groupings operating at different safety-critical levels.

Although these four requirements are pertinent to IMA, the specific way in which they are applied is a function of the requirements of a specific implementation. For example, figure 36 shows a possible approach that conforms to the architecture where each of the IMA software entities also has requirements to communicate with other entities that operate at their own software level. (Note: because the devices in this example need to communicate extensively with non-IMA devices at their own classification level, this particular IMA system does not qualify for the Requirement 6 system-high approach. Please also note that the connection of the encapsulation gateways to the high-assurance LAN is not shown in this figure.)

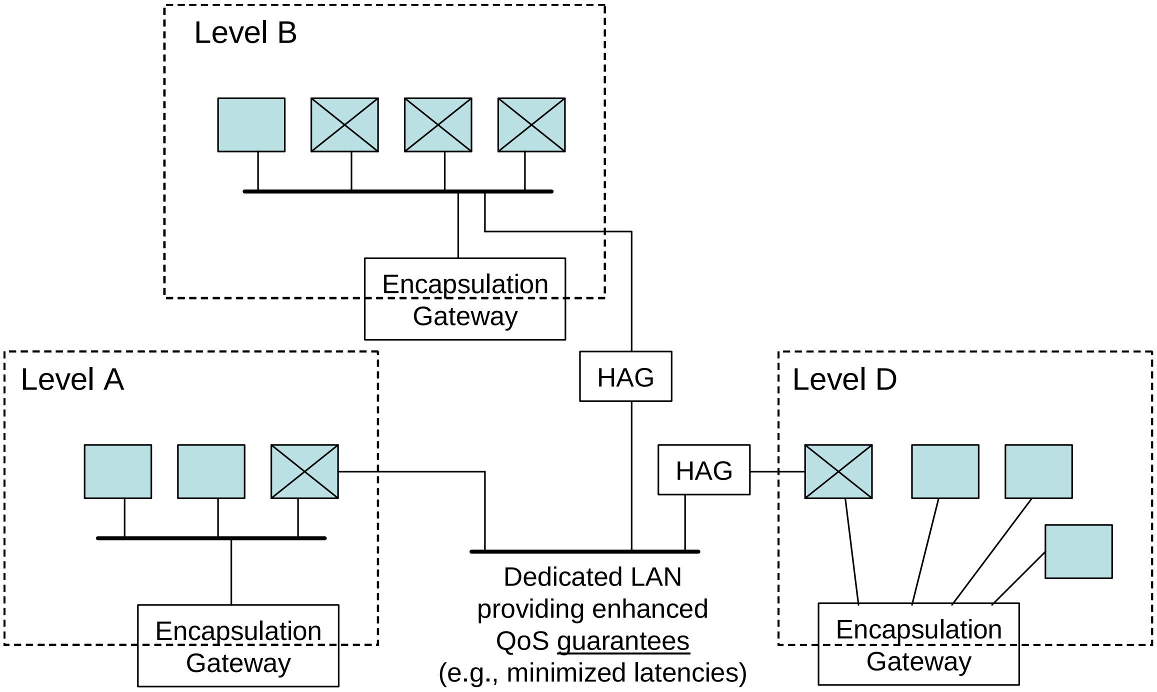


Figure 36. Notional IMA Design (Example 1)

The devices in the figure with an X are IMA devices. It is possible that the normal airplane VPN design shown will provide adequate support for IMA's real-time requirements. However, figure 36 assumes a worst-case scenario where this is not the case. Therefore, figure 36 provides an architecture where very tight real-time requirements for IMA interactions can be supported.

Figure 37 shows the same IMA devices that were in figure 36 except they are now deployed within a system-high environment (i.e., Requirement 6). There needs to be a special process or policy established within a system-high enclave to enable a system-high situation to exist, since it represents an exception to the direct application of the Biba Integrity Model, which naturally results in MSLS networks (i.e., see Requirement 1 of section 8.2).

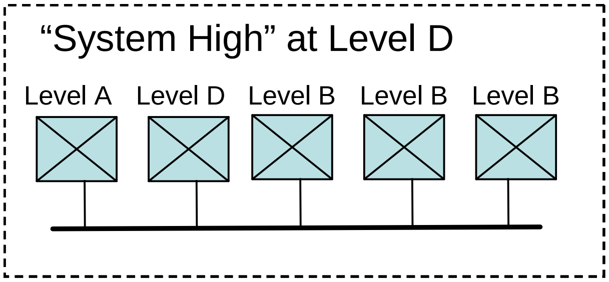


Figure 37. Notional IMA Design (Example 2)

9.3 USING PUBLIC IPs.

The model and architecture presented in this study does not rely upon any unique IP addressing posture. The only IP requirements of the model are that

- the nonenclaved devices within the airplane (e.g., the airplane's firewall, ASBR, etc.) need to be IP addressable by other airplane and NAS ground entities. The architecture simply does not care whether this is achieved by using public IP addresses, whether the entire aeronautical network uses the same common private address space,[35] or whether a combination of private IP addresses and an airplane-local NAT is used.

- the entities within each VPN enclave must be addressed from the same IP address space. The architecture does not care whether this IP address space is public or private.

The IETF community has had extensive internal discussions about whether private IP addresses are more secure than public IP addresses. While this remains a highly controversial topic, the majority's position is that private addresses have no appreciable security benefit over public IP addresses. The most powerful argument in favor of using private IP addresses for security purposes is that because private addresses have no uniqueness property outside of their enclave, use of private addresses cloaks internal networks from external visibility and limits access. The force of this argument diminishes the more closely one examines the details for maintaining private addresses within public spheres.

9.4 ELECTRONIC FLIGHT BAGS.

AC 120-76A [8] provides guidance for the certification, airworthiness, and operational approval of electronic flight bag (EFB) computing devices. EFB equipment refers to the replacement of historically report-based aviation data and calculations by onboard computing equipment (e.g., auxiliary performance computers or laptop auxiliary performance computers) to assist aircraft operations. EFBs may also host new types of database information and applications. Three distinct classes of EFB hardware devices are defined according to their relative integration with onboard resources such as electric power, data connectivity, and mounting.

- Class 1 hardware are portable COTS laptop or pen tablet computers with software applications that can include electronic documents, performance calculations and charts. Class 1 do not require certification, but they must be stowed for takeoff and landing. These devices are mostly employed in training and flight planning and for use with reference manuals and in performance calculations.

- Class 2 hardware are semipermanent in that they can dock with a certified crashworthy mount, can be powered all of the time and can tap into noncritical aircraft systems, allowing for cabin video displays and aircraft health monitoring and reporting, or links to an onboard file server.

[35] If the NAS does not use public IP addresses, then this alternative would mean that a NAT would be needed to provide airplane connectivity to non-NAS IP networks such as the Internet.

- Class 3 hardware is either a display mounted permanently to one side of the pilot or incorporated in a multifunction display on the forward panel. In other words, it is an installed piece of avionics that is part of an amended type certificate and, therefore, must satisfy all applicable regulations and policy.

Three types of EFB software applications are also identified.

- Type A software applications allow for report documents to be displayed. No certification is required to duplicate a report in electronic form. However, the FAA flight standards should evaluate and accept the applications for operational use in commercial carrier use, especially if intended to replace the report documents.

- Type B software applications have a higher operational approval level and are more capable and should also be evaluated by the FAA flight standards for operational use.

- Type C software applications have to meet FAA DO-178B assurance and policy and must be approved as part of the certified system. Type C applications also allow the aircraft's own-ship position to be depicted on the ground and in the air for situational awareness purposes (but not for use in navigation).

AC 120-76A specifies how the intersection of specific types of EFB hardware and specific types EFB software need to be designed and operated to achieve acceptable safety assurance.

EFBs fulfill many valuable aircraft functions. For example,

- EFBs often provide required flight performance data, including weight and balance, route and weather conditions, air traffic control inputs, and updated flight manuals.

- updating report documents proved to be a nuisance, with some aircraft needing updates several times a year. When that happens, that aircraft type fleet-wide would have to be upgraded, a very manually intensive process.

- MD-11 aircraft in the late 1990s were equipped with a central fault display information unit (CFDIU), a system that electronically queries various aircraft systems for faults and reports nonspecific information to pilots, in addition to storing diagnostic information that mechanics retrieve after landing. To make better use of the monitoring capability, however, FedEx developed and installed an onboard maintenance terminal (OMT), a rugged laptop computer with a small touch screen, which currently would be called a Class III EFB given its rigorous certification and integration with aircraft subsystems. Installed in the flight engineer's station, the OMT would query the CFDIU for information about faults and then radio the results to the ground through the Aircraft Communications and Reporting System (ACARS) so that maintenance teams could prepare for an aircraft's maintenance needs at the next stop. The EFB converted what had been a reactive system into a proactive one. If a redundant heating element in a pitot tube had burned out during a flight, for example, the MD-11 CFDIU would not have alerted the pilots because the failure was not a flight-critical item, leaving mechanics to find the discrepancy when downloading the CFDIU at a maintenance stop. With the OMT, however, ground crews now would be alerted via ACARS and would be ready to fix the problem at the next stop rather than having to find out about the problem later.

Some military airplanes are designed so that classified mission critical functionality (not flight) resides on laptop computers. The configuration of these laptop computers must be maintained to guarantee that they cannot corrupt other computer systems on the airplane network. This report presumes that EFB functionality must similarly be protected from unauthorized modification that could compromise the integrity of the data or affect other networked systems. The latter includes explicit protection against the introduction of viruses, worms, or other types of malware. Like military laptop computers, the EFB must be controlled through policies and procedures commensurate to its level of security (safety).

This architecture requires that EFB devices must be certified and deployed in conformance with the architecture just like any other networked nonpassenger device within the aircraft. Because these devices are COTS computers, this report states that they cannot themselves be certified at any higher-assurance level. Specifically, the viability of their security controls directly relies upon the vicissitudes of administrative configuration and management oversight and they are directly vulnerable to the problems discussed in sections 4.1 through 4.4. For this reason, any coupling of EFB devices within higher-assurance environments must occur via HAGs. Since HAGs are tailored for specific deployment environments, this requirement implies that EFP functionalities be directly coordinated with specific HAG devices in an FAA-supervised manner.

9.5 UPDATING SECURITY PROTECTION SOFTWARE.

Higher-assurance devices need to be designed so that they cannot be mismanaged or misconfigured.

By contrast, the security controls of lower-assurance devices have dependencies upon the vicissitudes of administrative configuration and management oversight. They are also often directly vulnerable to the problems discussed in sections 4.1 through 4.4. Nevertheless, their security protection software and update policies and procedures should be assessed as part of the certification process. That process should directly assess all networked devices in terms of documented network threats. However, the process also needs to evaluate lower-assurance devices in terms of their participation within a standard airborne defense-in-depth security architecture. That security architecture must conform to IATF-recommended and COTS supported protection mechanisms. These lower-assurance devices must be configured and managed so that they support the aircraft’s defense-in-depth security design and the certification process should ensure that this is possible.

The IA security design shall also address life cycle control issues (see section 5.1). The criticality of assured software update procedures and their potential safety impact must be considered and acceptable procedures developed. Updates to antivirus software signature files occur regularly and do not change the executable software on the computer; therefore, the safety impact of this type of update may be considered low as long as the integrity of the software module can be verified. Installation of OS patches to fix a reported security flaw will have a higher potential impact on system safety.

Many commercial companies require extensive testing before deploying these types of vendor patches. Similar procedures need to be followed on at least a fleet-wide basis. The DoD also has policies for testing and deploying security patches (e.g., the information assurance vulnerability alert process). Similar processes and procedures should be part of the aircraft's software update system that was discussed in section 7.1. Updates should only be authorized to become available to aircraft after a level of analysis, testing, and verification commensurate with the safety criticality of the system they are updating has been completed. Updates should only occur within the aircraft after the integrity and authorization of the update package is established.

As discussed in section 6.1.1, the U.S. DSS [81] provides a mature foundation to enable secure software load deliveries (new parts, security patches, software updates, etc.), including the update of protection software. The DSS standard provides an explicit mechanism to ensure the authenticity and integrity of signed software. The signer's PKI identity is provided as a constituent part of the signature. Should the signing have occurred within the auspices of officially sanctioned and well-defined FAA processes and mechanisms, then that signed identity can be leveraged to provide authentication and authorization within the airplane to determine whether the received code is authorized and trustworthy. Once that determination has been made, then the FAA-approved onboard software update system can securely distribute the software to update the appropriate device in a safe manner. This process is discussed in section 10.6.

9.6 RESPONDING TO SECURITY BREACHES.

The aircraft's IATF conformant defense-in-depth security design will attempt to block those security attacks that can be prevented, detect those that cannot be prevented, respond to those that are detected, and continue to operate through those that cannot be stopped. If the aircraft system architecture adequately addresses these four steps (see section 5.1), then analysis of onboard security failures that do not adversely affect safety of flight can be handled as maintenance events.

Responding to security breaches is a policy issue, so the stakeholders (manufacturer, owner, government agency, etc.) should determine what type of network monitoring to conduct and how to respond to incidents. There are a wide range of policies in the commercial and DoD domains for incident response that could be considered; however, the engineering process should focus on eliminating any safety-related events.

The flight crew will probably not have the expertise or time to perform anything beyond a minimal response to a security breach. The only exception would potentially be to address a safety condition. If the issue directly impacts the operational safety of the aircraft, then the pilots must be alerted.

In section 6.1, the impact of security controls upon airplane safety was considered. The architecture recommended by this study explicitly has focused on safety within networked environments. If the certification of networked nonpassenger airborne devices is trustworthy, the only security breaches that could directly affect aircraft safety would probably be associated with either the integrity or availability (or both) of networked airborne systems. Unfortunately, this also includes the possibility of (accidental) misconfiguring networked devices (e.g., misconfiguring the aircraft's ASBR). The danger from device misconfiguration is a very significant issue for networked systems in general. For this reason, high-assurance devices should be used for all network critical functions to the greatest extent possible because high-assurance devices need to be designed so that they cannot be misconfigured.

Because the critical airborne systems are protected within VPN enclaves, any hostile integrity or availability attack upon those networks or systems would require considerable sophistication on the part of the attacker (unless the vulnerability was caused by device misconfiguration) and would directly reflect significant aircraft design or process deficiencies potentially affecting other aircraft as well. Pilots and crew cannot be assumed to possess the computer and network knowledge to address these types of potentially sophisticated problems. Rather, pilot or crew members need aids that enable them to easily determine the nature of the problem (e.g., an error code or other monitoring status event) so that they can contact experts on the ground to determine remedial responses, just as they do for mechanical failures. In any case, the stakeholders need to anticipate this possibility and determine how ground-based entities should automatically receive and log real-time reports of all airplane safety-related failures. Operational logs should also be maintained and recorded within the airplane itself (hopefully integrated with airline maintenance processes), but safety-related incidents should also be reported to the ground in real time. If the aircraft crashes, there must be adequate information available to determine the root cause of the failure to prevent it from happening again.

9.7 ACCESS TO AIRCRAFT DATA.

Privacy is one of the elements of security engineering. A secure architecture does not necessarily guarantee privacy of all information stored on the system; rather, it will identify those data elements that must be kept confidential and will provide sufficient mechanisms to protect the data from credible threats. Airplane operators may have information that needs to be protected for business reasons; however, exposure of that information would not represent a safety concern for the airplane. The physical location of a plane or some other characteristics of its control channels may be considered sensitive. A credible threat scenario would generally be required as part of the safety and security methodology. Lacking a credible threat scenario, no countermeasures would be recommended. Privacy-enforcing mechanisms may still be warranted to protect sensitive company information or sensitive privacy information about humans in conformance to international or local law, but that would likely be outside of the certifier's scope unless there was a safety issue involved.

9.8 ADEQUACY OF EXISTING REGULATIONS.

Current certification guidelines focus on safety of flight issues. These are distinct from security issues that are commonly addressed by a security engineering process. For this report, the focus was on processes and procedures for identifying those security issues that may impact safety of flight. The addition of LANs to airplanes and interconnections with external public networks exposes previously isolated computational systems to new classes of failures resulting from both accidental as well as intentionally malicious attacks that could affect safety of flight.

Section 7 has suggested specific extensions that are needed to extend current FAA policy to address the additional threats and issues that occur in networked airborne environments. Sections 10.1 and 10.3 directly address the adequacy of existing regulations. Because this topic was a central element of phase 2, the more complete response to this topic occurs in section 10, with this section primarily being an initial description.

9.9 GROUND-TO-AIR COMMUNICATION.

This report recommends that the signals in space (e.g., radio or satellite communications) used for ground-to-air communications must use transport security cover (i.e., encryption of the wireless signal in space occurring at the OSI physical layer). This hinders nonauthorized entities from eavesdropping upon these communications and discourages attempts to potentially inject false communication signals into the data stream (e.g., possible man-in-the-middle attacks). However, these links will remain potentially vulnerable to availability attacks caused by hostile jamming unless mitigation techniques such as AJ waveforms or LPI/LPD waveforms are used.

9.10 WHAT IS THE EFFICACY OF CYCLIC REDUNDANCY CHECKS WITH RESPECT TO SECURITY?

Software parts are currently assured, in many cases, by having a 32-bit polynomial cyclic redundancy check (CRC) wrapped around each part packaged with other identifying information (aircraft type/serial, system part numbers, software part number, etc.) and then that package is wrapped within another CRC. This helps to ensure not only nontampering (internal CRC) but also error-free transmission of the software part and the entire data package (wrapping CRC).

This approach has semantically overloaded the CRC concept to handle two different purposes:

- Polynomial codes (CRCs) are mechanisms commonly used within data communications to detect and fix transmission bit errors. Industry uses different polynomial coding techniques in different environments to address specific network requirements. The wrapping CRC function of the previous paragraph corresponds well with this use case.

- The internal CRC is intended to provide identity and integrity protections for received software parts.

This study states that it is entirely appropriate to use CRCs as polynomial codes to assist in transmission bit error detection and correction. This is, after all, the historic reason for which CRC technology was created.

However, this study states that it is inappropriate and risky (potentially dangerous) to use internal CRCs to provide identity and integrity protections (i.e., the inner CRC) within networked environments. The United States and world standard mechanism by which the latter technique is securely accomplished is by code signing in conformance with the U.S. Federal

DSS (FIPS 186 [81]). Code signing is widely used by both government and industry (e.g., Java code signing). FIPS 186 was discussed in section 6.1.1 (see figures 24 and 25).

FIPS 186 has significant security advantages when compared to CRCs:

- FIPS 186 provides a high-assurance mechanism to establish identities. In most implementations, these identities are assured and certified by a highly trusted subject (i.e., the CA). Also, if the identity is subsequently modified after signing, that modification will be detected by the FIPS 186 verification process. By contrast, the identities of the CRC approach are not verified by a trusted third party or by any other mechanism (i.e., there is no mechanism to verify that the identity is what it claims to be) nor is there a mechanism to discern whether the identity was changed (modified) or not over time.

- FIPS 186 provides a superior approach to integrity protection when compared to CRCs. When CRCs are used for integrity, information (e.g., software, identities) can be modified and CRCs can be recomputed during man-in-the-middle attacks by the attacker in such a way that the received software parts can still pass the CRC checks. However, any attempt to alter FIPS 186 message digests (one-way hashes) will be detected during the FIPS 186 verification process (see figure 25). Thus, the integrity protection of all signed information, including both code and identity information, is trustworthy when using FIPS 186. However, the integrity of the CRC approach is questionable.

- FIPS 186 provides a mechanism to authenticate the established identity of the signer (if required) using a highly assured authentication mechanism based on PKI technology.

- FIPS 186 provides very strong nonrepudiation assurances, but CRCs do not have any nonrepudiation attributes.

10. ANSWERS TO THE PHASE 2 QUESTIONS.

This section discusses several certification concerns that were identified during the original FAA Screening Information Request for this study. These specific questions formed a starting point for the work performed in phase 2 of this study. The exemplar architecture presented in section 8.3 describes the generic airborne network environment that identifies how many of these specific questions should be answered.

10.1 ARE CURRENT REGULATIONS ADEQUATE TO ADDRESS SECURITY CONCERN?

The current regulations need to be extended to address network risks. Networks have very different attributes than the complex systems that the FAA has addressed to date. Section 7 identified specific changes needed to extend DO-178B and ARP 4754. However, other FAA regulations also need to become similarly enhanced. For example:

- ARP 4761 section 4.4. Common Cause Analysis is unlikely to recognize the gamut of possibly subtle effects resulting from the postattack actions of a compromised network device. This is true for all analysis mechanisms: zonal safety analysis, particular risks analysis, and common mode analysis.

- ARP 4761 section 5 states:

  > "Where the detection method is identified to be provided by test, assurance must be provided that the test procedures in fact detect the latent failures of concern."

  However, "failures of concern" in networked environments include latent software bugs that may not become visible or known until attacked. This possibility was not considered by ARP 4761.

- Similarly, the functional hazard assessments (see ARP 4761 Appendix A) also need to address software integrity issues (including software downloads and updates), network availability, and network security integrity and availability.

10.2 HOW DOES SECURITY ASSURANCE FIT INTO OVERALL CERTIFICATION PROCESS?

Security assurance is needed to provide integrity and availability protections to ensure that the DO-178B and ARP 4754 safety protections remain viable over time. If the Biba Integrity Model is used to extend ARP 4754 into network environments as this study recommends, then a mapping between the integrity of security controls and the inherent DO-178B and ARP 4754 safety concepts is needed. The nature of this mapping needs to be further studied, but the current study recommends that insights from the University of Idaho's study [72, 73, and 93] be used to provisionally equate the CC's EAL 5 with DO-178B Level A (see section 6.5).

10.3 WHAT SHOULD NETWORK SECURITY ASSURANCE PROCESS CONTAIN TO MEET XX.1309?

This study's conclusions and recommendations section (see section 11) together with the exemplar airborne network architecture (see section 8.3) provides the answer to this question. XX.1309 mentions many practical and important issues that the recommended architecture directly seeks to address and mitigate. Nevertheless, the current text of XX.1309 contains many statements and concepts that will be challenging to achieve in airborne network environments:

- The meaning of the word "system" in Section 23.1309 changes significantly within the context of an airborne environment. For one thing, systems become arbitrarily large in networked environments and, unless partitioned by VPNs, theoretically include all the devices and humans that can directly or indirectly access any part of the network. This creates the potential for danger and risk within airborne network environments in that equipment, which had no potential safety hazards in nonnetworking environments may have direct and potentially catastrophic safety effects through their fate sharing relationship with other equipment in networked environments. Attackers could potentially leverage the lower-assurance items to attack the higher-assurance items by hostilely changing the environment in which the higher-assurance items operate. The goal of the network security assurance process, which adds security controls within a Biba Integrity Model-based architecture, is to address and mitigate these dangers.

- Issues arise in regard to Section 23.1309 B 3:

> "Warning information must be provided to alert the crew to unsafe system operating conditions and enable them to take appropriate corrective action."

It may be challenging to warn against unintended or nonanticipated interactions resulting from other network-resident items that have no functional relationship to the system in question. Also, the attack vectors of crackers (hostile human attackers) are difficult to predict because the attacks are constantly evolving. Given this, it is unlikely that many dangers may not be discerned until it is too late.

- Issues arise in regard to Section 23.1309 B 4:

> "Compliance with the requirements of … may be shown by analysis and, where necessary, by appropriate ground, flight, or simulator tests."

Analysis is unlikely to address or recognize the gamut of possible subtle effects resulting from the postattack actions of a compromised network device. Similarly, preattack system interactions that these types of tests would address may have little relationship to the modified system interactions that occur during or after attacks.

- Issues arise in regard to Section 25.1309 b:

> "(b) The airplane systems and associated components, considered separately and in relation to other systems, must be designed so that – (1) the occurrence of any failure condition which would prevent the continued safe flight and landing of the airplane is extremely improbable, …"

This goal is difficult to achieve for networked software for fly-by-wire designs unless flight critical systems are partitioned by VPNs (or by an equivalently appropriate partitioning approach for networks) to limit and constrain unintended interactions that may occur during or after the system has been attacked.

- Issues arise in regard to Section 25.1309 d:

> "(d) Compliance with the requirements of paragraph (b) of this section must be shown by analysis, and where necessary, by appropriate ground, flight, or simulator tests."

Unless the safety risks of a networked system have been controlled by leveraging the Biba Integrity Model, any such analysis would be improbable to perform adequately because of the many items involved and their many possible (potentially very subtle) interactions. Any such tests would be for the preattack environment and thus would represent an ideal that may become greatly modified during or after attacks. Many of these issues are addressed in the control life cycle concepts that are an integral part of the IATF defense-in-depth approach.

- Issues arise in regard to Section 25.1309 e:

> "(e) Each installation whose functioning is required by this subchapter, and that requires a power supply, is an 'essential load' on the power supply. …"

The same logic that Section 25.1309 e explains in regard to power supplies is also needed in networked environments to be applied to all possible software interactions that could affect aircraft operation. This includes obvious as well as subtle affects, intended as well as nonintended, and preattack as well as postattack variants. These of issues are addressed in the control life cycle concepts that are an integral part of the IATF's defense-in-depth approach.

10.4 HOW WILL CONTINUED AIRWORTHINESS AND MAINTENANCE BE ADDRESSED?

The conclusions (see section 11) and exemplar airborne network architecture (see section 8.3) addresses how this study recommends that airworthiness be addressed.

Maintenance in networked software environments can potentially differ significantly from current practice, depending on the actual software design, because authorized maintenance personnel no longer need to be physically proximate to the airplane to maintain its software systems. Maintenance in networked environments requires a robust authentication of the maintainer. This study recommends that maintenance personnel be authenticated by two factored authentication systems. For example, the administrator's PKI identity (presuming that the civil aeronautical community selects PKI for its authentication technology) coupled with either what he knows (e.g., a pass phrase) or what he is (i.e., biometrics). It is often advisable that administrative authorizations be restricted in terms of separation of duties with least privilege. For example, different people are authorized to administer airborne security configurations than those who are authorized to handle the non-security-related network management functions, such as downloading software.

It is important that all activities performed by administrators be automatically logged. At a minimum, the log files should state exactly the actions performed by the maintenance person, contain the individual identification of the specific maintenance personnel who performed it, as well as a timestamp and the identification of the networked device from which the administration occurred. All log records should be protected against modification or erasure. One possible approach is to keep the log information both on the aircraft and on the ground and to create an alarm whenever the two copies contain different information (e.g., produce different hashes).

10.5 HOW CAN IT BE ENSURED THAT NETWORKED SYSTEMS CANNOT IMPACT SAFETY?

The recommendations and exemplar airborne network architecture of this study are the answer to this question. For example, see figure 32.

10.6 WHAT SHOULD THE PROCESS BE FOR UPDATING SECURITY PROTECTION SOFTWARE?

The aircraft design should specify the mechanism by which security protection software is updated. It is important that security protection software be updated using the same processes and the same FAA-approved system that handles the issuance of versions of all other aircraft software.

The system should include the following concepts: the FAA should ensure that a secure, ground-based software storage facility is created to house authoritative versions of aircraft software. All authorized versions and variants of airborne software are stored in this secure facility. An authorized human signs each software item previous to storing within this secure facility using the U.S. Federal DSS (FIPS 186). Authorized administrative personnel or systems securely retrieve the appropriate software from the secure facility and download it to the target device within an airplane via formally established processes. This could potentially occur during flight if doing so will not have a detrimental safety impact. To download this software, the administrator will need to establish his or her authentication credentials and to become authorized to download the software via the airplane software download system. That software download system then checks the DSS signature of the software that has been securely retrieved from the secure software storage facility to verify that

- the individual who originally signed that software is authorized to sign software for that airline.
- the signed software has not been modified subsequent to signing.
- the signed software is indeed intended to be deployed onto the device the administrator is attempting to download it onto (including being the appropriate variant).

The aircraft's software download system will only install the retrieved official software into the target device if it successfully passes all three checks. Regardless of whether the checks pass or fail, the maintenance event must be logged, listing the identity of the administrator, a timestamp, what was attempted, and the action taken.

10.7 HOW CAN SECURITY BREACHES BE HANDLED?

The security control life cycle (see section 5.1), which is associated with the IATF defense-in-depth concepts, addresses this issue, stating that it contains four different types of control elements:

- Protection—This study has focused on this part of defense, which is most clearly seen within the exemplar network airborne architecture.

- Detection—The architecture needs to include mechanisms (e.g., sensors) to discern that successful attacks have occurred. This report has only mentioned two such mechanisms, the deployment of Tripwire-like software integrity system and the systematic use of log files. Although not mentioned in this study, a variety of other detection mechanisms should be enabled within a real-life deployment:

  - The firewall, packet filter, and VPN gateways could be configured to provide alerts for certain types of identified behaviors.

  - The deployment would directly benefit from having a NIDS closely associated with the firewall if SWAP issues are not a problem.

  - The deployment should have well-thought-out network management capabilities, including the ability to fuse together health reports (e.g., alerts) from many different systems to form a common operational picture.

- Reaction/neutralization—This refers to automated policies that have been created to respond to certain types of events. For example, if a NIDS is deployed, then the NIDS could be potentially configured to provide an automated reaction to certain types of attack signatures. However, in many airborne systems, the reaction capabilities may be limited to providing alerts to the crew (potentially with real-time copies to ground-based administrative entities) that specifically identified problems have been observed. These administrators could then take appropriate steps to address those problems.

- Recovery/reconstitution—The possibility exists that the attacks were so successful that the system as a whole (or specific elements of the whole) is of doubtful integrity. Administrators or crew could theoretically download from the secure ground-based software site preattack versions of all software that they suspect were compromised due to reports from the Tripwire-like software integrity checker or other sources.

Regardless, a constituent part of any security architecture is to design safe, efficient, and secure mechanisms to completely reconstitute the entire system in an effective manner when needed so that the entire system could return to a known preattack state. It is probable that this complete reconstitution capability should only be permitted to occur when the aircraft is on the ground.

11. SUMMARY.

Current civilian aircraft certification safety assurance processes for airborne systems and equipment are based on ARP 4754, ARP 4761, and various certification authority advisory material (e.g., AC 25.1309-1A) and aircraft manufacturer standards. Civil airborne system software assurance is based on DO-178B, which defines a structured, rigorous development and verification processes for assurance of the embedded software, and other various certification authority and industry policies and standards. ARP 4754 provides guidance for the system development processes to address the safety issues that arise from highly integrated or complex airborne system relationships. It provides guidance for conducting system safety assessments, and references ARP 4761, which defines methods and approaches for conducting safety analysis techniques, such as functional hazard analysis, fault tree analysis, and failure modes and effects analysis.

Approving networked airborne systems should be recognized as being a significant extension to ARP 4754. Networked systems differ from the current ARP 4754 environment in several significant ways. Networked elements are systems that include all of the networks and their constituent elements and users to which the network is directly or indirectly attached. Networks are, therefore, arbitrarily huge, and the many interrelationships of the system items are often too subtle to discern. Networks are inherently complex systems in which every item in the network is inadvertently integrated, regardless of whether those items share any common functional goal. Approval of networked entities must now also address possible network interactions that occur during, and result from, network attacks. The various networked elements potentially have a fate-sharing relationship with each other because any compromised network entity can theoretically be used to attack other networked items or their shared network environment. Embedding airborne software within network systems represents an extension of the ARP 4754 environment to networked items that share limited common functional relationships with each other. This is because entities or components of a system are connected into a common network environment regardless of the original functional intent of the system design (e.g., multiple aircraft domains can be connected by a common network system).

Networks are inherently hostile environments because every network user, which includes both devices (and their software) and humans, are potential threats to that environment. Networked environments and the entities that comprise them need to be protected from three specific classes of threat agents: (1) the corrupted or careless insider, (2) the hostile outsider, and (3) client-side attacks. Because of these dangers, ARP 4754 needs to be extended for networked environments by ensuring network security protection and function/component availability and integrity. This, in turn, implies the need to strategically deploy IA security controls within network airborne systems.

Safety and security have, therefore, become intertwined concepts within networked airborne environments. Security engineering addresses the potential for failure of security controls caused by malicious actions or other means. Safety analysis focuses on the effects of failure modes. The two concepts (safety and security) are, therefore, directly related through failure effects. A shortcoming of either a safety process or a security process may cause a failure in a respective system safety or security mechanism, with possible safety consequences to the aircraft, depending on the specific consequence of that failure.

Previous studies have sought to address airborne software safety and security by correlating DO-178B safety processes with CC security processes. This correlation produces necessary but inadequate results. It is inadequate because it lacks mathematical rigor and therefore produces ad hoc conclusions. The results are ad hoc because even when safety and security are correlated, they are nevertheless distinct concepts from each other, addressing very different concerns.

This report states that the primary issue impacting network airborne system safety is how to extend existing ARP 4574, ARP 4761, DO-178B, and DO-254 assurance guidance processes into networked systems and environments in a mathematically viable manner. This study recommends to extend these processes into arbitrarily vast network environments in a mathematically viable manner by using the Biba Integrity Model framework. This report maps current DO-178B and ARP 4754 processes into the Biba Integrity Model framework using well-established system security engineering processes to define airborne safety requirements. It applies best current information assurance techniques upon those airborne safety requirements to create a generic airborne network architecture.

Since the Biba Integrity Model is an integrity framework, it has a natural mechanism for relating safety and security concepts in terms of their respective integrity attributes. Nevertheless, this study recommends that the model be implemented solely within the context of existing FAA safety processes. This results in airborne network systems being organized into networks that operate at specific safety integrity levels (e.g., the DO-178B software levels).

There are fortuitous secondary effects from using the Biba Integrity Model to extend current FAA processes into networked environments, which stem from its role as the direct analog of the Bell-LaPadula Confidentiality Model. The Bell-LaPadula Confidentiality Model forms the framework for confidentiality within U.S. DoD information processing. Consequently, the application of the Biba Integrity Model to airborne system assurance processes results in an airborne network architecture that remarkably resembles the emerging DoD network architecture, the global information grid, despite their very different underlying goals. Consequently, the generic airborne network architecture identified by this study greatly resembles the DoD's GIG architecture. While military technologies could be used to implement the airborne network architecture, this study recommends the use of civilian IPs deployed as a virtual private network. In addition, the similarities between the Biba Integrity Model and the Bell-LaPadula Confidentiality Model may result in increased synergies between DoD and FAA certification processes.

Deploying airborne systems into networked environments means that the FAA system safety assessment (ARP 4761), system development (ARP 4754), software assurance (DO-178B), and complex electronic hardware assurance (DO-254) processes need to be extended to address and mitigate network threats. For example, although security is primarily a systems concept involving system issues (e.g., ARP 4754), the Biba Integrity Model relies upon the networked items having integrity attributes that function at a known assurance level (e.g., specific DO-178B software levels). This means that the processes for developing those items for network environments should be extended to address network attack risks. The concept of high-assurance software in networked environments should therefore mean that items and systems will behave in the same manner before, during, and after network attacks, i.e., be immune to potential network-based threats. Exploits in network environments leverage latent software blemishes so that software items are subject to misbehavior, corruption, or compromise, possibly including being used as a launching pad to attack other systems and items. Current DO-178B processes do not currently include mechanisms to identify and fix well-known network attack vectors. This study identifies specific additional tests to perform that function. Unfortunately, software testing alone cannot result in high-assurance software. This is because tests only identify the flaws for which the tests are designed to identify, they cannot guarentee the absence of other flaws that were not addressed by the test suite. There is no existing security theory or process that can be leveraged to produce warranteed high-assurance results for networked environments. This is a very significant certification issue. Until a solution for this problem is found, this study recommends that the FAA ensure that high-assurance software complies with formal models and receives a rigorous line-by-line code inspection to demonstrate weaknesses that can be hostilely attacked. Software will also need to be verified when integrated in reapproved network environments.

11.1 FINDINGS AND RECOMMENDATIONS.

The following are the findings of this report:

1. The primary issue impacting network airborne system safety is how to extend existing ARP 4574, ARP 4761, DO-178B, and DO-254 assurance guidance processes into networked systems and environments in a mathematically viable manner.

2. Security models exist (e.g., Bell-LaPadula Confidentiality Model, Biba Integrity Model, Clark-Wilson Integrity Model) that are directly applicable for extending security or safety policies and processes into arbitrarily large and complex networked environments. The models map the policy goals to information system terms by specifying explicit data structures and the techniques necessary to enforce the policy and processes.

3. An attribute of high-assurance systems is that they cannot be misconfigured.

4. VPNs are viable mechanisms to partition network systems in accordance with ARP 4754 Section 5.4.1.1.

5. Airborne network environments are inherently complex integrated systems. Every entity in a network is potentially integrated via fate sharing unless explicitly separated by network partitions (i.e., VPNs). (Note: even though VPNs provide secure network partitions, this study recommends that VPN techniques be applied within a larger defense-in-depth context.)

6. Safety and security are intertwined concepts in airborne networked environments. Security controls (primarily for integrity and availability) need to be introduced if safety integrity is to be preserved within airborne networked environments. The following is

the minimal subset of security controls that have been identified by this study: VPN encapsulation, packet filter, firewall, ASBR, high-assurance LAN, and QoS.

7. Existing airborne system assurance processes need to be extended to recognize that networks are a complex integrated system with unique attributes. For example, ARP 4754 processes need to recognize that networks are potentially hostile environments and that humans are a constituent element within networked systems. Human access to networks should not be solely equated to the humans who are authorized to access airborne networks. Rather, it should also consider individuals who are only authorized to access remote networks to which the airborne network is indirectly linked. If airborne networks are directly or indirectly connected to the Internet, this means that over one billion people can theoretically potentially access airborne networks. Consequently, the processes need to be extended to address possible network attack threats upon the integrity and availability of the system and its items. This requires an assured software download process for airborne software using FIPS 186 (i.e., the U.S. Federal DSS [81]). A secure mechanism that automatically verifies the continued integrity of deployed airborne software items within airborne networks is also needed.

8. It is entirely appropriate to use CRCs as polynomial codes to assist in transmission bit error detection and correction across networks and data buses. However, it is inappropriate (and risky) to use CRCs for software identity and integrity protections within networked environments. Rather, document and code-signing mechanisms conforming to U.S. Federal DSS (FIPS 186, [81]) need to be used instead.

9. The networks of the NAS and the worldwide ground networks that communicate with airborne networks need to be designed with an architecture and design that is consistent with that used by airborne networks if the security and safety provisions of airborne networks are to be preserved. Specifically, ground-based entities that communicate with items or systems located within airborne network partitions (i.e., VPN enclaves) must themselves be within the same VPN enclave network partition as the airborne systems with which they communicate.

10. Items and systems that have been assured for stand-alone system deployments should be reassured whenever they are deployed within networked environments in accordance with extended airborne system assurance processes that support network deployments. Former assurance results must be reassured (revalidated and reverified) on an entity-by-entity basis before the device or software component is deployed in networked environments.

11. Larger software implementations (i.e., large numbers of lines of code) pose certification challenges for networked environments because of potential vulnerabilities to attack caused by (possible) latent software bugs. Large software programs or applications are more vulnerable in the general case because their large size increases the probability of latent blemishes within the code that can be exploited by network attacks.

12. COTS computer systems cannot be adequately secured within large network environments, in general, because their security controls cannot be trusted to perform as intended when attacked. These devices contain potential vulnerabilities potentially affecting security and safety of other networked entities whenever they are deployed within large networks. COTS computer systems, and the applications they support, cannot be high assurance.

The following are the recommendations of this study (see section 8.3 for a generic safety and security design implementing these recommendations and safety requirements).

1. Existing ARP 4574, ARP 4761, DO-178B, and DO-254 assurance guidance processes be extended into network environments by using the Biba Integrity Model framework to define network safety and security assurance concepts.

2. The Biba Integrity Model be implemented solely within the context of existing FAA safety processes. This results in airborne network systems being organized into networks that operate at specific safety integrity levels (e.g., the DO-178B software levels).

3. ARP 4754 and FAA policy be extended to address attack prevention and mitigation by using security controls. IA controls need to comply with best common IA practice, which is defined by the NSA's IATF [50]. These controls need to be implemented in accordance with best current defense-in-depth practices.

4. Aircraft be defined as Mobile ASs, which have embedded VPN network enclave partitions, each of which operates at a specific assurance level.

5. The aircraft should be configured as a mobile AS that moves in reference to other ASs within the larger worldwide aeronautical system. In this approach, each individual networked entity within aircraft is IP addressed and the network topology changes that occur as the aircraft moves are handled by the BGP protocol that links the aircraft to other ASs. IP addressing issues may arise with this model depending on whether the aircraft's IP addresses are associated with a specific service provider (e.g., CIDR; see RFC 1517) or not.

6. DO-178B and ARP 4754 processes be extended to include security vulnerability penetration tests of the integrated airborne network, systems, and each of its constituent items prior to initial certification and deployment. This includes examining their actual vulnerability to attacks as shown in appendix A (e.g., network mapping, vulnerability scanning, penetration testing, password cracking, etc.).

7. Devices operating at specific criticality levels (i.e., failure condition categories, ARP 4754 system development assurance levels, DO-178B software levels, DO-254 Hardware Design Assurance Levels) should be organized into specific network partitions (VPN network enclaves) that operate at a specific assurance level in a manner parallel to the DoD classification levels. Network enclaves for IP networks should be established by

leveraging IPsec's ESP in tunnel mode in direct parallel to current U.S. DoD COMSEC and some civilian VPN practices.

8. While military technologies could be used to implement the airborne network partitions, the use of civilian Internet protocols be deployed as a virtual private network. Specifically, this study recommends that the airborne community use the IETF's L3VPN IPsec variant of RFC 4364 [99] for its VPN technology.

9. Because of SWAP considerations and the network management issues associated with how to manage VPN enclaves, the VPN encapsulation be established by means of an encapsulation gateway middlebox, rather than the traditional dual PE and CE router approach (see figure 34) commonly used by reference 99 conformant implementations.

10. Although the network partition capabilities and assurance of VPNs are demonstrably sound, security vulnerabilities (depending on how it is implemented) may potentially be introduced by bringing in network management capabilities into the encapsulating gateways (see section 8.4) that otherwise could not occur within a VPN system. For this reason, this study recommends that the encapsulation gateways be deployed with the following additional defense-in-depth security control protections:

    - Firewall (and, if in a nonair gap target environment, the packet filter as well) to be configured to discard any non-IPsec packets addressed to airborne encapsulating gateways.

    - The encapsulating gateway should also be configured to discard any packet sent to it that does not use IPsec's ESP. It decapsulates and decrypts any received tunnel mode packets and forwards them to the VPN. Received transport mode packets are those communications to the encapsulating gateway itself. All transport mode packets must be successfully authenticated by the encapsulating gateway or else discarded.

    - QoS provisions ensuring that the VPN is provided adequate network capacity (e.g., to avoid DoS) are also needed to ensure the viability of VPN partitioning.

11. The encapsulation gateways will need to be certified as a high-assurance security item (i.e., EAL 5 or higher).

12. Onboard aircraft network LAN implementations should also support physical (i.e., hardware based) network protections to implement integrity enclave separation to physically isolate devices using a common LAN system into networked enclaves on a need-to-communicate basis [9].

13. Network communications between devices connected within each network enclave should be supplemented with IPsec's ESP in transport mode security protections whenever permitted by the specific communications performance requirements.

14. NAS and airborne network architecture and design should follow best common IA security practices [20, 83, and 85].

15. Approaches to authenticate and authorize network managers be carefully considered. This study recommends that administrative personnel be authenticated by two factored authentication systems; e.g., the administrator's PKI identity coupled with either what he knows (e.g., pass phrase) or what he is (i.e., biometrics). It is also recommended that administrative authorizations be restricted in terms of separation of duties with least privilege. For example, different people must work on airborne security topics than can work on other airborne administrative topics.

16. All activities performed by administrators upon aircraft software and systems must be automatically logged. At a minimum, the log files should: state exactly what the administrator did; contain the individual identification of the specific maintenance personnel who did it; and provide a timestamp and the identification of the networked device from which the administration occurred. All log records must be protected against modification or erasure. One possible approach is to keep the log information both on the aircraft and on the ground and create an alarm whenever the two copies contain different information (e.g., produced different hashes).

17. The signals in space (e.g., radio or satellite communications) used for ground-to-air communications must use transport security cover (i.e., encryption of the wireless signal in space occurring at the OSI physical layer). This hinders nonauthorized entities from eavesdropping upon these communications and discourages attempts to potentially inject false communication signals into the data stream (e.g., possible man-in-the-middle attacks). However, these links will remain potentially vulnerable to availability attacks caused by hostile jamming unless mitigation techniques such as AJ waveforms or LPI/LPD waveforms were also used.

18. Airborne or NAS systems should not be designed using technologies that require significant policy complexity for all (or a majority of) the networked devices or a high degree of policy coordination between all of the networked elements (see section 5.7).

19. Aircraft control and the cockpit (pilot) networks or their devices should not be physically accessible by aircraft passengers. If there is any possibility of passengers physically accessing the cockpit (pilot) network, then the high-assurance LAN within the cockpit must be connected to the aircraft control network via the packet filter. Otherwise, the high-assurance LAN in the cockpit can use the same physical high-assurance LAN as aircraft control. The noncockpit crew network devices should also not be accessible by passengers, but the design could accommodate situations in which passengers are not always physically excluded from the area where those devices are located. If physical separation is not possible, crew members must be very careful to not leave open applications running in situations when the crew member is not present (i.e., situations where passengers may access applications that have been opened with crew member authentications).

20. The packet filter in the aircraft control must be configured such that noncockpit crew network cannot address any encapsulation gateway. If the aircraft is using figure 1 target architecture (i.e., no air gap between the passenger and avionics systems), then the packet filter needs to additionally provide the following services:

- No device within the passenger network can access the noncockpit crew network or the cockpit-pilot network.

- No device within the passenger network can send packets to any encapsulation gateways (located within aircraft control).

- The packet filter, or a device closely associated with the packet filter comprising a common system with it (e.g., QoS middlebox), rate limits communications from the passenger network to ensure that passenger communications cannot exceed a certain threshold rate. This provision attempts to ensure that passengers alone cannot cause a DoS attack on the aircraft control's high-assurance LAN by consuming a disproportionate share of its capacity.

21. The firewall needs to be configured to be as exclusive as possible. Because of the presence of passengers in the network in the figure 1 target, the HTTP overt channel vulnerability (see section 4.1 and appendix A.1), unfortunately, cannot be fully plugged, unlike the figure 3 target alternative. However, if aircraft design restricts pilot and crew communications such that they never use HTTP, then the firewall can be configured so that HTTP traffic (i.e., both Port 80 and Port 443) is filtered out by the firewall whenever the packet's destination address is a nonpassenger device. Such a rule would provide pilot and crew devices helpful protection in figure 1 environments. Even if the pilot and crew were permitted to use secure HTTP only (i.e., Port 443), then at least the more dangerous Port 80 transmissions could be filtered. In any case, the firewall needs to be configured so that:

- All fingerprinting attempts (see appendix A, section A.1) originating from outside of the aircraft to any entity within the aircraft will fail (except for those that remain through the HTTP hole).

- All communications to encapsulation gateways from outside of an airplane are blocked by the firewall unless they use IPsec's ESP. (Note: both the firewall and the gateways themselves need to redundantly enforce this same rule for defense-in-depth reasons.)

- The firewall should also be configured to drop all packets originating from outside of the aircraft to IP destination addresses that are not deployed within the aircraft LAN. Please recall that the firewall only has visibility into VPNs since it only sees their encapsulating packet headers, which are solely addressed to encapsulation gateways.

22. If SWAP considerations permit, an NIDS should be deployed that is associated with the firewall system. The NIDS should be configured to recognize attack footprints and be configurable to optionally send alerts to designated crew members or ground systems alerting them should certain types of attacks occur.

23. The ASBR provides BGP connectivity with the remote air and ground networks with which the airplane is communicating. The airplane's ASBR must be configured such that all packets are sent with an ASBR interface, because the IP destination address must be dropped unless they use IPsec in transport mode and come from a network management or IDS device that is local to that airplane.

24. DO-178B should be extended to mitigate network attack vulnerabilities by introducing specific tests into the development processes (e.g., process maturity models, formally verify protocols, software fault injection, model checkers, buffer overflow tests, dead code tests). The software can provide for some self protection, similar to what is currently done for hardware failures (e.g., however, tests alone do not provide assurance, they only identify the presence or absence of problems for items contained within the test suite).

25. For network environments, existing DO-178B assurance processes should include the following three elements for higher-assurance software:

    - A series of penetration tests should be performed upon the completed software item. Specifically, the software (including its OS, if any) needs to be subjected to a range of network attacks described in appendix A. Any problems identified from these attacks need to be fixed.

    - Examine the software under evaluation to verify that its internal construction complies with formal models of software construction such as being modular and layered in terms of a structured presentation within the implementation itself.

    - Conduct a rigorous line-by-line code inspection of the software to demonstrate a lack of bugs that can be hostilely attacked.

    Software items that do not undergo, or cannot pass, these three additional tests cannot be stated to be high assurance when deployed in network environments

26. Very stringent application of existing software certification processes should be used for high-assurance software in networked environments. The line-by-line code inspection requirement for high-assurance software certification should ensure that high-assurance software code bases explicitly use formal software techniques and are comparatively small in size (in terms of the number of lines of code). The inspection should actively seek to identify (and fix) software bugs that can be attacked. The indeterminate number of bugs that are latently present in large-code bases represent unaddressed attack vulnerabilities in networked environments. Current software development methods cannot be trusted to produce high-assurance results unless those results are supplemented

with extensive scrutiny. The larger the code base, the more questionable the quality of the scrutiny. This means that software developers need to actively consider how to create high-assurance software for network environments so that the resulting software can be assured to be as bug free as possible. Until a theoretical solution is devised that produces guaranteed, high-assurance, bug free results, high-assurance software needs to undergo a very thorough (formal) line-by-line code inspection. A possible alternative is for the software developer to assemble high-assurance software modules. The integration of these modules faces the same types of integration issues addressed in ARP 4754, but this may potentially result in an approval approach in which only a select subset of the total software corpus will require a formal line-by-line code inspection.

27. All software in networked environments should comply with the processes established by an FAA-approved software distribution (i.e., storage and download) system. Software development processes need to include concrete plans for how software will be maintained and securely distributed over the software's life span.

28. Software that is currently hosted on COTS OSs should be evaluated to be ported to a more secure foundation. High-assurance software (i.e., Levels A and B) cannot reside on COTS OSs, because COTS OSs are not high-assurance and contain latent vulnerabilities that can be attacked. That software should be either ported to reside on a high-assurance OS or else rewritten to not reside on any OS.

29. The worldwide civil aviation community should identify common solutions for identity (see section 4.8), IP addressing (see sections 5.3 and 5.4), naming, routing (see section 5.5), protocol security (see section 4.5), and authentication (see section 4.9) subsystems. These common approaches need to be realized by consistent technology and configuration choices that produce a coherent worldwide civil aviation network infrastructure. These important technical issues need to be agreed upon by the aeronautical community before airborne avionics systems become networked to other aircraft or ground systems. This is because the safety of networked airborne LAN systems is potentially affected by the quality and integrity of the network system that is created by the worldwide civil aviation community. It is risky to permit networked airborne LAN systems to be created before the worldwide civil aviation community has decided on a common approach to address these key subsystems. Aircraft need to handle identity, IP addressing, naming, routing, protocol security, and authentication in a consistent manner with each other and with civil aviation ground systems if aircraft and NAS systems are to be networked together. The interfaces of both airborne and ground systems therefore need to be carefully articulated and designed if potentially significant security problems are to be avoided.

30. This study recommends that the FAA evaluate using AJ waveforms for air-to-ground communications.

31. Before a worldwide civil aviation network can be deployed, the worldwide civil aviation community explicitly should determine the policies and trust models that will pertain to the worldwide civil aviation network infrastructure.

11.2 TOPICS NEEDING FURTHER STUDY.

This report identifies the following topics as needing further study:

- Pertaining to the last recommendation noted above, what is the trust model between civil aviation regions? Will the trust model for the regions' Level A software network partitions (enclaves) be the same as for their Level C software network enclaves? What is the trust model between aircraft and ground entities? If air-to-air communications occur, what is the trust model between aircraft belonging to different airlines? Will the Level A VPN components of the NAS completely trust European Level A VPN components and vice versa, or will they establish distinct policies and SLA mappings between their components? What security protections (e.g., firewalls) will be inserted to protect the rest of the VPN elements at that safety level from a contamination that occurred within a specific region? How will aircraft that travel between regions maintain their connectivity in a seamless, safe, and secure manner? If air-to-air applications and systems are created, what mechanisms (e.g., firewalls) will protect the VPN at a given safety level in one airplane from (perhaps undiagnosed) misbehaviors occurring in the VPN at that same safety level in a different airplane? What policy systems will govern the interrelationship between aircraft and ground entities? Will SLAs be required?

- The worldwide civil aviation community needs to identify common solutions for identity (see section 4.8), IP addressing (see sections 5.3 and 5.4), naming, routing (see section 5.5), protocol security (see section 4.5), and authentication (see section 4.9) subsystems.

- Because network management issues for airborne networks are directly related to airline, manufacturer, and FAA concept of operations, this study has not provided a well-developed network management recommendation. Nevertheless, these issues need to be competently addressed and a viable network management system needs be designed if airborne LAN systems are to be safely networked. Therefore, network management designs and architectures need to be established for airborne networks.

- Carol Taylor, Jim Alves-Foss, and Bob Rinker of the University of Idaho have studied the issue of dual software certification [93] for CC and DO-178B. Their study suggested that security functionality certified at EAL 5 can be directly compared with DO-178B Level A. This report recommends that their conclusion should be verified by further evaluation and study for the specific issue of establishing assurance equivalencies between CC-certified security controls and safety assurance levels.

- Are there better mechanisms to address the problem of how best to remove latent software bugs that can be attacked from software items in networked environments? While testing probably provides part of the solution, it is obvious that testing alone cannot warranty the absence of bugs for elements unexamined by the test suite. Because of the vast array of possible software bugs that can exist, it is improbable that a complete testing corpus can be created. Therefore, best current practice is to continue to enforce line-by-line code inspection within the certification process for the highest assurance software items. Is there a security model or theory that could be discovered to authoritatively address this issue so that highly assured software could be definitively created within networked environments?

- Industry has experience creating Bell-LaPadula Confidentiality Model, HAGs. However, there is little or no experience creating Biba Integrity Model HAGs. Therefore, there is a need to study and articulate the controls needed within Biba Integrity Model HAGs. This study should distinguish the differences (if any) between Biba Integrity Model HAG technology and Bell-LaPadula Confidentiality Model HAG technology.

- What are the mechanisms that will integrate DoD and FAA certification processes and procedures? Is it possible to create a common certification system that reliably addresses safety in FAA environments and security in DoD environments, and both safety and security in joint certification environments?

- Should Level D software systems be treated as Requirement 1 systems and organized into VPN enclaves as this study currently states (see section 8.2), or should they rather become Requirement 2 systems and not be enclaved into network partitions (VPNs) such as Level E systems?

12. REFERENCES.

1. ARP 4754, “Certification Considerations for Highly-Integrated or Complex Aircraft Systems,” 1996, SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001.

2. ARP 4761, “Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment,” 1996, SAE International, 400 Commonwealth Drive, Warrendale, PA 15096-0001.

3. 14 CFR 23.1309, Equipment, systems, and installations, Revised January 1, 2006.

4. 14 CFR 25.1309, Equipment, systems, and installations, Revised January 1, 2006.

5. RTCA/DO-178B, “Software Considerations in Airborne Systems and Equipment Certification,” December 1, 1992, Prepared by SC-167.

6. RTCA/DO-254, “Design Assurance Guidance for Airborne Electronic Hardware,” April 19, 2000, Prepared by SC-180.

7. Knight, J., “Software Challenges in Aviation Systems,” NASA Grant number NAG-1-2290, 2002. http://dependability.cs.virginia.edu/publications/safecomp.2002.pdf
8. FAA Advisory Circular 120-76A, “Guidelines for the Certification, Airworthiness, and Operational Approval of Electronic Flight Bag Computing Devices,” March 17, 2003.

9. Lee, Y., Rachlin, E., and Scandura, Jr., P., "Safety and Certification Approaches for Ethernet-Based Aviation Databuses," FAA report DOT/FAA/AR-05/52, December 2005.

10. Yost, R., "Airplanes can be Networked," Abstract, *American Institute of Aeronautics and Astronautics*, 2002.

11. Donohue, G.L., "Air Transportation is a Complex Adaptave [SIC] System: Not an Aircraft Design," *American Institute of Aeronautics and Astronautics*, 2003.

12. Buede, D., Farr, J., Powell, R., and Verma, D., "Air Traffic Management Design Considerations," *IEEE AES Systems Magazine*, October 2003, pp. 3-8.

13. U.S. Department of Transportation Federal Aviation Administration Order 1370.82. Subject: Information Systems Security Program, June 9, 2000, Initiated by AIO-1, distribution A-WZYZ-2; A-FOF-O.

14. "The Transnational Dimension of Cyber Crime and Terrorism," Abraham D. Sofaer and Seymour E. Goodman, eds., *Hoover Institution Press*, 2001.

15. Birman, K., "The Untrustworthy Web Services Revolution," *IEEE Computer Magazine*, February 2006, pp. 98-100.

16. Campbell, S., "How to Think About Security Failures," *Communications of the ACM*, Volume 49, Number 1, January 2006, pp. 37-39.

17. "Future Communications Study: Initial Discussion of Radio Frequency Security Requirements, Version 1.5," prepared by FAA ACB-250, April 10, 2005.

18. Mehan, D., "Information Systems Security: The Federal Aviation Administration's Layered Approach," Transportation Research Board National Research Council, November-December 2000, TR *NEWS—Transportation Security—Protecting the System from Attack and Theft, Number 211.*

19. Mehan, D. and Potter, M., "Building Trustworthy Systems: An FAA Perspective," *Software Technology News*, Volume 4, Number 3, Federal Aviation Agency Issue, Data & Analysis Center for Software, Rome, NY, 2001.

20. Federal Aviation Administration Information System Security Technology Overview, Version 2.0, Prepared by The MITRE Corporation for the Office of Information Services, September 30, 2003.

21. Dunn, J.E., "Crypto Malware Close to Being 'Uncrackable,'" *ComputerWorld*, Security, July 25, 2006.
http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9001997&source=NLT_VVR&nlid=37

22. Robinson, D., "Safety Security Type Design Approval Consideration," *2006 Software and Complex Electronic Hardware Conference*, Atlanta, GA, June 28, 2006.

23. Cheswick, W., Bellovin, S., and Rubin, A., "Firewalls and Internet Security, Second Edition—Repelling the Wily Hacker," Addison-Wesley, 2003.

24. Wang, Y. and Strider, M., "HoneyMonkeys: Active Client-Side Honeypots for Finding Web Sites That Exploit Browser Vulnerabilities," Part of Works in Progress at the *14th Usenix Security Symposium*, Baltimore, July 31-August 5, 2005.
http://www.usenix.org/events/sec05/wips/wang.pdf and
http://research.microsoft.com/HoneyMonkey/

25. Weiss, T., "Trojan Horse Captured Data on 2,300 Oregon Taxpayers From Infected Gov't PC," *Computerworld*, Government, electronic version, June 15, 2006.
http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9001222&source=NLT_VVR&nlid=37

26. Spitzner, L., *Honeypots—Tracking Hackers*, Addison Wesley, 2003, pp. 11-12.

27. Ward, M., "Tracking Down Hi-Tech Crime," BBC News, Sunday, October 8, 2006.
http://news.bbc.co.uk/2/hi/technology/5414502.stm

28. Osterman, M., "Malware is Getting Very Serious," *NetworkWorld Magazine*, September 28, 2006.
http://www.networkworld.com/newsletters/gwm/2006/0925msg2.html

29. Messmer, E., "Software Vulnerabilities Already Outnumber Last Year's," *ComputerWorld*, Security, October 9, 2006.
http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9004000&source=NLT_VVR&nlid=37

30. The CERT web page as it existed on January 18, 2006.
http://www.cert.org/stats/cert_stats.html

31. Dunn, J., "China Attacks U.K. Government Using Windows Security Hole," *ComputerWorld* (on-line version), January 25, 2006.
http://www.computerworld.com/securitytopics/security/holes/story/0,10801,108037,00.html?source=NLT_VVR&nid=108037

32. Loscocco, P., Smalley, S., Muckelbauer, P., Taylor, R., Turner, S., and Farrell, J., "The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments," *Proceedings of the 31st National Information Systems Security Conference*, October, 1998, pp. 303-314.
http://www.nsa.gov/selinux/papers/inevit-abs.cfm

33. Skoudis, E., *Counter Hack*, Prentice Hall, 2002.

34. Klevinsky ,T.J., Laliberte, S., and Gupta, A., *Hack I.T.*, Addison-Wesley, 2002.

35. Hatch, B. and Lee, J., *Hacking Linux Exposed*, Second Edition, McGraw-Hill/Osborne, 2003.

36. Mourani, G., *Securing and Optimizing Linux, The Hacking Solution*, Third Edition, Open Network Architecture, Inc., 2002.

37. McClure, S., Scambray, J., and Kurtz, G., *Hacking Exposed: Network Security Secrets and Solutions*, Osborne/McGraw-Hill, 1999.

38. Rubin, A., *White-Hat Security Arsenal: Tackling the Threats*, Addison-Wesley, 2001.

39. Barrett, D., Silverman, R., and Byrnes, R., *Linux Security Cookbook*, O'Reilly and Associates, 2003.

40. Devanbu, P. and Stubblebine, S., "Software Engineering for Security: a Roadmap," ICSE 2000. http://www.stubblebine.com/00icse.pdf

41. Abrams, M., "FAA System Security Testing and Evaluation," The MITRE Corporation, MTR 02W0000059, May 2003.

42. DoD 5200.28-STD, "Department of Defense Trusted Computer System Evaluation Criteria," (TCSEC), December 26, 1985.
http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html

43. U.S. National Computer Security Center, "Trusted Network Interpretation of the Trusted Computer System Evaluation Criteria," NCSC-TG-005, Version 1, U.S. Department of Defense, Ft. Meade, MD, 31 July 1987.

44. Common Criteria for Information Technology Security Evaluation, Part 1, Version 2.1, CCIMB-99-031, August 1999.

45. Common Criteria for Information Technology Security Evaluation, Part 2, Version 2.1, CCIMB-99-032, August 1999.

46. Common Criteria for Information Technology Security Evaluation, Part 3, Version 2.1, CCIMB-99-033, August 1999.

47. Lee, Y.H. and Krodel, J., "Flight-Critical Data Integrity Assurance for Ground-Based COTS Components," FAA report DOT/FAA/AR-06/2, March 2006.

48. Stephens, B., "Aeronautical Telecommunications Using IPv6," http://spacecom.grc.nasa.gov/icnsconf/docs/2003/02_A1/A1-01-Stephens.pdf

49. Rao, S., "Trust and Security in Pervasive Computing," http://www.seinit.org/documents/publication_documents/Rao-Terena-conf-presentation.pdf , see page 12

50. "Information Assurance Technical Framework," Issued by the National Security Agency Information Assurance Solutions Technical Directors, Release 3.1, September 2002, Unclassified. http://www.iatf.net/framework_docs/version-3_1/index.cfm

51. Biba, K.J., "Integrity Consideration for Secure Computer Systems," The MITRE Corporation, MTR-3153, 1975.

52. Biba, K.J., "Integrity Consideration for Secure Computer Systems," USAF Electronic Systems Division Technical Report 76-372, 1977.

53. Xu, K., Hong, X., and Gerla, M., "An Ad Hoc Network With Mobile Backbones," *Proceedings of IEEE International Conference on Communications (ICC 2002)*, New York City, April 2002.

54. Gupta, P., Gray, R., and Kumar, P.R., "An Experimental Scaling Law for Ad Hoc Networks," May 16, 2001. http://black1.csl.uiuc.edu/~prkumar/

55. Lin, C.R. and Gerla, M., "Adaptive Clustering for Mobile Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 7. September 1997, pp. 1265-1275.

56. Gerla, M. and Tsai, J.T., "Multicluster, Mobile, Multimedia Radio Network," *ACM-Baltzer Journal of Wireless Networks*, Vol. 1, No. 3, 1995, pp. 255-265.

57. Krishna, P., Vaidya, N.H., Chatterjee, M., and Pradhan, D.K., "A Cluster-Based Approach for Routing in Dynamic Networks," *Proceedings of ACM SIGCOPMM Computer Communications Review*, 1997, pp. 372-378.

58. Banerjee, S. and Khuller, S., "A Clustering Scheme for Hierarchical Control in Multi-Hop Wireless Networks," *IEEE Infocom 2001*, Anchorage, Alaska, April 2001.

59. Raisinghani, V. and Sridhar, I., “Cross-Layer Feedback Architecture for Mobile Device Protocol Stacks,” *IEEE Communications Magazine*, Volume 44, No. 1, January 2006, pp. 85-92.

60. Fleischman, E., “JTRS WNW Mobility in Tactical Military Environments,” paper #1411 published in the *MILCOM 2005* classified section, May 10, 2005.

61. Jiang, H., Zhuang, W., and Shen, X., “Cross-Layer Design for Resource Allocation in 3G Wireless Networks and Beyond,” *IEEE Communications Magazine*, Volume 43, No. 12, pp. 120-126, December 2005.

62. Fleischman, E., “Mobile Exterior Gateway Protocol: Extending IP Scalability,” paper #314 published in the *MILCOM 2005* unclassified section, August 2005.

63. Feamster, N., Balakrishnan, H., and Rexford, J., “Some Foundational Problems in Interdomain Routing,” Third ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets), San Diego, CA, November 2004.
http://ramp.ucsd.edu/conferences/HotNets-III/HotNets-III%20Proceedings/camera.pdf

64. Wang, L., et al., “Observation and Analysis of BGP Behavior Under Stress,” ACM SIGCOMM Internet Measurement Workshop, November 2002.

65. Xiao, L. and Nahrstedt, K., “Reliability Models and Evaluation of Internal BGP Networks,” *Proc. IEEE INFOCOM*, March 2004.

66. Labovitz, C., et al., “Experimental Measurement of Delayed Convergence,” NANOG Presentation, October 1999.

67. Rosen, E. and Rekhter, Y., “BGP/MPLS IP VPNs,” February 2006, RFC 4364.
http://www.ietf.org/rfc/rfc4364.txt

68. Strassner, J., *Policy-Based Network Management: Solutions for the Next Generation*, Morgan Kaufmann Publishers, 2004.

69. Bellovin, S., “Distributed Firewalls,” *login magazine*, November 1999, pp. 37-39.
http://www.cs.columbia.edu/~smb/papers/distfw.pdf

70. Ioannidis, S., Keromytis, A., Bellovin, S., and Smith, J., “Implementing a Distributed Firewall,” *Proceedings of the 7th ACM International Conference on Computer and Communications Security (CSS)*, Athens, Greece, November 2000, pp 190-199.
http://www.cs.columbia.edu/~angelos/Papers/df.pdf

71. NIST, “Information Technology—Security Techniques—Evaluation Criteria for IT, Security—Part 1: General Model,” December 1999.

72. Alves-Foss, J., Rinker, B., and Taylor, C., "Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems," January 2002.

73. Taylor, C., Alves-Foss, J., and Rinker, B., "Merging Safety and Assurance: The Process of Dual Certification for Software," *Proc. Software Technology Conference*, March 2002. http://www.csds.uidaho.edu/comparison/stc2002.pdf
http://gulliver.trb.org/publications/security/dmehan.pdf

74. Payne, C., Froscher, J., and Landwehr, C., "Toward a Comprehensive INFOSEC Certification Methodology," *Proceedings of the 16th National Computer Security Conference*, Baltimore, MD, September 20-23, 1993, NCSC/MIST, pp. 165-172.

75. Cortellessa, V., Cukic, B., Del Gobbo, D., Mili, A., Napolitano, M., Shereshevsky, M., and Sandhu, H., "Certifying Adaptive Flight Control Software," *Proceedings of the ISACC2000*, The Software Risk Management Conference, Reston, VA, September 24-26, 2000.

76. Ibrahim, L., Jarzombek, J., Ashford, M., Bate, R., Croll, P., Horn, M., LaBruyere, L., and Wells, C., "Safety and Security Extensions for Integrated Capability Maturity Models," FAA, September 2004.

77. Foster, N., "The Application of Software and Safety Engineering Techniques to Security Protocol Development," PhD Dissertation at the University of York Department of Computer Science, September 2002.

78. Roy, A., "Security Strategy for U.S. Air Force to Use Commercial Data Link," *IEEE*, 2000.

79. McParland, T. and Patel, V., "Securing Air-Ground Communications," *Digital Avionics Systems*, DASC 20th Conference, Vol. 2, 2001, pp. 7A7/1-7A7/9.

80. Nguyen, T., Koppen, S., Ely, J., Williams, R., Smith, L., and Salud, M., "Portable Wireless LAN Device and Two-Way Radio Threat Assessment for Aircraft VHF Communication Radio Band," NASA/TM-2004-213010, March 2004.

81. FIPS Pub 186, "Digital Signature Standard," National Institute of Standards and Technology (NIST), 19 May 1994. http://www.itl.nist.gov/fipspubs/fip186.htm

82. Patel, V. and McParland, T., "Public Key Infrastructure for Air Traffic Management Systems," *Digital Avionics Systems Conference Proceedings*, Daytona Beach, FL, October 14-18, 2001, Piscataway, NJ, IEEE Computer Society, 2001.

83. Harris, S., "All In One CISSP Certification Exam Guide," McGraw-Hill/Osborne, 2002.

84. Bell, D.E. and LaPadula, L.J., "Secure Computer Systems: Mathematical Foundations and Model," Technical Report M74-244, The MITRE Corporation, October 1974. (Note:

the following is a pointer to a related article that Bell and LaPadula wrote in 1976 where they cite this reference for their work, as opposed to the more prevalent 1973 reference: http://csrc.nist.gov/publications/history/bell76.pdf)

85. Krutz, R. and Vines, R., *The CISSP Prep Guide*, Wiley Computer Publishing, 2001.

86. Executive Order 12958, "Classified Nation Security Information," April 17, 1995.
http://www.fas.org/sgp/clinton/eo12958.html

87. Executive Order 13292, "Further Amendment to Executive Order 12958, As amended, Classified National Security Information," March 25, 2003.
http://www.fas.org/sgp/bush/eoamend.html

88. Public Law 100-235 (H.R. 145), "Computer Security Act of 1987," January 8, 1988.
http://www.epic.org/crypto/csa/csa.html

89. Title 22, Chapter 1, Subchapter M, "International Traffic in Arms Regulations," Department of State, Revised April 1, 1992.
http://www.epic.org/crypto/export_controls/itar.html

90. MIL-STD 882D, "Department of Defense Standard Practice for System Safety," 10 February 2000.

91. Department of Defense Instruction (DoDI) 8500.2, "Information Assurance (IA) Implementation," ASD(C3I), 102 Pages

92. OMB Circular A-130, "Management of Federal Information Resources, Transmittal 4," November 30, 2000.

93. Alves-Foss, J., Rinker, B., and Taylor, C., "Towards Common Criteria Certification for DO-178B Compliant Airborne Software Systems," Center for Secure and Dependable Systems, University of Idaho, January 2002.
http://www.esds.uidaho.edu/papers/Taylor02d.pdf

94. http://sourceforge.net/projects/tripwire/

95. Ghosh, A., O'Connor, T., and McGraw, G., "An Automated Approach for Identifying Potential Vulnerabilities in Software," DARPA contract F30602-95-C-0282, *Proceedings of the 1998 IEEE Symposium on Security and Privacy*, IEEE Computer Society, May 1998, pp. 104-114.
http://www.cigital.com/papers/download/ieees_p98_2col.pdf

96. Cowan, C., Pu, C., Maier, D., Hinton, H., Walpole, J., Bakke, P., Beattie, S., Grier, A., Wagle, P., and Zhang, Q., "StackGuard: Automatic Adaptive Detection and Prevention of Buffer-Overflow Attacks," DARPA Contract F30602-96-1-0331 and F30602-96-1-

0302, *Proceedings of the 7th USENIX Security Symposium*, San Antonio, Texas, January 1998, pp. 63-78.
http://www.usenix.org/publications/library/proceedings/sec98/full_papers/cowan/cowan.pdf

97. Bolduc, L., "Verifying Modern Processors in Integrated Modular Avionics Systems," AlliedSignal Aerospace, Columbia, MD, 1999.
http://www.chillarege.com/fastabstracts/issre99/99110.pdf

98. Jacklin, S., Lowry, M., Schumann, J., Gupta, P., Bosworth, J., Zavala, E., Kelly, J., Hayhurst, K., Belcastro, C., and Belcastro, C., "Verification, Validation, and Certification Challenges for Adaptive Flight-Critical Control System Software," *American Institute of Aeronautics and Astronautics (AIAA) Guidance, Navigation, and Control Conference and Exhibit*, Providence, Rhode Island, August 16-19, 2004.

99. Rosen, E., De Clercq, J., Paridaens, O., T'Joens, Y., and Sargor, C., "Architecture for the Use of PE-PE IPsec Tunnels in BGP/MPLS IP VPNs," August 2005.
http://www.ietf.org/internet-drafts/draft-ietf-l3vpn-ipsec-2547-05.txt

100. http://www.ietf.org/html.charters/l3vpn-charter.html (Note: this link will only be active for as long as the L3VPN working group will exist in the IETF. After the working group is eventually disbanded, this URL will no longer be valid.)

101. Templin, F., "IPvLX—IP With Virtual Link eXtension," September 22, 2005.
http://www.ietf.org/internet-drafts/draft-templin-ipvlx-04.txt

102. APIM 04-012, "ARINC IA Project Initiation/Modification (APIM)," April 19, 2005.
http://www.arinc.com/aeec/projects/fms/04_012_apim_fmc.pdf

103. Adams, C., "Test Cards for the Airbus A380."
http://www.aim-online.com/PressRelease/afdx.pdf

104. http://www.afdx.net/
http://west.dtic.mil/whs/directives/corres/pdf2/d85001p.pdf

13. RELATED DOCUMENTATION.

Ibrahim, L., Jarzombek, J., Ashford, M., Bate, R., Croll, P., Horn, M., LaBruyere, L., and Wells, C., "Safety and Security Extensions for Integrated Capability Maturity Models," September 2004, FAA, 141 pages.

National Security Telecommunications and Information Systems Security Policy (NSTISSP) No. 11, "National Policy Governing the Acquisition of Information Assurance (IA) and IA-Enabled Information Technology Products," January 2000.

Department of Defense Instruction Number 5200.40, "DoD Information Technology Security Certification and Accreditation Process (DITSCAP)," December 30, 1997.

Barbir, A., Murphy, S., and Yang, Y., "Generic Threats to Routing Protocols," RFC 4593, October 2006, 22 pages.
http://www.ieff.org/rfc/rfc4593.txt

DoD 8510.1-M, "Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP) – Application Manual," Assistant Secretary of Defense for Command, Control, Communications, and Intelligence, July 31, 2000.

Defense Acquisition Guidebook, Version 1, October 17, 2004.
http://akss.dau.mil/dag/

Little, A., "Study Into the Potential Impact of Changes in Technology on the Development of Air Transport in the UK," Final Report to the Department of the Environment, Transport and Regions (DETR), November 2000.

14. GLOSSARY.

Accreditation—Accreditation is a formal declaration by a DAA that a software system or device is approved to operate in a particular safety and security mode using a prescribed set of safeguards at an acceptable level of risk.

Assurance—Assurance is the measure of confidence that a system's safety and security features have been implemented and work properly. Assurance properties must apply throughout a system's life cycle and is achieved through design, testing, and analysis.

Certification—The comprehensive evaluation of the technical and nontechnical safety and security features of a system and the other safeguards that are created in support of the accreditation process, to establish the extent that a particular design and implementation meets the set of specific safety and security requirements.

Control—A feature or function of the IT system used to mitigate the likelihood of a vulnerability being exercised and to reduce the impact of such an adverse event.

Crack—To electronically attack a computer or device by a successful exploit that compromises the machine and enables the attacker to take control over the machine, download Trojan Horses, and establish back doors so that the attacker could re-establish control over the machine at any subsequent time.

Daemon—A daemon is a background process that performs a specific function or system-related task (e.g., print). In Unix or Linux systems, daemons are programs rather than parts of the operating system's kernel. In other operating systems, they may be a constituent part of the operating system itself. Many daemons start at the operating system's boot time and continue to run as long as the operating system is up. Other daemons are started when needed and run only as long as they are useful.

Error—An omission or incorrect action by a crew member or maintenance personnel, or a mistake in requirements, design, or implementation.

Evaluation Assurance Level (EAL)—Part 3 of the CC [46] identifies seven EALs. The EALs are predefined packages of assurance components that comprise the CC's scale for rating confidence in the security of IT products and systems. EAL levels 2-7 are generally equivalent to the Trusted Computer System Evaluation Criteria (TCSEC) (see also TCSEC in this Glossary) (i.e., the "Orange Book" [42]) C2 through A1 security ratings.

Event—An occurrence that has its origin distinct from the airplane, such as atmospheric conditions, runway conditions, conditions of communication, navigation, and surveillance services, bird-strike, cabin and baggage fires. The term is not intended to cover sabotage.

Exercise a threat—(1) a malicious attempt to gain unauthorized access to an IT system to compromise system and data integrity, availability, or confidentiality or (2) a benign, but nonetheless purposeful, attempt to circumvent system security. (NIST 800-30)

Exploit—A purposeful action (or actions) by a threat source to accidentally trigger or intentionally cause, either directly or consequentially, a threat condition.

Failure—An occurrence that affects the operation of a component, part, or element such that it can no longer function as intended.

Failure condition—A condition that has an effect on either the airplane or its occupants, or both, either direct or consequential, which is caused or contributed to by one or more failures or errors, considering flight phase and relevant adverse operational or environmental conditions or external events.

Function—The lowest defined level of a specific action of a system, equipment, and flight crew performance aboard the airplane that, by itself, provides a complete recognizable operational capability.

ICMP—About a dozen types of Internet Control Message Protocol (ICMP) messages have been defined. These messages are used to report IP protocol errors to the sender as well as to provide IP-level services. The error messages most relevant to this report are: destination unreachable, time exceeded, parameter problems, source quench, and redirect. The two IP services most relevant to this report are (1) Echo request and Echo reply, which are used to see if a given destination is reachable and alive. Upon receiving the Echo message, the destination is expected to send an Echo reply message back. Another useful service is (2) the Timestamp request and the Timestamp reply, which are like Echo except that the arrival time of the message and the departure time of the reply are recorded in the reply.

Impact—Magnitude of harm that could be caused by a threat's exercise of a vulnerability

Information Assurance—The Department of Defense Directive 8500.1[*] defines information assurance as "Measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and non-repudiation. This includes providing for restoration of information systems by incorporating protection, detection, and reaction capabilities." Several synonyms to IA exist: IT security and information systems security.

Likelihood—Indication of the probability that a potential vulnerability may be exercised within the construct of the associated threat environment.

Logical network system—As used in this study, a logical network system is a partitioned network. Synonymous term is network enclave, e.g., a VPN.
National Information Assurance Partnership—A joint activity of National Institute of Standards and Technology (NIST) and NSA to establish an IT product security evaluation program based on the CC. This program is supported by a number of accredited, independent testing laboratories.

Physical network system—The physical media and intermediate system devices (e.g., router, bridge, hub) that physically create an actual functioning network. For example, LANs and/or WAN entities connected into a common network system. Physical network systems are the network system elements that physically convey network packets.

Risk—"Risk is a function of the likelihood of a given threat-source's exercising a particular vulnerability, and the resulting impact of that adverse event on the organization." (NIST 800-30 Risk Assessment)

Risk assessment—"Risk assessment is the first process in the risk management methodology. Organizations use risk assessment to determine the extent of the potential threat and the risk associated with an IT system throughout its lifecycle. The output of this process helps to identify appropriate controls for reducing or eliminating risk during the risk mitigation process." (NIST 800-30 Risk Assessment)

Severity—A measure of the effect of a failure condition on either the airplane or its occupants, including: (1) reduction in airplane safety margins or airplane functional capabilities including possible maintenance activity, (2) increase in flight crew workload or conditions impairing flight crew efficiency, and (3) distress or injury to airplane occupants.

SYN—The synchronous (SYN) bit in the TCP protocol header is used to establish TCP connections. In session establishment, it is associated with the acknowledgement (ACK) bit. A TCP connection request has SYN=1 and ACK=0 to indicate that the piggyback acknowledgement field is not in use. The connection reply does bear an acknowledgement, so it has SYN=1 and ACK=1. Therefore, the SYN bit is used to denote a connection request and a

[*] Department of Defense Directive (DoDD) 8500.1, "Information Assurance (IA)," October 24, 2002, ASD(C3I), http://west.dtic.mil/whs/directives/corres/pdf2/d85001p.pdf

connection accepted, with the ACK bit used to distinguish between those two possibilities. However, in the context of this report, the TCP SYN attack is a well-known denial of service (DoS) attack that involves sending the target multiple TCP SYN messages with no intention of following with an ACK. This forces the targets to process the SYN messages, to send out the SYN/ACK messages, and to maintain a half-open connection while waiting for an ACK that never arrives. The goal is to force the target to maintain so many of these connections that it is not capable of accepting any further connections requests.

TCP—The state machine that underlies the Transmission Control Protocol (TCP) uses a number of bits within the TCP Protocol Header. Various exploits leverage weaknesses in the TCP protocol itself and implementations of the protocol. The primary bits are:

- The Urgent pointer (URG) is used to indicate a byte offset from the current sequence number at which urgent data are to be found;

- The Acknowledgement number (ACK) is set to 1 indicates that the Acknowledgement number field in the TCP protocol header is valid;

- The PUSHed data (PSH) bit requests the receiver to deliver the data to the application upon arrival and not to buffer it until a full buffer has been received;

- The reset (RST) bit is used to reset a connection that has become confused due to a host crash or some other reason;

- The synchronous (SYN) bit is used to establish connections; and

- The final (FIN) bit is used to release a connection. It specifies that the sender has no more data to transmit. However, after closing a connection, a process may continue to receive data indefinitely. Both SYN and FIN segments have sequence numbers and are thus guaranteed to be processed in the correct order.

TCSEC—The U.S. Department of Defense developed the Trusted Computer System Evaluation Criteria, which is used to evaluate operating systems, applications and systems. It is also known as the “Orange Book” because it was originally issued with an orange cover. This criteria provides a security metric that can be used to compare different systems. It also provides direction for manufacturers so they can know what specifications to build to.

Threat—The potential for a threat-source to exercise (accidentally trigger or intentionally exploit) a specific vulnerability (NIST 800-30).

Threat source—Either (1) intent and method targeted at the intentional exploitation of a vulnerability, or (2) a situation and method that may accidentally trigger a vulnerability.

TLL field—The time to live (TTL) field indicates to routers whether the packet has been in the network too long and should be discarded. More specifically, this field in the IPv4 packet header is a counter used to limit packet lifetimes to protect against routing loops. It is decremented at each hop. When the TTL becomes zero, then the packet is discarded. This field is known as the hop limit field in IPv6 packet headers.

Traceroute—Traceroute is a popular application that was originally created by Van Jacobson to enumerate the series of IP network hops (e.g., routers) that a packet traverses to reach its destination. It is invoked as “traceroute hostname” where hotstname is the destination target. The user then will receive a report enumerating every router between the invocation location node and the destination node.

Trust—Reliance on the ability of a system to meet its specifications.

Vulnerability—A flaw or weakness in system security procedures, design, implementation, or internal controls that could be exercised (accidentally triggered or intentionally exploited) and could result in a security breach or a violation of the system’s security policy.

whois—Quote from http://en.wikipedia.org/wiki/Whois: “WHOIS (pronounced “who is”; not an acronym) is a TCP-based query/response protocol that is widely used for querying an official database to determine the owner of a domain name, an IP address, or an autonomous system number on the Internet. WHOIS lookups were traditionally made using a command line interface, but a number of simplified web-based tools now exist for looking up domain ownership details from different databases. Web-based WHOIS clients still rely on the WHOIS protocol to connect to a WHOIS server and do lookups, and command-line WHOIS clients are still quite widely used by system administrators” An example is http://www.whois.net/

APPENDIX A—HISTORIC ATTACK MECHANISMS AND TOOLS

The following sections contain technical details about historic attack mechanisms and tools that identify and exploit latent bugs within commercial off-the-shelf (COTS) computing and network systems [A-1 through A-8]. These mechanisms are not fully explained for non-information assurance (IA) security personnel, since an explanation of these details was outside of the scope of this research. Rather, these details provide partial evidence of the fact that the vast majority of modern computing equipment deployed within Internet protocol (IP) networks today cannot be adequately secured in general. Specifically, their security provisions, including their trusted paths and security controls, have repeatedly been demonstrated to not be viable when attacked. This point was initially mentioned in section 4.3 and then more fully discussed in section 4.4 of this report.

A.1 FINGERPRINTING (MAPPING AND TARGET ACQUISITION).

Fingerprinting is traditionally the first stage of an attack against IP-based systems. The goal of fingerprinting is to enable attackers to create a profile of the system and devices that they seek to eventually attack, including determining their relative security posture and defenses.

The earliest stage of fingerprinting consists of gathering whatever information one can about the target deployment and the technologies it uses. Increasingly web sites are providing an incredible wealth of information that can be used by attackers. For example, attackers frequently do web searches for network links to the target organization. E-mail messages originating from within the target environment, notably including the simple mail transfer protocol's (SMTP) e-mail headers, also contain much useful information about the target environment. Finally, news groups often reveal a surprising amount of information that is directly relevant to an attacker.

A.1.1 NETWORK ENUMERATION.

Attackers will seek to obtain information about domain names and the associated networks of the target deployment to learn about the networks within those environments. Due to interoperability and connectivity requirements, the target environment may be required to expose highly relevant domain name system (DNS) zone information to DNS servers elsewhere within the larger National Airspace System environment. Relevant information may also be available from whois servers (see section 14).

The American Registry for Internet Numbers database[1] contains information about who owns particular IP address ranges and given company or domain names. This database can also be searched to retrieve potentially useful information for domains located within the Americas. The Reseaux IP Europeens Network Coordination Centre[2] contains similar European information and the Asia Pacific Network Information Center[3] contains similar Asian information. The

[1] ARIN; see www.arin.net/index.shtml
[2] RIPE NCC; see www.ripe.net
[3] APNIC; see www.apnic.net

information found within these databases is public and serve as the "white pages phone book" for the worldwide Internet community.

The DNS is also an important component of the Internet's infrastructure. DNS is a hierarchical database distributed around the world that stores a variety of information, including IP addresses, domain names, and mail server information. DNS entries normally contain a great deal of information relevant to the attacker, including the registrant's name, the domain name, the administrative contact, when a record was created and updated, and the DNS servers within that domain. Using a process called "resolving," users and programs search the DNS hierarchy for information about given domain names. A list of dynamic host configuration protocol (DHCP) servers, e-mail servers, major file servers, major database servers, and other key system services is potentially available from DNS zone information, should the attacker eventually be able to access it[4]. These lists will normally include relevant information such as their IP addresses and DNS names of devices supporting essential infrastructure services of the target deployment. This information can be followed up in subsequent network queries, point-of-contact queries, and other mechanisms to subsequently learn increasing detail about these key devices within the target environment.

During this investigation, the attacker will also be looking for serious but far-too-common configuration mistakes, such as allowing untrusted users to perform DNS zone transfers. Should such a vulnerability exist, it can easily be exploited through nslookup and other means.

The explicit goal of the attacker at this stage is to learn as much as possible about the target organization, the domain, and network infrastructures to be attacked. This explicitly includes point of contact information that is very useful for many classes of social attacks. Most importantly, the attacker seeks to construct an accurate network map of the target environment, including an accurate classification of the operating systems (OS) and applications residing within both the routers and computers within the target environment, as well as their IP addresses and DNS names (if appropriate). Such information will be useful to maximize the efficiency and potency of the subsequent attacks.

A.1.2 NETWORK RECONNAISSANCE.

Once network enumeration has become somewhat complete, the attacker will usually attempt to determine the actual network topology as well as potential access paths into the network from the attacker's current location(s). On thoroughly mobile environments such as aircraft in flight, this information may prove to be largely transient in nature. During this stage, the attacker is likely to use tools, such as traceroute, since it directly aids in the construction of network diagrams. Other less useful tools, like ping, may also be employed.

One of the more useful historic general-purpose hacker tools for doing network enumeration and network reconnaissance is Sam Spade[5]. Many freely available web-based reconnaissance tools

[4] Organizations can use a technique called "Split-Horizon DNS" to reduce this threat. This technique maintains substantially more information about the local deployment on local private DNS servers than the externally accessible public DNS servers.

[5] see http://preview.samspade.org/ssw

exist to perform these network enumeration functions[6]. Of particular note is Cheops-ng[7], a link for open source software that maps and monitors a network.

A.1.3 SCANNING.

Once the larger network environment has been identified, potential targets within that environment are scanned. The goal here is to learn useful information such as specific user names and phone numbers; IP address ranges; and the identities of DNS servers, Web servers, document/file repositories, and e-mail servers within the domain. The attacker also seeks to "rattle doors and check windows" to identify preferential attack points. During the scanning phase, an explicit goal is to learn what systems are alive and reachable from wherever the attacker is located. Tools used include war-dialers, ping sweeps, port scans, and automated hacker discovery tools.

War dialing refers to using automated tools[8] to call all of an organization's telephone numbers to enumerate and identify the repeat dial tones and modems that are supported by that organization.

Ping sweeps systematically send Internet control message protocol (ICMP) (see request for comment (RFC) 791) Echo packets to systematically ping every IP address within a specified range of addresses to determine which active addresses can be reached from the attacking location. Tools like fping and shell scripts with gping are commonly used to conduct ping sweeps from Unix® devices and pinger, WS_PingProPack and other tools, such as Netscan, are available for Microsoft® Windows® devices. Because of this, firewalls and border routers need to be configured to block all incoming ICMP traffic so that ping sweeps originating outside of the autonomous system (AS) cannot penetrate inside that networked environment.

Should a device be reachable via ICMP Echo packets, then the attacker can learn a great deal about that device merely by sending ICMP packets to it. For example, icmpquery and icmpush are tools that the attacker can use to learn the current time on the system (ICMP type 13 message) or the device's netmask value (ICMP type 17 message). The former tells the time zone where the device is deployed, and the latter helps the attacker learn important information about how the subnetwork is configured where the target device is deployed.

Because of the growing prevalence of ICMP blocking (e.g., hopefully at firewalls and border routers), attackers have resorted to doing port scans at the transport layer instead of ping sweeps at the network layer. Tools such as nmap (using the (-PT) option[9]) and hping have automated port scanning capabilities. Because few, if any, firewalls filter hypertext transfer protocol

[6] see http://www.samspade.org/, http://www.network-tools.com/, http://www.cotse.com/iptools.html, http://www.securityspace.com/sspace/index.html, http://crypto.yashy.com/, https://grc.com/x/ne.dll?bh0bkyd2, and others.

[7] see http://cheops-ng.sourceforge.net/

[8] e.g., THC-Scan 2.0; see http://freeworld.thc.org/thc-scan/

[9] The -PT option means invoking nmap on the operating system's command line as "nmap –PT". This means that nmap will execute using its P and the T directives, which will change its execution behavior to do what those options signify.

(HTTP) traffic (port 80), port scans searching for reachable active devices are increasingly targeting Port 80 as a mechanism for defeating intermediate firewalls.

Several types of port scanning approaches exist to exploit weaknesses within the Internet's historic Transport Layer Protocols (i.e., TCP and UDP) to learn information about remote systems from transport layer responses, and these types include:

- The transmission control protocol (TCP) connect scan (i.e., connects to the target port by completing the three-way TCP handshake: SYN, SYN/ACK, and ACK).

- The TCP SYN scan (i.e., only partially connects to the port—enough so that it knows that the port is there and is active).

- TCP FIN scan (sends FIN packets to the target port, i.e., see RFC 793).

- TCP Xmas tree scan (Sends FIN, URG, and PUSH packets to the target port).

- The TCP Null scan (the technique turns off all flags. Based upon RFC 793, the target system should respond by sending back a RST for all closed ports).

- User Datagram Protocol (UDP) scan (that is looking for an ICMP port unreachable message—if no such reply, then the port is open).

Once it is determined that an active device is reachable, the attacker may want to scan the target device to discover what services it provides. The strobe, sc, netcat, portpro, portscan, nmap, and udp_scan tools are very useful for doing this. (Note: the latter was an element within "SATAN," which has subsequently been updated to become "SARA" and "SAINT.") Nmap is perhaps the most powerful of these tools since it performs many other functions and also provides decoy capabilities within its scans.

Target Unix systems may support port 113, making them vulnerable to ident scanning (see RFC 1413). In such systems, queries to port 113 will reveal all of the active ports within that system, the protocol (TCP and UDP) being used by that port, the service using that port (i.e., the identity of the application layer daemon), and the owner (e.g., root) of the daemon that is listening on that port. All of this information is very useful to an attacker.

Many other scanning attacks and exploits exist, including the file transfer protocol (FTP) bounce attack, which leverages the inherent security vulnerabilities of FTP to post virtually untraceable volumes of documents (e.g., mail and news) onto a third site, potentially filling up the disks of that third site, thereby creating a denial of service (DoS) attack. All other IP Advance Research Projects Agency (ARPA) services (as well as the Unix r- services) were similarly designed for trusted environments and are therefore similarly characterized by having ineffective security. The ARPA services include FTP (RFC 2228), trivial file transfer protocol (TFTP) (RFC 1782), Telnet (RFC 854), and SMTP (RFC 1652). For example, SMTP (port 25, i.e., Internet Electronic Mail) is so completely barefoot that one can readily spoof any aspect of the SMTP electronic mail header from one's own machine's port 25. However, the FTP is unique in that it permits one to instruct an FTP service on another machine to send files to an FTP service on a third machine, thereby cloaking the command origin—a very useful tool for attackers to hide the origin of attacks.

A.1.4 OPERATING SYSTEM DETECTION.

A second objective of port scanning is to determine the OS of that machine. Knowing the target machine's OS is invaluable in the vulnerability-mapping phase that immediately precedes launching exploits to attempt to actually take over the remote machine (see discussion about the cracking devices in section A.2). The OS identity can be learned from mechanisms such as banner grabbing;[10] however, the most useful approaches use stack fingerprinting.

Tools such as nmap, cheops, tkined, and queso are commonly used to do stack fingerprinting to quickly ascertain what the target machine's OS is, including the actual version of the OS implementation, with a high degree of probability. These tools leverage techniques such as the FIN probe (see RFC 793), the Bogus Flag probe, initial sequence number sampling, don't-fragment-bit monitoring, TCP initial window size, ACK value, ICMP error message quenching (see RFC 1812), ICMP message quoting, ICMP error message echoing integrity, type of service for "ICMP port unreachable" messages, fragmentation handling, and other TCP options (see RFC 1323) to make their calculations. Specifically, the RFCs that define TCP specify how a system should respond during connection initiation. However, they do not define how the system should respond to the various illegal combinations of TCP code bits. Rather, each implementation responds somewhat differently to the same set of illegal flags or finite state machine protocol violations. These differences provide a basis for these hacker tools to determine, with a high degree of probability, exactly what OS they are remotely accessing.

A.1.5 ENUMERATION.

Once the attacker has identified the OS of the target machine to crack, the attacker will want to learn the valid accounts or exported resource names of that system. This process is known as enumeration. The tools and approaches for accomplishing enumeration are largely a function of the target OS to be cracked. The default configuration of Microsoft Windows machines is particularly vulnerable for enumeration, though other machines are also vulnerable.

For example, within Unix devices, the Sun ONC services (e.g., Sun remote procedure call (RPC), network information system, and network file system (NFS)) are particularly vulnerable to enumeration. The finger utility is perhaps the oldest way to do enumeration on Unix systems. Similarly, r- commands such as rusers and rwho also provide enumeration services.

Enumeration can also occur via SMTP. The SMTP VRFY command confirms the names of valid users and the EXPN command reveals the actual delivery addresses of aliases and mailing lists.

[10] Services such as FTP, Telnet, SMTP, HTTP, POP3, IMAP4, and others frequently identify the operating system of their hosting machine. This identification is then leveraged by the attacker to focus the attack upon known weaknesses of that OS, often by using automated attack mechanisms.

The simple network management protocol (SNMP, see RFC 3413) also has a weakness in regard to enumeration. Many SNMP implementations readily enumerate the users of the host machine upon request. For example, the popular Unix NET-SNMP implementation of SNMP provides a management information base that is filled with a huge amount of information concerning the host OS, the IP, and mission assurance category addresses used by that machine, the network and route information of the machine's interfaces, and the active ports it is listening to. The book "Hacking_Linux Exposed Second Edition" (see page 158 of reference A-9) outlines how a single command "snmpwalk appropriate_DNS_address public" can retrieve all of this valuable information.

Another common exploit is to grab the /etc/password file from a target Unix machine. At this early stage of cracking preparation, this is usually done by using the TFTP (port UDP/69) to directly access (and copy) this file containing the list of the OS and user accounts on that platform.

Unfortunately, virtually all generic OSs have at least one well-known account that is usually present. Attackers repeatedly leverage this fact. There are many other ways to gain system permissions on these machines (see section A.2).

Several ports, including ports 111 (Sun RPC) and 32771 (FileNet RMI), also directly provide enumeration services that are exploited by hacker tools.

A.2 CRACKING DEVICES.

Because Unix devices are generally considered to be among the more difficult of the generic COTS OSs to crack (i.e., to take over via successful exploits), this section will solely discuss cracking Unix devices. Similar approaches can be used to crack other generic host OSs devices, such as Microsoft Windows or Apple Macintosh systems, as well as the special-purpose OSs used by routers.

A.2.1 ROOT ACCOUNT.

Most Unix systems have a root account that provides complete access to all functionalities and services within the OS. Many exploits consist of breaking the root password. Once the attacker has learned the root password, the attacker has effectively taken over that device. For this reason, most Unix systems have been configured to not permit remote root accesses, but rather require the administrator first log into the system via a user account and then subsequently use the Unix su command to become root. (This latter practice also enables the log files to identify the identity of the root user, which otherwise would not be known.) For this reason, many exploits first seek to break a user account and then break the root account.

A.2.2 USER ACCOUNTS.

Both root and user accounts can be broken through brute force mechanisms, data driven attacks, back channels, and social engineering attacks. Brute force mechanisms exist because weak and default passwords are historically the easiest mechanism to compromise Unix systems. Brute force mechanisms can be defended against by effective password management procedures and by limiting the number of failing account accesses that can occur within a given time period. The other types of attacks will be discussed below.

A.2.3 OPEN-NETWORK PORTS.

Ports[11] provide the avenue for device processes or applications to receive or send data across an IP network. IP communications are addressed in terms of a specific IP address that identifies a specific device within the network, and a specific port, identifying a specific application or process within that device (see RFC 2780). Open ports within a device provide opportunities for remote attackers to remotely attack the process or application using that port. All unneeded ports should to be closed. Indeed, devices should only support the minimum number of ports required to perform the device's mission(s). Bob Toxen observed:

> "Just as every account on a system is a potential path for a cracker, every network service [port] is a road to it. Most Linux [i.e., a type of Unix] distributions install 'tons' of software and services by default. They deliberately prefer 'easy' over 'secure.' Many of these are not necessary or wanted. Take the time to remove software and services you do not need. Better still—do not install them to begin with." [A-10]

For example, Department of Defense (DoD) instruction 8551.1 [A-11] requires that

> "Ports, protocols, and services that are visible to DoD-managed network components shall undergo a vulnerability assessment; be assigned to an assurance category; be appropriately registered; be regulated based on their potential to cause damage to DoD operations and interests if used maliciously; *and be limited to only the PPS required to conduct official business* or required to address Quality of Life issues authorized by competent authority." (Emphasis added, quoted from Section 4.1 of A-11.)

A.2.4 OLD SOFTWARE VERSIONS.

Vulnerabilities are continually being found and corrected in software systems. Thus, effective security requires that the administrators keep up with the current patches and software versions.

A.2.5 SESSION HIJACKING.

Session hijacking is the process used by an attacker to find an active TCP connection between two other computers and to take control of it, making it unusable by the actual source. Hacker tools, such as juggernaut and hunt, seek to leverage this vulnerability.

A.2.6 WEB HACKING.

Websites are subject to a host of security vulnerabilities that offer attackers numerous possible

[11] See http://www.iana.org/assignments/port-numbers

opportunities to crack the hosting server(s) that supports the web site. So many different vulnerabilities and exploits are associated with HTTP and web services that it is impossible to list all of them. Script inadequacies are among the greatest historical vulnerabilities within websites. This includes problems both within the script itself as well as problems with Common Gateway Interface (CGI) that interfaces to the scripts or other executables, and with server side includes (SSI)[12]. Vulnerabilities can also be introduced by inserting malicious code (through various means) into the user's web browser or by inserting corrupt web proxies. Many other security vulnerabilities are introduced by poor web design and programming mistakes on the part of the web developer, including bugs latent within the executables accessed by the website.

Fortunately, a number of tools have been created to identify specific well-known vulnerabilities within websites. Older tools include phfscan, cgiscan, grinder, and SiteScan. Unfortunately, attackers also are able to use these same tools to identify and leverage existing vulnerabilities within existing web sites, and new vulnerabilities may be inadvertently introduced during any subsequent web site modification.

A.2.7 MOBILE CODE AND MALICIOUS CODE.

Because the distinction between data and code is vanishing, malicious code (e.g., viruses and worms) may be introduced without a conscious decision on the part of a user. Malicious code can perform many functions, including providing a vehicle for an attacker to compromise a system. For example, malicious code may be introduced when installing executable code, by a Java applet, or by viewing apparently benign data within received e-mail or at remote websites. Mobile code, by contrast, is defined to merely be code that travels a network during its lifetime to execute on a destination machine. All current mechanisms to secure mobile code involves trade-offs [A-12]. Consequently, the current situation remains very much like Gary McGraw and Edward Felten observed back in 1998:

> "Today's diverse approaches to securing mobile code are all works in progress. Each different implementation of mobile code, including Java, ActiveX, and JavaScript, faces similar security risks; but each system presents a different way of dealing with the risks. In our opinion, Java's security design stands heads and shoulders above the competition. But Java is a complex system ... Securing Java and other forms of mobile code is still as much an art as it is a science" [A-13].

A.2.8 NETWORK TIME PROTOCOL ATTACKS.

Unix devices are potentially susceptible to network time protocol (NTP) spoofing attacks. Even though the NTP protocol is optionally equipped with authentication and integrity capabilities, it runs over the UDP protocol. More tellingly, publicly trusted NTP servers rarely use the NTP authentication provisions. Because of this, it is often possible for an attacker to forge NTP packets to a machine to make them appear as if they are coming from a trusted NTP server. The attacker's goal in doing this is to manipulate that receiving computer's systems clock, impacting key utilities on that computer such as cron and ntpdate. A common reason for doing this is that

[12] If SSI is used at all, its use should be limited by the "Includes NOEXEC" option.

if an attacker resets a Unix device’s time to a week from now, then it will trigger logrotate to rotate the Unix logging files. If the attacker does this five times, then the current Unix syslog files will be deleted, thereby eliminating the attacker’s tracks from the attacked device’s logging system.

A.2.9 DATA DRIVEN ATTACKS.

These are perhaps the best-known mechanisms for cracking remote systems. Data driven attacks are executed by sending data to an active service that causes unintended or undesirable results. These types of attack include:

- Buffer Overflow attacks. A buffer overflow occurs when a user or process attempts to place more data into a buffer (e.g., fixed array) than was originally allocated. A buffer overflow condition normally causes a segmentation violation to occur. This event can be potentially exploited to gain access to the target system. For example, if the process where the buffer overflow occurred is running as root (e.g., is a communications protocol), and if (at the appropriate place within the overflowing data) the data contained code that executed the command /bin/sh, then /bin/sh would be executed with root permissions, thereby giving the attacker a shell (e.g., command lines) to use that has root permissions. In this manner, attackers can gain control of OSs. Once they have gained control, they can establish backdoors and Trojan horses for subsequent access. Safeguards against buffer overflow attacks include improved software development practices. For example, validating arguments within code; using more secure routines such as (for the C programming language) fget(), strncpy() and strncat(); better test and audit practices; and using safe compilers such as Immunix’s StackGuard or Janus. Alternatively, rather than recompiling every program on the system, the Libsafe dynamic library file can be installed with either the environment variable $LD_PRELOAD specified or else list it in /etc/ld.so.preload. Unfortunately, these types of vulnerabilities only reduce the number of buffer overflows without eliminating all of them. Thus, this threat continues to exist even within systems whose developers have undertaken these types of safeguards.

- Input Validation attacks. An input validation attack leverages a programming flaw where (1) the program fails to properly parse and validate received input; (2) a module accepts this syntactically incorrect input; (3) the module fails to handle the missing input fields; and (4) a field value correlation error subsequently occurs. If a program accepts user-supplied input and did not properly validate it, it could be tricked into executing arbitrary code via leveraging Unix shell escape commands. Executing nonvalidated escape sequences provide a comparable capability to the attacker to crack the device as buffer overflows. The primary safeguard against this type of problem is improved software development practices.

These classes of attacks point out the importance of shell access within Unix systems. Within these OSs, shells provide command line capabilities to remote or local users. A successful logon, regardless of whether it is local or remote, results in the user receiving a shell. Once the user has a shell, then he or she is able to perform any function on that computer that he or she is authorized to perform.[13] However, attackers can also obtain shell access in an unintended (and nonauthorized) manner via the above-mentioned data driven attacks, as well as by Back channel attacks.

A.2.10 BACK CHANNEL.

This is a mechanism where the communication channel originates from the targeted system rather than from the attacking system. The attack consists of the attacker configuring his own system to automatically accept the target back channel communication for a particular protocol (e.g., via using the netcat or nc utility on his own machine) and then manipulating the target computer to contact the attacker's computer via that protocol. The attacker manipulates the target computer via data driven attacks (e.g., buffer overflow, input validation) or by other means. Possible back channel attacks include

- Reverse Telnet. This type of attack applies not only for telnet but also for the other Unix remote access mechanisms. For example, the following would potentially cause a reverse telnet to be executed from the cracked machine to the attacker's machine: /bin/telnet attackers_IP_address 80 | /bin/sh | /bin/telnet attackers_IP_address 25. Such a command would enable a remote hacker to execute instructions on the cracked machine via accessing the cracked machine through its normal web access port (e.g., this specific example could have been launched as the historic PHF[14] attack against the web-server's CGI script).

- Nc or netcat. If the attacker previously had inserted netcat on the cracked machine and assigned it to listen to a specific port, then netcat provides a ready back door for all subsequent accesses to that machine without needing to subsequently leverage a data driven attack for such access.

A.2.11 LOCAL ACCESS.

Most attackers seek to obtain local access within the OS via a remote access vulnerability of the OS. Once the attacker has an interactive command shell, they are considered to be local to the system. As previously mentioned, once attackers against Unix systems had local access, they traditionally sought to obtain privilege escalation by becoming root.

A.2.12 OTHER WELL-KNOWN ATTACKS.

Other well-known attacks include:

- Remote attacks using ARPA services: Telnet, TFTP, FTP, and Sendmail (SMTP). These very popular protocols are all used for a wide number of attacks leveraging the fact that

[13] Users with root permission are authorized to perform all available functions on that computer.

[14] PHF—a program name referring to a type of CGI script file. The PHF program with Apache 1.0.5 and earlier versions was distributed in the cgi-src directory, and needed active effort to both compile it and place it in the cgi-bin directory.

they predate today’s security awareness and therefore their security provisions are demonstrably inadequate and vulnerable.

- Attacks using the r- services: rsh, rcp, rexec, and rlogin. These services were historically very popular on Unix systems. Like the ARPA services, these services are designed for trusted environments only. Unlike ARPA services, they often lack security provisions altogether.

- Remote attacks by RPC services. These types of attacks either leverage buffer overflow problems within the RPC implementations itself or else leverage security weaknesses associated with the Sun Microsystems ONC protocol family services (e.g., Sun RPC, NFS, Network information systems).

- NFS, mountd, and portmap attacks.

- Leveraging X-windows insecurities.

A.2.13 SOCIAL ENGINEERING.

These attacks involve an attacker tricking a network manager to inappropriately reveal sensitive information, such as account-password information. As Ed Skoudis observed:

> “The most frustrating aspect of social engineering attacks for security professionals is that such attacks are nearly always successful. By pretending to be another employee, a customer, or supplier, the attacker attempts to manipulate the target person to divulge some of the organization’s secrets. Social engineering is deception, pure and simple” [A-2].

One of the first steps an attacker often takes after compromising a device is to eliminate any record of his actions from the cracked device’s audit logs and take steps to ensure that none of his future clandestine actions will be similarly recorded on the audit logs. In standard computer systems, the former is usually fairly straightforward and consists of modifying, corrupting, or deleting the audit files themselves.

The attacker usually inserts Trojan horses in key system utilities as a mechanism to hide his or her activities from the audit logs. He also will establish backdoors and logic bombs for continued control of the device after he “logs off.”

A Trojan, which is short for Trojan horse, is a program that purports to perform an authorized task, but actually carries on other activities behind the scenes. Many attackers replace common OS commands on the cracked OS with Trojans of their own design. For example, ps, netstat, ifconfig, killall, ls, ssh, who, find, du, df, sync, reboot, halt, and shutdown are commonly replaced by attackers of Unix OSs by a Trojan version contained within the attacker’s rootkit. A rootkit is a collection of tools an attacker downloads to a victim computer after gaining initial access. In addition to system binaries, rootkits for Unix systems usually also contain network sniffers, log-cleaning scripts, and back door remote-access daemon replacements such as a modified telnetd or sshd.

> “The fundamental problem in detecting rootkits is that you can’t trust your operating system. You can’t believe what the system tells you when you request a list of running processes or files in a directory. One way to get around this is to shut down the suspect computer and check its storage after booting from alternative media that you know are clean, such as a rescue CD-ROM or a dedicated USB flash drive. A rootkit that isn’t running can’t hide its presence, and most antivirus programs will find rootkits by comparing standard operating system calls (which are likely to be altered by the rootkit) against lower-level queries, which ought to remain reliable. If the system finds a difference, you have a rootkit infection” [A-14].

A logic bomb (also known as a time bomb) is a program that lies dormant until a specified event happens or until a condition is true, when the malicious code is activated. They are especially effective when coupled with a virus.

Worms and viruses are transport mechanisms for malicious code. A virus is a program that when run, inspects its environment and copies itself into other programs if they are not already infected, often without their users (or system administrators) knowing about the infection. A worm is a program that copies itself over computer networks, infecting programs and machines in remote locations. It primarily differs from a virus in that at does not require a human agency to activate it (e.g., a human (or a process) executes the affected program to propagate a virus, but a worm self-propagates itself).

A backdoor is a mechanism for an attacker to return to the device to continue to control (or attack) it once he has compromised it. One of the easiest backdoors for the attacker to add for ready future access into a cracked Unix system leverages adding the netcat (or nc) utility to the cracked system. Netcat is a common tool used for controlling TCP/IP systems if it is compiled with the #define GAPING_SECURITY_HOLE option that is associated with its –e invocation option. Netcat can be configured to listen on a certain port and launch an executable when a remote system connects to that port. By configuring a netcat listener to launch a shell for the remote attacker to use, normal security surrounding secure shell (SSH)-only remote access can be bypassed, permitting the attacker to have direct shell access without undergoing SSH’s authentication mechanisms. Of course, other backdoor possibilities also exist, but this one is particularly well known. Other common backdoors include attacker-modified startup files (particularly rc.d), BOOTP startup files that are provided to computers via DHCP servers, regularly scheduled jobs (e.g., crontabs), and others. In fact, so many backdoors are possible that the most viable mechanism today for recovering from being cracked is to restore and reinstall the OS from the original media.

In addition, cracked systems are vulnerable to port redirection that permits an attacker, located outside of a firewall, to access and control computers within the firewall. Redirection works by having a cracked system listen on certain ports and forward the raw packets to a specified secondary target. In this manner, an attacker can know what is occurring behind the firewall—

unless, of course, the firewall has established a reverse proxy that is equipped to handle this type of threat. Attackers can similarly control what is happening on devices within the firewall by communicating with the cracked device via HTTP (port 80), a protocol that is rarely blocked by any firewall.

Routers have similar vulnerabilities to end-systems except that they are more likely than end-systems to be identified by traceroute and they usually have substantially fewer resident application daemons for the attacker to potentially exploit.

Attackers often attack routers through SNMP. There are many security problems with SNMP (see section 4.6). These systems are particularly vulnerable if older versions of SNMP (i.e., SNMPv1 or SNMPv2) are being used or if the default SNMP community names have not been altered or removed from the network device previous to deployment (e.g., "public," "write," "user" are common default SNMP account names on routers, usually without any associated password protections). Similar vulnerabilities exist for the default accounts and maintenance accounts that come on most networking devices. In all other respects, the threats and exploits affecting network devices such as routers are the same as those affecting computers, except that the network devices traditionally have substantially fewer applications, and therefore less vulnerability for attackers to exploit.

A.3   AVAILABILITY ATTACKS.

These attacks do not seek to take over devices or network systems, but rather seek to make the network systems supporting devices become ineffectual.

A number of controls have been proposed to thwart specific classes of availability attacks. Some of these controls have been demonstrated in laboratory environments. However, other than securing the data communications protocols themselves (see section 4.5), few if any of these mechanisms have yet been demonstrated to be effective within actual operational network deployments. Thus, effective defenses against many classes of availability attacks are not yet available within today's best current practices.

A.3.1 DENIAL OF SERVICE ATTACKS.

There are a myriad of possible ways that DoS attacks may occur within networks. So many, in fact, that a complete enumeration of the possible mechanisms is probably not possible. However, RFCs 3704 and 3882 provide guidance to protect against certain classes of DoS attacks. The following are some of the more commonly known DoS exploits.

- Insertion of bogus routing data into the routing table causing routing loops, needlessly delaying, or needlessly dropping packets

- Computers sending vast amount of traffic to a device's port address

- Nmap-based scan attacks against devices using some other computer's source IP Addresses

- Hosts sending vast amounts of traffic to other hosts within networks

A.3.2 DISRUPTING ROUTING.

The Internet Engineering Task Force (IETF) has recently begun to enumerate the specific threats associated with standard IETF protocols. These threats can directly or indirectly disrupt routing systems[15]. They have produced three documents that address threats to routing protocols [A-15, A-16, and A-17].

Reference A-18 discusses the generic threats to routing protocols. Routing protocols are vulnerable to potential attacks against any one of the three functions that they share in common:

- "Transport Subsystem: The routing protocol transmits messages to its neighbors using some underlying protocol. For example, OSPF uses IP, while other protocols may run over TCP.

- "Neighbor State Maintenance: neighboring relationship formation is the first step for topology determination. For this reason, routing protocols may need to maintain the state of their neighbors. Each routing protocol may use a different mechanism for determining its neighbors in the routing topology. Some protocols have distinct exchange sequences used to establish neighboring relationships, e.g., Hello exchanges in OSPF.

- "Database Maintenance: Routing protocols exchange network topology and reachability information. The routers collect this information in routing databases with varying detail. The maintenance of these databases is a significant portion of the function of a routing protocol." (Quoted from Section 2 of reference A-15.)

[15] see http://www.ietf.org/html.charters/rpsec-charter.html

There are a variety of threats associated with attacking each of these subsystems. For example,

> "An attacker who is able to break a database exchange between two routers can also affect routing behavior. In the routing protocol data plane, an attacker who is able to introduce bogus data can have a strong effect on the behavior of routing in the neighborhood.
>
> At the routing function level threats can affect the transport subsystem, where the routing protocol can be subject to attacks on its underlying protocol. At the neighbor state maintenance level, there are threats that can lead to attacks that can disrupt the neighboring relationship with widespread consequences. For example, in BGP, if a router receives a CEASE message, it can lead to breaking of its neighboring relationship to other routers.
>
> There are threats against the database maintenance functionality. For example, the information in the database must be authentic and authorized. Threats that jeopardize this information can affect the routing functionality in the overall network. For example, if an OSPF router sends [Link State Advertisements] LSAs with the wrong Advertising Router, the receivers will compute a [Shortest Path First] SPF tree that is incorrect and might not forward the traffic. If a BGP router advertises a [Network Layer Reachability Information] NLRI that it is not authorized to advertise, then receivers might forward that NLRI's traffic toward that router and the traffic would not be deliverable. A [Protocol Independent Multicast] PIM router might transmit a JOIN message to receive multicast data it would otherwise not receive." (Quoted from Section 3 of reference A-15.)
>
> "In general, threats can be classified into the following categories based on their sources:
>
> - Threats that result from subverted links: A link become subverted when an attacker gain access (or control) to it through a physical medium. The attacker can then take control over the link. This threat can result from the lack (or the use of weak) access control mechanisms as applied to physical mediums or channels. The attacker may eavesdrop, replay, delay, or drop routing messages, or break routing sessions between authorized routers, without participating in the routing exchange.
>
> - Threats that result from subverted devices (e.g. routers): A subverted device (router) is an authorized router that may have been broken into by an attacker. The attacker can use the subverted device to inappropriately claim authority for some network resources, or violate routing protocols, such as advertising invalid routing information." (Quoted from Section 3.1.1 of reference A-15.)
>
> "There are four types of threat consequences: disclosure, deception, disruption, and usurpation.

- Disclosure: Disclosure of routing information happens when a router successfully accesses the information without being authorized. Subverted links can cause disclosure, if routing exchanges lack confidentiality. Subverted devices (routers), can cause disclosure, as long as they are successfully involved in the routing exchanges. Although inappropriate disclosure of routing information can pose a security threat or be part of a later, larger, or higher layer attack, confidentiality is not generally a design goal of routing protocols.

- Deception: This consequence happens when a legitimate router receives a false routing message and believes it to be true. Subverted links and/or subverted device (routers) can cause this consequence if the receiving router lacks ability to check routing message integrity, routing message origin, authentication or peer router authentication.

- Disruption: This consequence occurs when a legitimate router's operation is being interrupted or prevented. Subvert links can cause this by replaying, delaying, or dropping routing messages, or breaking routing sessions between legitimate routers. Subverted devices (router) can cause this consequence by sending false routing messages, interfering normal routing exchanges, or flooding unnecessary messages. (DoS is a common threat action causing disruption.)

- Usurpation: This consequence happens when an attacker gains control over a legitimate router's services/functions. Subverted links can cause this by delaying or dropping routing exchanges, or replaying out-dated routing information. Subverted routers can cause this consequence by sending false routing information, interfering routing exchanges, or system integrity.” (Quoted from Section 3.1.2 of reference A-15.)

“Within the context of the threat consequences described above, damage that might result from attacks against the network as a whole may include:

- Network congestion: more data traffic is forwarded through some portion of the network than would otherwise need to carry the traffic,

- Blackhole: large amounts of traffic are directed to be forwarded through one router that cannot handle the increased level of traffic and drops many/most/all packets,

- Looping: data traffic is forwarded along a route that loops, so that the data is never delivered (resulting in network congestion),

- Partition: some portion of the network believes that it is partitioned from the rest of the network when it is not,

- Churn: the forwarding in the network changes (unnecessarily) at a rapid pace, resulting in large variations in the data delivery patterns (and adversely affecting congestion control techniques),

- Instability: the protocol becomes unstable so that convergence on a global forwarding state is not achieved, and

- Overload: the protocol messages themselves become a significant portion of the traffic the network carries.

The damage that might result from attacks against a particular host or network address may include:

- Starvation: data traffic destined for the network or host is forwarded to a part of the network that cannot deliver it,

- Eavesdrop: data traffic is forwarded through some router or network that would otherwise not see the traffic, affording an opportunity to see the data or at least the data delivery pattern,

- Cut: some portion of the network believes that it has no route to the host or network when it is in fact connected,

- Delay: data traffic destined for the network or host is forwarded along a route that is in some way inferior to the route it would otherwise take,

- Looping: data traffic for the network or host is forwarded along a route that loops, so that the data is never delivered." (Quoted from Section 3.1.2.1 of reference A-15.)

A.3.3 DISRUPTING NETWORK MANAGEMENT.

The threats described in the previous subsection directly affect network functions other than routing. For example, the network management subsystem within that AS may be rendered ineffective (and inoperable) simply because the mechanisms for identifying or resolving the resulting network problems that were created by a compromised router have been subverted. That is, network management is dependent upon the viability of the underlying routing system.

For example, if the audit records of a compromised router have been modified to erase the attacker's presence, the network manager will have reduced basis for network fault management since he (or she) would be unable to identify which system was the corrupted one. This is particularly the case within mobile wireless networks, where network performance may be affected by the signal intermittence properties of the underlying wireless media. Should audit logs be successfully modified to cloak the fact that a router had been compromised within a mobile environment, then the network managers may have a difficult time determining whether the routing table fluctuations, for example, were a function of normal mobile network availability problems due to signal intermittence or whether they had a more sinister origin.

A.4 INTEGRITY AND CONFIDENTIALITY ATTACKS.

Perhaps the most common security threat that historically resulted from compromised routers has been compromising the confidentiality of the data contained within the packets that the router forwards. The prevalence of this class of attack is kept well hidden from the public due to possible detrimental business impacts should the general public learn of this threat type. However, beginning in 1994, major U.S. Internet service providers began to privately disclose during IETF meetings certain successful (and extremely clever) exploits of this nature. Because these types of attacks are not discussed publicly, it is impossible to know just how pervasive and widespread this problem remains. For it to occur, a successful exploit enables an attacker to insert a backdoor (for future access to the collected data that may be stored locally in the router or forwarded elsewhere) into the router's OS, coupled with an attacker-built, packet-reading utility that is inserted in the router's forwarding engine to glean and store (or forward) relevant information obtained from the router-forwarded packets.

It is also conceivable that if the attacker can insert a clandestine packet-listening program then he or she could also theoretically insert software to change select packet data, thereby affecting the integrity of the transmitted data itself. IETF protocols (see section 4.5) come equipped with integrity provisions to detect and reject malformed results from this latter type of attack. Thus, integrity attacks are more likely for the subset of communication protocols that have not been configured to provide integrity protections at the protocol level. Unless such packet corruption is sparingly done, it is possible that network managers may observe a higher percentage of message integrity failures, and thus become alerted to this particular activity. In any case, the best defense for recognizing this type of attack is deploying a network instrusion detection system (NIDS) capability on the network and ensuring that the NIDS has a highly intelligent expert system to correctly identify (with the lowest possible percentage of false positives) these classes of attacks.

However, the first line of defense for protecting network communications from these types of attacks is to universally use Internet Protocol security. Specifically, the use of the encapsulating security payload (see RFC 4303) is particularly well-suited for effectively protecting transmissions against possible integrity and confidentiality attacks.

A.5 REFERENCES.

A-1. Loscocco, P., Smalley, S., Muckelbauer, P., Taylor, R., Turner, S., and Farrell, J., "The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments," *Proceedings of the 31st National Information Systems Security Conference*, October, 1998, pp. 303-314. http://www.nsa.gov/selinux/papers/inevit-abs.cfm

A-2. Skoudis, E., *Counter Hack*, Prentice Hall, 2002.

A-3. Klevinsky ,T.J., Laliberte, S., and Gupta, A., *Hack I.T.*, Addison-Wesley, 2002.

A-4. Hatch, B. and Lee, J., *Hacking Linux Exposed*, Second Edition, McGraw-Hill/Osborne, 2003.

A-5. Mourani, G., *Securing and Optimizing Linux, The Hacking Solution*, Third Edition, Open Network Architecture, Inc., 2001.

A-6. McClure, S., Scambray, J., and Kurtz, G., *Hacking Exposed: Network Security Secrets and Solutions*, Osborne/McGraw-Hill, 1999.

A-7. Rubin, A., *White-Hat Security Arsenal: Tackling the Threats*, Addison-Wesley, 2001.

A-8. Barrett, D., Silverman, R., and Byrnes, R., *Linux Security Cookbook*, O'Reilly and Associates, 2003.

A-9. Hatch, B. and Lee, J., *Hacking Linux Exposed, Second Edition*, Osborne Press, 2003.

A-10. Toxen, B., *Real World Linux Security, Second Edition*, Prentice Hall, 2003, pp. 29.

A-11. DoD 8551.1, Department of Defense Instruction Number 8551.1, "Ports, Protocols, and Services Management (PPSM)," August 13, 2004.

A-12. Devanbu, P., Fong, P., and Stubblebine, S., "Techniques for Trusted Software Engineering," *Proceedings of the 20th International Conference on Software Engineering*, Kyoto, Japan, 1998.

A-13. McGraw, G. and Felten, E., "Mobile Code and Security," *IEEE Internet* Computing, Volume 2, Number 6, November/December 1998, pp. 26-29.

A-14. Kay, R., "QuickStudy: Rootkits," *ComputerWorld Magazine* (electronic version), January 30, 2006. http://www.computerworld.com/securitytopics/security/hacking/story/0,10801,108119,00.html?source=NLT_SEC&nid=108119

A-15. Barbir, A., Murphy, S., and Yang, Y., "Generic Threats to Routing Protocols," RFC 4593, October 2006, 22 pages.
http://www.ieff.org/rfc/rfc4593.txt

A-16. Jones, E. and Le Moigne, O., "OSPF Security Vulnerabilities Analysis," June 16, 2006. (a former [now expired] Internet-Draft document)

A-17. Murphy, S. "BGP Security Vulnerabilities Analysis," RFC 4272, January 2006. http://www.ietf.org/rfc/rfc4272.txt

A-18. Department of Defense Directive (DoDD) 8500.1, "Information Assurance (IA)," October 24, 2002, ASD(C3I).

APPENDIX B—THE FAA LAN SURVEY RESULTS

During June 2006, the Federal Aviation Administration (FAA) distributed the FAA local area network (LAN) survey to several hundred people who are associated with constructing airborne software in some manner. Twenty-two people responded. The following table tabulates the responses. The table is organized in terms of the function the responder's identified their employer as performing. None of the responders said that their employer was a commercial airline company, so that column is not included within the following tabulation results.

Survey Question	Survey Responses				
What is the primary role of your employer in regards to commercial aviation?	Build Components	Build Aircraft	U.S. Federal Agency	Consultant	Other
Number of respondents in each category:	7	3	5	5	2
1. Have you or are you currently planning to deploy, design, or build software or devices that reside on LANs onboard aircraft?	4 - Yes 2 - No 1 - NA	1- Yes 2 - No	3 - No 2 - NA	2 - Yes 3 - No	1 - Yes 1 - No
2. How many different LAN-attached devices have you already deployed, designed, or built for deployment on aircraft?	2 – 10+ 2 – 1 or 2 3 - NA	1 – 3 to 5 2 - NA	1 – zero 4 - NA	1 – 6 to 9 1 – 1 or 2 3 - NA	1 – 6 to 9 1 - NA
3. How many different LAN-attached devices do you currently have specific plans to build and/or deploy in the future for deployment on aircraft?	2 – 10+ 2 – 1 or 2 3 - NA	1 – 3 to 5 2 - NA	1 – 1 or 2 4 – NA	1 – 3 to 5 1 – 1 or 2 3 – NA	1 – 3 to 5 1 - NA
4. How many of these (built or planned) LAN-attached devices have (or will have) their intelligence primarily be silicon or firmware based?	2 – 10+ 1 – 3 to 5 1 – zero 3 – NA	1 – zero 2 - NA	1 – 1 or 2 4 – NA	1 – 10+ 1 – 1 or 2 3 – NA	1 – 3 to 5 1 - NA
5. How many of these (built or planned) LAN-attached devices contain (or will contain) software?	2 – 10+ 1 – 1 or 2 1 – zero 3 - NA	1 – 3 to 5 2 - NA	1 – 3 to 5 4 - NA	1 – 6 to 9 1 – 1 or 2 3 – NA	1 – 3 to 5 1 - NA
6. Does (or will) any of your software components communicate across an airborne LAN using the Internet Protocol?	1 – Yes 2 – No 4 - NA	1 –Yes 2 – NA	1 – Yes 4 – NA	2 – Yes 3 – NA	1 – Yes 1 - NA
7. Is any of your LAN-attached software: DO-178B Level A software?	2 – Yes 5 - NA	1 –Yes 2 - NA	1 – No 4 – NA	1 – No 4 – NA	1 – No 1 - NA
8. Is any of your LAN-attached software: DO-178B Level B software?	1 – Yes 6 – NA	1 –Yes 2 - NA	1 – No 4 – NA	1 – No 4 – NA	1 – No 1 - NA
9. Is any of your LAN-attached software: DO-178B Level C software?	1 – Yes 6 – NA	1 –Yes 2 - NA	1 – No 4 – NA	1 – No 4 – NA	1 – No 1 - NA
10. Is any of your LAN-attached software: DO-178B Level D software?	1 – Yes 6 – NA	1 –Yes 2 - NA	1 – Yes 4 – NA	1 – Yes 1 – No 3 – NA	1 – No 1 - NA

Survey Question	Survey Responses				
What is the primary role of your employer in regards to commercial aviation?	Build Components	Build Aircraft	U.S. Federal Agency	Consultant	Other
Number of respondents in each category:	7	3	5	5	2
11. What is the function performed by this software (e.g., a specific avionics function?)	Power Distribution; Protocol handling	Executive	Cabin network with no aircraft interface	Interconnect Laptops (WiFi)	NA
12. While the aircraft is airborne, can this software potentially communicate with ground-based National Airspace System entities?	1 – No 6 – NA	1 –No 2 - NA	1 – No 4 – NA	1 – No 4 – NA	2 - NA
13. While the aircraft is airborne, can this software potentially communicate with (entities within) other airborne aircraft?	1 – No 6 – NA	1 –No 2 - NA	1 – No 4 – NA	1 – No 4 – NA	2 - NA
14. Does your software development process that you used (or will use) to develop this software extend current DO-178B/ED-12B processes to explicitly address network security risks?	1 – No 6 – NA	1 –No 2 - NA	1 – No 4 – NA	1 – No 4 – NA	2 - NA
15. Is your software derived from autocode?	1 – No 6 – NA	1 –No 2 - NA	1 – No 4 – NA	1 – No 4 – NA	2 - NA
16. What programming language(s) is the software written in?	1 – C, C++, assembly 6 - NA	1 – C, assembly 2 - NA	1 – I don't know 4 – NA	1 – I don't know 4 – NA	2 - NA
17. What Operating System does your software use?	2 – No OS 5 – NA	1-High assurance OS 2 - NA	1 – COTS OS 4 – NA	1 – No OS 4 – NA	2 - NA
18. Approximately how many lines of code comprise (or is currently anticipated for) your software?	1 – between 4000 and 10000 lines 6 - NA	1 – More than 10000 lines 2 - NA	5 – NA	5 – NA	2 - NA
19. Once your software has been compiled, is the executable code signed in a manner that complies with the U.S. Federal Digital Signature Standard (FIPS Publication 186) previous to distribution and installation onboard aircraft?	7 – NA	1 -Yes 2 - NA	1 – No 4 – NA	5 – NA	2 - NA
20. What personnel (e.g., what roles) are authorized to sign the executable?	7 – NA	1 – don't know 2 – NA	5 – NA	5 – NA	2 – NA

Survey Question	Survey Responses				
What is the primary role of your employer in regards to commercial aviation?	Build Components	Build Aircraft	U.S. Federal Agency	Consultant	Other
Number of respondents in each category:	7	3	5	5	2
21. Are aircraft-resident mechanisms in place so that only software executables, which have passed the integrity and authentication provisions of the U.S. Federal Digital Signature Standard, are permitted to be installed within the aircraft?	7 – NA	1 –Yes 2 - NA	5 – NA	5 – NA	2 - NA
22. Have you defined a formal process to ensure the integrity and identity of your software after it has been installed onboard aircraft?	1 – Yes 6 – NA	1 –Yes 2 –NA	1 – No 4 – NA	1 – Yes 4 – NA	2 - NA
23. When can your software be installed onboard aircraft?	1 – locally, ground only 5 – NA	1 – local or remote, ground only 2 – NA	1 – locally, ground only 4 – NA	1 – locally, ground only 4 – NA	2 – NA
24A. In your personal opinion, could the following security failures during flight potentially result in major (or greater) safety failures to the aircraft: The contents of the data packets that traverse the airborne LAN become maliciously altered?	4 – Yes 3 – No	3 – Yes	4 – Yes 1 – NA	1 – Yes 4 – No	2 - No
24B. In your personal opinion, could the following security failures during flight potentially result in major (or greater) safety failures to that aircraft: The software that provides security control protections onboard the aircraft LAN becomes maliciously compromised?	3 – Yes 4 – No	2 –Yes 1 – No	4 – Yes 1 - NA	1 – Yes 4 – No	2 - No
24C. In your personal opinion, could the following security failures during flight potentially result in major (or greater) safety failures to that aircraft: Integrity of avionics software onboard the aircraft becomes maliciously compromised?	5 – Yes 2 – No	2 –Yes 1 – No	2 – Yes 2 – No 1 - NA	2 – Yes 3 – No	2 - Yes
24D. In your personal opinion, could the following security failures during flight potentially result in major (or greater) safety failures to that aircraft: The aircraft's LAN becomes congested so that packets sent across that LAN cannot arrive in a timely manner?	6 – Yes 1 – No	3 –Yes	1 – Yes 3 – No 1 - NA	1 – Yes 4 – No	2 - No

Survey Question	Survey Responses				
What is the primary role of your employer in regards to commercial aviation?	Build Components	Build Aircraft	U.S. Federal Agency	Consultant	Other
Number of respondents in each category:	7	3	5	5	2
24E. In your personal opinion, could the following security failures during flight potentially result in major (or greater) safety failures to that aircraft: The contents of the data traversing the aircraft's LAN become known to a remote electronic attacker?	5 – Yes 2 – No	1 –Yes 2 – No	2 – Yes 1 – No 2 - NA	1 – Yes 4 – No	2- No
24F. In your personal opinion, could the following security failures during flight potentially result in major (or greater) safety failures to that aircraft: The identities of passengers and crew onboard the aircraft become known to unauthorized hostile entities?	3 – Yes 4 – No	1 –Yes 2 – No	2 – Yes 2 – No 1 - NA	1 – Yes 4 – No	2 - Yes
24G. In your personal opinion, could the following security failures during flight potentially result in major (or greater) safety failures to that aircraft: The current location of the aircraft is accurately tracked in real time by unauthorized hostile entities?	4 – Yes 3 – No	2 –Yes 1 – No	2 – Yes 2 – No 1 - NA	1 – Yes 4 – No	2 - Yes
24H. In your personal opinion could the following security failures during flight potentially result in major (or greater) safety failures to that aircraft: The authentication infrastructure (e.g., Public Key Infrastructure) used by the NAS becomes maliciously compromised so that it is no longer trustworthy?	2 – Yes 4 – No 1 – I don't know	2 –Yes 1 – No	1 – No 4 – NA	1 – Yes 4 – No	1 – Yes 1 - No
25. Have you designed or deployed, or do you currently plan to design or deploy, LANs onboard aircraft?	1 – Yes 6 – No	1 –Yes 2 – No	1 – Yes 3 – No 1 – NA	2 – Yes 3 – No	1 – Yes 1 - No
26. Who handles LAN-related security breaches onboard aircraft during flight?	1 – intelligent systems and ground-based personnel 6 – NA	1 – intelligent systems 2– NA	1 – ground-based personnel 4 – NA	1 – intelligent systems 1 – ground-based personnel 3 – NA	2 - NA
27. Will the onboard LAN potentially provide connectivity of avionics equipment to the NAS?	1 – Yes 6 – NA	1 –Yes 2 - NA	1 – No 4 – NA	2 – No 3 - NA	1 – No 1 - NA
28. Will the onboard LAN potentially enable connectivity of avionics equipment to the worldwide Internet infrastructure?	1 – No 6 - NA	1 –Yes 2 - NA	1 – No 4 – NA	1 – Yes 1 – No 3 – NA	1 – No 1 - NA

Survey Question	Survey Responses				
What is the primary role of your employer in regards to commercial aviation?	Build Components	Build Aircraft	U.S. Federal Agency	Consultant	Other
Number of respondents in each category:	7	3	5	5	2
29. Will the same physical onboard LAN provide connectivity for avionics equipment, crew members, and aircraft passengers?	1 – No 6 – NA	1 –Yes 2 - NA	1 – No 4 – NA	2 – No 3 – NA	1 – Yes 1 - NA
30. Will avionics equipment, crewmembers, and passengers within an aircraft all share a common external communications system (i.e., a common radio or satellite capability for air-to-ground and/or air-to-air communications)?	1 – No 6 – NA	1 –Yes 2 - NA	1 – No 4 – NA	2 – No 3 – NA	1 – No 1 - NA
31A. Which of the following security controls are deployed within the onboard LAN: Encapsulating communications between specific communicating devices within virtual private networks?	7 - NA	3 - No	5 - NA	2 – Yes 3 - No	2 - NA
31B. Which of the following security controls are deployed within the onboard LAN: Deploying packet filters within the LAN?	7 - NA	1 –Yes 2 - No	5 - NA	5 - No	2 - NA
31C. Which of the following security controls are deployed within the onboard LAN: Deploying (perimeter defense) firewall systems within the aircraft?	7 - NA	1 –Yes 2 - No	5 - NA	5 - No	2 - NA
31D. Which of the following security controls are deployed within the onboard LAN: Explicitly enabling Internet Protocol Quality of Service provisions within the LAN?	7 - NA	1 –Yes 2 - No	5 - NA	1 – Yes 4 - No	2 - NA

NAS = National Airspace System
FIPS = Federal Information Processing Standard

NA means that the respondent did not answer that question.

The first column responses are from people who said that their employer designed and built devices or software components for commercial aircraft. The second column was from responders who said that their employer designed or built commercial aircraft. The third column was from responders who said that their employer was a U.S. Federal government entity. The fourth column was by individuals who said that they were consultant to one of the other previously mentioned entities. The final column was from respondents who said that they worked for some other type of company.

Most respondents found only a subset of the questions relevant to their corporate function. Question 24 was the only technical question that was generically relevant to all respondents. It should be emphasized that there is no right or wrong answer to the various question 24 topics, because the individual answers are in terms of a specific context envisioned by the responder, with no common context being provided for each respondent. Thus, whenever a respondent answered "yes," that person was thinking of a scenario in which that event could result in a major fault. If they answered "no," they did not think of such a scenario. There is a loose correlation for those individuals whose corporate function was actively involved with making software components that tended to answer "yes" more frequently to the question 24 A through H than those individuals who worked for entities that were not actively creating aircraft software.

Yes answers to questions 24F and 24G represent potential challenges to this study in that it has been stated that confidentiality is not particularly relevant for safety. Nine respondents thought that maintaining the confidentiality of passenger lists can have safety implications and 11 thought that revealing the current location of aircraft could have potential safety implications. Should the FAA determine that confidentiality can have safety implications, then appropriate confidentiality controls will need to be added to the exemplar airborne network architecture recommended by this study.

Two different answers greatly surprised the authors of the survey:

- On question 17, it was expected to see many answers that aircraft software did not use any operating system—which three of the five respondents did. From the point of view of the study, this is a fine choice for the reasons discussed in section 4.4 of this report. However, the authors were particularly pleased that one respondent is using "A commercial operating system independently designed for high assurance uses (e.g., GreenHills Integrity Kernel)." From the point of view of this study, this is an outstanding choice (see sections 4.3 and 4.4). On the other hand, another respondent is also using "A general purpose commercial-off-the-shelf (COTS) operating system such as MS-DOS, a Microsoft Windows variant (e.g., Windows XP), Mac OS, or a Unix variant (e.g., Linux)." Section 4.3 of the study discusses why this choice is problematic.

- The above summary does not identify the relationships between a single respondent's answers. However, when that correlation is preserved, the respondents who are building Level A software are doing so by constructing software components with more than 4000 lines of C/C++ code—indeed, one respondent stated that their Level A software has more than 10,000 lines of C code. (Note: this does not count the lines of code for the operating system, if any.) Software this large has many opportunities to have numerous latent software bugs that attackers could leverage should that software be deployed in a networked environment. This is a matter of significant concern.

Three respondents wrote in helpful observations that provided additional insight into their responses to question 24. The following are direct quotations of their hand-written comments:

- "Any breech of security can result in danger to the thing the security was intended to protect. If it was not possible to cause harm to the aircraft or passengers by breeching the security, then why is there any security there

- to breech? Or is the question trying to assess probabilities? With respect to LANs, the safety risk is entirely dependent on what is connected to the LAN. If a safety critical device is connected to the LAN and there is a possibility of bad data getting to it, then there is, by definition, a safety risk. The severity depends entirely on what devices are connected to the LAN and what they use the LAN for. As written, it may not be possible to provide a meaningful answer to the question."

- "Please note: responses to 24 assume airborne LAN is a separate system from flight avionics. If this is an invalid assumption, all answers are 'Yes.'"

- "Note: the answer to most of these questions is heavily dependent on the architecture of the LAN. If there are bridges to avionics software then there is a security risk. If we are only talking about internet use of a LAN that is not in any way tied to the flight essential or critical software then the security risk may be mute."