# Considerations in Developing Survivable Architectures For Global Information Grid (GIG) Systems

## Defense Advanced Research Projects Agency


**August 2001**



**DO NOT DISTRIBUTE**

## Abstract

An information processing system is *survivable* if it can continue critical operations in the face of cyber attacks. The purpose of this paper is to help system architects develop survivable architectures for Department of Defense Global Information Grid (GIG) systems. It summarizes previous generations of security technology, describes the notion of survivability, identifies survivability principles, reviews current and near-term security technology, and discusses vulnerabilities that will likely remain even after current security technology is fully deployed. To illustrate the considerations in developing a survivable architecture for a GIG system, this paper uses the United States Navy's Global Command and Control System – Maritime (GCCS-M) Ashore as a representative system. Finally, this paper serves as a single reference for current technology trends in survivable networked computer system architecture development.

Keywords: assurance, cyber attacks, Cyber Panel, DARPA, DoD, GCCS-M, GIG, Information Assurance, IA, survivability, third-generation technology, threats, vulnerabilities.

## Acknowledgments

This document was prepared for the Defense Advanced Research Projects Agency (DARPA) by members of the following organizations: Mitretek Systems, The MITRE Corporation, the Institute for Defense Analyses (IDA), and the Naval Research Laboratory (NRL).

# Table of Contents

## Executive Summary

The Defense Advanced Research Projects Agency (DARPA) is developing new technologies to enable Department of Defense (DoD) Global Information Grid (GIG) systems to continue critical operations in the face of cyber attacks, that is, to develop *survivable systems.* This paper presents the results of one area of investigation: how to create system architectures incorporating technologies that promote system survivability, that is, to create *survivable architectures*. The purpose of this paper is to help system architects develop survivable architectures for GIG systems. These considerations apply to cyber, rather than physical, attacks on networked information processing systems.

Investigators chose a representative GIG system, the United States Navy's Global Command and Control System – Maritime (GCCS-M) Ashore system, as a representative for incorporating survivability into a networked information processing system. In presenting these considerations, the paper also serves the following functions:

- Provides a single reference source for current/near-term technology trends in survivable networked information processing system development
- Provides considerations for designing solutions to enhance system survivability
- Introduces the concept of a Cyber Panel, which would function as a layered plan, monitor and assess, and control mechanism
- Summarizes the types of known cyber attacks and methods for countering them
- Provides an approach to documenting the assurance that an information system reaches its survivability goals

Investigators structured the considerations documented herein in the context of third-generation security technology concepts, which accept the reality that complete protection is unattainable. However, these considerations strive to allow a complex networked information system's critical functions to survive despite ongoing cyber attacks.

Estimating the cost of alternative survivability approaches is beyond the scope of this paper, but eventually system architects must address cost issues. These cost issues should ensure the incorporation of survivability that is both effective and cost-effective. Finally, any effective survivability implementation must also be mostly transparent to end users and not impose undue burdens upon them.

Also, the projections concerning GCCS-M Ashore made here and elsewhere in this document have not been coordinated with those who have operational or maintenance responsibilities for this system and do not necessarily reflect their views.

This paper concludes that system-specific, as well as survivability, requirements must be identified to address various issues, such as sufficient system redundancy, system criticality and protection from attack, and system infrastructure strengths/weaknesses. The identification of such requirements could also form the basis for the fundamental claims of an assurance argument for the specific system. Thus, this paper aids GIG system architects in stating and meeting requirements for ensuring critical GIG system survivability.

# 1. Introduction

## 1.1 Background

The Defense Advanced Research Projects Agency (DARPA) is developing new technologies to enable Department of Defense (DoD) Global Information Grid (GIG) systems to continue critical operations in the face of cyber attacks, that is, to be *survivable*. No single technology will transform a conventional system into a survivable one. Rather, a *survivable system architecture* will provide a framework for incorporating different technologies to counter various potential attacks and to support the plan, monitor/assess, and control functions. The intent is to provide an aid to GIG system architects in stating and meeting survivability requirements.

Investigators chose a representative GIG system, the United States (U.S.) Navy's (USN) Global Command and Control System – Maritime (GCCS-M) Ashore system, to serve as an example for incorporating survivability into a networked information processing system. In presenting these considerations, this paper also serves the following functions:

- Provides a single reference source for current technology trends in survivable networked information processing system development
- Provides considerations for designing solutions to enhance system survivability
- Introduces the concept of a Cyber Panel, which would function as a layered plan, monitor and assess, and control mechanism
- Summarizes the types of known cyber attacks and methods for countering them
- Provides an approach to documenting the assurance that an information system reaches its survivability goals

## 1.2 Purpose and Scope

The purpose of this paper is to present considerations in developing survivable system architectures for GIG systems. The approach is to apply survivability considerations to the GCCS-M Ashore system. These considerations apply to cyber, rather than physical, attacks on networked information processing systems. The paper focuses on identifying these considerations in the context of third-generation security concepts, which accept the reality that complete protection is unattainable, and strives to allow a complex networked information system's critical functions to survive despite ongoing cyber attacks.

Estimating the cost of alternative survivability approaches is beyond the scope of this paper, but eventually system developers must address cost issues. These cost issues should ensure the incorporation of survivability that is both effective and cost-effective. Finally, any effective survivability implementation must also be mostly transparent to end users and not impose undue burdens upon them.

Note that the projections concerning GCCS-M Ashore made here and elsewhere in this document have not been coordinated with those who have operational or maintenance responsibilities for this system and do not necessarily reflect their views.

## 1.3 Document Organization

This document contains six sections and four appendices, as follows:

- Section 1: Introduction
- Section 2: Survivability Overview
- Section 3: Current/Near-Term Information Technology Trends and Residual Vulnerabilities
- Section 4: Representative GIG System Description: GCCS-M Ashore
- Section 5: Potential Future GCCS-M Architecture and Survivability Considerations
- Section 6: Conclusions
- Appendix A: Framework for Describing Vulnerabilities and Attacks
- Appendix B: Cyber Panel Component Types and Functions
- Appendix C: Cyber Panel Interface Requirements
- Appendix D: Survivable GIG Assurance Argument Development
- Appendix E: Active DARPA Projects Exploring Survivability-Related Technologies
- Appendix F: Acronym List

# 2. Survivability Overview

An information processing system is *survivable* if it can continue critical operations even in the face of cyber attacks. The goals of this paper are to introduce the concepts of survivable systems to a technically knowledgeable reader and to indicate the types of technologies that might be employed to improve the survivability of critical information processing systems.

Survivability in this sense is a relatively new notion, and so some general concepts that can be used to develop an architecture for a survivable system are introduced first. These concepts serve as the foundation for identifying survivable architectural "principles," which are subject to revision as experience is gained with their application. This section presents an overview of survivability, addressing the following topics: security technology generations, security technology terminology, survivability concepts, and candidate survivable architectural principles.

The scope of this paper is specifically the systems in the DoD's GIG system, and the chosen representative system is the GCCS-M Ashore configuration. It is in wide operational use, the functions it provides will continue to be essential to DoD operations, and its architecture is representative of many other GIG systems. The GIG encompasses an extremely broad set of systems, however, so any single example will be atypical in some respects. Consequently, different GIG systems may yield different tradeoff decisions and different survivability architectures.

Technologies that can help a system survive cyber attacks, and how they might be applied within a system, are the focus of this effort. Cyber attacks include those mounted through networks above the physical layer, those that attempt to inject malicious software into computers (either when they are being developed or when they are in operation), and those that attempt to exploit accidental software flaws or system capabilities maliciously.

Physical attacks on the infrastructure are excluded from consideration, including attempts to destroy computer or communications facilities with kinetic or electromagnetic energy and the jamming of wireless links. This is not to say that these attacks are not significant and should not be taken into account during a system design, but they are not addressed by the technologies considered here, and effective protective techniques for those threats continue to evolve.

The representative GCCS-M Ashore system is a headquarters, rather than a battlefield system. This distinction will help segregate the handling of physical attacks from cyber attacks. This paper focuses on the cyber attack portion of system survivability.

## 2.1 Security Technology Generations

The traditional definition of computer security calls for enforcing three properties:

- Confidentiality: protecting sensitive information against unauthorized disclosure
- Integrity: protecting sensitive information against unauthorized modification
- Availability: protecting sensitive information against unauthorized withholding

Similarly, the field of information assurance (IA) has evolved according to three generations of security technologies (see Figure 2-1). The following paragraphs describe the three security technology generations.

***First-Generation Security (1GS) Technology***

In military systems, the first generation of computer security measures aimed primarily at the first two of these properties *preventing* unauthorized release or modification of sensitive information, and researchers developed technologies that could be counted on to prevent data from leaking from systems or from being contaminated by information of lower integrity. Assuring availability received less attention from military secure system developers, both because it was thought to be too difficult to achieve and because solving the problem also seemed to be the responsibility of those concerned with the system's functional operation, not just those concerned with security [GOLL98].

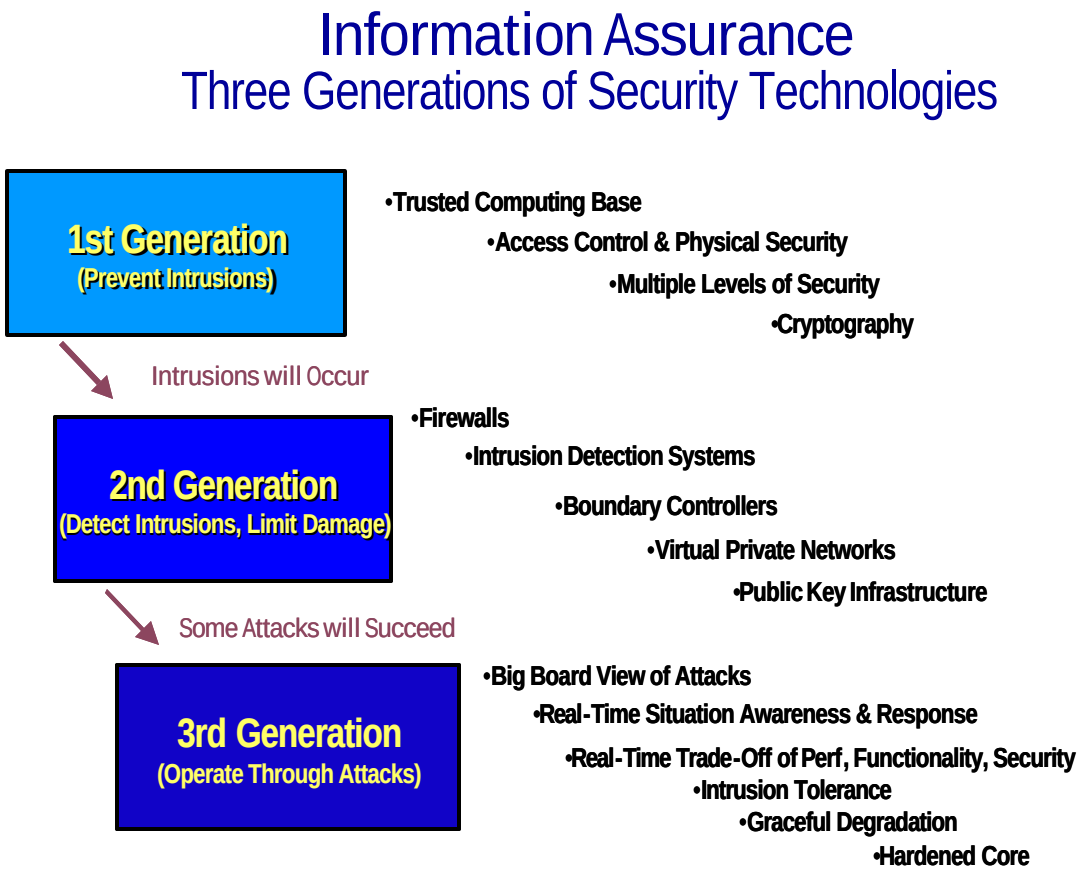


**Figure 2-1. Security Technology Generations**

Operational modes called *dedicated* and *system high* were defined, and time-separated *periods processing* was used to share scarce and expensive hardware among applications operating with different levels of classified data. As demand grew for more dynamic and flexible resource

sharing, the concept of *multilevel secure systems* was developed. Basic operating system protection structures, such as access control lists, were developed and analyzed, and, subsequently, security kernels and Trusted Computing Bases (TCBs) were prototyped and in some cases commercialized. The market adopted few of these technologies, however. Incorporating them into military systems without commercial market support raised cost, performance, and compatibility issues, and limited their adoption by the military as well. Cryptography continued to be an add-on technology for securing communications links, although efforts to develop end-to-end cryptographic systems were initiated.

***Second-Generation Security (2GS) Technology***

A second generation of security technology aims to *detect* intrusions and *limit* damage. Firewalls, intrusion detection systems (IDSs), virtual private networks (VPNs), and public key infrastructures (PKIs) characterize this generation, which has found much greater commercial acceptance. It applies strong access controls outside the individual system (via firewalls and other boundary control devices) and makes heavier use of cryptographic mechanisms for authentication, confidentiality, and integrity. Both commercial and military sectors are deploying this generation of technology now. However, commercial systems can also rely on insurance, law, and the military to protect them, so they are not generally so concerned with attacks by sophisticated adversaries with substantial assets as the military must be.

***Third-Generation Security (3GS) Technology***

Efforts are now underway to develop a third generation of security technologies and architectures for military systems that will have the ability to *tolerate* cyber attacks and continue to provide critical functions, possibly in a degraded mode, while an attack is in progress. These efforts are based on the observation that most security flaws are the result of program flaws [LAND94]. Because such flaws are unlikely to be eliminated from computer systems, system architectures that can tolerate their effects must be developed [BELL01]. Systems that have this ability are called *survivable*, because they can continue to operate even after an attack. This concept is explored in more detail throughout the paper.

Section 3 provides further details on these three security technology generations and the associated residual vulnerabilities.

## 2.2 Security Technology Terminology

***Vulnerability***

For the purposes of this paper, a *vulnerability* is a security weakness in a system. A vulnerability may be the result of a security flaw [LAND94]. There are many types of security flaws, and they may be accidentally or intentionally introduced. Security flaws may be introduced when the system is designed, during its implementation, or when it is being operated. The set of security vulnerabilities in any large system is likely to change as the system components and configurations change. The entire set of vulnerabilities at any time is probably not known by its developer, the operator of the system, or its attackers.

***Threat***

A *threat*, in the context of this paper, is the means through which the ability or intent of a threat agent to adversely affect an automated system, facility, or operation can be manifest; a potential violation of security.

***Threat Agent***

A *threat agent* is an opponent with the intent to harm a system; methods and things used to exploit a vulnerability in an information system, operation, or facility; fire, natural disaster, and so forth. Even a system with many vulnerabilities open to attack may not be damaged if there is no current threat agent operating against it. On the other hand, an increase in the level of threat agent activity can raise the significance of an otherwise minor vulnerability.

***Attack***

An *attack* attempts to exploit one or more system vulnerabilities and thus cause damage of some sort, either immediate or latent. Some vulnerabilities may be easier to exploit than others and the ease of exploiting a vulnerability may also change with time. Like vulnerabilities, attacks also change with time. As recently documented [ARBA00], the release of a script that automates an attack on a particular vulnerability can significantly increase the likelihood that a system will be damaged. An *insider* with legitimate access to a system may mount an attack by abusing privileges rather than exploiting a vulnerability.

Appendix A lists some known categories of vulnerabilities and attacks. These categories, also shown in matrix form, present a general framework for discussing attacks. The framework aims to be comprehensive, but it does not lend itself to providing a methodology for identifying system features or weaknesses that attacks could exploit. There is always the possibility that attackers will discover new exploitation avenues. Many successful attack methods depend upon exploiting human action (or inaction) as much as technical flaws. One of the most prevalent of these is the use of previously successful attacks to exploit system weaknesses for which countermeasures exist but have not been deployed.

## 2.3 Survivability Concepts

Network-centric warfare (as envisioned in [NATO00], for example) depends on highly available computing:

- Generic client workstations with processing and display capabilities
- Data servers providing shared access to large information pools
- Interface servers providing connectivity with external legacy systems
- High-bandwidth network communications connecting all components
- A wide variety of possible data push and pull arrangements

This dependence, however, makes these systems and the critical information they process attractive targets for a twenty-first century opponent. Hardening of DoD systems with 3GS technology, in concert with 1GS and 2GS technologies, will reduce their vulnerability to cyber attacks and bring the benefits of network-centric operations at lower risk.

Survivability differs from fault tolerance in that the faults to be tolerated are not random and accidental but instead malicious, persistent, intelligent, and adaptable. In particular, survivability relies on at least these primary capabilities:

- Structural defenses that avoid single points of vulnerability
- Detection of malicious intrusions, or their effects, throughout the computing and networking infrastructures used by the system under consideration
- Response to malicious attacks by thwarting, isolating, or confusing the attacker
- Recovery and reconfiguration techniques for enabling applications (systems) to continue despite successful attacks

Without identifying an attack, assessing damage, and countering the attack, 3GS techniques such as fragmentation and scattering, redundant servers, and dynamic reconfiguration can ensure system continuation for only a short time. This dependence on 1GS and 2GS protection technology complicates system recovery and reconfiguration because security state, as well as system state, must be captured and maintained to enable system survival.

To be survivable, a set of conventional (open-loop) systems requires plan, monitor and assess, and control functions to be added, so that attacks on the component systems can be recognized and appropriate defensive measures can be initiated, either manually or automatically. These functions should resemble a digital version of the notion of feedback control that gave rise to cybernetics [WIEN61].

The GIG, and indeed many lower level systems within it, integrates many individual systems into what is loosely called a *system of systems.* The collection of sensor, display, and control functions will be referred to herein as the *Cyber Panel* portion of the overall system of systems.

Because military systems will probably continue to be based largely on commercial off-the-shelf (COTS) technology being developed for a market that is unlikely to demand the high levels of survivability needed by military systems, much of the ability to tolerate cyber attacks must come from novel ways to tailor, organize, and exploit commercial systems so that the effects of their flaws are limited and will not cause overall system failures. To meet the goal of designing survivable, COTS-based systems, a small number of high-assurance (potentially non-COTS) components may be required, but their overall cost must be limited to a small fraction of total system cost.

This paper identifies considerations that a system architect must address to develop a system that can operate through cyber attacks, whether mounted through maliciously crafted packets sent over networks to evoke hidden system flaws, through implanted malicious code, or through actions of malicious insiders. As noted above, physical attacks (bombs or electronic jammers, for example) are not considered explicitly here. Nevertheless, many of the principles of survivable system design should lead to systems better able to cope with physical assault as well as cyber attacks.

### 2.4 Candidate Survivable Architectural Principles

The field of survivable system technology is too new to have a well-established set of architectural principles. Intuition and work in related fields stimulates identification of the following candidate survivable architectural principles.

***Use available cost-effective prevention mechanisms from 1GS technologies for basic protection***.

Most COTS operating systems implement protection domains at some degree of assurance. Even though flaws in those systems, and in the applications that run on them, can be exploited to defeat security controls, it is still true that good administration of the controls these systems provide can significantly increase the difficulty for an attacker.

***Use firewalls, intrusion detection, and commercial cryptography from 2GS technologies to further filter out unsophisticated attackers***.

Commercial security components and packages (again, well managed) can significantly limit the paths an outside attacker can use to mount an attack and increase the chance that an attack will be detected. They can improve the accountability of security-critical operations and detect changes to software components and configurations.

***Avoid single points of failure.***

This is a basic principle of defense against random faults as well as malicious attacks, and implies the need to provide redundant capabilities. *Redundancy* can be applied through various approaches: spatial, temporal, analytical, and so on. These approaches have been studied extensively for fault tolerance [LEEA90], but they are only now beginning to be applied to situations in which malicious attacks are expected. With redundancy comes a need for *redundancy management*, which has sometimes proven the Achilles heel of fault-tolerant schemes. Inadequately managed redundancy can present the single point of failure that the redundancy was intended to avoid. Redundancy management requires maintaining the synchrony and consistency of the redundant elements, detecting and confining errors, and rapidly reconfiguring the elements in case a failure (perhaps in the face of an attack) occurs.

***Design for graceful degradation.***

Shedding less critical functions when an attack damages some resources provides a better chance for accomplishing mission-critical functions. Degraded modes can take many forms. For example, rather than shedding functions entirely, the same set of functions may be provided, but with increased latency. Or, the system may be able to perform as it did before the attack, but with diminished resistance to a further attack if, for example, a primary system is replaced by a single backup system.

***Exploit diversity to increase the attacker's work factor.***

A set of identical redundant systems incurs the risks of a monoculture. If the attacker finds a way to subvert a single system, he/she can potentially defeat them all with little additional effort. Systems that are diverse in various dimensions (hardware, operating system, application, programming language, and so on) can be much more difficult to defeat entirely. This is another tactic to use redundancy to avoid a single point of failure, and it requires the same attention to redundancy management noted above.

***Disperse and obscure sensitive data.***

Concentrating valuable resources at a single, visible location provides a high-value target that can justify a high investment on the part of an attacker to overcome the defenses being deployed.

Obscuring the value of an asset and dispersing it so that capture of any fragment is of little value to the attacker removes this justification. An example is provided by the techniques of fragmentation and scattering. These employ cryptography to implement secret-sharing schemes, so that an attacker must intrude on several different systems to reconstruct particularly sensitive information. This is another way to avoid a single failure point.

***Make the system dynamic and unpredictable.***

A static target is easier to hit in cyberspace as well as in physical space. As flaws of systems become known, they become more vulnerable. A system that changes more rapidly than the rate at which flaws are exposed is harder to penetrate. Introducing *randomness*, *uncertainty*, and *unpredictability* into a dynamic system can multiply the attacker's difficulty. A system that varies its behavior randomly within a tolerance limit (as in random sequence number generation, for example) is harder for an attacker to imitate. Another approach is to create behavior that appears random to the attacker but is deterministic to a friendly system. Frequency-hopping communications systems exemplify this approach.

***Deceive the attacker.***

An attacker who cannot discover the characteristics of his/her target, or who thinks a target is one thing when it is really another, is at a disadvantage. Many attacks depend on knowledge of system details and sometimes on timing relationships. Masking a system's "fingerprint" (its characteristic responses to external requests) is another way to hinder the attacker. Finally, some systems can be made stealthy from a network perspective. What can't be seen is hard to attack.

# 3. Current/Near-Term Information Technology Trends and Residual Vulnerabilities

This section first presents a view of the computing facilities that are likely to be deployed in GIG systems in the next few years. Vulnerabilities that will probably remain even following this deployment are then discussed.

## 3.1 Current/Near-Term Information Technology Trends

3GS survivable architectures will not be available for a few years. In the meantime the information technology (IT) environment will continue to evolve. Technology advances and vendor product offerings influence and shape the technology environment. This section describes changes in the IT environment believed likely to occur while 3GS survivable architectures are being developed. These architectures should be able to accommodate the new developments. Some of the developments may contribute to the overall security and survivability of these future survivable architectures.

The significant changes in the IT environment that affect survivable architectures will likely involve the logical progression of the current COTS platforms and software suites, the incorporation of public key (PK) cryptography, software trends to support networks, and hardware trends. The remainder of this section describes these trends in more detail. Some technology likely will not change; for example, most systems will probably continue to operate at a single security level.

DoD organizations generally use COTS systems. Personal computer (PC) systems form the core of most user workstations. The primary software used on these systems generally consists of the typical office productivity suites. These suites provide functions for producing documents, spreadsheets, and presentations and for communicating through electronic mail. COTS server systems perform functions that involve large collections of data shared by multiple users or require computational resources beyond those typically available in workstations. The server systems use COTS systems to provide Web, database, mail, and collaboration services. The information that these services manage and provide is tailored to specific DoD functional communities and uses.

The hardware aspects of COTS systems are constantly changing to incorporate the latest technology. The changes generally include faster processors and more memory. The size of both fast, temporary memory (e.g., random access memory) and permanent, persistent memory (e.g., hard disk) will continue to increase.

COTS software also continues to evolve. The major operating systems and systems software products usually undergo major upgrades on about a 2-year cycle. Minor upgrades occur more frequently. Major trends include incorporation of more multimedia, particularly voice. Before long, the PC may well incorporate the functions that a telephone currently provides.

### *Public Key Cryptography*

Network security concerns have driven interest in public key cryptography and its use. Efforts are already underway to use PK technology in DoD systems. The Defense Messaging System [DEFE99] is relying on COTS e-mail systems that secure messaging using PK techniques following the Secure Multipurpose Internet Mail Exchange (S/MIME) standards. The DoD has deployed a PKI that will allow DoD organizations and personnel to communicate securely among themselves and with key non-DoD partners. The PKI allows parties to communicate without any prior arrangement, other than being listed in the key directories.

The DoD also has a pilot program underway that is intended to lead to issuing hardware-based tokens (e.g., smartcards) containing keys and certificates to all DoD personnel. These tokens would provide strong security for private keys. The use of these tokens, together with PK-aware systems, provides a basis for strong authentication and protection of data, particularly for network applications.

The personal tokens interact with PK-enabled applications to provide security, in the form of public key-based authentication, signature, and encryption, at the higher levels (e.g., application level) of the network protocol stack. Communications that support either unaware (i.e., not PK-enabled) applications or functions at lower levels of the protocol stack would not be protected. Standards to secure several of the lower level protocols have either been completed or are nearing completion, and products and services supporting the secured protocols are beginning to emerge.

Specifically, there are secure versions of the Internet Protocol (IP), known as IP Security (IPSEC) [KENT98]; the Domain Name System (DNS), known as DNS Security (DNSSEC) [EAST97]; and Border Gateway Protocol (BGP), known as Secure BGP (SBGP) [LYNN99, KENT00].

Enabling these protocols requires issuing additional tokens to devices and services. These tokens will provide a means for the devices and services to authenticate each other as well as secure their communications. The DoD PKI is already being used to issue digital certificates for devices and services. DoD systems will increasingly employ these secured protocols over the next few years.

### *Security Levels*

Systems will probably continue to operate primarily at a single security level. The mode of operation will be *system high*; all information is protected at the level of the most classified information the system is approved to process, and information to be exported from the system is treated as if it has that classification. In general, networks and the nodes that they connect will all operate at the same security level. Currently, selected systems are allowed to serve as an interface between networks and systems operating at different security levels. GCCS-M receives information from systems operating at both higher and lower levels. The information flows through interface nodes such as Radiant Mercury and other guard systems.

These interface systems allow GCCS-M to receive information from intelligence systems operating at higher security levels and to exchange e-mail through an e-mail guard with users of networks and systems at lower levels. These systems allow only selected information to flow between the two networks. Information passes from high to low only if certain conditions are met.

The conditions may range from manual review to strongly formatted information that must pass automated reviews of content and consistency.

New capabilities may be developed to allow users to interact more flexibly with systems operating at different processing levels from a single workstation. The new capabilities may involve special interface systems that enforce one-way flow of information from systems or networks operating at one level to systems or networks operating at a higher level or that have only volatile memory storage that allows fast switching of the system's operational security levels.

Security of one-way flow involves preventing any information flow from the higher network to the lower network, including acknowledgments that are part of many network protocols. One-way security also involves preventing covert channels such as timing channels. The one-way interface would have to provide a guaranteed throughput level that could not be affected by actions of elements operating on the high side of the interface.

Systems, particularly workstations, without persistent memory could relatively quickly change their security levels of operation. The level changes would have severe restrictions. Lowering the level of security operation would require that the memory be purged to prevent unintentional downgrading of information.

Permitting one-way upward flow of information introduces no confidentiality risk but can introduce an integrity risk. Viruses or other forms of malicious code could transit an upward link from a low system to a high system. Upward flows can be filtered, but today's filters can only eliminate malicious code that they can recognize as such. Alternatively, it may be possible to limit upward flows to only those that can be assured to be safe.

### *Software Control*

The increase in sophistication and capability of workstations has presented challenges in managing those workstations. Network administrator and IT staffs have the daunting task of maintaining the workstations. Ensuring that each workstation has the current versions of software can be overwhelming. Organizations acquire their systems on a continuous basis. Newly arrived systems often contain newer versions of operating systems and common software than systems that arrived earlier and are still in use. The difficulty of managing large collections of systems has driven the development of tools and techniques to support centralized system administration and will probably continue to do so. Such tools will allow systems to be configured and updated without requiring one-on-one interaction of the support staff with the individual systems and their users.

Systems will include measures to control a system's acceptance of new software. Some systems can require that programs must be digitally signed by an entity believed to be trustworthy before accepting the code for execution on the system. However, the digital signature checks only establish "pedigree;" which indicates that the program came from a known source. The signature does not guarantee that the programs cannot be maliciously used, although organizations responsible for the development and distribution of software for survivable GIG systems should establish both technical and procedural measures to ensure that programs are safe. The measures should ensure that no one was able to place malicious code in the distributed software and that the software was examined and tested to reduce the probability that it contains errors that would allow it to be exploited for malicious purposes.

The desire to minimize the effort required to distribute and maintain software has increased and will probably continue to spur the use of *thin clients.* Initial client-server applications involved a *thick* or *fat client*. The thick client provided significant processing capabilities to process and present information. For example, a database client would query a database server for information. The client would process, organize, and present the data. If the user had the ability to add to or modify the data, then the client could solicit the user's inputs and send the changes to the database. The principal disadvantages of the thick client approach are that developers need to create software for a variety of client platforms, and application managers need to provide the capability for and manage the distribution of client software to the entire user population. This distribution effort includes providing client software and supporting the installation of the software on each client's system.

The *thin client* approach offers some improvements. The thin client presents information to the user and accepts inputs and modifications. A thin client is not generally dedicated to a particular application and can support multiple applications. Examples of thin clients are Web browsers and windows terminals. Also, Microsoft Windows terminals act as servers for managing Windows systems and interact with other servers that host applications. With thin clients, application managers do not need to deliver and install (or support the installation of) new client software to all client workstations. Application-specific processing occurs at a server. The client and server together replace the thick client. The first-level server may in turn call on other servers for other information. Typically, the client is a Web browser, the first-level server is a Web server, and the second-level server is a database server. The second-level server is the server with which the thick client would formerly have interacted.

Use of thin clients shifts processing that had once occurred in the client to a server shared by multiple users. The server may then have to interact with other servers that manage and supply the data needed at the client. There have been efforts to develop systems for generalized distributed processing that support the interactions among the servers as well as between the clients and the first-level servers. These *middleware* systems allow applications processing on one system to call upon services provided by another system. In an object-oriented [FAYA99] model, an application can employ objects whose representations and methods are maintained on another system.

Initially, the relationship between the objects and the network locations that managed the objects was fixed or static. However, currently efforts are ongoing to allow applications to reference the services by the service name without knowing anything about the network address of the systems that provide the service. A directory service similar to that provided by the DNS will map the service name to the network location that provides the service. This directory-based service will allow services to be relocated or the level of resources providing the service to be changed transparently to the service users.

A final trend to be noted in deployment and management of 2GS technology is represented in the recent Navy-Marine Corps Intranet (NMCI) procurement. In effect, the Navy and Marine Corps have turned to private industry to deploy and manage not only office computing resources but also the related security functions, including deployment of 2GS technologies: firewalls, PKIs, VPNs, and IDSs. If this approach is successful, it could have major effects on how technology will be deployed and supported throughout the DoD and government as a whole. This approach seems unrelated to new technology development or the development and deployment of 3GS systems, however.

### 3.2 Residual Vulnerabilities

As discussed in detail in Section 2, 1GS technology addressed devising *protection* mechanisms against known types of attacks. While these mechanisms are quite effective against such attacks, they cannot protect against attacks for which they were either not designed or not configured to protect.

2GS technology addresses limiting damage and providing *detection* mechanisms for known types of attacks. By limiting the traffic that enters or leaves a protected system, firewalls limit the kinds of attacks that can be mounted against that system. PKI can improve authentication of users and software. VPNs can protect traffic flowing over an unprotected channel from observation or undetected modification. And IDSs can detect certain kinds of attacks and in some cases thwart them.

But even when 2GS mechanisms are fully deployed, significant vulnerabilities will remain. For example, as firewalls have limited the ports open to traffic, more protocols and systems are being designed to send their traffic over the few ports that virtually every firewall must leave open: the ports used for Web access. As protocols supporting distribution of mobile code begin to use these ports, firewalls may actually become less effective at stopping malicious code. PKI can be used to authenticate individuals, documents, and even software, but it cannot guarantee that an authenticated piece of software is free of malicious code.

VPNs can have a similar effect. Unless the VPN is terminated at or outside the firewall, it cannot monitor traffic flowing into the system it is trying to protect, because the traffic is encrypted. Also, VPNs protect traffic contents against eavesdropping and modification, but they do not protect against observation of unencrypted packet headers, which can reveal who is talking to whom.

Finally, 2GS IDSs are subject to numerous limitations. False positives are significant problems. Unless a method can be built to mitigate the base rate fallacy [AXEL99], false positives will continue to occur.

Both 1GS and 2GS technologies are vulnerable to attacks that, for any one of a number of reasons such as those presented below, do not match the criteria that have been programmed into them for protection and detection. For example:

- A hidden flaw in the software/firmware/hardware that is exploitable and is discovered by an attacker before this vulnerability becomes known to and is corrected by a defender.
- Malicious code that is already embedded in a system by an attacker who accesses and exploits it.

Additionally, any protection or detection system is vulnerable to attacks that exploit misconfigured settings of such systems. For example:

- Misconfigured firewalls (e.g., allowing hazardous ports to remain open)
- Misconfigured (or not configured at all beyond the default setting “out of the box”) operating systems. A typical example is retaining default passwords.
- Misconfigured intrusion detection software
- Misconfigured architectures (e.g., placing sensitive database servers outside of a protected environment)

The goal of 3GS technology differs from that of 1GS and 2GS technologies in a fundamental conceptual way. It accepts that, despite the best efforts to prevent attacks from succeeding, some will inevitably slip through for any one or more reasons such as those listed above.

Thus, the goal of 3GS technology is to endow the protected networked information system with the ability to function in the face of many attacks, possibly in a degraded mode where the most crucial services continue to be available, but less critical ones do not. Exactly how this goal can be attained will vary according to the system's mission, functions, threats, and resource constraints. As in any new endeavor, success is not guaranteed. Some principles to be applied have been outlined above, and further details on possible approaches are provided below.

# 4. Representative GIG System Description: GCCS-M Ashore

This section first describes GCCS-M systems and then focuses on GCCS-M Ashore, which is the GIG system selected as a representative for the application of survivability considerations described in this document. Topics covered below include the GCCS-M Ashore mission, major functions and information flows, constituent technologies, and potential degraded modes. Also presented is an operational scenario to provide insight into GCCS-M activities. The section concludes with a list of desired behaviors for survivable GCCS M systems.

As the U.S. Navy's implementation of the GCCS, GCCS-M is the Office of the Secretary of Defense's designated command and control migration system for the Navy. The stated mission of GCCS-M is to provide centrally managed services to the Fleet, allowing both U.S. and allied maritime forces to operate in network-centric warfare operations (for further information, see Section 2.3).

GCCS-M is a consolidation of multiple command and control systems into one Navy-wide system. After Operation Desert Storm in the early 1990s, a decision was made to integrate disparate Navy command and control systems into the Joint Maritime Command Information System. In 1998, this entity was renamed GCCS-M, to be consistent with the terminology used to refer to the Joint GCCS, GCCS-Army, and GCCS-Air Force, and to reflect the migration of GCCS-M applications to the Defense Information Infrastructure Common Operating Environment (DII COE). Figure 4-1 shows the GCCS-M high-level information flow and functions.

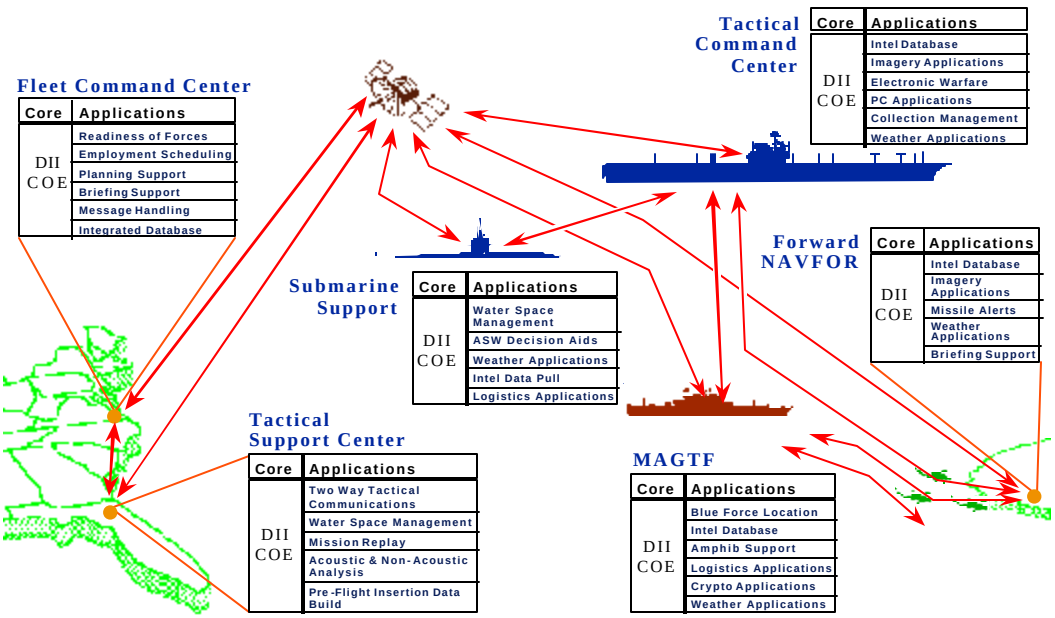


**Figure 4-1. GCCS-M High-Level Information Flow and Functions**

GCCS-M operates in the following deployments:

- GCCS-M Afloat is installed on ships and some submarines.
- GCCS-M Ashore is deployed at fixed command centers on shore.
- GCCS-M Tactical/Mobile is also deployed ashore but is tailored for various special purposes and is more oriented toward mobile and tactical operational use.

The rest of this paper focuses on GCCS-M Ashore. References to GCCS-M below, unless otherwise noted, should be assumed to refer to GCCS-M Ashore.

### 4.1 Mission

The GCCS-M mission is to provide Navy commanders with an integrated, multifunction system for command, control, communications, computers, and intelligence (C4I) that integrates the receipt, processing, analysis, display, and communication of a variety of data of tactical and strategic interest. More specifically, GCCS-M focuses on geolocation and intelligence information about friendly, hostile, neutral, and uncategorized targets on the sea, air, and land. The system receives, processes, and displays sensor data (e.g., tracks from processed radar data), but also processes and displays environmental data (e.g., weather) and intelligence information (e.g., additional information about specific objects being tracked).

A key function of GCCS-M is to present the commander with a Common Operational Picture (COP) of the Battlespace. GCCS-M also processes and displays information on unit characteristics, logistics supply status, combat readiness, and tactical disposition of U.S. and Joint Tactical Force coalition units. In addition, GCCS-M supports the sending, receiving, and review of a variety of kinds of message traffic (e.g., from Air Tasking Orders to messages with briefing attachments). GCCS-M supports near real-time weapon allocation and targeting data generation for submarines.

Other key functions supported by GCCS-M include:

- Mission planning (e.g., providing the ability to acquire, analyze, control, and disseminate pertinent antisubmarine warfare [ASW] mission-planning data and provide safety of flight planning for Maritime Patrol Aircraft to/from operational areas and for coordinating turnover on station).
- Imagery management and analysis (e.g., viewing imagery data, determining measurements from images [mensuration], and monitoring imagery data transmission/receipt, and output)
- Targeting and tracking analysis/support functions (e.g., providing contact location data and precise Over-the-Horizon Targeting [OTH-T]data to submarines equipped with Tomahawk Missile variants, correlating single link attributes or single emitter electronic intelligence [ELINT] tracks)
- Some aspects of displaying/managing networks (e.g., displaying GCCS-M segments loaded on each workstation and providing user-oriented network monitors)
- Briefing and office automation support, and continuous C4I, Surveillance, and Reconnaissance (C4/ISR) services to assigned U.S. and Allied Maritime Patrol Aircraft, Special Mission aircraft, and other ASW forces operating independently or as part of a Battle Group.

The list below identifies GCCS-M major functions and information flows.

## 4.2 Major Functions and Information Flows

***Core***

- Provides a single integrated C4I system that receives, processes, displays, maintains, and assesses the unit characteristics, employment scheduling, material condition, combat readiness, warfighting capabilities, positional information, and disposition of own and allied forces.
- Provides commanders with a timely, authoritative, fused, and common tactical picture with integrated intelligence services and databases.
- Enables commanders to plan, direct, and control the tactical operations of forces under the commander's operational control.

***Briefing Support/Office Automation***

- Provides a UNIX- and Microsoft Windows NT-based, multiscreen and multinode briefing display and control applications (enabling integration of office automation).
- Provides Unix- and NT-based applications for building Maritime Patrol Craft (MPA) briefs for the U.S. Navy and Allies in North Atlantic Treaty Organization standard format.
- Prepares, displays, and prints briefing text.

***Database***

- Maintains an authoritative history of tracking information in a relational database format.
- Provides database maintenance capabilities for Tactical Support Center (TSC) segments.
- Extracts data from received United States Message Text Format (USMTF) messages and populates various databases.

***Local Area Network/Wide Area Network***

- Displays GCCS-M segments that are presently loaded on each workstation; provides a user-oriented network monitor.
- Provides tools for Anchor Desk capabilities through a wide area network.

***Mission Operations***

- Provides contact location data and precise OTH-T data to submarines equipped with Tomahawk missile variants.
- Provides warfighting capabilities for surface, air, and subsurface platforms.
- Provides those forces with and integrates the waterspace picture for timely asset management.
- Detects and displays information on physical threats for warfare commanders embarked on GCCS-M-equipped platforms.
- Correlates, maintains, and analyzes tracks. Analyzes tactical platform sensor data for dissemination to other fleet units.
- Correlates single-link attributes or single emitter ELINT tracks.

- Provides a graphical post-ASW mission replay capability.

***Mission Planning***

- Provides Navy Command and Control Systems Ashore units as well as units afloat with the ability to acquire, analyze, control, and disseminate pertinent ASW mission-planning data.
- Provides safety of flight planning for Allied Maritime Patrol Aircraft to/from operational areas and for coordinating such aircraft turnover on station.

***Imagery***

- Supports the full range of imagery requirements, including viewing, mensuration, transmission/receipt, and output.
- Supports near real-time receipt and transmission of tactical imagery data to/from Antisurface warfare Improvement program (AIP) aircraft.

***Intelligence***

- Provides comprehensive military intelligence data and message applications.

***Logistics***

- Assesses short-term operational sustainability requirements for Naval forces afloat.

***Meteorological/Oceanographic***

- Provides applications and tools to process environmental data received from Meteorological and Oceanographic (METOC) production or regional centers.

***Communications***

- Provides continuous C4I/SR services to assigned U.S. and Allied Maritime Patrol Aircraft, Special Mission aircraft, and other ASW forces operating independently or as part of a Battle Group.

Figure 4-2 shows the GCCS-M architecture today.

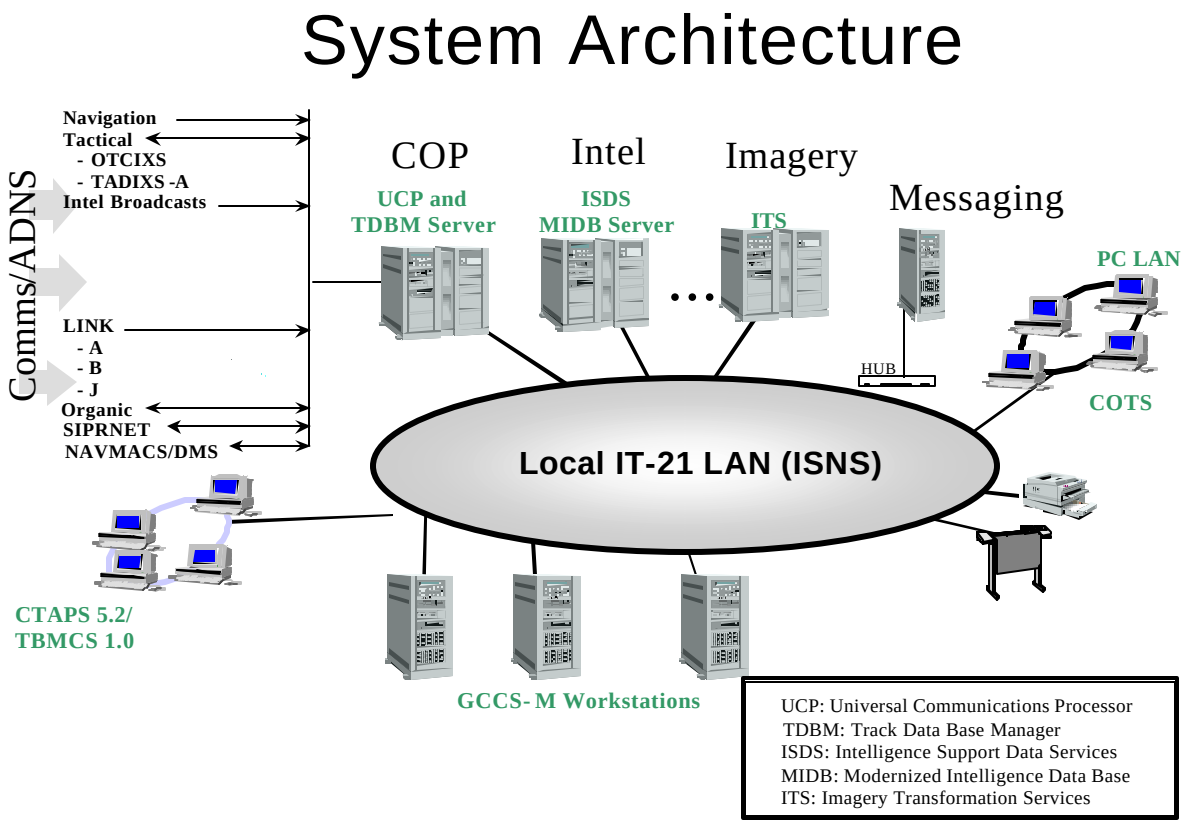


**Figure 4-2. GCCS-M Architecture Today**

### 4.3 Constituent Technologies

Constituent technologies also affect GCCS-M operations. One key aspect of the underlying implementation of GCCS-M arises because GCCS-M represents the consolidation of multiple disparate Naval command, control, and communications systems. Thus, GCCS-M comprises a number of different constituent subsystems, including Shared Data Server, Imagery Transformation Services, Track Management System, and the Naval Joint Surveillance and Target Attack Radar System (JSTARS) Interface. The specific list of constituent subsystems is not as important as the general observation that GCCS-M is the consolidation and integration of a number of applications that in most cases were *not* designed with each other in mind. The resulting integration challenges are especially important when a part of GCCS-M changes; that is, great care is required to reduce exposure to unintended consequences as various loosely integrated systems interact.

A second key aspect of the underlying implementation of GCCS-M is the adoption of the DII COE and commercial products. GCCS-M will use Sun Microsystems' Solaris servers for all Afloat, Ashore, and Tactical/Mobile installations. Workstations will be a mixture of Solaris machines and PCs, with the preponderance being PCs. The system will run on up-to-date versions

of the DII COE and commercial operating systems. GCCS also uses commercial products such as the Sybase database management system (DBMS).

### 4.4 Potential Degraded Modes

It is difficult to define desired degraded modes of operation in general, since so much of what is critical for GCCS-M to support depends on the context and situation (e.g., peacetime, war, interdiction, exercise). Three categories of degraded mode options are as follows:

- General ability to continue to deliver service despite compromise or loss of some infrastructure
- Reduced performance (e.g., worse track update latency, longer times to complete queries)
- Function shedding (e.g., fewer kinds of tracks displayed or fewer functions supported at all)

Although not really a specific degraded mode, the general ability to tolerate and recover rapidly from failures is a clear desire among operational users. As one example of current user requests in this area, the consensus of one group of GCCS-M users was that "capabilities are required to assist in optimizing configuration management, especially when unanticipated systems outages/reboots occur. Once channel configurations are set, there should be an archive capability that saves all channel configuration parameters that have been set up in the system, which is retrievable after catastrophic system failure."

The following functions have been suggested as the key issues for recovery:

- Core kernel processes (what processes are registered and how do I recover?)
- Core Track Management (Common Operational Picture) functions (what was the state of my track database and the server/client configurations?)
- Unified Communications Processor (UCP) configurations and display configurations (what was the configuration at the time of failure and how do I recover?)

Thus, all of the after-action review and analysis and training capabilities could be shed in a crisis (e.g., the graphical post-ASW mission replay capability). In most scenarios, the office automation and briefing preparation and display capabilities are not critical, although in some situations they may be important.

Time sensitivity is one criterion for determining those capabilities that would likely be retained as critical in a degraded mode. A clear example is the near real-time track display of friendly, neutral, and hostile units, especially fast-mover tracks (which may be an inbound hostile aircraft or missile).

### 4.5 Operational Scenario

This discussion provides an operational scenario for illustrating GCCS-M activities. As a basis, GCCS-M comprises many applications that share a common infrastructure built of COTS hardware and operating systems. Many of the GCCS-M applications exploit commercial software, such as browsers, office suites, and databases.

Whether a GCCS-M function is critical depends not only on the nature of the function but also on the purpose for which it is invoked. A weather report might provide information that is merely useful (for example, if it is needed to decide whether to wear a coat to a meeting, or critical (such as needed to help route a rescue mission).

Its heritage has led today's GCCS-M to operate with some servers providing dedicated functions. This architecture represents a local optimization: each system may be assured that no other can usurp its resources, yet excess, unusable resources will remain, and it will be very difficult to exploit this wasted capacity to improve survivability.

Based on investigations for this document, projections[1] for a future GCCS-M architecture would permit more flexible sharing of resources, so a pool of servers would be available to execute applications. Also, priority mechanisms will be in place to assure that the most critical applications would be served. In addition, future architectures will most likely use the Transmission Control Protocol (TCP)/IP protocol suite for communication among components and will exploit advanced middleware and other emerging commercial software technologies.

A future cyber attack intending to impede mission operations that depend on GCCS-M might well target its COTS infrastructure at a critical time. A survivable architecture aims to blunt the effects of such a hypothetical attack. To make both the attack more plausible and the characteristics of such an architecture more definite, the following hypothetical scenario is proposed.

GCCS-M is to be employed in the planning and scheduling of a maritime patrol mission. The task requires at least the following functional requirements: (a) access to mission-planning software, (b) access to information to identify the equipment needed and to determine its readiness, (c) access to personnel information to identify a crew, (d) access to weather and operational picture information to plan a route, and (e) the ability to receive and send messages for transmission of orders.

The above functional requirements translate to hardware and software that must be available and in working order. Databases, servers, and routers that handle access to mission-planning information and personnel information must be functional. Databases and online interfaces and servers that handle weather information must also be in full working order.

Finally, communications must be available to handle the traffic to and from the personnel who will execute each mission. This capability involves not only the radio equipment but also the computers that oversee channel selection, encryption, access authorizations, and uploading and downloading of data, for example.

While each function may seem self-contained and clearcut, it must involve numerous interfaces that must be maintained in working order. For example, transmission of sensitive operational data involves, among others, data servers, routers, and databases that contain information indicating who is authorized to access what data; computer-controlled setting of modern military software-controlled radios; and oversight of telecommunications protocols for error correction.

Those functions, in turn, depend on the availability of current data and thus the existence of an additional layer of hardware and software in operational status. This dependence chain can extend quite far both in terms of hardware/software and time; an outage in some feeder function may be tolerable in the short term, but it could quickly make operational data obsolete if that feeder function remains inoperable for long.

---

[1] The projections concerning GCCS-M Ashore made here and elsewhere in this document have not been coordinated with those who have operational or maintenance responsibilities for this system and do not necessarily reflect their views.

A cyber attack might target any of the information sources or flows listed above or it might target the COTS infrastructure that supports them. It could exploit publicly known but unrepaired flaws in the systems, flaws not publicly known but known to the attacker, or malicious code either embedded in the development process or inserted (perhaps through a malicious e-mail attachment or through a mobile code mechanism). A survivable GCCS-M architecture must equip the system so that even if some particular attack succeeds, the planning and scheduling tasks can be completed.

### 4.6 Desired Behaviors for Survivable GCCS-M Systems

The key behavior of a survivable GCCS-M system is its ability to continue operation, in a degraded mode if necessary, in the face of a cyber attack that causes some damage to the system. The basic desired behavior of survivable systems, which distinguishes them from earlier security technology generations, is the ability to monitor their own activities and their environment and to use the results of this monitoring to adapt their behavior to attacks in progress. That is, a survivable system must have *survivable critical functions.*

The types of attacks commonly reported today often proceed in stages and exploit security flaws in target systems. The attacker first pings a system looking for vulnerabilities, then finds a way to gain unprivileged remote access, and finally crosses internal barriers to gain fully privileged access to the target machine. Distributed denial-of-service (DoS) attacks involve launching this sequence for many separate hosts with the objective of installing malicious software that then can be triggered to mount other attacks against a victim from a wide variety of network locations.

Efforts to characterize system security in terms of the barriers between an attacker and fully privileged access were studied by Dacier, et al. [DACI96]. Security flaws, vulnerabilities, and attacks have been described and categorized by several authors for varying purposes [LAND94, ASLA96, BISH95, HOWA97, KEND99] and are available in databases (Internet Categorization of Attacks Toolkit [NATI01]) and dictionaries (Common Vulnerabilities and Exposures [COMO01]).

A serious cyber attack will have a purpose: disruption or usurpation of system operation at a critical time, misdirection of critical resources by modifying critical information, gaining access to sensitive information, or the like. In some cases, particularly attacks aimed at system disruption or usurpation, the attack's effects will at some point become highly apparent to the target. In other cases, where the goal is simply to gain access to confidential information or stealthily modify data to misdirect resources, the attacker may try to obscure the attack indefinitely.

To execute any such an attack, a rational attacker first collects as much information as possible about the target. This collection may be carried out without any access to the target system and so may not be detectable through technical means. The result of this activity may be to identify a security-relevant flaw in a component of the target system; it could even be the initiation of an activity to implant such a flaw. The actual execution of the attack may then involve simply triggering an implanted flaw or sabotaged mechanism at a particular time.

Ideally, a survivable system should, in the face of an arbitrary attack, degrade gracefully and recover fully following the attack. Although this behavior is desired, in practice it is difficult to reason about the effects of arbitrary, unanticipated attacks. Because new attacks are continually discovered, it is hard to specify even the set of all known attacks. It is usually necessary to settle for designs that can resist specified attacks to a greater or lesser degree.

Appendix A presents a matrix that relates known attacks to some available countermeasures. It is offered as a framework but should not be viewed as complete. A developer of a specific survivable system will need some similar framework to support specific assertions and assurances about the system's ability to survive attacks. This need is discussed further under the topic of assurance arguments.

### *Operating Through Attacks*

To operate through a phased attack, a capability to sense and report the state of the protected systems is essential and may be present at a variety of levels in the system. Specific modes of degraded operation for GCCS-M will need to be identified. These modes may involve shedding some noncritical functions, but they could alternatively increase the latency of some or all functions or they could mandate operating with the same functions and latency but diminished ability to withstand additional attacks.

At some levels and for specific attacks, automatic responses may be possible; for example, detection of a virus might result in automatic reconfiguration of firewalls. At higher reporting levels, humans will be involved for the foreseeable future, and so another aspect of the survivable system behavior will be a means to display system state and offer alternative courses of action to operators.

The GCCS-M operational scenario outlined above identifies five general functions as critical. Those functions depend on lower level functions and on system hardware and software resources, which, collectively, are by inference critical as well. Further, different higher level functions may depend on common lower level functions or resources. An attacker seeking to deny service may target such shared resources to maximize the effect.

### *Withstanding Specific Attacks for Specific Time Periods*

For any networked system that provides several different services, such as GCCS-M, it is more meaningful to think in terms of the temporary unavailability of a service *x* for *t* seconds than in terms of a cessation of service. Cessation of service or switched network capabilities is more applicable to machines that can break down rather than modern, packetized data networks.

Data networks, as distinct from switched communications networks of the past, do not have outages per se; they have delays, instead. In practical terms, however, a delay of more than *t* seconds for a particular service is just as bad as a total unavailability of that service and thus could be considered an outage. The time *t* beyond which a delay becomes operationally unacceptable depends both on the nature of that service *x* and also on the situation *S* in which the network finds itself (e.g., war or peace, or anything in between). A 10-minute delay may be tolerable for routine administrative traffic but intolerable for operational intelligence; similarly, a half-hour delay may be tolerable for logistics traffic during peacetime but intolerable for the same traffic during wartime.

The determination of those services that are mission critical also depends on what else is available or unavailable. A backup communications pathway for a network like GCCS-M is far less mission critical when the primary communications pathway is available, and it is extremely mission critical when the primary pathway has been rendered unavailable.

Accordingly, an outage of service *x* can be defined as the unavailability of that service *x* for more than $t_{x,s}$ seconds, where the two subscripts explicitly highlight that this time depends on what is service *x* and what is situation *s*.

This service availability issue underscores a fundamental problem with prioritizing mission critical services and defining an outage of each such prioritized service: the list is situation dependent. As such, there can never be a single static list but a matrix of lists.

The GCCS-M high-level user cares less about what caused an unavailability of a service and more about withstanding the problem by at least having a viable fallback position as "plan B."

Technical classes of approaches to reducing the unavailability of a service as seen by the end user include:

- The provision of redundancy
- The ability to recover from a temporary unavailability of a service

These approaches can be used in tandem with techniques that reduce the likelihood of an attack penetrating a protected system in the first place.

### *Withstanding Attacks That Cause Specific Damage*

Most cyber attacks, whose intent is to cause a specific damage, rather than a temporary denial of service, have a well-defined goal. The spectrum of possible intended damage is quite broad; it could be to overwrite the boot sector of a hard disk to render it unbootable, alter the contents of flash-ROMs so that the entire affected computer cannot be used, modify data files such as logs or access lists, or modify application software or any other software, and so on.

The extent of the ability of a networked computer system to withstand such attacks cannot be described by any single measure; it will depend on the specific details of the attack, whether the system is already operating in a degraded mode, whether the attack has been detected and defensive measures have been undertaken, and numerous other factors. Furthermore, the quantitative measure of the ability of a system to withstand attacks intended to cause specific damage cannot be defined in the abstract. One can certainly articulate the specific abilities themselves, such as "reconstitute all but the last 10 minutes of data in a disabled hard drive from backup within *m* minutes," but no single overall measure can meaningfully average-up all of the many individual such capabilities of a survivable system.

The ability to withstand an attack results from a combination of defensive measures. These measures can be specific, such as "no software will be allowed to rewrite a flash-ROM or to write on a hard disk's boot sector without manual concurrence by an authorized human operator." They should also include a judicious combination of the many techniques described in this paper, such as sufficient redundancy, the ability to reconstitute affected systems, the ability of individual servers to assume other servers' workloads, pseudorandomizing of the network addresses of nodes to disorient an attacker, and others.

# 5. Potential Future GCCS-M Architecture and Survivability Considerations

This section postulates an architectural direction for GCCS-M and, within that context, considers its survivability requirements and potential technologies that could help meet those requirements with respect to its middleware, clients and servers, network services, and the Cyber Panel. The section concludes with a discussion of an approach to assurance argument development. Also, Appendix E summarizes current research projects that are exploring the technologies discussed here and provides contact information and pointers to additional resources.

The architecture does not represent a simple incremental change to the architecture of the GCCS-M as it is today; it calls for a reorganization of the GCCS-M system. Although the reorganization is significant and it has not been coordinated with the organizations responsible for GCCS-M operations and maintenance, it aligns well with current trends in the technologies on which GCCS-M is based.

A major new factor in the architecture described here is the presence throughout the system of the plan, monitor and assess, and control functions of the Cyber Panel. At the highest level, these functions support a "big board" display of system state and enable control measures that help maintain critical operations across a system of systems when it is under cyber attack. At lower levels, the Cyber Panel's plan, monitor and assess, and control functions will enable better localized reporting and control of system or subsystem behavior.

A second factor that affects the architecture even though it is not part of the implemented system is the need for an assurance argument. The assurance argument provides the logical basis for believing that the system will behave as intended under specified conditions. While such an argument is essential for a specific survivable GCCS-M architecture, this paper only indicates a possible approach to developing one, both because full development of the argument is a major undertaking and because it must be guided by (and in turn influence) the system architecture. Ideally, the assurance argument should be developed together with the system architecture. Therefore, this section includes a discussion on assurance argument development.

## 5.1 Overall Survivable GCCS-M Architectural Considerations

The architecture of current GCCS-M systems was described earlier in Section 4. For the survivable 3GS GCCS-M, a number of significant changes are envisioned as depicted in Figure 5-1.

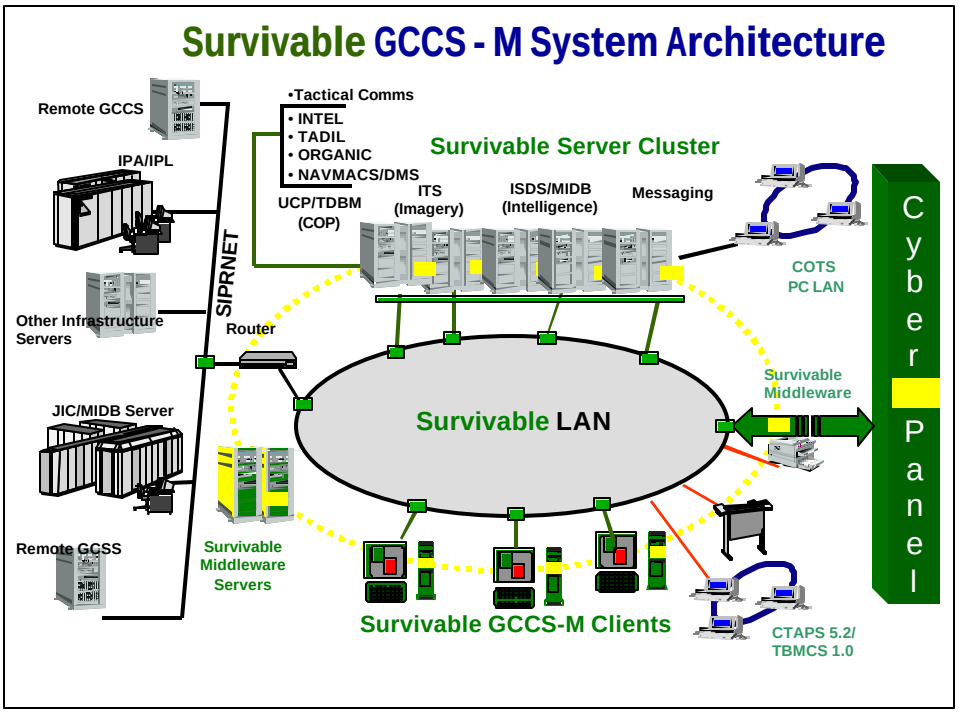


**Figure 5-1. Notional Survivable GCCS-M Architecture**

Specifically,

- Advanced middleware technologies distribute GCCS-M functions across all available servers so that critical functions have no centralized failure points where an attacker could disable all of GCCS-M at one time. The designation of those servers that are performing specific GCCS-M functions at any time is determined dynamically by middleware controls, making it difficult for would-be attackers to identify opportune targets.
- Individual client and server platforms are hardened against a wide variety of attacks. The GCCS-M track database containing mission-essential data, for example, is housed in redundant COTS database systems from different vendors, running on separate platforms behind a proxy server. The proxy server serializes all transactions and scans results from the redundant databases for inconsistencies that might indicate failures or an intrusion. If any of the primary servers come under attack, additional reserve servers on the network will stand ready to provide backup track information.
- Current local area network facilities are upgraded to provide redundant information paths, improved intrusion detection sensors, and intrusion response capabilities.
- The Cyber Panel component is incorporated to collect intrusion sensor data from clients, servers, the network, and applications; diagnose system-wide attack scenarios; and coordinate response actions across all system components to minimize the effects of any attempted attacks.

An underlying philosophy of system survivability is the incorporation of component and interface diversity. While this philosophy may not meet military uniformity, it guards against common mode vulnerabilities that could arise if all system components were identical. It also recognizes that today's new system is tomorrow's legacy system and that wholesale replacement of legacy components is not likely to occur as systems evolve. This philosophy also calls for a strong integration capability so that as systems and components are upgraded, replaced, consolidated, and reconfigured, the entire system not only continues to operate, but provides more and better services. Thus, a range of integration techniques is needed to adapt the new to the old, the old to the new, or both old and new to new configurations. These techniques include:

- Adapters or transformers, such as wrappers, scripts, tailored application programming interfaces (APIs), virtualized interfaces, and negotiated interfaces
- Dynamic protocols, such as variable micro-protocol assemblies
- Translators, both syntactic and semantic
- Directory-enabled services, policy-enabled services, object-services, and knowledge-based services

Adaptive techniques do not negate the use of standards. Standards are essential to meeting enterprise-wide interoperability, affordability, evolutionary, and performance needs. Standards change and evolve over time, however, and when outdated standards no longer meet systems' needs, resources must be applied to move them to newer and more appropriate standards.

The two primary integration components of the 3GS GCCS-M architecture are the middleware and the Cyber Panel. The middleware will house most of the adapters, transformers, translators, and other application integration services. The flexibility that middleware provides for accommodating new components while keeping legacy components in productive service is key to ensuring system evolution along with full operational capabilities.

The Cyber Panel has the responsibility of integrating a diverse collection of sensors, analyzers, and actuators that are deployed throughout the system. Sensors deployed in different layers of the system provide observations on the state of applications, middleware, operating systems, data storage, hosts, and network operations. Analyzers collect, aggregate, fuse, correlate, and interpret functions from local and remote sensors. Adaptation mechanisms or actuators may alter component operating parameters and reconfigure parts of the system in response to suspected intrusions and attacks. In addition to orchestrating these defensive components, the Cyber Panel provides the key human monitor and control interface for managing the survivability of the system as a whole.

## 5.2 Middleware

A key component in modern computing environments is a layer of software, called *middleware*, that provides services for assembling complex distributed information systems. The particular form of middleware proposed for the 3GS GCCS-M, called *enterprise application integration (EAI)* middleware, provides a rich environment in which general-purpose components (often COTS products) can be assembled flexibly into systems to achieve much more specialized mission capabilities [LINT99, KANG00a, KANG00b].

Planning and analysis for deploying EAI middleware starts at the business process level to identify functions and information flows that need to be supported by information systems. System construction tools then provide the middleware glue to configure highly flexible systems

using a variety of distributed, networked hardware and software components. These components may come from different sources and may span a wide range of price, performance, and vulnerability characteristics. EAI middleware connects software components across multiple computing platforms, providing the capability to distribute applications across available computing resources.

The use of EAI middleware is a popular approach to software architecture design for several reasons. It is cost-effective because mission-focused systems can be constructed from general-purpose components that have many potential uses. Component development costs can therefore be amortized over many applications. EAI middleware also supports good software engineering practices. It encourages the use of modular components with clean, well-specified interfaces. This modularity simplifies application programs by encapsulating service implementation details and complexity in separate components. It also allows component capabilities to evolve independently. Components can be upgraded without fear of the show-stopping incompatibilities often encountered in stovepipe systems.

A number of survivability-enhancing middleware capabilities are needed for the 3GS GCCS-M architecture. While not available in all of today's commercial EAI middleware products, these features are consistent with the direction in which most of these products appear to be evolving. These capabilities include, for example:

- Accommodation of legacy components (interface matching and protocol conversion) for adaptation of conventional applications to operate as part of a distributed system
- Support for monitoring application-level protocols to detect component behavior anomalies, with the possibility of blocking corrupted transactions
- Support for automatic recovery from certain types of failures, such as restarting a failed process or reestablishing a failed service on another platform
- Support for orderly termination of selected low-priority services to ensure performance of essential functions while fighting an attack and restoring full system capabilities
- Support for the orderly migration of services from one platform to another

DARPA research projects are currently developing more advanced middleware survivability capabilities. One project, for example, is developing the capability to switch applications dynamically among multiple service providers. An intruder hoping to disrupt applications using this capability by attacking a particular service provider would find those applications had automatically switched to alternate services with no disruption in operation. Successful research in process replication and migration would thwart an attacker's ability to use scripted attacks for breaking into a system because process allocation to servers would no longer be static. Also, investigations in developing techniques that allow applications to change their interfaces to services enables the correction of newly discovered interface vulnerabilities by changing the interface rather than replacing the entire application.

Other DARPA research projects use middleware monitor and control points to detect attacks and participate in containment and recovery operations. Several projects are investigating different techniques to monitor an application's transaction-level protocols and identify constraint violations that may indicate intrusion attempts. These techniques include wrapper technology to add monitors; others embed monitor and control capabilities in middleware.

DARPA research projects investigating these techniques are described in more detail in Appendix E. EAI middleware can play a major role in responding to attacks, limiting the extent of

attack propagation, adapting distributed systems to threat situations, and recovering and reconstituting systems damaged by attacks. Different GCCS-M applications will most likely adopt different combinations of these techniques, as appropriate, depending on their individual assurance needs and the other EAI middleware services they employ.

One potential disadvantage of the middleware approach is that, because of the IA and survivability advantages it can provide, middleware interfaces and services will become primary targets for cyber attacks. For example, many of today's commercial EAI middleware products employ centralized application management engines and brokers that register system-wide services. If an attacker can control or disable such an essential service, the entire enterprise would be affected. Any currently centralized middleware services will therefore need to be partitioned, replicated, distributed, dynamically configured, and otherwise protected to ensure continued operation if any one instance comes under attack.

Another disadvantage of middleware is that it introduces several additional layers of software that may raise performance concerns. Military computing systems are often overloaded, partly because of protracted hardware and software upgrade cycles and partly because it is easy to find more functions that need to be automated. Middleware workload estimates need to be included in system performance and sizing estimates. The business community has found that the additional hardware necessary to handle increased EAI middleware workloads is easily justified based on the integration capabilities and configuration flexibility it provides.

An example situation in which EAI middleware would provide a significant advantage for the 3GS GCCS-M is in the interfacing of applications and client/server software components with the Cyber Panel functions. This interfacing can be achieved by incorporating Cyber Panel functions and information flows in the EAI business process model for the system. Intrusion monitoring data, for example, will then be routed automatically to the Cyber Panel by the EAI system construction tools. Commands from the Cyber Panel directing specific response actions to stem the propagation of attacks will be similarly routed to designated EAI middleware control points. Without EAI middleware, integrating the Cyber Panel into all GCCS-M functions would be exceedingly difficult, if not impossible.

### 5.3 Clients/Servers

The client/server model, which divides software functions between relatively numerous, low-cost clients and relatively few, more powerful and expensive servers, is the basis for the architecture of most of today's distributed network-centric information systems. A client process typically requests information system resources from a server. A server, conversely, is a process that manages a system resource for use by clients when requested.

Also, a client may request a resource from a server and that server will, in turn, request the resource from another server, and so on. In this situation, only the originator of the request acts solely as the client and the final server that satisfies the request acts solely as the server. All other intermediate processes act both as clients and servers.

Thus, the division of system components between client and server is sometimes indistinct. This dual functionality of client and server occasionally permits certain survivability techniques to be applied bilaterally.

Although servers may seem to provide a higher value target for an attacker, clients too may store sensitive information, such as private keys. Clients can provide platforms from which other

systems may be attacked, either by allowing an attacker to masquerade as a legitimate user or by providing a place where the attacker can plant malicious code to be executed on command, as in recent distributed DoS attacks. Hence, both clients and servers require attention. In systems with thick clients, although the scale of clients and servers is different, often the basic hardware and operating system architectures are sufficiently similar that some common techniques can be used to secure them.

An intrusion-tolerant information system must also include survivable means of storing critical data. Methods to use redundancy in various forms to protect data have long been studied. An early, innovative approach to storing data securely, even when some hosts have been compromised, was studied at LAAS-Toulouse in the early 1990s. Researchers combined fragmentation and scattering of data with the use of threshold encryption. To compromise any particular file of data, several hosts would have to be compromised independently [FABR93].

Current research is revisiting this approach, known as Fragmentation, Redundancy, and Scattering (FRS), with a goal of quantifying the cost and performance tradeoffs involved in protecting data in this way [WYLI00]. Availability, as well as confidentiality, can be enhanced with this technique, because any number of surviving servers exceeding the threshold that have the fragments of a required file can reconstruct the critical data. Finally, integrity may be enhanced if surviving servers can allow a comparison of the output data of the threshold algorithm.

Another general technique providing intrusion-tolerant service is to combine redundancy with a voting mechanism, as is common in fault-tolerant system designs. The goal here is to develop a system that can survive dynamic faults resulting from an intrusion. Survivability with efficiency may be attempted through dynamic reconfiguration strategies, in which the level of redundancy and checking invoked is linked to an evaluation of the current level threat [SCAL01].

Specifically, the architecture of a survivable server capability could consist of replicated proxy servers, ballot monitors, and acceptance monitors. The proxy servers would maintain the state of client requests, include an IDS, control load balancing, and provide shared memory, among other features. The acceptance monitors would verify the plausibility of the results as well as perform trusted-state monitoring of the COTS servers. Finally, the ballot monitors would perform voting on output results and control any dynamic reconfiguration required because of intrusion induced faults.

Hardware mechanisms can also serve a role in creating intrusion-tolerant service as well. A hardware device that provides a filtering capability between a (potentially compromised) host and a network, and which is not under the control of the host, cannot be subverted by that host. Such a device, placed at the host/network interface, could not only filter packets but could provide IPSEC support, audit capabilities, and more. These interface cards, in turn, should be managed so that they provide the means for detecting intrusions, logging and reporting anomalies, and allowing response, recovery, and restoration in the face of attacks. However, the mechanisms used to control the hardware would need to be carefully designed to prevent their subversion as well. This approach to providing intrusion-tolerant service is also currently under investigation [MARK01].

Many servers are employed to store large amounts of data in an organized manner in the form of a DBMS. One way to provide intrusion tolerance at the DBMS level is to allow suspicious user masking and isolation [JAJO98]. For example, if a user is suspected to be an intruder, the user will be isolated at the DBMS level. Then any DBMS access by the suspected user will be rerouted to a virtual copy of the DBMS until a determination can be made whether the user is legitimate. If the

user in question is found to be legitimate, the virtual DBMS can then be merged with the original DBMS. Otherwise, the virtual DBMS data can be discarded or retained as evidence.

Monitors provide useful survival functionality and, in this context, can provide two types of functions: data monitors and execution monitors. The overall purpose of a monitor is to ensure that the monitored data or operations conform to an explicit security policy. An approach to data monitoring is to wrap data with integrity-assuring marks, record its processing history, and provide the ability to reconstruct it from this history if intruders corrupt it [TALL01]. Execution monitors, in contrast, normally provide a controlled interface between client software/target server and the operating system. For example, an execution monitor can be a library that encapsulates an API [HOLL01]. This encapsulating (known as *sandboxing*) of untrusted program execution is especially important with the growing use of mobile code. A wide variety of approaches to execution and data monitoring are the subject of current research, including investigations of widely deployed monitors that can be managed as configurations and security policies change [BALZ00, ERLI00a, ERLI00b, FRAS00, MCGR00]

The discussion now turns to the topic of determining how the servers of a future GCCS-M could be configured to improve system survivability. Figure 5-2 provides a notional view of a survivable server architecture that provides redundancy, diversity, execution monitoring, and other methods to enhance survivability.

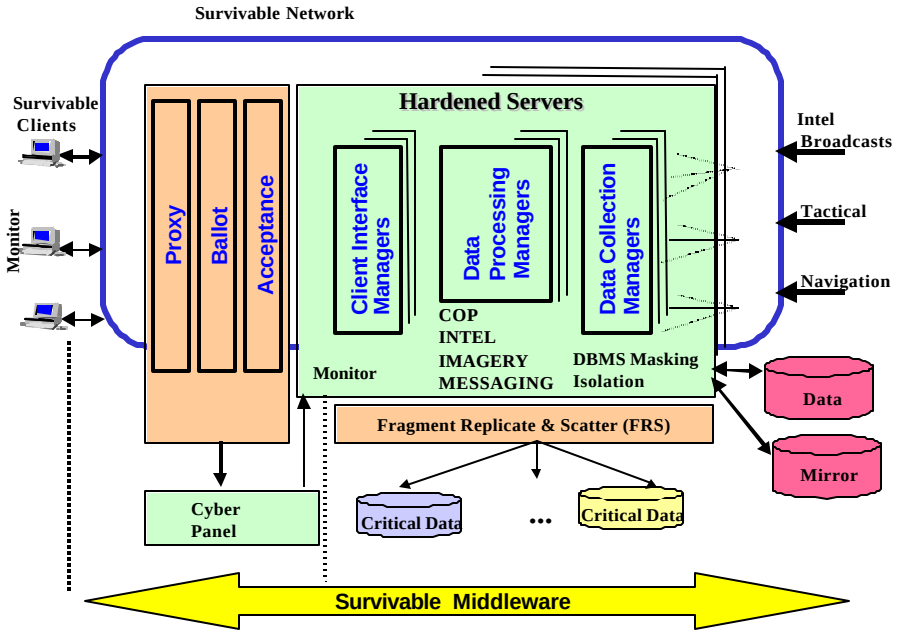


**Figure 5-2. Notional Survivable Server Architecture**

In the survivable GCCS-M architecture, data feeds from intelligence, tactical, and navigation sources are provided to GCCS-M servers via a survivable network. Each server consists of three primary components: Data Collection Managers (DCMs), Data Processing Managers (DPMs), and Client Interface Managers (CIMs). The primary responsibilities of the DCM are to gather incoming raw data from the survivable network and provide an input buffer for the DPM. The DPM is also responsible for storing the incoming data after performing general-purpose data processing, such as parsing and indexing of the data. Finally, the CIM is responsible for providing a session for each requesting client. Figure 5-2 shows a notional survivable server architecture.

Redundancy and diversity are common techniques to combat intentional and malicious faults [DESW98, BAIN00]. To this end, replicating each server using different platforms is recommended if it is possible. Such replication of information processing functionality ensures that no server becomes a single point of failure. Since GCCS-M encompasses many applications running on different servers, it is advisable to run multiple dissimilar GCCS-M applications on each server. This technique helps reduce the cost of server redundancy.

The employment of a balloting/voting scheme to improve the survivability of the servers is advisable. Survivability is accomplished by selecting the majority vote of the output from multiple redundant servers. The rationale for choosing the majority vote of the output is that it has the highest probability of being correct.

In the event that the balloting/voting system detects a recurring anomaly in one of the voting server members, the Cyber Panel is notified so that corrective action can be initiated. In addition, a DBMS masking and isolation scheme is triggered as necessary. The server that continues to provide incorrect output can be labeled as suspicious and isolated at the DBMS level until a positive determination is made about the status of that server. Every DBMS access by the suspect server is virtualized until a decision is reached about its status and appropriate corrective action is taken.

One of the major principles of survivability is containment of potentially suspicious or malicious programs and data. To improve survivability, it is useful to employ execution and data monitors throughout the architecture, because monitors help ensure the integrity, confidentiality, and availability of the system. For example, if an attacker succeeds in bypassing the preventive security mechanisms of the system, data integrity checks will similarly ensure that the damage caused by the intruder is kept to a minimum. Furthermore, the execution monitors will ensure that the intruder in such an example is not able to violate the organization's security policy.

A DBMS, or similar data storage mechanism, is included in the survivable GCCS-M architecture to provide persistent storage for servers. FRS technology could be used to store small amounts of critical data. Using FRS for the entire DBMS might not be advisable because of the high cost of providing multiple data repositories for fragmentation. Instead, a DBMS-mirroring scheme could be used for storing noncritical data.

While much of the above survivability techniques apply to servers, the same techniques could also be employed with clients if additional survivability is required. Employment of these survivability techniques is especially important with mobile code, which is becoming more commonly used. If such precautions were not in effect, malicious programs could compromise a client and then other hosts on the network. Clients will also be the end recipients of both data and mobile code.

To increase the survivability of the client, data and execution monitors should be employed (as in [NECU97, MCGR00, MYER99]). Finally, if survivable middleware is present as described above, all client/server communications should be performed through the middleware (see Figure 5-3).

Adding survivability features to clients and servers introduces tradeoffs. Survivability features often require compromise in areas such as storage requirements, system performance, and network bandwidth requirements. For example, FRS consumes additional processing and network bandwidth for assembling data fragments up to the required threshold. Similar tradeoffs can be found in other survivability technologies.

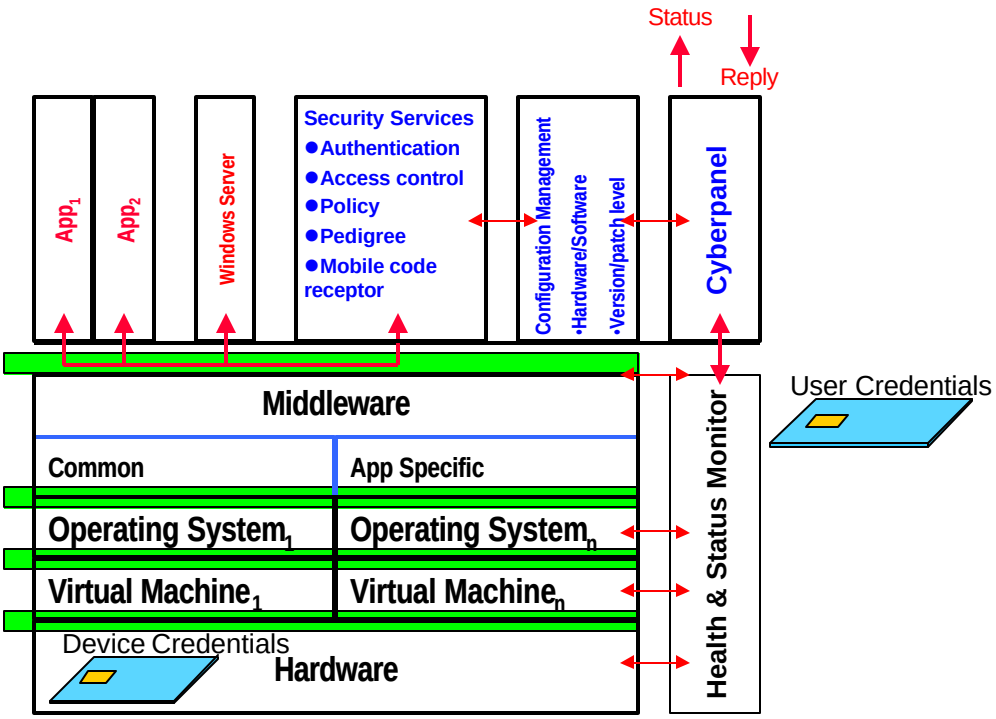


**Figure 5-3. Notional Survivable Client Architecture**

## 5.4 Network Services

### *Introduction to Network Services*

Network services will underlie a survivable GCCS-M and GIG systems generally. Survivability of the network infrastructure is essential for keeping the system operational and delivering information throughout the distributed system. To condition networking for survivability, emphasis should be placed on the architectural elements used in designing an enterprise network and the operational components detailing the technologies that combine to enable, protect, and adapt networked communications in the face of hostile activity.

### *Architectural Elements of Survivable Networks*

A survivable network should be developed with a systems approach. A resilient network architecture needs to encompass elements of connectivity, security, network management, and directory services. Connectivity defines the capabilities of an enterprise and incorporates communication links and redundancy solutions. Network management and security are layered on top of this design to provide the ability to protect against threats, control configuration, address equipment failures, and facilitate growth and change. Also, directory services enhance the usability of the enterprise for authorized users while adding efficiencies to the network management process.

A network's connectivity is defined by its ability to provide services to all of its authorized users. Basic connectivity is straightforward for nodes that are local to each other and that have the resources to supply sufficient bandwidth. Enterprise networks, spread over the globe and potentially reaching into space, pose greater challenges. Nodes that may fail and be reconstituted, along with communications links of varying latency and fluctuating bandwidth, make the behavior and management of large-scale networks complex. In addition, a survivable GCCS-M system must be able to provide access to critical services, or as much service as is possible, when portions of the enterprise are unavailable or impaired.

Essential to operating under degraded conditions is the ability to segregate traffic by priority or connectivity requirements (i.e., Quality of Service [QoS]). Redundancies are built into the architecture to ensure that communications paths are still available when portions of the enterprise fail or are disabled. In addition, resilient protocols and dynamic overlay networks are essential in providing reliable service and improved fault tolerance under adverse conditions. These technologies enable systems to use the redundancy of networks transparently and enhance the behavior of systems under adverse conditions.

Figure 5-4 shows an architectural model of survivable networks.

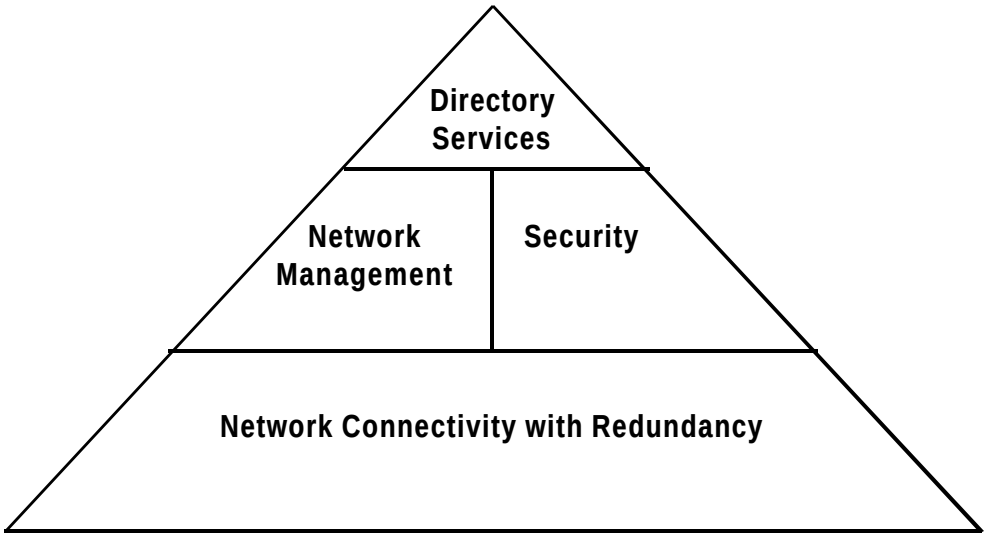


**Figure 5-4. Architectural Model of Survivable Networks**

Network security has two control mechanisms that broadly support survivability. The first is the ability to assert positive control over the resources of authorized network users. The second is the ability to deal with unauthorized activity, known as the *protect-detect-react* paradigm. In both control mechanisms, only the intended users of a system will have access to it. Users will need to be authenticated and have their networking privileges restricted to only those resources they require. Unauthorized users should not have access to network resources, and more importantly, they should not have the ability to deny the use of those resources by the authorized users. A well-engineered 2GS network security system should provide these control mechanisms.

An advanced security system would have cognizance of all activity on the network and also possess the intelligence to curtail any traffic that does not belong. Ideally, the integrated security functions will provide active response to an attack and initiate reasonable recovery processes. Security is also a key factor when designing a system to operate with a network infrastructure that is not wholly owned or not under complete administrative control.

This situation for the network infrastructure is likely to be the case with GCCS-M. Either the physical layer, which is the basic network fabric, is not owned in its entirety or the network in general is shared with related DoD organizations or other government agencies. Application security and virtual private networking then become important architectural considerations. In the shared case, the threat of misuse and unauthorized activity, is much higher and is therefore a greater priority when designing the protect-detect-react capabilities of the enterprise. The search for illicit activity is typically performed by network- or host-based IDSs. IDSs may also be used to look for insider threats.

A network of any complexity will require network management. This management will include not only the maintenance of existing nodes but also the configuration of new devices as they are added to the network. It will include the function of upgrading existing equipment, software, and services as the network evolves. Daily functions will include operating and maintaining the enterprise.

More granular management actively monitors bandwidth usage and QoS. This role can be expanded to include the operation and maintenance of user nodes, server nodes, and applications. Technical advances in controlling the flow of traffic will be used to preserve the QoS and to assist in preventing DoS attacks against the QoS provided by the network infrastructure.

Network management often has security implications for the survivable architecture. The organization responsible for managing the connectivity and usage of a network often performs the security monitoring as well. Information on monitoring and information detected by active response and recovery components within the network should feed into the Cyber Panel.

Directory services address usability features of a network. In its most basic form on an IP network, directory services provide the DNS. Additional functions can include dynamic addressing, service registration, user authentication, privilege and credentials management, and storage of configuration information. Security-specific features might include the key management features for a PKI and housing configuration parameters for networking nodes and security devices.

New technologies aimed at developing fault-tolerant trust will help maintain the basic functions of directory services and other distributed name services under adverse conditions. For example, policy information and policy mechanism information are highly important and can be effectively managed through the directory. Gathering these functions as part of the directory implements directory-enabled networks, a concept recently advocated by the Distributed Management Task Force (DMTF) and leading vendors [DIST01, CISC01]. Directory services link the underlying network with the domain of applications to increase usability. The aim is to facilitate the interoperability of distributed applications, management tools, and network elements.

In summary, various aspects of security, network management, and directory services must feed into the Cyber Panel as a way of coordinating information about cyber attacks on the system, gaining situation awareness, and planning protection, response, and recovery activities. The Cyber Panel must also exercise some control over the network in dealing with attacks.

### *Operational Components of Survivable Networks*

Once a survivable network infrastructure has been put in place and the usability features enabled, the ongoing challenge in implementing operational control is the incorporation of two disciplines: security and network management. These two disciplines are tightly interrelated and can be viewed as the operational components of survivable networks. Figure 5-5 depicts their joint operation. The main domains of the figure should be considered in the following order. First, the Configuration Control domain covers the initial setup of the network, including its services and their maintenance. Second, the Network Defenses domain covers services guarding the network against threats and attacks. Third, the Managing Degraded Service domain deals with situations occurring when network services have been compromised or lost. This service degradation could be the loss of links, reduced bandwidth, or noise generated by hostile activity. In a steady state, consider the figure in a circular fashion, with configuration changes implemented in response to

growth, modifications, and hostile activity. The hub is the Network Management and Security function, with efficient management and configuration control provided by Directory-Enabled Networking (DEN) technology. Expanded discussion on these parts of the figure follows.

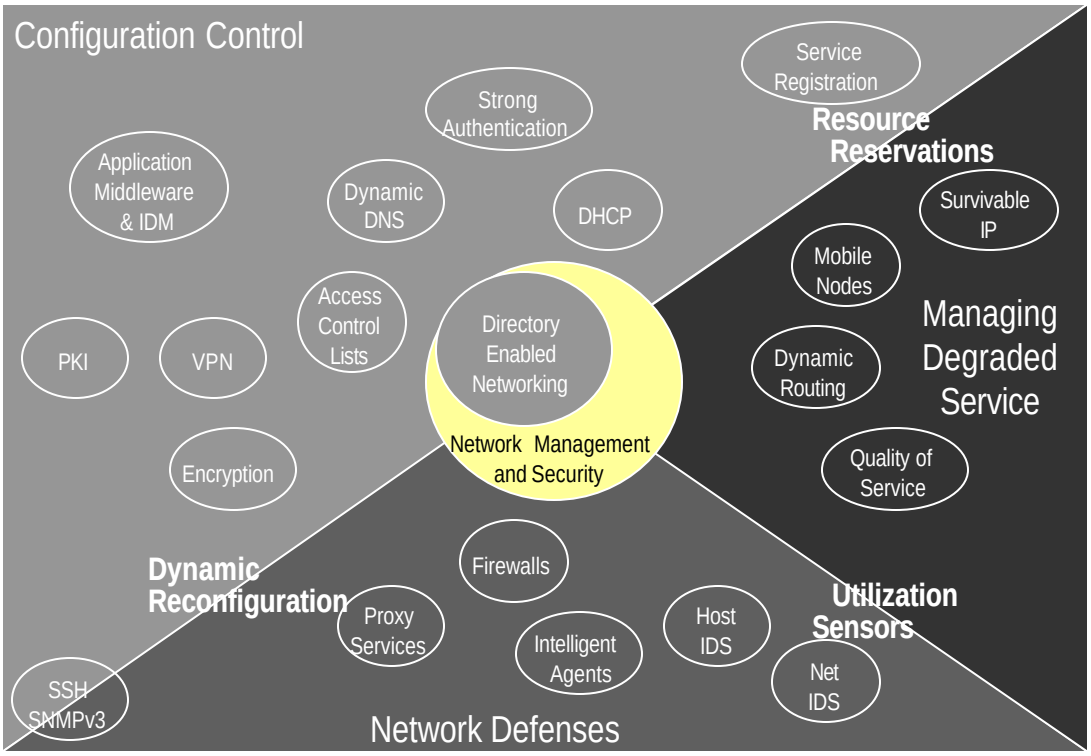


**Figure 5-5. Operational Components of Survivable GCCS-M Networks**

The Configuration Control domain strives to maintain the integrity of the enterprise and is essential to the DEN. It controls access to resources by authenticating users, registering services, and administering the address assignments of both. These are core functions provided by a DEN. Security services are part of this core, with user authentication being a primary service. They also come into play when traffic is protected by encryption or if either VPNs or PKI are used. Broad policy needs to be managed in this domain, so that it can govern the operations of the network. Any management of networking nodes (e.g., routers), either configuration modifications or statistical polling, should be performed securely (i.e., authenticated and authorized).

The Network Defenses domain presupposes that hostile forces wish to misuse or usurp functions of the network. Hostile acts may steal information, inject false information, monitor activity, use resources in an unauthorized way, or impair network operations (deny service). As stated above, traditional network security strives to protect, detect, and react. Protection is built into the boundary of the network infrastructure in the form of firewalls, proxy servers, and secure routers. Internal to the network infrastructure are authentication and access controls. Detection is

addressed by the implementation of IDSs: the IDSs are network-based to look for aberrant behavior, particularly on the network boundaries, and host-based to monitor critical nodes.

Detection can also be accomplished by monitoring the system logs of network devices and servers. Reaction is the least mature aspect of network defense operations and has internal (passive) and external (aggressive) components. The internal components are depicted in Figure 5-5. Within one's own network, some choices are straightforward and often include immediate connection termination and patching of holes, or a period of inclusive monitoring to uncover the perpetrator's methods and assess the information compromise suffered by the enterprise. The detection of hostile activity can also be used to trigger QoS activities and to activate policies prepositioned to respond to a threat. External reactions may include both efforts to notify other systems of an attack and attempting to strike back at the attacker. The latter kind of reaction is offensive in nature, technically and legally controversial, and will not be discussed further.

Current research is moving the three functions (protect, detect, and react) into all aspects of the network infrastructure. This transformation creates the opportunity for the network infrastructure to participate actively in assuring that it can continue to operate and provides a way for network monitoring and management systems to gather additional information. The change introduced by these technologies makes the network infrastructure more fault-tolerant and resilient to attacks.

The Managing Degraded Service domain handles degraded operating conditions. Warfare, both conventional and cyber, will cause attrition within the enterprise. Nodes will be lost, bandwidth will be reduced, and networking assets will be reconfigured. Despite this degradation, the mission to maintain communications will remain. Networks must employ protocols capable of withstanding often extreme variations in bandwidth availability, latency, and ingress points.

Routing protocols must be able to adjust to the disappearing and reappearing of subnetworks and nodes. Resilient routing protocols and support from within the network will allow routers to select paths based upon the QoS needed by the communications streams. Enhancements to the routing protocols are reducing the time needed to select and converge on a new routing path when one or more network components fail or misbehave.

Additionally, fault-tolerant network technologies are being developed to detect attacks and protect the QoS of legitimate traffic. The enterprise must additionally be sensitive to the fact that some communications are more critical than others, so technologies to support traffic prioritization and QoS become crucial.

Also significant in the operational view are the boundaries that separate the three survivable network domains. These boundaries highlight the requirements for networking technologies to interoperate with other technologies to ensure a survivable enterprise. The boundary conditions are those enabling technologies that allow for automation between the relationships in the protect-detect-react sequence of events.

The border between the Configuration Control and Managing Degraded Service domains is managed by the Resource Reservations function. To enable an adaptable enterprise, network resources should be reserved in an active or dynamic way. The availability of these resources, therefore, becomes resilient to change (the ability of users to find and access them remains unchanged) and makes network operations more survivable in the face of a threat. Also key are the management of QoS functionality and the ability to reconfigure networking equipment to adapt to changing network conditions. Networking equipment and protocols need to be engineered with

QoS features in mind. In addition, QoS preservation techniques need the ability to respond to commands from the directory to implement these features.

The border between the Configuration Control and Network Defenses domains is managed by the Dynamic Reconfiguration function. Being able to reconfigure a network's defense mechanisms dynamically against a sophisticated adversary offers an agile defense, that is, a way potentially to outwit the opponent or at least deny access to the vulnerability being exploited. Such dynamic reconfiguration must be well planned, so that the network remains available to the legitimate users.

Additionally, the survivable network infrastructure has much data to offer to the network defense effort; system logs, error messages, and usage statistics are among the types of useful data available. Active network defense mechanisms will have detailed information necessary to isolate attacks properly, such as the source addresses and paths of distributed DoS attacks.

For illustration purposes, consider the function of a dynamic routing protocol. It exists to handle degraded service. A dynamic routing protocol adapts connectivity to conditions in which networking links fail or have reduced bandwidth. Superior routing protocols become aware of degraded conditions, perform necessary adjustments in a minimal time, and can scale as the network grows.

Simply implementing the most capable routing protocol does not guarantee the most highly survivable network. Relationships exist between the Configuration Control and Network Defenses domains. Developing a routing protocol with a configuration control interface can make the system a superior architectural choice. A dynamic routing protocol also becomes much more survivable when it integrates an authentication and authorization service. Authentication and authorization services improve security by making the protocols less susceptible to attack. In addition, authentication and authorization services are beneficial in enforcing resource allocations and protecting the QoS within the network. Another facet of configuration control would be engineering the availability of configuration information, in a robust way, from the network directory service. The Cyber Panel should be involved in this control loop.

Dynamic network services can also provide agility and support security. Camouflaging techniques can be used to disguise the identity of systems or dynamically alter the fingerprint of systems. The use of these techniques impedes attacks by obscuring the true nature of the system and its potential vulnerabilities. Likewise, system addresses can be assigned dynamically with the Dynamic Host Configuration Protocol (DHCP). However, for deception purposes further dynamic addressing could be introduced to throw off the attacker. Dynamic DNS could pair node names with network addresses in a changeable way. This service could be coupled with DNSSEC to make it more resilient to attacks.

The Utilization Sensors function controls the boundary separating the Network Defenses and Managing Degraded Service domains. Sensors are essential to both operations. Utilization Sensors define the integration parameters for network defenses and the indicators for managing degraded services. Networking defense includes the active monitoring of the network. A network attack must be detected for it to be countered, just as shrinking bandwidth must be detected to activate QoS measures.

Utilization Sensors can also detect problems and trigger changes in the communications routing and behavior of priority queues that manage bandwidth allocation. Higher order monitoring may be able to detect subtle attacks against the systems or detect distributed attacks at

or near the source of the attacks. When problems are detected by the monitoring components of the networks, any number of security measures can be triggered into action. Hence, research on survivable sensor systems should include the identification of requirements from multiple areas.

***Considerations for Developing Survivable GCCS-M Networks***

Considerations for developing an architecture for survivable network services for a GIG system such as GCCS-M are based on the concept of a DEN. DEN facilitates the flow of control information between the applications on the end systems and the network nodes, so that the network infrastructure can meet the critical communications needs of the distributed system, even when under cyber attack. Protections may be grouped according to the following two types: network infrastructure reliability and IA. DEN offers a way to unite these two types of protection. The directory provides information for keeping the network both reliable and secure.

A survivable network infrastructure should provide robust network services. Specific services that should be considered in developing such a robust network include the following:

- **Dynamic Routing.** Routing protocols must be designed to adjust quickly to the disappearing and reappearing of network segments and individual nodes. Dynamic routing must offer robust means to adjust network traffic flow and overall routing patterns to deal with varying conditions, including conditions arising from cyber attacks. This function can be developed in several directions, since a number of new methods are now available to vary routing according to changing conditions.
- **Robust IP Service.** Robust IP service offers ways to prioritize traffic and adjust to lower levels of network service. It should be possible to handle extended latency and faults gracefully. Such an IP service should take account of provisioning down to the physical layer of the system.
- **Configured QoS.** Being able to configure quality of service in the network is highly desirable. Urgent traffic needs to be expedited through the network. In general, traffic needs to be segregated according to requirements of priority and connectivity. This segmentation should be determinable in advance through the application of appropriate QoS parameters. Also, networks must be protected from attacks against QoS. As a result, techniques are needed to detect such attacks and manage their impact on the service commitments. Thus, networking equipment and protocols need to be engineered with QoS features in mind.
- **Configured Addressing and Camouflaging.** Addresses must be assigned and managed in a consistent way. In addition, they may be dynamically configurable as in DHCP. This structure would provide flexibility to the resources available. Some forms of dynamic addressing can be used for deception purposes as a defense against an attacker. An attacker needs to know the location of the network resources, that is, the targets of attack. By using dynamically configurable addresses, an attacker would experience more difficulty in getting this information. Likewise, camouflaging techniques are needed to hide systems from attackers and also provide confusing information about these systems to attackers.
- **Configured Filters.** Filters and proxies at the protocol level on boundary devices, end systems, and key intermediate points within the system should control network access in a well-accepted manner. They could help manage access as a necessary part of the

protection of the network. Filters should be easily managed and responsive to information provided by sensors in the system while protecting themselves from detection, attack, and mismanagement.

- **Resource Location and Reservation.** Resource location is a significant part of DEN. The network nodes, through directory services, should become informed about places where resources are available and can be used. Being able to reserve some resources should be part of ensuring a more survivable network. This service should also enable dynamic addressing and mobile networking.
- **Resource Provisioning, Upgrades, and Maintenance.** Positive configuration control must be maintained for all network resources. This service should provide a current and uniform security posture while facilitating modifications.
- **Switch Partitioning.** Switch partitioning offers a way of controlling the end systems by being able to switch these resources on or off at the network nodes near the end systems, denying physical access to unauthorized usage. This service should be included in a survivable network.
- **Composable, Resilient Modules.** Components of the system should protect themselves from attacks, and reconstitute and recover once they have been attacked whenever possible. Resilient components must have well-defined interfaces that enable the composition of modules into high-assurance trusted systems.

IA ensures that network services are used as intended. Incorporation of the following IA network services should be considered for promoting system survivability:

- **User and Service Registration.** Both users who have access to and services available on the network need to be registered to manage each in a systematic and protected way.
- **Authentication.** Authorized users always need to be authenticated to the network system so that they can be distinguished from outsiders and potential attackers. This is a standard way to keep unauthorized users off the system.
- **Privilege Management.** The networking privileges of users will vary according to what they really need to use. These privileges need to be managed throughout the system. Fine-grained privileges can be used to reduce the level of trust between systems and to help control the propagation of attacks.
- **Access Control and Access Control Lists.** Access controls need to be placed on servers and other resources to control their use. When appropriate, access control lists should interface with sensors and IDSs to support dynamic modification of access controls based upon the current attack situation.
- **Intrusion Detection Systems.** IDSs must be placed at the host and network levels. If the system allows encryption and VPN connections, ways still must be found to perform intrusion detection at appropriate points in the network. Advanced methods of intrusion detection are needed to prevent attacks from slipping under the thresholds of detection. Intrusion detection methods must also minimize the number of false alarms.
- **System Log Monitoring and Auditing.** Sound log monitoring and auditing must be conducted throughout the system. In particular, monitoring and auditing must be conducted at all access points. Other auditing functions need to be in place.

- **Integrity Checking of System Resources.** System resources, such as programs, configuration files, and system or software updates, need to be thoroughly checked through devices like cryptographic checksums (e.g., "tripwires") to detect integrity problems in a timely fashion. When appropriate, system resources should be self-protecting.
- **Intelligent Agents.** In the far-term research arena, intelligent agents may be used to check on the status of the network while enforcing policies and configuration control. Intelligent agents are instances of mobile code that enforce network policies.

The entire array of protections needs to be well managed. Part of good management will be integrating all of these network services. Integration is the key to a sound, survivable network.

### 5.5 Cyber Panel

The Cyber Panel concept aims to improve the observability and controllability of operational computer systems and networks in the context of cyber attacks. A "big board" will display the status of large-scale DoD systems and networks with respect to cyber attacks. Staff who understand the network infrastructure will help commanders relate this status to operational entities, and associated tools will help them develop alternative courses of action and assess their potential effects. Access to relevant communication and control mechanisms will support initiation of response actions.

The Cyber Panel concept also applies at lower, more detailed levels of local system operation. Because this concept is new, this section describes the envisioned functions of a survivable Cyber Panel and a possible structure for their implementation. Appendices B and C provide further details on Cyber Panel components/functions and interface requirements, respectively. The implementation of these functions, particularly the aspects of sensing and control, will draw heavily on the survivable middleware, client/server, and network services technologies already described.

Today, system administrators and network security analysts must sift through an unmanageable barrage of low-level, apparently unrelated alerts, attempting to find those of special concern. They have few tools to aid in analyzing, intervening, or responding to the consequences of serious attacks. Operational commanders need to understand the operation and attack state of information systems and networks upon which they depend, at the scale of an entire theater and in terms relevant to operations unfolding in the kinetic battlespace.

The Cyber Panel's observation and control functions will be central to DoD's emerging concept of Network Operations. NetOps encompasses the tighter coupling and integration of telecommunications network management, information dissemination management, and IA. It captures the commander's need to visualize and influence all information flows over networks in an area of responsibility. Cyber Panel technology will focus on the display of information and the exercise of control relative to cyber attacks, which will be critical capabilities for NetOps across the GIG.

#### *Top-Level Functions*

The Cyber Panel will provide an integrated, survivable capability to plan for current/future operations, monitor system health, assess attack state, and control system security postures at local to theater scales. The Cyber Panel must perform the following top-level functions:

- Provide a big board view of system health and attack state in multinetwork areas of responsibility.
- Incorporate multilayered health-sensing and attack detection capabilities, including monitoring for misuse, anomaly, and time/value domain variances at host, network services, data services, middleware, and application levels.
- Provide correlation across layers and topology.
- Enable identification of large-scale attacks.
- Enable attack tracking and impact assessment.
- Allow security posture/change analysis (a priori planning and event response):
  - Provide capabilities for dynamically invoking security and survivability mechanisms and dynamically changing configurations.
  - Allow commanders to influence and monitor response actions.
  - Provide collective control of large-scale, coordinated responses to cyber attack.
  - Enable observation of defense and recovery performance.
- Retain a minimum set of situational awareness, and command and control capabilities when placed under extreme duress.

A Cyber Panel capability that performs these critical functions will naturally become a target of attacks, so the Cyber Panel itself, as noted above, must be survivable. That is, it must retain a minimum set of sensor, display, and control capabilities when it, as well as the systems it monitors and controls (referred to here as its *subject systems*, or *subjects*), is under cyber attack. To meet this requirement, the Cyber Panel is likely to employ the same technologies used to enhance the survivability of other systems. Because of its critical role in sensing and responding to attacks, however, its communications may require special protection.

A wary attacker pays attention to what the defender is doing: witness Robert Hanssen's recently publicized efforts to monitor the Federal Bureau of Investigation's own systems to detect whether he was under surveillance [JOHN01]. A cyber attacker who successfully penetrates some portion of a system will also check to see whether his/her efforts have been detected and, if communications mechanisms in that system are under his/her control, may make it his/her first priority to stop any warning message from being sent.

Even without such conscious intent to avoid monitoring and control, some attacks will jam or destroy those mechanisms as a side effect of the overall attack. In a distributed DoS attack, for example, the control path may be the same one being jammed by the attack. Consequently, providing a means for sensors and effectors to communicate with a Cyber Panel that uses paths other than the normal ones must be a strong consideration in survivable system design. Such alternative paths might use separate wireless links, for example, in the way that some copiers embed cell phones to communicate back to a service facility when the machine senses repairs are required.

### *Conceptual Architecture*

#### Concept

The Cyber Panel must interact with other survivable GCCS-M system components within and across each architectural layer to assert cyber situational awareness and command and control over

the cyber assets that enable military operations. This means it will operate across systems, across layers, and potentially across geographic domains. Figure 5-6 highlights the major architectural components of a Cyber Panel. Appendix B describes potential types of sensors, controller-actuators, analysis engines, models, and human-computer interfaces for situational awareness and system control.

Cyber Panel software will incorporate models of system interdependencies that will permit its operators and observers to understand the potential impact and damage of an attack at the level of operationally relevant computerized processes. That is, the Cyber Panel will not attempt to determine if an overall battle plan will fail as the consequence of a cyber attack, but it will identify critical software components (for example, software to generate an Air Tasking Order) that may fail or be degraded. The potential impact to mission survivability from varying degrees of information system service degradation, or other availability constraints due to resource allocation and shared processing, can then be presented and understood before using the Cyber Panel's control interface to implement system security management decisions.

Cyber Panel survivability will be achieved through the same techniques used to enhance the survivability of other components and systems, including distribution, redundancy, variation, randomization, active monitoring, threshold-based triage, situation-based adaptation, deception, concealment, and rapid service restoration.

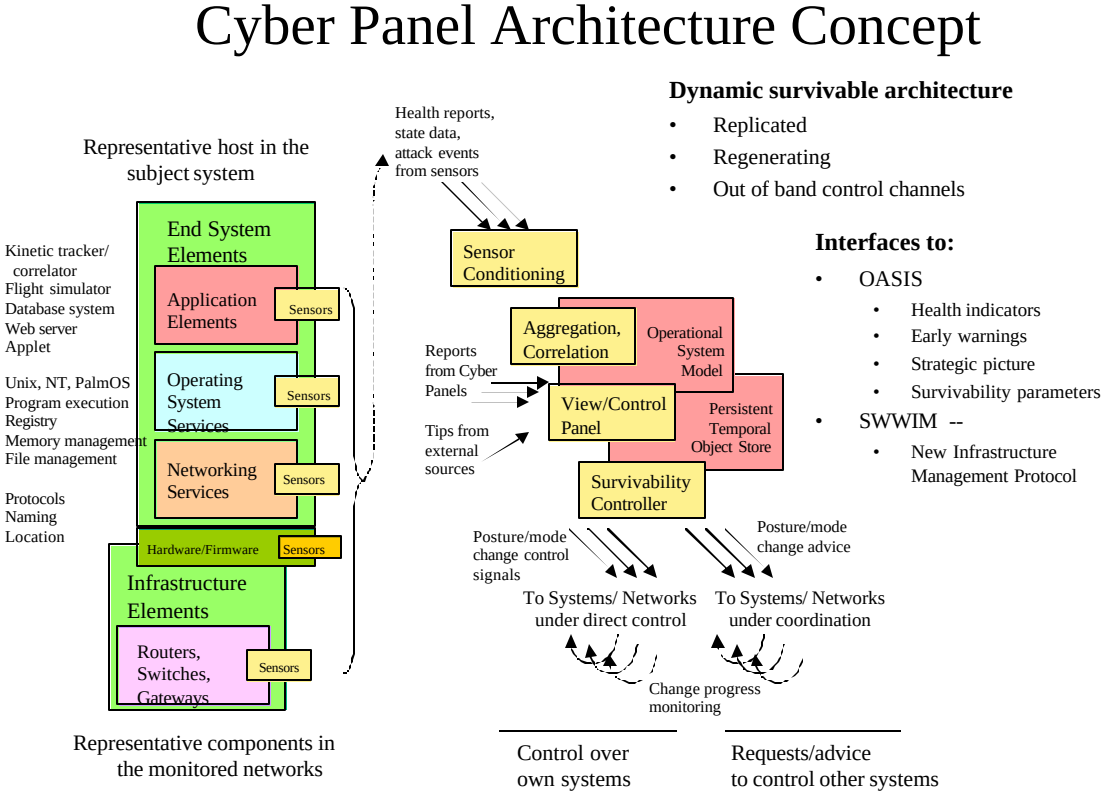


**Figure 5-6. Cyber Panel Architecture Concept**

**Architectural Interfaces**

The Cyber Panel must support a range of interface requirements within and across all components of a survivable GCCS-M system, largely achieved through middleware capabilities. The goal is to maximize the situational awareness and dynamic control capabilities of the Cyber Panel. Appendix C details more specific Cyber Panel interface requirements.

Ideally, these interfaces would be "plug and play." Publish and subscribe interface paradigms would be encouraged where possible. The degree of integration that the Cyber Panel can achieve with a particular legacy system will be governed by tradeoffs made in light of technology insertion points each legacy system offers, including its performance requirements and resource constraints.

***Operational Model***

The Cyber Panel will monitor system health across all architectural layers. It will collect information from numerous sensors and make it available for display. It will also process and correlate this information to support higher level conclusions about the possible existence of a coordinated strategic attack, its scope, severity, and potential mission-relevant impact. Cyber Panel operations will include a computerized analysis capability that assesses the significance of what is happening and recommends possible corrective action options, along with the implications of each recommended corrective action.

The scope of Cyber Panel *control* is the subset of all systems over which it has direct authority to initiate control actions in anticipation of or in response to cyber attacks. The scope of Cyber Panel *coordination* consists of all systems over which there is organizational authority but not direct control (e.g., component command assets). Internal system health information can be collected and reconfigurations recommended but not directly implemented. The scope of *awareness* is still broader, including other systems external to control or influence, that may cooperatively report data and can be watched to observe developing attack trends. The Cyber Panel will assist the collaborative plan, monitor and assess, and control functions to enhance survivable GCCS-M systems survivability; each of these functions is briefly described below.

**Plan**

The Cyber Panel will provide the inputs and tools to support planning for current and future operations. It will support the identification of critical mission and mission-support cyber assets and restoration priorities, including survivability thresholds, threat, vulnerability, and failure-propagation assessments, and cyber-resource constraints. The data that the Cyber Panel acquires from its sensors will be used to assess the health of existing systems in day-to-day use as well as in mission-specific planning.

**Monitor and Assess**

The Cyber Panel will enable sharable, local, and remote monitoring. Data sharing must be a flexible part of the Cyber Panel architecture to support redundancy and survivability to ensure that, where appropriate personnel are available, they can help monitor the system state.

Displays of sensor inputs must be flexible and effective. Displays should allow upper echelons of command to assess cyber situations in computers and networks associated with higher or lower levels of command. Operators could, for example, specify echelon views (e.g., one up, two down) they want presented to them on large display panels and workstations. The displays could also

support selectable situational awareness and control representations. Displays may be windowed with various system-state representations in the form of charts, statistics, text, three-dimensional graphics, and so on.

Simulations that can generate dynamic representations of system behaviors (e.g., representations using thermodynamic and self-similar behavioral models) will support Cyber Panel information displays. The Cyber Panel may exploit video teleconferencing, e-mail, office tools, and other collaboration tools available from the surrounding IT environment, although any such tools planned for use when the system is under duress must also be made survivable.

Cyber Panel assessment tools will be capable of identifying configuration conflicts in the runtime environment and will support threat assessment, vulnerability assessment, failure-propagation assessment, and incident assessment. Technical and mission models that can aid in relating the impact of a cyber incident to the mission will support Cyber Panel assessment tools. These tools will be capable of event correlation, aggregation, and semantic interpretation for higher echelon command situational awareness. Finally, Cyber Panel tools will support post-incident analysis.

**Control**

A secure and survivable remote configuration control capability will allow the Cyber Panel to exercise its control function. Cyber Panel administration must include establishing the scope of awareness, coordination, and control; managing system accounts; configuring network and system devices; setting and updating operating points to control or influence survivability operations in controlled and coordinated systems; managing and adaptively protecting information flow channels; and issuing additional procedures and guidance. The identification of configuration conflicts should be partially detectable within this administration function (e.g., through the use of constraint-based rules) and fully detectable in the assessment function (e.g., runtime testing and analysis).

The Cyber Panel will not have control of every aspect of the computing environment. For example, database transaction monitors will discover transaction problems and pass control to database recovery managers to resolve the transaction error condition. However, in some instances, the awareness of error conditions or the autonomous action of a system component may warrant notification to the Cyber Panel. Such notifications may be represented in the Cyber Panel as a frequency distribution of typed-error events over time or in the form of a resource request (e.g., a failure requiring activation of a backup data server). If an issue of resource contention exists, the Cyber Panel may be called upon to resolve the contention. Cyber Panel controls may be activated in the form of "throttles or selector switches." Controls may also be in the form of dispatched parameters or code.

The Cyber Panel must support collaborative responses in adapting its supported system(s) to avert undesirable consequences. Response may be autonomic or under human control. It may also be activated by direct control or by indirect command influence, and may require real-time, near real-time, and non-time-sensitive operations. Deconfliction support for appropriate, safe, and timely response is critical.

The Cyber Panel must also provide response-reporting capabilities and enable the selection of response tools. Responses may range from disconnection from the perceived source of threat to

reconfiguration of sensors; from deception to covert traceback; and from recovery of damaged resources to recovery via alternate resources.

Finally, the generation and selection of courses of action (COAs) requires integrated interaction of planning, assessment, response, and administration. The Cyber Panel will assist in COA development and selection, and may eventually automate it at some levels.

***Support Within the Survivable GCCS-M***

The Cyber Panel will be the decision-support interface to the IA and survivability functions inherent in the survivable GCCS-M architecture. Cyber Panel tools must allow the command staff to perform the following functions:

- **Plan** for the inclusion of mission-focused, survivable GCCS-M functions, allowing commanders to understand and display to commanders how the GCCS-M functions depend on their supporting IT capabilities to support planning for mission-focused GCCS-M survivability.
- **Monitor and assess** the survivability state of GCCS-M systems supporting a particular mission, emphasizing the failure modes and effects of the cyberspace environment on those systems.
- **Control** critical GCCS-M cyber configurations and response procedures as the cyber attack situation dictates, to ensure mission survivability in the face of evolving cyber-threat activity.

Cyber Panel capabilities will reside at multiple echelons of command. The Mission Commander will be responsible for the survivability configuration of GCCS-M and, through the Cyber Panel, will direct the necessary triage and adaptation operations for GCCS-M to retain mission survivability. Cyber Panel implications for GCCS-M include:

**Plan**

The command staff and supporting organizations using GCCS-M will use the Cyber Panel collaboratively to help plan GCCS-M survivability. This plan may include PKI-enabling specific GCCS-M applications, establishing restoration priorities for specific GCCS-M applications and support services, allocating Cyber Panel roles and responsibilities within the mission forces, and specifying critical GCCS-M performance thresholds.

**Monitor and Assess**

The command staff, through the Cyber Panel, will coordinate GCCS-M monitoring in support of a mission and inform the mission commander of the overall cyber situation of systems and networks assigned to maritime forces. Subordinate commands may be assigned specific cyber monitoring roles. All organizations will generally monitor their own local GCCS-M applications and supporting system services. Specified monitoring data will be shared through the Cyber Panel among the participating mission organizations; some organizations may be assigned to perform remote monitoring should local monitoring be disrupted.

The Cyber Panel will be used to assess GCCS-M health and possible attack state. The command staff will assist the commander in correlating relevant events at peer commands, as well as at echelons above and below his command level, to gain an understanding of their potential impact on GCCS-M and on the assigned mission.

**Control**

The Cyber Panel will support control of survivability aspects of GCCS-M system configurations. The configuration of Cyber Panel capabilities will also be adapted to support GCCS-M. This adaptation could include deployment of additional or different sensor/analysis suites, the re-configuration of existing sensor/analysis suites, changing the configuration of GCCS-M applications through the use of specialized integrity wrappers, and so on.

The Cyber Panel will be used to respond to anomalous behaviors or attacks on GCCS-M. Cyber Panel response actions will be fed into the higher echelon and subordinate local Cyber Panel displays to support collaboration on de-conflicting actions. If the Cyber Panel employs autonomic response capabilities in support of GCCS-M, the results triggered by an autonomic response will also be fed into the Cyber Panel displays.

***Considerations in Interfacing Cyber Panel Support for a Survivable GCCS-M***

Architectural enhancements will be required to perform the survivable plan, monitor and assess, and control functions that can help maintain a survivable GCCS-M in support of mission requirements. These architectural enhancements include:

**Sensor Replication, Distribution, and Variation**

The survivability architecture of the GCCS-M applications, servers, clients, network services, and supporting middleware will determine a basic sensor suite. Some sensors will be fixed in location and others may be deployable or adaptable. Some sensors will have autonomic response capabilities. This suite of sensors must also support monitoring the health and welfare of the Cyber Panel itself. Sensors must be capable of selectively exchanging information data with other sensors and/or analysis engines. Thus, the sensor data should be syntactically and semantically interoperable. Sensor data must be transformed into human-discernable form to support situational awareness. Some sensor data should remain local to the organization's Cyber Panel, as it may not be meaningful beyond the local enclave (e.g., error rates on local devices that remain within normal tolerances). Other sensor data, especially configuration compliance data or any unusual errors or occurrences, must be reported, even if their immediate effects are strictly local (e.g., disabling application audit functions, ensuring previously patched vulnerabilities are still correct when applying a new application service pack). This type of reporting would be necessary either because this information could, when viewed in a broader context, reveal an overall pattern that might indicate an attack or because it would be needed to verify overall system health. Various types of sensors should overlap in coverage (e.g., anomaly detectors) yet not interfere with one another. Sensors should always report locally and Cyber Panel report applications should report sensor data selectively to other echelons in accordance with normal reporting chains and mission-specific requirements.

**Controller-Actuator Replication, Distribution, and Variation**

Controller-actuators must also be arrayed within and across all layers of the GCCS-M system. Various types of controllers should overlap in coverage. Their actions must be coordinated so that they do not work against one another. The general approach is to activate controllers upon developing and selecting a COA that takes into account the potential failure modes and effects that could occur from a specific point and form of attack. If the control cannot preclude the advance or

spread of failure conditions from the point of attack, it should be removed from the COA alternatives. The placement decisions for autonomic controls (i.e., controls that are not invoked via a human-selected COA) must take into account whether such controls can be used as part of an attack as well as the need to communicate to the autonomic control changes in policy or rules of engagement that affect its response. Note that COA development must also take into account potential actions of autonomic controllers.

**Analysis Engine Replication, Distribution, and Variation**

Analysis engines must be capable of dealing with sensor data from various layers of the GCCS-M system. Data fusion, correlation, and aggregation must be supported for these engines along with the appropriate set of models to assist in developing situational awareness and potential courses of action. Correlation of results from different analysis engines supporting different aspects of the cyber-defense problem will be required. Replication and distribution of analysis engines should be established in conjunction with the replication and distribution of facilities where Cyber Panels operate.

**Cyber Panel Displays and Workstations**

The Cyber Panel should include adjustable large-panel and workstation displays. Multiple workstations could be resident within a single Operations Center, each being configured to perform independent functions or being capable of performing the full range of Cyber Panel display and control functions. Local Cyber Panel Operations Center capabilities may exist on a single dedicated workstation, but must be able to be set up and appropriately configured on any available workstation.

**Communications Among Cyber Panel Components**

Communications among Cyber Panel components must be maintained during a cyber attack. Out-of-band communications among all GCCS-M Cyber Panel components may not be feasible, but tradeoffs must be assessed and designs developed to meet overall needs for Cyber Panel survivability.

## 5.6 Approach to Assurance Argument Development

Why should anyone believe that a particular system design achieves a particular survivability requirement? The purpose of the assurance argument is to answer this question. Indeed, without a clear answer to this question, spending extra resources on a purportedly survivable system design would be a dubious investment.

The ability of a system to continue to operate in the face of a cyber attack depends not only on its hardware and software design, but also on the physical protection of the equipment, the procedures used to operate and maintain the system, and the ability and willingness of human operators to carry out those procedures faithfully. The assurance argument for a system needs to reflect all of these dependencies.

Many different forms of assurance can be applied in the creation of the overall assurance argument for a system. For example, a given system component may be relied on to detect a certain class of failure with a high degree of certainty because it has been tested in an appropriate environment and found to behave as expected, perhaps in accordance with some established

standard or interface, or because it was created using a process and methodology found to work well on similar problems, or because its structure is well-documented and has been subjected to outside reviews, or because its developers have experience and credentials that qualify them to design and build such a component, or because of a combination of such factors. The assurance argument provides a place for such evidence to be recorded so that an outside observer can understand the basis for believing a system's claim to be survivable, and also can identify areas where the argument might be made stronger through the inclusion of additional evidence or additional precautions.

The creation of the assurance argument itself can be a significant effort and needs to be planned. The plan for creating an assurance argument is sometimes referred to as an *assurance strategy* and may be displayed in an *assurance map*. In the best case, the assurance argument should probably be created in consonance with the system design. As the system design develops and evolves, so should the assurance argument. System designs that lead to simpler, more convincing assurance arguments should, other things being equal, be preferred to those with weaker or more complicated ones.

The assurance argument for a particular survivability property can be portrayed as a tree, with the particular claim at the top and supporting assertions or assumptions elaborated below. DARPA is supporting technology to help develop and display assurance arguments as described in Appendix D.

# 6. Conclusions

The integration of survivable middleware, clients and servers, and network services into an efficient, effective, survivable system with Cyber Panel capabilities is a challenging task. This paper has outlined a potential overall structure for a survivable GIG system, using the GCCS-M Ashore as a representative system, and presented some technologies that could fit into this structure, but to design and build such a system, numerous additional issues would need to be addressed, such as the following: How much redundancy is enough? Which data is most critical and deserves the strongest protection against attack? Which parts of the network infrastructure are weakest and most need to be strengthened?

To address these issues adequately, more specific system requirements are needed. The system requirements should actually stem from a consideration of those missions that the system supports or may be expected to support in the future. Survivability requirements would need to be identified as well. These would address the critical functions that must be maintained to support different missions, the kinds of attacks to be survived, and for how long. Other requirements might need to address the time it takes to recover from an attack, the time for reconstitution, the categories of attack that can be tolerated, and so on. With these requirements identified, assessing tradeoffs among the many technology options presented for engineering survivability into the system would begin to be possible. They would also form the basis for the fundamental claims of an assurance argument for the system.

Readers should also recognize that overlaps exist in the technologies and approaches discussed in Section 5 and that this paper has not attempted to resolve these overlaps. For example, both Directory-Enabled Networks and Enterprise Application Integration provide facilities for programs to address resources available in a network without specifying their location; the DEN or EAI infrastructures only provide the ability to locate the desired object or service. Perhaps an advantage would be to provide two such mechanisms, but certainly some economies would be realized in having only one. Cyber Panel functions are also likely to require substantial interaction with these facilities; how this interaction would proceed has not been addressed.

In summary, the considerations in developing survivable architectures for GIG systems, presented in this document, should help system architects to state and meet survivability requirements for critical systems.

## 7. References

[ARBA00] Arbaugh, W. A., W. L. Fithen, and J. McHugh. "Windows of Vulnerability: A Case Study Analysis." *IEEE Computer*, Vol. 33, No. 12, December 2000, pp. 52-59.

[ASLA96] Aslam, T., I. Krsul, and E. Spafford. *Use of a Taxonomy of Security Faults.* Technical Report 96-051. Purdue University Computer Science Dept., September 1996.

[AXEL99] Axelsson, S. "The Base-Rate Fallacy and its Implications for the Difficulty of Intrusion Detection." *Proc. 1999 ACM Conference on Computer and Communications Security,* 1999.

[BAIL97] Bailey, R., B. George. C. Bowers, et al. *The Network Rating Methodology: a Framework for Assessing Network Security.* National Security Agency, 1997.
http://chacs.nrl.navy.mil/projects/VisualNRM /nrm_3rd_draft.html

[BAIN00] Bain, C., D. Faatz. A. Fayad, and D. Williams. "Diversity as a Defense Strategy in Information Systems: Does Evidence from Previous Events Support Such an Approach?" *Proceedings of the 14th International IFIP TC-11 WG 11.5 Working Conference on Integrity and Internal Control in Information Systems (IICIS 2001)*. Brussels, Belgium, August 2000.

[BALZ00] Balzer, R. M., and N. Goldman. "Mediating Connectors: A Non-ByPassable Process Wrapping Technology." *Proc. DARPA Information Survivability Conference & Exposition (DISCEX),* Vol. II, January 2000.

[BELL01] Bellovin, S. M. "Computer Security – An End State?" *Communications of the ACM*, Vol. 44, No. 3, March 2001, pp. 131-132.

[BISH95] Bishop, M. *A Taxonomy of UNIX System and Network Vulnerabilities*. Technical Report CSE-95-10. The University of California, Davis, May 1995.
http://seclab.cs.ucdavis.edu/projects/vulnerabilities/scriv/ucd-ecs-95-10.pdf

[CISC01] Cisco Systems.
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/diren.htm

[COMM99] Common Criteria Project Sponsoring Organizations. *Common Criteria for Information Technology Security Evaluation*, Version 2.1, August 1999.

[COMO01] *Common Vulnerabilities and Exposures Dictionary*. http://cve.mitre.org/

[DACI96] Dacier, M., Y. Deswarte, and M. Kaaniche. *Quantitative Assessment of Operational Security: Models and Tools*. LAAS Research Report 96493, May 1996. http://dbserver.laas.fr/pls/LAAS/publis.rech_doc?langage=ENG&clef=18553

[DEFE99] Defense Information Systems Agency. *Defense Message System Product Plan*, Version 3.03. August 20, 1999. http://www.disa.mil/D2/dms/documents/download/0222-04.zip

[DEPA01] Department of Defense, *Mandatory Procedures for Major Defense Acquisition Programs (MDAPs) and Major Automated Information System (MAIS) Acquisition Programs.* DoD 5000.2-R (Interim), 4 January 2001.

[DESW98] Deswarte, Y., K. Kanoun, and J. C. Laprie. "Diversity against Accidental and Deliberate Faults." *Proc. ONR/NSF Workshop on Computer Security, Dependability, & Assurance: From Needs to Solutions*, IEEE Press, 1998, pp. 171-181. http://computer.org/proceedings/csda/0337/03370171abs.htm

[DIST01] Distributed Management Task Force: http://www.dmtf.org/standards/standard_den.php

[EAST97] Eastlake, D., and C. Kaufman. *Domain Name System Security Extensions*. RFC 2065, January, 1997. http://www.ietf.org/rfc/rfc2065.txt

[ERLI00a] Erlingsson, U., and F. B. Schneider. "SASI Enforcement of Security Policies: A Retrospective." *Proc. DISCEX*, Vol. II, January 2000.

[ERLI00b] Erlingsson, U., and F. B. Schneider. "IRM Enforcement of Java Stack Inspection." *Proc. 2000 IEEE Symposium on Security and Privacy*, May 2000.

[FABR93] Fabre, J-C, Y. DesWarte, and B. Randell. *A Framework for the Design of Secure and Reliable Applications by Fragmentation-Redundancy-Scattering*. Technical Report 410, University of Newcastle Upon Tyne, Newcastle Upon Tyne, February 1993.

[FAYA99] Fayad, M. E., D. C. Schmidt, and R. Johnson. *Implementing Application Frameworks: Object-Oriented Framework at Work*, Wiley, 1999.

[FRAS00] Fraser, T., L. Badger, and M. Feldman. "Hardening COTS Software with Generic Software Wrappers." *Proc. DISCEX*, Vol. II, January, 2000.

[GOLL98] Gollman, D. *Computer Security*. John Wiley, 1998.

[HOLL01] Hollebeek, T., and D. Berrier, "Interception, Wrapping, and Analysis Framework for Win32 Scripts," to appear, *Proc. DISCEX II*, June 2001.

[HOWA97] Howard, J. *An Analysis of Security Incidents on the Internet 1989-95*. PhD Dissertation, Engineering and Public Policy, Carnegie Mellon University, 1997. (See Chapter 6: "A Taxonomy of Computer and Network Attacks.")
http://www.cert.org/research/JHThesis/Start.html

[JAJO98] Jajodia, Sushil, Peng Liu, and Catherine D. McCollum. "Application-Level Isolation to Cope with Malicious Database Users." *Proc. 14th Annual Computer Security Applications Conference*, Phoenix, AZ, December 1998, pp. 73-82.

[JOHN01] Johnston, D. "F.B.I. Agent Charged As Spy Who Aided Russia 15 Years." *New York Times*, February 21, 2001, p. A1.

[KANG00a] Kang, M. H., and J. N. Froscher. "Software Architecture and Logic for Secure Applications." *Proc. DISCEX 2000*, Hilton Head Island, SC, 2000.

[KANG00b] Kang, M. H., and J. N. Froscher. "A Framework for Secure Enterprise Computing." *Proc. Fifth International Workshop on Enterprise Security*, Gaithersburg, MD, 2000.

[KEND99] Kendall, K. *A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems*. S.M. Thesis, MIT Department of Electrical Engineering and Computer Science, June 1999.

[KENT98] Kent, S., and R. Atkinson. *Security Architecture for the Internet Protocol*. RFC 2401, November 1998. http://www.ietf.org/rfc/rfc2401.txt

[KENT00] Kent, S., C. Lynn, and K. Seo. "Secure Border Gateway Protocol (Secure-BGP)." *IEEE Journal on Selected Areas in Communications*, Vol. 18, No. 4, April 2000, pp. 582-592.

[LAND94] Landwehr, C. E., A. R. Bull, J. P. McDermott, and W. S. Choi, "A Taxonomy of Computer Program Security Flaws, with Examples." *ACM Computing Surveys*, Vol. 26, No. 3, September 1994, pp. 211-254.

[LEEA90] Lee, A., and T. Anderson. *Fault Tolerance: Principles and Practice, 2nd Ed.*, Springer-Verlag, 1990.

[LINT99] Linthicum, D. S. *Enterprise Application Integration*. Addison-Wesley Information Technology Series, 1999.

[LYNN99] Lynn, C., J. Mikkelson, and K. Seo. *Secure BGP (S-BGP).* Internet Draft, October 1999. http://www.ir.bbn.com/sbgp/draft-clynn-s-bgp-protocol-00.txt

[MARK01] Markham, T., and C. Payne. "Security at the Network Edge: A Distributed Firewall Architecture." Secure Computing Corp. MN, *Proc. DISCEX II*, June 2001.

[MCGR00] McGraw, G., and G. Morrisett. "Attacking Malicious Code: A Report to the Infosec Research Council." *IEEE Software*, Vol. 17, No. 5, September/October 2000.

[MOOR00] Moore, A., B. Montrose, and B. Strohmayer. *A Tool for Mapping Enterprise Security Assurance*. Technical Report 5540-051a:apm, Naval Research Laboratory, September 2000.

[MYER99] Myers, A. C. "JFlow: Practical Mostly Static Information Flow Control." *Proc. 26th ACM Symposium on Principles of Programming Languages (POPL)*, 1999.

[NATI01] National Institute of Standards and Technology. *ICAT Metabase*. http://icat.nist.gov/icat.taf

[NATO00] National Research Council, Naval Studies Board. *Network-Centric Naval Forces: A Transition Strategy for Enhancing Operational Capabilities*. National Academy Press, 2000, 489 pp.

[NECU97] Necula, G. "Proof-Carrying Code." *Proc. 24th ACM Symposium on POPL*, 1997.

[PAYN93] Payne, C. N., J. N. Froscher, and C. E. Landwehr. "Toward a Comprehensive INFOSEC Certification Methodology." *Proc. 16th National Computer Security Conference (NCSC)*, pp. 165-172, Baltimore, MD, September 1993.

[ROBE81] Roberts, N., W. Vesely, and D. Haasl. *Fault Tree Handbook*. NUREG-0492, Office of Nuclear Regulatory Research, 1981.

[SCAL01] Scalable Intrusion-Tolerant Architecture for Distributed Services, http://www.anr.mcnc.org/projects/SITAR/SITAR.html

[TALL01] Tallis, M., and R. Balzer. "Document Integrity through Mediated Interfaces," to appear, *Proc. DISCEX II*, June 2001.

[WIEN61] Wiener, N. *Cybernetics or Control and Communication in the Animal and the Machine, Second Edition*. MIT Press, 1961, 212 pp.

[WILS96] Wilson, S., J. McDermid, P. Kirkham, and P. Fenelon. "The Safety Argument Manager: An Integrated Approach to the Engineering and Safety Assessment of Computer Based Systems." *Proc. IEEE Symposium and Workshop on Engineering of Computer-Based Systems*, pp. 198–205, 1996.

[WILS97] Wilson, S., P. Kirkham, and M. Cassano. *SAM 4 User Manual.* University of York, 1997.

[WULF96] Wulf, W., C. Wang, and D. Kienzle. "A New Model of Security for Distributed Systems." *Proc. New Paradigms in Security Workshop*, Lake Arrowhead, CA, 1996.

[WYLI00] Wylie, J., M Bigrigg, J. Strunk, G. Ganger, H. Kiliccote, and P. Khosla. Survivable Information Storage Systems. *IEEE Computer*, Vol. 33, No. 8, August 2000, pp. 61-68.
http://www.cs.cmu.edu/~jwylie/Publications/IEEEComputerAugust20000_SurvivableInformationSystems.pdf

# APPENDIX A. Framework for Describing Vulnerabilities and Attacks

The primary aims of the exploitation of vulnerabilities with respect to a survivable GIG system and of the execution of attacks on a survivable GIG system may divided into four large categories:

1. Denial of service
2. Data compromise
3. Data modification
4. Control of system or part of system

These are ordered more or less according to how severely the adversaries or attackers interact with the system. Another way to categorize exploitations of vulnerabilities and executions of attacks is to divide them into the periods when they are intended to happen:

1. Design phase
2. Implementation phase
3. Operational phase

Discussions among many interested in understanding this area show that it is difficult to find categorizations that are entirely satisfactory. The result below is just another tentative attempt.

To determine how well a survivable GIG system counters vulnerabilities and attacks the following more detailed list is to be considered. The list is drawn up with respect to system components and personnel. No such list can be exhaustive and every such list will reflect its authors' backgrounds and biases.

1. Network infrastructure vulnerabilities/attacks
    a. Vulnerabilities to routers, switches, and similar network nodes
    b. Vulnerabilities to boundary controllers, firewalls, proxies
    c. Routing protocol attacks
    d. Malicious protocol tunneling
    e. Protocol misuse
    f. Cause protocol changes
    g. Attacks on links
    h. Eavesdropping on network links
    i. Network flooding
    j. Cause links to go down
    k. Network packet and data modification
    l. Spoofing attacks
    m. Man-in-the-middle attacks
    n. Denial-of-service attacks on network
    o. Distributed denial-of-service attacks on network
2. Directory vulnerabilities/attacks
    a. Modify contents of directory
    b. Denial-of-service attacks against directory
3. Network management node vulnerabilities/attacks
    a. Take management control
    b. Denial-of-service attacks against management node

4. Network authentication server vulnerabilities/attacks
   a. Modify contents of authentication server
   b. Denial-of-service attacks against authentication server
5. PKI vulnerabilities/attacks
   a. Modify certification authorities, certificates
6. Network operations vulnerabilities/attacks
   a. Isolate operations center
7. IDS and Cyber Panel vulnerabilities/attacks
8. Client/server/host-based vulnerabilities/attacks
   a. Prevent authentication
   b. Circumvent authentication
   c. Prevent client/server access to network
   d. Circumvent or defeat access control mechanisms
   e. Denial-of-service attacks on clients/servers
   f. Distributed denial-of-service attacks against servers
   g. Modify clients/servers/hosts
9. Software/firmware/middleware vulnerabilities/attacks
   a. Viruses, worms, Trojan horse program vulnerabilities/attacks
   b. Mobile code vulnerabilities/attacks
   c. Life-cycle attacks/compromised software
   d. Attacks during software maintenance and updates
   e. Spyware and exfiltration vulnerabilities/attacks
   f. Covert channel vulnerabilities/attacks
10. Personnel-related vulnerabilities/attacks
    a. Vulnerabilities/attacks during development
    b. Vulnerabilities/attacks during installation
    c. Vulnerabilities/attacks during maintenance
    d. Vulnerabilities/attacks during service retirement
    e. Insider vulnerabilities/attacks
    f. Social engineering

The following is a list of the broad protections that may be offered by a survivable GIG system. It is really a selection among many possible protections:

A. Network connectivity protections
   a. Authentication of network administrator
   b. Robust network routing protocols
   c. Dynamic routing
   d. Firewalls, proxy services
   e. VPN, Ipsec
   f. QoS
   g. Dynamic IP addressing
   h. Mobile nodes
   i. Survivable IP
   j. Intelligent agents
B. Network management protections

   a. Network monitoring
   b. Service registration
   c. SNMPv3
   d. Robust and secure configuration and upgrading
   e. Host and network IDS/IDR
C. Directory protections
   a. DNSSec
   b. Dynamic DNS
   c. Robust PKI
   d. Directory-enabled networking protections
D. IDS and Cyber Panel vulnerability/attack protections
   a. Protected control path
   b. Robust IDS
E. Client/server/host protections
   a. Strong authentication of user to end system
   b. Authentication of client to network
   c. Access control mechanisms
F. Software/firmware/middleware protections
   a. Cryptographic checksums, "tripwire"
   b. EAI protections
   c. Spyware and exfiltration vulnerability/attack protections
   d. End system protections
   e. Boundary protections
   f. Covert channel vulnerability/attack protections
   g. Covert channel detection methods
G. Personnel-related protections
   a. Configuration management
   b. Personnel controls
   c. Internal system checks and protections

Table A-1 presents a matrix showing the rationale for protections countering vulnerabilities. This matrix is a notional view that relates known attacks to a greater or lesser degree. It is offered as a framework but should not be viewed as complete. A developer of a specific survivable system will need some similar framework to support assertions and assurances about the system's ability to survive attacks. [See the following: SGIG-Threat-Rationale-R2new.xls.]

# APPENDIX B. Cyber Panel Component Types and Functions

This appendix briefly overviews the potential types of sensors, controller-actuators, analysis engines, models, and human-computer interfaces (HCIs) for situational awareness and system controls. Note that not all sensors and controller-actuators are "owned" by the Cyber Panel, and many may not even be directly interfaced to the Cyber Panel. However, there may be cases where these independent components need to interface indirectly with the Cyber Panel.

For example, sensors may log data to databases that can be accessed through Cyber Panel tools; controllers that perform distribution, replication, and consistency management may register their results in some fashion (e.g., directory services or brokers), allowing the Cyber Panel to gain that information when and if needed. During system engineering, responsibilities and interfaces must explicitly be allocated to the Cyber Panel.

Table B-1 is not an all-inclusive list, but rather a representative one provided to aid the reader with a quick review of these potential Cyber Panel component types and their functions.

**Table B-1. Cyber Panel Component Types and Functions**

| Component | Type | Function |
|---|---|---|
| **Sensors** | Filters | Allow specified data to flow while disallowing other data. |
| | Detectors | Establish that a specified condition has been discovered. Detectors can be real-time or non-real-time. They may be signature-based, statistically based, and/or rule-based. |
| | Auditors | Record specified system and human activity along with the data associated with such activities. Can have detectors embedded in auditing applications or auditing applications can support detectors. Conditional specifications and thresholds for alarms may be set within auditors. |
| | State Monitors | Examine the reached state of one or more system components to see if that state agrees with a specified condition (e.g., pre- and post-conditions in executing code); if the reached state does not agree with the specified state an error/anomaly condition is flagged and error handling/response routines invoked. |

| Component | Type | Function |
|---|---|---|
| | Performance Monitors | Examine performance parameters of one or more system components to see if these parameters meet specified performance conditions; can be real-time or non-real-time and may or may not raise alerts |
| **Controller-Actuators** | In-line Application-Controllers | Perform distribution, replication, and consistency management; access and execution controls. May also include service, transformation, and recovery controls. |
| | Middleware Controllers | Perform distribution, replication, and consistency management; access and execution controls. May also include service, transformation, and recovery controls. |
| | Environment Controllers | Perform replication and consistency management; access, service, and recovery controls. |
| | Data Controllers | Perform distribution, replication, & consistency management; access, transformation, and recovery controls. |
| | Host/Network Controllers | Distribution, replication, and configuration management; access, service, transformation, and recovery controls. |
| **Analysis Engines** | Fusion/Correlation/ Aggregation | Combine, match, differentiate, and tabulate instances of specified data parameters from various sources to provide new perspectives through the resulting data. |
| | Inference | Infer a situation from a set of phenomena and other known facts. |
| | Deductive Reasoning | Conclude the elements creating a situation, given a set of evidence |
| | Statistical | Determine a situation from the application of statistical functions on collections of specified parametric data. |
| | Probabilistic | Determine the likelihood of a situation occurring or the likelihood of a set of elements producing such a situation. |
| **Models** | Mission Model | Provide mission abstractions and relationships that can be used in Cyber Panel analyses. |

| Component | Type | Function |
|---|---|---|
| | Process Model | Provide process abstractions and relationships that can be used in Cyber Panel analyses. Link higher-level mission models to system models. |
| | System Model | Provide system abstractions and relationships that can be used in Cyber Panel analyses. Failure models can be derived from such system models. |
| | Failure Model | Provide understanding of consequences of system component failures and how such failures can be produced. When used in combination with other models, provide means to inform triage response to failure situations. |
| | Threat Model | Provide abstractions of known or postulated threats. May be expanded and more tightly coupled to other models by incorporating appropriate abstractions for both known or postulated vulnerabilities and attacks. |
| | Performance Model | Provide understanding of system and component operations in normal and degraded modes using benchmarking behavioral parameters. |
| **HCIs for Situational Awareness and System Controls** | Alarms | Provide alerting indication of a specified condition that must be brought to the attention of the Cyber Panel operator(s). |
| | Charts | Provide visual mapping of parameters associated with a specified condition or set of conditions. |
| | Graphics | Provide two-dimensional representation of a specified component, condition, or set of conditions. |
| | Images | Provide photographic depiction of a specified condition or set of conditions. |
| | 3D Models | Provide three-dimensional representation of a specified component, condition, or set of conditions. |
| | Animation | Provide time-sequenced steps of the dynamics of a specified condition or set of conditions. |

| Component | Type | Function |
|---|---|---|
| | Control Instrumentation | Provide analog or digital state representation of specified parameters to report/direct component and system behaviors. Also, allow Cyber Panel operator(s) to influence or perform remote control through the manipulation of controller-interface representations for discrete controllers (e.g., on/off and selector switches) and continuous controllers (e.g., dials or sliders analogous to rheostats and potentiometers). |

# APPENDIX C. Cyber Panel Interface Requirements

This appendix captures the essential interface requirements for Cyber Panel.

**Table C-1. Cyber Panel Interface Requirements**

| Components | Interface Requirements |
|---|---|
| **Sensors** | • Have all available sensors registered in its databases.<br>• Have the ability to use inherent sensors built into the components and augmenting sensors to cover a specific component or range of components.<br>• Ensure that sensors can exchange data with other sensors and analysis engines.<br>• Be able to receive and manipulate all sensor data.<br>• Ensure that sensor representations for visualization and instrumentation have a common "look and feel."<br>• Be able to add new and deploy replacement sensors.<br>• Have a control interface to invoke sensor adaptation, where sensors can be made dynamically adaptive.<br>• Be able to monitor the health and welfare state of sensors.<br>• Ensure that the alarms and alerts established through direct sensor feeds are sufficiently designed to preclude masking.<br>• Ensure that sensor coverage is overlapped.<br>• Ensure that redundant sensors are managed whether in an "active" reporting state or a "ready" non-reporting state.<br>• Ensure that all available sensors are managed during any recovery process with potential sensor degradation and failure conditions explicitly incorporated in system response and recovery strategies. |
| **Controller-Actuators** | • Have a human-computer interface to support control activities.<br>• Have an autonomic controller-actuator reporting capability;<br>• Have all available controller-actuators registered in its databases.<br>• Be able to report current control states for all components.<br>• Have the ability to use inherent controls built into the components and augmenting controller-actuators to cover a specific component or range of components.<br>• Ensure that response actions and results are registered in its databases.<br>• Be able to use control state parameters generated for a |

| Components | Interface Requirements |
|---|---|
| | selected course of action.<br>• Ensure that controller-actuator representations for visualization and instrumentation have a common "look and feel."<br>• Be able to add new and deploy replacement controller-actuators.<br>• Have a control-actuator interface to activate sensor adaptation, where sensors can be made dynamically adaptive.<br>• Be able to monitor the health and welfare state of controller-actuators.<br>• Ensure that autonomic actuators cannot unintentionally create a failure condition or amplify a failure condition (e.g., denial of service).<br>• Ensure that controller-actuator coverage is overlapped.<br>• Ensure that redundant controller-actuators are managed whether in an "active" state or a "ready" state. |
| **Analysis Engines** | • Be able to receive, interpret, fuse, correlate, and aggregate data from all sources of system-state data and various models.<br>• Be able to reason using system-state data and various models to Infer a situation from a set of phenomena and other known facts.<br>• Conclude the elements creating a situation, given a set of evidence.<br>• Determine a situation from the application of statistical functions on collections of specified parametric data; and/or determine the likelihood of a situation occurring or the likelihood of a set of elements producing such a situation.<br>• Be able to generate alternative courses of action within a specified timeframe.<br>• Be able to use models in the generation of course-of-action alternatives.<br>• Support the selection of the optimal course of action within a specified time frame.<br>• Be able to link system state to mission needs and identify mission criticality for component service "X" in situation "S".<br>• Be able to use sensor and controller actuator data to identify degraded modes of operations and possible consequences to mission processes in the generation of |

| Components | Interface Requirements |
|---|---|
| | alternative courses of action.<br>• Be able to provide rationale for the course-of-action alternatives.<br>• Provide a common "look and feel" of courses-of action to the display elements.<br>• Ensure that redundant analysis engines are managed whether in an "active" state or a "ready" state. |
| **Models** | • Be able to maintain a varied set of models.<br>• Be able to display model abstractions;<br>• Be able to support simulation.<br>• Be able to have specified data produced from one model be used in another.<br>• Be capable of representing the system and its components;<br>• Be capable of representing system behaviors and performance thresholds.<br>• Be capable of representing acceptable degraded modes of operation.<br>• Be capable of representing system and component failure modes and effects.<br>• Be capable of representing mission goals and processes, and linking them to system component services.<br>• Be capable of representing known threats, vulnerabilities, and attacks.<br>• Ensure that redundant models are managed in a consistent fashion. |
| **HCIs for Situational Awareness and System Controls** | • Provide a common "look and feel."<br>• Support interactive representations of sensor-controller (continuous and discrete) "instrumentation."<br>• Support zoom-in and zoom-out with navigation to enable system state review as well as model review, maintenance, and development.<br>• Support development of common report forms.<br>• Support Web-based tools, office tools, e-mail, and video teleconferencing.<br>• Support special function keys.<br>• Provide for various selectable modes of human interaction, including voice, point-and-click, touch pad/screen, etc.<br>• Support tailoring of large-panel displays.<br>• Support a minimal Cyber Panel operator configurations on any available workstation. |

# APPENDIX D. Survivable GIG Assurance Argument Development

Providing an assurance argument for survivable GIG systems presents many challenges. System designers and assessors need to understand clearly the causality, relationships, vulnerabilities, threats, system-level viewpoints, and objectives of an entire enterprise. To design a system that can be trusted or to assess security and survivability properties of a system, the related assurance arguments need to be developed and described effectively, clearly, and systematically so that the comprehensive assurance argument for a system of systems can be evaluated.

To this end, this appendix introduces techniques for developing an assurance argument map, which depicts claim trees associated with assurance arguments related to the enterprise security objective, providing causality, relationships, vulnerabilities, threats, and other system and environment-related issues. Included is a summary of the underlying approach, language, and tools used to develop an assurance argument map and present a method for deriving the assurance map for survivable GIG systems.

The assurance argument map, or assurance strategy [PAYN93], provides a roadmap to generating the complete assurance argument. All the evidence that supports a claim can be linked to that claim. An assurance argument map permits tracking security vulnerabilities by identifying assumptions made in one branch of the map that are not matched by validating claims made in another branch of the map. In other words, a statement can be an assumption in one branch while it can be a claim in another branch. Such assumptions become dependencies of the argument. Assumptions that are not so linked become vulnerabilities that need to be considered when assessing residual risk. These vulnerabilities must be assessed when deciding whether the residual risk is tolerable in the operational environment. The map developer also needs to ensure the integrity of the assumption validation mapping itself. If a portion of the map for one application is reused for another application, a reviewer can identify the lack of any claims to ensure consistent application of the assumptions. Conscientious application of the above approach helps to uncover such gaps, identifying security vulnerabilities that were not previously considered.

There is always more than one way to provide survivability services for an enterprise, and there may also be many different methodologies and styles for describing assurance arguments for the same information system. Survivability depends on the measures used for protection, detection and response, and recovery/reconstitution.

To guide the production of a comprehensive description of an assurance argument for survivable GIG systems, a survivability strategy can organize the claims about countermeasures in the hierarchical assurance argument structure depicted in Figure D-1. This strategy does not propose to "hard code" the precise partitioning of arguments in the maps for every application. Different systems or applications may have a different partitioning of their arguments for improved understandability or a different level of assurance. However, this approach provides comprehensive categories for most survivable GIG systems.

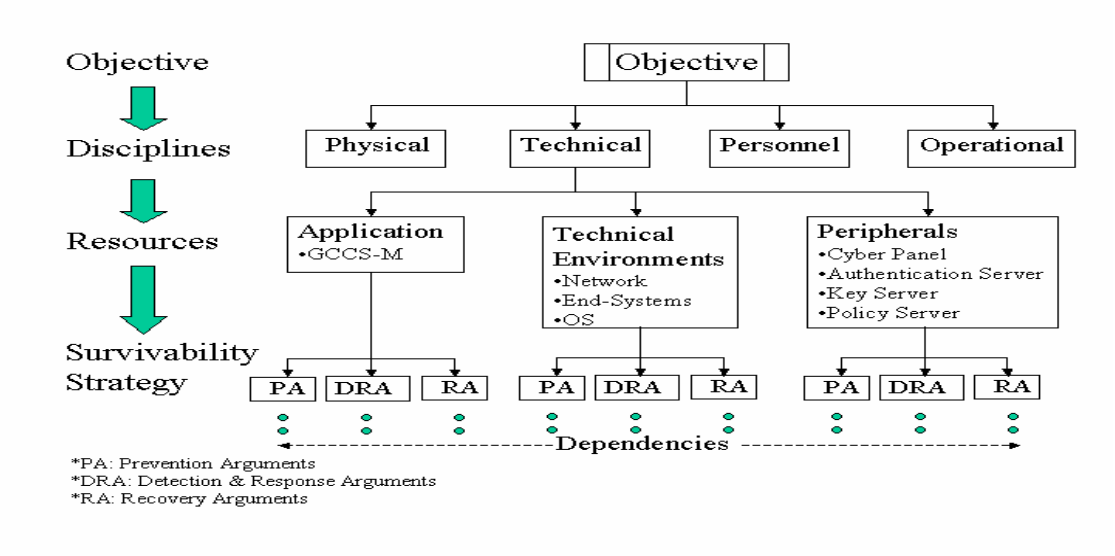


**Figure D-1. Assurance Argument Structure**

The assurance arguments that support the high-level objective are organized into four different disciplines in a map based on the NRM approach [BAIL97]: physical, technical personnel, and operational. Physical security involves the strength of physical mechanisms and structures used to protect and house the technology, such as strength of locks or safes. Technical security claims about the security and survivability of a system depend on physical, personnel, and operational security. Personnel security involves assurance about people, their trustworthiness and capabilities through some processes, such as personnel background investigation, training, and evaluation. Operational security involves the effectiveness of manual procedures, policies, and guidelines for handling and protecting information. The more conventional view of assurance comes from technical security, which involves security about combinations of hardware, software, and communications.

The assurance arguments under the technical security discipline (addressed here) are classified into three different resource branches: application, technical environments, and peripherals. The technical arguments within the application itself (e.g., survivable GCCS-M) are decomposed in the application branch. The technical arguments for system environments (e.g., network) are decomposed in the technical environments branch. Finally, the technical arguments for the peripheral components (e.g., authentication server), which assist the application's operations in the technical environments, are decomposed in the peripheral branch.

The technical arguments under each resource branch are decomposed in detail based on the survivability strategy described above. Each resource branch has three decomposition areas: Protection Arguments (PAs) (e.g., access control), Detection and Response Arguments (DRAs), and Recovery Arguments (RAs) to satisfy the survivability objective.

The Security Assurance Navigation and Environment (SANE) project (formerly ARGUS) supports the development of tools for producing assurance maps in the Composite Assurance Mapping Language (CAML) [MOOR00]. It will also provide capabilities for reuse of component assurance arguments as well as identification of assurance map patterns and linkage of this tool set and language with *Common Criteria* [COMM99] guidance.

CAML was developed by merging and extending several existing technologies [WULF96, PAYN93, ROBE81, WILS96, WILS97] to describe assurance arguments effectively in a well-organized format. The detailed description and usage information on CAML is available in [MOOR00]. Figure D-2 shows an example of the CAML structure with its primitives and definitions.

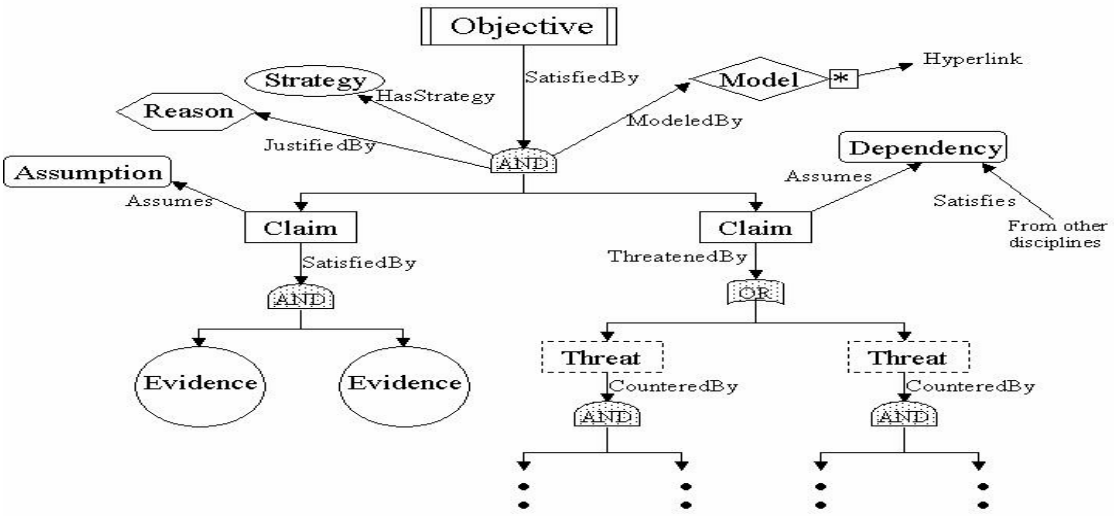


**Figure D-2. Assurance Argument Details Using CAML**

The following list briefly describe the rules and components of CAML:

- Objective: A statement expressing a security requirement of the countermeasure, system, network, or enterprise that is the reference of an argument.
- Claim: Statements that associate subjects with their attributes or properties.
- Assumption: A claim that is accepted without justification.
- Dependency: An assumption in one part of an argument that is validated by a claim in another part of the argument.
- Evidence: Data on which a judgment or conclusion about an assurance claim may be based.

- Hyperlink: A link from one component of an argument map to other components of the map or external components to provide for detail or clarification to the argument.
- Strategy: The approach taken for refining a claim into sub-claims or into evidences supporting the claim.
- Reason: A set of statements that ties together a set of sub-claims or evidences to establish a claim.
- Model: The architectural context on which a claim decomposition is based.
- Threat: A statement that expresses any circumstance or event with the potential to cause harm to an asset.

Distinct graphical primitives in different shapes represent key components of the assurance argument map. A textual summary of each component is shown inside each shape. The spine of an argument map hierarchically refines security claims about the system into sub-claims that, eventually, are linked with the evidence that a claim is satisfied. The flesh of an argument map describes supporting information about the refinement such as the general strategy, assumptions and dependencies, justifying reasons, and contextual models. Spine refinement may proceed using either AND-decomposition or OR-decomposition. By the AND-decomposition, all sub-claims or evidence must hold for the decomposed claim to hold. By the OR-decomposition, one of the sub-claims or evidences must hold for the decomposed claim to hold.

Not all CAML components are needed for every assurance argument map. Map developers use their discretion for choosing the necessary components to convey their argument satisfactorily. When a flesh component is connected to an AND/OR connector, it means this flesh component applies to all the arguments below the AND/OR connector. When a flesh component is connected to a spine (claim or evidence) directly, it means the flesh component applies to the particular spine. To provide more detailed descriptions of the map, the shapes can be hyperlinked. For instance, architectural diagrams can be hyperlinked to model shapes, and analytic proofs and tests can be hyperlinked to evidence shapes.

The Visual Network Rating Methodology (VNRM) helps users in drawing a graphical assurance argument map in CAML based on the approach (described above) that evaluates whether the target system adequately supports security services. The VNRM user's manual [MOOR00] is helpful in getting one started on using the tool. Additional information, including a packaged demonstration, is available at the VNRM Website [VNRM00].

VNRM was developed using Microsoft's Visual Basic and is dependent on external programs, such as Visio, MS Access, and MS Word. This requires a specific environment to use VNRM. Therefore, to provide higher portability and compatibility, currently under development is a successor to VNRM, SANE, purely in Sun Microsystems' Java. It will provide all of the drawing and documenting services without requiring external programs. It will also present new features to designers and assessors, associated with the reusability of assurance arguments, access control to CAML maps, and argument patterns.

# APPENDIX E. Active DARPA Projects Exploring Survivability-Related Technologies

## E.1 Technology Readiness Levels

This appendix lists active DARPA projects that are exploring survivability-related technologies, grouped according to the following programs: Organically Assured and Survivable Information System (OASIS), Cyber Panel, Survivable Wired and Wireless Infrastructure for Military Operations (SWIMM), Fault-Tolerant Networks, and Information Assurance. Table E-1 cites the technology readiness levels and their definitions [DEPA01], extracted from the following Website:

http://web1.deskbook.osd.mil/reflib/MDOD/031DR/013/031DR013DOC.HTM#T2

The technology readiness numbers provided in this appendix are estimates constructed by knowledgeable observers on the basis of project materials available in spring 2001. They were not constructed by DARPA personnel and do not represent an official DARPA assessment of projects in any respect.

**Table E-1. Technology Readiness Levels and Definitions**

| Technology Readiness Level | Description |
|---|---|
| 1. Basic principles observed and reported. | Lowest level of technology readiness. Scientific research begins to be translated into technology's basic properties. |
| 2. Technology concept and/or application formulated. | Invention begins. Once basic principles are observed, practical applications can be invented. The application is speculative and there is no proof or detailed analysis to support the assumption. Examples are still limited to paper studies. |
| 3. Analytical and experimental critical function and/or characteristic proof of concept. | Active research and development is initiated. This includes analytical studies and laboratory studies to physically validate analytical predictions of separate elements of the technology. Examples include components that are not yet integrated or representative. |
| 4. Component and/or breadboard validation in laboratory environment. | Basic technological components are integrated to establish that the pieces will work together. This is relatively "low fidelity" compared to the eventual system. Examples include integration of "ad hoc" |

| Technology Readiness Level | Description |
|---|---|
| | hardware in a laboratory. |
| 5. Component and/or breadboard validation in relevant environment. | Fidelity of breadboard technology increases significantly. The basic technological components are integrated with reasonably realistic supporting elements so that the technology can be tested in simulated environment. Examples include "high fidelity" laboratory integration of components. |
| 6. System/subsystem model or prototype demonstration in a relevant environment. | Representative model or prototype system, which is well beyond the breadboard tested for level 5, is tested in a relevant environment. Represents a major step up in a technology's demonstrated readiness. Examples include testing a prototype in a high fidelity laboratory environment or in simulated operational environment. |
| 7. System prototype demonstration in an operational environment. | Prototype near or at planned operational system. Represents a major step up from level 6, requiring the demonstration of an actual system prototype in an operational environment. Examples include testing the prototype in a test bed aircraft. |
| 8. Actual system completed and qualified through test and demonstration. | Technology has been proven to work in its final form and under expected conditions. In almost all cases, this level represents the end of true system development. Examples include developmental test and evaluation of the system in its intended weapon system to determine if it meets design specifications. |
| 9. Actual system proven through successful mission operations. | Actual application of the technology in its final form and under mission conditions, such as those encountered in operational test and evaluation. Examples include using the system under operational mission conditions. |

## E.2 OASIS Program

**1**

**Project Title:** New Approaches to Mobile Code: Reconciling Execution Efficiency with Provable Security

**Contact:** Michael Franz, UCI

**Objective:** Develop and demonstrate new mobile code transportation schemes that support the deployment of large mobile programs at a much better performance point than current solutions (e.g., Java) and with guaranteed statically verifiable security (i.e., representations that guarantee that any program written in the language will be type-safe).

**Survivability focus:** Programming infrastructure: Error/attack prevention. Provide underlying infrastructure for creating, distributing, and executing type-safe programs.

**Survivability principles/techniques applied/explored:** Hardened core.
**Technology Readiness Level estimate for December 2002:** 4

**2**

**Project Title:** A Binary Agent Technology for COTS Software Integrity

**Contact:** Dick Schooler, InCert; Anant Agarwal

**Objective:** Develop and demonstrate technology to instrument pre-existing binaries to detect violations of policy (e.g., out-of-bounds memory references, including buffer overruns, memory leaks) and report to system monitoring software.

**Survivability focus:** Programming infrastructure: error/attack detection and containment. Provide underlying infrastructure for detecting policy violations in running programs.

**Survivability principles/techniques applied/explored:**

**Technology Readiness Level estimate for December 2002:** 6

**3**

**Project Title:** Scaling Proof-Carrying Code to Production Compilers and Security Policies

**Contact:** Zhong Shao, Yale U.; Ed Felten, Andrew Appel, Princeton

**Objective:** Develop and demonstrate programming languages and infrastructure to support widespread deployment of type-safe mobile code programs together with proofs of advanced security policies that can be checked by the recipient.

**Survivability focus:** Programming infrastructure: error/attack prevention and detection. Provide program development and execution infrastructure.

**Survivability principles/techniques applied/explored:** Hardened core (smaller TCB); abstraction (disperse/obscure sensitive data); enforcement via formal specification and verification; self-monitoring and control; ACLs and authentication.

**URL:** http://www.cs.princeton.edu/sip/projects/darpapcc.php3 and
http://flint.cs.yale.edu

**Technology Readiness Level estimate for December 2002:** 5

**4**

**Project Title:** Sandboxing Mobile Code Execution Environments

**Contact:** Tim Hollebeek, Cigital

**Objective:** Develop software to detect and contain attacks attempting to exploit scripting mechanisms on Windows platforms.

**Survivability focus:** Programming infrastructure: error/attack detection, containment, and reporting.

**Survivability principles/techniques applied/explored:** Access control/intrusion detection

**Technology Readiness Level estimate for December 2002:** 5

**5**

**Project Title:** A Comprehensive Approach for Intrusion Tolerance Based on Intelligent Compensating Middleware

**Contact:** Amjad Umar, Telcordia

**Objective:** Develop more a generic approach for robust middleware through application of Fragmentation, Redundancy, and Scattering (FRS) techniques to a wide range of COTS middleware technologies, including CORBA, Message-Oriented Middleware (MOM), VoIP, and WAP.

**Survivability focus:** System design/composition: middleware.

**Survivability principles applied/explored:** Disperse/obscure sensitive data, deception, graceful degradation, and dynamism.

**Technology Readiness Level estimate for December 2002:** 3

**6**

**Project Title:** Intrusion Tolerance Using Masking, Redundancy, and Dispersion

**Contact:** Janet Lepanto, William Weinstein, Draper Laboratory

**Objective:** Develop and demonstrate system architecture to protect servers against attack, using redundant proxy servers, masking system fingerprint to attackers, and maintaining integrity of COTS backend database.

**Survivability focus:** System design/composition: servers, application (databases).

**Survivability principles applied/explored:** Disperse/obscure sensitive data, deception, diversity, dynamism, self-monitoring and control, recovery/restoration.

**Technology Readiness Level estimate for December 2002:** 5

**7**

**Project Title:** Active Trust Management for Autonomous Adaptive Survivable Systems

**Contact:** Howie Shrobe, MIT

**Objective:** Build self-monitoring and adaptive systems that detect failures, infer underlying compromises, and steer computations away from compromised or questionable resources.

**Survivability focus:** Programming infrastructure: execution, error/attack detection, containment, and reporting. System design/composition: application.

**Survivability principles/techniques applied/explored:**

**Technology Readiness Level estimate for December 2002:** 3

**8**

**Project Title:** Hierarchical Adaptive Control for QoS Intrusion Tolerance (HACQIT)

**Contact:** Jim Just, Teknowledge

**Objective:** Develop prototype survivable COTS-based server cluster for COTS/GOTS applications behind a firewall and with remote access for critical users via VPN. Cluster employs redundancy, diversity and migration, and decoy servers, internal sensors, adaptive content and separate communications paths for control in an effort to meet goal of four hours of uptime in the face of red team attack. Not dealing with flooding or other attacks on network infrastructure.

**Survivability focus:** System design/decomposition: servers.

**Survivability principles/techniques applied/explored**: No single points of failure (redundancy and redundancy management); graceful degradation (reconfiguration, self-monitoring and control).

**Technology Readiness Level estimate for December 2002:** 5

**9**

**Project Title:** A Scalable Intrusion-Tolerant Architecture for Distributed Services (SITAR)

**Contact:** Feiyi Wang, MCNC

**Objective:** Develop prototype server cluster based on proxy front ends to redundant COTS servers, and voting results, with degree of voting dependent on overall system survivability posture. Also, develop system models for prototype architecture to support reasoning about system behavior and system health.

**Survivability focus:** System design/decomposition: servers; also, system modeling.

**Survivability principles/techniques applied/explored:** Redundancy, graceful degradation, diversity, and dynamism.

**URL:** http:www.anr.mcnc.org/~sitar

**Technology Readiness Level estimate for December 2002:** 5

**10**

**Project Title:** Intrusion-Tolerant Distributed Object Systems

**Contact:** Greg Tally, Network Associates, Inc.

**Objective:** Design and develop prototype intrusion tolerant middleware (CORBA ORB), based on prior fault-tolerant CORBA work.

**Survivability focus:** System design/decomposition: middleware.

**Survivability principles/techniques applied/explored:** 1GS: link encryption; 2GS: firewalls; no single points of failure, graceful degradation, diversity, self monitoring and control, hardened core (in firewall).

**Technology Readiness Level estimate for December 2002:** 4

**11**

**Project Title:** Dependable Intrusion Tolerance

**Contact:** Alfonso Valdes, SRI International

**Objective:** Design and prototype intrusion-tolerant server architecture for intrusion detection application; tolerance proxy masks redundant server configuration with degree of operational redundancy/voting controlled based on detected attack level.

**Survivability focus:** System design/composition: server; also, intrusion detection.

**Survivability principles/techniques applied/explored:**

**Technology Readiness Level estimate for December 2002:** 5

**12**

**Project Title:** Intrusion-Tolerant Server Infrastructure

**Contact:** Dick O'Brien, Secure Computing Corp.

**Objective:** Prototype intrusion-tolerant server cluster based on hardened servers and custom hardware at network layer – policy-enforcing NIC card – packet filter controlled from outside host system.

**Survivability focus:** System design/composition: server.

**Survivability principles/techniques applied/explored:** Graceful degradation, no single points of failure, diversity, hardened core

**Technology Readiness Level estimate for December 2002:** 5

**13**

**Project Title:** Intrusion Tolerance by Unpredictable Adaptation (ITUA)

**Contact:** Partha Pal, BBN (also U. Illinois, Boeing)

**Objective:** Design and develop middleware-based mechanisms to make distributed systems intrusion tolerant using adaptation, redundancy, and uncertainty. Develop multi-mode redundancy mechanisms that present intrusion-tolerant view of system resources to application. CORBA base, designed to tolerate hybrid faults, combines 1GS and 2GS mechanisms; brings awareness and control of resources for intrusion tolerance.

**Survivability focus:** System design/composition: middleware, application objects.

**Survivability principles/techniques applied/explored:** No single point of failure, adaptation, uncertainty, dynamism, graceful degradation, incorporate 1GS and 2GS mechanisms.

**Technology Readiness Level estimate for December 2002:** 5

**14**

**Project Title:** Randomized Failover Intrusion-Tolerant Systems (RFITS)

**Contact:** Ranga Ramanujan, Architecture Technology Corp. (also ORA)

**Objective:** Develop and document in handbook design patterns for survivable systems resistant to DoS attacks, based on redundancy with failover and recovery process when attacked. Prototype selected survivability design techniques.

**Survivability focus:** System design/composition.

**Survivability principles/techniques applied/explored:** No single points of failure, graceful degradation, diversity, dynamism, deception (hiding, obfuscation, dodging).

**Technology Readiness Level estimate for December 2002:** 4

**15**

**Project Title:** Applicability of Model Predictive Control (MPC) to Intrusion Tolerance

**Contact:** Pavan Allaghatta or Walt Heimerdinger, Honeywell Labs

**Objective:** Model attack and control of intrusion-tolerant system responses using MPC mechanisms.

**Survivability focus:** System modeling/assurance.

**Survivability principles/techniques applied/explored:** Self-monitoring and control (esp. closed-loop control); automatic countermeasures (adaptation?) to improve survival probability.

**Technology Readiness Level estimate for December 2002:** 3

**16**

**Project Title:** Computational Resiliency

**Contact:** Steve Chapin, Syracuse

**Objective:** Intrusion tolerance through replication, migration and recovery of processes when attack is detected. Also, formal model using pi-calculus.

**Survivability focus:** Programming infrastructure: attack prevention/detection/recovery; also, system modeling/assurance.

**Survivability principles/techniques applied/explored:** Graceful degradation, deception, no single points of failure, dynamism.

**Technology Readiness Level estimate for December 2002:** 4

**17**

**Project Title:** Intrusion-Tolerant Software Architecture

**Contact:** Bruno Dutertre, Victoria Stavridou, SRI

**Objective:** Develop models of intrusion-tolerant system architectures, analyze models using game-theoretic techniques, develop intrusion-tolerant architectures for existing systems (GENOA, SEAS).

**Survivability focus:** System modeling/assurance.
**Survivability principles/techniques applied/explored:**
**Technology Readiness Level estimate for December 2002:** 3

**18**

**Project Title:** Survivability Analysis of Networked Systems

**Contact:** Jeannette Wing, Tom Longstaff, CMU

**Objective:** Develop and demonstrate models and methods for analyzing system survivability, incorporating probabilistic behavior and cost functions.

**Survivability focus:** System modeling/assurance.

**Survivability principles applied/explored:** Dynamism (attack/defender/intruder/system modeling).

**Technology Readiness Level estimate for December 2002:** 3

**19**

**Project Title:** Dependence Graphs for Information Assurance of Systems

**Contact:** Tim Teitelbaum, Grammatech

**Objective:** Develop tool for exposing control and data dependencies within a software component/system.

**Survivability focus:** Programming infrastructure: error prevention.
**Survivability principles/techniques applied/explored:**
**Technology Readiness Level estimate for December 2002:** 6

**20**

**Project Title:** Aspect-Oriented Security Assurance

**Contact:** Tim Hollebeek, Cigital

**Objective:** Capture security aspects of software development and make available to non-security-aware developers.

**Survivability focus:** Programming infrastructure: error prevention.

**Survivability principles/techniques applied/explored:** Hardened core at reduced cost.
**Technology Readiness Level estimate for December 2002:** 3

**21**

**Project Title:** Distributed Framework for Perpetually Available and Secure Information Systems (PASIS)

**Contact:** Greg Ganger, CMU

**Objective:** Apply fragmentation, scattering, redundancy techniques using threshold cryptography to prototype data storage subsystems to assess engineering tradeoffs, usability.

**Survivability focus:** System design/composition: client, server

**Survivability principles/techniques applied/explored:** Redundancy, redundancy management, disperse/obscure sensitive data.

**URL:** http://www.ices.cmu.edu/pasis

**Technology Readiness Level estimate for December 2002:** 6

**22**

**Project Title:** Self-Protecting Mobile Agents

**Contact:** Lee Badger, NAI Labs

**Objective:** Develop and prototype agent-based software system supporting computation on potentially hostile platforms, using Aglet infrastructure with heartbeats, periodic re-obfuscation.

**Survivability focus:** Programming infrastructure, attack prevention/system design/composition: application.

**Survivability principles/techniques applied/explored:** Redundancy and redundancy management, self-monitoring, disperse/obscure sensitive code/data.

**URL:** http://www.nai.com/nai-labs/asp-set/environments/spma.asp

**Technology Readiness Level estimate for December 2002:** 5

**23**

**Project Title:** Engineering a Distributed Intrusion Tolerant Database System using COTS Components

**Contact:** Peng Liu, UMBC

**Objective:** Detect, contain, mask, assess damage, and recover from malicious transactions submitted to COTS relational database.

**Survivability focus:** System design/composition: application.

**Survivability principles/techniques applied/explored:**

**Technology Readiness Level estimate for December 2002:** 4

**24**

**Project Title:** Tolerating Intrusions Through Secure System Reconfiguration (Willow)

**Contact:** Alex Wolf, U. Colo. (also John Knight, U. Va., and P. Devanbu, UC Davis)

**Objective:** Develop and demonstrate prototype system showing graceful degradation through reconfiguration as attacks/failures occur, using Software Dock for software distribution and modeling overall system behavior.

**Survivability focus:** System design/composition, system modeling/assurance.

**Survivability principles/techniques applied/explored:** Graceful degradation, self-monitoring and control.

**Technology Readiness Level estimate for December 2002:** 4

**25**

**Project Title:** Integrity through Mediated Interfaces

**Contact:** Bob Balzer, Teknowledge

**Objective:** Develop and demonstrate integrity protection for COTS (MS Office) application documents on Windows platform, using wrapper technology.

**Survivability focus:** Programming infrastructure: error/attack detection/containment; also, system design/composition: application.

**Survivability principles/techniques applied/explored:** Hardened core (application), disperse/obscure sensitive data, deception.

**Technology Readiness Level estimate for December 2002:** 7

**26**

**Project Title:** Enterprise Wrappers

**Contact:** Bob Balzer, Teknowledge, Mark Feldman, NAI Labs

**Objective:** Develop infrastructure for distribution and control of wrappers throughout diverse (Unix and Windows) system of systems.

**Survivability focus:** Programming infrastructure: development and distribution, execution, error/attack containment, reporting; system design/composition: system management.

**Survivability principles/techniques applied/explored:** Hardened core (application), disperse/obscure sensitive data, deception, self-monitoring and control.

**URL:** distribution of toolkit and papers at ftp:ftp.tislabs.com/pub/wrappers:

http://www.nailabs.com

**Technology Readiness Level estimate for December 2002:** 5

**27**

**Project Title:** Semantic Data Integrity

**Contact:** David Rosenthal, ORA

**Objective:** Develop and demonstrate techniques for detecting and repairing damage to the integrity of stored images; includes hierarchical hashing schemes (DSI mark).

**Survivability focus:** System design/composition: application.

**Survivability principles applied/explored:** Graceful degradation.

**URL:** www.oracorp.com/

**Technology Readiness Level estimate for December 2002:** 4

**28**

**Project Title:** Autonomix: Component, Network, and System Autonomy

**Contact:** Crispin Cowan, WireX

**Objective:** Develop methods for detecting and preventing damage from commonly exploited software vulnerabilities, such as buffer overflows, format bugs, etc.

**Survivability focus:** Programming infrastructure: Error/attack prevention and reporting, software execution integrity.

**System design/composition:** Clients and servers.

**Survivability principles/techniques applied/explored:**

**Technology Readiness Level estimate for December 2002:** 6

**29**

**Project Title:** Containment and Integrity for Mobile Code

**Contact:** Fred Schneider, Cornell (Andrew Myers)

**Objective:** Develop concepts and infrastructure to enforce security policies on low-level programs via type safety.

**Survivability focus:** System design/composition: server.

**Survivability principles/techniques applied/explored:** TCBs, no single points of failure, disperse/obscure sensitive data, reconfiguration, static analysis.

**Technology Readiness Level estimate for December 2002:** 6

**30**

**Project Title:** Intelligent Active Profiling for Detection and Intent Inference of Insider Threat in Information Systems

**Contact:** Joao Cabrera (Scientific Systems), Lundy Lewis (Aprisma)

**Objective:** Investigate the application of network management systems for the monitoring, detection and response of security violations carried out by insiders.

**Survivability focus:** Programming infrastructure: error/attack detection, attack response; attack/fault classification.

**Survivability principles/techniques applied/explored:**

**Technology Readiness Level estimate for December 2002:** 3

**31**

**Project Title:** A High Security Information System

**Contact:** Joe Johnson, U South Carolina

**Objective:** Assure high availability of Oracle-based operational state emergency management system against both natural and maliciously induced failures; mathematical models of system/components for evaluation.

**Survivability focus:** System design/composition: application, system management; system modeling/assurance.

**Survivability principles/techniques applied/explored:** 1GS, 2GS, no single points of failure, redundancy, redundancy management.

**Technology Readiness Level estimate for December 2002:** 5

**32**

**Project Title:** Efficient Code Certification for Open Firmware

**Contact:** Matt Stillerman, Odyssey Research Corp.

**Objective:** Detect potentially malicious firmware (Fcode) programs at boot time by detecting deviations from type-safe behaviors.

**Survivability focus:** Programming infrastructure: error/attack prevention (low-level/firmware).

**Survivability principles/techniques applied/explored:** Hardened core, self-monitoring and control.

**Technology Readiness Level estimate for December 2002:** 3

**33**

**Project Title:** Novel Applications of Military Science to Intrusion-Tolerant Systems

**Contact:** Matt Stillerman, Odyssey Research Corp.

**Objective:** Identify helpful analogs between conventional military science and cyber warfare, intrusion-tolerant systems (e.g., citadels, combined arms warfare, etc.).

**Survivability focus:** System design/composition.

**Survivability principles/techniques applied/explored:** (all? – paper study).

**Technology Readiness Level estimate for December 2002:** 2

**34**

**Project Title:** Encoded Program Counter: Self-Protection from Buffer Overflow Attacks

**Contact:** Akhilesh Tyagi, Iowa State University

**Objective:** Protect return addresses on stack and function pointers against malicious corruption by encrypting them; attacker cannot alter with predictable results.

**Survivability focus:** Programming infrastructure: error/attack prevention/detection.

**Survivability principles/techniques applied/explored:** Obscure sensitive data.

**Technology Readiness Level estimate for December 2002:** 4

## E.3 Cyber Panel Program

**1**

**Project Title:** Adaptive Knowledge-Based Monitoring/MAITA

**Contact:** Jon Doyle, MIT

**Objective:** Provide situational awareness and monitor/control system; facilitate rapid construction and adaptation of monitoring systems to counter new threats; provide mechanisms for automatically adapting monitor behaviors to changing situation.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system

**Email:** doyle@mit.edu

**Phone:** (617) 253-3512

**URL:** http://www.medg.lcs.mit.edu/projects/maita

**Technology Readiness Level estimate for December 2002:** 4

**2**

**Project Title:** Adaptive Model-Based Monitoring and Threat Detection

**Contact:** Al Valdes, SRI

**Objective:** Deliver a technology for monitoring and threat detection in large networks that has the sensitivity of signature techniques while retaining the generalization potential of statistical anomaly detection.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system

**Email:** alfonso.valdes@sri.com

**Phone:** (650) 859-4319

**URL:** http://www.sdl.sri.com/emerald/adaptbn-paper/adaptbn.html

**Technology Readiness Level estimate for December 2002:** 4

**3**

**Project Title:** ALPHATECH's Light Autonomic Defense System

**Contact:** Tiffany Frazier, ALPHATECH

**Objective:** Develop a quick-response, lightweight system that automatically responds to known and unknown automated assaults in execution time to either stop the attack or minimize its damage and restore the system to normal behavior; detect previously unknown assaults; detect and classify known assaults-in particular, (1) denial-of-service assaults, (2) slow-motion assaults, (3) Trojan horses in trusted software, and (4) insider misuse.

Maintain operation of high-priority mission-critical systems that are under automated attack while giving operators more time to define more effective countermeasures.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Make the system dynamic.

**Email:** tiffany.frazier@dc.alphatech.com

**Phone:** (703) 524-6263

**URL:** http://www.alphatech.com

**Technology Readiness Level estimate for December 2002:** 4

**4**

**Project Title:** Application Specific Intrusion Detection (ASID)

**Contact:** Anita Jones, U. Va.

**Objective:** Determine how to detect intruders in the context of application semantics.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** jones@cs.virginia.edu

**Phone:** (804) 982-2224

**URL:** http://www.cs.virginia.edu/~jones/IDS-research

**Technology Readiness Level estimate for December 2002:** 4

**5**

**Project Title:** ARGUS: Architecture for Cooperating Intrusion Detection and Mitigation Applications

**Contact:** Walt Heimerdinger, Honeywell

**Objective:** Correlate and analyze intrusion detection reports using Qualitative Bayesian estimation technology; combine results from multiple detectors at differing levels of detail; Create an intrusion reference model knowledge base that provides information for report analysis and for detector, firewall and aggregator configuration; suggest probable attacker plans based on intrusion reports.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** walt@htc.honeywell.com

**Phone:** (612) 951-7333

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 6

**6**

**Project Title:** Automatic Synthesis of Program-Based Triggers for Intrusion Tolerance

**Objective:** Promote reliable detection of events as triggers for intrusion-tolerant mechanisms based on system behavior and domain knowledge.

**Contact:** C.C. Michael, Cigital

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Make the system dynamic.

**Email:** ccmich@rstcorp.com

**Phone:** (612) 951-7333

**URL:** http://www.cigital.com/research/trigger.html

**Technology Readiness Level estimate for December 2002:** 3

**7**

**Project Title:** CIRCADIA: Automatically Synthesizing Security Control Systems

**Contact:** David J. Musliner, Honeywell

**Objective:** Demonstrate the feasibility of responding to automated computer attacks with adaptive dynamically generated real-time reactive controllers that are appropriate for the current state of the target system, and the level and type of threat, while satisfying timing and resource constraints.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Make the system dynamic.

**Email:** musliner@htc.honeywell.com

**Phone:** (612) 951-7599

**URL:** http://www.htc.honeywell.com/projects/circadia

**Technology Readiness Level estimate for December 2002:** 4

**8**

**Project Title:** Cyber Mission Interpretation Tool

**Contact:** David Levin, BBN

**Objective:** Explore ability to automatically translate the commander's operational planning and execution environment information into a prioritized set of cyber support requirements. Explore ability to automatically translate the current status of the cyber environment back into mission impact information.

Explore applying innovative concepts, such as an adaptive feedforward control loop and temporal translation of operational mission information into cyber requirements.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** mwilcomb@bbn.com

**Phone:** (703) 284-4779

**URL:** https://archive.ia.isotic.org/dscgi/ds.py/View/Collection-422

**Technology Readiness Level estimate for December 2002:** 4

**9**

**Project Title:** DASSA: Distributed Active Security Situation Assessment

**Contact:** Larry Clough, IET

**Objective:** Provide an extensible, distributed, active situation assessment associate by proactively monitoring through IDS sensors the current security status of multiple cooperating enclaves or networks, controlling selected (existing) network and host-based intrusion detection sensors, formulating situational hypotheses in light of external indicators and warnings.

Relate those hypotheses to the enclaves' ability to accomplish high-level objectives and display human-understandable representations of the situational information.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** lclough@teknowledge.com

**Phone:** (703) 353-9300 x211

**URL:** http://www.dc.teknowledge.com/external/DASSA/index.html

**Technology Readiness Level estimate for December 2002:** 4

**10**

**Project Title:** Data Mining Approach for Building Cost-Sensitive ID

**Contact:** Wenke Lee, NC State

**Objective:** Develop automated techniques for building cost-sensitive models that are optimized for user-defined cost metrics.

Design a system architecture for dynamically activating and configuring light intrusion detection modules that each specializes for a set of similar intrusions.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** wenke@csc.ncsu.edu

**Phone:** (919) 513-3506

**URL:** http://www.csc.ncsu.edu/faculty/lee/project/id.html

**Technology Readiness Level estimate for December 2002:** 4

**11**

**Project Title:** EBCOTE: Effects-Based CyberCOA Optimization Technology & Experiments

**Contact:** John Shaw, Alphatech

**Objective:** Help a cyber-commander formulate and select Courses of Action (COAs) that balance security objectives against mission effectiveness. Once selected, a COA will be continuously extended into new COAs as new information arrives from Situation Assessment components.

Each COA will consist of a sequence of process-level actions (e.g., kill, restart, reset, launch, or relocate) applicable to both mission applications and a set of security functions (e.g., authenticate, filter, validate, deceive).

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Make the system dynamic.

**Email:** john.shaw@alphatech.com

**Phone:** (781) 273-3388 (x219)

**URL:** http://www.alphatech.com

**Technology Readiness Level estimate for December 2002:** 3

**12**

**Project Title:** IA Cyber Ecology

**Contact:** Jane Jorgensen, IET

**Objective:** Composable trust: functionality and performance as a whole may be maintained by monitoring and managing system health.

Enhance abilities to describe attack agents and predict changes in network structure and behavior upon attack.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** jjorgensen@iet.com

**Phone:** (541) 752-7473 ext. 204

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 3

**13**

**Project Title:** IA Reliability Model

**Project Title:** IA Reliability Model

**Contact:** Ms. Roberta Gotfried, Raytheon

**Objective:** Provide methods and tools for specifying the desired IA and survivability characteristics of a target system, as well as a corresponding methodology for assessing the IA robustness of a given design or system. In particular, address system specification, design, and assessment as they relate to producing systems with a desired level of IA and survivability.

**Survivability focus:** Survivability specification.

**Survivability principles/techniques applied/explored:** Specify survivability requirements.

**Email:** rlgotfried@west.raytheon.com

**Phone:** (310) 334-7655

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 3

**14**

**Project Title:** Intelligent Visualization System for Situation and Course of Action Understanding

**Contact:** Allen Ott, Orincon

**Objective:** Apply the science of the Law of Requisite Variety, game theory, control theory, and intelligent software agents to control the IA environment.

**Survivability focus:** Survivability modeling.

**Survivability principles/techniques applied/explored:** Monitor the system; make the system dynamic.

**Email:** aott@orincon.com

**Phone:** (858)775-0001

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 4

**15**

**Project Title:** Internet Trap & Trace

**Contact:** Stuart Staniford, Silicon Defense

**Objective:** Develop a capability to trace the true location of attackers who attempt to disguise their activities by logging in through a chain of hosts.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** stuart@silicondefense.com

**Phone:** (707) 445-4355

**URL:** http://www.silicondefense.com/itrex

**Technology Readiness Level estimate for December 2002:** 4

**16**

**Project Title:** IPIB: Intelligence Preparation of the Information Battlespace

**Contact:** Ken Williams, ZEL Tech

**Objective:** Demonstrate an innovative means of detecting large-scale information attacks and performing IA situation assessment; achieve improved IA situation assessment, sensor placement, alerting, and inputs to incident response and recovery applications.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** kwilliams@zeltech.com

**Phone:** (757) 722-5565

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 6

**17**

**Project Title:** Multi-Community Cyber Defense

**Contact:** Randy Smith, Boeing

**Objective:** Develop intrusion correlation techniques and tools that scale up to regional and national levels; provide a detection and response system that can survive component failure and system-level attack; develop a trust model for IDR across disjoint administrative domains; develop techniques for assessing trust; develop the capabilities required for survivable, cooperating IDR systems across organizational boundaries.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system; make the system dynamic

**Email:** Randall.Smith@PSS.Boeing.com

**Phone:** (253) 657-2787

**URL:** http://seclab.cs.ucdavis.edu/mccd

**Technology Readiness Level estimate for December 2002:** 5

**18**

**Project Title:** Network Attack Detection

**Contact:** Jack May, TRW

**Objective:** Improve attack detection through use of avalanche and message pattern detection.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** jack.may@trw.com

**Phone:** (408) 743-6112

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 4

**19**

**Project Title:** PROPHETEER-Continuous Adversarial Planning

**Contact:** Tamitha Carpenter, Stottler-Henke

**Objective:** Create a Predictive Planning and Preemption (P3) system suitable for rapidly developing and executing adaptive large-scale cyber defense strategies.

Operate with incomplete and uncertain information including the likelihood of deception; continuously adapt COAs in the face of an adaptive threat, resulting reflexive defenses, as well as changes in mission and IA posture.

Recognize that adversarial engagements are completely unstructured, making even the most modern game theoretic approaches difficult to apply.

Ensure timely COA development to enable execution to occur in time for the actions to have the desired effect.

Balance planning and counter-planning by integrating predictive models of the adversary, with the overall goal of controlling recognized threats.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Make the system dynamic.

**Email:** tamitha@shai-seattle.com

**Phone:** (206) 545-1478

**URL:** http://www.shai.com/projects/cc2.htm

**Technology Readiness Level estimate for December 2002:** 3

**20**

**Project Title:** System Health and Intrusion Monitoring (SHIM): A New Approach to Triggering Intrusion Tolerant Mechanisms.

**Contact:** Calvin Ko, NAI Labs

**Objective:** Continuous monitoring of the health of a system by identifying system anomalies that could evolve into security compromises.

Detect novel attacks with a low false alarm rate.

Furnish strategic information on detected system anomalies to assist response analysis.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** calvin_ko@nai.com

**Phone:** (408) 346-3783

**URL:** http://www.pgp.com/research/nailabs/secure-execution.asp

**Technology Readiness Level estimate for December 2002:** 4

**21**

**Project Title:** Constellation: A Scalable Metrology to Support Theory and Practice of Anomalous Event Detection

**Contact:** Roy Maxion, CMU

**Objective:** Develop a basic science of anomaly detection and profiling; develop a diverse suite of anomaly detectors; provide custom, calibrated testbeds; provide statistically and methodologically rigorous assessment procedures.

**Survivability focus:** Survivability assessment.

**Survivability principles/techniques applied/explored:** Monitor the system; assess survivability.

**Email:**

**Phone:**

**Technology Readiness Level estimate for December 2002:**

**22**

**Project Title:** STAT: State Transition Analysis Technique

**Contact:** Richard Kemmerer, UC Santa Barbara

**Objective:** Model-based real-time intrusion detection system; scenario-based intrusion detection system development.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:** stat@cs.ucsb.edu

**Phone:**

**URL:** http://www.cs.ucsb.edu/~kemm/STAT/

**Technology Readiness Level estimate for December 2002:** 6

**23**

**Project Title:** NETFLARE: Network Fuzzy Logic Attack Recognition Engine

**Contact:** Chet Hosmer, Silicon Defense, Inc.

**Objective:** Mission/situation-based policy construction; policy-based risk analysis, situation assessment, and IDS decision support.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:**

**Phone:**

**Technology Readiness Level estimate for December 2002:** 4

**24**

**Project Title:** Global Guard

**Contact:** Karl Levitt, UC Davis

**Objective:** Model-based real-time intrusion detection system; scenario-based intrusion detection system development; attack response.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system.

**Email:**

**Phone:**

**Technology Readiness Level estimate for December 2002:** 4

**25**

**Project Title:** Correlated Attack Modeling

**Contact:** Ulf Lindqvist, SRI International

**Objective:** generalized models of composite attacks; correlation of multiple detector outputs; correlated attack specification language

**Survivability focus:** Survivability management

**Survivability principles/techniques applied/explored:** Monitor the system

**Email:** ulf@sdl.sri.com

**Phone:**

**Technology Readiness Level estimate for December 2002:** 4

**26**

**Project Title:** Assessing Strategic Intrusions

**Contact:** Phillip A. Porras, SRI International

**Objective:** Create an alarm management infrastructure to enable consolidated views of host and network operations from distributed (possibly remote) service centers; develop advanced mission-based techniques in alarm aggregation, correlation, prioritization, equivalence recognition, and complex scenario recognition.

**Survivability focus:** Survivability management.

**Survivability principles/techniques applied/explored:** Monitor the system

**Email:** porras@sdl.sri.com

**Phone:**

**URL:** http://www.sdl.sri.com/intrusion/

**Technology Readiness Level estimate for December 2002:** 4

**27**

**Project Title:** CyberAIM: Cyber Operations Center Study

**Contact:** Roshan Thomas, NAI Labs

**Objective:** Collect data from large commercial NOCs, ISPs on network monitoring, attack analysis and response; emphasis on understanding challenging problems in real-world, high-performance settings; Emphasis on real-world needs driving information analysis and visualization techniques for decision makers.

**Survivability focus:** Survivability requirements

**Survivability principles/techniques applied/explored:** Determine real-world

survivability requirements

**Email:** rthomas@nai.com

**Phone:**

**Technology Readiness Level estimate for December 2002:**

**28**

**Project Title:** Active Response Technology

**Contact:** Michael Winburn, Modus Operandi

**Objective:** Take a proactive defensive posture to intercept, track, redirect and respond to network-based attacks; create a virtual network of services to provide a framework for intrusion data collection, analysis, and response; develop and adapt advanced techniques to identify the identity and goals of the intruder; provide an informative view of an intruder's actions

**Survivability focus:** Survivability management

**Survivability principles/techniques applied/explored:** Monitor the system; make the system dynamic

**Email:** mwinburn@modusoperandi.com

**Phone:**

**Technology Readiness Level estimate for December 2002:** 3

**29**

**Project Title:** ASTER: Active Smart Targets for Effective Response

**Contact:** Frank Adelstein, Odyssey Research Associates

**Objective:** Develop active approach to intrusion detection: Active Smart Targets (ASTs) provide marked cards (marked information) to probes during reconnaissance; get marked cards back during an attack; increase effectiveness of log file information by "marking" the data that the probe sees; perform correlation to link probe and attacker; mislead and misdirect attacker with marked information; effectively focus response, such as router or firewall reconfiguration.

**Survivability focus:** Survivability management

**Survivability principles/techniques applied/explored:** Monitor the system; make the system dynamic; deception

**Email:** fadelstein@oracorp.com

**Phone:**

**Technology Readiness Level estimate for December 2002:** 3

**30**

**Project Title:** Visual Representation of Cyber Defense Situational Awareness

**Contact:** Anita D'Amico, Secure Decisions

**Objective:** Implement visualization aids to the discovery and analysis of time patterns in cyber security breaches; implement visualization aids to understanding the impact of cyber security breaches on mission-critical tasks; develop methods for easily interfacing visualization aids to most database schema containing temporal & mission impact data; speed IA analysts' access to information about the progression, sequence and time urgency of an impending cyber attack; improves speed of comprehending the impact of cyber threats to critical missions; improve maintenance of critical mission operations in the presence of cyber threats

**Survivability focus:** Survivability management

**Survivability principles/techniques applied/explored:** Monitor the system

**Email:** AnitaD@avi.com

**Phone:**

**URL:** www.SecureDecisions.com

**Technology Readiness Level estimate for December 2002:** 5

**31**

**Project Title:** SARA: Survivable Autonomic Response Architecture

**Contact:** Joshua Haines, MIT Lincoln Laboratory

**Objective:** Focus on enabling autonomic responses with fast, reliable, secure communication between Information Assurance components

**Survivability focus:** Survivability management

**Survivability principles/techniques applied/explored:** Make the system dynamic

**Email:** scl@sst.ll.mit.edu

**Phone:** 781-981-4337

**Technology Readiness Level estimate for December 2002:** 4

**32**

**Project Title:** SSARE: Security Situation Assessment and Response Evaluation

**Contact:** Suzanne Mahoney, Information Extraction & Transport, Inc.

**Objective:** Develop a mixed-initiative system that can detect a large-scale attack, display assessment of the situation, and identify effective responses; combine probabilistic domain models with uncertain data into a situation-specific model

**Survivability focus:** Survivability management

**Survivability principles/techniques applied/explored:** Monitor the system; make the system dynamic

**Email:** suzanne@iet.com

**Phone:**

**Technology Readiness Level estimate for December 2002:** 4

**33**

**Project Title:** Cyber Panel Modeling and Simulation

**Contact:** Joshua Haines, MIT Lincoln Laboratory

**Objective:** Learn to predict how intrusion detection and correlation systems respond to attacks and normal traffic in a theatre-wide IW scenario; generate alert and event streams data to support detailed grand challenge problem simulations and to use as input for real and modeled intrusion detection, correlation, situational awareness or COA systems

**Survivability focus:** Survivability modeling and simulation

**Survivability principles/techniques applied/explored:** Monitor the system

**Email:** jhaines@sst.ll.mit.edu

**Phone:** 781-981-4337

**Technology Readiness Level estimate for December 2002:**

**34**

**Project Title:** Strategic Attack Scenarios

**Contact:** Raymond Parks, Sandia National Laboratories

**Objective:** Develop an abstract mathematical view of attack and scenario spaces; develop a non-trivial relation on the attack space that yields 'equivalence classes'; develop a mapping between attack and scenario space that induces a non-trivial relation on scenario space that yields 'equivalence classes'; extract features for canonical attacks and canonical scenarios

**Survivability focus:** Survivability management

**Survivability principles/techniques applied/explored:** Monitor the system

**Email:** rcparks@sandia.gov

**Phone:** 505-844-4024

**Technology Readiness Level estimate for December 2002:**

## E.4 SWWIM Program

**1**

**Project Title:** Autonomix: Component, Network, and System Autonomy

**Contact:** Crispin Cowan, WireX Communications, Inc.

**Objective:** Use family of tools to guard components against common software

vulnerabilities: StackGuard: protection from "stack smashing" buffer overflows, **SubDomain:** lightweight mandatory access controls, PointGuard: generalized **StackGuard, FormatGuard:** protection from printf format bugs, RaceGuard: protection from temp file races, with the objective to eliminate 90-99% of software vulnerabilities.

**Survivability focus:** Provide light autonomic defenses, response mechanisms, and response selection techniques.

**Survivability principles/techniques applied/explored:**

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 6

**2**

**Project Title:** Advanced Mathematical Control and Identification Techniques for Autonomic Information Assurance

**Contact:** Dr. Edmond Jonckheere, University of Southern California

**Objective:** Investigate stochastic denial-of-service detection, denial-of-service mitigation, dynamic traffic modeling, robust and adaptive transport, robust and adaptive routing, worm defense, network hyperbolic geometry.

**Survivability focus:** Provide light autonomic defenses, modeling, response selection, and state estimation.

**Survivability principles/techniques applied/explored:** Modeling and simulation techniques.

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 3

**3**

**Project Title:** Continuous Experimentation for AIA

**Contact:** Kenneth Theriault, William Nelson, BBN Technologies

**Objective:** Formulate and execute a program of continuous, science-based experimentation.

**Survivability focus:** Experimentation.

**Survivability principles/techniques applied/explored:** Focus on automated experimentation for rapid exploration of issues and behaviors, such that experiments are well-controlled, repeatable, and cost-effective; develop a suite of automated experimentation tools with traffic generator (Skaion), automated attack simulator, and experimentation control and execution GUI/workstation; red team still essential for experiment formulation.

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 4

**4**

**Project Title:** Autonomic Distributed Firewalls

**Contact:** Tom Markham, Secure Computing Corporation

**Objective:** Provide robust, intrusion-tolerant networks via a firewall per host; provide defense in depth; provide protection from insiders; tie distributed firewall to autonomic response mechanisms.

**Survivability focus:** Provide response mechanisms.

**Survivability principles/techniques applied/explored:** Push the firewall closer to, but not on to, the host; create a "firewall" on a Network Interface Card (NIC) that is independent from the host; use a master-slave architecture to provide scalability and centralized security policy management.

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 4

## E.5 Fault-Tolerant Networks Program

**1**

**Project Title:** Adaptive Probabilistic Tools for Advanced Networks/Spinglass: Assuring the Integrity of Highly Decentralized Communication Systems

**Contact:** Ken Birman, Cornell University

**Objective:** Investigate approaches that allow distributed applications to scale, while strictly controlling the resources required; develop techniques for secure, scalable, and reliable operation; and implement and distribute software protocols (toolkit), infrastructure services support, and application development support.

**Survivability focus:** Provide a survivable network, thus survivable applications in certain cases.

**Survivability principles/techniques applied/explored:** Spinglass offers probabilistic guarantees, for example, by ensuring that "almost every" component of a large system will behave in a desired way; Spinglass involves the development of formal methods for automatically proving that a distributed protocol achieves a desired goal, or has a desired security property.

**URL:** http://www.cs.cornell.edu/Info/Projects/Spinglass/Spinglass-main.html

**Technology Readiness Level estimate for December 2002:** 5

**2**

**Project Title:** Scalable and Survivable Data Replication

**Contact:** Michael Reiter, Lucent

**Objective:** Achieve Byzantine fault-tolerance with efficient access, load balancing, and scalability of quorum-based access; build a survivable and scalable object store called Fleet, where Fleet provides the abstraction of persistent Fleet objects that

distributed applications can create and use to communicate and to coordinate distributed activities; Fleet objects are made persistent and highly available via their replication across a collection of Fleet servers; Fleet provides flexibility to applications.

**Survivability focus:** Provide improved broad survivability.

**Survivability principles/techniques applied/explored:**

**URL:** http://www.bell-labs.com/user/reiter/fleet/index.html

**Technology Readiness Level estimate for December 2002:** 5

**3**

**Project Title:** Protecting Network Quality of Service against Denial of Service Attacks

**Contact:** Doug Reeves, North Carolina State University

**Objective:** Protect network QoS against denial-of-service attacks

**Survivability focus:** Identify rogue or compromised routers that deviate from contracted behavior.

**Survivability principles/techniques applied/explored:** Solution has three components: (1) price network resources based on demand; (2) add per-flow traffic monitoring and intrusion detection capabilities to DiffServ; (3) protect the integrity of QoS signaling by using end-to-end authentication.

**URL:** http://www4.ncsu.edu/eos/users/r/reeves/rtcomm/ARQOS/main.htm

**Technology Readiness Level estimate for December 2002:** 4

**4**

**Project Title:** TBDS: Topology Based Domain Search

**Contact:** Bill Manning, University of Southern California Information Sciences Institute

**Objective:** Provide topology-based domain search.

**Survivability focus:** Allow continued DNS operation during transient breaks in communications connectivity of subnetworks, and discover and validate DNS resources when their availability or accessibility is restored.

**Survivability principles/techniques applied/explored:** Software enhancements to DNS code, based on the Internet Software Consortium's BIND.

**URL:** http://www.isi.edu/tbds

**Technology Readiness Level estimate for December 2002:** 7

**5**

**Project Title:** APOD: Applications that Participate in their Own Defense

**Contact:** Franklin Webber, BBN

**Objective:** Formulate response strategies to attacks that threaten survival of

applications and organize response mechanisms around a middleware infrastructure; facilitate the construction of defense-enabled applications.

**Survivability focus:** Provide broad survivability and response strategies.

**Survivability principles/techniques applied/explored:** Project's approach to improving system survivability despite intrusion, malicious attack, and failures is to use adaptable mechanisms; in this approach, an application is developed to be aware of and adapt to changing conditions in the environment in which it is running; the hypothesis is that relatively simple application-level adaptation, based on a modest selection of defensive strategies, can result in dramatic improvement in software survivability under attack.

**URL:** http://www.dist-systems.bbn.com/projects/APOD

**Technology Readiness Level estimate for December 2002:** 4

**6**

**Project Title:** Lighthouse: Detecting and Surviving Large-Scale Network Infrastructure Attacks

**Contact:** Farnam Jahanian, University of Michigan

**Objective:** Develop networks that can survive attacks, prototype capabilities in MichNet, and use fine and coarse-grained instrumentation tools.

**Survivability focus:** Provide more resilient network.

**Survivability principles/techniques applied/explored:** Develop a distributed detection and response system for global infrastructure survivability and assurance.

**URL:** http://www.eecs.umich.edu/lighthouse

**Technology Readiness Level estimate for December 2002:** 6

**7**

**Project Title:** Providing Survivable Real-Time Communication Service for Distributed Mission Critical Systems

**Contact:** Wei Zhoa, Texas A&M

**Objective:** Provide survivable real-time communication service for distributed mission-critical systems.

**Survivability focus:** Provide traffic stuffing to mask actual channel use without compromising quality of service, and provide intrusion detection and suppression that trigger in real time—countermeasures and rerouting of messages would guarantee delivery within deadlines.

**Survivability principles/techniques applied/explored:** Apply traffic modeling techniques in network security.

**URL:** http://netcamo.cs.tamu.edu

**Technology Readiness Level estimate for December 2002:**

**8**

**Project Title:** Active Network Intrusion Detection Response

**Contact:** Dan Sterne, NAI Labs

**Objective:** Provide better intrusion detection and response (IDR) capabilities via Active Network technology by better scanning, detection, traceback, response, repair for routers, firewalls, switches, hosts, and enable networks to become self-protecting.

**Survivability focus:** Provide IDR capabilities to counter intrusions, denial-of-service attacks, add new IDR capabilities, e.g., new responses, and add survivability by increasing attack tolerance, e.g., by moving away from adversary to avoid flooding attacks.

**Survivability principles/techniques applied/explored:**

**URL:** http://www.pgp.com/research/nailabs/adaptive-network/active-networks.asp

**Technology Readiness Level estimate for December 2002:** 6

**9**

**Project Title:** A Cost-Benefit Approach to Fault-Tolerant Communication and Information Access

**Contact:** Baruch Awerbuch, Yair Amir, Johns Hopkins University

**Objective:** Develop a cost-benefit framework to withstand strong network attacks, while providing theoretically provable QoS performance, and develop software to implement this framework.

**Survivability focus:** Use economic principles to equate various resources to a common base, and optimize defense resources.

**Survivability principles/techniques applied/explored:** Analysis of strong adversary models; new routing and dissemination protocols; new replication protocol; cost-benefit decision framework; an overlay network architecture.

**URL:** http://www.cnds.jhu.edu/funding/tolerant_networks/

**Technology Readiness Level estimate for December 2002:** 3

**10**

**Project Title:** Better Fault Tolerance via Application-Enhanced Networks

**Contact:** John Hartman, University of Arizona

**Objective:** Develop local resource management for active routers and construct application-enhanced networks with application-driven rerouting and data migration.

**Survivability focus:** Protect against denial of service and rogue application clients.

**Survivability principles/techniques applied/explored:** Local resource management; distributed terrain navigation; network-resident storage.

**URL:** http://www.cs.arizona.edu/ftn/

**Technology Readiness Level estimate for December 2002:** 4

**11**

**Project Title:** Building Secure and Reliable Networks through Robust Resource Scheduling

**Contact:** Larry Peterson, Princeton University

**Objective:** Unify solutions from the security and fault-tolerance communities to find solutions for denial of service and robustness; unified approach will produce completeness and generality.

**Survivability focus:** Provide network layer survivability.

**Survivability principles/techniques applied/explored:** Apply a unified set of mechanisms and algorithms to the problem of protecting a networked system from both failures and DoS attacks; view problem as a matter of resource allocation and management.

**URL:** http://www.cs.princeton.edu/nsg/

**Technology Readiness Level estimate for December 2002:** 6

**12**

**Project Title:** FNIISC: Fault-Tolerant Networking Through Intrusion Identification and Secure Compartments

**Contact:** S. Felix Wu, Lixia Zhang, North Carolina State University, University of California Los Angeles

**Objective:** Demonstrate how to partition the network into autonomous components to allow any component to fail without affecting entire network.

**Survivability focus:** Provide more robust networks that can tolerate attacks, such that attacks are confined in a secure compartment.

**Survivability principles/techniques applied/explored:**

**URL:** http://fniisc.east.isi.edu

**Technology Readiness Level estimate for December 2002:** 6

**13**

**Project Title:** Control Mechanisms to Prevent Maliciously Induced Network Instability

**Contact:** Ronald Skoog, Telcordia

**Objective:** Prevent networks from becoming unstable due to the propagation of system failures that are resultant from the triggering by system defects or attacks.

**Survivability focus:** Improve survivability by isolating failures caused by information warfare attacks or natural causes.

**Survivability principles/techniques applied/explored:**

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 6

**14**

**Project Title:** Fault Tolerant Network Protocols

**Contact:** Louise Moser, University of California Santa Barbara

**Objective:** Strengthen communication networks by providing reliable message delivery from source to destination despite the presence of malicious nodes in the network.

**Survivability focus:** Provide a protocol mechanism for fault-tolerant applications where objects are replicated to maintain consistency of the states of the replicas by delivering messages reliably and in the same causal and total order, and provide interoperability between different applications and different fault-tolerant infrastructures.

**Survivability principles/techniques applied/explored:** Fault-Tolerant Multicast Protocol (FTMP) is a multicast protocol being developed to provide reliable and efficient operation over the Internet; FTMP will be strengthened, so that it can resist malicious nodes in the network.

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 5

**15**

**Project Title:** SPIE: Source Path Isolation Engine

**Contact:** Luis Sanchez, BBN

**Objective:** Develop network packet traceback capability without significant overhead to routers.

Develop a network-wide Source Path Isolation Engine (SPIE) that can reliably trace the source of an attack back to its ingress point on a particular Autonomous System (AS) soon after the attack; every router in the network is instrumented with special software to record and cache a digest of every packet-forwarding event within the router; when an attack is detected by an intrusion detection system (commercial systems are now available), the attacking packet can be reverse traced by querying routers that may have seen the packet.

**Survivability focus:** Provides more resilient network and ability to track the source of attacks.

**Survivability principles/techniques applied/explored:** (1) Develop a system capable of isolating the source of a network intrusion attack by first defining the packet digest processing requirements for the infrastructure routers; these algorithms will reduce memory, CPU, and internal router bandwidth requirements; (2) define SPIE query messages and reply formats to carry messages across the ISP's network; specify trace-back algorithms and processing techniques for the SPIE servers; implement these algorithms and protocols in a commercial terabit router to test the entire system.

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 6

**16**

**Project Title:** RON: Private Resilient Overlay Networks

**Contact:** Frans Kaashoek , MIT

**Objective:** Provide applications with a new level of control over network quality and reliability; need for this control is highlighted by two recent developments: (1) the Internet routing system has been shown to react slowly to failures and frequently to choose needlessly low-quality paths; (2) denial-of-service attacks have revealed the difficulty of managing the Internet when it is under stress; these problems stem from the Internet's architecture as a large collection of loosely coupled and mutually-distrusting peer networks; RON will layer a new routing architecture over the Internet that will enable real-time collaboration and fault-resistant applications.

**Survivability focus:**

**Survivability principles/techniques applied/explored:** Key new idea in the RON project is a virtual application-level network overlaid on the Internet, consisting only of sites collaborating for a particular purpose.

**URL:** http://nms.lcs.mit.edu/DARPA/ron/

**Technology Readiness Level estimate for December 2002:** 5

**17**

**Project Title:** Fault Tolerant Internetworking

**Contact:** J. J. Garcia-Luna-Aceves, University of California Santa Cruz

**Objective:** Develop protocols that can protect, detect, and respond to attacks.

**Survivability focus:** Provide more resilient network.

**Survivability principles/techniques applied/explored:** Trust algebra for access control with delegation; fault-tolerant, secure internetworking; efficient authentication of routing updates; fault-tolerant QoS guarantees.

**URL:** http://www.cse.ucsc.edu/research/ccrg/ftn.html

**Technology Readiness Level estimate for December 2002:** 3

**18**

**Project Title:** TIARA: Techniques for Intrusion-Resistant Ad Hoc Routing Algorithms

**Contact:** Ranga Ramanujan, Architecture Technology Corporation

**Objective:** Develop general design techniques, collectively called TIARA, for protecting ad hoc networks against DOS attacks and demonstrate effectiveness of TIARA to sustain continued network operation despite intrusion-induced DOS attacks.

**Survivability focus:** Provide several types of protection against spurious traffic, packet replay, session flooding, flow disruption, and route hijacking.

**Survivability principles/techniques applied/explored:** An ad hoc network is a collection of wireless nodes that can be rapidly deployed as a multi-hop packet radio network without the aid of any established infrastructure or centralized administration and without any user-initiated configuration actions.

**URL:** http://www.atcorp.com/research/tiara

**Technology Readiness Level estimate for December 2002:** 4

**19**

**Project Title:** Active Network Fault Response

**Contact:** Sandra Murphy, NAI Labs

**Objective:** Enable active networks to tolerate faults, where the faults are active faults, unique to active networks, and security infrastructure faults.

**Survivability focus:** Provide fault tolerance in active networks, giving robust authentication and fault elimination.

**Survivability principles/techniques applied/explored:** Investigate mechanisms to perform fault passivation of faulty active code by searching the network for identified faulty active code and expunging that fault.

**URL:** http://www.pgp.com/research/nailabs/network-security/active-networks-fault.asp

**Technology Readiness Level estimate for December 2002:** 3

**20**

**Project Title:** Fault and Attack Management in Optical Networks

**Contact:** Hyeong-Ah Choi, George Washington University

**Objective:** Develop fault and attack models that can be used in designing robust optical networks; design fault and attack detection and localization algorithms; design protection and distributed restoration strategies; incorporate the developed methods in an open and modular simulation environment.

**Survivability focus:**

**Survivability principles/techniques applied/explored:** Provide fundamental performance limits and a cost-performance trade-off analysis for fault- and attack-tolerant network design; practical algorithms for fault and attack detection and localization, protection, and traffic restoration will be developed; graph-theoretic and probabilistic techniques will be employed in the design and evaluation of the algorithms.

**URL:** http://www.seas.gwu.edu/~fam/

**Technology Readiness Level estimate for December 2002:** 4

**21**

**Project Title:** Enforceable Network Protocols

**Contact:** Tom Anderson, Stefan Savage, University of Washington

**Objective:** Improve the robustness of TCP/IP-based network protocols, and add IP traceback functionality, including anonymous packets.

**Survivability focus:** Increase survivability by adding robustness to TCP/IP and other protocols, if needed, and provide source traceback, which is useful in countering DoS and other attacks.

**Survivability principles/techniques applied/explored:** Enforceable interfaces; dependable transient behavior; virtual edge services; network-level resource containers.

**URL:** N/A

**Technology Readiness Level estimate for December 2002:** 4

**22**

**Project Title:** Advanced Security Proxies

**Contact:** Stephen Schwab, NAI Labs

**Objective:** Develop an approach for using firewall security proxies in conjunction with high-speed networks.

**Survivability focus:**

**Survivability principles/techniques applied/explored:**

**URL:** http://www.pgp.com/research/nailabs/distributed/advanced-security.asp

**Technology Readiness Level estimate for December 2002:** 6

**23**

**Project Title:** Denial-of-Service Attack Assessment

**Contact:** Donna Gregg, Johns Hopkins University – Applied Physics Laboratory

**Objective:**

**Survivability focus:**

**Survivability principles/techniques applied/explored:**

**Technology Readiness Level estimate for December 2002:** 3

## E.6 Information Assurance Program

**1**

**Project Title:** Security Assurance Navigation and Environment (SANE)

**Contact:** Judith Froscher, NRL

**Objective:** Build the enterprise assurance argument map. This map depicts the

claim trees for the assurance arguments related to the enterprise security objective, providing causality, relationships, vulnerabilities, threats, and other system- and environment-related issues.

**Approach:**

Develop a methodology, Enterprise Certification Methodology (ECM), to derive and organize the related assurance arguments effectively.

Develop a visual language, Composite Assurance Mapping Language (CAML), to build the map of the assurance argument using ECM.

Develop tools, Visual Network Rating Methodology (VNRM) and Security Assurance Navigation and Environment (SANE), to help users develop a map to assurance arguments in CAML based on ECM and document it with related descriptions in a common environment.

Apply the above approaches to real systems.

**URL:** froscher@itd.nrl.navy.mil

**Technology Readiness Level estimate for December 2002:** 5

# APPENDIX F. Acronym List

| | |
|---|---|
| 1GS | first-generation security |
| 2GS | second-generation security |
| 3GS | third-generation security |
| AIP | Antisurface warfare Improvement Program |
| API | application programming interface |
| ASW | antisubmarine warfare |
| BGP | Border Gateway Protocol |
| C4/ISR | C4I, Surveillance, and Reconnaissance |
| C4I | command, control, communications, computers, and intelligence |
| CAML | Composite Assurance Mapping Language |
| CIM | Client Interface Manager |
| COA | course of action |
| COP | Common Operational Picture |
| COTS | commercial off-the-shelf |
| CTAPS | Contingency Theater Automated Planning System |
| CVE | Common Vulnerabilities and Exposures |
| DARPA | Defense Advanced Research Projects Agency |
| DCM | Data Collection Manager |
| DEN | Directory-Enabled Networking |
| DHCP | Dynamic Host Configuration Protocol |
| DII COE | Defense Information Infrastructure Common Operating Environment |
| DMS | Defense Message System |
| DMTF | Distributed Management Task Force |
| DNS | Domain Name System |
| DNSSEC | DNS Security |
| DoD | Department of Defense |
| DPM | Data Processing Map |
| DRA | Detection & Response Argument |
| EAI | Enterprise Application Integration |
| ELINT | electronic intelligence |
| FRS | Fragmentation, Redundancy, and Scattering |
| GCCS-M | Global Command and Control System – Maritime |
| GIG | Global Information Grid |
| IA | information assurance |
| ICAT | Internet Categorization of Attacks Toolkit |
| IDR | intrusion detection and response |
| IDS | intrusion detection system |
| IP | Internet Protocol |
| IPSEC | IP Security |
| ISDS | Intelligence Support Data Services |
| IT | information technology |

| | |
|---|---|
| ITS | Imagery Transformation Services |
| JSTARS | Joint Surveillance and Target Attack Radar System |
| LAN | local area network |
| METOC | Meteorological and Oceanographic |
| MIDB | Modernized Intelligence Data Base |
| MPA | Maritime Patrol Craft |
| NMCI | Navy-Marine Corps Intranet |
| OASIS | Organically Assured and Survivable Information System |
| OTH | Over-the-Horizon |
| PA | Prevention Arguments |
| PC | personal computer |
| PK | public key |
| PKI | Public Key Infrastructure |
| QoS | Quality of Service |
| RA | Recovery Arguments |
| ROM | read-only memory |
| S/MIME | Secure Multipurpose Internet Mail Exchange |
| SANE | Security Assurance Navigation and Environment (formerly ARGUS) |
| SBGP | Secure BGP |
| SIPRNET | Secret Internet Protocol Router (IPR) Network |
| SNMP | Simple Network Management Protocol |
| SSH | Secure Shell |
| STANAG | Standardisation Agreement |
| SWWIM | Survivable Wired and Wireless Infrastructure for Military Operations |
| TCB | Trusted Computing Base |
| TCP | Transmission Control Protocol |
| TDBM | Track Data Base Manager |
| TSC | Tactical Support Center |
| U.S. | United States |
| UCP | Universal Communications Processor |
| USMFT | United States Message Text Format |
| USN | United States Navy |
| VNRM | Visual Network Rating Methodology |
| VPN | virtual private network |

Filename: 0108030_SGIG_Master_Final_Cover_dc_10_16.doc
Directory: C:\program files\qualcomm\eudora\attach
Template: C:\Documents and Settings\Administrator\Application Data\Microsoft\Templates\Normal.dot
Title: Fjfjkdfjdjdkfjkjkjf
Subject:
Author: MITRE User
Keywords:
Comments:
Creation Date: 10/16/2001 9:12 AM
Change Number: 2
Last Saved On: 10/16/2001 9:12 AM
Last Saved By: croakes
Total Editing Time: 0 Minutes
Last Printed On: 12/6/2001 3:30 PM
As of Last Complete Printing
Number of Pages: 116
Number of Words: 39,350 (approx.)
Number of Characters: 224,300 (approx.)