National Scale INFOSEC Research Hard Problems List[1]

James P. Anderson (James P. Anderson and company), Stephen T. Kent (BBN),
Steven B. Lipner (Mitretek Systems), Robert V. Meushaw (National Security Agency)

Executive Summary

This document presents a list of "hard problems" that pose obstacles to the abilities of United States Government IT users to process sensitive information securely. The definition of security that guided the development of this list encompasses data confidentiality and integrity as well as the availability of information and processing resources. The sensitivity of the information requiring protection ranges from routine business information to information whose modification or disclosure could result in major financial loss or loss of life. The threats to the confidentiality, integrity, and availability of information similarly run the gamut from hackers executing scripts downloaded from the Internet to national governments and major criminal enterprises.

This document was developed at the request of the INFOSEC Research Council (IRC) whose members (DoD [including DARPA, NSA, OSD, Army, Navy, and Air Force], NIST, DoE, CIA) are the major government sponsors of research in information security. The "hard problems list" is intended to guide the research program planning of the IRC members by identifying the key problems whose solution would remove major obstacles to effective information security. It may also be useful to policy makers and planners in evaluating the contributions of ongoing and proposed research programs to the critical INFOSEC problems facing the nation.

The IRC is sponsoring an Information Security Technology Studies Group (ISTSG) study that will develop a twenty-year vision of Information Assurance (IA), and a roadmap to progress toward achieving that vision. This document will serve as an input to the ISTSG as it develops its roadmap. The ISTSG will also check its IA vision against this list of hard problems to ensure that the vision addresses the impact of solving (or failing to solve) the hard problems identified herein.

INFOSEC problems may be characterized as "hard" for several reasons. Some such reasons derive from the intrinsic technical challenges of building secure systems. Others derive from the realities of the modern IT market and the associated users' perceptions and expectations about INFOSEC. Some of the key technical factors that make INFOSEC problems hard include:

- users' insistence on INFOSEC solutions that permit the use of COTS hardware, software, and networks
- difficulty of widespread deployment of security technology
- difficulty of managing increasingly complex, networked systems securely
- dynamic security policy environments

[1] The workshops on which this report is based were convened under the auspices of the Infosec Research Council (IRC), with members from U.S. Government organizations that sponsor and conduct information security research. While IRC members use this list of hard problems to help organize their discussions, it does not necessarily reflect the specific research priorities of any IRC member organization.

- growing sophistication of the threat even from low-level hackers, e.g., increasing use of Trojan Horses to infiltrate target systems and exfiltrate data or provide a command platform for further attacks

Factors associated with the IT market and users' perceptions that make INFOSEC problems hard include:

- the fact that COTS products provide a high level of INFOSEC functionality, but they neither provide a high level of assurance nor the functions required to meet specific government needs
- government's diminishing influence as a market for COTS products and the associated diminishing interest of COTS vendors in meeting unique government requirements
- users' belief that COTS products will incorporate "sufficient" security without any need for government-unique technology or constraints
- unrealistic assumptions, e.g., about the ability to detect attacks that are not being prevented

This paper divides the problem space into a set of challenges associated with security features or functional requirements, and a set of challenges associated with the development of secure systems. For each problem category, the problem definition is followed by a discussion of factors that make the problem's solution important, factors that make the problem hard, and comments on approaches that seem either especially promising or especially unlikely to be successful.

The functional INFOSEC hard problems are:

1. Intrusion and Misuse Detection – providing IT system and network security managers with tools that can reliably detect attempts to defeat system security from without as well as instances of abuse by authorized users.
2. Intrusion and Misuse Response – providing IT system and network security managers with tools and techniques for responding to attack or misuse so as to identify, limit, and recover from the damage done by an attack and investigate the origin and mechanisms of the attack.
3. Security of Foreign and Mobile Code – providing users of IT systems with the ability to execute software of unknown or hostile origin without putting sensitive information and resources at risk of disclosure, modification, or destruction.
4. Controlled Sharing of Sensitive Information – Providing users of IT systems with the ability to process extremely sensitive information – including classified or compartmented information – in open, networked environments, while protecting that information from unauthorized disclosure.
5. Application Security – Providing tools and techniques that will support the economical development of IT applications that enforce their own security policies with high assurance.
6. Denial of Service – Providing system and network components and techniques for system design and operation that help to resist denial of service attacks.
7. Communications Security – Protecting information in transit from unauthorized disclosure, and providing support for anonymity in networked environments.
8. Security Management Infrastructure – Providing tools and techniques for managing the security services in very large networks that are subject to hostile attack.

9. Information Security for Mobile Warfare – Developing information security techniques and systems that are responsive to the special needs of mobile tactical environments.

The INFOSEC hard problems associated with the design and development of INFOSEC systems are:

1. Secure System Composition – Developing techniques for building highly secure systems in the case where few components or no components at all are designed to achieve a high level of security.
2. High Assurance Development – Developing and applying techniques for building IT components whose security properties are known with high confidence.
3. Metrics for Security – Developing techniques for measuring the security properties of IT systems and components.

A final discussion deals with the challenge of influencing the COTS vendors who are responsible for the development of most of the IT products and components that are used in real systems.

National Scale INFOSEC Research Hard Problems List[2]

This document presents a list of "hard problems" that pose obstacles to the abilities of United States Government IT users to process sensitive information securely. The definition of security that guided the development of this list encompasses data confidentiality and integrity as well as the availability of information and processing resources. The sensitivity of the information requiring protection ranges from routine business information to information whose modification or disclosure could result in major financial loss or loss of life. The threats to the confidentiality, integrity, and availability of information similarly run the gamut from hackers executing scripts downloaded from the Internet to national governments and major criminal enterprises.

This document was developed at the request of the INFOSEC Research Council (IRC) whose members (DoD [including DARPA, NSA, OSD, Army, Navy, and Air Force], NIST, DoE, CIA) are the major government sponsors of research in information security. The "hard problems list" is intended to guide the research program planning of the IRC members by identifying the key problems whose solution would remove major obstacles to effective information security. It may also be useful to policy makers and planners in evaluating the contributions of ongoing and proposed research programs to the critical INFOSEC problems facing the nation.

The IRC is sponsoring an Information Security Technology Studies Group (ISTSG) study that will develop a twenty-year vision of Information Assurance (IA), and a roadmap to progress toward achieving that vision. This document will serve as an input to the ISTSG as it develops its roadmap. The ISTSG will also check its IA vision against this list of hard problems to ensure that the vision addresses the impact of solving (or failing to solve) the hard problems identified herein.

INFOSEC problems may be characterized as "hard" for several reasons. Some such reasons derive from the intrinsic technical challenges of building secure systems. Others derive from the realities of the modern IT market and the associated users' perceptions and expectations about INFOSEC. Some of the key technical factors that make INFOSEC problems hard include:

- users' insistence on INFOSEC solutions that permit the use of COTS hardware, software, and networks
- difficulty of widespread deployment of security technology
- difficulty of managing increasingly complex, networked systems securely
- dynamic security policy environments
- growing sophistication of the threat even from low-level hackers, e.g., increasing use of Trojan Horses to infiltrate target systems and exfiltrate data or provide a command platform for further attacks

[2] The workshops on which this report is based were convened under the auspices of the Infosec Research Council (IRC), with members from U.S. Government organizations that sponsor and conduct information security research. While IRC members use this list of hard problems to help organize their discussions, it does not necessarily reflect the specific research priorities of any IRC member organization.

Factors associated with the IT market and users' perceptions that make INFOSEC problems hard include:

- the fact that COTS products provide a high level of INFOSEC functionality, but they neither provide a high level of assurance nor the functions required to meet specific government needs
- government's diminishing influence as a market for COTS products and the associated diminishing interest of COTS vendors in meeting unique government requirements
- users' belief that COTS products will incorporate "sufficient" security without any need for government-unique technology or constraints
- unrealistic assumptions, e.g., about the ability to detect attacks that are not being prevented

This paper divides the problem space into a set of challenges associated with security features or functional requirements, and a set of challenges associated with the development of secure systems. For each problem category, the problem definition is followed by a discussion of factors that make the problem's solution important, factors that make the problem hard, and comments on approaches that seem either especially promising or especially unlikely to be successful.

FUNCTIONAL HARD INFOSEC PROBLEMS

1. Intrusion and Misuse Detection

- This problem category addresses the need to build tools that can detect and localize both intrusions into computer systems and networks (by outsiders) and misuse of computer systems and networks (by authorized insiders).

  Intrusion and misuse detection systems and technologies are necessary in any real-world INFOSEC application. While preventive security techniques such as access control and authentication will prevent some instances of intrusion and misuse, such techniques are imperfect. In large-scale systems and networks, there will be residual vulnerabilities that are subject to exploitation by attackers. Furthermore, authorized insiders, by definition, have access to systems and networks that process sensitive information. Detecting when an insider has "gone bad" and is abusing his/her authorized access is critical to limiting the damage that such an insider can do.

  Intrusion and misuse detection are hard problems, fundamentally, because a well-executed attack or a subtle incident of misuse looks like ordinary system operation or use. The challenge for intrusion and misuse detection technology is to separate abuse from normal activity with a high alarm rate for real misuse (few Type I errors) and a low false alarm rate in the presence of normal authorized and responsible activity (few Type II errors).

  Today, most intrusion and misuse detection technology works like either virus detection (it recognizes "signatures" of attacks that have been previously encountered and analyzed) or it attempts to detect "anomalous" behavior (based on statistical

analysis and comparison to historical patterns) of systems and software. The former approach suffers from the fact that it can not detect new attacks. The latter is vulnerable to improperly tuned tradeoffs between Type I and Type II errors, "training" attacks that shift statistical norms over time, and, perhaps most importantly, it fails to provide near real time notification.

Recent efforts under the heading of "immune system" intrusion detection (also known as self/non-self discrimination) appear promising based on limited experiments, even though the metaphor may be somewhat strained. (The human immune system is readily defeated by biological weapons, which are analogous to sophisticated attacks. However, it responds well to many pathogens, which may be analogous to canned hacker attacks.) We have only two examples of the application of this technique so far, one for a specific privileged process in Unix and one for a CORBA application. The “immune system” approach assumes that one can characterize self by a very small set of trace parameters, but we have no proof that a well-designed attack can not exploit this assumption. The cost of creating the self database is high, because it is different for each site, and that may make the approach impractical. On the other hand, the fact that each site is characterized by its own database should improve the accuracy of this approach when compared to the normal signature-based intrusion detection system. The “immune system” approach shares with other INFOSEC technologies the need to be tested at scale – in this case, on a broader range of applications and against simulated hostile attacks.

Deployment of "honey pots” that are intended to attract attackers to a target under close observation and “canary" systems that signal the occurrence of an attack by expiring before a better-defended system would are two techniques under active investigation. Such approaches have proven effective at least against low level hackers. Wide spread deployment of these techniques must be preceded by system-level analysis. In particular, it is necessary to consider the fact that these techniques rely on attackers’ ignorance of their deployment, but standardization and proliferation of such systems may render them less effective as attackers encounter them in multiple systems.

More sophisticated analysis is another aspect of intrusion detection that deserves research. A goal would be to identify the precursors of an attack by developing a cyber indications and warnings technology. Techniques to allow the fusion of various forms of intelligence data with data collected from intrusion detection sensors also need to be investigated with the objective of providing complete visualization of the INFOSEC battle space.

Intrusion detection is a necessary component of a defense in depth, but COTS intrusion detection products are not nearly as capable (sufficient) as they are advertised to be today. COTS vendors are investing significant funds in the development and incremental enhancement of intrusion detection products--which continue to suffer the limitations cited above. Research funding in this area is justified only by truly innovative approaches to intrusion and misuse detection or cyber indications and warning. The paragraphs above have identified some promising approaches, but this is an area where fresh new ideas are needed. Because of the difficulty of the area, sponsors should be aggressive in testing the soundness of approaches against simulated attacks.

2. Intrusion and Misuse Response

- This problem category entails the development of tools and techniques for responding to attack or misuse, preferably in a sufficiently timely fashion to limit the damage done. Response will often be discussed in conjunction with intrusion and misuse detection, since an attack or incident of misuse must be detected before a response can be initiated.

  The problem of intrusion and misuse response would be unimportant if preventive mechanisms were completely effective. They are not, and it is unlikely that they ever will be. Therefore, response must be a part of a comprehensive system security solution.

  The problem of responding to an attack or incident of misuse is made hard in many cases by the uncertainty associated with the incident. What level of response to an ongoing attack is appropriate? How certain is it that a response will in fact target the attacker? What level of damage has been done by a successful attacker or malicious insider (what sorts of false data or malicious code have been introduced)? What data has in fact been compromised in the course of an incident? While all of these questions can be answered by a thorough review of a complete and accurate audit trail, the first action of a competent and malicious attacker will be to find and subvert the audit mechanism.

  Some intrusion detection systems offer the option of reacting to an attack, typically by notifying an administrator and by shutting down an offending process or closing a network connection. These responses may be appropriate and effective, provided the system can detect the attack before the attacker has done the damage he seeks to do, or gained access that bypasses the detection system or the controls available for response. One of the key problems associated with any response such as shutting down a service or network connection is that an attacker may provoke the response deliberately and thus accomplish denial of service on the protected system. The proper tuning of response mechanisms is an aspect of this hard problem.

  Some researchers have hypothesized, and some systems have implemented active responses that attempt to counterattack the system that is the presumed source of an

intrusion. Such strategies raise both technical (is the system being counterattacked really the originator of the attack?) and legal (is such a counterattack a violation of law?) issues. Resolving these issues is another aspect of this hard problem.

Beyond immediate response to an ongoing attack are the issues of damage assessment, investigation, and recovery. Forensic investigation of attacks for purposes of damage assessment and identification of perpetrators is a field of growing interest, though one dominated by manual effort and the use of existing system and network management tools. Recovery from an attack is straightforward though labor-intensive, if a "known good" backup is available. If such a backup is not available, recovery is a near-impossible task. Design principles for systems that would allow reliable recovery to an assured secure state constitute another hard problem.

Assuming that the response mechanism can be protected, a variety of intrusion response techniques and tools are needed to provide robust and flexible options. Policy based response, degraded mode operation, and rapid recovery are aspects of response that need to be addressed. Some form of assessment or analysis technique that could be used to help determine the impact and consequences of various response options prior to putting them into effect would also be a useful research product.

A new approach to intrusion response involves dynamic reconfiguration of applications in response to detected attacks. The ramifications of this interesting approach require thorough analysis. For example, this work proposes moving an application from one platform and operating system to another if an intrusion detection system produces evidence that the first is under attack. This tactic may provide an attacker with the ability to trigger the movement of an application from an operating system that is difficult to compromise to an operating system that has been compromised, simply by engaging in an attack that is designed to be detected. This example is one manifestation of a larger issue: intrusion response may add new "control surfaces" to our systems that create the potential for sophisticated adversaries to exploit these new facilities, especially for denial of service.

3. Security of foreign and mobile code

- Foreign code arises in two somewhat different contexts:
  1. Components of unknown provenance may be included in COTS or custom-developed system or application software; and
  2. Executable content may be introduced into a system during operation, whether as Java or ActiveX code downloaded from a web site or as Visual Basic macros embedded in a Microsoft Word or other document transmitted as an email attachment.

In the case of downloaded code, common practice today is to apply a COTS virus scanner to detect and eliminate harmful content. Virus scanning products represent a mature and widely-deployed technology. However, they operate either by recognizing specific viruses or by recognizing patterns that match the appearance of known viruses. Thus, they are dependent on users to detect early instances of a new

virus and on the vendor to abstract the characteristics of the virus and modify the scanner profiles as needed. A truly new virus can be expected to succeed until it has been detected and a scanner pattern developed and distributed.

Today, users may operate "safely" by disabling Java and ActiveX execution in browsers, by rejecting (or declining to enable) active documents, and by running an up-to-date virus scanner. However if enough web sites require Java or ActiveX to realize full functionality, or if enough users employ executable documents, then these approaches will fail. Some scanners that apply more or less standard virus scanning techniques to downloaded Java and ActiveX have been introduced, but they suffer from the same weaknesses as other virus scanners. Both scanning and rejection approaches suffer from the fact that they rely on individual users to configure their desktop environments to correctly resist attack. This level of reliance has often been misplaced.

Beyond the use of scanners, there are options of verifying the origin and presumed quality of downloaded code by means of a digital signature, and of limiting the harm that the code can do by confining it to a “sandbox” that limits its access to sensitive data and system resources. These options are viable provided that the environment that is receiving the downloaded code is sufficiently robust to reliably verify or confine the code. Today’s major signature verification technology (ActiveX) lets a user make a gross judgement about the provenance of software, and then trusts the developer completely. Signature of individual applets can be combined with fine-grained access controls to enforce specific limits on downloaded code, but this combination requires that the user correctly tune the profile of the code that he/she will accept – a requirement that is not likely to be met. The “sandbox” mechanisms available today have proven error prone and easy to circumvent. Moreover, sandboxes quickly become too confining and developers insist on moving beyond them, as with the evolution of Java Developer’s Kit Version 2.x.

In the case of code of unknown origin that is installed as part of a system rather than downloaded, the recipient has little choice today but to trust the developers and development process where the code originated.

The problem posed by foreign or mobile code is a hard one because, at a fundamental level, it appears that telling whether a program will attempt to do harm or not is equivalent to a Turing Machine halting problem – an unsolvable problem. Building an effective “sandbox” is roughly equivalent to the “confinement problem” – a software engineering problem that has proven very difficult in practice. Signature verification and virus scanning are straightforward and effective, but limited.

Research is needed to identify and exploit promising new approaches to dealing with the problems posed by foreign and mobile code. Such approaches would ideally result in the availability of tools that could reliably detect and block the execution of malicious code. At the same time, such tools would allow the execution of harmless code that had not been previously certified (no digital signature) and allow harmless code relatively full access to system facilities (no sandbox or confinement).

Approaches based on program proving, code analysis, reverse engineering and confined testing of potentially malicious code, and constrained properties of programming languages deserve additional consideration. Research teams may best be drawn from both the classical INFOSEC community and the community of researchers in theory of computation and programming languages.

Two promising specific avenues for research are identified in the NRC Trust in Cyberspace report: proof carrying code (PCC) and software fault isolation (SFI). The latter is intriguing in that it takes advantage of increased processor speed to support additional checking in software in lieu of added hardware checking. SFI is not a substitute for coarse-grained memory protection managed by an OS and enforced in hardware, but it may have a place in helping secure foreign code.

Since the growth of the Internet, organizations' networks have been protected by firewalls that were sensitive to the distinction between the organization's network and outside networks, and attempted to prevent harmful operations from entering the protected network. With deployment of mobile code and of objects with active content, users will be faced with the situation where information is only available if it comes accompanied by potentially hostile active content. Mobile code that migrates from platform to platform may carry its own security policy that conflicts with that of the host system where it attempts to execute. Mobile code that embodies a security policy may need to understand the trust attributes of the platform where it attempts to execute. Both of these topics are research problems that will require further consideration.

4. Controlled sharing of sensitive information

- Much of the sensitive information that the IRC sponsors and their parent agencies must process is formally classified or compartmented national security information. The most widely accepted formal security policy models for processing such information in shared (multilevel or multicompartment) systems date back to the 1970s. The development of systems that implement these models was a major research priority during the 1970s and 1980s, but those development efforts were never especially successful and have diminished almost to the point of vanishing over the last five years. Users who process classified information today do so almost entirely with COTS products that implement only a "discretionary" security model that is not specifically adapted to protecting classified information. The protection of classified information is left to discretionary controls, and to procedural and peripheral controls that may or may not be effective.

  Government sponsors of INFOSEC research should increase the priority that they place on systems that can enforce controlled information sharing in classified or compartmented environments. Users' growing demands for connectivity, including the availability of systems where unclassified information gleaned from the Internet or from administrative systems can be integrated with highly classified information, makes this problem more critical today than it was in the 1970s when research into

this topic was initiated. Increasing users' ability to share information without providing suitable controls is a very dangerous practice that plays into the hands of sophisticated attackers.

The problem of controlled sharing in environments where classified or compartmented information must be processed is a hard one for a variety of technical and non-technical reasons:

- Conventional security controls are "discretionary" controls that allow a user or program to disclose to another user any information that the user or program can observe, without regard to the classification of the information or clearance of the user. Such controls are unable to "confine" a Trojan Horse or other piece of malicious code that is determined to disclose classified or compartmented information. The discussion of foreign and mobile code is also relevant to this issue.
- Building a system that can effectively prevent the disclosure of classified or compartmented information to unauthorized individuals is a very difficult technical task. Even well-designed and well-implemented conventional systems are not good enough to protect classified or compartmented information from hostile attack.
- Given the choice between a system whose performance and functionality are up to current COTS standards and a system that can protect classified or compartmented information, almost all users will choose the former. To be used, a secure system must offer a high level of compatibility and performance compared to COTS systems, and meeting such a standard makes the system developer's task harder yet.
- The market for systems that can protect classified or compartmented information is a small one. Over the last decade or so, this market has become less important as a fraction of the total market for IT products.

Appropriate research topics in controlled sharing range from security models for multilevel processing to elimination or discovery and suppression of covert channels to the architecture and implementation of multilevel systems. The problem of managing and enforcing security for dynamic coalitions represents another aspect of controlled sharing. Managing the sharing of information among a "Community of

Interest" (COI) presents significant engineering challenges, and these challenges are made much harder in the dynamic coalition environment that changes the composition of a COI quickly.

The government should accept the fact that COTS vendors will not incorporate effective controls in their products for controlled sharing in classified or compartmented environments, and should seek to build systems or components that can augment COTS products. The development of such technology is critical to the protection of the government's most sensitive information.

We cite here some ongoing research and some potential directions for research in controlled sharing for classified environments. We do not intend to suggest that these are the only research directions to pursue, or even the best ones. Rather, we cite them to point out that there are research directions that can be pursued in this critical area.

It seems clear that the constraints that make the problem of controlled sharing hard will also mitigate against the development of full-function operating system products designed to protect classified or compartmented information. Instead of pursuing such costly and time-consuming development projects, a preferred path is to identify and develop architectures in which a few highly secure ("plug-in") components can assume the burden of security while most components are unmodified COTS. A number of concepts, architectures, and prototypes have been proposed that point the way along this path:

- The Starlight system provides a primitive multilevel capability that is currently being explored in trials at NRL and elsewhere. It exploits "thin client"-server architectures and simple physical isolation via switches and one-way links to give an end user a multilevel thin client environment that can import lower level information to a higher level environment and still use modern COTS software. While the technology and the current product have limitations, both have enough positive aspects to be worth further exploration.
- The Trusted File Server (TFS) architecture was proposed over a decade ago. The TFS architecture includes diskless workstations, LAN encryption, and the use of secure limited-function file, print and communication servers. NSA is exploring a similar approach applied to network computer technology under the name SLAT (single level at a time).
- Another candidate technology for supporting controlled sharing of classified or compartmented information is the virtual machine monitor (VMM). The use of a VMM to create a secure environment is attractive because it allows transparent use of COTS operating systems and applications while providing high assurance separation and control over the sharing of information. One might use a VMM in either a server or desktop context. In the former context, use of separate hardware for each security domain may often be economically feasible due to declining hardware costs. However, if domains must be created or destroyed quickly, or if there are large numbers of small domains, use of separate hardware platforms is not likely to be feasible. In the desktop

context, a VMM paired with a secure window manager provides a platform for quick domain switching.

Previous efforts to build secure VMMs failed because of incomplete support for modern computing features and insufficient market demand. However, we believe it is time to revisit this technique because recent results of VMM R&D (the DARPA-sponsored Disco project) appear promising and because the VMM approach is much less costly than the development of a full secure operating system. The pervasiveness the WINTEL architecture implies that a secure VMM for the Intel platform could see application in a wide variety of user environments.

Regardless of the specific approach taken, we believe that controlled sharing in classified and compartmented environments should be a primary focus for the government's INFOSEC research funding.

5. Application Security

- There is general agreement that the ultimate goal of building secure systems is to enable secure processing of information for users. Users view information processing in terms of the applications that are executed and the data that those applications manipulate and present. Applications often have security requirements that extend beyond what operating systems offer. For example, distributed collaboration applications may require policies unique to the application and user community. Thus there is increased interest in application security. Historically, applications built on top of insecure operating systems have been intrinsically vulnerable to a variety of attacks that cannot be thwarted by the application, so a secure OS base has been required if applications were to be secure.

  One hard problem is to revisit the assumption that, for applications and application data to be secure, there must be an underlying operating system with a set of well-defined security attributes. While there is adequate reason to suspect that this problem is impossible rather than merely hard, the payoff from its solution would be sufficient that it is worth examination, at least to the point of developing a proof that the problem is unsolvable.

  Assuming that a secure operating system is necessary as a base for secure applications, a second hard problem is to determine what forms of operating system security are necessary to provide the right underpinnings for application security. At a minimum, the operating system must protect the application program and its data from access by users who have no authorization whatever. Beyond this overall protection, however, there may be requirements that certain data can only be read or modified by users executing specific programs. Type Enforcement may be an appropriate operating system interface to support the implementation of many types of application security. Research into application security policies and the operating system mechanisms required to support them is an appropriate area for future funding. While the need for systems that can process classified and compartmented

information securely is a factor in this area, the problem of operating system support for application security is much more general than the problem of protecting classified or compartmented information.

6. Denial of Service

- Prevention of or resistance to denial of service attacks is a growing problem and a very hard problem to address. Network launched denial of service (DoS) attacks against end systems (desktops and servers) are almost always successful and there are many dimensions that attackers may exploit. Traditional robustness measures often do not prove effective against DoS attacks, since such measures usually are oriented toward protecting against benign, uncorrelated events. In fact, some forms of robustness measures may exacerbate DoS problems (e.g., automatic propagation of corrupted data to backup servers.)

  DoS attacks against networks are relatively rare today, but not unknown. Several protocols that are critical to the operation of the Internet are extremely vulnerable, (e.g., Open Shortest Path First, or OSPF, and Border Gateway Protocol, or BGP). OSPF supports routing within individual autonomous systems (ASs) while BGP interconnects ASs. Security countermeasures have been developed for OSPF and work is underway to improve the security of BGP. The former have not been incorporated into COTS routers nor deployed because of a perception that there is no credible threat to an individual AS; the latter may meet resistance because of the potential impact on network operating procedures imposed by any non-trivial security technology.

  While it is easy to cite DoS attacks, the development of technology to resist such attacks, in systems or in networks, has long been viewed as a very difficult problem. In some cases, such as countermeasures to the network DoS attacks mentioned above, enhancements to individual mechanisms show promise. More general models of continuous system or network operation, and architectures for systems that can resist DoS attacks, pose a problem worth emphasis by research sponsors. It may be that using an operating system base whose self-protection is strongly assured will go far in providing resistance to denial of service attacks. However, it should be noted that most network components incorporate dedicated rather than general-purpose operating systems, and that network vulnerabilities to DoS attacks often result from flaws in application programs or the underlying communications protocols.

  A more difficult variation on the DoS problem is that of developing systems that can continue to operate or to perform critical elements of their mission, even in the face of partly successful attacks. This area goes beyond DoS to encompass operation in the presence of systems that are trusted but may have been penetrated. Not much is known about theory or practice of building such systems, but it is clearly unrealistic to assume that no attack will succeed, and thus there is need to understand how to organize or partition systems so that some core functionality can survive and carry on.

The entire field of DoS attacks appears likely to benefit from some new ideas and fresh approaches that might supplement incremental improvements to existing techniques. A theoretical model of the best conceivable responses to DoS attacks would provide a useful benchmark against which more practical and near-term approaches could be calibrated and evaluated.

7. Communications Security

- The DoD and other government agencies have unique requirements for communications services that are unlikely to be met by standard commercial offerings. These services often pose a significant technological challenge within the context of modern networked communications systems. For example, threats that derive from the analysis of communications characteristics such as traffic patterns, timing, addressees, or other message externals can often pose as significant a risk to operations as the complete compromise of data. Protection of these characteristics is not available from end-to-end encryption alone. Techniques used in previous generations of circuit switched technology, such as full-time traffic flow security, would be extremely wasteful of bandwidth and may not apply well to advanced communications technology. At the same time, past claims of the inherently dynamic nature of packet switched communications have proven to be vastly overstated. The problem of traffic flow security will become aggravated as simple privacy protection becomes widespread and as more and more of the communications infrastructure and services supporting the government become outsourced**.**

  Anonymity in communications is another requirement that may demand special attention from the government. Concern with privacy in the Internet community has given rise to a number of anonymous re-mailer services. A few government projects have begun to deal with anonymity issues, notably the NRL onion routing technology. However, greater attention to government needs in this area and the technology to support a wide variety of high assurance anonymous services is required.

  Beyond these general COMSEC problems are specific requirements that should be addressed. DoD needs releasable cryptography and security functions (e.g. secure voice) that can be quickly deployed to coalition (i.e. NATO) partners and mobilized facilities (e.g. Maritime Sealift Command ships) as conditions require. DoD also needs technology for “sanitizing” information systems and “neutralizing” critical systems if a facility is overrun or abandoned. Finally, there is no current work aimed at providing a very high speed IP encryption capability for classified government information.

8. Security Management Infrastructure

- The introduction of security technology into large-scale distributed systems requires the introduction of distributed security tools and associated data to a large number of components. The range of services that must be deployed and managed encompasses certificate-based authentication and encryption facilities, certificate distribution and revocation, policy interoperability, interoperation with peer security management infrastructures, sharing of labeled sensitive information, and collection and correlation of audit trails, to name only a few examples. Commercial vendors are investing heavily in distributed security management tools, and commercial users are at least beginning to deploy those tools at the enterprise level.

  Government requirements for security management infrastructure include two facets that are especially demanding, if not unique to government. First, government requires management systems that are effectively hardened against hostile attack. Second, even though government is no longer as significant a market as it once was, government networks are still among the largest extant, and will stretch the scalability of management products and architectures.

  Commercial vendors are not likely to meet specific requirements for the management of security components that can protect highly sensitive or classified information, nor are they likely to build high-assurance management tools or platforms. It also seems unlikely that commercial vendors will invest in management facilities that are effective in the intense and dynamic threat environment likely to confront some of the government's most critical information systems. It is appropriate for government to invest in the research needed to ensure that evolving high-assurance systems (which we hope will be produced by research in response to other hard problems listed here) can be managed when deployed on a large scale.

9. Information Security for Mobile Warfare.

   A concept for a Joint Tactical Intranet that provides significant improvements in connectivity between land based nodes, maritime and air platforms in the battle space, along with internetworking services to the sustaining logistical and intelligence support systems, underlies DoD's long-term IT program. The doctrinal concept for "network-centric warfare" envisions a robust, seamless, digital data network built on existing wireless tactical communication systems, new information distribution capabilities, and commercial off-the-shelf products and services.

   The fact that DoD is attempting to leverage commercial technologies and standards while meeting the unique survivability requirements of the battlefield makes providing security in the mobile warfare environment a particularly hard problem. The mobile warrior network will have to communicate over noisy, congested, low-bandwidth radio channels. Information systems will have to operate in a hostile environment, contending with the classical military communication problems of noise, interference, jamming, interception, overrun, and physical destruction as well as a host of newer threats such as deception, masquerading, network flooding, insider misuse, malicious code, and other forms of network attacks. Highly mobile nodes will intermittently gain and lose network connectivity with other nodes. The warrior network will have to rapidly and continually reorganize itself to achieve reliable

connectivity. Network traffic required to reorganize and control the network will further stress low-bandwidth wireless channels. Commercial security approaches that rely on centralized authentication and services that use relatively high bandwidth or the ability to conduct interactive dialog are not practical in this environment.

More mundane reasons than those cited above also conspire to make this a difficult problem. The protocol stacks in PCs and routers fare very poorly in low-bandwidth tactical network environments. This was demonstrated years ago in the DARPA packet radio environment, and to some extent in MSE, where the IP routers are custom technology rather than COTS products. The situation is worse today because it is difficult or impossible to tune parameters in COTS TCP/IP implementations. COTS TCP/IP will work in a static wireless environment, will be marginal in a tactical environment not under attack, and will fail under hostile attack. Some of the tricks being used to improve performance of COTS wireless systems rely on access to TCP headers that is prevented by IPSEC encryption. Thus we have a good chance of creating a tactical environment where, in order to get acceptable performance, we give up end-to-end security.

The challenge posed by this hard problem is to devise security protocols and cryptographic mechanisms that can cope with the unique problems and constraints of the tactical environment. Solutions to this problem will achieve security dynamically in a low-bandwidth, unreliable multi-medium communications environment. They will adapt to network outages caused by hostile agents that will take active measures to prevent the restoration of service. They will provide reliable and effective security with minimal administrator support for ensuring the details of system security.

SECURITY ENGINEERING METHODOLOGIES

1. Secure system composition

- If one assumes that systems must be composed only from COTS products that vary considerably in their security features and assurance, then one needs a methodology for such composition. The previous statement assumes that highly secure systems can be composed from less secure, or insecure components. The "theory of insecurity" put forth in the NRC Trust in Cyberspace report suggests that one should not focus on trying to eliminate vulnerabilities in systems. Rather, it acknowledges that vulnerabilities will be present and that the best one can accomplish is to move insecurity around, to make it more manageable or to minimize or ameliorate its impact. We lack any methodology for security engineering based on this paradigm, but we also lack any demonstrably successful methodology for constructing secure systems from (the small number of) highly assured components! We agree with the NRC report that this approach to secure system development may be worth pursing, and we believe that the development of a well-defined methodology for composing secure systems from less secure components constitutes a hard problem.

  Explorations of this approach to composition should begin with limited experiments and with investigations into and articulation of the underlying reasons why more

secure systems can be composed of only less secure components. Full-scale engineering developments should follow studies into the theory of composition and successful small-scale developments.

- If it proves infeasible to build secure systems only from less secure or insecure components, one can relax slightly the constraints cited above, and allow the introduction of a small number of high assurance components into a system. This approach seems feasible because the majority of components would be COTS and thus the cost for deploying a small number of non-COTS high assurance components would not be prohibitive. The hard problem is to identify in a general way what security components need to be of high assurance and where they need to be placed in the system in order to improve system security significantly. If we can identify "security lynchpin" components that would be widely useful and practices for integrating them into systems to enhance overall security, these two developments would greatly simplify the task of building high assurance systems. This approach to secure system composition is closely related to the approach of providing “security plug-ins” discussed under the topic “Controlled sharing of sensitive information”.
- One might envision several types of tools that would function as security engineering and architecture tools. One such tool would help to track security dependencies, trust dependencies, and the like in an architecture to ensure that some security design goals or policies were correctly enforced. These tools would help in the top-down design of a system for a particular environment or set of threats. The goal of research into such tools would be to come up with more of a scientific, repeatable, demonstrable approach to architecture and system design.

2. High Assurance Development

- We are concerned that the expertise needed to develop high assurance components is not adequate to the need, and that the expertise that has been developed is being lost. High assurance operating system R&D in the commercial sector is moribund. Secure applications, such as military messaging systems, are on the wane. While there is little in the way of products or systems to show for it, the United States invested significant resources in high assurance development between 1973 and 1995. This investment generated some prototype systems, a few products of limited market success, and a few systems that achieved limited deployment in government. It generated experience with formal and structured design, and with security analysis of real systems. The country is at risk of losing the experience base that was developed over the last 25 years.

  We believe it is important to identify a few development programs targeted at producing high assurance software systems or components. Those development programs, to be successful, would need to yield high assurance products that would support COTS technology, would not be so expensive that their deployment is stymied, and that would not degrade performance so badly as to be shunned by prospective users. A specific development program might identify and isolate a security critical feature of a system or application and move it into a highly assured

component. Such an effort might both produce a useful component and provide experience from which to learn and generalize.

The high assurance development problem complements the controlled sharing problem. In the latter, the emphasis is on the security models, architecture, and interfaces for systems or components that can process classified information securely in a hostile environment. High assurance development emphasizes the techniques for designing and implementing systems whose security properties can be characterized and evaluated against a standard. The classic techniques for high assurance development include layered and structured designs, formal verification, testing, analysis, review, and rigorous processes. Application of "theories of insecurity" may prove to be a new alternative for achieving high assurance, and classic approaches to the design of high-assurance secure systems have paid little attention to the techniques used to produce and maintain cryptographic, man-rated, or nuclear release software. The development of vulnerability assessment tools, vulnerability data bases, and tools for simulating attacks and responses may help enhance capabilities for assessing the security of components and the systems that they compose. We believe that multiple approaches should be explored with the ultimate aim of the development and maintenance of a capability for building and maintaining high assurance secure systems. (The importance of "and maintaining" can not be overstated. Software-based systems must be able to change with evolving requirements, and a successful process for developing high assurance systems must address the need for timely and highly assured changes to those systems.)

A related problem involves the design and verification of high assurance network protocols. The protocols at issue may be either cryptographic protocols or other network protocols that must meet reliability or robustness requirements.

The High Confidence Systems working group that has been chartered by the HPCC program has developed a "National Research Agenda for High Confidence Systems". This document discusses the need for high confidence systems and recommends a set of research activities aimed at improving the nation's capabilities in the development of high confidence systems. The research agenda is the product of more in-depth analysis than the present document and should be considered as a complementary effort in the area of high assurance systems.

3. Metrics for security

- The notion of "managing risk" is a good one, but it suggests a degree of precision that is absent from the approaches to system security engineering that are currently employed. Designers' understanding of the vulnerabilities of individual components and the systems into which they are integrated is often intuitive and vague, and there is a significant probability of using a component so as to accept an unquantified risk - which hardly qualifies as risk "management." It may not be possible to quantify risks, but designers should at least be aware of vulnerabilities associated with the use of various components, under the guise of "risk management." There is a need for

better-structured vulnerability analysis and cataloging approaches that can be used to guide system designers and integrators.

- A commonly cited security metric is that of "work factor" as applied to cryptanalysis. While this metric is well established in the cryptographic community, its use in more general discussions may be misleading, since resistance to cryptanalysis is a necessary, but not sufficient requirement for the security of a deployed cryptosystem. More generally, it is fairly common to hear references to work factors that are not supported by consideration of the alternative or least time-consuming ways of defeating a security system.
- The development of metrics for the security of real-world systems is an extraordinarily difficult task. However, such a metric would be a high-payoff result, so it is worth seeking new approaches to this hard problem. Research should address metrics for the security of overall systems and for that of the components of which systems are built (along with rules for combining the metrics as components are combined). If successful, such research would have a profound impact on the ways in which components and systems are designed and built. Among other things, such metrics would lead to a refinement in the United States' application of the international Common Criteria (which are intended to address security requirements for systems or components), or to a revision based on a more quantitative understanding of system security. The National Research Agenda for High Confidence Systems referred to above includes recommendations for research effort in metrics.

INFLUENCING VENDORS

The nontechnical "hard problem" of influencing COTS vendors deserves mention in this paper. Government has often been ineffective in efforts to influence vendors through procurement processes. The procurement path has been largely unsuccessful for two reasons: first, the government represents a small and shrinking portion of a very large information technology market, and second, procurement regulations oriented toward security have often been ignored by government buyers.

However, vendors often can be influenced through standards processes, and this path represents another avenue for the government to achieve better security in COTS products. Both direct agency participation in standards development, and agency participation through contractors acting as surrogates can be effective approaches. The choice should be determined by the particular standards group and its willingness or reluctance to be influenced by government, and by the availability of contractors and their credibility as players in the technology and the market at issue.

Open publication of research results, either in the form of professional papers or reference software implementations, has occasionally had the effect of influencing or even creating markets. As government agencies sponsor research, it is a given that they will ask for the publication of research results in the literature. Sponsors should also consider what treatment of research prototypes will result in maximum enhancement of the nation's INFOSEC capabilities: transfer of commercial rights to the research organization or another party, release of prototype code to the public, or some other strategy.