Carnegie Mellon
Software Engineering Institute
Pittsburgh, PA 15213-3890

SEI Monographs on the Use of Commercial Software in Government Systems

DoD Security Needs and COTS-Based Systems

Scott A. Hissam
David Carney
Daniel Plakosh

September 1998

About this Series

Government policies on the acquisition of software-intensive systems have recently undergone a significant shift in emphasis toward use of existing commercial products. Some Requests for Proposals (RFPs) now include a mandate concerning the amount of COTS (commercial off-the-shelf) products that must be included. This interest in COTS products is premised on a number of factors, not least of which is the spiraling cost of software. Given the current state of shrinking budgets and growing need, it is obvious to almost any observer that appropriate use of commercially available products is one of the remedies that might enable the government to acquire needed capabilities in a cost-effective manner. In systems where the use of existing commercial components is both possible and feasible, it is no longer acceptable for the government to specify, build, and maintain a large array of comparable proprietary products.

However, like any solution to any problem, there are drawbacks as well as benefits: significant tradeoffs exist when embracing a commercial basis for the government's software systems. Thus, the policies that favor COTS usage must be implemented with an understanding of the complex set of impacts that stem from use of commercial products. Those implementing COTS products must also recognize the associated issues—system distribution, interface standards, legacy system reengineering, and so forth—with which a COTS-based approach must be integrated and balanced.

In response to this need, a set of monographs is being prepared that addresses the use of COTS software in government systems. Each monograph will focus on a particular topic, for example: the types of systems that will most benefit from a COTS approach; guidelines about the hard tradeoffs made when incorporating COTS products into systems; recommended processes and procedures for integrating multiple commercial products; upgrade strategies for multiple vendors' systems; recommendations about when not to use a commercial approach. Since these issues have an impact on a broad community in DoD and other government agencies, and range from high-level policy questions to detailed technical questions, we have chosen this modular approach; an individual monograph can be brief and focused, yet still provide sufficient detail to be valuable.

About this Monograph

Integration and incorporation of COTS components into legacy and emerging systems has never been more attractive in the information industry. The COTS marketplace has become very competitive with the increased number of vendors and the increasing number of products offered. This, combined with ever increasing pressures to deliver systems sooner and cheaper, has only hastened the call to use COTS. However, it is also important to recognize that most markets are driven by that which can be sold to the largest audience, and that audience may not always share the same perspective or notional model as that of any one buyer (in this case the system integrator). Security is one such area of interest that managers and system integrators must address. Each may find themselves in dire straits trying to reconcile what the market has to offer and what the needs are of the information system. This monograph offers a "heads-up" to decision makers who are building information systems that have security constraints, who feel the market imperatives, and who want to make opportunistic use of what the market has to offer.

DoD Security Needs and COTS-Based Systems

1 Introduction

The practice of acquisition, development, and sustainment of complex software systems has changed considerably over the past few years. Much of that change has come about due to a maturing software component industry and to the wide variety of components now commercially available. It is increasingly possible to find commercial off-the-shelf (COTS) components for many application domains, such as geographic information systems for command and control, product data management for sustainment support, and financial packages for comptrollers. Central in this growth of software components is the development of the world-wide web (WWW), together with the appearance of many rapidly changing component technologies that exploit the web, particularly the huge growth of browser-based architectures for information systems.

As with many organizations in both the public and private sectors, the U.S. Department of Defense (DoD) is committed to a policy of using COTS components in information systems.[1] However, the DoD also has a long-standing set of security needs for its systems, and the pressure to adopt COTS components can come into conflict with those security constraints. The principal source of this conflict is the DoD's overall approach to system security on one hand and the economic forces that drive the component industry on the other. This conflict becomes more prominent as DoD managers and system integrators look to the COTS marketplace for components to satisfy additional security requirements.

The traditional DoD approach to security is to isolate the systems it deploys from the rest of the world by limiting access and by guarding the information that is delivered. This is done physically by locked doors, armed guards, and other physical measures, thus making it difficult to touch the system, thereby lowering the likelihood of interfering with its functioning. Information is guarded electronically by encryption, private networks, and other electronic means, thus making it difficult to see the information being transmitted, lowering the likelihood of compromising the information.

In contrast to the DoD, the component industry's existence is based on the need to appeal to a broad market. Therefore, the software component market thrives on openness and accessibility: interconnectivity and heterogeneous information exchange are coveted attributes for commercial products. Knowledge about accessing systems and about product behavior is expected to be generally available. Information delivery is designed to be as seamless as possible. In this view of the world, such things as confidentiality are a secondary concern.

These different viewpoints can produce a mismatch between the security requirements of DoD systems and the characteristics of the COTS components available for those systems. It therefore becomes industry's challenge to attempt to meet the needs of the diverse marketplace while at the same time producing components that are useful to one critical market segment. Likewise, it becomes DoD's challenge to recognize the potential shortcomings of COTS components for its needs and to try to bridge the gap between its requirements and the realities that drive the component vendors. The purpose of this Monograph is to examine some of the realities of the marketplace and how to begin to address these conflicts relative to the security needs of the DoD.

The remainder of this Monograph is as follows. In Section 2 we define what we mean by the term "security" as we use it in this paper. In Section 3 we look at some of the key issues facing DoD programs when adopting or evaluating COTS components. In Section 4 we look at the marketplace imperatives that drive commercial vendor and how that conflicts with DoD requirements. In Section 5, we discuss some ways that these varying needs might be reconciled. Finally, in Section 6, we summarize the key points of this Monograph.

2 Definitions of Security Terms

The term "security" can have many different meanings and be used in many different contexts. For the purposes of this Monograph, we will separate the overall notion of security into attributes, pressures, or mechanisms. A security *attribute* is a characteristic that is desirable in a software component or system such as confidentiality, integrity, and non-repudiation. A security *pressure* is a force that can negatively impact the security attributes of a system, for example, threats, vulnerabilities, and risks. A security *mechanism* is an approach or method used to protect the security attributes of a system from security pressures, for example, identification and authentication, authorization, and cryptography. We now provide capsule definitions of each of these terms.

Attributes

- **Confidentiality**: Sensitive data is held in confidence, limited to an authorized set of individuals or organizations.
- **Integrity**: Processes do only and exactly what they are stated to do and data are not modified during storage or transmission.
- **Non-repudiation**: The sender of data is provided with proof of delivery and the recipient is assured of the sender's identity, so that neither can later deny having processed (i.e., having sent or received) that data.

Pressures

- **Threat**: Any circumstance or event with the potential to cause harm to a system in the form of destruction, disclosure, modification of data, and/or denial of service.
- **Vulnerability**: A weakness in system security procedures, system design, implementation, internal controls, or other weaknesses that could be exploited to violate system security policy. This may involve unauthorized disclosure, unauthorized modification, and/or loss of information resources, as well as the authorized but incorrect use of a computer or its software.
- **Risk**: The probability that a particular threat will exploit a particular vulnerability of the system, usually involving an assessed balance between threat and vulnerability.

Mechanisms

- **Identification and Authentication**: Verification of the originator of a transaction, similar to the signature on a check or a Personal Identification Number (PIN) on a bankcard.

- **Authorization**: The granting of access rights to a user, program, or process.

- **Cryptography**: The protection of data to render it unintelligible to other than authorized recipients. Many techniques are known for the conversion of data, called *plain text*, into its encrypted form, called either *cipher* or *ciphertext*.

3 Key Factors for Security in DoD Systems

An obtainable goal for any DoD system is to exhibit basic security attributes, remain operational, keep the information those systems contain safe, and protect themselves from intrusion. In the context of COTS-based systems, there are three factors that have a bearing on the degree to which these goals can be achieved. The first is the existence of recommended DoD standards in the domain of security. The second is the need to gain trust of commercial products that are used in DoD systems. The third is the essential nature of commercial products and their susceptibility to malicious attack.

3.1 DoD Security Standards

DoD has several security standards, some of which have gained conformance in commercial products, but only in a limited way.

Several security requirements have been identified and mandated for DoD computing systems for many years. The National Computer Security Center (NCSC) has identified many of those requirements in its "Rainbow Series" that started with DoD Standard 5200.28-STD Trusted Computer System Evaluation Criteria (TCSEC) in 1983, also known as the "Orange Book". Additionally, the National Institute for Standards and Technology’s (NIST) Federal Information Processing Standards (FIPS) define a number of computer security standards for government organizations in the domains of access control, cryptography, security labeling, risk analysis and contingency planning, and general computer security.

These standards have made some headway into commercial products, but they are by no means ubiquitous. For example, version 4.0 of Netscape Communicator is compliant with FIPS Security Requirements for Cryptographic Modules (FIPS 140-1), although the feature is disabled by default. Netscape’s Certificate Management Server (CMS), however, only partially supports FIPS Digital Signature Standard (FIPS 186); it replaces the actual signature generation algorithm with different algorithms from RSA (patented from RSA Data Security and licensed by Netscape Corporation). Microsoft’s Internet Explorer, makes no claim of conformance to FIPS 140-1. But Microsoft’s Internet Information Server (IIS) *does* fully support the Digital Signature Algorithm (DSA) as defined in FIPS 186.

To make the DoD’s security requirements better known, representatives from various DoD organizations have joined numerous organizations and standards groups. Groups such as the World Wide Web Consortium (W3C) Security Interest Group, Object Management Group

(OMG) Security Special Interest Group, and the Internet Engineering Task Force on Public Key Infrastructures (IETF-PKI), have strong representation from the Government in their working groups. While these efforts have contributed to industry awareness of the Government's security needs, the commercial marketplace continues to be a limited source of off-the-shelf components for systems that strictly adhere to the DoD's security standards.

3.2 DoD's Need for Trust in COTS Products

Trust of a COTS component comes from knowledge about it. This knowledge most often comes from others, whether a product's vendor or other (third-party) sources. Some knowledge can also be gained from first-hand examination (e.g., information about standards compliance).

An integrator who is incorporating commercial products into a DoD system has difficulty in trusting components when first-hand examination of the source code for the component is not possible. This issue is important for any system, but it is especially important for a DoD system that has stringent security requirements. Nor is this issue trivial, since COTS components have often contributed to the vulnerability of a system to outside attack. The "Internet Worm" incident in the fall of 1988 is probably the best known.[2] In this attack, a program (i.e., the "worm") took advantage of a archaic debug feature of a UNIX network daemon, `sendmail`. Sendmail is the workhorse that receives and delivers electronic mail for nearly all UNIX-based operating systems. At the time, it was installed and used with little additional consideration, and few had knowledge of the existence of the debug feature. Yet this forgotten feature led to numerous system failures throughout the country.

One form of trust derives from knowing whether anyone else has found problems with a COTS product and if so, what kind of problems. "Buglists" are a particularly important source of such knowledge. These are collections of known problems found in certain products; they are often published by a product's vendor. These buglists sometimes identify gaping security holes in the services provided by their products. (Unfortunately, buglists may be published by the vendor long after the component has been integrated into a complex system.).

Security advisories are also published regularly by organizations like the Computer Emergency Response Team Coordination Center (CERT®/CC) at Carnegie Mellon University, which was established to track and coordinate information regarding vulnerabilities in commercial off-the-shelf software. CERT itself has coordinated and released over 240 advisories, bulletins, and summaries dealing with exposed vulnerabilities in operating and network systems software. Additionally, CERT coordinates information with a number of incident response teams within the Government and around the world.

In addition to buglists and third-party advisories, trust in a component can come from knowledge about the open standards to which components are compliant (TCP/IP, COM, CORBA, etc.). Knowledge of these standards aid in understanding the mechanisms by which software components interact with other components. Techniques for unveiling some of the inner workings of a software component such as snooping network traffic, tracing call stacks, and interactive debugging can also provide greater understanding of the component's construction and design.[3][4]

3.3 The Essential Nature of COTS Components

COTS components are more susceptible to vulnerabilities than custom code because the "hacker" has access to the same third-party knowledge about the component as the system integrator.

It cannot be denied that the need for trust on one hand and the essential nature of commercial products on the other are somehow in conflict. An essential characteristic of the commercial world is the need for openness, which makes COTS components especially susceptible to malicious attack (and hence more difficult to trust). For instance, all of the information described above that documents known problems about a product is equally available to anyone, both the integrators that develop a system using off-the-shelf software components, and the threat agents (i.e., "hackers") who try to obtain access in a malicious or unauthorized manner.

Since the hackers have access to the same commercial components (and second-hand information about them) that the integrators do, they can purchase these components, install them in their environment, and pick and probe at the components in the privacy of their own testbed until a vulnerability is revealed. Once a vulnerability is revealed, "hackers" can then use it to compromise any system that employs the use of that component. Hackers can also read buglists, and use them to probe for other weaknesses.

One example of how hackers attack a system is a simple password attack. For some commercial operating systems, it is possible to retrieve the password database and copy it to an alternate location. In such a case, the hacker with the copy of the database can attempt to crack or guess valid passwords on his own computer using the resources to which he has access. Once the password is revealed, the hacker can return to the system from which the database was pulled, and on the first attempt, use the compromised password to gain access. This is just one example of the more general premise that everything that can be learned about a software component can be used by a hacker for purposes that are of malicious intent.

4 Imperatives in the Commercial Marketplace

These issues should not be interpreted to mean that security is unimportant for commercial products. On the contrary, security represents a growing concern for many COTS vendors. However, the software component industry, like any industry, is driven by demands of the consumers. With the emergence of the world-wide web demand for software components with security attributes and mechanisms has risen. But this demand has emerged independently from the established security needs of the DoD; the commercial need for security has been driven by electronic commerce and consumer privacy via the world-wide web.

The DoD, therefore, must now compete with other consumers, both domestic and international, to get desirable security attributes and mechanisms incorporated into commercial products. However, the influences that motivate the way components are designed, built, and marketed are complex. At least three factors affect the way that vendors include security factors in their marketing plans:

- Market priorities that govern product features
- Different models of security behavior.
- Laws and statutes that affect the export of software and hardware.

4.1 Market Priorities Govern Product Features

Features and capabilities of COTS security components are selected based on revenue opportunities.

Vendors prioritize features and capabilities based on existing and projected customer demand, the existing competition, and projected future technology direction. Features that are, from a vendor's point of view, relatively unimportant, are likely to be included in later product revisions (if at all). Thus, any DoD-specific requirements for secure features may take second place to features with a higher initial marketability. If some product feature appears obscure to the component vendor, it is probably not as much as a decisive factor to him as it is a requirement for the DoD customer.

The browser discussion presented earlier exemplifies this. The Netscape browser is compliant with FIPS 140-1 while the Microsoft browser is not. For systems that are based on Microsoft products, there is a conflict between their installed base and conformance with the Government's standard for security. While DoD may regard the most desirable solution as having Microsoft bring its product into compliance with this standard, that same view may not be shared by Microsoft. Microsoft will add (or not add) features based on the largest perceived market, forecasts about future trends, and other business-related factors.

4.2 Differing Models of Security Behavior

COTS components striving for the commercial marketplace will probably be based on a different security model from that needed by the DoD.

It is trivial to build a simple web-based information systems from available, off-the-shelf, components. One reason is that most of the available components are based on a widely held model of how information is delivered between distributed clients and servers via the WWW. This model is shared by the vendors of these components and the integrators that use them, which naturally increases the ease of integration. The component marketplace grew as this web-based model became dominant, and in fact helped to shape that model. Today, many of the advances in the WWW are coordinated through the World Wide Web Consortium (W3C) which has sitting members from industry, academia, and government—both international and domestic.

We have already noted that industry—including the W3C—has been slow to embrace security standards and mandates that have been imposed on DoD systems. One major cause of this is a difference in security models. The existing model for the DoD (i.e., the one that is implicit in the

"Rainbow Series") is very different from the security model being developed and build in the commercial community.

An example of how the two models can conflict can be seen in a DoD project that hoped to use commercial components to manage access through certificate revocation. Certificates are a computer-based file or structure used to convey digital information about a user for identification purposes. Certificate revocation lists (CRLs) are used to identify those certificates that were previously issued but have since been revoked. For instance, a system containing compartmented data is required to identify and authenticate users wishing to gain access. A mechanism used to meet this requirement is often CRLs. The system first verifies that a certificate presented is authentic, and then checks the CRL to ensure that the certificate has not been revoked. Authorized users with an authentic and non-revoked certificate are granted access.

However, when the DoD project examined some commercial products to provide secure certificates, a vulnerability was found to exist in this model, because updates to the CRL are periodically done incorrectly. On occasions when a system fails to properly update the CRL (i.e., the system fails to revoke a certificate that should be revoked), invalid users are permitted access. The DoD project proposed that this vulnerability be removed by logically inverting the purpose of the CRL. In this model, the CRL would become an access control list, and all valid users would be placed on the CRL. This would mean that for an authorized user to gain access to compartmented data, the certificate had to be authentic and *had to exist* on the CRL.

While this approach was attractive to the DoD project (since it models the traditional physical model of gaining access to compartmented data), it was in conflict with the behavior of commercial components, which are designed to treat CRLs as originally described. The project therefore was unable to use existing products, since *all* commercial servers and certificate managers in the project would still interpret all certificates on the CRL as revoked.

4.3 Laws and Statutes

Some COTS products may lack sufficient security attributes because vendors must conform to laws dealing with the export of software and hardware that is critical for national security.

Vendors often wish to sell their products in domestic and international markets and to both government and industry. Existing legal constraints have a significant bearing on how vendors can make products available, particularly those with critical security mechanisms. A good example of this can be seen in Title 22, Code of Federal Regulations (CFR), Parts 120-130, of the International Traffic in Arms Regulations (ITAR), which pertains to federal controls restricting export of high-grade encryption technology (e.g., DES, Triple DES, 128-bit SSL, etc.).

Any vendor of a product that involves encryption technology must choose one of two options. The first is to make two versions of his product, one a domestic product with high-grade encryption (i.e., that is subject to the ITAR controls), the other for export, that will necessarily use less secure technology. Maintaining two parallel products like this is generally not cost effective, and many vendors simply choose to market a single product that can be sold world-wide. The result for domestic consumers (including the DoD) is that fewer products are commercially available that have certain desirable features and characteristics (high-grade encryption in this example).

We are not suggesting that regulations such as the ITAR are not useful or are unnecessary. We are only noting that the realities of these laws and statutes can have a major impact on the vendors' products, which can in turn impact the Government's capability to build and deploy commercially based, yet acceptably secure systems.

5 Reconciling DoD Needs and the Marketplace

There are many ways that DoD programs can actively reduce the effect of the mismatch between the DoD and the commercial marketplace. The first is to find ways of increasing trust in COTS components in general. The second is through evaluation and negotiation. The third is vigilance: regardless of the promises or pitfalls of using COTS products, it is imperative that organizations establish a security policy and perform continuous threat assessment to determine the specific risks and identify applicable vulnerabilities. We examine each of these in detail below.

5.1 Increasing Trust in COTS Components

Learning about the COTS components the way that "hackers" learn them gives the integrator valuable insight and knowledge about potential vulnerabilities.

There are a number of steps that can be taken to increase trust in components. The first and easiest is to become aware of potential vulnerabilities in a component. Information about a component, and the potential risks that can come from the use of that component in a system can come from a variety of sources. These sources include everything from newsgroups (perhaps vendor moderated), product buglists described earlier, and independent reports from sanctioned advisory groups.

Another step that leads to increased trust is personal insight: know the component and know its capabilities. If the component itself has security enabling features, learn to use them. While third-party sources can give effective and low-cost initial insight, comprehensive testing of a component must still be done. Note that it is not sufficient to test and explore only those features of the components that are used in the integrated system, since such a subset does not typically represent the full vulnerability of the component (e.g., the Internet Worm incident described earlier in Section 3.2).

Learn about the inner workings of the components just as the hackers learn them. There are a number of techniques that can be used to gain insight into black-box components. There are techniques that use operating system services and shared library protocols to introspect and snoop the component's call stack and interfaces with other components. For example, the truss utility under UNIX makes it possible to observe the call interface between an executing component and the operating system. For example, if a component were to make an unexpected socket() call to an unknown network address it would raise suspicion about the integrity of the component.

Finally, isolate potentially vulnerable components from the rest of the system. A basic approach is to use a firewall, which essentially shields unnecessary or unwanted data or services between two or more networks. A component firewall only permits access, control, or information between a known and tested component service and the rest of the system. TCP Wrapper (a public domain component) is an example of a component firewall that isolates UNIX-native network services (telnet, ftp). Systems that have installed TCP Wrapper can effectively limit access to the system to network hosts that are known and authorized to connect. The services themselves are left unchanged, but are layered from the operating system's direct control in launching them.

5.2 Evaluate and Negotiate

Be prepared to evaluate the available technologies and components in the COTS marketplace. Weigh the opportunities against requirements and negotiate where appropriate.

Know the marketplace and the market imperatives that drive COTS component vendors. They compete for marketshare by differentiating products from those of their competitors; their goal is to attract more sales.

Some vendors do produce components that conform to Government requirements and standards. Over the last three years, according to the Evaluated Products List from the National Security Agency[5], approximately 30 products from a handful of vendors have been submitted and evaluated as level C2 or greater. As an example, a few vendors of POSIX-complaint operating systems sell specialized versions of their products (commonly referred to as Compartmented Mode Workstations (CMW)). Some major database vendors also have level C2, or greater, versions of their products, all of which have been evaluated as a trusted. But on the whole, the cost for TCSEC evaluation is prohibitively expensive for most vendors when compared to the relatively small market for such products. This is evidenced by the relatively small number of evaluated components.

Be prepared to make engineering trade-offs. Even in a carefully written set of requirements, some might be ill-stated, overgeneralized, or written without knowledge of commercial opportunities. Requirements that specify security attributes that cannot be met through COTS components should be examined and possibly reconsidered: it may be the case that requirements have been overstated. For instance, it is not unreasonable today to require that a WWW-based system be compatible with all commercially available browsers, and be secure from eavesdropping. But as recently as 1996, such a requirement would have been less reasonable. At that time, the popular Netscape WWW servers, which used 128-bit keys for Secure Socket Layer (SSL v3.0), could not interoperate with every commercial network browser. At this time, it would have been an unattainable requirement for a system to support all commercial browsers and use 128-bit encryption keys for secure communications. A typical trade-off made at the time was to regard security as a higher concern than interoperability, and negotiate a change in requirements to support only Netscape browsers.

5.3 Security Policies and Threat Analysis

Assume each COTS component is a potential source for vulnerabilities, and perform security risk assessments based on that assumption.

In the context of using COTS components, achieving system security lies somewhere between two extreme positions. One extreme is that no commercially available components are trustworthy, and therefore COTS should never be considered in designing and developing DoD systems. The other extreme is that the COTS component marketplace provides fully secure components that can meet all of DoD's needs.

Neither of these extremes is accurate. Before a decision to avoid the use of COTS or to embrace COTS as a security solution is made, the actual security risk to the system should be identified. Typically this is done through a security risk assessment, which consists of identifying the following:

1. Security policy: a formal statement of the rules by which people who are given access to an organization's technology and information assets must abide.[6]
2. Scope of the Assessment: some defined boundary to which the assessment will and will not address (electronic security (computers and networks) but not physical security (buildings and employees)).
3. Usage scenarios: system execution threads and functions under which the system is intended to function.
4. Assets: the data, information, and property that are under the protection of the system under assessment.
5. Threat Agent: identification of persons with an interest in obtaining access or information, modifying information, or interrupting services in a malicious or unauthorized manner (i.e., perpetrator).

In performing a security risk assessment, be suspicious. That is, immediately assume that the likelihood of compromise for any particular COTS component is high. In other words, it is probably easy to compromise the component and difficult to detect the compromise. By taking this worst-case perspective, it simplifies that task of assessing the actual impact upon the system. Consider the following:

- If the impact to the system is low, then regardless of how easily it is to compromise the COTS component, the end result (of the compromise) is of little consequence to the system as a whole.
- If the impact is high, but the countermeasure(s) are effective and affordable, then the potential for compromise is, again, of little consequence to the system as a whole.

When a security risk assessment has been performed, it is then possible to decide whether to use a COTS component or not. Without such evidence, no informed decision can be made. With such evidence, it is possible to know with some certitude what effect a component may have on the overall security profile of the system.

6 Summary

There is no simple answer to what security means in the context of COTS-based systems. For the purpose of this Monograph, we have examined security in the context of two orthogonal issues

- The essential security characteristics (e.g., integrity) of the COTS components themselves,
- The security requirements that must be satisfied by DoD systems.

There are many pressures on DoD managers and system integrators to use COTS software components. At the same time, the marketplace imperatives that drive the component vendors can sometimes contradict the longstanding needs of DoD systems. If we hope to make optimum use of commercial products and yet maintain DoD's high security requirements, we need greater insight into how to reconcile these contradictions. Testing and investigation of the components under evaluation can provide better insight into commercial components, and can aid the integrator in building suitably secure systems. Knowledge of the COTS marketplace and of the laws and forces that drive the vendor into making marketing decisions will help designers and users to evaluate and negotiate system requirements against component opportunities. Finally, good security practices can remove some of the obscurity about the vulnerabilities about COTS components and their role in the security of the overall system.

7 References

[1] Oberndorf, P., and Carney, D., "A Summary of DoD COTS-Related Policies", SEI Monographs on the Use of Commercial Software in Government Systems, Software Engineering Institute, Carnegie Mellon University, September 15, 1998

[2] http://www.cert.org/faq/cert_faq.html#A1

[3] Hissam, S. "Experience Report: Correcting System Failure in a COTS Information System" in Proceedings of the International Conference on Software Maintenance (1998), IEEE Computer Society Press, pp: 170-176, Los Alamitos, CA., 1998.

[4] Hissam, S., Carney, D., "Isolating Faults in Complex COTS-Based Systems", SEI Monographs on the Use of Commercial Software in Government Systems, Software Engineering Institute, Carnegie Mellon University, February 1998, http://www.sei.cmu.edu/cbs/papers/monographs/isolating-faults/isolating.faults.htm

[5] National Computer Security Center, "Evaluated Products List Indexed by Rating" December 1998, http://www.radium.ncsc.mil/tpep/epl/epl-by-class.html

[6] Fraser, B., Site Security Handbook, September 1997. http://www.faqs.org/rfcs/rfc2196.html

Feedback

Comments or suggestions about these monographs are welcome. We want this series to be responsive to the real needs of government personnel. To that end, comments concerning inclusion of other topics, the focus of the papers, or any other issues, will be of great value in continuing this series of monographs. Comments should be sent to:

Editor
SEI Monographs on COTS
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213
cots@sei.cmu.edu