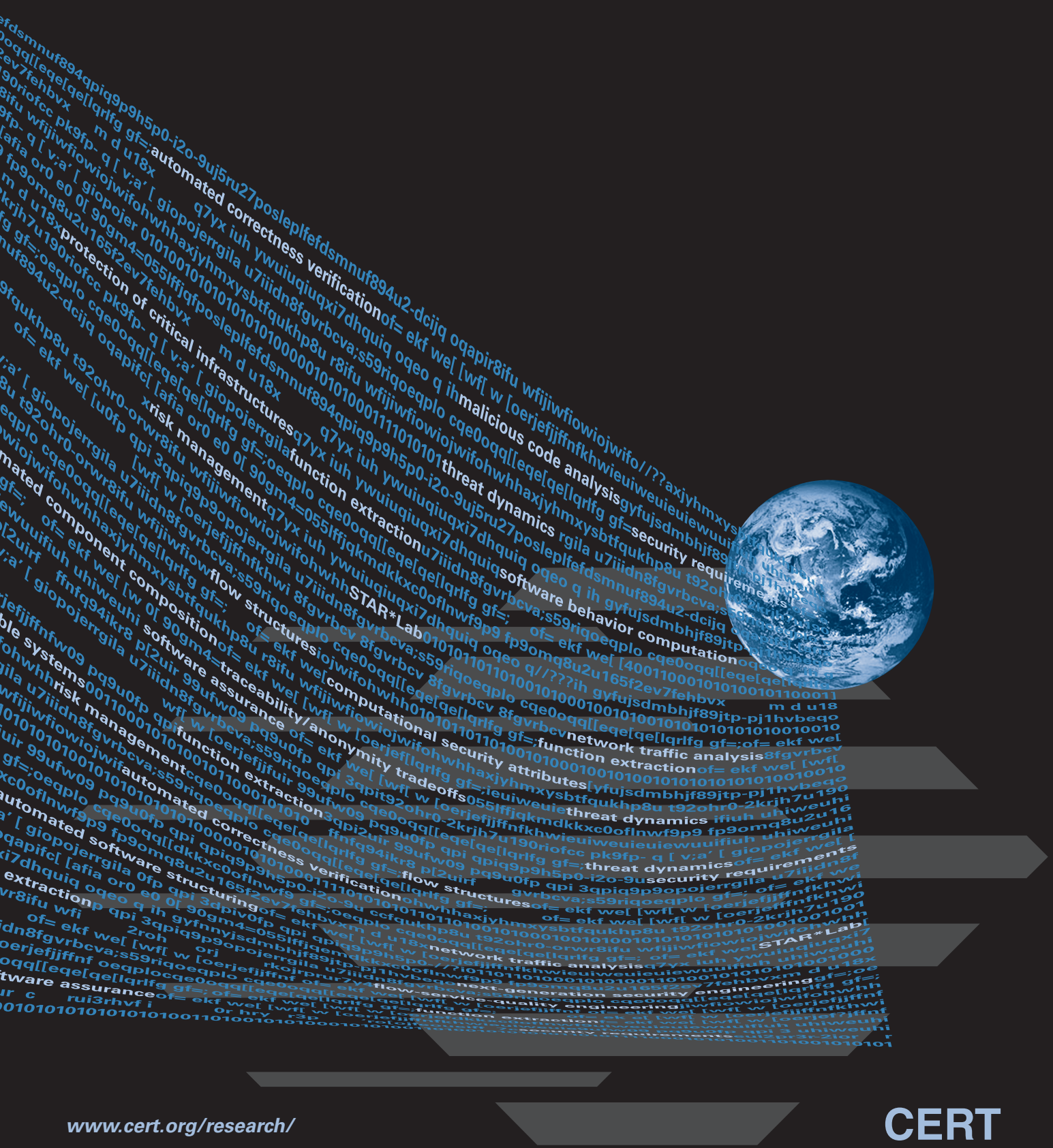
CERT Research

2005 Annual Report



CERT Research and the Rapidly Evolving Threat

Thomas Longstaff
Deputy Director of Technology, CERT Program

For many years, security on the Internet has been a rapidly escalating arms race between automated malware (worms, viruses, and web) and commercial security product updates that detect and block attacks. Vendors no longer have the luxury of time in preparing fixes for these newly propagated threats. What is worse, advances in phishing, pharming, and other attacks that blend automated threats with social engineering have forced security professionals to search for yet another set of signatures and firewall features that will prove effective.

The CERT® Program has been involved in this arms race at many levels, from analysis of vulnerabilities and malware to shared resolution of vendor problems across competitors. We have trained system administrators to be more knowledgeable, helped large enterprises evaluate their security improvement processes, and assisted system professionals in understanding the traffic on their networks. The CERT web site (http://www.cert.org) provides details on these activities and many others related to this work.

CERT researchers are looking beyond these current approaches toward a next-generation approach to security engineering. Our driving vision is a networked world in which software and systems can be understood far better and faster than is possible today, not only as they typically behave but how they always behave. In this view of the future, system responses to attacks, accidents, and failures are simply modes of their programmed behavior, modes that must be more thoroughly designed and analyzed than is practical with today's traditional methods. Next-generation security engineering will require automated support for this new level of behavior analysis. CERT projects on function extraction and secure coding, for example, help build a more complete understanding of how systems will behave before they are delivered. Similarly, automation for analyzing malicious code will speed the development of effective countermeasures.

Beyond today's world, we visualize software in new generations of ubiquitous computing and communication products, many of which we will not immediately recognize as networked computers. Cell phones provide a glimpse into this developing world by placing in our hands a combined audio-visual communications device, network browser, secure purchasing agent, geographic locator, gaming and entertainment system, and trusted wallet. We will see similar capabilities in cars, homes, offices, and more systems unimaginable today. All will be enabled by software and, in our vision, all will be subject to a full understanding of their behavior as a basis for engineering security into their operational features.

While system components will become smaller and more ubiquitous, systems themselves will become larger and more complex through integration and interconnection. Large-scale systems will continue to evolve and become ever more essential to modern society. In these systems, it is not only important to know the behavior of individual components but also of the assembled systems as whole entities. The CERT Research group is also focused on the integrated enterprise and (larger) environments of tomorrow. Our goal is to ensure that as these systems grow we approach a securely connected world, not untrustworthy networks of untrustworthy applications.

The vision is clear, but the path to achieving it is full of challenges. The research performed at CERT must take advantage of the most advanced theory and practice available, yet be flexible and adaptable in addressing a rapidly changing set of problems and constraints in the real world. These challenges have led to formation of the CERT Security Technology Automation and Research Laboratory (STAR*Lab), a new software development laboratory that will move concepts from theory to application to practice in a rapid and integrated approach. STAR*Lab researchers are dedicated to making a difference in the networked environments of the future through development of theory-based security engineering tools, not just through studies and publications that stop short of implemented solutions.

For CERT Research, 2006 will be a year of change as we focus on the problems in the computing environments of our customers and collaborators while maintaining the rigorous research approaches embraced by our scientists. This report begins to chronicle this change, and we hope you will find the results of our work as exciting as we find the journey.

Executive Summary

CERT® Research 2005

Reporting on the period ending September 30, 2005

CERT concentrates on the technical basis for identifying and preventing security flaws and for preserving essential services if a system is penetrated and compromised, and provides guidance to help organizations improve the security of networked computer systems. Our agenda consists of three elements: research, technology development, and technology transfer.

In our research activities, we aim to replace informal methods with precise software, security, and survivability engineering. In our technology development work, we create software, security, and survivability standards, technologies, and automation. In technology transfer, we attempt to incorporate results into key acquisition and development projects.

While all of these elements are necessary to achieve success, the focus of this report is on CERT's research work. Our research agenda is driven by the need to develop theoretical foundations and engineering methods to ensure the security and survivability of critical systems. We believe the projects described in this report are necessary elements in support of this agenda. We provide brief abstracts for our major research projects, followed by more detailed descriptions of the projects.

Executive Summary

LEVANT

The LEVANT (Levels of Anonymity and Traceability) project is addressing the engineering challenge of balancing the apparently conflicting needs of privacy and security with respect to the traceability of cyber attacks. LEVANT researchers are investigating the feasibility of a disciplined engineering design of Internet protocols (in the context of key policy issues) to allow optimal, fine-grained tradeoffs between traceability and anonymity to be made on the basis of specific mission requirements.

Scan Detection Using Bayesian Methods

Given that scans are an indicator of malicious activity, it is important that they be detected as part of an overall security strategy. This project deals with the detection of a scan when only unidirectional flow-level information is available and the internal configuration of the network is not known. These restrictions represent the operating conditions present on many large networks, such as those at ISPs and large organizations. Project researchers have developed the first system that, as opposed to simple thresholding approaches, actually models scans in this restrictive environment. The model, based on a combination of expert opinion and data analysis, uses multiple metrics in order to prevent adversaries from easily avoiding detection by adjusting their scan parameters to below a given threshold.

Security Quality Requirements Engineering (SQUARE)

It is well recognized in industry that requirements engineering is critical to the success of any major development project. Security requirements, however, tend to be developed independently of the rest of the requirements engineering activity. As a result, security requirements that are specific to the system and that provide for protection of essential services and assets are often neglected.

Through the SQUARE Project, CERT research has developed an end-to-end process for security requirements engineering. The SQUARE methodology generates a final deliverable of categorized and prioritized security requirements. The methodology has been enhanced and refined through application in case studies, and CERT researchers are working toward creating a CASE tool to support each of its stages.

STAR*Lab

STAR*Lab is an internal software development laboratory that CERT has established to create theory-based prototype automation that provides operational solutions to challenge problems in security and software engineering. STAR*Lab currently has two projects underway, Function Extraction for Malicious Code and Computational Security Attributes, and four potential projects, Automated Structuring for Understanding Legacy Systems, Automated Correctness Verification for Developing Reliable Software, Automated Component Composition for Developing Reliable Systems, and Flow-Service-Quality Engineering.

Function Extraction for Malicious Code

Because malicious code employs increasingly sophisticated intrusion strategies, analysts must understand all possible behaviors of the code in order to develop effective responses. While modern software engineering tools such as model checkers and testing and analysis tools provide useful information for specific cases of behavior, what is needed is an "all cases of behavior" view of what malicious code does. To help address this need, CERT is conducting research and development on the emerging technology of function extraction (FX). The goal of this project is to develop a system that computes the behavior of malicious code expressed in Intel assembly language, to help security analysts to quickly determine intruder objectives and strategies.

Computational Security Attributes

In the current state of practice, security properties of software systems are assessed through error-prone, labor-intensive human evaluation. The result is imprecise and a priori assertions that can be of limited value in the dynamics of system operation in threat environments where security attributes can change quickly. This project focuses on automated analysis of security properties, with the ultimate goal of providing foundations to help transform security engineering into a theory-based computational discipline.

Automated Structuring for Understanding Legacy Systems

The difficulty of maintaining and understanding legacy systems is often compounded by the unstructured, spaghetti-logic nature of the code. However, a well-structured program can be correctly understood as a composition of parts, each of which can be understood on its own. Legacy systems, compiled and optimized code, and intentionally obfuscated code often lack such structure, making it extremely difficult to understand their true control flow and functionality. The objective of this project is to create engineering systems for transforming the complex, arbitrary control flow of programs into function-equivalent structured form for improved understanding and analysis.

Automated Correctness Verification for Developing Reliable Software

In the current state of practice in software engineering, no practical means exists for automated, large-scale correctness verification of software with respect to intended behavior. As a result, much time and energy is devoted to inspection and testing activities that can provide only limited evidence of correctness. The objective of this project is to develop a proof-of-concept prototype of a function verification system that will analyze the correctness of programs.

Automated Component Composition for Developing Reliable Systems

Modern systems are characterized by large-scale heterogeneous networks with many components that must be correctly integrated to achieve mission objectives. It is often the case that the components are complex systems in their own right and must be dynamically combined to provide end-to-end capabilities. System integration today is a complex, labor-intensive process that can take months or even years for large systems. The objective of this project is to develop a proof-of-concept prototype of a component composition system that will automatically determine the net effect of combining components in network architectures.

Flow-Service-Quality (FSQ) Engineering

FSQ engineering provides foundations for mastering complexity and improving survivability in analysis and development of large-scale, network-centric systems. The FSQ project is defining rigorous engineering methods for developing complex systems that are characterized by shifting boundaries and users, uncertain functionality and security of commercial off-the-shelf (COTS) software, extensive asynchronous operations, unpredictable failures and compromises, and lack of visibility and control. Flow structures, a key element of FSQ engineering, help maintain intellectual control over complex system development.

Threat Dynamics

Threat dynamics is the study of the impact of an organization's threat environment on the ability of the organization to achieve its mission objectives. This project develops methods and tools that help model and analyze an organization's threat dynamics and improve the organization's security, survivability, and resiliency in light of those dynamics. The project's focus in 2005 was to apply the threat dynamics framework to the study of insider threat.

LEVANT: Protocols for Anonymity and Traceability Tradeoffs

Principal Investigators:
Howard Lipson and Sven Dietrich

Contact Points:
Howard Lipson 412-268-7237
Sven Dietrich 412-268-7711

Problem Addressed

Existing Internet protocols were never engineered for today's Internet, where the trustworthiness of users cannot be assumed and where high-stakes mission-critical applications increasingly reside. Malicious users exploit the severe weakness in existing Internet protocols to achieve anonymity and use that anonymity as a safe haven from which to launch repeated attacks on their victims. Hence, service providers and other victims of cyber attack want and need traceability for accountability, redress, and deterrence. Unfortunately, our current track-and-trace capability is limited in the extreme by the existing protocol and infrastructure design and requires a major re-engineering effort from both technical and policy perspectives, as described in an SEI special report sponsored by the U.S. State Department [1]. On the other hand, Internet users (both individuals and organizations) often want or need anonymity for a variety of legitimate reasons. The engineering challenge is to balance the apparently conflicting needs of privacy and security.

Research Approach

Traceability and anonymity are attributes that are central to the security and survivability of mission-critical systems. We believe that principled, fine-grained tradeoffs between traceability and anonymity are pivotal to the future viability of the Internet. However, such tradeoffs are rarely explicitly made, the current capability to make such tradeoffs is extremely limited, and the tradeoffs between these attributes have occurred on an ad hoc basis at best. The LEVANT (Levels of Anonymity and Traceability) project is developing the foundations for a disciplined engineering design of Internet protocols (in the context of key policy issues) to allow dynamic, fine-grained tradeoffs between traceability and anonymity to be made on the basis of specific mission requirements. We see this project as a first step toward the development of a discipline of Internet engineering, which would translate traditional design and engineering processes, such as thorough requirements gathering and attribute tradeoff analyses, into the unique context of the Internet environment and its associated security and survivability risks [2].

In any Internet transaction, trust ultimately depends not on IP addresses but on particular relationships among individuals and their roles within organizations and groups (which may be economic, political, educational, or social). Trust cannot be established while maintaining total anonymity of the actors involved. It goes without saying that there is a great need for privacy on the Internet, and it must be carefully guarded. However, trust and privacy tradeoffs are a normal part of human social, political, and economic interactions, and such tradeoffs can be resolved in a number of venues, such as in the marketplace. Consider

the telephone system, in particular the caller identification (caller ID) feature, which displays the phone number, and often the name, associated with incoming calls. Caller ID is a feature for which many customers are willing to pay extra in return for the privacy benefits associated with having some idea of who is calling before answering a call. However, callers are sometimes given the option of being anonymous (i.e., not identifiable by the caller ID feature) by default or on a call-by-call basis. To more fully protect their privacy, the caller ID customer can choose to block all incoming calls from anonymous callers. The anonymous caller is notified of this fact by an automated message. For callers that pre-arrange with their phone company to be anonymous by default, the only way to complete the call is to enter a key sequence to remove the anonymity for that particular call and to redial. Customers that achieve anonymity on a call-by-call basis (by entering a specific key sequence) can choose to redial without entering the key sequence that denotes anonymity. This choice is a form of negotiation between the caller and the intended recipient of the call, and it is a tradeoff between anonymity and trust that is supported by the technology of caller ID and the marketplace. There is no government mandate that all calls must be anonymous or that no calls are allowed to be anonymous. The individual caller chooses whether or not to relinquish anonymity (or some degree of privacy) in exchange for the perceived value of completing the call by increasing the degree of trust as seen by the recipient.

One can envision next-generation Internet protocols supporting this kind of marketplace negotiation of trust versus privacy tradeoffs. For example, we are exploring the possibility of third-party certifying authorities, which would serve as brokers of trust. These certifying authorities would provide mechanisms whereby packets would be cryptographically signed with very fine-grained authentication credentials of the sender. This is not the same as having an individual digitally sign a message, as a digitally signed message may be too coarse-grained for a particular scenario and may reveal too much. Another capability might be the escrowing, by these certifying authorities, of complete identifying information for a specified period of time, to be revealed in the event that one or more of a user's packets have been identified as participating in a confirmed attack.

We are investigating the fundamental concepts necessary to inform the design of Internet protocols that support dynamic, fine-grained tradeoffs between traceability and anonymity in a manner that satisfies the security, survivability, and anonymity (or privacy) requirements of the protocols' users. Our goal is to pro-

vide an exemplar for the application of principled software and systems engineering practices in the unique context of the Internet. A key part of this process is our exploration of alternative designs for new Internet protocols that allow the originator and the recipient of an Internet transaction or service to negotiate what levels of traceability and anonymity to accept.

In order to design and evaluate Internet protocols that support negotiated tradeoffs between anonymity and traceability, we need some way to quantify and measure levels of anonymity and traceability. The concept of *k*-anonymity provides some useful theoretical underpinnings.

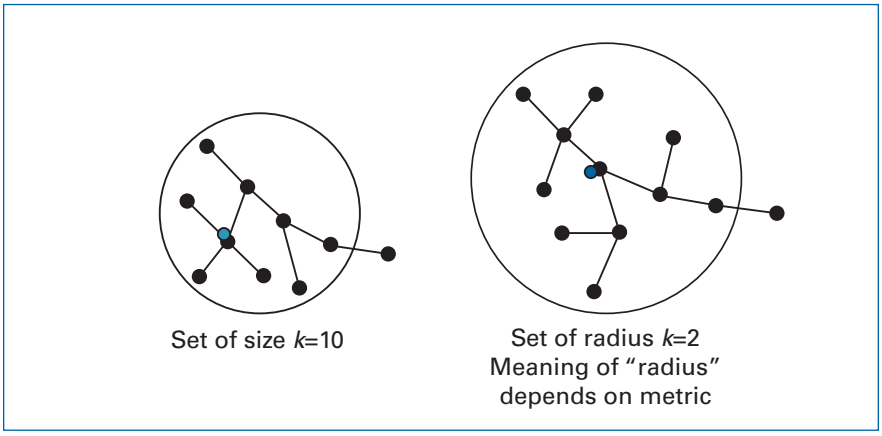


Figure 1: Examples of *k*-anonymity

Meaning of *k*-anonymity

We say that a user is *k*-anonymous in a network context if the user is only traceable to a set of measure *k*, where this could mean either a set of size *k* or a set of radius *k* in the topological sense of the network (as shown in Figure 1). Our goal is to explore the design of Internet protocols that assure traceability, but only to a group of *k* actors. Latanya Sweeney originally defined the notion of *k*-anonymity in the privacy context for medical patient data [3].

User and Service Provider Goals

Effective anonymity and traceability tradeoffs require an in-depth understanding of the specific goals of users and service providers. User goals may differ on a case-by-case basis. Some examples:

- User may want to hide its location and identity entirely (large *k*).
- User may want to hide its location somewhat (e.g., reveal the city but not street address).
- User may want to hide its location but not its identity.

Similarly, service providers may have different goals and/or requirements. Some examples:

- Provider may want to know both user's location and identity.
- Provider may want to know user's location somewhat.
- Provider may want to know user's identity but does not care about user's location.

It is important to note that the value of anonymity and traceability tradeoffs extends well beyond the relationships among individual users and service providers. The ability to explicitly make such tradeoffs can provide essential support for organizations engaged in a complex mix of collaborative and competitive (adversarial) relationships. Consider the scenario below.

LEVANT Scenario

Assume that a number of organizations collaborate to build a shared (highly distributed) knowledgebase that is more comprehensive and of higher quality than each could build on its own. This knowledgebase provides a significant competitive advantage for the collaborators over other organizations that are not participating in this effort.

Although the participating organizations collaborate on some projects, they are competitors in other areas. Each may use the knowledgebase to further its own strategies, tactical decisions, and so forth. Hence, each participating organization wants traceability in the event that the availability, integrity, or confidentially of the knowledgebase is compromised or threatened and to ensure that no external organizations get access to the data. Yet each organization wants its own members to be able to query the knowledgebase without revealing to the other collaborators (or, of course, to any outsider) the source of any query being made by that organization. LEVANT technology would provide network level protocol support for the traceability and anonymity tradeoffs that the collaborating organizations agree on, helping ensure the success of their cooperative *and* individual missions.

Some additional information on the LEVANT project is available in a summary report on SEI independent research and development projects [4].

Benefits

In this era of open, highly distributed, complex systems, vulnerabilities abound and adequate security, using defensive measures alone, can never be guaranteed. As with all other aspects of crime and conflict, deterrence plays an essential role in protecting society. Hence, the ability to track and trace attackers is crucial,

because in an environment of total anonymity, deterrence is impossible and an attacker can endlessly experiment with countless attack strategies and techniques until success is achieved. The ability to accurately and precisely assign responsibility for cyber attacks to entities or individuals (or to interrupt attacks in progress) would allow society's legal, political, and economic mechanisms to work both domestically and internationally to deter future attacks and motivate evolutionary improvements in relevant laws, treaties, policies, and engineering technology. On the other hand, there are many legal, political, economic, and social contexts in which some protection of anonymity or privacy is essential. Without some degree of anonymity or privacy, individuals or entities whose cooperation is vitally needed may not fully participate (or participate at all) in the use or operation of systems that support the critical functions of the global information society.

Hence, traceability and anonymity are attributes that are central to the security and survivability of mission-critical systems. The LEVANT project is exploring the essential engineering and policy issues associated with traceability and anonymity tradeoffs. A primary objective is to design Internet protocols that allow dynamic, fine-grained tradeoffs between traceability and anonymity to be made on the basis of the specific mission requirements of the protocols' users. An ultimate benefit of these new Internet protocols will be dramatically improved security and traceability for mission-critical applications and infrastructures, along with strong privacy and anonymity protection for legitimate users that act either as individuals or within specific organizational roles.

2005 Accomplishments

Only very limited funding was available for the LEVANT project in FY2005, so much of the principal investigators' time was redirected to other CERT efforts. Albeit at a reduced pace, we continued our work towards establishing a solid theoretical foundation on which to base principled engineering tradeoffs between traceability and anonymity. Progress includes some further work on a conceptual model that helps clarify the relationships between anonymity and traceability. We expect the model to continue to evolve into a foundation for understanding and expressing the full range of engineering requirements for the design of Internet protocols that support attribute tradeoffs and negotiations, as well as help us to generate examples of specific user requirements for anonymity and traceability that must be satisfied for particular applications, systems, or missions. One of the key engineering requirements for the design of such protocols is that they

effectively support anonymity–traceability tradeoff negotiations between service providers and their clients. There are strong security and survivability themes in this research, both in the engineering tradeoffs being explored for protocol design and in the policy issues relating to the design and use of protocols that support levels of anonymity and traceability for individual actors and for organizations.

A highlight of FY2005 is that the principal investigators authored a successful research proposal for project-level CyLab funding for FY2006. The CyLab award goes beyond earlier graduate-level "seed" funding and will directly support the principal investigators' work on the LEVANT project for FY2006.

2006 Plans

With renewed and expanded CyLab funding, the project has resumed its fully active status in FY2006. One or more technical reports or papers describing our research results are planned for this fiscal year, along with additional research to further develop the conceptual model on which the anonymity and traceability tradeoff protocols will be based. We will also continue to explore several of the economic and public policy issues relevant to this research area.

References

[1] Lipson, Howard F. *Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues* (CMU/SEI-2002-SR-009, ADA408853). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.cert.org/archive/pdf/02sr009.pdf.

[2] Lipson, Howard F. & Fisher, David A. "Survivability – A New Technical and Business Perspective on Security." *Proceedings of the 1999 New Security Paradigms Workshop.* Caledon Hills, ON, Sept. 21-24, 1999. New York, NY: Association for Computing Machinery. http://www.cert.org/archive/pdf/busperspec.pdf.

[3] Sweeney, Latanya. "k-anonymity: A Model for Protecting Privacy." *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems 10, 5* (October 2002): 557-570.

[4] Lipson, H. & Dietrich, S. "Levels of Anonymity and Traceability (LEVANT)." In Bergey, J.; Dietrich, S.; Firesmith, D.; Forrester, E.; Jordan, A.; Kazman, R.; Lewis, G.; Lipson, H.; Mead, N.; Morris, E.; O'Brien, L.; Siviy, J.; Smith, D.; & Woody, C. *Results of SEI Independent Research and Development Projects and Report on Emerging Technologies and Technology Trends* (CMU/SEI-2004-TR-018), pp. 4-12. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. http://www.sei.cmu.edu/publications/documents/04.reports/04tr018.html.

Scan Detection Using Bayesian Methods

Principal Investigator:
Carrie Gates

Contact Point: Carrie Gates
412-268-4134

Problem Addressed

It is widely believed that, before attacking an organization's computer systems, an adversary will often perform a scan to determine potential weaknesses in his target. This premise was examined by Panjwani et al. [1], who found that 50% of directed attacks were preceded by a vulnerability scan, while much of the remainder consisted of having a scan and the attack bundled as a single activity (such as is observed in worms, for example). Given that scans are an indicator of malicious activity, it is therefore important that scans be detected as part of an overall security strategy.

Scans are a reconnaissance technique that is aimed at multiple targets. While there are many different forms of scans (for example, based on the protocol used and how that protocol might be abused), they all share the commonality of attempting to gain information about a service or set of services on one or more hosts. The pattern formed by the targets being scanned is known as a scan footprint, with the three most common footprints being a horizontal scan (one service across multiple hosts), strobe scan (multiple services across multiple hosts) and vertical scan (multiple services on a single host). The footprint of a scan can be used by a defender to determine the true targets of an adversary and what services they might potentially exploit.

Several methods have been developed to detect scans. One of the most effective approaches is that of Jung et al. [2], who, as a source initiates new connections, use sequential hypothesis testing to determine if the source is indeed performing a scan. The key indicator that a scan is being performed is that the source is attempting to access a host that does not exist. This approach has been incorporated into Bro [3], a commonly used open source intrusion detection system. However, this approach requires either the ability to observe both sides of a conversation (in order to determine if there is a response, or if the destination host does not exist) or an oracle that can be used to determine if the source is contacting a known host. There are other approaches that do not have this limitation, but they require packet-level information and are not as effective.

The problem we have addressed here is the detection of a scan given that only unidirectional flow-level information is available and the internal configuration of the network is not known. These restrictions represent the operating conditions present on many large networks, such as those at ISPs and large organizations.

Research Approach

Given that scans consist of multiple communication attempts from a single source, we approached detecting scans by first grouping all communications by source IP address and then analyzing each group as a separate event. We proceeded to take a statistical approach to analyzing the events, which incorporated expert opinion.

Given that each event (all communications from a single source IP address within some period of time) can be thought of as having a binary property—either it contains a scan or it does not—a logistic regression approach was used to determine the probability that a given event contained a scan. For TCP scans, 21 variables were identified as potentially indicating whether a scan was present, and so these 21 variables were initially used in the model.

Events were extracted for three different time periods, with each time period being used for a different phase—elicitation, training, or testing. For both the elicitation and training phases, the events that represented the most extreme behaviors were identified, using 100 events for the elicitation phase and 200 events for the training phase. The values for the 21 variables were extracted from the elicitation data, and these were provided to an expert for analysis. Using just this information, the expert provided a probability that the event contained a scan. This information was used to generate a prior distribution for the coefficients for each variable in the logistic regression model [4]. The events in the training set were then analyzed by the expert, using the flow data rather than the values for the variables, for the presence of scans. The resulting data was used to generate a logistic regression model whose coefficients were also influenced by the priors obtained in the elicitation phase. The Akaike Information Criterion (AIC) [5] was then applied to the model to reduce the number of variables to just those that contribute significantly to the detection of a scan given an event. The result was a reduced model consisting of six variables.

Three hundred events were randomly sampled from the time period reserved for testing, and each event was manually analyzed by an expert for the presence of port scans. The values for the six variables for each event were also calculated and provided to the logistic regression model, comparing the results to the expert analysis. Given the 300 events, there were only 22 scans and 278 non-scans. Of these, the model correctly recognized 21 scans and 277 non-scans, resulting in a detection rate of 95.5% and a false positive rate of 0.4%, using the conditional probabilities as defined by Axelsson for these calculations [6]. This approach was then repeated for both UDP and ICMP scans, with similar results.

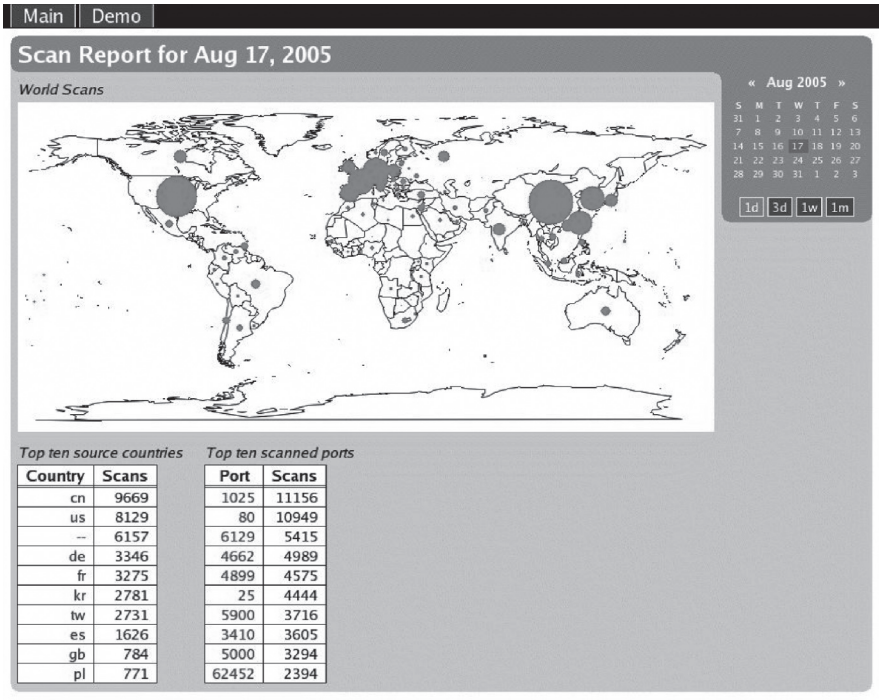


Figure 1: Example report on scanning activity

Expected Benefits

This is the first system that uses metrics, rather than simple thresholding, to detect the presence of a port scan given only unidirectional flow-level network traffic. The system can be deployed to recognize port scans on large networks, such as those at the ISP level.

There are two primary benefits that can be obtained from deploying this system:

1. The scan information can be saved to a database for later processing. In particular, this data can be used to determine changes in scanning trends that potentially indicate the presence of new vulnerabilities or exploits. Additionally, the data can be used to determine key targets by observing vertical scans or destination IP addresses that are more frequently targeted.
2. The scan information can be used to filter data. More specifically, given a large amount of network traffic, any scan can be extracted and stored to a different location. The result is a reduced amount of network traffic for a security analyst to process for other forms of malicious behavior.

2005 Accomplishments

This scan detection system has been deployed at a large client site and has been detecting TCP scans for several months, recording the results to an Oracle database. Additionally, both a command line interface and a web portal have been designed to allow analysts easy access to the database, without requiring SQL knowledge. The command line provides analysts with the ability to see scans meeting particular characteristics and to drill down into the details of particular scans. The web portal provides analysts with the ability to view graphs that indicate trending information. An example screen shot from the web portal is provided in Figure 1.

References

[1] Panjwani, Susmit; Tan, Stephanie; Jarrin, Keith M.; & Cukier, Michel. "An Experimental Evaluation to Determine if Port Scans Are Precursors to an Attack," 602-611. *Proceedings of the 2005 International Conference on Dependable Systems and Networks.* Yokohama, Japan, June 28-July 1, 2005. Los Alamitos, CA: IEEE Computer Society Press, 2005.

[2] Jung, Jaeyeon; Paxson, Vern; Berger, Arthur W.; & Balakrishnan, Hari. "Fast Portscan Detection Using Sequential Hypothesis Testing," 211-225. *Proceedings of the 2004 IEEE Symposium on Security and Privacy.* Oakland, CA, May 9-12, 2004. Los Alamitos, CA: IEEE Computer Society Press, 2004.

[3] Paxson, Vern. "Bro: A System for Detecting Network Intruders in Real-Time." *Proceedings of the 7th USENIX Security Symposium.* San Antonio, Texas, January 26-29, 1998. Berkeley, CA: USENIX Association, 1998.

[4] Agresti, Alan. *Categorical Data Analysis.* New York, NY: John Wiley and Sons, 2002.

[5] Akaike, H. "Information Theory as an Extension of the Maximum Likelihood Principle," 267-281. *Proceedings of the 2nd International Symposium on Information Theory.* Budapest, Hungary, 1973. Los Alamitos, CA: IEEE Computer Society Press, 1973.

[6] Axelsson, Stefan. "The Base-Rate Fallacy and the Difficulty of Intrusion Detection." *ACM Transactions on Information and System Security 3,* 3 (August 2000): 186-205.

SQUARE: Requirements Engineering for Improved System Security

Principal Investigator:
Nancy R. Mead

Contact Point: Nancy R. Mead
412-268-5756

Problem Addressed

It is well recognized in industry that requirements engineering is critical to the success of any major development project. Several authoritative studies have shown that requirements engineering defects cost 10 to 200 times as much to correct once fielded than if they were detected during requirements development. Other studies have shown that reworking requirements, design, and code defects on most software development projects costs 40 to 50 percent of total project effort, and the percentage of defects originating during requirements engineering is estimated at more than 50 percent. The total percentage of project budget due to requirements defects is 25 to 40 percent.

A recent study found that the return on investment when security analysis and secure engineering practices are introduced early in the development cycle ranges from 12 to 21 percent, with the highest rate of return occurring when the analysis is performed during application design. The National Institute of Standards and Technology (NIST) reports that software that is faulty in security and reliability costs the economy $59.5 billion annually in breakdowns and repairs [1]. The costs of poor security requirements show that even a small improvement in this area would provide a high value. By the time that an application is fielded and in its operational environment, it is very difficult and expensive to significantly improve its security.

Requirements problems are among the top causes of why projects

- are significantly over budget
- are significantly past schedule
- have significantly reduced scope
- deliver poor-quality applications
- are not significantly used once delivered
- are cancelled

Security requirements are often identified during the system life cycle. However, the requirements tend to be general mechanisms such as password protection, firewalls, virus detection tools, and the like. Often the security requirements are developed independently of the rest of the requirements engineering activity, and hence are not integrated into the mainstream of the requirements activities. As a result, security requirements that are specific to the system and that provide for protection of essential services and assets are often neglected.

In reviewing requirements documents, we typically find that security requirements, when they exist, are in a section by themselves and have been copied from a generic set of security requirements. The requirements elicitation and analysis that is needed to get a better set of security requirements seldom takes place.

Much requirements engineering research and practice has addressed the capabilities that the system will provide. So a lot of attention is given to the functionality of the system, from the user's perspective, but little attention is given to what the system should *not* do. In one discussion on requirements prioritization for a specific large system, ease of use was assigned a higher priority than security requirements. Security requirements were in the lower half of the prioritized requirements. This occurred in part because the only security requirements that were considered had to do with access control.

Research Approach

The Software Engineering Institute's Networked Systems Survivability (NSS) Program at Carnegie Mellon University has developed a methodology to help organizations build security into the early stages of the production life cycle. The Security Quality Requirements Engineering (SQUARE) methodology consists of nine steps that generate a final deliverable of categorized and prioritized security requirements. Although the SQUARE methodology could likely be generalized to any large-scale design project, it was designed for use with information technology systems.

The SQUARE process involves the interaction of a team of requirements engineers and the stakeholders of an IT project. It begins with the requirements engineering team and project stakeholders agreeing on technical definitions that serve as a baseline for all future communication. Next, business and security goals are outlined. Third, artifacts and documentation are created, which are necessary for a full understanding of the relevant system. A structured risk assessment determines the likelihood and impact of possible threats to the system.

Following this work, the requirements engineering team determines the best method for eliciting initial security requirements from stakeholders, which is dependent on several factors, including the stakeholders involved, the expertise of the requirements engineering team, and the size and complexity of the project. Once a method has been established, the participants rely on artifacts and risk assessment results to elicit an initial set of security requirements. Two subsequent stages are spent categorizing and prioritizing these requirements for manage-

Number	Step	Input	Techniques	Participant	Output
1	Agree on definitions	Candidate definitions from IEEE and other standards	Structured interviews, focus group	Stakeholders, requirements team	Agreed-to definitions
2	Identify security goals	Definitions, candidate goals, business drivers, policies and procedures, examples	Facilitated work session, surveys, interviews	Stakeholders, requirements engineer	Goals
3	Develop artifacts to support security requirements definition	Potential artifacts (e.g., scenarios, misuse cases, templates, forms)	Work session	Requirements engineer	Needed artifacts: scenarios, misuse cases, models, templates, forms
4	Perform risk assessment	Misuse cases, scenarios, security goals	Risk assessment method, analysis of anticipated risk against organizational risk tolerance, including threat analysis	Requirements engineer, risk expert, stakeholders	Risk assessment results
5	Select elicitation techniques	Goals, definitions, candidate techniques, expertise of stakeholders, organizational style, culture, level of security needed, cost benefit analysis, etc.	Work session	Requirements engineer	Selected elicitation techniques
6	Elicit security requirements	Artifacts, risk assessment results, selected techniques	Joint Application Development (JAD), interviews, surveys, model-based analysis, checklists, lists of reusable requirements types, document reviews	Stakeholders facilitated by requirements engineer	Initial cut at security requirements
7	Categorize requirements as to level (system, software, etc.) and whether they are requirements or other kinds of constraints	Initial requirements, architecture	Work session using a standard set of categories	Requirements engineer, other specialists as needed	Categorized requirements
8	Prioritize requirements	Categorized requirements and risk assessment results	Prioritization methods such as Triage, Win-Win, etc.	Stakeholders facilitated by requirements engineer	Prioritized requirements
9	Requirements inspection	Prioritized requirements, candidate formal inspection technique	Inspection method such as Fagan, peer reviews, etc.	Inspection team	Initial selected requirements, documentation of decision-making process and rationale

Table 1: Security requirements elicitation and analysis process

ment's use in making tradeoff decisions. Finally, an inspection stage is included to ensure the consistency and accuracy of the security requirements that have been generated.

SQUARE is a work in progress. Several case studies with real-world clients have shown that the methodology holds good promise for incorporation into industry practice. The SQUARE process has been enhanced and refined throughout the case studies. The current working model is summarized in Table 1. NSS is currently continuing research and application of the process and is working in parallel to create a CASE tool to support each stage of the methodology.

Expected Benefits

When SQUARE is applied, the user should expect to have identified and documented relevant security requirements for the system or software that is being developed. SQUARE may be more suited to a system under development or one undergoing major modification than one that has already been fielded, although it has been used both ways.

2005 Accomplishments

A workshop on Software Engineering for Secure Systems (SESS05) was held in conjunction with the International Conference on Software Engineering on May 15-16, 2005. The SQUARE method was presented at the workshop [4]. SQUARE is also described in the requirements engineering section of the Build Security In web site [5], and applied in a series of client case studies. Carnegie Mellon University graduate students worked on this project during the summer and fall of 2005. Some previous case study results were published [6] and a definitive technical report on SQUARE was produced [7]. In conjunction with Cylab, the prototype tool is being enhanced to be more modular and to have a more appealing user interface.

The methodology is most effective and accurate when conducted with a team of requirements engineers with security expertise and the stakeholders of the project. The requirements engineering team can be thought of as external consultants, though often the team is composed of one or more internal developers of the project. The effectiveness of SQUARE in eliciting requirements is dependent on representation from the project's stakeholders. Thus, the requirements engineering team must emphasize the importance of establishing a representative set of stakeholders to participate in the methodology.

SQUARE can be decomposed into nine discrete steps, which are outlined in Table 1. Each step identifies the necessary inputs, major participants, suggested

techniques, and final output. Generally, the output of each step serves as the sequential input to the following steps, though some steps may be performed in parallel. For instance, it might be more efficient for the requirements engineering team to perform Step 2 (Identify Security Goals) and Step 3 (Develop Artifacts) simultaneously, since to some extent they are independent activities. The output of both steps, however, is required for Step 4 (Perform Risk Assessment). In principle, Steps 1-4 are actually activities that precede security requirements engineering but are necessary to ensure that it is successful.

2006 Plans

The team plans to refine and continue to pilot requirements elicitation and analysis methods for security properties. Prototype tools development will continue in conjunction with the process application for leading-edge clients. SQUARE will also be proposed for software and security-related international standards. A book chapter on SQUARE is in progress.

References

[1] National Institute of Standards and Technology. "Software Errors Cost U.S. Economy $59.5 Billion Annually" (NIST 2002-10). http://www.nist.gov/public_affairs/releases/n02-10.htm (2002).

[2] Linger, R. C.; Mead, N. R.; & Lipson, H. F. "Requirements Definition for Survivable Systems," 14-23. *Third International Conference on Requirements Engineering.* Colorado Springs, CO, April 6-10, 1998. Los Alamitos, CA: IEEE Computer Society, 1998.

[3] Mead, N. R. *Requirements Engineering for Survivable Systems* (CMU/SEI-2003-TN-013). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2003. http://www.sei.cmu.edu/publications/documents/03.reports/03tn013.html.

[4] Mead, N. R. & Stehney, T. "Security Quality Requirements Engineering (SQUARE) Methodology." *Software Engineering for Secure Systems (SESS05), ICSE 2005 International Workshop on Requirements for High Assurance Systems.* St. Louis, MO, May 15-16, 2005. http://homes.dico.unimi.it/%7Emonga/sess05.html.

[5] https://buildsecurityin.us-cert.gov/

[6] Gordon, D.; Stehney, T.; Wattas, N.; & Yu, E. *Quality Requirements Engineering (SQUARE): Case Study on Asset Management System, Phase II* (CMU/SEI-2005-SR-005). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05sr005.html.

[7] Mead, N.R., Hough, E. & Stehney, T. *Security Quality Requirements Engineering (SQUARE) Methodology* (CMU/SEI-2005-TR-009). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tr009.html.

STAR*Lab A Software Development Laboratory for Security Technology Automation and Research

Developing Software Solutions to Security Challenge Problems

In response to the growing needs of its customers, CERT has established an internal software development laboratory. The mission of the laboratory is the creation of theory-based prototype automation that provides operational solutions to challenge problems in security engineering. Challenge problems are long-standing barriers to progress that have been identified by DoD and other organizations, whose solutions can have substantial impact on engineering capabilities. The laboratory applies sound theoretical foundations to create automated engineering tools that practitioners can apply to challenge problems of system security and dependability. The focus of STAR*Lab is not on studies and reports that may leave implementation speculative and undone, but rather on applied technology expressed through concrete instantiation in working tools. The laboratory is dedicated to helping CERT customers achieve three objectives:

Faster development: Solutions must replace time- and resource-intensive operations with engineering automation that permits faster system development.

Improved quality: Solutions must substitute foundations-based automation for fallible human processes to improve system security and dependability.

Fewer resources: Solutions must increase the span of intellectual control through automation to support effective use of resources in developing secure systems.

STAR*Lab Operating Principles

The laboratory operates according to three principles in developing engineering solutions:

The foundations-first principle. Solid theoretical foundations are necessary to ensure completeness and correctness in automated engineering solutions and confidence in the results they produce. All projects will start with sound foundations to avoid ad hoc solutions with limited applicability.

The proof-by-automation principle. Automation is essential to replace fallible and resource-intensive human operations with solutions that permit full intellectual control. All projects will demonstrate solutions through automated engineering tools.

The practical application principle. Automation must transform challenge problems into practical engineering operations that scale up for routine use by practitioners. All projects will scale up engineering solutions for practical application.

The STAR*Lab Development Model

STAR*Lab projects are managed within a gated review structure designed to maintain visibility, reduce risk, and ensure effective use of resources on behalf of sponsors. Projects must satisfy the requirements of each gate in order to receive funding to progress to the next gate:

Gate 1: Challenge problem definition. Each project must address a well-defined barrier to progress through a comprehensive project plan that defines tasks, schedules, and resources.

Gate 2: Theoretical feasibility. Each project must identify theoretical foundations for a challenge problem solution, to avoid ad hoc approaches of limited value for achieving comprehensive and confident application.

Gate 3: Proof-of-concept automation. Each project must develop prototype automation that demonstrates application of the theoretical foundations to address the challenge problem.

Gate 4: Scale-up for application. Each project must evolve the prototype automation to scale up engineering capabilities for routine and widespread application.

STAR*Lab Projects

STAR*Lab currently has two projects underway, which are described in the sections below:

Function Extraction for Malicious Code
This multiyear, customer-sponsored project has satisfied the requirements of Gate 2 and is proceeding to Gate 3.

Computational Security Attributes
This SEI-sponsored 2006 R&D project has satisfied the requirements of Gate 1 and is proceeding to Gate 2.

In addition to these ongoing projects, STAR*Lab stands ready to undertake any challenge problem projects to meet the needs of CERT sponsors and collaborators.

The following challenge problem areas are of interest because they can take advantage of technologies being developed in the Function Extraction project. They are described as potential STAR*Lab projects in the sections below.

Automated Structuring for Understanding Legacy Systems

This potential project has satisfied the requirements of Gate 3 and is ready to proceed to Gate 4.

Automated Correctness Verification for Developing Reliable Software

This potential project has satisfied the requirements of Gate 2 and is ready to proceed to Gate 3.

Automated Component Composition for Developing Reliable Systems

This potential project has satisfied the requirements of Gate 1 and is ready to proceed to Gate 2.

Flow-Service-Quality (FSQ) Engineering for Developing Network Systems

This potential project has satisfied the requirements of Gate 2 and is ready to proceed to Gate 3.

STAR*Lab

Function Extraction for Malicious Code (FX/MC)

Automated Calculation of Program Behavior for Security Analysis

Principal Investigator:
Richard C. Linger

Contact Point: Richard C. Linger
301-926-4858

Problem Addressed

As the volatility of malicious code on the Internet increases, fast and reliable understanding of what the code is doing becomes critical for developing timely countermeasures. But malicious code analysis today requires laborious code reading by security experts that can take days of effort, delaying an effective response.

How can this analysis be made faster and more reliable? Because malicious code employs increasingly sophisticated intrusion strategies, analysts must understand all possible behaviors of the code in order to develop effective responses. While modern software engineering tools such as model checkers and testing and analysis tools provide useful information for specific cases of behavior, what is needed is an "all cases of behavior" view of what malicious code does. To help address this need, CERT is conducting research and development on the emerging technology of function extraction (FX).

FX technology applies function-theoretic foundations of software to automate calculation of the behavior of malicious code to the maximum extent possible. Computing the behavior of code requires deriving its net functional effect, that is, how it transforms inputs into outputs in all circumstances of use. That information can be presented to analysts in behavior catalogs that define all the possible effects a program can have, essentially, the "all cases of behavior" view. The ultimate objective is to move from an uncertain understanding of malicious code derived in human time scale (days) to a precise understanding computed in machine time scale (seconds).

CERT has initiated a project to develop the Function Extraction for Malicious Code (FX/MC) system. The goal of FX/MC is to analyze the behavior of malicious code expressed in Intel assembly language, to help security analysts to quickly determine intruder objectives and strategies.

Research Approach

The function-theoretic model of software treats programs as rules for mathematical functions or relations, that is, mappings from domains (inputs, stimuli) to ranges (outputs, responses), no matter what subject matter they deal with [1, 2]. The key to the function-theoretic approach is the recognition that, while programs can contain an intractable number of execution paths, they are at the same time composed of a finite number of control structures, each of which implements a mathematical function or relation in the transformation of its inputs into outputs. In particular, the sequential logic of programs can be composed of single-entry,

single-exit composition, alternation, and iteration control structures, plus variants and extensions. This finite property of program logic viewed through the lens of function theory opens the possibility of automated calculation of program behavior. Every control structure in a program has a non-procedural behavior signature that defines its input-to-output transition function. Each behavior signature can be extracted and composed with others in a stepwise process that traverses the control structure hierarchy. The overall behavior signature of a program represents the specification that it implements. These concepts are a key element of function extraction technology [3]. Automated computation of software behavior is a difficult problem that requires innovative approaches. For example, while no comprehensive mathematical theory for loop behavior computation can exist, engineering solutions are feasible and under development.

Expected Benefits

FX/MC is expected to help analysts to quickly determine intruder strategies by providing precise information on the structure and function of malicious code. Successive versions of the system will provide increasing capabilities for malicious code analysis.

Beyond the application of FX to malicious code analysis, it is a formidable task to achieve security goals for systems without knowing what their programs do in all circumstances of use. In the current state of practice, this knowledge is sporadically accumulated from specifications, designs, code, and test results. And ongoing program maintenance and evolution often limit the relevance of this hard-won knowledge. However, programs are mathematical artifacts subject to mathematical analysis. Human fallibility still exists in interpreting the analytical results, but there can be little doubt that routine availability of calculated behavior would substantially reduce errors, vulnerabilities, and malicious code in software and make intrusion and compromise more difficult and detectable. In addition, broader questions about security capabilities for authentication, encryption, filtering, etc., are in large part questions about the behavior of the programs that implement these functions.

2005 Accomplishments

An essential first step in malicious code analysis is to transform its intentionally obfuscated, spaghetti-logic control flow into readable, structured form. Based on a structure theorem, this transformation helps analysts to understand the program logic and establishes a suitable foundation for behavior calculation. The result is a function-equivalent version that is traceable to the original program and can be used for program understanding and comparison. The initial version of FX/MC that provides this capability was completed in 2005.

To explore the potential of FX technology, STAR*Lab developed a proof-of-concept prototype that calculates the behavior of programs written in a small subset of Java and presents it to users in the form of behavior catalogs. A rigorous experiment was conducted to compare traditional methods of program reading and inspection with FX-based methods. Experienced programmers were divided into a control group using traditional techniques and an experimental group using the FX prototype. Each group answered questions dealing with comprehension and verification of three Java programs. Results showed that the experimental group was about four times better at providing correct answers to the comprehension and verification questions and required about one-fourth of the time to do so [4].

Function extraction technology can be applied to any programming language environment and has potential to impact many aspects of the software engineering life cycle. To better understand this impact, CERT conducted a comprehensive study to determine how FX could improve engineering operations in activities ranging from software specification and design to implementation and testing. This study produced the following recommendations for further FX development [5]:

- Prioritize development of FX automation for assembler language.
- Develop FX automation for correctness verification of software.
- Develop FX automation for high-level languages, starting with Java.
- Initiate research on FX automation for computing security attributes.

2006 Plans

FX/MC development will continue in 2006, and additional sponsors are welcome to participate in moving the technology forward. While the target of interest is malicious code, the system will in fact process any software written in assembly language. In addition, the technology developed for FX/MC can be applied to function extractor development for other languages such as Java and C, as well as to related capabilities such as automated correctness verification and component composition.

References

[1] Prowell, S.; Trammell, C.; Linger, R.; & Poore, J. *Cleanroom Software Engineering: Technology and Practice.* Reading, MA: Addison Wesley, 1999.

[2] Mills, H. & Linger, R. "Cleanroom Software Engineering." *Encyclopedia of Software Engineering, 2nd ed.* (J. Marciniak, ed.). New York: John Wiley & Sons, 2002.

[3] Pleszkoch, M. & Linger, R. "Improving Network System Security with Function Extraction Technology for Automated Calculation of Program Behavior." *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37).* Hawaii, January 5-8, 2004. Los Alamitos, CA: IEEE Computer Society Press, 2004.

[4] Collins, R.; Walton, G.; Hevner, A.; & Linger, R. *The CERT Function Extraction Experiment: Quantifying FX Impact on Software Comprehension and Verification* (CMU/SEI-2005-TN-047). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tn047.html.

[5] Hevner, A.; Linger, R.; Collins, R.; Pleszkoch, M.; Prowell, S.; & Walton, G. *The Impact of Function Extraction Technology on Next-Generation Software Engineering* (CMU/SEI-2005-TR-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tr015.html.

STAR*Lab

Computational Security Attributes

Engineering Automation for Security Assessment

Principal Investigator:
Gwendolyn H. Walton

Contact Point: Gwendolyn H. Walton
863-255-2932

Problem Addressed

Security strategies must be sufficiently dynamic to keep pace with organizational and technical change. However, in the current state of practice, security properties of software systems are assessed through error-prone, labor-intensive human evaluation. The result is imprecise and a priori assertions that can be of limited value in the dynamics of system operation in threat environments where security attributes can change quickly. This CERT STAR*Lab R&D study project for automated analysis of security takes a fundamentally different approach, focusing on the question "What can be computed with respect to security attributes?" to develop theory-based foundations for defining and computing attribute values with mathematical precision [1].

The ultimate goal of this work is to provide foundations to help transform security engineering into a theory-based computational discipline. Achieving this goal will require development of mathematical foundations and corresponding automation to permit both rigorous evaluation and improvement of the security attributes of software during development and real-time evaluation of security performance during operation.

Research Approach

The problem of determining the security properties of programs comes down in large measure to the question of how they behave when invoked with stimuli intended to cause harmful outcomes. Thus, the first step in security analysis is to understand program behavior at a level of completeness and correctness that is generally impractical with current technology. The emergence of the CERT STAR*Lab's new Function Extraction (FX) technology, unavailable to previous researchers, provides the basis for this critical first step by supporting the derivation of the functional behavior of programs as a starting point for the security analysis process. The foundations of FX treat programs as rules for mathematical functions or relations that can be computed from program logic. These foundations can be generalized to accommodate what are often termed "non-functional" properties, in this case security properties, but which in reality exhibit functional characteristics amenable to computational approaches.

A high-level description of automated evaluation of software security attributes consists of three major steps:

1. Specify security attributes in terms of their required functional behavior for the operational environment of the software being analyzed.
2. Apply FX technology to the software being analyzed to compute a behavior catalog that specifies its as-built functional behavior.
3. Perform computational analysis to verify that the extracted behavior of the software is correct with respect to the required security attribute behavior.

Expected Benefits

There are several advantages of this approach:

- A rigorous method is used to specify security attributes in terms of the actual behavior of code during execution and to verify that the automated processes are correct with respect to security attributes.
- The specified security behaviors provide requirements for a security architecture.
- Traceability capabilities can be defined and verified outside of the automated processes.
- Vulnerabilities can be well understood, making it easier to address evolution of code, environment, use, and users.
- The use of constraints provides a mechanism for explicitly defining all assumptions.

Computational security attribute technology can address specifying security attributes of software systems before they are built, specifying and evaluating the security attributes of acquired software, verification of the as-built security attributes of software systems, and real-time evaluation of security attributes during system operation.

2005 Accomplishments

While the mathematics of functions provides a solid point of departure for computational security analysis, much work remains to define mathematical approaches and engineering practices and to specify tools to support automated analysis of particular security properties. The SEI awarded funding to carry out this foundational work as an internal FY2006 R&D project.

2006 Plans

The CSA study will be completed and documented in an SEI technical report. Interested organizations are invited to provide sponsorship to expand the scope and application of this work.

References

[1] Linger, R.; Pleszkoch, M.; Walton, G.; & Hevner, A. *Flow-Service-Quality (FSQ) Engineering: Foundations for Network System Analysis and Development* (CMU/SEI-2002-TN-019). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu/publications/documents/02.reports/02tn019.html.

STAR*Lab

Automated Structuring for Understanding Legacy Systems

Transforming Complex Control Flow into Structured Form

Principal Investigator:
Stacy Prowell

Contact Point: Stacy Prowell
412-268-9205

Problem Addressed

The difficulty of maintaining and understanding legacy systems is often compounded by the unstructured nature of the code. However, a structured program can be correctly understood as a composition of parts, each of which may be understood on its own. Legacy systems, compiled and optimized code, and intentionally obfuscated code often lack such structure, making it extremely difficult to understand their true control flow and functionality. The complexity of unstructured code grows exponentially as the size of the program increases because it is possible for the control logic to jump from anywhere to anywhere. The understanding of what a portion of code does can be derailed by the discovery that another part of the system "jumps in" to the middle of the code just analyzed. Computed jumps and self-modifying code further compound the problem faced by analysts.

Research Approach

The structure theorem guarantees that any program can be automatically transformed into an equivalent structured form, and the constructive proof of the theorem defines an algorithmic approach for doing so. The resulting structured program is expressed in terms of a small number of control flow constructs (such as if-then-else, while-do, and sequences of instructions) which are composed in a disciplined manner [1]. The purpose of this project is to create a tool for generating a properly structured program from an arbitrarily structured input program for improved understanding and analysis.

Expected Benefits

Automated structuring of complex, spaghetti-logic code increases the speed and precision of human understanding for improved maintenance and evolution of software, as well as for analyzing security and other properties. It is particularly useful for understanding legacy code, where modifications and patches have obscured the original logic, and for assessing new but unfamiliar code.

2005 Accomplishments

CERT has developed the first version of a structuring engine for spaghetti-logic Intel assembly language executables as part of FX/MC system development. In this process, the input is disassembled, a de-obfuscation phase detects computed jumps and transforms them into case statements using automated program comprehension techniques, and the result is transformed into a structured program using only while-do, if-then-else, and sequences of instructions. This output program is function-equivalent to the input program, and is tagged with addresses for tracing to the original code.

2006 Plans

The assembly language structuring engine is embedded within the FX/MC system, and could be converted to a stand-alone system. In addition, the structuring algorithms can be applied to development of structuring engines for other languages such as Java and C. Sponsors are welcome to participate in this work.

References

[1] Prowell, S.; Trammell, C.; Linger, R.; & Poore, J. *Cleanroom Software Engineering: Technology and Process.* Reading, MA: Addison Wesley Longman, 1999.

STAR*Lab

Automated Correctness Verification for Reliable Software

Engineering Automation for Eliminating Errors and Vulnerabilities

Principal Investigator:
Mark Pleszkoch

Contact Point: Mark Pleszkoch
434-426-3754

Problem Addressed

Software containing errors and vulnerabilities cannot be trustworthy or secure. Yet most software is developed and delivered with incorrect and even unknown (and thus unspecified and untestable) behavior. In the current state of practice in software engineering, no practical means exists for automation support of large-scale correctness verification of software with respect to intended behavior. As a result, much time and energy is devoted to inspection and testing activities that can provide only limited evidence of correctness. Other engineering disciplines, for example, integrated circuit design, exhibit no reluctance to use substantial computational systems to verify the correctness of engineering artifacts. Achieving security and trustworthiness in software systems will require a similar level of computational support in their development.

Research Approach

The objective is to develop an operational prototype of a Function Verification (FV) system that will determine the correctness of programs expressed in a subset of the Java language. The system will employ the mathematics-based foundations of Function Extraction (FX) to achieve completeness and correctness of results, but the user will not be exposed to, or required to know, these foundations. The system will provide a proof of concept for functional verification technology in addressing the security and trustworthiness of software systems, and a foundation for elaboration into industrial-strength verification systems. In addition, the system will provide a standard, machine-processable form for representing intended behavior. Users will be able to code programs in the Java subset to satisfy intended behavior, and execute the FV system to check correctness.

Function Extraction and Function Verification are closely related. Functional correctness verification requires computing the as-built functional behaviors of program structures, just as in the Function Extraction process, and then comparing those behaviors to intended behaviors for equivalence or not. The function-theoretic model of software treats programs as rules for mathematical functions or relations, that is, mappings from domains (inputs, stimuli) to ranges (outputs, responses), no matter what subject matter they may deal with. While programs can contain an intractable number of execution paths, they are at the same time composed of a finite number of control structures, each of which implements a mathematical function or relation in the transformation of its inputs into outputs.

Table 1: Mapping of control structures into functional form

Structure	Program	Function Equation	Interpretation
Sequence	P: [f] do g; h enddo	f = [P] = [g;h] = [g] o [h]	For all possible arguments, does g followed by h do f?

Ifthenelse	P: [f]
	if p
	then
	g
	else
	h
	endif
		f = [P] = [if p then g else h endif] =
		([p] = true → [g] |
		[p] = false → [h])
			For all possible arguments,
			whenever p is true,
			does g do f,
			and whenever p is false,
			does h do f?

Whiledo	P: [f] while p do g enddo	f = [P] = [while p do g enddo] = [if p then g; while p do g enddo endif] = [if p then g; f endif] = f = ([p] = true → [f] o [g] | [p] = false → I)	For all possible arguments, is termination guaranteed, and whenever p is true, does g followed by f do f, and whenever p is false, does doing nothing do f?

A Correctness Theorem defines the mapping of these control structures into functional form for verification purposes, as shown in Table 1 [1, 2], where P represents the control structure, f represents the intended function, g and h represent subfunctions, p represents a predicate, square brackets represent the function of the enclosed program, "|" represents the "or" operator, "o" represents the composition operator, and "I" is the identity function. The table defines the function equations in terms of composition, case analysis, and for the whiledo, composition and case analysis in a recursive equation based on the equivalence of an iteration control structure and an ifthen control structure. These equations lie at the heart of the algorithms that will be implemented in the Function Verification prototype.

Expected Benefits

This project can provide substantial benefits to STAR*Lab customers and collaborators who must deal with software failures in enterprise operations. It is difficult to achieve trustworthiness and security goals for systems without knowing whether they are correct with respect to intended behavior. Routine availability of functional verification will substantially reduce errors, vulnerabilities, and malicious code in software. FV technology can replace much of the labor-intensive and error-prone work of program inspection and testing, with corresponding reductions in resource requirements and improvements in product quality.

2005 Accomplishments

Theoretical foundations developed for Function Extraction in 2005 are readily applicable to correctness verification [3].

2006 Plans

STAR*Lab is ready to initiate work on correctness verification and develop a proof-of-concept prototype for interested sponsors.

References

[1] Prowell, S.; Trammell, C.; Linger, R.; & Poore, J. *Cleanroom Software Engineering: Technology and Practice.* Reading, MA: Addison Wesley, 1999.

[2] Mills, H. & Linger, R. "Cleanroom Software Engineering." *Encyclopedia of Software Engineering, 2nd ed.* (J. Marciniak, ed.). New York: John Wiley & Sons, 2002.

[3] Hevner, A.; Linger, R.; Collins, R.; Pleszkoch, M.; Prowell, S.; & Walton, G. *The Impact of Function Extraction Technology on Next-Generation Software Engineering* (CMU/SEI-2005-TR-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tr015.html.

STAR*Lab

Automated Component Composition for Developing Reliable Systems

Engineering Automation for Combining Network Capabilities

Principal Investigator:
Gwendolyn H. Walton

Contact Point: Gwendolyn H. Walton
863-255-2932

Problem Addressed

Modern systems are characterized by large-scale heterogeneous networks with many components that must be correctly integrated to achieve mission objectives. It is often the case that the components are complex systems in their own right and must be dynamically integrated to provide end-to-end capabilities. System integration today is a complex, labor-intensive process that can take months or even years for large systems. Automation support for behavior analysis of component compositions could help reduce the time and effort required to achieve operational capabilities.

Research Approach

This project will define the extent to which component compositions can be automatically calculated. Automation support for determining composite behavior of components architected into systems could enable fast and reliable understanding and development. Such a capability is crucial for achieving confidence in distributed, component-based architectures and for creating just-in-time systems of systems. Composition computation must generate mathematically correct abstractions of behavior at any level and help scale up the reliable unit of construction for systems [1]. Because behavior calculation is essentially a compositional task, Function Extraction is the key underlying technology for component composition. Abstracted behavior of programs and components can be organized into behavior catalogs: repositories of as-built program behavior expressed in sets of conditional concurrent assignments and indexed according to the predicate expressions involved.

Expected Benefits

Automated derivation of the net effect of compositions can reveal combined functionality, illuminate mismatches, facilitate analysis of design alternatives, and support evaluation of COTS products. This approach can also guide rapid and reliable refactoring of components and systems in responding to new system requirements.

2005 Accomplishments

Research and development in the FX/MC project carried out in 2005 has direct applicability to automated composition of components.

2006 Plans

Creating Function Extractors to compose software components would provide automation support for construction and integration of entire systems. Investigations in Function Extraction technology on the FX/MC project have provided initial foundations for component composition. Additional mathematical work in unification and reduction is in progress. A key step toward creation of an automated composition capability would be to develop a proof-of-concept prototype to demonstrate the application of FX technology in automating the composition process. Sponsors are welcome to join in this effort.

References

[1] Pleszkoch, M. & Linger, R. "Improving Network System Security with Function Extraction Technology for Automated Calculation of Program Behavior." *Proceedings of the 37th Hawaii International Conference on System Sciences (HICSS-37).* Hawaii, January 5-8, 2004. Los Alamitos, CA: IEEE Computer Society Press, 2004.

STAR*Lab

Flow-Service-Quality (FSQ) Engineering

Foundations for High-Assurance Network-Centric System Development

Principal Investigator:
Stacy Prowell

Contact Point: Stacy Prowell
412-268-9205

Problem Addressed

Modern society is dependent on large-scale network-centric systems whose complexity can often exceed engineering capabilities for intellectual control. The result can be frustrations and delays in development and failures and compromises in operation. Intellectual control does not mean the absence of uncertainties or failures—they are inevitable—but rather the capability to address them in a rigorous engineering framework.

System complexity and survivability are closely related. Complexity diminishes survivability by masking errors and vulnerabilities and hiding unforeseen paths for intrusion. The survivability of complex systems that support national infrastructures is of particular concern. The problem lies not with developers but with the lack of engineering methods to cope with system complexities. More effective engineering technology is required across the life cycle for fast and precise development and evolution of network-centric systems.

A promising path lies in the investigation of unifying mathematical foundations as a basis for engineering practices and automation support. These foundations must explicitly accommodate the realities of large-scale network systems: highly distributed heterogeneous components, shifting boundaries and users, uncertain COTS function and quality, extensive asynchronous operations, unpredictable failures and compromises, and lack of visibility and control. They must also address enterprise needs for rapid development and evolution, predictable composition of components, and system interoperability to achieve mission goals. The objective of Flow-Service-Quality (FSQ) engineering is to develop unified engineering methods for network-centric system analysis, specification, design, verification, implementation, and operation. The focus of FSQ is on developing high-assurance systems, with special emphasis on complexity reduction and survivability improvement.

Research Approach

Initial research has identified three integrated engineering concepts that address the realities of network-centric systems:

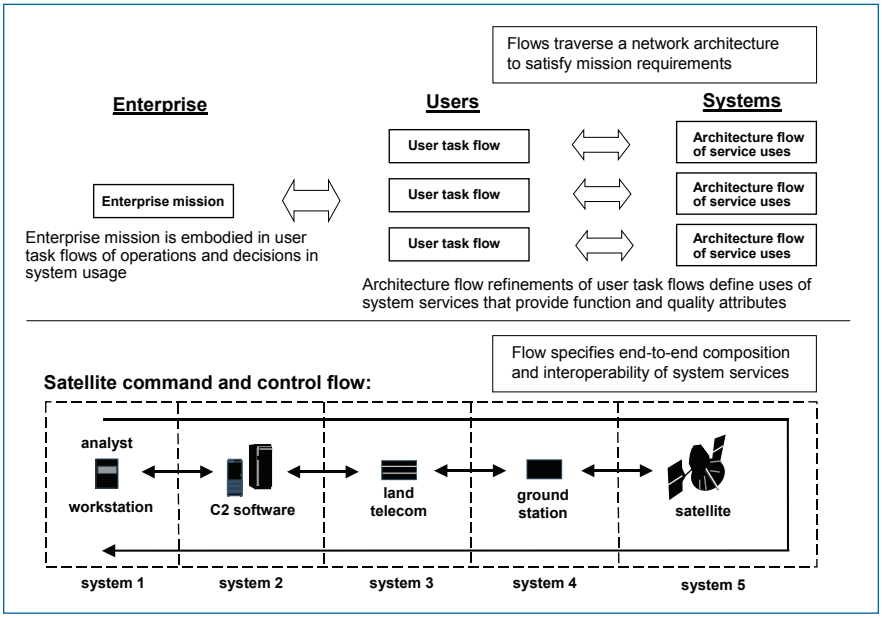
- Flow Structures: User task flows and their refinements into system service uses can provide engineering foundations for analysis, specification, design, verification, and implementation of system functionality and quality attributes.
- Computational Quality Attributes: Quality attributes can be associated with both flows and the system services they invoke and computed as dynamic functional properties, rather than treated as static, a priori estimates of limited utility in real-time system operations.
- Flow Management Architectures: Flow Structures and Computational Quality Attributes support architecture frameworks that manage flows, network services, and quality attributes in execution.

Flow Structures. Flow Structures are compositions of system services distributed across networks that combine to carry out user tasks to accomplish enterprise missions. They employ mathematical semantics that permit them to be deterministic for human understanding and analysis, despite the underlying asynchronism of network behavior. Flow Structure engineering requires designing for unpredictable events that can impact mission survivability. In addition, Flow Structures provide a vehicle for specification and management of quality attributes such as security and reliability. Thus, the first-class concepts of flow, service, and quality are the essential and primary artifacts of Flow-Service-Quality engineering [1, 2, 3].

Network-centric systems are usefully viewed as webs of asynchronously communicating components that provide services whose functions can be combined in various patterns to satisfy enterprise mission requirements. System services include all the functional capabilities of a system, from protocols, operating systems, and middleware, to databases and applications. The sequencing of system services in user task flows can be refined into compositions of network hardware, software, and human components that provide the services. These compositions are end-to-end traces that define slices of network architectures whose net effect is to carry out operations that satisfy user requirements.

The top of the figure below depicts notional refinement of user task flows into uses of system architecture components. Flows are essentially procedures that define compositions of network service uses at levels of abstraction ranging from an enterprise mission down to its system implementation. Flows can specify integration and traversal of many systems and components, as shown in the bottom of the figure. Flows can be expressed in simple control structures including sequence, alternation, and iteration, and can be refined, abstracted, and verified with precision [4]. Flows invoke services, which can be refined into flows, etc., in a recursive process that employs identical methods at all levels of design. The functional specification of a network system is envisioned as a set of Flow Structures, where the union of the flows defines a necessary network architecture, and the functional specification of each service in the network is based on the union of all its uses in flows where it appears.

Figure 1: FSQ Engineering



Flow Structures can engage in extensive traversals of network nodes and services whose behavior and quality attributes cannot always be known. Services may be unreliable, compromised, or simply unavailable. These uncertainty factors are pervasive behavioral realities of large-scale, network-centric systems. Flow Structure engineering requires designers to define appropriate actions by flows for uncertainty factors they may encounter, thereby addressing system survivability and risk management issues.

Computational Quality Attributes. FSQ engineering treats quality attributes as ever-changing functions that must be dynamically computed, rather than as static, a priori descriptions of limited utility in system operation. Attributes must be measurable in defined metrics as computable functions. While such functions rely on what can be computed and may differ thereby from traditional methods, they permit new approaches to attribute analysis and evaluation. Attribute requirements can be associated with system component uses embedded within Flow Structures and dynamically compared with computed attribute capabilities in operation. Future work will explore attribute-specific models within this framework. The Computational Security Attributes project discussed above will extend and amplify this initial work.

Flow Management Architectures. Flow Structures and Computational Quality Attributes support system architectures that carry out dynamic flow and attribute management in execution. Flow Management Architectures (FMA) can provide design and implementation frameworks for this purpose, as well as engineering processes for architecture development. An open family of such frameworks can be defined for architecture development both in the small and in the large. Future work will investigate FMA templates of system topologies and functional capabilities for managing flows and their quality attributes.

The mathematical semantics of FSQ are defined to support development and verification of flow structures for the uncertain environments of large-scale network systems as a standard engineering practice. For example, flow specification requires definition of appropriate actions by a flow for all possible responses of key services, both desired and undesired. Thus, if the behavior of an invoked service changes for any reason, the specification and verification of the invoking flow need not change. This approach accommodates the realities of today's network systems and offers important advantages. It requires for mission survivability that the uncertainty factors be dealt with explicitly in specification, design, and dynamic execution, thereby addressing important aspects of enterprise risk management. It permits flows and reasoning about them to be localized yet complete. And it permits flow structures to be defined by simple deterministic structures despite the underlying asynchronous behavior of their constituent services. These deterministic structures can be refined, abstracted, and verified using straightforward compositional methods for human understanding and analysis.

Expected Benefits

FSQ foundations prescribe engineering practices and tools for intellectual control and survivability engineering in network-centric system analysis and development. In particular, the deterministic nature of flow structures facilitates human understanding despite the underlying asynchronous behavior of network systems. Computational Quality Attributes permit automated reactions to dynamically changing quality values in system execution. And Flow Management Architectures provide systematic frameworks for managing flows and quality attributes in operation.

2005 Accomplishments

Research was carried out to relate FSQ engineering to web service and service-oriented architectures.

2006 Plans

STAR*Lab is interested in continued development and application of FSQ engineering methods, particularly for large-scale network systems. A partnership of interested organizations can be established to create a proof-of-concept prototype tool and associated engineering practices, with immediate application to network system development. Additional sponsorship of this project is welcome.

References

[1] Hevner, A.; Linger, R.; Sobel, A.; & Walton, G. "The Flow-Service-Quality Framework: Unified Engineering for Large-Scale, Adaptive Systems." *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS35).* Hawaii, January 7-10, 2002. Los Alamitos, CA: IEEE Computer Society Press, 2002.

[2] Linger, R.; Pleszkoch, M.; Walton, G.; & Hevner, A. *Flow-Service-Quality (FSQ) Engineering: Foundations for Network System Analysis and Development* (CMU/SEI-2002-TN-019). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu/publications/documents/02.reports/02tn019.html.

[3] Hevner, A.; Linger, R.; Pleszkoch, M.; & Walton, G. "Flow-Service-Quality (FSQ) Engineering for the Specification of Complex Systems." *Practical Foundations of Business System Specifications* (H. Kilov & K. Baclawski, eds.). Dordrecht, NL: Klewer Academic Publishers, 2003.

[4] Prowell, S.; Trammell, C.; Linger, R.; & Poore, J. *Cleanroom Software Engineering: Technology and Process.* Reading, MA: Addison Wesley Longman, 1999.

Threat Dynamics

Modeling and Analyzing the Dynamics of Organizational Threats

Principal Investigator:
Andrew P. Moore

Contact Point: Andrew P. Moore
412-268-5465

Problem Addressed

Information technology decision-makers face daunting challenges to provide and maintain inter-networked systems that ensure organizational mission success despite sophisticated computer network attacks. Exacerbating this situation, the extremely dynamic threat environment for Internet-based systems requires regular re-evaluation of organizational operations and systems in light of changes in attacker activity or, simply, an improved understanding of threats. Unfortunately, current technology provides little help in determining how attacks affect the survival of what is important to an organization and maintaining a survivability strategy as the threat environment evolves. This project develops methods and tools that help model and analyze an organization's threat dynamics and that improve the organization's security, survivability, and resiliency in light of those dynamics. We define *threat dynamics* as the study of the impact of an organization's threat environment on the ability of the organization to achieve its mission objectives.

Our focus in 2005 has been to use the threat dynamics framework to study the insider threat. Evidence from a comprehensive study of insider threats indicates that managers, at times, make decisions that are intended to enhance organizational performance and productivity but that have the unintended consequence of magnifying the likelihood of insider cyber attack and the organization's exposure to it [2,3]. In addition, the ultimate effect of business policy decisions on insider threat risks over time are often complex and sometimes counterintuitive, with short-term effects very different from long-term effects. The potential cascading effects and long-term consequences of personnel, policy, and technology decisions on the organizational culture and security posture are not always immediately evident.

Individuals across CERT are involved with this project, including Dawn Cappelli, Timothy Shimeall, and Bradford Willke.

Technical Approach

Methods to deal with the insider threat problem need to capture and analyze the complex interactions between behavioural, technical, policy, and cultural issues over time, so that an integrated risk management approach can be developed. We believe that threat dynamics provides a foundation for developing methods and tools that will help decision-makers understand, characterize, and communicate the potential risk due to insider attacks on organizational and system operations and their respective missions. Simulation of the threat dynamics model enables analysis of alternative strategic responses to counter insider threats so that strong and justifiable defenses can be identified and mounted.

This effort uses the threat dynamics modeling approach and commercial tools to

- develop empirically validated models of the insider threat problem based on insider threat data collaboratively collected by the U.S. Secret Service and CERT
- support simulation of the models with the purpose of enabling organizations to identify alternative combinations of insider threat countermeasures and analyze and compare their effectiveness
- use these same tools to develop a training or decision analysis environment (which we call the management simulator) based on the threat dynamics models

The management simulator will provide a protected, interactive environment for hands-on analysis of the effects of policy and technical decisions and countermeasures on malicious insider activity. The tool will provide an effective means to communicate insider threat risks and tradeoffs and will be useful for both technical and non-technical personnel, from system administrators to corporate CEOs.

Expected Benefits

The ultimate effect of business, policy, and technical decisions on insider threat risks is complex and often counterintuitive and can result in significant losses and operational impacts due to insider cyber attack. This work will develop, demonstrate, and validate technology that will help decision-makers better understand insider threat risks and the effects of decisions on the promotion or mitigation of those risks. The technology will empower organizations to develop comprehensive, efficient, and justifiable defenses against insider threats along with the organizational understanding and support needed to maintain a strong security posture over time. Broad application of tools developed will enable organizations across the United States and abroad to significantly reduce their risk and losses due to insider attacks. The capability that will result from this work promotes the security, survivability, and resiliency of all government, military, and commercial critical

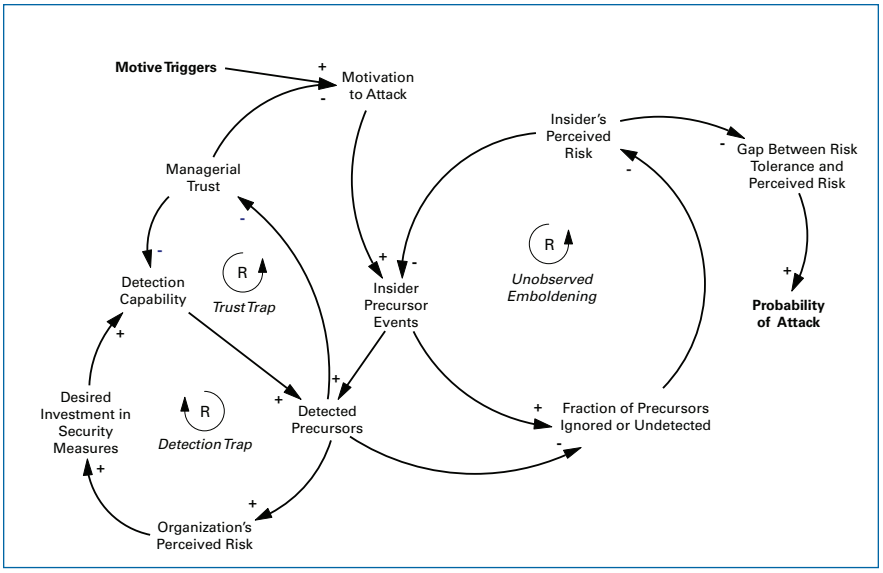


Figure 1: Insider threat dynamics

systems. The ultimate beneficiaries will be organizational stakeholders and, where the U.S. critical infrastructures are better protected, the general public as a whole.

2005 Accomplishments

Members of the Security Dynamics Network participated in two workshops hosted by CERT and CyLab in February 2004 and November 2004 [1, 4]. We developed a preliminary system dynamics model based on six insider threat cases in the public domain. These cases ranged broadly in terms of technical sophistication and the motivation of the attacker. Motivations ranged from greed to revenge, with a mix of motivations exhibited in most cases.

The first workshop resulted in a preliminary system dynamics model and the expression of the dynamic trigger hypothesis—that there are dynamic mechanisms operating within organizations that spur insider attacks to occur over time. The dynamic trigger hypothesis has three components, as described in text and depicted in Figure 1:

- Detection Trap (R1): Gaps in detection capability suppress detection of ongoing violations and precursor events. The lack of detection can be misinterpreted as an absence of attack threats, thereby suppressing desired investments in security measures (such as detection capability).
- Trust Trap (R2): Over time, excessive management trust can erode an organization's compromise detection capability, leading to fewer detected precursor

events and increased violations of best practices. In turn, fewer detected events can reinforce the (perhaps erroneous) conclusion that compromise detection is not needed.

- Unobserved Emboldening (R3): Left undetected, precursor events reduce an insider's perception of risk. In turn, reduced perceptions of risk lead to additional precursor events. This reinforcing cycle of emboldening can remain unobserved by management (absent detection of precursor events—see Detection Trap and Trust Trap).

The second workshop and subsequent work refined the preliminary model based on a fictional organization case description. The case description abstracted long-term fraud cases investigated in the Insider Threat Study into a compelling and concrete basis for the modeling effort.

2006 Plans

We plan to extend and refine the system dynamics model of insider threat to apply to insider sabotage based on findings from the Insider Threat Study [2]. We plan to use the model as a basis for gaining further insights into practical tradeoffs of insider threat risks, mitigations, and organizational performance. Finally, we expect to develop an interactive learning environment for transitioning lessons about insider threat to individuals in both government and industry.

References

[1] Rich, E.; Martinez-Moyano, I. J.; Conrad, S.; Cappelli, D. M.; Moore, A. P.; Shimeall, T. J.; Andersen, D. F.; Gonzalez, J. J.; Ellison, R. J.; Lipson, H. F.; Mundie, D. A.; Sarriegui, J. M.; Sawicka, A.; Stewart, T. R.; Torres, J. M.; Weaver, E. A.; & Wiik, J. "Simulating Insider Cyber-Threat Risks: A Model-Based Case and a Case-Based Model," *Proceedings of the 23rd International Conference of the System Dynamics Society.* Boston, MA, July 17-21, 2005. http://www.systemdynamics.org/conf2005/proceed/index.htm.

[2] Keeney, M. M.; Kowalski, E. F.; Cappelli, D. M.; Moore, A. P.; Shimeall, T. J.; & Rogers, S. N. "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors." Joint SEI and U.S. Secret Service Report, May 2005. http://www.cert.org/archive/pdf/insidercross051105.pdf.

[3] Cappelli, D. M. & Moore, A. P. "Analyzing Organizational Cyber Threat Dynamics." *Proceedings of the Workshop on System Dynamics of Physical and Social Systems for National Security.* Chantilly, VA, April 21-22, 2005. Contact Andrew Moore at 412-268-5465 for a copy of the paper.

[4] Anderson, D. F.; Cappelli, D. M.; Gonzalez, J. J.; Mojtahedzadeh, M.; Moore, A. P.; Rich, E.; Sarriegui, J. M.; Shimeall, T. J.; Stanton, J. M.; Weaver, E.; & Zagonel, A. "Preliminary System Dynamics Maps of the Insider Cyber-Threat Problem." *Proceedings of the 22nd International Conference of the System Dynamics Society.* Oxford, England, July 25-29, 2004. http://www.cert.org/archive/pdf/InsiderThreatSystemDynamics.pdf.

Additional Research Activities

Contact Point:
SEI Customer Relations
412-268-5800

Correlations Across Ports in Network Flow Records

Principal Investigator: Joshua McNutt

Analysis of network traffic to identify malicious or anomalous activity is challenging because traffic volume metrics are not well behaved (do not follow a "normal" or Poisson statistical distribution) [1, 2]. Therefore, methods of efficiently identifying and removing known background activity (vertical scanning, ephemeral port activity, etc.) are needed to concentrate analysts on the remaining traffic of potential interest.

This project has focused on methods for detecting the onset of anomalous port-specific activity by recognizing deviations from correlated activity. We have discovered that background activity in the form of vertical scanning often leads to temporal correlations between different ports. Correlation coefficients between time series on different ports, for the number of flows per hour on each port, are frequently >0.99.

Because of this relationship between activity on different ports, the median time series of a group of related ports can be used to filter out the (highly variable) background noise prior to identifying traffic anomalies unique to a specific port. This background-subtracted time series enhances the ability to detect the onset of scanning for new vulnerabilities and other traffic of interest.

Since correlation is a measure that is highly sensitive to outliers, we exclude extreme observations prior to calculating correlations. To this end, we calculate a robust correlation measure using a minimum volume ellipsoid approach [3] and cluster ports that share similar patterns of activity. Due to the highly variable nature of the background traffic, this method is likely to result in a lower false positive rate than we would expect in a paradigm in which all ports are evaluated individually.

References

[1] Paxson, V. "Wide-Area Traffic: The Failure of Poisson Modeling." *IEEE/ACM Transactions on Networking 3,* 3 (1995): 226-244.

[2] Leland, W. E.; Taqqu, Murad S.; Willinger, Walter; & Wilson, Daniel V. "On the Self-Similar Nature of Ethernet Traffic," 183-193. *ACM SIGCOMM Computer Communication Review Vol. 23 Issue 4, Conference Proceedings on Communications Architectures, Protocols and Applications, SIGCOMM '93.* San Francisco, October 1993. New York, NY: ACM Press, 1993.

[3] Rousseeuw, P. J. "Umasking Multivariate Outliers and Leverage Points." *Journal of the American Statistical Association 85* (1990): 633-639/648-651.

Additional Research Activities

Vendor Risk Assessment and Threat Evaluation (V-RATE) for Improved Security of COTS-Based Systems

Principal Investigator: Howard Lipson

Using commercial-off-the-shelf (COTS) components to build large, complex systems has become the standard way that systems are designed and implemented by government and industry. These components are being integrated into mission-critical systems where successful cyber attacks (or other causes of system failure) can lead to severe consequences. Yet with little access to the components' source code or development processes, it is difficult to evaluate the security and survivability attributes of these components or their contribution to the attributes of the composite system. The V-RATE project is developing methods for assessing vendor capabilities as a strong indicator of product quality.

Based on a taxonomy covering vendor-related risks and the acquiring organization's risk management skills, the V-RATE method provides criteria to help decide when and how COTS products can be used to build survivable systems and to assess and mitigate the risks of COTS usage. Factors that influence this decision include not only attributes of the COTS products themselves but also attributes of the system's mission, the vendor, the vendor's development life cycle processes, and the acquiring organization's risk management skills in dealing with vendors. The output of an assessment based on the V-RATE taxonomy is a *vendor-risk profile* for the system being evaluated. We envision a large and growing collection of vendor-risk profiles tied to real-world performance histories, providing empirical data against which a newly generated risk profile can be compared. A vendor-risk profile can be used to assess the risk associated with the use of a product in a particular threat environment and to identify areas for additional risk-mitigation activities. Because a single numerical rating would not provide sufficient guidance for these risk mitigation activities, the vendor-risk profile helps the acquiring organization to identify its risks in each of the V-RATE taxonomy areas and to consider its risk tolerance with respect to each element of the taxonomy. The V-RATE project team is seeking opportunities for piloting the method to help organizations attain assurance and reduce risk in creating critical systems with substantial COTS involvement.

Contact Point:
SEI Customer Relations
412-268-5800

Cyber Security of the Electric Power Grid for Improved Infrastructure Survivability

Principal Investigator: Howard Lipson

Howard Lipson of the CERT Survivable Systems Engineering team continued to serve as a source of security and survivability expertise to faculty and researchers at Carnegie Mellon's Electricity Industry Center (CEIC), as he has since its inception. In particular, Lipson continued as co-principal investigator on a National Science Foundation sponsored collaborative research project titled "Secure and Robust IT Architectures to Improve the Survivability of the Power Grid." This project is a joint effort of the CEIC and the Washington State University School of Electrical Engineering and Computer Science. Work on this project is helping to expand the role of CERT beyond traditional computer platforms into the protection of critical infrastructures.

Selected List of Publications

Book Chapters

Mead, Nancy R.; Davis, Noopur; Dougherty, Chad; Mead, Robert; & Seacord, Robert. "Recommended Practices," 275-308. *Secure Coding in C and C++*. Upper Saddle River, NJ: Addison Wesley, 2005.

Prowell, S. & Poore, J. "Reliability Computation for Usage-Based Testing." *Modern Statistical and Mathematical Methods in Reliability. Vol. 10, Series on Quality, Reliability and Engineering Statistics.* Edited by A. Wilson, N. Limnios, S. Keller-McNulty, and Y. Armijo. Hackensack, NJ: World Scientific Publishing, 2005 (ISBN: 981-256-356-3).

Reports

Collins, R.; Walton, G.; Hevner, A.; & Linger, R. *The CERT Function Extraction Experiment: Quantifying FX Impact on Software Comprehension and Verification* (CMU/SEI-2005-TN-047). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tn047.html.

Mead, N. R.; Hough, E.; & Stehney, T. *Security Quality Requirements Engineering (SQUARE) Methodology* (CMU/SEI-2005-TR-009). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tr009.html.

Ellison, Robert J. *Trustworthy Composition: Challenges for the Practitioner* (CMU/SEI-2005-TN-026). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tn026.html.

Gordon, D.; Stehney, T.; Wattas, N.; & Yu, E. *Quality Requirements Engineering (SQUARE): Case Study on Asset Management System, Phase II* (CMU/SEI-2005-SR-005). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05sr005.html.

Hevner, A.; Linger, R.; Collins, R.; Pleszkoch, M.; Prowell, S.; & Walton, G. *The Impact of Function Extraction Technology on Next-Generation Software Engineering* (CMU/SEI-2005-TR-015). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tr015.html.

Randazzo, M. R.; Keeney, M. M.; Kowalski, E. F.; Cappelli, D. M.; & Moore, A. P. "Insider Threat Study: Illicit Cyber Activity in the Banking and Finance Sector." Joint SEI and U.S. Secret Service Report, August 2004. http://www.cert.org/archive/pdf/bankfin040820.pdf.

Selected List of Publications

Papers

Lipson, Howard & van Wyk, Ken. *Integrating Business Applications into Your IT Security Infrastructure—Creating and Maintaining a Secure Application Environment: Topic 1: Application Firewalls and Proxies – Introduction and Concept of Operations.* https://buildsecurityin.us-cert.gov/ (2005).

Rich, Eliot; Martinez-Moyano, Ignacio; Conrad, Stephen; Cappelli, Dawn; Moore, Andrew; Shimeall, Timothy; Andersen, David; Gonzalez, Jose; Ellison, Robert; Lipson, Howard; Mundie, David; Sarriegi, Jose Mari; Sawicka, Agata; Stewart, Thomas; Torres, José Manuel; Wiik Johannes; & Weaver, Elise. "Simulating Insider Cyber-Threat Risks: A Model-Based Case and a Case-Based Model." *Proceedings of the 23rd International Conference of the System Dynamics Society (ISDC-2005).* Boston, MA, July 17-21, 2005. http://www.systemdynamics.org/publications.htm.

Fung, Casey K.; Hung, Patrick C. K.; Wang, Guijun; Linger, Richard C.; & Walton, Gwendolyn H. "A Study of Service Composition with QoS Management," 724. 2005 *IEEE International Conference on Web Services (ICWS'05).* Orlando, Florida, July 11-15, 2005. Los Alamitos, CA: IEEE Computer Society Press, 2005.

Gates, C. & Becknel, D. "Host Anomalies from Network Data," 325-332. *Proceedings from the Sixth Annual IEEE Systems, Man and Cybernetics Information Assurance Workshop.* West Point, New York, June 15-17, 2005. Los Alamitos, CA: IEEE Computer Society Press, 2005.

Moore, A. P. & Cappelli, D. M. "Analyzing Organizational Cyber Threat Dynamics." *Proceedings of the Workshop on System Dynamics of Physical and Social Systems for National Security.* April 21-22, 2005.

Cappelli, D. M.; Moore, A. P.; & Shimeall, T. J. "Common Sense Guide to Prevention and Detection of Insider Threats." *Proceedings of the 2005 CyLab Corporate Partners Conference.* Pittsburgh, PA, April 13-15, 2005.

Fung, Casey K.; Hung, Patrick C. K.; Linger, Richard C.; & Walton, Gwendolyn H. "Extending Business Process Execution Language for Web Services with Service Level Agreements Expressed in Computational Quality Attributes," 166a. *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS-38).* Big Island, Hawaii, January 3-6, 2005. Los Alamitos, CA: IEEE Computer Society Press, 2005.

Selected List of Publications

Gates, C.; Collins, M.; Duggan, M.; Kompanek, A.; & Thomas, M. "More NetFlow Tools: For Performance and Security," 121-132. *Proceedings of the 18th Large Installation Systems Administration Conference (LISA 2004).* Atlanta, Georgia, November 14-19, 2004. http://www.usenix.org/events/lisa04/tech/gates.html.

McHugh, J.; Gates, C.; & Becknel, D. "Situational Awareness and Network Traffic Analysis," 209-228. *Proceedings of the Gdansk NATO Workshop on Cyberspace Security and Defence: Research Issues.* Gdansk, Poland, September 6-9, 2004.

Anderson, D. F.; Cappelli, D. M.; Gonzalez, J. J.; Mojtahedzadeh, M.; Moore, A. P.; Rich, E.; Sarriegui, J. M.; Shimeall, T. J.; Stanton, J. M.; Weaver, E.; & Zagonel, A. "Preliminary System Dynamics Maps of the Insider Cyber-Threat Problem." *Proceedings of the 22nd International Conference of the System Dynamics Society.* Oxford, England, July 25-29, 2004. http://www.cert.org/archive/pdf/InsiderThreatSystemDynamics.pdf.

Linger, Richard C.; Hevner, Alan R.; Walton, Gwendolyn H.; & Pleszkoch, Mark G. "Flow-Service-Quality (FSQ) Engineering: Foundations for High-Assurance Network Systems Development," 255-256 (extended abstract). *Eighth IEEE International Symposium on High Assurance Systems Engineering (HASE 2004).* Tampa, Florida, March 25-26, 2004. Los Alamitos, CA: IEEE Computer Society Press, 2004.

Journal Articles

Mead, Nancy R.; Weiringa, Roel; Maiden, Neil; & Rolland, Colette. "Requirements Engineering Paper Classification and Evaluation Criteria: a Proposal and a Discussion." *Requirements Engineering Journal 11* (2006): 102-107.

Mead, Nancy R. & McGraw, Gary. "A Portal for Software Security." *IEEE Security & Privacy 2,* 4 (July-August 2005): 75-79.

Mead, Nancy R. "Outsourcing and Information Security: What are the Risks?" *Cutter IT Journal 17,* 10 (October 2004): 30-35.

Talks/Panels/Workshops

Gates, Carrie, Presenter. "Security Visualization: A Case Study." Annual Computer Security Applications Conference, Case Studies Track. Tucson, Arizona, December 7, 2005.

Mead, Nancy R.; Wieringa, Roel; Maiden, Neil; & Rolland, Colette. "A Classification of RE Papers: Are We Researching or Designing RE Techniques?" Third International Workshop on Comparative Evaluation in Requirements Engineering (CERE'05). Paris, France, August 29, 2005.

Lipson, H. Invited Participant, Assurance Cases for Security Workshop (follow-on workshop of the 2004 International Symposium on Dependable Systems and Networks). Arlington, Virginia, June 13-15, 2005.

Mead, Nancy R. & Stehney, Ted. "Security Quality Requirements Engineering (SQUARE) Methodology." Software Engineering for Secure Systems (SESS05). May 15-16, 2005, ICSE 2005, St. Louis, MO. http://homes.dico.unimi.it/%7Emonga/sess05.html.

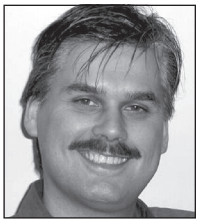
Mead, Nancy R., Panelist. "Software Assurance Education." IEEE Conference on Software Engineering Education & Training (CSEET'05). Ottawa, Ontario, Canada, April 18-20, 2005.

System Dynamics Modeling for Information Security: An Invitational Group Modeling Workshop. Pittsburgh, PA, February 16-20, 2004. http://www.cert.org/research/sdmis/.

Technical Leadership

Carrie Gates	• Program Committee Member, New Security Paradigms Workshop, Lake Arrowhead, California, September 2005 • Program Committee Member, Symposium On Usable Privacy and Security, Pittsburgh, Pennsylvania, July 2005
Howard F. Lipson	• Team leader and author, Assembly, Integration & Evolution content area of the DHS Build Security In Web site (https://buildsecurityin.us-cert.gov/) • Invited Participant, Assurance Cases for Security Workshop, Arlington, Virginia, June 13-15, 2005 • Member (founding), Carnegie Mellon Electricity Industry Center • Reviewer, International Symposium on Dependable Systems and Networks (DSN) • Reviewer, *IEEE Transactions on Dependable and Secure Computing* (TDSC)
Nancy Mead	• Named IEEE Fellow for leadership in software engineering education and development and application of software engineering methods in requirements engineering and survivable systems • Program Committee Member, 2005, IEEE Workshop Requirements for High Assurance Systems • Program Committee Member, 2005, IEEE Workshop on Requirements Engineering Education & Training • Practitioner Track Co-Chair, 2005, IEEE International Requirements Engineering Conference • Program Committee, 2005, Australian Workshop on Requirements Engineering • Editorial Board Member, 2003–present, *IEEE Security & Privacy* • Editorial Board Member, 2002–present, *Requirements Engineering Journal* • Steering Committee Member, IEEE International Requirements Engineering Conference, 1998–2005 • Steering Committee Member, IEEE Conference on Software Engineering Education & Training, 1995–2005
Stacy Prowell	• Program Committee Member, 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06), March 2006 • Member, IEEE CS Technical Committee on Software Engineering (TCSE)
Gwendolyn H. Walton	• Co-chair (with Alan Hevner and Richard Linger), conference track on Testing and Certification of Trustworthy Systems, 38th Hawaii International Conference on System Sciences (HICSS), January 2005 • Co-chair, HICSS conference minitracks, 2003–present
Richard Linger	• Co-chair (with Alan Hevner and Gwendolyn H. Walton), conference track on Testing and Certification of Trustworthy Systems, 38th Hawaii International Conference on System Sciences (HICSS), January 2005 • Program Committee Member, Hawaii International Conference on System Sciences (HICSS), 2002–present

Biographies



Sven Dietrich

Sven Dietrich is a member of the technical staff in the Networked Systems Survivability (NSS) Program at the SEI.

As a member of the CERT Research team, his work includes conducting research on computer security and survivable network technology. Dietrich also actively participates in the CyLab (previously Center for Computer and Communications Security—C3S), a collaboration between the SEI NSS program and several departments at Carnegie Mellon University (CMU), including Electrical and Computer Engineering and Computer Science.

Prior to joining the SEI, Dietrich was a senior security architect at the NASA Goddard Space Flight Center. His work included intrusion detection, distributed denial of service analysis, and the security of Internet Protocol communications in space. For his contributions to the latter he was granted the NASA Goddard Space Flight Center National Resource Group Achievement Award in 2000. Previously he had served on the faculty at Adelphi University for six years, where he taught mathematics and computer science. His research interests include computer security, cryptographic protocols, and quantum cryptography, and he gives presentations and talks on these subjects.

Dietrich holds a Doctor of Arts in Mathematics, an MS in Mathematics, and a BS in Computer Science and Mathematics from Adelphi University in Garden City, New York. He belongs to the Association for Computing Machinery (ACM), the Institute for Electrical and Electronics Engineers (IEEE) Technical Committee for Security and Privacy, and the National Mathematics Honor Society Pi Mu Epsilon.

Robert J. Ellison

Robert J. Ellison is a senior member of the technical staff in the NSS Program at the SEI. Ellison was part of the Carnegie Mellon University team that wrote the proposal for the SEI and joined the new FFRDC in 1985 as a founding member. While at the SEI he has served in both technical and management roles.

Before coming to CMU, Ellison taught mathematics at Brown University, Williams College, and Hamilton College. At Hamilton College, he directed the creation of the Computer Science curriculum. He joined the Carnegie Mellon Computer Science Department in 1981, where his research supported the Gandalf project, a prototype software development environment.

While at the SEI Ellison has worked in a number of technical areas. He was a project leader for evaluating software engineering development environments and associated software development tools. He was a member of the group that created the Quality Attribute Workshop, which is an elicitation technique for quality attribute requirements. He regularly participates in the evaluation of software architectures and contributes from the perspective of security and reliability measures. As a member of the NSS Program, he contributed to the development of the Survivable Systems Analysis Method and has been a team member for all applica-

tions of that approach. The objective of his current work, which is motivated by his experience with architecture assessments, is to better integrate security issues into the overall architecture design process.

Ellison received his MS and PhD in mathematics from Purdue University and a BA in mathematics from Lewis and Clark College in Portland, Oregon. He is a member of the IEEE Computer Society and the ACM.

Carrie Gates

Carrie Gates is a Member of the Technical Staff in the Networked Systems Survivability (NSS) Program at the SEI.

As a member of the CERT Network Situational Awareness Group, Gates performs research on the detection of malicious activity given highly aggregated network data for ISP-level networks. She has developed a scan detection algorithm that uses very limited network information to detect scans, an algorithm that has been deployed on a large network and is in operational use. Her current interests in the area of scan detection are focused on the detection of stealthy scans, such as those that scan only hosts that are known to exist or that scan particularly slowly. Her other interests focus on how the connection information for a network can be used to recognize that an internal host has been compromised, based on observing changes in the communication patterns it exhibits.

Prior to joining CERT, Gates was a Systems Manager for the Faculty of Computer Science at Dalhousie University, Canada. In this role she was responsible for the security of the computer systems and has a practical background in network and systems security, computer forensics, and intrusion detection, as well as experience in working with law enforcement officials.

Gates is expecting to receive her PhD from Dalhousie University in May 2006. She has an MS in Computing Science, specializing in Neural Networks, and a BS in Computing Science, both from Dalhousie University. She belongs to the Association for Computing Machinery (ACM), the Institute for Electrical and Electronics Engineers (IEEE) Technical Committee for Security and Privacy, and Usenix.

Richard C. Linger

Richard C. Linger is the manager of the Survivable Systems Engineering group and a senior member of the technical staff in the NSS Program at the SEI. Linger directs research in Flow-Service-Quality engineering for network-centric system survivability, Function Extraction technology for automated computation of program behavior, and Next-Generation Software Engineering for ultra-large-scale system development. He also serves as a member of the faculty at the CMU Heinz School of Public Policy and Management and lectures at the School of Computer Science.

At IBM, Linger partnered with Dr. Harlan Mills to create Cleanroom Software Engineering technology for development of ultra-reliable software systems, including box-structure specification, function-theoretic design and correctness verification, and statistical usage-based testing for software certification. He pioneered use of Cleanroom technology for product development, achieving zero-defect performance with improved productivity, and founded and managed the IBM Cleanroom Software Technology Center. Linger has extensive experience in project management; system specification, architecture, design, verification, and certification; software re-engineering and reverse engineering; and process improvement, technology transfer, and education. He has published three software engineering textbooks, twelve book chapters, and over 60 papers and journal articles. Linger is a member of the IEEE and the ACM.

Howard F. Lipson

Howard F. Lipson is a senior member of the technical staff in the CERT Program at the SEI. Lipson has been a computer security researcher at CERT for more than thirteen years. He is also an adjunct professor in Carnegie Mellon University's Department of Engineering and Public Policy. He has played a major role in extending security research at the SEI and Carnegie Mellon into the new realm of survivability, developing many of the foundational concepts and definitions and making key contributions to the creation of new survivability methodologies. Lipson has been a chair of three IEEE Information Survivability Workshops. His research interests include the foundational concepts of survivability, the analysis and design of survivable systems and architectures, survivable systems simulation, critical infrastructure protection (specifically the electric power grid), and the technical and public policy aspects of Internet traceability and anonymity. He is co-principal investigator on a National Science Foundation award to investigate "Secure and Robust IT Architectures to Improve the Survivability of the Power Grid."

Lipson's early research at Carnegie Mellon included detailed workflow analyses of the incident response and vulnerability handling activities at the CERT/CC. He later designed and developed tools to automate and improve key aspects of

the incident response and security advisory processes. His work was recognized as a primary factor in the CERT/CC's ability to sustain its effectiveness in the face of the rapid growth of the Internet.

Prior to joining Carnegie Mellon Lipson was a systems design consultant, helping to manage the complexity and improve the usability of leading-edge software systems. Earlier, he was a computer scientist at AT&T Bell Labs, where he did exploratory development work on programming environments, executive information systems, and integrated network management tools. Lipson holds a PhD in computer science from Columbia University. He is a member of the IEEE and the ACM.

Thomas Longstaff

Thomas Longstaff is the Deputy Director for Technology in the Networked Systems Survivability (NSS) Program at the Software Engineering Institute (SEI). Longstaff has spent the past 12 years managing and initiating many of the CERT/CC's projects and initiatives such as the CERT Analysis Center, CERT Research Center, many survivability projects, and most recently Network Situational Awareness. His current scope of work includes evaluating technology across the entire NSS program to assure continued quality and innovation of all the work at CERT.

Longstaff is responsible for strategic planning for the NSS program, technology scouting for promising avenues to address security problems, and operating as a point of contact between research projects at Carnegie Mellon University and the NSS program.

Prior to coming to the Software Engineering Institute, Longstaff was the technical director at the Computer Incident Advisory Capability (CIAC) at Lawrence Livermore National Laboratory in Livermore, California. Longstaff obtained his MS in 1986 and PhD from the University of California, Davis in 1992 in software environments, and his BA from Boston University in 1983 in Physics and Mathematics.

Longstaff's research interests include network situational awareness, netflow analysis, insider threat, network traceback, and cyber/physical vulnerabilities.

Nancy R. Mead

Nancy R. Mead is a senior member of the technical staff in the Networked Systems Survivability Program at the Software Engineering Institute (SEI). The CERT Coordination Center is a part of this program. Mead is also a faculty member in the Master of Software Engineering and Master of Information Systems Management programs at Carnegie Mellon University. She is currently involved in the study of secure systems engineering and the development of professional infrastructure for software engineers. She also served as director of education for the SEI from 1991 to 1994. Her research interests are in the areas of information security, software requirements engineering, and software architectures.

Prior to joining the SEI, Mead was a senior technical staff member at IBM Federal Systems, where she spent most of her career in the development and management of large real-time systems. She also worked in IBM's software engineering technology area and managed IBM Federal Systems' software engineering education department. She has developed and taught numerous courses on software engineering topics, both at universities and in professional education courses.

Mead has more than 100 publications and invited presentations and has a biographical citation in Who's Who in America. She is a Fellow of the Institute of Electrical and Electronic Engineers, Inc. (IEEE) and the IEEE Computer Society and is a member of the ACM. Mead serves on the Editorial Boards for IEEE Security and Privacy and the Requirements Engineering Journal and is a member of numerous advisory boards and committees.

Dr. Mead received her PhD in mathematics from the Polytechnic Institute of New York and received a BA and an MS in mathematics from New York University.

Biographies

Andrew Moore

Andrew Moore is a Senior Member of the Technical Staff at the CERT Research Center of Carnegie Mellon University's Software Engineering Institute.

As a participant in the Network Systems Survivability Program, Mr. Moore explores ways to improve the security and survivability of unbounded systems through computer network attack and defense modeling, incident processing and analysis, and enterprise architecture engineering and analysis. Before joining the SEI in 2000, he worked for the Naval Research Laboratory investigating high assurance system development methods for the Navy. He has over fifteen years experience developing and applying mission-critical system development methods and tools, leading to the transfer of critical technology to both industry and the military. Mr. Moore received his BA in Mathematics from the College of Wooster and MA in Computer Science from Duke University.

While at the NRL, Mr. Moore served as member of the US Defense Science and Technology review (Information Technology TARA) panel on Information Assurance; the International Technical Cooperation Program, Joint Systems and Analysis Group on Safety-Critical Systems, (TTCP JSA-AG-4); and the Assurance Working Group of DARPA's Information Assurance Program. He has served as Principal Investigator on numerous projects sponsored by NSA and DARPA. He has also served on numerous computer assurance and security conference program committees and working groups. Mr. Moore has published a book chapter and a wide variety of technical journal and conference papers.

His research interests include computer and network attack modeling and analysis, adversary modeling, survivable systems engineering, formal assurance techniques, and security risk analysis.

Mark G. Pleszkoch

As a member of the Survivable Systems Engineering Team, Mark G. Pleszkoch works in the area of automation of formal methods. His current project, Function Extraction for Malicious Code (FX/MC), involves the automatic derivation of the functional behavior of disassembled assembly language code. Previously, Pleszkoch worked at IBM for twenty-one years in various capacities. As a member of IBM's Cleanroom Software Technology Center, he provided education and consultation to clients in software process, software engineering technologies, and software testing. Pleszkoch was the principal architect of the IBM Cleanroom Certification Assistant tool set for statistical testing automation.

Pleszkoch received his PhD in Computer Science from the University of Maryland at College Park and an MA and a BA in Mathematics from the University of Virginia. He has several publications in formal methods, software engineering and other topics. He served on the adjunct faculty in the Computer Science department of the University of Maryland, Baltimore County, from 1986 to 1995. As an undergraduate, Pleszkoch was a Putnam fellow of the Mathematics Association of America. He is a member of the Association for Symbolic Logic.

Stacy Prowell

Stacy Prowell is a senior member of the technical staff of the NSS Program at the Software Engineering Institute (SEI) of Carnegie Mellon University.

He is currently working on applied research and development of the Function Extraction for Malicious Code (FX/MC) system and related technologies.

Prowell has managed both commercial and academic software development projects and consulted on the design, development, and testing of applications ranging from consumer electronics to medical scanners, from small embedded real-time systems to very large distributed applications.

Prior to joining the SEI in 2005, Prowell was a research professor at the University of Tennessee. To support wider adoption of rigorous methods in industry, Prowell started the Experimentation, Simulation, and Prototyping (ESP) project at the University of Tennessee, which develops software libraries and tools to support application of model-based testing and sequence-based specification. Software developed by this program is in use by over 30 organizations. Prior to working at the University, Prowell worked as a consultant in the software industry.

Prowell's research interests include rigorous software specification methods, automated statistical testing, and function-theoretic analysis of program behavior.

Prowell holds a PhD in Computer Science from the University of Tennessee and is a member of the ACM, IEEE, and Sigma Xi.

Biographies

Gwendolyn H. Walton

Gwendolyn H. Walton is a senior member of the technical staff in the NSS program at the SEI. As a member of the Survivable Systems Engineering Team, she is currently involved in research on theoretical foundations for computation and automated analysis of software security attributes and function extraction for malicious code.

Prior to joining the SEI, Walton held faculty positions at Florida Southern College and the University of Central Florida. She published over 30 journal and conference papers and directed the research of 2 PhD students, 15 MS students, and 4 undergraduate students. Previously Walton served as President of Software Engineering Technology Inc, Assistant Vice President, Division Manager, Project Manager, and Senior Systems Analyst for Science Applications International Corporation, Senior Data Systems Programmer for Lockheed Missiles and Space Company, and Research Associate for Oak Ridge National Laboratory.

Walton received her PhD in Computer Science, MS in Mathematics, and BS in Mathematics Education from the University of Tennessee. She is a senior member of IEEE and the IEEE Computer Society, a senior member of the Society of Women Engineers, and a member of ACM.

Notes

Notes


CERT | Software Engineering Institute | Carnegie Mellon