www.cert.org/research


# CERT RESEARCH 2006 ANNUAL REPORT

Software Engineering Institute | Carnegie Mellon


Software Engineering Institute
Carnegie Mellon

The primary goals of the CERT® Program are to ensure that appropriate technology and systems management practices are used to resist attacks on networked systems and to limit damage and ensure continuity of critical services in spite of attacks, accidents, or failures.

CERT is part of the Software Engineering Institute (SEI), a federally funded research and development center (FFRDC) sponsored by the U.S. Department of Defense and operated by Carnegie Mellon University. The SEI advances software engineering and related disciplines to ensure the development and operation of systems with predictable and improved cost, schedule, and quality.

This report describes how CERT research advanced the field of information and systems security during the 2006 fiscal year.

To download a PDF version of this annual report, go to http://www.cert.org/research/2006research-report.pdf

# TABLE OF CONTENTS

# CERT Research Vision

**Thomas Longstaff**
Deputy Director, CERT
Software Engineering Institute
Carnegie Mellon University

Our 2006 research projects are a carefully chosen mix of continuations from 2005 and new starts that reflect the changing nature of computer security. CERT researchers in function extraction continue to produce ever more capable prototypes that explore Intel binary code, searching for the complete behavior of the program, both expected and surprising. We have helped form the vision of ultra-large-scale systems of the future (http://www.sei.cmu.edu/uls) and broadened a security vision that addresses insider threat, network flow analysis, and dynamic worm response.

This report reflects the broad nature of our projects as we push the frontiers of securing our critical systems. In addition to our researchers' projects, 2006 brought new collaborations and partnerships. As members of a research institution, we have always focused on transition of our research results and have historically accomplished this through publications, government pilot projects, and collaborative agreements with private industry. Now we have significantly enhanced our transition activities with new collaborations; two examples are our work with the Center for Identity Management and Identity Protection and the Information Technology Process Institute.

We participate in the Center for Identity Management and Identity Protection along with Utica College, the Federal Bureau of Investigation, U.S. Secret Service, LexisNexis, IBM, TransUnion, Indiana University, and the Case Center. Through this collaboration, we are able to apply our expertise to the nationally critical problem of identity theft, as well as offer our expertise and research results to the companies and law enforcement agencies that can take immediate action to address the threat.

We are collaborating with the Information Technology Process Institute (ITPI) to help promote best practices and methodologies for IT administrators and managers. The ITPI sponsors research to identify which practices and methodologies really make a difference in the security and profitability of organizations. The ITPI uses these results to further develop evaluation methods and publications that dramatically improve organizations' overall security and survivability. CERT is working closely with the ITPI in both research and in the development of IT professionals.

Within the CERT Program, we are taking advantage of our research results by developing products for our U.S. government, industry, and international customers. For example, we are working with US-CERT (http://www.us-cert.gov) to produce the "Build Security In" web site, which provides a wealth of knowledge and practices gained from our research and collaborations (http://buildsecurityin.us-cert.gov). We are applying the results of our research in function extraction (FX), which automates the calculation of software's functional behavior to the maximum extent possible, to support our analysis of malware and vulnerabilities. We are incorporating our research results on insider threat into course material for our Virtual Training Environment (http://vte.cert.org). All these efforts ensure that our research does not simply remain on a shelf or in conference proceedings but is used to solve difficult problems in network security.

The CERT vision for the future is built along these lines of research, prototyping, collaboration, and transition. Through each of these activities, we address the rapidly changing set of networked systems and security vulnerabilities that we face daily. Solving these problems involves incremental improvements in security. It also entails disruptive changes—leaps forward—in security technology that change the way we perceive the problem. Making progress in both ways requires a balanced and cooperative approach.

Our incremental approaches to security include improving our ability to analyze network behavior, measure and promote best practices for security, and apply security architectures to environments such as electric power systems and embedded computing. These projects are essential to the continual improvement of security in our current environment; we cannot wait for a future breakthrough to come along before we address our current vulnerabilities.

Just as important, however, is our research that is designed to redefine the way we look at the security problem. In our FX work, we hope to change the way code is created and integrated, making the complete behavior of the code evident at the same time the code is created. Similarly, with our integration work, we support an environment where the integration of components can be well understood before we deploy them in a hostile environment. For these projects to be successful, we must shift our perspective on software development—from an art form with few constraints to a science of design that can compute the essential properties of software before it is created. This change is one of the key contributions we hope to make to the ultra-large-scale systems of the future.

The main difference between the security solutions available today and those we envision is a shift from reactive to predictive security. Today's solutions revolve around detection and response for security problems. Whether this be patching vulnerable software, tracking down an active bot network, or detecting a scan, we spend much of our resources on reacting to security problems. Our preventative solutions tend to be designed to repel known attacks, viruses, worms, and other problems.

On the other hand, predictive security involves the rigorous analysis of new and existing designs to predict where the security problems will arise in order to manage that risk before the implementation begins. To achieve this, we need to gain a deeper understanding of how our systems are susceptible to security flaws and to generate the alternative designs necessary to create systems that inherently require fewer responses to security problems. The achievement of these predictive technologies will change the nature of system protection; they are the core of our projects that we hope will result in major advances in disruptive solutions.

You will see evidence of other potentially disruptive technologies in the subsequent pages of this report as well as our incremental progress in achieving security in the current computing environment. Behind all these projects is a vision of the future that includes a scientifically rigorous understanding of every level of activity in our networked systems. We should never have network activity that cannot be linked to software behavior we understand and can predict.

We are striving for an environment where the complexities of software and networked systems can be managed through computation and analysis, where criminal behavior can be identified and attributed to the source, and where we are confident in the protection of our identity. We are also working to effectively train and equip our cyber-enabled workforce to take full advantage of this understood and predictable environment to produce value without fear of having that value stolen or misused.

This vision is not without significant challenges. Our current and future research agenda will continue to address the most difficult aspects of achieving this vision, both along paths we can see and those that may surprise us. We welcome participation with other organizations that feel as we do—that the networked environment of the future can be a place that is secure and well understood. If you see projects and ideas in this report that spark your imagination, work with us to help reach for this new, safer networked world.

# Executive Summary

*Reporting on the period ending September 30, 2006*

The CERT Program at Carnegie Mellon University's Software Engineering Institute concentrates on the technical basis for preventing security flaws and vulnerabilities from occurring in the first place, for identifying them once they have occurred, and for preserving essential services if a system is penetrated and compromised. We recognize that we must focus on multiple strategies for prevention, detection, and recovery from cybersecurity attacks. 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 and security engineering. In our technology development work, we create software and security standards, technologies, and automation. In technology transfer, we work with clients to incorporate results into key acquisition and development projects. We also provide training and other materials such as published books and articles to support technology transfer.

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 of critical systems and systems of 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. We also describe additional research activities, our publications, and technical leadership activities.

### LEVANT: Protocols for Anonymity and Traceability Tradeoffs

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.

### RAVE: Network Flow Visualization

Fully automated analysis is a valuable tool for network situational awareness, but few techniques can discern subtle patterns in noisy data as well as human visual inspection. The CERT Network Situational Awareness Group has developed the Retrospective Analysis and Visualization Environment (RAVE), an operational environment for generating visualizations and making them available to presentation applications.

The team has used RAVE to investigate improvements to the presentation of time-series network flow data and explore ways of characterizing host behavior. A particular challenge in automated visualization of volatile data such as network flow is that very few assumptions can be made about the data to facilitate visualization. In the detailed description of RAVE, the impact of this challenge on the work and how the team has overcome it is discussed.

### Port and Payload Agnostic Application Identification

To properly understand the threats that a network faces, network administrators must access current, accurate information about the composition of their networks. However, the considerable expense of continuous traffic monitoring and deep packet examination raises the need for a method that can identify applications without relying on payload.

This effort is focused on application identification without reliance on payload or dependence on the port number. By developing behavioral tests, CERT researchers can create accurate methods for classifying traffic and inventorying the network's behavior, even when the payload or port number is untrustworthy. The premise of this approach is that an application's design has definite goals that influence its behavior over a network.

For its analysis, the team uses network flow data that is classified and fed into various behavioral tests that are calibrated to identify existing applications. CERT has developed a small behavioral dictionary for identifying BitTorrent clients using this methodology, resulting in a technique that has been highly effective.

### SQUARE: Requirements Engineering for Improved System Security

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 researchers have developed an end-to-end process for security requirements engineering to help organizations build security into the early stages of the production life cycle. The SQUARE methodology consists of nine steps that generate a final deliverable of categorized and prioritized security requirements. The process has been baselined and several case studies with real-world clients have shown that the methodology holds good promise for incorporation into industry practice. CERT has prototyped a computer-aided software engineering (CASE) tool to support each stage of the SQUARE process.

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

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 is currently engaged in the Function Extraction for Software Assurance project, and is ready to expand application of function extraction technology in four projects: Computational Security Attributes, Automated Correctness Verification, Automated Component Composition, and Flow-Service-Quality Engineering.

**STAR*Lab Function Extraction for Software Assurance: Engineering Automation for Computing Software Behavior**

STAR*Lab recognizes the importance of software assurance to national defense. Software assurance depends on knowing and verifying the complete behavior of software, because behavior that is not known can contain errors, vulnerabilities, and malicious content. To help address this need, CERT is conduction research and development on the emerging technology of function extraction (FX). The goal of this project is to compute the behavior of software to the maximum extent possible with mathematical precision. Computation of the behavior of malicious code is of particular interest, to enable analysts to quickly determine intruder objectives and develop countermeasures.

**STAR*Lab Computational Security Attributes: Engineering Automation for Software Security Analysis**

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.

**STAR*Lab Automated Correctness Verification: Engineering Automation for Software Assurance**

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.

**STAR*Lab Automated Component Composition: Engineering Automation for Understanding System Behavior**

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 help determine the net effect of combining components in network architectures.

**STAR*Lab Flow-Service-Quality (FSQ) Engineering: Foundations for Developing Network-Centric Systems**

FSQ engineering provides foundations for mastering complexity and improving survivability in analysis and development of large-scale, network-centric systems. The objective of the FSQ project is to define 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.

**SAF: Survivability Analysis Framework**

Large systems and particularly systems of systems raise the importance of complexity management to mitigate complexity's negative effect on survivability. Current survivability analysis mechanisms are limited by their focus on single-system considerations. The objective of this research is to develop survivability analysis methods that are applicable to systems of systems and address the challenge of increased demands for interoperability and integration. Increasingly the functions and services needed to satisfy a specific mission span multiple systems. Such a system of systems might involve several agencies' systems involved in performing a work process. Through the SAF, the work process thread can be analyzed end to end and step by step to identify events that could lead to failure to successfully complete the process.

**The Uncleanliness Vector: Histories of Hostile Activity**

In an operational network security analysis environment, there is little time for gathering context information on attackers. Analysts need tools that present background information in an easily digestible way. The Uncleanliness Vector (UV) project is an attempt to provide context information on external networks to security analysts that indicates whether the network in question has engaged in other malicious activity and, if so, what kind. The Uncleanliness Vector describes the likelihood of attack from a continuous block of IP addresses using an uncleanliness score. The UV is composed of several distinct uncleanliness elements, weighted to indicate a prioritization by their relative threat level to the monitored network. The UV score is a weighted average of these elements. Through analysis of scanning data on the Internet, the team identified patterns and characteristics regarding the distribution and sources of malicious scanning activity and used these results to construct a useful measure for uncleanliness. This work led to definition of a corruption measure as a complement to the Uncleanliness Vector and development of corruption measures is also being explored.

# 3
# **LEVANT:** Protocols for Anonymity and Traceability Tradeoffs

**Howard Lipson**
Principal Investigator
412-268-7237

**Sven Dietrich**
Principal Investigator
412-268-7711

# **LEVANT:** Protocols for Anonymity and Traceability Tradeoffs

### 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 extremely limited by the existing protocol and infrastructure design and requires a major re-engineering effort from both technical and policy perspectives. This is discussed 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. This will 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 are routinely resolved in a number of venues, for example 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's 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, caller ID customers can choose to block all incoming calls from anonymous callers. The anonymous caller is notified of this fact by an automated message. For callers who pre-arrange with their phone companies to be anonymous by default, the only way to complete a call is to enter a key sequence to remove the anonymity for that particular call and to redial. Customers who 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 may 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 perceived 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 that 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 provide 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.

### 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. The concept of *k*-anonymity was first defined by Pierangela Samarati [3]. Samarati showed how generalization and suppression of data can be used to enforce *k*-anonymity in private databases (such as those containing medical and driver's license data) thereby reducing privacy loss. This concept was later reiterated by Latanya Sweeney in the context of medical databases [4].

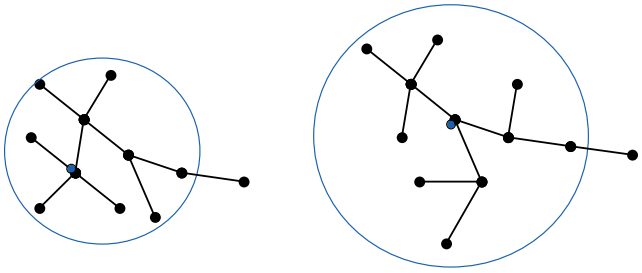


*Figure 1: Examples of k-anonymity*

### 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. Below are 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. Below are 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) knowledge base that is more comprehensive and of higher quality than each could build on its own. This knowledge base 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 knowledge base 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 knowledge base 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 knowledge base 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 upon, 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 [5].

### 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 be of critical value. It 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 these tradeoffs to be dynamic, fine grained, and based on the specific mission needs 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 who act either as individuals or within specific organizational roles.

**2006 Accomplishments**

In FY2006, we continued our work towards establishing a solid theoretical foundation on which to base principled engineering tradeoffs between traceability and anonymity. Our research has explored the range of engineering requirements for the design of Internet protocols that support traceability and anonymity attribute tradeoffs and the negotiations that are needed to set the desired level of each attribute. We've generated and analyzed several LEVANT protocol scenarios of use that include specific user requirements for anonymity and traceability that must be satisfied for particular applications, systems, and missions. We've also investigated the underlying security and survivability themes in this research, in particular with respect to the engineering tradeoffs being explored for protocol design. Our research has also examined policy issues relating to the design and use of protocols that support negotiated levels of anonymity and traceability for individual actors and for organizations.

**2007 Plans**

In FY2007, we plan to complete an SEI technical report on our LEVANT project research. One or more papers derived from this technical report are also planned, along with the submission of funding proposals seeking long-term support for the LEVANT project.

**References**

[1] Lipson, Howard F. *Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues* (CMU/SEI-2002-SR-009). 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, 1999. http://www.cert.org/archive/pdf/busperspec.pdf.

[3] Samarati, Pierangela. "Protecting Respondent's Privacy in Microdata Release." *IEEE Transactions on Knowledge and Data Engineering 13,* 6 (Nov./Dec. 2001): 1010-1027. http://seclab.dti.unimi.it/Papers/tkde_k-anonymity.pdf.

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

[5] Lipson, H. & Dietrich, S. "Levels of Anonymity and Traceability (LEVANT)" 4-12. In *Results of SEI Independent Research and Development Projects and Report on Emerging Technologies and Technology Trends,* 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. (CMU/SEI-2004-TR-018). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2004. http://www.sei.cmu.edu/publications/documents/04.reports/04tr018.html.

# 4
# **RAVE:** Network Flow Visualization

**Phil Groce**
Principal Investigator
412-268-5040

# **RAVE:** Network Flow Visualization

### Problem Addressed

Network situational awareness depends on informed and effective analysis of network activity. Network monitoring and measurement tools provide copious information on the behavior of actors in instrumented networks. Sifting through this data for useful, actionable information presents a constant challenge. Automated analysis approaches sometimes yield excellent results that would be difficult to produce with visual inspection. One example is CERT's work on detecting scanning activity [1].

Nonetheless, the human eye can quickly and reliably discern detail in extremely dense information sets [2]. It is one of the most sensitive instruments available to network and security operations. Automation can assist in analysis by selecting and manipulating data and by generating visualizations.

The Retrospective Analysis and Visualization Environment (RAVE) is an operational environment for generating visualizations and making them available to presentation applications. To facilitate rapid workflow, RAVE caches visualizations and intermediate analysis results for reuse in different contexts, for repeated viewings, or for browsing of related data. We have used RAVE to improve client operations and as a springboard to investigate a different problem—how to improve the visualization of network monitoring data.

### Research Approach

The core of RAVE is a file-based cache containing intermediate data (for use in RAVE analyses) and visualization data that external applications consume and present to the end user. This cache is presented to the Python-based programming environment through an API based on Python decorators [3]. The cache then stores the results of these analyses. If more complex analyses are composed of simpler ones, the return values of the simpler analyses will be implicitly cached and reused by any other code which invokes the analysis with the same arguments.

To return the results of analyses to end users, we provide raved, an HTTP-based visualization service provider. Presentation applications (usually web based) retrieve analyses from raved in a manner similar to the retrieval of static images. If a cached visualization exists, it is returned; otherwise, the analysis that generates that visualization executes, and that result is cached and returned instead. The raved application is both a working service provider and a proof of concept; other service providers could quickly be written for other network service architectures, such as XML Remote Procedure Calling protocol (XMLRPC).

### Time-Series Network Visualization

The analysis and visualization framework RAVE provides gives us a starting point for addressing problems in network flow visualization. One area in which we have focused our efforts is the visualization of network flow volume from disparate sources. The results of this work are available to RAVE users in the RAVE visualization layer.

Especially in large networks, a particular challenge in visualizing network flow volume over time is its high variability. We've found through our observations in client networks that flow volume can briefly deviate significantly from the normal flow volume for a sample: either positively (e.g., if a large file transfer during a period of inactivity increases the observed number of bytes) or negatively (e.g., if a router or sensor goes down). If we scale the visualization to include these important but obvious events, less obvious changes in flow volume become less noticeable.

We can remedy this by denoting such obvious events without directly plotting them on the scale. Carets at the top of the scale denote observations that fall outside the 95th percentile of observations for this time period. With the outliers noted but not plotted, detail that would have been lost is now clearly visible.

We see another common challenge in time-series network information when comparing the traffic volume of different sensors for the same time period. Frequently, the median volume between sensors is quite disparate. When the two series are plotted on the same axis, we encounter a problem similar to the one above; any scale that can accommodate both series "flattens" detail within the series, making it difficult, for example, to identify proportional changes that are highly correlated between the two series.

However, we wish to retain information on the magnitude of traffic observed by the sensors relative to one another. One approach is to adjust the scale to show both relative traffic magnitude and detail within each series. Alternately, we can plot both series over one another on different scales.

These approaches work for some data sets; for others with more variance within the data and/or similar magnitudes between the two data sets, observations within the two data sets may overlap in ways that make the visualization confusing. This is undesirable for automatic visualization; the approach should generate an informative data visualization in all but the rarest cases. Our response to this is to present independent series of observations in independent axes, all scaled equally by time and differently depending on the observations in the series. Because the scales are independent, it is impossible for one series to affect the scaling of another.

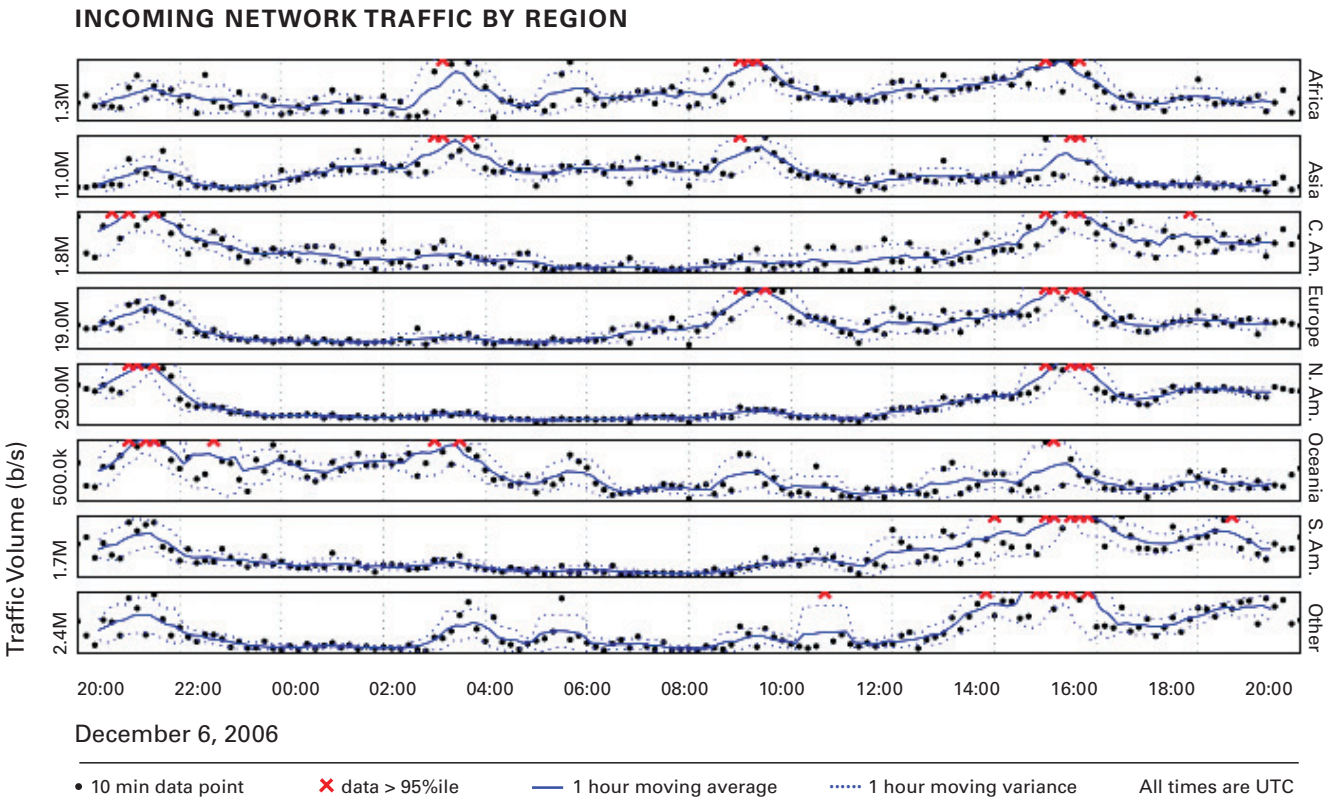


*Figure 1: Incoming Network Traffic by Region*

The maximum plotted value for each scale is available at the upper left of each plot for comparison. The technique described above for handling outlier observations prevents them from washing out detail within individual series.

Figure 1 illustrates our approach. Changes over time are clear and comparable. By plotting events outside the 95th percentile of observed activity outside the scale, we preserve a more detailed characterization of activity.

### Non-Time-Series Network Visualization

An exciting area for future research is visualizing non-time-series network monitoring information. One example of such research is the host characterization plot.

Figure 2 shows a radar plot [4] characterizing host activity over a given time period on seven axes. The axes on the left denote network activity in which the host behaved as a server, while the axes on the right denote activity as a client. Each side is broken down into web-related, mail-related, or other activity. The axis pointing straight down is activity that cannot be classified as either client or server behavior; this is implicitly "other" activity as well, as we cannot determine whether it is mail or web related. The labels at the end of the "other" axes display the most prevalent type of traffic in that sample, which can be TCP or user datagram protocol (UDP) traffic to/from a given port; Internet control message protocol (ICMP) traffic of a given message type; or traffic corresponding to a different protocol, in which case we display the Internet Assigned Numbers Authority (IANA) protocol number as listed in the IP header.

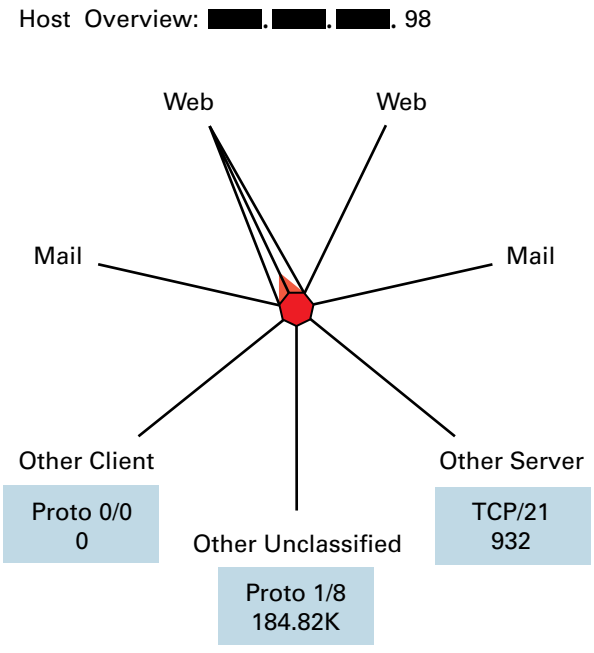


*Figure 2: Characterization of a Specific Network Host*

**Expected Benefits**

RAVE is most useful in operational environments, where we intend for it to facilitate good decision making by network and security operations personnel. In these environments, personnel frequently look at the same or very similar data repeatedly, either to review another's work or to refine an interesting observation from another analysis. In these environments, working from cached data can reduce time the operations personnel spend waiting for analyses to run, improving response time to network events. Caching also reduces load on the primary data store, improving performance for those queries which it must perform.

RAVE can provide visualization capability to analysis tools that currently have none, without introducing a requirement for a specific user interface. This includes the tools in CERT's NetSA Security Suite [5]; in addition, work on RAVE includes development of Python-based methods for manipulating data with these tools.

**2006 Accomplishments**

RAVE has been deployed with a sponsor as the analysis backend to a web portal. Feedback from this deployment has been used to improve RAVE's robustness and scalability. The portal is currently in a pre-production phase, with rollout expected to occur in 1Q 2007.

To facilitate further integration of RAVE into a web application, we have developed an Apache module to proxy RAVE connections. This allows us to use Apache for authentication and access control.

**2007 Plans**

We expect to further deploy RAVE with additional clients during 2007. We will integrate new analyses with RAVE and explore the feasibility of other user interface techniques such as AJAX or thick-client user interfaces.

**References**

[1] Gates, Carrie; McNutt, Josh; Kadane, Joseph B.; & Kellner, Marc. *Detecting Scans at the ISP Level* (CMU/SEI-2006-TR-005). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr005.pdf.

[2] Tufte, Edward. *The Visual Display of Quantitative Information,* 161-162. Cheshire, CT: Graphics Press, 2001.

[3] Smith, Kevin D.; Jewitt, Jim; Montanaro, Skip; & Baxter, Anthony. *Decorators for Functions and Methods.* http://www.python.org/dev/peps/pep-0318.

[4] Wilkinson, Leland. *The Grammar of Graphics,* 2nd ed., 214-215. New York, NY: Springer Science+Business Media, Inc., 2005.

[5] CERT Network Situational Awareness Group, Software Engineering Institute. *Monitoring for Large-Scale Networks.* http://tools.netsa.cert.org.

# 5 **Port** and **Payload** Agnostic Application Identification

**Michael Collins**
Principal Investigator
412-268-9124

# **Port** and **Payload** Agnostic Application Identification

### Problem Addressed

In order to properly understand the threats that a network faces, network administrators need access to current and accurate information about the composition of their networks. One solution to this problem is to continuously audit the network by monitoring traffic. However, deep packet examination is expensive; a method that can identify applications without relying on payload is therefore desirable.

This logistical problem is complicated by evasion: both attackers and certain users have an active interest in hiding their network activities. The most notable example of this evasive behavior involves peer-to-peer applications. Several implementations of peer-to-peer tools incorporate encryption in order to evade detection [1].

The focus of this effort is application identification on a network without reliance on payload or dependence on the port number. By developing behavioral tests, we can create accurate methods for classifying traffic and inventorying the network's behavior, even when the payload or port number is untrustworthy. In addition, these techniques may lead to engineering insights about how to construct a network that is unfriendly to attack.

### Technical Approach

The premise of this approach is that an application's design has definite goals that influence its behavior over a network. For example, given that bandwidth is a finite resource and often constrained at the endpoints of a network, an application will not use more bandwidth than is necessary to accomplish its task. Therefore, a chat application (such as AOL Instant Messaging, Internet Relay Chat, or Zephyr Online) will not use a 64 KB packet to send a 10-character message.

For our analyses, we use network flow data. The flows are divided into various flow classes using the classification tree in Figure 1; these classes define what type of information we can reasonably acquire from a flow. For example: in a Short Flow, there is little, if any, payload, but the flag combinations are less ambiguous than in a message or a file transfer.

The classified flows are fed into various behavioral tests that are calibrated to identify existing applications. For example, such a test might determine the number of client attempts to access a service that result in failed connection (that is, a SYN packet without any payload-bearing interaction). Automated connections for services such as Simple Mail Transfer Protocol (SMTP) or BitTorrent are more likely to generate failed connections than a user-driven application such as a web browsing.

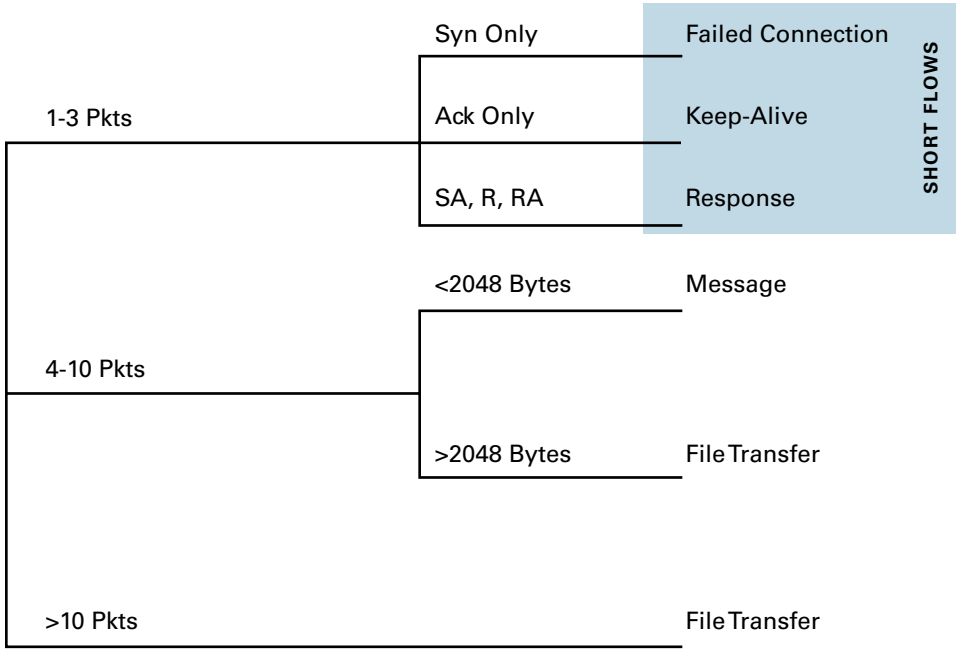


*Figure 1: Flow Classification Tree*

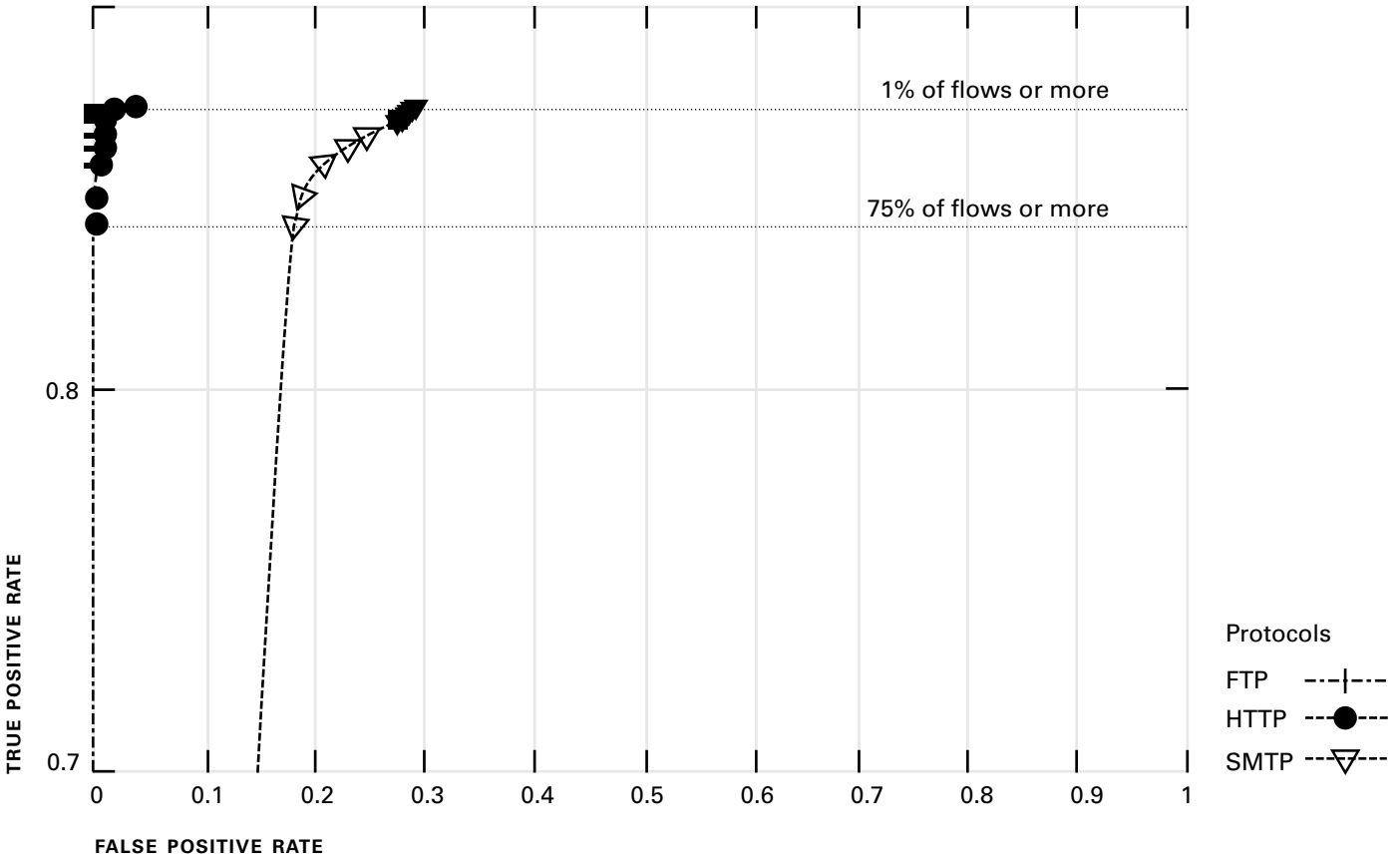


*Figure 2: Percentage of Failed Connections in Several Common Protocols vs. BitTorrent*

Figure 2 shows this behavior in depth; in this figure, we compare the probability that a high degree of failed connections involves BitTorrent traffic as compared to FTP, HTTP, or SMTP traffic. As this figure shows, BitTorrent is easily distinguished from both FTP and HTTP traffic, while the SMTP traffic is not as easily distinguishable.

**2006 Accomplishments**

In 2006, CERT developed a small behavioral dictionary for identifying BitTorrent clients using this methodology [1]. The resulting system was tested on netflow data collected on a large (multiple/8) network instrumented by CERT. The resulting technique was highly effective.

Figure 3 on the next page shows the results of this decision mechanism in depth. It compares the results of using a multiple-vote classification mechanism for distinguishing BitTorrent from a limited set of applications: email, HTTP, and FTP. Email and HTTP were chosen on the grounds that they were the two most common services crossing the client's network. FTP was chosen on the assumption that as a protocol primarily concerned with transferring files, it would closely resemble BitTorrent. The tests used involved examining the estimated bandwidth during large file transfers (BitTorrent should presumably be slower than the single server protocols), the number of failed connections (as above), a comparison of message sizes, and the maximum amount of data transferred in a single session.

As this figure shows, the resulting mechanism is capable of distinguishing between BitTorrent and other applications with a high degree of success: 90% of BitTorrent applications are identified by at least two tests, while nearly 100% of the other applications are eliminated after two affirmative tests. This mechanism, which involves no payload and which does not rely on the port number to identify the application, is consequently a feasible method for distinguishing applications.

CERT has developed similar behavioral tests for identifying and distinguishing spam. The behavioral methodology is very attractive for identifying spam sources as such efforts involve distinguishing abuse of a service from the service itself, meaning that the port number is no longer a relevant detection mechanism.

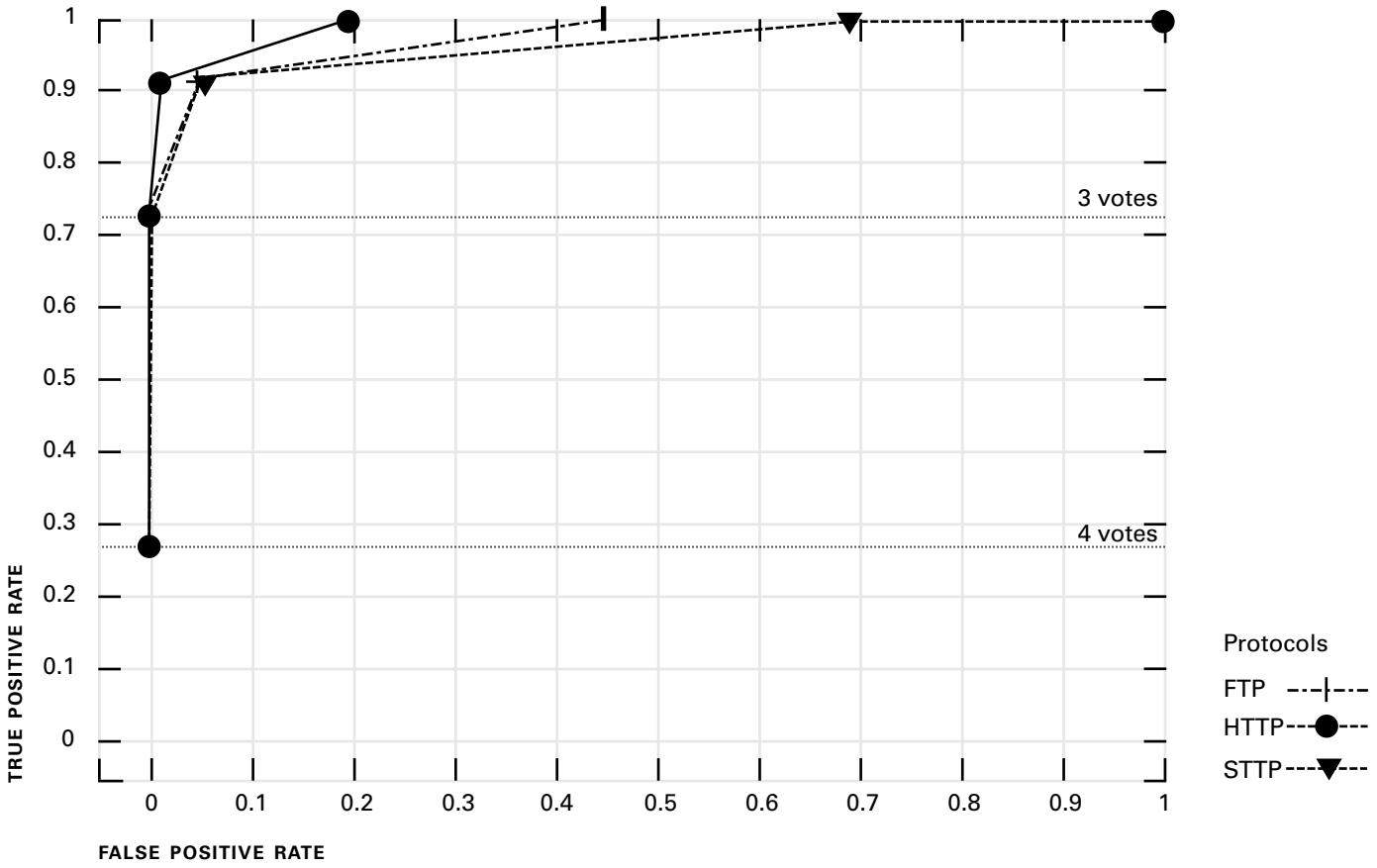


*Figure 3: Integrated ROC Curve for Identifying BitTorrent*

**2007 Plans**

In 2007, the CERT Network Situational Awareness Group plans to expand this methodology in two ways: we will expand the behavioral "dictionary" by examining and testing other forms of classification mechanisms. The methods used so far in this work are used to identify file-transfer protocols (e.g., BitTorrent, FTP, SMTP, and HTTP). Other protocols (e.g., AIM and RealVideo) will not be as easily identified using the current body of tests. In addition, as more tests are developed they will naturally overlap and contradict each other; consequently a more sophisticated decision-making mechanism will be required to identify applications.

Another venue for research is examination of the impact that these behavioral filtering mechanisms will have on application design. For example, identifying and filtering attributes that are part of the BitTorrent protocol will naturally encourage developers to modify the protocol in order to evade detection. It is conceivable that certain behaviors may be constrained to the point that the protocol is no longer attractive for its target audience. This is a particularly relevant question with spam, where a definite profit motive can be attached to sending spam emails.

**References**

[1] TorrentFreak Weblog. *Encrypting BitTorrent to take out Traffic Shapers.* http://torrentfreak.com /encrypting-BitTorrent-to-take-out-traffic-shapers (2006).

[2] Collins, M. & Reiter, M. "Finding Peer-To-Peer File-sharing Using Coarse Network Behaviors." *Proceedings of the 2006 European Symposium on Research in Computer Security (ESORICS) Conference.* Hamburg, Germany, Sept. 18-20, 2006. *Lecture Notes in Computer Science 4189,* Berlin/Heidelberg, Germany: Springer, 2006. http://www.springerlink.com/content/u265q75n3568.

# 6 **SQUARE:** Requirements Engineering For Improved System Security

**Nancy Mead**
Principal Investigator
412-268-5756

# **SQUARE:** Requirements Engineering For Improved System Security

### 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 are detected during requirements development [1, 2]. 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 [3], and the percentage of defects originating during requirements engineering is estimated at more than 50 percent [4]. The total percentage of project budget due to requirements defects is 25 to 40 percent [5].

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 [6]. The National Institute of Standards and Technology (NIST) reports that software faulty in security and reliability costs the economy $59.5 billion annually in breakdowns and repairs [7]. The costs of poor security requirements make apparent that even a small improvement in this area will provide a high value. By the time 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 reasons that 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, and virus detection tools. 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 while significant attention is given to the functionality of the system from the user's perspective, 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

CERT 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. This determination depends 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 management'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.

The methodology is most effective and accurate when conducted with a team of requirements engineers with security expertise and the stakeholders of the project. SQUARE's nine discrete steps, 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 ensuing 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.

The SQUARE process has been baselined. Several case studies with real-world clients have shown that the methodology holds good promise for incorporation into industry practice, and we are working informally with additional industry clients. The current model is summarized in Table 1. CERT 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, documented, and inspected 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.

| | 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*

**2006 Accomplishments**

A paper on the educational aspects of SQUARE was presented in April 2006 at the Conference on Software Engineering Education and Training (CSEET) [8]. SQUARE is also described in the requirements engineering section of the "Build Security In" web site [9], and in two books [10, 11]. The SQUARE baseline is described in a technical report on SQUARE [12]. In conjunction with Cylab, the prototype tool is being completed in a way that is more modular and has a more appealing user interface. Also in conjunction with Cylab, workshop, tutorial, and educational materials on SQUARE are being produced.

**2007 Plans**

A promising development is identification of the essential SQUARE steps (SQUARE-lite) to be used to supplement a client's existing requirements engineering process. The team plans to refine requirements elicitation and analysis methods for security properties in conjunction with process application for leading-edge clients. The prototype tool development will be completed, and we will seek to work with a commercial vendor to further develop the tool as a commercial product. Workshop, tutorial, and academic educational materials on SQUARE will be completed. SQUARE will be further documented in a requirements engineering chapter of a book based on the "Build Security In" web site.

**References**

[1] Boehm, B. W. & Papaccio, P. N. "Understanding and Controlling Software Costs." *IEEE Transactions on Software Engineering SE-4,* 10 (Oct.1988): 1462-77.

[2] McConnell, Steve. "From the Editor - An Ounce of Prevention." *IEEE Software 18,* 3 (May 2001): 5-7.

[3] Jones, Capers, ed. *Tutorial: Programming Productivity: Issues for the Eighties, 2nd ed.* Los Angeles, CA: IEEE Computer Society Press, 1986.

[4] Wiegers, Karl E. *Inspecting Requirements* (column). StickyMinds, July 30, 2001. http://www.stickyminds.com.

[5] Leffingwell, D. & Widrig, D. *Managing Software Technology–A Unified Approach.* Boston, MA: Addison-Wesley, 1999.

[6] Soo Hoo, Kevin; Sudbury, Andrew W.; & Jaquith, Andrew R. "Tangible ROI through Secure Software Engineering." *Secure Business Quarterly 1,* 2 (2001).

[7] 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).

[8] Mead, N. R. & Hough, E. D. "Security Requirements Engineering for Software Systems: Case Studies in Support of Software Engineering Education," 149-156. *19th Conference on Software Engineering Education & Training.* Turtle Bay, HI, April 19-21, 2006. New York, NY: IEEE Computer Society, 2006.

[9] Software Engineering Institute. "Build Security In." https://buildsecurityin.us-cert.gov/ (2005).

[10] Mead, N. R.; Davis, N.; Dougherty, C.; & Mead, R. Ch. 8, "Recommended Practices," 275-308. *Secure Coding in C and C++.* Robert Seacord. Upper Saddle River, NJ: Addison Wesley, 2005.

[11] Mead, N. R. Ch. 3, "Identifying Security Requirements Using the SQUARE Method," 44-69. *Integrating Security and Software Engineering: Advances and Future Visions,* H. Mouratidis & P. Giorgini. Hershey, PA: Idea Group, 2006, (ISBN: 1-59904-147-2).

[12] 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.

7

# STAR*Lab

**Richard Linger**
Principal STAR*Lab
Investigator
301-926-4858

**Gwendolyn H. Walton**
Principal Investigator
412-268-7700

**Mark Pleszkoch**
Principal Investigator
412-268-7700

**Kirk Sayre**
Principal Investigator
412-268-7700

**Stacy Prowell**
Principal Investigator
412-268-9205

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

### Developing Engineering Automation for Challenge Problems in System Security

CERT has established a software development laboratory in response to the growing needs of its customers. The mission of STAR*Lab (Security Technology Automation and Research Laboratory) is development of theory-based prototype automation that provides operational solutions to challenge problems in security engineering and software assurance.

Challenge problems are long-standing barriers to progress identified by Department of Defense 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 assurance. The focus of STAR*Lab is not on producing studies and reports that may leave implementation speculative and undone, but rather on applying theory to develop working tools. The laboratory is dedicated to helping customers achieve three objectives:

1. **Faster development:** Solutions must replace time- and resource-intensive operations with engineering automation that permits faster system development.
2. **Improved quality:** Solutions must substitute foundations-based automation for fallible human processes to improve system security and dependability.
3. **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:

1. **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.
2. **The proof-by-automation principle.** Automation is essential to replace fallible and resource-intensive human operations with solutions that enable full intellectual control. All projects will demonstrate solutions through automated engineering tools.
3. **The practical application principle.** Automation must solve challenges with practical engineering operations for routine use by practitioners. All projects will scale up engineering solutions for widespread 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 sponsor resources. 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 heuristic or partial 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 application.

### STAR*Lab Projects

Star*Lab is currently engaged in the Function Extraction (FX) for Software Assurance project. This multiyear effort has satisfied the requirements of Gate 3 and is proceeding to Gate 4. In addition, the laboratory is ready to capitalize on function extraction technology in four potential project areas:

- **Computational Security Attributes.**
  In 2006, STAR*Lab completed this IRAD (Internal Research and Development) study to develop foundations for computational evaluation of the security properties of software. This FX-based project has satisfied requirements for Gate 2 and is ready for sponsorship to Gate 3.
- **Automated Correctness Verification.**
  This FX-based project has satisfied the requirements of Gate 2 and is ready for sponsorship to Gate 3.
- **Automated Component Composition.**
  This FX-based project has satisfied the requirements of Gate 2 and is ready for sponsorship to Gate 3.
- **Flow-Service-Quality Engineering.**
  This FX-based project has satisfied the requirements of Gate 2 and is ready for sponsorship to Gate 3.

These projects are described in the following sections.

# **STAR*Lab** Function Extraction for Software Assurance: Engineering Automation for Computing Software Behavior

Principal Investigator: Richard Linger

**Problems Addressed and Research Approach**

STAR*Lab recognizes both the importance of software assurance to national defense and economic security and the difficulty of achieving it. Software assurance depends on knowing and verifying the complete behavior of software. Unfortunately, nothing less will do, because behavior that is not known can contain errors, vulnerabilities, and malicious content.

It is a sobering fact that current software engineering provides no practical means for developers to determine the full behavior of software, and no testing effort, no matter how elaborate, can exercise more than a small fraction of possible behavior. Because of the size and complexity of software systems, current-generation software engineering must often operate in a world of incomplete knowledge of software behavior.

Understanding behavior today is an error-prone, resource-intensive process carried out primarily through program reading and analysis. Software development environments and specialized program analysis tools in use today support some aspects of program comprehension, such as navigation, clustering, metrics, and hypothesis generation and verification. But these tools focus on supporting what remains a cognition-intensive activity, and cannot define behavior in all circumstances of use.

Complex software systems are difficult to understand because of their immense numbers of execution paths, any of which can contain errors and security exposures. Faced with innumerable execution possibilities, developers and analysts often achieve no more than a general understanding of system behavior. This technology gap is a challenge problem at the heart of many issues in present-day software and security engineering. Simply put, systems experience errors and vulnerabilities in large measure because their developers have no practical means to determine what they do in all possible uses.

It is becoming evident that costly human methods for analyzing software behavior are inadequate for coping with today's systems, let alone the ultra-large-scale systems of the future, and that automated methods of behavior analysis will be essential for scaling up. This problem has technical roots but substantial non-technical impact. It affects the software in systems ranging from defense and telecommunications to banking and transportation, to name a few.

While achieving software assurance has been limited by human capabilities in the past, it may not be so in the future. Function-theoretic foundations of software illuminate a challenging but feasible strategy for developing automated tools to calculate the behavior of software and present it to users in understandable form. This next-generation capability can help transform software engineering into a rigorous computational discipline. Other engineering disciplines have made this transformation to computational methods to their everlasting benefit. For example, electronics engineers routinely employ computational methods to design processors containing millions of circuits, a feat well beyond human capabilities and totally dependent on automated analysis.

STAR*Lab is conducting research and development in the emerging technology of function extraction (FX). The objective of FX is to move from an uncertain understanding of program behavior derived in a human time scale of days to a precise understanding automatically computed in a machine time scale of seconds. This technology applies function-theoretic foundations to automate calculation of the functional behavior of software to the maximum extent possible. These foundations define the transformation of code structures into procedure-free functional form and are the basis for the function extraction process. 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. While theoretical results impose some constraints on behavior calculation (for example, for certain forms of loops) STAR*Lab development of engineering solutions suggests that nearly all software behavior will be amenable to calculation. And any level of behavior calculation can help improve human capabilities for software understanding and analysis.

To explore the impact of FX technology, STAR*Lab developed a proof-of-concept function extractor prototype that calculates the behavior of programs expressed in a small subset of the Java programming language and presents it to users in the form of behavior catalogs. The catalogs provide procedure-free definitions of the net functional effect of programs from input to output in all circumstances of use, essentially their as-built specifications.

The FX prototype was employed in a controlled experiment sponsored by the SEI to compare traditional methods of program reading and inspection with FX-based methods [3]. Experienced programmers were divided into a control group using traditional methods and an experimental group using the FX prototype. Each group was required to answer questions dealing with comprehension and verification of three Java programs. The experiment produced the following results:

- The experimental group using the FX prototype reduced the time required to derive the functional behavior of the programs by several orders of magnitude compared to the control group.
- The experimental group was about four times better at providing correct answers to the comprehension and verification questions, and required a fourth of the time to do so, a productivity improvement of a factor of 15 over the control group.
- The experimental group achieved these results with 45 minutes of instruction on use of the function extractor, compared to years of experience for the control group.

Function extraction technology can be applied to any programming or design language and has potential to impact many aspects of the software engineering life cycle. To better understand this impact, STAR*Lab conducted an SEI-sponsored study with a major corporation to determine how FX could improve engineering operations in activities ranging from software specification and design to implementation and testing [4]. This study produced guidance for FX evolution from experienced software developers, including the following recommendations:

- Development of FX automation for assembly language should be a priority.
- FX automation should be developed for correctness verification of software.
- FX automation should be developed for high-level languages starting with Java.
- Research on FX automation for specification and architecture should be initiated.

Based on the findings of this study and additional recommendations from sponsors and stakeholders, STAR*Lab has identified the project areas described on the next several pages.

**Expected Benefits**

The Function Extraction for Malicious Code (FX/MC) system, an IDA Pro plug-in that operates on Intel Assembly Language, is expected to help security 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.

In terms of the broader application of the Function Extraction for Intel Assembly Language (FX/ASM) system, a stand-alone version, opportunities exist to make real inroads on the problems of malicious code detection, computational security analysis, correctness verification, legacy system understanding, creation of assured software repositories, and automated component composition. The basis for all of this is the realization that programs are mathematical artifacts subject to mathematical analysis. Human fallibility may still exist in interpreting the analytical results, but there can be little doubt that routine availability of calculated behavior would help reduce errors, vulnerabilities, and malicious code in software and make software development more manageable and predictable.

**2006 Accomplishments**

As a result of work accomplished in 2006, the FX/MC and FX/ASM systems now demonstrate transformation of spaghetti-logic assembly language programs into understandable structured form, and automated computation of behavior for sequence and alternation structures. Work on theoretical foundations and implementation of loop behavior computation is well underway.

**2007 Plans**

FX system development will continue in 2007, and additional sponsors are welcome to participate in moving the technology forward. Specific areas of system evolution have been identified and incorporated into project plans. STAR*Lab is also ready to apply FX to additional languages and phases of the software engineering life cycle.

**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, NY: John Wiley & Sons, 2002.

[3] 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.

[4] 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.

[5] 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). Waikoloa, HI, Jan. 5-8, 2004. Los Alamitos, CA: IEEE Computer Society Press, 2004.

[6] Walton, G.; Longstaff, T.; & Linger, R., *Technology Foundations for Computational Evaluation of Software Security Attributes* (CMU/SEI-2006-TR-021). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/publications/documents/06.reports/06tr021.html.

# **STAR*Lab** Computational Security Attributes: Engineering Automation for Software Security Analysis

Principal Investigator: Gwendolyn H. Walton

**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 often assessed through error-prone, labor-intensive human evaluation. The result is imprecise and a priori analysis that can be of limited value in the dynamics of system operation where threat environments and security attributes can change quickly. This STAR*Lab IRAD (Internal Research and Development) 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 project 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 STAR*Lab's new function extraction (FX) technology, unavailable to previous researchers, provides the basis for this critical first step by deriving the functional behavior of programs as a starting point for security analysis. 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.

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.

The properties analyzed in the study include authentication, authorization, non-repudiation, confidentiality, privacy, and integrity.

**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 specification of security attributes of software systems before they are built, specification and evaluation of security attributes of acquired software, verification of as-built security attributes of software systems, and real-time evaluation of security attributes during system operation.

**2006 Accomplishments**

The project was successfully completed with the publication of research results that define the required computational foundations for analysis of security properties [2].

**2007 Plans**

The findings of this study define requirements for prototype automation to compute software security properties. Interested organizations are invited to sponsor development of engineering tools for this purpose.

**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.

[2] Walton, G.; Longstaff, T.; & Linger, R. *Technology Foundations for Computational Evaluation of Software Security Attributes* (CMU/SEI-2006-TR-021). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/publications/documents/06.reports/06tr021.html.

# **STAR*Lab** Automated Correctness Verification: Engineering Automation for Software Assurance

Principal Investigator: Mark Pleszkoch

### 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 sophisticated computational automation 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 of this potential project is to develop an operational prototype of a function verification (FV) system that will help users to check the correctness of programs based on their actual computed behavior. The system will employ the mathematics-based foundations of function extraction 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 function 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 to satisfy intended behavior and execute the FV system to check its 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. As noted, the function-theoretic model of software treats programs as rules for mathematical functions or relations—that is, mappings from domains to ranges. 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. A theorem defines the mapping of these control structures into procedure-free functional form [1, 2]. These mappings are the starting point for the function extraction process and its application to correctness verification.

### Expected Benefits

Large-scale software assurance requires a commensurate scale of engineering automation. This project can provide substantial benefits to STAR*Lab sponsors who must deal with software failures and vulnerabilities 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 can 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 [3].

### 2006 Accomplishments

The function extraction system currently under development in STAR*Lab provides a foundation for implementing correctness verification capabilities.

### 2007 Plans

STAR*Lab is ready to extend FX technology for automation of correctness verification 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, NY: 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: Engineering Automation for Understanding System Behavior

Principal Investigator: Kirk Sayre

### 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 require 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 [1].

### Research Approach

This potential 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. Because behavior calculation is essentially a compositional task, function extraction (FX) is the key underlying technology for component composition. FX produces behavior catalogs of individual programs; the catalogs themselves can be composed to reveal the composite behavior of the programs when combined into systems.

### Expected Benefits

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

### 2006 Accomplishments

Research and development carried out in the FX project in 2006 has direct applicability to automated composition of components.

### 2007 Plans

Creating function extractors to compose software components would provide automation support to help construct and integrate entire systems. Research in FX technology has provided initial foundations for component composition. A key step toward creation of an automated composition capability is extension of FX technology to create a proof-of-concept prototype. Sponsors are welcome to join in this effort.

### Reference

[1] Feiler, Peter; Goodenough, John; Linger, Richard; Longstaff, Tom; Kazman, Rick; Klein, Mark; Northrop, Linda; Wallnau, Kurt; Gabriel, Richard; Schmidt, Doug; & Sullivan, Kevin. *Ultra-Large-Scale Systems: The Software Challenge of the Future.* Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, June, 2006. http://www.sei.cmu.edu/uls.

# **STAR*Lab** Flow-Service-Quality (FSQ) Engineering: Foundations for Developing Network-Centric Systems

Principal Investigator: Stacy Prowell

### 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 networked systems: highly distributed heterogeneous components, shifting boundaries and users, uncertain commercial off-the-shelf component 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:

1. **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.
2. **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.
3. **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 that accomplish enterprise missions. They employ mathematical semantics that permit 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 FSQ 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 operations 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.

Flow structures 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. They can be expressed in simple control structures including sequence, alternation, and iteration, and they can be refined, abstracted, and verified with precision. Flows invoke services, which can be refined into flows, and so forth, 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.

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. The Automated Component Composition project described earlier will provide support for flow structure analysis.

**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 earlier extended and amplified 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 networked systems as a standard engineering practice. For example, flow specification requires definition of appropriate actions by a flow for all possible responses, both desired and undesired, from the services it employs. 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 networked systems and offers important advantages. It requires for mission survivability that the uncertainty factors be dealt with explicitly in specification, design, and 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 structures despite the underlying asynchronous behavior of their constituent services. These structures can be refined, abstracted, and verified using straightforward compositional methods for human understanding and analysis [4].

#### 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. Computational quality attributes permit automated reactions to dynamically changing quality values in system execution. In addition, flow management architectures provide systematic frameworks for managing flows and quality attributes in operation.

#### 2006 Accomplishments

Work continued on relating FSQ engineering to web services and service-oriented architectures.

#### 2007 Plans

STAR*Lab is interested in continued development and application of FSQ engineering methods for large-scale networked systems. Interested organizations are invited to participate in creation of a proof-of-concept prototype and associated engineering practices.

#### 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).* Waikoloa, HI, Jan. 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.

# 8
# **SAF:** Survivable Analysis Framework


**Robert Ellison**
Principal Investigator
412-268-7705

**Carol Woody**
Principal Investigator
412-268-9137

# **SAF:** Survivable Analysis Framework

### Problem Addressed

Large systems and particularly systems of systems raise the importance of complexity management. Complexity is an aggregate of technology, scale, scope, and operational and organizational issues. While small-system security may have been implemented by a set of point solutions that mitigated specific threats, the mitigation of threats of the magnitude and diversity of those associated with large distributed systems of systems requires foundational support.

Separation of concerns is a powerful tactic for managing complexity. A software architect may try to maintain separation among security, performance, reliability, and other system quality attributes. We frequently have maintained separation among system operations, systems development, and business operations, but that separation was often reflected by the expression "toss it over the wall." Business integration requirements and the appearance of technologies such as Web services to support that integration for distributed systems challenge these traditional separations. It is now not unusual to find that an organization's development, operational, and business groups are tackling common problems with little coordination, or that some problems, often aspects of security, are falling between the cracks and have been ignored.

Complexity's negative effect on survivability renders its management all the more critical. We define *survivability* as the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures, or accidents. Survivability initially involves the availability aspects of security, but also incorporates confidentiality, integrity, and reliability considerations. Availability applies to the specific functions and services needed to satisfy a specific mission. Increasingly those services span multiple systems. Survivability efforts concentrate on the organizational activity supported by software systems rather than on the individual systems. Current survivability analysis mechanisms have the following limitations:

- a focus on survivability with respect to a single system
- consideration of only single-system architecture tradeoff decisions
- a lack of consideration of mission dependencies on multiple systems
- an inability to analyze components to identify the relationship of a system's survivability and performance to a joint mission

The objective of this research is to develop survivability analysis methods that are applicable to systems of systems to address the challenge of increased demands for interoperability and integration.

### Research Approach

Increasingly essential work processes span multiple systems that are geographically distributed and independently managed. The individual systems are useful in their own right, that is, the services are general purpose. The business demands for adaptability result in a mix of systems and work processes that are constantly changing. Development is evolutionary as functions and purposes are added, removed and modified; thus we have what Maier characterizes as a system of systems [1]. Metrics for the success of these work processes are end-to-end measures. We shall see why assuring such success in a system of systems is problematic and how this threatens survivability.

Consider Figure 1 where the ovals on the left side represent geographically distributed systems and the blue and black lines are business processes that use those systems. The right side of the figure expands one of those systems. For a military example, an oval might be a specific Service system whereas the work process might be joint activity that requires coordination across the Services. The specific Service system receives both joint and Service-specific requests. A joint Service activity would likely generate a sequence of actions similar to the actions generated for a Service-specific request.

We need to take two perspectives in analyzing Figure 1: the end-to-end work process view and the individual systems view. Usage is critical for the system participants. A work process, especially in a systems-of-systems environment, could create usage patterns that were not anticipated in the design of a specific system and hence could adversely affect the operation of that system. An important objective for the risk mitigation strategies for a specific system may be the protection of local resources, particularly when usage patterns are not predictable. The design for such systems has to consider abnormal behavior.

The success of the end-to-end work process depends on the successful composition of the individual process steps and an acceptable completion. The key relationship for the work process is in the services and the associated systems that are required to successfully complete that work process. We would like to assure the end-to-end behavior of a work process, but the limited access to controls for the member systems can limit the level of assurance that is achievable. The work process thread must be analyzed end to end and step by step to identify events that could lead to failure to successfully complete the process. Accomplishing this requires the following detailed process thread information: a description of work process success, expected work process quality attributes such as performance and reliability, and scenarios of both expected and unacceptable behavior. The latter includes the kinds of things that may go wrong and what will happen should they occur.

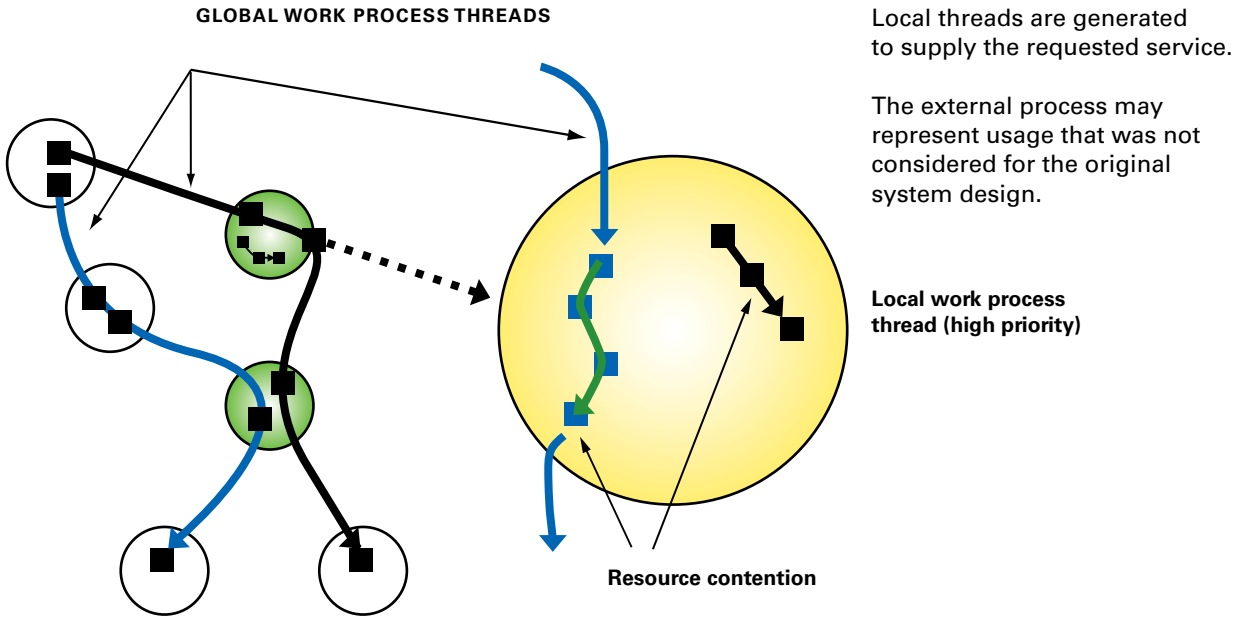


*Figure 1: System-of-Systems Resource Contention*

In addition, each work process to be analyzed must be decomposed into required steps with the following types of information about each step: roles in the process, pre-conditions, functions, post-conditions, constraints, and dependencies. Each step may be composed of multiple components (human, software, system, and/or hardware) acting independently or in a coordinated manner.

Systems and systems of systems can create failure states that are difficult to resolve. Historically, system failure analysis has sought to identify a single root cause, but for software intensive systems that involve human interactions a failure may be the result of multiple software, hardware, or human errors where each error when considered individually would be perceived as minor. Other failures may arise because of emergent behavior. Each system behaves as specified, but the collective behavior is unacceptable. For example, feedback among systems might generate unexpected resource contention. At this stage of our research we consider the stresses that might be induced by a work process thread. We initially focus on the interactions among the systems that participate in that thread and the stresses that might be induced by those interactions on the supporting systems. The stress types include

- interaction (data): missing, inconsistent, incorrect, unexpected, incomplete, unintelligible, out of date, duplicate
- resource: insufficient, unavailable, excessive, latency, inappropriate, interrupted
- people: information overload, analysis paralysis, fog of war, distraction (rubbernecking), selective focus (only looking for information for positive reinforcement), diffusion of responsibility, spurious correlations

The scenarios of potential problems, especially those with anticipated high impact, will be used to potentially limit the areas of each stress type to a subset of high-interest issues for the mission thread stakeholders. For each type of stress, the analysis framework will be applied to identify what is currently in place, what should be in place, and expected step and/or component behavior, should survivability be affected. The analysis framework will be applied at a specific point in time to a selected sample mission thread. In order to analyze the change in risk over time, an assessment is needed for the existing work process to establish a baseline of current risk.

Survivability involves what can go wrong. The issues considered by the SAF analysis are shown in Figure 2.

### Expected Benefits

The expansion of the scope and scale of systems induce new stresses. An objective of the initial phase of this project is to identify indicators of stress that may lead to system failures. Indicators that are appropriate to the early life-cycle phases of software development help to change current practice, whereas software failure analysis typically concentrates on the errors that are derived from testing. The goal is to generate a sufficient number of examples so that patterns emerge. A pattern, for example, may represent ways to reduce complexity by consolidating risk mitigations.

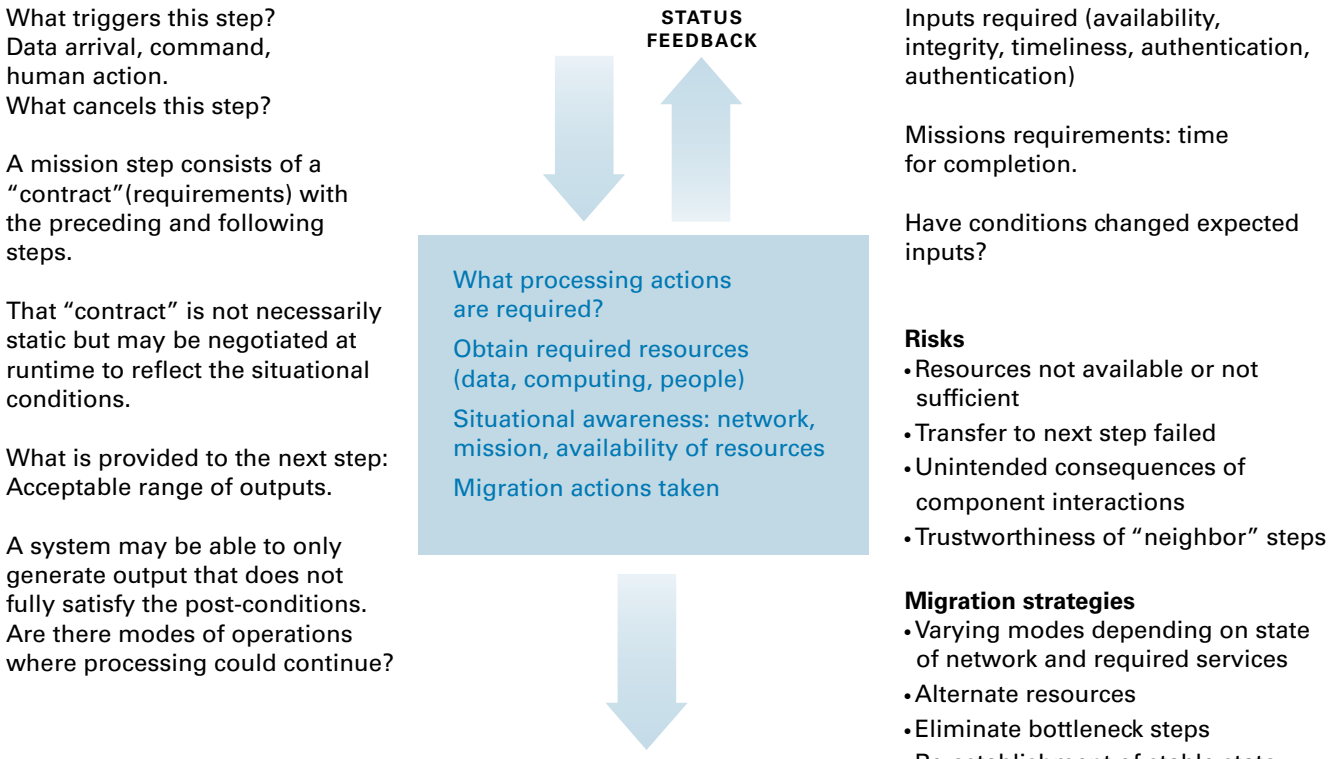


*Figure 2: SAF Analysis*

The survivability analysis framework with its emphasis on business process threads also enables better traceability between technology risks and business work processes. It may also enable better traceability of the design decisions to the requirements of multiple organizational levels.

**2006 Accomplishments**

The SAF was initially applied in two pilot applications in 2006. Both pilots involved applications to DoD systems. One pilot was relatively high level and focused on an improved definition of the problem and general design and acquisition guidance. The second pilot involved a more detailed evaluation of a DoD system. In addition, we developed a white paper describing the system-of-systems problem space for survivability and a tutorial based on a medical work process example.

**2007 Plans**

The general objective for 2007 is to apply SAF in more pilots. One specific objective is to increase the number of SAF work process examples available for analysis, and a second is to find examples that stress the SAF approach and suggest necessary expansion and changes. We also seek non-DoD SAF pilots projects. At least one workshop is planned for expert external review of SAF.

The complexity of systems of systems is reflected in the diversity of mitigation strategies that might be applied. Assurance cases may be a way to develop a strategy for how to demonstrate survivability given that complexity and diversity. Assurance cases have been used for safety. There is growing interest in using assurance cases for other system attributes, and we are considering them within the context of this research. An assurance case is a body of evidence organized into an argument demonstrating that some claim about a system holds. The argument for a simple assurance case might be that the claim holds if three subclaims hold. In addition to the organizing the argument in terms of subclaims, an assurance case also identifies the evidence that supports the claims. For example, evidence for a security claim for the lack of coding vulnerabilities could be provided by static analysis tools.

**References**

[1] Maier, Mark W. "Architecting Principles for Systems of Systems." *Systems Engineering 1*: 267-284, Hoboken, NJ: John Wiley & Sons, Inc., 1998. http://www.infoed.com/Open/PAPERS/systems.htm.

# 9 The **Uncleanliness Vector:** Histories of Hostile Activity


**Michael Collins**
Principal Investigator
412-268-9124

**Markus De Shon**

**Timothy Shimeall**

**Jeff Janies**

**Rhiannon Weaver**

**Sidney Faber**

**John Prevost**

# The **Uncleanliness Vector:** Histories of Hostile Activity

### Problem Addressed

In an operational network security analysis environment, there is little time for gathering context information on attackers. Analysts need tools that present background information in an easily digestible way.

The Uncleanliness Vector (UV) project is an attempt to provide context information on external networks to security analysts that indicates whether the network in question has engaged in other malicious activity and, if so, what kind.

### Technical Approach

The hypothesis behind the UV approach is that an attacker will take over as many machines as possible to serve as hosts for further attacks. Therefore the likelihood that a particular IP address is so used for hosting an attack (such as a scan or distributed denial-of-service attack) is a function of the security administration of the network that address resides on. A network with a strong security policy and rigorous maintenance will identify and remove compromised hosts, while one that does not demonstrate this vigilance will remove those hosts more slowly and see the same hosts repeatedly compromised.

The UV describes the likelihood of attack from a block of IP addresses using an uncleanliness score of 0 to 100; a network with a value of 0 is considered clean, while a network with an rating of 100 is considered totally unclean.

The UV is composed of several distinct uncleanliness elements, each of which was assigned an uncleanliness value. We characterized this set of numbers as a vector, and the aggregate score as a weighted average of the individual elements. The weights indicate a prioritization of the uncleanliness elements by their relative threat level to the monitored network.

The vector elements and their relative weights are as follows.

**Spyware site hosting** (weight 6). Spyware is software installed on a client machine, generally without the user's knowledge, that gathers information and reports that information back to the spyware site. The information gathered can range from host configuration data, to browsing history, to personally identifiable information (PII). A spyware site host is generally controlled by the entity that created the spyware.

**Phishing site hosting** (weight 5). Phishing attacks attempt to gather login credentials, financial information, or PII for the purpose of financial fraud or identity theft. Phishing sites are generally hosted on compromised machines, and the gathered information is reported back to email drops or other data gathering points.

**Botnet command-and-control server hosting** (weight 4). *Botnets* are networks of remotely controlled programs that allow an attacker to perform various actions such as denial-of-service attacks, spamming, and anonymous proxying. Botnet command-and-control servers are sites that the bots "call home" in order to receive commands. These traditionally are Internet Relay Chat (IRC) servers, but can include almost any type of service—one common alternative is web servers, where the bot regularly polls a URL to see if any commands have been issued. Botnet command-and-control server networks are not usually owned by the attacker, and thus either represent a compromised host or a service piggybacking on a legitimate server (such as an IRC server in both legitimate and illegitimate use).

**Botnet members installed** (weight 3). Hosts on which botnet members (the "bots" themselves) are installed are most likely compromised hosts.

**Open proxy hosting** (weight 3). Anonymous proxy networks could either be hosted 1) deliberately by the network owner, 2) unknowingly as the result of a compromised host on which a proxy application was installed, or 3) unknowingly through a misconfigured web server or internal proxy. Attacker-installed proxies could be "bot" programs, or simply specialized proxy software.

**Spamming** (weight 2). Hosts sending spam are usually compromised hosts, because the success of blacklists and other anti-spam measures has meant that dedicated spam hosting is no longer efficient.

**Scanning** (weight 1). Scanning is reconnaissance activity by attackers attempting to determine what services are available on a host. This involves probing ports on the host to determine which are open. Such probing could be purely investigatory or could be combined with an attack. Scanning usually originates from compromised hosts.

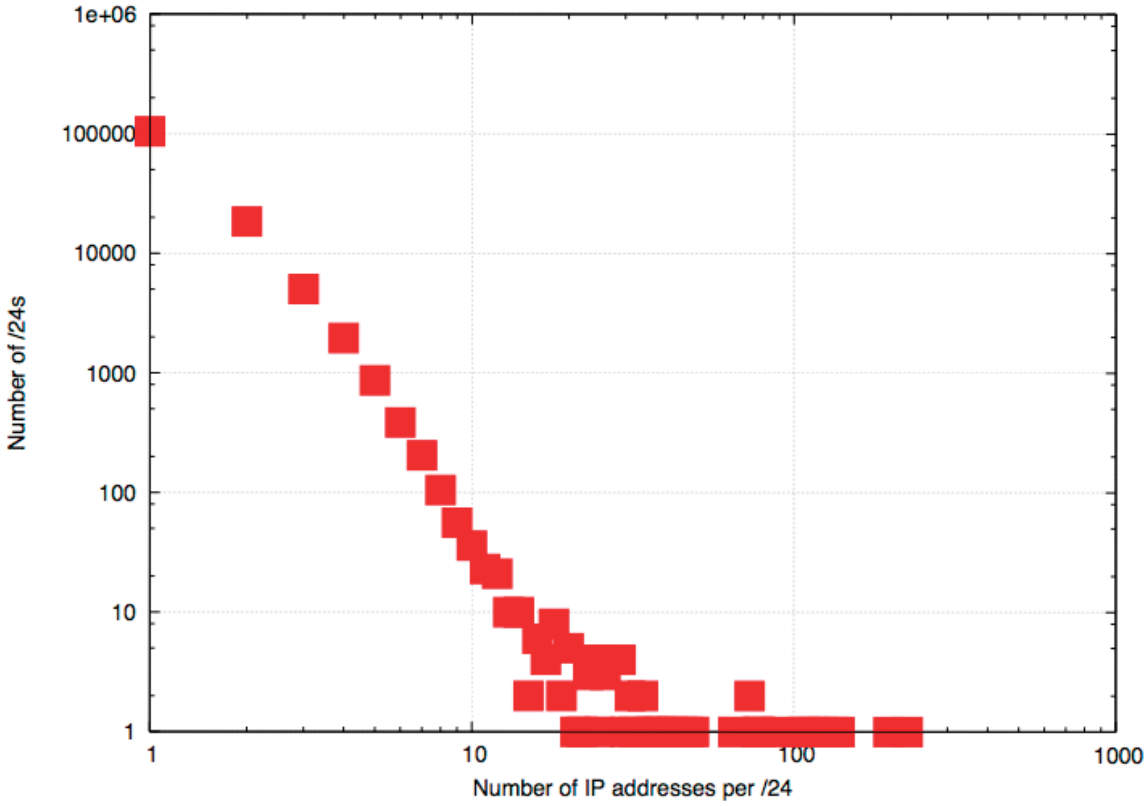


*Figure 1: Number of Networks Showing a Particular Number of Unique Scanning IP Addresses (log-log scale)*

**2006 Accomplishments**

Early in our analysis, we established that malicious activity is indeed distributed *heterogeneously* across the network. If malicious activity were distributed *homogeneously*, then the amount of the activity would depend only on the number of hosts on a network. The most complete data we had available for any of the UV elements was scanning data, so we analyzed how scanning was distributed across the Internet. We found that the scanning is concentrated, coming from only about 1% of networks, and that the amount of scanning does not depend strongly on the size or population of the network.

Figure 1 shows the distribution of scanners, rendered as a log-log scale plot. The x-axis indicates the number of IP addresses in a particular /24 network (class C network) that were observed scanning. The y-axis indicates the number of /24 networks that showed the particular number of unique IP's scanning shown on the x-axis. Note that the majority of networks have only one host scanning. The chance of having two or more hosts scanning is nearly 10 times lower. The chance of having 10 or more hosts scanning is nearly 10,000 times lower. This type of distribution is called a "power law" distribution, and its presence here implies that scanning is strongly inhomogeneous—there are a few networks that are heavy sources of scanning with most networks tending to be clean.

Different types of malicious activity are related. For example, when we compared data for bots and scanning, networks where bots were reported also showed significant increases in scanning of the monitored networks. Figure 2 shows the time behavior of scanning observed from reported botnets (red line) versus all networks (green line). The observed scanning from botnet networks was much higher than the expected scanning (dotted line). Interestingly, scanning started several weeks before the botnets were reported, implying that the bots went undetected for some time.

We needed to translate the data on scanning, spamming, and other activity into some measure of "uncleanliness" of a network as a score on the scale 0-100. During our investigations, it became clear that this measure should not be related to the volume of malicious activity observed. In some cases we did not have access to data on volume of activity. In other cases it was clear that such data was not very meaningful.

The quantity that instead appeared to provide the most useful measure of the uncleanliness of a network was the *number of unique hosts* (i.e., hosts with unique IP addresses) that engaged in the hostile activity. This is because in most cases, the vector elements were really measuring the likelihood that a host on that network could be compromised. If a network has a large number of compromised hosts, then the network is more likely to be in use by an attacker.

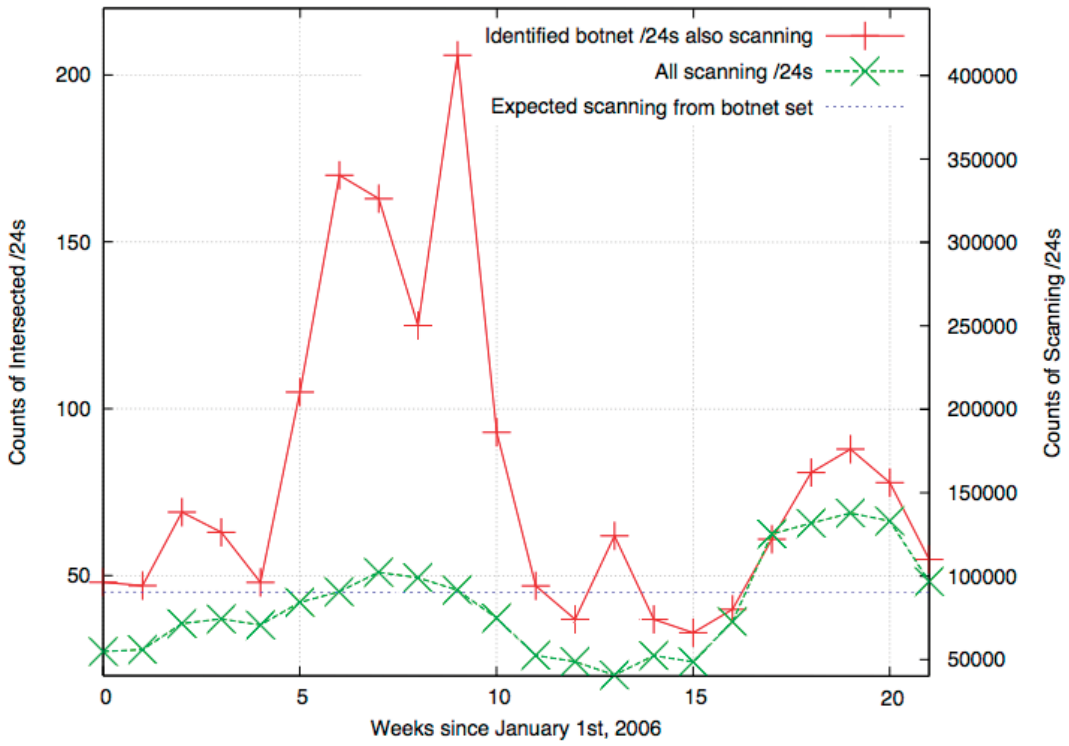


*Figure 2: Comparison of Scanning by Bots (red line) and Overall Scanning (green line) Over Time*

Figure 1 and similar figures for the other vector elements show that malicious activity is distributed sparsely, and that there are relatively few networks where there are many malicious hosts. This observation tended to support our plan of using unique hosts counts as the measure of uncleanliness.

For each vector element, we chose two thresholds ($s_0$, $s_1$) on the axis of unique host counts (x). The uncleanliness value was then chosen according to the equation:

$$\text{Uncleanliness}\ (x, s_0, s_1) = \begin{cases} 0 & \text{if } x = 0 \\ 10 & \text{if } 0 < x < s_0 \\ 10 + 90\ \dfrac{x - s_0}{s_1 - s_0} & \text{if } s_0 \le x \le s_1 \\ 100 & \text{if } x > s_1 \end{cases}$$

*(Eq. 1)*

In other words, uncleanliness is 0 if no hosts are unclean, 10 for any number of hosts greater than zero but below the low threshold, linearly interpolated from 10 to 100 between the thresholds, and 100 if the number of unclean hosts meets or exceeds the high threshold.

Based upon analysis of the data we used, we found the thresholds shown in Table 1 to be appropriate for our input data (these values can of course be changed based on the patterns observed in the actual input data used).

| Element | Low threshold (s0) | High threshold (s1) |
|---|---|---|
| Spyware | 8 | 192 |
| Phishing | 2 | 7 |
| Botnet C&C | 0 | 2 |
| Bot Members | 2 | 8 |
| Proxy hosting | 2 | 32 |
| Spamming | 1 | 12 |
| Scanning | 1 | 12 |

*Table 1: Thresholds Used for Calculating Uncleanliness Scores*

The individual elements are then combined into an overall uncleanliness score as a weighted average, that is,

$$\frac{\sum_i \beta_i U_i}{\sum_i \beta_i}$$

*(Eq. 2)*

where $U_i$ is the value of uncleanliness vector element i, and $\beta_i$ is the weight of that vector element, as given in the listing of vector elements above.

The data sources for each of the vector elements differed. In some cases, we found that publicly available blacklist information was useful (proxies, and the initial version of the spamming element), in other

cases we had access to large volumes of actual traffic and were able to identify the malicious activity (scanning and later versions of spamming), and in other cases we processed incident report information (botnets and botnet C&C). Of particular interest from a research point of view are the methods we have developed for identifying scanners and spammers. Scanner identification was performed using methodology discussed in the 2005 CERT Research Report, and explained in detail in a CMU/SEI technical report published this year [1].

The spammer identification was developed in the course of this research. We observed, using network flow data [2], that spam traffic had certain characteristics that were significantly different from the traffic observed from known good email servers. In particular, spam was not accompanied by Domain Name System queries, and the set of hosts with which spammers communicated were highly dynamic. We were able to construct a spam identification process using these heuristics, which now feeds our spam uncleanliness measurements.

While constructing measures of uncleanliness of the public Internet, it was natural to think about the reflected image of uncleanliness: how much has the external uncleanliness affected the internal network, leading to what one might call "corruption?" This led us to define a *corruption vector* as a complement to the uncleanliness vector.

For certain of the vector elements, communicating with unclean networks is not necessarily a good indicator of corruption (e.g., external hosts are constantly scanning; this does not automatically lead to corruption of the scanned hosts). For other vector elements, communication with an external unclean host is in fact an indicator of internal corruption. The latter is true for proxies (internal hosts attempting to anonymize activity), botnet command-and-control (possible bots on the internal network), and spyware (probable spyware installed on an internal host).

The measure of corruption was (similarly to uncleanliness) based on the *number of unique hosts* seen communicating with unclean external hosts. However, significant challenges in counting unique hosts were encountered, due to the presence of internal gateways (Network Address Translation devices or proxies). When a spyware-infected host, for example, is located behind a gateway, the gateway's IP appears to be infected with spyware, and it may be one of the few active IP addresses (or is perhaps the only active IP) on that network block. The result is a very high corruption score, when in reality the activity only represents a small number of the hosts behind the gateway.

The output of this research included software tools, primarily a daemon for querying a running uncleanliness database. The daemon makes it easy to integrate the uncleanliness vector into an existing operational environment. A UV server need only be set up somewhere on the network, and the daemon can be queried by the analyst interface server to retrieve information about a particular IP address. Since the granularity of the data is at the /24 level, a query for any IP in that range of addresses will yield the same result.

Additional tools make it simple for an operational environment to supplement UV with additional context data, including, for example, internal incident data. UV is a framework for simplifying large amounts of diverse data into a straightforward measure for analysts, and therefore we endeavored to make it as easy as possible to extend and include other data sources.

**2007 Plans**

We are investigating methods for better measuring the corruption of networks behind Network Address Translation (NAT) gateways and proxies. These will likely be based on traffic models that will estimate the number of hosts (and the number of infected hosts) based on traffic volumes of various types. For a sufficiently large number of NATed hosts, it should be possible to apply models of the average user based on observed web browsing and email activity, for example. Since the essential measure is actually the percentage of infected hosts, it may be possible to extract that information without actually estimating the number of hosts.

While examining various independent phishing data sources, it was observed that small overlaps exist in the data. We are in communication with the various data providers to determine the degree of independence of these data sources, and are attempting to construct statistical models (based on capture-recapture models used for measuring wild animal populations) to estimate the total population of phishing sites, both detected and undetected. Complications to constructing the model include that phishing reports tend to lead to their being shut down (referred to as the "trap death" problem in capture-recapture modeling), that some phishing data is not truly independent, and that some reporting biases exist in the data under study. Nevertheless, we are confident that such an estimate is possible and desirable.

**References**

[1] Gates, C.; McNutt, J.; Kadane, J.; & Kellner, M. *Detecting Scans at the ISP Level* (CMU/SEI-2006-TR-005). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/publications/documents/06.reports/06tr005.html.

[2] Gates, C.; Collins, M.; Duggan, M.; Kompanek, A.; & Thomas, M. "More Netflow Tools: For Performance and Security." *Proceedings of LISA '04: Eighteenth Systems Administration Conference.* Atlanta, GA, Nov. 2004. Berkeley, CA: USENIX Association, 2004. http://www.cert.org/netsa/publications/Lisa2004-gates-netflow.pdf.

## Additional Research Activities




**SEI Customer Relations**
412-268-5800

## 10
## Backdoor Analysis Using Network Flow

Principal Investigator: Timothy Shimeall

A *routing backdoor* is an incoming or outgoing path through a network that does not pass through any of the designated border or gateway routers for that network. These backdoors are of concern since they avoid security and performance monitoring and may circumvent other security controls. Our research is an analysis of traffic as summarized by sensors attached to a large organization's border routers, using the SiLK tool suite [1].[1] The analysis uses a set of identified and registered organizational addresses. The analysis identifies and profiles sources of traffic that is anomalous in any of three specific ways:

1. inversion of internal/external address with respect to flow direction
2. indication of non-organization traffic traversing the organizational network
3. organization addresses sensed in one direction only in non-scan traffic

The analysis process detects these separately in incoming and outgoing flows, yielding the six groups of flows diagrammed in the figure below.

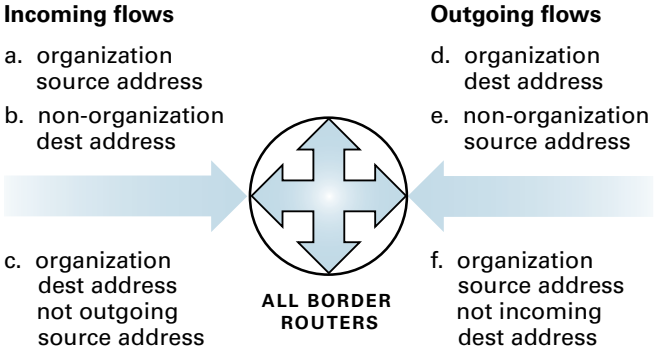


*Figure 1: Six Flow Groups Detected by Analysis*

The results characterize unintended network activity and topological anomalies in the organization's network.

**References**

[1] 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*. Atlanta, Georgia, Nov. 14-19, 2004. Berkeley, CA: USENIX Association, 2004.

1 SiLK, the System for Internet-Level Knowledge, is a collection of traffic analysis tools developed by the CERT Network Situational Awareness Group to facilitate security analysis of large networks.

## 11
## Construction and Verification of a Metamodel for Valuation of Software Assurance in Sliding Scale Applications

Principal Investigator: Nancy Mead

Collaborators:
Professor Daniel Shoemaker, University of Detroit, Mercy
Professor Antonio Drommi, University of Detroit, Mercy
Jeffrey Ingalsbe, Ford Motor Company
John Bailey, IDA

There are currently no standard methods for justifying investment in software security across the software life cycle. There are many existing models for estimation of IT costs. However, none of these models are specific to software assurance cost/benefit. Software cost estimation models such as COCOMO II have security extensions that look at cost estimation for software systems but are not yet focused on actual costs or the associated return on investment [1, 2].

Those organizations that do invest in software assurance typically do so because of an overwhelming concern, such as loss of life, or to mitigate risk of financial disaster. As a consequence, they are not concerned about cost/benefit calculations, because they *have* to make the investment, regardless of the cost. The Space Shuttle is a good example of this. The Federal Aviation Association would also fit this model. We know that there are many organizations that should be investing in software assurance but are not. For example, many critical infrastructure systems are exposed. Manufacturers, software vendors, retailers, and professional service organizations, among others, are reluctant to spend more money in this area. We all know that executives will not stand for investing in improvement solely because it is the "right" thing to do, be it in software assurance or other areas.

Our research objective is to identify or develop a model that can be used to consistently document the costs and benefits associated with software assurance. This will allow us to build a database of case studies and hopefully give staff members and managers the needed ammunition to convince executives that investment in software assurance is worthwhile and will in fact benefit them in the long term.

Through an initial literature search, we identified 14 economic models that can be used to obtain IT value estimates. These models can be factored into four general approaches to valuation: investment oriented, cost oriented, environmental/contextual oriented, and quantitatively oriented.

We will justify and build a metamodel that encompasses the relevant common elements of these 14 models. Once that model is built and rationalized, we will conduct a one-year multitrack parallel study in commercial application that will compare the outcomes of the metamodel against results obtained for each of the source models.

The goal of the research is to validate that the variables and application methods embodied in the model can accurately characterize the economic value of software assurance and that this classification technique can be generally applied. The intention is to longitudinally gather and validate empirical data that will allow organizations to make decisions about the value of software assurance with a reasonable degree of confidence. We expect this to lead to the resolution of the following research questions:

- Is there a common empirical approach to valuation of software assurance?
- What are the correct elements of a software assurance valuation model?
- Is the value of assurance influenced by various levels of organizational maturity?
- Is the value of assurance influenced by various levels of sensitivity?

The primary deliverable of this project is a report that documents the metamodel, the projects studied, and the results of the study. We should also be able to develop a conference or journal paper if we have good results. Success criteria are our ability to develop such a model, obtaining actual data, and getting experimental results. Another measure of success is that organizations are willing to continue to apply the model in-house, with or without our help, and provide us with the results. The ultimate goal is that organizations will be able to assess the cost/benefit associated with software assurance and become willing to increase their investment in software assurance best practices.

If this project is successful, further work could take place under the umbrella of the Business Case Factors Team, and a data repository of cost/benefit data for software assurance could be established.

**References**

[1] University of Southern California. *COCOMO*. http://sunset.usc.edu/research/COCOMOII (1995-2002).

[2] Colbert, Ed; Don Reifer; & Murali Gangadharan. "COCOMO II Security Extensions," *17th International Forum on COCOMO and Software Cost Modeling.* Los Angeles, CA, Oct. 22-25, 2002. Los Angeles, CA: University of Southern California, Center for Software Engineering. Available through http://sunset.usc.edu/events/2002/cocomo17/agenda_cocomo17.html#cocomoII (2002).

## 12
## Cyber Security of the Electric Power Grid for Improved Infrastructure Survivability

Principal Investigator: Howard Lipson

The Internet and its enabling technologies are playing an increasing role in electric power systems, improving the efficiency and sophistication of business and technical operations. However, Internet connectivity also introduces significant new risks to the power grid's automated control systems and hence to the power grid itself. Howard Lipson of the CERT Survivable Systems Engineering team continued as co-principal investigator on a National Science Foundation (NSF) 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 Carnegie Mellon's Electricity Industry Center and the Washington State University School of Electrical Engineering and Computer Science. Ongoing research includes investigating cyber risks to the electric power grid at the architectural level along with technical and policy approaches for enhancing the power grid's survivability in the face of cyber attacks.

In related activities, Lipson gave an invited tele-seminar, *Cyber Security and Control System Survivability: Technical and Policy Challenges,* for the Power Systems Engineering Research Center (PSERC)[1] and was acknowledged by the International Risk Governance Council (IRGC – Geneva, Switzerland) for contributing substantial review comments for their document *Managing and Reducing Social Vulnerabilities from Coupled Critical Infrastructures.*[2]

1 http://www.pserc.org/ecow/get/generalinf/presentati/psercsemin1/2psercsemin/ (05-10)

2 http://www.irgc.org/irgc/projects/critical_infrastructure/

## 13
## Developing a Worm Calendar

Principal Investigator: Michael Collins

While a variety of worm defenses and identification mechanisms have been developed by the research community, relatively little research has been done on the current state of the practice in worm development. To support our work on analyzing worm trends, the CERT Network Situational Awareness Group has been developing a corpus called the Worm Calendar.

The Worm Calendar is an application that provides a structured schema for worm information. In comparison to existing repositories, the Worm Calendar formalizes many attributes (notably the methods of propagation) that are otherwise handled using natural language.

The Worm Calendar provides support for both operational facilities and research. For operational analysts, the Calendar can be used to identify how much "noise" in the network is the result of recent worm developments. For researchers, it provides an inventory of existing worms and rapid access to worms, with data on their methods of propagation, exploitation, and design family.

## 14
## Development and Implementation of the IPFIX Flow Export Protocol Standard

Principal Investigator: Brian Trammell

A primary goal of the CERT Network Situational Awareness Group (NetSA) is to enhance security incident detection and response at the network level though the sharing of security-relevant network measurement data across organizational boundaries. One significant impediment to this effort is the wide variety of methods for representing and transporting this data. To that end, NetSA supports standardization efforts in this area.

The IP Flow Information Export Protocol (IPFIX) Working Group at the Internet Engineering Task Force is defining a standard protocol for the export and representation of IP flow information.[1] The working group membership is international and includes representatives from router and meter vendors, telecommunications operators, and network measurement and security research groups. The standard defined by this group is expected to displace the variety of disparate methods for representing and communicating information about flows.

The contribution of NetSA to this effort is threefold.

First, we have contributed to the development of the protocol document into a draft standard.

Second, we are actively working to enhance the applicability of the standard to the research problems we face at NetSA. The addition of a method to represent bidirectional flow data was substantially completed in 2006 [2][1] and is expected to be published as a Request for Comments in 2007. Future work includes a standard for extending the IPFIX message format as the basis for long-term storage of flow data,[3] expected to be completed in 2007.

Third, the NetSA Engineering Team is building and maintaining a reference implementation of the IPFIX standard; libfixbuf is available under an open source license from http://tools.netsa.cert.org and successfully performed in two interoperability tests in March and November, 2006.

**References**

[1] Boschi, E. & Trammell, B. "Bidirectional Flow Measurement, IPFIX, and Security Analysis." *Proceedings of FloCon 2006*. Vancouver, WA, Oct. 10-12, 2006. http://www.cert.org/flocon/2006/proceedings.html.

1 Claise, B., ed. "Specification of the IPFIX Protocol for the Exchange of IP Traffic Flow Information." INTERNET- DRAFT draft-ietf-ipfix-protocol-24, work in progress, Oct. 2006.

2 Trammell, B. and Boschi, E. "Bidirectional Flow Export using IPFIX." INTERNET-DRAFT draft-ietf-ipfix- biflow-00, work in progress, Aug. 2006.

3 Trammell, B.; Boschi, E.; Mark, L.; & Zseby, T. "Requirements for a Standardized Flow Storage Solution." To be published.

# 15
# Distributed Security Testing

Principal Investigator: Timothy Shimeall

Modern network defenses operate network wide, with a variety of controls supplied by host-based, perimeter-based, and interior-based software and hardware. As these defenses have grown, the technology to adequately test them has not kept pace. This research refines a series of capabilities to support adequate test of distributed, heterogeneous network defenses:

- principles for definition of detailed criteria to identify successful defense or revealingly unsuccessful defense
- methods for characterization of network behavior in sufficient detail to support artificial traffic generation
- techniques for collection of data regarding network defense from heterogeneous sources in a manner that is both trustable, auditable and suitable for analysis
- techniques for analysis of heterogeneous data to produce fused indications of defense behavior to be evaluated to assess the success criteria

The research is both theoretical and applied; these capabilities are being refined through a series of distributed tests of network security controls. We expect the development and refinement of these capabilities to enhance both the effectiveness of modern network defenses and the ability of defenders to understand the strengths and weaknesses of these defenses.

# 16
# DNS Traffic Patterns

Principal Investigator: Sidney Faber

We studied large-scale domain name system (DNS) patterns in order to survey normal DNS traffic and identify anomalous and potentially malicious traffic. Because DNS access control can be difficult to effectively manage, permissive rules present some concern for network security. However, we found that the most significant security risks for DNS on the target networks were legitimate DNS servers with insecure configurations. We found that the most common misconfigurations involved DNS servers providing public recursion on domains for which they were not authoritative. Our findings largely confirm those found earlier by The Measurement Factory [1].

Another common misconfiguration involved forwarding of DNS queries to prisoner.iana.org, blackhole1.iana.org and blackhole2.iana.org. These queries represent reverse Pointer (PTR) lookups to private address spaces or attempted client DNS name registration when no local DNS server is available. Normally these queries should be contained within the local network. However, many networks forward this private addressing information to the public Internet.[1]

In addition to potential misconfigurations, we found that a study of user diagram protocol (UDP) packet flows yielded a wealth of information about a network's DNS topology. We are able to profile DNS clients, gateways, local DNS servers, and authoritative DNS servers. In addition, we can identify which servers are most often used by clients within the monitored network. This data demonstrates the value of using summary data to build a topology of normal and abnormal behavior for large networks.

**References**

[1] The Measurement Factory (2005). DNS Survey: June 2005. http://dns.measurement-factory.com/surveys/200506.html (2006).

1 Abley, J. & Maton, W. "AS112 Nameserver Operations," Network Working Group Internet-Draft draft-jabley-as112-ops-00.txt, Dec. 2006.

## 17
## Economics of Worm Development

Principal Investigator: Michael Collins

In this work, we attempt to predict the development of worms by modeling attackers as disinterested parties. We hypothesize that the majority of attacks over the internet are conducted by *disinterested* attackers; that is, the attacker has no special interest in a target except that the target is exploitable. It is possible, especially with automated attack tools, that attackers may not even be aware of the existence of some targets until after they are compromised by their tools.

We therefore hypothesize that attackers will develop tools to attack as large a target group as possible. To test this, we examined the prevalence of various worm designs in passively spreading peer-to-peer worms. These worms spread by masquerading as other applications (such as installers for a video game) on peer-to-peer filesharing networks like Gnutella. When activated, these worms duplicate themselves to the host's filesharing partition in order to infect other targets.

We compare the creation of worms and their variants against the total population of major peer-to-peer networks. The resulting regression analyses show a direct relationship between the size of the market share of a peer-to-peer network and the number of new variants for a worm using that network [1]. Notably, this analysis shows no significant relationship between the number of new worms and the market share, but rather with the number of modifications and adoptions of existing worms for new use.

**References**

[1] Collins, M.; Gates, C.; & Kataria, G. "A Model for Opportunistic Network Exploits: the Case of P2P Worms." *Proceedings of the 2006 Workshop on the Economics of Information Security*. June 26-28, 2006, Cambridge, UK. http://weis2006.econinfosec.org/prog.html.

## 18
## A Hybrid Approach to Scan Detection on Large Networks

Principal Investigator: Anthony Cebzanov

Network scanning tools are often used by adversaries to narrow the focus of subsequent attacks to hosts and services that are known to exist on the target network. This constitutes a significant threat to information security. Our research is directed toward finding an effective method for detecting malicious scanning activity.

A Bayesian logistic regression approach to scan detection has proven effective for large networks where only unidirectional flow-level traffic detail is available and where internal network configuration is not known in advance. Sequential hypothesis testing (also called threshold random walk, or TRW) can yield superior classification accuracy, but typically requires more advance, detailed knowledge about network traffic and topology than does the logistic regression approach [1]. To optimize performance, we have implemented a hybrid scan detection system that combines elements of these two approaches with the goal of identifying more scans than either algorithm used separately.

In developing our approach, we adapted the TRW algorithm for use with unidirectional flow data. TRW's advantages over the logistic regression model are faster operation and a lower false positive rate, attributes that make it ideal as the primary algorithm in a hybrid system. The performance advantages of TRW's sequential hypothesis approach are significant, as TRW can often arrive at a decision by analyzing a small number of flows from each traffic source [2]. Logistic regression, on the other hand, requires calculation of metrics for all flows from each source, which can cause scalability problems. However, the more exhaustive logistic regression analysis has been shown to detect scans that are missed by TRW. Furthermore, the logistic regression model detects scans in User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP) traffic, whereas TRW was designed for detection of TCP scans only.

Analysis of these characteristics led to a hybrid design where the TRW algorithm is invoked first, after which the logistic regression model is only invoked for sources classified by TRW as benign (not scanning). If the logistic regression model determines that a source is a scanner, the TRW algorithm's benign classification for that source is overturned. Because the TRW and logistic regression algorithms miss different types of scanning activity, the hybrid approach is advantageous when false positives are deemed more acceptable than false negatives [3].

This hybrid scan detection system has been deployed for several months at a large client site. The performance advantages afforded by the more efficient TRW algorithm (along with incremental improvements to the logistic regression code) have reduced the hardware requirements by 90%, while also reducing the processing time for each batch of flow data. While initial tests of the hybrid system's classification accuracy look promising, further testing and refinement of the system is planned for 2007.

**References**

[1] Gates, Carrie. "Scan Detection Using Bayesian Methods." 13-16. *CERT Research 2005 Annual Report*. Pittsburgh PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.cert.org/sse/bayesian.html.

[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] Gates, Carrie; McNutt, Josh; Kadane, Joseph B.; & Kellner, Marc. *Detecting Scans at the ISP Level* (CMU/SEI-2006-TR-005). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/publications/documents/06.reports/06tr005.html.

# 19 Network Traffic Studies of Managed Networks Using NetFlow

Principal Investigator: Sidney Faber

Very large networks present an interesting challenge when researchers attempt to determine how they are used. Data volume on dynamic, high-availability networks often precludes any detailed widespread or thorough traffic monitoring. This project has focused on using network flow data to understand wide-scale usage patterns on very large networks. Network flow data is a summary representation of traffic data collected and analyzed with the SiLK toolset [1].

We have found that managed networks--those that exercise controls to limit access--exhibit different utilization characteristics than unmanaged networks. Studies on unmanaged networks often record a high percentage of bandwidth dedicated to peer-to-peer file sharing applications [2, 3]. However, managed networks, even when using only rudimentary port-based access control lists, defeat most of the popular peer-to-peer protocols under their default settings. While a clever user can change settings and bypass these controls, most users will abandon the application, resulting in an overall reduction in bandwidth consumption.

These same port-based controls cannot provide effective throttling of streaming media applications because these generally operate over the same port as web traffic. We observed a widespread use of both streaming audio and video contributing significantly to resource utilization.

**References**

[1] Collins, M.; Kompanek, A.; & Shimeall, T. *Using SiLK for Network Traffic Analysis.* CERT Network Situational Awareness Group. Pittsburgh PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://tools.netsa.cert.org/silk/analysis-handbook.pdf.

[2] Karagiannis, T.; Broido, A.; Faloutsos, M.; & Claffy, K. "Transport Layer Identification of P2P Traffic" 121-134. *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement.* Taormina, Sicily, Italy. Oct. 25-27, 2004. New York, NY: ACM Press, 2004.

[3] CacheLogic *Research: Peer-to-Peer in 2005.* http://www.cachelogic.com/home/pages/research/p2p2005.php (2006).

# 20
# Supporting Trust Decisions

Principal Investigator: Sven Dietrich

When Internet users are asked to make "trust" decisions they often decide badly. Implicit trust decisions include decisions about whether or not to open an email attachment or provide information in response to an email that claims to have been sent by a trusted entity. Explicit trust decisions are decisions made in response to specific trust- or security-related prompts such as pop-up boxes that ask the user whether to trust an expired certificate, execute downloaded software, or allow macros to execute. Attackers are able to take advantage of most users' poor trust-decision making through semantic attacks.

It is not always possible for systems to make accurate trust decisions on a user's behalf, especially when those decisions require knowledge of contextual information. The goal of this research is not to make trust decisions for users, but rather to develop approaches to support users when they make trust decisions. This work is being done in collaboration with Lorrie Cranor and others affiliated with the Center for Usable Privacy and Security [1]. It is sponsored by the U.S. National Science Foundation (Cyber Trust initiative) and ARO/CyLab.

**References**

[1] Center for Usable Privacy and Security. http://cups.cs.cmu.edu (2004).

# 21
# The YAF Flow Meter: Enhancing IP Flow Collection

Principal Investigator: Brian Trammell

YAF is an open source flow meter developed during 2006 by the CERT Network Situational Awareness Group. YAF[1] generates unidirectional or bidirectional flow records from packets in libpcap[2] dumpfiles, live packet capture over commodity network interfaces using libpcap, or live packet capture over specialized capture hardware (e.g., DAG cards from Endace). It uses the IPFIX (IP Flow Information Export) protocol, presently being standardized by the Internet Engineering Task Force [1], to represent and export the flow records it generates. This emerging standard was adopted in order to facilitate wider sharing of data generated by YAF across organizational boundaries.

In addition to being a reference Metering Process implementation for IPFIX, YAF supports the present and future research goals of the Network Situational Awareness Group. Because the IPFIX protocol uses templates in order to be self-describing and extensible, YAF can be rapidly updated to meet new requirements for flow data collection. Such requirements include those for extended TCP state information, layer-2 encapsulation information, and partial payload capture for application protocol analysis. This update occurs without requiring the definition of new flow export representations or the synchronized deployment of updated meters and collectors throughout a flow collection system.

2007 will see further enhancements to the flow data generated by YAF, including support for collection of flows on IPv6 networks, and wider deployment in the field.

**References**

[1] Internet Engineering Task Force. http://www.ietf.org/ (2007).

1 YAF stands for "Yet Another Flow Meter."

2 *libpcap* is a basic format for saving captured network data.

Selected List of Publications

Talks/Panels/Workshops

Technical Leadership

Biographies

# Selected List of Publications

**Book Chapters**

Mead, N.R., Ch. 3, "Identifying Security Requirements Using the SQUARE Method" 44-69. *Integrating Security and Software Engineering: Advances and Future Visions.* H. Mouratidis, P. Giorgini. Hershey, PA: Idea Group, 2006 (ISBN: 1-59904-147-2).

**Reports**

Feiler, Peter; Goodenough, John; Linger, Richard; Longstaff, Tom; Kazman, Rick; Klein, Mark; Northrop, Linda; Wallnau, Kurt; Gabriel, Richard; Schmidt, Doug; & Sullivan, Kevin. *Ultra-Large-Scale Systems: The Software Challenge of the Future.* Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/uls.

Walton, Gwen; Longstaff, Tom; & Linger, Richard. *Technology Foundations for Computational Evaluation of Software Security Attributes* (CMU/SEI-2006-TR-021). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/publications/documents/06.reports/06tr021.html.

Lipson, Howard. *Evolutionary Systems Design: Recognizing Changes in Security and Survivability Risks* (CMU/SEI-2006-TN-027). Pittsburgh PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/publications/documents/06.reports/06tn027.html.

Chung, L.; Hung, F.; Hough, E.; Ojoko-Adams, D.; Mead, N.R. (Faculty Advisor), *Security Quality Requirements Engineering (SQUARE) Case Study Phase III* (CMU/SEI-2006-SR-003). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2006. http://www.sei.cmu.edu/publications/documents/06.reports/06sr003.html.

Woody, C. *Eliciting and Analyzing Quality Requirements: Management Influences on Software Quality Requirements* (CMU/SEI-2005-TN-010). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tn010.html.

Ellison, Robert J. *Trustworthy Integration: 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.

Janies, J.; Johnson, N.; & Huang, C. *Secure Unicast Messaging for Wireless Ad Hoc Sensor Networks* (NIST GCR 05-878). Gaithersburg, MD: National Institute of Standards and Technology, 2005.

**Papers**

Ellison, Robert & Woody, Carol. *Assessing Security, Reliability, and Survivability Before Implementation.* Tutorial EDUCAUSE 2006, Oct. 2006.

Linger, Richard; Prowell, Stacy; & Pleszkoch, Mark. "Automated Calculation of Software Behavior with Function Extraction (FX) for Trustworthy and Predictable Execution." *Proceedings of NIST Software Assurance Metrics and Tool Evaluation (SAMATE) Static Analysis Summit.* Gaithersburg, MD, June 29, 2006. Gaithersburg, MD: National Institute of Standards and Technology, 2006.

Dietrich, Sven. "Connecting the dots: DDoS, botnets and other threats." *Proceedings of the 3rd Summer Workshop on Information Security.* Algiers, Algeria, June 17-19, 2006. Algiers, Algeria: National Institute of Informatics (INI). http://www.ini.dz/conference/ecole_printemps2006/Pdf/1.Long_Sven.pdf.

Janies, Jeff; Huang, Chin-Tser; Johnson, Nathan L. "SUMP: A Secure Unicast Messaging Protocol for Wireless Ad Hoc Sensor Networks." *IEEE International Conference on Communications (ICC 2006).* Istanbul, Turkey, June 11-12, 2006. New York, NY: IEEE Communications Society, 2006.

Mead, N.R. & Hough, E.D. "Security Requirements Engineering for Software Systems: Case Studies in Support of Software Engineering Education," 149-156. *19th Conference on Software Engineering Education & Training.* Turtle Bay, Hawaii, April 19-21, 2006. Los Alamitos, CA: IEEE Computer Society.

Linger, Richard; Daly, Tim; Pleszkoch, Mark; Prowell, Stacy; & Walton, Gwen. "Function Extraction (FX) Technology: Automated Calculation of Program Behavior for Software Assurance." *Proceedings of NSA High Confidence Software and Systems Conference.* Ft. Meade, MD, Feb. 2006. Los Alamitos, CA: IEEE Computer Society Press, 2006.

Lipson, Howard. "Evolutionary Design of Secure Systems–The First Step Is Recognizing the Need for Change," Department of Homeland Security "Build Security In" web site, 2006. https://buildsecurityin.us-cert.gov/daisy/bsi/articles/best-practices/assembly/467.html?branch=1&language=1.

Mead, N.R. & McGraw, G. "Building Security Into Software Development Lifecycle Processes," *Cutter Enterprise Risk Management and Governance Advisory Service Executive Update 2,* 24, 2006.

Lipson, Howard & van Wyk, Ken. "Application Firewalls and Proxies–Introduction and Concept of Operations," Department of Homeland Security "Build Security In" web site, 2005. https://buildsecurityin.us-cert.gov/daisy/bsi/articles/bestpractices/assembly/30.html?branch=1&language=1.

**Journal Articles**

Alberts, C. & Woody, C. "Consider Operational Security Risk During System Development." Scheduled for *IEEE Security & Privacy* (Jan.-Feb. 2007).

Mead, N.R. "Experiences in Eliciting Security Requirements." *CrossTalk 19*, 12, (Dec. 2006) 14-19.

Fung, Casey; Hung, Patrick; Linger, Richard; Wang, Guijun; & Walton, Gwendolyn. "A Service-Oriented Composition Framework with QoS Management." *International Journal of Web Services Research, 3,* 3. Hershey, PA: Idea Group Publishing, July-Sept. 2006.

Woody, C. "Securely Sustaining Software-Intensive Systems," *Cutter IT Journal* (Jan. 2006).

Faber, S. & Shawgo, J. *Windows Server 2003 Benchmark.* The Center for Internet Security, Oct. 2005.

## Talks/Panels/Workshops

Linger, Richard. "Function Extraction Technology for Software Assurance," CSIA Interagency Working Group, National Science Foundation, Arlington, VA, Dec. 2006.

Linger, Richard; Burns, Luanne; & Prowell, Stacy. "Function Extraction for Software Assurance," Senior Leadership Forum on Function Extraction Technology for Software Assurance, SEI, Arlington, VA, Nov. 2006.

Woody, Carol. CSI 2006, "Operational Risk in Software Development", Kissimmee FL, Nov. 2006.

Woody, Carol. EDUCAUSE 2006, "An approach for assessing the security, reliability and survivability risk," Dallas, TX, Oct. 2006.

Woody, Carol. *Better Software Conference 2006*, "Improve Software Operational Security," Las Vegas, NV, June 2006.

Linger, Richard; Longstaff, Tom; Pleszkoch, Mark; & Hevner, Alan. "Next-Generation Software Engineering: Transformation to a Computational Discipline," Workshop on Next-Generation Software Engineering, 39th Hawaii International Conference on System Sciences (HICSS), Koloa, Kauai, HI, Jan. 2006.

Lipson, Howard. "Cyber Security and Control System Survivability: Technical and Policy Challenges," Power Systems Engineering Research Center (PSERC) Tele-Seminar, Nov. 1, 2005. Invited Presentation.

# Technical Leadership

**Sven Dietrich**
Associate Editor, IEEE Cipher, 2003-present

Program Committee Member, Conference on Detection of Intrusions, Malware and Vulnerability Assessment (DIMVA), July 2006, Berlin, Germany

Program Committee Member, Financial Cryptography and Data Security Conference, Feb.-Mar. 2006, Anguilla, British West Indies

Program Committee Member, Workshop on Information Security Applications (WISA) Conference, Jeju Island, Korea, Aug. 2006

Member, IEEE CS Technical Committee on Security and Privacy, 1998-present

Reviewer, *Journal of Computer Security,* 2003-2006

Reviewer, *IEEE Transactions on Dependable and Secure Computing,* 2003-2006

Reviewer, *IEEE Security & Privacy Magazine,* 2003-2006

Reviewer, *ACM Transactions on Internet Technology,* 2003-2006

**Sidney Faber**
Editor, *The SANS Advisor. 2005-2006.*
http://www.sans.org/newsletters/advisor/

**Richard Linger**
Co-chair (with Alan Hevner and Gwendolyn H. Walton), Workshop on Next-Generation Software Engineering (HICSS), Koloa, Kauai, HI, Jan. 2006

Program Committee Member, Hawaii International Conference on System Sciences (HICSS), 2002–present

**Howard Lipson**
Team leader for the *Assembly, Integration & Evolution* content area and the *Assurance Cases* content area of the DHS "Build Security In" web site (https://buildsecurityin.us-cert.gov/)

Session chair for Digital Identity Management Frameworks for the Workshop on Digital Identity Management (DIM 2005), ACM Conference on Computer and Communications Security, Nov. 2005

Program Committee member for the Workshop on Digital Identity Management (DIM 2006), ACM Conference on Computer and Communications Security, Nov. 2006

Invited Speaker, Power Systems Engineering Research Center (PSERC) Tele-Seminar, "Cyber Security and Control System Survivability: Technical and Policy Challenges," Nov. 2005

Reviewer for the International Risk Governance Council (Geneva, Switzerland) of its document *Managing and Reducing Social Vulnerabilities from Coupled Critical Infrastructures*

Member (founding), Carnegie Mellon Electricity Industry Center

Member of the Advisory Board, Graduate Program in Computational Mathematics, Duquesne University

**Nancy Mead**
IEEE Fellow for leadership in software engineering education, 2006

Editorial Board Member, 2003–2006, *IEEE Security & Privacy*

Editorial Board Member, 2002–present, *Requirements Engineering Journal*

# Biographies

**Anthony Cebzanov**

Anthony Cebzanov is a software developer on the Network Situational Awareness team. He participates in the design and implementation of software systems used to measure and categorize security risks through network flow analysis.

Prior to joining the SEI, Cebzanov worked as an Information Systems Engineer for The Vanguard Group of Investment Companies, where he designed and developed authentication, authorization, and access control systems. Cebzanov holds an MS in Software Engineering and BS in Computer Science degrees from Pennsylvania State University.

**Michael Collins**

Michael Collins has been a member of the Analysis Team within the Network Situational Awareness Group since the group's founding. In this capacity, he developed the first version of the SiLK packing system and analysis suite in collaboration with the late Dr. Suresh Konda. He now focuses on questions in traffic analysis, particularly issues of anonymity and application identification. Collins has published and holds patents for work in several fields, including security and knowledge management.

Collins holds a BS in Physics and an MS in Electrical Engineering from Carnegie Mellon University and is currently a PhD candidate in Electrical Engineering. Prior to his work at CERT, he worked as a researcher for the Institute for Complex Engineered Systems and as a consultant for several Fortune 1000 companies.

**Markus De Shon**

As the analyst team lead for the Network Situational Awareness Group, Markus De Shon guides the research activities of the network security analysts and provides support to analysis operations for government agency customers.

Prior to joining the SEI, De Shon was Chief Scientist at SecureWorks, Inc. His work included designing intrusion prevention (IPS) technologies, developing IPS signatures based upon vulnerabilities and reverse engineering of malicious code, analyzing network activity, and acting as the final incident handling escalation point for an IPS service to nearly 1,000 small to medium-sized businesses.

De Shon holds a PhD in Nuclear Physics and an MS in Health Physics from the Georgia Institute of Technology.

**Sven Dietrich**

Sven Dietrich is a senior member of the CERT technical staff. As a member of the Survivable Systems Engineering Group, he investigates computer security and survivable systems engineering. Dietrich also actively participates in CyLab, a collaboration between the CERT Program and several departments at Carnegie Mellon University, 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 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 Ellison**

Robert J. Ellison is a senior member of the CERT technical staff. 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 Carnegie Mellon, 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 CERT Program, he contributed to the development of the Survivable Systems Analysis Method and is currently exploring how to analyze the security issues that arise with systems of systems.

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.

**Sid Faber**

As a member of the CERT Network Situational Awareness Group, Sid Faber supports customers by providing detailed reports of current and historical network activities.

Prior to joining the SEI, Faber worked as a security architect with Federated Investors, one of the largest investment managers in the United States. His experience includes over 10 years in software application development and evaluation, and 5 years in the U.S. Navy Nuclear Power program. Faber holds Global Information Assurance Certification for Intrusion Detection (GCIA), Windows Security Administrator (GCWN) and Forensics Analyst (GCFA).

**Phil Groce**

Phil Groce is a member of the technical staff in the CERT Program. As a member of technical staff on the Network Situational Awareness team, Groce develops software to support situational awareness activities in research and operational environments. Groce's current work focuses on improving the usefulness of data analysis and visualization tools in security operations.

Before joining CERT, Groce was a software engineer for the messaging security company CipherTrust, where he helped develop email security software. Prior to that, Groce was a Senior Researcher and Senior Software Engineer at SecureWorks, a managed security service provider. There he participated in the development of the company's intrusion prevention and security analysis capabilities. Groce holds a BA in history from the University of Memphis.

**Jeff Janies**

Jeff Janies is a member of the technical staff and performs research in the CERT Program. Janies is an analyst for the Network Situational Awareness Group. His research interests include anomaly-based intrusion detection systems and wireless ad hoc sensor network protocol design.

Prior to joining the SEI, Janies attended the University of South Carolina, where he obtained his MS in Computer Science and Engineering. He also holds a BS in Computer Science from Louisiana State University.

**Richard Linger**

Richard Linger is manager of the CERT Survivable Systems Engineering Group and STAR*Lab, where he manages the function extraction project. He has extensive experience and knowledge in function-theoretic foundations for software engineering. Linger also directs research in flow-service-quality (FSQ) engineering for network-centric system survivability, and next-generation software engineering for ultra-large-scale system development. He serves as a member of the faculty at the CMU Heinz School of Public Policy and Management. At IBM, Linger partnered with Dr. Harlan Mills, IBM Fellow, 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 certification of software fitness for use. He pioneered use of Cleanroom technology for software product development, achieving zero-defect performance with improved productivity, and founded and managed the IBM Cleanroom Software Technology Center. He 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, 12 book chapters, and over 60 papers and journal articles. He is a member of the IEEE and the ACM.

**Howard F. Lipson**

Howard F. Lipson is a senior member of the CERT technical staff. Lipson has been a computer security researcher at CERT for more than 14 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 has been 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 Coordination Center (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 CERT Program. Longstaff has spent the past 13 years managing and initiating many CERT projects and initiatives such as the former CERT Analysis Center, former CERT Research Center, many survivability projects, and most recently Network Situational Awareness. His current scope of work includes evaluating technology across the entire CERT Program to assure continued quality and innovation of all the work at CERT.

Longstaff is responsible for strategic planning for the CERT 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 CERT 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 Survivable Systems Engineering Group, which is part of the CERT 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 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.

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

**Mark Pleszkoch**

Mark Pleszkoch is a senior member of the CERT technical staff. He is an expert in function-theoretic mathematical foundations of software, and focuses on automation of formal methods. As a member of the function extraction research and development team, he is responsible for creating theoretical foundations and engineering automation for FX systems.

Prior to joining CERT, Pleszkoch worked at IBM for 21 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. He 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 and an MA and a BA in Mathematics from the University of Virginia. He has a number of publications in formal methods and software engineering. 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, and is a member of the IEEE and the Association for Symbolic Logic.

**John Prevost**

Since arriving at CERT, John Provost has worked primarily on database and user interface integration for the SiLK network data collection and analysis toolset. His past work is quite diverse, with a range that includes systems administration for the Carnegie Mellon School of Computer Science, developing data visualization software at MAYA Design, Inc., and designing database-backed web sites at ArsDigita.

**Stacy Prowell**

Stacy Prowell is a senior member of the CERT technical staff, and chief scientist of STAR*Lab. He is an expert in the function-theoretic foundations of software, and is currently conducting research and development for function extraction technology. Prowell has managed both commercial and academic software development projects and consulted on 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, he 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, he served as a consultant in the software industry. His 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.

**Kirk Sayre**

Kirk Sayre is an expert in the function-theoretic mathematical foundations that are the basis for function extraction technology. He is currently working on development of the core rewriting engine for the FX system, as well as on formal testing for the system.

Prior to joining CERT, Sayre was a research professor at the University of Tennessee, where he developed an automated testing framework for the certification of generic scientific computing libraries. In his position at UT, Sayre also developed a CASE tool to support the editing and creation of rigorous sequence-based software specifications. This tool is currently being used on software projects at Oak Ridge National Laboratory and Bosch. Sayre has developed software in many different areas, including educational web applications, automated testing tools, CASE tools, medical devices, and weapons systems. He received a BS in Computer Science from Bucknell University in 1992, an MS in Computer Science from American University in 1994, and a PhD in Computer Science from the University of Tennessee in 1999.

His primary areas of research are function extraction, automated testing, statistical analysis of test data, and formal methods of software specification.

**Timothy J. Shimeall**

Timothy J. Shimeall is a senior member of the technical staff with the CERT Network Situational Awareness Group, where he is responsible for overseeing and participating in the development of analysis methods in the area of networked systems security and survivability. This work includes development of methods to identify trends in security incidents and in the development of software used by computer and network intruders. Of particular interest are incidents affecting defended systems and malicious software that are effective despite common defenses. Shimeall is also an Adjunct Professor of the CMU Heinz School of Public Policy and Management, with teaching and research interests focused in the area of information survivability. He is an active instructor in information security management and information warfare and has led a variety of survivability-related independent studies.

**Brian Trammell**

Brian Trammell is the technical lead of the Network Situational Awareness (NetSA) Group's engineering team. His primary area of responsibility is the design, implementation, and deployment of network data collection and analysis tools for use by NetSA's Analysis team, internet security analysts at NetSA's sponsors, and the security-focused internet measurement community at large. He is active in the Internet Engineering Task Force, advancing standards for the representation, storage, and interchange of security-relevant network data in the IPFIX, PSAMP, INCH, and related working groups. He is the editor or co-author of several current Internet-Drafts.

Prior to joining NetSA, Trammell was a co-founder of Leapfrog Research & Development, LLC, a Pittsburgh-based software development firm; an engineer at PanGo Networks, Inc. in Pittsburgh, where he was principally responsible for the Site Surveyor component of the Proximity Platform; a UNIX System Administrator at the School of Civil and Environmental Engineering at the Georgia Institute of Technology; and a member of Georgia Tech's Collaborative Software Laboratory. He is a current member of the ACM.

**Gwendolyn H. Walton**

Gwendolyn H. Walton is a senior member of the CERT technical staff. 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 the ACM.

**Rhiannon Weaver**

Rhiannon Weaver joined the CERT Network Situational Awareness Group in September 2006 and provides support to various research initiatives in the group through statistical and mathematical modeling.

Weaver holds a BS in Mathematics and BS in Computer Science from Penn State University and an MS in Statistics from Carnegie Mellon University. She is currently working on her PhD in Statistics at Carnegie Mellon. Her interests include scientific computing, Bayesian statistics, and dynamic hierarchical modeling.

**Carol Woody**

Carol Woody is a senior member of the CERT technical staff. Her research is focused on ways to address software design and development that improve the security of the implemented results. She is leading two research projects for the DoDo that are developing and validating the Survivable Analysis Framework.

Woody is a member of the Survivable Enterprise Management team in the CERT Program. She participated in the development of the Operationally Critical Threat, Asset, Vulnerability Evaluation (OCTAVE®) methodology for applying good security practices through risk management. In addition she developed and piloted a version of OCTAVE for use in K-12 schools and school districts.

Woody has over 25 years of experience in software development and project management covering all aspects of software and systems planning, design, development, and implementation in large complex organizations. Before coming to the SEI, she consulted for New York City as a strategic planner for the Administration of Children's Services addressing the financial technology needs of the $2 billion organization during the formulation of the agency and its transition through Y2K. She also managed the user testing for a timekeeping application purchased by NYC to handle 160,000 employees in over 100 agencies; activities involved included work force scheduling for police, fire, sanitation, and correction. Woody has a biographical citation in *Who's Who in American Women and Who's Who in Finance and Industry.* She is a member of IEEE, the ACM, and PMI.

Woody holds a BS in Mathematics from The College of William and Mary, an MBA with distinction from Wake Forest University, and a PhD in Information Systems from NOVA Southeastern University where she was elected to Upsilon Phi Epsilon, the international honor society for computing and information disciplines.


Software Engineering Institute
Carnegie Mellon



The 2006 CERT Research Report was produced by SEI Communications.

**Executive Editor**
Nancy Mead

**Executive Reviewers**
Rick Linger
Thomas Longstaff
Nancy Mead

**Editor-in-Chief**
Claire Dixon

**Editorial**
Mindi McDowell
John Morley

**Design**
Cat Zaccardi

**Production**
David Gregg
Melissa Neely

**Software Engineering Institute**
Carnegie Mellon University
4500 Fifth Avenue
Pittsburgh, PA 15213-2612

Phone: 412-268-5800
Fax: 412-268-5758
www.cert.org

CERT

Software Engineering Institute
Carnegie Mellon