10/4

Volume 10 Number 4 • **Winter 2007**

IAnewsletter

The Newsletter for Information Assurance Technology Professionals

Information Assurance for the Net-Centric Environment: Making the Mission Possible

also inside

IATAC

contents

About IATAC and the *IAnewsletter*
The *IAnewsletter* is published quarterly by the Information Assurance Technology Analysis Center (IATAC). IATAC is a Department of Defense (DoD) sponsored Information Analysis Center, administratively managed by the Defense Technical Information Center (DTIC), and Director, Defense Research and Engineering (DDR&E).

Contents of the *IAnewsletter* are not necessarily the official views of or endorsed by the US Government, DoD, DTIC, or DDR&E. The mention of commercial products and/or does not imply endorsement by DoD or DDR&E.

Inquiries about IATAC capabilities, products, and services may be addressed to—

IATAC Director: Gene Tyler
Inquiry Services: Peggy O'Connor

***IAnewsletter* Staff**
Promotional Director: Christina P. McNemar
Creative Director: Ahnie Jenkins
Art Director: Don Rowe
Copy Editors: Louise Price, Gina Abruzzese
Designers: Ricardo Real, Tammy Black
Editorial Board: Ronald Ritchey, Tara Shea, Gene Tyler, Buzz Walsh

***IAnewsletter* Article Submissions**
To submit your articles, notices, programs, or ideas for future issues, please visit http://iac.dtic.mil/iatac/IA_newsletter.html and download an "Article Instructions" packet.

***IAnewsletter* Address Changes/ Additions/Deletions**
To change, add, or delete your mailing or email address (soft-copy receipt), please contact us at—

IATAC
Attn: Peggy O'Connor
13200 Woodland Park Road
Suite 6031
Herndon, VA 20171

Phone: 703/984-0775
Fax: 703/984-0773

email: iatac@dtic.mil
URL: http://iac.dtic.mil/iatac

Deadlines for future Issues
Spring 2008 March 14th, 2008

Cover design: Ricardo Real
Newsletter design: Bryn Farrar, Donald Rowe

Distribution Statement A: Approved for public release; distribution is unlimited.

in every issue

IATAC Chat

Gene Tyler. IATAC Director

For us to accomplish our mission, we must know the information, where to find the information, who has the information, and—most importantly—how to get the right information to the right people.

If you are one of the many individuals who has been receiving Information Assurance Technology Analysis Center (IATAC) information and products for years, then you are probably well aware of our overarching mission and goals. For those who are "new" to IATAC, then you may not. IATAC was set up as a Department of Defense (DoD) institution to *"provide the DoD a central point of access for information on Information Assurance emerging technologies in system vulnerabilities, research and development, models, and analysis to support the development and implementation of effective defense against Information Warfare attacks."* For us to accomplish our mission, we must know the information, where to find the information, who has the information, and—most importantly—how to get the right information to the right people.

Although we certainly believe we have the right people working for IATAC, we also know it is impossible for us to have all of the answers all of the time. It is for this reason that we must know how and where to find the best and most accurate information. IATAC has invested a considerable amount of time and effort into developing extensive resources, tools, and capabilities to obtain the *right* information. We also realize the importance of knowing not only where and how to obtain information but also *who* has the information. We know there are areas of information that are not readily available to us; therefore, to obtain this imbedded information, we must know who to turn to—the experts. IATAC has an extensive network of subject matter experts (SME) from across the information assurance (IA) community. SMEs are an essential resource to IATAC and the DoD community as a whole. These experts are individuals who volunteer to be part of this phenomenal institution by assisting IATAC with the tough technical inquiries we often receive. Throughout the years, SMEs have been pivotal in helping us respond rapidly and accurately to these inquiries. In fact, just recently, a top government official formally recognized several of our SMEs for their assistance with a technical inquiry.

We add the information we obtain and develop to the IATAC library to make it easily accessible for the IA community and our future use. IATAC uses various methods of information gathering that, when combined, provide a powerful and comprehensive research showcase that is functional and vital to the IA community. IATAC serves as a central authoritative source for IA and is dedicated to ensuring the continuity of information critical to the nation's defense in support of federal agencies.

IATAC ensures we consistently deliver information to the right individuals and that they can easily obtain it. We have numerous free products and services available to the IA community, including our State-of-the-Art Reports (SOAR), the biweekly *IA Digest,* our quarterly *Research Update,* our SME program, the *IAnewsletter,* and up to 4 hours of complimentary research—just to name a few. Individuals may visit our website, email us, or call us to find out more or inquire how they may receive these numerous products and services.

As with every edition, this edition has several fascinating, well-written, and timely articles. We hope you will find them as thought provoking and worthwhile as we do. ■

Information Assurance for the Net-Centric Environment: Making the Mission Possible

by Craig Harber

"Defense transformation hinges on the recognition that information is our greatest source of power… The information systems have to be secure… security is key."

John Grimes, DoD CIO/ASD(NII)

We've all heard that the Global Information Grid (GIG) is the future of secured information for our armed services. Together with the Department of Defense's (DoD) Net-Centric Environment (NCE)—the operational construct within the GIG that enables Net-Centric Warfare (NCW)—the GIG is also a high-value target that must be safeguarded against both outsider attacks and insider threats.

How do we best protect the GIG from our adversaries and secure it for use in a variable-trust environment? And how must our information assurance (IA) strategies differ from those we have used in the past?

Version 1.1 of the IA Component of the GIG Integrated Architecture, developed under the leadership of the National Security Agency (NSA) with participation from the Services, Joint Staff, and Defense Information Systems Agency (DISA) and the approval of the DoD Chief Information Officer (CIO), addresses these issues to help "make the mission possible" for our warfighters and coalition partners.

Background

DoD defines the NCE as a joint force framework for full human and technical connectivity and interoperability—one that allows all DoD users and mission partners to share the information they need, when they need it, in a form they can understand, and act on with confidence. DoD has also identified the NCE as the best way for our Armed Forces to achieve information superiority, which in today's world requires new methods of IA. Because users, services, information, and networks reside in a Net-Centric information sharing space, data encryption, and firewall deployment at the boundaries of our networks can no longer assure information and services.

Just as stealth, armor, and maneuverability help protect traditional weapons platforms, robust IA layered throughout the GIG helps protect the GIG against cyber warfare attacks and maintain DoD's decisionmaking ability, command and control (C2) capabilities, and operational effectiveness. Technology is a key component, but achieving an assured GIG (or National Security Enterprise) also requires enterprise-level governance, systems engineering, policy, risk management, operational doctrine, and training.

IA enables the assured sharing of information, provides mechanisms to protect systems and data from cyber attack, and facilitates the restoration of compromised information systems. As the GIG balances the "need to know" and "need to share" paradigms, user reliance on the integrity and availability of shared information to execute missions can only increase. IA is critical to the operational readiness of Net-Centric capabilities and for protection against sophisticated adversaries in the NCE.

The IA Component of the GIG Integrated Architecture is a first step in providing an IA framework for achieving the assured, integrated, and survivable information enterprise necessary to attain the strategic objectives of the National Security Community.

Assistant Secretary of Defense for Networks and Information Integration Mandates the Information Assurance Component

In a July 2003 memorandum, the Assistant Secretary of Defense for Networks and Information Integration—ASD(NII)—outlined the criticality of "trusting the GIG and protecting the information that is retained or flows through it" and asserted the need for an IA Component in the GIG Integrated Architecture. The ASD(NII) tasked NSA to lead the development effort, and NSA released Version 1.0 of the IA Component in October 2004.

As GIG IA Domain Agent, NSA also established the GIG IA Portfolio Management (GIAP) Office to serve as a community organization reporting to ASD(NII). GIAP ensures GIG-related IA investment,

research, development, and procurement activities are properly planned and synchronized; are consistent with the GIG IA Architecture; and provide the IA capabilities needed to enable an assured GIG Enterprise.

The strategy, technical framework, and transition plan outlined in the GIG IA Architecture also apply to the Intelligence Community (IC); Department of Homeland Security (DHS); information sharing environments of federal, state, and local entities; and critical National Security Community infrastructures nationwide.

Enhancements in Version 1.1

Since the release of Version 1.0, NSA has actively worked to evolve the IA Component of the GIG to meet IC needs. NSA is also working to align the GIG IA Component with DoD processes; incorporate IA content into DoD compliance "documents of standing," such as the Net-Centric Operations and Warfare Reference Model (NCOW RM) and Net-Centric Implementation Documents (NCID); and align it with IC-sponsored efforts, including development of the GIG IA Initial Capabilities Document (ICD) and the GIAP IA Capability Roadmaps.

NSA has restructured Version 1.1 of the IA Component to better support DoD's Decision Processes Planning, Programming, Budgeting, and Execution (PPBE); the Joint Capabilities Integration and Development System (JCIDS); and the Defense Acquisition System (DAS). Version 1.1 aligns with IA operational capabilities in the approved GIG IA ICD—the basis for development of IA operational activities, IA system functions, and incremental capabilities of the IA transition strategy—and the Capabilities-Based Assessment (CBA). All agree that to be effective, IA must be integrated into all aspects of the solution space.

The ongoing evolution of NSA Enterprise IA Architecture and Systems Engineering analysis efforts has focused on external IA IC efforts and products rather than the development of standalone documents. This path requires continual synchronization and alignment among NSA efforts in this area and those of other DoD organizations responsible for product development.

DoD Net-Centric Enterprise Information Assurance Vision

The DoD Net-Centric Enterprise IA vision is to dynamically protect information and systems needed to enable information sharing and collaboration in the NCE between users and systems with varying levels of trust and IA capabilities. This effort will involve ongoing research, development, analysis of solution alternatives, systems engineering, policy development, and operational evaluation. It will be achieved through robust IA functionality incorporated in information technology components and distributed in a defense-in-depth (DiD) construct across the full spectrum of strategic, operational, and tactical environments.

Five key elements have been identified as critical to this vision. Together, they help safeguard the GIG and the NCE by addressing the need to move beyond traditional "bolt on" perimeter methods of security to dynamic IA. The DoD Net-Centric Enterprise IA is built into the system from its inception, ensuring information is labeled and protected at appropriate levels and determining whether it should be shared, with whom, and under what conditions. It also guards against insider and outsider threats, and it contains both types of threats—whether intentional or otherwise. The five elements critical to the DoD Net-Centric Enterprise IA vision are as follows:

Transactional Information Protection
Granular, end-to-end security controls enabling protected information exchanges within the variable-trust, Net-Centric environment

System IA requirements have historically been based on the highest level of information that a system contains. Today's users require a more flexible and dynamic IA approach that considers information sensitivity, mission criticality, and the ability of systems to protect information end to end throughout its life cycle. Internal barriers are also necessary to block

adversaries who have managed to gain entry by accessing information elsewhere in the NCE.

The wide range of users operating in the NCE requires that access controls address the following: trust levels and user and system privileges that change over time, embedded information access criteria, the physical environment into which information will be released, life-cycle protection capabilities, and mission needs. Automated mechanisms will evaluate these criteria and share information if and when all necessary conditions have been met.

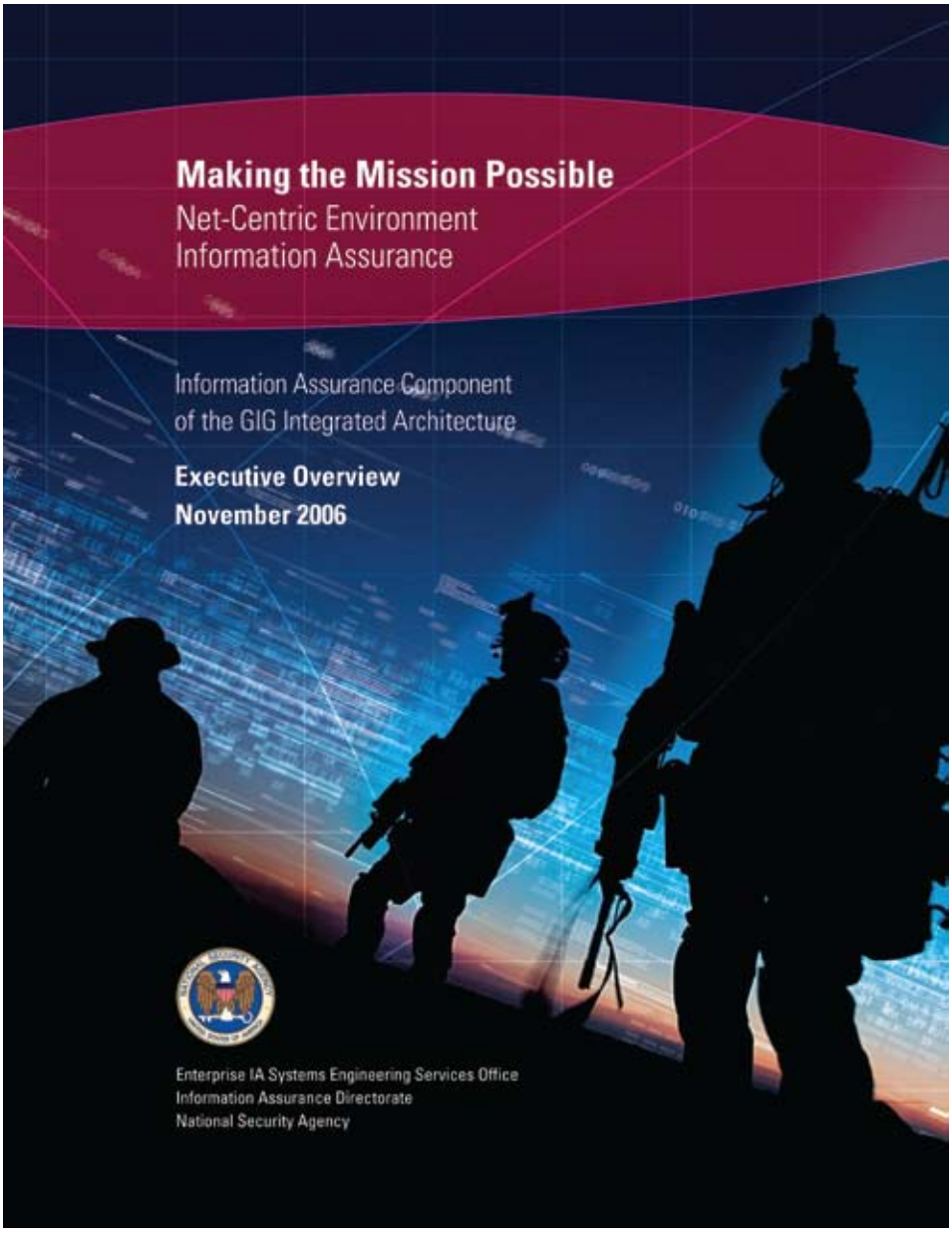
Because of the differing levels of trust and IA protection assigned to individual users and systems, collaboration and information sharing in the NCE must use a dynamic IA approach to ensure exchanges occur only when authorized and when systems involved can protect the information adequately.

Digital Policy-Enabled Enterprise
Dynamic response to changing mission needs, attacks, and system degradations through highly automated and coordinated distribution and enforcement of digital policies
The NCE is a high-priority target for adversaries of all skill levels and motivations. The interdependence and interconnection of systems that make up the NCE can also potentially increase attack avenues and decrease the ability to contain adversarial impacts, resulting in temporary system outages, degradations, and competing demands for limited resources.

Enabled by digital policy, IA can adjust resources and system configurations to ensure the highest priority missions continue to receive the resources they need while limiting the spread of attacks to adjacent parts of the system. Use of real-time IA situational awareness to assess the health and readiness of the environment; the ability to de-conflict digital policy at local, regional, and enterprise levels to ensure updates do not affect missions in unintended ways; and the ability to distribute and enforce policy across the enterprise are keys to addressing these challenges.

Defense Against an Adversary From Within
Persistently monitor, track, search for, and respond to insider activity and misuse in the enterprise
Today's wider audience for information sharing requires enhanced monitoring, misuse detection, and network defense capabilities to guard against potential adversarial operations in the enterprise. Strong IA in this area must include limiting configuration knowledge that could help identify attack avenues, restricting abilities to increase personal access or add new accounts, increasing capabilities to detect and identify unauthorized usage and changes, and limiting access in the environment to contain adversarial activities.

A distributed sensor grid, used with transactional IA access control mechanisms, can enhance the tracking of user actions throughout the enterprise via input collected across classification levels, communities of interest (COI), and missions.



A dynamic response capability can enable coordination and de-confliction of attack responses across the environment. It can also automatically generate digital policy updates to address and contain any identified threats. This enhanced ability to monitor, track, search for, and respond to attacks in the NCE not only reduces the possibility of unauthorized access but also improves response capabilities and damage control should a security breach occur.

Integrated Security Management
Dynamic automated Net-Centric security management, seamlessly integrated with operations management
To adequately protect the NCE—which constantly changes in response to attacks, outages, and revised mission needs—the security management infrastructure must evolve. The infrastructure must be able to support not only a more automated, Net-Centric key management capability but also enhanced IA management capabilities, such as identity management, privilege management, auditing, and security configuration management.

Automating these procedures wherever possible will reduce operational burden and configuration errors, improve real-time support, and minimize dependence on users and system administrators to enforce cyber security. Secure management and control exchanges in the NCE and application of robust physical, procedural, and technical security controls to the management infrastructure components are critical to enterprise IA protection.

Increased automation and the ongoing ability of the NCE to evolve creatively in response to changing circumstances are vital to safeguarding an environment that relies on transactional enterprise protection built around a distributed array of IA mechanisms.

Enhanced Integrity and Trust of Net-Centric Systems
Robust IA embedded within enterprise components and maintained throughout their life cycle
The NCE's end-to-end information protection requires the distribution of IA functionality in a DiD construct across system components. This distribution involves a greater reliance on commercially available products and software-based IA functionality—and greater trust in the platforms that perform IA functions. DiD helps maintain the integrity and trust of the NCE by providing multiple layers of protection for information and services.

Deliberately building diversity into the type, placement, configuration, robustness, and suppliers of protection mechanisms will improve effectiveness and increase the detection of inappropriate activity. These embedded protection mechanisms will also lessen the ability of an attack to compromise a major portion of the GIG.

This reliance on commercial resources to achieve the levels of integration, flexibility, and cost-effectiveness needed to realize the Net-Centric vision means that special care is required to ensure the integrity of every component used.

Global Information Grid Information Assurance Transition Strategy

As the GIG evolves from a collection of separate legacy systems into a seamless entity that serves our Armed Forces and selected coalition partners, a common IA strategy for information and infrastructure is needed across the National Security Community. The GIG IA Transition Strategy—an incremental plan to achieve the 15 IA operational capabilities required by the GIG IA ICD—was developed for this purpose.

This Strategy describes the transformation of existing DoD IA operations, technologies, processes, and people to the global, policy-based, assured capabilities and services needed for ongoing mission success. Each of the Strategy's increments corresponds to an increase in user-available operational capabilities, as well as the IA capabilities needed to manage risk to acceptable levels and facilitate transition to the next increment. Of the three primary increments identified, Increments 1 and 2 have clearly defined capabilities. Increment 3 displays less fidelity and could potentially generate additional increments as the NCE evolves.

The goal of Version 1.1 of the IA Component is to achieve Increment 1 as a set of capabilities and milestones rather than within a specific timeframe.

About Increment 1

Increment 1 of the GIG IA Transition Strategy will support US-only, releasable, bilateral, and COI-specific information sharing within a federated system-high environment consisting of DoD, the IC, the DHS, and close allies. It focuses on IA capabilities that improve discovery, sharing, and collaboration across a broad range of users and organizations while preventing inadvertent disclosure.

In Increment 1, traditional perimeter defense mechanisms will continue to provide the primary means of information protection, and all users will be cleared to at least the system-high level. Because of this relatively homogenous user base (common level of trust) and the low risk potential, commercial IA technology can provide needed capabilities.

The planned Increment 1 environment will enable DoD participants to share information, collaborate among related groups of users, and exchange higher risk file attachments. It will establish situational awareness within domains and permit the safe posting and retrieval of detailed information across federated system-high environments. It will also support the dynamic formation of COIs across organizational boundaries.

Although the potential for risk is lower in the Increment 1 environment than in a variable-trust environment, provisions are necessary to protect against cyber attacks and insider activity or misuse in the individual system-high environments and the federated environment.

Increment 1 will support the following core IA capabilities: consistent mechanisms to label users and information, fine-grained access controls (based on those labels) to information and

services, and persistent monitoring and misuse detection. A common Net-Centric Computer Network Defense (CND) infrastructure, a User Defined Operational Picture (UDOP), and enhanced Cross-Domain Solutions (CDS) capabilities should also result from Increment 1.

Follow-on efforts will build on Increment 1 to offer enhanced mission operational capabilities. These efforts will also provide opportunities for program adaptation, proof of new concepts and theories, and technology maturation.

The Need for IA Integration

To effectively protect the GIG and safeguard the NCE, IA must be integrated into service, agency, and acquisition program architectures.

Version 1.1 of the IA Component describes the IA Architecture Views that support the family of joint concepts, GIG IA ICD, and DoD strategies from an operational, systems, and technical perspective. Collectively, mission areas can use these views to develop architectures that identify aspects of IA that apply to the problem space, and to tailor solutions based on their operational environments. Establishing a common lexicon and methodology for IA Architecture Components and their relationships to DoD Enterprise Architectures here is critical.

The IA Operational View identifies and defines IA operational activities necessary to support mission needs and operational capabilities. This view aligns with Joint NetOps Architecture and CND and integrates with the NCOW RM.

The IA Systems View supports operational activities through a set of IA systems functions that applies across the NCE without regard to mission space. These functions are the building blocks for achieving the IA controls outlined in DoDI 8500.2, *Information Assurance (IA) Implementation* (6 February 2003), and they must be integrated into products and services developed as IA solutions.

The IA Technical View describes the IA technologies and standards that are available or necessary to support the NCE, as well as potential implications of introducing new technologies. When fully mature, this view will provide insight into technology required to achieve the end-state capabilities of the IA Component.

The Road Ahead

To realize Increment 1, further definition and planning are needed for the proposed activities outlined in the GIG IA Transition Strategy. NSA's Enterprise Systems Engineering (ESE) efforts must continue to analyze architecture alternatives, assess technology and risk considerations, develop implementation-level requirements and guidance, and recommend appropriate standards for the Increment 1 capabilities. This work will serve as input for the GIG IA Portfolio, the DoD's Enterprise Wide Systems Engineering (EW SE) activity, and the Combatant Commands/Services/Agencies (CC/S/A) Enterprise Architecture and system-specific developments. It will also provide IA technical input for the future development of GIG/Net-Centric JCIDS documentation, including ICDs, Capabilities Development Documents (CDD), and Capability Production Documents (CPD).

The Version 1.1 IA Architectural Views are reference models intended to enable GIG NCE programs and CC/S/As to consistently determine how IA activities, systems functions, and technology should be represented in their integrated architectures. Additional work under the ESE effort is necessary to mature and socialize the IA Architecture content and develop its supporting DoD Architecture Framework (DoDAF) artifacts.

Further IC participation in Increment 1 EW SE efforts and Increment 2 definition is encouraged and essential. NSA will continue to engage with CC/S/As in the analysis of Increment 1 IA capabilities and development of implementation-level guidance.

Because NetOps requires the close integration of IA management operations with the management of all GIG Components, NSA will continue to work with the Joint NetOps Architecture Working Group to integrate and harmonize the IA Operational Architecture with the broader NetOps operational activities. This will ensure that management of the IA aspects of GIG Components is defined consistently with the IA Component of the GIG Architecture—and that the IA portion of the architecture will be augmented as necessary to conform to the overall NetOps framework.

Together, the GIG IA Architecture and EW SE efforts represent a significant step forward in developing an IA strategy, technical framework, and transition plan that will help operationalize an assured GIG. These efforts will also potentially enable an assured, federated National Security Enterprise through the systematic and coordinated incorporation of IA throughout the Enterprise's constituent architecture, systems engineering, standards, policies, requirements, and guidance documents and processes. ■

About the Author

Craig Harber | leads the Enterprise Information Assurance (IA) Systems Engineering Services Office in the Information Assurance Directorate at the National Security Agency (NSA). His office is responsible for the development, evolution, and management of the IA Component of the national security Information Technology (IT) enterprise architectures for DoD's Global Information Grid, Intelligence Community, NSA/CSS, and the Department of Homeland Security. Additionally, his office supports clients in developing IA strategies, performing analysis, and defining IA requirements, standards, policies, technologies, capabilities, and guidance required to achieve assured IT enterprises. He may be reached by telephone at 410.854.7069, or by email at *gigia@missi.ncsc.mil.*

Outweighing the Virtual Risks

by Allan Carey

The introduction of any new technology inevitably introduces new security risks and associated concerns. Virtualization is no exception. Regardless of your industry or perspective, or whether you are in the public or private sector, virtualization will significantly affect how information technology is architected and managed in the future. Despite new risks introduced by virtualization, such as hackers attacking VMware (virtual machine) servers, virtualization remains a tremendous success for information assurance (IA).

Virtualization is an important achievement for several reasons. First, it offers significant security benefits. Virtualization solves important operational and deployment problems, and it simplifies certain processes, such as patching, segmentation, and access control. Second, virtualization attacks usually do not occur until all other defenses have failed. For the most part, the threats that exist only come to fruition when someone has compromised the kernel.

The structure of virtualization involves a virtual "guest" using a shared resource from a real network. The virtualization landscape consists of—

- **A guest**—This VM guest runs a virtualized or emulated operating system, such as Windows XP or Linux, inside VMware or Xen (a free software virtual machine monitor). The system thinks it is speaking to hardware, but it is not.
- **A "hypervisor"**—In a virtualized environment, the hypervisor is the host. It consists of shared resources, such as computing, disks, memory, and network switches.
- **The real environment**—The real environment is the real physical network and computer.

Security issues can occur at the boundaries between the different layers. Another way to think about the landscape is in terms of rings. Typically, a first ring has databases and web servers. A ring below that (referred to as "ring 0") is the kernel. Virtualization introduces a new ring below ring 0, which includes a shadowed state, the hypervisor, and a control state. Virtualization also introduces a new type of threat between the kernel and the new ring below it.

To date, there are no hypervisor rootkits in the wild. The two hypervisor rootkits that have been developed in the lab are detectable, despite claims that they are not. One such rootkit is the Blue Pill developed by Joanna Rutkowska. The rootkit can be detected by several techniques, including chipset features, cache timing, and direct timing. For example, one proposed method for combating timing attacks is by dedicating cryptographic hardware to the secured virtual machines that are not vulnerable to timing attacks from secured virtual machines. Another cost-effective measure for protection against exploit is proper segmentation.

From the perspective of a hacker, hypervisor malware is a worse place to hide malware than the kernel. The result is that virtualized malware is not currently a threat. Thus, overall, virtualization is a success for information assurance because its security benefits far outweigh its current risks. ■

About the Author

Allan Carey | is the Senior Vice President of Research and Product Development at IANS. In this position, he manages all research and intellectual property across the Institute. Prior to IANS, Mr. Carey spent seven years at IDC, a global provider of market intelligence and advisory services for the IT sector. He developed and managed the Security Services practice and provided in-depth analysis, intelligence and consulting on key aspects of the information security and business continuity services markets. He may be reached at the Institute for Applied Network Security, 15 Court Square, Suite 1100, Boston, MA 02108, by telephone at 617.399.8100, or by email at *acarey@ianetsec.com*

GIG Performance Assessment Framework

by Julie Tarr, Tony Modelfino, and George Case

Abstract

This article discusses the Global Information Grid (GIG) Performance Assessment Framework (PAF) developed by the Office of the Secretary of Defense for Networks and Information Integration (OSD NII) to evaluate end-to-end (E2E) application and service performance across the GIG, particularly to the tactical edge. The article describes the use case-based strategy developed to define GIG operational scenarios and describes the simulation models developed to predict E2E performance. In addition, it details the Performance Evaluation Tool (PET) developed to allow rapid assessment and parametric analysis of GIG performance. The article also describes the function of the GIG Performance Working Group and PAF pilot efforts.

Introduction

The Office of the Secretary of Defense for Networks and Information Integration (OSD NII) developed the Performance Assessment Framework (PAF) to evaluate Global Information Grid (GIG) end-to-end (E2E) performance and ensure E2E performance meets end-user expectations and needs. OSD NII developed the PAF to support the GIG Enterprise-Wide System Engineering effort to both identify GIG performance shortcomings and to serve as a methodology and tool to evaluate the effectiveness of E2E solutions. The PAF goal is to present E2E performance in metrics that end users can readily understand and evaluate, such as service or application availability and response time. These metrics are in sharp contrast to the packet-level performance metrics typically used to categorize GIG transport segment performance.

The PAF is required for a number of reasons. First, individual GIG transport development programs tend to focus on the performance of a single network and typically do not evaluate E2E performance across multiple networks. Therefore, segment engineering design decisions may be made to optimize intra-segment performance, without recognizing their impact on E2E performance across multiple segments. Second, GIG application and services development programs typically do not consider the full range of transport network performance—particularly tactical edge networks—when developing new end-user applications. As a result, tactical users may experience degraded application performance because of low bandwidth, high delay, and high packet loss, which often characterize tactical edge networks. Third, GIG component development programs seldom consider interactions between all of the layers of the data plane and control protocol stack when evaluating segment performance. For example, GIG transport programs validate segment performance based on segment-level packet performance rather than E2E application performance, which can result in misleading E2E performance and capabilities estimates.

Originally, the PAF was envisioned as a strategy for assigning portions of an E2E performance target to individual GIG segments and sub-segments. It became apparent that this top-down allocation strategy could not succeed for a number of reasons. First, there was no definitive acceptable E2E performance threshold for GIG applications and services. Second, it was impossible to allocate portions of E2E performance to GIG segments because segments do not specify performance using these metrics. For example, transport segments define and assess segment performance using packet delay and loss characteristics, not message delay. Similarly, services and application programs specify performance as measured at the Local Area Network (LAN) or Defense Information Systems Network (DISN) core interface, not at the tactical edge. Even if such an allocation were possible, a strategy for determining the most practical and cost-effective allocation between segments or programs does not exist. Third, GIG segment performance—particularly transport performance—is constrained by inherent physical and technical limitations *(e.g.,* satellite propagation delay and rain attenuation). PAF segment and sub-segment allocations might not be physically or technically possible given

these inherent constraints. Finally, even if all of these issues could be resolved or mitigated, it would not be practical to define and impose a large number of new requirements on existing development programs. The objective of the PAF process is not to optimize E2E GIG performance; therefore, the focus of the PAF shifted from performance allocation to performance assessment. Tools were developed that allow GIG segment developers and operators to evaluate the impact of their segment's (or segments') performance in an end-to-end context. This approach identifies shortfalls early in the requirements generation or technology development cycle and provides a way to assess the success of new requirements or design modifications aimed at correcting the shortfall.

The PAF assesses performance by defining a comprehensive set of E2E use cases that span the full spectrum of GIG user types, applications, networks, and service architectures. In addition, the PAF defines a set of operating conditions and a segment performance categorization strategy that are consistent with GIG segments' approach to evaluating and specifying performance. These definitions minimize the effort required from GIG segments to participate in E2E performance assessment by taking advantage of existing segment modeling, simulation, test, or monitored performance data. GIG segment performance data is integrated into a Performance Evaluation Tool (PET). The PET estimates performance for thousands of GIG use cases and provides the capability to rapidly assess the impact of segment performance or architecture changes on E2E use case performance.

The PAF and PET are intended for use by segment developers, planners and operators, and end users as the GIG capability evolves. Segment developers can use the process to assess the impacts of other GIG segments on their segment performance. They can also evaluate the impacts of performance changes on their segment on E2E performance. GIG planners and operators can use the process to evaluate the impacts of service architecture decisions on E2E performance, particularly for tactical users. End users can receive an accurate estimate of service and application performance to determine GIG impacts on mission performance and mission effectiveness. In summary, the PAF is meant to be an iterative process with multiple feedback loops. These feedback loops ensure that use cases are representative of critical Department of Defense (DoD) communication requirements; GIG component performance assumptions reflect actual component performance; performance shortcomings are real and warrant correction; and solutions consider the impacts on all GIG developers, operators, and users in a clear and transparent process. Because the PAF is an evolving process, a Performance Working Group (PWG) was initiated to bring users, developers, and operators together to ensure the PAF accurately represents and assesses segment performance and user needs. The following sections describe the development of GIG use cases, the E2E GIG modeling strategy, the tools developed as part of the PAF process, and the work of the GIG PWG.

Global Information Grid Use Cases and Performance Evaluation Tool

Ideally, application performance should not depend on the user type, user location, or networks involved. However, in practice, it does. GIG networks and access technologies have inherent bandwidth, latency, and loss characteristics that affect application performance. The PAF recognized that these limitations made it impossible to define a single application performance objective for all GIG users across all networks. Instead, performance must be assessed for each user type, access technology, and network connectivity. To this end, a GIG use case generation strategy was developed. Each use case comprises a GIG user or users, GIG ingress/egress access technology, a GIG transport network or networks connecting users and services, and a GIG service or application. Use cases were initially developed considering the Net-Centric Operating Environment Joint Integrating Concept (NCOE JIC), Net-Centric Enterprise

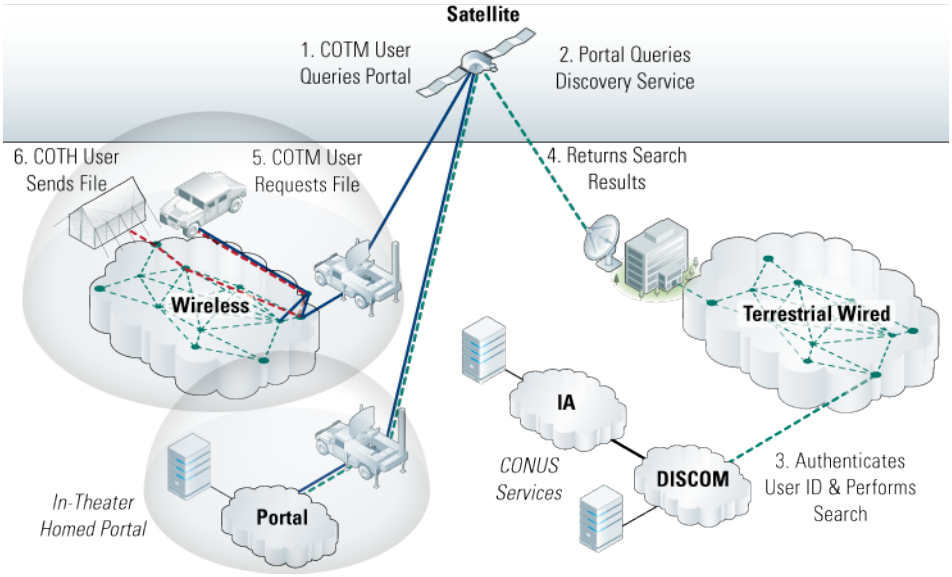


Figure 1 Typical GIG Use Case

Services (NCES), joint mission areas, and multiple transport and application program DoD Architecture Framework (DoDAF) operational views and information exchange requirements (IER). These user and mission requirements clearly show that the GIG is not a homogenous network, but rather a collection of networks with widely different performance capabilities and limitations. Similarly, GIG users span a broad spectrum, including strategic, tactical, and business functions. These users can access the GIG from locations as varied as fixed sites in the continental United States (CONUS) to high mobility multipurpose wheeled vehicles (HMMWV) in Iraq. In addition, these users employ a wide range of applications that will place different demands on the GIG.

Initially, the PAF attempted to identify only the most important GIG use cases in an effort to minimize the number of scenarios that required analysis. It became apparent that this strategy was destined to fail because performance requirements varied significantly across GIG users. Ultimately, the PAF chose to evaluate a broad range of use cases. Currently, more than 5,000 use cases have been identified and analyzed for a variety of service architectures. Figure 1 shows a typical use case. In this example, the commander in a communication-on-the-move (COTM) vehicle queries a portal located at a commander, joint task force (CJTF) for available imagery. The COTM terminal connects to the beyond line-of-site portal using a satellite network, such as Wideband Gapfiller Satellite (WGS) or Transformation Communications Satellite (TSAT). The wireless network interconnects with the satellite at a satellite communications (SATCOM) point of presence (PoP), while the portal server has a high-speed connection to a teleport. The portal relays the search query to a CONUS datacenter using a satellite connection through a teleport connected to the DISN core (Global Information Grid Bandwidth Expansion [GIG-BE]). The request is relayed to a CONUS datacenter, such as the Defense Enterprise Computing Center (DECC). The datacenter authenticates the user, performs a federated search, and returns the results to the portal, which then relays the results to the commander in the COTM vehicle. The commander downloads the imagery from the closest content delivery server, which in this case is a fixed command post located in theater. This service-oriented architecture allows for greater data dissemination and improved decisionmaking, but also requires additional user-to-service and service-to-service communications.

The objective of the PAF implementation in PET is to identify each communication exchange or processing step; define the transport paths involved and determine the E2E transfer time for each constituent message transfer; and combine the constituent response delays to determine the overall E2E service response time and availability.

GIG users are defined as the senders and receivers of information and can be either people or machines. The current version of PET defines 11 user types based on users' operational mode and technology employed to access the GIG. Table 1 lists these user types. Each user type has distinctly different operational capabilities and performance. GIG connectivity is defined as a network or series of networks that connect senders, receivers, and services. The PET currently includes 15 GIG terrestrial, wireless, and satellite network types. Table 2 shows the current PET network types and networks that have been integrated into the PET. The PAF recognized that significant capability variation exists between network types in a given class. For example, WGS, Advanced Extremely High Frequency (AEHF), and TSAT are each considered satellite networks; yet, significant performance and operational differences exist between these architectures. For that reason, PET defines the performance for each network separately.

PET defines the access capabilities for each user type to each GIG network type. Access metrics include bandwidth, ingress/egress delay and packet loss, and availability. In addition, PET categorizes

User Type	
Dismounted	Aircraft, Tactical
COTM	Aircraft, C2
COTP	ISR Aircraft
COTH	Ship
Fixed—CONUS	Submarine
Fixed—OCONUS	

Table 1 GIG User Types

Network Type	Pet Network
Wireless	JTRS, SRW, JTRS WNW
SATCOM	TSAT (1Hop), TSAT (Multi-hop), FDMA, TDMA, DAMA IP Modem, L-Band mobile
Wired	Intra-Theater GIG-BE, Inter-Theater GIG-BE, Intra-theater PTP, Inter-Theater-PTP, NIPRNET, SIPRNET

Table 2 GIG Network Types

the performance of each network type as a function of Internet Protocol (IP) service class. PET defines transport network performance for five IP service classes that are consistent with the Net-Centric Implementation Document (NCID) T300 Service Class segmentation. Network performance metrics include packet loss, packet delay, and transport segment availability. Finally, PET characterizes the performance of interconnection nodes used to connect networks, such as gateways, terrestrial and wireless PoPs, and teleports. Ideally, the performance attributes for these elements comes directly from the appropriate GIG segments, although it may come in various forms. For example, GIG transport programs have provided availability, packet delay, and packet loss rate performance data based on engineering requirements, modeling and simulation data, service-level agreement requirements, and/or testing or operational network monitoring data.

GIG user services are applications or sets of applications that users execute over the GIG. PET currently includes more than 30 different applications ranging from legacy applications to Net-Centric service-oriented applications such as collaboration and discovery. Table 3 summarizes these applications. The legacy services represent the services, from a bandwidth perspective, that dominate the GIG today (based on evaluation of Secret Internet Protocol Router Network [SIPRNET] and Non-Secure Internet Protocol Router Network [NIPRNET] network monitoring performed by OSD NII). These services also represent a significant portion of traffic identified for future tactical and satellite network programs. Net-Centric services are currently modeled using the NCES service descriptions as described in the NCES Performance Specification. PET developers recognize that additional services will be used over the GIG (particularly specialized community of interest [COI] services); therefore, PET was designed to be readily expanded to include additional services. PET application/service models seek to identify the messaging requirements and attributes for each GIG service based on service OV-6C event sequence diagrams or on analysis of application packet trace data. Each messaging event is decomposed into a series of standard event building-block components, which are then further decomposed into standard networking protocols and messages.

GIG use cases can be complicated, involving multiple user-user, user-service, and service-service communications as shown in Table 3. In addition, a single use case can involve multiple nodes and multiple networks. The number of possible GIG use cases grows as more user types, network types, and applications are added to the analysis. The current PET combination of GIG user types, composite networks, applications/services, and service architectures can potentially generate more than 1 million use cases. Fortunately, reasonable operational assumptions reduce the number of use cases significantly, although the number is still about 5,000 for a given service architecture. The objective of PET is to generate these use cases from a small number of user inputs automatically. PET manages a process that allows the user to rapidly select these parameters, generate use cases, estimate E2E performance, and evaluate that performance relative to end-user requirements. PET presents the results in a manner that allows the user to identify the messaging or processing events most responsible for a performance shortfall.

Legacy Applications	SOA Applications
VoIP	Discovery
VTC	Collaboration
Sensor Streaming	Enterprise Messaging
HTTP	Security Services
FTP	Content Delivery
Instant Messaging	Net Management
Email	Mediation

Table 3 Examples of GIG Applications in PET

A typical GIG service or application is not a single message, but rather a series of messages between users and services. Figure 2 shows the event sequence (ES) diagram for an audio collaboration session. PET application models decompose each application/service into its core messaging and processing steps. PET then adds an appropriate series of standard and optional event building blocks to replicate protocol interactions and to reflect the state of the network and the user. Table 4 shows examples of event building blocks used in PET. These building blocks define a series of user-user, user-service, and service-service messages or transactions. Figure 2 shows the number of messaging events for each building block. For example, the initial Transmission Control Protocol (TCP) handshake between the call-initiating user and the audio collaboration portal requires two messages (SYN, SYN-ACK). The authorize and authenticate user building block requires 15 messages between the audio collaboration portal and multiple security servers. A PET use case model ultimately defines a sender, receiver, network path, message size, and service class for each message or transaction. The message transfer time for each message is then determined using the

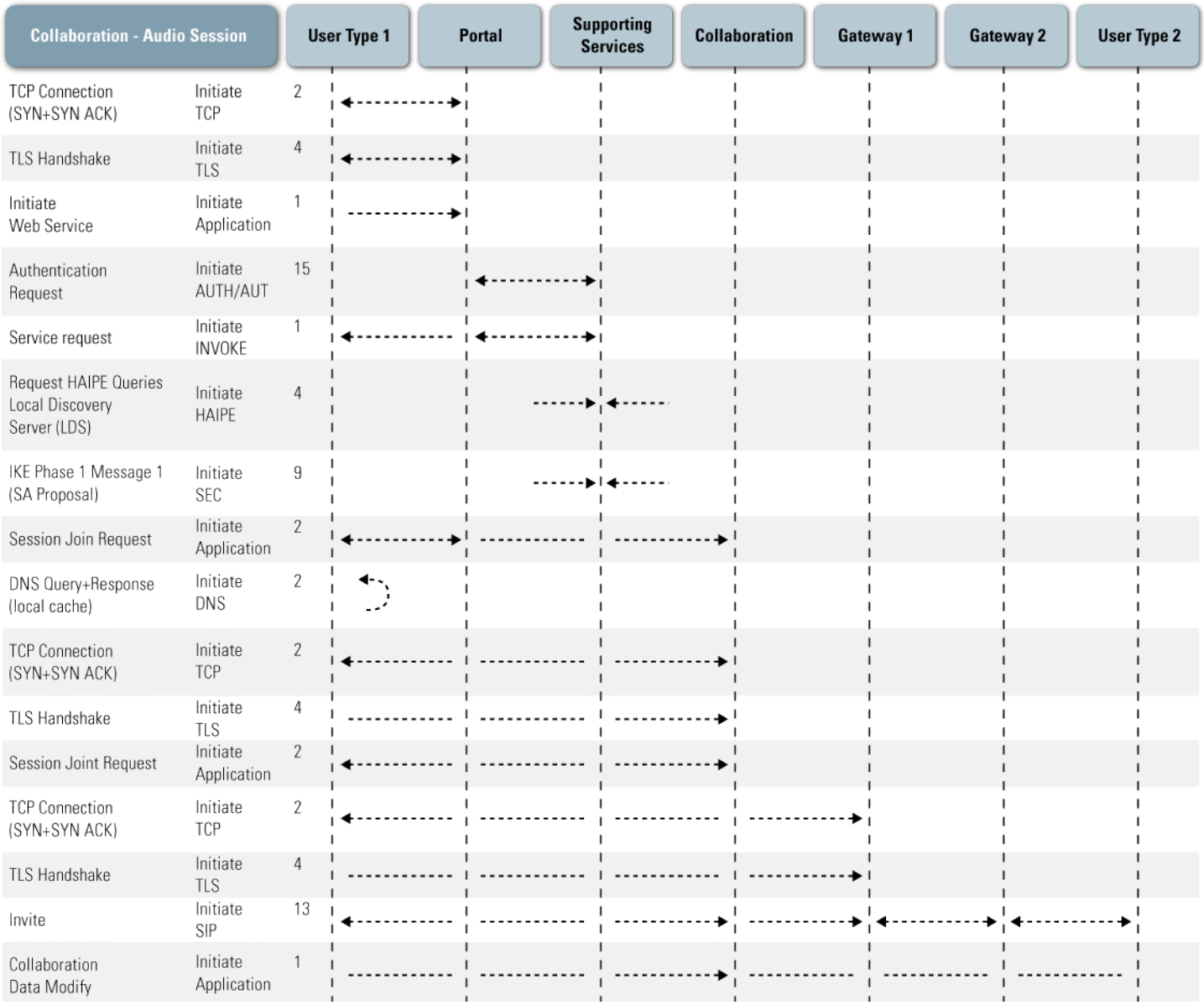


Figure 2 Typical GIG Use Case (abridged)

Event Building Blocks	
HAIPE Discovery	TCP Setup
Security Association	TLS Handshake
Authentication/ Authorization	Directory Lookup (LDAP)
Web-Service Invocation	DNS Query
Session Initiation Protocol (SIP)	HTTP Request/ Response
SMTP Setup	

Table 4 Sample Building Block Protocols

performance database. The PAF recognizes that the need for many of the building-block events will depend on the state of the user nodes and the network. For example, Domain Name Server (DNS) and High Assurance Internet Protocol Encryption (HAIPE) discovery may not be required if the user node is aware of the IP addresses for all services and users involved in the service or application. Therefore, PET provides an interface that allows the PET user to determine which building blocks must be executed for each service or application.

Total application response time and service availability are determined by combining the performance for each constituent message, taking into account that some messaging events occur serially while others are executed in parallel. The collaboration service requires 68 messages to initiate the service. Most of the messages involve a single small packet, such as a TCP handshake, which includes three 40-byte messages. Although the service-to-service messages are typically transported over a high-speed Wide Area Network (WAN) (GIG-BE) or a LAN if the

servers are located in the same data center, the user-to-service messages can experience significant delay and packet loss depending on the path. PET defines two response-time metrics for a service or application: service initiation time and service execution time. In the audio collaboration example, the service initiation time includes all of the delays before the first voice packet is transmitted, while service execution response time is the one-way Voice over Internet Protocol (VoIP) packet delay.

The PAF recognized that the large number of GIG use cases coupled with the wide array of GIG segment design and performance parameters requires an automated process to generate and assess use case performance. The PAF developed PET to serve this purpose. The PET was built in Excel to enable distribution beyond the GIG modeling and simulation community to the larger GIG system engineering and program engineering community. PET has a graphical user interface (GUI) that enables the user to analyze a single GIG use case or the full set of GIG use cases; add additional GIG users, networks, or applications; and automatically generate new use cases. A portfolio manager interface allows the PET user to select networks for inclusion in the use case evaluations and to create composite networks using any combination of these networks. Currently, PET allows the user to select up to six different networks from which it generates E2E network combinations. Typically, the six networks are combined to form between 20 and 30 composite network architectures. The PET user also defines the user types of interest and the user-to-user connectivity using these composite networks. In practice, this can generate more than 400 user-user and user-service connection paths.

PET includes output post-processing features that can identify performance shortfalls based on specified minimum performance thresholds for each GIG service, application, or use case. The tool includes target application performance thresholds derived from the NCOE JIC, program message/application speed of service requirements, and industry-standard performance thresholds. PET compares the performance of each use case to the specified threshold to identify performance shortfalls and allows the tool user to drill into any use case to isolate the cause of poor performance. The PAF built the PET to serve a wide range of purposes, and the software is available to the GIG user, developer, and operator communities. GIG end users can use the tool to estimate E2E performance relative to end-user mission requirements. Both transport and services GIG segment developers can use the tool to investigate the sensitivity of E2E performance to segment-level performance and to overall service architecture. The tool supports CONUS and global fixed-site centralized service architectures, as well as in-theater and forward-deployed decentralized service architectures.

Global Information Grid End-to-End Performance Modeling

The objective of the PAF network modeling is to estimate E2E application performance for each GIG use case based on the performance of the individual GIG components involved. The GIG performance modeling strategy needed to strike a balance between accuracy and calculation complexity. Given the wide range of GIG applications and network types, many of which are still in development, a single E2E GIG model that includes all network and application features does not exist. In addition, although various GIG segments have developed a variety of modeling and simulation tools to assess segment-level performance, it is not feasible to integrate these tools in the short term. A long-term modeling strategy for modeling the GIG should be to develop an integrated E2E model that combines program-supported GIG segment models. The success of this approach hinges on selecting a standardized core simulation model with which to integrate each of the segment models, and developing a standardized set of model interfaces that enables easy integration of these segment models. Fortunately, most GIG segment models are built using an OPNET core simulation model, which makes an integrated model possible. PWG discussions made it clear that it is not practical to integrate GIG segments models (particularly those of next-generation GIG segments) in the near term; therefore, the PAF decided to implement a short-term network modeling strategy that can predict E2E use case performance using GIG segment model outputs.

The short-term modeling strategy simulates GIG network connectivity as an IP cloud whose performance is defined by the IP packet delay, loss, and network availability of its constituent segment networks. The segment delay characteristics are assumed to have an offset gamma probability distribution function (PDF). The decision to use this delay distribution was made after considerable analysis of simulated and monitored network delay performance provided by GIG programs and measurements taken for commercial IP networks. The delay distribution for each GIG transport network was generated for each service class using the minimum delay, average delay, and delay variance provided by GIG network segments. The gamma distribution has the heavy tail characteristics typical of network congestion—particularly wireless bandwidth-on-demand networks. However, a standard gamma distribution would generate packet delays that vary randomly from packet to packet. As a result, one packet might experience a long delay while the next packet may experience a significantly shorter delay. This behavior is generally not consistent with measured and simulated packet delay performance for messages comprising multiple TCP transmission segments. In fact, E2E delay for packets associated with a common flow tend to be highly correlated because these packets typically follow the same network path and experience similar

network delay. The PAF delay model incorporates this behavior into the end-to-end delay model by correlating packet delay for individual messages or flows. This delay modeling strategy produces heavy-tailed packet and message delay distribution behavior characteristic of DoD networks.

The key to developing an interactive performance assessment tool was separating the packet-level simulation modeling from the PET. This was accomplished by developing a large performance database of message delay performance results generated using an OPNET-based, packet-level IP cloud simulation model. The database contains E2E message transfer time performance as a function of network ingress/egress load, message size, service class, E2E delay, E2E packet loss, and access bandwidth. In addition, the database includes performance for multiple TCP implementations. The database was generated by simulating multiple nodes for thousands of simulated seconds. A single simulation typically generated more than 100,000 messages and 10 million IP packets. The results of the simulations were processed to generate a statistical distribution of packet and E2E message transfer time, and these results were stored in the PET database. The PET model determines E2E message transfer time performance for any composite network by interpolating between entries in the performance database. The model allows PET users to modify the delay, loss, load, and bandwidth for any network, or the class of service or size of any message, and still predict E2E performance.

Performance Working Group and Pilot Efforts

The PWG was instituted in the spring of 2006 as a mechanism for refining and updating the PAF. The team coordinated with transport, services, and infrastructure developers, and it included members from a broad range of DoD agencies. The PWG met on a monthly basis for 4 months to review the network modeling strategies, refine GIG use cases, agree on GIG operating assumptions, develop a strategy for categorizing and obtaining GIG segment performance, and review GIG E2E performance results. PWG members also met regularly with NCID working groups (including the QoS, Services and Computing, and Infrastructure Working Groups) to ensure the compliance of PAF models with the NCID compliance requirements. The results of the PWG were captured in the GIG Performance Assessment White Paper V3.0, which was released in October 2006.

Upon conclusion of the first set of PWG meetings, a pathfinder pilot effort was initiated with a number of GIG development programs, including TSAT and Joint Tactical Radio System (JTRS). The efforts with the JTRS program illustrate the type of working relationship that makes the PAF successful. The JTRS system engineering team worked closely with the PAF team and provided performance requirements, radio test data, traffic models, and simulated performance data. The PAF team incorporated this data into existing wireless radio simulation models to predict JTRS E2E performance over a wide range of scenarios, which varied link bandwidth, link loss, coverage area, subnet size, and operating load. The objective of this pilot was to obtain performance data for JTRS and validate the PET model. In particular, the effort sought to—

- Validate the accuracy of the IP cloud gamma delay distribution model. Results have shown that the RMS error associated with this delay model is less than 10 percent for a broad range of operating conditions and network architectures.
- Determine network, link, and architectural configurations impacts on packet delay and E2E performance. The objective of this analysis was to determine the number of configurations required to represent JTRS network performance in the PET accurately. Results have shown that a small number (1–3) of configurations typically bracket segment performance.
- Evaluate the accuracy of the IP cloud model for predicting E2E performance. A comparison of the simulated E2E performance predicted using the IP cloud model and more accurate segment models shows differences of less than +/-25 percent between the two models for a broad range of network types, message sizes, packet delays, and packet losses. This accuracy is more than adequate for PET purposes.

The PET team is initiating a broader pilot effort in FY08 to obtain segment performance for additional GIG transport, service, and application programs. This data will be added to the PET, and use cases will be expanded to include these segments. In addition, the PET development team is evaluating upgrades and improvements that have been suggested for PET, including developing a mission modeling capability to link service/application performance thresholds to mission requirements; upgrading the model to include a broader range of operating loads and background traffic; and developing additional operating conditions such jamming, On-the-Move blockage, and Denial of Service (DoS) attack. ■

About the Authors

Julie Tarr | is a Senior Systems Engineer in the GIG Enterprise-Wide Systems Engineering Office in OSD(NII)/DoD CIO. Ms. Tarr led the development of the GIG Technical Direction, the NCID, and was a principal member of the team developing the GIG Technical Foundation. She is now leading the development of the Enterprise Engineering policy to implement the GIG Technical Foundation throughout the Department. Ms. Tarr led the GIG Routing Working Group defining the routing interoperability architecture for the GIG. Previously, Ms. Tarr was the Head of the

Secure Network Section at the Naval Research Lab which performs research, development, and support to the Navy in the areas of network security, intrusion detection, cross domain solutions, and networking requirements for encryption devices. Ms. Tarr is a member of the NATO Task Group on NEC Security and the TTCP Technical Panel on IA. Ms. Tarr received her BS in Electrical Engineering and Mathematics from the University of Maryland, College Park, in 1985. As a Naval Research Laboratory Fellow, Ms. Tarr received her MS in Electrical Engineering in Communication and Control Theory from the University of Maryland, College Park. She may be reached at *julie.tarr@jhuapl.edu*

Tony Modelfino | is the president of Stratogis Network. Prior to founding Stratogis in 2002, Mr. Modelfino spent three years as the vice president of Services and Applications at Astrolink International, where he was responsible for all product engineering. Mr. Modelfino began his career in engineering at Lockheed Martin in 1984; he left Lockheed Martin in 1986 to become a founding member of Atlantic Aerospace Electronics Corporation (currently a division of L3 Corp). In 1995, he left Atlantic to become

▷▷ *continued on page 31*

SUBJECT MATTER EXPERTS

Mary Linda Polydys

By Ron Ritchey

This article continues our profile series of members of the Information Assurance Technology Analysis Center (IATAC) Subject Matter Expert (SME) program. The SME profiled in this article is Mary Linda Polydys. Ms. Polydys has been the chair of the Information Operations and Assurance Department of the National Defense University's (NDU) Information Resources Management College (IRMC) for more than three years. She came to the position after serving as the chair of IRMC's Systems Acquisition Department (now the Systems Management Department). Under her guidance, IRMC has grown 30–40% and has implemented several unique learning opportunities, including the creation of an extensive—and mobile—laboratory to provide *experientattachedial learning*. This mobile lab gives students an opportunity to experience multiple security technologies and situations firsthand. Topics covered in the lab include biometrics, firewalls, intrusion detection, Supervisory Control and Data Acquisition (SCADA) system vulnerabilities, and computer forensics.

Ms. Polydys earned her master's degree in information systems from George Mason University (GMU). She also holds a bachelor's degree in decision sciences. She is designated as a Level III Certified Department of Defense (DoD) Acquisition Professional in both contracting and information technology. Ms. Polydys is currently pursuing a PhD at GMU, where she is researching ways to measure the knowledge level of acquisition personnel regarding software assurance requirements. She hopes these tools will allow organizations to improve the quality of software assurance by ensuring the inclusion of software assurance in software requirements.

Ms. Polydys is active in the software assurance community. She serves as a co-chair of the Department of Homeland Security (DHS) and DoD-sponsored Software Assurance Acquisition Working Group. As part of her work with the DHS/DoD Software Assurance initiative, Ms. Polydys served as one of the authors of the Software Assurance Common Body of Knowledge. [2] As co-chair of the Acquisition Working Group, she and Stan Wisseman developed Software Assurance in the Acquisition Process: Mitigating Risk to the Enterprise. [3] In developing this literature, she and Stan worked with 50–75 individuals from industry, academia, and government in the United States, as well as contributors from Canada and Australia.

If you have a technical question for Ms. Polydys or another IATAC SME, please contact *http://iatac.dtic.mil/iatac*. The IATAC staff will assist you in reaching the SME best suited to helping you solve the challenge at hand. If you have any questions about the SME program or are interested in joining the SME database and providing technical support to others in your domain of expertise, please contact *iatac@dtic.mil*, and the URL for the SME application will be sent to you.

References

1. Professor Polydys' home page at NDU is available at *http://www.ndu.edu/IRMC/ia/polydys.html*
2. Software Assurance Common Body of Knowledge is available at *https://buildsecurityin.us-cert.gov/daisy/bsi/resources/dhs/95.html*
3. Software Assurance in the Acquisition Process: Mitigating Risk in the Enterprise is available at *https://buildsecurityin.us-cert.gov/daisy/bsi/resources/dhs/908.html*

ForNet: Network Forensics for Detecting Stealthy Attacks

by Nasir Memon, Elliot Fischer, and Kulesh Shanmugasundaram

Introduction

Although the first worms released on the Internet were large-scale, easy-to-spot security incidents, future malware will be increasingly stealthy. Some botnets widely spreading on the Internet are difficult to detect. Whereas the motive of the first worm developers was self-gratification gained by compromising a large number of computers, the motives of recent malware developers are financial and political gains. Therefore, recent developers prefer to compromise a smaller number of computers as quietly as possible and over a longer period of time to avoid detection by security defenses. To remain stealthy, these attackers use sophisticated mechanisms, including encryption, metamorphism, and polymorphism. Furthermore, to evade detection, they use explicit hit lists to guide threat spread and avoid known honeypots. They often enter an endhost using malleable delivery mechanisms, such as email and P2P content, and exploit client software, such as Microsoft PowerPoint or Word.

To detect stealthy attacks, observations (data) and data analysis must have the following attributes—

- **Temporal Span**—Stealthy attacks are low and slow by nature. Therefore, any analysis must span data over a long period of time to detect such attacks reliably.
- **Spatial Span**—Stealthy attacks can span multiple networks using stepping stones. Therefore, data analysts must be able to collect data and propagate queries across multiple networks.
- **Foresight**—Detection of stealthy attacks require systems that look for potential attacks before the rest of the world realizes the nature of the attack. Therefore, detection cannot depend merely on signatures of known bad behavior or assumptions about the attack vectors.

Most of state-of-the-art security solutions address only one or two of these attributes. For example, perimeter defense systems, such as firewalls and intrusion detection (prevention) systems, assume knowledge of threats' modus operandi and only collect and analyze data about these specific threats. These systems generally lack foresight or temporal span. Others, especially commercial Network Forensic Analysis Tools (NFAT), take a brute-force approach of recording everything on the network. With such an approach, it is hard to achieve the temporal span needed to detect stealthy attacks.

In the past 3 years, researchers at Polytechnic University have developed a distributed network forensics system called ForNet. ForNet elements, called SynApps, create and save compact synopses of network events for later analysis (such as forensics and botnet detection) that depends on events that occur over a period of time. Synopsizing techniques—such as connection records, lossy counting, and Hierarchical Bloom Filters that store hashes of packet payload segment—are active within each SynApp to represent network events succinctly. Each domain also has a Forensics Server that controls and coordinates SynApps within its domain. Forensics Servers from multiple domains can cooperate to facilitate analysis that spans multiple domains.

ForNet has been deployed on the Polytechnic University campus for three years. It has been extensively tested and used to detect and analyze a broad variety of security attacks, including virus propagation, spyware distribution, malicious proxies, and botnets. ForNet synopses are roughly two orders of magnitude smaller than the raw packet data. Information about ForNet can be found at *http://isis.poly.edu/projects/fornet.*

Data Collection

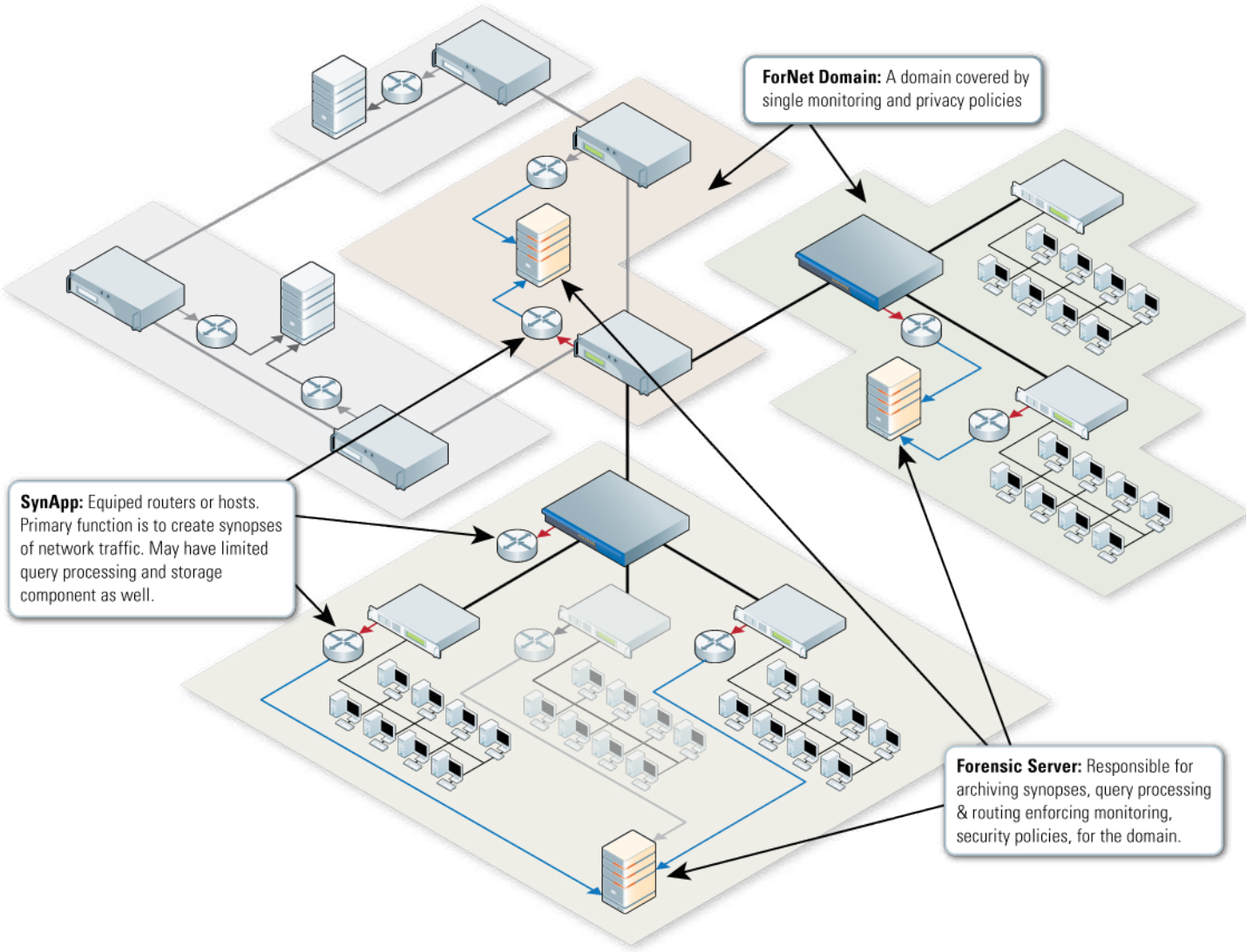
A key aspect of ForNet is the sophisticated set of techniques it employs for data collection. In the next section, we explore the major challenges of collecting network traffic for security analysis and forensics. Motivated by these challenges, we outline the general principles developers adhered to when designing ForNet.

Detection of stealthy attacks require systems that look for potential attacks before the rest of the world realizes the nature of the attack.

What to Collect?

Millions of network events occur every second. Collecting data without prior knowledge of what will be necessary for a future postmortem or analysis is a challenge for two reasons. First, we cannot selectively collect data, such as an Intrusion Detection System (IDS) or a firewall, because doing so would limit the scope of postmortems and analyses. Second, we cannot collect every piece of data because doing so would impose enormous storage requirements on the system. Below are the different types of data available and the challenges of collecting each type—

- **Link States**—Networks are formed by interconnecting a multitude of hosts. The hosts establish links or connections via a variety of protocols at different levels of protocol abstractions. These links can last for varying lengths of time, anywhere from a few seconds to many months. The links can also change over time. Old links may disappear and new links may be established. Therefore, it is useful to track what links to what on the Internet, as well as certain properties of these links. A system designed to support forensics should keep track of end-to-end links and hop-by-hop links.
  End-to-end links are established at transport and upper-level protocols that reveal which hosts are connected to which other hosts. This information can be inferred from network protocols, such as from a Transmission Control Protocol (TCP) connection or from a User Datagram Protocol (UDP) "connection."
  Hop-by-hop links are established at the infrastructure level and can help us determine which networks are connected to which other networks and their physical proximity. These links can be inferred from routing protocols, such as Border Gateway Protocol (BGP) and Open Shortest Path First (OSPF).
- **Link Content**—Links are established between hosts to carry a variety of content. Links may carry anything from audio streams to routing updates. Content traversing these links is the most useful source of evidence in any postmortem analysis. Ideally, a forensic system should capture and archive every single packet that traverses a link. However, when keeping raw packets is not feasible, the system may instead decide to keep what it perceives at the time as sufficient evidence for a postmortem. For example, instead of keeping raw packets, the system may infer the type of application using a link (such as Kazaa or Bittorent) or the type of content transferred over the link (such as audio or encrypted streams). Another approach is to keep hashes of content, which would then allow one to determine if some known content traversed a link.
- **Link Aggregates**—A lot of information about network behavior can be gleaned by examining link aggregate information. Network devices can generate these aggregates in the form of Simple Network Management Protocol (SNMP) statistics, or monitoring network links can gather these aggregates. The system can keep track of useful statistics about links, such as protocol types, amount of data transferred, number of packets transferred, and length of the link (in time).
- **Mappings**—To make heterogenous hosts on the Internet interoperate transparently, protocols and applications use many aliases or protocol mappings. An example of such a mapping is the Domain Name System (DNS), which maps a user-



ForNet Architecture

friendly domain name such as *isis.poly.edu* to a 32-bit Internet Protocol (IP) address such as 0x80EE400F. These mappings often change with time; therefore, a forensic system must keep track of these mappings to be able to find the correct host at a later time. These mappings usually fall under the following three categories:

- ***Protocol Mappings***—Various network protocols use protocol mappings to talk to each other or to translate between the mappings from lower layers to upper layers or vice versa. Examples of protocol mappings include Media Access Control (MAC) addresses and DNS names that map corresponding addresses or names to IP addresses. Other examples include IP multicast mappings, where one multicast IP address may map to a group of IP addresses, and Network Address Translation (NAT), where one IP address on one side of a network interface is mapped to one or many IP addresses on the other side.
- ***Application Mappings***—These mappings are unique to a particular application. The application uses them to improve scalability, reliability, or efficiency. Examples of application-level mappings include VirtualHost of HTTP and various routing protocol mappings in peer-to-peer networks, such as KeyId in FreeNet.
- ***Administrative Mappings***—These mappings are created and maintained for network or host administrative purposes and are generally not enforced by a protocol. An example of administrative mapping is mapping Autonomous System Numbers (ASN) to an IP address as assigned by Internet Assigned Numbers Authority (IANA).

How to Collect?

Once we decide what data to collect, we need a strategy for collecting this data properly. The type of data collected partly determines the collection strategy. For example, the centralized collection of network data at a network's traffic concentration point maximizes the visibility of the network's interactions with the rest of the Internet. Currently, this is the most popular strategy used to deploy network monitors. This centralized collection strategy, however, has two major flaws: visibility and correctness.

- **Visibility**—A centralized collection point does not see all of the packets traversing a network. Because each subnet of a network is usually a broadcast domain, and a switch (or a hub) does not leak packets sent within a domain, a collection point outside of a subnet does not see packets shared within the subnet. Hence, a centralized collection strategy lacks event visibility. Furthermore, even though a packet sent from a host in a subnet to a host on the Internet is visible to a centralized collection strategy, the MAC address of the host that sent the packet is not visible. Hence, a centralized collection strategy lacks data visibility. Postmortems require that a forensic system have both data and event visibility. For example, suppose a disgruntled employee decided to take revenge on the employer by installing a Trojan horse on critical systems. A postmortem of this incident would require elaborate information on network connections established within the organization so that an investigator could reconstruct the modus operandi for evidential and recovery purposes. A collection point at the edge of the organization's network would not be useful for such a postmortem.
- **Correctness**—A centralized collection point sees only one instance of an event, rather than the "big picture." The lack of integrity checks in the Internet protocol suite makes it impossible to infer the "big picture"' from a single observation. Because source IP addresses can easily be spoofed, only a hop-by-hop verification of the packet's passage can reliably identify the origin. Therefore, a proper collection strategy should be distributed throughout the Internet with full event and data visibility. The system should collect data from multiple points so it can corroborate the correctness of one observation with observations from many other points. The system should also provide the necessary information for an investigator to construct the big picture. A distributed system also creates additional challenges. The system must coordinate its data collection operation to avoid duplicate collection. A good system design must also include fault tolerance and redundancy to minimize the impacts of failures.

Deployment and Usage

ForNet was designed and developed to collect all of the data types discussed in the previous section efficiently. At the time of this writing, ForNet had been operational at Polytechnic University for more than two years. It has been used for network monitoring, troubleshooting, and retrospective analysis of incidents, such as denial of service attacks. It has identified frequent scanners of the University resources, bandwidth hogs, and service outages. ForNet's unique features make it especially useful for detecting certain types of attacks that other systems cannot detect with comparable ease or efficiency. The following are a few examples.

BotNet Detection

Based on ForNet, BotSentry is a novel botnet detection, discovery, and mitigation system. BotSentry is independent of signatures. It focuses on detecting the key symptoms of a bot infection that are not specific to a particular botnet but generalize over a broad set of botnets. For example, a bot-infected host may communicate via Internet Relay Chat (IRC) channels or P2P networks for purposes of command and control. An infected host may exfiltrate documents or perform reconnaissance activities in the internal network. The common characteristic of these activities is that they can be detected through a careful and thorough analysis of network traffic.

The following is a set of symptoms observed from a bot recently identified by BotSentry. The Command and Control Module (CCM) in most bots today is a modified IRC client. Once a host is infected, the CCM uses DNS to find its master on the Internet and establishes a Command and Control Channel (CCC). In doing so, a CCM exhibits the following symptoms—

- **Attempts to Access Non-Existing Hosts**—Set of DNS requests to which no IP addresses exist
- **Protocol Semantics Violations**—Connection to a host without a corresponding DNS
- **Interactive Session**—Flows exhibiting the characteristics of an interactive session indicates a CCC
- **Contact With a Mule**—A mule is a host that serves worker modules to bots.

Worker Modules (WM) in a botnet can be programmed to accomplish a variety of tasks. Each type of WM, however, has a set of unique symptoms that BotSentry uses to identify the purpose of a bot in a botnet, such as a spammer, phisher, or scanner. For example, a WM built to spam exhibits the following symptoms—

- **Protocol Semantics Violations**—Set of connections without corresponding DNS requests

- **Change of Host's Role (to Mail Server)**—A host resolves DNS MX records and sends email.

Note that the symptoms remain the same regardless of malware variations, targeted operating systems, or protocols used. Detailed analysis of synopses yields reliable symptoms for each infected host. BotSentry pitches the resulting symptoms against information about a host's environment to pinpoint the bots and any infected host. The approach has been tested and deployed on the Polytechnic production network, where it has discovered numerous hosts belonging to botnets.

Stealthy Attacks

Stealthy attacks, also known as low and slow attacks, attempt to evade detection by spreading out their activity over time. For example, the scanning phase of an attack could spread out over a week or more to evade detection. Using the HBF technology, ForNet can save months of synopsized data on a realistically sized network. For example, the system at Polytechnic University stores 3 month's worth of data for a network of more than 2,000 nodes on a terabyte server. Thus, any behavior that the attacker attempts to spread out over time can be detected. For example, if the scan pattern of an attack is known (*i.e.,* which ports are usually scanned and in what order), For Net can detect the scanning even if it is spread over many weeks. For any attack phase, if the behavior is known, ForNet queries can be constructed to identify the behavior even if it spreads out over time. Attack phases that spread out over time can be correlated to detect the attack—possibly while it is still in a preliminary phase and before the malicious software is activated. Correlation of information across hosts on a network can accelerate the early detection of these attacks.

For example, detecting partial scanning patterns on multiple hosts would raise the probability that the scanning phase is in progress, even though the entire scanning pattern has not been detected on any particular host. The HBF feature could be used to look for communication traffic that would occur, for example, during an attack phase in which the infected host attempted to communicate with a master. If enough of a standard communication message is discovered from one intercepted message from one node, ForNet can search for this data in traffic from other hosts. If found, this data raises the probability of an attack on that host. If the communications are encrypted, ForNet can use that information to find encrypted traffic messages spread out over time from hosts to the master. ForNet might even identify the master by looking for encrypted traffic sent in the last 2 months to a common host outside the network from multiple hosts inside the network. There are many ways ForNet queries can detect stealthy attacks based on correlation of evidence from multiple hosts over long time spans.

Exfiltration Detection

Unauthorized file exfiltration has become a problem for many organizations, and ForNet can detect unauthorized exfiltration in many ways. For example, if enough text in the exfiltrated file is known, a ForNet query to the HBFs can find the file's host and the destination IP address of the exfiltration. If the file is exfiltrated in sections, ForNet can possibly locate the destination address by looking for flows with the same (source, destination) pair, but that were spread out over time. If the exfiltrated files are encrypted, ForNet can detect the encrypted traffic. For example, if a particular host shows more than the usual amount of encrypted traffic to a particular destination over a time period, this could raise a flag that encrypted files are being exfiltrated. Again, there are many ways ForNet queries can be configured and correlated to detect unauthorized file exfiltration. ForNet has already been used on the Polytechnic network to detect both unauthorized proxies and illegal tunnel activity, both of which are examples of file exfiltration. ■

About the Authors

Nasir Memon | is a full-time Professor in the Department of Computer Science and the Director of the Information Systems and Internet Security (ISIS) Laboratory of Polytechnic University, Brooklyn, NY. He is the creator of the ForNet concept. His main research interests include digital forensics, network and computer security, data compression, steganography and watermarking. He is the associate editor of IEEE Security and Privacy, IEEE Transactions on Information Forensics and Security, and several other journals. He has published over 100 technical papers and has a PhD in computer science from the University of Nebraska. He may be reached at *memon@poly.edu*

Elliot Fischer | is a technical manager in the Internet Research Department at LGS Innovations, part of Bell Labs. His current research interests include attack graphs and forensic detection on network attacks. He holds a PhD in mathematics from the California Institute of Technology. He may be reached at *efischer@lgsinnovations.com*

Kulesh Shanmugasundaram | is a post-doctoral fellow at Polytechnic University and the chief architect of ForNet. His research interests include, network security and digital forensics. He received his PhD from Polytechnic University in 2006. His thesis on ForNet won 2nd prize in the International ACM 2005 research competition. He may be reached at *kulesh@isis.poly.edu*

IRMC IO & A Department

by Mary Linda Polydys and Mark Duke

The Information Resources Management College (IRMC) is the largest college in the National Defense University (NDU), with more than 3,000 active students each year. Of the three IRMC departments, the Information Operations and Assurance Department is the largest. This department was created to educate information leaders in information operations (IO) and information assurance (IA). [1] Graduate-level IO and IA courses are offered to qualified members of the military and federal civilian employees. The courses are free to Department of Defense (DoD) students and can be applied to master's and doctoral degree programs at several regionally accredited partner universities. [2] In addition, DoD students who are accepted into the DoD IA Scholarship Program (IASP)/ IRMC Option [3] are required to take a number of courses before completing their degrees at partner universities. Since the inception of the DoD IASP, the Information Operations and Assurance Department has taught and mentored more than 72 students through the IASP.

The department offers seventeen IA courses and three graduate-level certificates: the Information Systems Security Professionals (NSTISSI No. 4011) Certificate, the Senior System Manager (CNSSI No. 4012) Certificate, and the Chief Information Security Officer (CISO) Certificate. The 4011 and 4012 certificates satisfy the DoD 8570.1-M [4] education requirements for management personnel performing IA functions on national security systems. The CISO certificate supports the education needs of the senior agency information security officer identified in the Federal Information Security Management Act of 2002. The IA courses are designed to prepare graduates to—

- Exercise strategic leadership in the development and use of information security strategies, plans, policies, enabling technologies, and procedures
- Develop and lead programs to provide information security controls, security awareness training, risk analysis, certification and accreditation, security incident management, continuity of operations, and disaster recovery
- Link people, processes, information, and technology to critical IA decisions
- Develop and lead, in accordance with laws and regulations, an enterprise IA program that promotes and attains national security, agency, and interagency goals.

The IRMC also offers an IO concentration to students who attend the National War College (NWC) and Industrial College of the Armed Forces (ICAF). Although the concentration is specific to these colleges, the courses required for the concentration are available to all eligible students. These IO courses are not technical; rather, they are strategic-level courses that explore the impact of the information age on national security. The IO courses focus on information as both a component of national power and a strategic environment of increasing criticality to economists, diplomats, political leaders, military planners, and national security strategists.

Although IRMC's primary focus is education, professors are encouraged to perform research to improve the content of their courses. Department faculty frequently produce journal articles, books, and book chapters; conduct presentations; and speak at peer-reviewed conferences. Since 2000, department faculty have participated in more than 200 of these events.

One of the strongest benefits the department offers to students is its extensive laboratory. The laboratory covers a number of IA topics, including—

- **Biometrics**—Provides students the opportunity to experience various biometric technologies and see first-hand how false positives and false negatives can affect such systems
- **Supervisory Control and Data Acquisition (SCADA)**—Provides students an understanding of SCADA vulnerabilities

▷▷ *continued on page 34*

Accurate Application-Specific Sandboxing for Win32/Intel Binaries

by Wei Li, Lap-chung Lam, and Tzi-cker Chiueh

Abstract

Comparing the system call sequence of a network application against a sandboxing policy is a popular approach to detecting a control-hijacking attack, in which the attacker exploits software vulnerabilities such as buffer overflow to take control of a victim's application and possibly the underlying machine. The long-standing technical barrier to acceptance of the system call monitoring approach is determining how to derive accurate sandboxing policies for Windows applications whose source code is unavailable. In fact, many commercial computer security companies take advantage of this fact and fashion a business model in which their users must pay a subscription fee to receive periodic updates on the application sandboxing policies, much like anti-virus signatures. This article describes the design, implementation, and evaluation of a sandboxing system called BASS that can automatically extract a highly accurate application-specific sandboxing policy from a Win32/X86 binary, and enforce the extracted policy at run time with low performance overhead. BASS is built on a binary interpretation and analysis infrastructure called BIRD, which can handle application binaries with dynamically linked libraries, exception handlers, and multi-threading. BIRD has been shown to work correctly for a large number of commercially distributed Windows-based network applications, including IIS and Apache. The throughput and latency penalty of BASS for all of the applications we have tested except one is less than eight percent.

Introduction

One popular approach to host-based intrusion detection is to compare the run-time system call behavior of an application program with a predefined system call model, and declare an intrusion when a deviation between the two arises. This approach has been the linchpin of many research prototypes and commercial products under the names sandboxing, [20] behavioral blocking, [7] and restricted execution environment. [12] Although conceptually appealing, the technology has not been widely adopted in practice because the number of false positives—which disrupt legitimate applications—is still too high. Therefore, the main technical barrier of this system call-based sandboxing approach is determining how to automatically generate a system call model (or sandboxing policy) for arbitrary application programs that minimizes both the false positive rate and the false negative rate. This article describes the design, implementation, and evaluation of a system call-based sandboxing system called BASS that successfully removes this barrier for commercially distributed Win32 binaries running on Intel X86 architecture.

BASS's automated system call model extraction mechanism is an extension of PAID, [16] which analyzes an input program's source code and outputs a system call graph specifying the ordering among the program's system calls. BASS extends PAID in several important ways. First, BASS's system call model records the "coordinate" of each system call site, which is defined by the sequence of function calls from the program's main function to the function containing the system call site and the system call site itself. [2] Moreover, the run-time system call monitoring engine of BASS features a novel system call graph traversal algorithm that can efficiently map out the trajectory from one system call site to the next based on their coordinates. Second, BASS checks system call arguments in addition to system call ordering and coordinates. Finally, BASS supports load-time random insertion of null system calls to thwart mimicry attacks (see the X section). As a result of these techniques, the false positive rate of BASS is zero; *i.e.,* the intrusions PAID reports are guaranteed to be intrusions. In addition, the false negative rate of BASS with respect to control-hijacking attacks is very small; *i.e.,* the probability of successful control-hijacking attacks is miniscule, as explained later in the Attack Analysis section.

Another major difference between BASS and PAID is BASS is able to derive a system call model for an arbitrary

BASS's automated system call model extraction mechanism is an extension of PAID [16], which analyzes an input program's source code and outputs a system call graph specifying the ordering among the program's system calls.

Windows/X86 executable file and dynamically linked library (DLL). Because state-of-the-art disassemblers cannot distinguish between instructions and data in Windows/X86 binaries with 100% accuracy, [21] it is not possible to statically uncover all instructions of a binary image, let alone its system call model. To solve this problem, BASS is built on a general binary analysis and instrumentation infrastructure called BIRD, [18] which is specifically designed to facilitate the development of software security systems by simplifying the analysis and instrumentation of Windows/X86 binaries. Given a binary program, BIRD statically disassembles the program to uncover as many instructions as possible, rewrites it to allow run-time interception at all indirect jumps and calls, and dynamically disassembles those binary areas that cannot be disassembled statically.

The Windows operating environment introduces several additional issues that do not exist in PAID, which was designed for the Linux platform. First, Windows binaries are more difficult to disassemble than Linux binaries because the former tend to contain more handcrafted assembly instruction sequences that violate standard programming conventions, such as jumping from one function into the middle of another function. Second, because the procedural call convention is not followed strictly, deriving the coordinate of a system call site is non-trivial because it is not always possible to accurately infer the locations of the return addresses currently on the stack. Third, Windows applications use DLLs extensively, and common DLLs—such as Kernel32.DLL, User32.DLL, and NTDLL.DLL—are enormous. Therefore, it is essential to share the system call graphs for these DLLs across applications, as well as their code. BASS successfully solves these three problems and demonstrates for the first time that it is not only feasible but also efficient to sandbox Windows binaries with an automatically generated system call model that produces zero false positive and almost zero false negatives. As a result, we believe BASS makes a powerful building block for guarding enterprises against all Internet worms that use control-hijacking attacks, such as buffer overflow attacks.

Application-Specific Sandboxing

Abstract Model

By preventing applications from issuing system calls in ways not specified in their system call model, one could effectively stop all control-hijacking attacks. One way to derive a network application's system call model automatically is to extract its system call graph from its control flow graph (CFG) by abstracting away everything except the function call and system call nodes. A system call graph is a non-deterministic finite state automaton (NDFSA) model, due to if-then-else statements and functions with multiple call sites. The more impossible paths a system call model has, the more leeway is available to mimicry attacks, [27] which issue system calls exactly in the same order as specified in the system call graph before reaching the system call that can damage the victim system (*e.g.,* exec()). To reduce the amount of non-determinism in a system call graph, BASS uses a Call Site Flow Graph (CSFG), which captures both the ordering among system call sites and their exact locations. More specifically, a system call site's coordinate is uniquely identified by the sequence of return

addresses on the user stack when it is made and the return address of the system call's corresponding trap instruction.

In CSFG, a call node and a return node represent each function call, and each call node or return node is labeled with its return address. The manner in which BASS uniquely identifies each system call site removes the non-determinism caused by functions with multiple call sites. Despite the assignment of a unique coordinate to each system call site, CSFG is still an NDFSA, as illustrated by the functions foo6 and foo7 in Figure 1. Because of the if statement, foo6 and foo7 do not always make a system call. A function that may not always lead to any system call is referred to as a "may" function. Because of may functions, BASS cannot use a DFSA traversal algorithm to traverse the CSFG.

Because the edges between per-function CSFGs are uniquely labeled by their return addresses, transitions between these CSFGs is always deterministic. Consequently, the CSFG traversal algorithm is a combination of DFSA traversal, which is for inter-function traversal, and depth-first traversal, which is for intra-function traversal. The example in Figure 1 illustrates the basic concepts of this algorithm. (For a complete description of the CSFG traversal algorithm, see [15].) Assume the current system call is sys1, which is legitimate, and the current CSFG cursor points to sys1_r7_t1. When a new system call sys2 is called from the function r9_t2, if the CSFG traversal algorithm can successfully identify a path from the node sys1 r7_t1 to the node sys2_r9_t2 that does not contain any other system calls, sys2 is considered legitimate and allowed to proceed.

When a new system call comes in, BASS first extracts the return address chain from the user stack. For example, when sys2 is called, the return address chain is {r1, r4, r6, r9, t2}. The last two return addresses, r9 and t2, are not used for graph traversal because they are used to identify the corresponding system call site. Therefore, the CSFG traversal algorithm uses only {r1, r4, r6} as its new stack. The new stack of the last system call (sys1 in this case), is called the saved stack, and is {r1, r2, r5}.

The CSFG traversal algorithm first computes the prefix of the saved stack and the new stack, which is {r1}. Because the saved stack is longer than the prefix, the application must have returned back to the function foo1 before making the system call sys2. Each time the algorithm moves the cursor to a new function, it uses depth-first traversal to look for the exit node of the current function. This search is deterministic because every function has only one exit node and works correctly even when the CSFG contains may functions; *e.g.,* the call r8 node in foo4. The return address sequence after the prefix in the saved stack is {r2, r5}, based on which the algorithm performs the following operations to simulate function returns—

1. Find exit(foo4) using depth-first traversal
2. Consume r5 using DFSA traversal, and move the cursor to ret r5

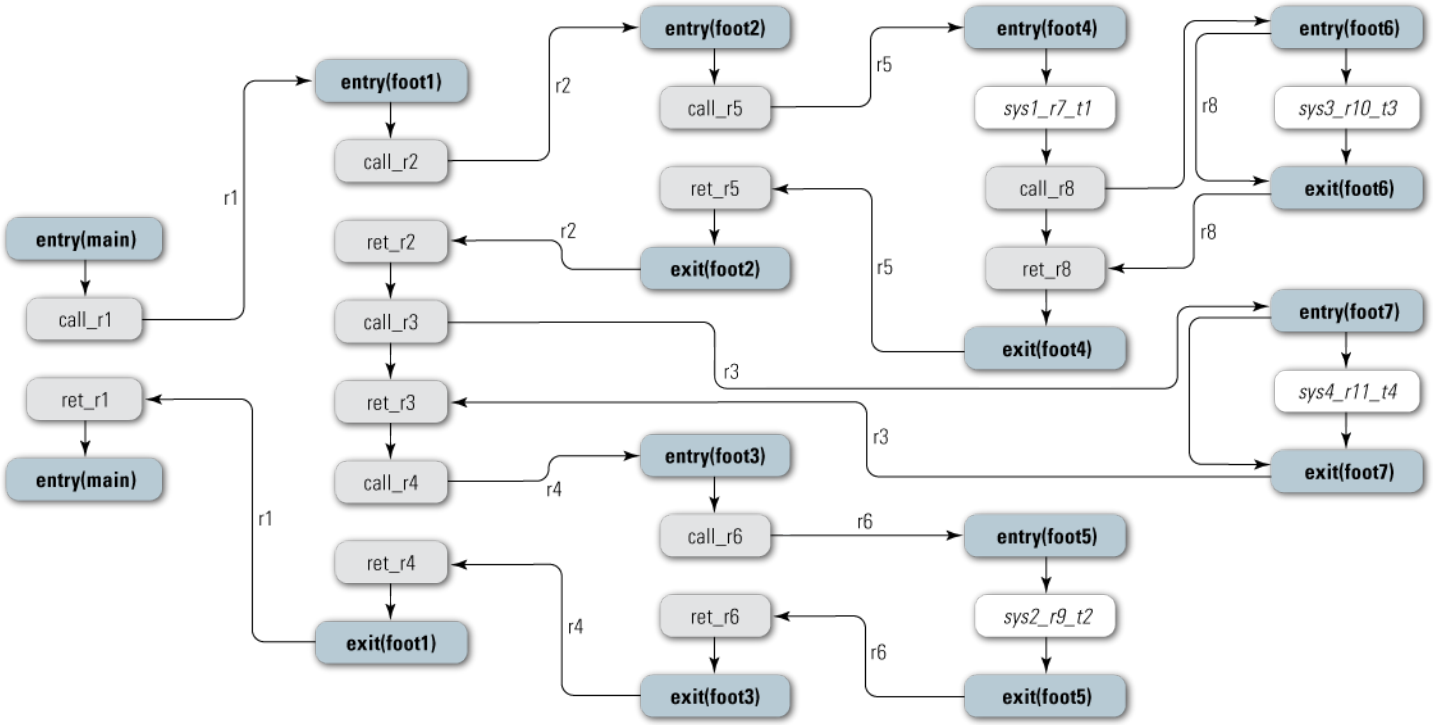


Figure 1 For the system call sequence {sys1, sys2}, when sys2 is called, the saved stack is {{r1, r2, r5}, the new stack is {r1, r4, r6}, and the prefix is {r1}. The run-time verifier needs to simulate the function returns and function calls to determine whether there is a path from the saved stack to the new stack.

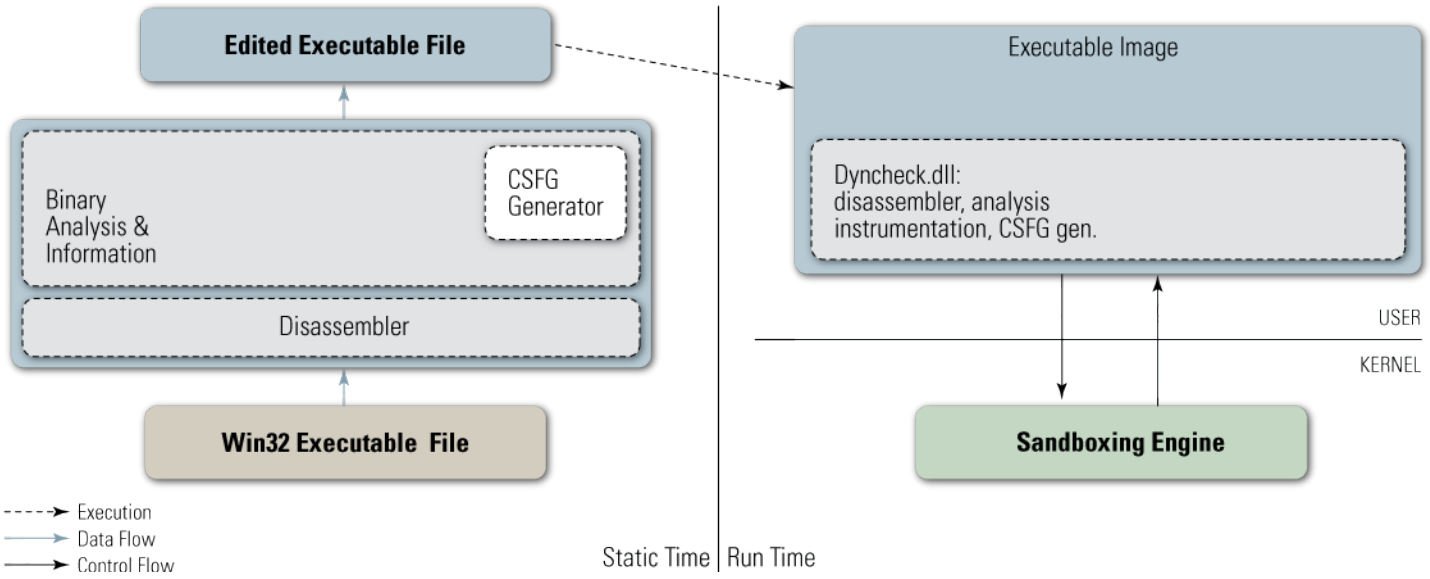


Figure 2 The system architecture of BASS, which consists of a static component that statically disassembles a binary file into instructions and extracts their system call model; a dynamic component that at run time disassembles those portions of the binary file that cannot be disassembled statically and extracts their system call accordingly; and a sandboxing engine that compares an application's dynamic system call patterns with its system call model.

3. Find exit(foo2) using depth-first traversal
4. Consume r2 using DFSA traversal, and move the cursor to ret r2.

After the above operations, the cursor is in the function foo1. Because the new stack is longer than the prefix, the application must have made some function calls before invoking the system call sys2. Therefore, the algorithm needs to simulate the call operations. Each time the cursor moves to a new function, the algorithm uses depth-first traversal to look for the call node that is labeled with the current stack symbol. This operation is deterministic because each call node is uniquely labeled by its return address. The return addresses after the prefix in the new stack are {r4, r6}, based on which the algorithm simulates the call operations using the following steps—

1. Find the call node labeled by r4 using depth-first traversal, which is call_r4
2. Consume r4 using DFSA traversal, and move the cursor to the callee of call_r4, which is entry(foo3)
3. Find the call_r6 node using depth-first traversal
4. Consume r6 using DFSA traversal, and move the cursor to entry(foo5). After completing the simulation of return and call operations, the CSFG algorithm uses depth-first traversal to reach the node sys2_r9_r2, which means the system call in question, sys2, is indeed legitimate.

Because of indirect calls (*i.e.*, function pointers), even if an application's source code is available, it is not always possible to construct a complete CSFG for that application. BASS solves this problem by inserting before every indirect call a notify system call, which informs the sandboxing engine of the actual target of the indirect call. The sandboxing engine uses this information to temporarily connect two potentially disconnected CSFG components and continue CSFG traversal. The disadvantage of this approach is additional system call overhead for every indirect call.

System Implementation

Figure 2 shows the system architecture of BASS. The following subsections describe its various components in detail.

Most existing binary analysis and instrumentation tools are developed on Unix/Linux OS and/or RISC architecture because it is generally easier to statically disassemble and analyze binaries on these platforms. However, Win32 binaries on the X86 architecture are much less susceptible to static disassembly and analysis because of handcrafted assembly routines and intentional obfuscation. To address this problem, we developed a new binary analysis/instrumentation system called BIRD [18], which performs both static and dynamic disassembly to guarantee that every instruction in a binary file will be properly examined before it is executed.

Because the instructions that BIRD recovers from an executable binary are meant to be transformed, it is essential that BIRD's disassembler be 100% accurate. In contrast, commercial disassemblers, such as IDA Pro, are designed for reverse engineering purposes, and therefore do not have to be as accurate as BIRD. To overcome the fundamental limitations of static disassemblers with respect to Win32 binaries, BIRD adopts a hybrid architecture that statically disassembles a binary

file as much as possible and defers the rest to dynamic disassembly at run time. Because most of the instructions in a binary file are disassembled statically, the performance overhead of dynamic disassembling is minimal. However, the flexibility of dynamic disassembly offers a simple and effective fallback mechanism for cases where static disassembling fails.

BIRD's static disassembler starts with a recursive traversal pass from the input binary's main entry point. Any instructions identified in this pass are guaranteed to be instructions. To improve the coverage of recursive traversal, BIRD applies data flow analysis to statically determine the target addresses of as many indirect jumps/calls as possible, and converts them into their direct counterparts. In addition, it exploits various PE header information, such as export table, relocation table, *etc.,* to identify places in a binary file that are known to be instructions.

The portions of a binary file that have been successfully disassembled are called known regions, whereas the rest are called unknown regions. Because of recursive traversal, the only way for a program's control to change from a known region to an unknown region is through an indirect control transfer instruction. Therefore, BIRD intercepts every indirect control transfer instruction at run time, and invokes the dynamic disassembler if it jumps to an unknown region. Run-time interception is through direct binary rewriting. This check-and-invoke logic forms the run-time engine of BIRD. The dynamic disassembler works similarly to the static one in that it also applies recursive traversal until the traversal encounters a known region or an indirect branch.

BASS intercepts system calls the same way as tools such as RegMon and FileMon, [24] which are designed to monitor run-time behaviors of application programs. Modern Windows OSs include a kernel executive, which provides core system services. All user-level API calls, such as those frequently used in KERNEL32.DLL, NTDLL.DLL, will eventually call these system services or Native APIs. The kernel executive dispatches native API calls through the system service dispatcher table (SSDT). By writing a kernel device driver, BASS can modify the function pointer entries in SSDT and intercept all system calls with additional functions. Consequently, each time a system call is invoked, BASS's interception function is called first, which performs the required sandboxing operation and decides whether to block the system call.

Application	Test Case	BIRD	Shadow Stack	CSFG Storage
Apache	Fetch a 1Kbyte file	2.5%	178.7%	106.3%
BIND	Query a name	2.5%	131.1%	270.0%
IIS W3 Service	Fetch a file	3.47%	107.1%	238.1%
MTS Email	Send a 1 Kbyte file	8.33%	108.34%	234.33%
Cerberus Ftpd	Fetch a 1Kbyte file	4.17%	67.4%	161.0%
GuildFTPd	Fetch a 1Kbyte file	4.24%	139.09%	120.5%
BFTelnetd	Login and list files	6.25%	87.5%	207.8%

Table 1 The network server applications used in the performance evaluation study; the test case for each of them; and the increase in their binary size under BASS due to BIRD, maintenance of shadow stack, and storage of CSFG.

Performance Evaluation

Methodology

The current BASS prototype can successfully run on Windows 2K, including Windows 2K Advanced Server, and Windows XP, with or without SP1 or SP2. Because BIRD needs to instrument known regions of executables and DLLs, we temporarily disable the Windows File Protection feature to modify the system DLLs and IIS. To evaluate the performance overhead of BASS, we measured the throughput and latency penalty of BASS with seven network server applications, which Table 1 briefly describes. Although BASS works on IE and Microsoft Office programs, we do not use them in the performance study because it is difficult to accurately measure the performance overhead for interactive applications that require user actions. We ran each of these applications under the following four configurations—

1. Native mode, in which applications are executed without interception or checking
2. BIRD mode, in which applications are executed with BIRD's interception
3. BIRD/BASS mode, in which applications are executed with BIRD's interception and BASS's system call checking,
4. BIRD/BASS/Random mode, in which null system calls are randomly inserted into applications at load time and the resulting binaries are executed with BIRD's interception and BASS's system call checking. For this study, we chose 38 sensitive system calls to monitor that are related to file system and registry manipulation.

To test the performance of each server program, we used two client machines that continuously send 2,000 requests to the test server applications. In addition, we modified the server machine's kernel to record the creation and termination time of each process. The throughput of a network server application is calculated by dividing

Application	BIRD		BIRD+BASS		BIRD+BASS+Random	
Apache	99.9%	0.9%	94.2%	5.5%	94.0%	5.6%
BIND	97.8%	3.1%	92.3%	7.7%	91.9%	7.9%
IIS W3 Service	99.1%	1.1%	93.9%	6.3%	93.5%	6.8%
MTS Email	99.7%	1.4%	97.3%	3.2%	97.3%	3.2%
Cerberus Ftpd	99.2%	1.2%	93.0%	7.6%	93.0%	8.2%
GuildFTPd	79.9%	25.3%	73.3%	32.7%	71.3%	33.2%
BFTelnetd	99.9%	1.5%	97.4%	3.4%	96.9%	3.5%

Table 2 The normalized throughput (left column) and latency penalty (right column) of the BIRD mode, the BIRD/BASS mode, and the BIRD/BASS/Random mode when compared with the Native mode for the seven test applications.

2,000 by the time interval between creation of the first forked process and termination of the last forked process. The latency is calculated by taking the average of the response times for each of the 2,000 requests. The server machine used in this experiment is a Windows XP SP1 machine with Pentium4 2.8-GHz CPU and 256-MB memory. One client machine is a 300-MHz Pentium2 with 128-MB memory and the other client is a 1.1-GHz Pentium3 machine with 512-MB memory. Both of them run Redhat Linux 7.2. The server and client machines are connected through a 100-Mbps Ethernet link. To test HTTP and FTP servers, the client machines continuously fetched a 1-KB file from the server, and the two client programs were started simultaneously. In the case of the mail server, the clients retrieved a 1-KB mail from the server. A new request was sent only after the previous request was complete. To speed up the request sending process, client programs simply discarded the data returned from the server.

Performance Overhead

Table 2 shows the throughput penalty of the test applications under the BIRD mode, BIRD/BASS mode, and BIRD/BASS/Random mode compared to the Native mode. For most applications except GuildFTPd, the majority of the throughput penalty comes from BASS, which accounts for a 1.8–6.4% drop in throughput, whereas BIRD accounts for a 0.9–3.1% throughput loss. The randomization component of BASS does not contribute much to throughput loss. With BIRD and BASS combined, the total throughput degradation remains within 8%, which is a generally acceptable performance penalty. The overall throughput penalty of GuildFTPd is about 29%; 20% due to BIRD and 9% due to BASS. GuildFTPd incurs a high BIRD-interception overhead because it uses heavily dispatching functions and small callback functions, which correspond to indirect calls. As a result, the check-and-invoke logic in BIRD is triggered so frequently that eventually this logic accounts for a significant portion of GuildFTPd's overall run time.

The latency penalties for different applications running under different configurations are similar to their throughput penalties. Overall, the latency penalty is also bounded under 8%, with the exception of GuildFTPd, whose latency penalty is more than 30%.

Conclusion

To the best of our knowledge, BASS is the first system call-based sandboxing system that can automatically sandbox arbitrary Windows binaries running on the Intel X86 architecture without any human inputs and with low performance overhead, while achieving a zero low false positive rate and an almost zero false negative rate. Because BASS operates at the binary level, it is independent of the source languages and the associated compilers/linkers, and thus is applicable to a wide range of applications. In addition, BASS offers users an effective way to protect themselves from potential bugs in third-party applications without support from the original application developers or from special computer security vendors. More concretely, this work makes the following contributions—

- A highly accurate system call model representation that checks system call ordering, system call coordinates, and system call arguments, which together greatly minimize the window of vulnerability to mimicry attacks
- A flexible and efficient Win32/X86 binary interpretation system that has been shown to correctly interpret a wide variety of Windows applications, including Microsoft Office suite and IIS, which state-of-the-art disassemblers fail to disassemble completely
- One of the most—if not the most—comprehensive system call pattern-based host-based intrusion detection systems that could automatically and accurately sandbox applications that involve dynamically linked libraries, multi-threading, and exception handlers. ■

References

1. M. Abadi, M. Budiu, Ifar Erlingsson, and J. Ligatti. Control-flow integrity. In Proceedings of the 12th ACM conference on computer and communications security, pages 340–353, Alexandria, VA, November 2005.
2. G. Ammons, T. Ball, and J. Larus. Exploiting hardware performance counters with flow and context sensitive profiling. In Proceedings of 1997 ACM SIGPLAN Conf. on Programming Language Design and Implementation, 1997.
3. V. Bala, E. Duesterwald, and S. Banerjia. Dynamo: A transparent dynamic optimization system. ACM SIGPLAN Notices, 35(5):1–12, 2000.

4. D. Bruening, E. Duesterwald, and S. Amarasinghe. Design and implementation of a dynamic optimization framework for windows. In 4th ACM Workshop on Feedback-Directed and Dynamic Optimization (FDDO-4), December 2000.
5. B. D. Bus, D. Kastner, D. Chanet, L. V. Put, and B. D. Sutter. Post-pass compaction techniques. Commun. ACM, 46(8):41–46, 2003.
6. S. Chen, J. Xu, E. C. Sezer, P. Gauriar, and R. Iyer. Non-control-data attacks are realistic threats. In Proceedings of 14th USENIX Security Symposium, August 2005.
7. A. Conry-Murray. Product focus: Behavior-blocking stops unknown malicious code. *http://www.networkmagazine.com/shared/article/showArticle.jhtml?articleId%20=8703363&classroom= (2002).*
8. H. H. Feng, J. T. Giffin, Y. Huang, S. Jha, W. Lee, and B. P. Miller. Formalizing sensitivity in static analysis for intrusion detection. In IEEE Symposium on Security and Privacy, page 194, Berkeley, CA, May 2004.
9. S. Forrest, S. A. Hofmeyr, A. Somayaji, and T. A. Longstaff. A sense of self for Unix processes. In Proceedings of the 1996 IEEE Symposium on Research in Security and Privacy, pages 120–128. IEEE Computer Society Press, 1996.
10. J. T. Giffin, S. Jha, and B. P. Miller. Detecting manipulated remote call streams. In Proceedings of the 11th USENIX Security Symposium, pages 61–79. USENIX Association, 2002.
11. J. T. Giffin, S. Jha, and B. P. Miller. Efficient context-sensitive intrusion detection. In Proceedings of the 11th Annual Network and Distributed System Security Symposium, Feb. 2004.
12. I. Goldberg, D. Wagner, R. Thomas, and E. A. Brewer. A secure environment for untrusted helper applications. In Proceedings of the USENIX Security Symposium, July 1996.
13. V. Kiriansky, D. Bruening, and S. Amarasinghe. Secure execution via program shepherding. In 11th USENIX Security Symposium, 2002.
14. C. Kruegel, E. Kirda, D. Mutz, W. Robertson, and G. Vigna. Automating mimicry attacks using static binary analysis. In Proceedings of the USENIX Security Symposium, Baltimore, MD, August 2005.
15. L. C. Lam. Program transformation techniques for host-based intrusion prevention. Ph.D. dissertation, Computer Science Department, Stony Brook University, December, 2005.
16. L. C. Lam and T. cker Chiueh. Automatic extraction of accurate application-specific sandboxing policy. In Seventh International Symposium on Recent Advances in Intrusion Detection, Sophia Antipolis, France, September 2004.
17. J. R. Larus and E. Schnarr. Eel: Machine-independent executable editing. In Proceedings of the ACM SIGPLAN'95 Conference on Programming Language Design and Implementation, pages 291–300, La Jolla, CA, June 1995.
18. S. Nanda, W. Li, L. chung Lam, and T. cker Chiueh. Bird: Binary interpretation using runtime disassembly. In Proceedings of the 4th IEEE/ACM Conference on Code Generation and Optimization (CGO'06), March 2006.
19. M. Prasad and T. cker Chiueh. A binary rewriting defense against stack based overflow attacks. In Proceeding of the 2003 Usenix Annual Technical Conference, June 2003.
20. V. Prevelakis and D. Spinellis. Sandboxing applications. In Proceedings of the FREENIX Track: 2001 USENIX Annual Technical Conference, pages 119–126, 2001.
21. T. Reps, G. Balakrishnan, J. Lim, and T. Teitelbaum. A next-generation platform for analyzing executables. In Proceedings of the 3rd Asian Symposium on Programming Languages and Systems, Tsukuba, Japan, Nov. 2005.
22. A. Srivastava, A. Edwards, and H. Vo. Vulcan: Binary Transformation in a Distributed Environment. Technical Report MSR-TR-2001-50, Microsoft Research, 2001.
23. A. Srivastava and D. W. Wall. A practical system for intermodule code optimization at link-time. Journal of Programming Languages, 1(1):1–18, December 1992.
24. SysInternals. *http://www.sysinternals.com/ntw2k/source/regmon.shtml*
25. UPX. The ultimate packer for executables. *http://upx.sourceforge.net.*
26. D. Wagner and D. Dean. Intrusion detection via static analysis. In Proceedings of the IEEE Symposium on Security and Privacy, pages 156–168, 2001.
27. D. Wagner and P. Soto. Mimicry attacks on host-based intrusion detection systems. In Proceedings of the 9th ACM conference on Computer and communications security, pages 255–264, Washington, DC, USA, 2002. ACM Press.

About the Authors

Wei Li | is currently a PhD candidate in the Department of Computer Science at the University of New York at Stony Brook. She has been working in the Experimental Computer Systems Lab with Professor Tzi-cker Chiueh. Her current research has focused on computer systems security, intrusion detection and prevention, program analysis and transformation. She may be reached at *weili@cs.sunysb.edu*

Lap-chung Lam | currently works for Rether Networks, Inc. as a chief engineer. His main research interests are system security, dynamic information flow control, OS level virtualization, and digital rights management. He holds a BA degree in computer science and mathematics from State University of New York at New Paltz. He received his MS and PhD degrees in computer science from Stony Brook University. He may be reached at *lclam@rether.com*

Dr. Tzi-cker Chiueh | is a Professor in the Computer Science Department of Stony Brook University, and the head of the Core Research group at Symantec Research Labs.

He received his BS in EE from National Taiwan University, MS in CS from Stanford University, and PhD in CS from University of California at Berkeley in 1984, 1988, and 1992, respectively. He received an NSF CAREER award in 1995, an IEEE Hot Interconnect Best Paper award from the 8th International Symposium on Systems and Information Security (SSI 2006), and a Best Paper award from the Third International Symposium on Information Assurance and Security (IAS 2007).

Dr. Chiueh has published over 160 technical papers in refereed conferences and journals. His current research interest lies in wireless networking, computer security, and storage systems. He may be reached at *chiueh@cs.sunysb.edu*

▷ continued from page 17

the director of system engineering for Lockheed Martin Telecommunications and was responsible for system engineering for all new commercial satellite programs. Mr. Modelfino holds an MS in Electrical Engineering and a BS in Physics from the State University of New York at Stony Brook. He may be reached at *tony.modelfino@stratogis.com*

George Case | is one of the co-founders of Stratogis Networks, LLC. and currently leads the development of Stratogis Network's software products. Mr. Case has 20 years of commercial and DoD satellite design and development experience, the last six focused on satellite network architecture design and traffic modeling. Prior to Stratogis Networks, Mr. Case was Director of Services Analysis for Astrolink International, LLC. From 1983 to 1997, Mr. Case held technical and program management positions with Lockheed Martin Commercial Space Systems (LMCSS). Mr. Case holds a BS degree in Mechanical Engineering from University of Virginia. He may be reached at *george.case@stratogis.com*

Letter to the Editor

Information assurance (IA) academic and educational centers matter to the Information Assurance Technology Analysis Center (IATAC). Since 2004, IATAC has been affiliated with the National Security Agency's (NSA) IA outreach program, Centers of Academic Excellence in Information Assurance Education (CAEIAE).

The CAEIAE program was created in the spirit of President Clinton's Decision Directive 63, *The Clinton Administration's Policy on Critical Infrastructure Protection*. The Department of Homeland Security (DHS), in support of President Bush's *National Strategy to Secure Cyberspace* (2003), is now a joint sponsor of the CAEIAE program. The joint program goal is to "reduce vulnerability in our national information infrastructure by promoting higher education in IA and producing a growing number of professionals with IA expertise in various disciplines." The *National Strategy to Secure Cyberspace* refers to cyberspace as the nervous system of our nation's critical infrastructures, and it indicates that the healthy functioning of cyberspace is essential to our economy and national security. Securing cyberspace presents a difficult strategic challenge, and IA education is a critical component in successfully meeting that challenge.

In June 2007, the CAEIAE held its annual Colloquium for Information Systems Security Education Conference in Boston, Massachusetts. During the conference, CAEIAE selected 86 centers across 34 states and the District of Columbia for the 2007–2012 academic years. A recognized university or center is acknowledged as having a certified curriculum and meeting the NSA's 10 IA criteria. In addition, each applicant receives a rigorous review demonstrating IA commitment. NSA's criteria are as follows—

- Have state-of-the-art IA resources
- Ensure faculty is active in IA practices and research and contributes to literature
- Have IA curriculum that reaches beyond the campus's geographic borders
- Have an academic program that encourages research in the field
- Encourage the best practices of IA
- Treat the program as a multidisciplinary science
- Create partnerships in information education
- Have a focus area or area of study in IA
- Have a full-time program facility
- Declare a center for IA education or research.

IATAC Director Gene Tyler states, "We've established relationships with the NSA and recognized IA Centers of Academic Excellence (CAEIAE) which are providing valuable sources of scientific and technical information (STI). As an example we routinely coordinate with and highlight the CIACAE to recognize their STI and IA efforts. The results of this coordination are highlighted in our Research Update as well as the Academic and SME Spotlight sections of the *IAnewsletter*."

In addition, for our website visitors, we have links to the CAEIAE universities and educational center on our Resources website. Most recently, we have spotlighted—

- George Mason University's Center for Secure Information Systems
- Center for Education and Research in Information Assurance (CERIAS) at Purdue University
- University of California at Davis
- Georgia State University's Department of Computer Information
- Systems at Mack Robinson College
- Johns Hopkins University's Information Technical Institute
- Mississippi State University's Center for Computer Security Research at Mississippi.

University of Maryland University College Security Studies Laboratory

by Don Goff

Founded in 1947, the University of Maryland University College (UMUC) offers a broad range of cutting-edge classes and has earned a global reputation for excellence. Headquartered in Adelphi, MD, UMUC has classroom locations in the Washington, DC, metropolitan area, Europe, and Asia and provides award-winning online classes to students worldwide. UMUC's Security Studies Laboratory (SSL) supports the largest information assurance (IA) student body among the university's named Centers of Academic Excellence in Information Assurance Education by the Department of Defense (DoD) and Department of Homeland Security (DHS).

Institutional Capabilities

The UMUC primarily serves adult, part-time students through traditional face-to-face and online instruction. UMUC offers IA-focused bachelor's and master's degrees and certificates through innovative online and classroom-based programs using various delivery formats and scheduling options. At the doctoral level, UMUC offers a security specialization in the Doctor of Management program, and all doctoral students take a course in information security. With more than 150,000 online enrollments in academic year 2006–2007, UMUC is the nation's largest online-enabled state university. UMUC's stateside programs annually award more than 800 undergraduate degrees in information technology (IT)—more than any other university in Maryland. In the past year, 22 percent of those degrees were awarded to African-American students. (Note that UMUC is the state's largest grantor of advanced technology degrees to African-Americans.) UMUC also awarded more than 300 master's degrees in IT.

Students may take UMUC courses in classrooms at more than 25 locations in Maryland and the greater Washington metropolitan area or in classrooms on US military bases across Europe and Asia through longstanding partnerships with the armed forces. In 1949, UMUC began providing educational service to US military overseas and today is the leading education provider for the US military. In fiscal year 2006, UMUC enrolled more than 60,000 active duty military and their dependents through its overseas programs under contract with the US DoD. UMUC enrolled additional active duty military through its stateside online and onsite programs.

UMUC was designated a Center of Academic Excellence in Information Assurance Education in 2002, renewed in 2005, and certified by the National Institute of Standards and Technology (NIST) for National Security Telecommunications and Information Systems Security Instruction (NSTISSI) 4011, 4012, and 4013.

With its mature and robust online delivery and direct access to students, UMUC is uniquely qualified and has a superior capability to support the mission of the Information Assurance and Technical Analysis Center (IATAC). UMUC's focus on developing very large-scale undergraduate and graduate programs offers an opportunity to broadly disseminate IA knowledge and skills directly with the IATAC community. UMUC's participation offers an opportunity for developing a large potential employee pool of qualified IA professionals at the entry level of their careers. Qualified students will receive specific skills and instruction with a strong emphasis upon applications, using laboratories employing state-of-the-art and industry standard tools. Qualified students will be certified to NSTISSI standards and may also receive Clinger-Cohen certification. In addition, UMUC offers an opportunity for working adults within the DoD and other IATAC user agencies to pursue a graduate education online.

Information Assurance Program

UMUC has a large, robust IA program. The IA specialization in the graduate school provides a thorough knowledge base for managers and technology professionals concerned with the design, development, implementation, operation, and management of secure information systems and with the protection of an

organization's information assets. It provides students with a practical understanding of the principles of data protection, network security, and computer forensics. The program further introduces students to the ethical, legal, and policy issues associated with information security. Laboratory exercises are included in some courses to enhance the learning experience. The Master of Science program in IT serves careers in entry, mid-, or upper-level positions, depending on the student's prior level of experience.

UMUC is unique in offering an undergraduate major in IA. Undergraduate students with a major in IA learn to identify the terms, functions, and interrelationships among the hardware, software, firmware, and other components of an information system; demonstrate a working knowledge of the principles and practices of information security; develop policies and procedures to ensure reliability and accessibility of information systems and to prevent and defend against unauthorized access to systems, networks, and data; conduct risk and vulnerability assessments of planned and installed information systems to identify vulnerabilities, risks, and protection needs; develop systems security contingency plans and disaster recovery procedures; develop and maintain strategic IA plans; establish metrics to measure and evaluate systems performance and total cost of ownership; and identify and address IT workforce planning and management issues such as recruitment, retention, and training.

The library contains full test databases in IA that are accessible online and second to none for their depth and completeness.

UMUC has committed substantial direct and in-kind resources to the development and expansion of its online delivery generally and specifically to the IA program. UMUC has made every effort to ensure that the accompanying cost proposal is accurate, realistic, and detailed.

The substantial experience of the full-time and adjunct faculty provides students with substantial security awareness that goes beyond simple text-based learning. Supplementing theoretical knowledge with practical experience gives the students insights into the real-world problems of developing and implementing IA programs. Research conducted by faculty and students is focused on applications rather than empirical knowledge. Examples of approaches are case studies, best practice surveys, policy and regulatory analyses, and operations research. UMUC requires students to complete independent inquiry, usually in the form of a research paper, in many undergraduate courses. In addition, UMUC is expanding its online laboratory capability to allow students to manage and configure security applications over the Internet, with a primary focus on IA applications for databases, networks, and software engineering.

Security Studies Laboratory

To support these programs with hands-on laboratory experiences, UMUC created the SSL in 2004. This lab provides a conduit for exchanging state-of-the-art teaching and learning environments with IA and security content. It supports curriculum development, information architecture, faculty training and development, and laboratory development and support at the undergraduate and graduate levels. It develops advanced teaching tools such as remote access laboratories, network test beds, and emerging technologies and methods of providing asynchronous online learning globally.

Among the SSL's accomplishments are the creation of the first remote access network security labs. In courses in intrusion detection and forensics, students log on from anywhere in the world to use real equipment to solve network security problems. The lab, a "micro-network," consists of a closed system of routers and switches with ancillary firewalls, sniffers, and intrusion detection devices. Students can solve problems ranging from developing a simple network access control list, to dealing with traffic management issues of bypass and rerouting to work around a compromised piece of equipment.

In addition to the physical lab at Adelphi, Maryland, additional labs are nearing completion overseas. The first, in Heidelberg, Germany, supports teaching at US bases in Europe and "down range." The second, in Yokota, Japan, supports students assigned to military duties in Japan, Korea, Okinawa, Thailand, and Guam. Plans call for integrating these three labs into the first global laboratory teaching environment.

To achieve scalability, the SSL has begun to develop a series of simulations and emulations that students can use to solve IA problems. The emulations provide highly realistic problems that can be practiced in a virtual environment and then checked in a physical environment.

To gain adequate faculty, the SSL created the first virtual, post-doctoral fellowships in IA. In this program, faculty members worldwide participate in an in-service learning experience from their own campuses. Upon completion of six courses, they gain a professional credential at the graduate level, their home institution gains a new qualification, and UMUC recruits them to teach online—a win-win-win solution for all.

Laboratory founder Don Goff stated, "The SSL provides a focus and the resources necessary to really move the ball in IA education." Although he recently returned full time to private sector employment, Mr. Goff remains as an Advisory Member to the SSL. He adds, "UMUC President Susan Aldridge has made a commitment to the program and to the lab, and we're looking forward to continued growth and development."

In the interim while a search is conducted for a new Executive Director, Ms. Ouanessa Boubsil serves as Director. She has been at SSL from its beginning and has been a key player in making it work. ■

References

1. Additional information about UMUC and the SSL can be found at *http://www.umuc.edu/* and at *http://www.umuc.edu/ssl/*

About the Author

Don Goff | is currently the Vice President for Information Assurance and Training at Criterion Systems, Inc, in Vienna, VA. Previously, he was Professor and Executive Director of the Security Studies Laboratory at University of Maryland University College (UMUC), designated a National Center of Academic Excellence in Information Assurance Education by the Department of Homeland Security and the Department of Defense. Dr. Goff received his PhD in Telecommunications from Northwestern University, Evanston, Illinois in 1991. The University of Illinois, Urbana-Champaign awarded him the Master's degree in 1970, and Western Illinois University the Bachelor's in 1969. He is retired from AT&T and the US Army Reserve.

▷ *continued from page 23*

- **Cyber Defense**—Allows students to gain experience in actively defending against cyber attacks by dividing them into two teams
- **Forensics**—Provides students an understanding of the processes used to preserve and investigate evidence associated with incident response.

Other laboratory topics include cryptography, firewalls, intrusion detection and prevention systems (IDPS), and wireless security. ■

References

1. More information about the IO and IA programs can be found at *http://www.ndu.edu/irmc/pcs_iscp.htm* and *http://www.ndu.edu/irmc/pcs_ia.htm*
2. More information on partner universities can be found at *http://www.ndu.edu/irmc/ntwk_list1.htm*
3. More information on the IASP for DoD personnel can be found at *http://www.defenselink.mil/cio-nii/iasp/DoDMembersMain.htm*
4. See paragraph C1.4.4.13 of the DoD 8570.1-M.

About the Authors

Mary Linda Polydys | is the Chair of the Information Operations and Assurance at the IRM College of the National Defense University. She leads the department in information operations/assurance curriculum development and in maintaining their status as a center of academic excellence in information assurance education. She has more than 34 years of government service.

Mark Duke | is an Associate Professor at the IRM College of the National Defense University. He is the course leader for the Information Assurance and Critical Infrastructure Course and teaches in several other information assurance courses. He has more than 15 years of information security/assurance experience and more than 25 years of government service. He is a retired army Lieutenant Colonel.

FREE Products Order Form

Instructions: All IATAC LIMITED DISTRIBUTION reports are distributed through DTIC. If you are not a registered DTIC user, you must do so prior to ordering any IATAC products (unless you are DoD or Government personnel). To register online: *http://www.dtic.mil/dtic/registration*. The *IAnewsletter* is UNLIMITED DISTRIBUTION and may be requested directly from IATAC.

Name _____ DTIC User Code _____

Organization _____ Ofc. Symbol _____

Address _____ Phone _____

_____ Email _____

_____ Fax _____

Please check one: ☐ USA ☐ USMC ☐ USN ☐ USAF ☐ DoD
☐ Industry ☐ Academia ☐ Government ☐ Other

Please list the Government Program(s)/Project(s) that the product(s) will be used to support: _____

_____

LIMITED DISTRIBUTION

IA Tools Reports (softcopy only)
☐ Firewalls ☐ Intrusion Detection ☐ Vulnerability Analysis

Critical Review and Technology Assessment (CR/TA) Reports
☐ Biometrics (soft copy only) ☐ Configuration Management ☐ Defense in Depth (soft copy only)
☐ Data Mining (soft copy only) ☐ IA Metrics (soft copy only) ☐ Network Centric Warfare (soft copy only)
☐ Wireless Wide Area Network (WWAN) Security ☐ Exploring Biotechnology (soft copy only)
☐ Computer Forensics* (soft copy only. DTIC user code MUST be supplied before these reports will be shipped)

State-of-the-Art Reports (SOARs)
☐ Data Embedding for IA (soft copy only) ☐ IO/IA Visualization Technologies (soft copy only)
☐ Modeling & Simulation for IA (soft copy only) ☐ Malicious Code (soft copy only)
☐ Software Security Assurance ☐ A Comprehensive Review of Common Needs and Capability Gaps

UNLIMITED DISTRIBUTION

IAnewsletters Hardcopies are available to order. The list below represents current stock.
Softcopy back issues are available for download at *http://iac.dtic.mil/iatac/IA_newsletter.html*

Volumes 4		☐ No. 2	☐ No. 3	☐ No. 4
Volumes 5	☐ No. 1	☐ No. 2	☐ No. 3	☐ No. 4
Volumes 6	☐ No. 1	☐ No. 2	☐ No. 3	☐ No. 4
Volumes 7	☐ No. 1	☐ No. 2	☐ No. 3	☐ No. 4
Volumes 8	☐ No. 1	☐ No. 2	☐ No. 3	☐ No. 4
Volumes 9	☐ No. 1	☐ No. 2	☐ No. 3	☐ No. 4
Volumes 10	☐ No. 1	☐ No. 2	☐ No. 3	☐ No. 4

Fax completed form to IATAC at 703/984-0773

Calendar

January

SANS Security 2008
11–19 January 2008
New Orleans, LA
http://www.sans.org/security08/?utm_source=web-sans&utm_medium=text-ad&utm_context=text-link_index,_featured_text_link&utm_campaign=SANS_Security_2008&ref=15471

Seattle Tech-Security Conference
15 January 2008
Seattle, WA
http://www.dataconnectors.com/events/2008/01seattle/agenda.asp

Network Centric Warfare 2008
22–25 January 2008
http://www.transformingncw.com/

12th Annual IA Workshop (IAWS)
28 January–1 February 2008
Philadelphia Marriot Downtown
Philadelphia, PA
http://www.nsa.gov/ia/events/conferences/index.cfm?ConferenceID=51

February

SANS Silicon Valley 2008
2–8 April 2008
Hilton Alexandria
San Jose, CA
http://www.gcn.com/events/19218.html?topic=events

AFCEA West 2008
5–5 February 2008
San Diego, CA
http://www.afcea.org/events/west/2008/introduction.asp

RSA Conference
The 25th IEEE International Performance
5–8 February 2008
San Francisco, CA
http://www.rsaconference.com/2007/US/Event_Overview.aspx

Homeland Security 2008
27–28 February 2008
Washington, DC
http://www.afcea.org/events/homeland/landing.asp

March

INFOWARCON 2008
2–4 March 2008
Sandia National Laboratories, Bethesda, MD
http://www.infowarcon.com

Warfighter's Vision 2008
4–5 March 2008
Tampa, FL
http://www.afei.org/brochure/8a04/index.cfm

INFOSEC World Conference & Expo 2008
10–12 March 2008
Orlando, FL
http://www.misti.com/default.asp?Page=65&Return=70&ProductID=5539

Wireless Network Security (WISEC '08)
31 March–2 April 2008
Alexandria, VA
http://discovery.csc.ncsu.edu/WiSec08/

DTIC 2008 Conference
7–9 April 2008
Alexandria, VA
http://www.dtic.mil/dtic/annualconf/

DISA Customer Partnership Conference 2008
5–9 May 2008
Disney's Coronado Springs Resort
Orlando, FL
http://www.disa.mil/conference/index.html

IATAC
Information Assurance Technology Analysis Center
13200 Woodland Park Road, Suite 6031
Herndon, VA 20171

To change, add, or delete your mailing or email address (soft copy receipt), please contact us at the address above or call us at: 703/984-0775, fax us at: 703/984-0773, or send us a message at: iatac@dtic.mil