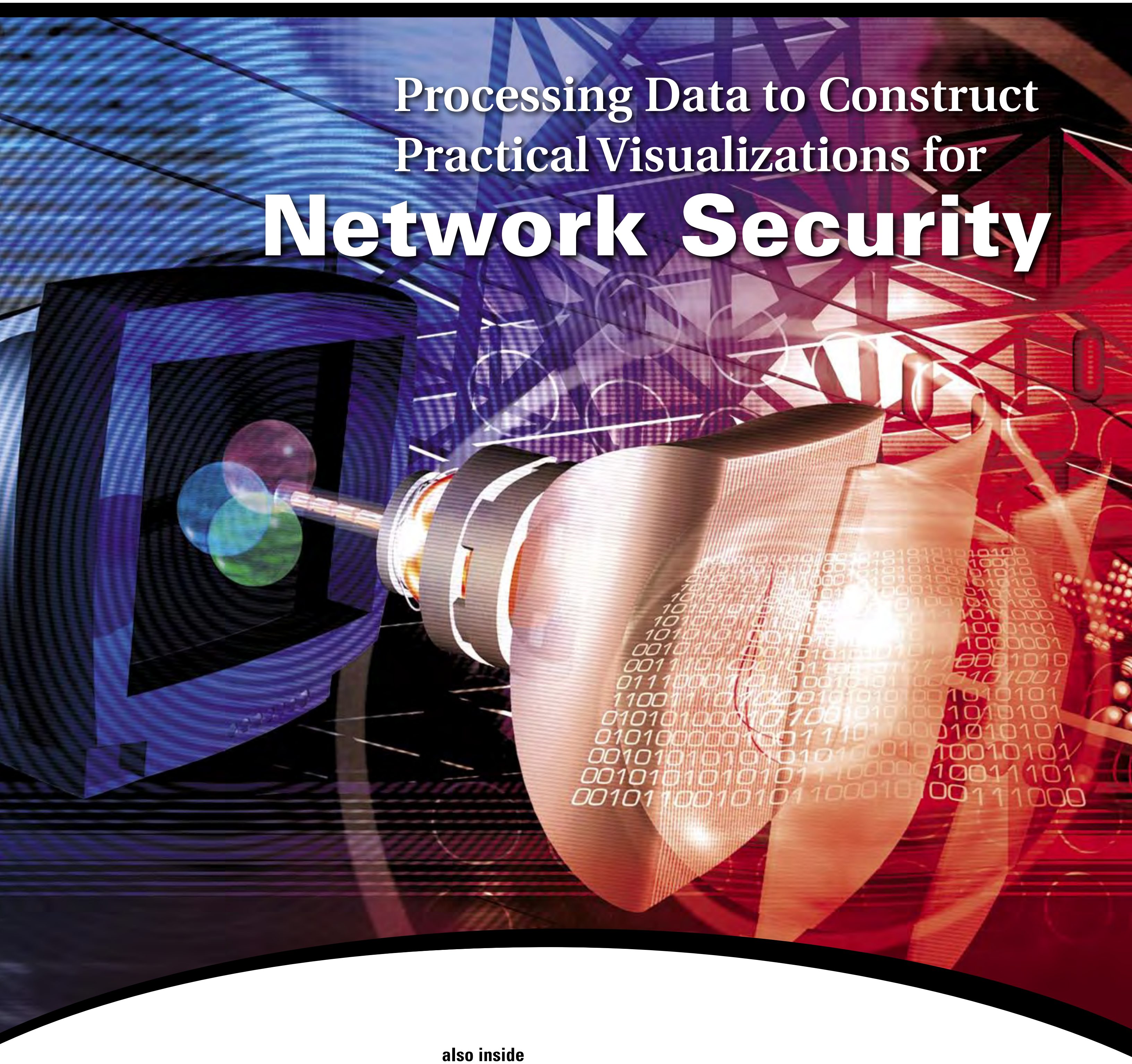
9/1

Volume 9 Number 1 • **Summer 2006**

# IAnewsletter

The Newsletter for Information Assurance Technology Professionals



# Processing Data to Construct Practical Visualizations for Network Security

**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, 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
Deputy Director: Greg McClellan
Inquiry Services: Peggy O'Connor

***IAnewsletter* Staff**
Promotional Director: Christina P. McNemar
Creative Director: Ahnie Jenkins
Art Directors: Bryn Farrar
Don Rowe
Copy Editor: Diane Ivone
Designers: Steve Green
Dustin Hurt
Brad Whitford
Editorial Board: Greg McClellan
Jim Peña
Ronald Ritchey
Tara Shea
Gene Tyler

***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 e-mail 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

E-mail: iatac@dtic.mil
URL: http://iac.dtic.mil/iatac

**Deadlines for future Issues**
Fall 2006 June 16, 2006

Cover design: Dustin Hurt
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

We have now been in our new offices for a couple of months and IATAC is looking better than ever.

In our last edition, I was filling you in on the move to our One Dulles facility in Herndon, VA. We have now been in our new offices for a couple of months and IATAC is looking better than ever. We worked diligently to ensure our move was seamless and transparent to you. However, if you did experience any delayed responses, let us know if we missed something—but please know we are back on track once again. Along with the move, I was telling you about some of the enhancements IATAC would be initiating, such as the new face of the *IAnewsletter*. We are very proud of our new look but are interested in your opinion too, so please let us know what you think.

In other exciting news, the Total Electronic Management System (TEMS) continues in its tremendous growth. By now, my hope is that all of our readers are well aware of the TEMS database, which we've featured in several previous editions of the *IAnewsletter* (including Volume 7 Number 4 and Volume 8 Number 3). TEMS allows users to search and access the latest Scientific and Technical Information (STI) across several IACs, via one central source. Currently, six of the nine DoD sponsored IACs have documents loaded in TEMS, with the three remaining IACs to come online shortly. Since a portion of this STI may contain information up to the Secret level and almost all are limited in distribution, users are required to register with the Defense Technical Information Center (DTIC). Once registered with DTIC, users will have access to more than 38,000 (and climbing) full-text documents, and more than 245,000 (and climbing) metadata entries. If you have not yet registered, please visit the DTIC Registration Web site to sign up (*http://www.dtic.mil/dtic/registration/index.html*).

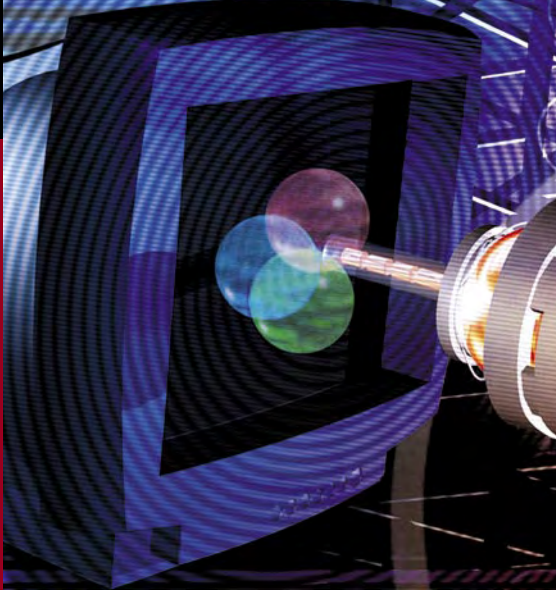
Currently, there are two means to access TEMS; both simply require your DTIC user ID and password. The first way is via DTIC's Private Scientific and Technical Information Network (STINET). After you register with DTIC, you will receive a confirmation e-mail which contains your user ID, DTIC User Code, as well as the link to Private STINET. The second method to gain access is through the Research & Engineering (R&E) Portal. As TEMS continues to grow, you will begin to see it in multiple locations. If you have any questions or concerns with either of these processes, please do not hesitate to contact us. As previously mentioned, registration through DTIC is required for access to either of these sources.

In this edition of the *IAnewsletter*, you will find some exciting articles of interest. Two articles in particular that I encourage you to read deal with our Warfighters. The first, "GIG-BE—Improving the Warfighter's Information Pipeline," talks about the Global Information Grid-Bandwidth Expansion Program and how it will provide a more robust and secure network environment, thus supporting the Warfighter and improving national security intelligence. The second Warfighter article, "Defending Warfighter Networks," goes into how to design networks that are self-defending and self-sustaining through attacks. Both articles, as well as the others of course, are well worth reading. ■

Gene Tyler

FEATURE STORY

# Processing Data to Construct Practical Visualizations for Network Security



by Kulsoom Abdullah, Gregory Conti, John Copeland, and Chris Lee

Network vulnerabilities are increasingly rampant despite advances in Intrusion Detection Systems (IDSs) and Intrusion Prevention Systems (IPSs). Even as funding and work by government, industry, and academia to counter these vulnerabilities increases, over 1,000 variants of worms and viruses have been discovered during the past six months, [1] and the level of network traffic increases as capacity increases. [2] Network monitoring systems are already choked performing packet analyses for large networks, and traffic increases worsens the problem. [3]

Information visualization methods deal with large datasets and provide far more insight and understanding to a human analyst than viewing text alone. [4] When techniques of information visualization have been applied to the network security domain, studies have shown a significant decrease in the time required to determine many types of network threats. The use of visualization with network data to aid in security is growing, but more work is still required. This article describes methods developed to scale a large amount of network data into meaningful visualizations for intrusion detection. These techniques were incorporated into the design and implementation of a tool to facilitate log analysis for IDSs. Capturing network traffic, the tool's design, the data-scaling method used before plotting, and definitions and illustrations of several threat models will be discussed.

### Capturing and Parsing Network Data

*Tcpdump*, a standard packet-capturing tool, collects network data, and the parameters used for visualization are then parsed from the network packet headers. The advantage of parsing network packets, compared to traffic-flow information, is that real-time processing on network packets can be performed instantaneously without having to wait for a flow to end compared to analyzing flow statistics. In our system, packet headers are parsed for information, but not the payload of the packet. This design choice was made because processing each packet payload would greatly increase the processing burden on the monitoring system.

During the design of our system, we considered requirements for both forensic analysis and real-time traffic monitoring. Forensic analysis is used on static network captures after an incident has occurred. This is often performed by browsing through the capture logs with tools such as Ethereal [5] and is considered a tedious process. Currently, we have used forensic Honeynet traffic captures from the Georgia Institute of Technology network [6] and the Honeynet Scan of the Month, [7] because they provide a good benchmark to test the effectiveness of the tool.

### Tool Description

A good visualization provides an overview of data by which to understand context and then provides more detail on demand. The data should be scaled and presented so that when an overall view is given, there is as little occlusion as possible in that view. Plotting data over time will show patterns and trends. Cumulative port statistics will show port activity.

Histograms are used because they are easy to interpret and good for visualizing large datasets. [8] Values can be compared, which is useful in visualizing time patterns. For three-variable plotting, we use 2D stacked, rather than 3D, for lower program complexity and processing and for more accurate value interpretation. In 3D, it is difficult to accurately determine values, as 3D is represented on a 2D surface, and this can permit an inaccurate perception. [4] The variables plotted on the graph are time, port count, and port number (or range) as illustrated in Figure 1.

### Preprocessing the Data

There are many network data parameters, and some of these variables have a large range of values. Because of this, the data must be scaled before it is plotted. In the overall graph, overlap and occlusion should be avoided to reduce confusion. Network traffic statistics are highly variable by nature. High values can skew the scale and hide values that are much lower. (For a comparison, see Figure 5 and Figure 6) To deal with this, cube root instead of a logarithmic scale is used to scale data quantities, because cube root can be applied to values from 0–1 and scaled to

positive values. We can also complement the scaling with information visualization best practices, such as filtering, zooming, and mouseovers, to deal with occlusion.

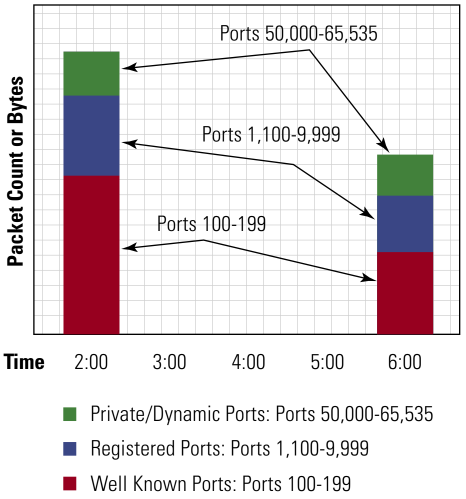


**Figure 1** Layout of the Visualization. The *x* axis represents time, while the *y* axis is an interval quantity of port count or total port bytes. Port-number ranges are grouped and mapped by color.

### Port Scaling

There are 65,536 possible port numbers, which makes it impossible to allocate each discrete value to one pixel on an axis. Port numbers have been grouped into ranges so that we can fit the range on the graph without losing context.

The well-known and commonly assigned ports (0–1,023) are grouped into bins of 100 per group. Most attacks start with these ports, which require more granularity in this range, and, because of this, we chose smaller groupings. The registered ports 1,024–49,151 can be used by an application or be connected to a server. This range is not as active as the well-known ports, so larger groups of 10,000 are used. Typically, no service should be assigned in the private or dynamic ports range (49,152–65,535), but these can still be used by malicious applications. (See Figure 1) These ports are divided into larger groups of 40,000–49,999 and 50,000–65,535. (The plot shown in Figure 1 is illustrative but not a plot of real activity for that port range.)

Singling out individual ports is a way to filter the graph. In Figure 5, the targeted ports of that time are separated from the other port ranges. These were chosen because a Honeynet traffic capture was used. In a regular network, ports that are used most of the time would be separated. This helps filter out high port counts from the other port ranges, which could drown out other possible anomalous activity that could be occurring in its respective port range.

### Time Scaling

Sampling rate and graph-update rate influence what kind of information is revealed in the data. A small time sample is good for quickly occurring activities such as fast network scans, Denial of Service (DoS) attacks, and fast- propagating worms. (See Figure 2 and Figure 3) A large time sample is better for viewing slow network scans and overall network trends over a long period of time.

Time is more crucial with real-time monitoring when activities happen quickly. However, with a time interval that is too small, too much detail may result, making it difficult to notice a pattern.

### Internet Protocol (IP) Address Scaling

Like port scaling, it is not possible to plot the four billion potential Internet Protocol (IP) addresses without filtering or scaling. A matrix method has been used in SnortView [9] and NVisionIP [10] to layout IP addresses across two perpendicular axes. VizFlowConnect [11] filters on a host and maps IP addresses on a parallel plot axis. Currently, we do not have IP address information in our tool, but we are considering filtering on IP addresses that actively connect to the local network.

### Results

We use typical attack captures from the Honeynet to show the effectiveness of our methods.

### Network Scanning and Mapping

A scan is more difficult to detect when performed on a network's commonly used ports. When the scan probes unused IP addresses and ports, this is clearer on the graph, because those ranges were never used before and would "be readily apparent."

Figure 3 shows 30 minutes of network scans. The popular network mapping tool (nmap) was used to perform Synchronize Flag (SYN), NULL, XMAS, and Transmission Control Protocol (TCP) Connect scans of all ports. The pattern of the way in which port ranges are targeted can be seen.

### Viruses, Worms, and Trojans

Distributed Denial of Service (DDoS) attacks are on the rise. Some occur when illicit Simple Mail Transfer Protocol (SMTP) servers are installed on a compromised host for Spam and Unsolicited Commercial E-mail (UCE), and some shut down other systems and the services they provide. Typically a DDoS attack is setup to compromise machines and gain control over them. Once control is established, those machines can be used to carry out attacks. In the graph, we would see higher traffic on the port/service being hacked, and then afterwards we would see activity on backdoor ports, which would be used to scan the other hosts and transmit and receive traffic. (See Figure 4)

### Backdoors and Rootkits

A botnet attack capture is used to illustrate the result of a successful takeover. We see an increase of traffic on those ports opened for use. For the most common botnets, the ports typically used are 6667 or any from 6660–6670.

In Figure 5, activity can be seen on ports 80 Hypertext Transfer Protocol (HTTP), 139 Network Basic Input Output System, (NetBIOS), 445 Server Message Block (SMB), 1434 Slammer/Microsoft Structured Query Language (MS SQL) Monitor, 4899 Remote Administration Tool (Radmin), and 28431 (Unallocated). The attackers exploited port 445 and installed a program that created an encrypted backdoor port on port 4899. They subsequently compromised Honeynet machines and then added  their Internet Relay Chat (IRC) botnet. In the last part of the graph on the right, we can see the successful botnet traffic on the IRC ports (light blue), which shows consistent network activity.

Figure 6 is a port-count graph of the Honeynet scan of the month. The botnet installation and subsequent botnet activity had a large number of packets transferred in and out of the network, thereby increasing the scale of packet counts and hiding the other port count values.

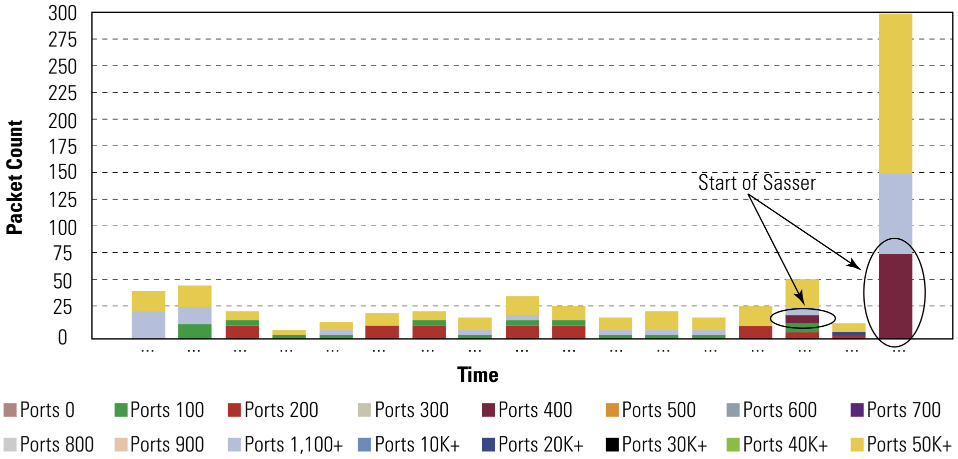


**Figure 2** Incoming port counts for every 5 minutes during the Sasser worm attack.This capture is from the GT Honeynet.

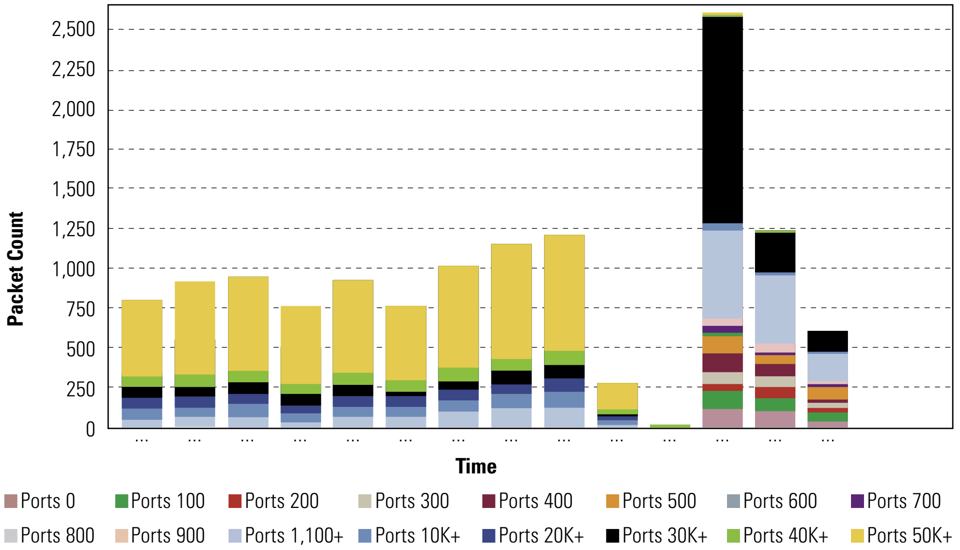


**Figure 3** Slow network scan occurring over 30 minutes, plotted every two minutes.

### Conclusion and Future Work

This tool has proven useful for detecting malicious activities that affect ports and for providing an effective overview of all port usage on a network. The tool can be used to determine anomalous behavior with an IDS and in situations in which human visual analysis can be used with anomaly-based algorithms and known signatures.

Non-port-based activity, such as illegitimate root access, cannot be detected with this tool alone. We would like to incorporate other packet header fields (*e.g.*, from ICMP and IP) for non-port-based attacks, implement more information visualization methods (*e.g.*, zooming and mouseovers), and conduct human-computer interaction studies. ■

### References

[1] C. Cooper, "Snoozing About Security," vol. I. CNET Networks, 2005.

[2] P. Lyman, "How Much Information 2003?" October 2003.

[3] P. Jungck & S. Shim, "Issues In High-Speed Internet Security," *Computer*, vol. 37, pp. 36–42, 2004.

[4] R. Spence, *Information Visualization*. England: ACM Press, 2001.

[5] G. Combs, "Ethereal," Open Source, GPL.

[6] J. Levine, H. Owen, D. Contis, and B. Culver, "The Use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks," Proceedings of the IEEE Workshop on Information Assurance, West Point, NY, 2003.

[7] "Honeynet project: Scan of the month," 2004.

[8] D. Keim, M. Hao & U. Dayal, "Hierarchical Pixel Bar Charts." *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, pp. 255–269, 2002.

[9] H. Koike & K. Ohno, "Snortview: Visualization System of Snort Logs," VizSEC/DMSEC'04, Washington DC, USA, 2004.

[10] K. Lakkaraju, W. Yurcik, A. Lee, R. Bearavolu, Y. Li, & X. Yin, "NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness," VizSEC/DMSEC'04, Washington DC, USA, 2004.

[11] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju " VisFlowConnect: NetFlow Visualizations of Link Relationships for Security Situational Awareness," VizSEC/DMSEC'04, Washington DC, USA 2004.

## About the Authors

**Ms. Kulsoom Abdullah** | is a graduate research assistant at the Georgia Institute of Technology Communications Systems Center (*http://www.csc.gatech.edu/*). She is completing her PhD, and her research focuses are network security and visualization. Her research may be found at *http://users.ece.gatech.edu/~kulsoom/research.html*. She may be reached at kulsoom@gatech.edu.

**Mr. Gregory Conti** | is an Assistant Professor of Computer Science at the US Military Academy, West Point, NY. He is currently at the Georgia Institute of Technology on a Department of Defense Fellowship where he is completing a PhD in Computer Science. His research may be found at *http://www.gregconti.com*.He may be reached at conti@cc.gatech.edu.

**Dr. John Copeland** | is the John H. Weitnauer, Jr. Chair at the Georgia Institute of Technology School of Electrical and Computer Engineering. In 2000, he co-founded Lancope, Inc., (*http://www.lancope.com*). His research interests include information visualization for computer security, network security and high-speed optical networks. Copeland has a BS, MS, and PhD in physics from the Georgia Institute of Technology. He is a Fellow of the IEEE, and received the Morris N. Liebmann award in 1970. He may be reached at john.copeland@ece.gatech.edu.

**Mr. Chris Lee** | is a graduate research assistant at the Georgia Institute of Technology Communications Systems Center under Dr. John A. Copeland. He is completing his PhD and his current research focuses on security visualizations for ubiquitous deployment of security systems. His research may be found at *http://www.csc.gatech.edu/people/chrislee.html*. He may be reached at chris@ece.gatech.edu.

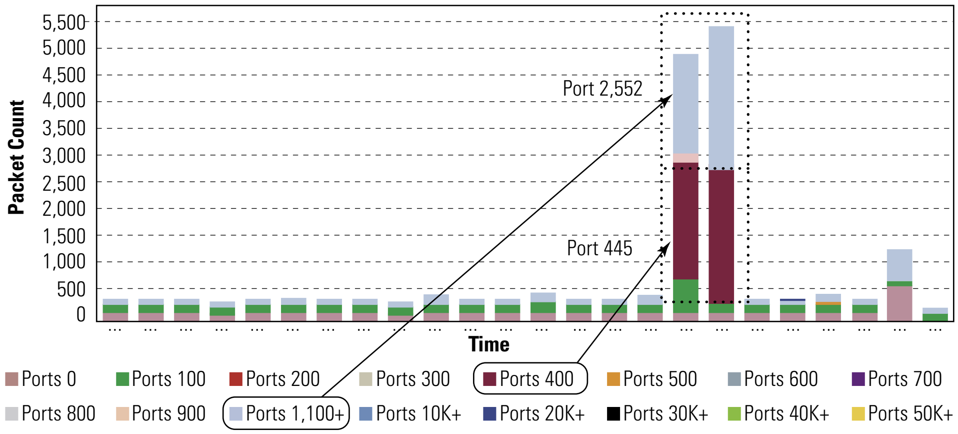


**Figure 4** Sasser attack. This shows normally occurring probes and chatter. The spikes indicate a significant increase in the number of packets destined for two port ranges (the incoming on port 445 and the outgoing on port 2552).

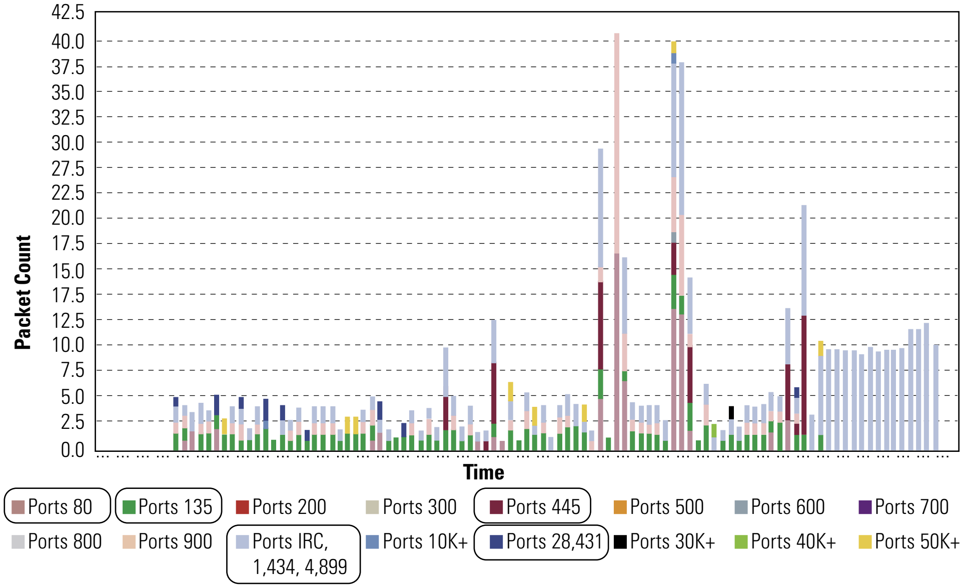


**Figure 5** Stacked Histogram of Botnet Attack (Normalized)

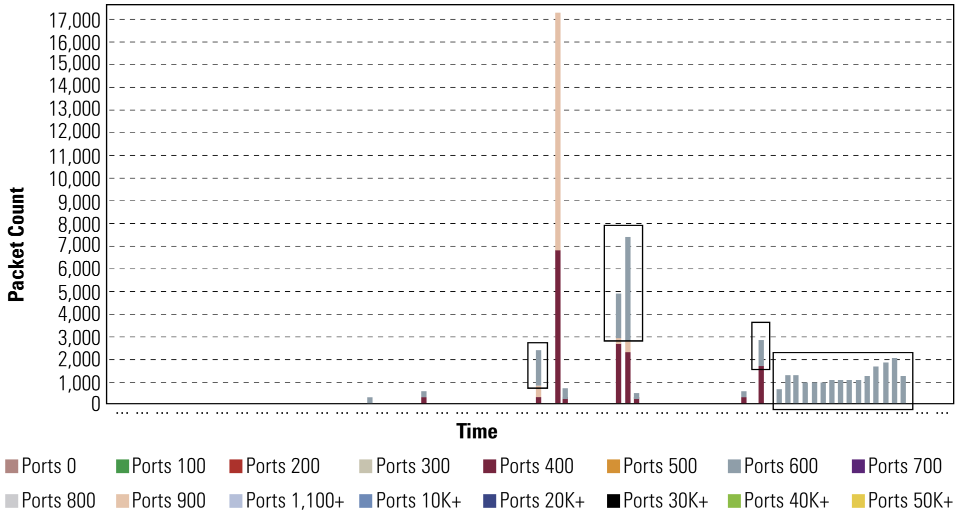


**Figure 6** Botnet Traffic

# GIG-BE—Improving the Warfighter's Information Pipeline

by David Smith

The Global Information Grid-Bandwidth Expansion (GIG-BE) Program is a Department of Defense (DoD) net-centric transformational initiative managed by the Defense Information Systems Agency (DISA). GIG-BE provides a ubiquitous "bandwidth on demand" network environment and protected sharing of surveillance, reconnaissance, and command and control information, thereby improving national security intelligence. This article discusses GIG-BE and the key strategies being implemented to improve bandwidth availability and enhance information assurance capabilities. The execution of GIG-BE greatly improves the end-to-end performance of the Defense Information Systems Network (DISN) by increasing bandwidth and physical diversity to selected locations worldwide. GIG-BE aims to increase bandwidth and provide diverse physical access to more than 100 sites in the Continental United States (CONUS) and in the Pacific and European theaters. The GIG-BE (DISN core) interconnects these sites; in particular, key intelligence, command, and operational locations in which high-bandwidth capability over physically diverse routes is essential. The majority of these locations are connected by a state-of-the-art, dense wavelength division, multiplexed optical network that provides high availability and survivability.

Figure 1 provides an overview of DoD's GIG. The next-generation DISN backbone depicted within the figure is provided by the GIG-BE. Specific DISN architecture goals served by the GIG-BE program include the following:

- ▶ Increasing the survival rate of DISN services by responding to increased threats that exist against GIG resources, both overseas and in the US
- ▶ Using economies of scale provided by the GIG-BE investment to improve "best value" supplied to the warfighter
- ▶ To structure and better position DISN services to evolve into Net-Centric Enterprise Services (NCES)

As a global, high-capacity, optical-communications system, the GIG-BE program reduces bandwidth constraints for data exchange and provides significantly increased bandwidth to support combat operations. Moreover, it provides physically diverse routing services for DoD and the Intelligence Community by expanding DoD's core telecommunications backbone. This greatly enhances the reliability and the survivability of the entire network. GIG-BE's state-of-the-art, optical-network design is achieved with double and even quadruple connections throughout the US and Europe. Diverse routing capability substantially decreases the chances of a single point of failure.

The government effectively owns both the fiber and the optical equipment deployed in GIG-BE, as compared with previous network implementations that relied on leased or contracted services and equipment. This enhances DoD's capabilities for positive control of the network and facilitates its ability to improve or change the network's security and perfor-

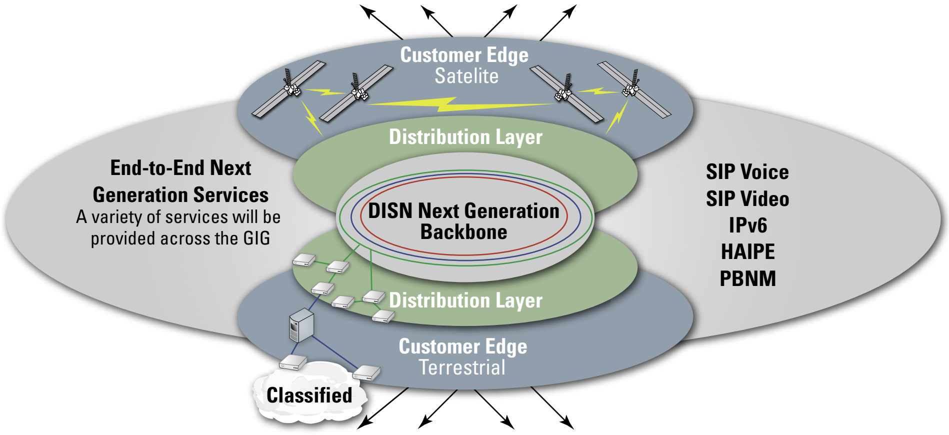


**Figure 1** Overview of Global Information Grid (GIG)

mance. GIG-BE is also critical to using "reach-back" capability, which will maximize investments in airborne Intelligence, Surveillance, and Reconnaissance (ISR) and future combat-support assets.

With the expanded bandwidth provided by GIG-BE, DISA can address high-capacity applications (*e.g.*, imaging, video streaming) and provide a higher degree of network security and integrity. The GIG-BE program also contributes significantly to meeting highly remote ground-transport needs for voice, video, and data.

The GIG-BE program is the first of its kind to bring high-speed High Assurance Internet Protocol Encryptor (HAIPE) devices to a DoD network. The HAIPE devices, introduced because of the National Security Agency's anticipatory development, will greatly increase the ability to bring secure, net-centric capabilities to the Intelligence Community and DoD operations.

The combination of data types that can be handled by GIG-BE permits the military to create a much smaller footprint in combat situations, because all types of communications—voice, video and data—can be accommodated by GIG-BE's IP-centric network design. For DoD, the advantages of the GIG-BE/DISN migration includes increased efficiencies, and advanced preparation for the changing face of combat. Figure 2 shows an example of a GIG-BE/DISN node. DISN services are migrated onto the GIG-BE backbone, allowing for managed network elements and consolidated service-delivery points. For the end user, the ultimate objective is to take the issue of bandwidth limitation out of the equation so that service remains consistent as needs increase.

The GIG-BE program delivers much-needed capabilities to decision makers and warfighters alike. It represents a clear and continuing commitment among Defense and Intelligence Community leaders to achieve net-centric capabilities. ■

## About the Author

**Mr. David Smith's** | experience includes four years in optical engineering specializing in Synchronous Optical Network (SONET) and Dense Wavelength Division Multiplexing (DWDM) applications. He supports DISA as an optical engineering Subject Matter Expert (SME) and assists the DISN Transition Office with the challenges of migrating from legacy systems to SONET. He also leads a planning effort that looks at each site and formulates a strategy to transition DISN services onto GIG-BE. Mr. Smith was an Enterprise Systems Engineer with Technica Corporation, where he worked on various optical projects with DoD, the White House Communications Agency, and DISA.

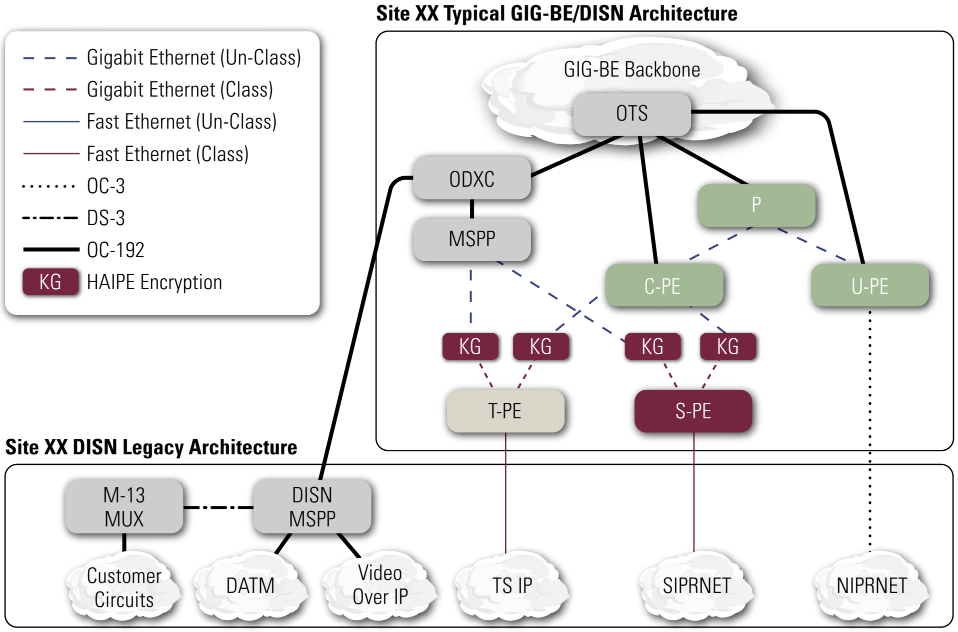


**Figure 2** New GIG-BE/DISN Node

# What is Secure Software?

by Karen Mercedes Goertzel

Secure software is software that cannot be intentionally forced to perform unintended functions. The motivation for producing secure software is the desire for software that can continue to operate correctly even when subjected to attack. This means that software at the level of granularity of an individual component, program, or application will be attack *resistant*. Attack resistance means the software will—

- Recognize attack patterns
- Avoid or withstand attack

Entire software *systems*, to be secure, must not only be attack resistant, they must also be attack *resilient*. This means they can recover from an attack by resuming operation at or above a minimum level of service as soon as the source of the attack has been isolated and blocked and damage has been contained.

Software that is *not* secure contains defects that can be exploited by an attacker, whether a person or a malicious process. These defects may be at the design level, the implementation level, or the configuration level.

## Why Does Software Security Matter?

Software, particularly software at the application level, is being increasingly targeted by attackers because—

- It has become difficult to compromise the security of networking and operating system software. Security measures for protecting operating systems, networks, and some middleware (*e.g.*, database management systems, Web servers) are proving effective in resisting many attacks.
- Software has increased hugely in value but not in security robustness. Because of its ubiquity and increasing criticality, attackers have begun to recognize that software's value as a target has greatly increased and continues to increase. Users trust and rely on software for nearly every business function they perform (as well as a number of non-business functions). Yet software today is no better (and, indeed, is arguably worse) in quality (correctness), reliability, and security than it was 10 years ago, when it wasn't being routinely exposed to the Internet.
- There is a higher chance of success in targeting software, because software technologies and their vulnerabilities have become so familiar. Attackers study the widely reported exploitable defects (vulnerabilities) in commonly used software technologies and products and thus can craft effective attacks against software systems that incorporate those technologies and products.

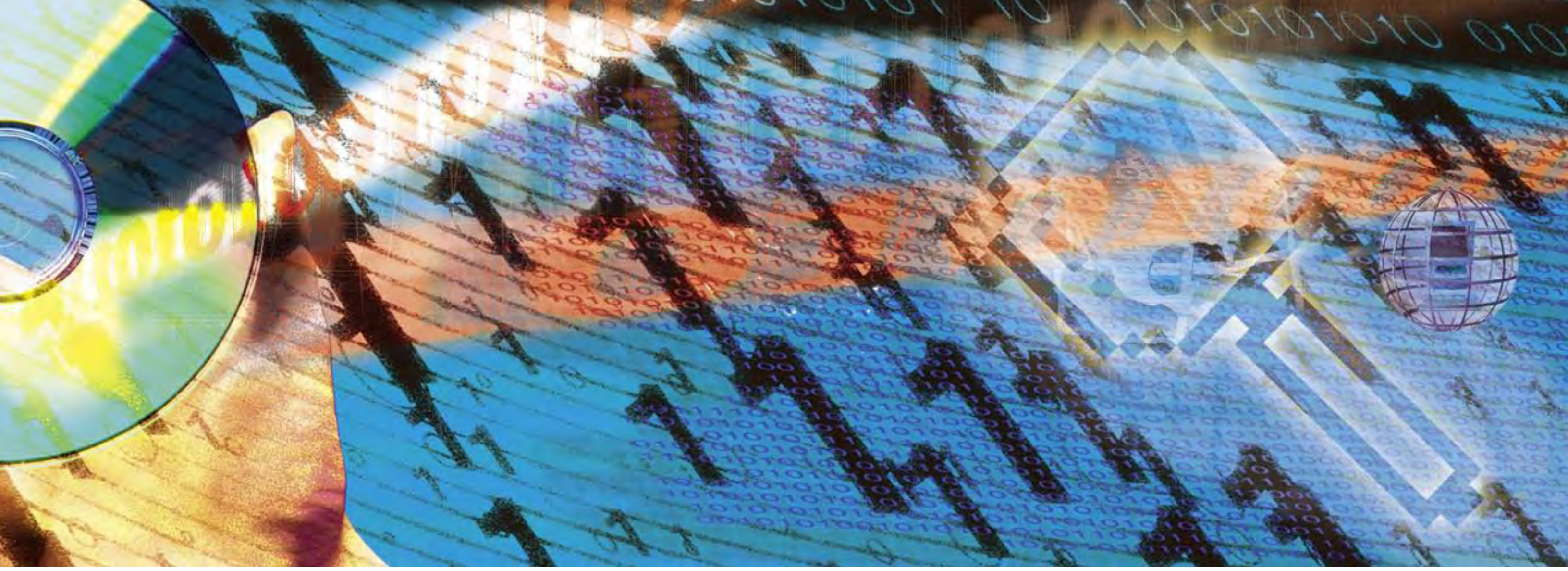
## Why So Much Software is Not Secure

With the exception of high-consequence software, most software in use today is the result of artistry, not engineering. (See box below) The typical software-development process far more closely resembles musical composition than engineering. The composer follows general rules of music theory but is otherwise unconstrained by anything except his imagination, the extent of his need for self-expression, and the imperative to finish his commissioned symphony in time for its first rehearsal so he can get paid.

It isn't that most developers consider themselves artists; *most* want to be engineers. But they are at the mercy of forces beyond their control—their managers' profit or cost-savings motive and/or their users' unrealistic demands. As a result, they often manage to do only the bare minimum needed to turn out software that does what it's supposed to more or less reliably, while keeping their efforts on schedule and within budget.

Even when quality—*correctness*—is considered as important (never *more* important) as profit and/or cost savings or user demand, it is defined as "The software operates correctly and satisfies all of its functional requirements," or "It does everything it's supposed to do when operating under expected conditions."

Correctness is seldom if ever defined as "The software never does anything it isn't supposed to do," or "It never does



anything it *isn't* supposed to do, even when its environment changes"—much less as "It cannot be intentionally forced into doing something it isn't supposed to."

Forcing software to do what it's not supposed to is exactly what attackers hope to accomplish. And the software, with its defects and flaws and errors and faults and bugs, cannot stop them from succeeding.

### Why Attackers Target Software

Successful attacks on software systems result from human ingenuity—a subtle design defect may be exploited, or a previously undiscovered implementation defect may be located through an attacker's engineering efforts. The majority of attacks have one or more of the following objectives: (See Table on page 12)

In an increasing number of cases, an attacker is an authorized user who constitutes what is commonly referred to as an *insider threat*. Protections against *un*authorized use combined with application security measures that detect and block externally originated attack patterns will not prevent such insider attacks, because insiders already have privileges authorizing them to access the software's network and host and to execute the software. Having the advantage of a position of greater access and privilege, insiders' attacks have a high likelihood of success and the worst potential consequences.

The insider threat is not limited to targeting software in deployment, however. Developers may be motivated by malice, criminal intent, or worse to embed malicious processes in software before it is deployed. The *rogue-developer* threat is a key reason to perform security reviews of source code before deployment and to strictly enforce secure configuration management of source code and executable images throughout the software's lifetime.

### High-Consequence Software

High-consequence software is software whose failure could result in serious harm to a human being. That harm may take the form of loss of life, physical injury or damage to health, loss of political freedom, loss of financial well-being, or disastrous damage to a human's environment.

Because of its critical nature, high-confidence software is subjected to much careful reliability and safety engineering and operational management, including rigorous patching, cautious installation-time configuration, and frequent post-deployment configuration checking. Examples of high-consequence software include software components of national security systems, medical control systems, and Supervisory Control And Data Acquisition (SCADA) systems.

Large-scale software systems that support a very large number of users are sometimes also considered of high consequence, not primarily because it would be so difficult to recover such a system to normal operation after a failure, but because it would be extremely difficult or costly to make reparations to users for damages resulting from such a failure. An example of this type of high-consequence system is an electronic voting system.

| To | By |
|---|---|
| Compromise confidentiality | Reverse engineering the software to discover how it works to more effectively target it |
| Compromise access control | Gaining access to software artifacts (executables, configuration files, *etc.*) an attacker isn't supposed to access |
| Compromise authorization | Performing software functions an attacker is not supposed to perform |
| Compromise availability | Rendering the software inoperable or inaccessible to its valid users, also known as Denial of Service (DoS) |
| Compromise integrity | Subverting the software's functionality to achieve one of the other compromises |
| Elevate privileges | Attempting to gain privileges an attacker shouldn't be permitted to have to accomplish one of the other compromises |

**Table 1** Threats to Software

### What Makes Software Vulnerable?

In their Report to the President, Cyber Security: A Crisis of Prioritization (February 2005), in the chapter entitled "Software Is a Major Vulnerability," the President's Information Technology Advisory Committee (PITAC) summed up the problem of insecure software concisely and accurately:

(See Callout box on page 13)

Software—and especially networked, application-level software—is most often compromised by the intentional exploitation of—

- Inherent deficiencies in the software's processing model (*e.g.*, a Web or service-oriented architecture model)
- Design or implementation defects in execution environment components, including components at the middleware, operating system, network device and protocol, firmware, and hardware levels
- Design or implementation defects in the software's interfaces with other environment-level and application-level components
- Design or implementation defects in the software's interfaces with its users (humans and software), especially defects that enable user input to avoid or subvert the software's input validation or error handling

Because of simple human fallibility, some number of exploitable software defects can be virtually guaranteed to be introduced into software during its implementation and deployment, even if its design can be deemed free of vulnerabilities. Moreover, if software continues to grow in size and complexity, at least some of these defects will elude detection even by the most rigorous and extensive testing regime. While it may be possible, through careful, extensive code review, to discover all defects in custom-developed code, discovering comparable defects in nondevelopmental components—particularly binary components—is for all intents and purposes impossible.

An exploitable defect will always represent a vulnerability. It is *not* true, however, that all vulnerabilities arise from exploitable defects. Consider the appearance of previously unobserved vulnerabilities after the software is re-hosted to an environment other than that for which it was designed. Such unanticipated vulnerabilities do not result from inherent defects in the software. Instead, they arise from violating, in actual deployment, assumptions made by the software's designers about how the software would be deployed. Similarly, some vulnerabilities appear only when the software's threat environment changes. Such vulnerabilities may result from previously undetected defects but are more likely to result from novel and ingenious attacker exploitation of wholly valid software features and interfaces. Software is no less vulnerable just because the means by which it was compromised were not inherent defects.

The incorrectness or incompleteness of developer assumptions often relate to this failure to anticipate how the software will behave under *all* possible conditions, not just conditions associated with normal use, which is a key source of exploitable defects (vulnerabilities) in software. For example, developers make assumptions about what state changes can and will occur in their software's execution environment and in their software as it responds to those environment state changes. These assumptions seldom include anticipating the kinds of state changes associated with intentionally induced faults.

The disciplines of threat modeling, attack-tree generation, and the modeling of abuse and misuse cases are all intended to help developers craft more accurate assumptions about all potential environment state changes, software state changes, and faults that could occur during the software's execution. Unfortunately, security testing is unlikely to be as effective in finding or anticipating non-defect-related vulnerabilities *vs.* finding simple exploitable defects. The imagination, time, and resources available to most testers does note permit them to exercise the necessary multiplicity of complicated, wholly speculative test scenarios. Given business realities of limited resources and short development schedules, the extra time and resources required to perform effective security testing on software are best focused on the following:

- High-consequence software, *i.e.*, software in which a high degree of trust will be placed. For example, the components that perform security functions and those that access or manipulate sensitive data or resources *Note:* Federal Information Processing Standard (FIPS) 199, *Standards for Security Categorization of Federal Information and Information Systems*, describes a process for categorizing a software system's impact level. This process can be adapted to prioritize the consequence of software components within a software system.
- Interfaces between the system's components

- Interfaces between the software system and its environment and between the software system and its users

Network connectivity provides "door-to-door" transportation for attackers, but vulnerabilities in the software residing in computers substantially compound the cyber security problem. As the PITAC noted in a 1999 report, the software development methods that have been the norm fail to provide the high-quality, reliable, and secure software that the Information Technology infrastructure requires.

Software development is not yet a science or a rigorous discipline, and the development process by and large is not controlled to minimize the vulnerabilities that attackers exploit. Today, as with cancer, vulnerable software can be invaded and modified to cause damage to previously healthy software, and infected software can replicate itself and be carried across networks to cause damage in other systems. Like cancer, these damaging processes may be invisible to the lay person even though experts recognize that their threat is growing. And as in cancer, both preventive actions and research are critical, the former to minimize damage today and the latter to establish a foundation of knowledge and capabilities that will assist the cyber security professionals of tomorrow reduce risk and minimize damage for the long term.

Vulnerabilities in software that are introduced by mistake or poor practices are a serious problem today. In the future, the Nation may face an even more challenging problem as adversaries—both foreign and domestic—become increasingly sophisticated in their ability to insert malicious code into critical software.

### Security as a Property of Software

Software can be characterized by its fundamental properties such as "functionality," "performance," "reliability," "cost," "usability," "manageability," "adaptability," and, of most interest to us, "security." The main objective of all software security practices is to ensure that behaviors associated with software's internal and external interactions are always secure in demonstrating all security properties considered desirable for software.

Security of software is considered an *emergent* property. Emergent properties are those that derive from the interactions among the parts—*components*—of software. As an emergent property, security of the software as a whole cannot be predicted solely by considering the software's separate components in isolation.

The security of software can only be determined by observing its behavior as a collective entity under a wide variety of circumstances. Such observation should reveal whether the desired property, in fact, *emerges* from the collective behavior of software's components as they prepare to interact, respond to, and recover from interaction. There are other desirable properties of software which, while they do not directly make software secure, make it possible to *assure* that software is secure:

- **Predictability**—Predictable software will never deviate from correct operation under anticipated conditions. Predictability is also critical if security is to be assured: unless the software can be trusted to operate as it is supposed to under normal conditions, determining that it also continues to operate correctly under abnormal conditions will be of little value.
- **Simplicity and traceability**—Software that is simple, with process and data flows that are traceable, is easier to comprehend in its internal operation and external interactions. Simplicity and traceability make it easier to discover exploitable defects, insecure behaviors and interactions, and insecure state changes and to quickly identify and implement remediations for those defects.
- **Correctness**—From the standpoint of quality, correctness is a critical attribute of software that must be consistently demonstrated under all anticipated operating conditions. Several advocates for secure software engineering suggest that quality engineering is all that is needed to produce secure software. Their thinking is that, by reducing the total number of defects in software, quality engineering will necessarily reduce some percentage of exploitable defects. However, by focusing on defect removal, software developers are very likely to overlook complex vulnerabilities such as those caused through a series of interactions among processes or components.

  Because correctness in software is demonstrated only under anticipated operating conditions, quality engineering does not attempt to demonstrate that the same software will remain correct under conditions that are *un*anticipated. Software operation that is correct under anticipated conditions, but which becomes incorrect under unanticipated conditions, cannot be considered secure. And it really shouldn't be considered correct, either.
- **Reliability and Safety**—High reliability, fault tolerance, high confidence, safety…when it comes to software, they all have the same objective: to sustain predictable, dependable execution in the face of unpredictable but *unintentional* faults.

  By contrast with reliability and safety, software *security* is concerned with sustaining predictable, dependable program execution in the face of *intentional* faults. These faults may result from defects or malicious logic planted in the software by a rogue developer, or they may be induced in the executing software by an attacker. *Intentionality* is at the core of what makes requirements for software security different from those for software reliability. Intentionality is also the reason why the ability of highly reliable software to handle stochastic (unintentional) faults extremely well is not sufficient for assuring software security.

The software reliability and safety communities are coming to recognize that handling of stochastic faults is also insufficient to assure reliability. Any network-accessible, safety-critical system that contains even a single exploitable defect cannot be considered safe, even if that defect cannot possibly lead to a stochastic fault. This is because there is no way of guaranteeing that the same defect cannot be used to intentionally induce a fault. Unless software can be assured to be reliable under *all* fault conditions, including conditions in which faults are intentionally induced (non-stochastic), that software cannot truly be considered reliable.

### Secure in Development *vs.* Secure in Deployment

A growing industry has emerged to address the need for what has been dubbed application security. Application security is not about making software less vulnerable. Instead, it relies on *secure-in-deployment* techniques to protect vulnerable software against exposure to threat agents.

Application-security measures focus on strengthening the protective boundaries around applications, *e.g.*, through deployment of application firewalls, vulnerability scanners, and host-based Intrusion Detection Systems (IDSs), to detect and then block or filter input that might be intended to exploit those vulnerabilities. Application security also focuses on constraining the amount of damage that will be caused when dangerous input does manage to reach the application, *e.g.*, by executing the vulnerable portions of the software within a virtual machine or *sandbox*. In short, application security reproduces at the application layer the same kinds of in-deployment security protections that have long been used at the network and operating system layers.

Application security techniques have the same strengths and shortcomings as their lower-layer counterparts. Application security techniques that focus on blocking input and constraining the extent of damage do not get to the heart of the problem because—

- They do nothing to eliminate the vulnerabilities that put the software at risk in the first place
- Dangerous inputs are only one potential way in which application compromise can be triggered
- Attackers are becoming increasingly ingenious at crafting inputs that appear to be completely innocuous but are designed to trigger a series of events within the application that will lead to its failure or subversion

*Software* security's first objective is to ensure that software is *designed* and *implemented* so that, by the time it is deployed, it will contain few vulnerabilities that can be exploited in *any* way. In software security, priority is placed on secure-in-development techniques so that secure-in-deployment techniques can be limited in focus to mitigating the few residual vulnerabilities that were not (or could not be) avoided or eliminated through secure development.

### Security in the Development Life Cycle

Until they adopt a disciplined, repeatable security-enhanced development process, most software development organizations will need to rely heavily on application security technologies to protect their deployed software against compromise. In their article, *Security Guidance for .NET Framework 2.0,* J.D. Meier *et al.* assert—

> *To design, build, and deploy secure applications, you must integrate security into your application development life cycle and adapt your current software engineering practices and methodologies to include specific security-related activities.*

As the experiences of software development organizations such as Microsoft (with its Security Development Lifecycle) have demonstrated, instituting security best practices throughout the development life cycle does produce demonstrably less vulnerable software.

Enhancing the security of activities that comprise the software development life cycle generally entails shifting the emphasis and expanding the scope of existing life-cycle activities so that security becomes as important as other objectives to be achieved in the developed software.

Risk-management activities and checkpoints must be integrated into the software's development activities—

- By starting with developing threat models, attack trees, and abuse and misuse cases to help drive security requirements and test plans for the software
- By iterative security reviews and testing throughout all of the software's conceptualization and implementation phases
- By conducting impact analyses associated with maintenance, including impact analyses before patching, component replacement, or satisfaction of new user requirements.

Adopting a security-enhanced life-cycle process model and supporting development methodology (or methodologies) can increase the likelihood that software produced by that process will be more secure. Fortunately, a number of efforts have been undertaken to adapt, extend, or define secure life-cycle process models and methodologies. Most noteworthy among these are—

- The Federal Aviation Administration (FAA) Proposed Safety and Security Extensions to the Integrated Capability Maturity Model (iCMM)/Capability Maturity Model integration (CMMI)
- The emerging International Standards Organization (ISO)/ International Electrotechnical Commission (IEC) Standard 15026, which defines a set of Assurance Activities and 19 Security Practice Areas that can be appended to the ISO/IEC Standard 12207 (Software Lifecycle Processes) or ISO/IEC Standard 15528 (System Engineering Lifecycle) processes.
- Microsoft's Trustworthy Computing Security Development Lifecycle (SDL)
- John Viega's Comprehensive, Lightweight Application Security Process (CLASP)

By contrast with life-cycle process models and full life-cycle methodologies, many software development methodologies have a narrower focus and applicability, such as defining lower-level, functional, and technical constraints for individual life-cycle phase practices, including requirements specification, design, and coding. With very few exceptions, these methodologies share a focus with process capability models of improving software *quality*, not software security.

This said, there are efforts under way to enhance security—or re-purpose— various features of existing development methodologies so that they expressly produce more secure software. The Department of Homeland Security (DHS) has released for public comment the final draft of Security in the Software *Lifecycle: Making Application Development Processes—and Software Produced by Them—More Secure*. This document includes extensive information on security-enhanced life-cycle process models and the secure use of popular development methodologies, including agile methods, object-oriented methods, aspect-oriented methods, and formal methods. The draft can be downloaded from BuildSecurityIn, https://buildsecurityin.us-cert.gov/portal/, sponsored by DHS. The final revised version, based on received public comments, is expected to appear in March 2006.

In summary, to be truly effective in improving the security of software, any software engineering process, method, or practice must intentionally and methodically address security. ■

## About the Authors

**Ms. Karen Mercedes Goertzel, CISSP,** | has 24 years in IA, software assurance, and net-centric architectures and applications. She supports the Department of Homeland Security and ASD(NII) as a software assurance subject-matter expert, and was lead technologist for 3 years on DISA's Application Security Program. She also leads ASD(NII)'s effort to define an approach for two-level GIG network management. Ms. Goertzel was lead author of DISA's 2002 IA Technology Forecast, Future IA Architecture, and IA Technology Roadmap. She has consulted in IA, software assurance, and CDS to DISA's Net-Centric Enterprise Services program, the National Geospatial Agency, NSA, the US/Canadian Anti-Drug Network, and Joint Staff, among others. In 2004, she authored *IAnewsletter* articles on Autonomic Computing and Computer Immunology. Previously, Ms. Goertzel was a CDS engineer for Getronics (now BAE/DigitalNet). For Wang and Honeywell she consulted to DoD, Civil agencies, and NATO on security architecture and policy, risk management, COOP, and software development.

# CPOL: High-Performance Policy Evaluation

by Kevin Borders, Xin Zhao, and Atul Prakash

Design of policy-evaluation systems has been a focus of the security community. However, the performance of such systems has generally not received much attention. Applications are emerging, such as privacy enforcement for a real-time location-tracking infrastructure, in which performance is an issue. Current policy-evaluation systems are unable to deliver the required throughput for these applications. This article presents a new policy-evaluation system, CPOL, which focuses heavily on delivering high performance. CPOL was developed at the University of Michigan under a National Science Foundation (NSF) infrastructure project to deploy a location-sensing network in the new computer science building that enforces privacy policies on behalf of the users.

Enforcing privacy in a location-sensing network presents unique challenges. Unlike traditional access control, access to a person's location information may depend on the current environment and state of the user such as the time of day and the user's location. For example, employees may not want their locations to be known after work hours or when they are outside of their workplace. Depending on the number of users, location-sensing networks also have the potential to receive a large number of queries. During one of our simulations, 1,000 users occasionally monitored the locations of friends, looked up information about others nearby, and browsed buildings to look for empty labs, conference rooms, or places to study—and generated 50,000 requests in one second Determining whether or not to grant access to a user's location by itself can be complicated because of the potential complexity of access conditions. This problem is exacerbated even further by having a large number of requests.

CPOL is a promising new policy-evaluation system designed to handle heavy workloads while maintaining a high level of expressiveness.

Our first step in solving the privacy-enforcement problem was to evaluate existing solutions. We looked at the KeyNote Trust-Management System and at the MyStructured Query Language (MySQL) Database Management System (DBMS) as potential candidates. These systems can express complex access conditions based on a user's location and time of day; however, they fall short of handling the large number of requests that are likely to be seen in a real location-sensing network. The current implementation of KeyNote is very inefficient because of a lack of indexing, run-time policy parsing, and policy chaining (*i.e.*, evaluating multiple policies in a "chain" to obtain one right.) In our experiments with the KeyNote, we found it could only handle a few requests per second with 500 users in the system. With only 100 users, it performed much better, but 100 users falls far short of what would be required for a medium or large location-sensing network.

After evaluating KeyNote, we proceeded to configure a MySQL DBMS to enforce privacy policies. A DBMS is convenient because it already has the power to manage large datasets and is flexible enough to allow for expressive access conditions. Restrictions based on time of day and a user's current location can be placed in policy-table entries and evaluated at query time using Boolean "WHERE" clauses. Despite its being fairly straightforward to set up and use, the maximum throughput of a dedicated and optimized MySQL DBMS is only a few thousand queries per second, which is not enough to handle real-time queries for a moderately sized location infrastructure.

To address the need for a high-throughput evaluation system, we created CPOL, a flexible C++ framework for policy management. In the remainder

CPOL Access Policy Fields

| | |
|---|---|
| Owner | The owner is the entity whose resources are controlled by this rule. |
| Licensee(s) | The licensee is the entity or group that will receive privileges. If multiple licensees are specified, then all licensees must request access together for the rule to apply. |
| Access Token | The access token contains information about the rights assigned by this rule. |
| Condition | CPOL verifies that the condition is true before granting the access token to the target |

**Table 1** A CPOL Access Policy has four fields: a Rule Owner, a Rule Target, an Access Token, and a Condition. Policies govern access to all entities in the system.

of this article, we focus on using CPOL to evaluate privacy policies for location-aware services. CPOL, however, is a general-purpose system and can be applied effectively in a variety of other domains such as mobile messaging policies, firewall rules, and file access control.

The main goal of CPOL is to deliver good throughput with a strong emphasis on maintaining expressiveness. Our experiments, which are described in more detail later in this article, indicate that CPOL can handle requests from clients in a medium-to-large location-sensing infrastructure of 10,000 or more users. In addition, CPOL can also express a wide range of policies so that users are able to have fine-grained control over their private information. The expressiveness of CPOL policies is comparable to those of KeyNote. Some features supported by CPOL include roles, arbitrary access conditions, and privilege delegation.

One key design feature of CPOL that enhances its performance is a *policy-evaluation cache*. CPOL can cache results more effectively than other systems, while still maintaining correctness, because it is better at determining when to invalidate entries. A typical DBMS cache, for example, only works if query parameters stay exactly the same between requests. If one query parameter is the current time, then caching in a real-time system becomes completely ineffective. With CPOL, however, an application developer can restrict the domain of access conditions. This helps, because a single policy can be forced to use a limited set of time intervals, which allows the caching subsystem to calculate an accurate time-to-live for each entry, significantly increasing the hit rate. A similar mechanism can be used with conditions on other variables to calculate how much they can change before invalidating the result.

## CPOL Policies

To provide a context for further discussion, we will first examine the contents of a CPOL policy. Table 1 shows the four fields of a policy. Owners and licensees are both entities in the system and are identified by unique integer values. (A licensee can also be a role.) An access token is an object containing a set of access rights. The contents of an access token are defined by the application developer. In the case of location-policy enforcement, an access token contains the resolution of access to the owner's location (exact room, building, or city); the level of access to the owner's identity (anonymous, first name, or full name); and any additional administrative privileges.

An access condition is also an object defined by the application developer. It can contain an arbitrary Boolean expression or a list of simple conditions. Depending on the system's state, the condition will either evaluate to *true* or *false*, thus determining whether or not to grant the access token to the licensee. For the privacy-enforcement application, the condition is restricted to a set of location modifiers and a time modifier. The location modifiers specify that the condition should be *true* or *false* if the owner is in a particular room, floor, or building. The time modifier objects contain an interval during which the condition can be *true* and a weekday mask. The weekday mask restricts the days of the week to which the time interval applies. This allows specification of time modifiers such as “Monday through Friday 9 AM to 5 PM.” To achieve efficient cache invalidation (discussed in more detail later in this article), note that the contents of a condition are limited to a fixed number of modifiers.

Figure 1 shows an example of a policy for the privacy-enforcement application. Here a user, Alice, gives Bob access to her location. The access right permits Bob to view Alice’s location with room-level accuracy and also to see her full name. The condition associated with this access rule, however, specifies that Bob should only be given the access token if it is between 9 AM and 5 PM, Alice is in the Library or the CS Building, and Alice is *not* in Conference Room 1010 CS. Each time Bob wants to access Alice’s location, CPOL verifies that the condition is still true, given the current state of the system, to ensure compliance with Alice’s privacy policy.

```
Owner: Alice
Licensee: Bob
AccessToken {
        LocationResolution = RoomLevel
        IdentityResolution = Name
}
Condition  {
        AfterTime = 9 AM
        BeforeTime = 5 PM
        InBuilding = {Library, CS}
        NotInRoom = {ConferenceRoom 1010 CS}
}
```

**Figure 1** CPOL policy giving Bob access to Alice’s location with room-level precision, with no delegation privileges, 9 AM–5 PM, when Alice is in the library or the Computer Science (CS) Building, except in Conference Room 1010 CS.

## Caching in CPOL

One key element of CPOL’s design is the *policy-evaluation cache*. Storing recent results can speed up evaluation time tenfold on standard policy sets . Effectively caching query results, however, does present a significant challenge because of access conditions. The cache entry containing the most recent access token must be invalidated immediately when the condition granting access becomes false *or* when a condition giving a higher level of access that was previously false becomes true.

CPOL is able to quickly and correctly cache policy-evaluation results by using a special object called a *cache condition*. A cache condition is similar to a normal access condition but much smaller. For example, in the privacy-enforcement application, instead of containing a time interval during which access is valid, the cache condition contains a time-to-live value. Also, instead of containing full location conditions, it only records the resolution of the most specific location condition (the room-level in the condition from Figure 1) and invalidates the entry when an owner’s location changes by more than the specified resolution; *i.e.*, when the owner changes rooms for a room-level condition.

The full cache condition fits in 16 bits (2 bytes) and is stored along with the owner, the licensee, the access token, and the previous system state (the time and owner’s location) at which the condition was true. Each cache entry is uniquely identified (keyed) by its [*owner, licensee*] pair. During our experiments, we were able to use a 500,000-entry cache, which occupied approximately 40 MB of memory.

## Evaluating CPOL

To evaluate CPOL, we first performed micro-benchmarks. These experiments involved evaluation of single policies in Keynote, MySQL, and CPOL under varying conditions. We also adjusted the number of policies in the system and found that Keynote’s lack of indexing caused its performance to suffer greatly. The results from evaluating a single policy with respect to the number of policies can be seen in Figure 2. A summary of the micro-benchmark results can be seen in Table 2. Note here that CPOL can evaluate a single policy in 0.33 µsec (3M/sec) on a cache hit, while MySQL takes 459.0 µsec (approximately 2,000/sec). and Keynote takes 101,000 µsec (approximately 10.0/sec). CPOL is two-to-three orders of magnitude faster than MySQL depending on cache hit rate, and four-to-five orders of magnitude faster than Keynote.

The next experiment we conducted used CPOL and MySQL to process queries from simulated movement data in a university building. We collected information about privacy preferences, usage scenarios, and daily habits by interviewing 30 potential users. Using that information, we simulated users going to and from class, labs, offices, restrooms, and vending machines throughout the day. Queries on the system were generated by users looking up the locations of acquaintances, others nearby, or everyone in a building. The results of this experiment using different query rates can be seen in Figure 3. Based on this experimental workload, CPOL was able to handle queries from approximately 300,000 users in real time with a sample interval of 30 sec. A MySQL database could only handle 5,000 users, and KeyNote could not even handle 1,000.

| Request Type (500 Users, 5,000 Rules) | | | Processing Times (µsec) |
|---|---|---|---|
| KeyNote | | | 101,000.00 |
| MySQL Database | | | 459.00 |
| CPOL | Cache Hit | | 0.33 |
| | Cache Miss | Access | 5.50 |
| | | No Rule | 3.50 |
| | No Cache | Access | 4.50 |
| | | No Rule | 2.00 |

**Table 2** Individual Request-Processing Times for KeyNote, MySQL, and CPOL

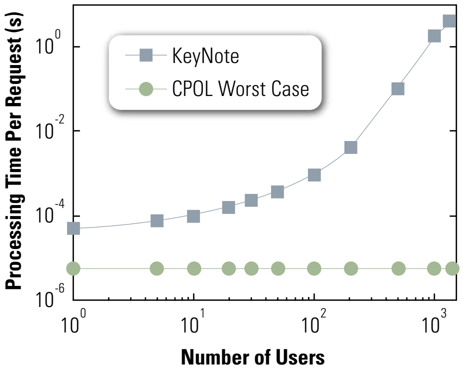


**Figure 2** KeyNote *vs.* CPOL Request-Time Comparison

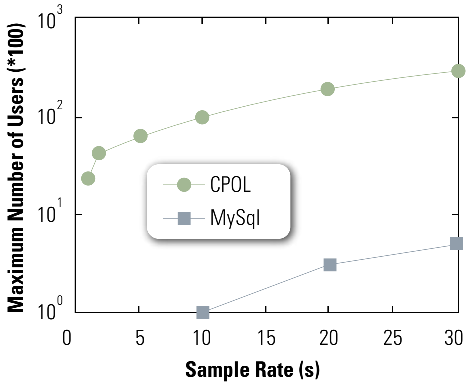


**Figure 3** Maximum Number of Supported Users for Different Location-Query Sample Rates

## Conclusion and Acknowledgements

CPOL is a promising new policy-evaluation system designed to handle heavy workloads while maintaining a high level of expressiveness. This article illustrates how CPOL can be used to efficiently enforce privacy policies in real-time for location-aware services. In addition, CPOL is a very expressive general-purpose system with potential to greatly improve performance in many other application domains that require policy enforcement.

For more information about CPOL, access to the full academic paper, or to download up-to-date source code, please visit http://cpol.sourceforge.net. The CPOL project was funded in part by a grant from Intel Corporation. The equipment for the research was funded by the NSF under Grants 0303587 and 0325332. We thank our colleague, Professor Jignesh Patel, for discussions on database research in the area of privacy and on continuous queries on spatial data. ■

## Bibliography

M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, "The KeyNote Trust-Management System Version 2," Internet RFC 2704, Sep. 1999.

M. Blaze, J. Feigenbaum, and M. Strauss, "Compliance Checking in the PolicyMaker Trust Management System," Proceedings of the Financial Cryptography Conference, Lecture Notes in Computer Science, vol. 1465, pages 254–274, Springer, 1998.

K. Borders, A. Prakash, "CPOL: High-Performance Policy Evaluation," Proceedings of the 12th ACM Conference on Computer and Communication Security (CCS 2005), Alexandria, VA, Nov. 2005.

N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," Proceedings of Policy Workshop, 2001, Bristol UK, Jan. 2001.

J. Hong and J. Landay, "An Architecture for Privacy-Sensitive Ubiquitous Computing," Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (Mobisys 2004). Boston, MA. pp. 177–189, 2004.

S. Lederer, C. Beckmann, A. Dey, and J. Mankoff, "Managing Personal Information Disclosure in Ubiquitous Computing Environments," University of California, Berkeley, Computer Science Division, Technical Report *UCB-CSD-03-1257*, Jul. 2003.

MySQL, Inc., "The mysql database manager," *http://www.mysql.org, 2004.*

L. Opyrchal, A. Prakash, A. Agrawal, "Designing a Publish-Subscribe Substrate for Privacy/Security in Pervasive Environments," In First Workshop on Pervasive Security, Privacy and Trust (PSPT), Boston, MA, Aug. 2004.

M. Spreitzer and M. Theimer, "Providing location information in a ubiquitous computing environment," Proceedings of Fourteenth ACM Symposium on Operating System Principles, Asheville, NC, Dec. 1993.

## About the Authors

**Mr. Kevin Borders** | is a PhD student at the University of Michigan where he researches computer and network security. His previous work includes research on intrusion detection, discovering data leakage through covert channels, and spyware mitigation. He received a BSE degree in computer engineering from the University of Michigan in May 2004 and is a graduate of the National Security Agency Stokes Scholar program. He may be reached by e-mail at kborders@umich.edu. His Web site is *http://www.kevinborders.com.*

**Mr. Atul Prakash** | is a Professor in the Department of Electrical Engineering and Computer Science (EECS) at the University of Michigan. He received a BTech from the Indian Institute of Technology (IIT) Delhi in 1982 and MS and PhD degrees in Computer Science from the University of California, Berkeley, in 1989. His research interests include system security, groupware systems, and software engineering. He has published over 50 papers in these areas. Smithsonian-ComputerWorld Innovation Awards Program selected his project, the Upper Atmospheric Research Collaboratory (UARC), for inclusion in the Smithsonian permanent archives. He is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the Association for Computing Machinery (ACM) and may be reached by e-mail at aprakash@eecs.umich.edu.

**Mr. Xin Zhao** | received BS and MS degrees in computer science from Nanjing University, China, in 1996 and 1998, respectively. He is currently a graduate student in the Computer Science PhD Program at the University of Michigan, Ann Arbor. His research interests include computer security and virtual machine systems. He may be reached by e-mail at zhaoxin@eecs.umich.edu.

# Johns Hopkins University Information Security Institute (ISI)

The Johns Hopkins University (JHU) Information Security Institute (ISI) was created, as Founding Director Gerald M. Masson explains, to address "the fact that a lot of Internet systems are vulnerable, and Internet Security is vitally important in today's society." An NSA IA Center of Academic Excellence since May 2003 [1] and charter member of the Institute for Information Infrastructure Protection (I3P), [2] the ISI is a multi-discipline collaboration of several JHU schools.

The ISI's Technical Director, Associate Professor and multiple author, Dr. Aviel "Avi" D. Rubin, leads an impressive roster of dedicated and adjunct faculty from across the collaborating JHU Schools, including the Schools of Engineering, Public Policy, Law, Medicine, and Public Health, all of whom share a common interest in Information Security and Assurance.

In practice, the ISI combines twelve Centers of Research on different Information Security issues, each of which performs both basic research and development of practical applications. Several key Centers are described below.

### 1. Accurate e-Voting Center [3]

With its activities expanded in 2006, thanks to a five-year National Science Foundation $7.5M, this Center focuses on security of Electronic Voting (e-Voting), with research areas that include:

- Creation of voting system software that is not susceptible to Trojans, and has built-in audit capability;
- Reduction of Trusted Computing Base (TCB) size through techniques such as use of TPMs (trusted processing modules) to add trustworthiness to voting machines.
- Pre-rendering of ballot images to deal with psychological/ sociological/ human factors issues.

### 2. RFID Security [4]

Partially DoD-funded, this Center includes an extensive laboratory with state-of-the-art equipment donated by the National Security Agency (approximately $100,000 worth) and RSA Security Inc. (approximately $60,000 worth) that is used to explore the limitations of and improvements to current RFID security approaches, with specific research in:

- Development of crypto-ciphers optimized for power consumption rather than processor size or network latency;
- Security and vulnerability analyses of RFID security solutions and applications such as toll tokens, credit cards, and future RFID-enabled passports.
- Vulnerability analyses of "state of the art" RFID security solutions

### 3. Privacy of Medical Records/Security of Patient Data

This Center works on projects such as development of cryptographic primitives to anonymize data before aggregation in order to preserve privacy when aggregated data is analyzed. The center was granted a National Science Foundation (NSF) CAREER Award in January 2006. [5]

> The Institute also offers a dual MS degree that combines the MSSI with a Masters in Health Management, which educates students into industry privacy standards and programs, such as the Health Information Portability and Accountability Act (HIPAA).

### 4. Other Centers

The Information Security Institute includes several other research Centers, focusing in the following areas:

- Network "Dark Space" Analysis,
- DDoS (Distributed Denial of Service) Prevention,
- Biometrics,
- Surveillance and Sensor Networks,
- Computer Forensics,
- Computer Viruses.

The ISI focuses on achieving three primary objectives:

1. To be a leader in both Information Security research and practical applications;
2. To provide a richer, more robust educational experience to students;
3. To prepare students for Information Security jobs in industry, academia, and government. [6]

The ISI offers a means to achieve the first two of these objectives through a unique MS program, the Masters in Security Informatics (MSSI), which converges both the Information Technology (IT) and Policy, Privacy, and Human Factors aspects of Information Security. MSSI Candidates have access to scholarships funded either by DoD or through a $2.92M NSF Scholarship for Service (SFS) [7] grant; in both cases, the scholarships provide a way for students to transition into DoD or other government careers after receiving their degrees.

The Institute also offers a dual MS degree that combines the MSSI with a Masters in Health Management, which educates students into industry privacy standards and programs, such as the Health Information Portability and Accountability Act (HIPAA). Both degree programs are consistent with Masson's core philosophy, which has shaped the Institute: information security is not just a technical problem, but includes aspect of human factors, public policy, and legislation. [8] As Masson puts it, "*You can't have technology in a vacuum, particularly when it comes to Information Security.*" ■

### References

[1] View the official NSA announcement: *http://www.nsa.gov/releases/relea00051.cfm*

[2] View the I3P profile of the ISI and other charter members: *http://www.thei3p.org/about/members.html*

[3] Visit the website for this center at *http://www.accuratevoting.org*

[4] Visit the website for this center at *http://www.RFID-analysis.org*

[5] View the Award at: *http://www.nsf.gov/award-search/showAward.do?AwardNumber=0546350*

[6] See the expanded goals at the ISI website: *http://www.jhuisi.jhu.edu/institute/index.html*

[7] For more information on the Federal Cyber Service Scholarship for Service program, visit: *http://www.sfs.opm.gov/*

[8] JHU has other similarly cross-disciplinary institutes and centers, such as the center on Computer Integration of Surgical Techniques.

# Dr. Avi Rubin, Technical Director

## JHU Information Security Institute

This article is the fifth in a series of profiles of members of the Information Assurance Technology Analysis Center (IATAC) Subject Matter Expert (SME) program. The SME profiled in this article is Dr. Aviel "Avi" D. Rubin. Dr. Rubin is a Professor of Computer Science and has served as the Technical Director of the Information Security Institute at Johns Hopkins University since 2003. During his tenure, he also co-founded the start-up consultancy Independent Security Evaluators with some of his former students, [1] which focuses on penetration testing and redesign of "non-secure systems" revealed through testing.

Since 2001, Dr. Rubin has also authored or co-authored nearly a half-dozen books on security topics, ranging in variety and scope. His most recent book, Brave New Ballot, published in September 2006, explores Dr. Rubin's studies into the vulnerabilities and privacy-related issues inherent to modern electronic voting techniques—a cornerstone of his current research activities. His research attempts to overcome a significant challenge in e-Voting: Balancing a sufficiently robust audit trail without compromising the anonymity of voters. Over the period of nearly a decade, Dr. Rubin has also either contributed to or refereed dozens of journal and conference publications related to security and e-Voting, and he has served as a Co-Editor and Associate Editor for past IEEE publications on multiple occasions. [2] Currently, Dr. Rubin's research interests focus in two major areas:

1. Security limitations, challenges, and implications to electronic voting systems, and
2. Security robustness of Radio-Frequency Identification (RFID) cryptography.

> Privacy, according to Rubin, continues to be a persistent problem in modern electronic voting. Paper log records of ballots, for example, provide advantages over electronic logging, as a physical audit trail is more tangible.

Dr. Rubin has asserted that "privacy needs to be a design requirement from the beginning." Privacy, according to Rubin, continues to be a persistent problem in modern electronic voting. Paper log records of ballots, for example, provide advantages over electronic logging, as a physical audit trail is more tangible. Unfortunately, maintaining the privacy of physical logs can be challenging. To date, there have been research efforts that attempt to address this issue, however none that have been fielded have been met with a sufficient level of success. Dr. Rubin's research in this area has revealed both the limitations to other current approaches as well as a potential solution that may minimize some of the security challenges inherent to the e-Voting process. To view one of his collaborative studies, refer to the footnote below. [3]

Dr. Rubin's other current major research effort focuses on RFID security. Rubin led a team that compromised (*i.e.*, "broke") the proprietary crypto-cipher used in the Texas Instruments Radio-frequency Identification System (TIRIS) microchip, which is built in the Exxon-Mobile "SpeedPass" RFID device. When outlining his two-prong approach to his research in this area, Dr. Rubin offered, *"You can't do the 'making' without a track record for 'breaking',"* clarifying that secure design ultimately requires a balance between "developing secure systems (Prong 1)" and "breaking insecure systems (Prong 2)."

If you have a technical question for Dr. Rubin or other IATAC SMEs, please contact iatac@dtic.mil. 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 domains of expertise, please contact iatac@dtic.mil, and the URL for the SME application will be sent to you.

## References

[1] View the website at *http://www.securityevaluators.com*

[2] View Dr. Rubin's complete list of publications and accolades at *http://avirubin.com/vita.html*

[3] "Analysis of an Electronic Voting System" Tadayoshi Kohno, Adam Stubblefield, Avriel D. Rubin, Dan S. Wallach. February 27, 2004. *http://avirubin.com/vote.pdf*

# Letter to the Editor

**Q** ***I'm vaguely familiar with the term "IPv6" but was hoping you might be able to share some more information with me.***

**A** Internet Protocol Version 6, also known as IPv6 or Next Generation Internet Protocol, was designed by The Internet Engineering Task Force (http://www.ietf.org) to replace the current version Internet Protocol, IPv4. Most of the internet uses IPv4, which today is close to twenty years old. IPv4 has held up extremely well over time, but it's beginning to show its age. Most importantly, there is a growing shortage of IPv4 addresses, which are required by all new machines that are added to the Internet.

This next-generation protocol fixes a number of problems and setbacks that appear in IPv4 such as the limited number of available IPv4 addresses. Other IPv6 capabilities have been developed in direct response to current requirements for more scalable network architectures, improved security and data integrity, integrated quality of service, autoconfiguration, mobile computing, data multicasting, and more efficient network route aggregation at the global backbone level.

IPv6 is expected to gradually replace IPv4, with the two coexisting for a number of years during a transition period. The Department of Defense (DoD) Global Information Grid (GIG) is also part of the transition from IPv4 to IPv6. In a June 2003 memo, Mr. John P. Stenbit, Assistant Secretary of Defense of the C3I successor organization, Networks and Information Integration [ASD (NII)] and DoD Chief Information Officer (CIO), stated that DoD has as its goal to complete the transition to IPv6 for all inter- and intra-networking across DoD by FY 2008.

IPv6 is a very hot topic, and there is an abundance of information about it. In fact, in our IAnewsletter, Vol. 7, No. 3, we featured an article entitled, IPv6, the Next Generation Internet Protocol, which discussed in detail the differences between IPv4 and IPv6, the transition and deployment process, and DoD's time line for transitioning from one to the other. If you would like to obtain more information on IPv6, please do not hesitate to contact us. ■

# Creating a Network Warfare Operations Career Force

by Rhonda Richardson

The Secretary of Defense approved the development of an Information Operations (IO) Roadmap as part of his effort to transform the US Department of Defense (DoD) to meet future challenges. The Roadmap, released in 2003, provided a vision for IO and directed each Service to develop IO as one of its core competencies. In response, the Air Force developed an IO Concept of Operations (CONOPS) and revised Air Force Doctrine Document (AFDD) 2-5, Information Operations.

The Air Force IO CONOPS, published 06 February 2004, and AFDD 2-5, published 11 January 2005, organizes IO into three operational capabilities: Influence Operations, Electronic Warfare Operations, and Network Warfare Operations (NW Ops). NW Ops encompasses the following mission areas:

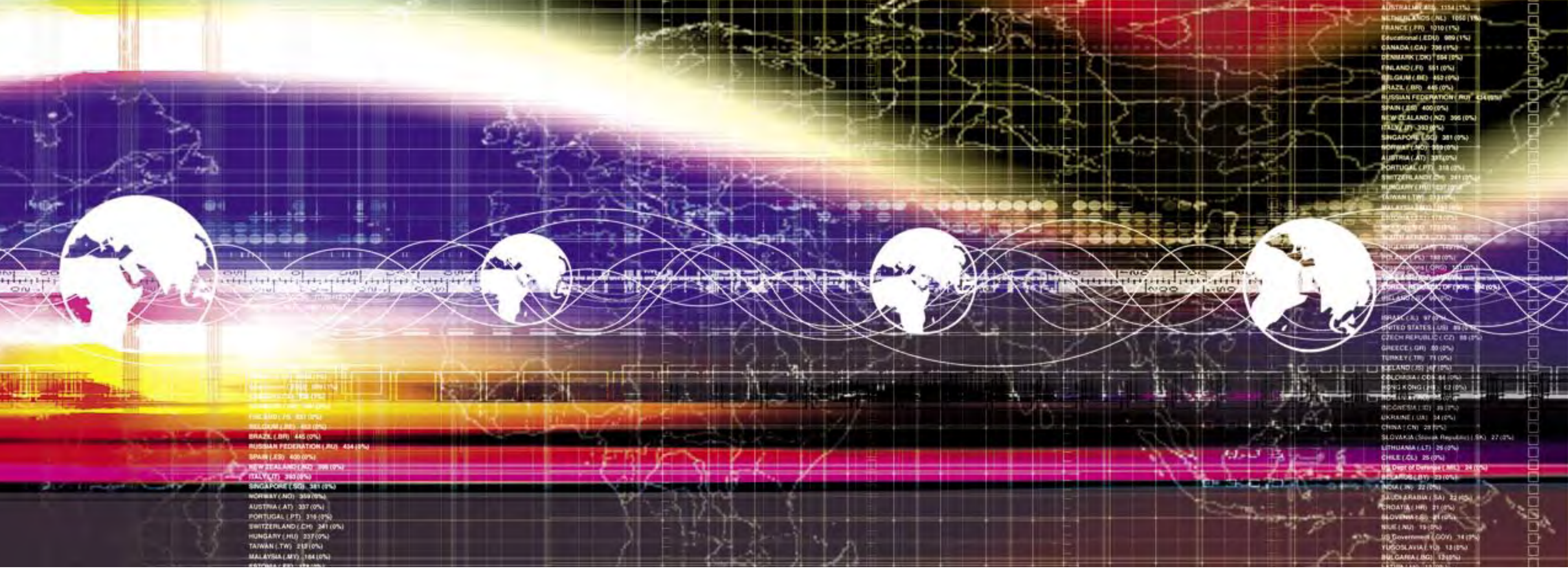
- **Network Attack (NetA)**—the employment of network-based capabilities to disrupt, deny, delay, degrade, or destroy information resident in or transiting through networks
- **Network Defense (NetD)**—the employment of network-based capabilities to protect and defend networks, as well as the information resident in or transiting through networks, against adversary efforts to destroy, disrupt, corrupt, deny, delay, degrade, or usurp it
- **Network Warfare Support (NS)**—actions tasked by or under direct control of an operational commander to search for, intercept, identify, and locate or localize sources of access and vulnerability for the purpose of immediate threat recognition, targeting, planning, and conduct of future operations

Taken together, NW Ops can be used to achieve a desired effect across the interconnected analog and digital portion of the information battle space. To achieve the Air Force's vision for NW Ops, the groundwork is being laid for developing a cadre of trained professionals who can integrate NetA, NetD, and NS into air, space, ground, and maritime operations. A major component of this effort is the partnering of the Secretary of the Air Force, Force Development and Transformation Division, the Air Force Deputy for Information Warfare, and Air Combat Command to establish NW Ops Basic and Intermediate Schools. The Schools, projected to be available in FY07, will initially focus on the officer corps. The specific feeder Air Force Specialty Codes (AFSCs) for the NW Ops Schools will include Space and Missile (13Sx), Intelligence (14Nx), Communications and Information (33Sx), Developmental Engineer (62Ex), and Special Investigator (71Sx) career fields. As the Air Force further expands its focus on the enlisted and civilian force, the Schools will adapt to meet their specific training needs.

> The Air Force strongly depends on its terrestrial, airborne, and space networks to conduct offensive and defensive operations. The NW Ops Basic and Intermediate Schools will supply the vital "missing piece," a career force capable of defending the Global Information Grid and/or providing a desired, non-kinetic effect.

### NW Ops Basic School

The Basic School will provide a general understanding of NW Ops capabilities and will target officers at the one-to-four year



career point (second and first lieutenants). Selection to attend the School will require self-nomination by the individual and recommendation by the unit commander. Although specific technical prerequisites are currently under development, the individual should possess a strong background in networking and computer skills. On completing the basic course, officers will take additional qualification courses such as NetD for the Air Force Network Operations Security Center or Network Aggressor training for the 92nd Information Warfare Aggressor Squadron.

The curriculum will be broken into four phases: IO Fundamentals, NW Ops Fundamentals, NW Ops Certifications, and a scenario-based graduation exercise. The curriculum will focus on developing highly skilled, capable specialists trained to execute NetA, NetD, and NS across air, space, and terrestrial networks. It is not intended to duplicate either courses awarded by the AFSC or local training.

## NW Ops Intermediate School

The Intermediate School will target officers at the four-to-eight year career point (captains) with at least two years' experience in the field. Individuals must complete the NW Ops Basic School to be eligible for the Intermediate School, which will focus on developing NW Ops planners with the advanced skills to articulate NW Ops requirements and integrate NetA, NetD, and NS into kinetic and non-kinetic theater operations.

Like the Basic School, the Intermediate School curriculum will consist of four phases: Fundamental Skills and Knowledge of NW Ops; Planning and Integration; intermediate topics such as integration and interoperability, Joint, Coalition, and multi-service operations and target development; and a large-scale, scenario-based graduation exercise.

## Special Experience Identifiers (SEIs)

The SEIs will allow Air Force career-force managers to easily identify IO positions and personnel to ensure the right people get the right jobs at the right time. Air Force Director of Personnel and the career-force managers of the Air Staff/Major Command (MAJCOM) are scheduled to complete the SEI tagging of officer billets in the fall of 2006, at which time the Air Force will begin assigning the trained IO officers to these duty positions. Building the criteria for tagging the enlisted and civilian IO billets and trained personnel is scheduled to begin in late 2006.

The NW Ops IO SEIs include NetA Capability Specialist (9I), NetD Capability Specialist (9J), NS Capability Specialist (9K), and Network Warfare Operations Planner (9L). NW Ops professionals can be sourced from any of five AFSCs (13X, 14N, 33S, 62E, and 71S) meeting the mandatory training requirements as specified in Air Force Manual (AFMAN) 36-2105. The Air Force IO career-force managers will spend the next several years gathering force-management data, at which time they will lead an effort to examine the need for a separate IO career field.

## Conclusion

The Air Force strongly depends on its terrestrial, airborne, and space networks to conduct offensive and defensive operations. The NW Ops Basic and Intermediate Schools will supply the vital "missing piece," a career force capable of defending the Global Information Grid and/or providing a desired, non-kinetic effect. The development and maturation of a cadre of IO professionals will improve the way in which the Air Force employs NW Ops to support the kill chain and provide a robust combat capability to the Air Force and Joint warfighter. ■

## About the Author

**Ms. Rhonda Richardson** | is an Information Technology Specialist (Policy and Planning) serving as the Information Operations Cell Chief, Networks Division, Directorate of Communications, Headquarters Air Combat Command, Langley Air Force Base, VA. Ms. Richardson develops policies and programs to seamlessly integrate Combat Air Forces IO capabilities.

# Cyber Security Dimensions of Critical Infrastructure Protection (CIP) Conference

by Avery-Lynn Dickey

In March 2004, the Information Assurance Technology Assessment Center (IATAC) assisted the Office of the Assistant Secretary of Defense Networks and Information Integration's [OASD(NII)] International IA Program with planning and executing the conference on Political-Military Dimensions of Cyber Security. The conference was co-sponsored by OASD(NII), United States European Command (USEUCOM), and the George C. Marshall Center for Security Studies. The focus of the conference was the military aspect of executing cyber defense in a political and international environment. Because of the success of this conference, a follow-on conference was planned, organized, and held in October 2005.

The Cyber Security Dimensions of Critical Infrastructure Protection (CIP) Conference of 25–28 October 2005 was again co-sponsored by OASD/NII, USEUCOM, and the George C. Marshall Center in Munich, Germany. Military, government, and industry representatives from over 31 countries in Europe and Eurasia attended the conference and participated in plenary presentations, panel discussions, and a three-day cyber security scenario breakout group event.

"We must all work together!" quickly became the tone of the conference, as retired German Air Force Major General Dr. Horst Schmalfeld, Deputy Director of the Marshall Center, opened the conference by telling participants, "We must all work together. By bringing together private industry and competent government agencies involved with cyber security, we can look at ways of identifying areas of mutual concern that require close partnership—that is why this conference is so important."

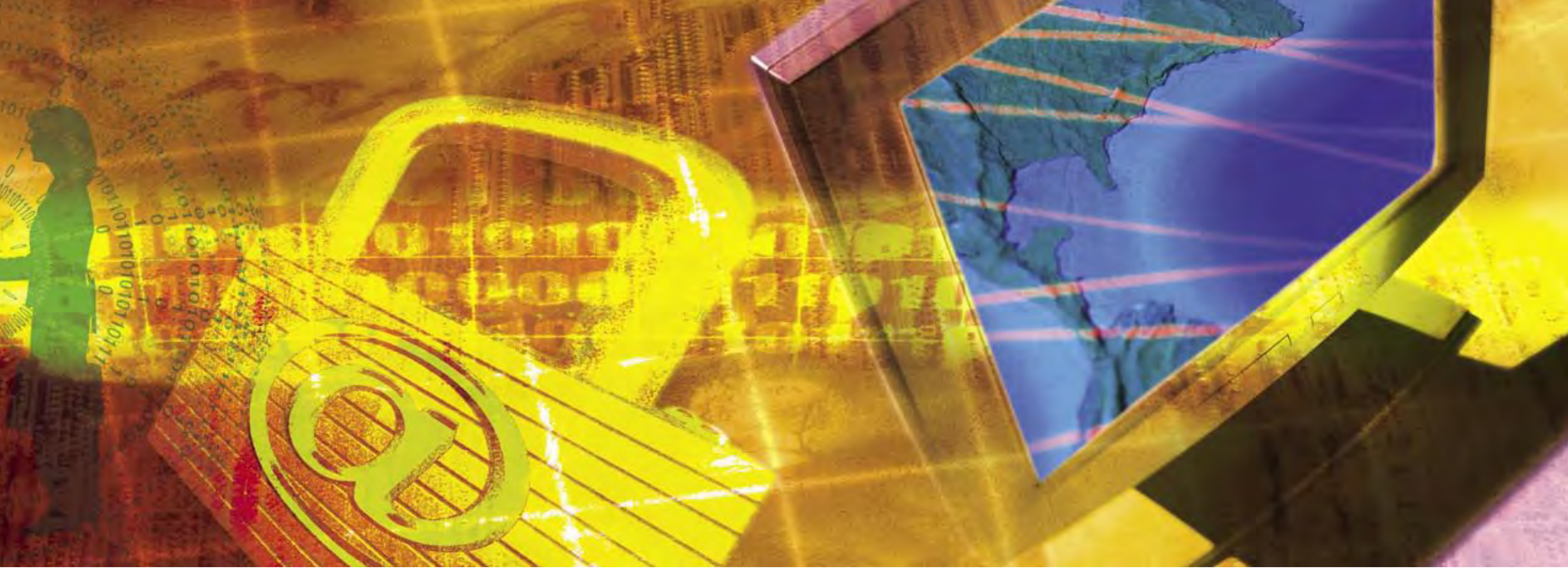
MG (USA, Ret) Dave Bryan, Vice President, Northrop-Grumman, echoed that sentiment in his keynote address, saying, "Now more than ever, the issue of cyber security is extremely critical. As the threat of terrorism continues, attacks against each of your nation's critical infrastructure and financial systems also increase. We are all in this together. There is so much at risk, and if we don't act together, cyber attacks can put us all at risk. We must coordinate our activities in order to maximize our efforts." Focusing on the idea of TEAM—"Together Everyone Achieves More"—General Bryan stressed that, as we become increasingly connected *via* networked computers and the growth of the Information Age, we must work together to protect the networks and prevent cyber attacks. If ties are severed, catastrophe will ensue.

In addition to the keynote address, the conference comprised a series of panels, presentations, and scenario working groups, including the following:

- Protecting Critical Infrastructures
- Cooperation with Alliances
- Military and Government Perspectives
- Creation of a Computer Emergency Response Team (CERT)
- Training and Education

The panels and presentations featured speakers from military, government, and private industry who are responsible for working with the challenges of cyber security from various different perspectives. Speakers hailed from Armenia, France, Hungary, Italy, Lithuania, Poland, Romania, Sweden, Switzerland, Turkey, the United Kingdom, and the United States.

The three-day cyber security scenario was developed by IATAC Subject Matter Experts (SMEs) to reveal international cyber security and critical infrastructure issues and to foster communication among nations. IATAC facilitated six scenario working groups that enabled countries to discuss multiple approaches to solving a hypothetical cyber security incident in a non-attributional environment. The working groups focused on different perspectives during an incident; specifically, military and government, intelligence, impact on critical infrastructure protection, CERT response, and international coordination. Each working group comprised participants from various countries who were able to share their respective positions, reactions, and



limitations. After the scenario played out, each working group delivered a concluding brief to all participants that discussed actions and conclusions. Some major issues were uncovered in dealing with cyber security on an international level, including the following:

- a lack of resources and qualified personnel
- a need for more robust and mature international law
- a need for improved international communication

Overall, the conference increased international awareness of the importance of securing networks to prevent a potential catastrophic event that will impact national and international critical infrastructures. The conference gathered valuable Scientific and Technical Information (STI) from presenters with an international perspective on IA. The next steps are to develop solutions to the issues of international cyber security that were uncovered throughout the conference. Through this conference, the Marshall Center, USEUCOM, and OASD(NII) laid a solid foundation of international coordination regarding cyber security issues.

To learn more information about IATAC Conference and Event Planning, visit its Web site at *http://iac.dtic.mil/iatac/con_event_planning.html.* ■

The picture above is of the sponsor, Mark Hall of OSD-NII, delivering the opening remarks.

### About the Author

**Ms. Avery-Lynn Dickey** | is an IATAC Conference and Event Planner. She received a degree from the University of Maryland in International Business. She may be contacted at iatac@dtic.mil.

# Privileged Escalation Through Trusted E-mails

by Jared Trent

Consider the following scenario: You have been receiving a number of e-mails recently from an online retailer informing you that your password has expired and asking you to click on a link to resolve the issue. The e-mail has a corporate logo and looks and sounds legitimate, but, after some preliminary research on the link, it leads not to a legitimate corporate entity but to an address in South Korea.

Almost anyone with an e-mail account is familiar with or has experienced this type of activity, which is better known as SPAM. While most SPAM activity is generally harmless, such as offering cheap prescription drugs or free trips to the Caribbean, a more nefarious form exists in the form of "phishing." Phishing is the practice of trying to obtain credit-card, password, or other sensitive information and is a far more dangerous threat than offering discount products online. Within the US Department of Defense (DoD), Phishing can pose a serious threat to security. Fortunately, DoD users are generally aware of this threat and immediately delete such e-mails.

There is the potential, however, for a greater threat—a properly executed Phishing scam that exploits the implicit trust of e-mails sent from senior officials to subordinates. This scam gives a hacker or a foreign nation unrestricted or privileged access to DoD systems.

### Tools of the Trade

A spoofed e-mail has two key features:

(1) It disguises the real originator by using a name other than that of the sender, as in the scenario presented above.

(2) It incorporates social engineering by targeting on a human propensity to read messages that appear to be from a trusted source.

There are several ways to create a spoofed e-mail. The simplest method is to modify the Simple Mail Transfer Protocol (SMTP) settings, which sends e-mail using an e-mail client, such as Microsoft® Outlook. When creating an e-mail account, the hostile intruder can give the account any name of his choosing. However, this method is easily detectable by opening the e-mail message and clicking *View | Options* to determine the origin of the e-mail; the only drawback to this is the amount of time and effort required to check every single message.

Although DoD has recognized its technical vulnerabilities, solutions to mitigate socially engineered threats remain a challenge for the individuals responsible for protecting the Global Information Grid (GIG).

There exist more sinister and complicated methods, one of which is to use an open-mail relay in which neither sender nor recipient is a local user. Since a majority of mail relays are open, this method can be easily exploited, making it impossible to determine the actual source of an e-mail.

### The Target

Once hostile intruders appropriately modified their SMTP settings or have found an open mail relay to use, they then move to the next step: identifying a victim and determining the type of attack. Because operational security management is often poor, this stage generally requires far less effort.

Hostile intruders are motivated by only one purpose: to gain unrestricted or privileged access to a proprietary network such as that of DoD. The following scenario best illustrates how an intruder might gain access to a certain organization. Our intruder first begins

with open-source Internet research on the target, locating its unclassified home page. Because of poor security administration, this Web site is a treasure trove of information. With additional research, the intruder has located the biographical ("Bios") section of the target's Web site. From this, the intruder learns that the Commanding Officer is Brig Gen Stephanie Owens. The intruder also learns the names of the Logistics, Intelligence, and the Operations officers.

Once the intruder has this information, he seeks to learn the naming convention for e-mail addresses within the command. Again, from the "Bios" section, he determines that e-mails are written using the following format: [firstname.lastname@unit.mil]. The intruder now has targets: the Logistics, Intelligence, and Operations officers. He has obtained an e-mail address he will spoof: stephanie.owens@unit.mil. Without committing a single illegal activity, or drawing attention to himself, the hostile intruder has gathered sufficient information to conduct his Phishing activity. All that is missing is an unwitting participant.

### The Setup

The average spoofed e-mail would, at this point, attempt to craft some well-designed attempt to learn a password or gather credit card information, but this is no ordinary attacker—this is a hostile intruder operating with a definite purpose. He has carefully crafted an e-mail attachment, with a title vague enough to cause, at most, puzzlement on behalf of its recipients. Because of the level of trust that typically exists between the Brig Gen Owens and her subordinates, it is unlikely that anyone will question the veracity of an e-mail attachment she sends them.

This attachment, however, is more than it seems. Embedded in its code is a self-extracting executable file, which, when opened, inserts into the system directory a backdoor to the target's system. This executable file may be

- key-logger software, which captures every keystroke on a computer and then transmits it to a third party;
- a file that gathers all documents, spreadsheets, and presentations, zips them, and sends them to a third party; or
- a root-kit, which permits the hostile intruder to remotely log in to the target's system and browse through it as if he had been given unrestricted access.

The intruder can be even more deceptive, using well-known ports, such as port 53 for a Domain Name Server (DNS), which converts convenient Web-site names, such as www.unit.mil, into a numbered Internet Protocol (IP) counterpart (*e.g.*, 10.10.10.5). Or he could use port 80, Hypertext Transfer Protocol (HTTP), which provides Internet access. Activity over these ports would go largely unnoticed by firewalls and Intrusion Detection Systems (IDSs), as it is standard network traffic.

The possibilities of this type of attack are endless: the intruder can gain unrestricted access to the Logistics, Intelligence, and Operations systems for the entire fictitious unit. While there should be no classified information on these systems, it is likely there will be information on the troop movements, future operations, and current intelligence requirements of the unit. With this information, the intruder no longer needs a network of spies within the unit—he has already gained unrestricted access because of poor network security.

### Vulnerabilities

Unless a hostile intruder uses a security vulnerability that has not yet been discovered or fixed (Zero-Day Exploit), security patches or anti-virus software should protect the victim's system. This, of course, assumes that the systems are maintained by diligent system administrators. Unfortunately, this is not always the case.

In many instances, particularly at smaller units, DoD systems are maintained by individuals who have received little or no training and are given nothing more than an instruction manual and an order to "make it work." Because insufficient training results in poor security, this means that DoD networks are wide open to security threats. Even with highly skilled network administrators, security becomes a concern when they are faced with the overwhelming task of maintaining large, complicated networks.

### Mitigation

In the face of these potential security threats, what procedures does DoD plan to implement to protect against such attacks? Is there a method whereby members of DoD can guarantee the e-mails they receive are from a trusted source? The answer is surprisingly simple: the Agency already has the proper safeguards to protect against these threats and the policy to implement it; however, there has been poor or limited implementation.

**Public Key Infrastructure (PKI)**

Public Key Infrastructure (PKI) creates a trusted environment among users by integrating a private key and a public key, which are managed by either a third party or maintained by the local organization. The public key is shared with the receiver; only to the sender knows the private key. An example of this would be if Brig Gen Owens were to send an e-mail to Capt Hodge, the Unit Intelligence Officer, she would encrypt her message with her private key. Capt Hodge would then decrypt the message using both her public key and his own private key. With a PKI infrastructure in place, these e-mails are digitally signed, verifying to Capt Hodge that the sender is the General, and providing a commensurate level of trust between the General and her subordinates. DoD understood the need to integrate secure procedures such as PKI when it issued the following statement in December of 2000: "We recognize that the only practical way to extend IA (Information Assurance) features to over 3.5M DoD employees (active military, reservists, and civilians) and to the hundreds of software applications and the thousands of network devices across the Department is to deploy a modern, commercially based infrastructure." [1]

One of my first assignments in the Marine Corps in 2001 was to get my unit up to speed with the Corps-wide implementation of PKI. I arranged training for users and set up the proper software for our systems. However, the PKI program was abandoned several months later. Now, nearly five years since the *Roadmap* was issued, few members of DoD are using PKI.

**Common Access Card (CAC)**

Another potential security method is integrating Smart Card technology. A Smart Card can be described as a cross between a credit card and a driver's license. It has a photograph of the user, a magnetic strip, and a computer chip, all of which serve to identify the user and store his or her relevant data.

DoD's implementation of the Smart Card technology has been accomplished with the Common Access Card (CAC). In accordance with the Clinger—Cohen Act of 1996, DoD was directed to "improv [e] the acquisition, use, and disposal of information technology." [2] In keeping with this Act, DoD issued the following directive: "The initial implementation of smart card technology shall be effected as a Department-wide common access card (CAC). The CAC shall be the standard ID card for active duty military personnel (to include the Selected Reserve), DoD civilian employees, and eligible contractor personnel. It will also be the principal card used to enable physical access to buildings and controlled spaces and will be used to gain access to the Department's computer network systems." [3]

Proper implementation of the CAC system would provide DoD with multifactor authentication, which means that a user must possess more than one attribute to access a system. This can take the form of three things: something you have, something you are, or something you know. The CAC is something you have, and a password would be something you know. CACs are able to implement biometrics (*e.g.*, a fingerprint scanner), thereby proving users are who they say they are. CAC can be integrated with a fingerprint scanner, defeating the ability of an intruder to steal the CAC and a user password. With a CAC inserted to access a DoD network, PKI can then be more easily implemented, as the user is already verified. However, as with PKI, CACs have been issued to members of DoD, but few organizations have implemented their full capability.

### Conclusion

The weakest link in any communication network is the human factor. Although DoD has recognized its technical vulnerabilities, solutions to mitigate socially engineered threats remain a challenge for the individuals responsible for protecting the Global Information Grid (GIG). Increased awareness of Information Assurance (IA) policy, planning, and compliance, combined with multi-tiered security measures, will serve to minimize socially engineered exploits. ■

### References

[1] *Public Key Infrastructure Roadmap for the Department of Defense*, Version 5.0, 18 December 2000, *http://www.dod.mil/nii/org/sio/ia/pki/PKI_Roadmap.pdf*

[2] *National Defense Authorization Act for Fiscal Year 1996*, also known as the Clinger-Cohen Act, Division E of Public Law 104–106.

[3] Common Access Card Memorandum, Deputy Secretary of Defense, 10 November 1999, *http://www.defenselink.mil/nii/org/sio/ia/pki/smartcard_11101999.pdf*

### About the Author

**Mr. Jared Trent** | supports the US Strategic Command (USSTRATCOM) Joint Task Force-Global Network Operations (JTF-GNO) Law Enforcement/Counter Intelligence Center (LE/CI). His main area of expertise encompasses the JTF-GNO Information Assurance (IA) threat analysis and operations. Mr. Trent served in the US Marine Corps as an Intelligence Analyst for Marine Aircraft Group 13 and in Operation Iraqi Freedom. He received a BA from Charter Oak State College, is an MBA candidate at the University of Wisconsin, and is a Certified Information Systems Security Professional (CISSP). Mr. Trent may be reached directly at the JTF-GNO LE/CIC at 703/607-6544 or by e-mail at jared.trent@jtfgno.mil.

# IATAC Conference and Event Planning

IATAC

Experienced Assistance for Your Classified or Unclassified Event

**Are you a government client in need of planning and hosting assistance for an upcoming conference? Look no further… IATAC's conference and event planners provide the assistance you need.**

Since 1998, we have offered a full range of services to support classified and unclassified conferences, meetings, and other gatherings for groups ranging from 20 to 300+ participants. From site selection and registration to catering and security requirements coordination, we can plan and execute an event that complies with government conference regulations and provides a high level of customer satisfaction. All members of our staff hold active security clearances ranging from Secret to Top Secret/SCI.

Services are available to all government clients regardless of whether or not they are currently affiliated with the IATAC contract. Support can be arranged through Technical Area Tasks (TATs) subscription accounts with payments *via* Military Interdepartmental Purchase Requests (MIPRs), if applicable.

**Our experienced planners offer service and support for all phases of your event.**

### Before the event

- Site selection
- Budget oversight
- Contract negotiation
- Secure online registration and payment
- Graphics support
- Audio/visual coordination
- Agenda development
- Sponsorship/exhibitor solicitation
- Marketing and promotion
- Security requirements coordination (classified events)

### During the event

- Check-in and registration
- Note-taking (session minutes)
- Speaker assistance
- Problem resolution
- Catering coordination

### After the event

- After-action report
- Conference surveys and evaluations
- Distribution of conference proceedings
- Reconciliation of invoices

### Want more information?

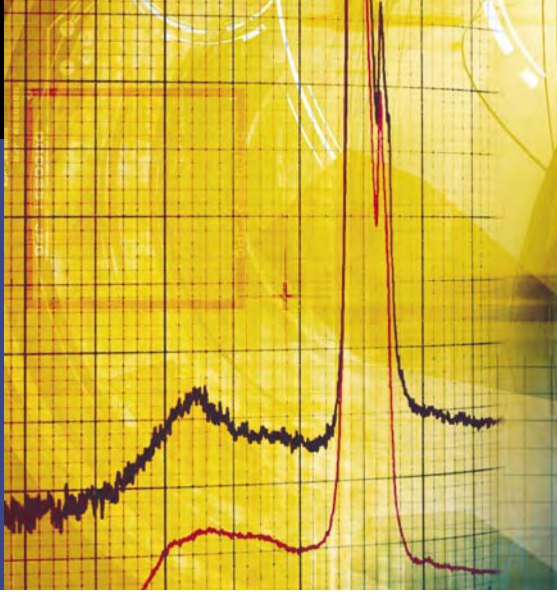
To find out more about IATAC's conference and event planners and what they can do for you, please contact:

**April Perera**
Director, Conference and Event Planning
703/984-0769

**Avery-Lynn Dickey**
Conference and Event Planner
703/984-0766
*iatac@dtic.mil*

### Examples of recent events

- Intel Support to CND Conference, August 2003
- Second Intel Support to CND Conference, February 2004
- Fourth Intel Support to CND Conference, March 2005
- Fifth Annual Intel Support to CND Conference, March 2006
- The Political/Military Dimensions of Cyber Security, March 2004
- The Cyber Security Dimensions of Critical Infrastructure Protection, October 2005
- The 2nd Annual USSOUTHCOM Force Protection Detachment Conference, October 2005
- CI & HUMINT 2006 Conference March 2006
- DoD Defense Continuity Conference, September 2004
- 2006 Defense Continuity Conference, April 2006



# Defending Warfighter Networks

by Dr. Anup Ghosh

*The following article is a transcript of a speech delivered by Dr Ghosh at the DARPATech conference, at the Anaheim Marriot Hotel, Anaheim, California on 9–11 August, 2005—Editor.*

Imagine, for a moment, a network that behaves like a living organism: It is fully cognizant of its environment. It can recognize attacks and failures and adapt to its new environment. It can monitor the behavior of its users and the state of its software and detect when it is being misused. It knows how to repair its own faults. And, above all else, it can defend itself, allowing it to provide sustained service even during continuous attacks.

This is the level of sophistication in network design we must achieve to realize Department of Defense (DoD) goals for network-centric warfare. We're not there yet, not by a long shot.

The network is the most important weapons platform for the military of the future. My Advanced Technology Office (ATO) colleagues have spent much of the last hour describing the potential of network-centric warfare. They have discussed the projects they are managing to bring this potential online within our military. The technologies they're working to deploy have the ability to powerfully expand our warfighting capabilities. They will allow our forces to operate with greater accuracy and lethality, while putting fewer of our soldiers in harm's way.

Overcoming the technological challenges they've discussed will be an awesome job. But that is just the start. Because of all the potential of network-centric warfare, all the future capabilities DoD is counting on to fight and win our wars hinge on a crucial, self-evident fact: The networks must work.

That is my area of research within ATO, and what I'd like to talk about today: How do we design networks that are self-defending and self-sustaining through attack?

As my colleagues made clear: DoD is counting on the Global Information Grid (GIG) to fight its wars. Our military leaders envision a GIG that provides reliable access to a rich stream of data and information for every DoD user, from war planners to individual soldiers in the most forward-deployed units.

To fulfill this critical mission, the GIG must offer a reliable, secure, and robust computing network. Yet there are many technological Everests to climb and conquer before the scientific community can make good on the GIG's promise. As it stands today, because of the fundamental problems of our current computing systems, the GIG is likely to fail in delivering on its promise if we don't develop robust, self-defending, and self-sustaining networks.

Today's networked Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) are predicated on the ability to share information in a secure, timely manner. Our military personnel expect to have a continuous picture of the battlefield, a picture they must be able to relay through networks and data links both up and down the entire chain of command and laterally within units in an ever-wider circle of information sharing. This network must be secure against all natures of attack, even though the network is based on infrastructure that is sometimes mobile, ad hoc, and always under attack.

Networks enable our C4ISR capabilities, but, in an era in which networks are constantly under attack, we must develop C4ISR capabilities for the network itself. That means a network that can conduct its own surveillance; a network that can process intelligence about the threats it faces; a network that can command and control itself. This requires not merely evolutionary, but revolutionary, changes to the way we build networking and computing technology.

Our current approach to network defense is medieval. Network defenses are designed with a fortress mentality. We build the toughest possible shell to repel as many attacks as possible. We use firewalls, anti-viral systems, Intrusion Detection Systems (IDSs), and patch-compliance tools to strengthen the fortress walls in a valiant, but ultimately futile, effort to keep intruders outside the

network's walls. We treat network defense like a war of attrition, hoping we can somehow outlast the enemy's siege.

The flaw is, once the fortress walls are breached, attacks can systematically undermine the network one node at a time—and often the entire enterprise—within a matter of seconds. One Trojan horse can defeat a carefully constructed fortress. And while attackers need only exploit one vulnerability, we have to close every hole, including holes we don't know about. It is ironic how brittle our systems are, often crashing or becoming compromised when presented with unexpected input. Meanwhile, viruses and spyware are infamously robust, often able to withstand a barrage of detection and cleanup tools and keep on ticking. Talk about asymmetric warfare…

> The initial groundwork for self-defending networks has been laid by two ongoing projects of the Defense Advanced Research Projects Agency (DARPA): Dynamic Quarantine of Computer-Based Worms and Defense Against Cyber Attacks on the Mobile Ad Hoc Network (MANET) Systems programs.

Obviously, we need to radically upgrade our cyber-defense mentality to 21st century standards. We must wage the equivalent of a war of network maneuver rather than hope to survive a war of attrition. To secure our goals for network-centric warfare, our networks must be designed to be self-defending and self-sustaining. What are the attributes of a self-defending network?

First, it has to be cognizant of its own behavior. Our networks must be their own doctors, with the ability to develop a baseline of health and to recognize when they are sick.

Second, self-defending networks need their own command and control function. This would allow the network to recognize attacks and failures, distinguish between malicious and benign users, determine when software is misbehaving, and provide traceback and attribution of attacks. Based on this intelligence, the network would need to adaptively reconfigure in the face of attacks and failures. In other words, an autonomic command and control system for networks.

Third, self-defending networks must be self-correcting. After sensing and evading attacks, our networks must be able to adapt, developing new immunities so that they are no longer vulnerable to the same attacks.

What are the technology challenges that need to be overcome to achieve this vision? What are some of the approaches that might prove effective in meeting these challenges? Without intending to limit creative input, here are a few ideas of where we'd like to go.

- **Software assurance controllers**—We need on- or off-board devices that execute control algorithms for monitoring and controlling the dependability and security of essential software systems. These devices not only monitor applications for run-time failures or security violations but also apply appropriate correctives in case of failure or compromise. Any corrective actions autonomously applied must have high precision and be cost optimized to preserve as much normal system operation as possible while isolating and correcting the problem. And we

must accomplish all of this without the aid of a human in the loop.

- **Dynamic measures of system health**—We must define what constitutes system health and then use these as inputs to our assurance-control models. Examples include uptime, network latencies, available memory, hung processes, system restarts, and intrusion alerts.
- **Real-time, large-scale network health status**—To make this possible, we require scalable real-time measures of the operational impact of degraded system services. For example, systems' functions may be lost or degraded because of attacks, malfunctions, or autonomous corrective actions. When this happens, we must be able to alert human operators and decision makers to the operational impact of these lost services. We also need to give real-time mission health assessments to global network operations centers.
- **Out-of-band network defenses**—Host defenses must run on separate hardware from that of the applications they are defending. Ideally, we should use a different instruction set or operating system and a separate command and control channel for our defenses. We must reverse the long trend of building our defenses on a house with a broken foundation.
- **Trust-based Credentials**—We must develop a credentialing system that can discriminate between trustworthy and untrustworthy patterns of system-resource use. Individual users' credentials, system permissions, and accesses must be continually reassessed in light of each user's current behavior. And user credentials must be automatically degraded, revoked, or restored with fine-grained controls when untrustworthy behavior is detected.

The initial groundwork for self-defending networks has been laid by two ongoing projects of the Defense Advanced Research Projects Agency (DARPA): Dynamic Quarantine of Computer-Based Worms and Defense Against Cyber Attacks on the Mobile Ad Hoc Network (MANET) Systems programs. While these programs have begun to show the promise of autonomic defense technology, the challenges described today remain.

DoD vision for the GIG and future military operations requires all DoD and Intelligence Community users to have timely, assured access to information. To realize this vision, to be able to achieve victory in the future, we need you to solve these technical challenges in network-centric warfare.

DoD vision for the GIG and future military operations requires all DoD and Intelligence Community users to have timely, assured access to information. To realize this vision, to be able to achieve victory in the future, we need you to solve these technical challenges in network-centric warfare.

We look forward to solving these problems together in the future.

## About the Author

**Dr. Anup Ghosh** | is Research Professor and Chief Scientist in the Center for Secure Information Systems (CSIS) at George Mason University. Dr. Ghosh was previously Senior Scientist and Program Manager in the Advanced Technology Office of the Defense Advanced Research Projects Agency (DARPA) where he managed an extensive portfolio of information assurance and information operations programs. Dr. Ghosh is also author of three books on computer network defense. Dr. Ghosh serves on the editorial board of IEEE Security and Privacy Magazine and has been guest editor for IEEE Software and IEEE Journal on Selected Areas in Communications. Dr. Ghosh is a Senior Member of the IEEE. For his contributions to DoD's information assurance, Dr. Ghosh was awarded the Frank B. Rowlett Trophy for Individual Contributions by the National Security Agency in November 2005, a Federal government wide award.

# FREE Products

# Order Form

Instructions: All IATAC LIMITED DISTRIBUTION reports are distributed through DTIC. If you are not registered DTIC user, you must do so prior to ordering any IATAC products (unless you are DoD or Government personnel). To register On-line: *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 ______________

______________________________ E-mail ______________

______________________________ 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)
□ Data Mining (soft copy only)
□ Wireless Wide Area Network (WWAN) Security
□ Computer Forensics* (soft copy only. DTIC user code MUST be supplied before these reports will be shipped)
□ Configuration Management
□ IA Metrics (soft copy only)
□ Defense in Depth (soft copy only)
□ Network Centric Warfare (soft copy only)
□ Exploring Biotechnology (soft copy only)

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

## 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 | | | |

**Fax completed form to IATAC at 703/984-0773**

# Calendar

## May

**DISA Customer Partnership Conference 2006**
01–04 May 20006
Las Vegas Hilton
Las Vegas, NV
*http://www.disa.mil/conference/index.html*

**REDTEAM2006 Conference**
02–04 May 06
Sandia National Laboratories
Kirtland AFB, Albuquerque, NM
*www.sandia.gov/redteam2006/*

**Cross Domain Solutions Workshop**
02–04 May 2006
Sheraton Colorado Springs
Colorado Springs, CO
*http://www.nsa.gov/ia/events/conferences/index.cfm?ConferenceID=33*

**IA Summit—United States Air Force Space Command**
02–05 May 2006
Colorado Springs, CO
*www.fbcinc.com/iasummit*

**Transformation TechNet TEAM Transformation …A Must for International Security**
09–10 May 2006
Hampton Roads Convention Center
Hampton, VA
*http://www.afcea.org/events/transformation/transformation06/info.asp*

**Armor Warfighting Symposium 2006**
15–17 May 2006
Fort Knox, Kentucky.
*http://www.fbcinc.com/event.asp?eventid=Q6UJ9A00AE8L*

**17th Annual National OPSEC Conference and Exhibition**
15–19 May 2006
Hyatt Regency Dallas At Reunion
Dallas, TX
*http://www.nsa.gov/ia/events/conferences/index.cfm?ConferenceID=35*

**Heartland TechNet 2006—Net Centricity: Empowering The Warfighter**
23–25 May 2006
Offutt Air Force Base
Omaha, NE
*http://afcea.com/calendar/eventdet.jsp?event_id=12070*

## June

**ECIW 2006: The 5th European Conference on Information Warfare and Security**
01–02 Jun 2006
Helsinki, Finland
*http://academic-conferences.org/eciw/eciw2006/eciw06-home.htm*

**TechNet International 2006—Information Sharing—Adaptability Across the Spectrum of Operations**
19–20 June 2006
Washington, DC Convention Center
Washington, DC
*http://www.afcea.org/events/technetinternational/*

**Making it Real in 2008 – Information Assurance (IA) Implementation Plan for the Global Information Grid (GIG) Architecture**
15 June 2006
*forum2@iatf.net*

**18th Annual FIRST Conference**
25–30 June 2006
Baltimore, MD
*http://www.first.org/conference/2006/papers.html*

IATAC

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

To change, add, or delete your mailing or e-mail 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