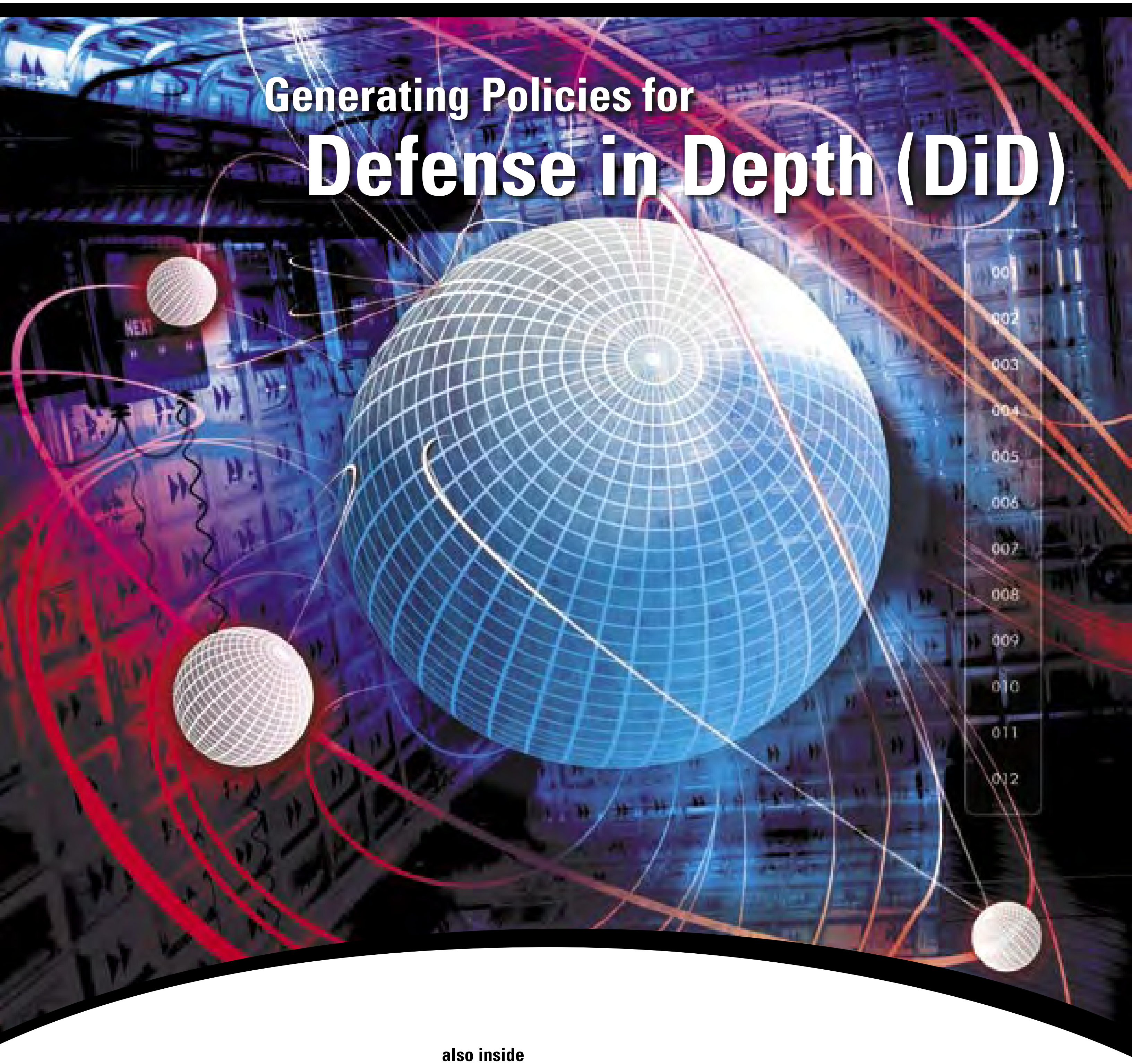
9/3

Volume 9 Number 3 • **Fall 2006**

IAnewsletter

The Newsletter for Information Assurance Technology Professionals



Generating Policies for Defense in Depth (DiD)

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: Dustin Hurt
Brad Whitford
Brittany Biggs
Kacy Cummings
Editorial Board: Greg McClellan
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
Spring 2007 Nov. 30th, 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

> The signing of DIACAP's Guidance document was not the only recent development in the IA community—the Enterprise-Wide Solutions Steering Group (ESSG) has also undergone some transformation.

6 July 2006 was a date that many had been looking forward to, especially several individuals from IATAC. You see, this was the date that Mr. John G. Grimes signed the *Interim Department of Defense Information Assurance (IA) Certification and Accreditation (C&A) Process Guidance*, also known as DIACAP. Mr. Grimes is the Assistant Secretary of Defense for Networks and Information Integration (ASD NII) and Department of Defense (DoD) Chief Information Officer (CIO). As you may recall from the article entitled, *Net-Centric Assured Information Sharing: Moving Security to the Edge through Dynamic Certification & Accreditation*, published in the Volume 8, Number 3, Winter 2005/2006 *IAnewsletter*, IATAC played a significant role in developing technical solutions pertaining to DIACAP. Superseding DoD's Information Technology Security Certification and Accreditation Process (DITSCAP), DIACAP now officially establishes DoD's IA C&A process for authorizing the operation of DoD information systems. It is DoD's approach to implementing a C&A process that supports Net-Centricity.

The signing of DIACAP's Guidance document was not the only recent development in the IA community—the Enterprise-Wide Solutions Steering Group (ESSG) has also undergone some transformation. The ESSG was established to integrate and synchronize solutions, to advocate adherence to IA strategic goals, and to field enterprise-wide Computer Network Defense (CND) solutions. The ESSG is chaired by United States Strategic Command (USSTRATCOM) with its Coordinator a part of USSTRATCOM. On 4 August 2006, Mr. Bob Ferguson stepped down as the ESSG Coordinator and Mr. John Palumbo assumed coordination responsibilities. One of Mr. Palumbo's first duties as Coordinator was to organize the September 2006 ESSG Meeting. Held in Omaha, NE from 12–14 September 2006, this meeting included a number of briefings from various voting members, including the National Security Agency (NSA), the Defense Information Systems Agency (DISA), and the Defense-wide Information Assurance Program (DIAP).

On 3 August 2006, IATAC was given the opportunity to lend support to the Army Inspector General (IG). Looking for information primarily in IA compliance and inspections, IA personnel and training, standards, and emerging technologies, we briefed a senior Army IG representative on the services and capabilities IATAC has to offer. Focusing on these areas of interest, we informed the Colonel of several free products and services such as the Total Electronic Migration System (TEMS), our various technical reports, our Scientific and Technical Information (STI) repository, the IA Digest, the *IAnewsletter*, our inquiry services, and much more. We also provided information on the Director, Defense Research and Engineering (DDR&E) Research and Engineering (R&E) Portal, the various Defense Technical Information Center (DTIC) Information Analysis Centers (IACs), and the Scientific and Technical Information Network (STINET). The brief went very well, and the Army IG left knowing of all the resources IATAC can provide for his office.

In this edition of the *IAnewsletter* you will once again find some very interesting articles. We are privileged to have an article from the Air Force Institute of Technology entitled, *Digital Forensics Education at the Air Force Institute of Technology (AFIT)*. We also feature a fascinating article on Border Gateway Security, *Efficient Path Authentication for Border Gateway Protocol (BGP) Security*, which reviews the analyses of performance issues raised by adding security measures to BGP. *Generating Policies for Defense-in-Depth (DiD)* is a wonderful article covering cost-effective and practical policies under development for a multi-level defense. Our featured Subject Matter Expert (SME) is Dr. Tzi-cker Chiueh of the State University of New York, Stony Brook, which is also our featured institute. For a glimpse of work Professor Chiueh is performing, be sure to read *A Virtual Environment for Safe Vulnerability Assessment (VA)*. ■

Gene Tyler

Generating Policies for Defense in Depth (DiD)

by Paul Rubel, Charles Payne, Michael Ihde, Steven Harp, and Michael Atighetchi

In 2002, the Defense Advanced Research Projects Agency (DARPA) challenged the research community to design and demonstrate an unprecedented level of survivability for an existing US Department of Defense (DoD) information system by combining Commercial-Off-The-Shelf (COTS) technologies with those developed by DARPA. In particular, DARPA required that the undefended system—a large, distributed, Publish/Subscribe/Query (PSQ) system, implemented using the Joint Battlespace Infosphere (JBI)—be defense enabled in such a way as to survive 12 hours of sustained attack from a Red Team modeling a nation-state adversary. (See Figure 1 for a notional diagram.) The development team, led by BBN Technologies, produced a solution architecture entitled *Designing Protection and Adaptation into a Survivability Architecture* (DPASA) [1] that combines the following elements:

- **Protection**—the ability to detect or prevent attacks;
- **Detection**—the ability to detect and report attack-related events; and
- **Adaptation**—the ability to modify system behavior and structure to repair attack damage or to degrade gracefully. (See Figure 2.)

In this article we focus on the defense mechanisms and policies that protect the system's network communications.

From a protection perspective, DPASA's goal was to block an attacker using the Defense in Depth (DiD) strategy illustrated in Figure 3. (The defense layer is shown in boldface, while the prevention technology(s) used at that layer appears in italics.) At the system layer, redundant hosts were deployed so that the failure of a single host would not stop the entire system. At the network layer, hosts were grouped into enclaves, and enclave-to-enclave communication was restricted and encrypted using a Virtual Private Network (VPN) firewall and router. At the host layer, authorized host-to-host communication was enforced by the Autonomic Distributed Firewall (ADF), a host-based, embedded, distributed firewall that is implemented on the host's Network Interface Card (NIC) and performs ingress and egress packet filtering. ADFs protect the host from the network and the network from the host. All host-to-host communication was also encrypted using ADF's Virtual Private Groups (VPG) [4], which provided a unique encryption key for each collection of hosts. At the process layer, authorized process behavior was enforced either by the National Security Agency's (NSA) Security Enhanced Linux (SELinux) or by Cisco System's Cisco Security Agent (CSA) for non-Linux hosts. At the application layer, the Java Virtual Machine (JVM) enforced authorized JBI application behavior. An application's network communications were subjected to defense mechanisms in each network, host, process, and application layer.

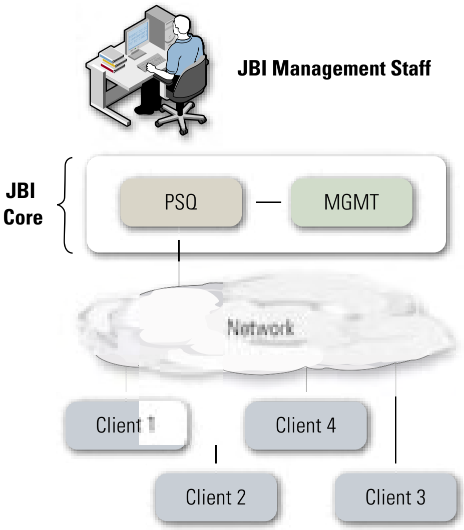
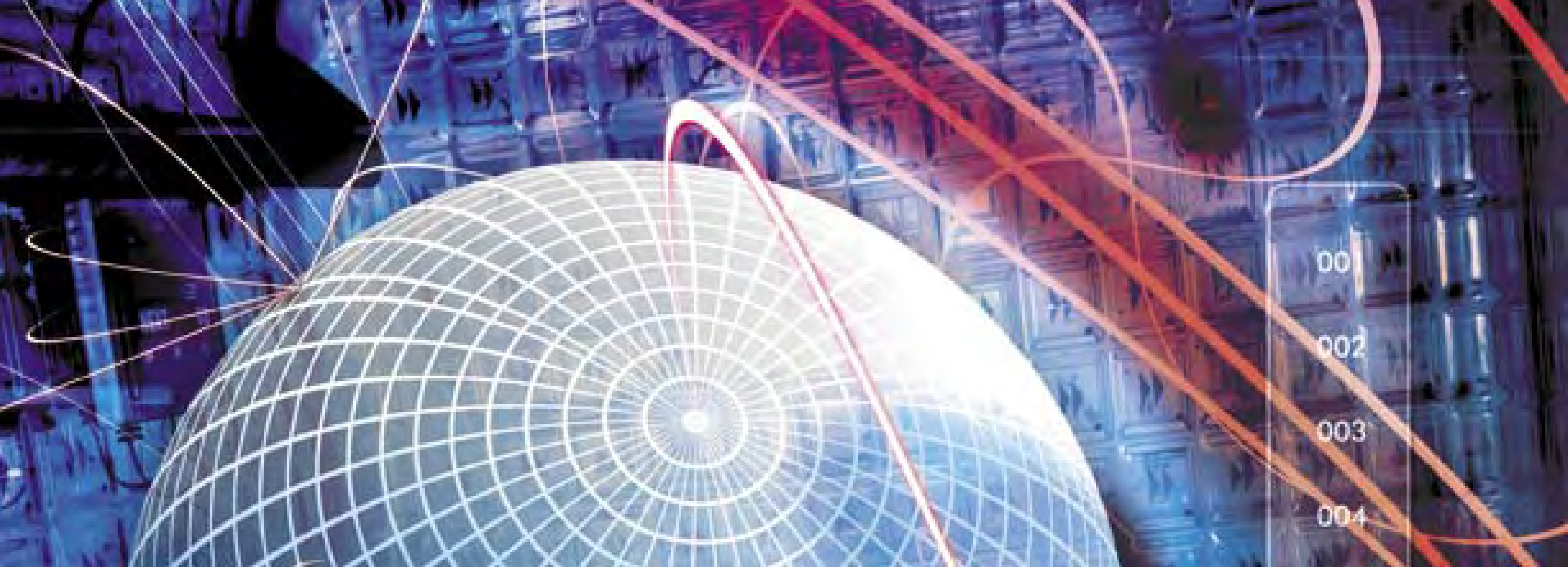


Figure 1 Baseline Joint Battlespace Infosphere (JBI)

Constructing the policies that govern these defense mechanisms across the multiple layers in a coherent and mutually consistent way proved to be a challenge on several fronts. The richness of DPASA's DiD strategy meant there was significant *vertical* duplication of logical policy rules across the defense layers. Because of the nature of DiD, significant *horizontal* duplication also occurred between redundant hosts and network elements. For example, to minimize common mode failures, a mix of operating systems was used when providing redundant services. This meant that



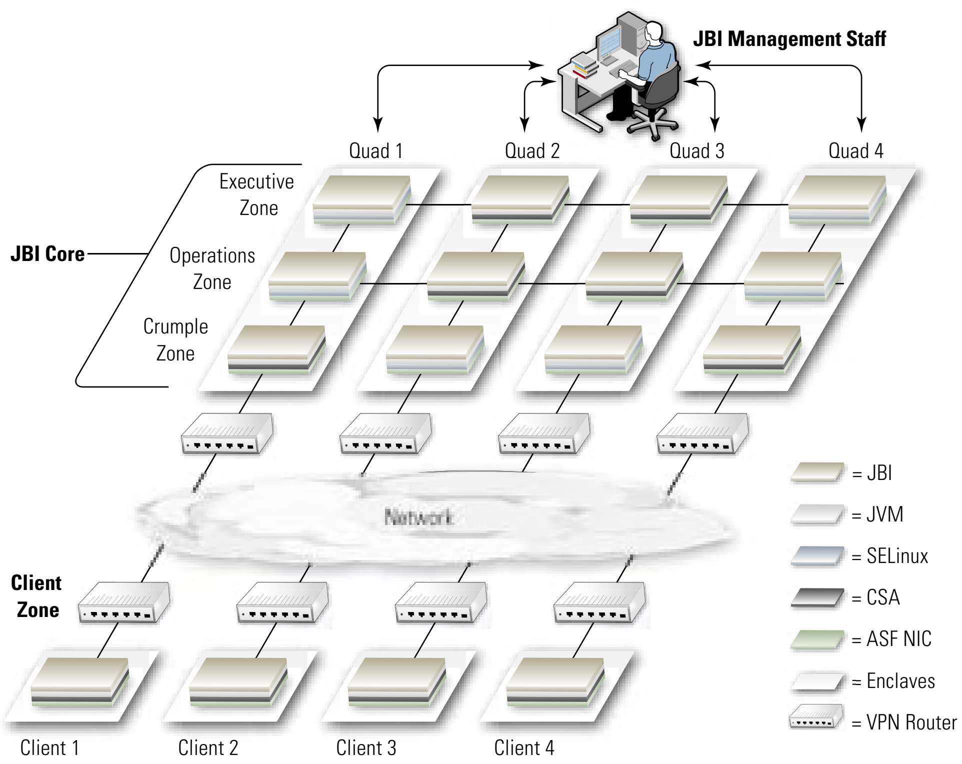


Figure 2 Survivable Joint Battlespace Infosphere (JBI)

actual policies enforced by similar hosts could differ significantly (*e.g.*, between an SELinux host and a CSA-enabled host), even though those hosts were performing identical logical functions.

Strategy

There is a strong motivation to develop a single specification from which all policies will be derived. This topic has been the focus of significant research (*cf.*, [5], [6], [7]), which has demonstrated that a master specification can eliminate unnecessary duplication and be analyzed effectively for desired properties. However, those research efforts focused on policy coordination for identical or similar defenses within a single defense layer. What about policy coordination across multiple defense layers, as in a DiD system? The variety of enforcement targets and range of abstractions, from IP addresses and gateways to network services and processes, means that any useful master specification must contain many details at discordant levels of abstraction; *i.e.*, not all details are required at all defense layers, and unnecessary details tend to get in the way when reasoning about a layer at which they are not required. A master specification also raises concerns about hidden assumptions that might yield exploitable vulnerabilities and circumvent any gains promised by DiD by compromising all layers at once.

We initially pursued the master specification approach for selected layers. For example, we began by generating the host-layer policy automatically from the application-layer policy. However, constraints on policy construction at the host layer soon made this process unwieldy. We then moved to a hybrid approach that created policies in a coordinated but largely independent fashion. This yielded the best balance of flexibility, autonomy (an important quality for DiD), and assurance of correctness. The approach relied on a central specification of common values, such as host names and port numbers, plus policy templates; *e.g.*, for Java (which enforced application-level defenses), SELinux (which enforced process-level defenses), and ADF (which enforced host-level defenses), that relied on these values. Changes to these common values could then be propagated automatically to all affected policies. Where possible, we used existing policies

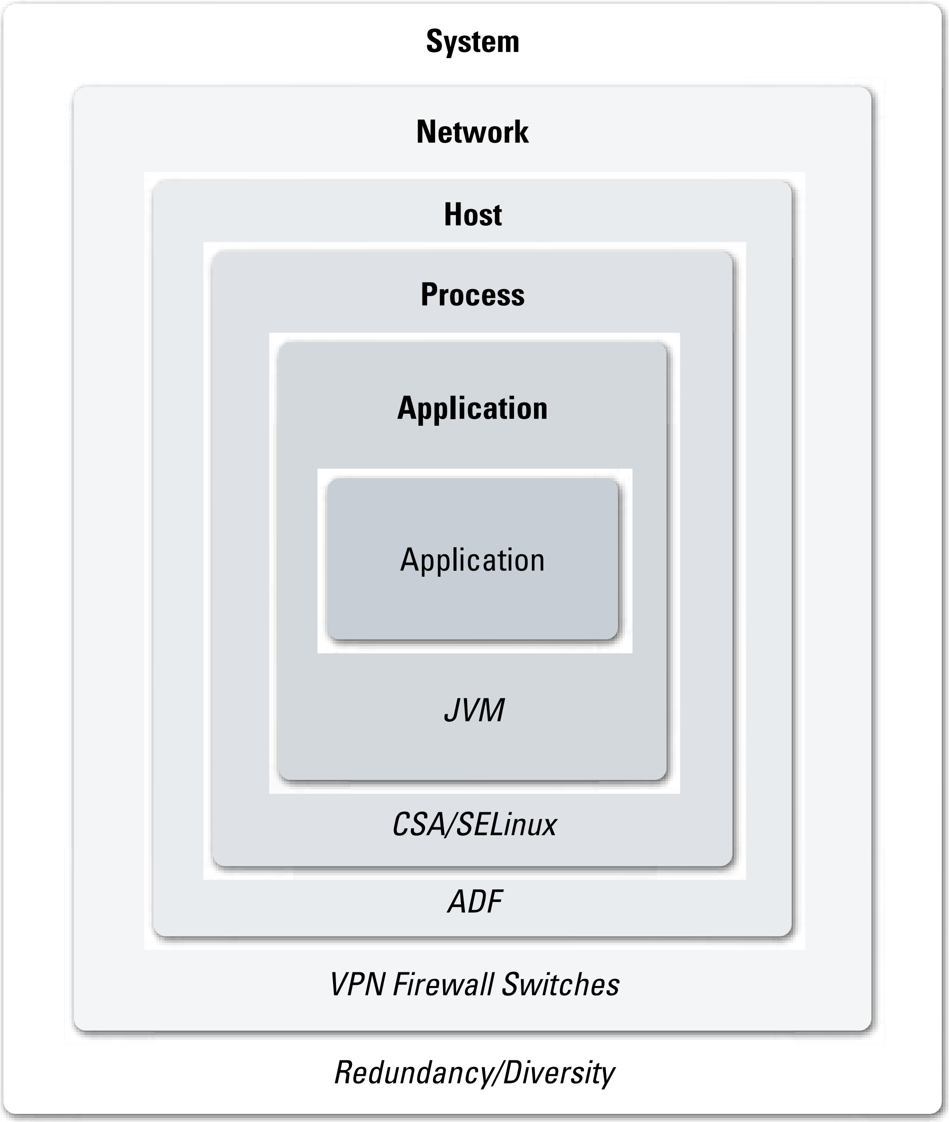


Figure 3 Attacker's Perspective of DPASA's Defense in Depth. (The defense layer is shown in boldface, while the prevention technology(s) used at that layer appears in italics.)

to inform new policies but did not automatically generate one from the other. For example, policies for CSA (which enforced process-level defenses) were developed independently from SELinux policies but were heavily informed by them.

To avoid simple misconfigurations, our hybrid approach shared information where appropriate but was not dogmatic about creating a single master specification. This approach also allowed us to use a single source to coordinate static policy elements shared by all policies. Separating functional roles from actual values allowed roles and values to change independently but kept then in synchronization between layers. This hybrid approach further minimized the risk of hidden assumptions by:

- ▶ Specifying each policy separately, using a different author,
- ▶ Structuring each policy to deny everything that is not explicitly allowed and then defining the policy according to observed failures to achieve a policy that was minimally sufficient, and
- ▶ Generating output to support manual and automated policy validation.

Generating validation support tools, such as visualizations and automated probing and testing, enabled software developers to review policies for correctness even if they did not understand the syntax of the policy-enforcement mechanism. For example, one automated validation task compared ADF policy against a thorough network scan. To conduct this task, we initiated a network scan from each host in the system to every other host, capturing the combined effects of egress filtering on the sending hosts and ingress filtering on the receiving hosts. The scan results were then automatically compared with the ADF policy to discover misconfigurations and unnecessary communication paths.

Lessons Learned

As in any large project, coordination and separation of concerns were important for making progress. In DPASA, we learned a number of valuable lessons along these lines as we developed and deployed the system.

- ▶ Our hybrid-policy construction worked well, because various authors could develop policies both independently and simultaneously. This was especially important, since policy authors were geographically dispersed. It also meant that it was not necessary for all authors to develop expertise in all technologies.
- ▶ Integration was greatly improved by having tools that supported command-line and file-based interactions. While Web-based interfaces are probably the friendliest for a novice user, they are awkward to integrate into a larger, multi-policy environment such as DPASA.
- ▶ Co-development of system functionality and policy is risky—developing policy against evolving system functionality causes undesirable churning. This can be avoided using one of two methods. First, by setting an acceptable but perhaps overly broad policy early on and then implementing the system to fit within the specified policy. This can be followed by a final refinement and tightening of the policy. Second, by implementing the system, being mindful of security concerns, and then creating the policy to tightly

fit the system requirements, as implemented. Since the undefended system did not have an existing set of policies, and we needed to incrementally build up the defense under a tight schedule, we chose the second option, which requires only one phase of policy construction.

- ▶ As the system grew, isolated testing became more difficult. Many policy refinements depended on observing the system in operation in a permissive mode, while collecting denial audits. In most cases, it was impossible to fully test applications in isolation, as related applications also had to be running. The constantly changing and challenging-to-test system impeded policy development to a surprising degree and underscored the need to reduce coupling between applications.

Conclusions

In DiD-enabled systems, constructing each policy in isolation is labor intensive and can lead to configuration errors. However, generating all policies from a single specification—an approach advocated for policies within a particular defense layer such as the network layer—is perhaps even more labor intensive and prone to error for DiD solutions, because too many details in that specification will apply only to specific layers, creating an unwieldy specification. Instead, we advocate a hybrid approach that:

- ▶ Encourages selective sharing of policy elements, while maintaining policy autonomy,
- ▶ Encourages independence between policy authors to reduce common, faulty assumptions,
- ▶ Builds policies from observed failures to be minimally sufficient, and
- ▶ Integrates validation tools to support other policy stakeholders.

Such an approach minimizes the risk of exploitable vulnerabilities that could circumvent the benefits of DiD.

A critical measure of success, of course, is how well the resulting policies and the defense mechanisms enforcing them perform against a determined adversary. At this writing, analysis of the Red Team assessment of DPASA is ongoing; however, preliminary results confirm that the overall DPASA solution architecture puts up a formidable defense. We believe that this approach is a solid step forward for making enforcement of multi-layer defense both cost-effective and practical. ■

Acknowledgement
This work was supported by DARPA under Contract number F30602-02-C-0134.

References

1. Atighetchi, Michael, Paul Rubel, Partha Pal, Jennifer Chong, & Lyle Sudin. Networking Aspects in the DPASA Survivability Architecture: an Experience Report. *Proceedings of The 4th IEEE International Symposium on Network Computing and Applications (IEEE NCA05)*, Cambridge, MA, 2005.
2. Chong, Jennifer, Partha Pal, Michael Atighetchi, Paul Rubel, & Franklin Webber. Survivability Architecture of a Mission Critical System: The DPASA example. *Proceedings of the 21st Annual Computer Security Applications Conference*, Tucson, AZ, 2005.
3. Rubel, Paul, and Michael Ihde, Steven Harp, & Charles Payne. Generating Policies for Defense in Depth. *Proceedings of the 21st Annual Computer Security Applications Conference*, Tucson, AZ, 2005.
4. Markham, Tom, Lynn Meredith, & Charles Payne. Distributed Embedded Firewalls with Virtual Private Groups. *DARPA Information Survivability Conference and Exposition Volume II*, 2003.
5. Bartal, Yair, Alain Mayer, Kobbi Nissim, & Avishai Wool. Firmato: A Novel Firewall Management Toolkit. *ACM Transactions on Computer Systems*, 22(4):381–420, November 2004.
6. Uribe, Tomas E., & Steven Cheung. Automatic Analysis of Firewall and Network Intrusion Detection System Configurations. *Proceedings of the 2004 ACM Workshop on Formal Methods in Security Engineering*, Washington, DC, 2004.
7. Guttman, Joshua D. Filtering Postures: Local Enforcement for Global Policies. *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, Oakland, CA, 1997.

About the Authors

Paul Rubel | is a staff scientist in the Quality Objects (QuO) group at BBN Technologies. His research focuses on adaptive distributed systems, specifically fault-tolerant middleware and survivable systems. Mr. Rubel is a member of the Institute of Electrical & Electronics Engineers (IEEE) and can be reached at prubel@bbn.com.

Charles Payne | has performed computer security research for nearly 20 years at industry and government laboratories. His work has focused on simplifying the definition and management of computer security policies and on streamlining the construction of convincing assurance arguments to ensure that these policies are upheld. Other research interests include host-based, distributed firewalls, high-assurance computing systems, and formal methods. Mr. Payne is a member of the IEEE Computer Society and may be reached at charles.payne@adventiumlabs.org.

Michael Ihde | received an MS from the University of Illinois and is currently a Software Engineer with Northrop Grumman Information Technology. As a graduate student, he conducted research on verification and validation of survivable, distributed systems, with an emphasis on distributed firewall technologies. He may be reached at mike.ihde@randomwalking.com.

Steven Harp, PhD | is a staff scientist at Adventium Labs with a background in cognitive science, artificial intelligence, and statistics. He has been studying applications of automated reasoning to problems in computer and physical security for the last five years. He may be reached at steven.harp@adventiumlabs.org.

Michael Atighetchi | is a Senior Scientist in the Quality Objects (QuO) group at BBN Technologies. He conducts research on enabling technologies for advanced distributed systems, with a major focus on using adaptation in survivable systems, network and operating system security, and distributed coordination. He may be contacted at matighet@bbn.com.

Footnotes

[1] Designing Protection and Adaptation into a Survivability Architecture. More details about the architecture can be found in [1], [2], and [3].

A Virtual Environment for Safe Vulnerability Assessment (VA)

by Yang Yu, Fanglu Guo, and Dr. Tzi-cker Chiueh

As vulnerabilities in computer systems have multiplied in recent years, Vulnerability Assessment (VA) has emerged as an important security administration technique that can identify system vulnerabilities before they are exploited. However, current vulnerability scanners cannot guarantee that their scanning is not intrusive to the network applications being scanned. This article discusses the problem with safe VA and introduces a *VA supporting system*, which can quickly duplicate the production-mode network applications to a virtual environment for vulnerability scanning and can automate the entire scanning process. This technique makes it feasible for system administrators to perform VAs on their systems safely and frequently.

Vulnerability Assessment

Every computer system may have vulnerabilities. Programming errors in the software code (*e.g.*, buffer overflow vulnerability) cause some vulnerabilities, and others are related to misconfigurations of the system (*e.g.*, running unnecessarily vulnerable services). As attack tools become more user friendly and automated, more script kiddies can use them to randomly scan the Internet for victims with unpatched vulnerabilities. To make things worse, while system administrators need to patch every possible security hole in their systems, an attacker only needs to locate one to break in.

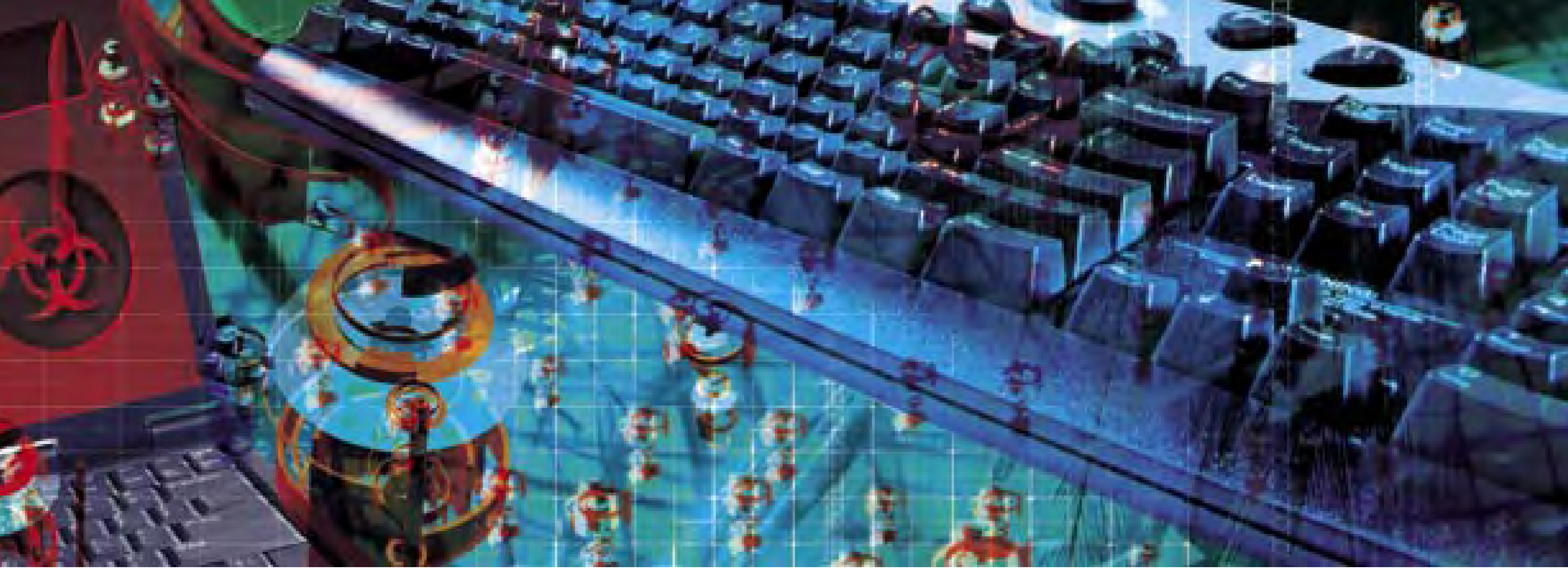
VA is the process of identifying known vulnerabilities in computer systems and networks. Different from an *Intrusion Prevention System* (IPS), which filters out network packets containing attack payloads, and antivirus software, which scans host computers for malicious contents, VA permits system administrators to proactively find security holes before any real attacks can exploit them. Once identified and patched properly, vulnerabilities exposed to attackers can be significantly reduced.

A *VA scanner* is a program specifically designed to scan specified computer systems and networks to perform VA automatically. It determines if the system being scanned contains certain known vulnerabilities by probing for open ports, checking patch levels, scanning registries, or simulating real attacks. According to the location at which the scanner is running, a scanner can be classified as *host-based* or *network-based*. The scanning process itself can be further classified as *passive* or *active*: passive scanning does not generate network traffic and works more like an IPS, while active scanning generates probe packets and is more helpful in proactively finding vulnerabilities. There are dozens of vulnerability scanners on the market, such as Nessus [1], FoundStone [2], Microsoft Baseline Security Analyzer [3], eEye's Retina [4], *etc.* Each product differs in scanning techniques used, vulnerabilities detected, assessment reports generated, detection accuracy, and performance.

Side Effects of Vulnerability Scan

Although VA scanners can efficiently collect information about a system's security state, these tools may introduce a safety problem to the production-mode network server applications, such as a Web server. The problem is the undesired side effects during the vulnerability scan. In a report that tested 11 VA scanners [5], all scanners caused adverse effects on the

Because of the potential undesirable impact on production-mode network servers, vulnerability scanners may not be able to work as extensively as IPS, firewall, or antivirus software.



scanned network servers. One scanner even crashed at least five servers during an assessment run. According to Nessus documentation [1], with every network-based vulnerability scanner comes the risk of crashing the scanned systems and services or, even worse, leaving permanent, damaging side effects. This security issue is not specific to some scanners but a hard fact about vulnerability scanning in general.

Most VA scanners, such as FoundStone [2], HARRIS's STAT Scanner [6], and Nessus [1], are aware of this security issue and allow system administrators to do an optional "safe scan" or "non-intrusive check," which can help to avoid probable Denial-of-Service (DoS) and mitigate the intrusiveness of the scan. However, such a scanning option sacrifices the accuracy of vulnerability detection [2]. Moreover, it turns out that "safe scan" may not be truly safe, as reported by Kevin Kovak in "VA Scanners Pinpoint Your Weak Spots." [5]

In one sense, these testing results are not surprising, because many types of vulnerability scanning packets *should* behave like real attacks to expose vulnerabilities. There are several reasons why some side effects could happen in practice. For example, some protocol implementations of scanned applications do not handle errors very well and may crash the process on receiving unexpected inputs. Also, if the vulnerability is related to memory errors; *e.g.*, buffer overflow vulnerability, a scanner may send enough data to overflow the buffer, and the overflow could result in unpredictable program execution, including program crash or undesirable modifications to the system state.

Because of the potential undesirable impact on production-mode network servers, vulnerability scanners may not be able to work as extensively as IPS, firewall, or antivirus software. Many only scan their systems once a month or once a quarter, even though new vulnerabilities may be found every day. In a case study [7], the author was aware of the importance of regular vulnerability scanning, but he eventually decided to scan his system only after midnight once a week because of the safety concern. So the question is: *Can we use a VA scanner to scan our systems frequently, without worrying about its impact to our business-critical server applications?*

Scanning a Duplicated Environment

To protect network server applications in a production-mode environment from side effects of intrusive VA scanners, we developed a *VA supporting system* based on an *intrusion isolation* idea. The idea is to run the vulnerability scan against a separate testing environment, which has the same network applications and system configurations as the production-mode environment. This testing environment should have the following attributes:

- **Full-State Isolation**—The testing environment should be isolated from the production-mode environment, so server crashes and other side effects caused by a vulnerability scan will not affect production-mode network servers.
- **High Testing Fidelity**—Network applications duplicated to the testing environment should be as close as possible to original production-mode applications, so the VA report in the testing environment can be used by system administrators to patch the production-mode environment.
- **Fast Duplication**—To support frequent vulnerability scans, the system should duplicate production-mode network applications to the testing environment very quickly.
- **Small Overhead**—Performance overhead imposed to the production-mode environment should be negligible. The resource requirement of the testing environment should also be minimal.

Given this idea, the question is how to actually create such a testing environment. Two available options present themselves—using a separate physical machine or using a virtual machine. A separate physical machine can be fully isolated from the production-mode network servers. However, it takes too long to duplicate the entire production-mode environment to a separate machine. It is also difficult to synchronize patches or reconfigurations of the production-mode environment to the testing machine in a timely manner. Finally, using a separate machine incurs a fixed cost. Therefore, using a virtual machine may be a better choice.

Conventional Virtual Machines

Virtual Machine (VM) is a technology that creates one or multiple isolated operating environments on a single physical machine through a layer of software. Each virtual machine represents a distinct operating environment that gives users an illusion of directly accessing a physical machine. There are different levels of abstraction at which virtualization can take place, such as the *hardware level* and the *Operating System (OS) level*. The *hardware level* virtualization layer presents a hardware abstraction to run guest operating systems, while the OS level virtualization layer presents an OS interface to run application programs, as shown in Figure 1. Most conventional virtual machines, such as VMware and Microsoft Virtual PC, have the virtualization interface at the hardware-abstraction layer. They virtualize common PC hardware such as processor, memory, and peripheral Input/Output (I/O) devices such that multiple OS instances of different types can be installed on a single, physical x86 machine.

Using conventional hardware-level virtual machines as the testing environment for vulnerability scans avoids the cost of a separate physical machine. Besides, such virtual machines provide full-state isolation so that even intrusive vulnerability scans in the testing environment cannot interfere with production-mode network services. However, each virtual machine is a full-fledged operating environment, which includes not only the duplicated network applications but also the OS and file-system image. When duplicating a production-mode environment or synchronizing its configuration changes to a virtual machine, it may take as long a period of time as the same operations using a physical machine. Consequently, conventional virtual machines cannot satisfy the requirement for fast cloning a production-mode environment. In contrast, the OS level virtual machines share a common OS and file-system image and are therefore more suitable in meeting the requirement of fast duplication.

A Featherweight Virtual Machine (FVM)

OS level virtual machines, such as Linux VServer [8] and Solaris Containers [9], create multiple virtual machines atop the same host operating system. These virtual machines share most system resources and are normally isolated from one another by namespace separation and resource copy-on-write. In our VA supporting system, we developed an OS level virtual machine, the *Featherweight Virtual Machine* (FVM) [10], on a Microsoft Windows platform. (See Figure 1.) Before a vulnerability scan takes place, the production-mode Windows applications will be duplicated to an FVM, which acts as the target of VA scanners.

The key idea behind FVM is *namespace virtualization*, which renames machine resources at the system-call interface. FVM also uses *resource sharing* and *copy-on-write* to reduce the resource requirement and duplication overhead of each virtual machine. When an FVM is created, it shares everything with a host machine by default. However, when it tries to update any resource, the target resource will be copied to the virtual machine's own workspace. For example, suppose a process in one virtual machine (say **vm1**) tries to access a file **/a/b**. If the access is a "read" request, FVM allows the process to access **/a/b** directly. If it is a "write" request, FVM will copy **/a/b** to **/vm1/a/b** and transparently redirect subsequent file access from **/a/b** to **/vm1/a/b**. In this way, any permanent side effects in the virtual machine are isolated from the host environment. To prevent DOS attacks and to also support performance isolation, FVM allows a set of policies regarding resource quota and network access to be specified for each virtual machine, such as Central Processing Unit (CPU) scheduling priority, memory limit, disk-space utilization, and network traffic. When processes in a virtual machine leave undesired side effects, these effects can be easily cleaned by terminating and deleting the virtual machine.

FVM is designed to be a comprehensive OS level virtualization technique that can achieve strong isolations between different virtual machines and a host machine. It virtualizes different types of system resources with the following virtualization modules:

- Console-process virtualization
- Service-process virtualization
- File virtualization
- Object virtualization
- Interprocess communication confinement
- Network interface virtualization

Through extensive virtualization and confinement, an FVM provides several features that meet all requirements of a testing environment in our VA supporting system. First, FVM allows a virtual machine to be fully isolated

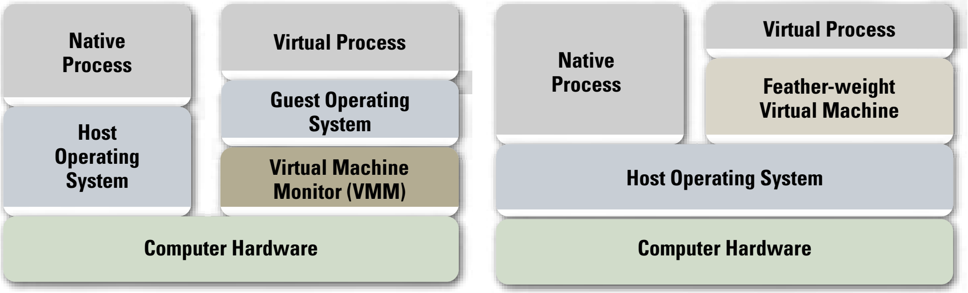


Figure 1 Conventional VM at hardware-level and FVM at OS level

The safety issue with vulnerability scanning is a well-known problem, because every vulnerability scanner can be intrusive to production-mode network applications. A VA scanner crashing scanned services can lead to the same financial loss as that caused by a real attack. How intrusive the scanning process can be has become an important criterion in selecting VA scanner products. However, no matter how gentle the scanner is, the risk remains.

from a host machine. Any updates to a virtual machine's persistent state, such as files or registries, are contained within the virtual machine without affecting any applications running on the host machine. Second, an FVM has the same patches, configurations, and operating environment as the host machine. Running vulnerability scanners against a virtual machine can achieve high testing fidelity. Third, an FVM can be cloned from a host machine in seconds. This permits VA to be performed frequently and efficiently. Finally, resource renaming at the system-call interface creates very small performance penalties to the production-mode network applications running on the host machine.

VA Supporting System

With FVMs that can support fast duplication and strong isolation, we implement a *Vulnerability assessment supporting engine (Vase)* that can automatically direct a VA scanner, such as Nessus, to perform VA on duplicated Windows network applications in an FVM. Figure 2 shows the architecture of *Vase*.

To automate a vulnerability scanning process, *Vase* needs to create an FVM on a production-mode environment and to start all network applications currently running on the production-mode machine in the new virtual machine in exactly the same way as they were started originally. Each vulnerability scan controlled by *Vase* is composed of the following four phases:

1. Enumerate all network applications running on a target network server for VA and obtain a complete list of these applications, including their configuration information.
2. Create a new FVM and install and start all the network applications identified in the first phase in the virtual machine.
3. Invoke a VA scanner, such as Nessus, against the FVM that runs all duplicated network applications. The VA scanner will generate an assessment report.
4. After the vulnerability scanning is complete, stop all processes running in the virtual machine and, finally, stop and delete the virtual machine.

Obviously, even if the VA scanners are intrusive and crash some duplicated network applications being scanned in an FVM, the original production-mode network applications are still unaffected.

Conclusion

The safety issue with vulnerability scanning is a well-known problem, because every vulnerability scanner can be intrusive to production-mode network applications. A VA scanner crashing scanned services can lead to the same financial loss as that caused by a real attack. How intrusive the scanning process can be has become an important criterion in selecting VA scanner products. However, no matter how gentle the scanner is, the risk remains.

We designed a VA supporting system to resolve the problem. The basic idea is intrusion isolation and tolerance: We duplicated production-mode network applications into an isolated environment and ran the VA scanners against that environment. To support fast duplication and small resource requirements, we developed an OS level virtual-machine

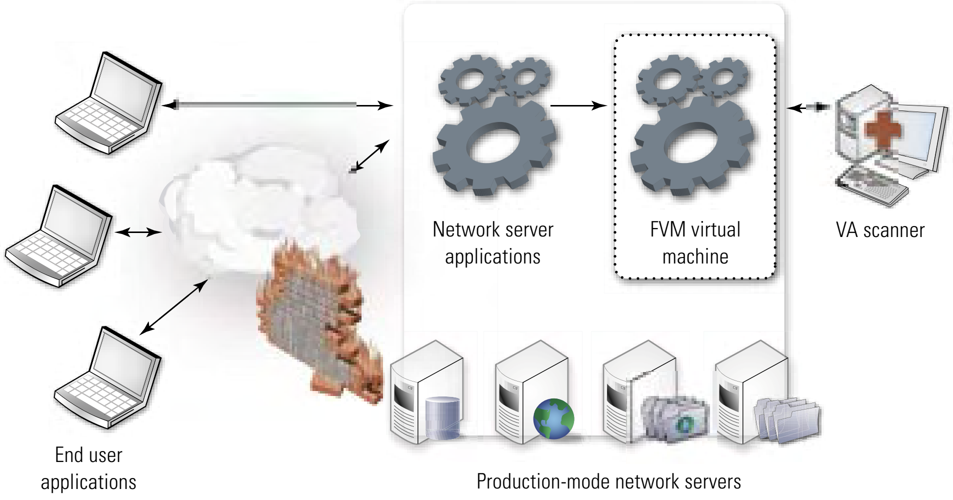


Figure 2 Architecture of the *Vase*

technique on a Windows platform and used it as the isolated scanning target. Experiments on a working prototype demonstrate that our approach can provide strong isolation, high fidelity, automation, and transparency to the vulnerability scanning process. Moreover, the performance impact on the original production-mode network services is as low as 3%. As a result, system administrators can run any VA tools to scan their systems safely and frequently, without worrying about the intrusiveness to their business-critical system and services.

In the future, we will further improve the isolations of the FVM environment used to achieve unintrusive VA. Such a technique can also be used to confine untrusted mobile code on desktop computers or by an *application streaming* [11] client to host application processes whose images are maintained on the server while executed on the client machine. ■

Reference

1. Nessus, “How enabling local security checks can improve my scanning experience,” *http://www.nessus.org/documentation/index.php?doc=ssh#BETTER.SCANNING.EXPERIENCE*
2. Foundstone Labs, “Scanning Safety,” November 2003, *http://www.foundstone.com/resources/whitepapers/wp_scanningsafety.pdf*
3. Microsoft TechNet, “Microsoft Baseline Security Analyzer,” August 2005, *http://www.microsoft.com/technet/security/tools/mbsahome.mspx*
4. eEye Digital Security, “Retina Network Security Scanner,” *http://www.eeye.com/html/products/retina/index.html*
5. Kevin Novak, “VA Scanners Pinpoint Your Weak Spots,” June 2003, *http://www.networkcomputing.com/1412/1412f2.htm*l
6. Alec Yasinsac, “Managing Network Vulnerability: A Paper on Vulnerability Management,” *http://www.stat.harris.com/technologies/whitepapers/index.asp*
7. Kevin Austin, “Implementing Vulnerability Assessment with eEye’s EVA Suite—Case Study,” January 2004, *http://www.giac.org/certified professionals/practicals/GSEC/3595.php*
8. Herbert Pötzl, “Linux-VServer Technology,” 2004, *http://linux-vserver.org/Linux-VServer-Paper*
9. Sun Microsystems, “Solaris containers: Server virtualization and manageability,” September 2004, *http://www.sun.com/software/whitepapers/solaris10/grid_containers.pdf*
10. Yang Yu, Fanglu Guo, Susanta Nanda, Lap-chung Lam and Tzi-cker Chiueh, “A Feather-weight Virtual Machine for Windows Applications,” to appear in the Proceedings of the 2nd ACM/USENIX Conference on Virtual Execution Environments (VEE’06), June 2006.
11. Andy Dornan, “Application Streaming: The Virtual Thin Client,” January 2006, *http://www.itarchitectmag.com/shared/article/showArticle.jhtml?articleId=175001526&pgno=1*

About the Authors

Yang Yu | is a PhD candidate in Computer Science Department at Stony Brook University. His research focuses on virtualization and on OS and system security. He received BS and MS degrees in Computer Science from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and an MS degree in Computer Science from Stony Brook University in 2005. He may be reached at yyu@cs.sunysb.edu.

Fanglu Guo | is a PhD candidate in Computer Science Department at Stony Brook University. His research focuses on network security and wireless networking. He received a BE degree from Xi’an Jiaotong University, Xi’an, China, and an ME degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1996 and 1999, respectively. From 1999 to 2001, he was a Research Engineer with Huawei Beijing Research, Huawei Technologies for high-performance routers, Beijing, China. He may be reached at fanglu@cs.sunysb.edu.

Dr. Tzi-cker Chiueh | is a Professor in the Computer Science Department at Stony Brook University. He received a BS degree in Electrical Engineering from National Taiwan University, an MS degree in Computer Science from Stanford University, and a PhD in Computer Science from the University of California at Berkeley in 1984, 1988, and 1992, respectively. He received a National Science Foundation (NSF) CAREER award in 1995, an Institute of Electrical & Electronics Engineers (IEEE) Hot Interconnect Best Paper award in 1999, a 1997/2004 Long Island Software Award, and a Best Paper Award from the 2005 Annual Computer Security Applications Conference (ACSAC). Dr. Chiueh has published over 140 technical papers in refereed conferences and journals. His current research interests are focused on wireless networking, computer security, and storage systems. He may be reached at chiueh@cs.sunysb.edu.

Black Hat and DEFCON

Two of the largest hacker conferences were held during the first week of August in Las Vegas, NV. Black Hat 2006 was held 2–3 August at Caesar's Palace, and DEFCON 14 was held 4–6 August at the Rivera on the Las Vegas strip.

For five days, security professionals met to hear presentations by their peers on cutting-edge security topics. Black Hat is often considered by many to be the "good guy" hacker conference; DEFCON is considered just the opposite; however, both are considered the top offerings in the cyber-security industry. Black Hat hosted over 2,500 attendees and DEFCON over 6,000 attendees. Both conferences contained excellent technical material. Black Hat had tracks such as Database Security, Voice Services Security, Forensics, Zero Day Attacks, Web Security, Hardware Security, Rootkits, Cross-side Scripting and Windows Vista Security; and DEFCON had similar tracks such as Hardware Hacking, Malware, Wireless, and Privacy.

Below is a sampling of some high-level items of interest from the conferences:

- ▶ Claudio Merloni and Luca Carettoni presented the Bluebag, a lazy man's Bluetooth scanner. The luggage-style bag includes a small portable computer running on batteries; a long-range, omni-directional antenna; and eight Bluetooth adaptors. The Bluebag is capable of sniffing up to 200 m and can run for 10 hr on batteries.
- ▶ "Johnny Cache" and David Maynor gave a demonstration on hacking hardware drivers. The team was able to get root access to a Mac book through a vulnerability in a third-party device driver. The team would not release the name of the manufacturer of the wireless device.
- ▶ Microsoft gave away several early copies of its new operating system, Windows Vista, and invited conference attendees to try to hack it. The security professionals at Black Hat were successful in hacking the new operating system, and Joanna Rutkowska was successful in circumventing some of the security controls built into Vista and to run unsigned code.
- ▶ Colin Mulliner of the Trifinite Group presented a flaw discovered in Windows CE 4.2. The flaw allows a hacker to connect to a portable device, such as a phone or PDA, install remote software, and connect to the Internet. The flaw was reported to Microsoft before the presentation but was not corrected before the conference.

For five days, security professionals met to hear presentations by their peers on cutting-edge security topics. Black Hat is often considered by many to be the "good guy" hacker conference; DEFCON is considered just the opposite; however, both are considered the top offerings in the cyber-security industry.

Both Black Hat and DEFCON are renowned for bringing together security individuals from around the world. While the intent of the individuals who attend the two conferences may vary greatly, there is no disputing that the technical information provided at the conferences provides knowledge equally. It is then up to an individual to determine how he or she intends to use this information. ■

Efficient Path Authentication for Border Gateway Protocol (BGP) Security

by Meiyuan Zhao, Sean Smith, and David Nicol

The *Border Gateway Protocol (BGP)* [1] controls inter-domain routing of the Internet and is subject to many attacks. The root problem is that routers rely on hearsay information from neighbors. *Secure BGP (S-BGP)* [2] applies digital signatures to provide route authentication and can mitigate many risks. However, several performance and deployment issues prevent S-BGP's real-world deployment. S-BGP not only introduces significant processing latencies, it also has higher space costs created by increased message size and memory cost.

In this article, we summarize our analyses of performance issues raised by adding security measures to BGP and discuss the Aggregated Path Authentication (APA) schemes originally proposed by Meiyuan Zhao *et al.* in "Aggregated Path Authentication for Efficient BGP Security." [3] Previously, we proposed a Signature Amortization (S-A) scheme to speed up S-BGP path authentication [4] but observed that it increases memory costs, even beyond S-BGP. APA schemes solve the memory problem without sacrificing processing speed. APA schemes combine two efficient cryptographic techniques—S-A and *aggregate signatures*. We design six constructions for APA. These schemes dramatically improve performance beyond S-BGP in both processing speed and memory requirements. Compared with original BGP, the slowdown of our algorithms is minimal. Furthermore, APA schemes can reduce by as much as 60% of the space overhead by S-BGP path authentication. With such efficiency, we conclude that APA schemes provide the same security as does S-BGP, without the overhead.

BGP and S-BGP

BGP is an inter-domain routing protocol that controls routing between *Autonomous Systems (ASs)*. Each AS has a unique *AS number* as its identifier. ASs are connected *via* their BGP routers, which establish BGP sessions with their neighbors. Routing tables are established using the information from *Update messages*. Whenever there is a change in the routing table, the router sends *Update message*s to inform its neighbors, either announcing routes or withdrawing routes that were previously announced. A *route* has two major attributes: an *IP prefix* representing the destination and an *AS path*, a sequence of AS numbers describing the path to the destination. The AS path is extended at each hop of the route propagation. That is, each router prepends its local AS number to the path before sending the route to the next hop. Receiving multiple routes to the same prefix, *p*, the router chooses the "preferred" route, based on the route attributes and local domain policies.

It has been recognized for some time that the BGP security is a critical problem [2], [5]. BGP is vulnerable to malicious adversaries and configuration errors. Because BGP routers completely believe the route information sent from neighbors, falsified route information may be used and quickly propagated through the network. The Internet needs a good authentication mechanism to provide route announcement authenticity.

S-BGP [2] was proposed to provide authenticity to the information conveyed in messages and authorization for BGP routers to represent certain ASs. Routers digitally sign Update messages before

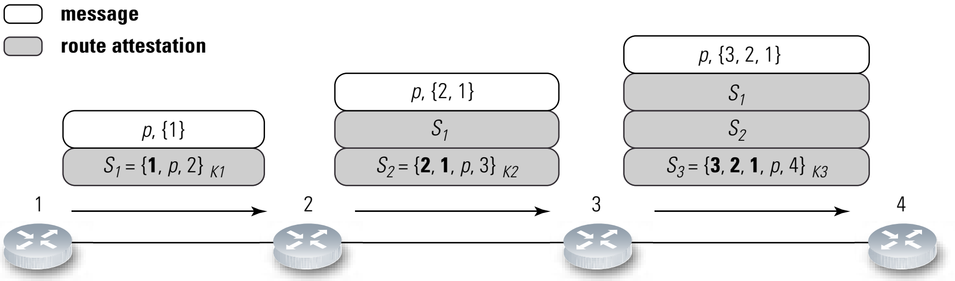

message
route attestation
p, {1}
$S_1 = \{\mathbf{1}, p, 2\}_{K1}$
p, {2, 1}
S_1
$S_2 = \{\mathbf{2}, \mathbf{1}, p, 3\}_{K2}$
p, {3, 2, 1}
S_1
S_2
$S_3 = \{\mathbf{3}, \mathbf{2}, \mathbf{1}, p, 4\}_{K3}$
1
2
3
4

Figure 1 This figure, adapted from Meiyuan Zhao *et al.* in "Evaluating the Performance Impact of PKI on BGP Security," [6] sketches the process of sending route announcements and their route attestations. There are four routers that speak for AS 1, 2, 3, and 4, respectively. Each router uses its corresponding private key *Ki* to generate route attestations. The figure shows the AS path components in bold.

sending them out. Recipients only accept routes that can be properly verified. Digital signatures are carried in *attestations*. *Address Attestations (AAs)* validate IP prefixes in routes. *Route Attestations (RAs)* authenticate path information. S-BGP uses public key certificates to distribute and validate public keys.

Figure 1 uses an example to illustrate the process of sending route announcements and associated route attestations for *path authentication*, ensuring that a propagated path is authorized by each AS in the path. A route attestation is signed by a BGP router to authenticate the existence and position of an AS number in an AS path. [2] The process is as follows: First, the origin BGP router signs the AS number of the origin AS, the prefix, and the intended receiver (in the form of AS number). The next signer who receives this RA computes and signs the concatenation of the new AS path, the prefix, and intended receiver. The process continues as the route is propagated. The inclusion of the intended receiver prevents "cut-and-paste" attack by effectively linking RAs together. RAs by all routers on the path are required to validate the entire AS path.

The route attestation has several serious performance issues. First, computational overhead is introduced by signing and verifying RAs. Routers perform one signing operation and *k* verification operations for each preferred route announcement, if AS path length is *k*. Second, message size increases. Additional *k* RAs are required for one route announcement. To mitigate this issue, S-BGP uses Digital Signature Algorithm (DSA), since the signatures are relatively short compared with Rivest-Shamir-Adleman (RSA). Finally, there exists a memory problem. As one of the S-BGP optimizations, routers cache the signed and verified routes in local storage to decrease the number of operations. Noted by Steve Kent in "Securing the Border Gateway Protocol: A Status Update" [7] for S-BGP in 2003, it required about 30–35 MB to store RAs for one neighbor, and the total requirement for a router with tens of neighbors may reach gigabytes. This amount is beyond the configured amount of memory for typical BGP routers (*e.g.*, 128 MB or 256 MB of RAM).

Aggregated Path Authentication (APA)

Unlike S-BGP, which expedites process at the cost of space, APA schemes improve processing latency and also reduce memory burden on both routers and network. APA schemes maintain the strong security that S-BGP provides, while also providing more efficient BGP path authentication. Such design facilitates its practical deployment on the Internet. As we will show later, APA schemes dramatically reduce memory requirements and perform much faster than S-BGP, even with its performance optimizations.

APA schemes combine two techniques—S-A and aggregate signatures. S-A [4] speeds up processing by reducing the number of signing operations that routers perform. Aggregate signatures [8], [9], [10] are space efficient cryptographic techniques that allow multiple signers to cooperatively generate one aggregate signature on their own messages. By combining these two techniques, the resulting APA schemes can be both fast and space efficient.

Figure 2 sketches the design of APA. It illustrates how routers generate route attestations using an APA scheme. This process entails two steps:

1. A router aggregates as many messages as possible to generate an aggregated message to be signed. In this way, only one signing operation is required for multiple outgoing update messages. This step uses S-A techniques.
2. On each hop of route propagation, a router generates an aggregate signature for route attestation, using the aggregated value computed locally and the aggregate signature received from the previous hop. The main advantage of using aggregate signatures is that one signature is enough to convince verifiers that each signer did correctly sign its own message. This way, routers use one signature to validate the entire AS path. This step uses aggregate signature techniques.

As shown in Figure 2, Step One, the router constructs the message to be signed using the S-A techniques. BGP routers keep outgoing Update messages in output buffers, one buffer for each neighbor. The S-A technique allows a router to collect

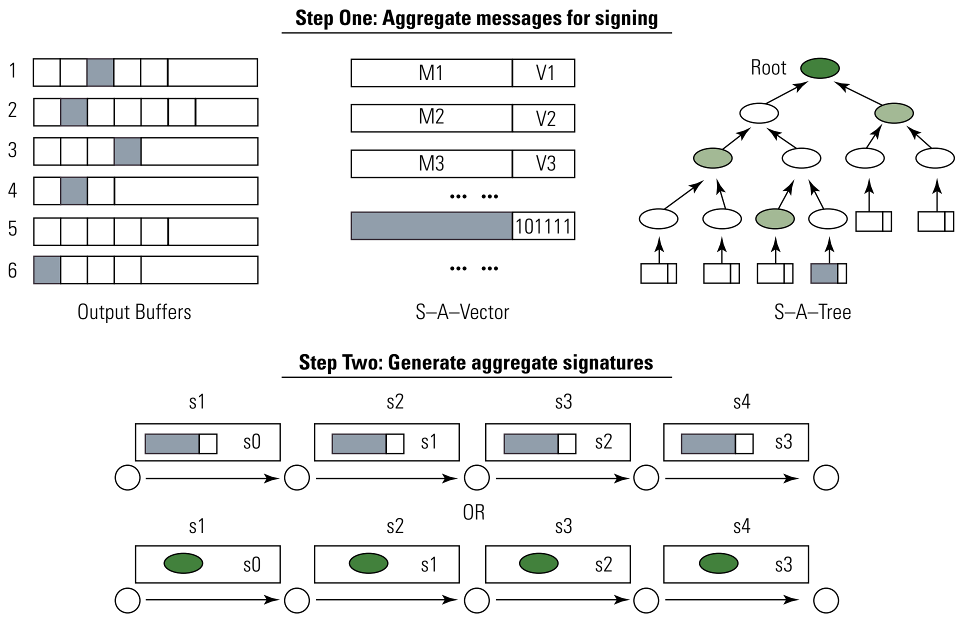


Figure 2 Sketch of APA schemes

all unsigned Update messages from these buffers. Then the router picks out the same route announcement to be sent to different neighbors and uses a bit vector to express which router's neighbors are the recipients. The aggregated message is the concatenation of the original route announcement and bit vector. The router signs this aggregated message to create an RA. The verifier of this RA uses the bit vector to check the intended receiver. To do this, the router needs to pre-establish an ordered list of its neighbors and distribute this list to potential verifiers. A router's public key certificate can be used for this purpose. Since such information is relatively stable, routers can use caching optimization with S-A-Vector to speed up the process. With the S-A-Tree scheme, the router constructs a hash tree, using the aggregated messages by S-A-Vector. The "leaves" in the "tree" are the hash values of these aggregated messages. Figure 2 illustrates the concept of constructing a hash tree. Now the root of the tree is the new aggregated value to be signed. The bit vector and a sequence of hash values (shown as light green nodes in Figure 2) shall be sent, together with the Update message, to permit the verifier to reconstruct the root. Details of the S-A scheme are given by David M. Nicol *et al.* in "Evaluation of Efficient Security for BGP Route Announcements using Parallel Simulation." [4]

In Step Two, the router applies the `AggrSign` algorithm to generate the aggregate signature using the aggregate value constructed locally by S-A-Vector or S-A-Tree and the aggregate signature from the previous hop. There are two choices for implementing the `AggrSign` algorithm, the General Aggregate Signature (GAS) scheme [9], an algorithm based on elliptic curves, and the Sequential Aggregate Signature (SAS) scheme proposed by Anna Lysyanskaya *et al.* in "Sequential Aggregate Signatures from Trapdoor Permutations." [10] It is constructed on a variation of the RSA algorithm. Both GAS and SAS are functionally equivalent for APA. Their difference is mainly in performance overhead. Finally, the `AggrVerify` algorithm permits validation of the route announcement, using the aggregate signature, messages by each hop, and the public key of each signer. The most important algorithm involved in `AggrVerify` for GAS is the pairing calculation. Currently, it can be implemented using software or dedicated hardware, which have different computational requirements. We denote them as GAS(SW) and GAS(HW).

To construct aggregate signatures for RAs, one uses either S-A-Vector or S-A-Tree in Step One and GAS or SAS in Step Two. Thus, there are four combinations and six actual implementations (considering software or hardware implementation). We denote them as GAS-V(HW), GAS-V(SW), GAS-T(HW), GAS-T(SW), SAS-V, and SAS-T, respectively. Details of each implementation are presented by Meiyuan Zhao *et al.* in "Aggregated Path Authentication for Efficient BGP Security." [3]

Performance Advantages

We use network simulation to compare the performance of APA schemes with S-BGP route attestations and with S-BGP, using all optimizations [denoted as S-BGP(CP)]. We implement simulation models of these schemes using the Scalable Simulation Framework Network (SSFNet) [11], a powerful simulator for large-scale networks. The processing and memory overheads by each cryptographic algorithm are modeled using benchmarks, as shown in Table 1.

	1024-bit RSA	1024-bit DSA	SAS	GAS
Sign (*ms*)	50.0	25.5	50.0	11.0
Verification (*ms*)	2.5	31.0	2.5	43.0 × k
SW Aggregate Verification (*ms*)	–	–	2.5 × k	43.0 × (k + 1)
HW Aggregate Verification (*ms*)	–	–	–	1.3 × (k + 1)
Aggregate Sign (*ms*)	–	–	50.0	11.0
Signature Length (bytes)	128	40	128	20

Table 1 This table, adapted from Meiyuan Zhao *et al.* in "Aggregated Path Authentication for Efficient BGP Security," [3] demonstrates the benchmarks of signature algorithms with the same level of security. Running times are normalized to 200 MHz CPU, a typical type of processor by edge BGP routers on the Internet, except hardware implementation of aggregate verification. We assume that aggregate verification handles *k* distinct messages.

We use the same BGP activity for evaluation—the router rebooting process. The workload on routers could be much higher than normal BGP activities. When re-establishing BGP sessions with its neighbors, the rebooting BGP router receives routing table dumps from each of its neighbors in a short period of time *via* a large number of route announcements. We use this case to understand the impact by security on BGP when routers are under pressure—as has been observed, for instance, during worm attacks.

We evaluate performance in terms of time and space. The following are the metrics:

- **Convergence Time**—We measure the reboot convergence time; *i.e.*, the time between when a crashed router returns to life and all the changes that percolate through the network. Convergence time is a good measure of routing stability. We compare the resulting convergence time with original BGP without security.
- **Message Size**—We measure the bytes for basic Update message fields and bytes for additional signatures, bit vectors, and hash values.
- **Memory Cost**—We also use experiments to report memory cost for route announcements, signatures, and bit vectors.

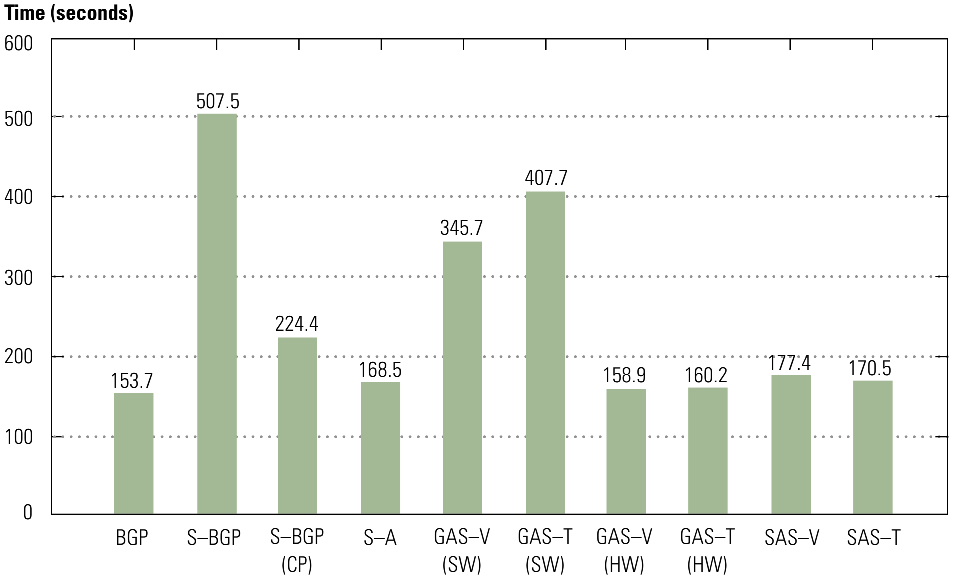


Figure 3 This figure, adapted from Meiyuan Zhao *et al.* in "Aggregated Path Authentication for Efficient BGP Security," [3] presents the comparison of convergence time.

Fast Convergence

Most APA schemes achieve very fast convergence time. The improvements come from the extreme savings by S-A techniques. APA schemes can save up to 98% of signing operations by S-BGP. The convergence time comparison is shown in Figure 3. In these experiments, original BGP requires at least 153 sec to converge. APA schemes increase this number by only a few seconds. On the other hand, S-BGP significantly lengthens convergence time, even using all optimizations. Recall that we measure the convergence time during the router rebooting process. We can conclude that APA schemes have minimal impact on BGP convergence, even when routers are under pressure.

Shorter Messages

Because of the techniques of aggregate signature, APA schemes generate shorter messages compared with S-BGP. In particular, GAS-V generates very short aggregate signatures (20 bytes), thus successfully shortening S-BGP messages by up to 66%. Figure 4 shows the experimental results. Notice that tree-based schemes actually increase message size, owing to extra hash values carried in messages.

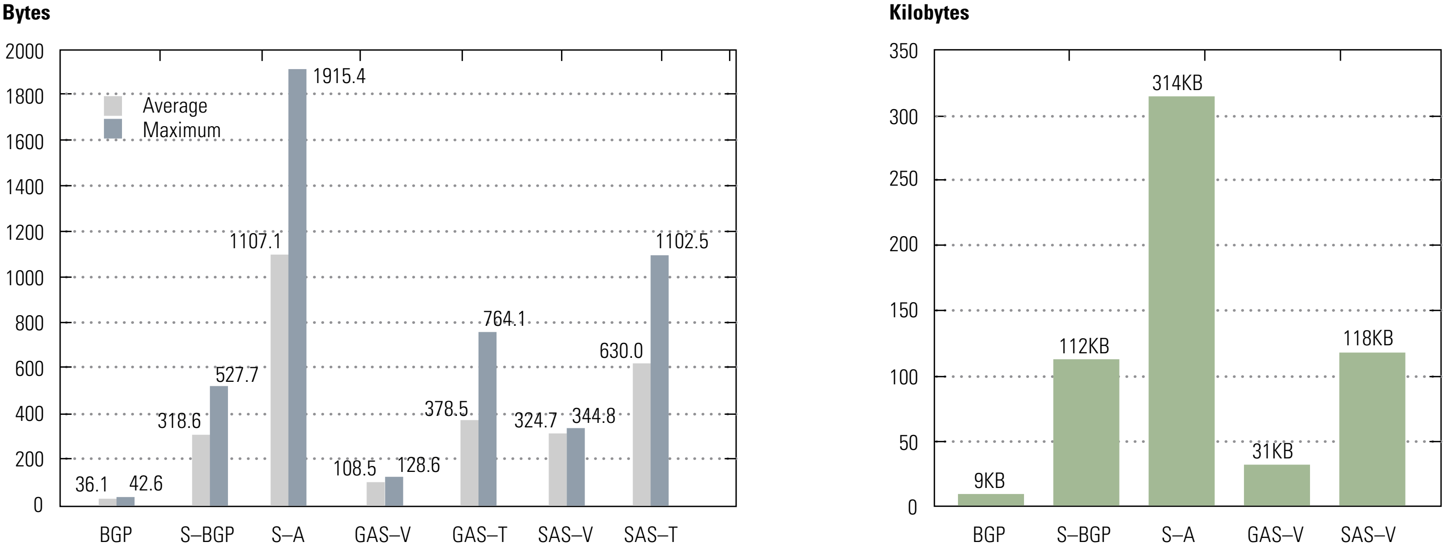


Figure 4 This figure, adapted from Meiyuan Zhao *et al.* in "Aggregated Path Authentication for Efficient BGP Security," [3] presents the comparison of message size and memory consumption.

Less Memory Overhead

Shown in Figure 4, GAS-V consumes significantly less memory on caching routes and RAs. It reduces by more than 70% the amount used by S-BGP. Once again, this saving is from short signatures by GAS algorithm.

Our experiments indicate that GAS-V is not just efficient—it helps reduce the extra memory costs needed for BGP path authentication. Using the BGP data collected from real routing tables, we analyze the memory usage by S-BGP and APA schemes. S-BGP can add more than 180% to the BGP routing table size, while GAS-V increases this number by only 52%. Taking into account other data structures, such as AAs and certificates, the overall savings by GAS-V as compared to all memory overheads by S-BGP can be 60% (*ibid.*, [3] for detailed discussion).

Conclusions

The S-BGP proposal provides comprehensive security countermeasures to authenticate routing information propagated by BGP routers. However, route attestations are expensive. The performance overheads—particularly memory overheads—are cited as contributing to the resistance to S-BGP's practical deployment.

We propose APA schemes to reduce the processing and memory overheads of S-BGP route attestations. We combine the efforts by S-A and aggregate signatures and construct six APA schemes. Performance evaluation using simulation has shown that the GAS-V scheme using hardware implementation of pairing calculation achieves the best performance. Our experiments suggest that GAS-V is an efficient and practical solution for BGP security. ■

Acknowledgment

The authors are grateful to Steve Kent, Russ Housley, B. J. Premore, Hongsuda Tangmunarunkit, and anonymous reviewers for their valuable suggestions. This research has been supported in part by Sun Microsystems, Inc; Cisco Systems, Inc; The Andrew W. Mellon Foundation; the National Science Foundation (NSF) (CCR-0209144, EIA-9802068); AT&T/Internet2; and the Office for Domestic Preparedness, Department of Homeland Security (2000-DT-CX-K001). This article does not necessarily reflect the views of the sponsors.

References

1. Y. Rekhter & T. Li. "A Border Gateway Protocol 4 (BGP-4)," RFC1771, March 1995.
2. Stephen Kent, Charles Lynn & Karen Seo, "Secure Border Gateway Protocol," *IEEE Journal of Selected Areas in Communications,* 18(4):582–592, April 2000.
3. Meiyuan Zhao, Sean W. Smith & David Nicol, "Aggregated Path Authentication for Efficient BGP Security," *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS 2005)*, pages 128–138, November 2005.
4. David M. Nicol, Sean W. Smith & Meiyuan Zhao, "Evaluation of Efficient Security for BGP Route Announcements using Parallel Simulation," *Simulation Practice and Theory Journal, special issue on Modeling and Simulation of Distributed Systems and Networks*, 12(3–4):187–216, July 2004.
5. S. Murphy, "BGP Security Vulnerabilities Analysis," *Internet-Draft, draft-murphy-bgp-vuln-01*.txt, October 2004.
6. Meiyuan Zhao, Sean W. Smith & David M. Nicol, "Evaluating the Performance Impact of PKI on BGP Security," *4th Annual PKI R&D Workshop*, Gaithersburg, MD, April 2005.
7. Steve Kent, "Securing the Border Gateway Protocol: A Status Update," *Seventh IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, October 2003.
8. Dan Boneh, Craig Gentry, Ben Lynn & Hovav Shacham, "A Survey of Two Signature Aggregation Techniques," *RSA CryptoBytes*, 6(2):1–10, 2003.
9. Dan Boneh, Craig Gentry, Ben Lynn & Hovav Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," *Proceedings of Eurocrypt 2003*, number 2656 in LNCS, pages 416–432. Springer-Verlag, 2003.
10. Anna Lysyanskaya, Silvio Micali, Leonid Reyzin & Hovav Shacham, "Sequential Aggregate Signatures from Trapdoor Permutations," *Eurocrypt 2004*, number 3027 in LNCS, pages 74–90, Springer-Verlag, 2004.
11. Andy T. Ogielski & James H. Cowie,"SSFNet: Scalable Simulation Framework Network Models," *http://www.ssfnet.org.*

About the Authors

Meiyuan Zhao | is a researcher working in the Communications Technology Lab at Intel Corporation. She is responsible for developing and proliferating Intel's guidelines to ensure secure networked communications. She received a BE degree in Computer Science and Engineering from the University of Electronic Science and Technology of China in 1998 and a PhD in Computer Science from Dartmouth College in 2005. Her research interests are in network security, cryptography, routing, and simulation and modeling.

Sean Smith | is a faculty member of the Department of Computer Science at Dartmouth College. His current research focuses on how to build trustworthy systems in the real world. He previously worked as a scientist at IBM's T.J. Watson Research Center, where he developed secure co-processor design, implementation, and validation; and at Los Alamos National Laboratory, where he developed security designs and analyses for a wide range of public-sector clients. Dr. Smith received a BA degree in Mathematics from Princeton and MS and PhD degrees in Computer Science from Carnegie Mellon University.

David Nicol | is Professor of Electrical and Computer Engineering at the University of Illinois, Urbana-Champaign, and a member of the Coordinated Sciences Laboratory. He received a BA degree in Mathematics from Carleton College in 1979 and MS (1983) and PhD degrees in Computer Science in 1983 and 1985, respectively, from the University of Virginia. His research interests are in high-performance computing, performance analysis, simulation and modeling, and network security. He is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE) and of the Association for Computing Machinery (ACM).

7th Annual IEEE Information Assurance Workshop (IAW)

The 7th annual Institute of Electrical and Electronics Engineers (IEEE) Information Assurance Workshop (IAW) was held 21–23 June 2006 at the Thayer Hotel at the US Military Academy, West Point, NY. This conference, sponsored by IEEE Systems, Man, Cybernetics (SMC), and the National Security Agency (NSA), showcased academic institutions and researchers on Information Assurance (IA) from all over the globe who presented papers on cutting-edge technologies and ground-breaking tactics.

Some institutions of higher learning that participated in the workshop included, but were not limited to, the Naval Post Graduate School, State University of New York at Buffalo, James Madison University, Carnegie Mellon University, Mississippi State University, and the University of Idaho. IATAC reviews IA technologies, such as those presented at this conference, to explore emerging technologies several years before they are made commercially available.

Topics included papers on Data Protection, Privacy, Visualization, Honeynets, Wireless Security, and Information Warfare. Among the top papers were the following:

- *Design and Implementation of Fire Transfer and Web Services Guard Employing Cryptographically Secured XML Security Labels*
- *Quantitative Analysis of Efficient Antispam Techniques*
- *Foundations for Visual Forensic Analysis*
- *A Dynamic Filtering Technique for Sebek System Monitor*
- *Effects of Denial of Sleep Attacks on Wireless Sensor Network MAC Protocol*
- *Investigating the Effect of an Attack on a Distributed Database*

The award for Best Paper was presented to Alexander Volynkin, Victor Skormin, Douglas Summerville, and James Moronski of the Air Force Office of Scientific Research for *Evaluation of Run-Time Detection of Self-Replication in Binary Executable Malware*. This paper describes how to detect self-replicating malware, known or unknown, based not on signature but on the malware's activity.

Dr. Frederick R. Chang, Research Directory, NSA, delivered the keynote address on NSA's current research activities. Wendy Seltzer, Professor, Brooklyn Law School, spoke about legal issues and privacy in IA. Glenn Schoonover, Executive Director, Worldwide Technical Strategy–National Security, Defense and Public Safety, at Microsoft, gave an overview of Microsoft's new operating system, Windows Vista, from a security standpoint. The last speaker, Bruce Potter of The Schmoo Group, spoke about Bluetooth and other wireless insecurities and defenses.

Proceedings from this year's workshop may be ordered from the IEEE Web site, *http://www.ieee.org*, and information about next year's conference is available at *http://www.itoc.usma.edu/workshop.* ■

Significant New Developments in Cyberlaw

by Richard Aldrich

As cyberspace has increasingly become the situs of choice for myriad life activities, including commerce, journalism, crime, and even warfare, the law has followed. The law's incursion into these areas has not always been smooth. Judges have tried to analogize events in cyberspace with those in the physical world with mixed results. Legislators have sometimes passed laws that were so technologically bound that they became obsolete as soon as the technology changed. Nevertheless, the law has continued to adapt and mature. This article addresses some recent developments in cyberlaw that are of special significance to those who work in and with the Department of Defense (DoD).

International Treaties

One significant development is the US Senate's ratification of the Cybercrime Convention. [1] This international treaty was originally signed by the US in 2001. On 17 November 2003, President Bush referred the treaty to the Senate for its "advise and consent" process, as required by the Constitution. President Bush urged the Senate to ratify it at that time, but the treaty did not emerge from the Senate Foreign Relations Committee for another year and a half and then was not ratified by the Senate until 3 August 2006. The treaty had already become effective on 1 July 2004, shortly after the fifth country ratified it, and it has since had been ratified by 15 other countries in addition to the United States. These countries are Albania, Bosnia-Herzegovina, Bulgaria, Croatia, Cyprus, Denmark, Estonia, France, Hungary, Lithuania, Norway, Romania, Slovenia, the former Yugoslavian Republic of Macedonia, and Ukraine. [2] There are also 27 other countries that have signed the treaty but not yet ratified it, including Japan, Canada, and most European countries. Of course, hackers can easily operate from virtually any country; to thwart investigators, hackers frequently hop through various countries en route to their target. Because of this activity, there is a thrust to broaden the number of states that are party to the treaty.

At an international conference entitled, *Cybercrime: A Global Challenge, A Global Response*, held in Madrid, Spain, last December, members of the Organization of American States were provided information to familiarize them with the process of accession to the Convention on Cybercrime. The treaty could prove useful to military criminal investigators in various ways. Perhaps most importantly, the treaty contains a mutual legal-assistance provision that will greatly assist in receiving the prompt, effective investigative assistance so crucial to cybercrime investigations. The treaty also requires states that are party to the treaty to ensure that their domestic laws cover a wide spectrum of computer offenses. This could greatly assist in ensuring cybercriminals are either prosecuted abroad or extradited to the US and will avoid the problems encountered in the "I Love You" virus case. In that case, even after investigators identified the alleged virus writer in the Philippines, the case died because the Philippines had no law proscribing the conduct and could neither prosecute nor extradite the individual. (Extradition treaties generally have a dual-criminality requirement, which mandates that the offense for which the individual is being extradited be a felony in both countries.) The hope is that, as more countries accede to the treaty by modernizing their laws and improving their investigative cooperation with other states, hackers will have fewer places to hide.

Domestic Law

Domestically, perhaps the most significant development for cybercrime investigators has been the re-authorization of the USA PATRIOT Act. [3] Sixteen provisions were scheduled to sunset on 31 December 2005. Re-authorizing these provisions became a very contentious process, as debate intensified over the proper balance between privacy and security. The provisions were granted several temporary extensions while legislators continued to debate, but ultimately, in two separate bills [4] passed in March 2006, 14 of the 16 sunset provisions [5] were permanently re-authorized, including the following:

Sec. 201	Electronic Communications Privacy Act (ECPA)—Wiretapping in certain terrorism investigations
Sec. 202	ECPA—Wiretapping in computer fraud and abuse investigations
Sec. 203(b)	Law enforcement sharing of court-ordered wiretap-generated foreign intelligence information wiretap information
Sec. 203(d)	Law enforcement sharing of foreign intelligence information notwithstanding any other legal restriction
Sec. 204	Technical exception for foreign intelligence pen register/trap and trace device use
Sec. 207	Duration of Foreign Intelligence Surveillance Act (FISA) wiretap and search orders involving agents of a foreign power
Sec. 209	Seizure of stored voice mail by warrant rather than ECPA order
Sec. 212	Communications providers emergency disclosures of communications content or related records to authorities
Sec. 214	FISA pen register order amendments, including extension to electronic communications; *e.g.*, Internet use
Sec. 217	Law enforcement access to computer trespassers' communications within the intruded system
Sec. 218	FISA wiretap or search orders with an accompanying law enforcement purpose (removal of "the wall" of separation between criminal catchers and spy catchers)
Sec. 220	Nationwide service of court orders directed to communication providers
Sec. 223	Civil liability and disciplinary action for certain ECPA or FISA violations
Sec. 225	Civil immunity for assistance in executing a FISA order

The remaining two provisions were re-authorized with a new sunset date of 31 December 2009. These provisions cover "roving" wiretaps under the FISA and the "lone wolf" amendment to FISA. "Roving" wiretaps permit investigators to wiretap a target regardless of which phone he or she uses. The "lone wolf" provision removed the "agent of a foreign power" requirement for conducting electronic surveillance or searches under FISA, thereby permitting the targeting of terrorists who worked independent of any foreign power.

Another significant domestic development is the proliferation of bills in Congress addressing the protection of Personally Identifiable Information (PII). These bills received added emphasis after the loss of PII on some 26M veterans by an employee of the Department of Veteran Affairs. It is unclear at this time which, if any, bill will pass in the current legislative session. Even if none pass, however, DoD will soon issue an interim, directive-type memorandum that will establish new policy for PII, both to protect it and report its loss.

Foreign Law

Across the Atlantic, the European Parliament issued a data retention directive [6] at the end of 2005 that could be very significant in combating cybercrime. The directive requires that telephone data be retained for at least 24 months and that e-mail and Internet Protocol (IP) data be retained for 6 to 24 months. Access to the data would generally be limited to investigations of "serious crimes," of which some 32 are listed on the European Arrest Warrant. The directive must be implemented by each state in the European Union. This could take time, and at least one state has expressed concern about the legality of implementing the directive. If the directive were enacted across the European Union, it would permit investigators to access data critical to most investigations—data that is often quickly destroyed under the current legal structure.

Case Law

In the courts, there were several interesting new developments. The case of *United States v. Long* [7] is of special interest because it challenges the efficacy of DoD's consent banner. In this case, Marine LCpl Long was charged with various drug offenses. The evidence against her consisted of eyewitness testimony and some incriminating e-mails she had sent to friends. Investigators requested that the senior network administrator retrieve Long's e-mails from the government server. At trial, Long moved to suppress her e-mails as the result of unreasonable search and seizure. The military judge denied the motion to suppress. As most in DoD are no doubt aware, every time a DoD computer is booted up, the following banner appears:

> *This is a Department of Defense computer system. This computer system, including all related equipment, networks and network devices (specifically including Internet access), are provided only for authorized U.S. Government use. DoD computer systems may be monitored for all lawful purposes, including to ensure that their use is authorized, to manage the system, to facilitate protection against unauthorized access, and to verify security procedures, survivability, and operational security. Monitoring includes active attacks by authorized DoD entities to test or verify the security of this system. During monitoring, information may be examined, recorded, copied, and used for authorized purposes. All information, including personal information, placed on or sent over this system may be monitored. Use of this DoD computer system, authorized or unauthorized, constitutes consent to monitoring of this system. Unauthorized use may subject you to criminal prosecution. Evidence of unauthorized use collected during monitoring may be used for administrative, criminal or other adverse action. Use of this system constitutes consent to monitoring for these purposes.* [8]

The intent of the banner was to obtain consent, either expressly, by one's clicking "OK" or "I agree," or by implication, by one's continued use of the computer, after being notified that such use was subject to monitoring. The banners, together with regulatory provisions, briefings, and user training about the monitoring of DoD computer usage was also intended to reduce one's subjective expectation of privacy and, by so alerting enough people over a period of time, to also make the expectation of privacy objectively unreasonable.

Despite this, the Navy-Marine Corps Court of Criminal Appeals (N-MCCA) held that LCpl Long had a subjective expectation of privacy in using her government PC and in the e-mails stored on the government server *vis-à-vis* the law enforcement investigators (but not necessarily with respect to the system administrators). The court also held that such expectation was objectively reasonable but found that the remaining evidence of Long's guilt was sufficient to render this error harmless. The case was then appealed to the military's highest court, the Court of Appeals for the Armed Forces (CAAF).

That CAAF issued its decision on 27 September 2006. In a 4–1 decision authored by Judge Gierke, with Judge Crawford dissenting, the court upheld the N-MCCA's finding that Long had a subjective expectation of privacy and that that expectation was objectively reasonable. The court also held the error was not harmless, since the trial counsel made repeated references to the importance of Long's e-mails during his closing argument and so set aside her conviction. One key factor in the court's rationale was the fact that Long had a user-defined password for accessing her e-mail, which was unknown to the system administrator. (The court seemed to ignore the fact that the system administrator did not need a user's password to read the user's e-mails.) The government argued that the password was for government computer security, and that the DoD Notice and Consent banner should have undermined the reasonableness of any subjective expectation of privacy that Long had. The court rebuffed those arguments, however, claiming that, in spite of the banner, the network administrator testified at trial that there was a "privacy issue" relating to monitoring e-mails. As such, the court held that under all the facts and circumstances of this case, Long did have a subjective expectation of privacy and that that expectation was objectively reasonable.

It was unclear before the publication deadline for this article whether the Navy would further appeal the case to the US Supreme Court. Because the CAAF rested its decision in part on a *military-specific* reading of *O'Connor v. Ortega* [9], it seems unlikely the Supreme Court would grant review, since it normally chooses to grant review only on issues of broad national interest.

DoD has already rewritten its Notice and Consent banner to specifically respond to the court's concerns, and by the time this article is published, it will likely already be issued as new policy. Even without any new banner, however, the court did nothing to strike down the "system provider" exception to the Federal Wiretap Act, which permits system providers to intercept, disclose, or use communications to protect the rights and property of the provider. It also did not address DoD information system user agreements, required by CJCSM [10] 6510.01, which, if well written, could separately undermine the reasonableness of one's expectation of privacy or even grant explicit consent to searches. It held only that under the specific facts of this case a user of a DoD computer could have a reasonable expectation of privacy.

Several other cases are also of interest. In *United States v. Conklin* [11], the CAAF was asked to decide whether one could have a reasonable expectation of privacy in a *personally owned* computer that was kept in an on-base dormitory room shared with another person and subject to routine room inspections. The court held that one could have a reasonable expectation of privacy under such conditions.

In *United States v. Romm,* [12] the Ninth Circuit Court of Appeals held that Customs agents can search a computer as part of a border search without any warrant or probable cause under the border-search exception to the Fourth Amendment. This also applies to international points of entry, such as the Seattle-Tacoma airport, as applies in this case. The court also held that one could be found guilty of "possessing" child porn even if one never explicitly saved the files to one's hard drive or removable media but only "viewed" the files through one's browser, causing them to be automatically stored in the browser's temporary cache.

In *United States v. Martinelli,* [13] the CAAF held that the federal Child Pornography Prevention Act (CPPA) [14] did not apply extra-territorially and therefore could not be used to convict a soldier who otherwise violated its provisions off post in Germany. It ruled, however, that "sending" child pornography was a "continuing offense," which extended from the time of sending until receipt, and therefore sending child pornography from Germany through servers in the United States was a domestic offense not affected by the lack of extra-territoriality. (For other reasons, however, the court also overturned that conviction. Lesser offenses included under Article 134 of the Uniform Code of Military Justice, [15] the General Article, which would not have required assimilating the CPPA, could not be upheld, as the judge failed to cover all of the elements of such offenses during the providency inquiry into the CPPA offense.) With cyberspace law still changing, evolving, and maturing, practitioners of all sorts—whether cybercrime investigators, Information Assurance (IA) professionals, Information Operations (IO) warriors, or lawyers who counsel these personnel—would be well advised to stay abreast of new developments in the field. It can help you take full advantage of the many beneficial provisions in the new laws while avoiding legal pitfalls. ■

References

1. Council of Europe Treaty Series No. 185.
2. See *http://conventions.coe.int/Treaty/Commun/ChercheSig.asp?NT=185&CM=&DF=&CL=ENG* for an up-to-date listing of signatories. The United States was not yet listed as having ratified the treaty as of the writing of this article because of administrative filings still to be made with the treaty body.
3. Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism Act (USA PATRIOT Act) of 2001, P.L. 107–56, 115 Stat. 272 (2001).
4. USA PATRIOT Improvement & Reauthorization Act (P.L. 109–177) and USA PATRIOT Additional Reauthorizing Amendments Act (P.L. 109–178), both signed into law by President Bush on March 9, 2006.
5. The list was obtained from "USA PATRIOT Improvement and Reauthorization Act of 2005: A Legal Analysis," Congressional Research Service, March 24, 2006.
6. Copy available at *http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//NONSGML+TA+P6-TA 20050512+0+DOC+WORD+V0//EN&language=EN*
7. 61 M.J. 539 (N-M. Ct. Crim. App. 2005)
8. The source for this language is a DoD General Counsel Memorandum, "Communication Security and Information Systems Monitoring," March 17, 1997.
9. See, *e.g.*, CJCSM 6510.01, Defense-In-Depth: Information Assurance (IA) and Computer Network Defense (CND).
10. 63 M.J. 333 (CAAF 2006).
11. No. 04-10648 (9th Cir. Jul. 24, 2006).
12. 62 M.J. 52 (CAAF 2005).
13. 18 U.S.C. 2252A.
14. 10 U.S.C. 934.

About the Author

Rick Aldrich | is Senior Computer Network Operations Policy Analyst for IATAC. He previously served as the Deputy Staff Judge Advocate for the Air Force Office of Special Investigations, specializing in the cybercrime and information operations portfolios. He has multiple publications in this field, including most recently a chapter on information warfare in a national textbook on National Security Law. He is a co-author of DoD's award-winning CyberLaw 1 computer-based training product and the soon-to-be-released CyberLaw 2 product. He has presented at several national and international conferences. He holds a BS degree in Computer Science from the Air Force Academy, a JD degree from the University of California Los Angeles, an LLM degree in Intellectual Property Law from the University of Houston.

ESSG Corner

by John Palumbo

In a previous *IAnewsletter* (Vol 8 No 3, Winter 2005/2006), Mr. Wise explained the background of Department of Defense (DoD) Enterprise-wide Information Assurance/Computer Network Defense (IA/CND) Solutions Steering Group (ESSG) and its efforts. In this entry of what we hope will be a continuing series, I will attempt to update the information to let you know of changes within the ESSG organizational structure, provide you with some high-level details of the ESSG's activities, and offer a brief overview of upcoming actions.

Organizational Structure

As you may remember, the ESSG holds quarterly meetings. Our last meeting was held 12–14 September 2006 in Omaha, NE. This marks the third meeting since adding a new co-chair, the Joint Task Force-Global Network Operations (JTF-GNO). The JTF-GNO provides a much-needed operational perspective into the mix of IA/CND professionals, ensuring that the ESSG keeps an eye on the end goal of supporting the operator in defending the Global Information Grid (GIG). A new Project Management Office has also been added through Mr. Montemarano's Program Executive Office (PEO) IA/NetOps in the Defense Information Systems Agency (DISA). This will permit greater integration of the ESSG's selected tools across all IA initiatives, while permitting the ESSG to reach deep into the DISA Core Programs. The PEO IA/NetOps will also co-chair a Configuration Control Board (CCB) with the DISA Field Support Office (FSO). The CCB will be charged with managing any changes to products that the ESSG has purchased for enterprise-wide use.

High-Level Updates

The ESSG continues to deploy both the Secure Configuration Compliance Validation Initiative (SCCVI) and Secure Configuration Remediation Initiative (SCRI) tools, jointly known as the Secure Configuration Tool Suite (SCTS). The tools are continually updated: the SCRI package has just recently been released in version 4.0. The other item of significant interest is the Host-Based Security System (HBSS) that continues in the pilot stage. Training for the HBSS can be found on the Information Assurance Support Environment's Web site, *http//iase.disa.mil*. The anticipated deployment date for the HBSS tool is December 2006. The HBSS' primary function will be to provide security applications to the host level, with the possibility of additional capabilities to be applied as modules. The ESSG also voted to adopt a Government-Off-The-Shelf (GOTS) solution for the Wireless Detection requirement. The Navy's "Flying Squirrel" will be used as the enterprise solution, which will allow all Combatant Commands, Services, and Agencies access to the same tool suite at little to no cost.

In the next *IAnewsletter*, I will not only provide an update on the most current happenings, but also explain in greater detail how the ESSG takes a requirement and converts it into an enterprise tool, including the major steps each Sub Working Group takes to contribute to this effort.

Upcoming Actions

The ESSG's next quarterly meeting will be sponsored by US Southern Command (USSOUTHCOM) on 12–14 December 2006. The meeting will cover

the standard ESSG format and include information on the HBSS pilot progress and spiral one of the User-Defined Operational Picture (UDOP) plus updates on Insider Threat Tools and the Tier III Security Integration Manager (SIM). The ESSG's Sub Working Groups will also continue their respective activities. The Technical Advisory Group (TAG) continues its monthly meetings to focus on refining both requirements and technical solutions. The Acquisition Working Group (AWG) will continue implementing funding to move solutions into the hands of forces helping to protect the GIG. The Concept of Operations (CONOPS) Working Group (CWG) will also be making progress in refining the CONOPS for tools in pilot stages and for those being fielded. Lastly, the CND Architecture Working Group (CAWG) continues to make progress in both documenting the architecture and leading the effort for Data Strategy that will permit all employed enterprise tools to communicate.

In the next *IAnewsletter*, I will not only provide an update on the most current happenings, but also explain in greater detail how the ESSG takes a requirement and converts it into an enterprise tool, including the major steps each Sub Working Group takes to contribute to this effort.

If you have questions about the ESSG or any of its Sub Working Groups, feel free to contact me in Omaha, NE, at 402/294-5890 or *via* e-mail at john.palumbo@stratcom.mil. ■

About the Author

John Palumbo | currently acts as coordinator for the ESSG. For the past 10 years, he has supported both the United States Strategic Command (USSTRATCOM) and the United States Space Command (USSPACECOM) as an IA and Information Operations (IO) professional, both as a US Navy Officer and as a contractor. He earned his Certified Information Systems Security Professional (CISSP) certification in 2002 and holds an MS degree in Information Technology Management from the Naval Postgraduate School, Monterey, CA.

State University of New York at Stony Brook (SUNY-SB) Center for CyberSecurity

The Center for CyberSecurity promotes research and education in computer, network, and information security, fostering research collaborations between Information Assurance (IA) and computer science researchers into security aspects of database systems, operating systems, programming languages, and formal methods and verification.

The Center is funded by the National Science Foundation (NSF), the National Institutes of Standards and Technology (NIST), the Defense Advanced Research Projects Agency (DARPA), the Office of Naval Research (ONR), Army Research (AR), the Air Force Office of Scientific Research (AFOSR), and Computer Associates (CA). An NSA Information Assurance Center of Excellence, the Center seeks comparable partnerships in the private sector.

Under the direction of Dr. R. Sekar, the Center comprises six research labs:

- **Secure Systems Laboratory (SSL)**—Researches high reliability and security for networks and software systems, with specific projects in network- and software-based attack and intrusion detection and confinement, and defect-detecting software analysis and debugging tools for complex systems.
- **Applied Logic Laboratory (ALL)**—Researches principles and use of logic-based methods for databases, concurrent system verification, data mining, and Web information systems. The lab has three major projects:
  - FLORA, a programming language for knowledge-intensive applications,
  - Logic Programming-Based Model Checking (LMC), and
  - XSB, a high-performance logic programming and deductive database system.

The lab also has several smaller projects in data mining, agent-based systems, and related technologies.

- **Concurrency and Verification Laboratory (CVL)**—Researches and creates integrated toolsets for specifying, simulating, verifying, and implementing concurrent systems (*e.g.*, communication protocols, process control systems). Projects include the following:
  - Concurrency Workbench of the New Century (CWB-NC) for specifying and verifying finite-state concurrent systems,

> SUNY-SB was designated a Center of Excellence in Wireless and Information Technology (IT) by the State of New York. The school received $20M of a total $230M in state grants awarded to a group of Long Island academic and industry institutions and intended to fund collaborative research into several aspects of wireless and IT systems, including cyber security. On raising $20M of matching funds, SUNY-SB will receive another $10M from the state.

- Process Algebra Compiler, the Concurrency Factory,
- Probabilistic Input/Output Automata (PIOA) PIOATool suite, and
- Modeling and verification of the Rether real-time ethernet protocol and the Java virtual machine meta-locking algorithm.

▶ **Design and Analysis Research Laboratory (DARL)**—Develops methods and tools for modeling, specifying, analyzing, verifying, designing, optimizing, generating code, and testing for the construction of reliable, efficient reactive systems, embedded systems, database applications, and information-retrieval systems.

▶ **Experimental Computer Systems Laboratory (ECSL)**—*This lab is described in a companion article in this issue on Dr. Tzi-cker Chiueh and the ECSL. Dr. Chiueh is a Subject Matter Expert in software security and vulnerability detection and prevention whose research is driven by a single key impetus:* How to dynamically detect, rather than statically locate, the software bugs that are likely to manifest as security vulnerabilities.

▶ **File-systems and Storage Laboratory (FSL)**—Researches security aspects of operating systems, file systems, storage, and networking, with emphasis on the following:

- Balancing security, performance, and usability,
- Improving portability of operating system code, and
- Improving programmer and administrator productivity.

Projects include the following:

▶ FiST, a stackable file system language,

▶ Compound System Calls (CoSy), a method of aggregating multiple system calls into a single call,

▶ A next-generation cryptographic file system (NCryptfs),

▶ Elastic Quotas, a disk-space management method, and

▶ Versionfs, a stackable versioning file system.

Also affiliated with the Center is the Network Security and Applied Cryptography Lab (NSACL), which researches cryptography and its application to real-world problems such as secure data outsourcing, wireless and sensor network security, querying and searching encrypted data, security and privacy for networked storage, security policies for computational and data grids, Digital Rights Management, and secure reputation systems.

The Center for CyberSecurity does not offer scholarships, but all staff positions are paid research assistantships. Currently, 50 PhDs and 50 Masters degree candidates staff the Center. ■

References:

1. SUNY-Stony Brook Center for Cybersecurity (CCS) *http://ccs.cs.sunysb.edu/*
2. Center of Excellence in Wireless and Information Technology *http://www.cewit.org*

Dr. Tzi-cker Chiueh, Director

Experimental Computer Systems Laboratory State University of New York at Stony Brook

The State University of New York at Stony Brook (SUNY-SB) is a research leader in information, network, and software security. One faculty member who leads this research is Dr. Tzi-cker Chiueh. His research in software security and vulnerability detection and prevention is driven by a single key impetus: *How to dynamically detect, rather than statically locate, the software bugs that are likely to manifest as security vulnerabilities.*

To achieve this goal, Dr. Chiueh is pursuing a fundamental solution to the software security problems associated with C and C++ programs. More specifically, Dr. Chiueh is working on a security error-correcting compiler that automatically prevents buffer overflow, integer overflow, and format string errors from turning into exploitable vulnerabilities. The compiler adds logic to programs that check various types of overflow errors at run time and then invokes appropriate exception handling when these errors occur. Compared with current overflow detection solutions, such as StackGuard, that also try to eradicate overflows in executing software, Dr. Chiueh's overflow prevention approach is more general. It prevents overflows from happening in the first place and is thus capable of detecting more attacks, such as Data attacks, than are existing technologies. The key innovation of Dr. Chiueh's compiler is its low performance overhead. For example, his compiler performs array and buffer bound checking for large programs, such as Apache Web server and Bind, with a performance overhead of less than 10%, the fastest result ever reported in the literature.

Dr. Chiueh and his team extend the same dynamic checking approach to the Web application security problem such as Standard Query Language (SQL) injection, Hypertext Preprocessor (PHP) injection, and cross-site scripting attacks. The key enabling technology for this line of research is a general information-flow tracking compiler that automatically tracks application-specific tags through distributed Web application programs that are written in interpretive languages such as PHP, Perl, and Java. Over the next two years, Dr. Chiueh hopes to produce a compiler that will correct these Web application security bugs in the same way as overflow errors. Another ambitious research project undertaken by Dr. Chiueh's team and sponsored by National Scientific Foundation (NSF) is automatic generation of attack signatures and patches. More concretely, Dr. Chiueh's group aims to develop the following:

- Techniques that will detect an attack and then automate the process of generating signatures that firewalls will use to block future instances of the detected attack, and
- Patches that will permanently fix the vulnerabilities being exploited.

Dr. Chiueh's research is mainly performed in SUNY-SB's ECSL, of which he is Director. Falling under the umbrella of the university's Center for CyberSecurity, the ECSL is staffed by one postdoctoral student, 15 PhDs, and six Masters degree students.

An additional goal of this project is to derive characterizing attack signatures for vulnerable programs as soon as their patches are published. This technology can be applied to both cyber defense and offense.

The team's work on behavior-blocking intrusion detection eschews the approach

used by many intrusion and anomaly-detection systems—computer learning to identify correct *vs.* anomalous behavior. Instead, the team uses source code to construct a control flow graph from which unnecessary system calls are stripped. The resulting system-call control flow graph accurately represents the system's correct behavior and thus enables the resulting Intrusion Detection System (IDS), PAID (Program-semantics Aware Intrusion Detection), to detect intrusions with zero false positive and close-to-zero false negative. The team's next challenge is to apply the same system-call graphing approach to Win32 binary executables.

Dr. Chiueh's research is mainly performed in SUNY-SB's Experimental Computer Systems Laboratory (ECSL), of which he is Director. Falling under the umbrella of the university's Center for CyberSecurity, the ECSL is staffed by one postdoctoral student, 15 PhDs, and six Masters degree students. ECSL has developed several technologies that are being commercialized.

One technology transfer example is Feather-weight Virtual Machine (FVM), which is designed to support the same level of state isolation as conventional VMs (*e.g.*, VMWare and Xen) but incur much less start-up overhead and resource requirement. For example, FVM could easily support up to dozens of FVM instances on a single-processor desktop PC, as opposed to a maximum of two to three using existing VM technologies. FVM enables an isolation approach to malware detection that detects zero-day attacks: For example, potentially malicious code—scripts contained in e-mail attachments that are Microsoft Word documents—runs in its own isolated VM. With this architecture, even if a user mistakenly opens a virus-containing e-mail attachment, the damage is confined within the corresponding VM and can never infect the host operating system or other VMs. This technology is being evaluated for use in a commercial embedded system.

Another example of successful technology transfer from the ECSL is Rether Networks' Display-Only File Server (DOFS), which is designed to prevent theft of confidential information by insiders. DOFS guarantees the following invariant: once a confidential file is checked into a DOFS server, the bits of this file never physically leave the server; however, users may still interact with this confidential file in the same way as if it were stored on their local machines. In essence, when a file is accessed, DOFS first determines whether this file is too sensitive to be physically moved to the requesting client machine. If so, DOFS transparently launches the file's corresponding application on the server and uses standard thin-client technology, such as Microsoft's Terminal Services, to permit a user to read the file and write to it. DOFS also includes a client-side component that prevents users from taking a screen dump of windows that display confidential files.

While this approach cannot prevent insiders from retyping a file's contents on their clients, Dr. Chiueh contends that by increasing the effort it takes to "steal" sensitive data, the DOFS creates a significant deterrent to many insider attacks. He notes that the transparency of the DOFS process avoids the problems of Digital Rights Management (DRM) solutions, which have had low market acceptance because they are too difficult to correctly configure and because they cannot stop information theft by authorized users. "Enterprises are not willing to do security at the expense of convenience and productivity," he observes. ■

Note: *Dr. Chiueh is co-author, with Yang Yu and Fanglu Guo, of the article entitled, "A Virtual Environment for Safe Vulnerability Assessment," published in this issue on page 8.*

References:

1. Tzi-cker Chiueh *http://www.cs.sunysb.edu/people/faculty/TzickerChiuehAB.html*
2. Experimental Computer Systems Lab *http://www.ecsl.cs.sunysb.edu*
3. Rether Networks Inc. Display-Only File Server (DOFS) *http://www.rether.com/DOFS.htm*

Digital Forensics Education at the Air Force Institute of Technology (AFIT)

by Gilbert Peterson, Richard Raines, and Rusty Baldwin

The Department of Electrical and Computer Engineering (AFIT/ENG) at the Air Force Institute of Technology (AFIT) offers a graduate-level introductory course in digital forensics, during which students are introduced and exposed to several challenges and topics in digital forensics. We address ethical and legal procedures and basic forensic-science principles in a general manner. A larger percentage of time is spent examining the technical details of media analysis, including proper media duplication, and methods for locating hidden information. The network-forensics and digital-device analysis topics begin to breach technical-level details but do not attempt to reach mastery. This course provides our students with a real-world experience of digital forensics to prepare them for the challenges they will face in post-graduate employment.

Introduction

The digital forensics course at AFIT is currently in its third year. Enrollment has steadily increased over this period from a handful of students to approximately twenty for this year's offering. The course is part of three MS degree programs: Information Assurance, Computer Science, and Computer Engineering. The course is tightly integrated with our other computer security courses. For example, the techniques that students learn build on the events that occur during the Cyber Defense Exercise (CDX), an event sponsored by the National Security Agency. During a week-long exercise, students administer a network and defend it against attacks. Specifically, students must determine what went wrong after a successful attack. One-fourth of the digital-forensics course is spent on live network response, which gives students exposure to the tools they will use when faced with these situations in the future.

This article introduces the structure of material for the digital forensic course, then discusses general course content using the laboratories and topic divisions, followed by items we have found that improve the course and improvements we will implement in the future.

AFIT's Digital Forensics Course

Our graduate course is offered within the Graduate School of Engineering and Management and can be used to partially fulfill the requirements for MS degrees in Computer Science, Computer Engineering, or Information Assurance. We specifically emphasize the technical details of digital forensics rather than legal, law enforcement, and policy issues.

Rather than presenting the material using an investigator's process or by general forensic areas, we break the course material into five areas. This approach allows us to best obtain the balance required to meet our students' requirements. Table 1 shows the percentage of course time spent within a topic area.

Course Subject Area	Percentage of Course
Ethics and Legal Procedures	10
Basic Forensic Science	10
Media Capture and Analysis	40
Network Forensics	25
Digital Device Analysis	15

Table 1 Digital Forensics Course Material Breakdown

Ethics and legal procedures cover material on ethical behavior as it relates to computer usage. We discuss where individuals learn computer ethics (at home, school, and/or from the community) and how ethical behavior translates into a networked environment. The digital forensics aspect of these issues includes examining the criminal mind and how individuals reject ethics. We also define cyber crime and address search-and-seizure rights, the Fourth Amendment, and the large base of legal precedent that is currently being developed.

Basic forensic science covers both the law enforcement view of forensics and general laboratory policies. Some topics include Locard's Principle (as shown in Figure 1); Inman & Rudin's Forensic Science Paradigm; and questions of what should be seized at a crime scene, what must be in a warrant's text to ensure that the seizure is legal, what happens to items once they are seized, and how items are treated in the laboratory. Of these topics, some are addressed *via* a general overview and guided by the Department of Justice (DOJ) on Search

and Seizure of digital media. [1] Also discussed are the certification procedures for forensics and digital forensics laboratories of The American Society of Crime Laboratory Directors (ASCLD). [2]

Media capture and analysis covers the correct and accurate handling of media. This includes proper techniques for acquiring and verifying an image of the media and analyzing the media's physical and logical structure to extract evidence. The data-analysis portion includes some of the most difficult problems that forensics investigators encounter; *i.e.*, information hiding in the logical structure of the media and in the network traffic itself. This includes such topics as steganalysis, Domain Name Service (DNS) messaging, document metadata, and encryption.

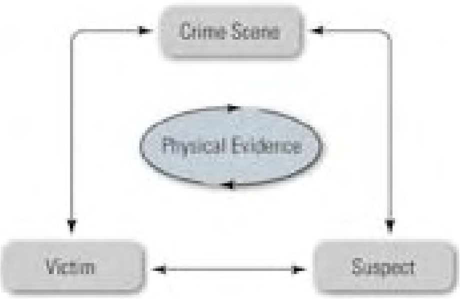


Figure 1 Locard's Principal

Network forensics investigates the situation from a network's standpoint. Here, the evidence can be contained within the network log files. Coverage includes the type of available logging information and how additional information about the network traffic itself can be extracted from this information.

Digital-device analysis looks at all the disparate devices that may confront investigators—the storage and extraction of information from Universal Serial Bus (USB) flash drives and MP3 players—and introduce the topic of mobile phone forensics, although not to the detailed level of media and data analysis.

Any introductory course in digital forensics should introduce all these topics. Depending on the program, the depth to which each is covered can vary. The text that we use for the course is Mandia and Prosise's *Incident Response and Computer Forensics*, 2nd Edition [3], supplemented with several documents on best practices, search and seizure, and class notes. For search-and-seizure best practices, we use the National Institute of Justice's *Electronic Crime Scene Investigation: A Guide for First Responders* [4] and the crime-scene training manual of the Air Force Office of Special Investigations. [5]

We have found that the best principles, methods, and science of the topics in Table 1 are learned in a joint setting that combines lecture and laboratory. During the lectures, we discuss the science and technology. While our students have generally taken course work in operating systems and computer architectures, they are usually unaware of how interfaces between components in today's desktop PCs are actually put together. Most lecture time is spent discussing these technical details, because a good forensics analyst must know how to manually perform the drive analysis even though there may be tools to do some tasks automatically. For example, a student should be able to describe the process of locating and undeleting a file in both general and technical terms. Because of the heavy laboratory component of this course, we will present the course contents in terms of the laboratories the students complete.

Laboratory Structure, Requirements, and Type: What Worked and What Didn't

AFIT is on the academic quarter system. This means there are ten weeks of instruction time available for a course. Most AFIT courses are four quarter-credit hours, which allows us to minimally interact with students for forty hours during the course of a term. Typically, AFIT student-instructor interaction increases by close to 50% (60 hours) for laboratory courses.

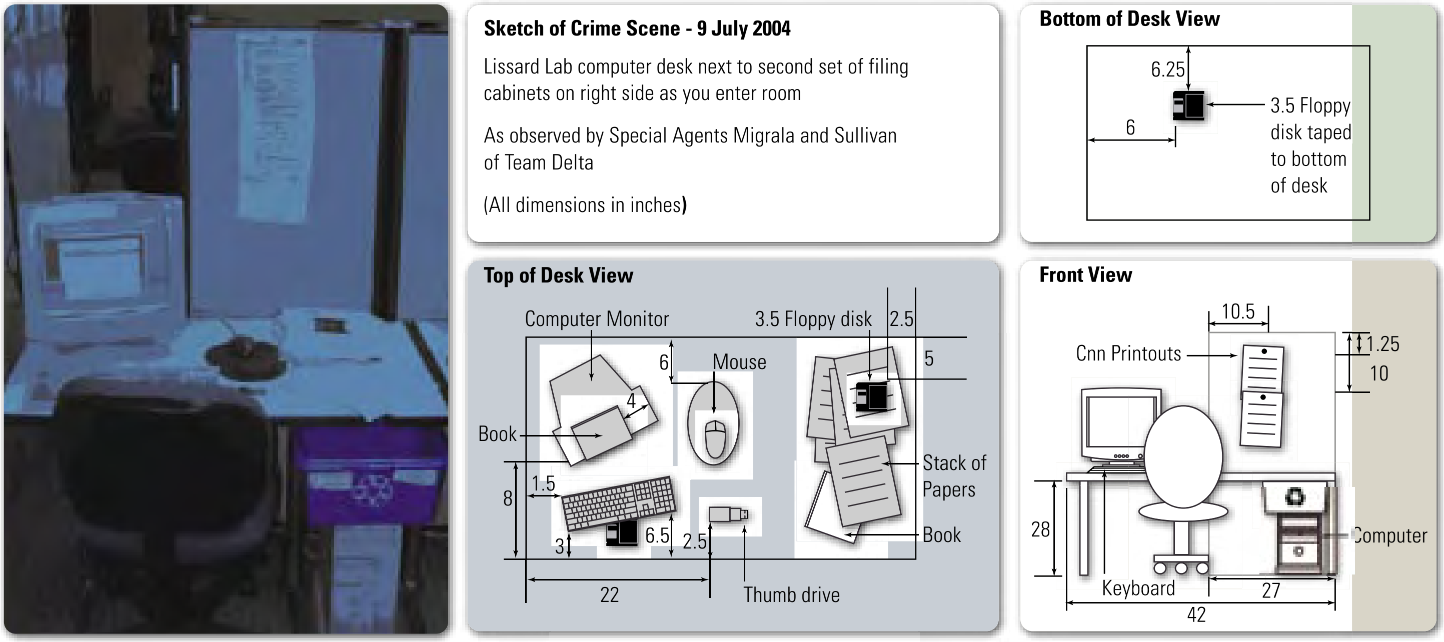
During the ten weeks, there are seven laboratories and a class project. The laboratories themselves are structured similarly to those at the University of Tulsa and cover the range of topics shown in Table 1. Students work in teams of three, a group process that provides two noticeable benefits. The first is associated with students' background and experience levels. Since the majority of AFIT students are military, a great wealth of operationally diverse experience is brought into the laboratory, and different experiences and ideas come together when solving laboratory problems. This improves each student's opportunity to complete the laboratory. The second benefit is seen from the group laboratory team structure. Because schedules vary, students are forced to maintain a chain of evidence, as it is not always possible for a student group to collectively meet in the laboratory at the same time.

Our digital forensics laboratory setup includes 16 machines, one of which is the victim/evidence computer. Students are issued their own hard drive for imaging, analysis, and retention of chain of evidence. The available software consists of a mixture of freeware and commercial products. We use the Helix and Penguin Sleuth bootable CDs, both of which include the `dd` imaging tool and the autopsy analysis tool suite. The commercial tools range from Winhex, which allows students the lowest-level view of the media, to EnCASE and Forensic Toolkit (FTK), which provide Graphical User Interfaces (GUIs) with advanced recovery and analysis tools.

The Laboratory 1 (Ethics and Legal Procedures) develops a first responder's policy for search and seizure, which forces students to think about the different cases they might confront when they attend Laboratory 2. An added twist is that each team must use another team's policy when conducting the Laboratory 2 (Basic Forensic Science). This gives students experience in following a policy they have not written and provides a differing view of the search-and-seizure procedure. Laboratory 2 itself requires students to conduct a search and seizure. In Laboratory 2, students must locate and seize all media and other physical evidence related to a fictitious case of an individual selling secrets. Figure 2 shows a typical setup for conducting the search and seizure.

Laboratories 3 and Laboratory 4 focus on incident response, since many graduates fill network support positions at military installations around the world. For these networks, 100% availability—or as close as possible—is an absolute must. Laboratory 3 (Media Analysis) addresses a live network response, in which the machine must remain on, and students must determine what has gone wrong and reverse it without loss of service. Specifically, students must open a secure command-line interface and create a network connection to another machine. Students then transfer as much volatile information from the machine, along with logs, registry keys, and anything else they believe may be relevant. After the transfer, the results are analyzed. In Laboratory 4 (Media Analysis I), students are locked out of the machine and must gain re-entry by circumventing the computer's security. This includes gaining access to both Basic Input Output System (BIOS) and log-in passwords.

The fictitious scenario continues in Laboratory 5 (Media Analysis II), in which students must seize the machine and image the hard drive. After imaging the drive, students analyze the drive and the file system for hidden information in Laboratory 6 (Media Analysis and Device Analysis). The first time the course was run, the drive that students analyzed was the same as the information one found in the evidence machine. For this, a few files were planted on the drive, along with logging on and off the network under different user names and


Sketch of Crime Scene - 9 July 2004

Lissard Lab computer desk next to second set of filing cabinets on right side as you enter room

As observed by Special Agents Migrala and Sullivan of Team Delta

(All dimensions in inches)

Figure 2 Laboratory 2 Setup and Crime Scene Watch

a few other normal user behaviors. The analysis of the drive took students two weeks to complete rather than the one week scheduled for analysis because of the slow indexing and searching of the 20-GB drive. Additionally, the size of the drive made it difficult to tailor the image and include enough "evidence" to provide an interesting search for students. On the positive side, the time required to analyze the 20-GB drive did provide students with the very real experience of how time-consuming media analysis is.

In the following year, students imaged the hard drive and a USB flash drive and then performed the analysis on the USB flash drive alone. Because of the much smaller scale (128-MB vs. 20-GB), it was much easier to hide a larger number of items in different ways and to fill the drive space. The resulting image was designed so that no single forensics tool would find everything. The image itself contained information hidden in all the different slack spaces, in the boot cylinder, a hidden partition, deleted files, bridging sectors in a reverse order (*i.e.*, the keyword is only locatable by searching in the file), steganography, and very simple cryptography. There was also the addition of a compression bomb. If students do not pay attention to the analysis tools settings and search the file, the bomb causes the machine to freeze. This approach has worked much better but is still not perfect because, although some of hidden items point to other items, they are not all set up as a set of "clues" that lead to some incriminating piece of evidence. We hope to build an image along these lines this year.

In Laboratory 7 (Network Forensics), students analyze two days of network capture logs and track individuals who attack the system as far as their Internet Service Provider (ISP). In the past, the network traffic logs have been pulled from the Lincoln Laboratories Intrusion Detection System Dataset. [6] Because of the datasets' statistically normal behavior [7], the Lincoln Laboratories dataset is being replaced with one that students captured during this year's Cyber Defense Exercise, which provides a much richer, more realistic environment for forensics analysis.

The laboratory structure of the course has been changed in two ways to provide better flow and challenges for the students. The first change was to Laboratory 6, which was originally two laboratories: restoring deleted files and then the analysis. These two laboratories were converted to one laboratory at the same time as the move from the 20-GB image to the 128-MB flash drive. Originally, with the available tools, the undelete process took very little time, while the analysis took twice the allotted time. Laboratory requirements were also altered to include the entire forensics process from search and seizure, through live response, collection, and analysis to final project. The second change was the addition of a final project, which allows students to explore an area of forensics that interests them but may not be covered in the course. Some student topics have included steganalysis, analyzing anonymous routing networks, wireless network penetration, and compression file cryptography.

Other Lessons Learned

In addition to working closely with the course instructor, we have found that inviting Subject Matter Experts (SMEs) to speak to the class is very important to a robust learning experience. SMEs address specializations that an instructor may not have. They are also typically actual practitioners, who give students a real-life perspective on digital forensics. This year, the director of the Federal Bureau of Investigation's Miami Valley Regional Computer Forensics Laboratory will discuss the law enforcement view of digital forensics. During the past two years, we have had an expert in malware analysis speak on catching and reversing engineering viruses, Trojans, and other software security risks. In the near future, we hope to bring in a legal expert to present and conduct a mock trial.

Our outreach to local law enforcement is another rewarding aspect. Since the course's inception, we have offered local law enforcement personnel the opportunity to attend the course without charge. Attending officers have enjoyed the course, commenting that the level of difficulty with the USB image analysis required more of them than most cases they worked on a regular basis. The other benefit is that, during class, they are very similar to SMEs and provide a real-world view of the topics in the course.

An additional benefit of the digital forensics course is our collaboration with Sinclair Community College (SCC), Dayton, OH. Through a grant sponsored by the National Science Foundation, we are partnering with SCC to develop courseware appropriately structured for first responders who attend classes at the community college level. We are in the process of sending a survey to Chief Information Officers of large corporations in the Miami Valley to gather information on their preparedness to deal forensically with a computer security problem and on their interest in a course at the community college level. Working closely with SCC faculty, we will use this information to tailor their course to best meet the requirements of the corporate and first-responder communities. Our vision is to assist in preparing SCC students who will be hired by area corporations to deal with and understand the ramifications of mishandling possible evidence and how to interface with local law enforcement.

Conclusion

Currently, we offer our digital forensics course once each year. We continue to improve course content and make the laboratories as relevant and realistic as possible. Our students' feedback indicates a positive learning experience and a feeling of high value received for the course-content exposure. We believe education and research in digital forensics is critical to our national security. In future Air Force and US Department of Defense (DoD) assignments, our graduates will face many issues presented in class. Their digital forensics exposure will give them a distinct advantage over our potential adversaries, be they nation states or malicious hackers.

We plan to extend the digital forensics offerings at AFIT by adding courses that offer more depth in both Network Forensics, Digital Device Analysis, and even in Data Analysis, including more in-depth coverage of information hiding and its role in steganalysis, metadata, and network protocols. ■

Acknowledgments

The views expressed in this paper are those of the authors and do not reflect the official policy or position of the US Air Force, DoD, or the US Government.

References

1. Department of Justice, computer crime and intellectual property section, *http://www.usdoj.gov/criminal/cybercrime/searching.html*, June 2006.
2. The American Society of Crime Laboratory Directors, *http://www.ascld.org*, June 2006.
3. Mandia K. and C. Prosise, "Incident Response and Computer Forensics, 2nd Edition, MacMillan Publishing, 2005.
4. National Institute of Justice, *http://www.ojp.usdoj.gov/nij/*, June 2006.
5. U.S. Air Force Office of Special Investigations, *http://public.afosi.amc.af.mil/*, June 2006.
6. Lippmann, R.P. and J. Haines, Analysis and Results of the 1999 DARPA Off-Line Intrusion Detection Evaluation, *Recent Advances in Intrusion Detection, Third International Workshop, RAID 2000*, 162–182.
7. Mahoney, M.V., and Chan, P.K., An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection, *Recent Advances in Intrusion Detection, RAID 2003.*

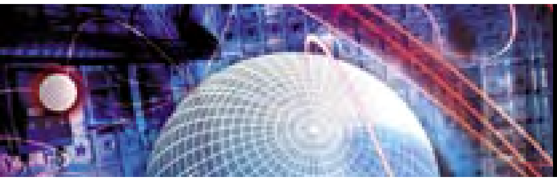
About the Authors

Dr. Gilbert "Bert" Peterson | is an Assistant Professor of Computer Engineering at the AFIT. Dr. Peterson received a BS degree in Architecture, an MS in Computer Science, and a PhD in Computer Science from the University of Texas at Arlington. He teaches and conducts research in digital forensics and artificial intelligence.

Dr. Richard "Rick" Raines | is the Director of the Center for Information Security Education and Research (CISER) at the AFIT. Dr. Raines received a BS degree in Electrical Engineering from the Florida State University an MS degree in Computer Engineering from AFIT, and a PhD in Electrical Engineering from Virginia Polytechnic Institute and State University. He teaches and conducts research in information security and global communications.

Dr. Rusty Baldwin | is an Associate Professor of Computer Engineering in the Department of Electrical and Computer Engineering at the AFIT. He received a BS degree in Electrical Engineering (*cum laude*) from New Mexico State University in 1987, an MS degree in Computer Engineering from the AFIT in 1992, and a PhD in Electrical Engineering from Virginia Polytechnic Institute and State University in 1999. His research interests include computer communication networks, embedded and wireless networking, information assurance, and reconfigurable computing systems.

The authors may be reached at the Air Force Institute of Technology, Department of Electrical and Computer Engineering, 2950 Hobson Way, WPAFB, OH 45433-7765 or by e-mail at gilbert.peterson@afit.edu, richard.raines@afit.edu, and rusty.baldwin@afit.edu



Letter to the Editor

Q ***While attending both the Black Hat and DEFCON conferences this year, there was mention of something called fuzzing. Apparently, this has been around for quite some time, but I'm unfamiliar with this term. Might you be able to explain what it is?***

A Fuzzing, also known as fuzz testing, is simply a software testing technique. Developed in 1989 at the University of Wisconsin-Madison, fuzzing is used to find bugs by feeding random input into software applications. This random data is the fuzz used for testing the application. If the program crashes or hangs up, than the test is considered a failure and the software must be repaired.

Fuzz testing is generally used as part of an overall software security program and is not meant to substitute for other, formal methods. As happens in many cases, passing a fuzz test may only demonstrate that an application can handle an exception without crashing; it does *not* mean that the application is actually behaving correctly. Therefore, fuzz-test failures should be viewed as bug-finding tools, rather than viewing passes as an assurance of the quality of an application.

Now, as is the case with many technologies, fuzzing tools in the wrong hands can have negative results. As already stated, fuzzing is designed to find bugs in software. Therefore, adversaries wanting to exploit software now have the capability to do so using these same tools.

If this *Letter to the Editor* on fuzzing has peaked your interest, please stay tuned, as we intend to have a feature article on fuzzing in an upcoming *IAnewsletter*. In the meantime, should you have any additional questions, please do not hesitate to contact us. ■

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	□ No. 2	□ No. 3	

Fax completed form to IATAC at 703/984-0773

Calendar

November

ANG SLC
5–8 November 2006
Adam's Mark
Dallas, TX
https://www.technologyforums.com/6NS/

2006 Techno Forensics Conference
6–8 November 2006
NIST Headquarters 100
Bureau Drive, Administration Bldg. 101
Gaithersburg, MD
http://www.thetrainingco.com/html/TechnoForensics2006.htm

Information Assurance Conference 2006
6–8 November 2006
Radisson Hill Country Resort & Spa
San Antonio, TX
https://www.technologyforums.com/6AI/

SIGAda 2006 - ACM Special Interest Group on Ada International Conference
12–16 November 2006
Hotel Albuquerque at Old Town
Albuquerque, NM
http://www.acm.org/sigada/conf/sigada2006/

7th Annual Security Conference & Exhibition
15–16 November 2006
Ronald Reagan Building
Washington, DC
http://www.e-gov.com/EventOverview.aspx?Event=SEC06&NoCache

The Gartner Identity & Access Management Summit
29 November–01 December 2006
JW Marriott Las Vegas Resort
Las Vegas, NV
http://www.gartner.com/2_events/conferences/iam1_section.jsp

December

Annual Computer Security Applications Conference (ACSAC)
12–13 December 2006
Wyndham Miami Beach Resort
Miami, FL
http://www.fbcinc.com/event.asp?eventid=Q6UJ9A00AODJ

January

Gartner Wireless and Mobile Summit
5–7 January 2007
Dallas, TX
http://www.gartner.com/2_events/conferences/ra10.jsp

Department of Defense Cyber Crime Conference 2007
21–26 January 2007
St. Louis, MO
https://www.technologyforums.com/7CC/index.asp

Network Centric Warfare 2007
22–25 January 2007
Ronald Reagan Building and
International Trade Center
Washington, DC
http://idga.org/cgi-bin/templates/genevent.html?event=11275&topic=221

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