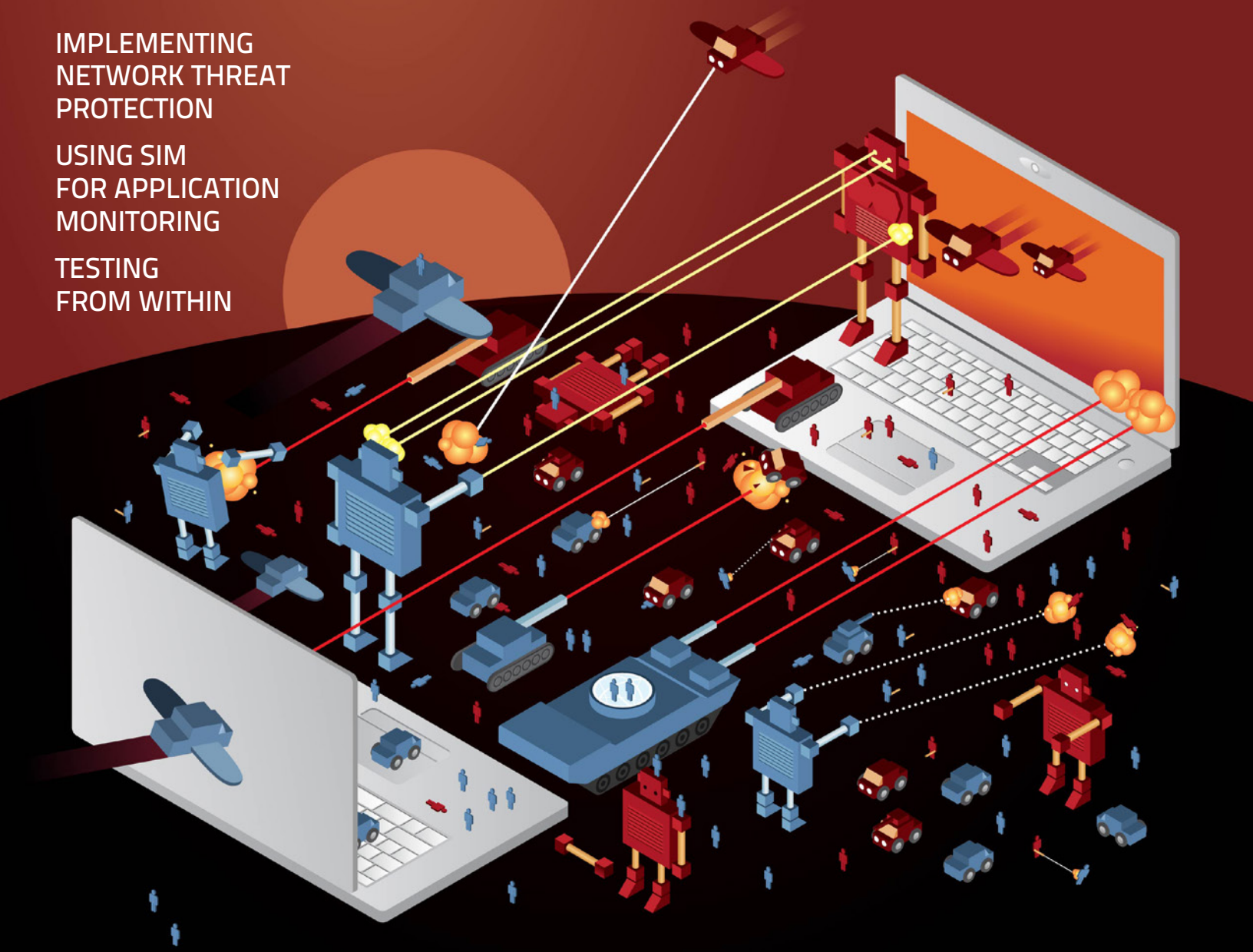
INFORMATION SECURITY

ESSENTIAL GUIDE

IMPLEMENTING NETWORK THREAT PROTECTION

USING SIM FOR APPLICATION MONITORING

TESTING FROM WITHIN



NETWORK SECURITY

Learn how to develop strategies to gain network security visibility and make the most of your SIM, GRC, log management, network monitoring, penetration testing and forensic efforts.

TechTarget

Network Security: Complexity and Coordination

BY ROBERT RICHARDSON

YOUR NETWORK SECURITY is increasingly complex and the ostensibly simple matter of just keeping track of its components has spawned a slew of network discovery tools. But if that complexity weren't enough, the attacks that chew at your network every day are also growing in complexity and stealth at breakneck speed. So you need a correspondingly complex set of network security tools. And more than that, you need strategy.

The way to get that maximum effectiveness is to focus on coordination, and John Burke points out that this is perhaps best initiated with the adoption of a centralized policy engine. With an overarching view of security policies, what you get is context-sensitive tuning of your security controls. If there's any way for an organization to be flexible and secure at the same time, this is where its security initiatives must be centered.

This special *Essential Guide* also offers views into SIM and application monitoring from Joel Snyder, as well as Dave Shackleford's spirited and sensible argument in favor of developing an internal penetration testing team. For a lot of organizations, pen testing is that "one last thing" that never quite gets covered. But if you think you've got your network defenses coordinated and you want to see if your perceived gains are real, regular probing by a skilled internal team can be a vital way to keep security policy discussions real. ■

ROBERT RICHARDSON *is director of TechTarget's Security Media group and* Information Security *magazine. Send comments on this column to feedback@infosecuritymag.com.*

IMPLEMENTING NETWORK THREAT PROTECTION

Products need to work in concert to deliver threat monitoring, detection, analysis, testing, and containment.

By John Burke

IN JUST THE past few years, the threat universe has changed markedly. Cyber-threats now target individuals and small, specific groups all the way up to broad swathes of the global population, whether via old-fashioned malware-infected email attachments or through throw-away Trojan horse websites that exist for no reason but to be a vehicle for malware. Denial-of-service (DoS) attacks have become more frequent and larger in scale, and are motivated now not only by attackers' desire for profit, but also by political goals such as hacktivism, protests against specific policies and cyberwarfare aimed at hurting a government's ability to operate. DoS attacks threaten the online presence, services

and even baseline Internet access of organizations of all sorts, commercial and non-profit, large and small, whether by saturation of network links or by depletion of limited resources like SSL sessions, or by crashing applications, servers, firewalls or routers.

Add to this such things as:

- The burgeoning field of mobile phone compromise in a BYOD world;
- The still-increasing use of laptops and other mobile computing devices;
- The ongoing problem of data leak prevention (DLP);
- The movement of attacks up the network stack to the application layer; and
- The growing use of public clouds for data storage and application hosting.

All this amid the growing use of multimodal attacks that work from multiple angles toward the same goal—gaining entry to the targeted systems, your systems—and a security landscape emerges that is dramatically different from the one information security professionals have known through much of their careers.

There are myriad security "point products" to address these various threats: firewalls, intrusion-prevention systems, DLP tools, endpoint protection offerings, DDoS appliances and services, and on and on. Although these single-purpose-driven products are growing more sophisticated as they rise to meet the challenge of the multimodal adaptive persistent threat, they are not sufficient to protect today's perimeter-less, virtualized, distributed, mobile, multi-application and multi-device environments.

What's the No. 1 reason these technologies fail to detect critical attacks? Coordination. The more places an enterprise technology staff have to look to monitor security threats and responses, the more places they have to manage policies, the more processes they need to ensure that everything is kept synchronized and coordinated, hence the more difficult it gets for them to do so and the more likely it becomes that gaps will appear, persist and grow, and that inconsistencies will persist. Unconnected and uncoordinated, the defenses become less like the Walls of Constantinople and more like the Maginot Line—easily bypassed islands of security.

What's required instead is an integrated approach to security architecture: pervasive network threat protection to address pervasive threats; coordinated

and linked security systems to address multimodal and persistent attacks. Pervasive in that it reaches into, out to or around all the critical components of the enterprise technology infrastructure: protecting systems, users and data in a coordinated fashion. It necessarily comprises many parts, since no single security product or single vendor's portfolio covers everything. Some parts are focused on the coordination of all the others.

One such coordination tool, and the one at the heart of a pervasive security strategy, is a centralized policy engine. A policy engine is at its core a repository for security policies. Generally written in the standard eXtensible Access Control Markup Language (XACML), the policy engine is also a central distribution tool for those policies, pushing them out to other systems, or producing them when requested by other systems. It may also provide for workflows around policy development, maintenance and audits.

One policy, maintained in one place with one process, is the goal and justification of a policy engine and the heart of a pervasive protection strategy.

Policy engines exist from a variety of vendors, and Nemertes Research recently found that nearly 30% of organizations are ready to call their architectures pervasive now, whether built around a commercial policy engine or around something they developed in house. With a centralized policy engine, it becomes easier to define the kinds of context-aware security policies that, for example, let employees use some Facebook applications but not others, depending on the employee, the application, and the time and place and platform.

One policy, maintained in one place with one process, is the goal and justification of a policy engine and the heart of a pervasive protection strategy. Although such a tool is an additional investment, it both increases the value of previous investments in security point products and allows process changes that reduce operating costs as well. When a tool is responsible for ensuring consistency and coordination among security systems, IT and security staff spend less time getting everything in line with policy. When policy is implemented automatically across the security infrastructure, the various security systems relying on that coordinated policy to do what the organization wants and needs them to be doing, increasing their value.

As an example, consider the use of social media again. Many organizations are relaxing prohibitions on use of social tools at work, but others are still reluctant to do so because they lack granular application function control. If an organization implements a next-generation firewall capable of distinguishing between actions like posting messages, sending files and playing games, and if the organization is also pushing out a DLP client that can watch for certain kinds of data flowing out into the social space, that new combination of tools enables manageable, secure use of social software to a degree that wasn't previously possible.

Log analysis is either built in or layered on and can provide both warnings when suspicious or dangerous things are happening.

However, as policy decisions get made about who can use the tool, when they can use it and what each group can do with it, there are now at least two places where security staff have to implement those policies: two interfaces, two change management events, two audit trails and possibly two sets of staff making the changes. With a policy engine in place, however, the changes could be made once, in one place and pushed out to both systems. The same would be true if there were also a mobile device management (MDM) tool in the mix to control smartphone usage and a WLAN security system to enforce a policy specific to wireless users. Yet even with four enforcement systems, with a centralized policy engine there would still be just one change process, one audit trail, one set of staff setting up one policy definition in one place.

Centralizing and automating the front end of the security process, the implementation of security policies in the various enforcement tools on hand, is a change most IT shops have not explored. On the other hand, many have looked at (and to varying degrees implemented) centralization and automation on the other end of the process: the collection, monitoring, and analysis of log data. Security information and event management (SIEM) tools are broadly available and widely implemented, ranging from open source and free systems to major commercial tools. Log analysis is either built in or layered on and can provide both warnings when suspicious or dangerous things are happening, as well as forensic help dissecting and understanding what happened after the fact in the event of a breach or attempted breach. The key, of course, is to make sure that the various tools don't operate as silos and to make sure logs are

brought together for analysis. Just as it is important to have consistent application of policy across all security point products, so it is important to have a view across what all those tools are seeing and doing.

When assessing the effectiveness of new or existing security products and services, security teams should think in terms of not only individual functionality—the tool's fitness for its niche in the ecosystem—but also in terms of its ability to work in concert with complimentary technology to deliver effective threat monitoring, detection, analysis, testing and containment. Such tools need to be able to consume policies using industry standard methods like XACML, either directly or through an automatable transformation that can be built into an enterprise service bus or the like.

Moreover, security tools need a steadily increasing level of granularity, so that they can enforce policies sensitive to the time, manner and place of person or system's attempt to do something. As employees become increasingly mobile, as telework and telecommuting become increasingly common, and as the use of contractors and other consultants continues to increase, the ability to finely distinguish between what different people can do in the same place, and what a given person can do in different places or via different devices, will become increasingly important.

And, of course, all need to be able to fit into a unified log and event management infrastructure, often based on a SIEM system, so that security professionals have a "single pane of glass" to look into to see the state of the organization's security. Whether in-house or as-a-service, such a centralized picture of decentralized protections and responses to decentralized attacks is crucial to making security not just successful but scalable and sustainable.

By unifying policy, management, protection and monitoring, a system of pervasive protection can meet the challenge of pervasive threats we all now face. ■

JOHN BURKE *is a principal research analyst with Nemertes Research, where he advises key enterprise and vendor clients, conducts and analyzes primary research, and writes thought-leadership pieces across a wide variety of topics. Burke leads research on virtual enterprise, focusing primarily on the virtual and mobile desktop, application delivery optimization and management and orchestration tools for the virtualized data center and the cloud. He also covers the security aspects of all of these topics, as well as server and storage virtualization, network and application performance management and monitoring, branch office IT and SOA.*

USING SIM FOR APPLICATION MONITORING

Security information management systems aren't just for network security monitoring anymore.

By Joel Snyder

ENTERPRISES HAVE ADOPTED security information management systems (SIMs) for their value in correlating, reporting, and alerting on network security. By feeding firewalls, intrusion detection and prevention, and vulnerability analysis into a common platform, network and security managers have a valuable window, giving greater visibility and helping to clear out the noise.

Despite their name, though, SIMs can be used for more than network security monitoring. In many cases, the same tools can bring value to application managers if they're used correctly. With attacks targeting the application layer, SIMs can help find security problems in enterprise applications that otherwise

might get missed. But SIMs can do more than identify security threats: Any hard-to-find event or application performance issue can show up through careful analysis.

We'll walk through the four steps application managers need to integrate applications into enterprise security information management systems and begin analyzing, reporting and alerting.

FEEDING APPLICATION DATA INTO A SIM

Before you begin using a security information management system, you've got to start feeding it your application data. In the world of networking and security, this is easy because network and security devices universally support SYSLOG, a way to ship log data over the network to a central point. SIMs love to get data via SYSLOG, so that's always your first choice if your application supports it. You'll have to work a bit harder for the rest of them.

For enterprise applications that send logs to Windows Event Log (Microsoft Exchange is the most common candidate here), getting those into a SIM will be easy. Most SIMs already have a simple strategy to connect to Windows Event Logs, either using a SYSLOG converter that sends Windows logs out via SYSLOG, or through some native tool that pulls logs directly from Windows.

Applications that write standard log files or send their logs to databases will pose a greater challenge. Some SIMs have already dealt with this problem out of the box and have tools or daemons that will monitor databases for changes or watch log files as they grow. If your SIM doesn't do that, you'll need to figure out some way to get those logs sent over. One approach is to wait for hourly or even daily intervals and use a batch procedure to copy the logs for the last time interval over to the SIM all at once. If your goals in using the SIM are more focused on reporting, correlation and forensics, then having a delay of an hour for logs to be sent over may be fine.

When you start thinking about what logs to send to the SIM, make sure you're realistic about performance and about your goals.

When you start thinking about what logs to send to the SIM, make sure you're realistic about performance and about your goals. The temptation is to

send everything, but some logs, such as back-end database logs, may be more than your SIM can handle. In general, you should start from the edge of the network and work your way backwards. Try and capture the full stack that represents the application from beginning to end. For example, if you have load balancers in front of an application, you'll definitely want those logs. This is doubly true if you are doing source network address translation (NAT), a common strategy in load balancers for application servers because of the simplified deployment model. Without logs from the load balancers, you won't be able to track transactions back to the true originating IP address or identify some types of denial-of-service (DoS) attacks. Keep going, including your Web front ends, application servers and database servers. If you're adding logging for email, make sure you include the whole set of products: antispam/antivirus gateway, email relay, and finally, messaging server pieces.

As you work in toward the back end of the application, keep in mind the potential goals of SIM analysis: correlation and analysis, alerting, reporting, and forensics. If the logs don't advance one of those goals, they aren't going to help you much and will just clog up your SIM.

PARSE, NORMALIZE AND STORE APPLICATION DATA

The difference between a SIM and a log storage system is in the ability of the SIM to "understand" the logs you send it, a process generally called parsing or normalization. If you expect to get anything useful out of your SIM, you'll need to make sure it is able to interpret the logs, collect information from various fields, and generate reports, calculate rates, and correlate information across log entries.

In many SIMs, the parsing and normalizing are dark magic (OK, that's a technical term) reserved to the SIM vendor's engineering team. In that case, you'll have to provide them with a sample of your log files so they can add the necessary logic to their product. You may also need to add fields to the SIM's database, a task that can either be easy or next to impossible, depending on how flexible your SIM is.

At this stage, be prepared to give up—if your SIM vendor says you won't be able to do something useful with these logs, don't try and force a round peg in a square hole. Even if your SIM can't parse and normalize your application log files, you can still get useful information from the logs. Unparsed log files,

though, won't give you good reports, statistics or analysis. For example, many SIMs will treat unparsed log entries as blobs of text, which would let you raise an alert on certain specific entries, such as fatal errors or security alerts. That's not giving you the full value of a SIM, but alerting combined with some forensics and log search capabilities can make it worth your time to go down this path.

If you're parsing the logs yourself or advising your SIM vendor, you'll need to figure out which fields to pull out of each log entry. Focus on fields that will make sense for reporting, alerting, and correlation. For example, if you're pushing email application logs over, you'll want to try and track fields such as message ID, envelope-from, envelope-to, subject, and date as a bare minimum. When adding more complex applications, such as ERP systems, you'll have to pick and choose fields from a fairly complex environment. Parse fields that will help you trace a transaction from the originating IP all the way back to the application. That means starting at the front end and moving back one level at a time, always looking for some way to link entries at one layer to the next. For example, if you have a load balancer changing IP addresses, you'll want to capture the original IP address and the changed IP address. Then, you can link Web server logs with the changed IP address to the original IP address by going through the load balancer logs.

If you're parsing the logs yourself or advising your SIM vendor, you'll need to figure out which fields to pull out of each log entry.

Depending on your log management strategy, you may want to set a shorter expiration date for application-layer logs than other logs you're collecting. Most application stacks are going to have their own log management systems, making the SIM a duplicate of what's already being collected elsewhere. In that case, keep what you need for your reporting, alerting and forensics requirements, but don't look to your SIM for long-term log archiving if that's going to mean keeping two (or more) copies of everything in two different systems. Most SIMs have built-in reporting tools that will provide some trending information, but unless there is specific report for application-type logs, you're not going to get much useful trend data over the long term. That means there's no point in saving three years worth of application logs if you never go back more than three months in forensics.

WRITING BUSINESS RULES AND ALERTS

A SIM will never know what you want to do with your logs unless you tell it. All SIMs are rule-based, which means rules are going to be the best way to add value to the task of looking at application logs with your SIM.

Writing business rules isn't the most fun part of this exercise, but it is the one that leverages the power of the SIM best. When you begin feeding application data to the SIM, you'll want to focus on two fronts: reusing existing rules and writing new ones to leverage new types of data.

If your SIM has parsed and normalized log files properly, then some of your existing rules will probably apply. For example, a login failure on a database server will be caught by the same rules you have looking for login failures on Web servers or firewalls. This is what you want—you want your application log messages to look as close to existing log messages as you can, so any business logic you've put into your SIM will apply to new applications.

However, you will probably have other rules that are specific to your applications. Most SIMs can watch for regular expressions in individual log entries, patterns and rates of one type of log entry, and correlation between two (or more) entries. Application logging can take advantage of each of these features.

For example, the MySQL database will log "slow queries," ones that take more than ten seconds to run. If you get five of these a day, or even 50, that may be normal, but if the rate of these suddenly jumps up, you want to know about it. This is an ideal job for a SIM, which can dig out slow queries from all

Glossary

THERE ARE A LOT OF ACRONYMS FOR THE SAME TECHNOLOGY.

Call it what you will: security information management, security event management, or some combination of letters, the difference between SIM, SEM, SEIM, and ESM is marketing.

Security information managers, known as SIMs, accept security and networking information from multiple sources within the enterprise and analyze it to provide a higher level of understanding. —JOEL SNYDER

the other logged messages and let you know when the rate goes above a threshold. Or you may have certain queries that you know will be "slow," but want to know about any others. That's a good use for the rule-based engine in a SIM, because the engine should be able to correlate different log messages together to catch the interesting ones.

Business rules can be simple if you want to just look for regular expressions. For example, you may want to be alerted whenever certain banned IP addresses try to connect to the application, or if a user with a disabled account tries to log in. And if you want to identify application-layer attacks such as SQL injection, for example, it's easy to write a handful of rules that look for sequences such as "xp_" in traffic going towards a normal Web server.

As you're writing business rules, don't forget the value of combining sources of information from outside the application with your application logs. For example, you may already have IDS/IPS events being sent to the SIM. Those can be correlated with logging to filter out problems that have already been solved at the network layer, or to help identify more information about suspicious transactions.

The last step of real-time log analysis is alerting: telling someone or something about whatever you've found in the logs. Each deployment will be different, but if your SIM is already installed, you'll want to try and fit smoothly into the existing alerting strategy. However, you may find network and security teams have a different way of looking at alerts since they are more focused on real-time issues, ("something bad is happening right now!") while application logs are often more useful seen over a longer period of time, such as a day or a week ("here are all the interesting slow queries from last week"). Don't be afraid to point this out to the team managing the SIM so they don't get the wrong idea about what's most important to you.

USING SIM TO HANDLE REPORTING, ARCHIVING AND FORENSICS

Security information management systems can help you with reporting, forensics (looking backward at logs to understand the root cause of problems), and archiving of your application logs. If your application is long-standing, you may already have strategies for all three of these activities. Just because you're adding a SIM into the mix doesn't mean you have to change your strategy for long-term log retention. With compliance regimes requiring three to seven years of

logs, SIMs and their high-speed databases can be an expensive alternative to a simple log server.

Regardless, whether you're using the SIM for archiving and forensics, you want to filter logs going to the SIM. For example, many applications can write detailed debugging information as well as normal transaction information in their log files. That debugging information might be useful to the application manager, but not so useful to the SIM. You can filter either by ignoring some events when they arrive at the SIM or by filtering them out before sending from the application to the SIM. Which technique you use depends on your application tools and your SIM, but the best approach, if your SIM supports it, is to filter out at the SIM. This way, you can easily increase the types and level of logs you save at the SIM without having to go back to the application manager and ask them to change anything.

Reporting can be a mixed bag with SIMs when trying to add application logs into the mix.

Reporting can be a mixed bag with SIMs when trying to add application logs into the mix. If you are generating your own alert data based on correlation and analysis of logs, then reports on your alerts will be invaluable. However, don't be disappointed with stock reports. The off-the-shelf SIM reports are likely to be aimed more at security-type events, and may not be adaptable to the kinds of things you want to summarize and count in your application logs.

Adding application log data to your enterprise SIM can bring a wealth of information and give you new insight into what is going right, and wrong, with your enterprise applications. However, SIMs are focused on security and the path to non-security data from application logs can be arduous, involving the SIM vendor and work on your part. A successful integration will take time, but will increase your security and application awareness all at the same time.

JOEL SNYDER *is a senior partner with consulting firm Opus One in Tucson, Ariz. He has worked in IT for more than 25 years. Send comments on this article to feedback@infosecuritymag.com.*

TESTING FROM WITHIN

You need an internal pen testing team. Here's why and how to build one.

By Dave Shackleford

IN TODAY'S COMPLEX security landscape, new threats are emerging on a regular basis, and we have more vulnerabilities than ever before. As part of a sound security program, most mature security teams have developed a vulnerability management program that includes network and application scanning, patching, and risk assessment. However, many organizations are now asking themselves if it's time to take these programs to the next level by adding penetration testing capabilities into the mix. For many reasons, ranging from compliance mandates to improved vulnerability and threat intelligence, the answer should be a resounding "yes."

Yet there's often some confusion on how best to approach pen testing, what kinds of skills are needed, the tools to use, how often to do it, and what the process should look like. We'll clarify best practices for security pen testing and explain how to build an internal testing program and measure its success.

WHY YOU NEED AN INTERNAL PEN TESTING TEAM

There are many reasons why organizations should seriously consider performing penetration tests with an internal team. While most organizations today have some type of penetration test performed annually (usually for compliance reasons), penetration tests should ideally be done more often. In essence, a penetration test is a highly specialized, security-specific validation of controls in place. This can range from testing network and application access controls, to software code and IT operational processes.

When developers test code as part of a typical quality assurance (QA) cycle, the testing is done for each iteration of code production to catch bugs. Penetration testing is really a form of QA that looks for flaws in network architecture and design, operating system and application configuration, application design, and even human behavior as it relates to security policies and procedures. If you only performed QA once a year on your applications, would that be enough?

The goal of penetration testing is to find vulnerabilities or flaws in networks, applications, and operating platforms that could potentially be exploited by attackers or simply cause business disruption of some sort. Due to the rapid pace of change in most enterprise IT environments, as well as the rising complexity of most infrastructure, the likelihood of configuration issues and less-than-adequate security controls being implemented increases significantly. Regular penetration testing can be a useful way to determine with a higher degree of certainty that flaws really do exist. However, in order to effectively find these issues before attackers, the testing regimen you put together needs to be focused on consistent, repeatable testing.

In addition to being a security best practice overall, many compliance mandates and industry regulatory guidance specify penetration testing as a requirement or recommendation. There are numerous examples to choose from, some of which are more concrete and others that are more implied. Likely the most well-known standard that explicitly requires penetration testing is the Payment

Card Industry Data Security Standard (PCI DSS), which requires pen testing at least once a year "and after any signification infrastructure or application upgrade or modification."

In addition, the PCI Council has released a separate information supplement for penetration testing that specifies who can perform them, as well as suggestions on a methodology and other useful guidelines. The supplement clarifies that internal teams can perform their own penetration tests to meet compliance, as long as they are experienced and are operationally distinct from the areas being tested.

While not explicitly requiring pen tests, HIPAA requires thorough assessments of risks and vulnerabilities to electronic protected health information. According to general security best practices, this should include a technical assessment like pen tests. Pen testing is also included in the Consensus Audit Guidelines (20 Critical Controls) from the SANS Institute.

THE PEN TESTING PROCESS

The first thing to understand is there are many different types of penetration testing your team may need to perform. Network and system-focused pen testing is still the most common, with an emphasis on traditional server technology, operating systems like Microsoft Windows and Linux, and network-based access and security controls like firewalls, VPNs, routers, and intrusion detection and prevention systems (IDS/IPS). However, more and more organizations are testing Web and other applications regularly, too, and this coincides with the threat landscape we face today. An examination of some of the biggest breaches of 2011 by Chris Wysopal, co-founder and CTO at Veracode, found that two-thirds were caused by some sort of application issue. To that end, most mature organizations will need to diligently perform both network-focused and application-focused penetration tests.

Most penetration testing regimens follow this cycle, or one similar to it:

1. **Reconnaissance:** The "recon" phase of a pen test is focused on information gathering, often through online and openly available sources. Targeted Google queries, DNS and WHOIS lookups, and website searches can provide a vast amount of valuable information about an organization that will come in handy in later phases of the test. Common information gathered

includes names and usernames, email addresses, DNS records, keywords that may be included in passwords, evidence of Web-based configuration issues, and sensitive data that has been inadvertently published. Although some organizations skip this phase, those that choose to regularly perform "recon" generally know much more about their organizations' online presence and potential exposure areas.

2. **Scanning and enumeration:** Scanning may include website crawling (looking for directory structures and site structure), network scanning with tools like Nmap (looking for active IP addresses and TCP/IP ports that are open), and more in-depth vulnerability scanning that actively probes for specific evidence of existing vulnerabilities in both Web applications and network and operating system components.

3. **Exploitation:** Once vulnerabilities have been identified, a pen tester will actively exploit them to try and gain access to systems and applications. Remote server-side exploits are still the most common, followed by Web application exploitation using common techniques like SQL injection, and will also likely include authentication attacks like brute-force password guessing.

4. **Pivoting:** After gaining access to systems and/or data, most pen tests should allow for "pivoting," which simply means establishing a new source of attack on the newly compromised target and continuing the scanning and enumeration cycle from that vantage point.

There are quite a few other types of activities that may occur during both network and application pen tests. For instance, if a pen tester gains access to encrypted password hashes, she might choose to start cracking those passwords in the background while continuing with other tasks (if this is in scope for the test). For application tests, client-side code may be enumerated and exploited, possibly allowing for authentication bypass to sites. Some organizations are starting to employ a technique called "fuzzing" to send unusual data into applications to see if error conditions can be caused, thus exposing potential weak points. Advanced security teams may even develop their own exploits. Another type of pen test is social engineering, where the team tries to exploit insecure employee behavior.

HOW TO BUILD AN INTERNAL PEN TEST PROGRAM

To build a pen testing program, the simplest way to get started is by leveraging any existing vulnerability scanning processes and tools you have in-house today. Then, you'll need a skilled team. The skills your team will need vary widely depending on the infrastructure you have in place. Very few organizations perform pen tests against their mainframes, and this is a rare skill for pen testers (although not unheard of). In addition to creativity and curiosity—which are hard to measure but critical for successful testing—the following are some general skills that can apply to most pen testing team members, although specialization in networking or applications is common:

- 3-5 years of operating system knowledge with Linux and Windows.
- 2-5 years of networking knowledge and skills.
- Basic scripting/coding skills are critical for pen testers. Real coding experience in C++ and Java are often invaluable, and basic shell scripting is very useful, too. The most common languages used for penetration testing scripting today are Perl, Python, and Ruby.
- Application development and assessment skills are a must for internal platforms like .NET, PHP, etc. The deeper understanding the team has of the technologies in use, the more effective and efficient application pen tests and code analysis will be.
- Database skills, with a base understanding of SQL syntax, are becoming increasingly important.

This is not an exhaustive list, of course. Many organizations will have very specific platforms and apps that require specialized knowledge. For example, energy utilities will need testers that understand SCADA platforms and applications and how to test them. There are many sources of good training for penetration testing teams. The SANS Institute has a Penetration Testing course and an Advanced Penetration Testing course that both cover a broad range of topics, and also have specific classes on hacking Web applications, wireless, mobile, and other technologies. Offensive Security, a training and pen test services firm, offers several courses that teach students how to perform penetration tests with the popular BackTrack toolkit, exploit wireless networks, and create exploits, among others.

There are many tools to choose from for performing penetration testing.

For enterprises, a number of commercial options are available, including testing suites from Core Security, Rapid7, and SAINT. Open source tools abound, and an excellent starting point for most teams is the aforementioned BackTrack distribution, which is packed with tools that perform scanning, Web application testing, wireless assessment, exploitation, and password attacks. Many specific tools are also included for assessing VoIP and network devices. An excellent Web application-specific toolkit is the Samurai distribution led by SANS instructor Kevin Johnson.

Many organizations have a legal or compliance requirement to have an external party perform at least one penetration test per calendar year.

With any of these tools, the key for internal teams coming up to speed is to practice, and then practice some more. The easiest way to accomplish this is with a test lab, preferably containing actual platforms and applications found in the environment. You should also include a variety of vulnerable testing platforms such as Rapid7's Metasploitable, UltimateLAMP, LAMPSecurity, Damn Vulnerable Web Application (DVWA), OWASP WebGoat, and Mutillidae from Adrian Crenshaw. For testing mobile applications, developers can use OWASP iGoat and GoatDroid and the ExploitMe Mobile Labs for iPhone and Android.

THIRD-PARTY PEN TESTING

Many organizations have a legal or compliance requirement to have an external party perform at least one penetration test per calendar year. In addition to this, it's a good idea to have external firms perform some tests that require extensive knowledge on platforms that your team may not know well or tests your team is not capable of performing for some other reasons.

It's also a very good idea to rotate through several pen testing firms for a variety of reasons. First, you can ensure the firm you are using does not become too comfortable with your environment and its details; performing regular testing could lead to a scenario where the testing firm becomes complacent and misses potential vulnerabilities. Second, you can get a different perspective from a variety of pen testers, each of whom brings his or her own skills and approach to the table. In general, the more eyes you can get on your environment, the more

potential security issues you can find.

Other than these specific cases, performing your own testing is the best way to proceed for all other scenarios. You will be able to test your own environment more often, learn more about the infrastructure, and develop consistent processes for working with IT operations teams to remediate issues. Professional pen testing software companies are creating internal testing tools to allow for team collaboration and consolidated analysis and reporting, as well. Examples include Core INSIGHT from Core Security and Metasploit Pro from Rapid 7.

MEASURING PEN TESTING EFFORTS

The key to a successful penetration testing program is to mimic the actual threats you face, emulating attacker behavior and approaches. While this will give you a more reasonable idea of how you may be vulnerable, the key to proving the value is demonstrating the actual impact to business systems and assets by successfully accessing and exploiting sensitive data. Just telling management that you have successfully compromised systems means very little; showing executives how you have accessed payroll or HR data, payment card information, or intellectual property has a much more definitive impact and drives home the severity of issues in the environment.

What kinds of metrics make sense for penetration testing and vulnerability assessments? For vulnerability assessments, common measurements to track include:

- Number of vulnerabilities found;
- Criticality and types of vulnerabilities;
- Percentage of systems and applications scanned;
- Number of "unowned" or questionable assets detected.

For penetration tests, the key is a baseline:

- How many critical vulnerabilities were found vs. the last test?
- User accounts and/or passwords compromised;
- Data records accessed.

A pen test should demonstrate real value, both by helping discover

vulnerabilities, and potentially making other aspects of the vulnerability management cycle more productive and effective. For example, pen tests can help to cut down on false positives found with vulnerability scanners. Pen tests can also be used as an additional measure to validate application development practices.

Overall, building an internal pen test team is a good idea for many reasons, ranging from false positive validation to compliance requirements. By developing test scenarios that mimic real world threats, and demonstrating true business impact from exploited systems and applications, pen testing teams can help improve the security of your organization dramatically. Most security teams have the skills to get started now and plenty of training is available to fill in the gaps. ■

DAVE SHACKLEFORD *is owner and principal consultant at Voodoo Security, senior vice president of research and CTO at IANS, and a SANS analyst, instructor, and course author. He is a VMware vExpert and has extensive experience designing and configuring secure virtualized infrastructures. He coauthored the first published course on virtualization security for the SANS Institute, serves on the board of directors at the SANS Technology Institute and helps lead the Atlanta chapter of the Cloud Security Alliance. Send comments on this article to feedback@infosecuritymag.com.*

TECHTARGET
275 Grove Street,
Newton, MA
02466
www.techtarget.com

About TechTarget:
TechTarget publishes media for information technology professionals. More than 100 focused Web sites enable quick access to a deep store of news, advice and analysis about the technologies, products and processes crucial to your job. Our live and virtual events give you direct access to independent expert commentary and advice. At IT Knowledge Exchange, our social community, you can get advice and share solutions with peers and experts.