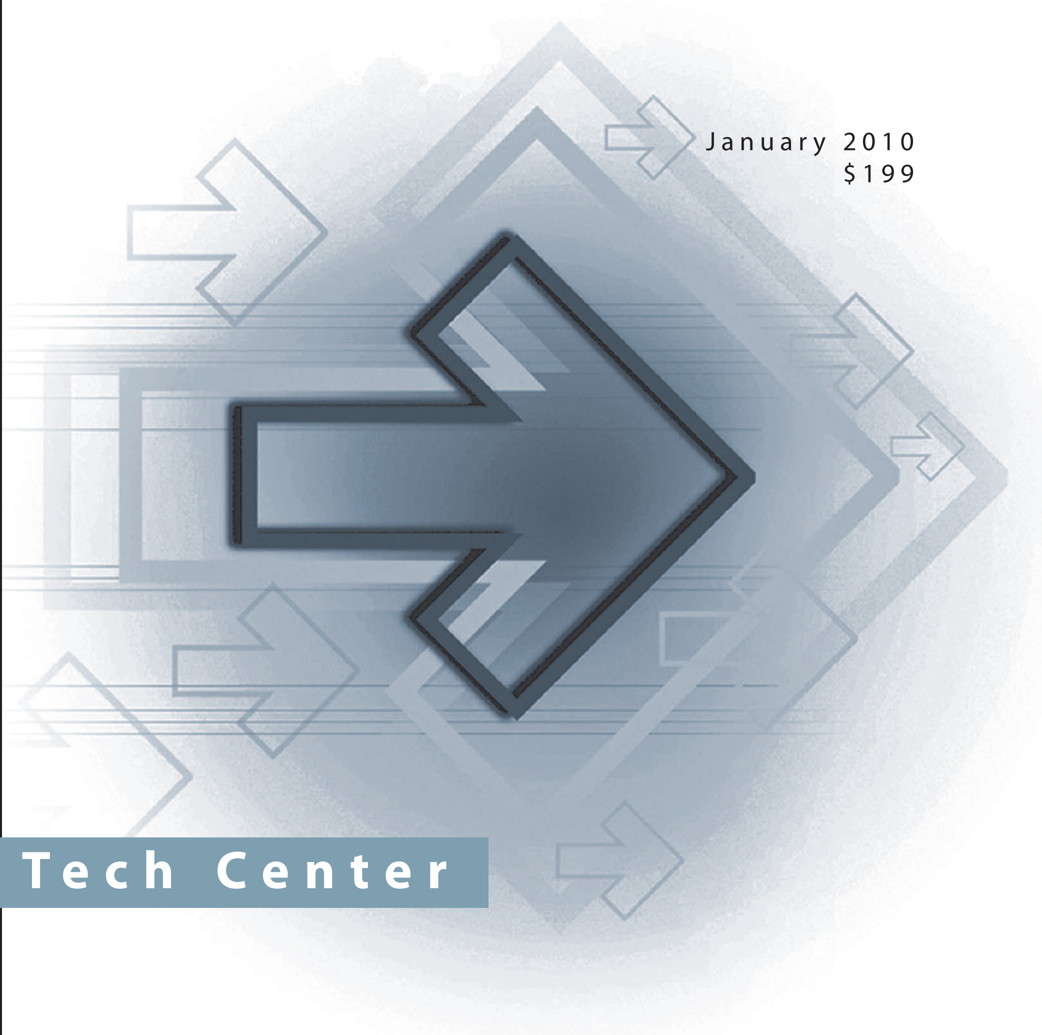
InformationWeek
:: analytics



January 2010
$199

SECURITY
dark READING Tech Center
Protect The Business Enable Access

Contents

Compliance 101: Creating a Strong Vulnerability Management Strategy

Assessing new threats is only the first step in finding and shoring up weak spots in your defenses. Most infosec groups must also factor in a broader audit of compliance with regulatory standards, including HIPAA and PCI. In this **Dark Reading** Tech Center report, we outline best practices for compliance-oriented vulnerability management; discuss helpful technologies; and address the process of mapping compliance requirements to vulnerability detection and remediation.

By Richard Dreger

Richard Dreger
CISSP, CISA, CWNE

Richard Dreger is president of WaveGard, a vendor-neutral security consulting firm. Rick has significant, broad-based technology experience with extensive skills in the information assurance, security and wireless networking fields. He has consulted for a wide breadth of clients in both the public and private sectors, and his professional background includes over 15 years of experience in *Fortune* 100 companies as well as smaller technology consulting firms.

Rick has complemented his hands-on consulting experience by leading courses such as the CWNP wireless curriculum and the (ISC)2 CISSP review. In addition to being one of the 11 founding members of the Certified Wireless Network Experts (CWNE) roundtable, he is also co-author of the *Certified Wireless Security Professional (CWSP) v2* study guide and numerous *InformationWeek* articles. Rick obtained his BSE from Duke University and his Masters from Villanova University.

Catch Us if You Can

It's a high-stakes game of cat and mouse as IT security pros constantly scramble to identify and mitigate vulnerabilities while staying on the right side of compliance auditors. If you're currently feeling more like the mouse in this scenario, then we'll bet you lack a comprehensive vulnerability management strategy.

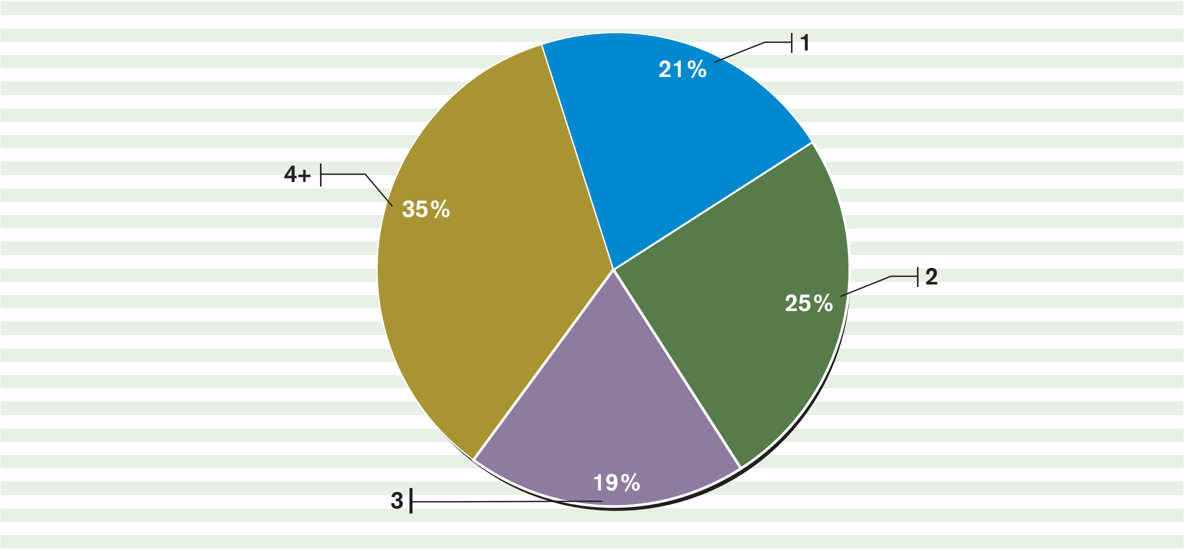
Take heart, you're not alone. In our recent *InformationWeek Analytics* Regulatory Compliance Survey of 379 business technology professionals, when we asked how many different regulatory compliance requirement sets their organizations were currently required and/or choosing to comply with, the No. 1 answer was four or more, at 35%. HIPAA was the most common reg, at 65%; SOX and PCI tied for second, at 51%. However, just 23% of respondents characterized their information security programs as very secure and well documented.

Vulnerability management is essentially the process of testing and vetting your own security controls to not only validate what you *think* you have, but to proactively discover any hereto-

Figure 1

Number of Regs in Play

How many different regulatory compliance requirement sets is your organization currently required and/or choosing to comply with?



Data: *InformationWeek Analytics* Regulatory Compliance Survey of 379 business technology professionals

fore unknown weaknesses. The real value of this procedure—aside from appeasing auditors—is to ensure you're not caught with your pants down (as it were) when some malware hits your network and exploits a vulnerability that *should* have been patched months ago.

It's tougher to explain to upper management what happened if you had absolutely no visibility into the problem to begin with. Ignorance is no defense, as the adage goes.

So how can IT create a comprehensive vulnerability management plan? To crack this nut, we recommend a three-pronged approach that combines strong policies, well-disciplined operational procedures and effective software validation tools. The first two we'll discuss in more detail later on, but we want to update the common view of vulnerability scanning based on what we're seeing in our practice.

The traditional approach is to drop a system on the network, grab a network range, tweak up a few configuration settings and then start scanning away. Once the software is done, a report is generated to provide the next step: a to-do list. Simple enough. The problem, however, always seems to come when the report is actually scrutinized and voluminous action items are being generated—there are just too many false positives. A further annoyance is that if incremental delta scans are not being performed, it can be difficult to determine what has changed in the environment, so time is wasted reanalyzing items that have already been reviewed.

What can be done to address these issues? Tool selection will go a long way toward determining just how valuable your scan results end up being. Look for scanners that:

> Provide the option of **attempting to exploit** discovered vulnerabilities;

> Offer help to quickly **whittle down** the false-positive list;

> **Create baselines** of the environment;

> Can **discover new hosts** on the fly; and

> Enable IT to **configure regularly recurring scans** to minimize the gaps between identifying new vulnerabilities and testing them in the environment.

Why is this last item so important? Some tests can cause serious disruptions to your production environment, so it's essential to ensure that all your bases are covered while minimizing duplicative poking and prodding as much as is humanly possible.

Time to Build the Stool

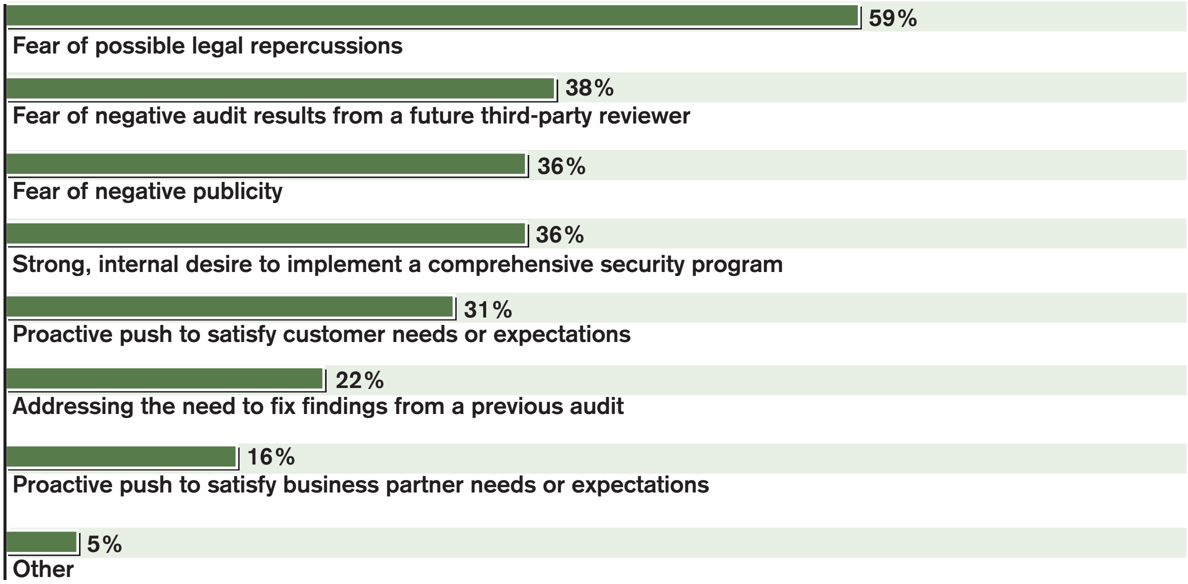
For our purposes, at a gut level, a vulnerability is defined as a security problem or weakness that exists somewhere within the IT infrastructure; it could be any management, operational or technical issue that not only raises our risk level, but has the potential to be exploited, which could then lead to some form of data or system compromise.

The main weapon in IT's unending struggle to stay ahead of the bad guys isn't the hottest new security system. It's a process in which we identify vulnerabilities; rank them in a meaningful way based on business and compliance realities; and then decide whether to accept the risk, mitigate problems with appropriate fixes or offload the risk to a third party. Not sexy, but vital.

Figure 2

Compliance Drivers

What are the primary drivers for pushing compliance initiatives?



Note: Three responses allowed
Data: *InformationWeek Analytics* Regulatory Compliance Survey of 379 business technology professionals

In this Dark Reading Tech Center report, we'll zero in on how to build just such a vulnerability management strategy that integrates tightly with a broader security program and compliance plan. Given the omnipresent specter of proscriptive regulatory requirements such as SOX, HIPAA and PCI, coupled with limited IT resources, we must "think twice and execute once," as it were.

The Program

This is where is all starts—at the security program level. As we have detailed in other articles and papers, there are myriad approaches that can be taken when building an information security management program; however, we like the frameworks provided by ISO and NIST. ISO's 27001/27002 documents lay out an 11-category approach for building an effective, if perhaps overly extensive, information security management system (ISMS). NIST, through its 800-series special publications, covers a very wide range of technology, and in some cases security-centric, topics, such as providing the definitive guideline for certifying and accrediting (C&A) major applications and general support systems for the federal government.

Employing either the ISO or the NIST framework saves an organization from reinventing the wheel; both groups have already invested serious effort in creating the initial baseline. The key is to take the time to abstract these sometimes monolithic documents and tailor their generalized guidance to fit the specific needs of your organization. In this report, we're focusing specifically on the vulnerability management aspect of the security program, so let's see what both ISO and NIST have to say about this subset of the guidance.

As a first step, let's define the environment in which we'll be working. Security controls can be grouped loosely into three broad areas: management, operational and technical. Management controls include topics such as policies and the security posture. Operational controls involve how things are done in production, and technical controls address the more tangible software and/or hardware protections that implement the requirements specified by our policies. In practice, all three of these areas are required for a complete vulnerability management strategy. Let's look at each in more depth:

Management: It's essential that upper management understand the risks that face key IT systems and connect the dots on how vulnerabilities increase that overall risk profile. Of course, this will be easier or harder depending on your organization.

“A risk analysis is the first thing that must be addressed. You need to understand exactly what the probability is of issues happening that could impact your operation,” says Robert A. Clines, former CIO of Dallas County. “When working with long-existing government institutions, it’s hard to convince elected officials that there is a chance that things could go south. When the cost is identified for the mitigation of that risk, they are not inclined to finance it. ‘This has never happened in the 160 years that we have been around; it will not happen now.’ Mitigation of risk is just insurance and has no tangible benefit unless it is used.”

We’re certain that mindset isn’t limited to the public sector. In our survey, we asked about the level of financial, personnel or other resources being made available to address compliance needs. Despite the high level of regulatory adherence required of our poll respondents, just 20% say they’re getting significant support. The best way to remedy this is by promoting business involvement at every turn; since conducting regular risk assessments is a critical part of identifying vulnerabilities and quickly eradicating them, upper management should be involved in creating, supporting and incrementally enhancing strong IT security policies that promote a risk-based approach to protecting data.

Operational: Constant vigilance is the cost for securing our dynamic IT environments. Vulnerabilities can be subtle and are often accidentally injected into the network when IT makes minor changes or installs incremental patches. By following specific procedures and policies for maintaining, testing and approving systems prior to going live in production, these vulnerabilities can be minimized. Follow this up with regular vulnerability testing and validation, and you’ll be rewarded with a fairly effective strategy.

Technical: This is where the more tangible vulnerability management work is done. Would-be attackers use vulnerability scanning, red-teaming, exploit attempts and other techniques to poke and prod your systems. Conversely, we can employ these methods to see how well our defensive controls are securing the organization. Of course, these controls, while providing the lion’s share of protection, must exist in an environment that also has strong management and operational controls to be truly effective.

Regardless of the specific security program or regulatory requirements that you might be dealing with for vulnerability management, you will likely need to address each of the three facets described above. Now let’s discuss how to tailor your approach to closely map to specific tenets of popular regulatory requirements.

Mapping Compliance Requirements

Given the wide range of interpretations for how vulnerability management might proceed, let's start by discussing how both ISO and NIST describe this process before moving into specific regulations, like HIPAA, PCI and SOX.

While ISO 27002 does not consider vulnerability management as one of its 11 main focus areas, it does explicitly include Technical Vulnerability Management as a subset of the Information System Acquisition, Development and Maintenance section. There is also significant discussion about creating a risk-based approach to security that relies on thoroughly understanding the underlying vulnerability picture. Similarly, numerous NIST SPs focus on risk management; for example, 800-37 states: "Part of risk management incorporates threat and vulnerability analyses, and considers mitigations provided by security controls planned or in place."

Figure 3

Regulatory Perspective on Vulnerability Management

	HIPAA	PCI
Vulnerability Language?	"Identify Potential Vulnerabilities" listed as a key part of the Risk Assessment approach from the NIST 800-66 HIPAA Resource Guide.	Very explicit. The PCI-DSS actually uses the phrase "Vulnerability Management Program."
Approach	Vulnerability language is strongly integrated with the risk assessment approach and takes a more traditional view of scanning, mitigating and keeping on top of new threats when specifically discussing vulnerabilities.	The PCI DSS is much more explicit in its definition of specific control areas and not only mentions technical fixes, such as antivirus, but also stresses good change control and more policy-driven protections.
Stated requirements for vulnerability management	> Vulnerability scans > System security tests > Reviewing vulnerability databases > Regular scheduled vulnerability testing	> Antivirus > Software patching > New vulnerability notifications > Good development environment practices > Change control > Strong coding for applications > Code review > Application-layer firewalling

So, what we are seeing is that vulnerability management is intimately linked with risk management. The upside to this approach is that a risk-based program ranks problems in the environment using concepts such as the potential impact of a compromise occurring to a system and the likelihood of such a problem occurring. Based on defined risks, our finite resources can then be smartly parceled out to address the biggest issues first, and so on. While this clearly makes sense at a conceptual level, we also need to understand how various regulations take this higher-level risk/vulnerability concept and inject more specific drivers.

After reviewing the regulatory perspective on vulnerability management, we come somewhat full circle—regardless of which reg an enterprise is seeking to address, IT must have a comprehensive, risk-based approach to managing security. This approach, regardless of the actual structure selected, must include strong supporting policies, some form of regular scanning for validation and ongoing control enhancements to fix identified weaknesses.

Before we close the discussion on the policy facet of the vulnerability management program and move on to tool-based technical controls, let's lay out a few best-practice recommendations. The chart, below, discusses some of the key considerations regarding relevant policies and procedures. This is not an exhaustive list, but we touch on the top items to review.

Figure 4

Basic Vulnerability Management To-Do List

Topic	Recommendations
Policies	> Security policy (using a risk-based approach) > Change control policy > System development lifecycle documents (software/applications) > Data classification & handling > User awareness training policy
Procedures	> Risk assessment procedures > Change control procedures > Vulnerability scanning process (identification, mitigation and management) > System build standards > Patch management process
Reference documents	> **ISO:** *http://www.iso.org/iso/home.htm* > **PCI-DSS:** *https://www.pcisecuritystandards.org/index.shtml* > **NIST Special Publications:** *http://csrc.nist.gov/publications/PubsSPs.html*

Creating a strong security program, including supporting documents for managing vulnerabilities, all seems logical enough, but how can we put some rubber on the road and make progress creating a tangible vulnerability management program? In the next section, we'll discuss defining testing suites and technical control validation.

Gear Up

Now (finally!) it's time to focus on the vulnerability assessment itself. When we think about vulnerability assessment, the first image that comes to mind is network scanning. Scanning is typically the process of utilizing a network-connected device running highly customized security software to identify and probe systems. This approach is commonplace and provides a solid way of validating systems and devices to see if they might be vulnerable to attack. Popular commercial products in the marketplace include those from Nessus, Qualys, Retina, SAINT and many others.

Keep in mind, however, that any time a system is built, hardened and tested, it represents a "snapshot in time" for that device. New zero-day vulnerabilities are constantly coming online that can make an otherwise well-secured system highly at risk literally overnight. This is one reason why regular, ongoing vulnerability testing and validation should be conducted to try and shorten the window between new exposures and implementing fixes.

Now it's time for you to think like an auditor and create a testing strategy that will mirror the logic used by outside testers. If we can create a robust vulnerability management strategy that supersedes what we might be tested on, we know that the majority of our security problems will be proactively found and our systems will at least be reasonably protected. Keep in mind that auditors are generally trying to ensure that due diligence is being performed and that reasonable controls have been implemented. If we can provide a strong, documented security program, backed by robust technical controls and a regularly exercised security testing and evaluation plan, we should have nothing to worry about during an audit.

We recommend a standardized approach for network scanning that includes:

Preparation: Before conducting any type of potentially invasive scan, proper preparations must be made. For consultants working at a client site, a rules of engagement (ROE) document must be drawn up that outlines the types of testing to be conducted and the proposed targets. This document helps to square the understanding between the client and the consultant and must be signed by the client prior to any hands-on work being performed. For in-house self-testing, an approved policy and procedure set should be established to ensure that all responsible parties know when testing will be occurring and on what systems. This helps to coordinate alerting teams and minimize false alarms. Also, if a problem does arise, such as the corruption of a server, it is essential to demonstrate that you followed an approved process.

Initial tool configuration: This is where parameters for the test are established. Although specifics will vary according to product, common options include the depth of testing to be conducted (light to exhaustive), TCP/UDP ports to scan, username/passwords for authenticated scans and other performance settings. These settings help determine exactly what the tool is going to be doing for the tester and how it is going to perform.

Discovery: Once testing parameters are decided on and traffic is ready to begin traversing the network, targets must be identified and selected. Scanning tools allow for IT to inputt specific network ranges, host names or IP addresses when there is prior knowledge about the desired targets. Devices can also be discovered using either ICMP (ping) or a UDP/TCP frame to identify responding hosts that are "alive" and potentially reachable.

Once discovery is complete, more extensive probing and testing can occur.

Port discovery: Now that we have our target list, we seek to profile the hosts and see what ports they might be listening on. This process will give us some insight into the services and daemons that are running and set the stage for even deeper testing. For example, if 80/TCP is open, then we likely have a Web server, and if port 23/TCP is listening, then we may have a telnet service we can try to interact with. By proceeding through the port scan, we can define exactly what is running on the different hosts. Note that, depending on the time and depth of access available, either a subset of the most common ports can be scanned or a full, exhaustive scan of all 65,535 ports for both TCP and UDP could be checked. Or, somewhere in between.

More invasive testing: So our initial checks have been performed, and we now have targets and available ports. If we just stop here, we have some useful data, but very limited vulnerability information. We have no idea if the services that we've discovered are potentially exploitable and could raise risk levels. Thus, the next step is to probe more deeply to understand what's running on the various open ports, try to discover possible version information and gather even more data to profile our targets. Depending on the scan level, a variety of potential findings can be given relative risk levels to help sort the output.

Note that this is typically the level at which vulnerability scanning stops. The scanning team, whether this is IT or an external audit group, is provided with a list of services, and we have more detailed data on potentially exploitable problems. The output and subsequent reports are useful in developing a mitigation strategy, particularly if easily fixable flaws such as missing patches or old code versions are discovered. The problem is that this level of testing stops short of actually validating the findings, meaning that there will be many false positives that need to be followed up on and then later discarded.

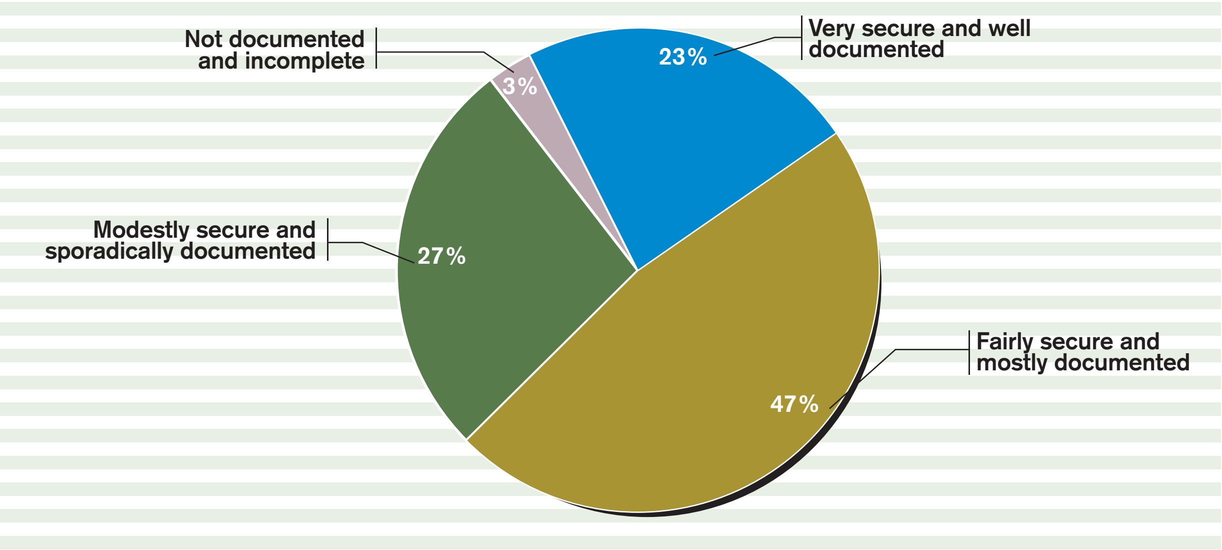
To get even more useful information, consider conducting subsequent penetration testing and exploit testing.

Penetration testing: The art of penetration testing revolves around the concept of exploiting identified vulnerabilities to see if the targets can be compromised to gain access or data. Targeted pen testing is a common next step after host mapping and vulnerability scanning has been completed. Essentially, what happens is that the output of the probing reveals a set of

Figure 5

Security Program Maturity

What is the maturity level of your information security program?



Data: *InformationWeek Analytics* Regulatory Compliance Survey of 379 business technology professionals

services/applications that likely have known vulnerabilities or an ability to be manipulated. More sophisticated toolsets (commercial or open source) and knowledge of how various software components work can be used to gain progressively more information and access into vulnerable systems, until full root (administrator) level access is achieved. Tools that have the option of performing penetration testing and exploits with the click of a button include products such as SAINT scanner and CORE-IMPACT; such offerings provide testers with a relatively straightforward way of performing fairly thorough penetration testing on target systems.

Of course, these extra features do come with a licensing cost, and pen tests can be significantly time consuming to conduct, particularly if customized tests or home-grown tools are being run. But they also provide a better assurance level that a given system is being adequately protected.

Reporting: The ultimate deliverable of the vulnerability scan is an actionable report that summarizes the activities that have been conducted and lists results in a digestible format. The style of these reports will vary, but essentially, each lists the target devices and the number of vulnerabilities that were discovered in rank order (critical, high, moderate, low). Additional, more detailed information is then provided later in the report to describe in more depth exactly what the findings were and suggest some proposed remediation to fix the issues—for example, upgrade the application or patch the OS. Reports are essential for quickly identifying problems, directing resources and getting a handle on progress.

Next Steps

Our goal has been to lay out the key facets of a vulnerability management strategy. For most shops, a three-pronged technique that blends strong policies with disciplined operations and the liberal application of testing tools will minimize vulnerabilities and help IT proactively identify new areas of weakness—and avoid being surprised by the results of compliance audits.

Complacency is the enemy; enterprises must consider vulnerability management an ongoing battle. Strategies should constantly evolve to encompass the regular validation of systems and the use of progressively smarter tools. A high level of security is within reach, however, since a broad suite of testing tools is readily available to those with the right expertise and, arguably, a big enough budget. In the end, building an effective vulnerability management strategy is an essential component of the overall security program and a core piece of managing risk throughout the enterprise.