

Realtime  
publishers

"Leading the Conversation"

# The Essentials Series

# Messaging & Web Security

*by Dan Sullivan*

---

Article 1: Botnet Threats.....	1
Botnets at Your Service .....	1
Botnet Building Blocks.....	2
Combating Botnets.....	3
What Can You Learn from Biology?.....	3
New Defenses .....	4
Article 2: Combating Spam .....	5
Spam Detection and Blocking Techniques.....	5
Integrity Analysis.....	5
Heuristic Analysis.....	5
Content Filtering.....	5
URL Filtering.....	6
Content Scanning.....	6
Domain Name Reputation.....	6
Blacklists and Whitelists.....	6
Self-Tuning .....	7
Responding to Shifting Tactics.....	7
Preventing Your Resources from Being Used by Spammers .....	7
Article 3: Content-Filtering Technologies .....	8
URL Blocking.....	8
Content Scanning .....	9
Bayesian Filtering .....	10
Collaborative Filtering.....	10
Policy Controls.....	10
Summary .....	11
Article 4: Messaging Security: Defense in Depth (and Breadth) .....	12
Technologies for Messaging Defense in Depth and Breadth.....	13
Security Management Practices for Messaging Security.....	15
Article 5: Email Authentication .....	16
To Trust or Not To Trust .....	16
Sender ID Framework.....	17
DomainKeys Identified Mail .....	18
Moving to a Trust-But-Verify Model .....	18

---

Article 6: Email Compliance and Regulations.....	19
Major Regulations Affecting Email.....	19
Impact on Email Management Objectives.....	20
Email Security.....	21
Email Policies.....	21
Monitoring and Auditing Email Operations.....	22
Archiving Messages.....	22
Summary.....	22
Article 7: Essential Policies for Messaging Security.....	23
Risk Tolerance and Messaging Policies.....	24
User-Centric Policies.....	25
Implementation Policies.....	25
Summary.....	26
Article 8: FISMA and Messaging Security.....	27
Risk Assessment.....	28
External Threats to Messaging.....	28
Internal Threats.....	28
Messaging Policies and Procedures.....	29
Identification and Authentication.....	29
Awareness Training.....	29
Contingency Planning.....	30
Auditing and Accountability.....	30
Summary.....	30
Article 9: Host Intrusion Detection and Prevention.....	31
Threats to Hosts.....	31
Data Theft.....	31
Data Tampering.....	32
Loss of Control.....	32
DoS Attacks.....	32
Detection Techniques.....	33
Limitations of Host IPSs.....	33
Article 10: Instant Messaging Worms and Other IM Threats.....	34
Examples of IM Malware.....	34

General Principles of IM Threats.....	35
Tips on Controlling IM Malware.....	36
Summary.....	36
Article 11: IT Audits: What to Expect.....	37
In the Beginning: SAS 94.....	38
COSO and COBIT Frameworks.....	38
Steps to the IT Audit Process.....	39
Article 12: Physical and Digital Security Convergence.....	41
Business Drivers.....	41
Benefits of Consolidated Physical and Digital Security Management.....	42
Industry Standards Supporting Convergence.....	43
Article 13: Rootkit Challenges.....	44
Types of Rootkits.....	44
Application-Level Rootkits.....	44
Library Rootkits.....	45
Kernel Rootkits.....	45
Virtualized Rootkits.....	45
Firmware-Based Rootkits.....	46
Blocking and Removing Rootkits.....	46
Article 14: Vulnerability Scanning 101.....	47
Purpose of Vulnerability Scanners.....	47
How Do Vulnerability Scanners Relate to Other Security Tools?.....	48
Packet Sniffers.....	48
Port Scanners.....	49
Intrusion Detection and Prevention.....	49
Architectural Options.....	49
Article 15: Web Application Testing.....	50
Watch Out for Top Vulnerabilities.....	51
Tools for Testing.....	51
Testing Procedures and Methodology.....	52
Summary.....	52
Article 16: Web Services Security.....	53
SAML.....	53

---

Authentication.....	54
Authorizations.....	54
Attributes.....	54
WS-Security .....	54
WS-Trust.....	55
Article 17: When Patching Is Not Enough: Zero-Day Threats.....	56
What Kinds of Zero-Day Threats Have Occurred? .....	56
Reducing the Threats of Zero-Day Exploits in the Short Term.....	57
Reducing the Threats of Zero-Day Exploits in the Long Term .....	58
Article 18: Configuration Management and Security.....	60
Hardening Systems .....	61
System Settings.....	62
Access Controls .....	62
Authorized Hardware and Software.....	62
Configuration Auditing.....	63

---

## Article 1: Botnet Threats


Cybercrime seems to continue on its evolution toward greater sophistication in techniques. You see it in higher rates of infections of Trojan horses, more targeted phishing attacks (aka *spear phishing*), and perhaps most disturbingly, the increasingly robust nature of botnets. Bots are tools for taking control of computers, and a collection of computers compromised by bots is known as a botnet. The botnet herder manages the botnet and issues commands directing the operation of the botnet. The herder may or may not be the person who originally created the botnet—it seems cybercriminals are not above stealing botnets from one another (no honor among thieves).

### Botnets at Your Service

So what good is a collection of compromised machines? Botnets may be used for:

- Conducting distributed denial of service (DDoS) attacks
- Distributing spam
- Launching phishing attacks
- Conducting click fraud
- Stealing personal information

DoS attacks are becoming a dominant problem for Internet service providers (ISPs). For example, *Info Security* magazine is reporting that DoS attacks have become the top threat, demanding the most resources to combat (Source: [http://www.infosecurity-magazine.com/news/060915\\_network\\_operator\\_attacks.htm](http://www.infosecurity-magazine.com/news/060915_network_operator_attacks.htm)). When one of the ISP's customers is targeted for attack, all the customers suffer. One quick fix is to redirect all traffic sent to victim's address, both the legitimate and the attack traffic. The problem, of course, is that the victim is knocked offline and the attackers succeed in their attempt. Rather than simply throwing the victim of the attack overboard, the targeted servers could be put behind another server that can scan each packet and determine whether the packet is part of a DDoS or is legitimate traffic. Such a server would have to be able to handle high volumes of traffic.

 For details about a DDoS attack and the various responses, see Scott Berinato's "Attack of the Bots" ([http://www.wired.com/wired/archive/14.11/botnet\\_pr.html](http://www.wired.com/wired/archive/14.11/botnet_pr.html)), which chronicles an attack on one company's site.

---

The majority of spam, more than 70 percent by some estimates, is now originating from botnets. Spammers simply rent botnets on an as-needed basis. This situation is analogous to just-in-time inventory for manufacturers and distributors; there is no need to maintain assets for extended periods of time when you can get them only when needed.

Click fraud is another method for financial gain from botnets. The perpetrator sets up Web sites that serve pay-per-click ads, then commands the members of the botnet to visit the site and simulate clicks on the ads. This is a serious problem for advertising services such as Google and Yahoo!. In fact, Google agreed to pay a \$90 million settlement in a class action suit against the firm for charging for illegal clicks. The company did so despite the finding by an independent expert that Google's four-layer defense against click fraud—which includes pre-filtering, online filtering, automated offline detection, and manual offline detection—is reasonable. Botnets are hitting more than major Internet companies.

The theft of personal information is another target of botnets. Once installed on a device, botnets can download keyloggers and capture all keystrokes, which can then be scanned for the names or URLs of banks, online auctions, stock brokerages, and other financial services firms. From there, it is a simple matter to scan for text of usernames, passwords, account numbers, Social Security numbers, and other useful personal information.

As you can see from these examples, botnets are general-purpose tools for cybercrime. Bot software is not like other malware, such as viruses, worms, and keyloggers, although botnets can be used to distribute those programs. It is botnets rather generic nature that makes them so adaptable for different tasks.

## Botnet Building Blocks

As previously stated, a botnet is a set of compromised devices that respond to commands from a botnet herder. Although there are different kinds of botnets, they have some common characteristics. First, they communicate with the herder. Internet Relay Chat (IRC) servers and protocols are commonly used for communications. IRC servers are easy to set up, and the command set is more than sufficient for controlling bots. Some bots may use other communications channels, such as AOL chat rooms, but the principle is the same. Once a bot is established on a compromised machine, the bot establishes communication so that it can receive commands.

Second, they try to find other hosts to compromise. Just as worms and viruses will propagate on their own, a bot program is designed to duplicate and spread. One way to spread is to exploit vulnerabilities of commonly used protocols, such as NetBIOS Name Service, NetBIOS Session Service, and Remote Procedure Call (RPC) services. The herder may control the rate at which the bots spread by issuing commands and setting configuration parameters.

Third, bots have the ability to download files using ftp, http, or other protocols. This allows herders to download updates, send instructions, and receive data back from bots (especially useful for information theft operations). Bots may also make use of other malicious programs. For example, bots may spread as payload on viruses and worms. They can also use rootkits to disguise their presence on compromised machines.

---

## Combating Botnets

Botnets are difficult to combat because they are highly distributed and the command and control structures are flexible. Botnets with centralized command structures can be shut down temporarily by shutting down the command and control (C&C) server. C&C servers are easy to move and can be back online in relatively short times. And the fact that the new server will have a different IP address is no problem—bots can reference a dynamic DNS name. The herder just has to update the domains entry in the dynamic DNS database, which can be done in real time. (Dynamic DNS was developed so that machines with dynamic IP addresses could still be used as servers.)

Some botnets do not even use a centralized C&C model; instead, they use a peer-to-peer model that allows herders to send a command to one bot, and it is distributed to other bots in the botnet. This eliminates the single point of failure in the bot network.

So what you have is a highly distributed, self-propagating network that uses the common infrastructure of the Internet. How do you combat this? Information security has appropriately used analogies from biology when describing viruses and worms, and the botnet situation is just as likely to find some enlightenment from the life sciences.

## What Can You Learn from Biology?

Organisms have evolved over eons and have created a wide array of defense mechanisms. Some organisms are strong and mobile while others, especially plants, are physically limited and immobile. Yet both have survived and thrived in a hostile environment. You can learn a few things from nature.

First, you need multiple lines of defense. Animal skin is the first line of defense; you do not have to battle a pathogen if it cannot get into your system. In the world of computers and networking, the first line of defense is a perimeter security mechanism such as a firewall, content filter, and intrusion prevention system (IPS). These systems have moved within the network, and now everyone should be running a personal firewall and antivirus applications. There are plenty of perimeter defenses, but we need to keep improving them. The more that is detected and blocked at the perimeter, the less that has to be dealt with by secondary defenses.

Higher organisms employ a host of biochemical defenses to combat internal threats. Antibodies in animals detect foreign objects and respond with a complex mechanism that ultimately destroys the compromising agent. Plants have evolved to produce a wide array of proteins that can protect the organism under different stress conditions. We currently lack this kind of secondary defense in information security—and that needs to change.

---

## New Defenses

New defenses are needed if we are to maintain control of our computing and networking resources. For starters we need:

- Better monitoring of botnet activity—Projects such as the Honeynet Project (<http://www.honeynet.org/>) have created tools and conducted studies that provide information about the form and function of botnets. We need more.
- Distributed countermeasures—We can no longer expect to shut down a botnet by shutting down a centralized C&C server. We need better tools to identify and eradicate bots in the wild. Antibodies are a good analogy of what we need.
- More secure operating systems (OSs)—Of course, there will always be a way to breach a home computer, but at least we could make it somewhat of a challenge for an attacker.
- Better user education—No one expects to drive without a license and no one should have a driver's license without a basic understanding of car operations and rules of the road. Once you connect to the Internet, you are participating in a shared resource and that entails responsibilities.

It does not take long browsing through articles, blogs, and discussion threads by information security professionals to detect the sense of frustration and malaise about our current abilities to contain cybercrime. The assessment is justified but the problem will not just go away—we are the ones that need to do something about it. The challenges are daunting and we will witness fundamental changes in things ranging from Internet protocols to user behavior before we can even begin to think about winning a few rounds.

---

## Article 2: Combating Spam

Spam, or unwanted and unsolicited email, wastes the time and resources of the recipient but is profitable enough for the sender that this practice still thrives. Spamming, like cybercrime, is driven by economic factors. The low cost of spamming, especially when using compromised hosts in botnets to distribute the messages, makes the endeavor profitable at very low response rates. Given the market incentives for spamming, you can expect it to continue and grow. So if it is not going away, what can you do to minimize its impact?

### Spam Detection and Blocking Techniques

Several spam detection and blocking techniques are available, and they are best used in combination. Some techniques work to identify sources of spam and prevent messages from ever reaching their recipients. Others work by analyzing the content of messages and identifying likely spam messages; still other techniques work by preventing recipients from accessing spam-related sites.

#### *Integrity Analysis*

Integrity analysis examines the structure of a message looking for tell-tale signs of spam such as:

- Invalid headers
- Suspicious time stamps
- Invalid time zones
- Text patterns indicative of spam, such as opening text in upper case

Integrity analysis depends on common patterns related to the structure of the message, but the same principles can apply to detecting spam by looking at patterns in the content of spam.

#### *Heuristic Analysis*

Heuristic analysis uses rule of thumb about content to identify spam. For example, a message with phrases such as “Free offer” or “Act now to save” are more likely to be spam than messages that do not have those phrases. The presence of HTML graphics near bold text is also commonly seen in spam messages. The challenge with heuristic analysis is crafting the rules in such a way that spam is blocked without blocking legitimate messages.

It is highly unlikely that any single rule could reliably detect spam without also falsely categorizing legitimate messages as spam. Instead, groups of rules are used and each can contribute to an overall measure of the likelihood that a message is spam. For example, one rule might say “If the word free appears in the subject line add 2.0 to the spam score,” while another says “If upper-case text appears just before an HTML image, add 1.0 to the spam score.” If the spam score exceeds a threshold, the message is categorized as spam.

#### *Content Filtering*

Content filtering, as the term is typically used, includes two types of technologies for blocking content: URL filtering and content analysis. Both types depend on a network configuration in which all traffic being monitored passes through a content-filtering device.

---

## URL Filtering

URL filtering prevents access to Internet content based on the URL of the site. These systems use large databases of millions of Web sites that have been categorized according to their content, such as shopping, social network, adult, gambling, and so on. These applications are best known for controlling Web browsing and blocking inappropriate material from coming into an organization from banned sites. This technology is also useful in mitigating the adverse effects of spam. If a user were to respond to an offer in a spam message, perhaps to a shopping offer or an invitation to a gambling site, the URL filter could block the access.

URL filtering has a number of advantages, including the ability to block all access to well-known but banned sites. Vendors can continually update their databases and distribute updates to clients, but these measures are not always sufficient to keep up with spammers.

## Content Scanning

In addition to blocking based on a URL, anti-spam applications can filter based on the content of the message. Unlike heuristic filtering, which uses complex sets of rules for identifying spam, content scanning checks for banned or suspicious words and phrases. Another technique that can preempt spam from making it to the more content-oriented filters is domain name reputation.

### ***Domain Name Reputation***

Spammers will change domain names frequently to avoid being blocked by URL filters. To counter the frequent changes, anti-spam researchers have developed techniques that analyze network and server information to determine whether a newly registered domain is likely to generate spam.

Domain name reputation analysis uses a combination of domain registration information and IP reputation information. By analyzing this information shortly after a domain is registered, blacklists can be updated, ideally, before the spammer has a chance to start spamming.

### ***Blacklists and Whitelists***

Blacklists are databases of known sources of spam, malware, phishing attacks, and other unwanted content and programs. As it is unlikely that any legitimate email will originate from one of these sites, all content from them can be blocked rapidly without having to execute the more computationally demanding filters, such as heuristic analysis.

Whitelists are known sources of legitimate email. Once a message is determined to be from a site on the whitelist, the message can be sent through without any further processing.

---

## **Self-Tuning**

Self-tuning is the process of adjusting classification rules based on the particular needs of a user or organization. For example, if a number of spam messages are not correctly categorized, a user or email administrator may identify those messages as spam and submit them for analysis by the anti-spam program. The program then uses those as examples of spam and adjusts the categorization parameters accordingly.

Bayesian classification algorithms are often used for this type of tuning for their speed and accuracy. Of course, spammers will try to slip past Bayesian filters, too. Some of the tricks they have used include adding long strings of random text or inserting randomly selected words into the message. Of course, this kind of pattern can then be detected by other anti-spam techniques such as heuristic analysis. No one technique may be able to detect all spam, but a combination of techniques can be quite effective at keeping users' inboxes under control (or at least virtually spam-free).

## **Responding to Shifting Tactics**

These techniques can be quite effective in combating spam. Unfortunately, the spammers adjust their techniques accordingly. For example, once a domain name has been added to a blacklist, the utility of that domain name is lost; anti-spam filters easily block messages referencing a site in that domain. In response, spammers have resorted to high volume mailings over relatively short periods of time in an attempt to beat the clock on detection and blocking.

Of course, anti-spam developers respond in kind. Anti-spam systems now use real-time behavior analysis to monitor email traffic and identify the IP addresses of machines that are conducting spamming operations. As long as customers' databases of blocked IP addresses are updated frequently (in minutes, not hours), this technique can be used against botnets that rapidly shift the source of spam messages.

## **Preventing Your Resources from Being Used by Spammers**

In the early days of spamming, it was the responsibility of email administrators to keep their email servers locked down. One of the most important steps was not to provide an open relay. An open relay allows spammers to send email through your server and hide the true source of the message.

The various techniques described earlier can effectively block spam from open relay servers, and the number of open relays is far less than the number of potentially compromised PCs. This fact is not lost on spammers who are now using botnets to generate spam. In addition to blocking spam coming into a network, take steps to prevent it from generating from within the network by using antivirus solutions, personal firewalls, and content filtering, and keep clients up to date on operating system (OS) and browser patches.


---

## Article 3: Content-Filtering Technologies

Content filtering is a term that is often used to refer to a collection of technologies that can be applied at client, server, and network levels. This article discusses the various types of content-filtering technologies and their uses. The technologies addressed include:

- URL blocking
- Content scanning
- Bayesian filtering
- Collaborative filtering

These are complementary and sometimes overlapping technologies, but they are distinct approaches to the problem of blocking unwanted content.

 You might hear that one product or another is a “true” content-filtering technology because it blocks based on phrases in content and not just URLs. One can argue about whether a particular technology belongs in this category of security tools, but, frankly, outside of product marketing, this is not a useful distinction. The term content filtering is commonly applied to software and appliances that use multiple techniques. This article will assume the broader definition of the term.

### URL Blocking

URL blocking, as the name implies, is a technique for disrupting the use of a Web resource based on its address or URL. Pattern-matching techniques are used to detect parts of the URL that indicate a banned site, for example:

- `http://*.serveadsonline*`
- `http://*.*casino*.com`
- `http://*.gaming*.*`

The first pattern would block traffic to and from any domain that starts with “serveadsonline;” the second pattern would block traffic to and from any .com sites with the word “casino” in the domain name; the final pattern blocks traffic with domain names that start with the term “gaming.”

---

There are limits with URL blocking. First, URLs are easy to change, so once a domain becomes blocked at enough sites, it can simply be switched to another domain name. Using real-time databases of categorized URLs solves this problem. Content-filtering vendors maintain substantial databases of Web sites that are continually updated. Automated techniques may be used to categorize the content of sites tracked in the database. This setup allows administrators to specify entire categories of content to block, for example, all gambling, adult, or hate speech sites.

Second, there is the problem of false positives—sites that are blocked but should not be. This occurs when a pattern is too general and can be addressed with more specific patterns. The most specific patterns identify particular sites.

Of course, when using URL blocking, the system you choose should block corresponding IP addresses as well as URLs. IP addresses are easily obtained using ping or other network utilities.

Proxy-based URL blocking is not easily bypassed, unlike some client-side filters used commonly in home or small office environments. When client-side URL blocking is employed, access to the registry and other administrative functions must be strictly controlled.

## Content Scanning

Content-scanning techniques examine the content within network traffic as it pass through a proxy device. These devices are usually just inside the firewall and can be configured to scan for multiple threats, including

- Inappropriate content
- Malware
- Spyware
- Spam
- Phishing lures

Content scanning complements URL filtering by providing a secondary defense. URL filtering blocks known banned sites, content scanning blocks content that is considered harmful that originates from non-banned sites.

Content-scanning techniques include searching for phrases, filtering on Platform for Content Selection (PICS) metadata, filtering on file extensions and MIME. For example, content filters often block executable images based on file extensions.

 For more information about PICS, see <http://www.w3.org/2000/03/PICS-FAQ/>. For details about MIME media types, see <http://www.iana.org/assignments/media-types/>.

Broad-range content filtering also includes malware, spyware, and spam filtering. Malware and spyware may be identified with signature-based detection techniques; spam is often detected based on common patterns within spam. Bayesian filtering is an efficient and effective method for this type of detection.

---

## Bayesian Filtering

Bayesian filtering is a categorization technique that uses positive and negative examples. For instance, email clients now allow users to indicate some messages are spam while others that may have been categorized as spam are actually legitimate messages. Bayesian techniques use patterns in both negative and positive examples of spam to assess messages. If a message shares a common characteristic of positive examples, the message is likely spam; otherwise it is not.

Spammers will often try to throw off these kinds of filters by adding irrelevant, random text, but heuristic analysis can compensate for some of these techniques. With these techniques, rules can identify irrelevant text (for example, a large number of characters near the end of an email) and remove them from consideration by the Bayesian filter. The result is a highly effective method for blocking spam.

## Collaborative Filtering

Collaborative filtering is a technique that uses a database of “votes” to categorize potentially unwanted content. The technique works in the same manner as the recommendation and rating system common in commerce sites.

When a user receives a spam message and the user categorizes it as spam, the information is retained locally in non-collaborative solutions, such as personal email filters. In collaborative applications, the categorization information is also sent to a centralized database that essentially tallies the number of times the message was classified as spam.

To be effective, collaborative filtering needs an efficient representation of messages that accounts for small random variations used in spamming. Consider two spam messages with the subject line “Great Interest Rates Now!” and “Great Interest Rates Now!!” Assuming the body of the message is exactly the same, a simple message digest function would render two different values because one has a single exclamation mark and the other has two exclamation marks. Something akin to a fingerprinting method that takes into account several features is a better option for representing spam in a collaborative-filtering system.

## Policy Controls

Content filtering is best done with a combination of techniques. Even within a single security device, such as a content-filtering appliance, a defense-in-depth approach is best. Given the combination of techniques and the varying requirements across organizations, it is not surprising that content-filtering applications and appliances are configured with user defined policies.

Policies are useful for defining:

- What categories of URLs to block
- How to filter based on PICS metadata
- How to quarantine malware-infected messages
- What protocols to filter
- How to report on content-filtering activities

---

## Summary

Content filters serve multiple functions. Although the term has grown to include several distinct techniques, they complement each other and serve a common objective: blocking unwanted traffic based on the content of that traffic.

---

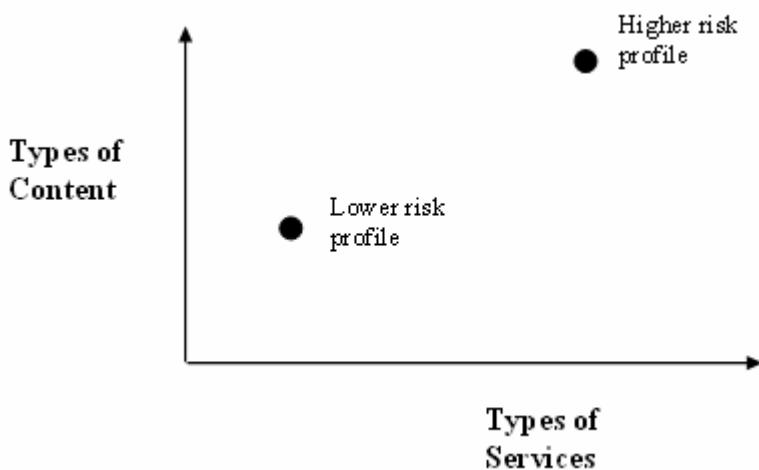
## Article 4: Messaging Security: Defense in Depth (and Breadth)

Defense in depth is a phrase you often hear when discussing information security. The principle behind it is sound: do not depend upon one mechanism or practice to secure a system. It is also insufficient for protecting the integrity of messaging security. In addition to the depth of defenses, you must think in terms of the breadth of defenses.

Consider a simple example that justifies the need for defense in depth: A firewall has been put in place, so there is no need to worry about an attack that depends on a protocol that uses one of the blocked ports, right? Not exactly—tunneling through an allowed protocol, such as HTTP, on an open port, such as port 80, can be an entry point to the network. With a defense-in-depth strategy, you assume that any one countermeasure maybe compromised and therefore multiple countermeasures are required to mitigate security risks. So far so good, however, you do not want to be too limited in how you think of threats.

As more services are provided on the Internet, from email and instant messaging to IP telephony and Web video conferencing, you are creating more potential vulnerabilities that can be exploited by attackers. A heap overflow vulnerability has been discovered and fixed in the Skype Internet phone client (Source: <http://www.skype.com/security/skype-sb-2005-03.html>). Frankly, there is nothing unusual about this; buffer overflows are a common security vulnerability. It, along with the other 4279 vulnerabilities found this year (according to the National Vulnerability Database at <http://nvd.nist.gov/>), does demonstrate that the breadth of the vulnerabilities in your systems is correlated with the number of services you provide.

Vulnerabilities are also associated with the type of content that moves through a network. Leaked trade secrets can compromise competitiveness. A database security breach can result in the disclosure of personal financial or protected healthcare information resulting in regulatory violations. Offensive material brought into the office can create a hostile work environment leading to civil actions against an employer. Content type is another dimension that comes into play when you consider the breadth of security requirements.



**Figure 1: The risk profile associated with messaging security is determined by both the breadth of network services provided and the breadth of content distributed.**

---

Clearly, when the subject of security turns to messaging security, you must think in terms of breadth as well as depth. The first step in a defense in depth and breadth approach is to prioritize information assets. Which applications, if they were unavailable, would severely disrupt operations? What data, if disclosed, would compromise competitive advantages or leave the organization liable for violating regulations? At the same time, consider threats to maintaining an appropriate work environment. Having a clear understanding of what is to be protected, you can apply countermeasures to protect those assets. Technology alone will not mitigate security risks—well-defined and implemented policies and procedures are required, too.

## Technologies for Messaging Defense in Depth and Breadth

If you had to summarize messaging security in its most basic form, it would have to be keeping the bad stuff out and keeping the confidential stuff in. There is plenty of malicious content to get in, including viruses, worms, Trojan horse, keyloggers, screen scrappers, root kits, and behavior-tracking adware. Add to that list offensive material that could create a hostile work environment, peer-to-peer file sharing services that consume storage as well as network bandwidth, and the use of non-work-related sites that can raise productivity concerns. Then there is the material that needs to stay within an organization's network, such as customer account information, healthcare data, and other confidential and proprietary information. Protecting these assets is done with several technologies: anti-malware, anti-spyware, firewalls, intrusion prevention systems (IPs), and content filters.

Anti-malware applications are commonly called antivirus, but that is a bit a misnomer. They certainly can detect and eliminate viruses, but anti-malware programs can effectively detect worms, Trojan horses, and other malware as well. Anti-malware should be deployed both on the network and on individual devices. There are several reasons for this. From the defense-in-depth strategy, a network-based anti-malware program may have vulnerabilities not present in desktop versions and vice versa. When the two anti-malware deployments use programs from different vendors, there is a chance that malware not detected by one will be identified by the other. It is also better to eliminate a threat earlier rather than later. The human body can fight off many pathogens but, as a rule, you are better off if they if they do infiltrate the body to begin with. Similarly, malware is better stopped at the perimeter than at the desktop. Once inside, the malware may be able to exploit vulnerabilities in the operating system (OS), network services, or applications.

Anti-spyware may detect some of the same threats as anti-malware, especially programs such as keyloggers. If adware is missed by anti-malware, anti-spyware can detect and eliminate the potentially unwanted program (PUP).

---

Firewalls are the first line of defense for messaging security. Again, following the principle of detect and block early, firewalls prevent potential threats from entering the network. Packet-filtering firewalls work well to prevent basic threats. For example, someone may unknowingly install an ftp server while setting up an ad hoc development server. (In the ideal world, this would never happen and policies and procedures would be followed; reality is a different story). A firewall can effectively block traffic to the file transfer server known for security vulnerabilities. Application proxy and circuit proxy firewalls operate at higher levels of the network stack and more effectively control traffic based on content.

Like anti-malware, firewalls belong both on the network and on servers and client devices. The need for multiple layers of firewall protection is obvious for mobile devices: if they are disconnected from the network, they are vulnerable unless local countermeasures are deployed. Even stationary devices, such as servers and workstations, should be protected with personal firewalls. If a client device were compromised by a Trojan horse that included botnet software for emailing spam, a properly configured personal firewall could block SMTP traffic and prevent the spam from reaching its recipients.

IPSs can also help to improve the security of messaging. The assumption behind the use of an IPS is that other countermeasures have failed and there has been a security breach. IPSs work at the network and host levels. Network-based IPSs may detect unusual traffic patterns; for example, large volumes of SMTP traffic from a workstation that has been compromised and included in a botnet. (A personal firewall should block this, but in case that fails, the IPS could detect and block the problem; this is another example of defense in depth).

So far, the technologies described have an important element in common: they operate to protect different types of content. Anti-malware can be configured to filter HTTP as well as SMTP traffic; firewalls can block thousands of ports; and IPSs can detect many types of anomalous patterns. Together they can provide both depth and breadth in security measures. Another technology that provides for a finely targeted breadth of coverage is content filtering.

Content filtering is essentially a traffic-scanning process that detects patterns of banned content. For example, employees of a company may not be allowed to use company resources to peruse gambling, entertainment, or music and video download sites. The other technologies listed cannot reasonably detect and block access to those sites without also potentially blocking access to legitimate use site. Content filters, however, can. They also provide breadth of protection by blocking offensive material, such as adult, hate speech, and similar sites.

One of the better aspects of content filtering is that it can work on traffic going out of as well as into an enterprise. The technology can therefore provide a measure of protection against information leaks.

If technical solutions alone were enough to reduce security risks to an acceptable level, you could stop here. They don't—which brings us to the other dimension of defense in depth and breadth: security management practices.

---

## Security Management Practices for Messaging Security

Technical solutions are necessary but not sufficient to obtaining reasonable levels of messaging security. You also need four security management practices: patching, policy development, auditing, and training.

All software past a somewhat minimal level of complexity is likely to have bugs. Some of those bugs will present security vulnerabilities. Probably one of the best remembered is the flaw in SQL Server that was exploited by the SQL Slammer worm so effectively and rapidly that traffic on major segments of the Internet was effectively shutdown. What is less well recalled is that a patch for that vulnerability was released by Microsoft months before the attack. Vulnerabilities exist in OSs, network services, databases, Web servers, Web browsers, enterprise resource planning systems (ERPs), email servers, and just about any other major category of software. (Peruse the vulnerability database at <http://www.securityfocus.com/vulnerabilities> for specifics). Protecting the integrity of messaging systems requires that you protect the integrity of shared infrastructure, and patching is a key element of that process.

Policy development is essential to establishing the goals and objectives of messaging security. Policies define what is to be protected as well as the measures used to protect those assets. Without adequate policies, managers and systems administrators are left to make decisions that may or may not align with broader enterprise objectives.

Auditing is another key practice that can help to maintain the integrity and confidentiality of messaging systems. This practice enables you to know what has actually happened within the infrastructure. Policy development defines what should be done, implementation and maintenance procedures execute those policies, and auditing verifies that they are effective.

The final key practice is training. You cannot underestimate the human dimension of messaging security. Users may not need to know the intricate details of IP protocols, OS vulnerabilities, or the inner workings of polymorphic viruses. What is important is an appreciation for the threats to information assets and users' role in protecting them. Simple acts—such as not downloading a file-sharing client, contacting the service desk when the personal firewall displays an unfamiliar message, or reporting phishing messages—can contribute to the overall effort to protect messaging and its underlying infrastructure.

There are many threats to messaging services. The common practice of defense in depth works as well here as in other areas of information security, but it works best when the full breadth of network services is accommodated within that framework.

---

## Article 5: Email Authentication

Spam and phishing continue to plague email systems. Estimates of the amount of spam clogging the email infrastructure reach as high as 75 to 80 percent of all email messages. Techniques such as blocking lists, content filters, and reputation filters all help to identify spam and phishing lures. The problem of unwanted email is so challenging that no single technique will work in all cases, and the effectiveness of techniques will vary. Part of the problem is that spammers are quick to adapt to new blocking technologies. Blocking based on content is not the only option, though.

### To Trust or Not To Trust

Implicit in many Internet protocols is the assumption of trust. Consider a couple of examples:

- If an email message arrives at a mail transfer agent (MTA) with a From: line indicating the message originated with JaneSmith@mycompany.com, the Simple Mail Transfer Protocol (SMTP) proceeds with that information without question. The assumption is that the message is truly from a user named Jane Smith who is a legitimate user of the mail service at a domain called mycompany.com.
- Similarly, if a domain name service (DNS) server returns information about a particular domain, say the IP address of mycompany.com is 192.111.222.123, then the requesting program, following standard DNS protocol, would proceed as if the information is valid.

The problem in both examples is that the information provided may have been tampered with and the recipient using basic protocols has no way of knowing that tampering occurred. In the case of the From: address, a spammer may have *spoofed*, or substituted a bogus address, for the address of the actual sender. In the case of the DNS information, the DNS server could have been compromised and the DNS database corrupted in an attack known as DNS poisoning. Again, there is no way for the recipient of the information to know that the integrity of the information has been compromised.

Implicit trust is no longer a sustainable model for several Internet protocols, especially those supporting email. Too much fraud and abuse is being perpetrated. A method is needed to preserve the integrity of messaging data. At the same time, the features we have come to expect from Internet protocols, such as ease of implementation and scalability, must be preserved. This can be done.

A number of proposals have been made to improve email integrity, and two protocols, Sender ID Framework (SIDF) and Domain Key Identified Mail (DKIM), are growing in popularity. These two protocols use different approaches and can complement each other or work independently. In both cases, the level of integrity of messaging data is improved and along with it, the ability to better filter spam, phishing lures, and other illegitimate messages that are taxing the email infrastructure.

Authenticating emails allows senders to assume responsibility of email messages. If a bank uses authenticated emails, customers can be more confident of the origin of a message from their financial institution. Recipients can also use that information to filter unwanted messages by allowing only messages from authenticated senders.

---

## Sender ID Framework


SIDF is the product of the merger of two earlier protocols, the Sender Policy Framework (SPF) and the Caller ID protocol proposed by Microsoft. There are a few basic steps in the SIDF protocol.

In the first step, a sender site will register information about its email servers in the DNS system. SPF records are added to the DNS entry for a domain. These SPF records specify the servers used to send email. A simple example of an SPF policy is:

```
mycompany.com TXT "v=spf1 a:outserver.mycompany.com -all"
```

which specifies that for the domain mycompany.com, the SPF version is 1, the server outserver.mycompany.com is authorized to send mail on behalf of the domain, and all other servers are not allowed.

Specifying an SPF record is only half of the solution. The receivers of messages need to look up this information and compare it with the header data in a message. The return-path, also known as the envelope sender address, is compared with the contents of the SPF record; if they match, the message is considered legitimate. If the sender server does not match the criteria in the SPF record, it is considered invalid.

 There are some subtle differences in the use of SPF records in Sender ID (RFC 4406) and in the SPF protocol itself (RFC 4408). For details about these differences, see [http://www.openspf.org/SPF\\_vs\\_Sender\\_ID](http://www.openspf.org/SPF_vs_Sender_ID).

After an email has been sent and reaches its destination, the next step is to conduct a check. The check can be based on either the purported responsible address (PRA) or a *mail from* check. The PRA check examines the sender, from, recent-sender, and resent-from fields to determine the sender. (The details of how these fields are used are defined by the RFC 2822.) The mail from check uses the sender address used to notify the senders of bounces. Forwarding agents and mailing lists have to change the mail header to support PRA checks, so it is not generally used.

It is not perfect, but SIDF provides several advantages over the implied trust model of SMTP:

- Wide adoption
- Low performance overhead
- Helps protect the integrity of domains
- Supports recipient confidence about the legitimacy of a message
- Useful for filtering based on failed checks

SIDF uses information about the path a message follows from start to finish and especially about the server from which the message originated. DKIM takes a different approach.

---

## DomainKeys Identified Mail

DKIM uses public key cryptography to verify the sender of a message. Although different from SDF, there are similarities in the implementations.

The first step in implementing DKIM is to update MTA software. (The email infrastructure needs to work with encryption.) Next, the email administrator must generate a pair of keys, one public and one private. The public key is published in a DNS record, and the private key is kept secret.

When a message is sent, the MTA signs the message using the private key and transmits the message. Once the message arrives at the destination email server, the server looks up the sender's public key in the DNS record. Decrypting the encrypted signature validates the message; if the decrypted data matches header data, the message is legitimate. This is predicated on the fact that because the private key is kept secret, only the sender could create a signature that could be decrypted with the public key. Legitimate messages are processed normally; others can be quarantined, deleted, flagged as potentially invalid, or otherwise segregated from other messages.

DKIM has wide support like SDF. It has the added advantage of working well under redirection. A disadvantage of DKIM is that software upgrades are required, but the benefits of authentication can often outweigh those costs.



Another point that should be noted is that both SDF and DKIM depend on the integrity of the DNS service. DNS servers must be hardened to prevent tampering.

## Moving to a Trust-But-Verify Model

Implicit trust is no longer a sustainable model for messaging. SDF and DKIM are two options for verifying the legitimacy of a message sender. These models build on existing protocols and add a minimal level of verification. Although they do not address other concerns, such as encrypting the content of messages, they do address the key issue of authentication.

Is email authentication the silver bullet that will end spam? Not a chance, but it is one more level for use in a defense-in-depth approach to improving email integrity.

---

## Article 6: Email Compliance and Regulations

Privacy and corporate governance regulations have introduced several additional responsibilities in IT management. As organizations have adapted to the requirements imposed by regulations, it has become clear that, in many cases, there is a confluence of interests in those concerned with compliance and those concerned with sound IT management practices. Email compliance and management is one of those cases.

This article examines some of the more well-known regulations that have an impact on email management practices, then explores the most effective way to comply with these regulations. The best practice is to implement policies and procedures that should already be in place in response to common business drivers. A corollary to this position is that sound email management practices will bring an organization into compliance with multiple regulations at once. There should be little need for silos of compliance procedures crafted for particular regulations.

### Major Regulations Affecting Email

Not surprisingly, the major regulations that demand the most attention of organizations, and corporations in particular, are also the ones with the most significant impact on email management. These include:

- The Sarbanes-Oxley Act—The major U.S. legislation on corporate governance is probably best known in the IT area for section 404. That part of the legislation requires adequate internal controls to protect the integrity of financial reporting.
- Health Insurance Portability and Accountability Act (HIPAA)—Addresses the privacy of individuals' healthcare information. The law requires, among other things, that doctors, hospitals, and other covered entities take reasonable steps to ensure that communications with patients are kept confidential.
- The Gramm-Leach-Bliley Act (GLBA)—Applies to financial services companies and requires that they follow three basic privacy rules. First, personal data of customers must be stored securely. Second, institutions must advise customers of the firm's privacy policies. Finally, customers must have the option to direct the institution to not share their information with third parties.
- Security Exchange Commission (SEC) Rule 17a-4—Applies to securities brokers and dealers and requires comprehensive recordkeeping. One of the requirements of the rule dictates that records of communications related to securities business must be preserved for not less than 6 years and, for the first 2 years, those records must be in an easily accessible format.

- 
- National Association of Securities Dealers (NASD) Rule 3010 and 3110—Approved by the U.S. SEC, this regulation requires its members to review incoming and outgoing communications to customers.
  - U.S. Patriot Act—Allows for email surveillance techniques, analogous to older telecommunication monitoring techniques, such as pen register and trap and trace. It also provides authority for government officials to access email records.
  - Cyber Security Enhancement Act—Allows U.S. government officials to receive email records from ISPs without warrants.

As this list demonstrates, the motivations for regulations affecting email management range from corporate governance and consumer protection to law enforcement and counter-terrorism. There is always the chance that focus on a particular public policy will wax and wane, but the broad base of legislation applying to email ensures that regulations of one form or another will be with us for the foreseeable future.

## Impact on Email Management Objectives

Looking across the major regulations, you can see that IT has several responsibilities with respect to email; they boil down to keeping email messages confidential, maintaining the integrity of the application, and ensuring that records of communication are available for some period of time. Even without regulations, many organizations would implement most if not all of these practices because these practices make sense from a business perspective. There would, of course, be differences in some details, such as how long to retain communications records and with whom to share certain types of information. The overall management framework, however, would remain the same.

Such a framework includes several procedures:

- Securing email servers, clients, and related infrastructure
- Defining email policies, including acceptable use and retention policies
- Monitoring and auditing email operations
- Archiving messages

These practices serve to meet the requirements of multiple regulations.

---

## **Email Security**

The need for email security is nothing new. Viruses, worms, Trojan horses, and blended threats have spread through email for years. It is difficult to imagine connecting a device (or at least a Windows device) to the Internet without anti-malware protection. In addition to client-based anti-malware, detection and filtering applications can be deployed on email servers or at the network perimeter. This setup provides a defense-in-depth strategy widely used in information security.

Access controls should be used to protect both confidentiality and integrity of the email system. Of course, authentication mechanisms should be in place to prevent unauthorized access to a user's account, but strict access controls should be in place on email servers to reduce the chance of a breach. Access events should be monitored and audited.

Deploying security measures is not a one-time activity. Anti-malware applications require updates both to their signature database for detecting malware and to the application code itself. Like any complex software, these programs can have bugs. Change management and patch management procedures should be used to ensure security measures are up to date. These procedures, like other email management operations, should be governed by well-defined policies.

## **Email Policies**

Email policies should set the scope of acceptable use, privacy and confidentiality, and retention. Acceptable use policies define the types of ways email systems may be used. Typically, organizational email is restricted to company business. These policies can also be used to define the management activities carried out in the process of managing the organization. For example, management may retain the right to monitor all email, keep copies of all emails, and treat email messages as assets of the organization. In effect, users should not expect or assume any degree of privacy with regards to email communications.

Privacy and confidentiality, as it applies to customer, patient, and client information as well as proprietary company information, is a different story. Organizations have a responsibility to protect private information that has been provided for business purposes. Several regulations make this explicit. Email policies should identify categories of information that may be sent through email and under what conditions. For example, private patient information may be sent from a hospital to a doctor only if it is encrypted and digitally signed.

Email archiving policies define what content is archived and how long it is retained. In many cases, archiving all email may seem like the appropriate measure, but such is not necessarily the case. For example, should messages quarantined as spam or phishing messages be archived? The volume of these messages alone could result in significant increases in storage volumes. There are also questions around personal mail folders (.pst files in Microsoft parlance). Should those be archived along with centralized mailboxes? In some cases, such as in securities industries, regulations require records of all communications, regardless of how they are stored internally in the email system; in other cases, the requirements may not be as stringent.

---

## ***Monitoring and Auditing Email Operations***

Auditing email operations can fit both regulatory and business requirements. Regulations may require that you not only perform a specific action but also that you can document that you have carried it out. Understanding trends in email use is essential for capacity planning and maintaining service availability. Monitoring and auditing operations can serve both of these objectives. Specifically, you should monitor:

- Provisioning of email accounts and changes in access controls
- Rates of malware and spam detection
- Growth rates in storage volumes
- Network traffic patterns

## ***Archiving Messages***

Another key area of regulatory compliance is archiving. Email storage requirements can grow quickly enough that it is not practical to keep all messages online. Archiving is required to meet compliance and business needs without sacrificing performance or increasing costs unnecessarily.

Archiving should not be conflated with backup and recovery. The purpose of backup and recovery is to ensure operational continuity in the event of a system failure or loss of data. Backups typically operate at the file-system level or lower. They do not necessarily provide ways to easily restore objects within a file, such as a series of emails.

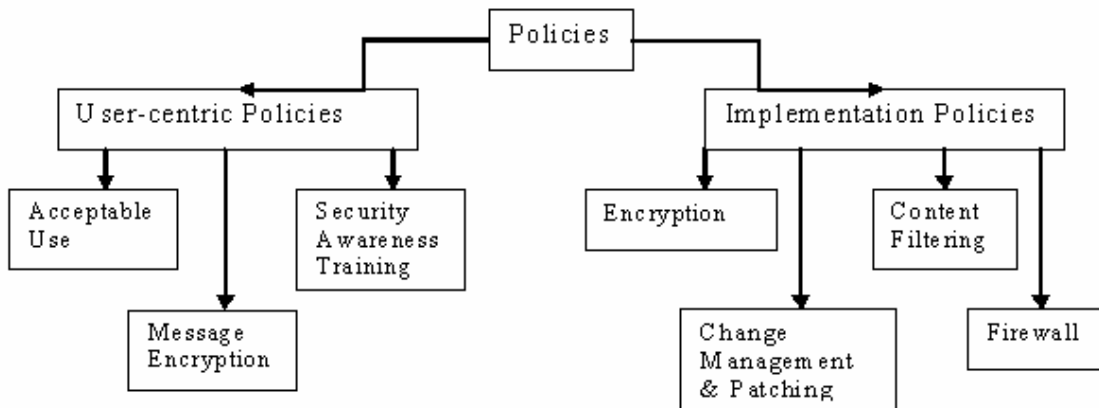
## **Summary**

Email compliance is just one instance of the regulatory impact on IT operations. There are a number of government regulations that apply to email services, and the list of such laws is likely to grow. Fortunately, many regulatory requirements coincide with business requirements for security, business continuity, and operations management. Sound email management driven by business needs can go a long way toward compliance as well.

## Article 7: Essential Policies for Messaging Security

Securing a messaging infrastructure is a multi-faceted challenge that begins with balancing functionality and security. It is often said that there is a tradeoff between the usefulness of a system and its level of security. This is not to say that reasonably secured systems are unusable or that feature-rich, user-friendly applications are inherently insecure. It does mean that as networks and applications become more complex, the potential security risks increase.

Messaging is a prime example. As email clients began to include improved functionality with features such as macros, malware developers took advantage of those features. Worms can threaten instant messaging (IM). Flaws in Internet phone services, such as buffer overflows, can provide entry points to attackers. In spite of these types of vulnerabilities, the cost/benefit ratio of messaging can be substantially in favor of the benefits, as long as certain protections are in place. A coordinated response to threats to messaging depends on a set of essential policies governing the use and implementation of messaging services. Figure 1 shows a hierarchy of these policies.



**Figure 1: Messaging policies span both user-centric and implementation issues.**

As Figure 1 shows, policies that have a direct impact on messaging extend well beyond basic email and IM applications. Messaging services on the network, server, and application infrastructure must be managed as well.

Broadly speaking, messaging security policies can be divided into two groups: user-centric policies and implementation policies. The former focus on what users should do, what they should not do, and what they should know (like the practice of monitoring message traffic). The latter group of policies is designed for IT administrators who are responsible for protecting the overall integrity, confidentiality, and availability of system services.

For more information about security policies in general, see the SANS Institute's Policy Project at <http://www.sans.org/resources/policies/>.

---

## Risk Tolerance and Messaging Policies

The details of an organization's policies will be driven, in part, by that enterprise's risk tolerance. Risk tolerance will vary from one business to another, and even within a single organization. For example, a CIO might have a moderate risk tolerance with regards to messaging security but extremely low tolerance for risk when it comes to protecting the company's financial system. The spectrum of tolerance for risk with regards to messaging can lead to several different approaches to messaging services:

- Preventing the use of messaging technologies
- Allowing controlled use of messaging technologies by providing only services that are well understood and believed to be reasonably well protected
- Allowing controlled use of messaging technologies and blocking features that present known vulnerabilities
- Not limiting messaging services and depending on network and host defenses to mitigate security risks

The first option is too draconian for most organizations, especially if email were included in the ban on messaging services. These services are efficient and reliable in most cases; it does not make sense to eliminate them completely.

The second option takes a conservative "trust only what is known" approach. This is a prudent stand in high-security environments in which the cost of a security breach could easily outweigh the benefits of the messaging service. In this case, a messaging service should be deployed in a lab environment without access to production environments, where the application is thoroughly tested. Testing might include analyzing network traffic between clients and servers and probing for vulnerabilities such as buffer overflows.

The next approach begins with a different assumption—essentially that services are allowed until they are known to be vulnerable to attacks. And the last option may suffice in some cases, as under the following circumstances (these are the minimal requirements; there may be others depending on the situation):

- Network traffic across the Internet is encrypted
- All messaging traffic, including HTTP, is scanned for malware and controlled content, such as private customer information and proprietary business information
- Message applications are under change control and patched appropriately
- Access controls are in place and enforced to ensure application code is not compromised
- Sufficient monitoring practices are in place to detect malicious activity, such as a Denial of Service (DoS) attack

Once the organization has identified its level of acceptable risk and the corresponding approach to messaging protection, policies can be defined.


---

## User-Centric Policies

Three user-centric policies should be defined: an acceptable use policy, a message encryption policy, and an awareness training policy. Acceptable use policies define what users may do with a messaging system. Typically, these policies limit business systems to legitimate business uses but allow for reasonable personal use. Emailing a friend about a lunch date is not likely to raise any eyebrows but running an eBay business on the side using the company email system is another story. These policies may also notify users that the contents of emails are considered company property and may be monitored to prevent the loss of intellectual property or the distribution of inappropriate content that could contribute to a hostile work environment.

Message encryption policies describe the need for secure communications and define when encryption is required. For example, if personally identifying information about a customer is sent outside the company network to a business partner or some other party with a legitimate need for the information, it must be encrypted. There is no need to bother with technical details such as encryption algorithms and key lengths in user policies; presumably, IT will manage those details.

User awareness training is important enough to warrant a policy. Well-designed and implemented security measures can be undermined by user error. Social engineering tricks such as impersonating an authority figure over the phone can be used to solicit private or confidential information.

 The board of Hewlett-Packard has made news recently because of internal frictions on the board that led to the use of pretexting to collect information about some members. See “Fuzzy Laws Come Into Play in H.P. Pretexting Case” at <http://www.nytimes.com/2006/09/19/technology/19hewlett.html?ref=business> for more information.

As with message encryption, there is no need to train users on the technical details of how attackers and malware writers exploit vulnerabilities, but make users aware of the broad details of threats and vulnerabilities.

## Implementation Policies

Implementation policies for messaging should have a two-part focus: first on preserving the integrity and confidentiality of communications, and second, on protecting the organization. Policies, in general, describe what should be done, not necessarily how to do it; encryption policies are sometimes the exception to this rule. For example, a policy might define the need for confidential messaging and specify a minimum work factor for cracking an encrypted message (the “what”) or it might denote certain algorithms, such as Advanced Encryption Standard (AES—the “how”). Either way will work and the important point is that a standard be in place so that there is no question about what constitutes acceptable encryption.

---

A messaging-specific policy is content filtering. The policy should define what types of content are blocked from either entering or leaving the network. Obvious candidates for incoming blocks are content from sites providing:

- Adult material
- Hate speech
- Gambling
- Gaming

The content-filtering policy should also define material that should not leave the organization, such as trade secrets, personally identifying information, and company confidential material.

Organizations probably already have a number of broadly applicable policies in place that affect messaging, including:

- Change management, which is the process for controlling modifications to operations systems that takes into account dependencies between systems and serves to minimize disruption in services because of changes to hardware and software.
- Patch management, which is the practice of testing and installing software updates to correct flaws that present either functional or security issues.
- Firewall configuration, which defines what protocols are allowed into and out of the organization network.

## Summary

Messaging is a fundamental service provided by IT. Protecting it begins with defining policies that address both its human and technical aspects. You can attend to the human element with clear acceptable use policies and some basic training. The technical aspects are addressed by broadly applicable policies, such as change management and firewall policies, and enhanced with messaging-specific policies such as content filtering.

---

## Article 8: FISMA and Messaging Security

The Federal Information Security Management Act (FISMA) is a broad set of regulations that addresses the management and control of information resources within the federal government and is applicable to many federal agencies. The high-level objectives of the act include:

- Ensuring information is appropriately categorized according to security objectives and the impact of a security breach
- Setting standards for minimal security requirements for federal information systems
- Selecting and implementing appropriate security controls based on a risk assessment model
- Assessing and certifying the effectiveness of security controls

Volumes could, and probably have been, written about FISMA. The goal of this article is to consider FISMA with respect to messaging security. The most relevant areas of FISMA to messaging security are:

- Risk management
- Policies and procedures
- Identification and authentication
- Awareness training
- Contingency planning
- Auditing and accountability

There are several other areas within FISMA that address broad range issues in information security.

---

## Risk Assessment


The first step in any comprehensive security management strategy is risk assessment. FISMA requires periodic risk assessments to understand the potential impact of security threats. With regards to messaging, the impacts can include:

- Loss of confidentiality if messages are intercepted
- Loss of availability as a result of Denial of Service (DoS) attacks, rapidly spreading malware, or other disruption to agency operations
- Loss of integrity from tampering, such as via a man-in-the-middle attack

Threats to messaging security are well known, and they can originate both within and outside of an agency.

### ***External Threats to Messaging***

External threats include viruses, worms, Trojan horses, spyware, keyloggers, botnets, and rootkits—all of which can compromise a messaging system. Malware can disrupt operations, and the speed at which malware can spread and the sophistication of the software is making the control of malicious code more difficult. Some malware, particularly botnets and rootkits, can commandeer computing resources and hide the fact that a device has been compromised. Previously effective security measures, such as reformatting hard drives and reinstalling the operating system (OS) are no longer effective as rootkits may be using advanced power control and PCI subsystems for storage.

 For more information about emerging techniques for persistently storing rootkits, see “Can You Trust Your Firmware?” ([http://www.realtime-websecurity.com/articles\\_and\\_analysis/2006/10/malware\\_continues\\_to\\_evolve.html](http://www.realtime-websecurity.com/articles_and_analysis/2006/10/malware_continues_to_evolve.html)) and “More Rootkit Threats—This Time It’s PCI Devices” ([http://www.realtime-websecurity.com/2006/11/more\\_rootkit\\_threats\\_this\\_time.html](http://www.realtime-websecurity.com/2006/11/more_rootkit_threats_this_time.html)).

The consequences of external threats can be severe. In October 2006, the U.S. Commerce Department reported that attacks on computers at the Bureau of Industry and Security forced the department to shut down Internet access in September. InformationWeek reported that the bureau had decided to replace workstations compromised with rootkits rather than trust the devices after the attack (Source: <http://www.informationweek.com/management/showArticle.jhtml?articleID=193105227&subSection=Global>).

### ***Internal Threats***

Risk assessments of messaging must also consider threats from within the organization. Information theft and leaks are clearly confidentiality problems. The well-publicized theft of a Veterans Administration (VA) employee’s laptop containing more than 28 million personal information records is an obvious example. Information leaks can also occur through messaging systems without adequate filtering. The product of a risk analysis is a set of threats and impacts that can be used to prioritize security policies and procedures.

---

## Messaging Policies and Procedures

Messaging policies and procedures are the bridge between the strategic security objectives that emerge from the risk assessment process and the operational aspects of implementing security controls. Policies describe the mechanisms for controlling threats without delving into technical detail. For example, an email retention policy may describe which types of messages must be retained and the length of time they must be retained but not detail specific steps or technologies for implementing those guidelines.

The operational details for implementing messaging policies should be described in procedures that address the key components and processes of messaging services, including:

- Network security
- Server and client security
- Encryption
- Message filtering
- Backup and recovery

Together, policies and procedures provide the guidance needed to implement effective control structures around messaging security.

## Identification and Authentication

Identification and authentication governs the process by which users and user agents are granted access to systems and resources. This is especially challenging in highly distributed environments such as those found in federal agencies. For example, the use of federated identity management adds a layer of trust between agencies or departments that must be governed by policies, implemented with comprehensive procedures, and audited to ensure proper implementation.

## Awareness Training

Security professionals and other information technology (IT) specialists cannot maintain security on their own. All users of information systems must be aware of risks related to information management and act in a responsible manner.

FISMA's requirements are twofold in this case. First, users must be made aware of relevant laws, executive orders, policies, and procedures in the area of computer security. Second, personnel must be trained in information security so that they can carry out their responsibilities knowledgeable of security threats.

---

## Contingency Planning

Contingency planning is essential to messaging security because it is one of the foundations for ensuring service availability. Contingency planning includes day-to-day operations, such as backup and recovery, as well as longer-term planning and operations, including failover systems, emergency response plans, and transition plans for moving operations to backup systems and then back to production systems.

## Auditing and Accountability

Up to this point, all aspects of FISMA are generally applicable to information security best practices. Although auditing and accountability are also elements of these best practices, they are especially important to the practice of FISMA compliance.

In addition to implementing security controls, security managers must be able to prove with documentation that their systems are compliant, their users are trained, and their risk assessments are up to date. A sound security strategy can be undermined by weak documentation, which leads to audit reports that do not accurately reflect the state of an agency's or department's security status.

## Summary

FISMA is a broad and complex regulation, but for the most part, the practices that are demanded by FISMA are generally considered best practices in information security management. The need for comprehensive documentation places additional burdens on security professionals that may be more akin to document management than to information security practices.

---

## Article 9: Host Intrusion Detection and Prevention

In the defense-in-depth security framework, host intrusion prevention systems (IPSs) are one of the last lines of defense. Host IPSs reside on the devices that they protect. These systems use a combination of signature- and behavior-based analysis to detect attacks and mitigate the impact of those attacks.



There are also IPSs designed to protect networks using similar techniques to host IPSs, but this article will focus on the device-level systems.

This article will examine three aspects of host intrusion detection and prevention:

- Threats to hosts
- Detection techniques
- Limits of host intrusion prevention technologies

### Threats to Hosts

Hosts can be any type of device, from a high-end server running enterprise applications to notebook computers used for email, browsing, and basic office productivity. Regardless of how a host is used, it is subject to a number of broad-ranging types of attacks, including:

- Data theft
- Data tampering
- Loss of control
- Denial of service

#### ***Data Theft***

Data theft can target personal information as well as proprietary corporate information, especially intellectual property. In the case of the [recent breach at the University of California, Los Angeles](#) (UCLA), the Los Angeles Times reported “an attacker found one small vulnerability and was able to exploit it, and then cover their tracks.” The purpose of the attack may have been identity theft, as the Los Angeles Times also reported that a former student had been the victim of identity theft that included a \$24,500 car loan made in the student’s name.

The theft of intellectual property is significant enough that the U.S. Justice Department has established a Computer Hacking and Intellectual Property (CHIP) unit within the agency to address intellectual property crime. In some cases, the goal is not to steal information but to tamper with it.

---

## **Data Tampering**


Tampering with data is a direct threat to the integrity of information. Threats of this type can come from both outside and inside an organization. Although external attackers can breach a system with the intent of changing data, perhaps to disrupt services, insiders also have plenty of motivation to change records, either for their benefit or to damage the owner of the information. For example, someone committing fraud against a company may tamper with data to cover their tracks. An unethical super user or systems administrator may change personnel records or other protected information. In both cases, auditing and monitoring measures can detect the breach after the fact; intrusion prevention and detection may be able to block such tampering as it is attempted.

Another form of tampering that is emerging is known as ransomware. In this kind of attack, a victim's files are encrypted and held for ransom. If payment is made, the encryption key is provided so that the data can be unencrypted. Ransomware may be subject to the same economics as spam—sufficient returns on investment are earned even when an extremely small percent of victims respond to the attack.

 See [Cybercrime and the Economics of Micro-markets](#) for more information about this topic.

## **Loss of Control**

Botnets, networks of compromised computers controlled by a single individual or organization (known as a bot herder), represent the impact of losing control of computing devices. Botnets are widely believed to be responsible for generating most spam; they have also been implicated in Distributed Denial of Service (DDoS) attacks. Botnets may also be used to perpetrate click fraud.

 For more information about botnets' involvement in click fraud, see [Cybereconomy Spills Over to Real Economy](#).

A compromised host may not have a direct, noticeable impact on the person whose computer is compromised other than slower performance and inordinate bandwidth consumption.

## **DoS Attacks**

Hosts are subject to DoS attacks, which can range from moderately disruptive, as in the case of an attack on [Akamai Technologies servers](#), to so disruptive that they can shutdown a business, as in the case of Blue Security documented in a recent article in [Wired](#) magazine. DoS attacks are best addressed at the network rather than the host level.

---

## Detection Techniques


Attacks on a host can be detected based on well-known patterns or because of activity indicative of an attack. In the first case, signatures are used to determine whether an attack is underway. For example, if a large number of SYN packets is sent to a host, it could be part of a simple DoS attack known as SYN flooding. This type of attack is easily detected (although not as easily terminated).

The more challenging case is when the behavior of a program indicates an attack is underway. For example, buffer overflows are common vulnerabilities. Attackers exploit overflow I/O buffers in vulnerable programs and inject malicious code to compromise a system. Host IPSs can detect when shell code used by an attacker starts to execute.

Another set of methods employed by host IPSs entails protecting OS files and configurations, such as registry keys. This can be done with cryptographic techniques. When an OS is first installed, message digests can be computed for critical files. Those message digests are used as a baseline for comparison; at regular intervals, new message digests are computed for those critical files. If the message digests do not match, the file has been tampered with.

## Limitations of Host IPSs

Host intrusion prevention, although useful and effective in many cases, is not infallible. Like other security measures, host IPSs can be circumvented in some cases. For example, it may be possible to bypass some detection techniques by making calls to low-level OS services rather than intermediary services. Also, unless configured properly, host IPSs may not protect critical files that have multiple paths leading to them (known as symbolic links.)

 For more information about these limitations, see Eugene Tsyklevich's Blackhat presentation at [www.blackhat.com/presentations/bh-usa-04/bh-us-04-tsyklevich.pdf](http://www.blackhat.com/presentations/bh-usa-04/bh-us-04-tsyklevich.pdf).

---

## Article 10: Instant Messaging Worms and Other IM Threats

As email systems are becoming more difficult to penetrate with viruses and worms, it is not surprising that malware writers are turning to other communication channels to push worms, Trojan horses, and botnet software. Instant messaging (IM) services have become an ideal mechanism for malware developers for the following reasons:

- Malware can be spread quickly over IM, especially with users with large numbers of contacts
- Users are not necessarily as cautious with IM as they are with email, especially when dealing with a message from an apparent friend or colleague
- The activities carried out by malware delivered by IM can be relatively low profile (for example, opening an IRC channel and listening for commands)
- Rootkits can be sent along with worms and Trojans to hide the presences of the malware
- IM communications are relatively short, so it is easier to create a message that appears legitimate in a wide range of circumstances

This article will examine some examples of IM worms and related threats, identify general principles of underlying IM threats, and discuss tips for minimizing the impact of these threats.

### Examples of IM Malware

Examples of IM malware are not difficult to find; they have been with us for years. Like malware spread by other means, IM malware is getting more sophisticated and now includes blended threats. The following list highlights examples that demonstrate the breadth of techniques and tactics used by malware writers in the IM realm:

- The Opanki worm is an IM-based threat that establishes a backdoor on TCP 443 and then waits to receive commands from an IRC server. This worm can also exploit the Server Service Vulnerability that allows for remote code execution. The worm hides itself with the NTRootkitJ rootkit.

 See [http://vil.nai.com/vil/Content/v\\_140546.htm](http://vil.nai.com/vil/Content/v_140546.htm) for more information about this worm.

- Adware-BuddyLinks, a potentially unwanted program (PUP), fits in the category of a Trojan in some ways because of its somewhat hidden behaviors. Users download an installer, which has a Terms of Use agreement that includes a section stating that the user agrees to allow a message to be sent to all contacts listed in the user's AOL Instant Message buddy list.

- 
- The Pipeline worm establishes botnets by exploiting instant messages that lure users to click on links within a message. This worm is more sophisticated than many because the payload is broken down into multiple files that can be downloaded in several ways. The program is designed to minimize the dependencies between components so that victims can still be infected without using the same set of servers or download orders.
  - The Heartworm worm lures recipients to click a link to receive a virtual greeting card. When they do, data-theft malware is downloaded and the worm spreads to the user's IM contacts. The unusual part of this worm is that it appears to be a hoax—the messages displayed by the worm are modeled after a reported hoax. Obviously, the goal is to lead victims to believe that they are victims of a hoax, not an actual infection.

IM threats have all the sophistication of malware that has already affected email.

## General Principles of IM Threats

From the known examples of IM threats, there are a few generalizations you can make about them. First, the payload is similar to other current malware and PUPs. These threats contain information-stealing programs, botnet code, rootkits, and other tools from the attacker's toolbox.

Second, IM threats are going to some length to disguise their presence. The user of rootkits, multiple command and control servers, and even hoaxes indicates that the malware writers want their code to stay in place for a while.

Third, the motivations are not vandalism and destruction but more likely economic. Botnets are a preferred means for distributing spam, and IM worms can deliver information-stealing programs to copy usernames, passwords, credit card numbers, PayPal account details, and other financially useful data.

Next, malware delivered by IM is not following the same development curve as email-based malware. Email malware went through several evolutionary stages with encrypted viruses, polymorphic viruses, blended threats, and so on. IM malware leverages what has already been developed, so it can be just as stealthy and destructive as its email counterparts.

Finally, IM malware involves a social engineering dimension unique to instant messages. IM communications are quick and often little is questioned about a benign-looking message from someone on a contact list. By luring users to a site with generic but plausible "hey check this out" kinds of messages, the malware writers are getting their victims to do some of the work for them. So what is to be done with these threats that have something old and something new?

---

## Tips on Controlling IM Malware

First and foremost, you need to focus on the fundamentals: anti-malware systems and end user training. Antivirus and anti-spyware tools should be used to scan messaging content. Scanning email alone is obviously not enough to keep malware at bay. Network- and client-based tools should both be used. Tools with centralized management programs make it easier to track the level of threats making their way into an organization.

IM malware spreads so quickly in part because users do things they should not do. At the very least, users should

- Avoid downloading files using IM services
- Not open messages from unknown users
- Not overload contact lists; highly connected individuals can spread malware to large numbers of IM users rapidly

Because botnets depend on communication with command and control servers, using personal firewalls can help block the communications needed to exploit compromised hosts. Personal firewalls should allow only approved applications to access network services and then only using specified ports. A host may be compromised but at least it will not be responsible for spreading spam, PUPs, or malware.

## Summary

Since the time of the first viruses, we have known that malware would evolve to adapt to new countermeasures and to exploit new vulnerabilities. Thus, it is not surprising that techniques that worked well for email viruses are now showing up on IM systems. What may have been less anticipated is how well malware writers are using social engineering techniques to hone their lures to the particular medium they use to spread their malware. IM, by its nature, is less complex than email communication and because of that easier to make a malware message appear legitimate.

Content-filtering technologies are still the first line of defense, but user awareness is also a key ingredient to reducing the threat of IM malware. At the very least, do not make it easy for the bad guys.

---

## Article 11: IT Audits: What to Expect

The section of the Sarbanes-Oxley Act (SOX) known as SOX 404 has gotten much attention from IT management. This is not a passing fad; there will not be any regulatory equivalent of a dot com bust—compliance with SOX 404 is here to stay in some form or another. In fact, even if SOX 404 is changed, as many have argued for, other regulatory schemes are in place that influence how IT does its job. With these regulations come requirements for verification—and that means auditing.

For starters, it is important to remember that IT auditing has been around longer than SOX. Auditing is nothing new, but it is different now. In the distant past (that is, pre-SOX), auditors would review transactions, key systems, and personnel; validate access controls; and review software development and change management practices. The level of scrutiny varied, but few lost any sleep over the thought of an IT audit.

Today, IT audits are more comprehensive and more thorough—systems and controls are not just reviewed, they are tested. At the same time, the audit practices of today have evolved from earlier practices. It helps to understand the past (even if you are not condemned to repeat it) because what you can expect in IT audits is found in part in three sets of guidelines or frameworks:

- Statement of Accounting Standards 94 (SAS 94)
- Committee of Sponsoring Organizations (COSO)
- Control Objectives for Information and related Technology (COBIT)

There is some overlap between these, but they have different areas of emphasis; COBIT, for example, is primarily focused on IT governance. Taken together, these three provide a good idea of what you can expect in an IT audit. The major areas of emphasis are:

- Controls for the process of collecting, deriving, analyzing, and reporting on financial transactions
- Controls over the implementation of generally accepted accounting principles
- Controls to prevent and detect fraud
- Controls over non-standard transactions that can have a significant impact on financial state
- Controls on period-ending reporting

Each of the three frameworks listed earlier contributes to the practice of contemporary IT audits. Audits will vary according to the sophistication of IT systems and the risk tolerance of the businesses being audited. Although it is impossible to say with 100 percent certainty what will happen in an audit, the auditing and control frameworks provide good guidance.

---

## In the Beginning: SAS 94

SOX auditing combines something old and something new. SAS 94, Consideration of Internal Control in Financial Statement Audit, established standards for examining the role of IT in financial systems as early as 1990. The SAS 94 guidelines called for audits of:

- Control environment
- Risk assessment
- Control activities
- Information and communication systems
- Monitoring

The control environment is the organizational structure in which security is practiced. It is the most abstract of the SAS 94 subject areas and can be thought of as the alignment of business strategy and IT implementation and the relative importance IT governance has in the organization.

A risk assessment is a formal analysis of the vulnerabilities, threats, and impact of realized threats on an organization. Risk assessments are a starting point for understanding the kinds of control activities that must be in place to protect information assets.

Control activities are defined by policies and procedures put in place to ensure that executive directives are implemented. For example, information classification policies may define various categories of information (for example, public, sensitive, private, and confidential) and the levels of protection each should have. Procedures are then defined for creating, managing, and destroying the various categories. Control activities address the full range of protections dictated by a risk posture and sufficiently enforced.

Information and communication systems are designed to store, process, and transmit information needed for production operations. Clearly, this definition is broad enough to encompass all IT infrastructure within an organization. The purpose of monitoring is to determine the effectiveness of the controls in production environments.

## COSO and COBIT Frameworks

The COSO framework builds on SAS 94 and adds four areas of control:


- Internal environment
- Objective setting
- Event identification
- Risk response

---

These topic areas are something of a bridge to the COBIT framework, which is the most IT-specific of the frameworks used to establish IT audit practices. COBIT provides a governance framework centered around four key functions:

- Planning and organizing
- Acquiring and implementing
- Delivering and supporting
- Monitoring and evaluating

Each of these areas is further divided into operations that can be mapped to COSO- and SAS 94-related audit tasks. For example, managing systems configurations within COBIT maps to a control objective in COSO.


 For more information about COBIT, listen to the [“COBIT and Governance Best Practices”](#) at the Realtime Messaging and Web Security Community.

Now, given the kinds of focus suggested by these frameworks, what will auditors do when they arrive at your office?

## Steps to the IT Audit Process

The following list highlights steps in IT auditing. Auditors

- Plan the scope of the audit
- Assess complexity of IT systems relevant to financial reporting

 This step can include fundamental services, such as networking and database management, as well as higher-level systems, such as ERPs.

- Define acceptable risk levels for the business
- Conduct risk assessment and identify mitigation measures
- Determine controls to test
- Formulate and validate tests
- Conduct test of controls
- Assess results of tests and document remediation steps required

Again, auditing is not new to IT, but it is now more extensive. The purpose of the audit is to assess controls on IT processes. Fortunately, frameworks such as COBIT that are used by auditors are also best practices for managing IT operations. Implement the COBIT framework, and you can rest as easily as you did before the advent of SOX.

---

### Resources

COBIT at <http://www.isaca.org>

Generally Accepted Accounting Principle at <http://www.fasab.gov/accepted.html>

M. Virginia Cerullo, "How the New Standards and Regulations Affect an Auditor's Assessment of Compliance with Internal Controls" *Information Systems Control Journal Volume 4, 2005* (subscription required)

SAS 94, "The Effect of Information Technology on the Auditor's Consideration of Internal Control in a Financial Statement Audit" at <http://www.aicpa.org/download/members/div/auditstd/AU-00319.pdf>

---

## Article 12: Physical and Digital Security Convergence

Security management entails a number of dimensions, including logical security over information in whatever form it takes; electronic security over networks, servers, and other devices that store, manage, and transmit information; and physical security, which addresses the protection of persons and property. Not surprisingly, logical and electronic security are tightly coupled. Information residing on a server is dependent on the access controls and other security measures of the server. Encryption can protect data when it is transmitted across insecure channels, such as the Internet, but it can also provide additional protection against information theft. For example, the use of full disk encryption can reduce the chance of information theft when a mobile device is lost or stolen.

Today, we are witnessing the convergence of digital forms of security (that is, logical and electronic security) with physical security. The convergence is understandable given the general acceptance for aligning security practices with business drivers. The goals of business operations do not distinguish between the types of technologies that provide security. This article will examine several topics in the convergence of digital and physical security:

- Business drivers behind the convergence of physical and digital security
- Benefits of consolidated physical and digital security management
- Industry standards supporting consolidated physical and digital security management

### Business Drivers

The consolidation of physical and digital security is largely driven by the fact that it makes sense from a business perspective. This is not primarily a bottom-up driven effort. In fact there are natural and organizational barriers to consolidation. Different departments within an organization often manage digital security and physical security. The departments use different systems to manage their domains and even the types and granularity of information managed is different. For example, network security systems can generate thousands of log entries in short periods of time from fine-grained access controls such as accessing a file. Physical security tends to log more granular events, such as a person entering or leaving a building. In spite of these inherent differences, the business needs of the organization provide enough momentum to overcome the differences.

Some of the key business drivers are:

- The increasing complexity of physical and digital assets
- The increasing value of information assets
- Demands for compliance with regulations
- Efficiency and cost-cutting drivers

---

The increasing complexity of digital assets is well known. Distributed systems that depend upon services provided by multiple applications running on a variety of platforms each with their own life cycle and maintenance schedules are standard in today's IT realm. Physical security is also becoming more complex. Assets are no longer restricted to a single location. Mobile devices are taking valuable information assets outside normal organizational boundaries. Organizations are also physically distributed. Headquarters may be in one country, the research and development campus in another country, and customer support and sales offices in several others. Securing and protecting these widespread assets is more difficult than providing the same services to more localized facilities.

Another factor is the increasing value of information. The increasing threat of cybercrime is targeting information assets, such as proprietary information, trade secrets, and confidential personal information (for example, credit and other financial information of customers).

Government regulations designed to protect privacy and the integrity of business information do not necessarily distinguish between physical and electronic security. For example, a regulation requiring that businesses not disclose personal credit information may not differentiate between information disclosed by hacking into a database versus a thief walking into an office and stealing a paper file. Businesses need to protect information regardless of the form this protection takes.

Yet another driver behind convergence is the need to control costs. If efficiencies can be achieved by combining overlapping security structures, organizations will merge those structures. If leveraging physical security can improve digital security while containing costs, it will happen.

## **Benefits of Consolidated Physical and Digital Security Management**

In addition to potential cost savings, there are a number of benefits to merging physical and digital security:

- Improved access controls
- Improved incident response
- Expansion of defense in depth

Combining physical and digital access controls can provide for more effective monitoring of network and system activities. An obvious example is that physical and digital events should be consistent. For example, if a salesperson in the Brussels offices enters the building at 3:00 p.m. local time and 30 minutes later appears to connect to a wireless network in the New York office, her account may have been compromised.

When breaches or attempted breaches do occur, using both physical and digital information can improve responsiveness. Consider an attempted breach of a wireless network at a company office. The perpetrators would have to be in relatively close proximity to the building or within the building itself to access the network. Surveillance cameras may record vehicles coming and going from the parking lot around the time of the attack or record visitors in the main reception area that might have been trying to hack into the network.

---


Another benefit of consolidation is the expansion of the scope and breadth of the defense-in-depth principle. This is a fundamental practice in information security: trust no one type of defense and always deploy multiple security measures. Physical security adds measures that may not be easily implemented otherwise. For example, by strictly controlling physical access to critical devices, systems administrators may be able to allow functions, such as the ability to use USB flash drives, that would otherwise have to be disabled.

## Industry Standards Supporting Convergence


Standards are emerging to promote the convergence of physical and digital security. The Physical Security Bridge to IT Security (PHYSBITS) provides a framework as well as data models for combining several aspects of physical and digital security:

- Security monitoring
- User administration
- Policy management
- Log management
- System monitoring

The objective of the data model is to map physical security data into an IT security framework, making both sources of data accessible from a single management framework.

 For more information about PHYSBIT, see [http://www.oasis-open.org/committees/download.php/7778/OSE\\_white\\_paper.pdf](http://www.oasis-open.org/committees/download.php/7778/OSE_white_paper.pdf).

Another standard for supporting physical and digital security is the Open Building Exchange (oBIX), which is focused on developing information exchange standards and Web service standards for intelligent building applications. The objective is to allow better integration of building management systems with building automation, security, and other services that can benefit from physical state data.

 For details about oBIX, see <http://www.obix.org/>.

The convergence of physical and digital security is driven by business needs but enabled by technologies provided by an array of vendors. The speed at which standards such as PHYSBITS and oBIX are adopted as well as the time required to deploy consolidated security management systems will influence how quickly organizations can begin to realize the benefits of consolidated physical and security management.

---

## Article 13: Rootkit Challenges

Rootkits are one of the threats that can keep security professionals up at night. Viruses and worms are real threats but can be reasonably well controlled. Trojan horses, keyloggers, and other information-stealing programs are growing threats but, if detected, can usually be removed. In fact, the first step in many security countermeasures, such as antivirus solutions, is detecting the presence of malicious and unwanted programs. The role of rootkits is to hide them.

Rootkits use a number of techniques to hide themselves and other malicious programs. Rootkits alter the process list so that executing malware does not appear with other running programs. They might alter the information returned by operating system (OS) functions; for example, a function call to get the size of an altered file may return the original size, not the new size after alteration, effectively masking the file tampering. Rootkits can also prevent OS functions from listing files, such as files for a Trojan horse or a program designed to launch a Denial of Service (DoS) attack. Several types of techniques are used to implement rootkits, and each attacks at a different level of the OS.

### Types of Rootkits

There are several types of rootkits; they are, in increasingly threatening order:

- Application-level rootkits
- Library rootkits
- Kernel rootkits
- Virtualized rootkits
- Firmware-based rootkits

#### ***Application-Level Rootkits***

Application-level rootkits alter programs that run in the user space of the OS. At this level, they do not have access to the lowest-level, trusted routines in an OS. Application-level rootkits can use techniques similar to Trojan horses to hide their presence. As they do not alter low-level OS functions, application-level rootkits can be detected by methods that depend on host OS functions.

---

## **Library Rootkits**

Library rootkits alter shared code used by multiple applications. Using OS hooks and monitoring system activity, rootkits can intercept calls to OS shared code and execute its own code instead. For example, a call to an OS function that lists all the currently executing processes could be intercepted by the rootkit. The rootkit then calls the legitimate OS function, receives the results, filters the results to remove any hidden programs, and returns the altered results to the calling program.


## **Kernel Rootkits**

Kernel rootkits alter the core code of an OS. These are especially threatening because the kernel, at least in theory, is the immutable, trusted foundation of the OS. The kernel is a small amount of code that controls the lowest-level functions of the OS and provides basic services, such as process and memory management, to other processes running on a system. If the kernel cannot be trusted, security measures that depend on kernel services, such as antivirus programs, should be assumed compromised as well.

The Windows OSs have long allowed changes to the kernel through device drivers. (This model is changing with Windows Vista.) Device drivers, such as printer drivers, have been allowed to run in kernel mode; the same mechanisms that allow legitimate device drivers to run in kernel mode can be exploited to allow rootkits to run in kernel mode. UNIX OSs are vulnerable to kernel rootkits as well; for example, when key OS binaries are replaced with modified versions.

## **Virtualized Rootkits**

OS virtualization is a technique for running multiple OSs on a single machine. Programs such as VMware and Xen implement a monitor that allows multiple OSs to share the same underlying hardware. This is especially useful for developers who need to work with multiple platforms or users who require tools that run on different OSs. The same techniques, however, can be exploited by rootkits.

 For more information about virtualization, see the Xen project at <http://www.cl.cam.ac.uk/research/srg/netos/xen/> and for VMware, see <http://www.vmware.com/>.

A virtualized rootkit is installed by modifying the boot sequence of a device so that a virtual machine monitor with the rootkit functionality is loaded first and then the target OS is loaded. In addition to the target OS, an attack OS can be loaded to perform tasks on behalf of the attacker, such as launching DoS attacks or mass-mailing spam and email messages.


 For details about how a virtualized rootkit can be implemented, see “SubVirt: Implementing Malware with Virtual Machines” at <http://www.eecs.umich.edu/virtual/papers/king06.pdf>.

Reinstalling the OS and any compromised applications can eliminate the types of rootkits described thus far. Such is not the case for firmware-based root kits.

---

## Firmware-Based Rootkits

Disk storage is the most commonly used method for maintaining the persistence of malware, but it has disadvantages from the perspective of the attacker because OSs are so easily reinstalled. Security researcher John Heasman has demonstrated how the advanced power configuration interface (APCI) and PCI devices with flashable expansion ROM can be used to persist rootkits.

 For more information about Heasman's findings, see [http://www.ngssoftware.com/jh\\_bhf2006.pdf](http://www.ngssoftware.com/jh_bhf2006.pdf) and [http://www.ngssoftware.com/research/papers/Implementing\\_And\\_Detecting\\_A\\_PCI\\_Rootkit.pdf](http://www.ngssoftware.com/research/papers/Implementing_And_Detecting_A_PCI_Rootkit.pdf).

When firmware storage devices are used to store rootkit code, reinstalling the OS does not eliminate the rootkit threat. The threat of persistent rootkits even after OS reinstallation is costly. For example, the U.S. Commerce Department has gone so far as to replace computers compromised with rootkits and spyware, according to *Information Week* (Source: <http://www.informationweek.com/management/showArticle.jhtml?articleID=193105227&subSection=Global>). Attackers are using a variety of techniques to hide their malicious code, and multiple countermeasures are required to reduce this threat.

## Blocking and Removing Rootkits

Several steps should be taken to prevent a rootkit compromise. First, administrator or root privileges should be strictly limited. Installing a virtualized rootkit requires privileged access as does installing device drivers that can be a venue for rootkits. Second, keep OSs patched. Vulnerabilities in an OS can be exploited by attackers to install rootkits and other malware.

Next, use host-based intrusion detection. These systems create cryptographic signatures for critical files that can be used to detect tampering. The signatures should be generated after a fresh install of the OS and applications and again after any updates.

Systems administrators should also scan for rootkits. Although virtualized rootkits and kernel rootkits may not be readily detected, library and application rootkits can be detected with scanning programs. In some cases, rootkit scanners can detect the more challenging types of rootkits as well by using information collected directly from the disk without depending on the OS.

In addition to anti-malware suites that may help block rootkit infection, a number of tools are available for detecting rootkits:

- Rootkit Revealer (<http://www.microsoft.com/technet/sysinternals/utilities/RootkitRevealer.mspx>)
- Rootkit Hunter ([http://www.rootkit.nl/projects/rootkit\\_hunter.html](http://www.rootkit.nl/projects/rootkit_hunter.html))
- GMER (<http://www.gmer.net/>)

The history of malware is replete with examples of attackers adapting to countermeasures and deploying more advanced, difficult-to-detect malicious code. The same trend will likely be followed with rootkits; the potential for virtualized rootkits and rootkits persisted in ROM are just two examples of the types of emerging enhancements you can expect to see with rootkits.

---

## Article 14: Vulnerability Scanning 101

Vulnerability scanners have come a long way. When tools like SATAN first came out, there was a lot of discussion about the wisdom of having vulnerability scanners. After all, these were tools for hackers to employ to attempt an attack on your network and servers. Proponents of the tools argued that it was better to learn about your vulnerabilities with a tool rather than through an attack. The debate about the relative value of vulnerability scanners is essentially over—they are useful tools for network and systems administrators in spite of the fact that they could be used for malicious purposes. This article will examine

- The purpose of vulnerability scanners
- How vulnerability scanners relate to other security tools
- Architectural options for implementing vulnerability scanners

### Purpose of Vulnerability Scanners

The purpose of vulnerability scanners is to identify weaknesses in software and configurations. These can be known software vulnerabilities, such as those tracked in vulnerability databases, or system configurations that, although not incorrect, create potential weaknesses in the system. Scans may be performed on multiple types of assets:

- Operating systems (OSs)
- Network devices
- Third-party applications
- Custom applications

OS scans can look for missing patches, questionable services or daemons, improper registry or other system settings, and applications installed on the system that may present problems. Network devices can be scanned for several vulnerabilities, including open ports, improper configuration settings, and missing patches. Third-party applications, such as Web servers, application servers, and databases may be scanned for vulnerabilities. These types of vulnerabilities are specific to the type of applications. For example:

- Web servers that support the TRACE request can be used in cross-site scripting attacks.
- Microsoft IIS Web server supports http access through multiple file types, such as Internet Databases Connectors (IDC), which if unpatched, can allow attackers to execute arbitrary code on the server.
- Several known vulnerabilities in the Simple Network Management Protocol (SNMP) can allow attackers to launch Denial of Service (DoS) attacks or gain access to the server.
- Copy protection mechanisms used by some music publishers have known security vulnerabilities.

---

Custom applications should be analyzed using source code analyzers. These tools work like compilers, at least in the early stages, by lexically scanning and parsing code. Some tools build an internal representation of the code that is then analyzed with a database of security rules. These are not vulnerability scanners as you typically think of them, but they serve much the same purpose when working with custom applications.

At minimum, a security scanner should offer:

- Comprehensive and up-to-date vulnerability database
- Ability to discover network accessible assets
- Ability to determine vulnerabilities given the current software and configuration
- Capacity to make remediation recommendations

After the minimum requirements have been met, other features to look for include:

- Prioritizing vulnerabilities
- Support for compliance audits
- Scalability
- Integration with service support applications
- Ability to schedule scans
- Management reporting tools
- Multi-platform scanning capabilities

## How Do Vulnerability Scanners Relate to Other Security Tools?

Vulnerability scanners complement other network security tools such as:

- Packet sniffers
- Port scanners
- Intrusion detection and prevention systems

### ***Packet Sniffers***

Packet sniffers are used to analyze network traffic. They are often used to help identify bottlenecks in the network and can help debug and troubleshoot Web applications. Attackers can use packet sniffers to gather information about a network. Vulnerability scanners do not analyze traffic but can help identify vulnerabilities in network applications that can be exploited by an attacker with access to information collected with tools such as packet sniffers.

---

## **Port Scanners**

Port scanners check open TCP and UDP ports by sending packets to ports and checking for any response. Depending on the type of packet sent (for example, SYN or FIN packets), the information returned will vary. Port scanners are useful systems administrator tools for verifying which ports are opened and which are closed. A vulnerability scanner goes further by providing information about any known vulnerabilities associated with open ports.

## **Intrusion Detection and Prevention**

Intrusion detection and prevention systems monitor the state of a network or host and determine whether suspicious activity occurs. This type of detection is useful for detecting and blocking attacks as they occur and is one of the few defenses for zero-day attacks, which target previously unknown vulnerabilities. The goal of using a vulnerability scanner is to detect weaknesses in a network or host before it is exploited for an attack.

## **Architectural Options**

Security scanners are available in three architectures:

- Software only
- Network appliance
- Hosted service

Software scanners run on your hardware. This option allows you to run the scanner on the same server as other applications, allowing for optimal use of hardware. As long as peak times for each application do not overlap, this setup can be a cost-effective way to start with vulnerability scanning. This is especially true if you start with an open source application.

Network appliances also have their advantages—particularly simple management. Devices are added to the network and managed remotely. They can be configured to update the vulnerability database, download patches, and perform other routine tasks without concern for other services running on the server.

Hosted services effectively eliminate the need to deploy and manage hardware in return for allowing the provider to scan your network. Understandably, this setup will leave some network administrators uncomfortable. Nonetheless, this may be an ideal option for small businesses or large organizations that can leverage the economies of scale and specialized expertise of a security service provider.

Vulnerability scanners are one of many tools in network administrators' and security professionals' toolbox, and one that can prevent problems before they occur.

---

## Article 15: Web Application Testing

Testing is an important part of any software development methodology, but testing security features is essential for Web applications. Those who come from a software development background are familiar with functional testing: start with required functions, formulate test plans, and define test cases for each feature. Ideally, these steps are automated in a regression test that is run routinely to make sure you do not lose ground as you correct errors. Just as important, and more important if you have to answer to auditors, is testing a key non-functional requirement—security.

Security testing touches on many aspects ranging from technologies deployed with an application to user behaviors with the system. Web applications, in particular, bring risks not found in other types of applications. For example, with client-server applications, a significant barrier to entry was the fact that a custom application was needed to access an application. For example, a payroll application may have required a Visual Basic or Oracle Forms program locally installed to use the back-end system. Web applications, in contrast, are readily accessible to anyone with a browser.

To adequately test Web applications, you need

- To understand the kinds of security risks posed by Web applications
- Tools for testing the security of applications
- Methodologies for managing the process and ensuring testing quality

Understanding the risks to Web applications is a continuous process. Seemingly well-developed countermeasures to threats have to change sometimes, such as the radical shift in antivirus technology that was needed to deal with mutating viruses (for example, polymorphic viruses). In other cases, new threats emerge, such as botnet-generated spam, that is more difficult to shut down at the source than more centralized spam generators. With Web applications, as new features and programming methods evolve, such as AJAX, exploits are soon to follow.


Testing is always a combination of careful, thoughtful procedures and high volume, mind numbingly repetitive tasks. Fortunately, tools exist to automate the former so that you can concentrate more time on the latter. Security testing methodologies, like their software development counterparts, have matured and resources are available to security and systems management professionals to help establish and execute comprehensive security testing.

---

## Watch Out for Top Vulnerabilities

In spite of the fact that vulnerabilities continue to emerge and threats are evolving to exploit them, there seem to be certain vulnerabilities that occur frequently enough to warrant testing in every Web application. The following list is comprised of the top-10 vulnerabilities identified by the Open Web Application Security Project (OWASP):

1. Unvalidated input
2. Broken access controls
3. Broken authentication and session management
4. Cross-site scripting
5. Buffer overflow
6. Injection flaws
7. Improper error handling
8. Insecure storage
9. Application denial of service
10. Insecure configuration management.

 OWASP will be releasing an updated top 10 list for 2007. Visit [http://www.owasp.org/index.php/Main\\_Page](http://www.owasp.org/index.php/Main_Page) for the latest news from this project.

Given that the types of vulnerabilities found in Web applications are numerous and constantly changing, how can one keep up? Automation is essential.

## Tools for Testing

Many tools are available for testing Web applications, and these tools fall into several categories. At one end of the spectrum are developer-oriented tools such as the Lightweight Analysis for Program Security in Eclipse ([LASPSE](#)) and [Solex](#), two plug-ins for the popular Eclipse development environment. Tools such as Brute Force Tester ([BFBTester](#)) can provide automated checks for common vulnerabilities such as buffer overflows. Application testing tools include the [CAL9000](#) tool from OWASP that supplement the functions of automated scanners. There are many other open source tools, including Web Application Testing in Ruby ([Watir](#)). There are also vulnerability scanning tools for Web servers, such as [Nikto](#).

Tools alone are not enough. Just as banging on a keyboard will not produce a Shakespearean sonnet, running testing tools without a formal process will not likely improve security. Formal methodologies help to ensure that you derive the maximum benefit from the tools you use.


---

## Testing Procedures and Methodology

Security is often lumped into the “non-functional” requirement category because it is not an isolated feature that can be checked off as present or absent. Security is a function of how systems are designed and built; thus, testing security must be methodical and comprehensive. Fortunately, a well-developed testing methodology is available from the Institute Security and Open Methodologies, known as the Open Source Security Testing Methodology (OSSTM).

The OSSTM breaks down testing into broad functional areas:

- Information and data controls
- Personal security awareness
- Fraud and social engineering controls
- Computer and communication networks
- Physical security access controls

 For complete details about the OSSTM, see <http://www.isecom.org/osstmm/>. Also, OWASP has released version 2 of its testing guide and is available at [http://www.owasp.org/index.php/Category:OWASP\\_Testing\\_Project](http://www.owasp.org/index.php/Category:OWASP_Testing_Project).

## Summary

Testing Web application security is challenging—there are many potential vulnerabilities and the process requires a number of tools as well as formal procedures to ensure that all essential steps in the process are completed. There are, however, tools at your disposal that allow for automated testing, beginning with the development process. In addition, you also have methodologies that can provide the basis for formalizing the entire testing process.

---

## Article 16: Web Services Security

Web services are an established method for building distributed and federated applications. Using Web services protocols, developers can provide access to application functions by publishing the interface to the service using the Web Services Definition Language (WSDL), providing data in XML structures, and transmitting data between applications using the Simple Object Access Protocol (SOAP). Service consumers can discover Web services that have been registered using the Universal description, discovery, and integration (UDDI) protocol. As with any application, questions of authentication, authorization, and trust must be addressed in the Web services architecture.

In addition to the basic Web services protocol designed for packaging, sharing, and discovering information and services, security mechanisms have been developed. Three of these are the

- Security Assertion Markup Language (SAML)
- WS-Security
- WS-Trust

Each of these provides a subset of necessary security services.

### SAML

Web services can provide varying levels of information and functions depending upon the user or proxy for a user that requests the information. For example, a manufacturer may provide a Web service to all customers that accepts a product code and returns a product description and price. For business partners with established business ties, the same Web service could provide additional details, such as cost to that partner under current contract, delivery options, volume discount schedules, and any other details not shared with other customers.

This kind of arrangement has a number of advantages. The manufacture maintains a single Web service that provides varying responses based on the customer using the service. There is no need to maintain a separate Web service for every conceivable usage/response scenario. For customers, as their relationship to the manufacturer changes, their application interfaces will not need to change except to make use of the additional information available to them.

Ensuring that such an information-sharing policy works correctly requires the Web service to verify:

- Authentication
- Authorization
- Attributes

---

## **Authentication**

Authentication is the process of verifying the identity of a user or a proxy. SAML provides a mechanism for asserting the identity of the agent making the Web service request. The source system of the Web service request provides the SAML data asserting the identity of the user, and the Web service then determines whether to accept the assertion. If the assertion about identity is accepted, access to information and functions can be refined based on the authorizations granted to that user.

## **Authorizations**

Authorizations are entitlements to do certain things. For example, a user may be authorized to place orders up to a certain value or to request information about past transactions. The authorizations can be asserted using the Extensible Access Control Markup Language (XACL), another security standard, which allows for access control decisions and policy enforcement within a federated application.

## **Attributes**

Attributes allow one system to assert information about roles and privileges of a user or proxy that can then be used by the target system. For example, a user may be authenticated on a source application with a manager role; that application then formulates a Web services request and asserts the user is a manager. The target application can then use (or not) this attribute to determine whether additional access to information or functionality is warranted.

SAML provides the foundation for establishing identities across organizational boundaries. This foundation is based on trust between the organizations in this federated model. One of the elements of that trust is the assurance that information is exchanged securely; for that, you need another standard, WS-Security.

## **WS-Security**

When transmitting information to and from Web services, you need to ensure two things: the data is not tampered with en route and confidential information is not compromised during transmission. WS-Security includes mechanisms to do both.

The XML Signature standard defines the protocol for creating and representing digital signatures. If the message is compromised en route, decoding and comparison of digital signature information will reveal the tampering.

The XML Encryption standard addresses issues of encoding data and encapsulating it within XML structures for transmission. This standard is used with SOAP messages to ensure confidential information is not disclosed after leaving its source or before arriving at its destination. A third element of Web services security is WS-Trust.

---

## WS-Trust

The WS-Trust language is a protocol for requesting and using credentials from a trusted provider. WS-Trust is built on WS-Security. This protocol is useful, for example, because a Web service would only need to trust the WS-Trust credentials provider, not every separate entity that requests credentials.

By combining the ability to establish identity across boundaries, protect the integrity and confidentiality of information exchanged, and form trust relationships, existing Web services standards are providing the foundation for securing federated application development.

---

## Article 17: When Patching Is Not Enough: Zero-Day Threats

Zero-day threats were in the press over the past month as previously unknown vulnerabilities were exploited in Microsoft Office products. Zero-day attacks get their name from the fact that there are zero days between the time a vulnerability becomes known to software developers, security researchers, and application users and the time the vulnerability is exploited on fully patched and updated devices. The problem is not limited to Microsoft products, and it would be unwise to assume that using alternative products alleviates the zero-day vulnerability problems.

This article will examine zero-day threats from three perspectives:

- What kind of zero-day threats have occurred?
- What can be done to minimize the problem in the short term?
- What are the longer-term challenges to mitigating the risk of zero-day threats?

Let's begin with some recent history.

### What Kinds of Zero-Day Threats Have Occurred?

The most well-known zero-day attacks have targeted desktop applications. Microsoft products are common targets, but other applications are vulnerable as well. Some of the zero-day vulnerabilities include:

- A vulnerability in Windows Shell that allows remote code execution
- A vulnerability in Microsoft PowerPoint that allows remote code execution
- A vulnerability in Microsoft Internet Explorer (IE) that allows attackers to gain control of the compromised machine
- A vulnerability in Linux RealPlayer that allows attackers to execute commands remotely

These vulnerabilities are not necessarily different from others that are found and disclosed by developers and security researchers. The distinctive characteristic of zero-day vulnerabilities is who finds them first.

---

As in other areas of software engineering, tools can make vulnerability testers much more productive. Programs that test for vulnerabilities are called *fuzzers* or fault injectors. Their objective is to send input to a program that causes the program to fail or generate some type of fault exception. For example, a fuzzer might send a long input string to a Web form and force a buffer overflow exception in the application. Other fuzzers might specialize in database applications and send common forms of SQL injection attacks to a database application. In theory, any program that accepts input—regardless of whether it is a Web form, desktop application, database listener, or firewall—can be tested with a fuzzer.

In the early days of malware, programmers with knowledge of operating systems (OSs) and computer architecture were the ones building the first viruses; programmers that understood the intricacies of network protocols and network application vulnerabilities wrote the first worms. Before long, tools were created that allowed anyone that could run a program and select a few options to create a virus. Fuzzers are widely available on the Internet, making knowledge of vulnerabilities accessible to those with limited programming skills. Fortunately, these tools do not produce code that exploits those vulnerabilities.

It's prudent to assume that with the availability of tools, the ability to share exploit code, and the increasing influence of economic motivations driving cyber attacks, you will continue to hear about and have to live with zero-day exploits. The question is: What do you do about it?

## Reducing the Threats of Zero-Day Exploits in the Short Term

There are no direct solutions to zero-day threats, so you need to concentrate on indirect measures, including:

- Patching applications and OSs
- Using intrusion prevention
- Using malware detection
- Limiting privileged access rights

By definition, patching will not eliminate the vulnerability used by a zero-day exploit. Nonetheless, you need to keep OSs and applications patched. There are always some preconditions on the ability to exploit a vulnerability—for example, a user has to browse to a Web site with infected malware, someone has to open a compromised document in email, or an application has to be written in such a way to allow an attacker to reach the vulnerable code. As long as known vulnerabilities are patched, you reduce the number of possible paths to unknown vulnerabilities. This is not a panacea by any means, but it might limit an attacker's options.

Another step you can take is to deploy intrusion prevention systems (IPS). Both host- and network-based intrusion detection can be used to detect and block anomalous behavior. It's less than ideal because if an IPS is detecting a problem, the vulnerability has already been exploited.

---

Anti-malware programs may be able to detect exploit code when analytic or behavior-based detection techniques are used. Signature-based detection, a staple of antivirus detection for years, only works when you know what to look for. If you know what to look for, you would know about the vulnerability; thus, as with intrusion prevention, if you have a signature for an attack, it has been detected in the wild and antivirus vendors have had time to create a signature. With the potential for fast-spreading attacks, behavior-based detection becomes the first line of defense from anti-malware tools.

Vulnerabilities that allow for remote code execution are especially dangerous when they can execute with full privileges to the OS. If a vulnerability allows for remote commands and those commands execute in the user space of a limited-privilege user, the damage that can be done is reduced. Files can be deleted and information stolen but without appropriate privileges, a process cannot update the registry or perform other actions with system-wide impact.

These measures can be implemented today to help mitigate the problem of zero-day threats. You will not eliminate the threat of zero-day exploits, but in the long run, you will be better prepared for them.

## Reducing the Threats of Zero-Day Exploits in the Long Term

One of the reasons there are so many vulnerabilities in software is the complexity of software. (If you have doubts about the number and scope of vulnerabilities, browse one of the vulnerability databases, such as <http://nvd.nist.gov/> or <http://www.kb.cert.org/vuls/>.) Ironically, the one element of the long-term response to these vulnerabilities is with more complexity.

Complex systems may be brittle (like some software) or robust (like many organisms). Brittle systems tend to break when external conditions vary from the expected; robust systems continue to function in spite of such changes. This is not to say that robust systems keep doing the same thing—they may change how they accomplish their objectives, but what they accomplish remains stable. Biological systems have a variety of ways of compensating for changes in the environment. Under stress conditions, an organism's metabolism may change to conserve resources or maintain a certain state. Software systems should exhibit the same type of response.

IPSS display the basics of this concept. When an apparent intrusion is detected by a monitoring system, a response is generated that blocks the intrusion. Developing more robust software systems will require even more complex software than exists today. But given the inability to write complex software without vulnerabilities, how can we expect to produce something even more complex without even more vulnerabilities?

---

There are two answers to that question. First, we do need to write better software. An old quip among software engineers is that if builders built buildings the way programmer program software, one woodpecker would have destroyed civilization. We know how to program and we know how to test programs. Better software engineering discipline is part of the solution. However, we need to understand that there are constraints on developers—with enough time and resources, we can push software quality to whatever level we choose. Conversely, without sufficient time and resources, we are not going to achieve the levels we want. This brings us to the second answer.

Software need not be completely vulnerability free if it is robust enough to respond to failures. Again, we need to look to biological systems for examples. Organisms are subject to all kinds of threats, yet they have survived and evolved in spite of their limitations and their own vulnerabilities, such as harmful mutations and limited defense mechanisms. Those of us who have spent decades in software engineering and related fields are all too familiar with the limitations of our craft. It is time to look beyond our traditional boundaries to find examples of complex yet robust systems. We will probably never eliminate zero-day threats; we now need to find ways to live with them, both in the short term and the long term.

---

## Article 18: Configuration Management and Security

Information security is commonly described through the adage “as strong as the weakest link”—and too often the weakest link is in systems configurations. Regardless of all the time and money poured into anti-malware, intrusion prevention, content filtering, and all the other measures we deploy, if configuration is not controlled, our networks and systems will be vulnerable.

Configuration management is one of the areas of information security that often falls on systems managers and network administrators. Many of the security-oriented tasks are also applicable to good systems management practices; this fact just adds weight to the notion that good systems management is good security. Much has been written about effective configuration management practices and comprehensive best practices are readily available. Rather than delve into the details of these broad frameworks, this article will focus on several basic functions and areas that are critical to leveraging the benefits of configuration management to improve security:

- System hardening
- System settings
- Access controls
- Authorized hardware and software
- Configuration auditing

These are just a subset of all the elements of configuration management, but these are some of the most important from a security perspective.

---


## Hardening Systems

When new systems are procured, one of the first steps should be to harden the operating system (OS); that is, to remove unnecessary system services and applications to reduce and eliminate potential vulnerabilities. The process will vary by type of OS, but some common steps include:

- Changing any default passwords
- Minimizing the number of accounts with administrator/root privilege
- Establishing strong password policies
- Removing any unnecessary system services or daemons from automatic startup
- Removing unwanted applications from servers that do not need them.
- Reducing unnecessarily broad access controls
- Encrypting the file system
- Removing unnecessary users or groups
- Revoking unnecessary privileges from user accounts
- Disabling clear-text IP protocols used for authentication
- Password protecting boot operation controllers
- Enabling appropriate event logging
- Disabling anonymous ftp access

In addition, vulnerability scanning programs may be able to detect remaining applications that can be eliminated or require patching.

This process of stripping out unnecessary applications and minimizing authorizations before placing a device on the network helps to protect it and other assets. A further benefit is that these devices can be easier to manage—there are fewer applications to patch and users are limited in their ability to inadvertently, or intentionally, damage the system.

 For more information about hardening Linux, UNIX, and Mac OS X, see the Bastille Project at <http://www.bastille-linux.org>; for Windows, see the Windows XP Security Guide at <http://www.microsoft.com/technet/security/prodtech/windowsxp/secwinxp/default.mspx>.

---

## System Settings

Configuration management requires controlling system settings. In the case of Windows OSs, this means controlling the registry; in UNIX and Linux-based systems it means controlling system configuration files. During the device's initial deployment process, the system should be hardened as described earlier and then monitored for all changes.

Monitoring can be done automatically with host intrusion detection tools. These work by calculating a message digest for configuration files when first installed and then periodically recalculating. If there is a difference, a change has been made. Of course, the changes could be legitimate, but the value of these tools is detecting unauthorized changes. They are not silver bullets, however.

An inexperienced systems manager (and yes, even experienced ones) can easily make a mistake when changing a configuration, especially when changes are applied to multiple devices. To minimize the chance of this kind of configuration error, policy-based configuration changes can be used. These allow administrators to define changes, review the changes, and then apply them consistently across all target devices.

## Access Controls

Access controls begin with physical access. Although servers are often physically secured in locked data centers with limited access, client devices such as desktop workstations and laptops can also be physically secured. Mobile storage devices such as USB memory devices, external disk drives, and even iPods, can be used to copy and remove sensitive and confidential information. Devices should be configured to control the use of mobile storage devices.

Logical access controls should also be configured. These can be used, for example, to limit use of devices to specific groups and to specific times of day. This is one area in which the responsibilities between systems management and network management begin to overlap. For example, laptop users may be granted remote access to network resources if a VPN is used and if the laptop is properly configured (for example, has up-to-date anti-malware, is properly patched, and so on).

## Authorized Hardware and Software

Defining and enforcing policies about what kinds of hardware and software are allowed on devices can help to improve security. For example, policies about the use of instant messaging, peer-to-peer file sharing, browser plug-ins, and other restricted software can prevent the introduction of vulnerable or compromised applications. Similarly, restricting the use of removable storage, such as the use of USB memory devices, can reduce the chances of data loss as well as the chance of introducing malicious code to a device.


Users can unknowingly introduce vulnerabilities by adding hardware to a network. For example, an unauthorized wireless access point might be set up temporarily as a convenience for a project team and, in the process, open the network to potential attacks. Configuration management should also include some measures to detect devices that should not be on the network as well as verify the configuration of devices that should be there.

---

## Configuration Auditing

Configurations will change over time. New requirements emerge, additional applications are installed, and administrators make “temporary” changes to address immediate problems. Without regular reviews of configurations, these changes can become permanent and leave systems vulnerable, in some cases, and more difficult to manage in many cases.

Configuration management is an essential element of system security. By focusing on system hardening, system settings, access controls, authorized hardware and software, and finally configuration auditing, system managers can go a long way to support system security.

 For more information about configuration management, see the Institute of Configuration Management Web site at <http://www.icmhq.com/whitepapers.htm> and the Security Configuration Checklists from NIST at [http://checklists.nist.gov/download\\_sp800-70.html](http://checklists.nist.gov/download_sp800-70.html).