



# Testing Your Security Configuration

*by Roger A. Grimes,*  
*CPA, MCSE: Security (2000/2003/MVP), CISSP, TICSIA*

Hackers and malware (i.e., malicious programs) are using scanning techniques to find vulnerable computers and networks to exploit. If a vulnerable computer is attached to the Internet, it's usually only a matter of days or hours until a hacker exploits it. Vulnerabilities happen because of

- software bugs
- malware
- poor patch management
- misconfiguration
- security policy weaknesses
- social engineering

Each of these issues can be minimized using security configuration testing along with other offsetting defenses.

Network administrators must audit and test their own network to discover the vulnerabilities before the hackers do. A company's security policy can be tested using an IT security policy audit, vulnerability assessment scanning, and penetration testing.

## → Contents

Identifying and Fixing Vulnerabilities .....	2
<i>Understanding How Vulnerabilities Occur</i> .....	2
<i>Identifying and Mitigating Vulnerabilities</i> .....	3
<i>Vulnerability Assessment and Penetration Testing Methodology</i> .....	4
Vulnerability Assessment Tools ....	5
<i>Microsoft Security Configuration Testing Tools</i> .....	6
Evaluating Your Security Investment .....	7
Summary .....	8
Other Resources .....	8
About the Author .....	8

**Microsoft**<sup>®</sup>

Copyright 2005 Microsoft. All rights reserved

# Testing Your Security Configuration

An IT security policy audit samples and audits an IT security policy. If a security policy is appropriately written, followed, and applied consistently, the risk of a successful exploit is minimized. An IT security policy audit can also help minimize the effects of a social engineering attack.

A vulnerability assessment scan uses software to search for known vulnerabilities on one or more computers. They are an efficient way to check for vulnerabilities and to develop comparative metrics.

A penetration test is a manual test conducted by a trained professional with hacking expertise. Administrators should conduct IT security policy audits and penetration testing at least once per year and perform vulnerability assessment scanning more often. You can use the latter tool to confirm critical patch effectiveness and deployment success.

When you find vulnerabilities, you need to remedy them using acceptance, mitigation, transference, or avoidance. A security investment strategy should seek to minimize security risk, which can never be completely eliminated, to an acceptable cost/benefit level. The security investment strategy should be reviewed and updated at least annually. Most companies can improve their strategy by re-focusing on low-cost basics and spending fewer resources on unproven, but more expensive, devices and software.

## Identifying and Fixing Vulnerabilities

Every network administrator is tasked with fixing vulnerabilities on the systems under their control by understanding how vulnerabilities occur and identifying and mitigating vulnerabilities. It's crucial to understand that you can't eliminate all potential vulnerabilities on a computer unless you disconnect it from the network and lock it in a closet. Any input mechanism is a potential security hole. Outside of a complete lockdown, the goal is to minimize the most substantial risks. The first step is to understand how vulnerabilities occur.

### Understanding How Vulnerabilities Occur

Vulnerabilities happen because of

- software bugs
- malware
- poor patch management
- misconfiguration
- security policy weaknesses
- social engineering

You can use security configuration testing and other off-setting defenses to minimize each of these concerns.

**Software bugs.** Ultimately, many vulnerabilities wouldn't happen if the software was coded better. But perfectly coded software is a theoretical concept that doesn't exist in the real world. Code flaws are measured in bugs per thousand lines of code ([http://www.sei.cmu.edu/news-at-sei/columns/watts\\_new/2004/1/watts-new-2004-1.htm](http://www.sei.cmu.edu/news-at-sei/columns/watts_new/2004/1/watts-new-2004-1.htm)). Carnegie Mellon University's Software Engineer Institute (<http://www.sei.cmu.edu>), which did the most respected study on the subject, claims the average commercial software program contains 15 bugs per 1000 lines of code. Microsoft Windows has tracked significantly better build quality than the expected estimate because of its quality software engineering controls and secure coding methodologies. Still, bugs do happen that allow security exploits.

Software vulnerability sleuths will often either contact the vendor or publicly announce the vulnerability and the consequent exploit to a mailing list. If the individual announces the vulnerability to the vendor first, the vendor can create and test a patch to release to its customers. In either case, history has proven that once the vulnerability and exploit have been publicly released, patch available or not, malicious exploits using the vulnerability are not far behind. For this reason, administrators should subscribe to one or more security vulnerability mailing lists (see the

Resources section below) and follow a strong patch management strategy.



#### Note

If you or your company writes code for a living, all programmers should be trained in writing secure code. *Writing Secure Code* by Microsoft's Michael Howard and David LeBlanc (<http://www.amazon.com/exec/obidos/ASIN/0735617228>) is an excellent guide.

**Malware.** Malware lets hackers quickly find and exploit vulnerabilities. For example, if a buffer overflow is found in an exposed service, it's usually only weeks to months before a malicious coder automates the process. If a hacker can trick a user into executing malware, the hacker can create a vulnerability where none existed before. Most malware programs come in the form of a virus, worm, Trojan, buffer overflow, or other type of hybrid. The natural defense against malware is proactive and reactive malware scanning tools. The former prevents recognized malware from being installed and executed in the first place, and the latter passively warns the user of the malware's existence.

**Poor patch management.** Most security vulnerabilities have available patches to close the exploitable holes for weeks to months before the malware is released. Unfortunately, patching can be difficult to manage—administrators must test each patch and apply it with roll-back strategies in case it unduly affects legitimate operations. Microsoft's Software Update Services (SUS) and Microsoft Windows Server Update Services (WSUS—<http://www.microsoft.com/windowssserversystem/updateservices/default.aspx>), as well as System Management Server (SMS <http://www.microsoft.com/smsserver>) have proven valuable in many company's patch management strategies.

**Misconfiguration.** A hacker can exploit a fully patched system with no known vulnerabilities if that system isn't properly configured. These types of vulnerabilities can only be fought by educating administrators and support staff in appropriate configuration settings and in change management. Microsoft provides many documents and security templates to aid in the appropriate security baselines at <http://www.microsoft.com/security>. For example, the Windows XP Security Guide (<http://www.microsoft.com/downloads/details.aspx?familyid=2d3e25bc-f434-4cc6-a5a7-09a8a229f118&displaylang=en>) provides a comprehensive list of security recommendations for Windows XP Pro clients. Many other organizations, such as the

Center for Internet Security (<http://www.cisecurity.org>) and the National Institute of Standards and Technology (<http://csrc.nist.gov>) contain excellent baseline recommendation documents, as well.

**Security policy weaknesses.** A corporate security policy communicates the security policies and procedures that all employees should follow in regard to covered assets. Security policies are customized for each environment and should reflect an organization's commitment to secure practices. Because technology is always evolving, along with the subsequent threats, security policies quickly become dated after their approval unless a concerted effort is made to review and update the policies on a regular basis. Policies can also be less effective than desired because poor or weak assumptions, ignorance of a particular threat, or lack of compliance. A policy must be audited to ensure that it's being consistently and appropriately applied. IT security auditing can help affirm adherence to the security policy. Vulnerability assessment scanning and penetration testing, either against all machines or an appropriately selected sample, can help you pinpoint strengths and weaknesses, and let you make corrections.

**Social engineering.** No amount of technology can defeat a correctly implemented social engineering attack. You can have the strongest authentication mechanism in the world, but if your user's implement weak passwords, you can't stop a brute force cracker. If someone can persuade a user to reveal his or her password, even a legitimate complex password won't help. A few studies have been done to show many users are willing to give complete strangers their password in exchange for a low-value gift (<http://news.bbc.co.uk/1/hi/technology/3639679.stm> and [http://www.theregister.co.uk/2003/04/18/office\\_workers\\_give\\_away\\_passwords](http://www.theregister.co.uk/2003/04/18/office_workers_give_away_passwords)).

Administrators can only defeat social engineering by providing end-user education and enforcing good security policies. Most other hacking vectors have other offsetting technical controls, but social engineering subverts all those mechanisms by gaining privileged information from unknowing privileged insiders. Penetration testing should include some form of social engineering testing if it's to be inclusive.

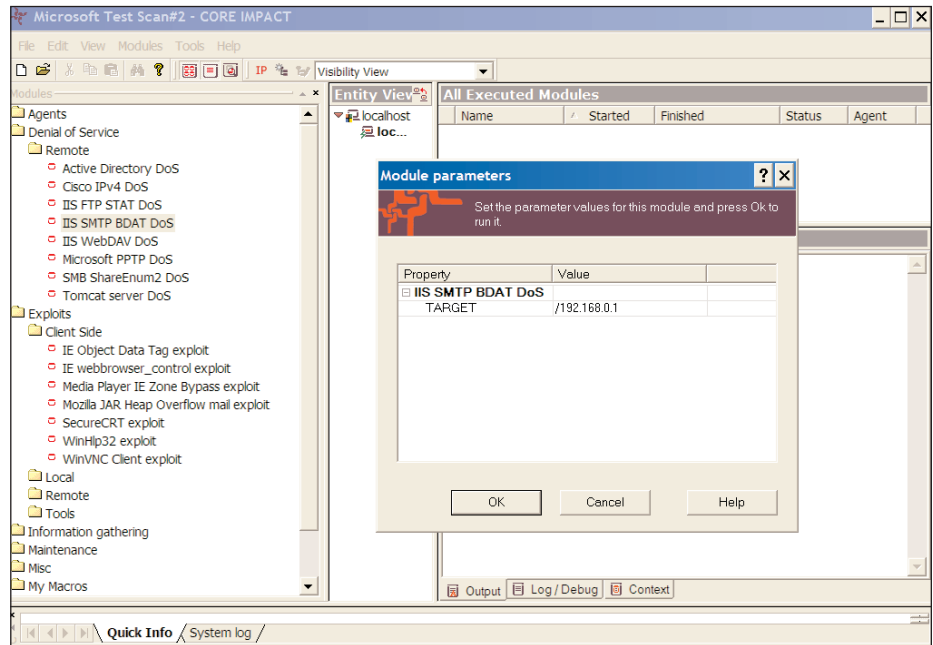


Figure 1: CORE IMPACT vulnerability assessment tool example

## Identifying and Mitigating Vulnerabilities

You can use vulnerability assessment scanning and penetration testing to identify and audit all the vulnerabilities mentioned above. Then, you can rank each identified vulnerability and offset the highest priority items through risk management. Vulnerability assessment scanning is the process of using automated software tools to check for the presence of known vulnerabilities. Figure 1 shows an example of this type of software tool.

Vulnerability assessment scanning can be done by simply checking for the presence of a specific patch on a specific OS or application or by conducting a test that would only succeed if the vulnerability is present. Penetration testing usually involves a skilled vulnerability tester who uses their expertise to manually probe specific computer or human targets.

Both vulnerability assessment scanning and penetration testing have their place in discovering and mitigating vulnerabilities. Vulnerability assessment scanning should be done on a regular basis, optimally at predetermined periods, say once a quarter, against all computers to establish comparative baselines. You can also perform such scanning before or after specific crucial patch roll-outs to establish how many machines need the patches and to document patch success.

Penetration testing frequency depends on whether your company has the internal and external resources to conduct the relevant procedures. Certainly, companies with internal resources can conduct more frequent tests at a

lower overall cost. Penetration testing is usually done less frequently than vulnerability assessment scanning because of the expertise and expense involved, but it can be more accurate on high value assets. Although there are no industry standards, conducting a penetration test once per year to measure security policy compliance isn't unusual. Both vulnerability assessment scans and penetration testing follow a similar methodology.

## Vulnerability Assessment and Penetration Testing Methodology

Let's examine the steps to take when performing vulnerability assessment scanning and penetration testing.

1. Obtain approval
2. Determine scope
3. Develop test strategy
4. Determine reporting requirements
5. Inventory assets
6. Run tests
7. Collect and collaborate results
8. Determine conclusions and report findings
9. Recommend vulnerability fixes

**Obtain approval.** Always obtain appropriate, written approval before beginning and implementing vulnerability assessment scanning or penetration testing. The written approval document should contain the scope of the engagement, what will be tested, how it will be tested, any potential operational interruption issues, and dates and times of the tests. If any of the tests might affect operations, communicate potential problems to the affected managers and users before beginning the scanning/testing process. Invasive vulnerability assessment scanning is well known for unintentionally causing denial of service (DoS) interruptions on tested assets. Consider testing new vulnerability assessment scanning tools during low activity hours until the potential consequences are known. Never, ever begin a test without appropriate approval. Failure to do so could result in dismissal or criminal prosecution.

**Determine scope.** The objectives of the scanning/testing must be understood by implementers and management, and documented. For example, will the test include only certain assets (e.g., only Windows-based computers)? Will the test include network devices, legacy systems, and human elements? Will the test be limited to servers or include all computers? Will all departments be tested or just specific departments? Should the test include physical security, wireless, and remote avenues?

Will the test be black box, gray box, or white box? Black

box testing assumes the tester has no insider information. A white box test provides all the information needed about the network and assets to be measured. A gray box test is somewhere in-between.

A black box test is often conducted externally from a remote location into the private network. It essentially acts as an outside hacker with no specific internal information. A white box test begins inside the network and assumes that the hacker will be successful in penetrating outside defenses and has absolute knowledge of the network. Although a black box test will reveal the risk of an external hack attack, the white box test provides more overall assurance, especially because many attacks occur by insiders using unauthorized access and methods.

Another question to consider is whether to run only tests known to work against a specific platform, say Windows, or run all tests regardless of the known platform. The former testing strategy is based on the idea of security relevance. For example, why run tests known only to work against UNIX machines against Windows? The answer is that security vulnerabilities are often found in the systems you don't expect or know you have. For example, you might think you have only Windows computers, but any appliances (e.g., firewalls, Intrusion Detection Systems, mail gateways) could be running vulnerable versions of Linux, UNIX, or other OSs you never considered.

**Develop test strategy.** After you've set the scope and objectives, you can develop a strategy. A vulnerability assessment scanner runs a predetermined battery of tests using software, while the penetration tester begins with the most fruitful tests first based on his or her expertise and the environment. In either case, the most likely candidates should be tested first, but all computers in the scope and all vulnerabilities in the scope should eventually be tested.

**Determine reporting requirements.** Ultimately, management must read and approve the report. How do they want the report presented? Are summary statistics enough? Do they want the details behind every test and the output report from every computer tested? In most cases, management wants a 1- or 2-page Executive Summary followed by the appropriate supporting detail.

**Inventory assets.** The first step in the actual test is to obtain an updated list of assets within the scope. This list can be a document handed to the tester by management or obtained using an auditing tool. Many vulnerability assessment tools come with inventory programs that will use Internet Control Message Protocol (ICMP) echo requests (i.e., pings) to find assets. Is this acceptable? Many computers come with host-based firewalls that disable ICMP echo

replies by default (as does Windows XP's Windows Firewall). A vulnerability assessment tool that lets you test only those computers responding to a Ping test will miss many active assets. Hopefully, the tool used is flexible enough to allow Layer 2 (i.e., Address Resolution Protocol requests) or TCP or UDP scans to determine available assets; or will allow testing on IP addresses not actively located. Compare any provided inventory list with the findings and note discrepancies.

**Run tests.** The tests should be run against the computers defined in the scope. Most vulnerability assessment scanning software is launched to cover one or more IP addresses or subnets in sequential order. A vulnerability assessment scanning tool should be monitored to make sure it functions appropriately during the test period. They're known to lock up their host machines or the machines they are run against.

**Collect and collaborate results.** After the tests are completed, results should be collected and collaborated. Care should be taken to independently confirm any critical results. For instance, if a particular port or service is found to be vulnerable across multiple machines, ensure it isn't a false-positive result by sampling one or more reported machines. Vulnerability assessment scanners are notorious for reporting false-positives, so a little independent collaboration will go a long way to giving more credence to the results.

**Determine conclusions and report findings.** Your conclusions should recommend closing the most critical issues first. Each found issue should be ranked according to criticality, the severity of potential abuse, the likelihood of occurring, and the value of affected assets. For example, a blank sa password or potential SQL injection attack on a mission-critical SQL Server system exposed to the Internet would most likely be a critical item. Similarly, a potential buffer overflow on an inactive FTP server would rank as a mid- to low-level item. Many times the vulnerability assessment scanning software will provide its own rough estimate to vulnerability criticality; ultimately, however, it's your experience and understanding of the network that should prevail. For example, even a low ranked vulnerability should rank higher if it's across a wide swath of computers or exists on an Internet-facing server. The key is to provide recommendations to management on what to fix first. The lowest hanging fruit isn't always the best or most popular choice.

**Recommend vulnerability fixes.** No security configuration testing report is complete without recommending vulnerability fixes. Security risks are offset using acceptance, mitigation, transference, or avoidance solutions. Acceptance is when the estimated cost to fix the risk far outweighs the potential damage from the vulnerability or if there's no

significant way to lessen the risk. In these cases, the risk is accepted and communicated to management. For example, many forms of popularly downloaded content are often subject to various hacking attacks. You might decide a particular type of content is so embedded within the company that to discontinue its use would do far greater harm to the company's revenues than any resulting exploit.

Mitigation is taking steps to remove the vulnerability. Install patches in a timely manner, configure software and hardware securely using best practice recommendation documents as your guide, and test. Make sure all employees know and follow the company's IT security policy.

Transference involves shifting the risk to another party. For example, you can buy IT security insurance that pays if a particular type of attack is successful or hire another company to manage the security of your network.

Last, avoidance is choosing to remove the potential avenue for the vulnerability before it can be exploited. For instance, running a publicly accessible Web server with read/writable drives on your company's demilitarized zone might expose the company to an unacceptable amount of risk, so the Web server project might be denied. Using acceptance, mitigation, transference, or avoidance, you can minimize your company's security risk.



#### Note

You do not have to develop your testing methodology in a vacuum. Several excellent guides exist, including guides from Microsoft (<http://www.microsoft.com/technet/itsolutions/msit/security/attackandpenetest.msp>), *The Open Source Security Testing Methodology Manual* (<http://www.isecom.org/osstmm>) by Pete Herzog, and guidance provided by NIST (<http://csrc.nist.gov/publications/nistpubs/800-42/NIST-SP800-42.pdf>).

## Vulnerability Assessment Tools

There are literally dozens of vulnerability assessment tools and vendors to choose from, each with its own strengths and weaknesses. First, you must choose between running your own tool or hiring a company that specializes in vulnerability assessment. The latter approach might even be less expensive when you factor in the cost of the software and time involvement. When reviewing a vulnerability assessment tool, consider the following:

- Area and platform of expertise (i.e., Windows)
- Number of vulnerability checks
- Types of tests: Checks for missing patches,

vulnerability checking, misconfiguration checking, banner grabbing, fingerprinting, etc.

- Accuracy level of false positives and false negatives
- Speed
- Does an agent need to be installed on scanned hosts?
- Level of access required (hopefully administrative access isn't needed for all tests)
- Stability of host and scanned systems
- Reports—standard, customized, and export options
- How often is the tool updated, and how?
- Vendor/community support

Many vulnerability assessment tools and vendors work over the Internet. If you choose to run your own vulnerability assessment scanning program, you can choose among dozens of free and commercial programs. As expected, the commercial products typically do a better job and are easier to use than the free counterparts. You can also choose tools that install client-side “agent” software on the hosts to be scanned. Although these types of tools tend to be more accurate in determining vulnerabilities, their increased accuracy might actually result in unrealistic metrics as compared to the true threat. As an overall recommendation, be sure to choose a tool or company that specializes in the platforms covered by the scope of the test. For instance, many vulnerability assessment tools specialize in UNIX exploits. When auditing a Windows computer, you want a tool that specializes in Windows vulnerabilities. The following articles review and rank various vulnerability assessment tools:

- [http://www.windowsitpro.com/Windows/Article/ArticleID/43888/Windows\\_43888.html](http://www.windowsitpro.com/Windows/Article/ArticleID/43888/Windows_43888.html)
- <http://www.nwc.com/1201/1201f1b1.html>
- <http://www.cotse.com/tools/vuln.htm>
- <http://www.sans.org/rr/whitepapers/tools/1060.php>
- <http://www.sans.org/rr/whitepapers/auditing/1353.php>

## Microsoft Security Configuration Testing Tools

Although Microsoft doesn't offer any publicly available tools as complete as some of its security partners, it does

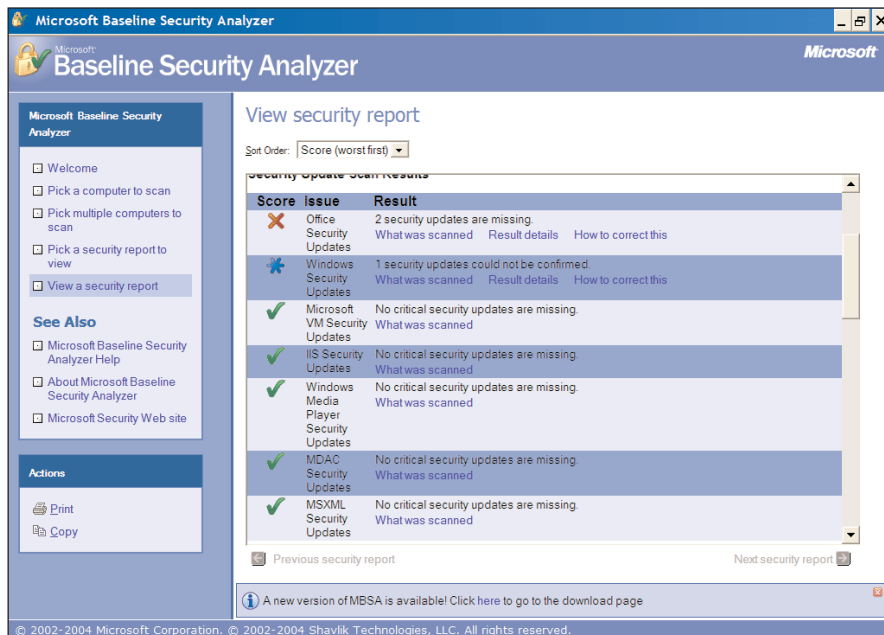


Figure 2: Microsoft Baseline Security Analyzer

provide many programs that can assist in any vulnerability assessment testing program, including the Microsoft Baseline Security Analyzer (MBSA—<http://www.microsoft.com/technet/security/tools/mbsahome.msp>) and the Microsoft Management Console (MMC) Security Configuration and Analysis snap-in ([http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/sag\\_SCMtopnode.asp](http://www.microsoft.com/resources/documentation/WindowsServ/2003/standard/proddocs/en-us/sag_SCMtopnode.asp)). The MBSA tool checks Windows NT 4.0 and later systems for missing patches and common misconfiguration mistakes that can lead to vulnerabilities. Figure 2 shows the MBSA interface.



### Note

**MBSA can scan an NT 4.0 computer remotely, but can't be installed on NT.**

The Security Configuration and Analysis tool lets you compare an existing security template against a particular PC to discover the differences. Figure 3 shows the Security Configuration and Analysis snap-in interface.

You can make a security template that mimics your company's IT security policy and audit the results. If you find a computer out of compliance, you can use the same security template to apply the correct security settings. Whether you use Microsoft's or another third party's vulnerability assessment tools, you need to start testing and auditing your computers before the hackers do it for you.

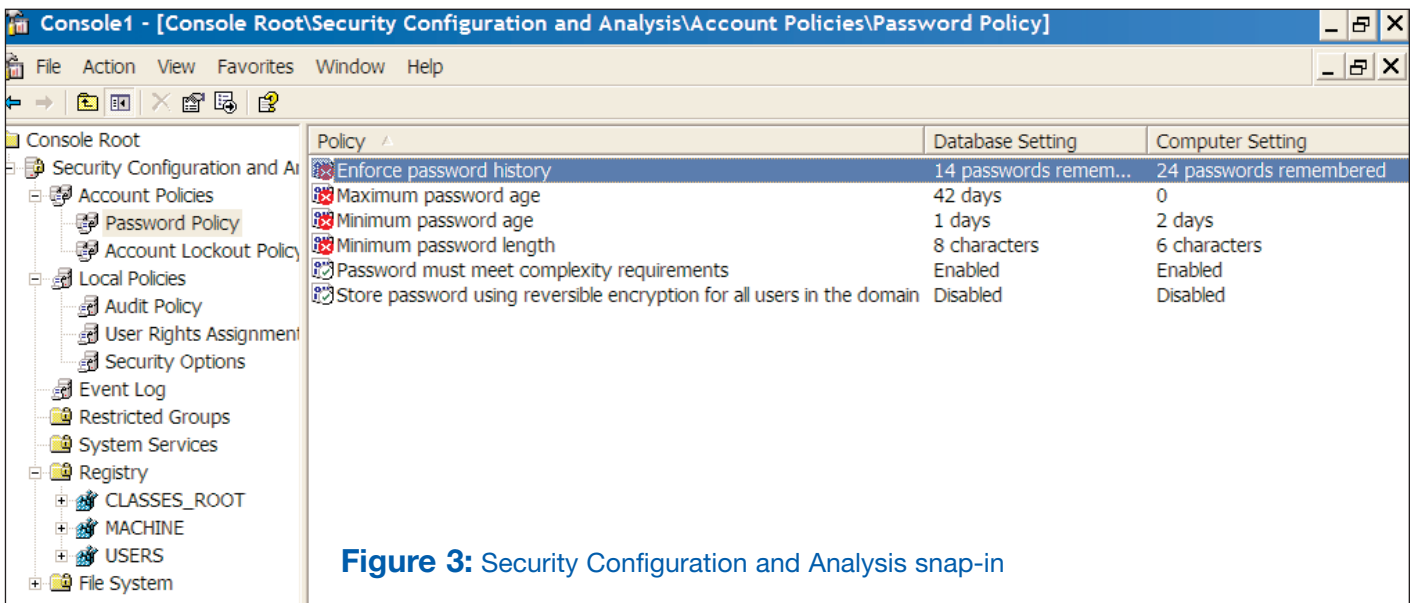


Figure 3: Security Configuration and Analysis snap-in

## Evaluating Your Security Investment

To many CEOs and businesses, security investment is purely a cost/benefit ratio. The defenses you deploy should outweigh the potential cost of not deploying the defenses. For example, you would not deploy a \$100,000 investment in email antivirus technology if the expected losses from an email malware attack were less. At least annually, you should be evaluating your security investment dollars and asking the following questions:

- Is the current strategy working?
- Are expenses being made in the right places?
- How does your company compare with similar companies with similar requirements in the same industries?
- Do expenses more than offset the risk?
- Do you see future malicious hacking trends that indicate additional investment in new areas, or moving resources away from old areas (e.g., macro and boot viruses aren't much of a threat anymore, but wireless attacks are becoming more news worthy)?
- What new training programs do staff and employees need to attend?

First, is the current strategy working? Have the expenditures resulted in the expected expenses and decreased downtime due to successful hack attacks and malware exploits as compared to what was expected. For example, if you spent \$100,000 on a new antivirus solution, did it result in less malware getting executed on desktops? If not, what's wrong with the product or assumption? If the solution isn't as successful as you'd like it to be, is the money being spent

in the right places? A common mistake many companies make is to spend large sums of their investment dollars on "advanced" security tools while neglecting the basics.

For example, they purchase unproven intrusion detection technology while forgetting to ensure that basic Windows permissions are properly configured. Or they spend a large portion of their security dollars trying to prevent the dedicated manual attacker from being successful against their network systems when 99 percent of the attacks are malicious mobile code (e.g., viruses, worms, Trojans). How does your company's costs compare to the industry average? You can measure costs against your industry by reading trade journals and contacting industry associations. Also, high dollar, high-risk trades, such as banking or securities trading, can be expected to spend more on computer security than the home building industry.

After evaluating your current strategy and spending, what changes do you need to make for forthcoming financial periods? Particular types of security threats come in popularity waves. Boot and macro viruses used to be a big problem, but Internet scanning and email worms are the biggest threat right now. Root kits and wireless threats seem to be gaining popularity during 2004 and 2005, while email threats seem to be peaking. Rely in expert articles and opinions about where to invest your future security dollars.

You can easily measure costs quantitatively or qualitatively, but even the qualitative costs have to be imputed to an estimated cost. The basic formula for determining security risk, and hence how much security investment is acceptable, is to estimate the total expected cost due to different vulnerabilities in a given time period as compared to the cost of the defenses to defend successfully against those

Vulnerability Description	Estimated ARO	Estimated SLE	Total Risk
Email virus outbreak	2	\$20,000	\$40,000
SQL Server Compromise	0.05	\$10,000	\$500
Stolen Confidential Secrets	0.005	\$10,000,000	\$50,000
		<b>Total Annual</b>	<b>\$90,500</b>

**Figure 4:** Calculating total risk example

vulnerabilities. You can calculate expected risk by multiplying the annualized rate of occurrence (ARO) for vulnerabilities by their single loss expectancy (SLE). Figure 4 shows an example of measuring ARO against SLE to assess the total risk.

In this example, defenses to prevent the expected security risks should be no more than \$90,500. In most business cases, the payback is expected to be significantly more than a break-even proposition. The key is to justify security expenses and compare and offset these expenses against the computer security risks they're offsetting. Only by keeping the environment reasonably safe and preventing attacks can you justify a security expense budget.

Microsoft has several risk management guides that might prove useful during a security evaluation, including the Microsoft Security Risk Management Guide (<http://www.microsoft.com/technet/security/topics/policiesandprocedures/secrisk/default.aspx>) and the new Microsoft Security Risk Self-Assessment for Midsize Organizations Tool (<http://www.securityguidance.com>). The latter guide will lead you through a series of interview questions to determine how your organization might measure up against similar companies in the same industry.

## Summary

Hackers and malware can scan your networks and computers to look for common security vulnerabilities. If they can contact your computers over the Internet, these hackers and malware can exploit any possible vulnerabilities. Use IT security policy auditing, vulnerability assessment scanning, and penetration testing techniques to ensure the security of your network. Projects of this type must be sponsored and approved by company executives (e.g., CEO, CSO, CIO). Vulnerability assessment scanning and penetration testing are just the first steps in a security assessment project. Management needs to see the business value of doing such an exercise. Without their support, it will be hard to obtain the security assurance budget necessary to implement remedies to the vulnerabilities.

## Other Resources

*Assessing Network Security* by Kevin Lam, David LeBlanc, and Ben Smith (Microsoft Press, 2004, ISBN 0-7356-203304)

Microsoft Security Update bulletins at <http://www.microsoft.com/security/bulletins/default.msp>

NTBugtraq at <http://www.ntbugtraq.com>

The SANS Institute security newsletters at <http://www.sans.org/newsletters>

Security Focus newsletters and mail lists at <http://www.securityfocus.com/subscribe>

US Computer Emergency Readiness Team (CERT) at <http://www.us-cert.gov>

CERT Coordination Center at <http://www.cert.org>

Federal Computer Incidence Response (FedCIRC) at <http://www.us-cert.gov/federal/>

Department of Energy's (DOE's) Computer Incident Advisory Capabilities at <http://ciac.llnl.gov/ciac/index.html>

MITRE's Common Vulnerabilities and Exposures at <http://www.cve.mitre.org/cve>

Security Tracker at <http://www.securitytracker.com>

Distributed Intrusion Detection System at <http://www.dshield.org>

The SANS Institute Internet Storm Center at <http://isc.sans.org>

Message Labs at <http://www.messagelabs.com>

## About the Author

**Roger A. Grimes** ([roger@banneretcs.com](mailto:roger@banneretcs.com)) is a full-time writer and consultant on Windows security. He has written three books and more than 100 magazine articles on the subject and is a highly ranked presenter at national security conferences. His certifications include: CPA, CISSP, MCSE: Security (NT/2000/2003/MVP), CNE (3/4), TICSA, A+, and Network+. He is a contributing editor to *Windows IT Pro* magazine, and has developed several courses on Windows security and incident response.