PLANNING ASSUMPTION

June 28, 2001

Security Maintenance — Essential Components in Security Architectures

Michael Rasmussen

Giga Position

Whether a threat lies inside an organization or comes from a hacker on the Internet, security incidents are increasing at an alarming rate. Vulnerabilities are constantly discovered with automated exploit tools — that any would-be attacker can use. Operating system and application vendors fight vulnerabilities by releasing patches to fix the exposure in their software. But the story remains — successful penetration of systems utilizing patched vulnerabilities continues to rise after the patch is released.

The problem lies in the fact that organizations are not maintaining their security. Vendors release security updates, but organizations fail to apply them in a timely manner. Giga believes, in fact, that 90 percent of exploits related to security hardware/software vulnerabilities (as opposed to misconfigurations or poor security policies) could be eliminated if organizations would properly maintain the security of their systems. Companies that properly maintain the security of their systems will eliminate 90 percent of all potential exploits. Companies that fail to take these precautions should prepare for breaches at an increasing rate.

Proof/Notes

The Problem

Exploits to compromise systems abound, and they continue to grow at an alarming rate. One need only look at the Bugtraq database at **SecurityFocus** (www.securityfocus.com) to find an exploit that could compromise nearly any commercially deployed system.

While novices stumble on a small percentage of exploits, most take painstaking research by a small number of elite hackers. Once these exploits are released (often with an automated attack script or attack tool), the attack falls into the hands of the malicious masses. The Internet is then progressively scanned by numerous individuals looking for hosts susceptible to compromise. Once a vulnerable system is discovered — *game over* — your system is what hackers called "owned."

In the past, this game of search and destroy was done for the challenge. Then "hactivism" became a primary motive — breaking into sites to promote a cause. Today, in addition to the previous causes, there is increased activity centered on financial or personal gain. The FBI and other agencies continuously send warnings about individual and organized hacking rings exploiting known vulnerabilities to gain access to sensitive data (e.g., credit cards) to sell, use or to extort the company it was stolen from. We are already seeing the stages of the next evolution of hacking — cyber-warfare and cyber-terrorism (see Glossary) — a war where more than the military strength of a country is at risk; it is a war where a nations businesses and infrastructure are the targets.

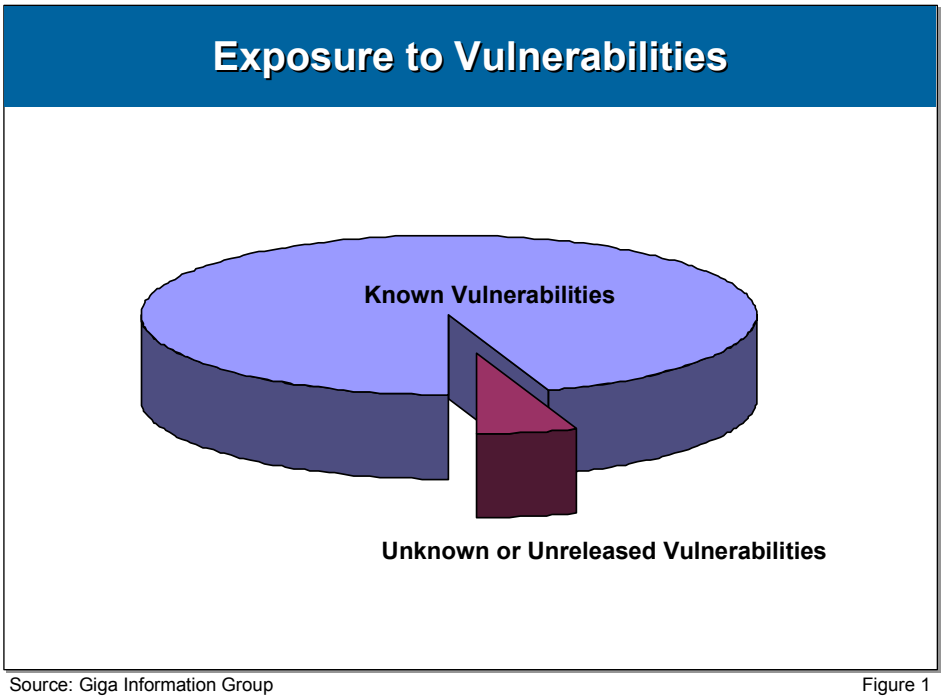
The threat from exploitation of vulnerabilities is not only external. A company needs to protect itself internally as well. Exploits are used internally by individuals within the organization for the very same purposes an external attacker would use them. The exposure here, however, is greater, since internal users know more about the internal workings, systems and processes of an organization, which they can use for greater gain, damage and to mask or disguise their activities. Vendors, while perpetually behind the curve of

patching security vulnerabilities, do their best to combat this by promptly releasing security updates for known vulnerabilities.

Organizations tend to rely only on their firewall(s) to combat external threats. As long as their servers sit behind the firewall, an organization feels safe. This is a false sense of security. Firewalls only offer a moderate level of protection to the servers that sit behind them. There are many different layers to attack. Firewalls typically operate at the network layer and, thus, they are primarily effective at combating network layer attacks. They are, with the exception of application proxy firewalls or air-gap devices, not effective at combating attacks at the application layer — the layer where we see the predominant focus in attention from attackers these days. This mandates that organizations secure their systems located behind the firewall.

If a firewall rulebase allows "Web" traffic to a server behind the firewall, then most firewalls will look at the incoming network traffic to verify that it claims to be Web traffic. This traffic could very well be an application layer attack aimed at compromising the system, or an attack masquerading as Web traffic. Hardening servers and maintaining the security combats this threat.

The majority of successful attacks against organizations are a result of the malicious masses using known vulnerabilities in which a patch has been released. A recent FBI announcement in March 2001 revealed that individuals and organized hacking rings were compromising systems for the purpose of extortion via vulnerabilities that were known and for which patches had been released — some of these exploits dated back to 1998 (see IdeaByte, FBI Security Advisory — A Lesson in Poor Maintenance, Michael Rasmussen).



Source: Giga Information Group Figure 1

As shown in Figure 1, Giga maintains that 90 percent or more of the successful system penetrations by software and hardware vulnerabilities (as opposed to poor passwords, misconfiguration and social engineering attacks) could be eliminated if organizations would commit to securing their systems on initial deployment and maintaining the security thereof on an ongoing basis.

The steps to combat your exposure to vulnerabilities is a two-fold process:

1. Harden systems and applications upon initial deployment.
2. Maintain the security of systems and their corresponding applications regularly.

Security Hardening

Before a system goes into production, it needs to be "hardened." Hardening is the process of securing the system through modifying its default configuration and applying security patches. Operating systems and applications by default are insecure. Manufacturers design their systems so any novice can use and deploy them — at the sake of security. This allows for systems that are easily compromised. The hardening process takes a system from its default state to a state where the security of the system can be trusted.

Many organizations neglect the process of hardening systems because of ignorance — they do not realize the exposure these systems present to their organization, as follows:

- The more unnecessary services left active, the greater risk an organization faces to security exposures.
- The weaker the security settings and policies on the system, the easier it will be for an attacker to penetrate the system.
- If security patches/updates are not applied to the system, the greater the exposure to compromise.

When hardening a system, it is best to adopt a holistic approach to security. Some companies approach hardening at just the operating system level and ignore the application and content on the system. An intensive hardening project can be overturned by an insecure application on the system.

The following layers need to be kept in mind when hardening a system — as you move up the layers, you often find less and less attention being paid to the security of that particular layer:

- **Content** — This is the most often ignored component in securing a system. An example is in HTML code. Many times HTML code, which is plainly viewable to anyone accessing a Web site, has hard-coded account names and passwords within the code itself. Content should be inspected to validate that it is secured appropriately.
- **Third-party applications** — These consist of extensions to applications — plug-ins, scripts. These applications, if left unsecured and untested, can lead to a compromise of the entire system.
- **Primary applications** — These refer to the main services of the system — Web, file transfer, mail. Often, little effort is given to secure these applications. It is at this level where we see much of the efforts in the hacking community to uncover vulnerabilities — it is also the layer that firewalls are the most ineffective at protecting against.
- **Operating system** — This is the foundation for securing the underlying processes required for a computer to operate. If the foundation is left insecure, it weakens the entire structure.
- **Network access** — This is the layer where most efforts at security are commonly focused. This is where firewalls operate. It is an important layer to secure, but it does not provide end-to-end security.

System hardening first needs to be defined in an organization's information security policies. Security policies should mandate that systems be secured and maintained to set standards and procedures. The policy is enforced by developing security standards and procedures documents:

- The security standards document outlines the operating system and application standards that have been set for an organization — and it should be strictly adhered to. It is difficult to secure and

maintain a wide array of competing operating systems and applications. By setting an organization standard, economies of scale can be developed that minimize the resources and effort needed to secure these systems.

- The security procedures document provides a step-by-step guide to securing the operating systems and applications outlined in the security standards document. This document is the one that is most frequently updated, since it has to stay on top of new developments in exposures to the standardized systems.

System hardening is a process that involves both business and technical aspects and is done through the stages of the four D's:

1. **Define** — The concept phase where a systems purpose is determined, it primarily involves the business needs the system is to address.
2. **Design** — The phase where the system is specked and a technical design is developed to address the business need.
3. **Develop** — The process of developing the functional system to the design specifications.
4. **Deploy** — The point at which the developed and tested system is moved into production.

The first step in hardening any system is not a technical step — it is a business step. Before any system is designed, developed and deployed, a risk analysis of the system must be performed. A risk analysis identifies the value the system will provide to the business, the threats to it and the exposure the system faces to those threats. This allows an organization to assign cost-effective counter-measures in the system to combat threats through an understanding of what level of security is appropriate for the system.

During design and development of the system, developers need to understand the objectives and security needs to allow for the appropriately identified security controls to be integrated into the system. Developers should think comprehensively about system security during the design and development process.

During deployment, the process of hardening a system varies from operating system to operating system and application to application, but some common steps found throughout different systems are the following:

- **Turn off unnecessary services** — Default installations leave many services turned on that are otherwise unnecessary. Turning off unnecessary services and deleting the executables from the system provide a more secure system. In addition, the system will perform better, since the CPU does not spend time monitoring and running services that are not being used.
- **Patch the system** — The system should have all service packs/patches/hot fixes applied to it, primarily those that pertain to the security of the system. Once applied, all hardening procedures should be validated that they have not been changed, because some service packs roll back configuration settings.
- **Configure file system, directory and registry settings** — The appropriate rights to the file system, directory service and the registry should be reviewed and enforced. Many applications provide global read/write access to key directories, which could lead to a security exposure — in most cases, this level of permission is unnecessary.
- **Configure the system security policies** — The system security policies should be reviewed to make sure they accurately reflect the organization's information security policy. Examples are the systems password, account and audit policies.
- **Configure and tune logging** — Logging is abysmal in default operating system and application installations. This can be enhanced by configuring the system to log more detail and security relevant information. The logs, and monitoring thereof, are often one of the best indicators of

attempted and successful security breaches.

- **Use and enable access controls** — The proper access controls should not only be set on the file/directory structure — but also on the network interface. This allows for firewall functionality on the server itself. If the server is a Web server, then only Web traffic should be allowed in on the network interface.
- **One service to a system** — The best practice in securing a system is to only allow one service/function to a server. It is more difficult to secure a multipurpose system than it is to secure a system with a single dedicated purpose.
- **Physical security** — The strong technical controls and hardening can mean nothing if the system itself is not physically secure from unauthorized access.
- **Install the operating system fresh** — A system is best secured when security is done from the beginning when the system is first installed. Hardening a server that has been in production can be very difficult, and there is a chance that the server being hardened might have previously been compromised — efforts at hardening then could be circumvented.
- **Disk storage** — The most secure file system structure should be selected (e.g., Windows NT/2000 offers NTFS, which offers more security than the FAT file system).
- **Pick a strong administrator password** — The administrator password holds the keys to the kingdom. It is important that this password be the strongest password on the network and closely guarded. The administrator account should only be used in emergencies, with system administrators using their own administrator-equivalent accounts to provide accountability for their actions.
- **Use a minimal install** — Do not install components on the system that are not needed in the ongoing operation of the system. Additional components can introduce more points of vulnerability into the system.
- **Install only necessary network protocols** — If the server only needs TCP/IP for communication, this is the only protocol that should be installed. Additional protocols allow for more vulnerabilities and cause higher performance on the system and network than is necessary.
- **Encrypt system databases** — System databases, such as password databases, should be encrypted to protect them.
- **Enforce minimal rights** — Applications and users should only be provided with the minimal rights required to do the tasks they need to accomplish.
- **Install host-based intrusion detection** — Use host intrusion-detection features and products to monitor and identify security incidents. Host-based IDS technologies include monitoring of the following five key areas:

    1. Log/event — Defines a process that watches system and application logs for significant security events.
    2. File integrity — A process that watches key system and application files for unauthorized changes made to them.
    3. Network traffic monitor — Monitors and controls network traffic coming into the protected hosts looking for traffic that violates security policies or represents a security incident.
    4. System monitor — Monitors the system for overall performance and stability. Watches for rogue unauthorized processes that an attacker might attempt to run.
    5. Policy compliance — Verifies system configuration to validate that it conforms to defined organizational policies and checks to make sure that changes that violate policy have not been performed on the system.

- **Verify *all* security settings** — After configuring/hardening the security on the host, validate all settings to make sure they are intact. It is a known fact that, in many operating systems, applying patches and making changes to settings can often undo other changes that were previously made.

The process of hardening a system can be difficult. Defining and adhering to corporate-wide security standards eases this process as expertise is developed and maintained on a focused set of applications. Still, the process of hardening a system often involves breaking the system/application by tightening security controls to the extreme — and then fixing it by turning on a piece at a time to allow the application to function.

Hardening a system can also protect a system from unknown vulnerabilities. An example is the recent **Microsoft** ISAPI printer vulnerability in Internet Information Server (IIS) 5 discovered by eEye (see IdeaByte, Serious Vulnerability in Microsoft IIS 5.0 Calls for Fast Action to Patch, Michael Rasmussen). If organizations had followed Microsoft's guidelines for securing IIS 5.0, this feature would have been disabled if it was not being used in the application. On the other hand, Microsoft does not disable this feature by default — a point that many in the security community find bothersome. Is it better to provide a product with everything turned on, than to turn off what is not needed? Or is it better to provide a product with everything turned off and only turn on what is needed? The security point of view sides with the latter.

Security is often not a primary concern to operating system and application vendors. This makes it very difficult for organizations to protect their data. To assist in the process of securing systems, security vendors have developed products that assist in securing operating systems and applications. These products can be broken into the following three categories:

1. **Operating system security** — Products such as **Argus**' PitBull provide a more trusted operating system architecture that integrates with commercial operating systems.
2. **Application security** — Software like **Sanctum**'s AppShield provides security against known and unknown vulnerabilities at the application layer.
3. **Host-based intrusion detection** — Vendors like **PentaSafe**, with its Vigilent Security Manager, and **Symantec** with Intruder Alert and Enterprise Security Manager, offer a suite of products to set security policies and monitor the operating system and applications for security events.

These products, and others like them, can be used to assist in securing and maintaining the security of the systems deployed in an organization. The interesting thing about many of them is the simple reason why these products exist: because the operating system and application vendors/developers fail to integrate security controls and checks into their systems.

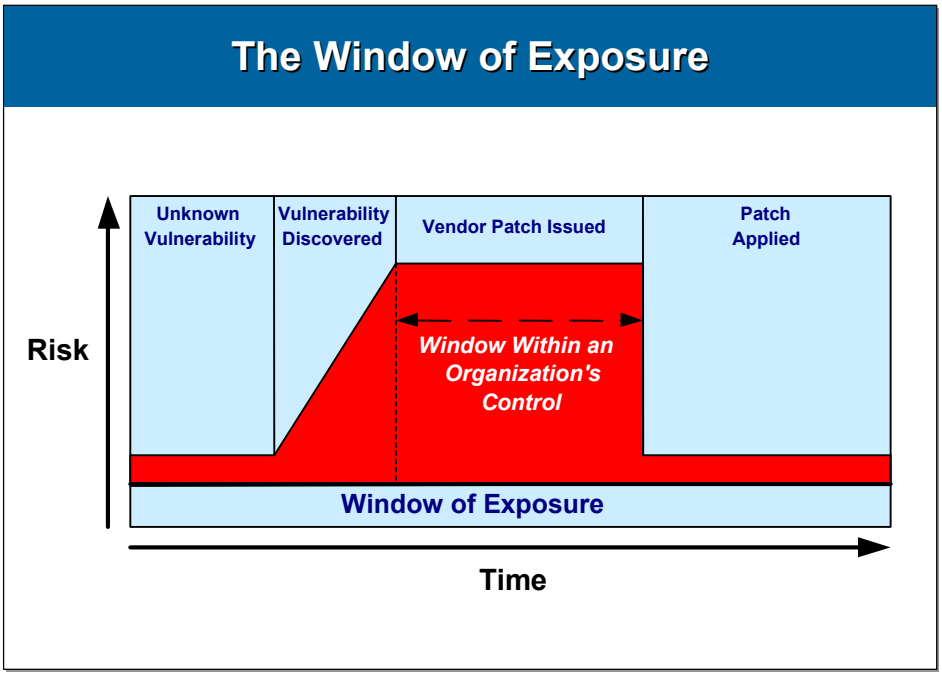
Security Maintenance

Security maintenance and security hardening go hand and hand — you cannot have one without the other. It makes no sense to maintain the security of a system that was never secured to begin with. Likewise, it makes no sense to harden a system if you do not plan on maintaining the security of it.

Many efforts in security hardening go wasted, since a year after deployment, the system is exposed to a number of vulnerabilities because security was never maintained on the system. A predominant number of systems that are compromised are done so through known, patched vulnerabilities.

The world needs a call to arms in security maintenance. Through security maintenance, an organization can stay on top of, and proactively manage, the security of its systems. It comes down to a basic concept called the "window of exposure." The window of exposure is the period of time that an organization finds itself susceptible to a vulnerability in a system (see Figure 2). A portion of this window is out of the organization's control — that is the time from when the vulnerability is discovered (whether publicly known or not), to the

point where a patch is supplied by the vendor. What is in the organization's control is the time from when the patch is available to the time when the patch is applied.



Source: Giga Information Group Figure 2

In an organization that takes security maintenance seriously, the window of exposure is reduced by promptly applying patches to fix security holes shortly after they are released. This window is completely under the organization's control. This is the exact area where most organizations are failing. In the recent FBI announcement, system penetrations are occurring as a result of vulnerabilities that were patched three years ago.

Only after a system is hardened can a successful security maintenance program be initiated. The following are some suggested guidelines in defining a security maintenance program for the enterprise:

- **Regularly apply patches —** Security patches *must* be applied regularly. The frequency of application will vary based on the exposure of the system involved and the risk it brings to the business. Patches for vulnerabilities that pose a significant risk to the enterprise should be applied as soon as possible. Before being deployed in production, patches should be tested for stability and reliability. One of security's objectives is availability — it is counter-productive to apply a patch to protect a system, just to have it go down because of a poorly developed patch.
- **Review security configuration —** Review the security configuration on systems regularly to validate that changes have not been made through maintenance that will weaken system security (e.g., quarterly). Last September, **Western Union** experienced a compromise in which nearly 16,000 credit card numbers were stolen. The attackers were successful because of a security configuration change that was done through routine maintenance and left open.
- **Vulnerability scanning** — Systems should be scanned for vulnerabilities on a regular basis (e.g., quarterly). Popular tools are **NAI**'s CyberCop Scanner and **ISS** System Scanner, or for those that prefer a free, but comparable product, there is the **Nessus** scanner (www.nessus.org) (see IdeaByte, Vulnerability Scanners — What Vendors Don't Want You to Know, Michael Rasmussen).

- **Security audit** — Conduct a third-party security audit of an organization's systems regularly (e.g., annually). When an audit is conducted, the following components should be addressed:
  - Application security — A secure operating system can often be overcome by an insecure application.
  - Third-party application security — Plug-ins, scripts and third-party components installed on top of applications should be reviewed for security vulnerabilities.
  - Content — Review the content of the system; often, it may contain items that need to be addressed, such as passwords stored in HTML.
  - Operating system security — The security controls and configuration of the operating system should be reviewed.
  - Network security — The security controls, counter-measures and configuration of the network should be audited on a regular basis.
- **Monitoring** — Security monitoring involves ongoing review and audit of system and application logs (e.g., minimum of weekly).
- **New vulnerabilities** — Bug/vulnerability mailing lists should be continually monitored for new vulnerabilities that might affect an organization's systems. Some vendors are offering this as an automated service in which they profile an environment and send alerts that are only of significance to the profiled environment — thus reducing the amount of time and cost in monitoring these lists. Some vendors offering this service are SecurityFocus.com's Security Intelligence Alert Service, **Vigilinx** Security Intelligence Service and **eSecurityOnline**'s Online Vulnerability Service.
- **News/trends analysis** — Attacks can often be thwarted by paying attention to what security incidents other companies are experiencing. Security incidents often come in waves as attack mechanisms and tools grow in popularity. The ability to forecast and predict incidents is a nirvana that many would like to achieve. The US federal government has created the National Infrastructure Protection Center (NIPC) to provide a level of forecasting of security incidents. The NIPC has met this with limited success. Other organizations, such as SANS Global Incident Analysis Center and SecurityFocus.com's new Attack Registry and Intelligence Service, are taking this to a new level.

The biggest struggle in security maintenance is often defining who is responsible for security maintenance. Organizations often have stringent dividing lines between internal departments, which can be a political nightmare to overcome. Systems administrators often do not want anything to do with security, while the security folks are at their mercy to see that the systems are secure. If an organization wants to take the security process seriously, it is mandatory that security be distributed throughout the roles in the organization.

Best practices in security maintenance roles are as follows:

- It is the role of the information security department to communicate and validate that systems are being maintained.
- It is the role of the systems administrator to test and apply patches and maintain the security of the system.

If the security department is given the role to maintain security, in addition to validating and communicating security, then it is a conflict of interest, since the auditor/validator would be the maintainer. Security staffs are often faced with limited personnel. It would be an impossible task for many security departments to take on the responsibility of maintaining system security throughout the enterprise. The task of maintenance needs to be distributed throughout the system/application administrators to see that security is accomplished. The downside to this is that, in many vendor certification programs, they do not adequately teach the process of securing their product. It is part of the security department's job to communicate and train the systems administrators to secure systems properly according to the organization's security policies, standards and procedures.

Alternative View

Some organizations are willing to gamble — they feel that they can play the odds, assuming that they are not targets. The fact is that *everyone* is a target. This has been seen recently with the rise of an unsanctioned hacking war between the US and China — it didn't matter who you were, if you were on the other side, you were a target.

Ignoring a standard policy of server hardening and maintenance is acceptable if a company engages in some degree of business risk management. After all, some systems face less risk to a corporation than others. It gets down to risk management once again. If it is going to cost an organization more money to maintain a level of security than the asset is worth or produces, then a lesser level of security may often be justified. A word of warning though: There are many facets to risk. Leaving a server vulnerable to attack because it does not seem cost effective to maintain the security of it may introduce legal liabilities if the system is compromised and access to confidential data is gained through it or if it is used to attack someone else.

Findings & Recommendations

Vulnerability awareness is like a disease — popularity of exploits quickly spreads as hackers use known vulnerabilities to compromise systems. Once an exploit has been published, hackers will use it to scan and compromise a multitude of systems, soon spreading throughout the world. This is evidenced with the popularity of the BIND vulnerabilities in domain name system (DNS) servers earlier this year. Right after discovery, the world saw an unparalleled increase in scans for, and compromises of, vulnerable DNS servers.

Just like a disease, the best way to fight this threat is through healthy security maintenance of systems. This starts with system hardening. When a system is first defined, perform a risk analysis to determine the appropriate security controls to integrate into the system. This is then fulfilled through the design, development and deployment stages of the system.

Organizational policies, standards and procedures for system hardening and maintenance need to be defined and maintained. It is in these documents that an organization can define, implement and manage an ongoing security maintenance program. Security standards set guidelines outlining what systems/products should be deployed, and procedures outline the proper steps in hardening and maintaining the security of those systems.

If an organization is going to make the effort to secure its systems upon design and deployment, then it must commit to ongoing security maintenance. It makes no sense to secure a system that will not be maintained — it will be vulnerable to the next exploit that comes out. Likewise, it makes no sense to maintain the security of a system when the proper security foundation of hardening was not done: The two go hand in hand.

References

Related Giga Research

IdeaBytes

Where Firewalls Fail, Michael Rasmussen

Deploy an Application/System Securely, Michael Rasmussen

FBI Security Advisory — A Lesson in Poor Maintenance, Michael Rasmussen

Serious Vulnerability in Microsoft IIS 5.0 Calls for Fast Action to Patch, Michael Rasmussen

Vulnerability Scanners — What Vendors Don't Want You to Know, Michael Rasmussen

Relevant Links and Other Sources

SecurityFocus, www.securityfocus.com

SecurityPortal, an **AtomicTangerine** site, www.securityportal.com

Microsoft Security, www.microsoft.com/security/default.asp

Glossary

Cyber-warfare — the attack and defense of systems as it relates to conflicts between government bodies.

Cyber-terrorism — the attack of systems with motives based in terrorism.

Exploit — utilizing a vulnerability to gain access to a system.

Hactivism — attacking a system to further a cause: the Internet form of activism.

Hardening — the process of securing a system upon initial deployment.

Intrusion detection — the process and technology used to identify security incidents on a network or system.

Vulnerability — the exposure of a system to a specific method of compromise.

Window of exposure — the time that a system is susceptible to a vulnerability — between the time it is discovered and the time that a patch is applied.