



**National Institute of
Standards and Technology**

Technology Administration
U.S. Department of Commerce

NIST Special Publication 800-40

Procedures for Handling Security Patches

Recommendations of the National Institute of Standards and Technology

Peter Mell and Miles C. Tracy

NIST Special Publication 800-40

Procedures for Handling Security Patches

*Recommendations of the National
Institute of Standards and Technology*

Peter Mell and Miles C. Tracy

C O M P U T E R S E C U R I T Y

Computer Security Division
Information Technology Laboratory
National Institute of Standards and Technology
Gaithersburg, MD 20899-8930

April 2002



U.S. Department of Commerce
Donald L. Evans, Secretary

Technology Administration
Phillip J. Bond, Under Secretary for Technology

National Institute of Standards and Technology
Arden L. Bement, Jr., Director

Reports on Computer Systems Technology

The Information Technology Laboratory (ITL) at the National Institute of Standards and Technology (NIST) promotes the U.S. economy and public welfare by providing technical leadership for the Nation's measurement and standards infrastructure. ITL develops tests, test methods, reference data, proof of concept implementations, and technical analysis to advance the development and productive use of information technology. ITL's responsibilities include the development of technical, physical, administrative, and management standards and guidelines for the cost-effective security and privacy of sensitive unclassified information in Federal computer systems. This Special Publication 800-series reports on ITL's research, guidance, and outreach efforts in computer security and its collaborative activities with industry, government, and academic organizations.

National Institute of Standards and Technology Special Publication 800-40
Natl. Inst. Stand. Technol. Spec. Publ. 800-40, xx pages (Mon. 2002)

CODEN: XXXXX

Certain commercial entities, equipment, or materials may be identified in this document in order to describe an experimental procedure or concept adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 2002

For sale by the Superintendent of Documents, U.S. Government Printing Office
Internet: bookstore.gpo.gov — Phone: (202) 512-1800 — Fax: (202) 512-2250
Mail: Stop SSOP, Washington, DC 20402-0001

Executive Summary

Timely patching is critical to maintain the operational availability, confidentiality, and integrity of IT systems. However, failure to keep operating system and application software patched is the most common mistake made by information technology (IT) professionals. New patches are released daily, and it is often difficult for even experienced system administrators to keep abreast of all the new patches. Further complicating the matter is that monitoring for new patches is not sufficient since one must monitor also for vulnerabilities.

Vulnerabilities are weaknesses in software that can be exploited by a malicious entity to gain greater access and/or permission than it is authorized to have on a computer. Not all vulnerabilities have related patches; thus, system administrators must not only be aware of vulnerabilities and patches, but also mitigate “unpatched” vulnerabilities through other methods (e.g., workarounds, firewalls, and router access control lists).

To help address this growing problem, we recommend that organizations have an explicit and documented patching and vulnerability policy and a systematic, accountable, and documented process for handling patches. We recommend creating a patch and vulnerability group (PVG). This group will facilitate the identification and distribution of patches within the organization. Its duties will include:

1. Creating an organizational software inventory
2. Identifying newly discovered vulnerabilities and security patches
3. Prioritizing patch application
4. Creating an organization-specific patch database
5. Testing patches
6. Distributing patch and vulnerability information to local administrators
7. Verifying patch installation through network and host vulnerability scanning
8. Training system administrators in the use of vulnerability databases
9. Deploying patches automatically (when applicable)

The existence of the PVG does not diminish the responsibility of each systems administrator to patch the systems under their control. Each systems administrator should:

1. Apply patches identified by the PVG
2. Test patches on the specific target systems
3. Identify patches and vulnerabilities associated with software not monitored by the PVG

Besides creating a PVG, organizations should be aware that applying patches and mitigating vulnerabilities is not always a straightforward process. There are a variety of issues covered in this document that are associated with prioritizing patches, obtaining patches, testing patches, and applying patches.

Acknowledgements

The authors, Peter Mell of NIST and Miles Tracy of Booz Allen Hamilton (BAH) wish to express their thanks to Timothy Grance, John Wack, and Murugiah Souppaya of NIST and Alexis Feringa, Jennifer Gray, Jonathan Holleran, and Mark McLarnon of BAH for their research, technical support, and written contributions to this document.

Table of Contents

1.	INTRODUCTION.....	1
1.1	AUTHORITY.....	2
1.2	PURPOSE AND SCOPE.....	2
1.3	OBJECTIVE.....	3
1.4	AUDIENCE AND ASSUMPTIONS.....	3
1.5	DOCUMENT STRUCTURE.....	3
2.	CREATING AND IMPLEMENTING A PATCHING PROCESS.....	5
2.1	THE PATCH AND VULNERABILITY GROUP.....	5
2.2	SYSTEMS ADMINISTRATOR PATCHING RESPONSIBILITIES.....	7
3.	IDENTIFYING VULNERABILITIES AND APPLICABLE PATCHES.....	10
3.1	VENDOR WEBSITES AND MAILING LISTS.....	11
3.2	THIRD-PARTY WEBSITES.....	12
3.3	THIRD-PARTY MAILING LISTS AND NEWSGROUPS.....	12
3.4	VULNERABILITY SCANNERS.....	13
3.5	VULNERABILITY DATABASES.....	15
3.6	OTHER NOTIFICATION TOOLS.....	17
3.7	WINDOWS UPDATE.....	17
4.	GOVERNMENT VULNERABILITY IDENTIFICATION RESOURCES.....	19
4.1	CVE VULNERABILITY LIST.....	19
4.2	NIST ICAT VULNERABILITY INDEX.....	20
4.3	NATIONAL INFRASTRUCTURE PROTECTION CENTER.....	21
4.4	CERT/CC.....	23
4.5	FEDERAL COMPUTER INCIDENT RESPONSE CENTER (FEDCIRC).....	24
5.	PATCHING PROCEDURES.....	26
5.1	PATCHING PRIORITIES.....	26
5.2	OBTAINING PATCHES.....	27
5.3	PATCHING PRECAUTIONS.....	27
5.4	TESTING PATCHES.....	30
5.5	APPLYING PATCHES.....	31
5.6	UPDATING LINUX/UNIX OPERATING SYSTEMS AND APPLICATIONS.....	32
5.7	UPDATING NETWORK INFRASTRUCTURE COMPONENTS.....	32
5.8	UPDATING WINDOWS OPERATING SYSTEMS AND APPLICATIONS.....	32
5.9	AUTOMATED PATCH DISTRIBUTION AND APPLICATION TOOLS.....	34
5.10	REDUCING THE NEED TO PATCH THROUGH SMART PURCHASING.....	34
5.11	TRAINING USERS TO PATCH.....	35
5.12	PATCHING AFTER A SECURITY COMPROMISE.....	36
6.	CONCLUSION.....	38
	APPENDIX A: GLOSSARY.....	1

APPENDIX B: PATCHING RESOURCES.....	1
APPLE.....	1
CISCO.....	1
MICROSOFT WINDOWS OPERATING SYSTEM.....	2
POPULAR WEB CLIENT AND MAIL CLIENT APPLICATIONS.....	3
POPULAR END-USER APPLICATIONS.....	3
POPULAR SERVER APPLICATIONS.....	4
LINUX/UNIX DISTRIBUTION WEBSITES.....	5
POPULAR LINUX/UNIX DISTRIBUTION DOWNLOAD/UPDATE/SECURITY WEBSITES.....	8
VIRUS SOFTWARE DOWNLOAD/UPDATE/SECURITY CENTERS.....	10
APPENDIX C: IDENTIFYING VULNERABILITIES WITH ICAT.....	1
ACCESSING THE ICAT METABASE.....	1
NAVIGATING THE ICAT METABASE WEBSITE.....	2
ICAT METABASE SIDEBAR.....	3
COMMON VULNERABILITIES AND EXPOSURES.....	3
SEARCHING THE ICAT METABASE.....	4
DEFINING SEARCH PARAMETERS.....	5
DEFINING SEARCH FILTERS.....	7
ICAT METABASE SEARCH RESULT.....	9
APPENDIX D: VULNERABILITY ADVISORY RESOURCES.....	1
FEDERAL VULNERABILITY ADVISORY WEBSITES.....	1
PRIVATE SECTOR VULNERABILITY ADVISORY WEBSITES.....	1
APPENDIX E: WINDOWS UPDATE.....	1
APPENDIX F: MICROSOFT PERSONAL SECURITY ADVISOR.....	1
APPENDIX G: MICROSOFT NETWORK SECURITY HOTFIX CHECKER.....	1
APPENDIX H: MICROSOFT QFECHECK HOTFIX CHECKER.....	1

1. Introduction

Failure to keep operating system and application software up to date is the most common mistake made by information technology (IT) professionals. Despite extensive testing, all operating systems and applications are released with “bugs” (errors in the software) that affect security, performance, and stability. Most estimates for the number of bugs in published software range from 5 to 20 bugs per 1,000 lines of code¹.

As software programs expand, the potential number of bugs grows. Windows 3.1, released in 1992, had an estimated 3 million lines of code.² Thus, according to common opinions, it would contain an estimated 15,000 to 60,000 potential bugs. In 1999, Windows 2000 was released. With a low estimate of 35 million lines of code, there would be 175,000 to 700,000 potential bugs within Windows 2000. In 1992, the first graphical user interface (GUI) version of Linux was released with about 170,000 lines of code, equating to an estimated 850 to 3,400 potential bugs in the software. With the release of a distribution of Linux, RedHat 7.1 in 2001, the number of lines of code has grown to about 30 million. This equates to an estimated 150,000 to 600,000 potential bugs. Most of these bugs do not create vulnerabilities in our systems. It does, however, reflect the complexity and difficulty of delivering highly trustworthy code on systems developed with that level of scale and complexity. Nonetheless, it is critical that we promptly fix those vulnerabilities that weaken our defenses.

Security-related bugs are generally discovered only after a large number of users start using the software and hackers start attempting to compromise it. Once a bug is discovered, the software manufacturer often releases a piece of software to correct the bug. This software is often called a patch, hotfix, or service pack.

Today more than ever, timely patching is critical to maintain the operational availability, confidentiality, and integrity of IT systems. New patches are released daily, and it is often difficult for even experienced system administrators to keep abreast of all the new patches.

Patches are usually released for three reasons:

- To fix faults in an application or operating system. Many hacker attacks are based on exploiting faults in the computer code of applications and operating systems. Patches are also released to correct performance or functionality problems.
- To alter functionality or to address a new security threat. An example of this is new virus definitions for an antivirus application. There was nothing “wrong” with the code of the antivirus program, but it had to be updated to detect new viruses that did not exist when the application was first released.
- To change or modify the software configuration to make it less susceptible to attacks and more secure.

¹ For more information see <http://panko.cba.hawaii.edu/HumanErr/>.

² Throughout this document, examples of bugs and software applications will be employed for illustrative purposes only. This does not imply that a product is defective or that it should not be used.

The outbreaks of the Code Red and Nimda worms demonstrate why patching applications and operating systems are critical. During June 2001, a network security company discovered a serious vulnerability in the Microsoft IIS web server application. Within days, Microsoft released a patch to eliminate the vulnerability, but many system administrators did not update their systems. In July, the Code Red worm, exploiting the vulnerability discovered in June, infected more than 300,000 computers in one week, even though the patch had been available for several weeks. Unfortunately, system administrators did not learn from the experience. Within two months, the even more virulent Nimda (admin spelled backwards) virus, which exploited multiple vulnerabilities for which patches existed, infected a large number of additional computers. Neither virus would have had much effect if those systems had been patched in a timely manner.

CERT/Coordination Center (CC)³ (<http://www.cert.org>) estimates that 95 percent of all network intrusions could be avoided by keeping systems up to date with appropriate patches. In an increasingly interconnected world, it is critical that system administrators keep their systems patched to the most secure level. A common misperception among some system administrators is that a secure firewall reduces the need for timely patching. Unfortunately, this is incorrect because a firewall generally permits some level of traffic between most internal and external hosts. As long as a communication is allowed between the internal network and the Internet or other external network, there is a risk of compromise; thus patching becomes critical.

1.1 Authority

This document has been developed by the National Institute of Standards (NIST) in furtherance of its statutory responsibilities under the Computer Security Act of 1987 and the Information Technology Management Reform Act of 1996 (specifically, 15 United States Code [U.S.C.] 278 g-3 (a)(5)). This is not a guideline within the meaning of 15 U.S.C. 278 g-3 (a)(3).

These guidelines are for use by federal organizations that process sensitive information. They are consistent with the requirements of Office of Management and Budget (OMB) Circular A-130, Appendix III.

This document may be used voluntarily by non-governmental organizations. The document is not subject to copyright.

Nothing in this document should be taken to contradict standards and guidelines made mandatory and binding upon federal agencies by the Secretary of Commerce under his or her statutory authority. Nor should these guidelines be interpreted as altering or superseding the existing authorities of the Secretary of Commerce, Director, OMB, or any other federal official.

1.2 Purpose and Scope

This document presents a systematic approach for identifying and installing necessary security patches. Following this systematic approach will reduce the number of incidents in an organization. However, it does not address specific patches or

³ Note: CERT is no longer an acronym. Previously, it stood for Computer Emergency Response Team.

vulnerabilities (except as examples) or how vulnerabilities might be mitigated beyond installing the appropriate patch.

1.3 Objective

The objective of this document is to provide guidance and advice on patching IT systems so as to better secure them from attack.

1.4 Audience and Assumptions

This document is written for system administrators, technical managers, functional managers, and other IT staff members who manage information systems. It provides a structured approach to identifying and implementing security patches. Management personnel who are responsible for systems can use the topics discussed in this document to become familiar with the status of the assets under their stewardship. This document can also assist personnel in evaluating their compliance with their organization's security standards and requirements. Finally, management personnel can use this guide to provide a technical basis that supports their decision-making processes.

This document assumes that readers will have some minimal operating system and application expertise. Because of the volatile nature of vulnerabilities and patches, this document is generic; and readers are expected to take advantage of other resources (including those listed in this document) for specific vulnerability and patch information.

1.5 Document Structure

The document is divided into six sections followed by seven appendices. The remainder of this document is structured as follows:

- Section 2 describes how to create and implement a patching policy, process, and system.
- Section 3 presents an overview of the various methods of identifying vulnerabilities and applicable patches.
- Section 4 gives an overview of specific government patch and vulnerability resources.
- Section 5 specifies patching procedures.
- Section 6 concludes by summarizing our patching recommendations.
- Appendix A gives a glossary of terms used throughout this document.
- Appendix B specifies patching resources for a variety of platforms and applications.
- Appendix C provides guidance on using the ICAT website to identify vulnerabilities and applicable patches.
- Appendix D identifies some commonly used vulnerability advisory resources.

- Appendix E details instructions for using the Windows Update feature included with most newer versions of Microsoft's Windows Operating System.
- Appendix F presents detailed instructions on using Microsoft's Personal Security Advisor.
- Appendix G gives detailed instructions on downloading and using Microsoft's Network Security Hotfix Checker.
- Appendix H provides detailed instructions on downloading and using Microsoft's Qfecheck hotfix checker.

2. Creating and Implementing a Patching Process

Given the number of patches and the complexity inherent in any but the most basic network, most organizations need to create a systematic and accountable process for identifying and applying patches. This section presents an approach for handling patches and vulnerabilities within medium to large organizations. This approach advocates creating a centralized group in charge of patches and vulnerabilities that supports the patching efforts of local administrators. Alternate approaches exist for implementing a patching process. The essential point is that organizations should create or adopt a systematic, comprehensive, documented, and accountable patching process. This model is provided as a starting point for organizations developing an organized patching process.

2.1 The Patch and Vulnerability Group

We recommend creating a "Patch and Vulnerability Group" (PVG). The size of the PVG will vary depending on the size and complexity of the organization. The PVG may consist of full-or part-time personnel. The personnel involved should have broad knowledge of patches, systems administration, and computer vulnerabilities. In addition, it is helpful to have specialists in particular operating systems, applications, and servers. Personnel who already perform vulnerabilities scanning or who operate intrusion detection systems are likely candidates for this group.

The duties of the PVG will be to support local administrators in finding and fixing vulnerabilities in the organization's software. The PVG will generally not patch vulnerabilities themselves; rather they will work with a local administrator to apply and test patches. The duties of the PVG are to:

1. **Create An Organizational Software Inventory.** The PVG should create a database containing the software packages and version numbers of those packages most used within the organization. This inventory will enable the PVG to monitor for information about vulnerabilities and patches that correspond to the software within the inventory. Specific attention should be given to those software packages that are used on important servers or that are used by a large number of systems. Note that some organizations may attempt to create a detailed inventory of the software on every system. For most organizations, however, this will probably not be cost effective (or even possible). The key point is that the inventory is reasonably representative of the systems in the organization. The PVG should make the inventory available to systems administrators so that the administrators know which software packages the PVG will be checking for new patches and vulnerabilities.
2. **Identify Newly Discovered Vulnerabilities and Security Patches.** The PVG is responsible for monitoring security sources for vulnerabilities and patches that correspond to the software within the PVG's organizational software inventory. A variety of sources should be monitored to ensure that they are aware of all the newly discovered vulnerabilities. Sections 3 and 4 discuss where and how to monitor for patches and vulnerabilities.
3. **Prioritize Patch Application.** The PVG should be aware of the resource constraints of local administrators and should attempt to avoid overwhelming them

(when possible) with a large number of patches. The PVG must prioritize the set of known patches and provide advice to local administrators on the criticality of each patch. A distinction must be made between servers and end-user systems when making patching recommendations because often it is more important to patch servers before end-user systems and to more thoroughly patch the servers. Information on prioritizing patches is contained in Section 5.1.

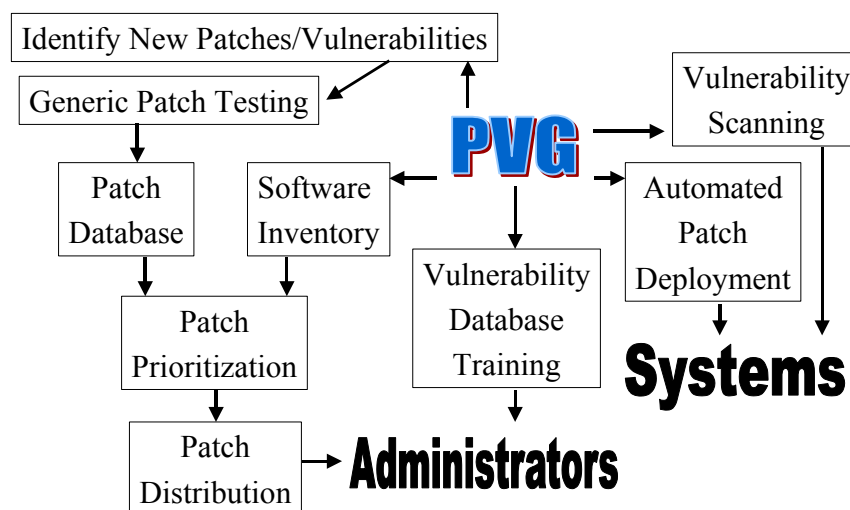
4. **Create an Organization-Specific Patch Database.** The PVG should create a database of information on the patches that apply to the organization. Ideally, the database should contain the actual patches and instructions on installing those patches. The reason that it is recommended that an organization maintain a copy of each patch is for situations when the Internet may not be accessible or the vendor's website may have been compromised. In addition, it is probably easier for local administrators to apply a patch using the PVG database as opposed to a vendor site that might overwhelm administrators with a large array of available patches. While the creation of a patch database is recommended, resource constraints may limit an organization to listing only websites that contain each patch. Such a solution should be workable when each hyperlink to a patch is associated with textual advice from the PVG.
5. **Conduct Generic Testing of Patches.** If an organization uses standardized host configurations, the PVG will be able to test patches on those configurations. This will avoid the need for redundant testing by each local administrator. The PVG should also work closely with local administrators to test patches on important servers systems. Information on testing patches is contained in Section 5.4.
6. **Distribute Patch and Vulnerability Information to Local Administrators.** The PVG is responsible for informing local administrators about patches that correspond to software packages included on the organizational software inventory. E-mail lists should provide an effective method for distributing patch information. However, to decrease the chance of a spoofed e-mail containing a Trojan horse patch, actual patches should be distributed from an internal secured website instead of within the e-mails themselves. Several e-mail lists may be maintained that include administrators that are responsible for various types of systems (e.g., Unix versus Windows administrators).
7. **Verify Patch Installation Through Network and Host Vulnerability Scanning.** The PVG will probably not have the resources to verify that every patch has been installed on every machine. However, the PVG should perform periodic network and host vulnerability scanning to identify systems that have not been patched. In addition, such scanning will provide the PVG with another data source for new vulnerabilities and patches. However, the PVG should be aware that network and host vulnerability scanners do not check for every known vulnerability and thus cannot be relied on as a sole source of vulnerability information. The PVG should inform local administrators that they are performing such periodic scanning because it will make the administrators more accountable to install each patch. NIST Special Publication 800-42, *Guidelines on Network Security Testing*, offers advice on techniques for vulnerability scanning.
8. **Train System Administrators in the Use of Vulnerability Databases.** Although the PVG will monitor for new patches and vulnerabilities found within the software

listed in the organizational software inventory, local administrators may use software not listed in the inventory. This situation may exist as a result of a management decision that the PVG only has resources to focus on the more popular software packages. In this situation, it is essential that local administrators have some knowledge of how to identify new patches and vulnerabilities. By providing them with such knowledge, we create a second line of defense in our patching process. Local administrators should be trained by the PVG on the various vulnerability and patching resources described in Sections 3 and 4. Organizations may choose to train their administrators with only a few tools that are known to be comprehensive.

9. **Perform Automatic Deployment of Patches (When Applicable).** Some organizations with largely homogeneous computing platforms can use automated distributed patch deployment services. Thus, an administrator from a single console can update hundreds or even thousands of systems. This job function could be effectively performed by the PVG. If not, the PVG should work very closely with the administrator in charge of the patch deployment system to ensure that all applicable patches are applied.

These nine PVG duties are interrelated and dependent upon each other. Table 2.1 shows how the PVG duties relate to each other.

Table 2.1: Relationships Between the Duties of the PVG



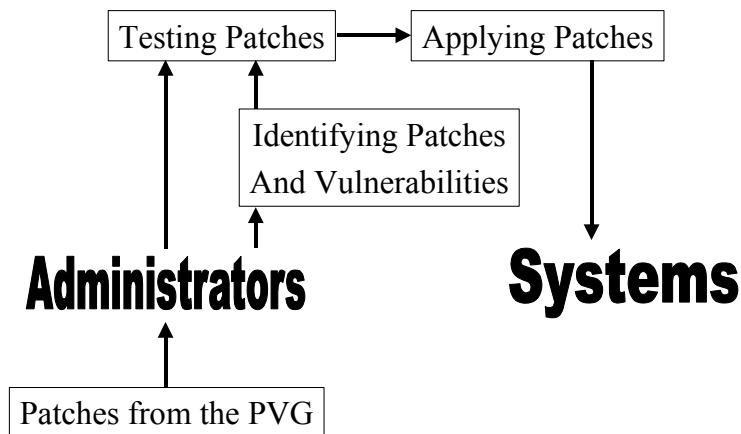
2.2 Systems Administrator Patching Responsibilities

Despite the existence of the PVG, system administrators should remain personally responsible for patching the systems under their control. System administrators will use the PVG as a primary resource for patching their systems and may work directly with PVG personnel when applying a patch. However, system administrators will also be responsible for identifying patches and vulnerabilities software not being monitored by the PVG. They will be trained how to identify such patches and vulnerabilities by the PVG. The duties of systems administrators will be as follows:

1. **Applying Patches Identified by the PVG.** Systems administrators will receive patch and vulnerability information via e-mail or a website from the PVG. The administrators must then decide if each patch is applicable to the systems under their control and to decide the prioritization of how they will install their patches. In some cases, the administrators may decide not to install the patches at all (e.g., where the patch would harm legacy systems). The PVG will recommend patches to administrators and recommend a priority, but the final implementation responsibility should usually lie with each administrator to protect the systems under their control.
2. **Testing of Patches on Specific Target Systems.** The PVG will test patches whenever possible, thus alleviating the need for each administrator to test each patch separately. However, patches must be tested on a system configured almost exactly like the target system because system configuration issues often cause patches to fail. Thus, except for organizations with largely homogenous software platforms, it will be difficult for the PVG to perform patch testing. When a patch has not been tested by the PVG, the local administrators are responsible for performing such testing. The PVG should provide guidance to administrators on how to test patches. Section 5.4 provides information on testing patches.
3. **Identifying Patches and Vulnerabilities Associated with Software Not Monitored by the PVG.** As previously mentioned, the PVG’s organizational software inventory may not contain all software used within an organization. Local administrators should be aware of which software packages are covered by the PVG. They are responsible for investigating patches and vulnerabilities for software not in the inventory but that is installed on systems under their control. The PVG should provide guidance to administrators on how to research patches for such software. Sections 3 and 4 discuss where and how to monitor for patches and vulnerabilities.

These three systems administrator duties are interrelated and dependent upon each other. Table 2.2 shows how the duties relate to each other.

Table 2.2: Relationships Between the Duties of Systems Administrators



This model addresses the burden that exists on systems administrators to patch their systems by providing them a central patching resource in the form of the PVG.

However, the model keeps the responsibility for patching in the hands of local administrators. A drawback to this approach is that some systems administrators may not install patches that were sent to them by the PVG. An internal administrative process could be created for the administrator to acknowledge whether the patches were implemented. Additional accountability can be obtained by the PVG performing periodic network and host vulnerability scanning. However, note that such automated scanning tools do not generally cover all vulnerabilities and patches.

One variation on this model is to have the PVG confirm with each administrator that the appropriate patches have been installed. In this variation, the level to which administrators are held accountable will rise dramatically. For large organizations that take this approach, they may wish to create a website where administrators can input the action they took for each released patch. The PVG would simply update the website with the latest patches and provide a simple form (e.g., two or three check boxes will probably suffice) for each administrator to fill out. For the systems under their control, the administrator may input that they, (1) installed the patch on all systems, (2) installed the patch on servers only, (3) installed the patch on desktop systems only, (4) did not install the patch, or (5) thought that the patch was inapplicable to their particular environment. Although the overhead involved in attempting to keep each administrator accountable for properly handling every patch may be high, this cost may be acceptable given an organization's ability to quantify the level of security (in regard to patching) of their systems.

3. Identifying Vulnerabilities and Applicable Patches

Vulnerabilities are weaknesses in software that can be exploited by a malicious entity to gain greater access and/or permission than it is authorized to have on a computer. Not all vulnerabilities have related patches; thus, system administrators must not only be aware of vulnerabilities and patches, but also mitigate “unpatched” vulnerabilities through other methods (e.g., firewalls, router access control lists). It is a common mistake among system administrators to monitor only patches and not vulnerabilities. Although this omission is understandable given the time pressures many system administrators face, this omission can be dangerous because the system administrator’s chief adversary, the attacker, spends considerable time monitoring and exploiting vulnerabilities. The number of new vulnerabilities being discovered is increasing each year. NIST’s ICAT Metabase (<http://icat.nist.gov>) reports the number of vulnerabilities discovered within the last several years (see Table 3.1).

Table 3.1: Number of New Vulnerabilities Reported to ICAT

Year	Vulnerabilities
1999	859
2000	983
2001	1329

The ICAT data demonstrates that approximately 80 new vulnerabilities are published every month. Within the last three years, ICAT shows that Microsoft’s IIS web server has had over 65 vulnerabilities and Sun’s Solaris has had over 75. For most system administrators, the most difficult part of the patch implementation process is keeping abreast of the latest developments in patches. This can be a complex task, particularly for system administrators responsible for maintaining a heterogeneous environment.

System administrators have several resources at their disposal to monitor the status of vulnerabilities and patches for the systems they support. Each type of resource has its own strengths and weaknesses. Often a system administrator will refer to more than one source to ensure accurate and timely knowledge of new vulnerabilities and patch releases. The most common forums for monitoring the release of patches and identification of vulnerabilities are as follows:

- Vendor websites and mailing lists
- Third-party websites
- Third-party mailing lists and newsgroups
- Vulnerability scanners
- Vulnerability databases
- Other notification tools

- Windows Update.

Note: This section does not provide specific examples of these resources. However, Appendix B provides lists of patching resources for commonly used applications and operating systems.

3.1 Vendor Websites and Mailing Lists

Vendor websites are probably the most popular resource for information used by system administrators to learn about new patches. These sites offer significant amounts of information and are the primary sources for downloading patches. Vendor websites offer several advantages:

- Patches are released by the application vendors.
- Patches downloaded from vendor websites are the most likely to be free of malicious code.
- Vendors provide a wealth of information about vulnerabilities associated with their applications, methods of mitigation, and instructions for installing and using patches.
- Vendors have unique expertise concerning their products.

Vendor website limitations:

- Vendor sites do not generally provide active notification (system administrator must take the effort to visit and review the site frequently).
- Vendors may not provide all relevant information (e.g., alternative mitigation procedures).
- System administrators managing a heterogeneous environment may have to peruse numerous vendor websites for the various products they support.
- Vendors may not be timely in listing new vulnerabilities and patches (many vendors will not report the vulnerability until the patch is available).

Many large vendors maintain mailing lists that enable them to send e-mail messages and notifications of vulnerabilities, patches, and updates to the product users. These lists inform users of new vulnerabilities in a particular vendor's product line without having to regularly visit the vendor's security website. A drawback to these lists is that administrators may have to subscribe to numerous vendor lists if they are managing multiple operating systems or a large number of applications. In addition, vendors may succumb to the temptation to use their mailing lists for marketing purposes (spam), resulting in system administrators ignoring or "filtering" all messages from the list. Note that vendors do not generally distribute actual patches within e-mails since e-mail is not a secure delivery mechanism. If patches are distributed in e-mail, they should be digitally signed and the signature checked before being trusted.

3.2 Third-Party Websites

A third-party vulnerability or patch website is one that is not affiliated with an application vendor, and it may offer more detailed information than a vendor site. These websites may cover a large number of vendors and products or may specialize in a specific vendor or product. The websites often report new vulnerabilities before the vendor reports them because the latter often delay notification until they have confirmed the vulnerability and created a patch or other mitigation techniques. Third-party websites offer several advantages:

- Timely release of new vulnerabilities.
- Depending on the site:
 - Coverage of more than one vendor or product, allowing the system administrator to visit fewer websites to gather information (i.e., “one-stop shopping”).
 - Specialization in a particular product or platform (saving the system administrators time because they do not have to navigate through unrelated data).
- Provision of potentially more acceptable alternatives to the official mitigation techniques provided by the vendor.
- Provision of information that the vendor chooses not to provide.

Third-party websites usually have several disadvantages:

- System administrators need to be cautious of third-party patches because these types of patches are more likely to have unintended consequences or contain malicious code.
- They may not include comprehensive information on patching the vulnerability requiring administrators to research multiple resources. However, this is generally not a problem because administrators should obtain the patches directly from vendor sites, and third-party websites usually provided direct links to such patches.

3.3 Third-Party Mailing Lists and Newsgroups

Mailing lists and newsgroups are threaded discussion groups that rely on e-mail. They are a way for users with similar interests to communicate with each other. The primary advantage of third-party mailing lists and newsgroups is that they allow system administrators and other users to interact in two-way communications, whereas vendor mailing lists support only one-way (vendor to user) communications. This allows system administrators to share their experiences and to ask questions. The principal difference between a newsgroup and mailing list is that a newsgroup is an “officially” recognized Internet forum and, as such, can only be established by following lengthy procedures. In contrast, anybody with a mail server and Internet access can set up a mailing list. In addition, mailing lists may be moderated and participation controlled.

The advantages of third-party mailing lists and newsgroups are as follows:

- Allow interaction between system administrators
- Reduce the number of sites that a system administrator is required to actively search
- Allow a system administrator to learn directly from the experiences of others (e.g., are there problems associated with a particular patch, does it really correct the problem)
- May provide a workaround to be used until a patch is released.

The disadvantages of third-party mailing list and newsgroups are as follows:

- Generate large number of e-mails that may not be useful to system administrators
- Potentially release sensitive information to unauthorized entities (a system administrator who asks questions relating to their system can inadvertently invite a hacker to try to exploit that vulnerability)
- Potentially increase exposure to malicious code because third-party fixes and workarounds are often created by unaccountable parties
- Expose an organization to unsolicited advertising (spam)
- Require verification of information because the third-party source may not be recognized authorities
- May provide links to self-testing sites that automatically launch an exploit against hosts that visit the site (this may cause problems if an unpatched system visits the site).

3.4 Vulnerability Scanners

Vulnerability scanners are commonly used in many organizations to identify vulnerabilities on their organization's hosts and networks. A vulnerability scanner automatically identifies not only hosts and open ports on those hosts, but also any associated vulnerabilities. It will identify a host's operating system and active applications and then compare these with its database of known vulnerabilities. Vulnerability scanners employ large databases of vulnerabilities to identify vulnerabilities associated with commonly used operating systems and applications. When a match is found, the scanner will alert the operator to a possible vulnerability. Most vulnerability scanners also generate reports to help administrators fix the discovered vulnerabilities. See NIST Special Publication 800-42, *Guidelines on Network Security Testing*, for detailed advice on the use of vulnerability scanners.

Vulnerability scanners provide the following capabilities:

- Identifying active hosts on networks.
- Identifying active and vulnerable services (ports) on hosts.

- Identifying vulnerabilities associated with discovered operating systems and applications.
- Testing compliance with host application usage/security policies.
- Vulnerability scanners can help identify out-of-date software versions and applicable patches or system upgrades. They can also be configured to validate compliance with, or deviations from, the organization's security policy. In addition, vulnerability scanners can sometimes automatically make corrections and fix certain discovered vulnerabilities. (This assumes that the operator of the vulnerability scanners has “root” or “administrator” access to the vulnerable host.)

3.4.1 Advantages and Disadvantages

Vulnerability scanners provide system and network administrators with tools that can be used to proactively identify and address vulnerabilities before an adversary discovers them. A vulnerability scanner is a relatively fast and easy way to quantify an organization's exposure to surface vulnerabilities.⁴

However, vulnerability scanners have some significant weaknesses. Generally, they identify only surface vulnerabilities and are unable to address the overall risk level of a scanned network. The scan process itself is highly automated. Because vulnerability scanners can have a high false positive error rate (reporting vulnerabilities when none exist), an individual with expertise in networking and operating system security and administration must interpret the results.

Vulnerability scanners can generate significant amounts of network traffic. This traffic may have a negative impact on the hosts or network being scanned or on the network segments the scanning traffic is traversing. Many vulnerability scanners also include tests for denial-of-service (DoS) attacks that, in the hands of an inexperienced user, can have a considerable negative impact on scanned hosts.

Another significant limitation of vulnerability scanners is that their ability to recognize the latest vulnerabilities depends on the constant updating of the scanner's vulnerability database. Before running any scanner, a system administrator must be sure to install the latest updates to its vulnerability database. Some vulnerability scanner databases are updated more regularly than others (frequency of updates should be a major consideration in choosing a vulnerability scanner).

Vulnerability scanners are better at detecting well-known vulnerabilities than they are at finding more esoteric ones because it is impossible for any one product to incorporate all known vulnerabilities in a timely manner. In addition, manufacturers may elect to exclude some vulnerability detection in order to keep the speed of their scanners high

⁴ A surface vulnerability is a weakness as it exists in isolation—that is, without any other vulnerability. The difficulty of identifying the risk level of vulnerabilities is that they rarely exist in isolation. For example, several “low-risk” vulnerabilities could exist on a particular network that, when combined, present a high risk. A vulnerability scanner would generally not recognize the danger of the combined vulnerabilities and thus would assign a low risk to each, leaving the network administrator with a false sense of confidence in his or her security measures. A more reliable way to identify the risk of vulnerabilities in aggregate is through penetration testing. For more information on network security testing, see NIST Special Publication 800-42, *Guideline on Network Security Testing* (<http://csrc.nist.gov/publications/>).

(more vulnerabilities detected require more tests, which slows the overall scanning process).

3.4.2 Types of Vulnerability Scanners

Vulnerability scanners can be of two types: network scanners and host scanners. Network scanners are used to map an organization's network and identify open ports, vulnerable software, and misconfigured services. In most cases, these scanners are not limited by the operating system of targeted systems. They can be installed on a single system on the network and can quickly locate and test numerous hosts. Host scanners, on the other hand, must be installed on each host to be tested. These scanners are used primarily to identify specific host operating system and application misconfigurations and vulnerabilities. Host scanners have high detection granularity and usually require not only host (local) access but also a root or administrative account. Some host scanners offer the capability of repairing misconfigurations.

Vulnerability scanners vary widely in capability and performance. Some of them perform optimized searching and can scan a host or network much faster than other systems. Some of them provide detailed reports and information about fixing each discovered vulnerability while others provide only the most basic information about which vulnerabilities were found.

3.4.3 Vulnerability Scanning Practices

Organizations should conduct vulnerability scanning to validate that operating systems and major applications are up to date on security patches and software version. Vulnerability scanning results should be documented and discovered deficiencies corrected. The following corrective actions may be a necessary follow-on to vulnerability scanning:

- Upgrade or patch vulnerable systems to mitigate identified vulnerabilities as appropriate.
- Deploy mitigating measures (technical or procedural) if the system cannot be immediately patched (e.g., if application system upgrade will make the application running on top of the operating system inoperable) to minimize the probability of the system being compromised.
- Tighten the configuration management program and procedures to ensure that systems are upgraded routinely.
- Modify the organization's security policies, architecture, or other documentation to ensure that security practices include timely system updates and upgrades.

3.5 Vulnerability Databases

Vulnerability databases are collections of searchable information on information system vulnerabilities. Many of these databases are publicly accessible via the web. These websites, generally run by third parties not affiliated with software vendors, can provide a wealth of information to system administrators and security professionals. They strive to cover most operating systems and software applications. Because they are not

affiliated with software vendors, they often provide information that the vendor, or other organizations affiliated with the vendor, may not provide.

Vulnerability databases tend to be the quickest to report new vulnerabilities, which is a benefit and disadvantage. On the one hand, they provide timely information on vulnerabilities that are critical to the success of a system administrator in securing their network. On the other hand, the sites do not provide the organized vetting of vulnerabilities that occurs with the Common Vulnerabilities and Exposures (CVE) list and ICAT (see Sections 4.1 and 4.2). This deficiency means the same vulnerability may be reported more than once, which dramatically increases the occurrence of false reports.

Although the quantity and quality of information vary to some degree from site to site, vulnerability databases typically include the following types of information:

- **Vulnerability Overview**—This generally consists of an introduction to the vulnerability that includes the following:
 - CVE Number—Number assigned by CVE, if applicable (see Section 4.1)
 - Classification—Type of vulnerability (buffer overflow, design error, etc.)
 - Date of First Publication—Date the vulnerability was first publicly identified
 - Date of Last Update or Revision—Date the vulnerability or patch information was last updated
 - Vulnerable Systems—Operating system, application, or hardware affected by the vulnerability.
- **Discussion or Analysis**—Detailed information on the vulnerability. Information can range in length from one paragraph to several pages depending on the complexity of the vulnerability. This discussion can be highly technical.
- **Solution**—A detailed discussion on mitigating or eliminating the vulnerability. Generally contains hyperlinks to the pertinent vendor’s website for patches and updates. If available, other mitigation techniques will also be included. This solution section may also discuss any negative impacts of the vendor’s patch, if applicable.
- **Exploit**⁵—Includes information on exploiting the vulnerability and any applicable software code. May also contain links to other sites that have more information and exploit code. This information can be useful to the system administrator in testing whether their system is susceptible to exploitation (before or after the patch is applied). However, great care should be exercised in using these techniques so as to not cause unintended harm to other systems.

⁵ Exploits are documented procedures, programs, and/or scripts that take advantage of vulnerabilities. Many vulnerability databases provide exploit instructions or code for most identified vulnerabilities. Exploit programs or scripts are actually just specialized software tools for exploiting a specific vulnerability.

- **Credit**—Recognizes those who identified the vulnerability, created the exploit, or otherwise provided information or assistance. Often contains hyperlinks to the website(s) of the contributor(s).

Overall, vulnerability databases are one of the most powerful weapons in the system administrator's arsenal. Even if a system administrator relies principally on other sources for vulnerability information, the general news and discussions provided on the vulnerability database sites can prove invaluable.

3.6 Other Notification Tools

Because the task of keeping up with releases of patches and reports of vulnerabilities has become more burdensome, new tools and applications have been created to allow system administrators to receive automated and customized notifications for the systems they support. These tools are provided by vendors (e.g., Microsoft's Critical Update Notification application) and third parties (e.g., Cassandra and Security Focus). For more information about obtaining these and similar products, see Appendix B. Some products, such as Cassandra, are free, while others require a one-time fee or subscription.

The advantages of these notification tools are as follows:

- Customization so that notification can be limited to those applications and operating systems of interest (reducing the time spent scrutinizing multiple alerts that do not apply to one's systems)
- Real-time alerts to the system administrator (e.g., not requiring them to visit a web page).

The disadvantages of these notification tools are as follows:

- Cost (for fee-based services)
- Information quality (these sources are only as good as the underlying information database)
- Lag time inherent in certain of these services
- They are somewhat invasive since an administrator must tell a third party organization their network's configuration details.

3.7 Windows Update

Microsoft's Windows Update website does simplify the update notification and installation process. In essence, Windows Update is an online extension of Windows. Currently Windows Update has two separate sections: the Product Updates catalog and the Support Information area.

The Product Updates catalog provides fixes, updates, and enhancements to Windows and many programs that are included with Windows (e.g., Internet Explorer and Media

Player). It allows users to select updated or new components to add to their Windows installation, including security patches.

Users can select as many components and updates as they require and download and install them straight from the Web to their computers. Specifically, when a user first accesses the Product Updates page, Windows Update scans their system to ascertain the system's characteristics (including Windows version, service pack level, and applications installed). This scan is performed without sending any information to Microsoft or over the Internet. After Windows Update has determined the exact parameters of the Windows installation on the user's computer, it creates a personalized catalog of available updates.

When this catalog page appears, users can select the updates required. Users should download most Critical Updates recommended for their system because these will rectify known problems (often performance or security issues). As with any patch, critical updates may cause unforeseen problems; however, such problems are unlikely on standard desktop systems. More information on using Windows Update is provided in Appendix E.

The Support Information area of the Windows Update site provides additional information and assistance on updating and patching Microsoft Windows systems. It contains sections on frequently asked questions, known issues, and contact information for Microsoft support. It also provides links to other Microsoft sites that support Microsoft products not yet supported by Windows Update (e.g., Exchange Server, SQL Server).

4. Government Vulnerability Identification Resources

This section identifies specific U.S. government-funded resources that system administrators and computer security officers can use to identify vulnerabilities and patches for their systems.

4.1 CVE Vulnerability List

The CVE vulnerability naming scheme is a dictionary of standardized names for most publicly known IT vulnerabilities. This emerging industry standard has achieved wide acceptance by the security industry and a number of government organizations. This standard effort is funded by Federal Computer Incident Response Center (FedCIRC) (see Section 4.5) and the technical analysis work is done at MITRE Corporation. General CVE information is available at <http://cve.mitre.org>. The vulnerabilities listed in CVE can be viewed using the NIST ICAT vulnerability index at <http://icat.nist.gov> (see Section 4.2).

CVE provides the computer security community with the following:

- A comprehensive list of publicly known vulnerabilities
- An analysis of the authenticity of newly published vulnerabilities
- A unique name to be used for each vulnerability.

In the context of this publication, the CVE is useful as an authoritative listing of most known vulnerabilities. It is unlikely that systems administrators and computer security personnel will directly use CVE when finding vulnerabilities and patching systems. However, we do recommend using CVE-compatible vulnerability resources monitoring for vulnerabilities. See <http://cve.mitre.org/compatible> for a list of CVE-compatible security products and services. It is also important to be aware of how completely a vulnerability service covers the vulnerabilities listed in CVE. For example, many administrators rely on the CERT/CC advisories (see Section 4.4) to warn them of vulnerabilities in their networks. However, CERT/CC publishes advisories only on the highest impact and time-critical issues, around 100 advisories per year, whereas CVE lists almost 1,000 new vulnerabilities every year⁶. Thus, systems administrators should augment the vulnerability information they receive from the CERT advisories with other vulnerability sources. It is important that administrators have access to resources that cover nearly all known vulnerabilities.

Decisions regarding which vulnerabilities and security exposures should be included in the CVE are made by the CVE Editorial Board. The board includes members from numerous information security-related organizations, including commercial security tool vendors, academia, research institutions, government agencies, and other prominent security experts.

Through open and collaborative discussions, the board identifies which vulnerabilities or exposures should be included in CVE and then determines the common name and description for each entry. The process begins with the discovery of a potential security

⁶ CERT also issues vulnerability notes and these cover a much larger set of vulnerabilities than the advisories.

vulnerability or exposure. The information is then assigned a CVE candidate number (e.g., CAN-2001-0002). The Editorial Board discusses the candidate and votes on whether it should be given a CVE number (e.g., CVE-2001-0002). Figure 4.1 depicts the CVE naming process.

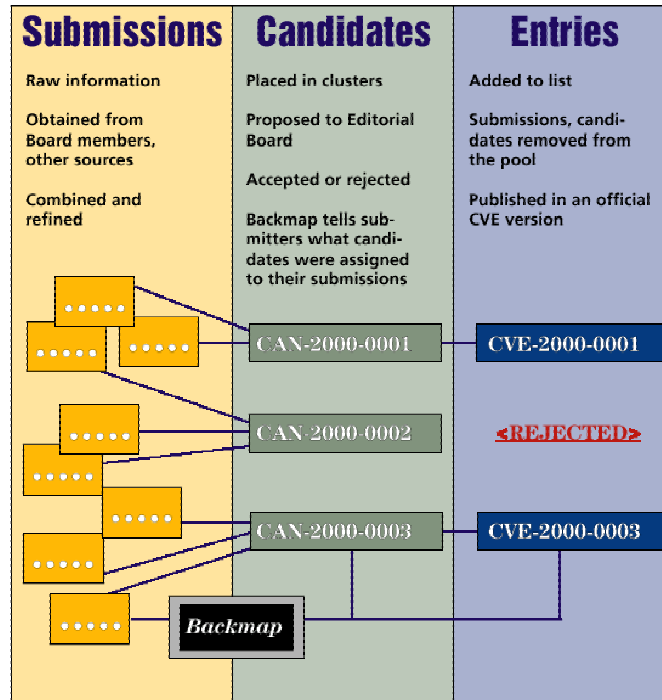


Figure 4.1: CVE Naming Process

As of January 2002, CVE contained more than 3,500 vulnerabilities.

4.2 NIST ICAT Vulnerability Index

The NIST ICAT Metabase is a searchable index of computer vulnerabilities based completely on the standard CVE vulnerability list (see Section 4.1). It links users to a variety of publicly available vulnerability databases and patch sites, thus enabling system administrators to find and fix the vulnerabilities existing on their systems. The ICAT Metabase takes the CVE list to the next level by including detailed information about each vulnerability or security exposure contained in the CVE and CVE candidate lists. ICAT allows users to search with a fine granularity, a feature unavailable with most vulnerability databases. Each vulnerability is characterized with more than 40 attributes (e.g., vulnerable software names, vulnerability consequences, and a range of the related exploits).

In the context of this publication, the usefulness of ICAT is that systems administrators and security officers can identify vulnerabilities that may exist in their network. They can perform this analysis by searching ICAT for the software that is used in their network and reviewing the resultant ICAT vulnerability summaries. Each ICAT vulnerability summary will then provide links to more detailed vulnerability and patch information.

ICAT is updated whenever new vulnerabilities are added to the CVE list. ICAT analysts currently process up to 40 vulnerabilities per week, and they update ICAT weekly whenever new vulnerabilities are available⁷. Periodically, the CVE standards committee releases a large batch of vulnerabilities, and it takes ICAT several weeks to catch up. Thus, given the time it takes for a vulnerability to be added to the CVE list and then to ICAT, we recommend that ICAT be used in conjunction with a service that provides immediate notification of extremely serious vulnerabilities (such as the CERT/CC advisories).

In addition to being accessible via the web, ICAT can also be downloaded in Microsoft Access 2000 and delimited text file formats. This allows system administrators to conduct searches without access to the Internet and to integrate ICAT <http://icat.nist.gov/icat.cfm?function=download>.

In conjunction with NIST, CERIAS at Purdue University distributes a vulnerability notification product named Cassandra that is based on ICAT. Cassandra allows a system administrator to enter the names and versions of the software used on their computers and networks into a database. Cassandra then sends the system administrator e-mails about new CVE entries and candidates that meet the system administrator's software profile. Unlike the alerts sent by many advisory systems, which may or may not be applicable to a system administrator, almost every vulnerability notification sent by this service will represent a vulnerability in the software that the system administrator included in their profile. Further, using their Cassandra interface, system administrators can search ICAT based on their software profile. Because Cassandra stores the products used by the system administrator, this eliminates having to search ICAT separately for each product every time they need to check for new vulnerabilities. More information about Cassandra, including configuration, can be found at <https://cassandra.cerias.purdue.edu/main/index.html>.

Detailed instructions on using the web-based ICAT are provided in Appendix C of this document.

4.3 National Infrastructure Protection Center

Established in February 1998, the National Infrastructure Protection Center (NIPC) is the part of the Federal Bureau of Investigation that provides threat assessment, warning, and investigation concerning threats or attacks against critical network infrastructures. A subset of its duties is to provide information regarding computer vulnerabilities.

The NIPC produces three levels of computer security warnings that address issues of significant impact. The three levels of NIPC threat warnings are as follows:

⁷ These numbers may change as ICAT is provided with greater or fewer resources.

- **Assessments**—The lowest level of warning. These address broad, general incidents or issue awareness information and analyses that are significant and current, but that do not necessarily suggest immediate action.
- **Advisories**—These address significant threat or incident information that suggests a change in readiness posture, protective options, and/or response.
- **Alerts**—The highest level of warning. These address major threat or incident information concerning imminent or in-progress attacks targeting specific national networks or critical infrastructures.

Links to these warnings can be found on the NIPC website at <http://www.nipc.gov/warnings/warnings.htm>. Note that these warnings are issued for only very high-profile threats and thus will cover only a small percentage (albeit the most important subset) of the vulnerabilities discovered each year. Figure 4.2 illustrates a NIPC alert sample.

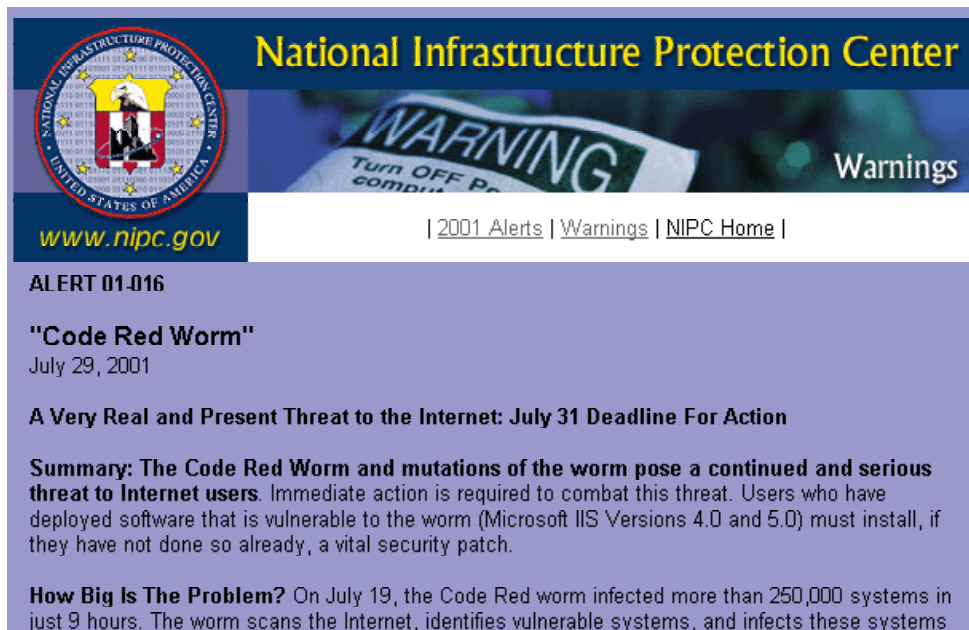


Figure 4.2: Sample NIPC Alert Sample

In addition to its warnings, the NIPC jointly publishes with the Systems and Network Security (SANS) Institute a list of the top vulnerabilities that should be fixed in today's networks. This list is determined by a group of experts in government, academia, and industry. It is a valuable starting point for organizations that do not already have a regular patching system in place. This list is CVE-compatible, and each vulnerability is cross-referenced into NIST's ICAT. The list is available at <http://www.sans.org/top20.htm>.

Lastly, the NIPC produces a biweekly publication called "Cybernotes," which lists every new vulnerability published during the previous two weeks. "Cybernotes" is an excellent resource for a digest of the latest known vulnerabilities. "Cybernotes" is

available at <http://www.nipc.gov/cybernotes/cybernotes.htm>.

4.4 CERT/CC

The CERT/CC is a center of Internet security expertise at the Software Engineering Institute, a federally funded research and development center operated by Carnegie Mellon University. CERT/CC responds to major vulnerabilities and threats by issuing advisories, incident notes, and vulnerability notes.

Advisories are warnings about the most critical vulnerabilities, and organizations should take steps to address these vulnerabilities immediately. CERT/CC advisories can be found at <http://www.cert.org/advisories/>. We also recommend subscribing to the CERT/CC advisory e-mail list in order to be more quickly informed of the latest threats. To subscribe to the mailing list, send e-mail to majordomo@cert.org. In the body of the message, type “subscribe cert-advisory.”

The CERT advisories date back to the inception of CERT in 1988, and advisories are organized sequentially by year. For example, the first advisory for 2001 is numbered CA–2001-01 (CA = CERT Advisory, 2001 = year and 01 = first advisory of the year).

Several methods are provided for identifying relevant advisories on the CERT site: searching for keywords using the CERT search engine, accessing the CERT advisories page and selecting the appropriate year (which will list summaries of all advisories for a given year), and accessing the CERT Current Activity web page at <http://www.cert.org/advisories/> (which provides a list “of the most frequent, high-impact types of security incidents and vulnerabilities currently being reported to the CERT” [see Figure 4.3]).

The screenshot shows the CERT/CC Current Activity page. The header includes the Carnegie Mellon Software Engineering Institute logo and navigation links: Home, Site Index, Search, Contact, and FAQ. Below the header, there are links for vulnerabilities, incidents & fixes, security practices & evaluations, survivability research & analysis, and training & education. The main content area is titled "CERT/CC Current Activity" and contains the following text:

The CERT/CC Current Activity web page is a regularly updated summary of the most frequent, high-impact types of security incidents and vulnerabilities currently being reported to the CERT/CC.

Last reviewed: Wednesday, October 10 2001 @ 10:08:23 EDT (0400 UTC)

	Date Added	Last Updated
• W32/Nimda	September 18	October 1
• W32/SirCam	July 23	September 20
• Scans and Probes	-	August 17

Below the table, there are sections for "W32/Nimda" and "W32/SirCam". The "W32/Nimda" section states: "The CERT/CC continues to receive a steady stream of reports of W32/Nimda although the volume of reports has dropped significantly since it first appeared on September 18th. Sites are strongly encouraged to read [CERT Advisory CA-2001-26](#) for detailed information on W02/Nimda." The "W32/SirCam" section states: "The CERT/CC continues to receive reports of a piece of malicious code known as W32/SirCam."

Figure 4.3: CERT Current Activity Page

Each CERT advisory provides significant amounts of information to assist the system administrator and security professional. The advisories are updated as new information

is discovered and patches become available. Each advisory begins with a title, the advisory's original release date, and date of last revision. After this introductory information, the advisory is divided into several major sections that help a system administrator identify and mitigate vulnerabilities and apply patches.

- **Systems Affected**—Provides a list of software and/or hardware affected by the vulnerability
- **Overview**—Provides a brief description of the vulnerability
- **Description**—Presents a detailed analysis of the vulnerability and provides hyperlinks to additional sources of information
- **Impact**—Describes the possible effects of a successful exploitation
- **Solution**—Provides information regarding correcting the problem, including patches, if available.

CERT/CC also produces incident notes that describe current hacking or virus activity. These notes are available at http://www.cert.org/incident_notes/ and are of a less time-critical nature than the advisories.

Lastly, the CERT/CC produces vulnerability notes. These notes cover a larger set of vulnerabilities than are covered with the advisories (but still cover only a fraction of the CVE entries). Although these vulnerabilities were not important enough for advisory, CERT/CC still deemed them important. The vulnerability notes are stored in a database with a web front end that allows easy perusal of the vulnerabilities collection. This database is available at <http://www.kb.cert.org/vuls>.

4.5 Federal Computer Incident Response Center (FedCIRC)

The FedCIRC, a program of the General Services Administration, Federal Technology Service (FTS) provides a focal point for incident reporting and handling within the Federal civilian government.

In regard to vulnerabilities, FedCIRC funds the CVE work at MITRE. It also provides vulnerability alerts in cooperation with the CERT/CC. These vulnerability alerts are identical to the ones produced by CERT/CC, except that they now have a stamp of approval by a civilian government entity. It is recommended that federal employees sign up to receive the CERT/CC advisories through FedCIRC instead of directly from CERT/CC, because the FedCIRC is the official channel for such information. To sign up to the FedCIRC advisories, send an e-mail to fedcirc@fedcirc.gov. In addition to receiving FedCIRC advisories through the e-mail, the various CERT/CC warnings can be viewed at the FedCIRC Website (<http://www.fedcirc.gov>).

FedCIRC has immediate plans to launch a patch capability that will enable US government agencies to obtain notification of vulnerability alerts and patches. The Patch Authentication and Dissemination Capability is a free service for all government agencies. In order to gain access to the service, agencies are required to register for the subscription-based service. The service will provide technology profile management, vulnerability management, intelligence alerts and bulletins, as well as the ability to

receive timely notification of alerts and patch information via email, pager, or other wireless devices. The technology profile management service allows agencies to create profiles for their agency specific technologies and receive alerts and patches on those technologies only. Vulnerability management allows agencies to keep a record of alert notifications and track the progress of vulnerability mitigation. Intelligence alerts are provided to keep subscribers informed of up to date security vulnerabilities, Trojan horses, worms, and denial of service attacks.

This service will complement any agency implementing the PVG methodology since it will provide the PVG with timely alert notification and generically tested patches. However, the initial capability will not cover all operating systems and applications. Thus, use of it will have to be combined with other vulnerability services in order to obtain complete coverage. FedCIRC plans to expand the list of covered technologies in the future based upon subscriber requests. For more details on the Patch Authentication and Dissemination Capability please send an email to fedcirc-info@fedcirc.gov or look on the FedCIRC website.

5. Patching Procedures

5.1 Patching Priorities

With the exception of small networks, it is a complex and difficult endeavor for network administrators to install all patches in a timely manner. This is attributed not only to time and resource constraints but also to the much greater complexity and heterogeneity of larger networks. Thus, setting priorities for which systems to patch in what order is essential for an effective patch process.

The first step in this prioritization process requires an organization's systems to be inventoried (if such an inventory does not already exist). This inventory would include the following:

- **Hardware**—Type, manufacturer, serial number, and firmware or basic input output system (BIOS) version
- **Operating System**—Type, manufacturer, version number, and current patch level
- **Major Applications**—Type, manufacturer, version number, and current patch level.

Once a reasonably complete inventory is available, a risk assessment needs to be performed to determine the prioritization of the systems to be patched. Consult NIST Special Publication 800-30, *Risk Management Guide for Information Technology Systems*, for guidance on performing risk assessments. The general idea in performing a patch-focused risk assessment is to evaluate each of your systems for the following:

- **Threat**—An activity with the potential of causing harm to a computer system or network. Examples of systems that frequently face high threat levels are web servers, e-mail servers, and other hosts traditionally accessible to external users and servers that contain high-value information such as financial databases, proprietary information, or other items of interest to internal and external entities. Threats can be determined through the analysis of the specific threat level historically faced by the organization and by general threat updates from external organizations (e.g., CERT/CC, NIPC).
- **Vulnerability**—A flaw or weakness that may allow harm to occur to a computer system or network. Systems that often have significant vulnerabilities include web servers, systems installed by inexperienced personnel, and systems that allow unauthenticated access to resources. A quick search of vulnerability resources (e.g., ICAT) should assist in the identification of vulnerable hosts.
- **Criticality**—A measure of how important or valuable a system is to the organization's mission. Systems frequently considered mission-critical include mail servers, database servers, and network infrastructure nodes. Interviews with appropriate staff should help determine mission-critical systems. For additional guidance on criticality, see NIST Special Publication 800-18, *Guideline for Developing Security Plans for Information Technology Systems*.

Systems that face significant threats, are vulnerable, and are mission-critical should be patched before hosts that face few threats, are secure, and are not mission-critical.

5.2 Obtaining Patches

As discussed previously, the most important step in maintaining an up-to-date system is identifying and installing the patches or updates available for a system or the applications running on that system. Although vendor websites should always have the most up-to-date patch information for their software, relying on information available through vendor websites is not always the easiest means to remain up to date. For example, it is difficult for system administrators in heterogeneous environments to frequently visit multiple websites. For this reason, many third-party security sites have emerged that maintain databases for identifying available patches. Users can use these databases to search for particular software or operating systems. The search results usually contain information about the vulnerabilities associated with the specified software and system along with links to vendor sites with the patch information or downloads.

Third-party websites, mailing lists, and newsgroups are often one of the first places where information on a new vulnerability becomes available and can be useful for identifying patches. System administrators should be cautious when retrieving patches from third parties. Certain third-party websites, mailing lists, and newsgroups are notorious distribution points for a variety of malicious programs. These programs are often advertised as an official patch or tool for correcting a known vulnerability, whereas in fact they are programs enabling the installation of a backdoor that hackers can use to compromise or otherwise identify vulnerable systems. Patches should be downloaded only from the vendor or other trusted sources.

5.3 Patching Precautions

Whatever the source of the patch, a system administrator should take several precautions before attempting installation. First, most vendors today provide some type of authentication mechanism. The downloaded patch should be checked against any of the authenticity methods the vendor provides.

- Verify cryptographic checksums—usually Message Digest 5 (MD5) checksum
- Verify Pretty Good Privacy (PGP) signatures
- Verify digital certificate.

Some of these methods, such as verifying digital signatures, are highly automated, requiring little user interaction. Others, such as MD5 checksum, require the user to visit the vendor's website to check the MD5 checksum listed there against the downloaded patch. Although these methods add another level of authentication, they are not foolproof. For example, hackers have obtained digital signatures that enabled them to distribute patches in the name of a major software company. Although this was only a proof of concept and the hackers did not attempt to exploit the fake digital signatures, they easily could have done so.

A virus scan should also be run on all patches before installation. Before running the scan, the system administrator should ensure that the virus signature database in the antivirus program is up to date. Again, this system is not foolproof. If a hacker has created an entirely new Trojan and included it with the patch, it might not be detected by the virus scan.

If the patch is not distributed compiled (that is, if the user must compile the source code before installation), the system administrator may wish to perform a code review before compiling and installation. The administrator is simply checking that the source code appears to be legitimate patch code as opposed to a hacker tool. Distributing patches as source code is a common situation for Linux patches and some Unix operating systems, and it gives administrators more control and knowledge of the patching process. If administrators are not comfortable performing the code review, they may want to find an expert who is or ensure that the source code is downloaded from a trusted source.

Before installing the patch, and especially if they do not have the time or resources to perform a test on the patch before employing it on a production system, administrators should find out what experiences others have had in installing or using the patch. For instance, the system administrator should attempt to learn whether the patch:

- Corrects the vulnerability
- Opens an old vulnerability
- Creates a new vulnerability
- Reduces reliability
- Degrades performance
- Is incompatible with other required applications.

If one or more of the above problems applies, the system administrator will need to consider whether the disadvantages outweigh the benefits of installing the patch. If installing the patch is not critical, it may be better to wait until the vendor releases a newer patch that corrects the major issues (this is a common occurrence). The more complex the patch (the file size of the patch gives some indication of complexity), the more likely one or more of the above issues will arise. For example, a Microsoft Windows hotfix (usually just a single patch) is much less likely to cause problems than a service pack that contains a large number of fixes. Also, the ability to “undo” or uninstall a patch should be considered; however, even when this option is provided, the uninstall process does not always return the system to its previous state.

Another consideration is the frequency of patch releases. For example, during the first nine months of 2001, Microsoft released 49 security bulletins with as many as seven bulletins released in one month. In the year 2000, Microsoft released 100 security bulletins, releasing as many as 13 bulletins in one month. Thus, patching could easily become a full-time job for a system administrator.

Before applying a patch, a system administrator and management must consider the following:

- Cost of deploying one patch versus cost of deploying a bundle of patches
- Automated versus manual deployment
- Whether the patch can be consistently deployed throughout all vulnerable systems.

If the decision is made to gather several patches and combine the deployment, there are additional considerations:

- Risks associated with delaying installation of one or more patches so that a few may be bundled
- Complications arising from combining patches (e.g., new vulnerability, incompatibility with current configuration)
- Possibility that installing several patches at once will complicate troubleshooting (e.g., which patch caused the problem?); this potential needs to be weighed against the time saved in installation.

The risk of delaying the application of patches must be weighed carefully. For example, when the vulnerability that was eventually exploited by the Code Red worm was discovered, because there was no immediate danger, many system administrators delayed installing the patch. However, within a month a very virulent worm (Code Red) was released that exploited many of the unpatched systems. In weighing the risk of delay against the labor-saving benefit of combining patches, the following issues must be considered:

- **Threat Level**— Does the organization or systems requiring patching face numerous and/or significant threats? For example, public web servers and most federal government organizations may face high threat levels. Generally, timely patching is critical for these systems. In contrast, for an intranet site that is inaccessible from the Internet, patching can often be delayed because such a site faces a much lower threat level.
- **Risk of Compromise**—What is the likelihood that a compromise will occur? If the vulnerability is easy to exploit, then the patch should be applied swiftly.
- **Consequences of Compromise**—What are the consequences of compromise? If the system is critical or contains sensitive data, then the patch should be applied immediately. This holds true even for noncritical systems if a successful exploitation would lead to “rooting”⁸ of the system.

Unfortunately, neither decision—to apply or not apply a patch—is risk-free. The correct decision is not always clear. Too often, a decision is made to make “no decision,” which may result in a compromised system. System administrators and management must work together to create a systematic process for evaluating patches and determining the appropriate decision within the context of their organization.

⁸ “Rooting” a system is a hacker term for gaining administrative or root-level access to a targeted system. This means that the attacker has gained full control over the targeted system. Any vulnerability that could lead to this level of access should be corrected or patched immediately, even on internal systems.

5.4 Testing Patches

In a perfect world, all patches would be widely tested before release and would work flawlessly. As previously discussed, bugs occur in all software, and patches are no exception. Many patches are extremely complicated and contain significant amounts of code (e.g., Microsoft service packs are often 100 megabytes a piece). In addition, patches are often released in haste in order to quickly repair a vulnerability, which means that they often receive less testing than the original software. Lastly, a patch may change the system behavior such that it causes other programs to crash or otherwise fail (this is especially true of operating system patches). In summary, patches can easily produce unintended consequences.

One of the most important aspects of testing patches is to determine that the patch has not broken any existing software. It is important that this testing be performed on the system being patched or on a system similarly configured because so many possible system configurations exist that the vendor cannot possibly test the patch against all of them. Thus, the patch may have unintended consequences only on your particular configuration. After patch installation, administrators should check that all related software is operating correctly. For operating system patches, the administrator may have to check all software on the system.

In addition to the need to identify any unintended consequences, patches may be tested to ensure that they have patched the vulnerability or corrected the performance issue as intended. This can be accomplished by several methods:

- Check that the files or configuration settings that the patch was intended to correct have been changed as documented in the vendor's documentation.
- Scan the host with a vulnerability scanner that is capable of detecting known vulnerabilities. Note that this technique may not work because vulnerability scanners may not check for the actual presence of the vulnerability but instead may simply look at software version numbers or patch levels.
- Employ exploit procedures or code and attempt to exploit the vulnerability (i.e., perform a penetration test). Only an experienced administrator or security officer should perform such a test since they will be launching actual attacks within a network or on a host. In general, it will be too time-consuming to verify each patch with a penetration test. However, for extremely high threat level vulnerabilities, an administrator may want the additional assurance that a penetration test will provide.

This issue of whether a patch is working correctly is complicated by the fact that the sequence in which patches are applied can be critical to their successful operation. It is quite possible to undo a previous patch by installing another patch. To avoid this problem, it is critical that system administrators follow the vendor's instructions exactly for applying all patches and then test the patches after installation.

Before applying a patch, system administrators must decide whether they should install the patch directly on a production, development, or some other system. Only the responsible system administrator or security officer can make the appropriate determination on whether to perform testing on a development system or on a

production system. This complicated issue is influenced by numerous other issues, as follows:

- Organization's configuration management policies
- Seriousness of the vulnerability
- Threat level of the system with the vulnerability
- Ability to temporarily mitigate the vulnerability through other methods (e.g., firewall rules, permission changes)
- Whether an appropriate system exists on which to test the patch
- Complexity of the patch
- Complexity of the production system
- Number of systems to be patched
- Experiences of others in installing the patch
- Vendor guidance (this includes all vendors whose applications are running on the systems to be patched)
- Previous experience in patching the systems on which the patch will be installed.

A comprehensive patch test consists of not only verifying that the vulnerability has been fixed but also checking to see if the patch caused any existing software to fail. However, for many organizations, it may be too technically difficult or too costly to verify that the vulnerability has been fixed. In these situations, it is still very important to ensure that all existing systems continue to function correctly.

5.5 Applying Patches

Patches can come in various forms; they can be applied by a simple click from a web browser through complex operating systems and kernel-specific compilation of source code. The patching of vulnerabilities may be as simple as modifying a configuration setting or may require the installation of a completely new version of the software. No simple patch application methodology applies to all software and operating systems.

Each vendor of an operating system and application will have a specific—often unique—methodology for applying a patch and updating its product. Consequently, it is recommended that the system administrator read relevant documentation provided by vendors. The tools or utilities used to assist and/or automate this process (see Appendixes F through H) may also vary from vendor to vendor. The guidance provided in this document is an adjunct, not a substitute, for the documentation and recommendations of the product vendors.

5.6 Updating Linux/Unix Operating Systems and Applications

Within a Linux or Unix system, many methods are available for installing a patch or update, depending on the particular operating system and version. The process usually involves compiling the source code for the patches for the specific operating system and kernel version in use. On certain installations, the sequence in which the patches are applied may be important, in addition to location in the directory tree where the installation updates should occur.

Certain distributions also have additional utilities that should be used to install a patch or update. Because of the large variance in procedures for applying a patch from the distributions, the user should consult the vendor and distribution-specific user manuals for detailed instructions. References for common Linux and Unix distributions can be found in Appendix B.

5.7 Updating Network Infrastructure Components

Network infrastructure components (e.g., routers, firewalls, switches, and intrusion detection systems (IDS)) are some of the most critical systems to keep patched. Many different methods are available for patching or updating these systems depending on component type and manufacturer. The process involves obtaining the software from an appropriate vendor and following their specific instructions for installing it.

A few components of the network infrastructure require special consideration when patching. Border routers and firewalls are an organization's first line of defense and should be updated swiftly once a vulnerability is identified. A compromise of one of these systems could lead to the compromise of the entire network. There is at best minimal time (hours to days) to test the patch before application, because attacks attempting to exploit these vulnerabilities are likely to be occurring as soon as the vulnerability is discovered or publicized.

Virus detection programs that are installed on the firewall, e-mail servers, and file servers need to be updated frequently. Antivirus programs rely on a database of virus signatures to recognize a virus. If this database is not up to date, the program cannot recognize newer viruses that represent the greatest threat. These updates should occur at a minimum on a weekly basis and on an ad-hoc basis when a particularly virulent new virus is traversing the Internet. Failure to update these antivirus programs could result in the widespread distribution of one or more viruses within an organization.

Many IDSs, like antivirus programs, rely on a database of attack signatures to recognize attacks. Because new attack techniques are constantly being developed, updating this database is critical to the ability of the IDS to detect and report attacks.

5.8 Updating Windows Operating Systems and Applications

Many methods exist for applying patches to Windows and Windows-based software. Most versions of Windows now have simple "update" buttons that, when clicked, automatically access the Internet and check for an update. If one is available, download and install it. This type of automated feature can be of great assistance to system administrators. If an automatic update feature is not built into the software, the system administrator will have to manually monitor for updates using the procedures described

previously. Even when automated procedures are available, many system administrators prefer to manually install patches because this generally gives them greater control over the process.

To assist in updating its operating systems and certain applications, Microsoft has created a variety of tools and automated techniques to identify necessary patches and install them. These tools are relatively new, and additional functionality is being added continually. Currently, Microsoft offers the following capabilities:

- **Windows Update**—This scans a computer(s) to find operating system updates available through Microsoft. This scan will identify any hotfixes or security patches that are needed in addition to listing other software updates that are available. See Appendix E for more information about Windows Update.
- **Microsoft Office Update**—Like Windows Update, this scans a computer(s) to find Office updates available through Microsoft.
- **Microsoft Personal Security Advisor (MPSA)**—MPSA checks the security stance and patch state of a Windows NT 4.0 Workstation or a Windows 2000 Professional computer. Using this web application, the system administrator can scan a computer and receive a detailed report on that computer's security settings, along with recommendations for updates and improvements. (See Appendix F for more information about MPSA.)
- **Microsoft Network Security Hotfix Checker (HfNetChk)**—This is a command line tool written by Microsoft to assess the patch status for Windows NT 4.0 and Windows 2000 operating systems, as well as the status of hotfixes for IIS 4.0 and 5.0, SQL Server 7.0 and 2000, and Internet Explorer 5.01 and later. (See Appendix G for information and instructions on its use).
- **Qfecheck**—This is a command-line tool released by Microsoft that gives network administrators the ability to track and verify installed Windows 2000 and Windows XP hotfixes (it does not currently support Windows NT) (see Appendix H).
- **Microsoft Security Toolkit, Strategic Technology Protection Program (STPP)**—Microsoft has recently started this program, which with the Microsoft Security Toolkit, is a two-phased program. The first phase of the program is to become secure, and the second phase is to stay secure. More information about this new program and the associated toolkit can be found at <http://microsoft.com/security>.
- **Windows Critical Update Notification**—This tool checks for an Internet connection every five minutes and, when a connection is found, checks for any updates. The tool connects to the Windows Update site and then notifies the user of any critical updates or patches that are available. The frequency of checking for an Internet connection slows to once per hour after the first hour of unsuccessful attempts and stops for 1 day after a successful update. This tool and additional information can be found at <http://support.microsoft.com/support/kb/articles/Q224/4/20.ASP>.

- **Microsoft Security Notification Service**—This free service provides e-mail notification from Microsoft about the security of Microsoft products. Information regarding this service and how to subscribe can be found at <http://www.microsoft.com/technet/security/bulletin/notify.asp>.

All of these Microsoft services or applications provide a means for checking and installing patches. Many forms of patches and methods of applying patches exist. Critical patches can come in the form of critical updates and hotfixes, whereas noncritical updates can come in the form of service packs and product updates. Noncritical updates are usually feature-enhancement packages.

Although service packs can be classified as noncritical updates, they do contain previously released critical updates within them. All Hotfixes and Critical Updates that are released before a service pack (including all patches and updates from previous service packs) are usually bundled into the next service pack when it is released.

When Microsoft Hotfixes are installed, a reboot is usually required. It can be a time-consuming task when multiple hotfixes must be downloaded and installed. Microsoft has released a tool, QChain, that enables hotfixes to be bundled into one package and installed at one time. QChain and additional information can be found at <http://support.microsoft.com/support/kb/articles/Q296/8/61.asp>. This product also allows system administrators to create a customized patch installer for multiple systems that require the same patch(s).

5.9 Automated Patch Distribution and Application Tools

Although applying patches to multiple servers may seem a daunting task, it is especially daunting compared with implementing a patch on hundreds or thousands of desktop systems. This task can be made less burdensome through applications that automatically distribute updates to end-user computers. Some of these patch automation tools are included with network operating system software, whereas third-party vendors distribute others.

The capabilities of these systems vary greatly. Some of these applications focus on the distribution of patches that rely on the system or network administrator to identify a necessary patch and set up the tool to deliver and install the patch. Other tools actively search for necessary patches and automatically notify the system administrator of the available ones. The system administrator, if they approve of the patches, can tell the patch application tool to install the patches on the appropriate hosts.

These patch distribution applications also vary greatly in their support of different operating systems and applications. Those that are bundled with an operating system tend to support the fewest operating systems and applications. Those from third-party vendors are generally compatible with the widest range of systems.

5.10 Reducing the Need to Patch Through Smart Purchasing

Some software products have more vulnerabilities than other products with equivalent purpose and functionality. By considering several factors during the purchasing process, one can reduce the number of future vulnerabilities experienced and thus reduce the need to patch the software. For example, a high-profile analyst group recently

recommended that companies stop using a major web server product because of its consistent history of discovered vulnerabilities. If one continues to use this product, one is likely to experience further vulnerabilities and need to be consistently patching this product. The future likelihood of vulnerabilities, should not be the only factor in purchasing a product, but it should be an element in the decision-making process. The following is a list of techniques for choosing products that are less likely to experience vulnerabilities in the future:

- Search a vulnerability database (such as ICAT) for known vulnerabilities of products under consideration. Examine the kind, type, severity, and quantity of vulnerabilities in the product under consideration. This is not foolproof because it often takes longer for vulnerabilities to be discovered (and patches released) for less popular software products.
- Select a mature product. Recently released products usually require more patches.
- Select less complicated products. More code, features, and services can mean more bugs, vulnerabilities, and patches. Do not purchase a product that has more features than needed. To the extent possible delay implementing recently released major operating systems or applications until the experiences of others can be included in the decision-making process.
- Purchase products that conform to appropriate national or international standards (e.g., NIST Federal Information Processing Standard 140-2 for encryption modules). See NIST Special Publication 800-23, *Guideline to Federal Organizations on Security Assurance and Acquisition/Use of Tested/Evaluated Products*, for more information.

5.11 Training Users to Patch

Making users an integral part of the patching process can be critical. Every user has time to perform lightweight patching of their system (e.g., Windows Update or a virus update) whereas a system administrator may not have time to update each user's systems. Even in organizations that have automatic patch distribution systems, users who are not a part of the automated process (e.g., users on travel with laptops or home telecommuters) need to know how to patch the systems. Users should be trained in easily accomplished patching methodologies that are unlikely to cause problems with their system. This is especially important for organizations that allow telecommuting as a vulnerable home system can threaten the security of an organization's network.

The advent of "one-click" updates (e.g., Windows Update and Norton's Live Update) has made training users much easier. As part of their information systems training, users should be trained to check monthly for Windows Updates. Users should be trained to update their antivirus signatures biweekly or whenever a serious virus outbreak occurs. No patching process is complete without the participation of end users.⁹

⁹ Another important lesson for users is that vendors do not e-mail updates to users. This is critical because several e-mail viruses have masqueraded as patches or updates from vendors to fool users into installing the malicious program on their computer.

5.12 Patching After a Security Compromise

Patching after a security compromise is significantly more complicated than merely applying the appropriate patch. Although applying a patch after a security compromise will generally correct the vulnerability that was exploited, it will not eliminate backdoors¹⁰ or most other changes that might have been introduced by the intruder. For example, the Code Red II worm placed backdoors on compromised systems and then subsequently those backdoors were exploited by the Nimda worm.

Systems that are known or suspected to have been compromised must be patched more carefully than uncompromised systems. If a system has been or is suspected of being compromised, it must be reformatted and reinstalled or restored from a known safe and trusted backup. If that is not possible, significant expertise will be required to manage the possible dangers inherent in compromised systems.

To recover a system from a compromise, take the following steps:

- Disconnect (absent extraordinary circumstances) compromised system(s) from network
- Consult the organization's security policy
- Consult with management, legal counsel, and law enforcement as appropriate
- Analyze the intrusion, including:
 - Modifications made to the system's software and configuration
 - Modifications made to the data
 - Tools or data left behind by intruder
 - System logs, intrusion detection, and firewall log files.
- Restore the system
 - Two options:
 - Install clean version of operating system, or
 - Restore from backups (much more risky)
 - Disable unnecessary services
 - Apply all security patches
 - Change all passwords (even on other uncompromised hosts as required)

¹⁰ A backdoor is a secret avenue of access placed on a compromised computer system by a hacker that allows future unauthorized access.

- Reconfigure network security elements (e.g., firewall, router, intrusion detection system, etc.) to provide additional protection.
- Document any lessons learned.

System administrators should consider the following when deciding whether to reinstall the operating system of a compromised system:

- Level of access gain by the intruder (e.g., root, user, guest, system, etc.)
- Purpose of compromise (e.g., web page defacement, illegal software repository, platform for other attacks)
- Method of system compromise
- Hacker's actions during and after compromise (see log files, intrusion detection reports, etc.)
- Duration of compromise
- Extent of compromise on network (i.e., the number of machines compromised)
- Results of consultation with management and legal counsel.

The lower the level of access gained by the intruder and the more the system administrator knows about the hacker's actions, the less risk there is in patching the vulnerability. The less known about the intruder's actions, the more highly recommended it is to reinstall all software on the host.

6. Conclusion

Most systems administrators are overwhelmed by the myriad of patches and vulnerabilities that are released every month. Consequently, many systems are not patched, are patched using an ad hoc methodology, or are not patched in a timely manner. The result is that the software of many organizations is vulnerable to penetration by hackers.

To mitigate this problem, this document provides a systematic patching model along with background information on how to obtain, apply, and test patches. It is our hope that you will find this model useful and implement or adapt it to your organization. We believe that creating a patching process like the one proposed will result in the timely application of the most critical patches.

Appendix A: Glossary

This document uses the following terms extensively. For this document, their definition is as follows:

Application—Any data entry, update, query, or report program that processes data for the user. It includes not only the generic productivity software (spreadsheets, word processors, database programs, etc.) but also custom and packaged programs for payroll, billing, inventory, and other accounting purposes.

Host—A computer that acts as a source of information or signals. The term can refer to almost any kind of computer, from a centralized mainframe that is a host to its terminals, to a server that is host to its clients, to a desktop personal computer (PC) that is host to its peripherals. In network architectures, a client station (user's machine) is also considered a host because it is a source of information to the network in contrast to a device such as a router or switch that directs traffic.

Hotfix—Microsoft's term for a bug fix, which is accomplished by replacing one or more existing files in the operating system or application with revised versions.

Network Administrator—A person who manages a local area communications network (LAN) within an organization. Responsibilities include network security, installing new applications, distributing software upgrades, monitoring daily activity, enforcing licensing agreements, developing a storage management program, and providing for routine backups.

Operating System—The master control program that runs the computer. The first program loaded when the computer is turned on, its main part, the "kernel," resides in memory at all times. The operating system sets the standards for all application programs that run in the computer. The applications "talk to" the operating system for all user interface and file management operations.

Patch—A patch (sometimes called a "fix") is a quick repair job for a piece of programming. A patch is the immediate solution that is provided to users; it can sometimes be downloaded from the software maker's website. The patch is not necessarily the best solution for the problem, and the product developers often find a better solution to provide when they package the product for its next release. A patch is usually developed and distributed as a replacement for or an insertion in compiled code (that is, in a binary file or object module). In larger operating systems, a special program is provided to manage and track the installation of patches.

Service Pack—A software patch that is applied to an installed application. It is either downloaded from the vendor's website or distributed via Compact Disk-Read Only Memory (CD-ROM). When executed, it modifies the application in place.

System—See Host.

System Administrator—A person who manages a multi-user computer system. Responsibilities are similar to that of a network administrator. A system administrator would perform systems programmer activities with regard to the operating system and other network control programs.

Vulnerability—A security exposure in an operating system or other system software or application software component. A variety of organizations maintain publicly accessible databases of vulnerabilities based on version number of the software. Each vulnerability can potentially compromise the system or network if exploited.

Appendix B: Patching Resources

Apple

Topic	Website
Apple Support	http://www.apple.com/support/
Apple Operating System and Application Patches	http://www.info.apple.com/support/downloads.html

Cisco

Topic	Website
Cisco Security Advisories	http://www.cisco.com/warp/public/707/advisory.html
Cisco Technical Assistance Center (TAC)	http://www.cisco.com/public/support/tac/home.shtml
Cisco Internetworking Operating System (IOS) Reference Guide	http://www.cisco.com/warp/public/620/1.html
Cisco Security Tips	http://www.cisco.com/warp/public/707/
Improving Security on Cisco Routers	http://www.cisco.com/warp/public/707/21.html
Cisco Product Security Incident Response	http://www.cisco.com/warp/public/707/sec_incident_response.shtml
Troubleshooting Security	http://www.cisco.com/warp/public/112/chapter24.htm
Subscription to the Cisco TAC Newsletter	http://www.cisco.com/public/news_training/itsnews/subscribe.shtml
Cisco Tool Index	http://www.cisco.com/public/support/tac/t_index.shtml

Microsoft Windows Operating System

Topic	Website
Microsoft	http://www.microsoft.com
Strategic Technology Protection Program	http://www.microsoft.com/security
Windows Critical Update Notification	http://support.microsoft.com/support/kb/articles/Q224/4/20.ASP
Microsoft Security Notification Service	http://www.microsoft.com/technet/security/bulletin/notify.asp
Qchain	http://support.microsoft.com/support/kb/articles/Q296/8/61.asp
Qfecheck	http://support.microsoft.com/support/kb/articles/Q282/7/84.ASP
Windows Update	http://windowsupdate.microsoft.com
Corporate Windows Update	http://corporate.windowsupdate.microsoft.com
Office Update	http://office.microsoft.com/downloads
Microsoft Downloads	http://www.microsoft.com/downloads
Microsoft Product Support Services	http://support.microsoft.com/directory
Microsoft Personal Security Advisor	http://www.microsoft.com/technet/mpsa/start.asp
HFNetChk	http://support.microsoft.com/support/kb/articles/q303/2/15.asp?id=303215&sd=tech
Maximized Software Hotfix Reporter	http://www.maximized.com/freeware/hotfixreporter/

Popular Web Client and Mail Client Applications

Topic	Website
Eudora Security Advisories	http://www.eudora.com/security.html
Internet Explorer	http://www.microsoft.com/windows/ie/downloads/critical/
Mozilla Patches and Updates	http://www.mozilla.org/binaries.html
Netscape	http://home.netscape.com/smartupdate/index.html
Opera	http://www.opera.com/support/
Outlook Patches	http://office.microsoft.com/Downloads/default.aspx

Popular End-User Applications

Topic	Website
Adobe Patches	http://www.adobe.com/support/downloads/main.html
Corel (WordPerfect Office, CorelDraw, etc.) Patches	http://www.corel.com/support/downloads/index.htm
Corel Knowledge Base	http://venus.corel.com/kbsearch/
Lotus Notes Patches	http://www.support.lotus.com/
Lotus Smart Suite Patches	http://www.support.lotus.com/
Microsoft Office Support	http://www.microsoft.com/office/support/default.asp
Microsoft Office Patches	http://www.microsoft.com/office/downloads/default.asp

Popular Server Applications

Topic	Website
Apache	http://www.apache.org/dist/
iPlanet (knowledge base)	http://www.iplanet.com/support/support_services_10_0.html
iPlanet (patches)	http://www.iplanet.com/downloads/patches/
Lotus Notes Server	http://www.support.lotus.com/
Lotus Domino	http://www.support.lotus.com/
Oracle	http://otn.oracle.com/software/content.html
Microsoft Exchange	http://www.microsoft.com/exchange/downloads/
Microsoft Internet Information Server (patches)	http://www.microsoft.com/windows2000/downloads/critical/
Novell (patches)	http://support.novell.com/filefinder/
Novell (online support)	http://support.novell.com/online.html

Linux/Unix Distribution Websites

Operating System	Website
Armed Linux	http://www.armed.net/
Astaro Security Linux	http://www.astaro.com/
Beehive Linux	http://www.beehive.nu/
BestLinux	http://www.bestlinux.net/
BlueCat Linux	http://www.lynxworks.com/
Caldera OpenLinux	http://www.calderasystems.com/
ChainSaw Linux	http://www.chainsawlinux.com/
Conectiva Linux	http://en.conectiva.com/
Corel Linux	http://linux.corel.com/
Coyote Linux	http://www.vortech.net/coyote/
CRUX	http://crux.nu/
Debian GNU/Linux	http://www.debian.org/
Demo Linux	http://demolinux.org/en/qui/qui.html
Dettu[Xx] Linux	http://dettus.dyndns.org/dettuxx
Devil-Linux	http://www.devil-linux.org/
DLX Linux	http://www.wu-wien.ac.at/usr/h93/h9301726/dlx.html
DragonLinux	http://www.dragonlinux.net/
easyLinux	http://www.eit.de/
Elfstone Linux	http://www.elflinux.com/linux.html
EnGarde Secure Linux	http://www.linux.org/dist/
FlightLinux	http://flightlinux.gsfc.nasa.gov/
FreeBSD	http://www.freebsd.org/
Freesco	http://www.freesco.org/
GCL – Grey Cat Linux	http://www.greycatlinux.myWeb.nl/
Gentoo Linux	http://www.gentoo.org/
Gentus Linux	http://www.gentus.com/
hal91 Floppy Linux	http://jspirop.tripod.com/linux/hal91.htm
Hard Hat Linux	http://www.mvista.com/

Operating System	Website
Icepack Linux	http://www.icepack-linux.com/
Immunix OS	http://www.wirex.com/
KRUD	http://www.tummy.com/krud/
KYZO	http://www.kyzo.com/
L13Plus	http://l13plus.mp3italia.com
Linux Antarctica	http://www.linuxantarctica.com/
Linux by LibraNet	http://www.libranet.com/
Linux Mandrake	http://www.linux-mandrake.com/
LinuxWare	http://www.trans-am.com/index1.htm
LoopLinux	http://www.tux.org/pub/people/kent-robotti/looplinux/index.html
Lycoris Linux	http://www.lycoris.com/
MaxOS	http://www.maxos.com/
Midori Linux	http://midori.transmeta.com/
MkLinux	http://www.mklinux.org/
Monkey Linux	http://www.spsselib.hiedu.cz/monkey/
MuLinux	http://sunsite.auc.dk/mulinux/
OpenBSD	http://www.openbsd.org/
Peanut Linux	http://metalab.unc.edu/peanut/
Phat Linux	http://www.phatlinux.com/
Pocket Linux	http://www.pocket-lnx.org/
Progeny Debian	http://www.progeny.com/
Pygmy Linux	http://pygmy.penguin.cz/
RedHat Linux	http://www.redhat.com/
Rock Linux	http://www.rocklinux.org/
RT-Linux	http://www.fsmlabs.com/community/
Slackware Linux	http://www.slackware.com/
Spinix	http://www.ibiblio.org/spinix
Stampede Linux	http://www.stampede.org/
SuSE Linux	http://www.suse.com/
ThinLinux	http://www.thinlinux.org/

Operating System	Website
TINY Linux	http://tiny.seul.org/
Tomsrtbt	http://www.toms.net/rb/
Trustix Secure Linux	http://www.trustix.net/
TurboLinux	http://www.turbolinux.com/
White Dwarf Linux	http://www.emjembedded.com/linux/dimmpc.html
WinLinux 2000	http://www.winlinux.net/
Yellow Dog Linux	http://www.yellowdoglinux.com/
Yggdrasil Linux	http://www.yggdrasil.com/
ZipHam	http://zipham.free.fr/

Popular Linux/Unix Distribution Download/Update/Security Websites

Operating System	Website
Debian	
Debian Security Information	http://www.debian.org/security/
Debian Distribution	http://www.debian.org/distrib/
Debian Support	http://www.debian.org/support
Debian Mailing Lists	http://www.debian.org/MailingLists/
Mandrake	
Mandrake Security Information	http://www.linux-mandrake.com/en/security/
Mandrake Distribution	http://www.linux-mandrake.com/en/ftp.php3
Mandrake Support	http://www.linux-mandrake.com/en/fdoc.php3
Mandrake Mailing Lists	http://www.linux-mandrake.com/en/flists.php3
RedHat	
RedHat Security Information	http://www.redhat.com/support/alerts/
RedHat Distribution	http://www.redhat.com/apps/download/
RedHat Support	http://www.redhat.com/apps/support/
RedHat Mailing Lists	http://www.redhat.com/mailling-lists/
SuSE	
SuSE Security Information	http://www.suse.com/us/support/security/index.html
SuSE Distribution	http://www.suse.com/us/support/download/suse_linux/index.html
SuSE Updates	http://www.suse.com/us/support/download/updates/index.html
SuSE Support	http://sdb.suse.de/en/sdb/html/
SuSE Mailing Lists	http://www.suse.com/de/support/maillinglists/index.html

Operating System	Website
Slackware	
Slackware Distribution	http://www.slackware.com/getslack/
Slackware Support	http://www.slackware.com/support/
Slackware Mailing Lists	http://www.slackware.com/lists/
Caldera	
Caldera OpenLinux Security Information	http://www.caldera.com/support/security/
Caldera OpenLinux Distribution	http://www.caldera.com/download/
Caldera OpenLinux Support	http://www.caldera.com/support/
FreeBSD	
FreeBSD Security Information	http://www.freebsd.org/security/index.html
FreeBSD Distribution	http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/mirrors.html
OpenBSD	
OpenBSD Security Information	http://www.openbsd.org/security.html
OpenBSD Patches	http://www.openbsd.org/errata.html
OpenBSD Support	http://www.openbsd.org/docum.html
TrustedBSD	
TrustedBSD Distribution	http://www.trustedbsd.org/downloads/
TrustedBSD Support	http://www.trustedbsd.org/documentation/
Solaris	
Solaris Security Information	http://www.sun.com/security/
Solaris Distribution	http://www.sun.com/software/solaris/get.html
Solaris Live Upgrade	http://www.sun.com/solaris/liveupgrade/
Solaris Support	http://www.sun.com/software/solaris/services.html

Virus Software Download/Update/Security Centers

Topic	Website
McAfee Antivirus	
McAfee Antivirus	http://www.mcafee.com/antivirus/
McAfee Antivirus Updates	http://download.mcafee.com/updates/updates.asp
McAfee Antivirus Upgrades and Patches	http://download.mcafee.com/updates/upgrade_patches.asp
McAfee Evaluation Download	http://download.mcafee.com/eval/
McAfee Mailing List	http://dispatch.mcafee.com/
McAfee Hoax Page	http://vil.mcafee.com/hoax.asp?
Symantec Norton Antivirus	
Symantec Norton Antivirus	http://www.sarc.com/avcenter/
Norton Antivirus Definitions	http://www.symantec.com/avcenter/defs.download.html
Removal Tools	http://www.sarc.com/avcenter/tools.list.html
Updates/Downloads	http://www.symantec.com/techsupp/files.html
Symantec Product Security Advisories	http://www.sarc.com/avcenter/security/SymantecAdvisories.html
Symantec Online Virus and Security Check	http://www.symantec.com/securitycheck/
Mailing List/News Bulletins	http://www.symantec.com/techsupp/bulletin/index.html
Virus Hoax Page	http://www.sarc.com/avcenter/hoax.html
Panda Antivirus	
Panda Antivirus Home	http://www.pandasecurity.com/platinuminfo.htm
Panda Antivirus Global	http://www.pandasecurity.com/gviinfo.htm
Sophos Antivirus	
Sophos Antivirus	http://www.sophos.com/products/antivirus/
Sophos Evaluation	http://www.sophos.com/downloads/products/
Sophos Virus Definition Updates	http://www.sophos.com/downloads/ide/
Sophos Mailing List	http://www.sophos.com/virusinfo/notifications
Sophos Supports	http://www.sophos.com/support/

Topic	Website
Central Command	
Central Command	http://www.centralcommand.com/products.html
Central Command Support	http://support.centralcommand.com/cgi-bin/command.cfg/php/enduser/home.php
Central Command Updates	http://www.centralcommand.com/update.html
F-Secure Antivirus	
F-Secure Antivirus	http://www.fsecure.com/products/antivirus/
F-Secure Virus Info	http://www.fsecure.com/virus-info/
Miscellaneous Antivirus Resources	
Virus Bulletins	http://www.virusbtn.com/
Antivirus Product Developers List	http://www.virusbtn.com/AVLinks/
Virus Bulletins Hoax Page	http://www.virusbtn.com/Hoax/

Appendix C: Identifying Vulnerabilities with ICAT

The ICAT Metabase is a searchable index of computer vulnerabilities. ICAT links users to a variety of publicly available vulnerability databases and patch sites, thus enabling system administrators to identify and correct vulnerabilities that exist on their systems. ICAT is not itself a vulnerability database, but is instead a searchable index leading one to vulnerability resources and patch information. ICAT allows one to search with a fine granularity, a feature unavailable with most dedicated vulnerability databases, by characterizing each vulnerability with more than 40 attributes (including software name, version number, impact, and exploitable range). ICAT indexes the information available in CERT advisories, ISS X-Force, Security Focus, NT Bugtraq, Bugtraq, and a variety of vendor security and patch bulletins. ICAT is maintained by the National Institute of Standard and Technology (NIST).

Some uses for ICAT are as follows:

- Systems administrators and computer security officers use ICAT to identify the known vulnerabilities (and patch information) associated with the software on critical systems.
- ICAT can be used in forensics activities to determine the set of possible vulnerabilities that a hacker might have used to penetrate a system.
- Computer security researchers use ICAT to identify sets of vulnerabilities that have particular characteristics of interest.
- Auditors can use ICAT to print out vulnerabilities known to exist within audited systems. Then, the auditor can be asked whether or not a patch has been installed to fix each vulnerability.

ICAT uses, and is completely based on, the common vulnerabilities and exposures (CVE) naming standard, an industry standard naming scheme for computer vulnerabilities and exposures. Information on the CVE can be found at <http://www.cve.mitre.org/>.

Accessing the ICAT Metabase

ICAT is accessible to all users with a web browser and access to the Internet. It is located at <http://icat.nist.gov> (see Figure C.1).

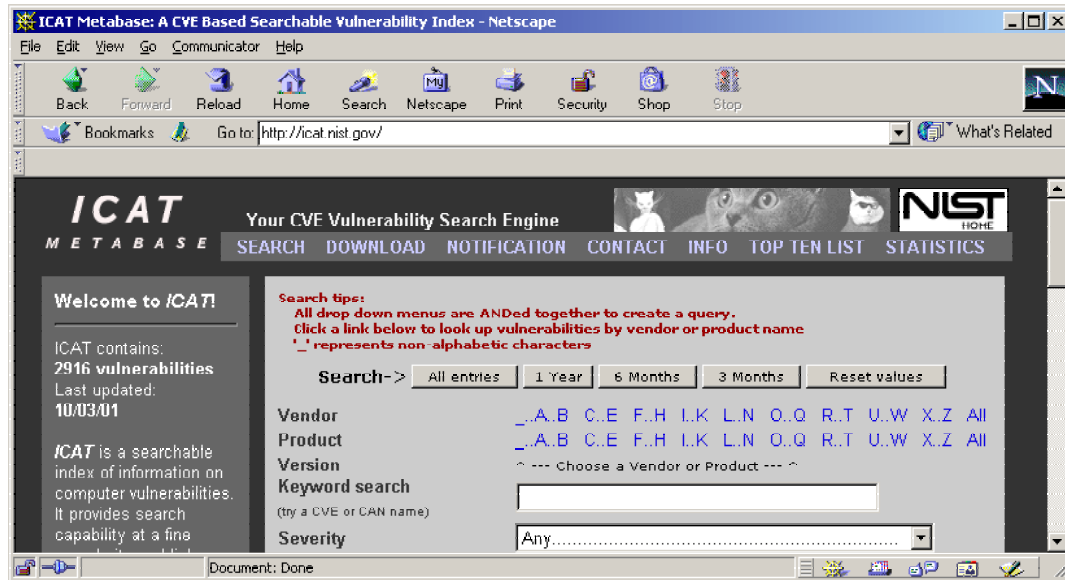


Figure C.1: Accessing ICAT Metabase

Navigating the ICAT Metabase Website

The ICAT Metabase homepage provides a variety of information and links that help the user to navigate the ICAT Metabase.

The menu bar at the top of the screen allows users to select an area in which they want to navigate (see Figure C.2).

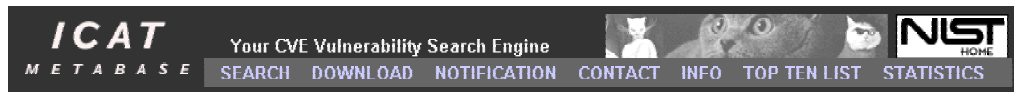


Figure C.2: ICAT Metabase Navigation Bar

- **Search**—Returns the user to the homepage/search page. From there, the user can conduct searches of the ICAT Metabase (see searching the ICAT Metabase, below).
- **Download**—Opens the *ICAT Download and Product Integration Section*, which allows users to download standalone versions of the ICAT database. This is useful for users who wish to have access to the database when not connected to the Internet or for developers who wish to integrate ICAT information into their application.
- **Notification**—Opens the *ICAT-Based Vulnerability Notification Systems* page. This page provides information about automated notification applications that employ the ICAT Metabase. These applications notify users of new entries to the ICAT Metabase that are related to the software employed by the user. Links on this page allow users to download these applications from their creators' websites.



Figure C.3: ICAT Sidebar

- **Contact**—Opens the *ICAT Contact Information* page. This page contains contact information for the ICAT staff. Users are encouraged to provide information regarding how they use the ICAT, to download ICAT advertising banners, and to provide a link to the ICAT Metabase from their website.
- **Info**—Opens a new browser window and then opens the *ICAT Metabase Documentation* page. This page provides answers to frequently asked questions (FAQ) about the ICAT Metabase.
- **Top Ten List**—Opens the *ICAT Top Ten List* page. This page contains a table of the ten most “popular” vulnerabilities, as defined by the number of requests for information received for a particular vulnerability through the ICAT Metabase. To maintain the timeliness of this information, only vulnerabilities published within the last year are included in the list.
- **Statistics**—Opens the *ICAT Vulnerability Statistics* page. This page contains statistics on the characteristics of the vulnerabilities contained in the ICAT Metabase.

ICAT Metabase Sidebar

The sidebar can be found on the ICAT homepage and most other pages on the site (see Figure C.3).

- The top of the sidebar presents the number of vulnerabilities contained in the metabase, the date of the last update, and a brief overview of the ICAT.
- The sidebar next contains an item that allows the user to register for the ICAT mailing list. Important announcements about ICAT are sent to subscribers. The number of announcements sent to subscribers is low (only a few e-mails per year).
- The lower half of the sidebar provides links to the websites of organizations that support the ICAT. These sites may be of use or interest to ICAT users. Some of these links are to sites external to NIST. (When a user attempts to access an external site via a hyperlink, the NIST website presents notification of the pending exit from the NIST site before allowing access to the external site. To speed loading of the external site, the user should click on the hyperlink provided after reading the disclaimer.)

- The side bar also provides links to press articles concerning ICAT.

Common Vulnerabilities and Exposures

The CVE is an industry standard naming scheme for information system and network vulnerabilities and exposures that the ICAT employs to index its vulnerability information. Vulnerabilities are named either CVE-xxxx-yyyy or CAN-xxxx-yyyy. The CVE prefix is used for vulnerabilities that have been reviewed by the CVE standard

advisory committee. The CAN prefix is used for candidates under review by the committee. The candidates have been filtered by MITRE to ensure some degree of accuracy, but there is no guarantee that a CAN entry is a unique or real vulnerability. The xxxx part of each entry represents the year in which the vulnerability entered the CVE process. The yyyy part of each entry is a unique number assigned to entries submitted to the CVE committee that year. When a CAN entry is approved by the CVE committee, the prefix is changed from CAN to CVE, while leaving the number the same.

Searching the ICAT Metabase

The ICAT Metabase has a variety of powerful search features. The search engine is available from the ICAT Metabase homepage (see Figure C.4). Searches can be conducted by the date when the vulnerability was published, vendor name, product name, version (available only when a product is selected), keyword search, and severity level. ICAT also provides filters that allow users to limit their search results to particular sources, exploit range, vulnerability consequence, vulnerability type, operating system type, exposed component type, entry type, and those vulnerabilities published after a particular date.

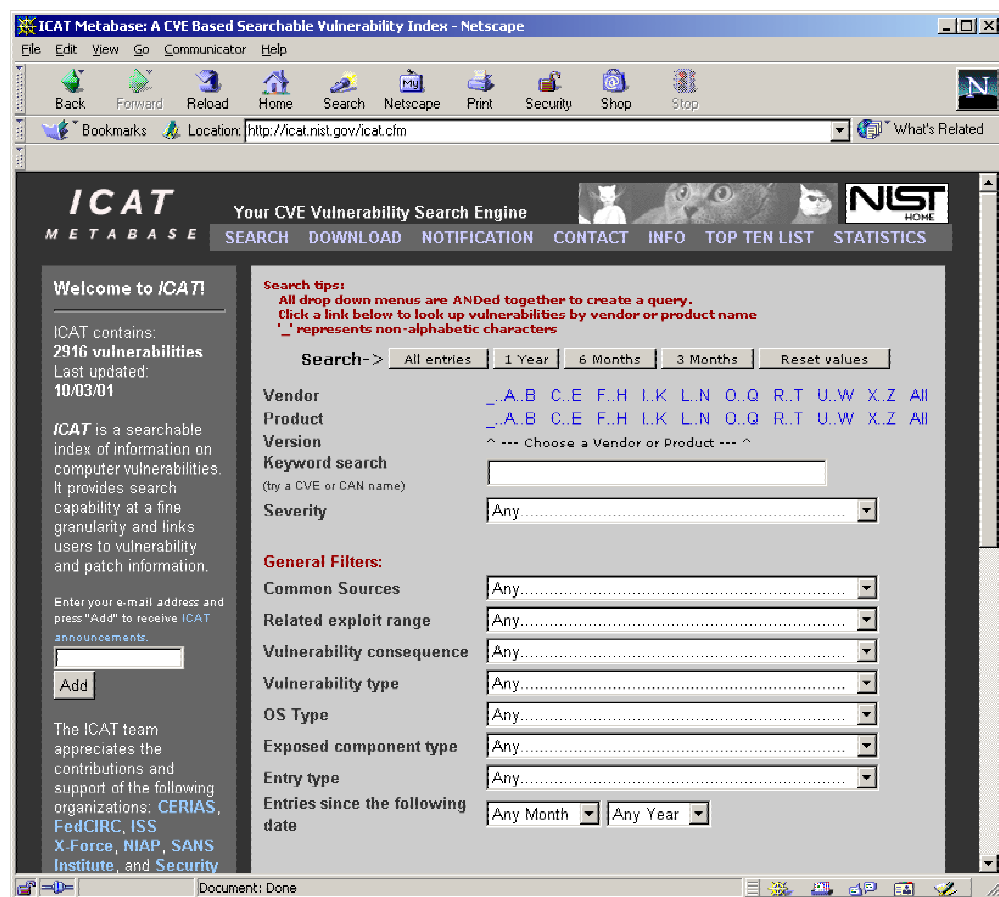


Figure C.4: ICAT Metabase Navigation Bar

Defining Search Parameters

Before conducting a search, one must define the search parameters. A number of methods of limiting searches are available in the ICAT Metabase. The “Reset values” button resets all of the search parameters and filters to the default values (see Figures C.4 and C.5). With limited exception noted below, users can use as few or as many of the search parameters and filters as required.

Date of Publication—The buttons across the top of the search parameters section (Figure C.5) allow users to search for vulnerabilities by date of publication. The results will be ordered from most recent to oldest within the parameters set.

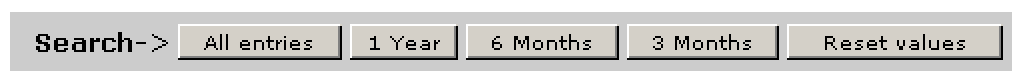


Figure C.5: Search by Date of Publication

Vendor—The letters on the vendor line allow users to search for vulnerabilities by the name of the vendor. Selecting a letter on the vendor line will open a drop-down menu with all vendor names starting with that letter (see Figure C.6). A user can select the vendor of choice by scrolling down the list and clicking on the vendor name. The user can then start the search by clicking on the appropriate search button (see Figure C.5) or narrow the search further by selecting a product from the vendor’s product list (see below). (As with all drop-down boxes on the ICAT search page, the user may select only one vendor at a time.)

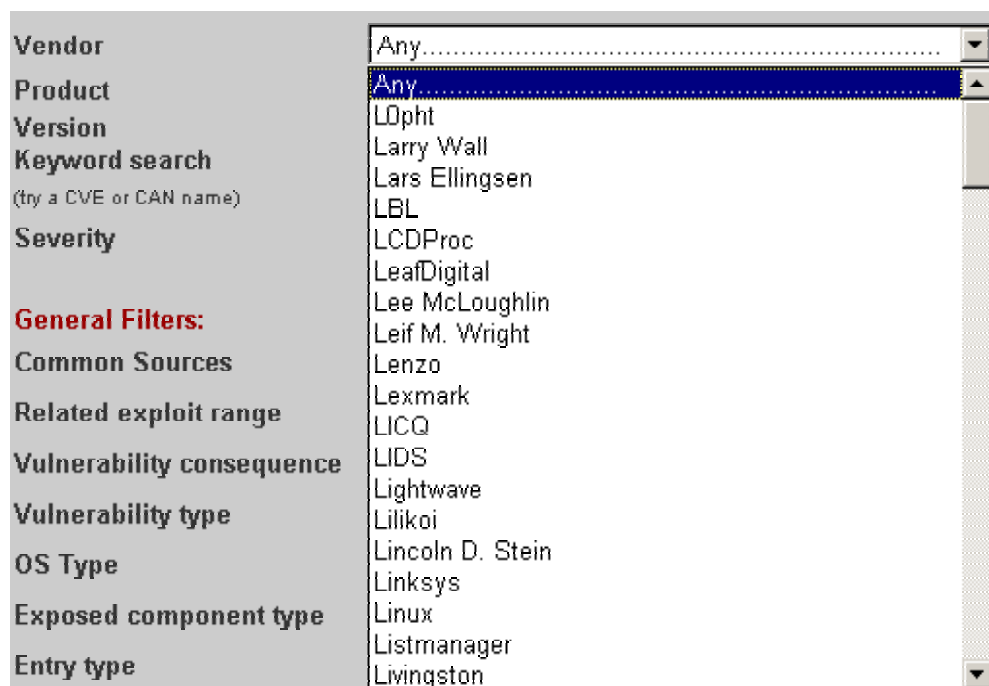


Figure C.6: Search by Vendor

Product—If a vendor has not already been selected, the user can click on the letter that begins the name of the product being searched. If the user has already selected a vendor

(see above), a drop-down box will appear with a list of vendor’s products (see Figure C.7). The user can select a product by scrolling down the list and clicking on the appropriate choice. (As with all drop-down boxes on the ICAT search page, the user may select only one product at a time.)

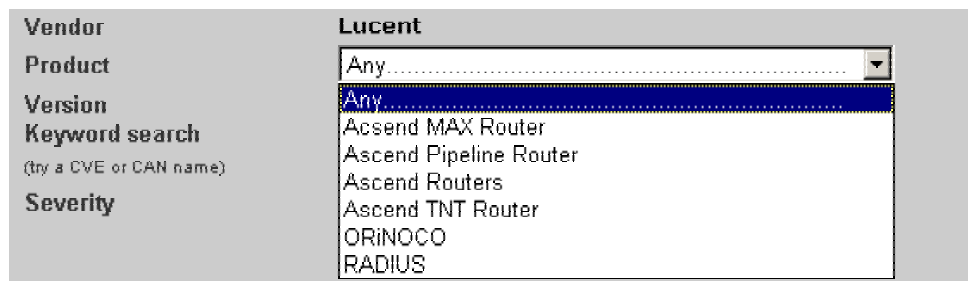


Figure C.7: Search by Product

Version—This option is available only if the user has selected a product. It allows a user to select a particular version of a product (see Figure C.8). Given the large number of versions available for certain products, use of this option may cause the metabase to fail to return all possible vulnerabilities. (As with all drop-down boxes on the ICAT search page, the user may select only one version at a time.)

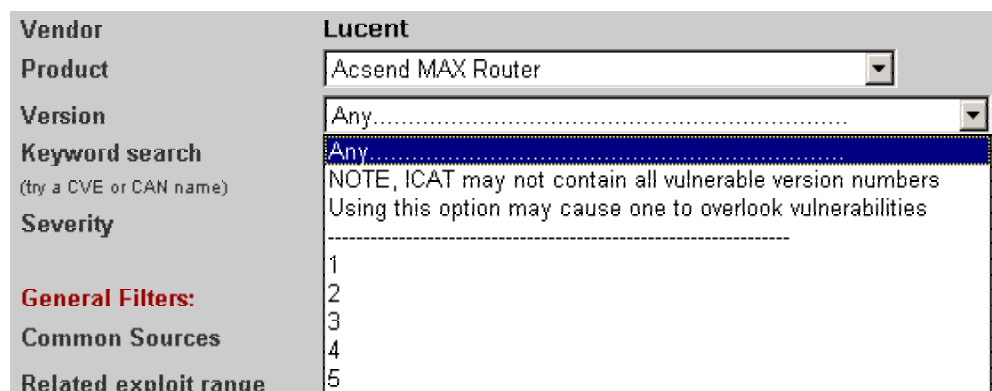


Figure C.8: Search by Product Version

Keyword—This keyword option allows the user to narrow the search using keywords (see Figure C.9).

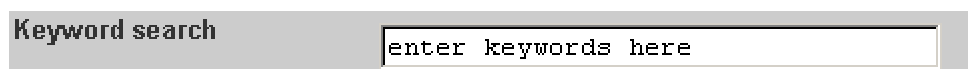


Figure C.9: Search by Keyword

Severity—This option allows the user to narrow the search by severity level (see Figure C.10). (As with all drop-down boxes on the ICAT search page, the user may select only one severity level at a time.)

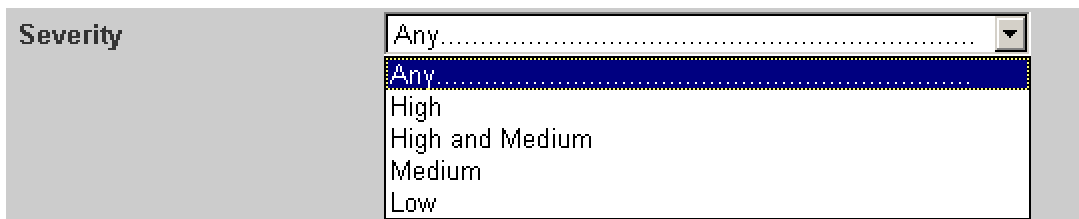


Figure C.10: Search by Keyword

Defining Search Filters

Optionally, the user can select filters to further narrow a search. The “Reset values” button resets all of the search parameters and filters to the default values (see Figures C.4 and C.5). With limited exceptions noted below, users can use as few or as many of the search parameters and filters as they require.

Common Sources—This filter allows a user to select a specific source of vulnerability (see Figure C.11). As with all drop-down boxes on the ICAT search page, the user may select only one source at a time.)

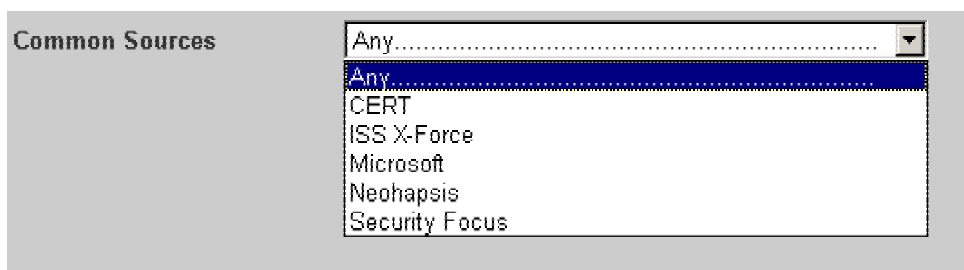


Figure C.11: Common Source Search Filter

Related Exploit Range—This filter allows a user to limit a search to local vulnerabilities (those that require local access for exploitation) or to remote vulnerabilities (those that can be exploited without local access to a host). This filter is shown in Figure C.12. (As with all drop-down boxes on the ICAT search page, the user may select only one range at a time).

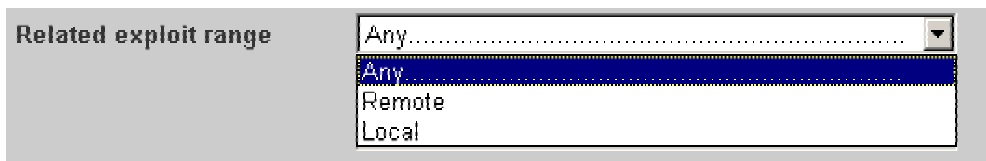


Figure C.12: Exploit Range Search Filter

Vulnerability Consequence—This option filters the results based on the consequence of the vulnerability (e.g., root-level access, availability). Figure C.13 shows the options for this filter. As with all drop-down boxes on the ICAT search page, the user may select only one consequence at a time.)

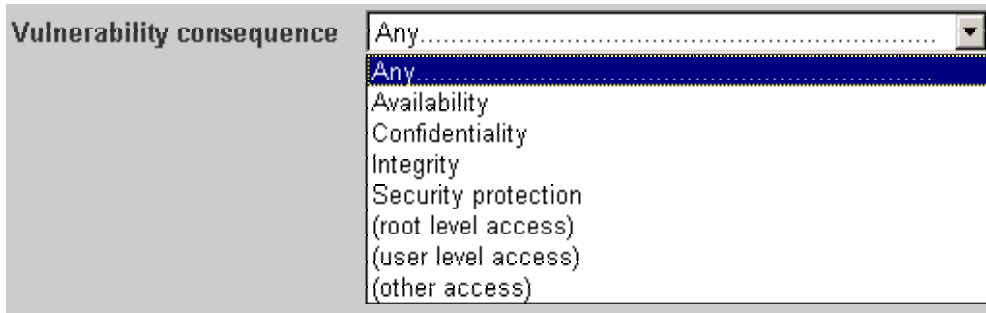


Figure C.13: Vulnerability Consequence Search Filter

Vulnerability Type—This choice filters the results based on the root cause of the vulnerability (e.g., root-level access, availability). Figure C.14 shows the options for this filter. (As with all drop-down boxes on the ICAT search page, the user may select only one vulnerability type at a time.)

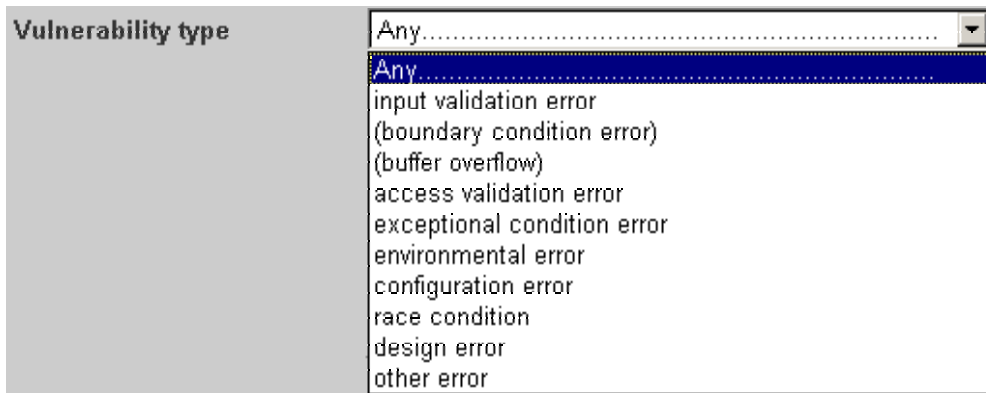


Figure C.14: Vulnerability Type Search Filter

Exposed Component Type—This filter provides an ability to filter results by the type of component affected (e.g., user application, communications protocol, hardware). Figure C.15 shows the options for this filter. (As with all drop-down boxes on the ICAT search page, the user may select only one component at a time.)

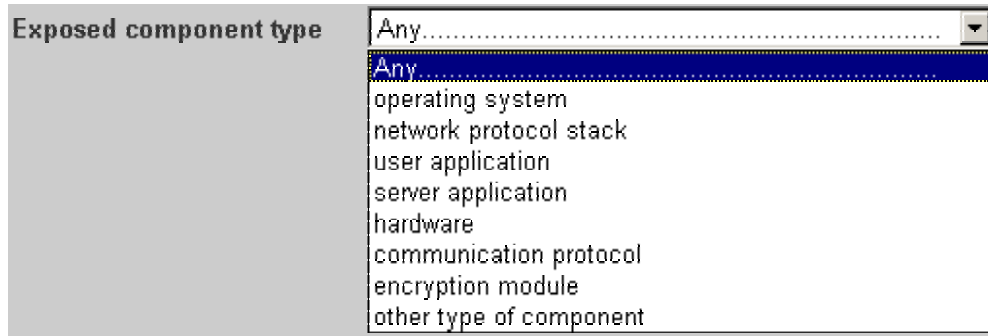


Figure C.15: Exposed Component Type Search Filter

Entry Type—This option provides an ability to filter results by the type of entry. The ICAT contains two types of entries. The primary entries are those that have been accepted by the CVE advisory committee. These entries start with the prefix CVE. The other types of entries are those that are still under review by the CVE committee; these begin with the prefix CAN. Figure C.16 shows the options for this filter. (As with all drop-down boxes on the ICAT search page, the user may select only one type of entry at a time.)

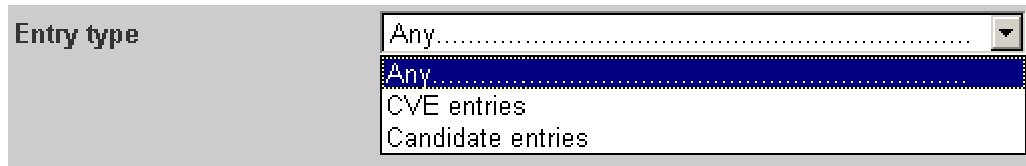


Figure C.16: Entry Type Search Filter

Entries Since the Following Date—This filter provides an ability to select only vulnerabilities published after a particular month and year (back to January 1995) as shown in Figure C.17.



Figure C.17: Entries Since the Following Date Filter

ICAT Metabase Search Result

Once a user has specified a search parameter and clicked on the appropriate search button, the ICAT will provide a list of all results matching the search criteria (see Figure C.18).

There are **2916** matching records. Displaying matches **1** through **20**.

Next 20 Matches

CAN-2001-0670	
<i>Summary:</i>	Buffer overflow in BSD line printer daemon (in.lpd or lpd) in various BSD-based operating systems allows remote attackers to execute arbitrary code via an incomplete print job followed by a request to display the printer queue.
<i>Published Before:</i>	10/3/2001
<i>Severity:</i>	High
CAN-2001-0710	
<i>Summary:</i>	NetBSD 1.5 and earlier and FreeBSD 4.3 and earlier allows a remote attacker to cause a denial of service by sending a large number of IP fragements to the machine, exhausting the mbuf pool.
<i>Published Before:</i>	9/20/2001
<i>Severity:</i>	Medium
CAN-2001-0709	
<i>Summary:</i>	Microsoft IIS 4.0 and before, when installed on a FAT partition, allows a remote attacker to obtain source code of ASP files via a URL encoded with Unicode.
<i>Published Before:</i>	9/20/2001
<i>Severity:</i>	High
CAN-2001-0708	
<i>Summary:</i>	Denicomp REXECD 1.05 and earlier allows a remote attacker to cause a denial of service (crash) via a long string.
<i>Published Before:</i>	9/20/2001
<i>Severity:</i>	Medium

Figure C.18: ICAT Search Results List

To assist the user in identifying the search results that are applicable to their system(s), the ICAT Metabase provides summary information on each result. Clicking on the vulnerability name (the number in blue) will provide additional details on that vulnerability or exposure (see Figure C.19).

Vulnerability Name: This reference is to a non-NIST site. (disclaimer)	CAN-2001-0709
Published before:	9/20/2001
Summary:	Microsoft IIS 4.0 and before, when installed on a FAT partition, allows a remote attacker to obtain source code of ASP files via a URL encoded with Unicode.
Severity:	High
Vulnerability type:	Environmental Error
Exploitable Range:	Remote
Loss type:	Confidentiality
Reference 1: This reference is to a non-NIST site. (disclaimer)	Source: Security Focus Type: General and Patch Name: VIGILANTE-2001001 http://www.securityfocus.com/archive/1/192802
Reference 2: This reference is to a non-NIST site. (disclaimer)	Source: Security Focus Type: General and Patch Name: bid2909 http://www.securityfocus.com/bid/2909
Reference 3: This reference is to a non-NIST site. (disclaimer)	Source: ISS X-Force Type: General and Patch Name: iis-unicode-asp-disclosure(6742) http://xforce.iss.net/static/6742.php
Vulnerable software and versions:	Microsoft, IIS, 4.0, and previous

Figure C.19: Search Result Detail

- **Vulnerability Name**—ICAT uses the CVE standard vulnerability naming scheme to name each vulnerability. Vulnerabilities are named either CVE-xxxx-yyyy or CAN-xxxx-yyyy. The CVE prefix is used for vulnerabilities that have been reviewed by the CVE standard advisory committee. The CAN prefix is used for candidates under review by the advisory committee. The candidates have been filtered by MITRE to ensure some degree of accuracy, but there is no guarantee that a CAN entry is a unique or real vulnerability. The xxxx part of each entry represents the year in which the vulnerability entered the CVE process. The yyyy part of each entry is a unique number assigned to entries submitted to the CVE committee that year. When a CAN entry is approved by the CVE committee, the prefix is changed from CAN to CVE, whereas the number remains as it was.
- **Published Before**—This field indicates when a vulnerability was discovered. With older vulnerabilities, the date used is the earliest date shown in publicly available sources. For vulnerabilities released in the year 2001 and beyond, ICAT uses the date that the vulnerability was added to ICAT.
- **Summary**—The summary item provides a one-line description of the vulnerability. This description is the same as that found in the particular vulnerability's standard CVE or CAN entry. ICAT provides only this short description of each vulnerability because ICAT is intended to be a searchable index of vulnerabilities, not a

vulnerability database. The purpose of this summary and the other vulnerability attributes is to allow users to determine whether a particular vulnerability is relevant. Users should access the references found in each ICAT entry to obtain complete descriptions of each vulnerability from a variety of outside resources (see References section below).

- **Severity**—ICAT provides a severity field to enable a user to quickly judge the impact of a vulnerability. Vulnerabilities can have one of three severity levels: high severity, medium severity, or low severity. Although it is difficult to accurately assign such ratings, because different vulnerabilities will have differing levels of impact depending on the installed software base of an organization, severity labels are still useful indicators of vulnerability impact.
 - A vulnerability is high severity if—
 - It allows a remote attacker to violate the security protection of a system (i.e., gain access to a user or root account)
 - It allows a local attack that gains complete control of a system
 - It is important enough to have an associated CERT/CC advisory.
 - A vulnerability is medium severity if—
 - It does not fit the definition of either high or low severity.
 - A vulnerability is low severity if—
 - It does not typically yield valuable information or control over a system but instead gives the attacker knowledge that may help them find and exploit other vulnerabilities
 - It is inconsequential for most organizations.
- **Exploitable Range**—A vulnerability can enable either a local and/or remote attack.
 - Local—Attacks that must be launched directly on the system that is being attacked. The attacker must have had some previous access to the system in order to launch an attack locally. (Note: ICAT still defines an attack as local if an attacker legally telnets to a host and then initiates an attack on that host. The attack is considered local because the attacker did not attack the telnet server itself but a component visible only to logged-in users.)
 - Remote—Attacks that are launched across a network against the system without the user having previous access to the system.
- **Loss Type**—Includes the traditional three types (availability, confidentiality, and integrity) and another category called “security protection.”

- **Availability**—A vulnerability is given the “availability” label if it enables an attack to directly inhibit a user (human or machine) access to a system resource. Denial-of-service attacks are availability violations by ICAT’s definition.
- **Confidentiality**—A vulnerability is given the “confidentiality” label if it enables an attack to obtain information from a system.
- **Integrity**—A vulnerability is given the “integrity” label if it enables an attack to change or modify information residing on or passing through a system.
- **Security Protection**—A vulnerability is given the “security protection” label if it enables an attack to give the attacker privileges in a system that the attacker is not allowed to have according to the access control policy of the system. The security protection label may appear alone or in three other variations: security protection (gain superuser access) when the attack gives a hacker complete control of a system, security protection (gain user access) when the attack provides a hacker partial control over a system, and security protection (other) when the attack gives the hacker some other privilege on the system.

The availability, confidentiality, and integrity attributes are included in a vulnerability description only if exercising the vulnerability directly violates these properties. For example, if a vulnerability can give an attacker increased privileges, thereby allowing him or her to violate availability, only the security protection box would be checked. However, if a single vulnerability enables two different attacks (as is typical with buffer overflow vulnerabilities), one of which violates security protection and the other of which violates availability directly, then both the Security protection and the Availability boxes would be checked.

- **Vulnerability Type**—The ICAT Metabase characterizes each vulnerability in such a way that one can understand the type of software problem that produced the vulnerability. Each vulnerability may exhibit one or more of the following characteristics:
 - **Access Validation Error**—A vulnerability is characterized as an access validation error” if a system is vulnerable because the access control mechanism is faulty. This problem lies not with the user-controllable configuration of the access control mechanism, but with the mechanism itself.
 - **Exceptional Condition Handling Error**—A vulnerability is characterized as an exceptional condition handling error if a system somehow becomes vulnerable resulting from an unexpected condition that has arisen. The handling (or mishandling) of this condition by the system causes a vulnerability.
 - **Environmental Error**—A vulnerability is characterized as an environmental error if the surroundings in which a system is installed somehow cause the system to be vulnerable. This may be attributed, for example, to an unexpected interaction between an application and the operating system or between two applications on the same host. Such a vulnerable system may be perfectly configured and provably secure in the developer’s test environment, but the

installation environment somehow violates the developer's security assumptions.

- **Configuration Error**—A vulnerability is characterized as a configuration error if user-controllable settings in the system are set such that the system is vulnerable. This type of vulnerability is not attributed to how the system was designed but to the way in which the system administrator or user configures the system. ICAT also considers a configuration error to occur when a system ships from the manufacturer with a weak default configuration.
- **Race Condition**—A vulnerability is characterized as a race condition if the non-atomicity of a security check causes the vulnerability. For example, a system verifies whether an operation is allowed by the security model and then performs the operation. However, between the time the security verification is performed and when the operation is performed, the environment changes such that the operation is no longer allowed by the security model. Attackers can take advantage of this small window of opportunity and convince systems to perform illegal operations.
- **Design Error**—A vulnerability is characterized as a design error if there are no errors in the implementation or configuration of a system, but the initial design causes a vulnerability.
- **Input Validation Error**—A vulnerability is characterized as an input validation error if the input being received by a system is not properly checked such that a vulnerability is present that can be exploited by a certain input sequence. This vulnerability type and its subcategories apply only to input that is malicious or otherwise malformed. The input validation error label may appear by itself or in two other variations: input validation error (boundary overflow) and input validation error (buffer overflow).” These two categories are discussed below:
 - **Boundary Overflow**—A vulnerability is characterized as a boundary overflow when the input being received by a system, whether human or machine-generated, causes the system to exceed an assumed boundary, thereby causing a vulnerability. For example, the system may run out of memory, disk space, or network bandwidth. Another example is that a variable might reach its maximum value and roll over to its minimum value. A third example is that the variables in an equation might be set such that a division by zero error occurs. Boundary overflow errors are a subset of the class of input validation errors. Although it could be argued that buffer overflow (discussed next) is a type of boundary overflow error, ICAT labels buffer overflows in a distinct category given their importance.
 - **Buffer Overflow**—A vulnerability is characterized as a buffer overflow if the vulnerability is caused by a system receiving input that is longer than the expected input length. If the system does not check for this condition, then the input buffer fills up and overflows the memory allocated for input. By cleverly constructing this extra input, an attacker can cause the system to crash or even to execute instructions on behalf of the attacker.

- **Exposed System Component**—The exposed system component field identifies exactly where in a system the vulnerability exists. The possible types are operating system, protocol stack, server application, non-server application, hardware, communication protocol, encryption module, and other type of component. A vulnerability may in some cases be assigned multiple types in this field.
- **Exposed System Type**—The exposed system type field identifies in what type of system the vulnerability is usually present. The possible types are server, workstation, networking/security device, and other device type.
- **References**—ICAT provides references along with each vulnerability entry. These references link an ICAT user to publicly available vulnerability databases and patch sites that contain entries about the particular vulnerability being viewed. ICAT or NIST cannot endorse, verify, or guarantee the information on those sites. For each reference, ICAT may provide the following information:
 - **Source**—The source is the name of the vulnerability database or patch site on which the vulnerability is described.
 - **Type**—The type field describes what kind of information is found in this source. A source may contain general information on the vulnerability, a patch, or a combination of both. The exact values entered into this field are of the following set: general, patch, or general and patch.
 - **Name**—The name field is the vulnerability name used by this particular source for the CVE entry. This field enables one to search ICAT using nonstandard vulnerability names from other vulnerability databases.
 - **Link**—The link field contains a hyperlink to the related vulnerability entry in the source field. ICAT links users directly to the desired vulnerability entries. This aspect of ICAT makes it a true “metabase” of information.
- **Vulnerable Software and Versions**—ICAT provides a list of vulnerable software names and version numbers for most CVE and CAN entries. This list is provided to enable users to search for vulnerabilities associated with their software versions. If software is listed on ICAT entry, the related references can be used to obtain the appropriate patches or mitigation techniques. ICAT entries typically list each vulnerable version separately. Each entry contains a vendor name, software name, and vulnerable version number. This information is obtained from various public and private sources.

Appendix D: Vulnerability Advisory Resources

Federal Vulnerability Advisory Websites¹¹

Website	URL
NIST ICAT Vulnerability Database	http://icat.nist.gov
Federal Computer Incident Response Center (FedCIRC)	http://www.fedcirc.gov/
DoD Computer Emergency Response Team (DoD-CERT)	http://afcert.kelly.af.mil
Navy Computer Incident Response Team (NAVCIRT)	http://infosec.spawar.navy.mil
Department of Energy's Computer Incident Advisory Capability (CIAC)	http://ciac.llnl.gov
National Infrastructure Protection Center (NIPC)	http://www.nipc.gov/

Private Sector Vulnerability Advisory Websites

Website	URL
The Software Engineering Institute's CERT Coordination Center (CERT)	http://www.cert.org/
Computer Security Incident Response Team-World Site (CSIRT.WS)	http://www.csirt.ws/
Internet Security Systems X-Force (ISS X-Force)	http://xforce.iss.net/
Security Focus	http://www.securityfocus.com/
CERIAS	http://www.cerias.purdue.edu/
SANS Institute	http://www.sans.org
SANS Incidents.org	http://www.incidents.org

¹¹ Note: several of these sites require users to be on a “.gov” or “.mil” domain in order to access some or all of the functionality of the website.

Appendix E: Windows Update

Windows Update is a utility provided by Microsoft in most versions of Windows (including some versions of 95 and NT and all versions of 98, ME, 2000 and XP) that allows users to scan their computers to find any updates that are available at that time from Microsoft and other participating vendors. Figures E.1 and E.2 demonstrate two different methods of accessing the Windows Update utility. It is suggested that users close all other applications before initiating the Windows Update feature.

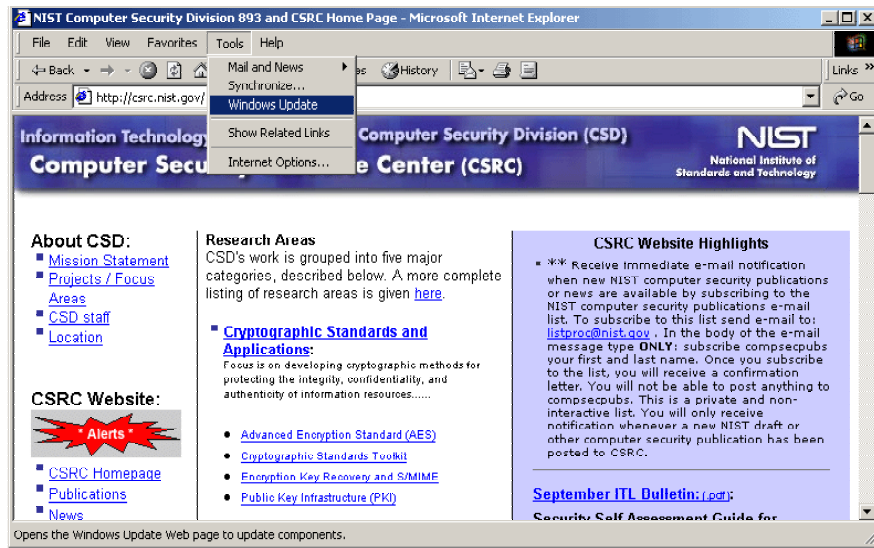


Figure E.1: Accessing Windows Update Though Internet Explorer

To access Windows update from within Internet Explorer browser, click on *Tools* and then *Windows Update* in the pull-down menu.

Alternatively, a user can access Windows Update from the Start Menu as demonstrated in Figure E.2. From the Windows desktop, click on the *Start* bar. From the menu, click on the *Windows Update* icon.

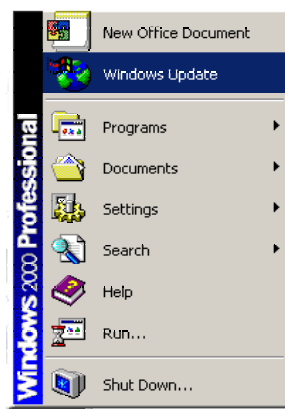


Figure E.2: Accessing Windows Update Though the Start Menu

Either option will launch Microsoft Internet Explorer (if it is not already active) and load the Microsoft Windows Update website (<http://windowsupdate.microsoft.com>). See Figure E.3 for the Windows Update homepage.

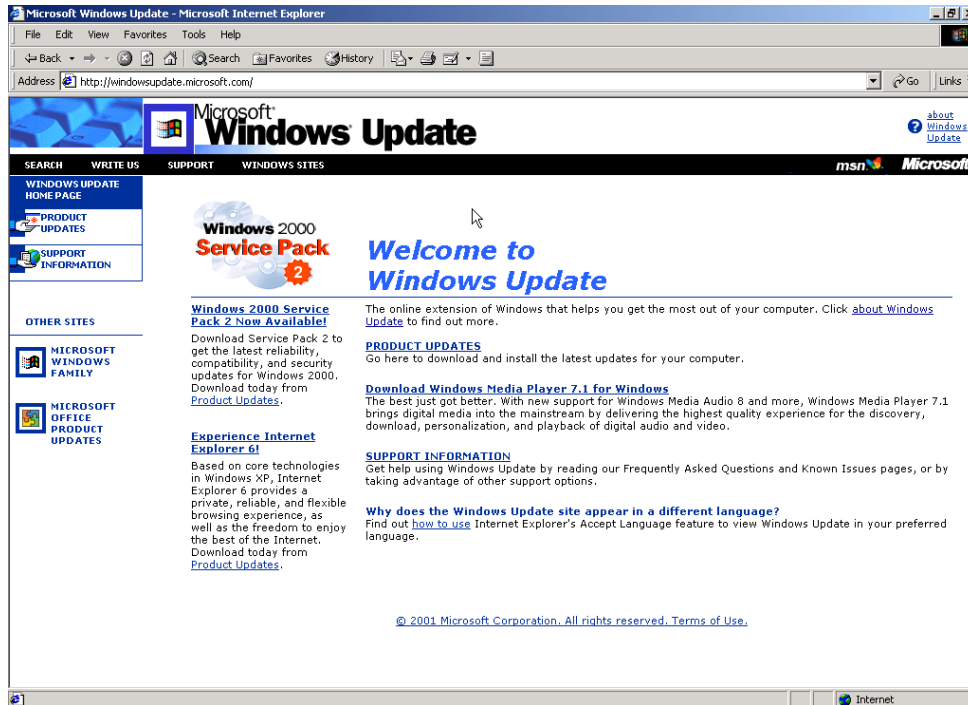


Figure E.3: Windows Update Homepage

To have the Windows Update scan a computer for updates, click on the “[PRODUCT UPDATES](#)” link. Note: This action is accomplished without sending any information to Microsoft or transmitting sensitive information on the host over the Internet. The Windows Update utility will commence its scan of the user’s computer and derive a customized product update catalog specific to that computer (see Figure E.4). Having Windows Update automatically check the system has several advantages. This check assures that users will get the most up-to-date and accurate versions of the items chosen for download from the site. Additionally, users will not waste time downloading components that are already installed.



Figure E.4: Windows Update Scan

Once Windows Update has finished scanning the user's machine, it will generate a list of recommended updates (see Figure E.5). Users can browse the list, select components, and download the selected components.

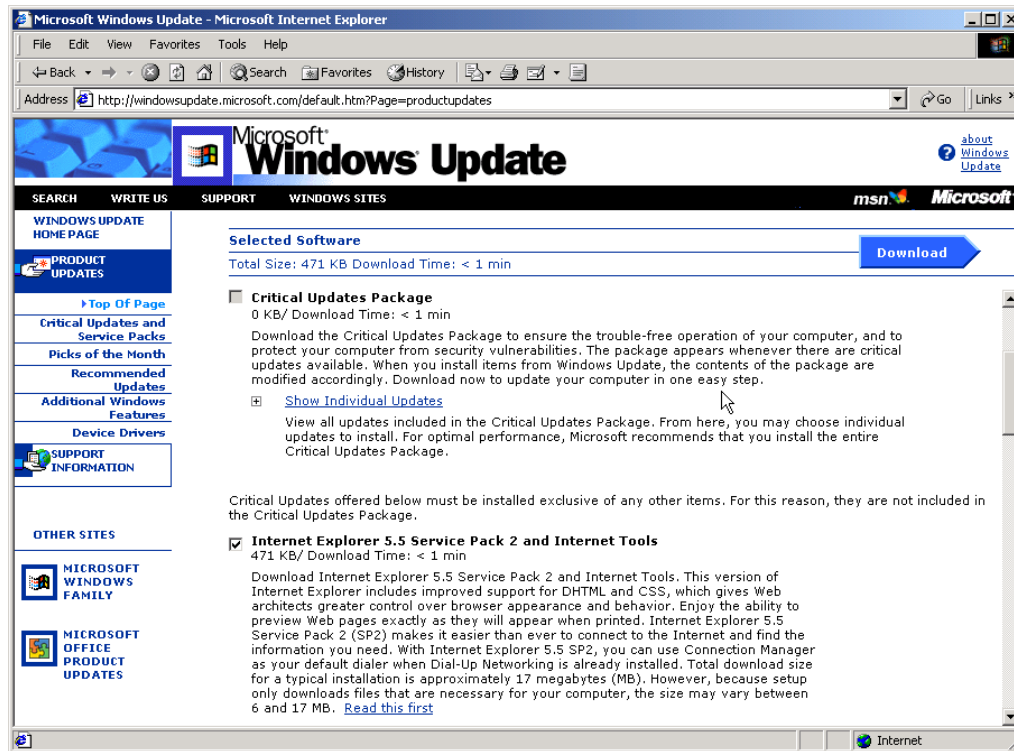


Figure E.5: Windows Update Recommended Updates

The product updates are organized into five sections:

- **Critical Updates and Service Packs**—It is suggested that users download all Critical Update Packages because these packages fix known problems (often security issues) with their specific installation.
- **Picks of the Month**—These new releases add functionality to Windows but are not required to fix a known problem.
- **Recommended Updates**—These are older releases that add functionality to Windows but are not required to fix a known problem.
- **Additional Windows Features**—These are updates to other applications that are included with Windows (e.g., Internet Explorer, Media Player).
- **Device Drivers**—Listed here will be any updated device drivers for the computer. A device driver is a program that controls a hardware device (e.g., printer, monitor, disk drive, or video card) that is attached to the computer. Note: Third parties that manufacture hardware and device drivers will not be listed unless the manufacturer has an agreement with Microsoft. A user should refer to the appropriate manufacturer's website to obtain device driver updates.

Certain updates can be downloaded only individually. If this is the case, Windows Update will provide notification as shown in Figure E.6. If this happens, the user must repeat the process delineated above.

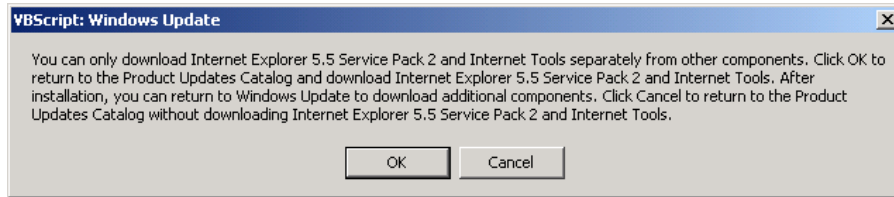


Figure E.6: Windows Update Warning

After selecting the patches to download, the Download Checklist page loads to confirm the selections (see Figure E.7). At this point, the user may choose to view the instructions, start the download and install, or return and reselect the software.

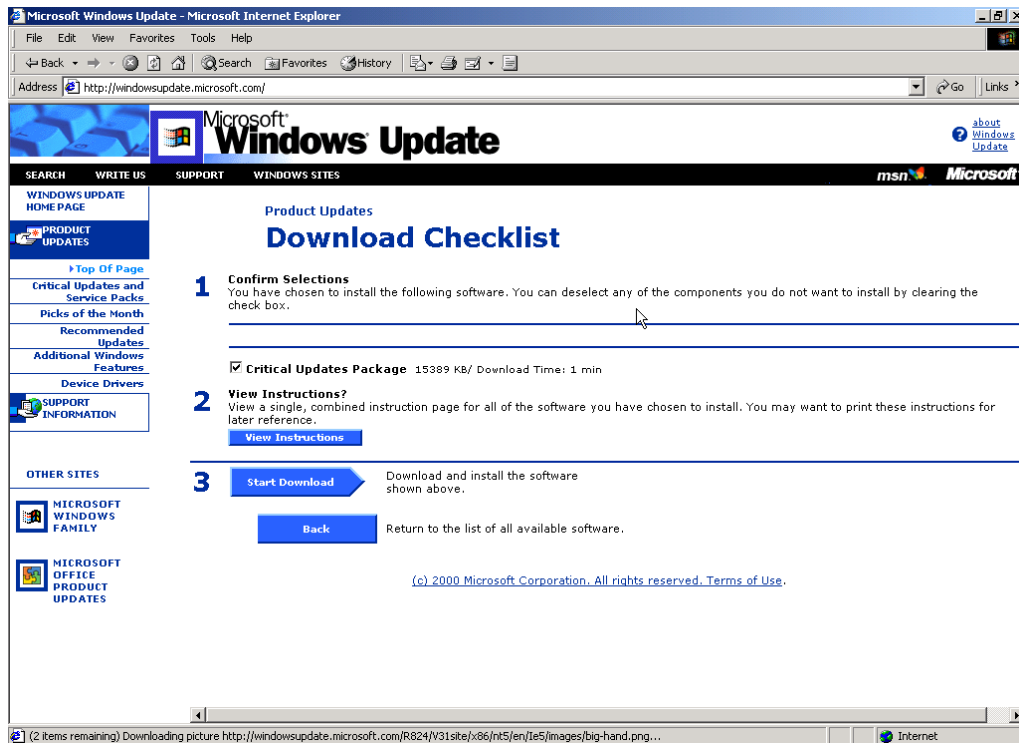


Figure E.7: Windows Update Download Checklist

After selecting “Start Download” from the Download Checklist page, an additional screen pops up to confirm the selection (see Figure E.8). At this point, the user may choose to view the instructions, license agreement, start the download and install (by clicking on the “Yes” button), return and reselect the software (by clicking on the “No” button).

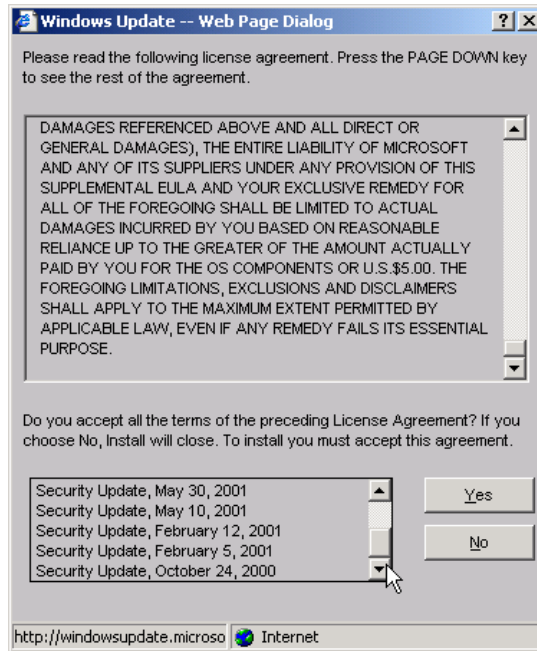


Figure E.8: Windows Update Confirmation and License Agreement

Upon acceptance of the license agreement, the selected patches and software will be downloaded (see Figure E.9). The duration of the download will depend on several factors, including the file size and connection speed.

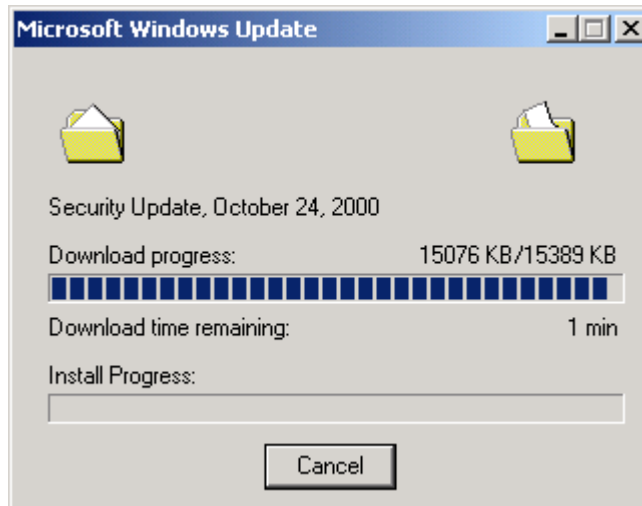


Figure E.9: Windows Update Download Status Window

After the download is complete, Microsoft Windows Update will start the install process, which may take up to several minutes to complete (see Figure E.10).

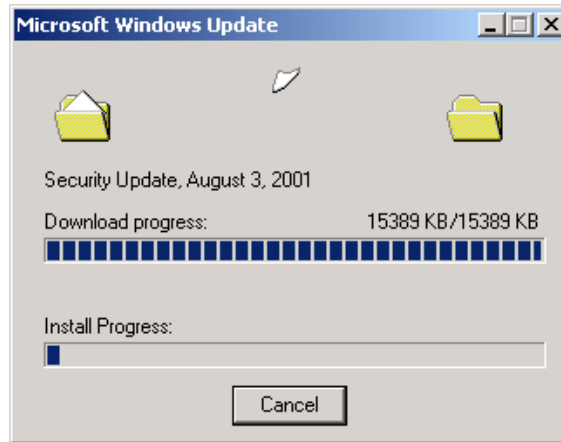


Figure E.10: Windows Update Install Status Window

Once the install is successfully completed, the browser window will confirm the success (see Figure E.11).

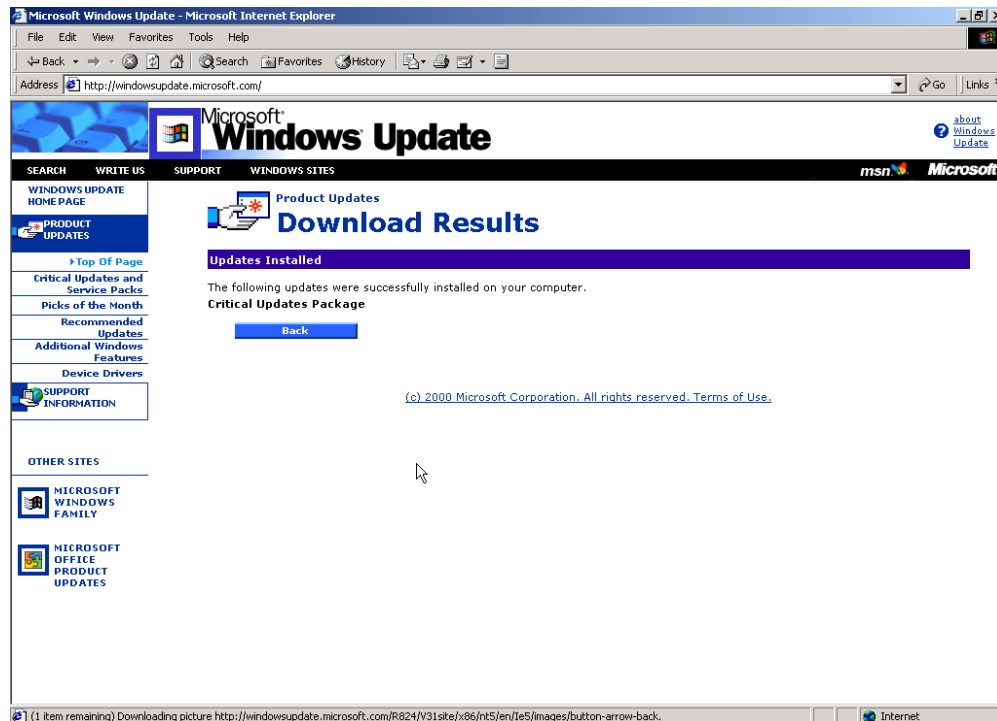


Figure E.11: Windows Update Download Results Window

Often, a reboot may be necessary to activate the updates (see Figure E.12). Click on the “Yes” button to restart the computer. Click the “No” button to continue the current Windows session (changes will NOT take effect until the computer has successful rebooted). If Windows Update does not prompt for a reboot, then the changes do not require it and are effective from the time of a successful install (see Figure E.11).

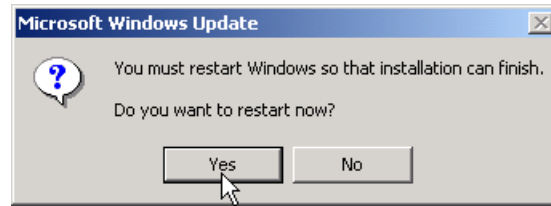


Figure E.12: Windows Update Reboot Dialog Box

If additional patches were required but could not be downloaded simultaneously, repeat the Windows Update process as required.

Appendix F: Microsoft Personal Security Advisor

Microsoft Personal Security Advisor (MPSA) is a site provided by Microsoft that allows a user to easily check the security and patch state of a Windows NT 4.0 or Windows 2000 computer (<http://www.microsoft.com/technet/mpsa/start.asp>). By using this website and associated application, a user can scan a Windows NT/2000 computer and receive a detailed customized report of the computer's security settings and patch level along with recommendations for updates and improvements.

Microsoft will continue to update this service as additional patches are released or when suggested security settings change. At the time of writing, the following checks are included in MPSA:

- Password Strength and Length
- Installed Service
- Macro Virus Protection
- Service Packs
- Hotfixes
- RASMAN
- Auto Login
- Auditing Events
- Secured File System
- Restriction of Anonymous Connections
- Shares
- Unusual Administrators
- Security Zones for Explorer and Outlook.

MPSA does have some limitations that users should consider before using:

- The hotfix checking portion of this tool is valid for English-language systems only.
- Presently, MPSA is intended to assess only Windows NT 4.0 Workstation and Windows 2000 Professional.
- MPSA does not check for web server-related patches. Users running Internet Information Server (IIS) or Personal Web Services (PWS) should use the HFNetChk (see Appendix G) tool to check for appropriate web server patches.

An example of a scan using MPSA is documented in the following example. To access the MPSA site, enter its Uniform Resource Locator (URL) (see above) into the Address Line of the Internet Explorer (note: MPSA requires Internet Explorer 5.0 or later). This will open the MPSA homepage (see Figure F.1).

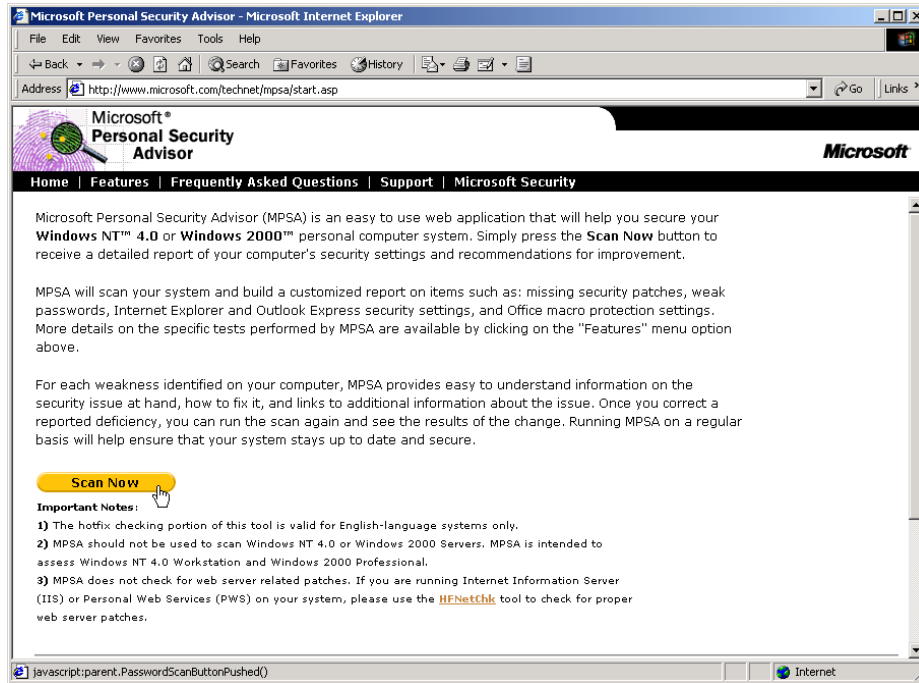


Figure F.1: MPSA Homepage

To start the scan, press the yellow button labeled “Scan Now.” Note that MPSA conducts a scan without sending sensitive client information to Microsoft. MPSA is implemented as an ActiveX control (application). To install and run MPSA, click the “Yes” button when the Security Warning window opens (see Figure F.2).



Figure F.2: MPSA Security Warning

Once a user selects yes, the ActiveX control is downloaded. This action usually takes less than a minute and automatically initiates and processes the scan (see Figures F.3 and F.4).

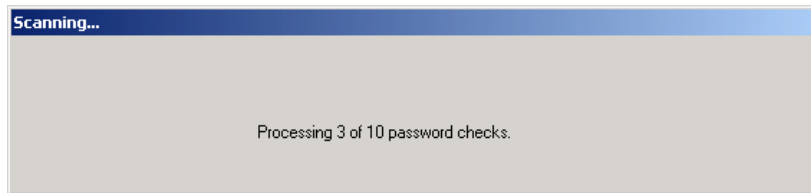


Figure F.3: MPSA Active Scanning

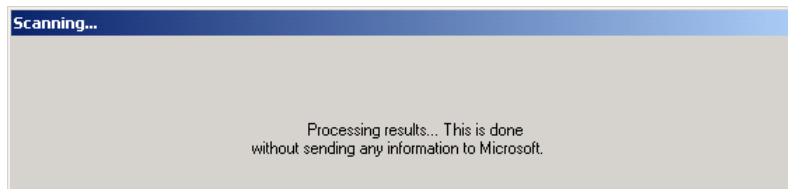


Figure F.4: MPSA Scan Processing

Once the scan is complete, MPSA generates a report on the overall security and patch level of the scanned computer. Three risk levels are assigned to each item as shown in Figure F.5.

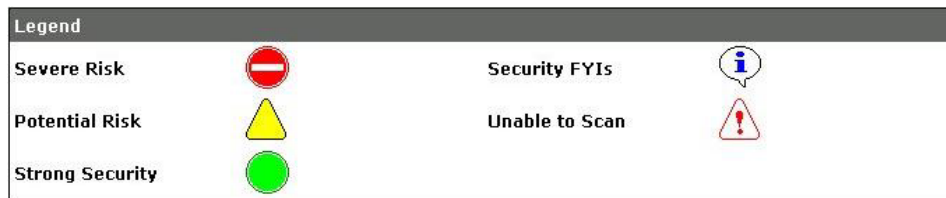


Figure F.5: MPSA Report Legend

- **Severe Risk**—There is a serious vulnerability in the system, and change is recommended immediately.
- **Potential Risk**—There is a potential vulnerability in the system, and change is highly recommended.
- **Strong Security**—No vulnerability is detected in the scanned system.
- **Security FYIs**—There is an issue that might be a vulnerability, but it requires further investigation.
- **Unable to Scan**—The MPSA was unable to scan for this item.

The final report provides details and advice regarding each of the tests performed (see Figure F.6).

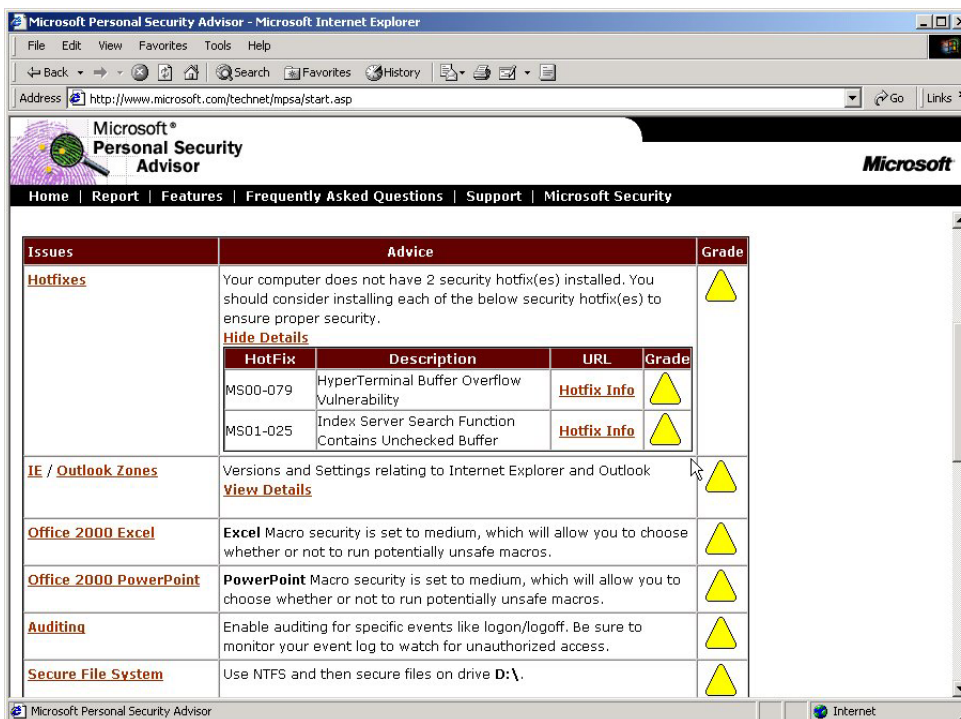


Figure F.6: Final Report

Clicking on the hyperlinks will provide additional details as shown Figure F.7.

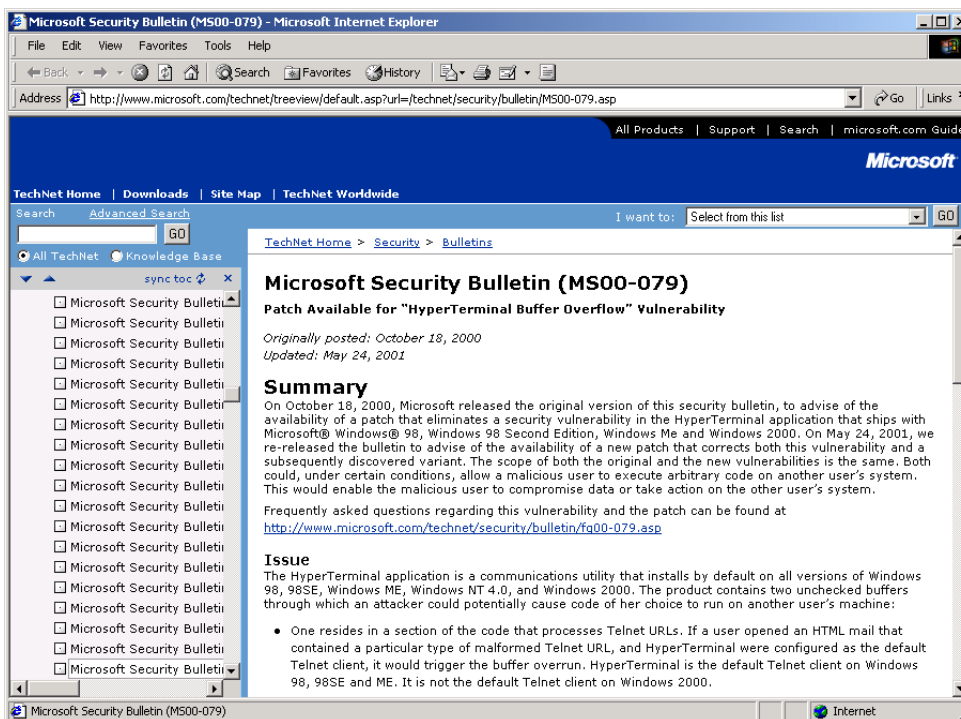


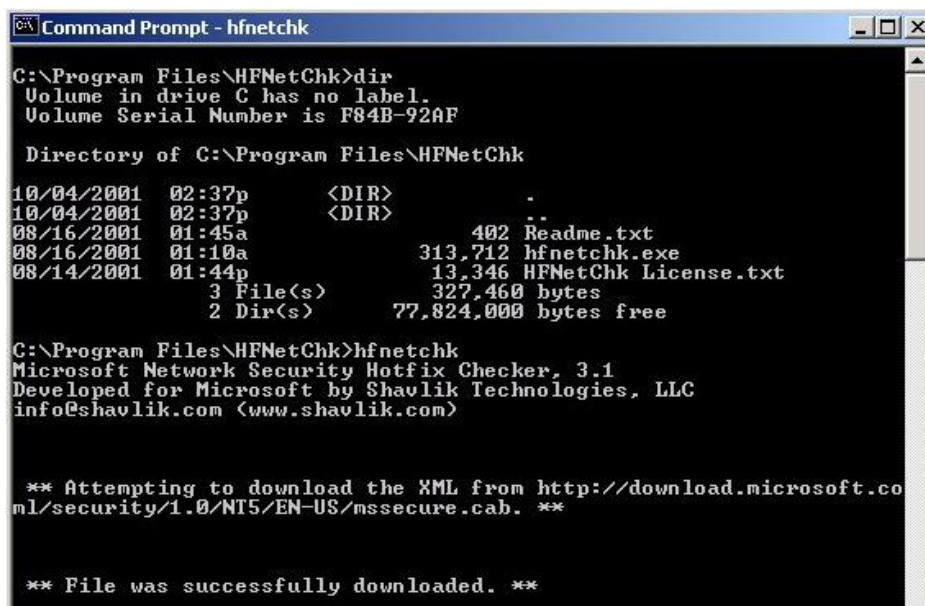
Figure F.7: Final Report Details

Note that when using the MPSA scanner, it must be run after any vulnerability is patched. Although all steps may be correctly implemented to patch a system, a possibility exists that one patch may “unfix” another. Therefore, it is important to always double-check the system by rerunning the scan.

Appendix G: Microsoft Network Security Hotfix Checker

Microsoft Network Security Hotfix Checker (HfNetChk) is a command line tool written by Microsoft to access the patch status for Windows NT 4.0 and Windows 2000 operating systems and hotfixes for Internet Information Services (IIS) 4.0 and 5.0, SQL Server 7.0 and 2000, and Internet Explorer 5.01 and later. Although not as easy to use, it supports more Microsoft products than either Microsoft Windows Update or MPSA.

After downloading and installing HfNetChk, run *hfnetchk.exe* from the command line. The program will then attempt to download an Extensible Markup Language (XML) file from Microsoft. This XML file contains the information on the current patch and update status of the programs and operating systems being checked (See Figure G.1).



```

C:\Program Files\HFNetChk>dir
Volume in drive C has no label.
Volume Serial Number is F84B-92AF

Directory of C:\Program Files\HFNetChk

10/04/2001  02:37p    <DIR>          .
10/04/2001  02:37p    <DIR>          ..
08/16/2001  01:45a             402 Readme.txt
08/16/2001  01:10a           313,712 hfnetchk.exe
08/14/2001  01:44p          13,346 HFNetChk License.txt
           3 File(s)              327,460 bytes
           2 Dir(s)             77,824,000 bytes free

C:\Program Files\HFNetChk>hfnetchk
Microsoft Network Security Hotfix Checker, 3.1
Developed for Microsoft by Shavlik Technologies, LLC
info@shavlik.com (www.shavlik.com)

** Attempting to download the XML from http://download.microsoft.co
ml/security/1.0/NT5/EN-US/mssecure.cab. **

** File was successfully downloaded. **

```

Figure G.1: Running Hfnetchk

To start the actual scan, agree to the installation and running of the downloaded XML file (see Figure G.2).



Figure G.2: Installing XML File

After clicking the “Yes” button, the program will load the XML files and scan the computer. The results listed on the command line screen list provide limited information (see Figure G.3).



Figure G.3: Hfnetchk Output

Although the listed information can be useful, the format and lack of detail is not comprehensive enough for most users. To correct this deficiency, there is a freeware

tool Maximized Software Hotfix Reporter. This utility works in conjunction with hfnetchk to display the results in an easy-to-read hypertext markup language (HTML) format. The utility has hyperlinks that make it easy to download the associated patches and hotfixes. This utility can be found at <http://www.maximized.com/freeware/hotfixreporter/> (see Figure G.4).

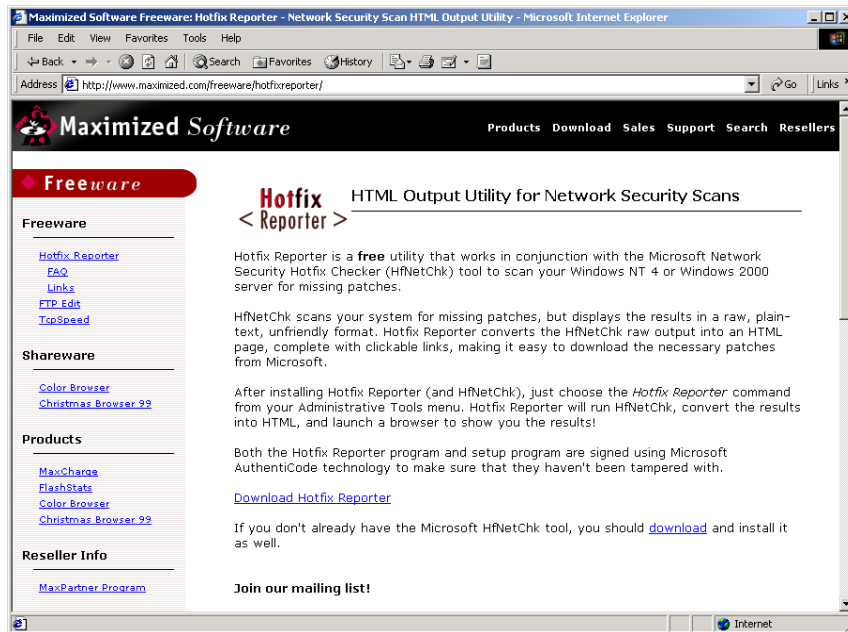


Figure G.4: Maximized Software Website

Download by clicking on the hyperlink [Download Hotfix Reporter](#).

Run the setup program to install Hotfix Reporter. The Hotfix Reporter must be installed in the same directory as HfNetChk (see Figure G.5).

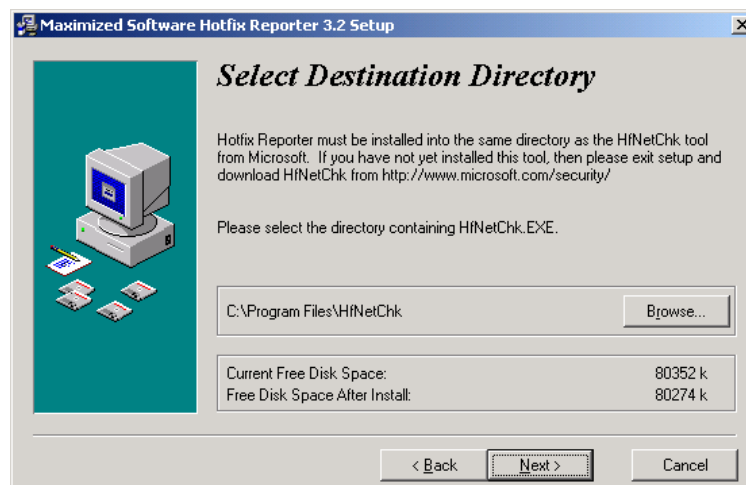


Figure G.5: Installing Hotfix Reporter

After installation, launch the Hotfix Reporter from the Windows Start menu. The execution of the Hotfix reporter is very similar to HfNetChk (see Figure G.6).

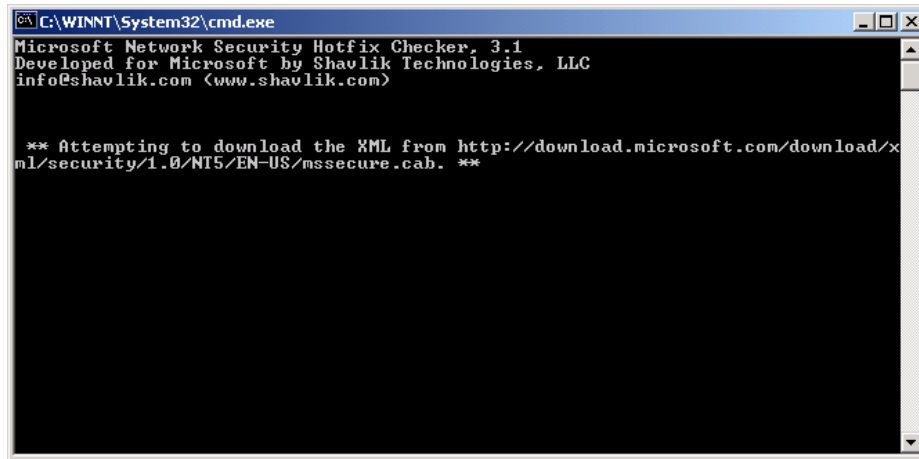


Figure G.6: Running HfNetChk With Hotfix Reporter

When HfNetChk attempts to install and run the latest XML file from Microsoft, the user will need to click “Yes” to continue the operation (see Figure G.7).



Figure G.7: Installing Microsoft XML Data File

Once HfNetChk has completed the scan, the Hotfix Reporter will then open a HTML file in the default web browser. This provides results in a more readable and usable format than that of HfNetChk. These results are also stored locally and can be reviewed as needed. As with the MPSA, it is important to rerun the Hotfix Reporter again after updates are installed to ensure that the system is running appropriately.

Appendix H: Microsoft Qfecheck Hotfix Checker

Qfecheck.exe is a command-line tool released by Microsoft (<http://support.microsoft.com/default.aspx?scid=kb;en-us:Q282784&ID=282784>) that gives network administrators the ability to track and verify installed Windows 2000 and Windows XP hotfixes (it does not currently support Windows NT). This tool provides the following capabilities:

- Assists administrators in identifying which hotfixes are installed on a computer. This simple tool easily enumerates all of the installed fixes by Microsoft Knowledge Base article number. Administrators can then confirm that they have installed the appropriate set of patches.
- Assists administrators in ensuring that Windows 2000 and Windows XP hotfixes are applied in a consistent manner across their organization, this tool allows administrators to create logs for each computer in their organization that show which fixes are installed. Once those logs are created, an administrator can easily scan them for consistency.
- Assists administrators in situations when the update itself does not install or a subsequent update that improperly overwrites a previous fix. Qfecheck ensures that not only have the fixes been installed, but that they are properly operating on the computer.

Qfecheck.exe determines which hotfixes are installed by reading the information that is stored in the following registry key:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates

Using the file version information that is stored in this key by each hotfix that is installed, the Qfecheck.exe tool checks the installed binary files to verify that they match.

Qfecheck.exe identifies the following types of issues:

- Files that have been hotfixed, but for which the installed binary file is not current.

Qfecheck reads the registry key that is associated with each update and checks the version number that is recorded in the registry against the current version of the same file that is installed. If the current version is lower than the version that is recorded in the registry, Qfecheck reports an error.

- Hotfix files that are current, but are not considered valid by the installed catalogs.

For each file that is installed by a hotfix, Qfecheck checks to see that the current catalogs on the computer contain the information that would be used by Windows File Protection (WFP) to validate the file. If a file is valid according to the hotfix information in the registry, but the installed catalogs do not concur, Qfecheck reports an error.

NOTE : If WFP were to be triggered in this case, the hotfixed file would be rolled back to an earlier version.

Qfecheck displays its information in a command-prompt window when you run it. If you log the results of Qfecheck to a log file with the /l switch, the log file is stored in the current folder unless you specify a location. This location can be any valid path, including a Universal Naming Convention (UNC) path. Qfecheck does not log information in the event log.

Once Qfecheck has been downloaded from the Microsoft website, it will be necessary to install it. To accomplish this, double click on the downloaded file (see Figure H.1).

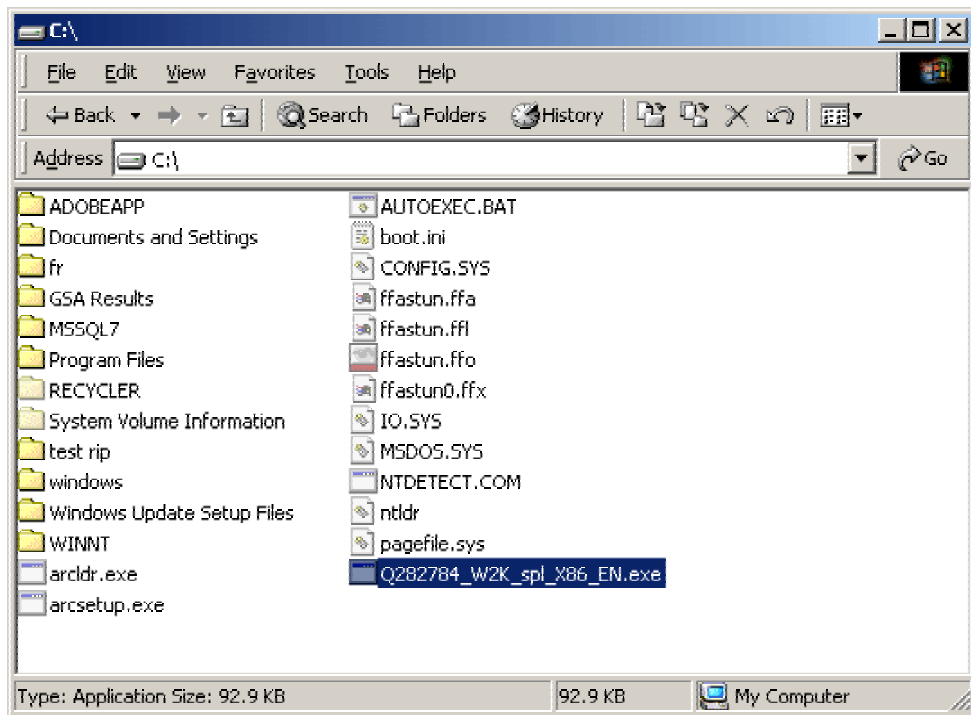


Figure H.1: Qfecheck Install

If the install is successful the administrator will be notified (see Figure H.2).

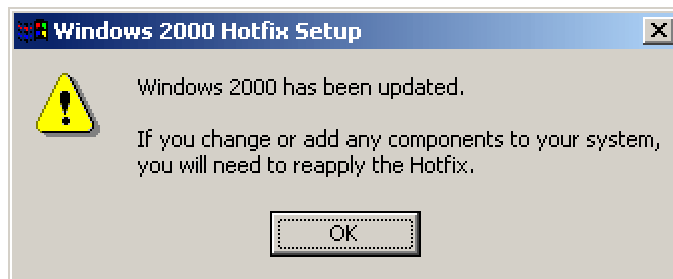
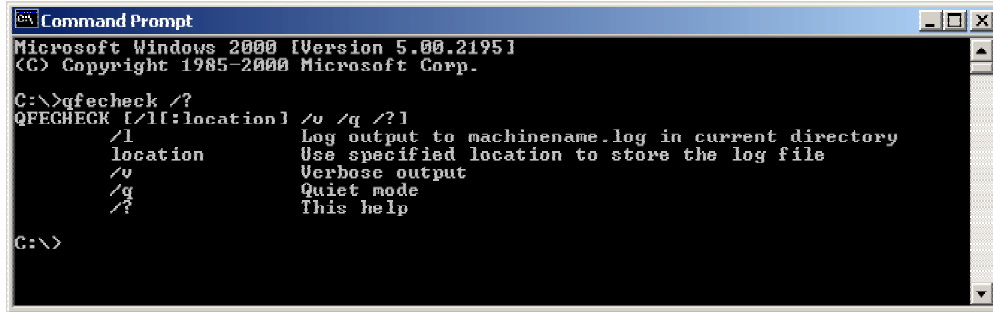


Figure H.2: Qfecheck Successful Install Confirmation

To run Qfecheck, open a Command Prompt window and type qfecheck.exe /?. This will provide the administrator with a list of qfecheck options (see Figure H.3).



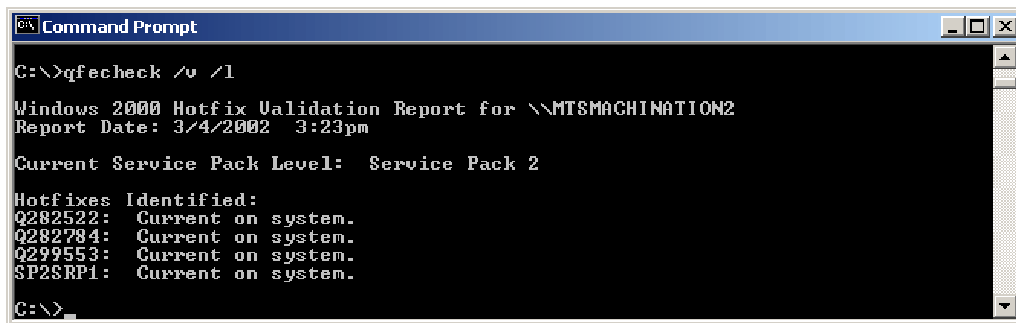
```

C:\>qfecheck /?
QFECHECK [/I[:location] /v /q /? ]
    /I          Log output to machinename.log in current directory
    location    Use specified location to store the log file
    /v          Verbose output
    /q          Quiet mode
    /?          This help
C:\>

```

Figure H.3: Qfecheck Options

To start qfecheck, type qfecheck followed by desired options (if any). Figure H.4 demonstrates qfecheck operating in verbose mode with an output log.



```

C:\>qfecheck /v /I
Windows 2000 Hotfix Validation Report for \\MTSMACHINATION2
Report Date: 3/4/2002 3:23pm

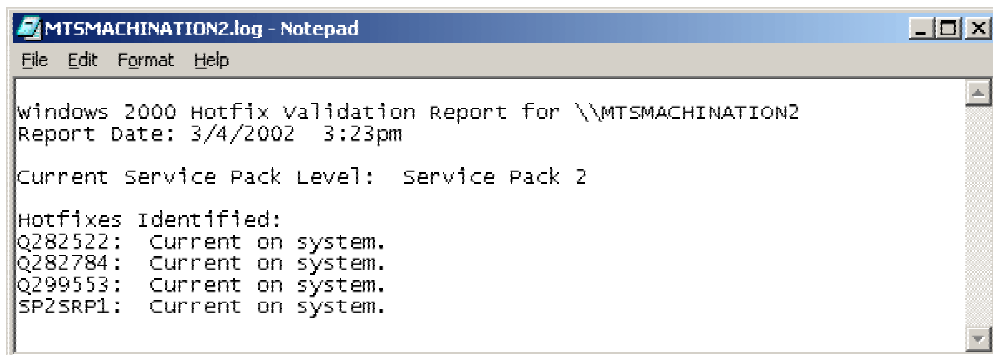
Current Service Pack Level: Service Pack 2

Hotfixes Identified:
Q282522: Current on system.
Q282784: Current on system.
Q299553: Current on system.
SP2SRP1: Current on system.
C:\>

```

Figure H.4 Running Qfecheck

Unless a particular directory was specified, the log file will be created in the %systemroot%\winnt\system32 directory. The log file name (unless otherwise specified) will be computername.log (where “computername” is the name of the host being checked). The log provides the same information as seen on screen while qfecheck is running (see Figure H.5).



```

windows 2000 Hotfix Validation Report for \\MTSMACHINATION2
Report Date: 3/4/2002 3:23pm

Current Service Pack Level: Service Pack 2

Hotfixes Identified:
Q282522: Current on system.
Q282784: Current on system.
Q299553: Current on system.
SP2SRP1: Current on system.

```

Figure H.5: Qfecheck Logfile