Applying Security Patches
Draft Release

October 23, 2001

CONTRACT NUMBER SPO 700-98-D-4002, DO 86

Information Assurance Technology Analysis Center (IATAC)
3190 Fairview Park Drive • Falls Church, VA 22042

Distribution A
Approved for public release, distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE (DD-MM-YYYY)	2. REPORT TYPE	3. DATES COVERED (FROM - TO)
23-10-2001		xx-xx-2001 to xx-xx-2001

4. TITLE AND SUBTITLE Applying Security Patches (Draft) Unclassified	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME AND ADDRESS Booz Allen & Hamilton 8283 Greensboro Drive McLean, VA22102	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME AND ADDRESS NIST U.S. Dept. of Commerce ,	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
APUBLIC RELEASE
,

13. SUPPLEMENTARY NOTES
Per conversation with Abe Usher, IATAC, performing organization is Booz Allen & Hamilton.

14. ABSTRACT
See report.

15. SUBJECT TERMS
IATAC COLLECTION

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE	Same as Report (SAR)	87	EM145, (blank) lfenster@dtic.mil
Unclassified	Unclassified	Unclassified			19b. TELEPHONE NUMBER International Area Code Area Code Telephone Number 703767-9007 DSN 427-9007

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10/23/01	Report 10/23/01

4. TITLE AND SUBTITLE
Applying Security Patches (Draft)

5. FUNDING NUMBERS

6. AUTHOR(S)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Booz Allen & Hamilton
8283 Greensboro Drive
McLean, VA 22102

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

NIST United States Department of Commerce
National Institute of Standards and Technology

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; Distribution is unlimited

12b. DISTRIBUTION CODE
A

13. ABSTRACT ***(Maximum 200 Words)***

Failure to keep operating system and application software up to date is the most common mistake made by information system professionals and users. Unfortunately, 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 up to 20 bugs per 1000 lines of code. Most of these bugs do not represent significant errors and do not affect the performance or security of a system, and therefore are not usually noticed. However, some may have a negative impact on security and performance.

14. SUBJECT TERMS
IATAC Collection, security, information technology, operating systems, virus, worm, patches, vulnerabilities

15. NUMBER OF PAGES
85

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED

NIST United States Department of Commerce
National Institute of Standards and Technology

Applying Security Patches

Certain commercial products are described in this document as examples only. Inclusion or exclusion of any product does not imply endorsement or nonendorsement by NIST or any agency of the U.S. Government. Inclusion of a product name does not imply that the product is the best or only product suitable for the specified purpose.

Table of Contents

Introduction

Failure to keep operating system and application software up to date is the most common mistake made by information system professionals and users. Unfortunately, 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 ~~up~~ to 20 bugs per 1000 lines of code. Most of these bugs do not represent significant errors and do not affect the performance or security of a system, and therefore are not usually noticed. ~~They do sometimes , h~~However, some may have a negative impact on security and performance. (This last sentence contradicts the second to last regarding security.)

As software programs expand, the potential number of bugs grows. Windows 3.1, released in 1992, had an estimated 3 million lines of code.[1] Thus according to common opinions it would contain an estimated 15,000 to 60,000 potential bugs within Windows 3.1. In 1999, Windows 2000 was released. Estimates for the number of lines of code within Windows 2000 differ by as much as 30 million lines. With a low estimate of 35 million lines of code there would be a 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 in RedHat 7.1.

These 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 maintaining the operational availability, confidentiality, and integrity of information systems. New patches are released daily and it is often difficult even for experienced system administrators to keep abreast of all the new patches.

Generally speaking patches are released for two reasons:

- To fix faults in an application or operating system. An example of this is a security hotfix for Microsoft's Internet Information Server (IIS). 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 add functionality or to address a new security threat. An example of this is new virus definitions for an anti-virus application. There was nothing "wrong" with the code of the anti-virus program, but it had to be updated to detect new viruses that did not exist when the applications was first released.

[1] Throughout this document examples of bugs and software applications will be employed for illustrative purposes only. This does not mean to imply that product is defective or that it should not be used.

The recent outbreaks of the Code Red and Nimda worms show why patching is so critical. During June 2001, a network security company discovered a serious vulnerability in Microsoft IIS web server application. Within days, Microsoft released a patch to eliminate the vulnerability; 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 1 week despite the fact that the patch had been available for several weeks. Unfortunately, people 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 thousands of additional computers. Neither of these viruses would have had much effect if not for thousands of computer administrators neglecting to patch their systems.

CERT[2] (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 levelwith the most recent updates and/or releases. Doing so significantly reduces the security vulnerabilities of their systems.

Purpose and Scope

This document provides guidance for system administrators and technical managers processes and procedures for installing and verifying on when and how to check for and install application and operating system patches. It describes how to keep apprised of patch releases and when and how to install patches.

This document does not address specific patches or vulnerabilities (except as examples) or how they might be mitigated. However, it does present a systematic approach for identifying and installing necessary patches. Following this systematic approach will potentially reduce the number of vulnerabilities incidents to which a network may be subject.

Audience

This document is written for system administrators, technical managers, functional managers, and other information technology (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 for, and support during decision-making processes.

[2] Note: CERT is no longer an acronym; it does not stand for anything. Previously, it stood for Computer Emergency Coordination Team. (I believe within DOD, "CERT" remains a valid acronym.)

Implementing A Patch Application System

Given the number of patches and ~~the~~ complexity inherent ~~in~~of ~~any but the most basic~~ information networks, ~~most organizations need to create a~~ systematic approach to identifying and applying patches is essential. This approach will vary from organization to organization depending on the ~~organizations~~ structure, threat level, and security posture. Each organization will need to create its own system that addresses its particular needs and priorities. The system must be a collaborative effort ~~created by the system and~~among network administrators, security officers, and management in order to fully ~~reflect~~ meet the needs of the organization.

Although each organization will rank its precautions and issues differently, all patched application and operating systems will include the following:

- Identification of applicable vulnerabilities and patches
- Testing of patches
- Deployment of patches.

Each of these steps is summarized bellows in and discussed greater detail in later sections of this document.

(NOTE: There is a little bit of a disconnect here. The introduction identifies patches to "application" and "operating systems." The above paragraph uses more specific language of "application systems." This is a hybrid of what was previously presented as distinct elements. I started to make some changes and have decided to stop, pending clarification. I would recommend you maintain the more specific language of "application software" and "operating systems."

Identifying ~~Applicable~~ Vulnerabilities and Applicable Patches

At all times, a group or individual should be responsible for identifying vulnerabilities in the current production system and locating any associated patches or mitigation techniques. ~~–~~Priorities at this level should include identifying patches in a timely manner for the most critical or threatened systems, implementing mitigation techniques to ~~the~~ lessen the risk associated with the vulnerability until the patch can be tested and applied, and identifying the experiences of others in using the patch to assist in the testing and deployment phases. (NOTE: This paragraph speaks to both itself and the subsequent paragraphs of "testing patches" and "deploying patches." The paragraph goes on to introduce some very worthwhile priorities such as "implementing mitigation techniques" and "identifying experiences…" There are no subsequent discussions of each (as there is with "testing patches" and "deploying patches", which might be both useful and logical. For example… the reason it is important to "identify experiences" is to ensure the ultimate instructions that accompany dissemination of a patch are properly executed. Improperly executed patches can leave a system equally if not more vulnerable.)

Testing Patches

Patches always need to be tested. Testing ~~S~~sometimes occurs in conjunction with deployment (e.g., when a patch is applied directly to production system) and at

other times occurs prior to deployment (e.g., when a patch is applied to a test system preceding deployment). There are advantages and disadvantages to both options that are discussed in greater detail later.

There are two types of testing that need to be conducted on a patch. One, the patch needs to be tested to ensure that it ~~as~~ does not ~~affected~~ negatively affect the performance of the system ~~negatively~~. Two, the patch also needs to be tested to ensure it has adequately corrected the vulnerability.

Deploying of Patches

Only after the patch (or bundle of patches) has been completely verified and tested ~~in~~ should it be deployed ~~to or throughout~~in the production environment (there are some exceptions to this rule discussed later). Methods should be developed to automate the deployment as much as possible. This not only reduces the cost of staff-hours required to manually deploy the patch, but also minimizes the risk of human error. If the patch is not deployed properly, it has a higher risk of introducing new vulnerabilities.

See Figure 1 for an example of a simplified patch application system. (NOTE: it might be useful for this diagram to use the same language as used above in describing the system. The above words should describe this system in the form of a coherent word picture. For example…the arrow on the box for "Development/Testing" points to itself. I'm unclear on its intent or the implication.)

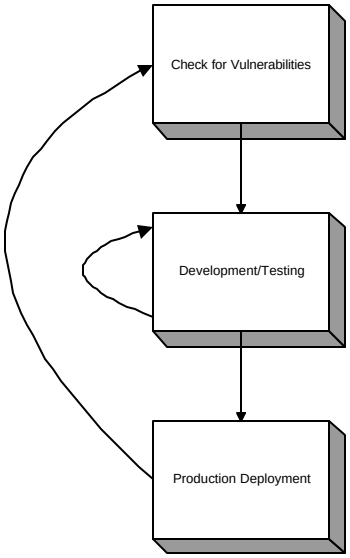


Figure 1 — Patch Application System

Identifying New Patches and Vulnerabilities

Perhaps the most difficult part of the patching process for most system administrators is keeping abreast of all of the latest developments in patches and updates. This can be a monumental task, particularly for system administrators responsible for maintaining a heterogeneous environment. Statistics in eWeek and Security Focus, two on-line publications, show overwhelming this undertaking can be. According to eWeek (http://www.eweek.com) Microsoft has released more than 21 security bulletins for Internet Information Server 5.0 alone in just 1 year. Security Focus (http://www.secutityfocus.com) estimates that 60 new security patches are released every month. To make matters worse, system administrators must watch two different (but related) moving targets: patches and vulnerabilities. Patches, as discussed below, are released to fix problems in software these are often, but not always, security related.

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 target computer. Not all vulnerabilities have related patches, and thus system administrators must be aware of vulnerabilities as well as patches and must mitigate "unpatched" vulnerabilities though other methods (firewalls, router access control lists, etc.). 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, it can crease serious problems since that the system administrator's chief adversary, the hacker, spends most of his or her time monitoring and exploiting vulnerabilities. The number of new vulnerabilities being discovered is increasing each year. CERT (https://www.cert.org) reports a dramatic increase since 1995 (see Table 1).

Table 1 — Number of New Vulnerabilities Reported to CERT

Year	Vulnerabilities
1995	171
1996	345
1997	311
1998	262
1999	417
2000	1,090
2001(first half)	1,151

The system administrator has numerous resources at his or her disposal to monitor the status of patches and vulnerabilities for the systems he or she support. Each type of resource has its own strengths and weaknesses. Often a system administrator will want to refer more than one to ensure timely knowledge of new patch releases and vulnerabilities. The most common forums for monitoring the release of patches and identification of vulnerabilities are:

- Vendor websites

- Vendor mailing lists
- Third-party websites
- Third-party mailing lists and newsgroups
- Other notification tools.

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

Vendor Websites

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

- Most patches are released by vendors.
- Often the vendor website is the only reliable place to acquire a patch (be suspicious of patches from third-party sources).
- Vendors often provide a wealth of information on vulnerabilities, methods of mitigation, and instructions on installing and using patches.
- Vendors often have unparalleled expertise concerning their products.

Vendor website limitations:

- These sites provide no active notification (system administrator must the effort to visit and review the site frequently).
- Vendors may not provide all relevant information (or alternative mitigation procedures).
- System administrators managing a heterogeneous environment may have to peruse numerous vendor websites for the various products they support.
- There is frequently a several-day delay between the time a vulnerability is discovered and the time the vendor releases a patch (unfortunately, many vendors will not report the vulnerability until the patch is released).

Vendor Mailing Lists

Mailing lists are a more interactive way for vendors to interact with the users of their software. A mailing list is nothing more than a massive list of users that have requested to be included on a particular list (i.e., subscribers). Many larger vendors maintain mailing lists that allow them to send e-mail messages and notifications of patches and updates to the users of their products. (Some vendors prefer to rely on newsgroups [see section below]).

Vendor mailing lists offer similar advantages to vendor websites:

- Most patches are released by vendors.
- Use of a mailing list does not require a system administrator to visit a website instead, notification is sent by e-mail to the system administrator's inbox.
- Vendors often provide a wealth of information on vulnerabilities, methods of mitigation, and instructions on installing and using the patch.
- Vendors often have unparalleled expertise concerning their products.

Vendor mailing list limitations include:

- The vendor may not always provide all relevant information (or alternative mitigation procedures).
- System administrators managing a heterogeneous environments may have to subscribe to numerous vendor mailing lists for the different products they support, which may overwhelm their inboxes with e-mails.
- Vendors sometimes succumb to the temptation to use their mailing lists for marketing e-mail (SPAM), which may cause administrators to ignore or "filter" all messages from the list.
- There is often a several-day delay between the time a vulnerability is discovered and the time the vendor sends an e-mail notification, due to vendor reluctance to provide information on a vulnerability before a patch release.

Third-Party Websites

Third-party websites are not maintained, or generally endorsed, by product vendors. These websites may cover a large number of vendors and products or may specialize in a specific vendor or product. They are often report new vulnerabilities before the vendor 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 (as noted above).
- 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 (saving the administrator time since he or she does 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.

The limitations of third-party websites include:

- No active notification (the system administrator must make the effort to visit and review the site frequently).

- Administrators need to be cautious of third-party patches.

- May not include all needed resources (e.g., patches available at vendor sites).

Third-Party Mailing Lists and Newsgroups

The primary advantage of third-party mailing lists and newsgroups is that they allow system administrators and other users to interact supporting two-way communications where vendor mailing lists support only one-way (vendor to user) communications. This allows administrators to share their experiences and ask questions. The principal difference between a newsgroup and mailing list is that newsgroup is an "officially" recognized Internet forum and, as such, can only be established by following fairly lengthy procedures. In contrast, anybody with a mail server and Internet access can set up a mailing list. Also mailing lists may be moderated and membership controlled.

The advantages of third-party mailing lists and newsgroups include:

- Allow interaction between administrators

- Do not require a system administrator to visit a website (information is sent via e-mail to his or her inbox)

- Allow system administrators to learn directly from the experiences of others (e.g., are there problems associated with a particular patch, does it really correct the problem).

Third-party mailing list and newsgroups limitations include:

- Because third-party mailing lists generally allow anyone to participate, they tend to generate large amounts of e-mail this may overwhelm the valuable data. This problem is sometimes referred a low signal-to-noise ratio.

- Administrators must be careful what information they release on the mailing list, because malicious entities might watch the list (for example, an administrator who says his or her system is having a problem with security in a particular area may invite hackers to try to exploit that vulnerability.)

- Administrators need to be cautious of third-party patches.

- Many newsgroups have high levels of advertising messages (Spam) which are often annoying and offensive.

- Administrators must be careful to verify the information they receive from these forums, since there is generally no way to verify the source.

Other Notification Tools

As 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 both by vendors (e.g., Microsoft's Critical Update Notification application) and by third parties (e.g., Cassandra and Security Focus SIA). For more information on obtaining these and similar products, see Appendix A. Some of these products, such as Cassandra, are free, while others require a one-time fee or subscription.

The advantages of these notification tools include:

- They are customizable 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)

- They provide real-time alerts to the administrator (e.g., not requiring him or her to visit a web page).

The disadvantage of these notification tools include:

- 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.

Specific Vulnerability Identification Resources

This section identifies specific resources administrators can use to identify vulnerabilities and patches for their systems. The section discusses vendor and third-party web sites as well as applications that assist the administrator in discovering vulnerabilities and applying patches. These resources are just a few of the many that are available and many additional resources are provided in Appendixes A-G.

Mitre CVE

Mitre's Common Vulnerabilities and Exposures (CVE) (http://cve.mitre.org) is a list of standardized names for vulnerabilities and other information security exposures. The principal goal of CVE is to standardize the names for all publicly known vulnerability and security exposures. This list should be considered a dictionary NOT a database (like some other websites that will be discussed later in this section). The standardization provided by the CVE makes it easier to share data among separate vulnerability databases and security tools. The CVE is:

- A dictionary rather than a database
- Accessible for review or download from the Internet
- Industry-endorsed via the CVE Editorial Board.

The CVE provides:

- A single name for each vulnerability or exposure
- A standardized description of each vulnerability or exposure
- A means for disparate databases and tools to speak the same language (e.g., one tool can interpret the results of another since they both recognize the CVE name for a given vulnerability)
- A path to interoperability and better security coverage
- A basis for comparing among tools and databases.

CVE was created because although most information security tools include a database of security vulnerabilities and exposures, there is significant variation among them and no easy way of ascertaining when different databases are referring to the same problem. The consequences of these disparities are potential gaps in security coverage and no effective interoperability among the disparate databases and tools. In addition, each tool vendor currently uses different metrics to state the number of vulnerabilities or exposures it detects, meaning that there is no standardized basis for evaluation among the tools.

Through use of a standard list of vulnerabilities and exposures such as CVE, databases and tools can communicate in a common language. This means that IT security professionals will know exactly what each tool covers, because CVE gives them a baseline for evaluating the coverage. As a results, IT security professionals

can determine which tools are most effective and appropriate for their organization's needs. In short, CVE-compatible tools and databases will give administrators better coverage, easier interoperability, and enhanced security.

The CVE Editorial Board makes decisions on which vulnerabilities and security exposures should be included in the CVE. 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 are to be included in the CVE, and then determines the common name and description for each entry. The process begins with the discovery of a potential security vulnerability or exposure. The information is then assigned a CVE candidate number. The Editorial Board discusses the candidate and votes on whether or not it should become a CVE entry. If the candidate is accepted, it is entered into the CVE and published on the CVE website. Candidates can be searched on the site, but the CVE and candidates lists are separate. (See Figure 2 for more information on the CVE naming process.)

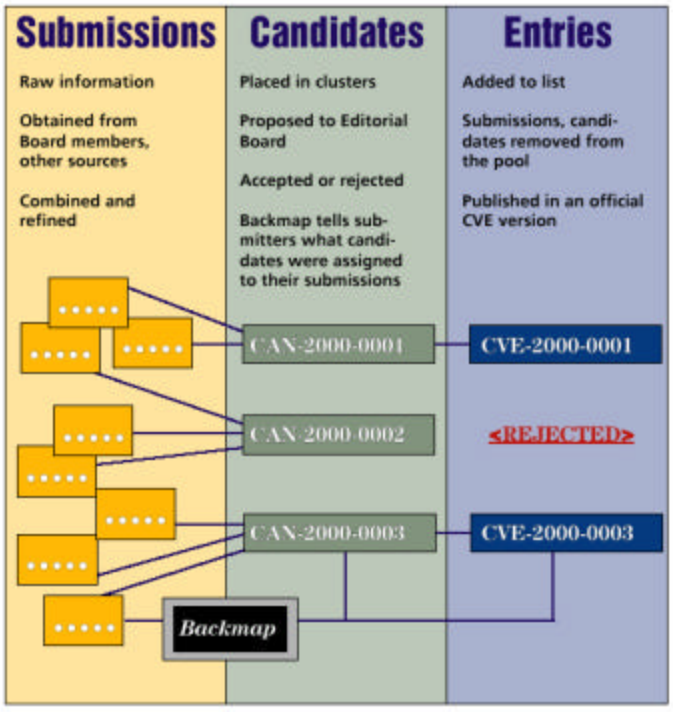


Figure 2 — CVE Naming Process

As of September 2001, the CVE contained 1,604 official entries. In addition, the CVE Editorial Board was evaluating 1,796 candidates.

Searching the CVE

The CVE offers several methods of searching its list of vulnerabilities and exposures (Figure 3).

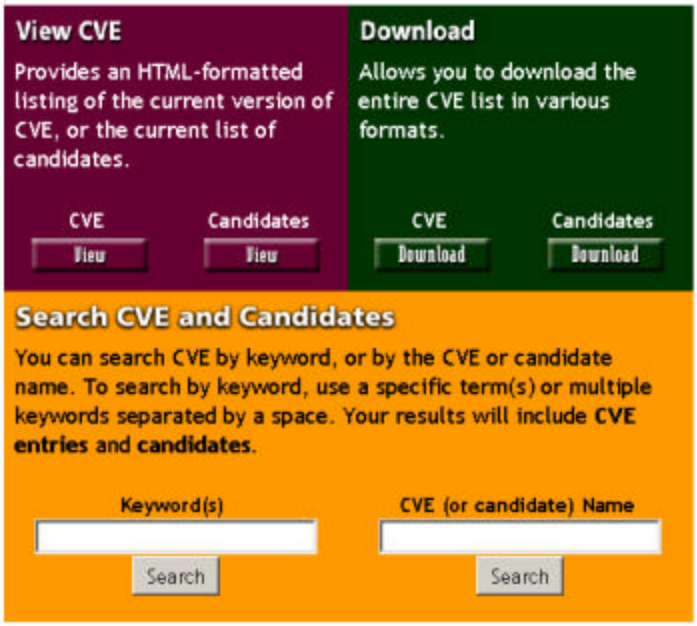


Figure 3 — CVE Search

Users can view the entire CVE list or candidates list by clicking on the appropriate view button in the purple "View CVE" square. This will list all the CVE or candidate entries starting with the earliest. Users can also download the CVE or candidate lists in their entirety in variety of formats by clicking on the appropriate button in the green "Download" square. Users may freely download, copy, redistribute, and reference the CVE but they may not modify them. Users of the CVE can also search each list via keyword or the CVE or candidate name.

While the CVE is useful from providing standardized names for vulnerabilities and of identifying potential vulnerabilities or exposures in systems, it provides very little, if any, detail on those vulnerabilities (Figure 4). To get that additional information, system administrators will have to use other resources, such as ICAT (described in the next section).

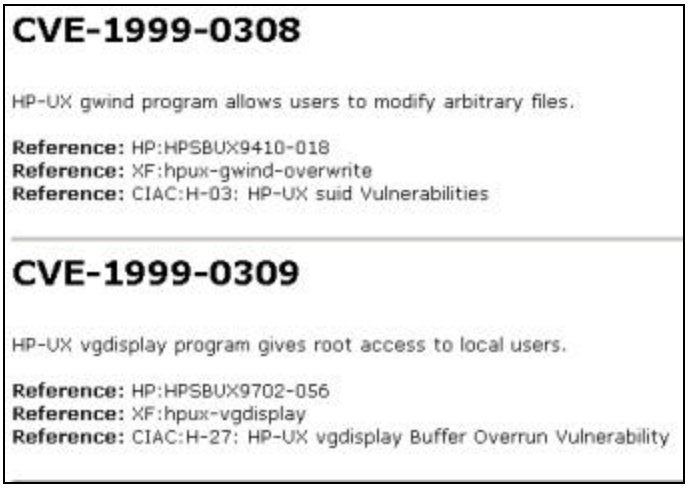


Figure 4 — CVE Search Results

ICAT

The National Institute of Standards and Technology (NIST) ICAT Metabase is a searchable index of computer vulnerabilities. It links users to a variety of publicly available vulnerability databases and patch sites, thus enabling administrators to find and fix the vulnerabilities existing on their systems. In this way, the ICAT Metabase takes the CVE list to the next level by including detailed information on each vulnerability or security exposure contained in the CVE and CVE Candidate lists. ICAT is not a vulnerability database, but a searchable index leading users to vulnerability resources and patch information on other participating websites. ICAT allows one to search at a fine granularity, a feature unavailable with most vulnerability databases, by characterizing each vulnerability by more than 40 attributes (e.g., software name, version number, vulnerability exploited).

ICAT indexes information from the following sources:

- CERT advisories
- ISS X-Force
- Security Focus
- NT Bugtraq
- Bugtraq
- Various other patch sites.

ICAT does not compete with publicly available vulnerability databases, but instead is a search engine that drives traffic to them. ICAT uses and is completely based on the CVE vulnerability naming standard (see previous section).

ICAT is updated as the CVE is updated generally every 2 to 4 weeks. The ICAT vulnerability analysts examine and publish all of the CVE entries and candidates within 1 week of their being announced by the CVE Standards Committee. Thus, there is a minor lag between the time a vulnerability is discovered and the time it appears on the ICAT website.

In addition to being accessible by web-based search, ICAT can also be downloaded in a variety of formats, including Microsoft Access 2000. This allows administrators to conduct searches without using the Internet and to integrate ICAT into their own applications. This standalone ICAT can be downloaded from 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 an administrator to enter the names and versions of the software used on his or her computers and networks into a database. Cassandra will then send the administrator e-mails about new CVE entries and candidates that meet the administrator's software profile. Unlike the alerts sent by many advisory systems which may or may not be applicable to an administrator, almost every vulnerability notification sent by this service will represent a vulnerability in the software that the administrator's organization employs. Further, using their Cassandra interface,

administrators can search ICAT based on their software profile. Since Cassandra stores the products the administrator uses, this eliminates having to search ICAT separately for each product every time he or she needs to check for new vulnerabilities. More information on 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 B of this document.

Vulnerability Databases

Even though ICAT conducts the most extensive search of different vulnerabilities provided by a single search engine, the lag-time inherent in the CVE naming process may require administrators to use one the many public available vulnerabilities databases as well. An extensive list of these databases is provided in Appendix C, and hyperlinks to several of these sites are provided in the ICAT Metabase.

These sites, generally run by third parties not affiliated with the software vendors, can provide a wealth of information to system administrators and security professionals. They generally 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.

These sites tend to be the quickest to report new vulnerabilities, which is both a benefit and a disadvantage. On the one hand, they provide timely information on vulnerabilities which is critical to the success of a system administrator in defending his or her network. On the other hand, the site do not provide the organized vetting of vulnerabilities that occurs with the CVE and ICAT. This deficiency means the same vulnerability may be reported more than once dramatically increasing the chance of false reports.

Although the amounts and type of information vary to some degree from site to site, sites generally include the following types of information:

- **News**—General information system security articles that provide system administrators and security professionals with the background information they may find useful by providing articles and news on all aspects of information security. The news section often includes information on ongoing or predicted threats including:
    - Malicious code (virus, worms, and Trojans)
    - Hacktivisim (hacking to advance a political or economic agenda often associated with website defacement)
    - Other kinds of new remote attacks and attack techniques not specifically related to a vulnerability.
- **Basics or Introduction**—Provides introductory information on networking and information system security. Useful to administrators who need some background in systems security.

- **Vulnerability Database**—Provides information on known vulnerabilities and patches, and generally includes most of the following.
    - Vulnerability Overview—This general includes a introduction to the vulnerability that generally includes.
        - CVE Number—Number assigned by CVE, if applicable
        - 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—The operating system, application, or hardware affected by the vulnerability.
    - Discussion or Analysis—More detailed information on the vulnerability. Information can be range in length from one paragraph to several pages depending on the complexity of the vulnerability and 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 applicable, will also include other mitigation techniques. May also discuss any negative impacts of the vendor's patch, if applicable.
    - Exploit[3]—Includes information on exploiting the vulnerability and any applicable software code. May also contain links to the other sites that have more information and exploit code. This information can be useful to the administrator in testing whether his or her system is susceptible to exploitation (before or after the patch is applied). However great care should be exercised in using these techniques.
    - 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 an administrator relies principally on other sources for vulnerability information, the general news and discussions provided on the vulnerability database sites can prove invaluable.

[3] 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.

Other Methods of Vulnerability Identification

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 just hosts and open ports but any associated vulnerabilities. It will identify a host's operating system and active applications and then compare these to 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 provide information on how to mitigate discovered vulnerabilities.

They can also help identify out-of-date software versions and applicable patches or system upgrades. And can also validate compliance with, or deviations from, the organization's security policy. In addition, vulnerability scanners can 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.)

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 however, 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 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 an 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).

[4] A surface vulnerability is a weakness as it exists in isolation, that is without any other vulnerability. The difficultly of identifying the risk level of vulnerabilities is that they rarely exist in isolation. For example, there could be several "low -risk" vulnerabilities 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. The reliable way to identify the risk of vulnerabilities in aggregate is through penetration testing. For more information on network testing see NST Special Publication SP-XX, *Information System Security Testing Overview.*

Vulnerability scanners are better at detecting well-known vulnerabilities than they are at finding more esoteric ones. This is due to fact that it is impossible for any one product to incorporate all known vulnerabilities in a timely manner. In addition, manufacturers may elect not include every possible vulnerability, to keep the speed of their scanners high (more vulnerabilities detected requires more tests which slows the overall scanning process).

Vulnerability scanners provide the following capabilities:

- Identifying active hosts on network
- 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 be of two types: network scanners and host scanners. Network scanners are used primarily to map an organization's network and identify open ports. 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 any misconfigurations.

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 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 is being compromised
- Tighten 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.

Network and host-based vulnerability scanners are available for free or for a fee. Appendix D contains a list of readily available vulnerability scanning tools and a simple example of scanner use.

Vulnerability Advisories

Certain websites issue ad hoc advisories for specific vulnerabilities and threats that exceed a particular threshold or magnitude. By addressing not only patches and vulnerabilities but also threats, these sites go beyond the other resources discussed so far. These advisories can be of great help to a system administrator. The two primary sources for these types of advisories are the Federal Bureau of Investigation's (FBI) National Infrastructure Protection Center (NIPC) and Carnegie Mellon University's CERT.

NIPC

Established in February 1998, the NIPC serves as the U.S. Government's focal point for threat assessment, warning, investigation, and response concerning threats or attacks against critical network infrastructures. The center is a joint Government and private-sector partnership that includes representatives from the relevant federal, state, and local government agencies as well as representatives of relevant commercial organizations.

The NIPC produces three levels of infrastructure warnings, which are developed and distributed in accordance with the FBI's National Threat Warning System. Collectively, these "threat warning" products are based on material that is significant, credible, and timely, and that addresses cyber and/or infrastructure dimensions with possibly significant impact. These warnings will often be based on classified material and include dissemination restrictions, but NIPC will generally produce a "sanitized" unclassified version for public use.

The three levels NIPC of threat warnings:

- **Assessments**—The lowest level of warning. These address broad, general incident or issue awareness information and analysis that are both significant and current, but that does 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. A partial sample of a NIPC alert is provided in Figure 5.

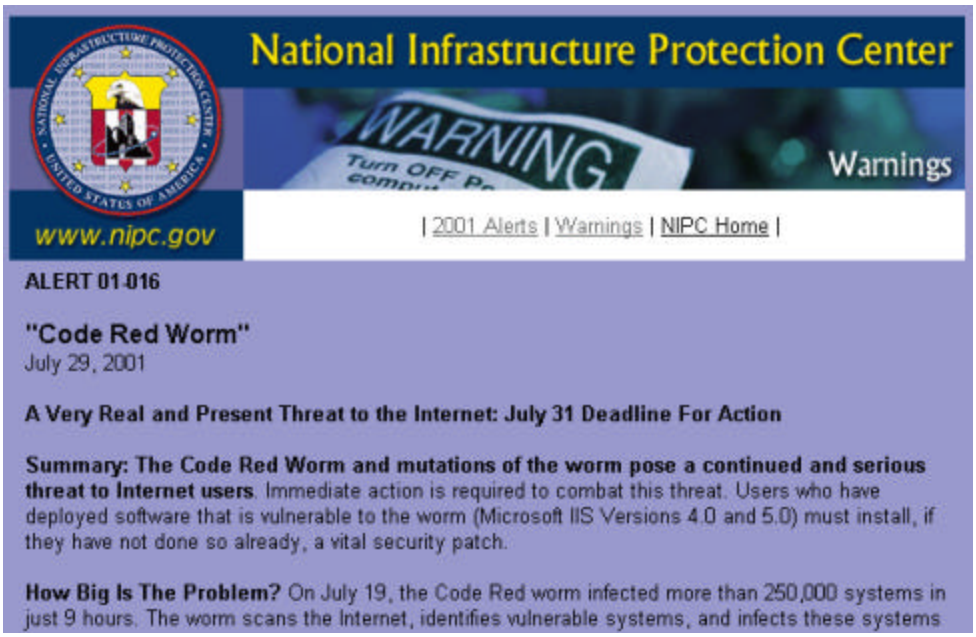


Figure 5 — Sample NIPC Alert

CERT

The CERT Coordination Center (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. The CERT/CC was established in November 1988, after a Cornell University graduate student released the Morris worm, which brought down much of the Internet and demonstrated the growing network's susceptibility to attack.

CERT responds to major vulnerabilities and threats by issuing advisories which can be found at http://www.cert.org/advisories/. Here users will find all CERT advisories dating back to the inception of CERT in 1988. Advisories are organized by year and then sequentially numbered. For example the first advisory for 2001 is numbered CA–2001-01 (CA = CERT Advisory, 2001 = year and 01 = 1st 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 6]).

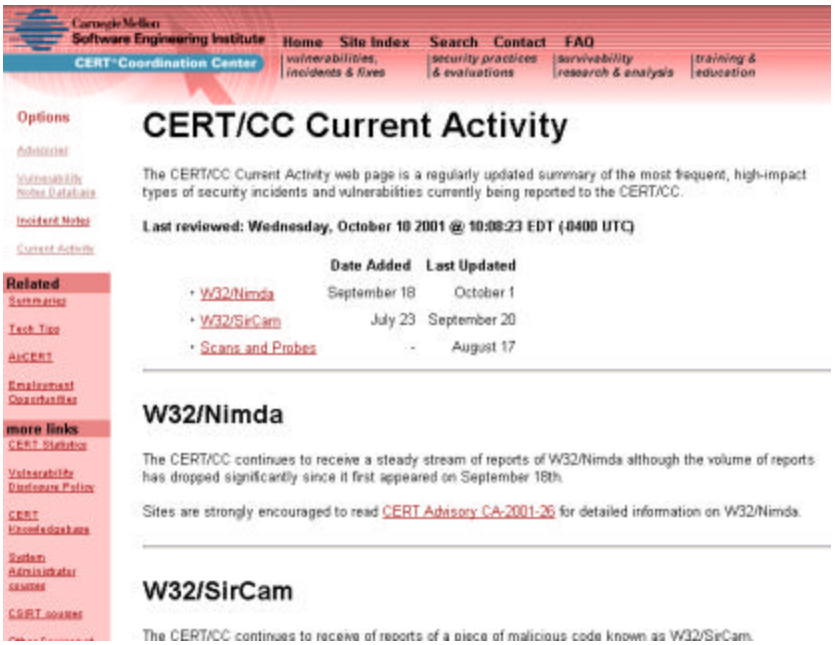


Figure 6 — CERT Current Activity Page

Each CERT advisory provides significant amount information to assist the system administrator and security professional. The advisories are also 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 on correcting the problem, including patches, if available.

Windows Update

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

The Product Updates area is a catalog of 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/updates as they require and download and install them straight from the web to their computers. More specifically, when a user first accesses the Product Updates page, Windows Update scans his or her system to ascertain the system's characteristics (including Windows version, service pack level, 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 he or she requires. Users should download any Critical Updates recommended for their system since these will rectify known problems (often performance or security issues). 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).

Patching Procedures

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, they are not always the easiest means of keeping up to date. For example, it is difficult for 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. These databases can be searched for the particular software or operating systems employed at the administrator's site. The 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.

Although 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, they generally should not be used to download patches. Administrators should generally be very suspicious of patches from third parties. Certain third-party website, 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 when in fact they are just a backdoor distributed by hackers in an attempt to compromise or otherwise identify vulnerable systems. Generally, patches should only be downloaded from the vendor or other trusted source.

An illustration of malicious code distribution in this manner involves the "removal" programs for the Windows Trojan backdoor, Back Orifice. The two programs are BOSniffer and BODetect. One of these programs, BODetect, actually does detect and remove Back Orifice; the other program, BOSniffer, is actually an install program for Back Orifice.

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 (MD) 5 checksum)
- Verify 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 an additional level of authentication, they are not foolproof. For example, hackers have obtained digital

signatures in the past that have provided them with the ability to distribute patches in the name of a major software company. Although this was just a proof of concept and the hackers did not attempt to exploit the fake digital signatures, they easily could have. The bottom line is authentication is no substitute for common sense.

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

If the patch is not distributed compiled (that is if the user must compile the source code prior to installation) the administrator should perform a code review before compiling and installation. This is a common situation for the patches of Linux and some Unix operating systems. If an administrator is not comfortable performing the code review he or she should use an expert who is, or ensure that the source code is download from a trusted site.

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 administrator should attempt to learn whether other have experienced:

- Patch does not correct the vulnerability
- Patch opens an old vulnerability
- Patch creates a new vulnerability
- Patch reduces reliability
- Patch reduces performance
- Patch is incompatible with other required applications.

If one or more of the above problems applies, the 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 issues above 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 although, even when this option is provided, the uninstall process does not always return the system to its previous state.

Another consideration that needs to be taken into account is the frequency of patch releases. For example, during the first nine months of 2001, Microsoft alone released 49 security bulletins with as many as 7 bulletins released in 1 month. In the year 2000, Microsoft released 100 security bulletins, releasing as many as 13 bulletins in 1 month. Thus, patching could easily become a full time job for an administrator.

Before applying a patch, an administrator and management must consider.

- 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 together 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 together
- Complications arising from combining patches
- 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 instillation.

The risk of delaying the application of patches, particularly, must be carefully weighed. For example, when the vulnerability that was eventually exploited by the Code Red worm was discovered, because there was no immediate danger many 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 computer(s) requiring patching face numerous and/or significant threats? For example, public web servers and most Federal Government organizations face fairly high threat level. Generally speaking, for these systems, timely patching is critical. In contrast for an internal intranet site that is inaccessible from the Internet, patching can often be delayed, since 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 promptly.
- **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 non critical systems if a successful exploitation would lead to "rooting"[5] of the system.

Unfortunately neither the decision apply nor the chose of not applying a patch is risk free. The correct decision is not always clear and too often the decision made is to make "no decision" which generally results in a compromised system. System administrators and management must work together to create a systematic process

[5] "Rooting" a system is hacker slang for gaining administrative or root-level access to a target system. This means the attacker has full control over the attacked system. Anyvulnerability that could lead to this level of access should be corrected or patched immediately.

for evaluating patches and determining the appropriate decision within the context of their organization.

Testing Patches

Testing of patches is critical for several reasons. In a perfect world, all patches would be widely tested before release and would work flawlessly. Unfortunately, matters are much more complicated. As previously discussed, bugs are 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 10 to 20 megabytes apiece). In addition, patches are often released in haste, in order to quickly repair a vulnerability; this means they often receive less testing from the vendor than did the original software. Thus, for all of these reasons, patches can easily produce unintended consequences. It may be possible for patches that do not contain a bug to change a system enough to cause some other software component or application to fail.

In addition to the need to identify any unintended consequences, patches should be tested to ensure that they have patched the vulnerability or corrected the performance issue as intended. Administrators should never assume that a patch is functioning on their system unless they have tested it themselves. This issue is complicated by the fact that the order 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 administrators exactly follow the vendor's instructions for applying all patches and then test the patches after installation.

Before applying a patch an administrator must decide whether he or she should install the patch directly on a production or install it on a development system or some other system. This issue is extremely complicated and is influenced by number of issues, including:

- Organization's configuration management policies
- Seriousness of vulnerability
- Threat level of system with vulnerability
- Ability to temporarily mitigate vulnerability through other methods (e.g., firewall rules, permission changes, etc.)
- Whether there is an appropriate system on which to conduct the test
- Complexity of patch
- Complexity of 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 system[s] to be patched)

- Previous experience in patching the systems on which the patch will be installed.

Only a system administrator, in conjunction with management, can make the appropriate determination on whether to perform testing on a development system or perform it on the production system.

Whether the administrator decides to install and test the patch on a development server or on the production server he or she will need to test both the operation of the system and the vulnerability or vulnerabilities the patch addresses. Once the patch has been installed successfully and the system rebooted (if required), the administrator should compare the performance of the system to previous baselines to ensure that the patch has not affected performance negatively. If the result of that test appears to be acceptable, the administrator should test the functionality of the system and ensure the system is operating correctly. If that test is completed successfully, the administrator should test whether the vulnerability has been corrected. This can be accomplished by several methods:

- Scan host with vulnerability scanner capable of detecting vulnerability
- Employ exploit procedures or code and attempt to exploit the vulnerability (i.e., perform a penetration test)
- Check that the files or configuration settings the patch was intended to correct have been changed as documented in the vendor's documentation.

A patch installation is not complete until the system's functionality and the remediation of the targeted vulnerability have been fully tested and determined to meet the organization's requirements.

Applying Patches

Patches can come in many different forms; can be applied by just a simple click from a web browser through complex operating system and kernel-specific compilation of source code. Patching of vulnerability may be as simple as the modifying a configuration setting, or may require the installation of a completely new version of the software. There is no simple patch application methodology that applies to all software and operating systems.

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

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 order 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. Due to the large variance in procedures for applying a patch from the distributions, the user should consult the vendor and distribution-specific user manuals for more detailed instructions. References for common Linux and Unix distributions can be found in Appendix A.

Updating Network Infrastructure Components

Network infrastructure components (e.g., routers, firewalls, switches, IDS etc.) are some of the most critical systems to keep patched. There are many different methods available for patching or updating these systems depending on component type and manufacturer. The process generally involves obtaining the software from the 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 as an organization's first line of defense should be updated as soon as there is a known vulnerability. A compromise of one these systems could lead to the compromise of the entire network. There is at best minimal time (hours to days) to test the patch or update prior to application, as attacks attempting to exploit these vulnerabilities are likely to be occurring as soon as the vulnerability is discovered.

Virus detection programs that are installed on the firewall, e-mail servers and file servers need to be updated very frequently. Anti-virus programs rely on a database of virus signatures to recognize a virus. If this database is not up to date then the program cannot recognize newer viruses that generally 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 anti-virus programs could result in the widespread distribution of one or more viruses within an organization.

Intrusion detection systems, similar to anti-virus programs rely on a database of attack signatures to recognize attacks. Since new attack techniques are being developed all the time, updating this database is critical to the ability of the IDS to detect and report attacks.

Updating Windows Operating Systems and Applications

There are many methods for applying patches for Windows and Windows-based software. Most version of Windows and many programs now have simple "update" buttons that when clicked automatically access the Internet and check for an update, and, if one is available, download and install it. This type of automated feature can be of great assistance to administrators. If no automatic update feature is built into the software, the administrator will have to manually monitor for updates using the procedures described previously. Even when automated procedures are available, many administrators prefer to manually install patches because this generally gives them greater control over the process. This is largely a matter of personal preference, with either option having both pros and cons.

To assist 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 on a continual basis. As of the writing of this document, Microsoft offered the following capabilities:

- **Windows Update**—This allows an administrator to scan his or her 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 allows an administrator to scan his or her computer(s) to find Office updates available through Microsoft.

- **Microsoft Personal Security Advisor (MPSA)**—MPSA an administrator to easily check the security stance and patch state of a Windows NT 4.0 Workstation or a Windows 2000 Professional computer. Using this web application, the 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 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).

- **Microsoft Security Toolkit, Strategic Technology Protection Program (STPP)**—Microsoft has recently started this program. This along with the Microsoft Security Toolkit, is a two-phased program. The first phase of the program is to get secure and the second phase is to stay secure. More information on 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 5 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 down 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 about it can be found at http://support.microsoft.com/support/kb/articles/Q224/4/20.ASP.

- **Microsoft Security Notification Service**—This free service that provides e-mail notification from Microsoft about the security of Microsoft products. Information on 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 means of 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 while

noncritical updates can come in the form of service packs and product updates. Noncritical updates usually are 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 prior to 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 applied, a reboot is usually required after the installation. When multiple hotfixes must be downloaded and installed, this can become a time-consuming task. To correct this problem, Microsoft has released another tool, QChain, that enables hotfixes to be bundled into one package and installed at one time. QChain and additional usage information can be found at http://support.microsoft.com/support/kb/articles/Q296/8/61.asp. This product can also allows administrators to create a customized patch installer for multiple systems that all require the same patches.

Automated Patch Distribution and Application Tools

While applying patches to 10-15 servers may seem a daunting task, it is nothing compared to rolling out a patch to hundreds or thousands of desktop systems. This task can be made easier through applications that automatically distribute updates to end-user computers. Some of these patch automation tools are included with network operating system software while third-party vendors distribute others.

The capabilities of these systems vary greatly. Some of these applications focus on the distribution of patches. They rely on the system or network administrator to identify a necessary patch and setup the patch distribution tool to deliver and install the patch. Other will actively search for necessary patches and suggest these to the system administrator and, at his or her command, install the patches on the appropriate hosts.

These patch distribution applications also vary greatly in their support of different operating systems and applications. Those 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.

Patching After a Security Compromise

Patching after a security compromise is significantly more complicated than merely applying the appropriate patch(es). While applying a patch after a security compromise will generally correct the vulnerability that was exploited, it will not eliminate backdoors[6] or most other changes that might have been introduced by the intruder. For example, the Code Red II worm placed backdoors on compromised systems. Hackers, and even the Nimda worm that came after Code Red II, then exploited these backdoors. Administrators who patched their systems after a compromise by Code Red II fixed the original vulnerability but not the backdoor.

Systems that are known or suspected to have been compromised must be patched more carefully than uncompromised systems. Generally speaking, if a system has

[6] A backdoor is a secret avenue of access placed on a compromised computer system by a hacker that allows him or her future access to the system.

been, or is suspected of being compromised it must be reformatted and reinstalled or restored from a known good backup. If that is not possible, significant expertise will be required managing the possible dangers inherent in compromised systems.

Some to consider when in deciding whether to reinstall a compromised system are as follows:

- Level of access to the intruders attained (root, user, guest, system, etc.)
- Purpose of compromising the system (e.g., web page defacement, illegal software repository, platform for other attacks, etc.)
- Method of system compromise
- Actions of hacker during and after compromise (e.g., erased log files, intrusion detection reports, etc.)
- Duration of compromise
- Extent of compromise on network (i.e., the number of machines compromised on network)
- Results of consultation with management and legal counsel.

The lower the level of access and the more the administrator knows about the hackers actions the less risk there is in patching the vulnerability as opposed to reinstalling the whole operating system.

System administrators also need to keep in mind if they wish to pursue any sort of criminal prosecution for the compromise they cannot in any way alter the compromised host or else it may not be used for evidence. There are very specific guidelines on the handling of a host after a compromise if it is to be used in evidence for criminal prosecution of civil litigation.

To recover from a compromise the following steps should be taken:

- Immediately disconnect compromised system(s) from network
- Consult organizational security policy
- Consult with management, legal counsel, management and law enforcement as appropriate
- Analyze intrusion including:
    - Modifications made to system software and configuration
    - Modifications made to data
    - Tools or data left behind by intruder
    - Review system, intrusion detection and firewall log files.

- Restore 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
    - Reconfigure network security elements (firewall, router, intrusion detection system, etc.) to provide additional protection.
- Update security policy
- Document lessons learned.

Appendix A: Patching Resources

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 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

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/

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.lynuxworks.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://jspiro.tripod.com/linux/hal91.htm
Hard Hat Linux	http://www.mvista.com/
Icepack Linux	http://www.icepack-linux.com/
Immunix OS	http://www.wirex.com/
JB Linux	http://www.jblinux.com/
Jurix Linux	http://www.jurix.org/

Operating System	Website
KRUD	http://www.tummy.com/krud/
KYZO	http://www.kyzo.com/
L13Plus	http://l13plus.deuroconsult.ro/
Linux Antartica	http://www.linuxantarctica.com/
Linux by LibraNet	http://www.libranet.com/
Linux Embedded	http://linux-embedded.com/
Linux Mandrake	http://www.linux-mandrake.com/
Linux Pro	http://www.wgs.com/
LinuxOne	http://www.linuxone.net/
LinuxPPC	http://www.linuxppc.org/
LinuxWare	http://www.trans-am.com/index1.htm
LoopLinux	http://www.tux.org/pub/people/kent-robotti/index.html
LuteLinux	http://www.lutelinux.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://pocket-linux.coven.vmh.net/about.html.en
Progeny Debian	http://www.progeny.com/
Pygmy Linux	http://pygmy.penguin.cz/
RedHat Linux	http://www.redhat.com/
Redmond Linux	http://www.redmondlinux.org/
Rock Linux	http://www.rocklinux.org/
RT-Linux	http://www.rtlinux.org/rtlinux/
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/
TINY Linux	http://tiny.seul.org/

Operating System	Website
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
WholeLinux	http://www.wholelinux.com/
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/mailing-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/mailinglists/index.html
Slackware	
Slackware Security Information	
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/

Operating System	Website
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
FreeBSD Support	http://www.freebsd.org/support.html
FreeBSD Mailing Lists	http://www.freebsd.org/support.html#mailing-list
OpenBSD	
OpenBSD Security Information	http://www.openbsd.org/security.html
OpenBSD Distribution	http://www.openbsd.org/ftp.html
OpenBSD Patches	http://www.openbsd.org/errata.html
OpenBSD Support	http://www.openbsd.org/docum.html
OpenBSD Mailing Lists	http://www.openbsd.org/mail.html
TrustedBSD	
TrustedBSD Security Information	
TrustedBSD Distribution	http://www.trustedbsd.org/downloads/
TrustedBSD Support	http://www.trustedbsd.org/documentation/
TrustedBSD Mailing Lists	http://www.trustedbsd.org/mailinglists/
Solaris Security Information	
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
Solaris Forums, Patch Documentation, and Patch Downloads	http://supportforum.sun.com/freesolaris/

Virus Software Download/Update/Security Centers

Topic	Website
McAfee Anti-Virus	
McAfee Anti-Virus	http://www.mcafee.com/anti-virus/
McAfee Anti-Virus Updates	http://download.mcafee.com/updates/updates.asp
McAfee Anti-Virus 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 Anti-Virus	
Symantec Norton Anti-Virus	http://www.sarc.com/avcenter/
Norton Anti-Virus 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 Bulletin	http://www.symantec.com/techsupp/bulletin/index.html
Antivirus Hoax Page	http://www.sarc.com/avcenter/hoax.html
Panda Anti-Virus	
Panda Anti-virus Home	http://www.pandasecurity.com/platinuminfo.htm
Panda Anti-virus Global	http://www.pandasecurity.com/gviinfo.htm
Sophos Anti-Virus	
Sophos Anti-Virus	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/
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

Topic	Website
F-Secure Anti-Virus	
F-Secure Anti-Virus	http://www.fsecure.com/products/anti-virus/
F-Secure Virus Info	http://www.fsecure.com/virus-info/
Miscellaneous Anti-Virus 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 B: Using the ICAT Website

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 administrators to identify and correct vulnerabilities existing 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 at a fine granularity, a feature unavailable with most dedicated vulnerability databases, by characterizing each vulnerability by over 40 attributes (including software name, version number impact, exploitable range, etc.). 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 does not compete with publicly available vulnerability databases but instead is a search engine that drives traffic to them. ICAT is maintained by NIST.

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 that ICAT uses to index its vulnerability information. 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 B-1).

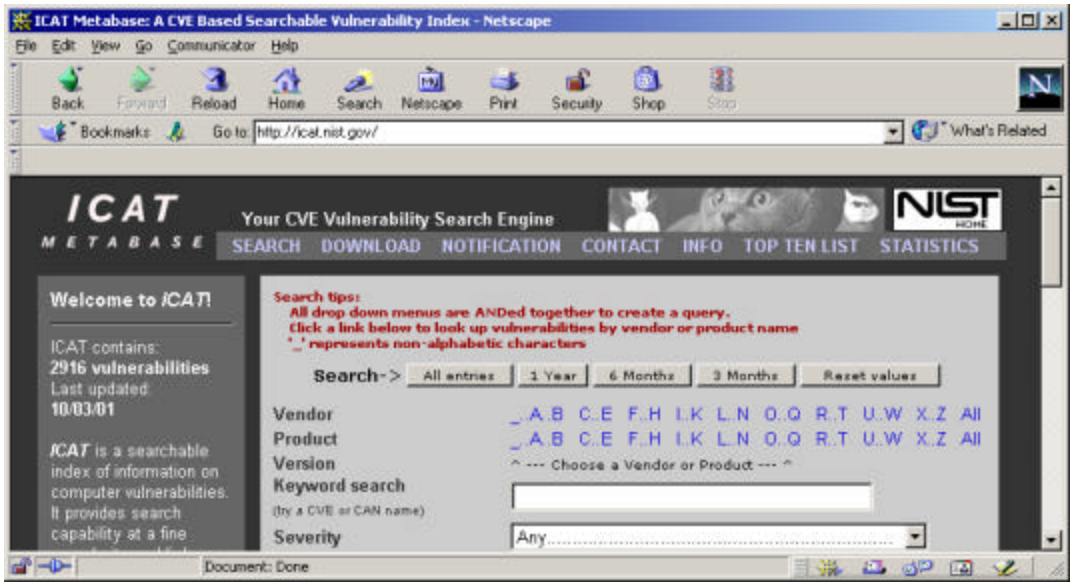


Figure B-1 — Accessing ICAT Metabase

Navigating the ICAT Metabase Website:

The ICAT Metabase homepage provides a variety of information and links that will assist the user in utilizing the ICAT Metabase.

The navigation bar at the top of the screen allows users to use and navigate the ICAT Metabase website (see Figure B-2).

Figure B-2 — ICAT Metabase Navigation Bar

- **Search**—Returns the user to the homepage/search page. From here the user can conduct searches of the ICAT Metabase (see section on searching below for more information).
- **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 connect 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 on automated notification applications that employ the ICAT Metabase. These applications will 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.
- **Contact**—Opens the *ICAT Contact Information* page. The page contains contact information for the ICAT staff. Users are encouraged to provide information on how they use the ICAT, to download ICAT advertising banners and provide a links to the ICAT Metabase from their website.
- **Info**—Opens a new browser window and 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* vulnerabilities page. This page contains a table that contains the top ten most popular vulnerabilities as defined by the number or requests made for a particular vulnerability through the ICAT metabase. To maintain 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.

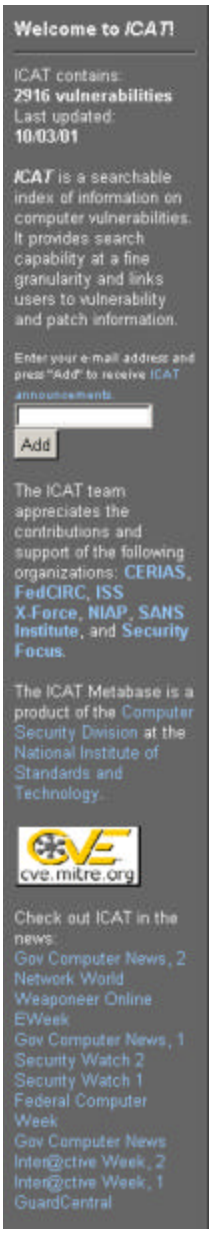


Figure B-3 — ICAT Sidebar

ICAT Metabase Sidebar

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

- The top of the sidebar provides the number of vulnerabilities contained with the metabase, the last update and a brief overview of the ICAT.
- The sidebar also allows a user to register for the ICAT mailing list. Important announcements about ICAT are sent out to subscribers of this list. It has an extremely low volume (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 users of the ICAT. Some of these links are to sites external to NIST. Note: When attempting to access an external site, there is an exit notification from the NIST website prior to accessing the external site. To speed loading of the external site click on the hyperlink provided after reading the disclaimer.
- The side bar also provides links to press articles concerning ICAT.

Common Vulnerabilities and Exposures (CVE)

The CVE is the common vulnerabilities and exposures list. It is an industry standard naming scheme for computer vulnerabilities and exposures that the ICAT employs to index its vulnerability information. Vulnerabilities will be named either CVE-xxxx-yyyy or CAN-xxxx.yyyy. The CVE prefix is attached to vulnerabilities that have been reviewed by the CVE standard advisory committee. The CAN prefix is 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 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 B-4). It is possible to search by the date when the vulnerability was published, vendor name, product name, version (available only when a product is selected), keyword search and/or severity level. ICAT also provides filters allowing users to limit their search results to particular sources, exploit range, vulnerability consequence, vulnerability type,

operating system type, expose component type, entry type and those vulnerabilities published after a particular date.

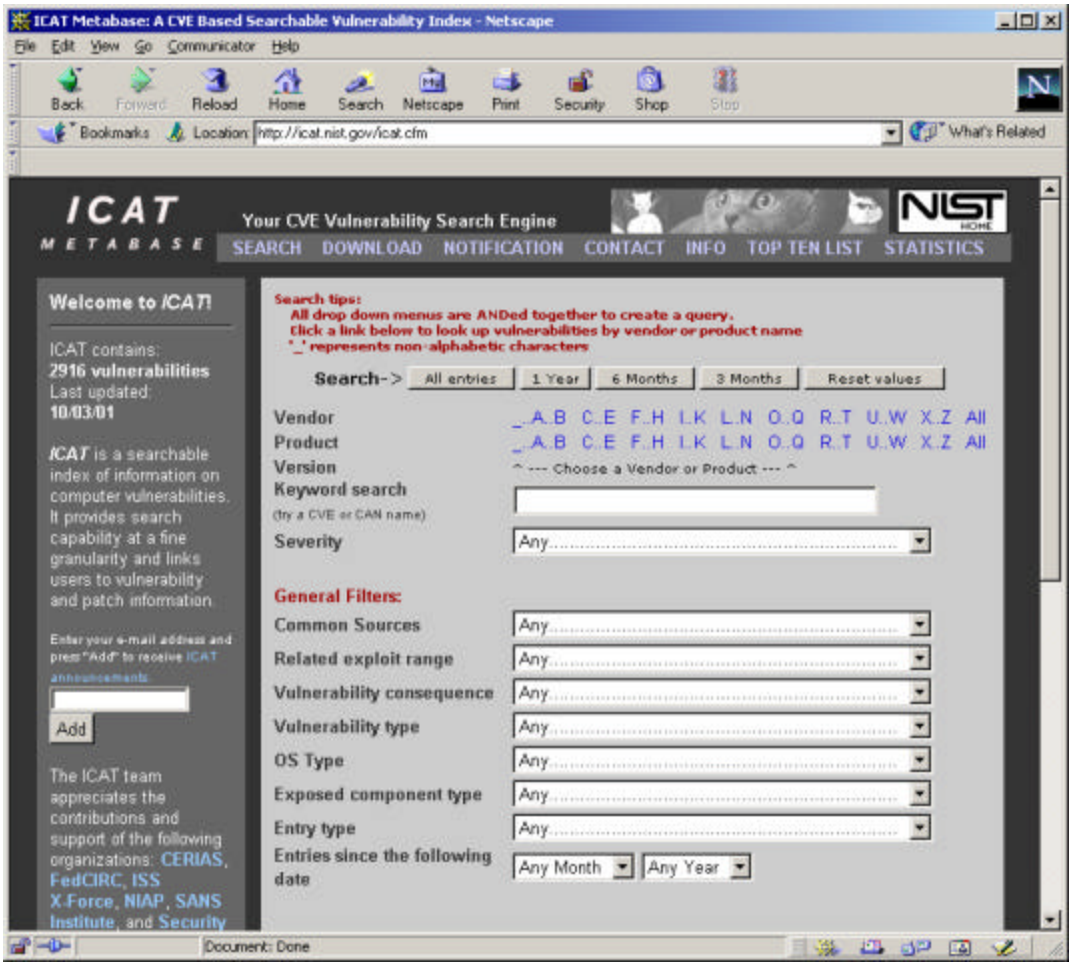


Figure B-4 — ICAT Metabase Navigation Bar

Defining Search Parameters

Before conducting a search, it will be necessary to define the search parameters. A number of methods are available in the ICAT Metabase. Multiple parameters can be set in order to narrow the search. The "Reset values" button resets all the search parameters and filters back to the default (see Figures B-4 and B-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 along the top of the search parameters section (see Figure B-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 by the user.

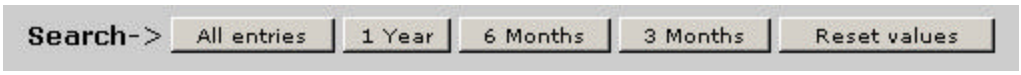


Figure B-5 — Date of Publication Search

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 B-6). A user can select the vendor of their choice by scrolling down the list and clicking on the vendor of their choice. The user can start their search by clicking on the appropriate search button (see Figure B-5) or narrow their search further by selecting a product from the vendor' s product list (see below). Note: as with all drop down boxes on the ICAT search page the user may select only one vendor at a time.

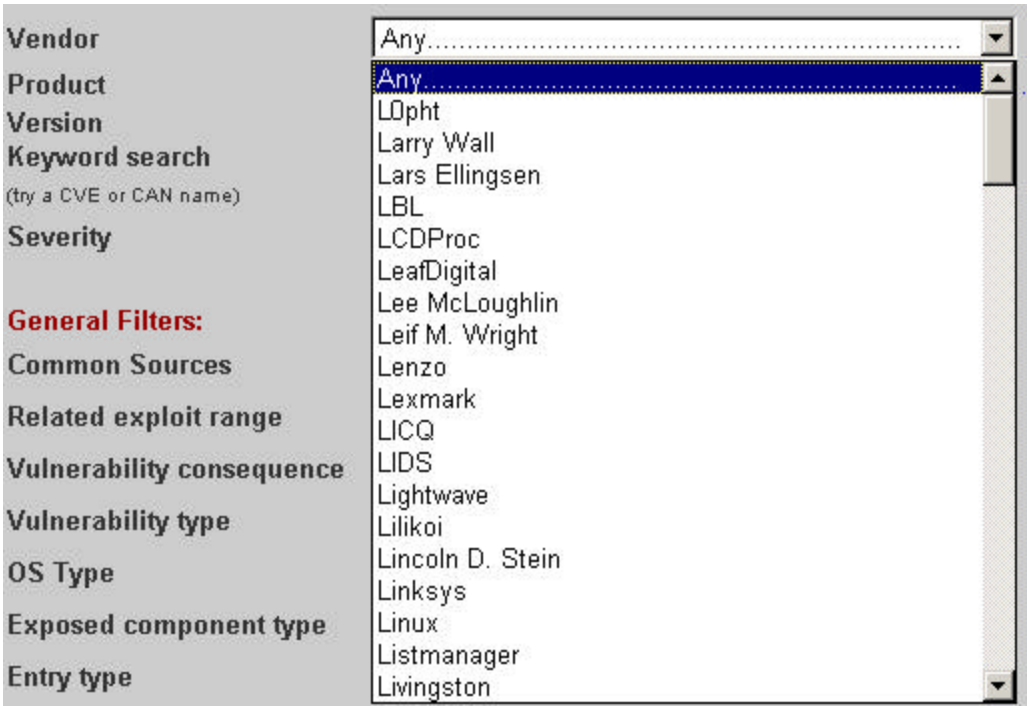


Figure B-5 — Vendor Search

Product—If a vendor has not already been selected, the user can click on the letter that begins the name of the product that they are interested in searching. If they have already selected a vendor (see above) then there will be a drop down box with that vendors products listed (see Figure B-6). The user can select a product by scrolling down the list and clicking on the appropriate choice. Note: as with all drop down boxes on the ICAT search page the user may select only one product at a time.

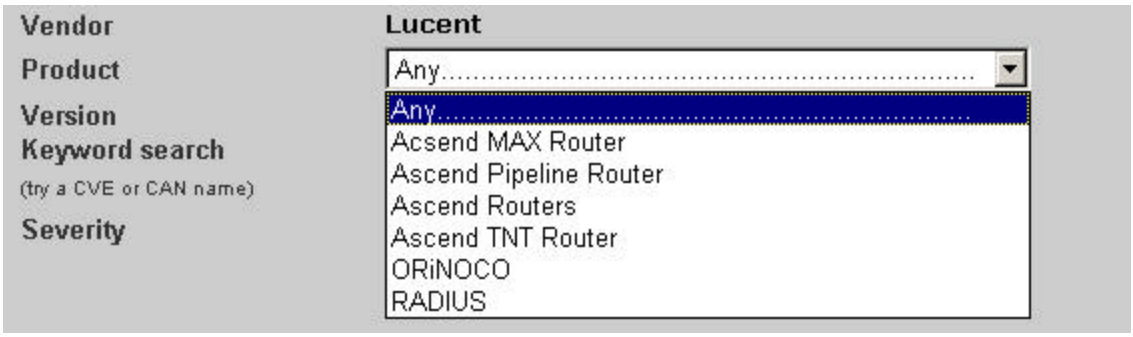


Figure B-6 — Product Search

(Product) **Version**—This option is only available if the user has selected a product. This search allows a user to select a particular version of a product (see Figure B-7). Given the large number of versions for certain products, it is possible that using this option will cause the metabase to not return all possible vulnerabilities. Note: as with all drop down boxes on the ICAT search page the user may select only one version at a time.

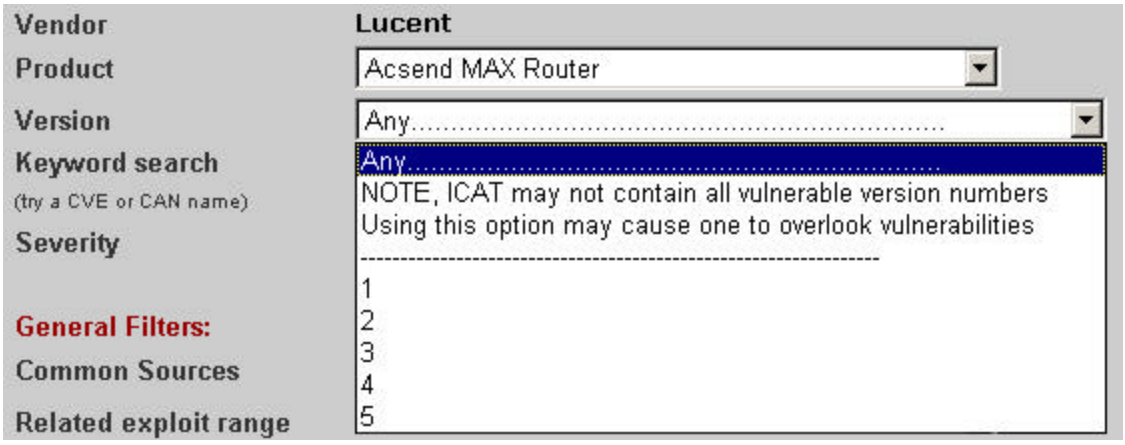


Figure B-7 — Product Version Search

Keyword—Allows the user to narrow their search using keywords (see Figure B-8).

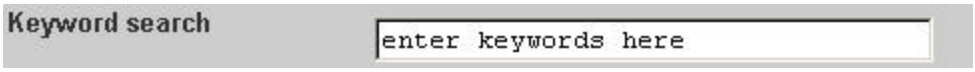


Figure B-8 — Keyword Search

Severity—Allows the user to narrow their search by severity level (see Figure B-9). Note: as with all drop down boxes on the ICAT search page the user may select only one severity level choice at a time.

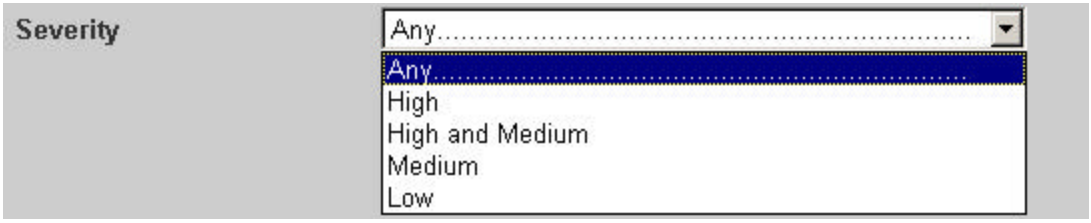


Figure B-8 — Keyword Search

Defining Search Filters

Optionally, the user can select filters to further narrow their search. The "Reset values" button resets all the search parameters and filters back to the default (see Figures B-4 and B-5). With limited exceptions noted below users can use as few or as many of the search parameters and filters as required.

Common Sources—This filter allows a user to select a specific source or vulnerabilities (see Figure B-9). Note: as with all drop down boxes on the ICAT search page the user may select only one source at a time.

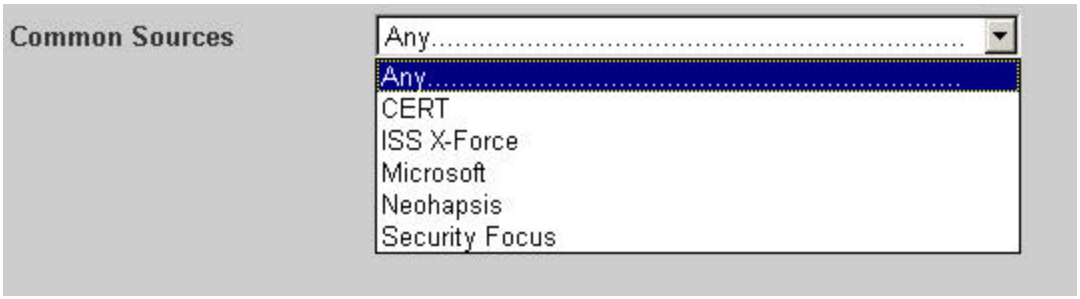


Figure B-9 — Common Source Search Filter

Related Exploit Range—This filter allows a user to limit their search to local vulnerabilities (those that require local access to exploit) or remote vulnerabilities (those that can be exploited without local access to a host). This filter option is shown in Figure B-10. Note: as with all drop down boxes on the ICAT search page the user may select only one option at a time

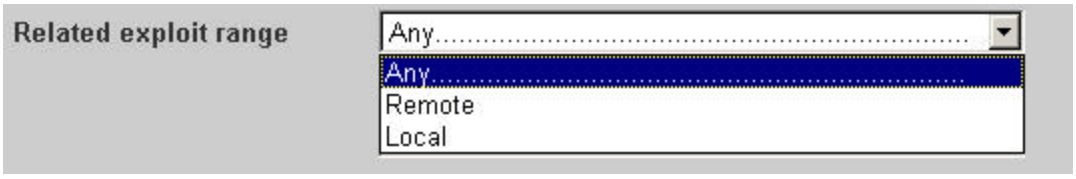


Figure B-10 — Exploit Range Search Filter

Vulnerability Consequence—Filters the results based on the consequence of the vulnerability (e.g., root level access, availability, etc.). Figure B-11 shows the options for this filter. Note: as with all drop down boxes on the ICAT search page the user may select only one consequence at a time.

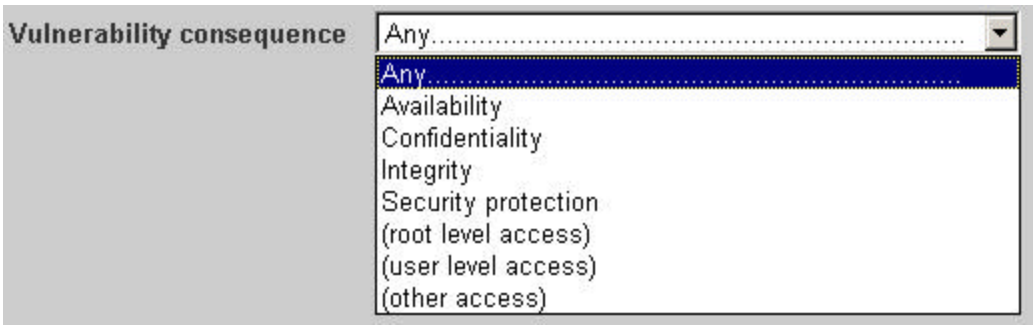


Figure B-11 — Vulnerability Consequence Search Filter

Vulnerability Type—Filters the results based on the root cause of the vulnerability (e.g., root level access, availability, etc.). Figure B-12 shows the options for this filter. Note: as with all drop down boxes on the ICAT search page the user may select only one vulnerability type at a time.

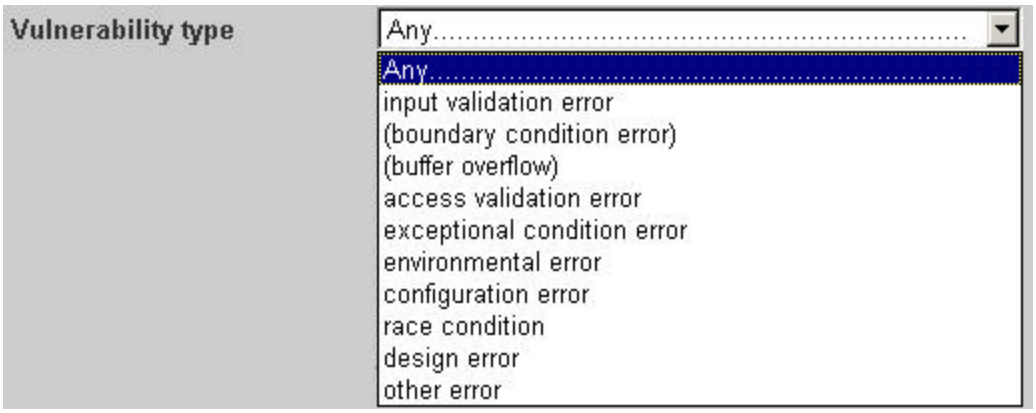


Figure B-12 — Vulnerability Type Search Filter

(Operating System) **OS Type**—This filter provides the ability to separate the results based on the operating system effected. Figure B-13 shows the options for this filter (the Windows NT option includes Windows 2000 and XP). Note: as with all drop down boxes on the ICAT search page the user may select only one OS at a time.

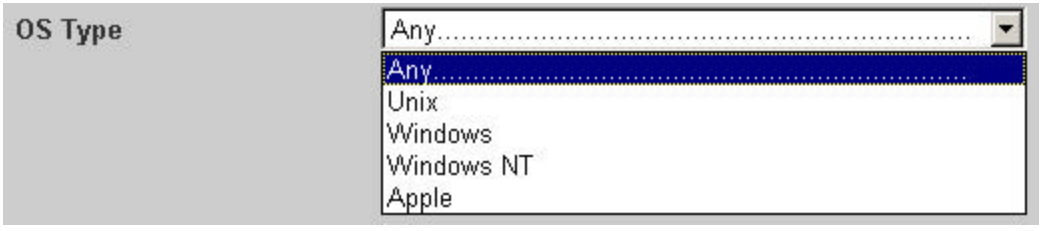


Figure B-13 — Vulnerability Type Search Filter

Exposed Component Type—Provides the ability to filter the results by the type of component effected (e.g., user application, communications protoc ol, hardware,

etc.) Figure B-14 shows the options for this filter. Note: as with all drop down boxes on the ICAT search page the user may select only one component at a time.

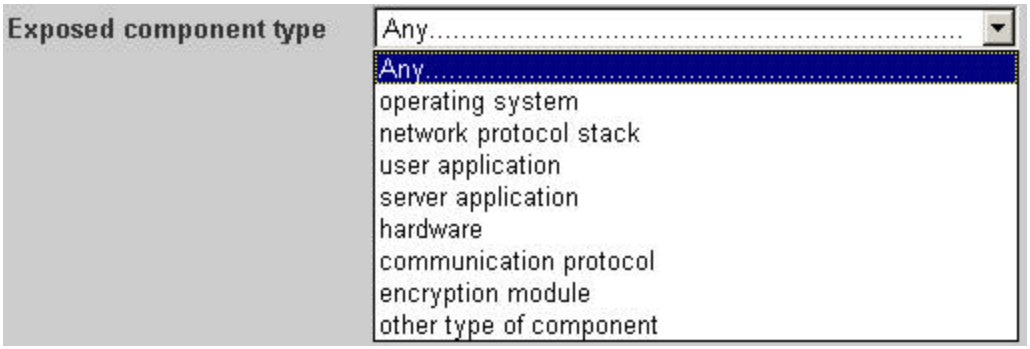


Figure B-14 — Exposed Component Type Search Filter

Entry Type—Provides the ability to filter the results by the type of entry. The ICAT contains two type of entries . The primary entries are those that have been accept by the CVE advisory committee. The entries start with the prefix "CVE." The other types of entries are those that are still being reviewed by the CVE committee and begin with the prefix "CVN." Figure B-15 shows the options for this filter. Note: as with all drop down boxes on the ICAT search page the user may select only one type of entry at a time.

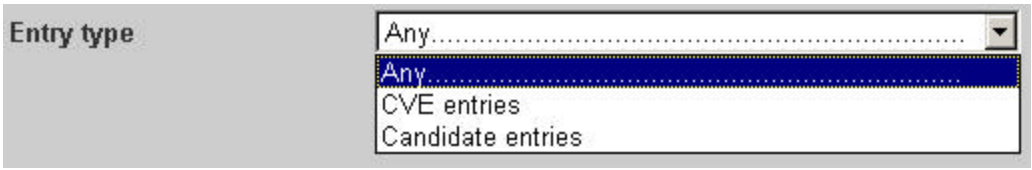


Figure B-15 — Entry Type Search Filter

Entries Since the Following Date—Provides the ability to select only vulnerabilities published after particular month and year (back to January, 1995) as shown in Figure B-16.

Figure B-16 — 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 his/her search criteria (see Figure B-17).

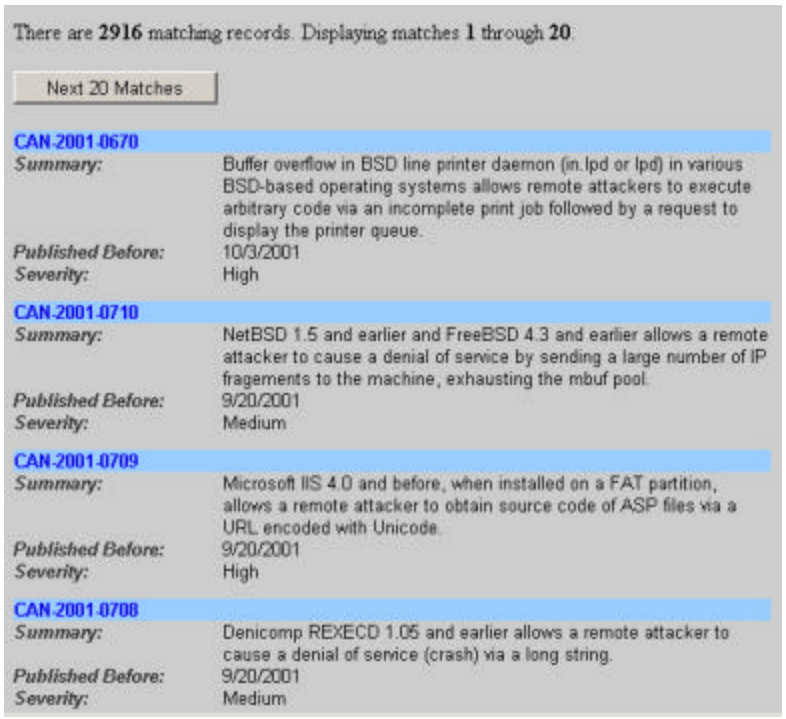


Figure B-17 — ICAT Search Results List

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. Click on the vulnerability name, which is the number in blue, will provide additional details on that particular vulnerability or exposure (see Figure B-18).

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 B-18 — Search Result Detail

- **Vulnerability Name**—ICAT uses the CVE standard vulnerability naming scheme to name each vulnerability. Vulnerabilities will be named either CVE-xxxx-yyyy or CAN-xxxx-yyyy. The CVE prefix is attached to vulnerabilities that have been reviewed by the CVE standard advisory committee. The CAN prefix is 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 while leaving the number the same.

- **Published Before**—This field gives an indication of when a vulnerability was discovered. With older vulnerabilities, the date used was the earliest dating from among publicly available sources. For vulnerabilities released in the year 2001 and beyond, ICAT uses the date on which the vulnerability was added to ICAT.

- **Summary**—Provides a one-line description of the vulnerability. This description is the same description as found in the particular vulnerability's

standard CVE or CAN entry. ICAT provides only this short description of the vulnerability because the purpose of ICAT is to be a searchable index of vulnerabilities instead of a vulnerability database. The purpose of this "summary" and the other vulnerability attributes is to quickly allow users to determine whether or not a particular vulnerability is relevant. Users should use the references found in each ICAT entry to obtain complete descriptions of each vulnerability from a variety of outside resources (see next section).

- **Severity**—ICAT provides a severity field to enable one to quickly judge the impact of a vulnerability. Vulnerabilities can have one of three severity levels: "High severity", "Medium severity", or "Low severity". It is difficult to accurately assign such ratings, as different vulnerabilities will have differing levels of impact depending upon the installed software base of an organization. Despite this problem, 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 some 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 meet the definition of either "high" or "low" severity.
    - A vulnerability is "low severity" if:
        - The vulnerability does not typically yield valuable information or control over a system but instead gives the attacker knowledge that may help the attacker find and exploit other vulnerabilities.
        - The vulnerability is inconsequential for most organizations.
- **Exploitable Range**—A vulnerability can enable either a "local" and/or "remote" attack.
    - Local—Attacks that utilize the vulnerability can be launched directly on the system that is being attacked. The attacker must have 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 local because the attacker did not attack

the telnet server itself but a component only visible to logged in users.

- Remote—Attacks that utilize the vulnerability can be launched across a network against a system without the user having previous access to the system.

- **Loss Type**—Includes the traditional three types ("availability", "confidentiality", and "integrity") plus an additional category called "security protection".

  - Availability—A vulnerability is given the “availability” label if it enables an attack that directly inhibits a user (human or machine) from accessing a particular 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 that can directly steal information from a system.

  - Integrity—A vulnerability is given the “integrity” label if it enables an attack that can directly change the information residing on or passing through a system.

  - Security Protection—A vulnerability is given the “security protection” label if it enables an attack that gives the attacker privileges in a system that the attacker is not allowed to have by the access control policy of the system. The "security protection" label may appear by itself or in three other variations: “security protection (gain superuser access)" when the attack allows a hacker complete control of a system, "security protection (gain user access)" when the attack allows a hacker partial control over a system, “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 one of these properties. If, for example, a vulnerability can give an attacker increased privileges thereby allowing the attacker 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 availability directly, then both corresponding 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:

  - 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 only apply 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, be it 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. Yet another 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. While it could be argued that buffer overflow (discussed next) is a type of boundary overflow error, ICAT puts 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 input being received by a system 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 the 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.

o Access Validation Error—A vulnerability is characterized as a “Access validation error” if a system is vulnerable because the access control mechanism is faulty. The problem lies not with the user controllable configuration of the access control mechanism but with the mechanism itself.

o Exceptional Condition Handling Error—A vulnerability is characterized as an “Exceptional condition handling error” if a system somehow becomes vulnerable due to an exceptional condition that has arisen. The handling (or mishandling) of the exception by the system enables a vulnerability.

o Environmental Error—A vulnerability is characterized as an “Environmental error” if the environment in which a system is installed somehow causes the system to be vulnerable. This may be due, 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 developers 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 a system are set such that the system is vulnerable. This vulnerability is not due to how the system was designed but on how the end user configures the system. ICAT considers it a configuration error when a systems ships from a developer with a weak configuration.

- Race Condition—A vulnerability is characterized as a "Race condition" if a the non-atomicity of a security check causes the existence of a vulnerability. For example, a system checks to see if an operation is allowed by the security model and then performs the operation. However, between the time the security check 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 like writing to the password file.

- Design Error—A vulnerability is characterized as a "Design error" if there exists no errors in the implementation or configuration of a system, but the initial design causes a vulnerability to exist.

- Other—Since the above vulnerability characteristics are not a true classification scheme, it is possible that a vulnerability will not fall in any of them. Any such vulnerability is characterized as vulnerability type "Other."

- **Exposed System Component**—The exposed system component field identifies exactly where in a system the vulnerability occurs. 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 database and patch sites that contain entries about the particular vulnerability being viewed. The referenced sites have no affiliation with 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 viewed CVE entry. This field enables one to search ICAT using non-standard 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. It is this aspect of ICAT that 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 mitigations techniques. ICAT entries typically will list each vulnerable version on a separate line. Each line will contain a vendor name, software name, and vulnerable version number. This information is obtained from various public and private sources. Much of this information is obtained (with permission) from CERT, Security Focus, and ISS X-Force.

Appendix C: Vulnerability Advisory Resources

Federal Vulnerability Advisory Websites[7]

Website	URL
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

[7] Please 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 D: Vulnerability Scanners

Vulnerability Scanning Tools[8]

Tool	Website	Operates on	
		Linux	Win32
CyberCop Scanner	http://www.pgp.com/products/	✓	✓
ISS Internet Scanner	http://www.iss.net/		✓
Nessus	http://www.nessus.org/	✓	✓ (client only)
SAINT	http://www.wwdsi.com/saint/	✓	
SARA	http://www-arc.com/sara/	✓	
SATAN	http://www.fish.com/satan/	✓	

Using Cybercop Vulnerability Scanner

One popular vulnerability scanner is CyberCop Scanner by Network Associates. This commercial vulnerability scanner like most commercial and freeware scanners can identify a numerous common vulnerabilities. It can also identify a limited number of esoteric vulnerabilities. However it does provide a good indication of the overall security stance of a network. Before using any vulnerability scanner, it is critical to get the latest vulnerability database update. All scanners rely on a database of known weaknesses and if this is not the latest version, the scanner will not be able to identify the latest vulnerabilities. To update CyberCop, select "**AutoUpdate**" see Figure D-1.

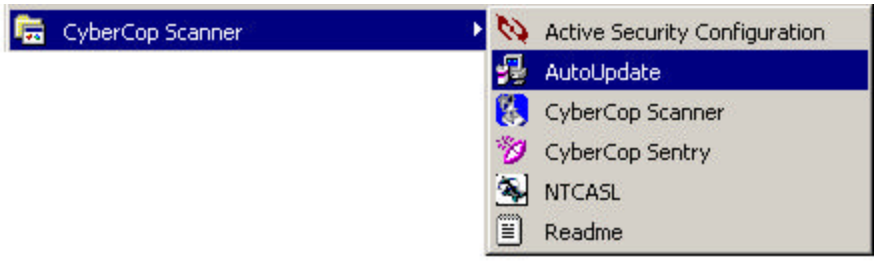


Figure D-1 — CyberCop AutoUpdate

[8] There are number of excellent freeware vulnerability scanners available including several listed in this section. However great care should be used in selecting freely available tools. Generally, freeware/shareware tools should not be used unless an expert has reviewed the source code or they are widely used AND are downloaded from a known safe repository. There are many cases of malicious code being hidden in freeware tools. Even if the tool appears to do what it was advertised to do there is no guarantee that it is not doing something else(such as sending user ids and passwords to a malicious entity). The bottom line is that one can never be too careful when using freeware.

When the CyberCop vulnerability database is updated, start the CyberCop Scanner application by clicking on "**CyberCop Scanner**" in the Windows Start menu (see in Figure D-11 just below AutoUpdate feature). This will open the CyberCop Scanner application (see Figure D-2)

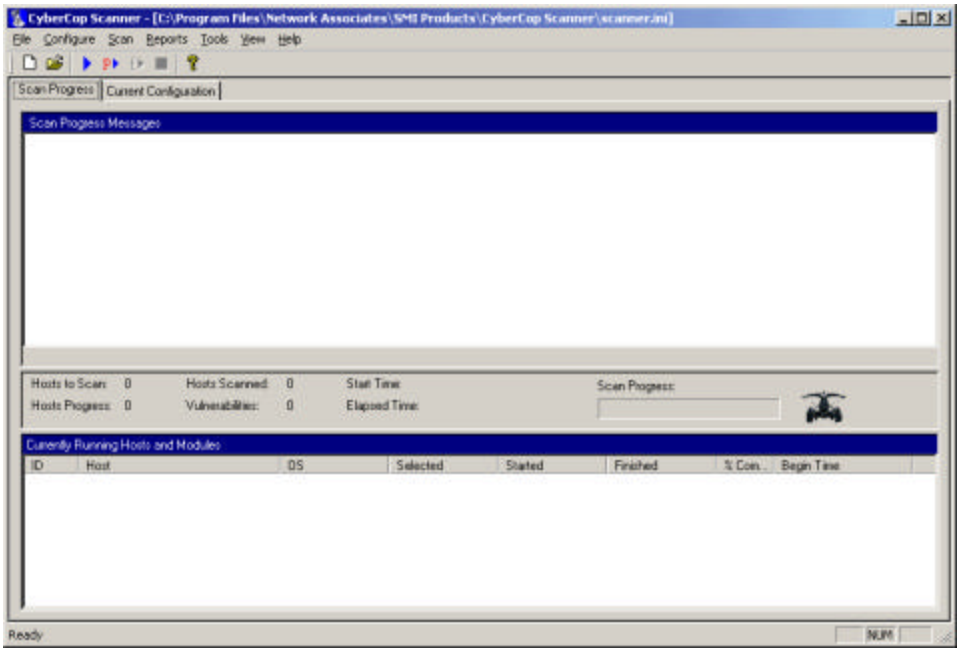


Figure D-2 — CyberCop Scanner Primary Window

Before running the actual scan it will be necessary to select what vulnerabilities to test for and what hosts or subnets to scan. To configure the tests to run:

1. Click on "**Configure**" from the CyberCop menu bar.
2. Click on "**Module Settings.**"

This will open the "Module Configuration Dialog" window (see Figure D-3).

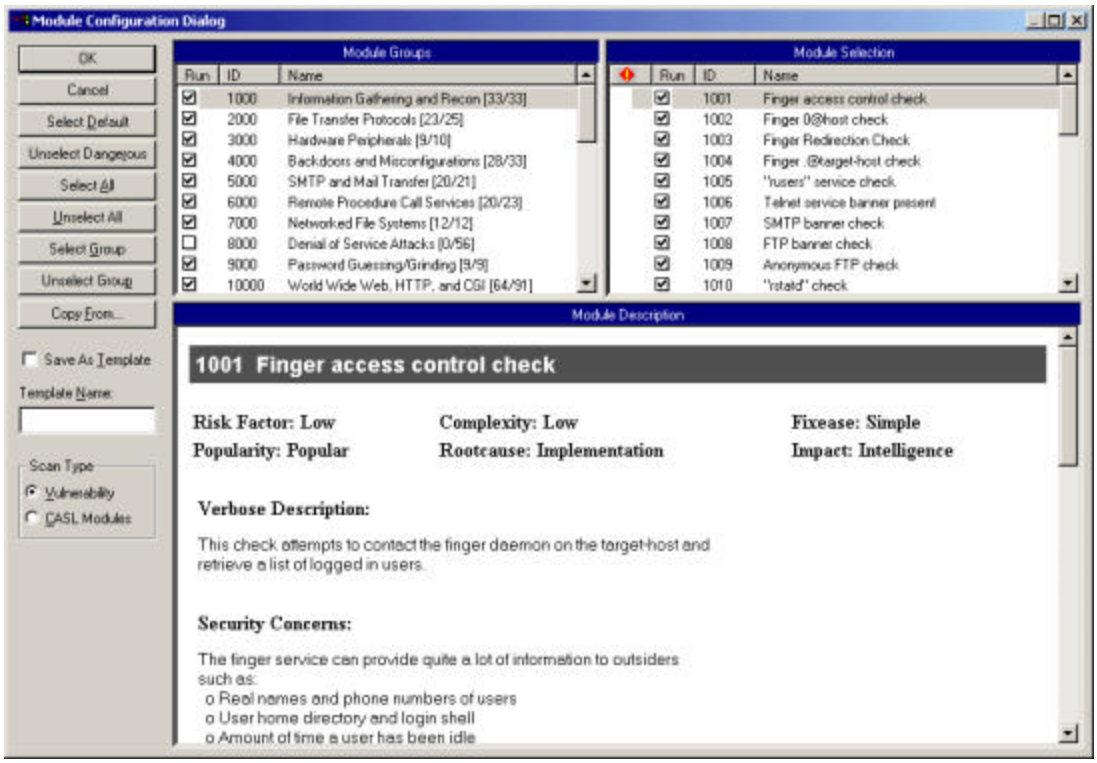


Figure D-3 — Module Configuration Dialog

From this window, a user can select the tests to process. This window has three sub-windows. In the upper left is the "Module Groups" sub-window (CyberCop refers to each test as a "module"). Similar tests are sorted in module groups (e.g., FTP tests, Windows tests, etc.). The "Module Groups" window allows a user to select and deselect entire groups of tests (modules). The "Module Selection" sub-window (upper right) allows the user to select/deselect specific tests. The contents of this window change with the group selected in the "Module Groups" sub-window. The "Module Description" sub-window (lower half) provides a description of the selected module including risk level, complexity, popularity, detailed description of vulnerability and sources for correction (if available). Modules with a next to them are "dangerous." That is they can have a negative impact on the host being scanned (e.g., crash targeted application, denial of service, freeze TCP/IP stack, etc.). To ensure these items are not selected, make sure to click on the "**Unselect Dangerous**" button. This is the recommended setting except for experienced users who fully understand the implications of selecting these tests.

Next it will be necessary to configure the hosts or subnet to scan. To configure this:

1. Click on "**Configure**" from the CyberCop menu bar.
2. Click on "**Scan Settings.**"

This will open the "CyberCop Scanner Setup" window (see Figure D-4).

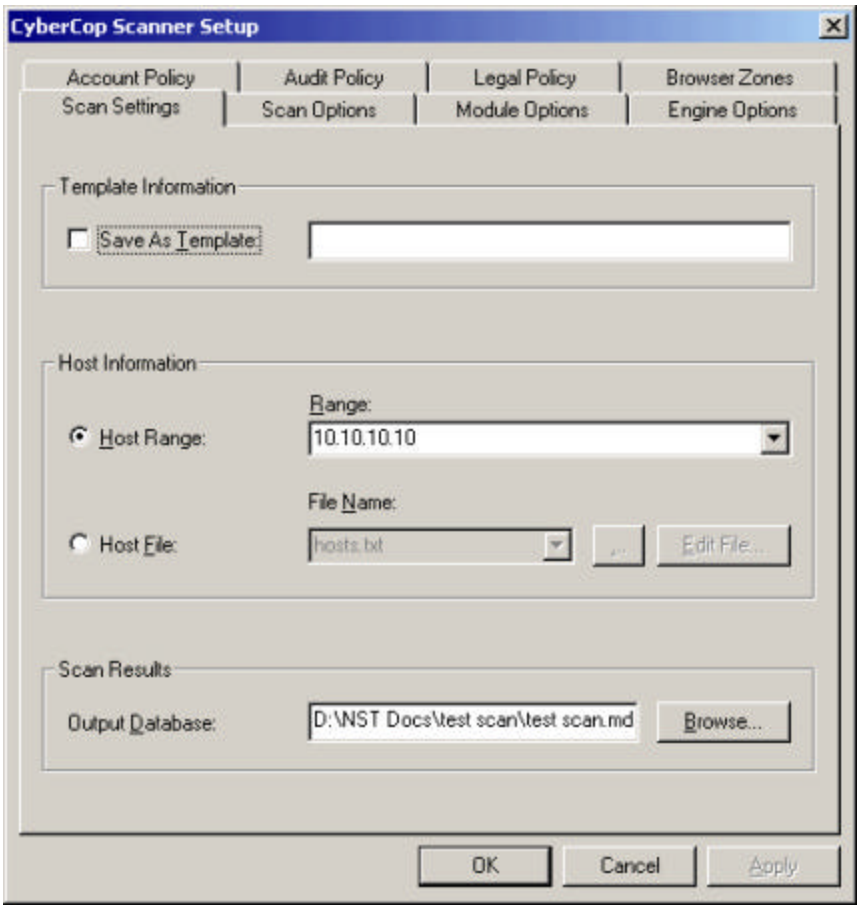
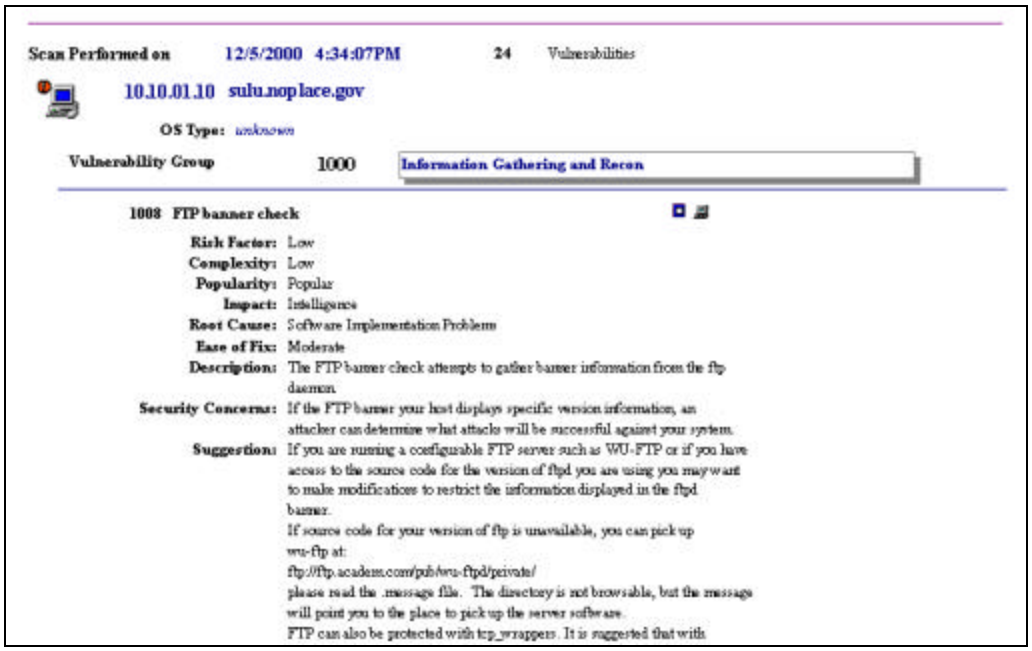


Figure D-4 — CyberCop Scanner Setup

From this Window, the user can select the hosts or subnets to scan and where to store the results. Once these have been selected, they can be saved to a template for future use, if required, and the scan can be started.

The time a scan will take depends on the number and type of tests selected and the number of hosts scanned. It can vary from minutes to hours. A general rule of thumb is approximately 2-10 minutes per host depending on the number of tests (modules) selected. Once the scan is completed, CyberCop will provide a detailed report itemizing the vulnerabilities discovered. This report can be customized to meet the needs of the user. A partial sample of a Cyber Cop report is provided in Figure D-5.



Scan Performed on 12/5/2000 4:34:07PM 24 Vulnerabilities

10.10.01.10 sulu.noplace.gov

OS Type: unknown

Vulnerability Group 1000 Information Gathering and Recon

1008 FTP banner check

Risk Factor: Low
Complexity: Low
Popularity: Popular
Impact: Intelligence
Root Cause: Software Implementation Problem
Ease of Fix: Moderate
Description: The FTP banner check attempts to gather banner information from the ftp daemon.
Security Concerns: If the FTP banner your host displays specific version information, an attacker can determine what attacks will be successful against your system.
Suggestion: If you are running a configurable FTP server such as WU-FTP or if you have access to the source code for the version of ftpd you are using you may want to make modifications to restrict the information displayed in the ftpd banner.
If source code for your version of ftp is unavailable, you can pick up wu-ftp at:
ftp://ftp.academ.com/pub/wu-ftpd/private/
please read the .message file. The directory is not browsable, but the message will point you to the place to pick up the server software.
FTP can also be protected with tcp_wrappers. It is suggested that with

Figure D-5 — CyberCop Report Partial Sample

Appendix E: Using 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 a user to scan their computer to find any updates that are available at that time from Microsoft and other participating vendors. Figures E-1 and E-2 demonstrate the two different methods of accessing the Windows Update utility. It is suggested that users close all other applications before initiating the Windows Update feature.

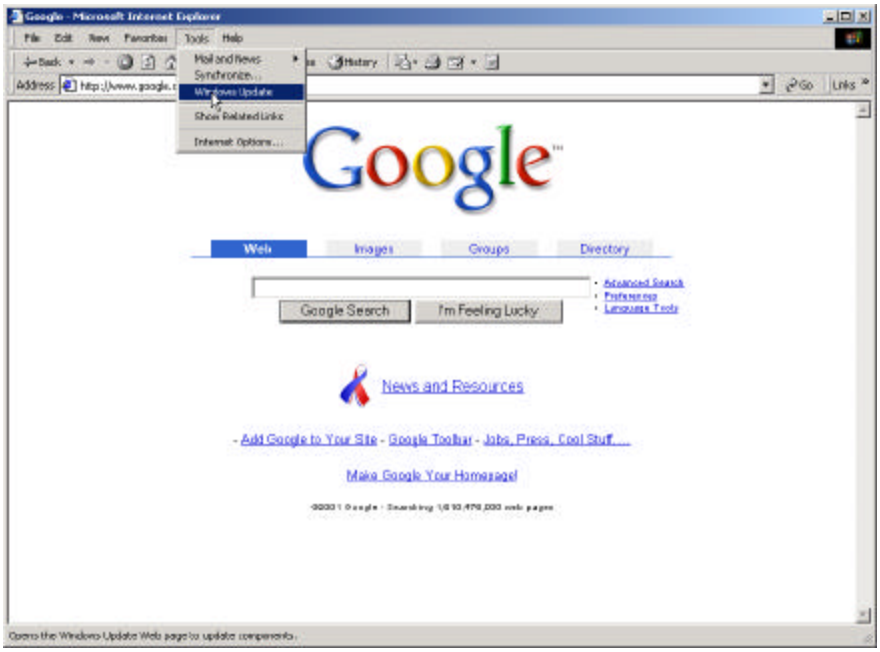


Figure E-1 — Accessing Windows Update Though Internet Explorer

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 of these options will launch Microsoft Internet Explorer (if it is not already active) and go the Microsoft Windows Update website (htttp://windowsupdate.microsoft.com). See Figure E-3 for the Windows Update homepage.

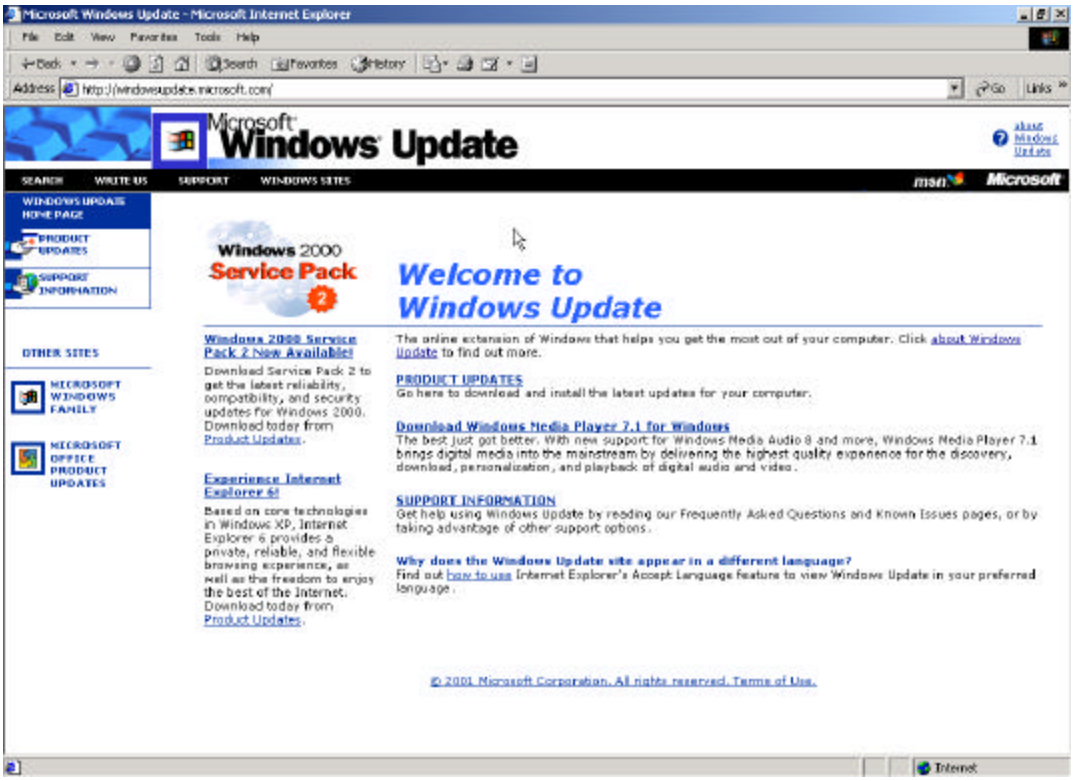


Figure E-3 — Windows Update Homepage

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

Figure E-4 — Windows Update Scan

Once Windows Update has finished scanning the users machine, it will generate a list of recommended updates (see Figure E-5). Users can browse the list, decide which components they want, and download them right to their computer.

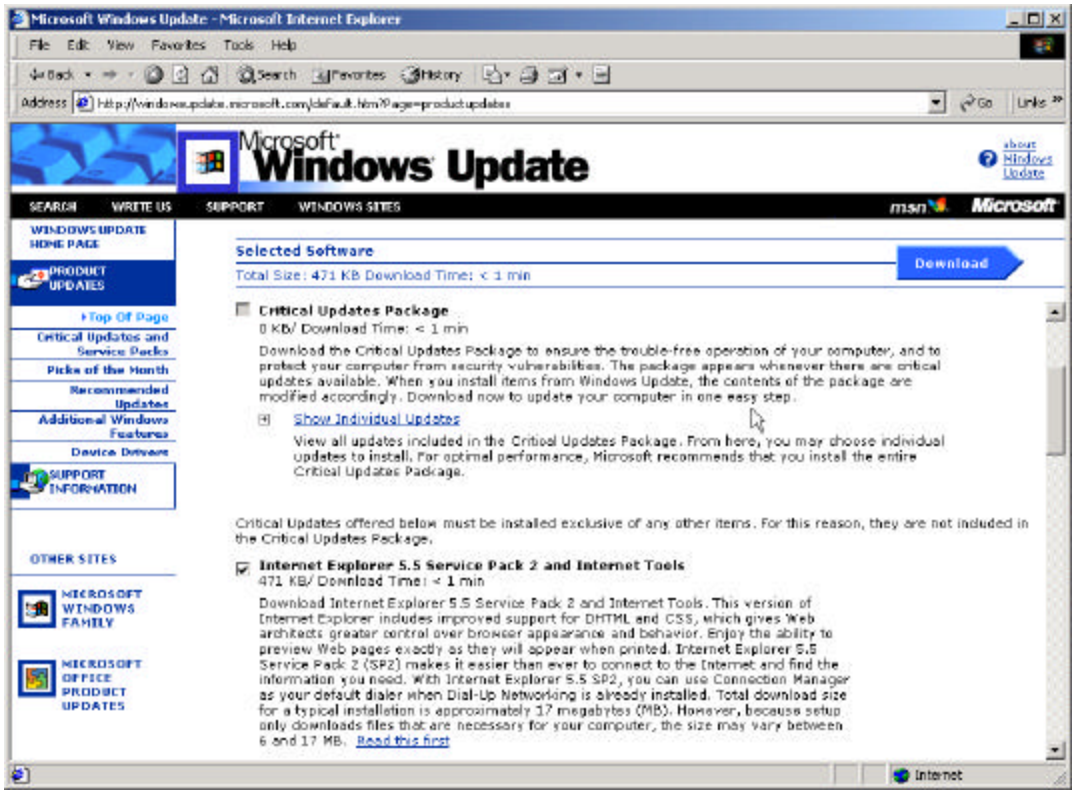


Figure E-5 — Windows Update Recommend Updates

The product updates are broken down into five different sections:

- **Critical Updates and Service Packs**—It is generally suggested that users download all Critical Update Packages as these fix now known problems (often security issues) with their specific installation.

- **Picks of the Month**—These are new releases that 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, etc.).

- **Device Drivers**—Listed here will be any updated device drivers for you computer. A device driver is a program that controls a piece of hardware (such as a printer, monitor, disk drive, or video card) that is attached to your computer. Note: Third parties manufacture most hardware and device drivers for this hardware will not be listed here unless the manufacturer has an agreement with Microsoft. Generally a user should go to the appropriate manufacturer's website to get device driver updates.

Certain updates can only be downloaded individually. If this is the case, Window Update will provide notification as shown in Figure E-6. If this happens the user will have to repeat the process delineated here.

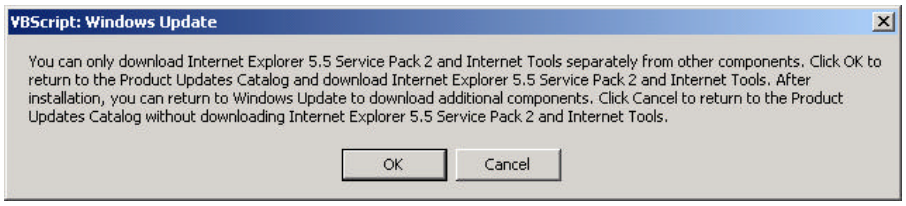


Figure E-6 — Windows Update Multiple Downloads not Permitted Warning

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

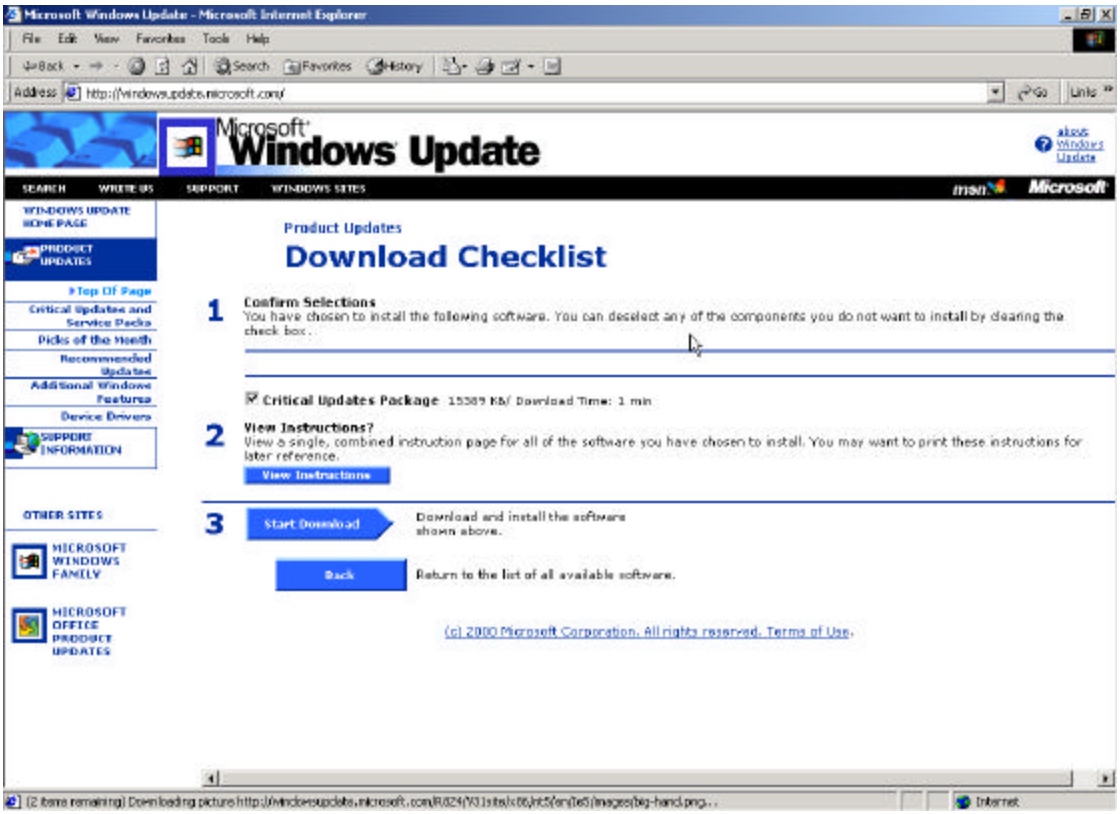


Figure E-7 — Windows Update Download Checklist

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

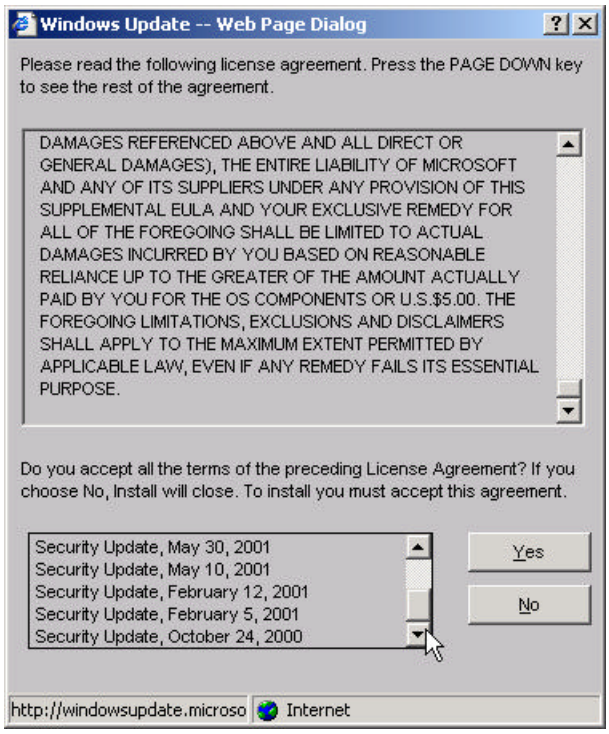


Figure E-8 — Windows Update Confirmation and License Agreement

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 files size of the software selected and connection speed.

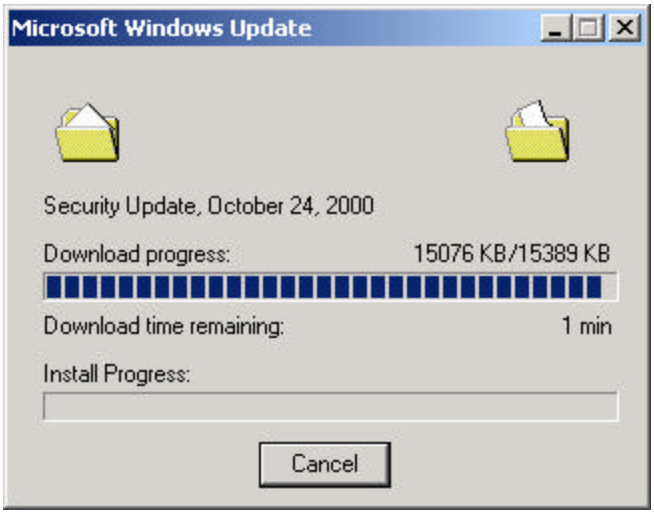


Figure E-9 — Windows Update Download Status Window

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).

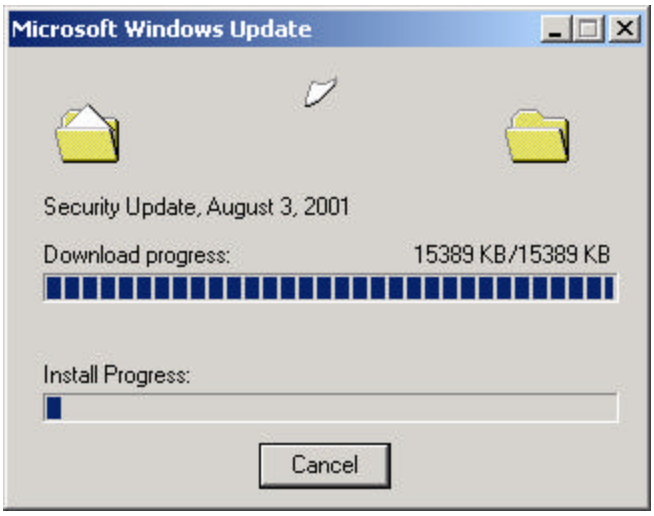


Figure E-10 — Windows Update Install Status Window

Once the install is successfully completed that browser Window will confirm the success (see Figure E-11).

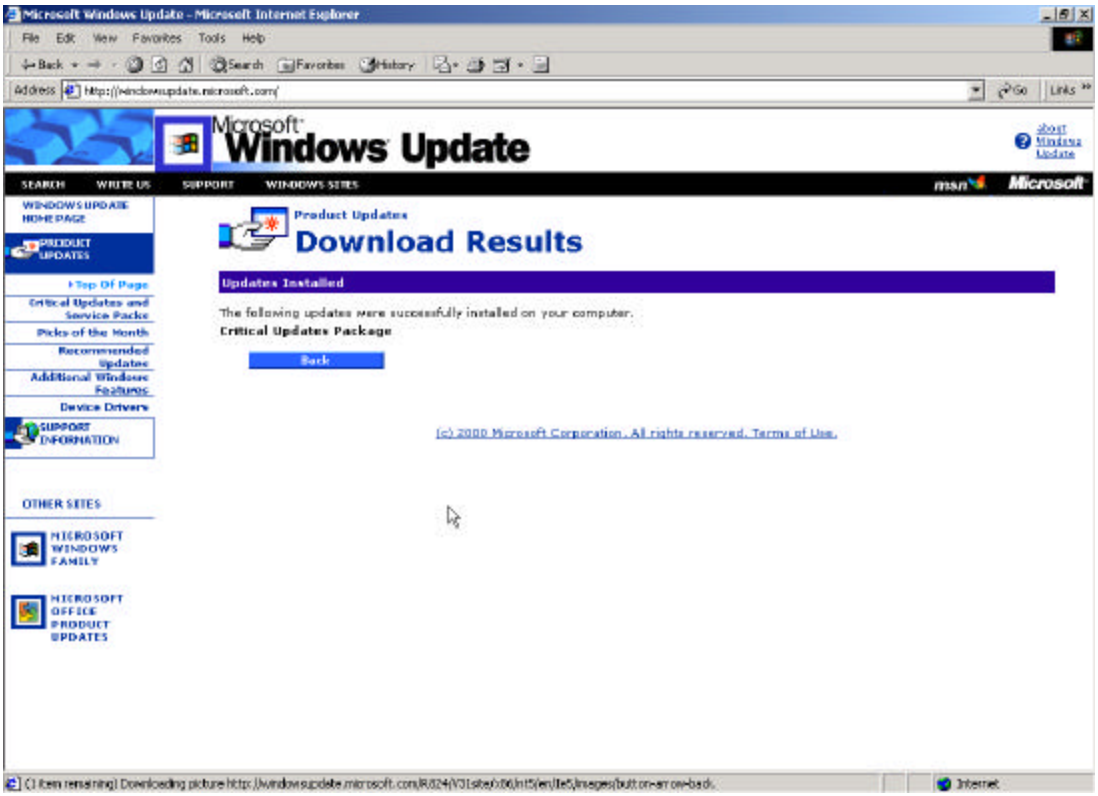


Figure E-11 — Windows Update Install Success Confirmation Window.

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 if you do not wish to reboot immediately (changes will NOT take effect until a the computer has

successful rebooted). If Window update does not prompt for a reboot, then the changes to not require it and are effective from the time of a successful install (see Figure E-11)

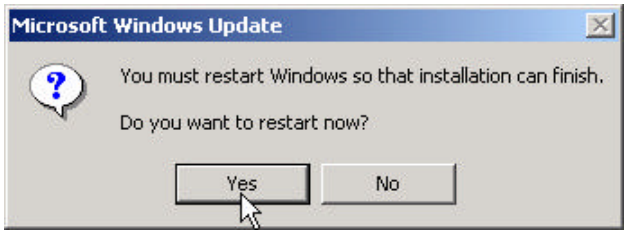


Figure E-12 — Windows Update Restart Dialog Box

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

Appendix F: Using 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 their Windows NT/2000 computer and receive a detailed customized report of their computer's security settings and patch level along with recommendations for updates and improvements.

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

- Password Strength and Length
- Services
- 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. For users running Internet Information Server (IIS) or Personal Web Services (PWS) on their system, its is recommended that they use the HFNetChk (see Appendix G) tool to check for appropriate web server patches.

An example scan using MPSA is documented in the following example. To access the MPSA site enter its 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).

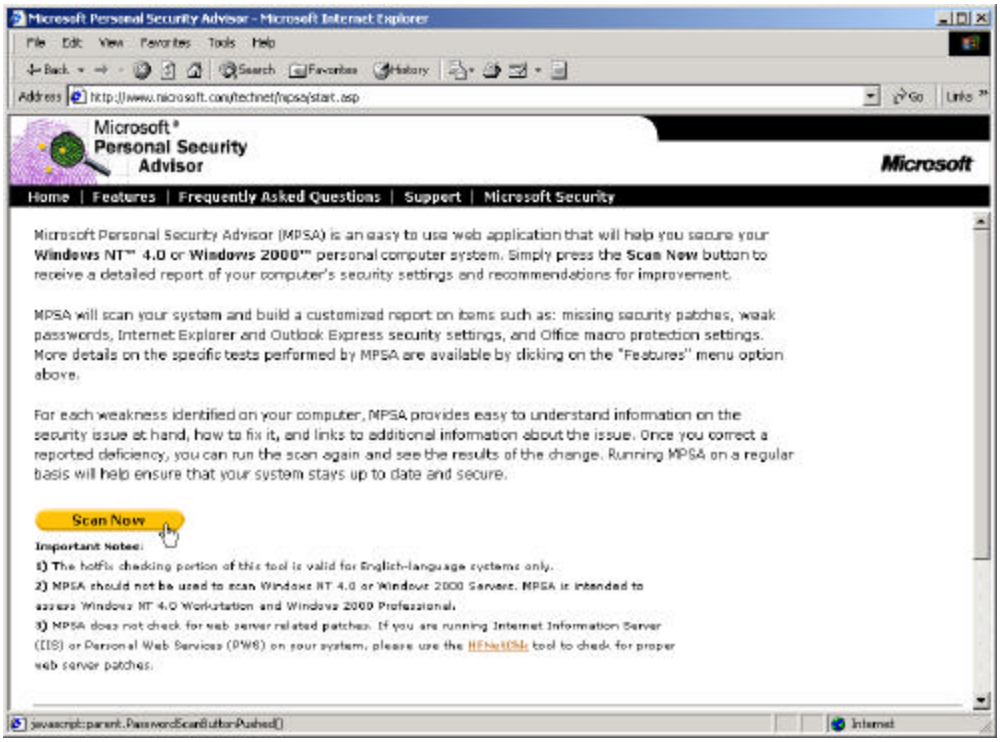


Figure F-1 — MPSA Homepage

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

Figure F-2 — MPSA Active-X Control Security Warning

Once a user selects yes, the Active X control is downloaded (should take less than a minute) and it initiates and processes its scan (see Figures F-3 and F-4).

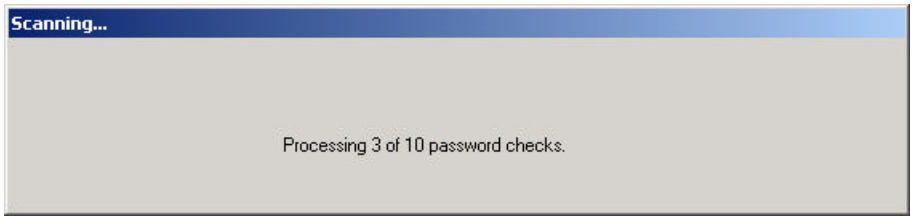


Figure F-3 — MPSA Active Scanning

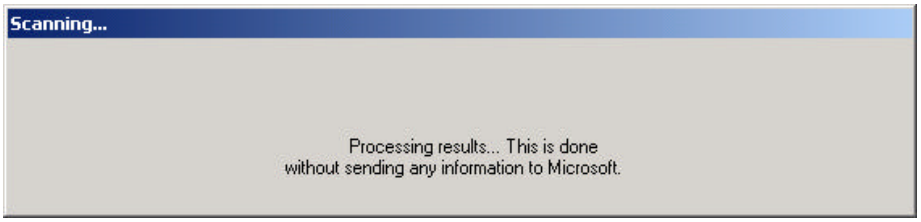


Figure F-4 — MPSA Scan Processing

Once the scan is completed MPSA will generate 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.

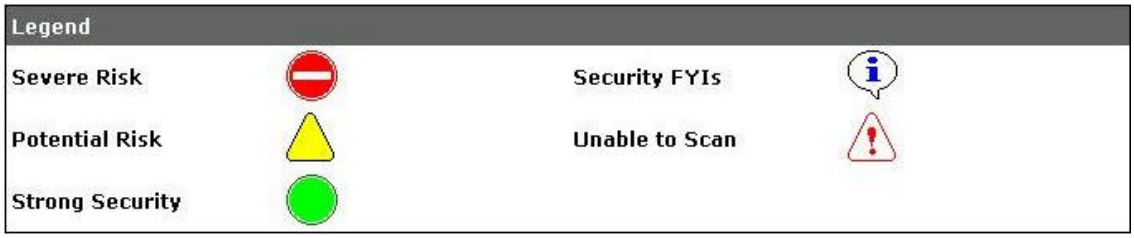


Figure F-5 — MPSA Report Legend

- **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, change is highly recommended.
- **Strong Security**—No vulnerability is detected in the scanned system.
- **Security FYIs**—An issue that may be a vulnerability but requires further investigation.
- **Unable to Scan**—The MPSA was unable to scan for this item.

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

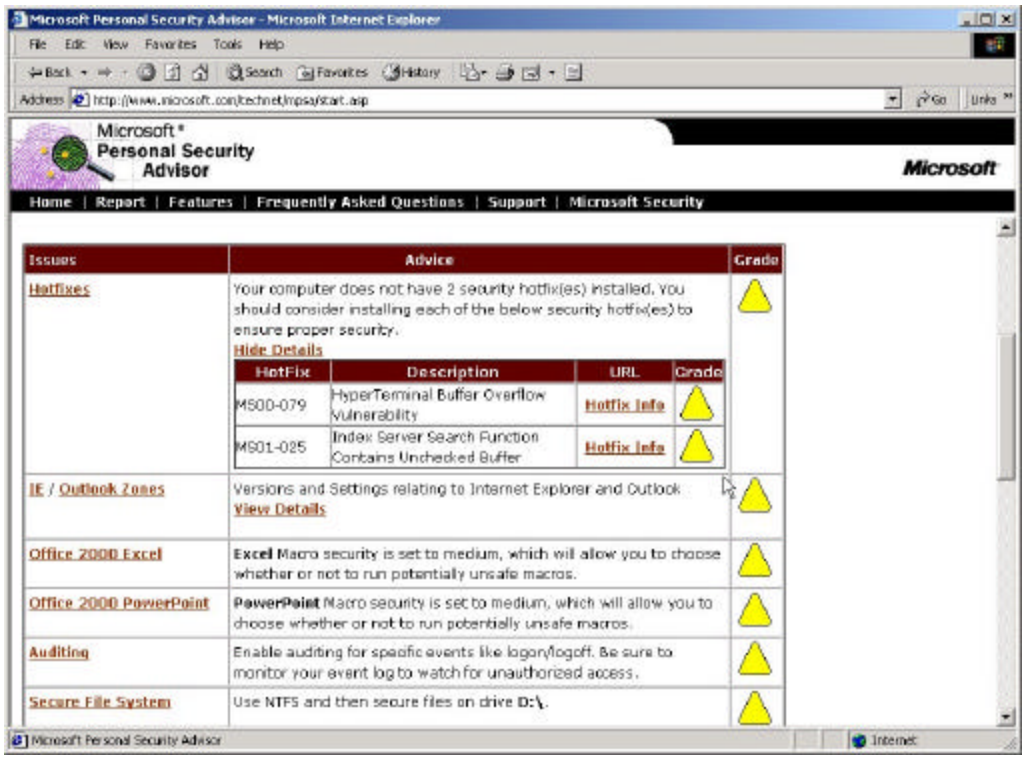


Figure F-6 — Final Report

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

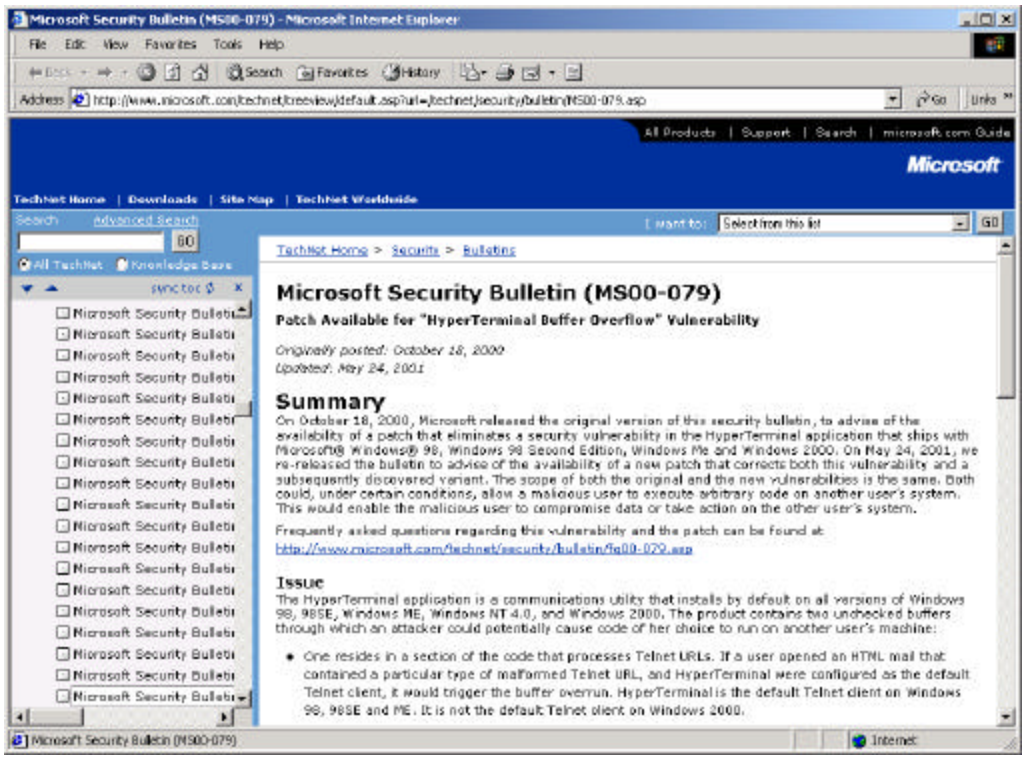


Figure F-7 — Final Report Details

An important point to note when using the MPSA scanner is that it must be run again after any vulnerability is patched. While all steps may be correctly implemented to patch a system, there still exists the possibility that one patch may "unfix" another. For this reason it is important to always scan again to double check for correct operation.

Appendix G: Using 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 as well as hotfixes for Internet Information Services (IIS) 4.0 and 5.0, SQL Server 7.0 and 2000, and Internet Explorer 5.01 and later. While less 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 Extend 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.

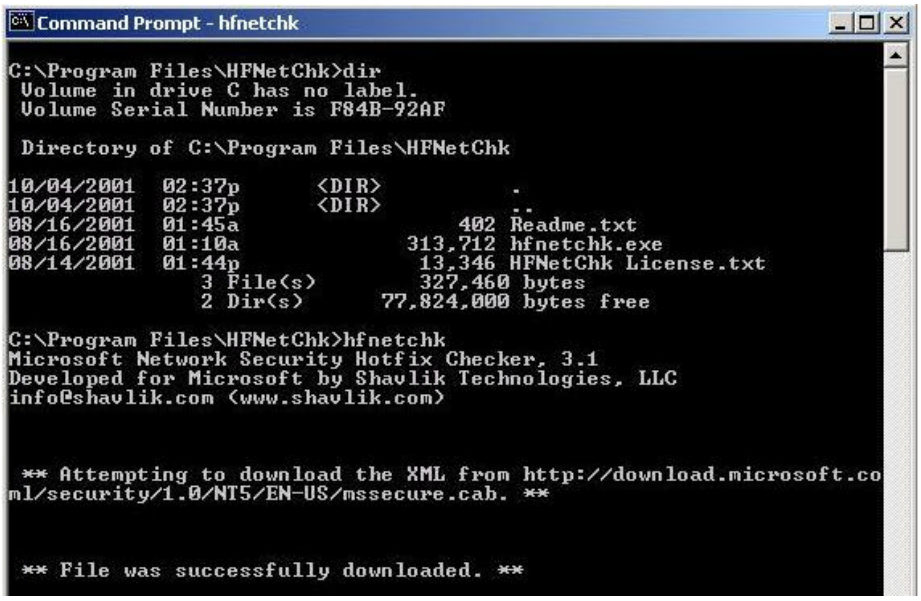


Figure G-1 — Running Hfnetchk

Before continuing with the actual scan, you must agree to the installation and running of the downloaded XML file (see Figure G-2).

Figure G-2 — Installing XML File

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

Figure G-3 — Hfnetchk Output

Although the listed information can be useful, the format and lack of additional information can leave the user desiring more. To correct this deficiency there is a freeware tool Maximized Software Hotfix Reporter comes in to play. This utility works in conjunction with hfnetchk to display the results in an easy to read HTML format with hyperlinks making 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).

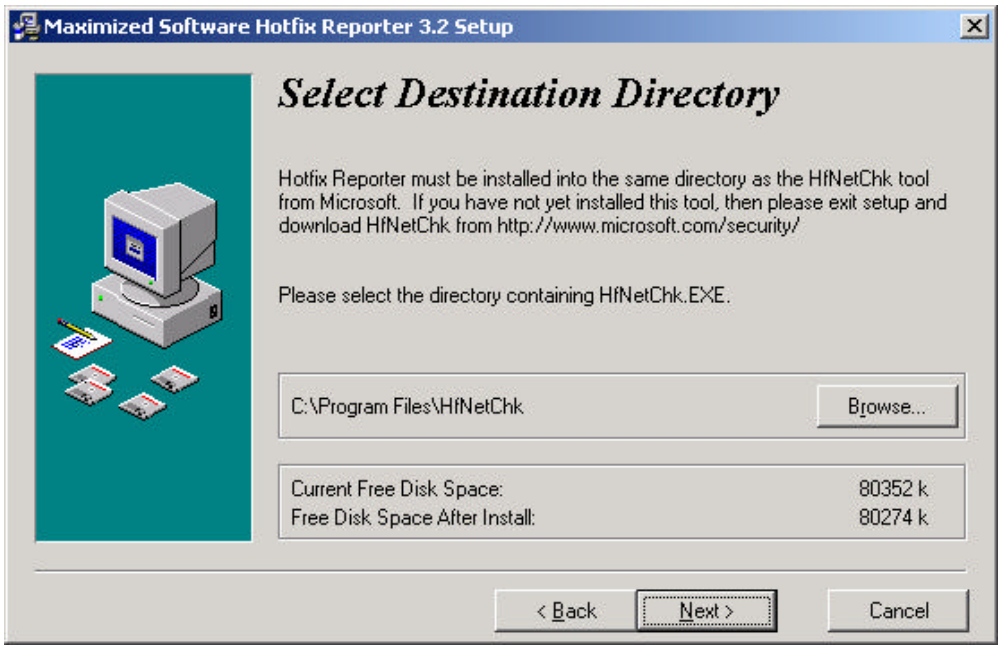


Figure G-5 — Installing Hotfix Reporter

After installation, launch the Hotfix Reporter from the Windows Start menu. The execution of the Hotfix reporter will very similar to that of HfNetChk since they work together (See Figure G-6).

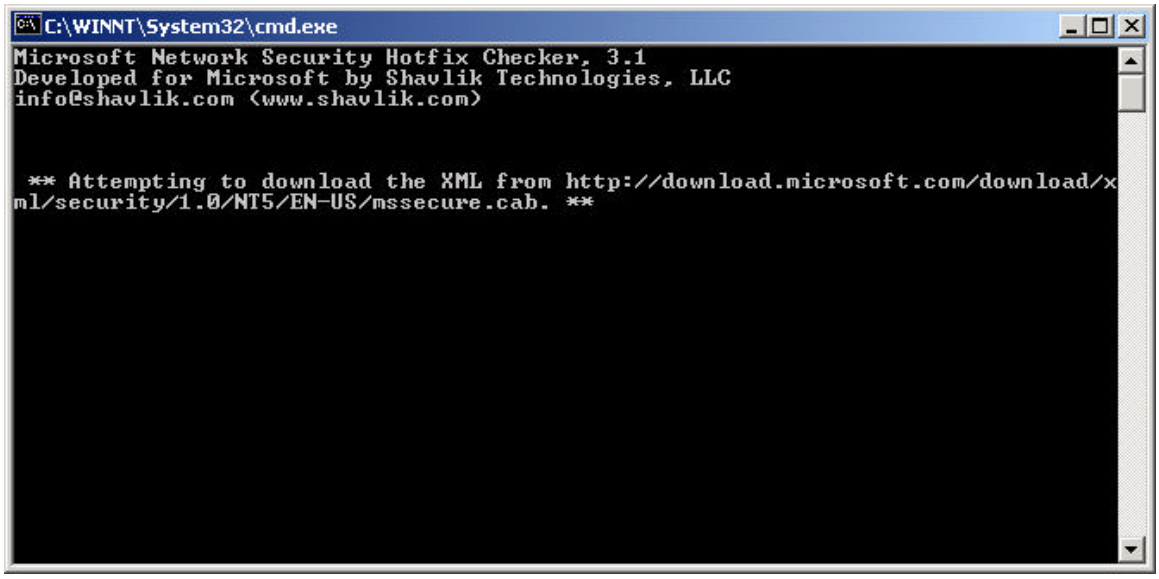


Figure G-6 — Running HfNetChk with Hotfix Reporter

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-6 — Installing Microsoft XML Data File

Once HfNetChk has completed it scan, the Hotfix Reporter will then open a Hypertext Markup Language (HTML) file in the default web browser. This provides results in a more readable and usable format than that of HfNetChk alone. These

results are also stored locally and can be reviewed at a later date. As with the MPSA, it is important to run the Hotfix Reporter again after any and all updates are run to ensure that nothing has been undone.