



The Twenty Most Critical Internet Security Vulnerabilities (Updated) ~ The Experts' Consensus
Version 2.6 October 1, 2002 Copyright © 2001-2002, The SANS Institute

www.fbi.gov

www.nipc.gov

www.sans.org

----- [Jump To Index of Top 20 Threats](#) -----

Introduction

The majority of the successful attacks on operating systems come from only a few software vulnerabilities. This can be attributed to the fact that attackers are opportunistic, take the easiest and most convenient route, and exploit the best-known flaws with the most effective and widely available attack tools. They count on organizations not fixing the problems, and they often attack indiscriminately, scanning the Internet for any vulnerable systems. System compromises in the Solar Sunrise Pentagon hacking incident, for example, and the easy and rapid spread of the Code Red and NIMDA worms can be traced to exploitation of unpatched vulnerabilities.

Two years ago, the SANS Institute and the National Infrastructure Protection Center (NIPC) released a document summarizing the Ten Most Critical Internet Security Vulnerabilities. Thousands of organizations used that list, and the expanded Top Twenty, which followed a year later, to prioritize their efforts so they could close the most dangerous holes first. The vulnerabilities that led to all three examples above - the Solar Sunrise Pentagon incident, and the Code Red and NIMDA worms - are on that list.

This updated SANS/FBI Top Twenty is actually two Top Ten lists: the ten most commonly exploited vulnerable services in Windows, and the ten most commonly exploited vulnerable services in Unix. Although there are thousands of security incidents each year affecting these operating systems, the overwhelming majority of successful attacks target one or more of these twenty services.

While experienced security administrators will find the Top Twenty to be a valuable resource in their arsenal, the list is especially intended for those organizations that lack the resources to train, or those without technically-advanced security administrators. The individuals with responsibility for networks in those organizations often report that they have not corrected many of these flaws because they simply do not know which vulnerabilities are most dangerous, they are too busy to correct them all, or they do not know how to correct them safely. Traditionally, auditors and security managers have used vulnerability scanners to search for five hundred or a thousand or even two thousand very specific vulnerabilities, blunting the focus administrators need to ensure that all systems are protected against the most common attacks. When a system administrator receives a report showing thousands of vulnerabilities across hundreds of machines, he is often paralyzed.

The Top Twenty is a prioritized list of vulnerabilities that require immediate remediation. The list is sorted by service because in many cases a single remedy -- disabling the service, upgrading to the most recent version, applying a cumulative patch -- can quickly solve dozens of specific software flaws, which might show up on a scanner. This list is designed to help alleviate that problem by combining the knowledge of dozens of leading security experts. They come from the most security-conscious federal agencies, the leading security software vendors and consulting firms, the top university-based security programs, and CERT/CC and the SANS Institute. A list of participants may be found at the end of this document.

The SANS/FBI Top Twenty is a living document. It includes step-by-step instructions and pointers to additional information useful for correcting the security flaws. We will update the list and the instructions as more critical threats and more current or convenient methods are identified, and we welcome your input along the way. This

Useful Related Resources

- [FBI/GSA/SANS/British NISCC/Canadian OCIEP Press Release on the Top Twenty](#)
- [Tools that Test for the Top Twenty](#)
- [Testing for the Top Twenty Internet Security Vulnerabilities](#)
- [Staying Current: The Critical Vulnerability Assessment \(e-mail every Monday, free\)](#)
- [Monitoring All New Vulnerabilities \(email every Thursday, free\)](#)
- [Upcoming Training programs for Hardening Windows or UNIX Systems](#)
- [GISRA Scanning Requirements and NASA Case Study](#)
- [SANS/FBI Top 20 List, October 2001](#)
- [Air Force CIO John Gilligan's remarks at 2001 Top 20 Announcement](#)
- [SANS/FBI Original Top 10 List, July 2000](#)

Weekly Update of Critical New Vulnerabilities

- 10/01/02 - Version 2.6 Posted

is a community consensus document -- your experience in fighting attackers and in eliminating the vulnerabilities can help others who come after you. Please send suggestions via e-mail to info@sans.org with the subject "Top Twenty Comments."

Notes For Readers:

CVE Numbers

You'll find references to CVE (Common Vulnerabilities and Exposures) numbers accompanying each vulnerability. You may also see CAN numbers. CAN numbers are candidates for CVE entries that have not yet been fully verified. For more data on the award-winning CVE project, see <http://cve.mitre.org>.

The CVE and CAN numbers reflect the top priority vulnerabilities that should be checked for each item. Each CVE vulnerability reference is linked to the associated vulnerability entry in the National Institute of Standards and Technology's ICAT vulnerability indexing service (<http://icat.nist.gov>). ICAT provides a short description of each vulnerability, a list of the characteristics of each vulnerability (e.g. associated attack range and damage potential), a list of the vulnerable software names and version numbers, and links to vulnerability advisory and patch information.

Ports to Block at the Firewall

At the end of the document, you'll find an extra section offering a list of the ports used by commonly probed and attacked services. By blocking traffic to these ports at the firewall or other network perimeter protection devices, you add an extra layer of defense that helps protect you from configuration mistakes. Note, however, that using a firewall to block network traffic directed to a port does not protect the port from disgruntled co-workers who are already inside your perimeter, or from hackers who may have penetrated your perimeter using other means.

[Back to Top ^](#)

Top Vulnerabilities to Windows Systems

- [W1 Internet Information Services \(IIS\)](#)
- [W2 Microsoft Data Access Components \(MDAC\) -- Remote Data Services](#)
- [W3 Microsoft SQL Server](#)
- [W4 NETBIOS -- Unprotected Windows Networking Shares](#)
- [W5 Anonymous Logon -- Null Sessions](#)
- [W6 LAN Manager Authentication -- Weak LM](#)

Hashing

- [W7 General Windows Authentication -- Accounts with No Passwords or Weak Passwords](#)
- [W8 Internet Explorer](#)
- [W9 Remote Registry Access](#)
- [W10 Windows Scripting Host](#)

Top Vulnerabilities to Unix Systems

- [U1 Remote Procedure Calls \(RPC\)](#)
- [U2 Apache Web Server](#)
- [U3 Secure Shell \(SSH\)](#)
- [U4 Simple Network Management Protocol \(SNMP\)](#)
- [U5 File Transfer Protocol \(FTP\)](#)
- [U6 R-Services -- Trust Relationships](#)
- [U7 Line Printer Daemon \(LPD\)](#)
- [U8 Sendmail](#)
- [U9 BIND/DNS](#)
- [U10 General Unix Authentication -- Accounts with No Passwords or Weak Passwords](#)

[Back to Top ^](#)

Top Vulnerabilities to Windows Systems (W)

W1 Internet Information Services (IIS)

W1.1 Description

IIS is prone to vulnerabilities in three major classes: failure to handle unanticipated requests, buffer overflows, and sample applications. Each will be addressed briefly here.

1. *Failure to Handle Unanticipated Requests.* Many IIS vulnerabilities involve a failure to handle improperly (or just deviously) formed HTTP requests. A well-known example is the Unicode directory traversal vulnerability, which was exploited by the Code Blue worm. By crafting a request to exploit one of these vulnerabilities, a remote attacker may:
 - View the source code of scripted applications.
 - View files outside of the Web document root.
 - View files the Web server has been instructed not to serve.
 - Execute arbitrary commands on the server (resulting in, for example, deletion of critical files or installation of a backdoor).
2. *Buffer Overflows.* Many ISAPI extensions (including the ASP, HTR, IDQ, PRINTER, and SSI extensions) are vulnerable to buffer overflows. A well-known example is the .idq ISAPI extension vulnerability, which

was exploited by the Code Red and Code Red II worms. A carefully crafted request from a remote attacker may result in:

- Denial of service.
 - Execution of arbitrary code and/or commands in the Web server's user context (e.g., as the IUSR_servername or IWAM_servername user).
3. *Sample Applications.* Sample applications are generally designed to demonstrate the functionality of a server environment, not to withstand attacks, and are not intended to serve as production applications. Combined with the facts that their default location is readily known and their source code is readily available for scrutiny, this makes them prime exploit targets. The consequences of such exploits can be severe; for example:
- A sample application, newdsn.exe, allowed the remote attacker to create or overwrite arbitrary files on the server.
 - A number of such applications allow remote viewing of arbitrary files, which may be used to gather information such as database userids and passwords.
 - An iisadmin application, ism.dll, allows remote access to sensitive server information including the Administrator's password.

W1.2 Operating Systems Affected

- Windows NT 4 (any flavor) running IIS 4
- Windows 2000 Server running IIS 5
- Windows XP Professional running IIS 5.1

W1.3 CVE Entries

[CVE-2001-0241](#), [CVE-2001-0333](#), [CVE-2001-0500](#), [CAN-2002-0079](#), [CVE-2000-0884](#), [CVE-2000-0886](#), [CAN-2002-0071](#), [CAN-2002-0147](#), [CAN-2002-0150](#), [CAN-2002-0364](#), [CAN-2002-0149](#), [CVE-1999-0191](#), [CAN-1999-0509](#), [CVE-1999-0237](#), [CVE-1999-0264](#), [CVE-2001-0151](#), [CAN-1999-0736](#), [CVE-1999-0278](#), [CAN-2002-0073](#), [CVE-2000-0778](#), [CVE-1999-0874](#), [CVE-2000-0226](#), [CAN-1999-1376](#), [CVE-2000-0770](#), [CVE-2001-0507](#)

W1.4 How to Determine if you are Vulnerable

Given the number of vulnerabilities, some of which are addressed only in a cumulative security roll-up package from Microsoft, it is simplest to presume that you are vulnerable if the cumulative roll-up has not been applied. To determine whether the cumulative roll-up has been applied on your server, check the registry for the entry listed for your platform below.

Windows NT 4:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Hotfix\Q319733

Windows NT 4 Terminal Server Edition:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Hotfix\Q317636

Windows 2000:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates\Windows 2000\SP3\Q319733

Windows XP:

- HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Updates\Windows XP\SP1\Q319733

Alternatively, you may use HFNetChk (see "Stay Current" under W1.5) to verify the presence of the corresponding patch:

- NT 4: Q319733
- NT 4 Terminal Server Edition: Q317636

- 2000 or XP: Q319733

You are probably vulnerable to sample application exploits if any of the following files resides in your %wwwroot%/scripts directory (e.g., C:\inetpub\wwwroot\scripts or D:\web\scripts) or any subdirectory thereof:

- code.asp
- codebrws.asp
- ism.dll
- newdsn.exe
- viewcode.asp
- winmsdp.exe

W1.5 How to Protect Against It

1. *Apply the current patches.* In the case of IIS 4 on NT 4 with Service Pack 6a, this means applying a cumulative security roll-up package and a single hotfix. In the case of IIS 5 or 5.1 on Windows 2000 or XP (respectively), the roll-up and the hotfix are included in service packs. URLs are provided below.

IIS 4 on NT 4:

- Service Pack 6a: <http://www.microsoft.com/ntserver/nts/downloads/recommended/SP6/allSP6.asp>
- Security Rollup: <http://www.microsoft.com/ntserver/nts/downloads/security/q319733/>
- Hotfix: <http://www.microsoft.com/ntserver/nts/downloads/security/q321599/>

IIS 4 on NT 4 Terminal Server Edition:

- Service Pack 6:
<http://www.microsoft.com/ntserver/terminalserver/downloads/recommended/tsesp6/>
- Security Rollup: <http://www.microsoft.com/ntserver/terminalserver/downloads/critical/q317636/>
- Hotfix: <http://www.microsoft.com/ntserver/nts/downloads/security/q321599/>

IIS 5 on Windows 2000:

- Service Pack 3: <http://www.microsoft.com/windows2000/downloads/servicepacks/sp3/>

IIS 5.1 on Windows XP:

- Service Pack 1: <http://www.microsoft.com/WindowsXP/pro/downloads/servicepacks/sp1/>

2. *Stay Current.* These service packs, rollup patches and hotfixes only remedy vulnerabilities that are already known. As new IIS weaknesses are uncovered, you will need to patch accordingly. HFNetChk, the Network Security Hotfix Checker, assists the system administrator in scanning local or remote systems for current patches. The tool works on Windows NT 4, Windows 2000, and Windows XP. The current version can be downloaded from Microsoft at <http://www.microsoft.com/technet/security/tools/hfnetchk.asp>.
3. *Eliminate Sample Applications.* Sample applications, including the iisadmin tool, may be used to verify that a server installation works as expected, but should be deleted immediately thereafter. These applications can be found in the %wwwroot%/scripts directory. Ideally, however, the administrator should choose not to install the sample applications and Web-based administration tools at all.
4. *Unmap Unnecessary ISAPI Extensions.* Most IIS deployments have no need for most of the ISAPI extensions that are mapped by default, particularly .htr, .idq, .ism, and .printer. All unused ISAPI extensions should be unmapped. This can be done by hand through the Internet Services Manager, but the IIS Lockdown Wizard from Microsoft will also do the job. The current version can be downloaded from Microsoft at <http://www.microsoft.com/technet/security/tools/locktool.asp>.
5. *Filter HTTP Requests.* Many IIS exploits, including Code Blue and the Code Red family, use maliciously formed HTTP requests in directory traversal or buffer overflow attacks. The URLScan filter can be configured to reject such requests before the server attempts to process them. The current version has been integrated into the IIS Lockdown Wizard, but can be downloaded separately from Microsoft at

W2 Microsoft Data Access Components (MDAC) -- Remote Data Services

W2.1 Description

The Remote Data Services (RDS) component in older versions of Microsoft Data Access Components (MDAC) has a program flaw which allows remote users to run commands locally with administrative privilege. Combined with a flaw in Microsoft Jet database engine 3.5 (part of MS Access), this exploit may also provide anonymous external access to internal databases. These flaws are well-documented and solutions have been available for more than two years, but outdated or misconfigured systems remain exposed and subject to attack.

W2.2 Operating Systems Affected

Most Microsoft Windows NT 4.0 systems running IIS 3.0 or 4.0, Remote Data Services 1.5, or Visual Studio 6.0.

W2.3 CVE Entries

[CVE-1999-1011](#) W2.4 How to Determine if you are Vulnerable

If you are running Microsoft Windows NT 4.0 and IIS 3.0 or 4.0, then check for the existence of "msadcs.dll" (this is typically installed in "C:\Program Files\Common Files\System\Msadc\msadcs.dll", but that may vary depending on your system).

W2.5 How to Protect Against It

An excellent guide to the RDS and Jet weaknesses and how to correct them is available at <http://www.wiretrip.net/rfp/p/doc.asp?id=29&iface=2>.

Microsoft has also issued several security bulletins detailing this exploit and how to repair it via configuration changes:

- <http://support.microsoft.com/support/kb/articles/q184/3/75.asp>
- <http://www.microsoft.com/technet/security/bulletin/ms98-004.asp>
- <http://www.microsoft.com/technet/security/bulletin/ms99-025.asp>

Alternatively, you can prevent this problem by upgrading to MDAC version 2.1 or greater (although this may introduce compatibility issues). The most recent MDAC versions are available at <http://www.microsoft.com/data/download.htm>

W3 Microsoft SQL Server

W3.1 Description

The Microsoft SQL Server (MSSQL) contains several serious vulnerabilities that allow remote attackers to obtain sensitive information, alter database content, compromise SQL servers, and, in some configurations, compromise server hosts.

MSSQL vulnerabilities are well-publicized and actively under attack. A recent MSSQL worm in May 2002 exploited several known MSSQL flaws. Hosts compromised by this worm generate a damaging level of network traffic when they scan for other vulnerable hosts. Additional information on this worm can be found at

- <http://www.incidents.org/diary/diary.php?id=157>
- <http://www.eeye.com/html/Research/Advisories/AL20020522.html>

Port 1433 (MSSQL default port) has also been regularly registered as one of the top scan ports in the Internet Storm Center. More detailed information about recent MSSQL exposures can be found in [CERT Advisory 2002-22](#).

W3.2 Operating Systems Affected

Any Microsoft Windows system with Microsoft SQL Server 7.0, Microsoft SQL Server 2000 or Microsoft SQL Server Desktop Engine 2000 installed.

W3.3 CVE Entries

[CAN-2002-0056](#), [CAN-2002-0649](#), [CAN-2001-0542](#), [CAN-2000-1081](#), [CVE-1999-0999](#), [CAN-2002-0624](#), [CAN-2002-0154](#), [CAN-2000-1209](#), [CAN-2002-1123](#), [CAN-2002-0186](#), [CVE-2000-0202](#), [CVE-2000-0402](#), [CVE-2000-0485](#), [CVE-2000-0603](#), [CVE-2001-0344](#), [CVE-2001-0879](#), [CAN-2000-0199](#), [CAN-2000-1082](#),

[CAN-2000-1083](#), [CAN-2000-1084](#), [CAN-2000-1085](#), [CAN-2000-1086](#), [CAN-2000-1087](#), [CAN-2000-1088](#),
[CAN-2001-0509](#), [CAN-2002-0187](#), [CAN-2002-0224](#), [CAN-2002-0641](#), [CAN-2002-0642](#), [CAN-2002-0643](#),
[CAN-2002-0644](#), [CAN-2002-0645](#), [CAN-2002-0650](#), [CAN-2002-0695](#), [CAN-2002-0721](#), [CAN-2002-0729](#),
[CAN-2002-0859](#), [CAN-2002-0982](#)

W3.4 How to Determine if you are Vulnerable

If the registry key HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MSSQLServer\MSSQLServer is defined, then you have SQL Server or SQL Server Desktop Engine installed. If you are running an un-patched system or you have not updated your system with the latest patch, your system is very likely to be vulnerable.

Microsoft has developed the Microsoft Baseline Security Analyzer (MBSA). MBSA will scan for missing hotfixes and vulnerabilities in SQL Server 7.0 and 2000. It is available at

<http://www.microsoft.com/technet/security/tools/Tools/MBSAhome.asp>.

Microsoft also has a HOWTO document to help you check your current version: [HOW TO: Identify Your SQL Server Service Pack Version and Edition](#).

To ensure the fix is installed properly, verify the individual files by consulting the date/time stamp of the files listed in the file manifest in Microsoft Knowledge Base article. They can be found at:

- [Microsoft SQL Server 7.0](#)
- [Microsoft SQL Server 2000](#)

W3.5 How to Protect Against It

Summary:

1. Apply the latest service pack for Microsoft SQL server.
2. Apply the latest cumulative patch that is released after the latest service pack.
3. Apply any individual patches that are released after the latest cumulative patch.
4. Secure the server at system and network level.

Detail:

1. *Apply the latest service pack for Microsoft SQL server.* The current Microsoft SQL Server service pack version is:
 - [SQL Server 7.0 Service Pack 4](#)
 - [SQL Server 2000 Service Pack 2](#)

To ensure that you are current with any future upgrades, monitor [Make Your SQL Servers Less Vulnerable](#) from Microsoft Technet.

2. *Apply the latest cumulative patch that is released after the latest service pack.* The current cumulative patch for all versions of SQL Server is available at [MS02-043 Cumulative Patch for SQL Server \(Q316333\)](#).

To ensure that you are current with any future upgrades, you can check for the latest cumulative patch for Microsoft SQL Server at:

- [Microsoft SQL Server 7.0](#)
- [Microsoft SQL Server 2000](#)
- [Microsoft SQL Server Desktop Engine 2000](#)

3. *Apply any individual patches that are released after the latest cumulative patch.* Currently, there is no individual patch after the release of the [MS02-043 Cumulative Patch for SQL Server \(Q316333\)](#). But to ensure that you are current with any future upgrades, you can check for any newly released individual patches at:

- [Microsoft SQL Server 7.0](#)
- [Microsoft SQL Server 2000](#)
- [Microsoft SQL Server Desktop Engine 2000](#)

4. *Secure the server at system and network level.*
 1. One of the most commonly attacked MSSQL exposures is that the default administrative account (known as "sa") is installed with a blank password. If your SQL "sa" account is not password-protected, you effectively have no security and can be affected by worms and other exploits.

Therefore, you should follow the recommendation from the "System Administrator (SA) Login" topic in [SQL Server Books Online](#) to make sure that the built-in "sa" account has a strong password, even if your SQL server does not run using this account.

Microsoft Developer's Network has documentation on [Changing the SQL Server Administrator Login](#) and how to [Verify and Change the System Administrator Password by Using MSDE](#).

2. Run the MSSQLServer service and SQL Server Agent under a valid domain account with minimal privileges, not as a domain administrator or the SYSTEM (on NT) or LocalSystem (on 2000 or XP) account. A compromised service running with local or domain privileges would give an attacker complete control of your machine and/or your network.
3. Enable Windows NT Authentication, enable auditing for successful and failed logins, and then stop and restart the MSSQLServer service. Configure your clients to use NT Authentication.
4. Packet filtering should be performed at network borders to prohibit non-authorized externally-initiated inbound connections to services. Ingress filtering of TCP ports 1433 and 1434 could prevent attackers outside of your network from scanning or infecting vulnerable Microsoft SQL servers in the local network that are not explicitly authorized to provide public SQL services.
5. If TCP ports 1433 and 1434 need to be available on your Internet gateways, enable and customize egress/ingress filtering to prevent misuse of this port.

Additional information on securing Microsoft SQL Server can be found at

- [Microsoft SQL Server 7.0 Security](#)
- [Microsoft SQL Server 2000 Security](#)

[Back to Top ^](#)

W4 NETBIOS -- Unprotected Windows Networking Shares

W4.1 Description

Microsoft Windows provides a host machine with the ability to share files or folders across a network with other hosts through Windows network shares. The underlying mechanism of this feature is the Server Message Block (SMB) protocol, or the Common Internet File System (CIFS). These protocols permit a host to manipulate remote files just as if they were local.

Although this is a powerful and useful feature of Windows, improper configuration of network shares may expose critical system files, or may provide a mechanism for a nefarious user or program to take full control of the host. One of the ways in which both the Sircam virus (see [CERT Advisory 2001-22](#)) and Nimda worm (see [CERT Advisory 2001-26](#)) spread so rapidly in the summer of 2001 was by discovering unprotected network shares and placing a copy of itself in them. Many computer owners unknowingly open their systems to hackers when they try to improve convenience for co-workers and outside researchers by making their drives readable and writable by network users. But when care is taken to ensure proper configuration of network shares, the risks of compromise can be adequately mitigated.

W4.2 Operating Systems Affected

Windows 95, Windows 98, Windows NT, Windows Me, Windows 2000, and Windows XP are all vulnerable.

W4.3 CVE Entries

[CAN-1999-0519](#), [CVE-2000-0979](#), [CAN-2000-1079](#), [CVE-2000-0044](#), [CAN-1999-0621](#), [CAN-1999-0520](#), [CAN-1999-0518](#)

W4.4 How to Determine if you are Vulnerable

For Windows NT (SP4), Windows 2000 or Windows XP, the [Microsoft Baseline Security Advisor](#), will report hosts are vulnerable to SMB exploits, and may be used to fix the problem. The tests can be run locally or on remote hosts.

Most commercially-available network-based scanners will detect open shares. A quick, free, and secure test for the presence of SMB file sharing and its related vulnerabilities, effective for machines running any Windows operating system, is available at the Gibson Research Corporation web site at <http://grc.com/>. Follow links to "ShieldsUP" to receive a real-time appraisal of any system's SMB exposure. Detailed instructions are available

to help Microsoft Windows users deal with SMB vulnerabilities. Note that if you are connected over a network where some intermediate device blocks SMB, the ShieldsUP tool will report that you are not vulnerable when, in fact, you are. This is the case, for example, for users on a cable modem where the provider is blocking SMB into the cable modem network. ShieldsUP will report that you are not vulnerable. However, the 4,000 or so other people on your cable modem link can still exploit this vulnerability.

W4.5 How to Protect Against It

Several actions can be taken to mitigate the risk of exploitation of a vulnerability through a Windows Networking Shares:

- Disable sharing wherever it is not required. If the host does not need to share files, then disable Windows network shares in the Windows network control panel. If an open share should be closed, you can disable it through Explorer's properties menu for that directory, in Server Manager for Domains or in Group Policy Editor.
- Do not permit sharing with hosts on the Internet. Ensure all Internet-facing hosts have Windows network shares disabled in the Windows network control panel. File sharing with Internet hosts should be achieved using FTP or HTTP.
- Do not permit unauthenticated shares. If file sharing is required then don't permit unauthenticated access to a share. Configure the share so a password is required to connect to the share.
- Restrict shares to only the minimum folders required. Generally only one folder and possibly sub-folders of that folder.
- Restrict permissions on shared folders to the minimum required. Be especially careful to only permit write access when it is absolutely required.
- For added security, allow sharing only to specific IP addresses because DNS names can be spoofed.
- Block ports used for Windows shares at your network perimeter. Block the NetBIOS ports commonly used by Windows shares at your network perimeter using either your external router or perimeter firewall. The ports that should be blocked are 137-139 TCP and 137-139 UDP, and 445 TCP and 445 UDP.

[Back to Top ^](#)

W5 Anonymous Logon -- Null Sessions

W5.1 Description

A Null Session connection, also known as Anonymous Logon, is a mechanism that allows an anonymous user to retrieve information (such as user names and shares) over the network, or to connect without authentication. It is used by applications such as the Windows Explorer to enumerate shares on remote servers. On Windows NT, 2000 and XP systems, many local services run under the SYSTEM account, known as LocalSystem on Windows 2000 and XP. The SYSTEM account is used for various critical system operations. When one machine needs to retrieve system data from another, the SYSTEM account will open a null session to the other machine.

The SYSTEM account has virtually unlimited privileges and it has no password, so you can't log on as SYSTEM. But SYSTEM sometimes needs to access information on other machines, such as available shares, user names, etc. -- the type of functionality offered by Network Neighborhood. Because it cannot log into the other systems using a UserID and password, it uses a Null session to get access. Unfortunately attackers can also log in as the Null Session.

W5.2 Operating Systems Affected

All flavors of Microsoft Windows NT, 2000 and XP.

W5.3 CVE Entries

[CAN-2000-1200](#)

W5.4 How to Determine if you are Vulnerable

Try to connect to your system via a Null session using the following command:

```
net use %a.b.c.d%ipc$ "" /user:""
```

(where a.b.c.d is the IP address of the remote system).

If you receive a "connection failed" response, then your system is not vulnerable. If no reply comes back that means that the command was successful and your system is vulnerable.

"Hunt for NT" can also be used. It is a component of the NT Forensic Toolkit from <http://www.foundstone.com>.

W5.5 How to Protect Against It

Domain controllers require Null sessions to communicate. Therefore, if you are working in a domain environment, you can minimize the information that attackers can obtain, but you cannot stop all leakage. To limit the information available to attackers, modify the following registry key:

```
HKLM/System/CurrentControlSet/Control/LSA/RestrictAnonymous=1
```

Whenever you modify the registry, it could cause your system to stop working properly. Therefore any changes should be tested before hand. Also, the system should always be backed up to simplify recovery.

Setting RestrictAnonymous to 1 will still permit certain information to be made available to anonymous users, but will minimize leakage. This is the tightest host-level restriction in NT. In Windows 2000 and XP, you can set the value to 2 instead. Doing so will bar anonymous users from all information where explicit access has not been granted to them or the Everyone group, which includes null session users. But this higher setting may affect domain synchronization or other services, and therefore should be thoroughly tested. For this reason, it is recommended that only those machines which are visible to the Internet have this value configured. All other machines should be protected by a firewall configured to block NetBIOS and CIFS.

If you do not need file and print sharing, unbind NetBIOS from TCP/IP.

Note here that configuring RestrictAnonymous on domain controllers and certain other servers can disrupt many normal networking operations.

Internet users should never be allowed to access any internal domain controller or other computer not specifically built for external access. To stop such access, block TCP and UDP ports 135, 137, 138, 139 and 445 at the external router or firewall.

[Back to Top ^](#)

W6 LAN Manager Authentication -- Weak LM Hashing

W6.1 Description

Although most current Windows environments have no need for LAN Manager (LM) support, Microsoft locally stores legacy LM password hashes (also known as LANMAN hashes) by default on Windows NT, 2000 and XP systems. Since LM uses a much weaker encryption scheme than more current Microsoft approaches (NTLM and NTLMv2), LM passwords can be broken in a very short period of time. Even passwords that otherwise would be considered "strong" can be cracked by brute-force in under a week on current hardware.

The weakness of LM hashes derives from the following:

- Passwords are truncated to 14 characters.
- Passwords are padded with spaces to become 14 characters.
- Passwords are converted to all upper case characters.
- Passwords are split into two seven character pieces.

This hashing process means that an attacker needs only to complete the trivial task of cracking two seven-character, upper-case passwords to gain authenticated access to your system. Since the complexity of cracking hashes increases geometrically with the length of the hash, each seven-character string is at least an order of magnitude simpler to attack by brute-force than would a combined fourteen-character string. Since all strings are exactly seven characters (including spaces) and entirely upper-case, a dictionary-style attack is also simplified. The LM hashing method therefore completely undermines good password policies.

In addition to the risk posed by having legacy LM hashes stored in the SAM, the LAN Manager authentication process is often by default enabled on clients and accepted by servers. As a result, Windows machines capable of utilizing stronger hash algorithms instead send weak LM hashes across the network, making Windows authentication vulnerable to eavesdropping by packet sniffing, and therefore easing the efforts of an attacker to obtain and crack user passwords.

W6.2 Operating Systems Affected

All Microsoft Windows operating systems.

W6.3 CVE Entries

N/A

W6.4 How to Determine if you are Vulnerable

If you are running a default installation of NT, 2000 or XP, you are vulnerable since LAN Manager hashes are stored locally by default.

If you have legacy operating systems in your environment that require LM authentication in order to communicate to servers, then you are vulnerable because those machines send LM hashes which can be sniffed off the network.

The more sophisticated Windows-based automated password cracking tools like LC4 (l0phtcrack version 4, available at <http://www.atstake.com/research/lc/download.html>) will show all hashes found in the SAM database (LM, NTLM or NTLMv2), and distinguish between the success cracking each. *PLEASE NOTE: Never run a password scanner, even on systems for which you have administrative access, without explicit and preferably written permission from your employer. Administrators with the most benevolent of intentions have been fired for running password cracking tools without authority to do so.*

W6.5 How to Protect Against It

1. *Disable LM Authentication Across the Network.* The best replacement in Windows for LAN Manager authentication is NT Lan Manager version 2 (NTLMv2). NTLMv2 challenge/response methods overcome many weaknesses in LM by using stronger encryption and improved authentication and session security mechanisms. The registry key that controls this capability in both Windows NT and 2000 is:

Hive: HKEY_LOCAL_MACHINE
Key: System\CurrentControlSet\Control\LSA
Value: LMCompatibilityLevel
Value Type: REG_DWORD - Number
Valid Range: 0-5
Default: 0

Description: This parameter specifies the type of authentication to be used.
0 - Send LM response and NTLM response; never use NTLMv2 session security
1 - Use NTLMv2 session security if negotiated
2 - Send NTLM authentication only
3 - Send NTLMv2 authentication only
4 - DC refuses LM authentication
5 - DC refuses LM and NTLM authentication (accepts only NTLMv2)

If all of your systems are Windows NT SP4 or later, you can set this to 3 on all clients and 5 on all domain controllers to prevent any transmission of LM hashes on the network. However, legacy systems (such as Windows 95/98) will not use NTLMv2 with the default Microsoft Network Client. To get NTLMv2 capability, install the Directory Services Client. Once installed, the registry value name is "LMCompatibility," and the allowed values are 0 or 3.

If you cannot force your legacy clients to use NTLMv2, you can gain a slight improvement over LM hashing by forcing NTLM (NT Lan Manager, version 1) at the domain controller (set "LMCompatibilityLevel" to 4). But the most secure option with regard to legacy systems is to migrate them to newer systems, since the older operating systems do not allow this minimum security level to be supported.

2. *Prevent the LM Hash from Being Stored.* One major problem with simply removing the LM hashes being passed over the network is that the hashes are still created and stored in the SAM or Active Directory. Microsoft has a mechanism available for turning off the creation of the LM hashes altogether, but only in Windows 2000 and XP.

On Windows 2000 systems, the following registry key controls this function:

Hive: HKEY_LOCAL_MACHINE
Key: System\CurrentControlSet\Control\LSA\NoLMHash

If this key is created on a Windows 2000 Domain Controller, the LanMan hashes will no longer be created and stored in Active Directory.

On Windows XP, the same functionality can be implemented by setting the registry value:

Hive: HKEY_LOCAL_MACHINE
Key: System\CurrentControlSet\Control\Lsa
Value: NoLMHash
Type: REG_DWORD - Number
Data: 1

After making these modifications to the registry, the system must be restarted in order for the change to take effect. IMPORTANT NOTE: This only prevents new LM hashes from being generated. Existing LM hashes are removed individually the next time each user changes his or her password.

The following Microsoft articles provide useful references:

- [How to Disable LM Authentication on Windows NT \[Q147706\]](#) details the required changes in the registry for Windows 9x and Windows NT/2000.
- [LMCompatibilityLevel and Its Effects \[Q175641\]](#) explains interoperability issues with this parameter.
- [How to Enable NTLMv2 Authentication for Windows 95/98/2000/NT \[Q239869\]](#) explains how to use Windows 2000's Directory Services Client for Windows 95/98 to overcome the compatibility limitation for NTLMv2.
- [New Registry Key to Remove LM Hashes from Active Directory and Security Account Manager](#)

[Back to Top ^](#)

W7 General Windows Authentication

-- Accounts with No Passwords or Weak Passwords

W7.1 Description

Passwords, passphrases and security codes are used in virtually every interaction between users and information systems. Most forms of user authentication, as well as file and data protection, rely on user-supplied passwords. Since properly authenticated access is often not logged, or even if logged not likely to arouse suspicion, a compromised password is an opportunity to explore a system from the inside virtually undetected. An attacker would have complete access to any resources available to that user, and would be significantly closer to being able to access other accounts, nearby machines, and perhaps even administrative privileges. Despite this threat, accounts with bad or empty passwords remain extremely common, and organizations with good password policy far too rare.

The most common password vulnerabilities are that (a) user accounts have weak or nonexistent passwords, (b) regardless of the strength of their password, users fail to protect it, (c) the operating system or additional software creates administrative accounts with weak or nonexistent passwords, and (d) password hashing algorithms are known and often hashes are stored such that they are visible by anyone. The best and most appropriate defense against these is a strong password policy which includes thorough instructions for good password habits and proactive checking of password integrity.

W7.2 Operating Systems Affected

Any operating system or application where users authenticate via a user ID and password.

W7.3 CVE Entries

[CAN-1999-0506](#), [CAN-1999-0504](#), [CVE-2000-0222](#), [CAN-1999-0505](#)

W7.4 How to Determine if you are Vulnerable

Although there are observable symptoms of general password weakness, such as the existence of active accounts for users who have departed the organization or services which are not running, the only way to know for certain that each individual password is strong is to test all of them against the same password cracking tools used by attackers. *PLEASE NOTE: Never run a password scanner, even on systems for which you have administrative access, without explicit and preferably written permission from your employer. Administrators with the most benevolent of intentions have been fired for running password cracking tools without authority to do so.*

The best cracking tools available are:

- [LC4 \(l0phtcrack version 4\)](#)
- [John the Ripper](#)
- [Symantec NetRecon](#)

W7.5 How to Protect Against It

The best and most appropriate defense against password weaknesses is a strong policy which includes thorough instructions to engender good password habits and proactive checking of password integrity.

1. *Assure that Passwords are Strong.* Given enough hardware and enough time, any password can be cracked by brute force. But there are simpler and very successful ways to learn passwords without such expense. Password crackers employ what are known as dictionary-style attacks. Since encryption methods are known, cracking utilities simply compare the encrypted form of a password against the encrypted forms of dictionary words (in many languages), proper names, and permutations of both. Therefore a password whose root in any way resembles such a word is highly susceptible to a dictionary attack. Many organizations instruct users to generate passwords by including combinations of alphanumeric and special characters, and users more often than not adhere by taking a word ("password") and converting letters to numbers or special characters ("pa\$\$w0rd"). Such permutations cannot protect against a dictionary attack: "pa\$\$w0rd" is as likely to be cracked as "password."

A good password, therefore cannot have a word or proper name as its root. A strong password policy should direct users to generate passwords from something more random, like a phrase, or the title of a book or song. By concatenating a longer string (taking the first letter of each word, or substituting a special character for a word, removing all the vowels, etc.), users can generate sufficiently long strings which combine alphanumeric and special characters in a way which dictionary attacks will have great difficulty cracking. And if the string is easy to remember, then the password should be as well.

Once users are given the proper instructions for generating good passwords, procedures should be put in place to assure that these instructions are followed. The best way to do this is by validating the password whenever the user changes it by employing [Passfilt](#).

Cracking utilities should be run in a stand-alone mode as part of routine scanning. *AGAIN PLEASE NOTE: Never run a password scanner, even on systems for which you have administrative access, without explicit and preferably written permission from your employer. Administrators with the most benevolent of intentions have been fired for running password cracking tools without authority to do so.* Once you have acquired authority to run cracking utilities on your system, do so regularly on a protected machine. Users whose passwords are cracked should be notified confidentially and given instructions on how to choose a good password. Administrators and management should develop these procedures together, so that management can provide assistance when users do not respond to these notifications.

Another way to protect against nonexistent or weak passwords is to use an alternative form of authentication such as password-generating tokens or biometrics. If you are having trouble with weak passwords, use an alternative means of authenticating users.

2. *Protect Strong Passwords.* Even if passwords themselves are strong, accounts can be compromised if users do not protect their passwords. Good policy should include instructions that a user should never tell his or her password to anyone else, should never write a password down where it could be read by others, and should properly secure any files in which a password is stored to automate authentication (passwords are easier to protect when this practice is only used when absolutely necessary). Password aging should be enforced so that any passwords which slip through these rules are only vulnerable for a short window of time, and old passwords should not be reused. Make sure that the users are given warning and chances to change their password before it expires. When faced with the message: "your password has expired and must be changed," users will tend to pick a bad password.
3. *Tightly Control Accounts.*
 - o Any service-based or administrative accounts not in use should be disabled or removed. Any service-based or administrative accounts which are used should be given new and strong passwords.
 - o Audit the accounts on your systems and create a master list. Do not forget to check passwords on systems like routers and Internet-connected digital printers, copiers and printer controllers.
 - o Develop procedures for adding authorized accounts to the list, and for removing accounts when they are no longer in use.
 - o Validate the list on a regular basis to make sure no new accounts have been added and that unused accounts have been removed.
 - o Have rigid procedures for removing accounts when employees or contractors leave, or when the accounts are no longer required.
4. *Maintain Strong Password Policy for the Enterprise.* In addition to operating system or network service-level controls, there are comprehensive tools available to help manage good password policy.

Symantec's Enterprise Security Manager (ESM) is a host-based monitoring tool that monitors any changes in policy, new account creation, and password strength. ESM will also attempt to crack passwords as it is performing a policy run on your network. ESM uses a client-manager environment: the agent is placed on the servers or workstations which in turn report to a centralized manager. Using a remote console, logs can be viewed and reports generated of the current status of the enterprise. ESM will monitor the audit logs and any change that has been made to the baseline of your network.

[Back to Top ^](#)

W8 Internet Explorer

W8.1 Description

Microsoft Internet Explorer (IE) is the default web browser installed on Microsoft Windows platforms. All existing versions of Internet Explorer have critical vulnerabilities. A malicious web administrator can design web pages to exploit these vulnerabilities on a user's Internet Explorer while browsing these web pages.

The vulnerabilities can be categorized into multiple classes including web page spoofing, ActiveX control vulnerabilities, Active scripting vulnerabilities, MIME-type and content-type misinterpretation and buffer overflows. The consequences may include disclosure of cookies, local files or data, execution of local programs, download and execution of arbitrary code or complete takeover of the vulnerable system.

W8.2 Operating Systems Affected

These vulnerabilities exist on Microsoft Windows systems running any version of Microsoft Internet Explorer. It is important to note that IE is installed with a wide variety of Microsoft software, and is therefore typically present on all Windows systems, even on servers where browsing is rarely necessary.

W8.3 CVE Entries

[CAN-2002-0193](#), [CAN-2002-0190](#), [CVE-2002-0027](#), [CVE-2002-0022](#), [CVE-2001-0875](#), [CVE-2001-0727](#), [CVE-2001-0339](#), [CVE-2001-0154](#), [CVE-2001-0002](#)

W8.4 How to Determine if you are Vulnerable

If you are using Internet Explorer on your system and have not installed the latest cumulative security patch, you are most likely vulnerable. If Windows Updates are enabled on your network, you can verify whether IE is installed and which Internet Explorer patches are installed on your system by visiting <http://windowsupdate.microsoft.com>. If Windows Updating is not available for your system, you can use [HFNetChk](#), the Network Security Hotfix Checker, or the [Microsoft Baseline Security Analyzer \(MBSA\)](#) to do the same.

You can also go to <http://browsercheck.qualys.com> to assess the impact of these vulnerabilities on your system.

W8.5 How to Protect Against It

Patches for these vulnerabilities are available for Internet Explorer versions 5.01, 5.5, 6.0. Earlier versions of Internet Explorer are also vulnerable, however patches may not be available for earlier versions. If your system is running an earlier version of IE, you should consider upgrading.

If you are running IE 5.01 or later, start by upgrading to the most recent service pack for Internet Explorer. The latest versions can be found at:

- [Internet Explorer 6, service pack 1](#)
- [Internet Explorer 5.5, service pack 2](#)
- [Internet Explorer 5.01, service pack 2](#)

After upgrading IE 5.5 or IE 5.01 to service pack 2, you should also add the latest [cumulative security patch \(Q323759\)](#), which repairs additional vulnerabilities. (This patch is already included in IE 6 service pack 1.) For more information about the vulnerabilities this patch repairs and appropriate changes to your configuration which can mitigate the risks, please see the related [Security Bulletin](#) and [Knowledge Base article](#).

Each of these articles discusses a variant on a cross-site scripting vulnerability, some aspects of which may not yet be completely solved by the patch. Please see <http://sec.greymagic.com/adv/gm010-ie/> for more information. If possible, it is generally good strategy to disable scripting wherever it is not necessary.

To maintain your system's protection, keep abreast of any new IE updates with [Windows Update](#), [HFNetChk](#), or

the [Microsoft Baseline Security Analyzer \(MBSA\)](#). You can also get general IE update information from Microsoft's [Internet Explorer Home](#).

[Back to Top ^](#)

W9 Remote Registry Access

W9.1 Description

Microsoft Windows 9x, Windows CE, Windows NT, Windows 2000 and Windows XP employ a central hierarchical database, known as the Registry, to manage software, device configurations and user settings. Improper permissions or security settings can permit remote registry access. Attackers can exploit this feature to compromise the system or form the basis for adjusting file association and permissions to enable malicious code.

W9.2 Operating Systems Affected

All versions of Microsoft Windows 9x, Windows CE, Windows NT, Windows 2000 and Windows XP.

W9.3 CVE Entries

[CAN-1999-0562](#), [CVE-2000-0377](#), [CVE-2000-0663](#), [CVE-2002-0049](#), [CAN-2001-0045](#), [CAN-2002-0642](#)

W9.4 How to Determine if you are Vulnerable

NT Resource Kit (NTRK) available from Microsoft contains an executable file entitled "regdump.exe" that will passively test remote registry access permissions from a Windows NT host against other Windows NT/Windows 2000 or Windows XP hosts on the Internet or internal network.

In addition, a collection of command line shell scripts that will test for registry access permissions and a range of other related security concerns is available for download at <http://www.afentis.com/top20>.

W9.5 How to Protect Against It

To address this threat, access to the system registry must be restricted and the permissions set for critical registry keys reviewed. Users of Microsoft Windows NT 4.0 should also ensure that Service Pack 3 (SP3) has been installed before adjusting the registry. *PLEASE NOTE: Editing the system Registry can have serious effects on the performance and operation of the computer and in extreme cases may cause irreparable damage and require reinstallation of the operating system.*

- **Restrict Network Access.** To restrict network access to the registry, follow the steps listed below to create the following Registry key:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg
Description: REG_SZ
Value: Registry Server
```

Security permissions set on this key define the Users or Groups that are permitted remote Registry access. Default Windows installations define this key and set the Access Control List to provide full privileges to the system Administrator and Administrators Group (and Backup Operators in Windows 2000).

Changes to the system registry will require a reboot to take effect. To create the registry key to restrict access to the registry:

1. Start Registry Editor ("regedt32.exe" or "regedit.exe") and go to the following subkey:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control
```

2. On the "Edit" menu, click "Add Key".

3. Enter the following values:

```
Key Name: SecurePipeServers
Class: REG_SZ
```

4. Go to the following subkey:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers

5. On the "Edit" menu, click "Add Key".

6. Enter the following values:

Key Name: winreg
Class: REG_SZ

7. Go to the following subkey:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg

8. On the "Edit" menu, click "Add Value".

9. Enter the following values:

Value Name: Description
Data Type: REG_SZ
String: Registry Server

10. Go to the following subkey:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg

11. Select "winreg." Click "Security" and then click "Permissions." Add Users or Groups to which you want to grant access.

12. Exit Registry Editor and restart Microsoft Windows.

13. If you at a later stage want to change the list of users that can access the registry, repeat steps 10-12.

- *Limit Authorized Remote Access.* Enforcing strict restrictions upon the registry can have adverse side effects upon dependent services, such as the Directory Replicator and the network printer Spooler service.

It is therefore possible to add a degree of granularity to the permissions, by adding either the account name that the service is running under to the access list of the "winreg" key, or by configuring Windows to bypass the access restriction to certain keys by listing them in the Machine or Users value under the AllowedPaths key:

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg\AllowedPaths

Value: Machine
Value Type: REG_MULTI_SZ - Multi string
Default Data: System\CurrentControlSet\Control\ProductOptions\System\CurrentControlSet\Control\Print\Printers\System\CurrentControlSet\Services\EventlogSoftware\Microsoft\WindowsNT\CurrentVersion\System\CurrentControlSet\Services\Replicator

Valid Range: (A valid path to a location in the registry)
Description: Allow machines access to listed locations in the registry provided that no explicit access restrictions exist for that location.

Value: Users
Value Type: REG_MULTI_SZ - Multi string
Default Data: (none)
Valid Range: (A valid path to a location in the registry)
Description: Allow users access to listed locations in the registry provided that no explicit access restrictions exist for that location.

In the Microsoft Windows 2000 and Windows XP Registry:

Value: Machine
Value Type: REG_MULTI_SZ - Multi string
Default Data: System¥CurrentControlSet¥Control¥ProductOptionsSystem¥
CurrentControlSet¥Control¥Print¥PrintersSystem¥CurrentControlSet¥
control¥Server ApplicationSystem¥CurrentControlSet¥Services¥Eventlog¥
Software¥Microsoft¥Windows NT¥CurrentVersion
Value: Users (does not exist by default)

For more information, please see Microsoft Knowledge Base Article Q153183, [How to Restrict Access to NT Registry from a Remote Computer](#).

[Back to Top ^](#)

W10 Windows Scripting Host

W10.1 Description

In the spring of 2000, "The Love Bug" (also known as "ILOVEYOU") Visual Basic script (VBScript) worm caused millions of dollars in damages. This worm, and others which have followed it, took advantage of Windows Scripting Host (WSH), which permits any text file with a ".vbs" extension to be executed as a Visual Basic script. With WSH enabled, a typical worm propagates by including a VBScript as the contents of another file and executes when that file is viewed or in some cases previewed.

While administrators should always keep applications like browsers, mail clients and productivity suites patched and updated, patching these applications to eliminate their susceptibility to a particular worm is an incomplete (and no better than reactive) solution to the risks posed by scripting. Windows Scripting Host can be safely disabled on most systems in a proactive effort to prevent worms from spreading.

W10.2 Operating Systems Affected

Windows Scripting Host can be installed manually or with Internet Explorer 5 (or higher) on Windows 95 or NT. It is installed by default on Windows 98, ME, 2000 and XP machines.

W10.3 CVE Entries

[CAN-2001-1325](#), [CVE-2001-0149](#)

W10.4 How to Determine if you are Vulnerable

If you are running Windows 95 or NT with IE 5 or higher, or are running Windows 98, ME, 2000 or XP, and have not disabled WSH, then you are likely vulnerable.

W10.5 How to Protect Against It

- Disable or remove Windows Scripting Host as outlined instruction sets provided by [Symantec](#) and [Sophos](#).
- Always keep your Anti-Virus software and definitions up-to-date. Some Anti-Virus software includes options to block scripts.

[Back to Top ^](#)

Top Vulnerabilities to Unix Systems (U)

U1 Remote Procedure Calls (RPC)

U1.1 Description

Remote procedure calls (RPCs) allow programs on one computer to execute procedures on a second computer by passing data and retrieving the results. RPC is therefore widely used for many distributed network services such as remote administration, NFS file sharing, and NIS. However there are multiple flaws in RPC which are being actively exploited. In many cases, RPC services execute with root privileges, and as a consequence, systems that offer vulnerable RPC services can provide an attacker with unauthorized remote root access. There is compelling evidence that the majority of the distributed denial of service attacks launched during 1999 and early 2000 were executed by systems that had been victimized through these RPC vulnerabilities. The broadly successful attack on U.S. Military systems during the Solar Sunrise incident also exploited an RPC flaw found on hundreds of Department of Defense computer systems.

U1.2 Operating Systems Affected

Nearly all versions of Unix and Linux come with RPC services installed and often enabled.

U1.3 CVE Entries

[CVE-1999-0166](#), [CVE-1999-0167](#), [CVE-1999-0168](#), [CVE-1999-0170](#), [CVE-1999-0211](#), [CVE-1999-0977](#), [CVE-1999-0018](#), [CVE-2000-0666](#), [CVE-1999-0002](#), [CVE-2001-0803](#), [CVE-1999-0493](#), [CAN-2002-0573](#), [CVE-2001-0717](#), [CVE-1999-0003](#), [CVE-1999-0019](#), [CVE-1999-0208](#), [CVE-1999-0696](#), [CVE-1999-0693](#), [CVE-1999-0008](#), [CVE-2001-0779](#), [CAN-2002-0033](#), [CAN-2002-0391](#), [CAN-2002-0677](#), [CAN-2002-0679](#), [CVE-1999-0207](#), [CVE-1999-0219](#)

U1.4 How to Determine if you are Vulnerable

Use a vulnerability scanner or the 'rpcinfo' command to determine if you are running one of the most commonly exploited RPC services:

RPC Service	RPC Program Number
rpc.ttdbserverd	100083
rpc.cmsd	100068
rpc.statd	100024
rpc.mountd	100005
sadmind	100232
cachefs	100235
snmpXdmid	100249

RPC services are typically exploited through buffer overflow attacks which are successful because the RPC programs do not perform sufficient error checking or input validation. Buffer overflow vulnerabilities allow an attacker to send unexpected data (often in the form of malicious code) into the program memory space. Due to poor error checking and input validation, the data overwrite key memory locations that are in line to be executed by the processor. In a successful overflow attack, this malicious code is then executed by the operating system. Since many RPC services execute with root privileges, a successful exploitation of one of these services can provide unauthorized remote root access to the system.

U1.5 How to Protect Against It

Use the following steps to protect your system against RPC attacks:

1. Turn off or remove any RPC service which is not absolutely necessary for the function of your network.
2. Install the latest patches for any services you cannot remove:

For Solaris Software Patches:

<http://sunsolve.sun.com>

For IBM AIX Software Patches:

<http://www.ibm.com/support/us>

<http://techsupport.services.ibm.com/server/fixes>

For SGI Software Patches:

<http://support.sgi.com>

For Compaq (Digital Unix) Software Patches:

<http://www.compaq.com/support>

For Linux Software Patches:

<http://www.redhat.com/apps/support/errata>

<http://www.debian.org/security>

3. Regularly search the vendor patch database for new patches and install them right away.
4. Block the RPC port (port 111) at the border router or firewall.
5. Block the RPC "loopback" ports, 32770-32789 (TCP and UDP).

6. Enable a non-executable stack on those operating systems that support this feature. While a non-executable stack will not protect against all buffer overflows, it can hinder the exploitation of some standard buffer overflow exploits publicly available on the Internet.
7. For NFS exported file systems, the following steps should be taken:
 1. Use host/IP based export lists.
 2. Setup exported file systems for read-only or no-suid wherever possible.
 3. Use 'nfsbug' to scan for vulnerabilities.

A summary document pointing to specific guidance about three principal RPC vulnerabilities - Tooltalk, Calendar Manager, and Statd - may be found at: http://www.cert.org/incident_notes/IN-99-04.html

Summary documents pointing to specific guidance about the above RPC vulnerabilities may be found at:

- Statd: <http://www.cert.org/advisories/CA-2000-17.html>
<http://www.cert.org/advisories/CA-1999-05.html>
<http://www.cert.org/advisories/CA-1997-26.html>
- Tooltalk: <http://www.cert.org/advisories/CA-2002-26.html>
<http://www.cert.org/advisories/CA-2002-20.html>
<http://www.cert.org/advisories/CA-2001-27.html>
- Calendar Manager: <http://www.cert.org/advisories/CA-2002-25.html>
<http://www.cert.org/advisories/CA-1999-08.html>
- Cachefs: <http://www.cert.org/advisories/CA-2002-11.html>
- Sadmind: <http://www.cert.org/advisories/CA-1999-16.html>
<http://www.cert.org/advisories/CA-2001-11.html>
- Mountd: <http://www.cert.org/advisories/CA-1998-12.html>
- SnmpXdmid: <http://www.cert.org/advisories/CA-2001-05.html>

[Back to Top ^](#)

U2 Apache Web Server

U2.1 Description

Web administrators too often conclude that since Microsoft's Internet Information Server (IIS) is exceptionally prone to compromise (see W1. Internet Information Server), the open-source [Apache web server](#) is completely secure. While the comparison with IIS may be true, and although Apache has a well-deserved reputation for security, it has not proved invulnerable under scrutiny.

There have been weaknesses found in Apache. Even [the apache.org website was defaced in early 2000](#). Exploits of core Apache or its modules in the recent past have been few, but they have been well-documented and quickly utilized in attacks. Among the most recent:

- [Apache/mod_ssl Worm \(CERT Advisory CA-2002-27\)](#)
- [Apache Chunk Handling Exploit \(CERT Advisory CA-2002-17\)](#)

Moreover, no web server can be considered secure until it is considered in the context of its interaction with web applications, especially CGI programs and databases. A hardened Apache configuration can still yield unauthorized access to data if CGI scripts are not themselves verified or database access controls not properly set. CGI scripts execute with the same permissions as the web server, so a malicious or just poorly written CGI script is just as dangerous as a software flaw in Apache. Unfortunately, these weaknesses on the back end of the web server remain problems today.

It is also imperative to harden the OS to truly prevent a web content from being modified or stolen. Although that is true for all running services, the fact that web services tend to have an external exposure lends itself to a false impression that they and the data they protect are somehow independent of the rest of the system. How failure

to address this issue left one system vulnerable to attack is explained in <http://www.wired.com/news/technology/0,1282,43234,00.html>.

U2.2 Operating Systems Affected

Nearly all Linux systems and many other Unix systems come with Apache installed and often by fault enabled. All Unix systems are capable of running Apache. (Windows administrators should be aware that the version of Apache for Windows is likely subject to the same or similar vulnerabilities.)

U2.3 CVE Entries

[CAN-2002-0392](#), [CAN-2002-0061](#), [CVE-1999-0021](#), [CVE-1999-0172](#), [CVE-1999-0266](#), [CVE-1999-0067](#), [CVE-1999-0260](#), [CVE-1999-0262](#), [CVE-2000-0010](#), [CVE-1999-0174](#), [CVE-1999-0066](#), [CVE-1999-0146](#), [CAN-2002-0513](#), [CAN-2002-0682](#), [CAN-2002-0257](#), [CVE-2000-0208](#), [CVE-2000-0287](#), [CVE-2000-0941](#), [CAN-2000-0832](#), [CVE-1999-0070](#), [CVE-2002-0082](#), [CVE-1999-0708](#), [CVE-1999-0751](#), [CAN-2002-0656](#), [CAN-2002-0655](#), [CVE-2001-1141](#), [CAN-2002-0657](#), [CAN-1999-0509](#), [CVE-1999-0237](#), [CVE-1999-0264](#)

U2.4 How to Determine if you are Vulnerable

Check to see what the latest version and patch level is at the Apache web site: <http://httpd.apache.org>. If your version is not the most recent, then your server is likely vulnerable. This site also maintains a list of most recent vulnerabilities and documentation on how to determine if you are vulnerable to them.

U2.5 How to Protect Against It

The following steps should be taken to help protect an Apache web server:

1. Get the latest patches from Apache at <http://www.apache.org/dist/httpd/patches/>. If possible, upgrade to the latest version.
2. Modify the default Apache HTTP Response token. This will allow your Apache server to return false information in its response header, which helps hide the web server's software. While this technique will not prevent a determined attacker from discovering your software, it can greatly protect your Apache web server from worms which trigger their attack code based on the information returned from headers. Please see the [Security Focus discussion](#) on how this can deter the Apache/mod_ssl Worm described in [CERT Advisory CA-2002-27](#).
3. Only compile in the Apache modules that your server requires to function properly. Much like an operating system running unneeded services, Apache itself should be minimized so as to reduce the exposure to future security issues.
4. Consider running Apache in a chroot() environment. To prevent these malicious HTTP requests from being successfully executed, a web server should be configured to initialize with the Unix chroot() function. When a web server starts chroot-ed, it is essentially placed within a "Silver Bubble" environment. From this configuration, the web server cannot access any part of the OS directory structure outside of the designated chroot() area. Each web server implements the chroot() differently, and therefore software documentation should be consulted for assistance. Additional information can be found in the [WWW Security FAQ](#).
5. Do not run Apache as root. Create a new user with minimal privileges on your network and in the databases offered by your web services and run Apache as that user. Do NOT use the nobody account, for this account is used to map the root account over NFS.
6. Remove the default html content, including the two CGI scripts test-cgi and printenv. Weaknesses in default content are very well-known and frequently attacked.
7. Best practices for handling CGI scripts:
 - o Do not configure CGI support on Web Servers that do not need it.
 - o Remove all sample CGI programs from your production web server.
 - o Audit the remaining CGI scripts and remove unsafe CGI scripts from all web servers.
 - o Ensure all CGI programmers adhere to a strict policy of input buffer length checking in CGI programs.
 - o Make sure that your CGI bin directory does not include any compilers or interpreters.
 - o Remove the "view-source" script from the cgi-bin directory.
 - o Configure your Apache server to use CGI alerting scripts for Error Responses. WebAdmins need to keep tabs on all of these security related issues with their web servers. To assist with this

monitoring, the web server should be configured to use custom CGI error response pages for server response codes 401, 403, 413 and 500. The error pages are PERL CGI scripts that are initiated every time the server issues either of these response codes. These scripts accomplish many important tasks including issuing an html warning banner to the client and immediately sending an e-mail notification to the WebAdmin. The e-mail message automates the process of manually collecting security related session information from the web server access and error logs for the request.

- Do not allow Directory Indexing. Directory indexing can give an attacker too much information about your site's directory structure and naming conventions.
- Do not use Server Side Includes (SSI). SSIs can potentially be abused and cause the web server to execute OS code which was not intended by the developer.
- In order to contain the directories which can be offered to clients, do not allow the Apache server to follow symbolic links.
- Create CGI Alerting Scripts to catch CGI Scanners. Use a CGI alerting script and rename it to vulnerable script names such as: test-cgi, phf, php.cgi, etc. When a CGI Vulnerability scanner is run against your web server, these scripts will be executed and the WebAdmin will be notified via email.

8. Perhaps most importantly, ensure that the underlying operating system and running services are hardened, or all of your steps until now will be for naught. Follow the other Top 20 entries, the [SANS Consensus Security Guides](#), and the [Center for Internet Security's Benchmarks](#).

For more Apache security information, see <http://www.sans.org/Gold/apache.php> and http://www.infosecurymag.com/articles/april01/features1_web_server_sec.shtml.

[Back to Top ^](#)

U3 Secure Shell (SSH)

U3.1 Description

Secure shell (ssh) is a popular service for securing logins, command execution, and file transfers across a network. Most Unix-based systems use either the open-source [OpenSSH](#) package or the commercial version from [SSH Communication Security](#). Although ssh is vastly more secure than the telnet, ftp, and R-command programs it is intended to replace, there have been multiple flaws found in both implementations. Most are minor bugs, but a few are major security issues which should be repaired immediately. The most dangerous of these actively exploited holes allow attackers to obtain root access on a machine from a remote location.

The SSH1 protocol itself has been demonstrated to be potentially vulnerable to having a session decrypted in transit given certain configurations. For this reason, administrators are encouraged to use the stronger SSH2 protocol whenever possible.

In addition, users of OpenSSH should note that the OpenSSL libraries against which OpenSSH is typically built have software vulnerabilities of their own. Please see [CERT Advisory 2002-23](#) for more details. They should also be aware that a trojan-horse version of the OpenSSH was being distributed for a short-time in summer 2002. Please see http://www.openssh.org/txt/trojan_adv for details about ensuring that your version is not affected

U3.2 Operating Systems Affected

Any Unix or Linux system running OpenSSH 3.3 or earlier, or SSH Communication Security's SSH 3.0.0 or earlier.

U3.3 CVE Entries

For ssh from SSH Communication Security: [CVE-2000-0575](#), [CVE-2000-0992](#), [CVE-2001-0144](#), [CVE-2001-0361](#), [CAN-2001-0471](#), [CVE-2001-0553](#), [CVE-2001-0259](#)

For OpenSSH: [CVE-2000-1169](#), [CVE-2001-0144](#), [CVE-2001-0361](#), [CVE-2001-0872](#), [CVE-2000-0525](#), [CAN-2001-0060](#), [CAN-2002-0002](#), [CAN-2002-0575](#), [CAN-2002-0639](#), [CAN-2002-0083](#), [CAN-2002-0640](#), [CAN-2002-0656](#), [CAN-2002-0655](#), [CVE-2001-1141](#), [CAN-2002-0657](#)

U3.4 How to Determine if you are Vulnerable

Use a vulnerability scanner to see whether you are running a vulnerable version, or check the software version reported by running the command 'ssh -V'.

U3.5 How to Protect Against It

1. Upgrade to the most recent version of either [OpenSSH](#) or [SSH](#). Or if SSH or OpenSSH came installed with your operating system, retrieve the latest patches from your operating system vendor. If you use OpenSSL, be sure to use the latest version of those libraries.
2. If at all possible, avoid the use of the SSH1 protocol, as there are known weaknesses corrected in the SSH2 protocol.
3. Both the ssh implementations include a variety of configuration options to restrict what machines can connect, and what users are allowed to authenticate, and via what mechanisms. Administrators should determine how these options can most appropriately be set for their environment.

[Back to Top ^](#)

U4 Simple Network Management Protocol (SNMP)

U4.1 Description

The Simple Network Management Protocol (SNMP) is used extensively to remotely monitor and configure almost all types of modern TCP/IP-enabled devices. While SNMP is rather ubiquitous in its distribution across networking platforms, it is most often used as a method to configure and manage devices such as printers, routers, switches, and to provide input for network monitoring services.

Simple Network Management communication consists of different types of exchanged messages between SNMP management stations and network devices which run what is commonly referred to as agent software. The method by which these messages are handled, and the authentication mechanism behind such message handling, both have significant exploitable vulnerabilities.

The vulnerabilities behind the method by which SNMP handles and traps messages are outlined in detail in [CERT Advisory CA-2002-03](#). There exists a set of vulnerabilities in the way trap and request messages are handled and decoded by management stations and agents alike. These vulnerabilities are not restricted to any specific implementation of SNMP, but instead affect a variety of vendors' SNMP distributions. The result of attackers exploiting these vulnerabilities may range anywhere from denial of service to unwanted configuration and management of your SNMP-enabled machinery.

The inherent authentication mechanism within SNMP also poses a significant vulnerability. SNMP uses an unencrypted "community string" as its only authentication mechanism. Lack of encryption is bad enough, but the default community string used by the vast majority of SNMP devices is "public," with a few supposedly clever network equipment vendors changing the string to "private" for more sensitive information. Attackers can use this vulnerability in SNMP to reconfigure or shut down devices remotely. Sniffed SNMP traffic can reveal a great deal about the structure of your network, as well as the systems and devices attached to it. Intruders use such information to pick targets and plan attacks.

SNMP is not unique to Unix; it is extensively used on Windows, in networking equipment, printers and embedded devices. But the majority of SNMP-related attacks seen thus far have occurred on Unix systems with poor SNMP configurations.

U4.2 Operating Systems Affected

Nearly all Unix and Linux systems come with SNMP installed and often by default enabled. Most other SNMP-enabled network devices and operating systems are also vulnerable.

U4.3 CVE Entries

[CAN-2002-0013](#), [CAN-2002-0797](#), [CAN-2002-0012](#), [CAN-2002-0048](#), [CAN-2002-0796](#), [CAN-1999-0516](#), [CAN-1999-0517](#), [CAN-1999-0254](#), [CAN-1999-0186](#), [CAN-1999-0615](#), [CVE-2001-0236](#)

U4.4 How to Determine if you are Vulnerable

You can verify whether SNMP is running on network-connected devices by running a scanner or checking manually.

SNMPing – You can obtain the free SNMPing scanning tool from the SANS Institute by emailing a blank mail message to snmptool@sans.org. You will get a return message with the URL where you can download the tool.

SNScan – Foundstone created another easy-to-use SNMP scanning tool called SNScan, which can be obtained at http://www.foundstone.com/knowledge/free_tools.html.

If you can not use any of the above tools, you should manually verify if SNMP is running on your systems. Refer to your operating system documentation on how to specifically identify its particular SNMP implementation, but the basic daemon can usually be identified by grepping for "snmp" in the process list, or by looking for services running on ports 161 or 162.

A running SNMP instance is probably sufficient evidence that you are vulnerable to pervasive trap and request handling errors. Please see [CERT Advisory CA-2002-03](#) for additional information.

If SNMP is running and any of these additional variables are met, you may have a default or easily guessable string-related vulnerability:

1. Blank or default SNMP community names.
2. Guessable SNMP community names.
3. Hidden SNMP community strings.

Please see <http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm> for information on how to identify the presence of those conditions.

U4.5 How to Protect Against It

- *Trap and Request Handling Vulnerabilities:*
 1. If you do not absolutely require SNMP, disable it.
 2. If you must use SNMP, make sure you are running the latest patched version from your vendor. A good starting point in obtaining vendor specific information is Appendix A of [CERT Advisory CA-2002-03](#).
 3. Filter SNMP (port 161 TCP/UDP and 162 TCP/UDP) at the ingress points to your networks, unless it is absolutely necessary to poll or manage devices externally.
 4. Employ host-based access control on your SNMP agent systems. While this capability may be limited by SNMP agent operating system capabilities, control of what systems your agents will accept requests from may be possible. On most Unix systems this can be accomplished through a TCP-Wrappers or Xinetd configuration. An agent-based packet filtering firewall on the host can also be used to block unwanted SNMP requests.
- *Default and Guessable String-Related Vulnerabilities:*
 1. If you do not absolutely require SNMP, disable it.
 2. If you must use SNMP, use the same policy for community names as used for passwords. Make sure they are difficult to guess or crack, and that they are changed periodically.
 3. Validate and check community names using `snmpwalk`. Additional information can be found at <http://www.zend.com/manual/function.snmpwalk.php>. A good tutorial on this tool can be found at <http://www.sans.org/newlook/resources/IDFAQ/SNMP.htm>.
 4. Filter SNMP (port 161 TCP/UDP and 162 TCP/UDP) at the ingress points to your networks, unless it is absolutely necessary to poll or manage devices externally.
 5. Where possible make MIBs read-only. Additional information can be found at http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm#xtocid210315.

[Back to Top ^](#)

U5 File Transfer Protocol (FTP)

U5.1 Description

FTP daemon is used to distribute files to anonymous or authenticated (via username and password) users. Anonymous FTP services do not require a unique password (any will do) and all users use the same login name ("anonymous" or "ftp"), thus allowing everybody to access the service.

Authenticated FTP services do require a username and a password, but each is transmitted over the network in the clear, permitting a third party to eavesdrop on the exchange of credentials. To steal the FTP login information, an attacker needs to place a network sniffer somewhere along the connection path, such as on the FTP server LAN or on the client LAN. Attackers have deployed such sniffers in many recent security incidents.

In addition to this inherent transmission insecurity, critical flaws have been found in many versions of FTP server software, both those provided by operating system vendors (Sun, HP-UX, etc) and those developed by the open source community (WU-FTPD, ProFTPD, etc). Many exploits allow an attacker to gain root access to the machine hosting the FTP server, while others simply permit user-level command execution. For example, recent WU-FTPD exploits allow attackers to gain root and upload their tools such as rootkits and then use the system

for their nefarious purposes. Most of the exploits require the anonymous access to be enabled, but some will work even when anonymous access is denied so long as the FTP server listens on the network port. It should be noted that although FTP server uses a chroot() system call to confine an anonymous user into a specified directory, it can still be exploited due to major bugs in the implementation.

U5.2 Operating Systems Affected

Nearly all Unix and Linux systems come with at least one FTP server installed and often by default enabled.

U5.3 CVE Entries

[CVE-1999-0368](#), [CVE-2001-0550](#), [CVE-1999-0080](#), [CVE-1999-0878](#), [CVE-1999-0879](#), [CVE-1999-0950](#), [CAN-2001-0249](#), [CAN-1999-0527](#), [CAN-1999-0911](#), [CVE-1999-0955](#), [CVE-2000-0573](#), [CVE-2001-0187](#), [CAN-2001-0935](#), [CVE-1999-0880](#), [CAN-2000-0574](#), [CAN-2001-0247](#), [CVE-2001-0053](#), [CVE-2001-0318](#), [CAN-2001-0248](#), [CVE-1999-0934](#), [CVE-1999-0082](#), [CVE-1999-0083](#), [CVE-2000-0856](#), [CAN-2001-0065](#), [CAN-2001-0283](#), [CVE-2001-0456](#)

U5.4 How to Determine if you are Vulnerable

Various versions of UNIX FTP daemons have a large number of vulnerabilities and must be regularly updated and patched. Check to see what the latest version and patch level is for your particular FTP server software by looking at your operating system vendor or FTP software vendor website. If it is not the latest, chances are that your version is vulnerable and that exploits of the flaw are publicly available in the underground community.

One may also use the freely available Nessus scanner (<http://www.nessus.org>) to scan for FTP flaws.

U5.5 How to Protect Against It

The following steps should be taken to protect the FTP service:

1. Upgrade to latest version of your FTP. The most popular free FTP servers are [WU-FTPD](#) and [ProFTPD](#). If your version of FTP came with your operating system, check your OS vendor's website for upgrade information.
2. Disable anonymous access to FTP services if it is not needed. Follow the instructions in the software manual for your particular version. For WU-FTPD and ProFTPD, create or edit the `/etc/ftpusers` file and add the usernames "anonymous" and "ftp" in it (on separate lines). This file sets which users should **not** be allowed to login to FTP server. To add an additional layer of security, also remove the "ftp" user from the password file.
3. In case anonymous functionality is needed, at least make sure that anonymous upload functionality is disabled so that users need a valid username and password to put files on your server. Anonymous upload functionality is disabled by default in most FTP daemons. To verify that it is indeed disabled, connect to your FTP server and try to execute a "put whatever.file" command. If the instruction fails, the error message will indicate that uploads are disabled.
4. Restrict access to the FTP server by the IP address or domain using TCP wrappers. TCP wrappers are installed by default on most recent Unix and Linux distributions. If not, you can build it from the source located at ftp://ftp.porcupine.org/pub/security/tcp_wrappers_7.6.tar.gz, and deploy. By putting a line similar to "in.ftpd: 10.164.168.15" or "in.ftpd: .good_domain.com" into your `/etc/hosts.allow` file, you will allow access only from a specific IP address or domain. You should then put "in.ftpd: ALL" in `/etc/hosts.deny` to block access from all others, and confirm that FTP daemon is started via "tcpd" in `/etc/inetd.conf`. Some Linux distributions (such as RedHat) use an enhanced version of inetd called xinetd, which contains the TCP wrapper code and will check the above files by default. Refer to the manual for the tips on xinetd configuration.
5. Implement restrictive file permissions on the FTP server so that users are able to only access files needed. Most FTP servers have an ability to impose granular access control for FTP users in addition to UNIX file permissions.
6. Add all administrative accounts (such as root, daemon, sys, etc.) to the `/etc/ftpusers` file so that those accounts cannot be accessed by FTP.
7. Consider replacing FTP with more secure software solutions such as SFTP or SCP (parts of the Secure Shell software package) and use a web server to distribute files to a wide audience.

8. Disable unused FTP servers completely and remove the software from the system. Firewall off port 21 on the perimeter device if FTP is not used for business reasons.

[Back to Top ^](#)

U6 R-Services -- Trust Relationships

U6.1 Description

Remote shell (rsh), remote copy (rcp), remote login (rlogin), and remote execution (rexec) -- known collectively as the "R-commands" -- are widely used in the Unix world. Organizations with multiple Unix compute servers will often configure the corresponding "R-services" (in.rshd, in.rlogind, in.rexecd) in such a way that users can move from one machine to another without having to enter a user ID and password each time. Even on networks where a given user's resources are contained to a single system, administrators are often responsible for dozens or even hundreds of systems, and therefore configure the R-services to ease their own movement from machine to machine. A single user can rsh, rcp, rlogin or rexec from machine A to machine B without having to re-authenticate by placing the name or address of machine A in his or her `~/rhosts` file on machine B. All users can rsh, rcp, rlogin or rexec from machine A to machine B without having to re-authenticate if the name or address of machine A is in machine B's `/etc/hosts.equiv` file.

R-services suffer from the two most fundamental flaws in network connections: lack of encryption and poor host authentication. The transmission of information between R-command clients and R-services in plain-text permits data or keystrokes to be intercepted. The fact that R-services simply accept the name or address presented by a connecting client permits that information to be forged. Without established trust relationships, users are forced to send passwords over the network in the clear. With trust relationships, an attacker can assume the identity of a valid user on a valid host, and use it to gain access to all other machines that trust the hacked machine.

U6.2 Operating Systems Affected

Nearly all versions of Unix and Linux come with R-services installed and often enabled.

U6.3 CVE Entries

[CVE-1999-0113](#), [CVE-1999-0627](#), [CVE-1999-0180](#), [CAN-1999-0651](#), [CAN-1999-0515](#)

.4 How to Determine if you are Vulnerable

The R-services run out of a meta-server called "inetd," or on some systems, "xinetd." Inetd will permit rsh or rcp connections if there is an entry for "in.rshd" (the specific name may vary slightly for your distribution) in `/etc/inetd.conf` or `/etc/inet/xinetd.conf`. Likewise, rexec requires an entry for "in.rexecd," and rlogin an entry for "in.rlogind." Xinetd works similarly, expecting a file named after each service it starts to appear in the `/etc/xinetd.d` directory.

Trust relationships have been established on a machine if there are entries in the `/etc/hosts.equiv` file, or in the `~/rhosts` file of any valid user.

U6.5 How to Protect Against It

Disable the R-services on any system where they are not absolutely necessary. Secure shell (ssh, available from either [OpenSSH](#) or [SSH Communications Security](#)) and its compliments of scp and sftp can far more securely replace the functionality of all R-services. If R-services are absolutely necessary, disable trust relationships and use [TCP Wrappers](#) to log all connection attempts, restrict access to specific hosts, and provide host verification. TCP Wrappers functionality is already built into xinetd.

To disable trust relationships, remove the `/etc/hosts.equiv` file and the `~/rhosts` file of any user. If you must use trust relationships, never use the "+" (wildcard) character, as it can be used to allow any user or any machine (or worse, any user from any machine) to login with proper credentials, and be sure to use [TCP Wrappers](#). Never use `~/rhosts` to permit password-less root authentication.

[Back to Top ^](#)

U7 Line Printer Daemon (LPD)

U7.1 Description

The Berkeley line printer daemon (LPD) is historically the service which lets users connect to a local printer from a local machine or from a remote machine on TCP port 515. Although there are replacement servers available, LPD remains the most commonly used print server across Unix and Linux distributions. Many implementations of LPD, however, contain programming flaws which have led to buffer overflows allowing attackers to run arbitrary code with root privileges. So many different Unix operating systems contain vulnerable LPD daemons that CERT issued a general advisory (<http://www.cert.org/advisories/CA-2001-30.html>) in late 2001 to provide specific information about compromises and remedies for dozens of various Unix systems.

U7.2 Operating Systems Affected

Nearly all Unix systems and many Linux systems come with a version of LPD installed and by default enabled.

U7.3 CVE Entries

[CVE-2001-0353](#), [CVE-1999-0299](#), [CVE-2000-0534](#), [CVE-2001-0670](#), [CAN-1999-0061](#), [CAN-2000-1208](#), [CAN-2001-0671](#)

U7.4 How to Determine if you are Vulnerable

Since every Unix or Linux operating system comes with some sort of print server installed, and since even those which use a replacement for LPD (like LPRng) call their service "lpd" or "in.lpd," you should check with your operating system vendor to verify that you are running the latest version or patch provided, and if not consider your system vulnerable.

U7.5 How to Protect Against It

Please see [CERT Advisory 2001-30](#) for specific remedy information for your operating system. Solaris users should also see [CERT Advisory 2001-15](#) and [Sun Security Bulletin #00206](#).

If your machine does not need to act as a print server for remote requests, you may be able to minimize the risk of future vulnerabilities in LPD by disabling the "in.lpd" service in inetd or xinetd. For inetd, comment out the "in.lpd" entry in `/etc/inetd.conf` or `/etc/inet/inetd.conf` and restart inetd. For xinetd, add a "disable = yes" line to the "in.lpd" file and restart xinetd. If you do need to service remote print requests, restrict what hosts can connect to in.lpd with [TCP Wrappers](#).

You can provide some protection against buffer overflows by enabling a non-executable stack on those operating systems that support this feature. While a non-executable stack will not protect against all buffer overflows, it can hinder the exploitation of some standard buffer overflow exploits publicly available on the Internet.

[Back to Top ^](#)

U8 Sendmail

U8.1 Description

Sendmail is the program that sends, receives, and forwards most electronic mail processed on Unix and Linux computers. Sendmail's widespread use on the Internet has historically made it a prime target of attackers, resulting in numerous exploits over the years.

Most of these exploits are successful only against older versions of the software. In fact, Sendmail has not had a 'high' severity vulnerability in more two years. Despite the fact that these older problems are well documented and have been repaired in newer releases, there remain so many outdated or mis-configured versions still in use today that Sendmail remains one of the most frequently attacked services.

The risks presented by running Sendmail can be grouped into two major categories: privilege escalation caused by buffer overflows, and improper configuration that allows your machine to be a relay for electronic mail from any other machine. The former is a problem on any system still running older versions of the code. The latter results from using either improper or default configuration files, and is a chief obstacle to fighting the proliferation of spam.

U8.2 Operating Systems Affected

Nearly all Unix and Linux systems come with a version of Sendmail installed and often by default enabled.

U8.3 CVE Entries

[CVE-1999-0206](#), [CVE-1999-0203](#), [CVE-1999-0204](#), [CVE-1999-0047](#), [CVE-2001-0236](#), [CAN-1999-0512](#), [CVE-1999-0130](#), [CVE-1999-0131](#), [CVE-1999-0393](#), [CVE-1999-1309](#), [CVE-2001-0653](#), [CVE-2000-0319](#), [CVE-1999-1109](#), [CVE-1999-0129](#), [CVE-1999-0095](#)

U8.4 How to Determine if you are Vulnerable

Sendmail has had a large number of vulnerabilities in the past. Don't always trust the version string returned by the daemon as that is just read from a text file on the system that may not have been updated properly.

Check to see what the latest version (if you built from source) or patch level (if it came packaged with your

operating system) is for Sendmail; if you are not running it, you are probably vulnerable.

U8.5 How to Protect Against It

The following steps should be taken to protect Sendmail:

1. Upgrade to the latest version and/or implement patches. The source code can be found at <http://www.sendmail.org/>. If your version of Sendmail came packaged with your operating system, patches should be available at your operating system vendor's website (various vendor-specific information, including compile-time and configuration suggestions, is also available at <http://www.sendmail.org/>).
2. Sendmail is typically enabled by default on most Unix and Linux systems, even those which are not acting as mail servers or mail relays. Do not run Sendmail in daemon mode (turn off the "-bd" switch) on these machines. You can still send mail from this system by invoking "sendmail -q" periodically to flush its outgoing queue.
3. If you must run Sendmail in daemon mode, ensure that your configuration is designed to relay mail appropriately and only for systems under your purview. See <http://www.sendmail.org/tips/relaying.html> and <http://www.sendmail.org/m4/anti-spam.html> for assistance in properly configuring your server. Starting with Sendmail 8.9.0, open relaying was disabled by default. However, many operating system vendors re-enabled it in their default configurations. If you are using the version of Sendmail which shipped with your operating system, take special care to ensure that your server is not used for relaying.
4. When you upgrade Sendmail binaries make sure to also update or verify the configuration file, as older configurations may still allow relaying even when running the newest code.

[Back to Top ^](#)

U9 BIND/DNS

U9.1 Description

The Berkeley Internet Name Domain (BIND) package is the most widely used implementation of the Domain Name Service (DNS), the system that allows one to locate a server on the Internet (or a local network) by using its name (e.g., www.sans.org) without having to know its specific IP address. The ubiquity of BIND has made it a frequent target of attack. While BIND developers have historically been quick to repair vulnerabilities, an inordinate number of outdated or misconfigured servers remain place and exposed to attack.

A number of factors contribute to this condition. Chief among them are administrators who are not aware of security upgrades, systems which are running BIND daemon (called "named") unnecessarily, and bad configuration files. Any of these can effect a denial of service, a buffer overflow or DNS cache poisoning. Among the most recently discovered BIND weaknesses was a denial of service, discussed in [CERT Advisory CA-2002-15](#). In this case, an attacker can send specific DNS packets to force an internal consistency check which itself is vulnerable and will cause the BIND daemon to shut down. Another was a buffer overflow attack, discussed in [CERT Advisory CA-2002-19](#), in which an attacker utilizes vulnerable implementations of the DNS resolver libraries. By sending malicious DNS responses, the attacker can explore this vulnerability and execute arbitrary code or even cause a denial of service.

In addition to the risk a vulnerable BIND poses to the server which hosts it, a single compromised machine may provide a platform for malicious activity targeting other machines on the Internet, or be used as a repository of illicit material without the administrator's knowledge.

U9.2 Operating Systems Affected

Nearly all Unix and Linux systems come with a version of BIND installed and often by default enabled. Binary versions of BIND do exist for Windows.

U9.3 CVE Entries

[CVE-1999-0009](#), [CVE-1999-0833](#), [CVE-2001-0010](#), [CVE-2001-0011](#), [CVE-2001-0013](#), [CVE-2001-0011](#), [CVE-1999-0024](#), [CVE-2001-0012](#), [CVE-1999-0837](#), [CVE-1999-0848](#), [CVE-1999-0849](#), [CAN-2002-0400](#)

U9.4 How to Determine if you are Vulnerable

If you are running a version of BIND that came with your operating system, verify that you are current with the patches released by your vendor. If you are running BIND as built from source from the [Internet Software Consortium \(ISC\)](#), ensure that you are using the latest version of BIND. Any unpatched or outdated version of the software is likely to be vulnerable.

For most systems, the command "named -v" will show the installed BIND version, enumerated as X.Y.Z where X is the major version, Y is the minor version, and Z is a patch level. There are currently three major versions for BIND: 4, 8 and 9. If you are running BIND built from source, you should eschew version 4 for the latest version of 8 or preferably 9. You can retrieve the latest source from the [ISC](#).

A more complete approach would be to use an updated vulnerability scanner to periodically check your DNS system against new flaws.

U9.5 How to Protect Against It

- *To generally protect against BIND vulnerabilities:*
 1. Disable the BIND daemon (called "named") on any system which is not specifically designated and authorized to be a DNS server. To prevent this change from being reversed, it may be wise to also remove the BIND software.
 2. Apply all vendor patches or upgrade your DNS Server to the latest version. For more information about hardening your BIND installation, see the articles about securing name services as referenced in CERT's [Unix Security Checklist](#).
 3. To complicate automated attacks or scans of your systems, hide the "Version String" banner in BIND by replacing the actual version of BIND with a bogus version number in the "named.conf" file options statement.
 4. Permit zone transfers only to secondary DNS servers in your domain. Disable zone transfers to parent or child domains, using delegation and forwarding instead.
 5. The Padded Cell: To prevent a compromised "named" from exposing your entire system, restrict BIND so that it runs as a non-privileged user in a chroot()ed directory. For BIND 9, see <http://www.losurs.org/docs/howto/Chroot-BIND.html>
 6. Disable recursion and glue fetching to defend against DNS cache poisoning
- *To protect against recently discovered BIND vulnerabilities:*
 1. For the Denial of Service Vulnerability on ISC BIND 9: <http://www.cert.org/advisories/CA-2002-15.html>
 2. For the Buffer Overflows in Multiple DNS Resolver Libraries: <http://www.cert.org/advisories/CA-2002-19.html>

[Back to Top ^](#)

U10 General Unix Authentication -- Accounts with No Passwords or Weak Passwords

U10.1 Description

Passwords, passphrases and security codes are used in virtually every interaction between users and information systems. Most forms of user authentication, as well as file and data protection, rely on user-supplied passwords. Since properly authenticated access is often not logged, or even if logged not likely to arouse suspicion, a compromised password is an opportunity to explore a system from the inside virtually undetected. An attacker would have complete access to any resources available to that user, and would be significantly closer to being able to access other accounts, nearby machines, and perhaps even root. Despite this threat, accounts with bad or empty passwords remain extremely common, and organizations with good password policy far too rare.

The most common password vulnerabilities are that (a) user accounts have weak or nonexistent passwords, (b) regardless of the strength of their password, users fail to protect it, (c) the operating system or additional software creates administrative accounts with weak or nonexistent passwords, and (d) password hashing algorithms are known and often hashes are stored such that they are visible by anyone. The best and most appropriate defense against these is a strong password policy which includes thorough instructions for good password habits and proactive checking of password integrity.

U10.2 Operating Systems Affected

Any operating system or application where users authenticate via a user ID and password.

U10.3 CVE Entries

[CAN-1999-0502](#)

U10.4 How to Determine if you are Vulnerable

On local systems, password hashes are stored in either `/etc/passwd` or `/etc/shadow`. `/etc/passwd` needs to be readable by all users on the network to permit authentication to complete. If that file also includes the password hashes, then any user with access to the system can read the hashes and attempt to break them with a password cracker. `/etc/shadow` can be used alternatively to store the hashes, and should only be readable by root. If your local accounts are not protected by `/etc/shadow`, then the risk to your passwords is extremely high.

If you use NIS, then password hashes are readable by all users and passwords are similarly high risks. This may also be the case with some implementations of LDAP as a network authentication service.

But even if password hashes are protected, passwords can be guessed by other means. Although there are observable symptoms of general password weakness, such as the existence of active accounts for users who have departed the organization or services which are not running, the only way to know for certain that each individual password is strong is to test all of them against the same password cracking tools used by attackers. *PLEASE NOTE: Never run a password scanner, even on systems for which you have root-like access, without explicit and preferably written permission from your employer. Administrators with the most benevolent of intentions have been fired for running password cracking tools without authority to do so.*

The best cracking tools available are:

- [Crack](#)
- [John the Ripper](#)
- [Symantec NetRecon](#)

U10.5 How to Protect Against It

The best and most appropriate defense against password weaknesses is a strong policy which includes thorough instructions to engender good password habits and proactive checking of password integrity.

1. *Assure that Passwords are Strong.* Given enough hardware and enough time, any password can be cracked by brute force. But there are simpler and very successful ways to learn passwords without such expense. Password crackers employ what are known as dictionary-style attacks. Since encryption methods are known, cracking utilities simply compare the encrypted form of a password against the encrypted forms of dictionary words (in many languages), proper names, and permutations of both. Therefore a password whose root in any way resembles such a word is highly susceptible to a dictionary attack. Many organizations instruct users to generate passwords by including combinations of alphanumeric and special characters, and users more often than not adhere by taking a word ("password") and converting letters to numbers or special characters ("pa\$\$w0rd"). Such permutations cannot protect against a dictionary attack: "pa\$\$w0rd" is as likely to be cracked as "password."

A good password, therefore cannot have a word or proper name as its root. A strong password policy should direct users to generate passwords from something more random, like a phrase, or the title of a book or song. By concatenating a longer string (taking the first letter of each word, or substituting a special character for a word, removing all the vowels, etc.), users can generate sufficiently long strings which combine alphanumeric and special characters in a way which dictionary attacks will have great difficulty cracking. And if the string is easy to remember, then the password should be as well.

Once users are given the proper instructions for generating good passwords, procedures should be put in place to assure that these instructions are followed. The best way to do this is by validating the password whenever the user changes it. Most flavors of Unix can use [Npasswd](#) as a front-end to check entered passwords against your password policy. PAM-enabled systems can also be extended to include [cracklib](#) (the libraries which accompany Crack).

If passwords cannot be verified against dictionary libraries when they are entered, then cracking utilities should be run in a stand-alone mode as part of routine scanning. *AGAIN PLEASE NOTE: Never run a password scanner, even on systems for which you have root-like access, without explicit and preferably*

written permission from your employer. Administrators with the most benevolent of intentions have been fired for running password cracking tools without authority to do so. Once you have acquired authority to run cracking utilities on your system, do so regularly on a protected machine. Users whose passwords are cracked should be notified confidentially and given instructions on how to choose a good password. Administrators and management should develop these procedures together, so that management can provide assistance when users do not respond to these notifications.

Another way to protect against nonexistent or weak passwords is to use an alternative form of authentication such as password-generating tokens or biometrics. If you are having trouble with weak passwords, use an alternative means of authenticating users.

2. **Protect Strong Passwords.** If you store password hashes in `/etc/passwd`, update your system to use `/etc/shadow`. If your system runs NIS or LDAP in such a way that hashes cannot be protected, anyone (even non-authenticated users) can read your password hashes and attempt cracking. You should therefore secure proper permission and run proactive cracking as a regular practice.

Even if passwords themselves are strong, accounts can be compromised if users do not protect their passwords. Good policy should include instructions that a user should never tell his or her password to anyone else, should never write a password down where it could be read by others, and should properly secure any files in which a password is stored to automate authentication (passwords are easier to protect when this practice is only used when absolutely necessary). Password aging should be enforced so that any passwords which slip through these rules are only vulnerable for a short window of time, and old passwords should not be reused. Make sure that the users are given warning and chances to change their password before it expires. When faced with the message: "your password has expired and must be changed," users will tend to pick a bad password.

3. **Tightly Control Accounts.**
 - o Any service-based or administrative accounts not in use should be disabled or removed. Any service-based or administrative accounts which are used should be given new and strong passwords.
 - o Audit the accounts on your systems and create a master list. Do not forget to check passwords on systems like routers and Internet-connected digital printers, copiers and printer controllers.
 - o Develop procedures for adding authorized accounts to the list, and for removing accounts when they are no longer in use.
 - o Validate the list on a regular basis to make sure no new accounts have been added and that unused accounts have been removed.
 - o Have rigid procedures for removing accounts when employees or contractors leave, or when the accounts are no longer required.
4. **Maintain Strong Password Policy for the Enterprise.** In addition to operating system or network service-level controls, there are comprehensive tools available to help manage good password policy. Symantec's Enterprise Security Manager (ESM) is a host-based monitoring tool that monitors any changes in policy, new account creation, and password strength. ESM will also attempt to crack passwords as it is performing a policy run on your network. ESM uses a client-manager environment: the agent is placed on the servers or workstations which in turn report to a centralized manager. Using a remote console, logs can be viewed and reports generated of the current status of the enterprise. ESM will monitor the audit logs and any change that has been made to the baseline of your network.

[Back to Top ^](#)

Appendix A – Common Vulnerable Ports

In this section, we list ports that are commonly probed and attacked. Blocking these ports is a minimum requirement for perimeter security, not a comprehensive firewall specification list. A far better rule is to block all unused ports. And even if you believe these ports are blocked, you should still actively monitor them to detect intrusion attempts. A warning is also in order: Blocking some of the ports in the following list may disable needed services. Please consider the potential effects of these recommendations before implementing them.

Keep in mind that blocking these ports is not a substitute for a comprehensive security solution. Even if the ports are blocked, an attacker who has gained access to your network via other means (a dial-up modem, a trojan e-mail attachment, or a person who is an organization insider, for example) can exploit these ports if not properly secured on every host system in your organization.

1. Login services-- telnet (23/tcp), SSH (22/tcp), FTP (21/tcp), NetBIOS (139/tcp), rlogin et al (512/tcp through 514/tcp)

2. RPC and NFS-- Portmap/rpcbind (111/tcp and 111/udp), NFS (2049/tcp and 2049/udp), lockd (4045/tcp and 4045/udp)
3. NetBIOS in Windows NT -- 135 (tcp and udp), 137 (udp), 138 (udp), 139 (tcp). Windows 2000 – earlier ports plus 445(tcp and udp)
4. X Windows -- 6000/tcp through 6255/tcp
5. Naming services-- DNS (53/udp) to all machines which are not DNS servers, DNS zone transfers (53/tcp) except from external secondaries, LDAP (389/tcp and 389/udp)
6. Mail-- SMTP (25/tcp) to all machines, which are not external mail relays, POP (109/tcp and 110/tcp), IMAP (143/tcp)
7. Web-- HTTP (80/tcp) and SSL (443/tcp) except to external Web servers, may also want to block common high-order HTTP port choices (8000/tcp, 8080/tcp, 8888/tcp, etc.)
8. "Small Services" -- ports below 20/tcp and 20/udp, time (37/tcp and 37/udp)
9. Miscellaneous-- TFTP (69/udp), finger (79/tcp), NNTP (119/tcp), NTP (123/udp), LPD (515/tcp), syslog (514/udp), SNMP (161/tcp and 161/udp, 162/tcp and 162/udp), BGP (179/tcp), SOCKS (1080/tcp)
10. ICMP-- block incoming echo request (ping and Windows traceroute), block outgoing echo replies, time exceeded, and destination unreachable messages **except** "packet too big" messages (type 3, code 4). (This item assumes that you are willing to forego the legitimate uses of ICMP echo request in order to block some known malicious uses.)

In addition to these ports, block "spoofed" addresses-- packets coming from outside your company sourced from internal addresses, private (RFC1918 and network 127) and IANA reserved addresses. Also block source routed packets or any packets with IP options set.

[Back to Top ^](#)

Appendix B – The Experts Who Helped Create The Top Twenty Vulnerable Service Lists

Jeff Campione, Federal Reserve Board -- Editor	Jeni Li, Arizona State University
Eric Cole, Editor, 2001 Edition	Nick Main, Cerberus IT, Australia
Ryan C. Barnett, Department of the Treasury/ATF	Jose Marquez, Alutiq Security and Technology
Chris Benjes, National Security Agency	Christopher Misra, University of Massachusetts
Matt Bishop, University of California, Davis	Stephen Northcutt, SANS Institute
Chris Brenton, SANS Institute	Craig Ozancin, Symantec
Pedro Paulo Ferreira Bueno, Open Communications Security, Brazil	Alan Paller, SANS Institute
Anton Chuvakin, Ph.D., netForensics	Ross Patel, Afentis, UK
Rob Clyde, Symantec	Marcus Ranum, ranum.com
Dr. Fred Cohen, Sandia National Laboratories	Chris Rouland, Internet Security Systems
Gerhard Eschelbeck, Qualys	Bruce Schneier, Counterpane Internet Security Inc.
Dan Ingevaldson, Internet Security Systems	Greg Shipley, Neohapsis
Erik Kamerling, Pragmeta Networks	Gene Spafford, Purdue University CERIAS
Gary Kessler, Gary Kessler Associates	Koon Yaw Tan, Infocomm Development Authority of Singapore
Valdis Kletnieks, Virginia Tech CIRT	Mike Torregrossa, University of Arizona
Jamie Lau, Internet Security Systems	Viriya Upatising, Loxley Information Services, Thailand
Scott Lawler, Veridias Information Solutions	Rick Wanner

People who helped prioritize the individual CVE entries to help define the tests to be used in the Top 20 scanners. For details on the process used, see www.sans.org/top20/testing.pdf

Charles Ajani, Standard Chartered Bank, London, UK	Susan Koski, Mellon Financial, Pittsburgh PA
Steven Anderson, Computer Sciences Corporation, North Kingstown RI	Kevin Liston, AT&T, Columbus OH
John Benninghoff, RBC Dain Rauscher, Minneapolis MN	André Mariën, Ubizen, Belgium
Layn Bro, BAE Systems, Denver CO	Fran McGowran, Deloitte & Touche, Dublin, Ireland, UK
Thomas Buehlmann, Phoenix AZ	Derek Milroy, Chicago IL
Ed Chan, NASA Ames Research Center, San Jose CA	Bruce Moore, Canadian Forces Network Operations Center, DND, Ottawa Canada
Andrew Clarke, Computer Solutions, White Plains NY	Castor Morales, Ft. Lauderdale FL
Brian Coogan, ManageSoft, Melbourne Australia	Luis Perez, Boston MA
Greg Cunningham, WSRC, Aiken SC	Reg Quinton, University of Waterloo, Ontario Canada
Paul Docherty, Portcullis Computer Security Limited, UK	Bartek Raszczyk, UWM Olsztyn, Olsztyn Poland
Arian Evans, U.S. Central Credit Union, Overland Park KS	Teppo Rissanen, Plasec Oy, Helsinki Finland
	Alan Rouse, N2 Broadband, Duluth GA

Rich Fuchs, Research Libraries Group,
Mountain View CA
Mark Gibbons, International Network Services,
Minneapolis MN
Dan Goldberg, Rochester NY
Shan Hemphill, Sacramento CA
Michael Hensing, Microsoft
Simon Horn, Brisbane Australia
Bruce Howard, Kanwal Computing Solutions, Jilliby
NSW Australia
Tyler Hudak, Akron OH
Delbert Hundley, MPRI Division of L-3COM, Norfolk VA
Chyuan-Horng Jang, Oak Brook IL
Kim Kelly, The George Washington University,
Washington DC
Martin Khoo, Singapore Computer Emergency
Response Team (SingCERT), Singapore

Denis Sanche, PWGSC ITSD/IPC, Hull, QC Canada
Felix Schallock, Ernst & Young, Vienna, Austria
Gaston Sloover, Fidelitas, Buenos Aires Argentina
Arthur Spencer, UMASS Medical School, Worcester
MA
Rick Squires
Jeff Stehlin, HP
Koon Yaw TAN, Infocomm Development Authority of
Singapore, Singapore
Steven Weil, Seitel Leeds & Associates, Seattle WA
Lance Wilson, Time Warner Cable/Broadband IS,
Orlando FL
Andrew Wortman, Naval Research Laboratory,
Washington DC
Carlos Zottman, Superior Tribunal de Justiça,
Brasilia Brazil

Additional security experts who helped with the 2001 Top Twenty and 2000 Top Ten lists which provided the foundation on which the 2002 Top Twenty is built.

Billy Austin, Intrusion.com
Phil Benchoff, Virginia Tech CIRT
Tina Bird, Counterpane Internet Security Inc.
Lee Brotzman, NASIRC Allied Technology Group Inc.
Mary Chaddock
Steve Christey, MITRE
Scott Conti, University of Massachusetts
Kelly Cooper, Genuity
Scott Craig, KMart
Sten Drescher, Tivoli Systems
Kathy Fithen, CERT Coordination Center
Nick FitzGerald, Computer Virus Consulting Ltd.
Igor Gashinsky, NetSec Inc.
Bill Hancock, Exodus Communications
Robert Harris, EDS
Shawn Hernan, CERT Coordination Center
Bill Hill, MITRE
Ron Jarrell, Virginia Tech CIRT
Jesper Johansson, Boston University
Christopher Klaus, Internet Security Systems
Clint Kreitner, Center for Internet Security
Jimmy Kuo, Network Associates Inc.
Jim Magdych, Network Associates Inc.
Dave Mann, BindView
Randy Marchany, Virginia Tech
Mark Martinec "Jozef Stefan" Institute
William McConnell, Trend Consulting Services

Peter Mell, National Institutes of Standards and
Technology
Larry Merritt, National Security Agency
Mudge, @stake
Tim Mullen, AnchorIS.com
Ron Nguyen, Ernst & Young
David Nolan, Arch Paging
Hal Pomeranz, Deer Run Associates
Chris Prosize, Foundstone Inc.
Jim Ransome
RAZOR Research - BindView Development
Martin Roesch, Snort
Vince Rowe, FBI, NIPC
Marcus Sachs, JTF-CNO US Department of Defense
Tony Sager, National Security Agency
Gene Schultz, Lawrence Berkeley Laboratory
Eric Schultze, Foundstone
Derek Simmel, Carnegie Mellon University
Ed Skoudis, Predictive Systems
Lance Spitzner, Sun Microsystems, GESS Team
Wayne Stenson, Honeywell
Jeff Stutzman
Frank Swift
Bob Todd, Advanced Research Corporation
Jeff Tricoli, FBI NIPC
Laurie Zirkle, Virginia Tech CIRT

[Back to Top ^](#)