Important NT Security Patches

Hotfixes appear in between service packs. We explain the simplest way to install large numbers of fixes, and discuss which fixes exist to help you plug the more important security loopholes.

By Simon Pride

In *Windows NT Service Pack 4* [PCNA 102, File: E1708] we listed several elements of Service Pack 4 that addressed security issues in NT. This article updates that list, surveying security issues that have arisen or been discovered since then and detailing how to fix them. Many of these patches came out first as hotfixes, and were subsequently incorporated into the next service pack.

Microsoft and other operating system vendors react quickly to new security threats to their products. Microsoft, in particular, responds in two ways:

- Service Packs are a regular series of software releases which address bugs in the operating system and close security loopholes or vulnerabilities.
- Hotfixes are patches which address a single bug or security issue and are released as interim fixes until the same protection can be supplied via the next service pack.

Managing Hotfixes

Managing hotfixes can be a difficult task. The situation following the release of Service Pack 3 was a particularly trying one for the systems administrator.

Firstly, there were over 40 different hotfixes released between Service Packs 3 and 4 and, secondly, many of the hotfixes were implemented as changes to the same system component such as TCPIP.SYS, the TCP/IP protocol driver. This meant that the order in which hotfixes were installed was absolutely crucial if the administrator was not to risk, at best, accidentally downgrading a system's hotfix level or, at worst, making the system unstable.

The hotfix application tool HOTFIX.EXE is of some help in listing the hotfixes already installed on a system, but unfortunately only reports installed hotfixes by the Microsoft Knowledge Base Q article number that describes the problem.

The same information is available by inspection of the registry key HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Hotfix, where again the fixes are listed as a series of subkeys named for the Q article number.

The hotfixes detailed below are all fully documented in Knowledge Base articles. To obtain a KB article, go to **support.microsoft.com/support/kb/articles/qxxx/x/xx.asp**, where xxxxxx is the six-digit KB article number.

Batch Processing

Once you have identified which hotfixes to apply, there is another obstacle to easy management. The hotfix installation procedure will normally cause a reboot after each hotfix, which makes upgrading a machine very time-consuming. You can apply a series of hotfixes in a batch process from a server share or removable drive by the following procedure:

1. Download the .EXE files for all the hotfixes you wish to apply.
2. Create a new folder called "hotfixes".
3. Within the new hotfixes folder, create one new folder for each hotfix to be applied.
4. Copy or move each hotfix .EXE file into its respective folder.
5. Open a command prompt in each folder in turn and issue the command hotfixfile /X, where hotfixfile is the name of the .EXE archive you downloaded in step 1 (these are often named after the Q article describing them, eg, Q242294.EXE). This will expand the archive into a set of installation files and the HOTFIX.EXE program.

In the root of the hotfixes folder create a .BAT or .CMD file which resembles Figure 1.

The -z argument means "do not reboot when update completes" and is vital in order for the batch process to continue after each update. The -m option means "run in unattended mode" and requires no action by the user.

Run the file to apply the hotfixes. Remember to construct the batch file so that the fixes are applied in the correct order.

Once the last hotfix has completed you should reboot the computer. As with service packs, if you add or alter any system component such that Windows NT prompts you for the original distribution CD, you will need to reapply both the current service pack and any of its subsequent hotfixes again.

Hotfixes For Security

Let us now look at important security-related hotfixes in order, starting with those that came after Service Pack 4. Most of these are fixed in Service Pack 6, so if you don't want to install them separately you can simply upgrade your machines to SP6.

DoS Attack

An attack on NT which sends either the Local Security Authority (LSASS.EXE) or the Spooler Service (SPOOLSS.EXE) to 100% utilisation of the processor is possible by opening multiple named pipe connections to these services and sending random data to them. The RPC (Remote Procedure Call) service will detect that the RPC requests are invalid, but in trying to send a response to the invalid caller and close the connection will go into a loop, using 100% of the CPU time, and additionally leaking memory. The affected computer will therefore slow to a halt and may eventually hang.

To prevent Denial of Service (DoS) from such an attack you should obtain and install nprpc-fix. Details are in Microsoft Knowledge Base article Q195733. The fix itself was incorporated into SP5.

Password Changes

This is a serious vulnerability affecting networks that have what Microsoft refers to as "downlevel" clients such as Windows for Workgroups, OS/2 or an Apple Macintosh. It was introduced by the changes made to logon validation in Service Pack 4.

The client software on Windows for Workgroups and the Microsoft UAM (User Authentication Module) for Macintosh uses the older, less secure LAN Manager hash of the password. When a user changes his or her password from an older system such as the ones mentioned above, NT Server will store the LAN Manager hash in the SAM (Security Accounts Manager) database, and set the stronger NT hash form of the password to NULL. Once the password has been changed this error allows anyone to log into that account from any computer which uses the NT hash for authentication (Windows 95, 98 and NT) using a blank password - a clear security violation.

To close this loophole obtain and install msv1-fix. Details are in KB article Q214840. The problem was fixed in SP5.

```
getadmin\hotfix.exe -z -m
simp-tcp\hotfix.exe -z -m
tear\hotfix.exe -z -m
srv\hotfix.exe -z -m
y2k\hotfix.exe -z -m
euro\hotfix.exe -z -m
lsa2\hotfix.exe -z -m
```

Figure 1 - Example batch file used to install hotfixes.

KnownDLLs Exploit

This exploit potentially allows an unprivileged user to gain local administrator privileges on an NT computer. The attack would require the attacker to write a program which manipulates NT system files in memory. While this may seem unlikely in the vast majority of enterprises, it is always possible for a single malicious individual to write a program that exploits this vulnerability in a way that end-users can use easily and then distribute it via the Internet.

The way it works is by using a loophole in the protection of NT system objects, which are the means by which NT controls access to processes and resources. When a system DLL (Dynamic Link Library) is required by two or more processes, NT economises on the use of memory by loading a single copy of the DLL into memory and mapping a copy into the process space of the calling process. When a process calls a function in such a DLL, NT refers to a system object called the KnownDLLs list, which gives the location in memory of the DLL called.

System DLLs can't be modified in memory, as NT's security system prevents it. However, the security protection on base system objects, of which the KnownDLLs list is one, is lax and allows all users read and write access to the KnownDLLs list. An attacker would therefore first write a malicious program as a DLL and give it the name of an NT system DLL. They would also need to write another program which manipulates the KnownDLLS list.

He or she would then go to a computer running NT and run the program, loading the malicious DLL. Then they would run the second program, which accesses the unprotected KnownDLLs list and patches it so that calls to the NT system DLL are instead diverted to the malicious version now in memory. As system DLLs run with an elevated security context the malicious DLL could more or less do anything its writer wished, such as gaining Administrator access to the workstation. This access can then be used to do whatever the attacker wanted, including installing Trojan horses, password-capturing utilities or network sniffers to capture sensitive network traffic.

The hotfix to correct the problem patches the Session Manager (SMSS.EXE) to increase the levels of protection possible on the base system objects. Note that, as with some other hotfixes, simply applying the patch does not itself fix the problem, it merely makes it possible for the problem to be fixed. Once you have applied Smss-fix, you must add a registry subkey of "ProtectionMode" to HKEY_LOCAL_MACHINE\Sytem-\CurrentControlSet\Control\Session Manager and set its value to be a REG_DWORD of 0x1. Once you have done this and rebooted, the computer will be protected from this attack.

More details are in KB article Q218473, and the fix was first posted in Service Pack 5.

MaxRequestThreads

Exceeding the MaxRequestThreads number may crash NT. This is another Denial of Service attack, this time targeting the Client Server Runtime Subsystem (CSRSS). NT's architecture is based around a client-server model, where every time a user process wishes the operating system to do something on its behalf it must issue a request, as a client, to the OS and await the result of its request. In this way, unprivileged user processes can use facilities that only privileged processes have access to.

The Client Server Runtime System is the process that manages these requests. This attack depends again on malicious code being executed at the computer - code that makes a request to a system process that requires user input, but never supplies that input. The thread that CSRSS.EXE creates to

NT Security Patches

service the request waits forever for the input which is not supplied. When 16 such processes have been started, the pool of worker threads available by default to service the requests is exhausted, and NT will hang. The patch changes CSRSS.EXE so that the last worker thread available is never used to process a request requiring user input.

More details are in KB article Q233323, and the first Service Pack to incorporate the fix was 6.

Buffer Overflows

Buffer overflows are among the oldest techniques for attacking operating systems and their services. In simple terms, a program running on the computer will accept data from a user or another program, and will allocate storage to receive that request. Unless the program takes special care to check that the size of the data it is being passed is less than or equal to the allocated storage prepared for it, the incoming data will overflow the storage or buffer.

What happens then depends largely on the receiving application and operating system. In the early days of Unix-based Internet services, the overflowing portion of the data could "escape" into a command shell, and if the overflowing requests were composed of valid shell commands they could execute in a shell which had the same privilege level as the process they escaped from - often a level much higher than that a normal user of the system would have. On Windows NT, the main problem with buffer overflow exploits is that they can place arbitrary code onto the stack, which is then executed.

In one particular exploit, the overflow is created by any user adding data to the RAS (Remote Access Service, Windows NT's version of Dial-Up Networking) phonebook such that it overflows the component that processes entries, and gives the same kind of ability to execute arbitrary code as above, and because the RAS client-end software runs as LocalSystem the code could take any action its author wished.

This vulnerability only affects computers that have RAS installed, and using best practice the average networked workstation will not have RAS installed. For those computers that are using this service, applying ras-fix will remove the vulnerability.

More details are in KB article Q230677. If you have Service Pack 6, you already have this patch.

Malformed API Call

A malformed API call hangs the LSA (Local Security Authority) process. This is another Denial of Service attack, and again requires the attacker to write a program to exploit the vulnerability. The LSA is the process on NT which is responsible for enforcing security and controlling access to resources. Programs can call security-related functions using the Win32 APIs provided for accessing the LSA.

However, some of the LSA APIs do not handle invalid arguments correctly, and a call made to such API functions with bad data will cause the LSA to hang. The attacker would therefore write a small program which knowingly passed invalid arguments to an LSA API function, and execute it on the attacked computer, causing the LSA to hang. As every request for access to a process or an object is referred to the LSA to see if the requester is authorised to access the desired target, the effect is to prevent the computer from operating.

The only way to recover from the situation is to reboot the affected computer. Lsa3-fix removes the vulnerability by improving the argument-checking carried out by the LSA API calls.

More details of this problem, which was fixed in SP6, are in KB article Q231457.

IOCTL Buffers

Although Windows NT has an architecture which passes all processes' requests to interact with the hardware via the Executive, it also provides a mechanism for processes to interact directly with device drivers. The interface between the process and the driver is called an Input Output Control (IOCTL), and is most commonly used to give a process access to the mouse and / or keyboard.

Unfortunately the security descriptors on the IOCTLs for mouse and keyboard have been left wide open, so that an unprivileged process can take control of the interface and block any other process from access to them. The attacker would write a small program which captured the mouse and keyboard IOCTLs and then did nothing. This would have the effect of disabling the mouse and keyboard, and the computer would need to be power-cycled to restore access.

This may not seem a particularly grave vulnerability, and in the case of Windows NT Workstation it is not, but consider that a Windows Terminal Server shares a single set of IOCTLs among its users. An attack of this kind launched by a single user will therefore disable all sessions on the server, potentially affecting many users at once.

The patch exists in versions for standard NT and NT Terminal Server Edition. Apply the relevant version to your computers to defend against this exploit. Be aware that the Q article describing this problem still lists the Terminal Server version of the patch as not having been fully regression-tested. The fully-tested version is contained in Service Pack 5 for Windows NT Terminal Server Edition, which should be used in preference.

The fix is explained further in KB article Q236359. The fix is incorporated into SP6 for NT Workstation and Server, and SP5 for the Terminal Server edition.

Malformed Help File

The Help system is at the root of yet another buffer overflow vulnerability. The vulnerability arises out of the combination of lax permissions on the folder %SYSTEMROOT%\Help and an unchecked buffer in WINHELP.-EXE.

The attacker would create a modified help file containing binary data containing the exploit (this must be done to an already existing Help file, using a hexadecimal editor, as the Help Compiler cannot be used to generate invalid Help files) and then add

it to the Help folder, whose default permissions grant read and write access to all users. If the attacker replaces a commonly-used NT Help file the exploit will be triggered the first time a user calls up Help for that topic.

Obviously an immediate workaround is to tighten the permissions on the Help files in the folder to prevent users modifying them in any way. Do make sure, however, that you have opened each help file in the folder once and searched for a word in the file in order to generate the .GID and .FTS files used by the Help engine. If a user without relevant permissions opens a Help file which has no corresponding .GID or .FTS files, the Help engine will attempt to create these, fail, and exit, leaving the user without access to help. The patch updates WINHELP.EXE to remove the unchecked buffer.

More details are in KB article Q231605. The patch is incorporated into SP6.

Phone Dialer

A buffer overflow in the Phone Dialer applet can cause arbitrary code to be run by the Administrator account. This is yet another buffer overflow attack, but the nature of the exploit is quite subtle. The Phone Dialer accessory (if installed) is used to cause an attached modem to call a phone number held in an information manager such as Outlook. It runs in the security context of the logged-in user, but has an unchecked buffer, and its .INI file is also unprotected by default.

The attacker logs onto an NT Workstation that has Phone Dialer installed and modifies DIALER.INI by hand, adding data to an entry in the section headed [Last Dialed Numbers] so that the entry is longer than 128 characters but shorter than 256 characters. The arbitrary code to be run is contained in the data above 128 characters. When an Administrator logs onto the same computer and uses Phone Dialer, the .INI file is parsed and the overflow occurs. Arbitrary code could then be run in the security context of Administrator, with the usual freedom to compromise the system in any way.

Microsoft Knowledge Base article number Q237185, which describes the problem, omits to mention any security-related issues surrounding this exploit, describing it merely as an Access Violation bug. Dialer-fix removes the unchecked buffer. The problem is also fixed in SP6.

TCP Source Routing

TCP source routing creates a vulnerability. This is a problem introduced by Service Pack 5 which allowed source routing to be disabled. Source routing refers to a means by which TCP / IP packets can be directed to take only one particular route of all those available from one host to another. This sometimes has legitimate uses, such as determining the addresses of routers in a network. However, it can be used by an attacker to discover details of your network topology.

Service Pack 5 introduced a means for administrators to disable source routing (see Microsoft Knowledge Base article number Q217336) but it was discovered after the service pack was issued that, if a data structure in an IP packet called the Route Pointer Field contained incorrect data, it could bypass disabled source routing and probe the network again. The vulnerability does not just concern multihomed NT computers acting as routers, but extends to workstations, which can also be used to launch routing attacks. The patch contained in spoof-fix will correct this issue on both single and multihomed computers. It's also fixed by Service Pack 6.

RASMAN

A malicious user may cause a different program to run in place of RASMAN. RASMAN is the application used on NT to manage dial-up network connections, and runs as an NT Service. It therefore has a high privilege level. When a process on NT needs to interact with a service, NT consults the Service Control Manager (SCM) to determine the name and location of the service. Normally only Administrators and privileged programs can use the SCM to manage services, because the SCM consults the security settings, called Discretionary Access Control Lists (DACL), before carrying out any operation on running services.

However, the default DACL for RASMAN.EXE has an entry (Access Control Entry, or ACE) which allows Everyone to manage the RASMAN service. A malicious user could therefore replace RASMAN.EXE with his or her own code, and then use SCM programmatically to run his or her code in place of RASMAN. As RASMAN runs with SYSTEM privilege, the attacker's code could potentially take any action it wished in order to compromise the system. Running Rasman-fix will reset the faulty permissions in the ACE back to the intended values.

Rasman-fix is explained in KB article Q242294. The problem is not fixed in SP6 so you need to obtain and install the patch. Or wait for SP7.

PCNA

The Author

Simon Pride runs the PC support department at Cambridge University and specialises in NetWare, Windows NT and networks. He can be contacted as simon.pride@itp-journals.com.

New Reviews from Tech Support Alert

Anti-Trojan Software Reviews

A detailed review of six of the best anti trojan software programs. Two products were impressive with a clear gap between these and other contenders in their ability to detect and remove dangerous modern trojans.

Inkjet Printer Cartridge Suppliers

Everyone gets inundated by hundreds of ads for inkjet printer cartridges, all claiming to be the cheapest or best. But which vendor do you believe? Our editors decided to put them to the test by anonymously buying printer cartridges and testing them in our office inkjet printers. Many suppliers disappointed but we came up with several web sites that offer good quality cheap inkjet cartridges with impressive customer service.

Windows Backup Software

In this review we looked at 18 different backup software products for home or SOHO use. In the end we could only recommend six though only two were good enough to get our “Editor’s Choice” award

The 46 Best Freeware Programs

There are many free utilities that perform as well or better than expensive commercial products. Our Editor Ian Richards picks out his selection of the very best freeware programs and he comes up with some real gems.