

[NT] Windows 2000 Default Permissions Could Allow Trojan Horse Program

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-10/0136.html>

From: support@securiteam.com

Date: 10/31/02

From: support@securiteam.com

To: list@securiteam.com

Date: 31 Oct 2002 11:35:54 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

When was the last time you checked your server's security?

How about a monthly report?

<http://www.AutomatedScanning.com> – Know that you're safe.

Windows 2000 Default Permissions Could Allow Trojan Horse Program

SUMMARY

On Windows 2000, the default permissions provide the Everyone group with Full access (Everyone:F) on the system root folder (typically, C:\). In most cases, the system root is not in the search path. However, under certain conditions – for instance, during logon or when applications are invoked directly from the Windows desktop via Start | Run – it can be.

This situation gives rise to a scenario that could enable an attacker to mount a Trojan horse attack against other users of the same system, by creating a program in the system root with the same name as some commonly used program, then waiting for another user to subsequently log onto the system and invoke the program. The Trojan horse program would execute with the user's own privileges, thereby enabling it to take any action that the user could take.

The simplest attack scenario would be one in which the attacker knew that a particular system program was invoked by a logon script. In that case, the attacker could create a Trojan horse with the same name as the system program, which would then be executed by the logon script the next time someone logged onto the system. Other scenarios almost certainly would require significantly greater user interaction – for instance, convincing a user to start a particular program via Start | Run – and would

Securiteam: [NT] Windows 2000 Default Permissions Could Allow Trojan Horse Program

necessitate the use of social engineering.

The systems primarily at risk from this vulnerability would be workstations that are shared between multiple users, and local terminal server sessions. Other systems would be at significantly less risk:

- * Workstations that are not shared between users would be at no risk, because the attacker would require the ability to log onto the system in order to place the Trojan horse.

- * Servers would be at no risk, if standard best practices have been followed that advocate only allowing trusted users to log onto them.

- * Remote Terminal server sessions would be at little risk, because each user's environment is isolated. That is, the system root is never the current folder – instead, the user's Documents and Settings folder is, but the permissions on this folder would not enable an attacker to place a Trojan horse there.

DETAILS

Affected Software:

- * Microsoft Windows 2000

Mitigating factors:

- * An attacker would require the ability to log onto the system interactively in order to place the Trojan horse program. It could not be placed remotely

- * As discussed above, dedicated workstations, servers and remote terminal server sessions would be at less risk (or, in some cases, none at all) from the vulnerability.

Patch availability:

Download locations for this patch

- * This vulnerability requires an administrative procedure rather than a patch. The needed changes are discussed in the FAQ.

What's the scope of the vulnerability?

This is a Trojan Horse vulnerability. An attacker who was able to successfully exploit this vulnerability could place a program on a system that, at some later point, could be inadvertently run by another user. The program would run with the privileges associated with the user, and could take any action the user could take, including loading and running programs, altering data on the system, reformatting the hard drive, and so forth.

In most cases, exploiting the vulnerability would not be a straightforward task, but would instead require at least some social engineering on the attacker's part. In addition, the vulnerability does not affect all systems equally. Instead, workstations that are shared between users would

Securiteam: [NT] Windows 2000 Default Permissions Could Allow Trojan Horse Program

be at greatest risk, and servers – including terminal servers – would typically be at significantly less risk.

What causes the vulnerability?

The vulnerability results because of a combination of two factors: the default permissions on the system root folder, and the search path Windows uses when locating programs. The former would enable an attacker to place a Trojan horse program on the system; the latter could cause it to be inadvertently executed by a subsequent user on the system.

What's the system root folder, and what does it do?

The system root folder is the topmost folder on the drive that contains the operating system (typically, C:\). The primary purpose of the system root is to provide a logical "anchor point" for the file system – a reference point from which other files and folders can be located. No sensitive files are stored there by default.

What are the default permissions on the system root?

The default permissions are Everyone Full Access. This allows any user who can log onto the system to read files located on the system root, write new files there (or modify existing ones), or execute programs that reside there.

If there aren't any sensitive files stored on the system root, why do the default permissions pose a security vulnerability?

The reason has to do with the way Windows locates programs. In a nutshell, there are cases in which the system root can become the first place the system looks when searching for a program. If an attacker had previously created a program, given it the same name as a frequently requested program, and placed it in the system root, the attacker's program – rather than the legitimate program – could execute.

How does Windows locate programs?

If a program is invoked using an absolute path – that is, if the caller specifies the exact location of the program – Windows will check that location and only that location, and return an error if the program isn't found there. However, if only the name of the program was provided, Windows will undertake a search process to find and execute it. First, it checks within the so-called current folder – that is, whatever folder the user (or calling application) happens to be working in at the time – then checks the folders specified in an environment variable called %PATH%.

It's the second case that plays a role in this vulnerability.

Specifically, there are cases in which the system root can become the current folder. When this happens, the system root will be the first place Windows looks for programs. Under these circumstances, if a program in the system root had the same name as, say, a legitimate system program, the bogus program would have precedence in the search order and be executed instead of the system file.

Securiteam: [NT] Windows 2000 Default Permissions Could Allow Trojan Horse Program

Under what circumstances is the system root also the current folder?

It's not possible to provide an all-encompassing answer, because it depends in some degree on the specific way the system is configured.

However, two examples of circumstances in which this occurs in default configurations include:

- * At system startup. Specifically, when logon scripts are run, the system root is the current folder.

- * When launching programs from the Windows desktop. For instance, if a program is started using the Start | Run command, or if Task Manager is started via the ctrl-alt-del sequence, the system root is the current folder.

It's worth pointing out that the foregoing is not true in the case of a remote terminal server session. Terminal Server isolates each user's session so, in the cases mentioned above, the current folder would be the user's own Documents and Settings folder rather than the system root.

What could an attacker do via this vulnerability?

An attacker who had the ability to log onto a system that's shared with other users could place a program in the system root, in the hope that a subsequent user would inadvertently run it. The attacker's program, if run, would do so with the user's own privileges, and would be able to do anything that the user could do.

The attack would likely proceed along the following lines:

- * The attacker would log onto the system and identify a program that he or she believed other users might invoke. For purposes of illustration, let's assume that the program is named x.exe.

- * The attacker would create a hostile program, name it x.exe, and put it in the system root.

- * At some later point, another user might log onto the system and invoke x.exe. If this happened during a time when the system root was the current folder, the attacker's version of x.exe rather than the legitimate version would run.

Who could exploit the vulnerability?

In order to exploit the vulnerability, the attacker would need valid logon credentials on the system – the Trojan horse could not be emplaced remotely. However, simply putting the file on the system would not be enough. To actually make the program to run, the attacker would also need some means of getting the user to invoke it.

How might the attacker get the user to invoke the Trojan horse program?

The simplest scenario would be one in which the attacker knew that a logon script would be run anytime a user logged onto the system, and would call a particular system program. In this case, the attacker could simply give the Trojan horse program a matching name and wait for another user to log

Securiteam: [NT] Windows 2000 Default Permissions Could Allow Trojan Horse Program

on. Other scenarios would be more difficult to carry out and almost certainly require a significant degree of social engineering. For instance, the attacker could attempt to create a scenario in which the user would be likely to invoke a particular program via the Start | Run command, open Task Manager via ctrl–alt–del, and so forth.

You said that the system root is the current folder for the Window desktop. Does that mean that clicking on an icon on the desktop could cause a Trojan horse to run?

No. Desktop icons, in almost every case, use an absolute path name to reference the programs they're associated with. As discussed above, when an absolute path is used to invoke a program, the search path never comes into play.

What systems are at great risk from the vulnerability?

Shared workstations are far and away the most likely systems to be affected by this vulnerability, but terminal servers can also be at risk if the user logs on locally instead of via a remote session. The reason for this can be most easily seen through a process of elimination.

- * Workstations that aren't shared are at no risk from the vulnerability. The vulnerability could only be exploited if the attacker could plant the Trojan horse and then persuade another user to log onto the same machine.

- * Servers would tend to be at less risk, if standard security practices have been followed and only trusted users are allowed to log onto them. Without the ability to log onto the system interactively, an attacker would have no way to place the Trojan horse.

- * Remote Terminal Server sessions would also be at significantly less risk, because each user's session is isolated. That is, system root doesn't ever become the current folder – the user's Documents and Settings folder does, but permissions on that folder do not allow anyone but the user and the system administrator to write to it.

Does the vulnerability affect Windows NT 4.0?

In Windows NT 4.0, all folders – not just the system root – default to Everyone Full Access. As a result, while it would be possible for an attacker to mount a Trojan horse attack through the same scenario as described above, it would be simpler for him or her to simply overwrite system files directly. However, the need to appropriately configure the permissions on Windows NT 4.0 systems is well–known to most system administrators, and is discussed in Microsoft's security checklists for Windows NT 4.0 as well as most third–party guides.

Does the vulnerability affect Windows XP?

In most cases, no. The default permissions for the system root in Windows XP allow only read access. The single case in which this is not true is if a Windows 2000 system is upgraded to Windows XP – in this case, the existing permissions on all folders, including the system root, are retained. However, even then, the vulnerability would be difficult to

Securiteam: [NT] Windows 2000 Default Permissions Could Allow Trojan Horse Program

exploit because the current directory for the Windows XP desktop is the user's Documents and Settings folder, rather than the system root.

Why isn't there a patch for this vulnerability?

A patch simply isn't practical for this case. The right permissions vary from organization to organization and can depend on factors like the specific applications in use. Any set of permissions set by a patch would almost certainly need to be adjusted in many cases, and as a result we believe the most effective way to remediate the problem is for system administrators to set the permissions that are right for their organizations.

What's a good baseline set of permissions?

The default permissions for Windows XP can serve as a guide for a set of permissions which have been thoroughly designed and tested. The default permissions for the root directory on the system drive for Windows XP are:

- * Administrators: Full (This Folder, Subfolder and Files)
- * Creators Owners: Full (Subfolders and Files)
- * System: Full (This Folder, Subfolder and Files)
- * Everyone: Read and Execute (This Folder Only)

Can I use security templates to apply the new permissions?

Yes. Customers using a security template can add the following to the [File Security] section to set the permissions to be the same as those for Windows XP.

```
"%SystemDrive%\",0,"D:AR(A;OICI;GA;;;BA)(A;OICI;GA;;;SY)(A;OICIO;GA;;;CO)
(A;CIOI;GRGX;;;BU)(A;CI;0x00000004;;;BU)(A;CIIO;0x00000002;;;BU)(A;;GRGX;;;WD)"
```

ADDITIONAL INFORMATION

The information has been provided by

<mailto:0_40079_E51E4D7D-DECD-43AE-9A29-36080E8D4C3C_US@Newsletters.Microsoft.com>
Microsoft Product Security.

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

list-unsubscribe@securiteam.com

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@securiteam.com

=====

=====

DISCLAIMER:

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.

Securiteam: [NT] Windows 2000 Default Permissions Could Allow Trojan Horse Program

- **Previous message:** support@securiteam.com: "[NT] AN HTTPD Cross-Site Scripting Vulnerability"
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)