

# [NT] NTFS Hard Links Subvert Auditing

*Source:* <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-08/0052.html>

---

*From:* [support@securiteam.com](mailto:support@securiteam.com)

*Date:* 08/18/02

From: [support@securiteam.com](mailto:support@securiteam.com)

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: Sun, 18 Aug 2002 14:22:03 +0200 (CEST)

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.

-----

NTFS Hard Links Subvert Auditing

---

## SUMMARY

The NTFS file system supports hard links. A hard link is another directory entry that points to the same physical file on disk. This allows you to have multiple pathnames to the same file within a partition. Once the hard link is created any file I/O operations on the hard link act as if they were done on the original file. The ACL of the original file are used. The auditing entries for the original file are used.

The auditing mechanism of Windows NT and Windows 2000 does not understand hard links so it produces some erroneous results. The results allow an attacker to access files through hard links such that the name of the file being accessed does not appear in the security event log. Instead, the file name of the hard link appears in the event log. The hard link can be deleted after accessing the file thus eliminating any trace of the file I/O activity.

Since the ACL on the file is enforced, the hard link does not grant the user any extra privileges. The hard link does however allow a user to access the file within her privileges without leaving an audit trail. Since this problem has existed for many years all archived audit logs are suspect.

## DETAILS

## Securiteam: [NT] NTFS Hard Links Subvert Auditing

NTFS has always supported hard links in order to support the POSIX subsystem that requires them. They are seldom used by NT users. Windows NT 3.51 and NT 4.0 have the Win32 API function BackupWrite() which can create hard links. Windows 2000 introduced a new, simpler Win32 API function CreateHardLink(). The usage of these functions, as well as a sample hard link creation program, in were documented by Microsoft in a knowledge base article:

Q234727 HOWTO: Create Hard Links in Windows NT and Windows 2000  
<<http://support.microsoft.com/support/kb/articles/Q234/7/27.ASP>>  
<http://support.microsoft.com/support/kb/articles/Q234/7/27.ASP>

(Unfortunately, this KB article has been deleted.)

Note: If you are going to compile and use the Microsoft example, you will want to make one small change. The CreateFile function does not need the FILE\_WRITE\_ATTRIBUTES flag. Elimination of this flag allows you to create hard links without creating a WriteAttributes access event. Chris made this change to the ln.exe and ran it for his examples below.

If full auditing of a file is enabled, one entry will be created in the security event log when the hard link is created. This event is ReadAttributes. This same audit event is generated if a user views the properties of a file. If ReadAttributes auditing is not enabled then no auditing event will be generated when the hard link is created.

It is worth noting that since the ReadAttributes Success event is an event that occurs when the properties of a file are read, it is an event that is not often audited. If this event is not audited, there is no trace of the hard link creation in the event log.

For the example however, we will assume the most secure and stringent auditing of all events.

Example:

Using Windows 2000.

1. Create a file c:\audited\foo.txt
2. Enable auditing of all events for c:\audited\foo.txt
3. Create a hard link named c:\temp\link.txt that links to c:\audited\foo.txt using ln.exe compiled from KB  
<<http://support.microsoft.com/support/kb/articles/Q234/7/27.ASP>> Q234727  
in c:\audited\foo.txt c:\temp\link.txt

4. Security log will show a Success Audit:

Object Open:  
Object Server: Security  
Object Type: File  
Object Name: c:\audited\foo.txt  
New Handle ID: 48  
Operation ID: {0,14421507}

## Securiteam: [NT] NTFS Hard Links Subvert Auditing

Process ID: 1148  
Primary User Name: user  
Primary Domain: DOMAIN  
Primary Logon ID: (0x0,0xA8F7)  
Client User Name: –  
Client Domain: –  
Client Logon ID: –  
Accesses SYNCHRONIZE  
          ReadAttributes  
Privileges –

To the audit reviewer it looks like the user has read the properties of c:\audited\foo.txt. There is no trace that c:\temp\link.txt is linked to foo.txt.

5. Read the file c:\temp\link.txt. Security log will show a Success Audit:

Object Open:  
Object Server: Security  
Object Type: File  
Object Name: c:\temp\link.txt  
New Handle ID: 96  
Operation ID: {0,14425896}  
Process ID: 1364  
Primary User Name: user  
Primary Domain: DOMAIN  
Primary Logon ID: (0x0,0xA8F7)  
Client User Name: –  
Client Domain: –  
Client Logon ID: –  
Accesses READ\_CONTROL  
          SYNCHRONIZE  
          ReadData (or ListDirectory)  
          ReadEA  
          ReadAttributes  
Privileges –

To the audit reviewer it looks like the user has read the data of c:\temp\link.txt when they have really read the data in foo.txt.

An audit event was recorded when the file was read but it contains the \*wrong\* file name. There is no audit entry for the link creation so there is no correlation in the audit log connecting the new file name with the original file that is being audited. Because of the lack of connection, we were able to read the contents of the file c:\audited\foo.txt without a ReadData audit event occurring on that file name.

After the file has been read or copied, the hard link can be deleted thus eliminating any traces of malfeasance.

Vendor Response:

The vendor was informed of this issue on 8/15/2000. It was determined that

## Securiteam: [NT] NTFS Hard Links Subvert Auditing

the issue was too risky to fix in a Hotfix patch so the fix was scheduled for Windows 2000 SP3. XP and .Net server beta were fixed before they shipped.

The solution was to this vulnerability was to create a new "Hard link creation attempt" audit event. This creates a audit entry that connects the new hard link file name to the target file name.

The audit entry looks like this:

Event Type: Success Audit

Event Source: Security

Event Category: Object Access

Event ID: 568

Date: 8/5/2002

Time: 6:29:32 PM

User: DOMAIN\user

Computer: COMPUTER

Description:

Hard link creation attempt:

Primary User Name: user

Primary Domain: DOMIAN

Primary Logon ID: (0x0,0xFF10)

File Name: C:\audited\foo.txt

Link Name: C:\temp\link.txt

A tool has also been created so that you can search for hard links that already exist on your system prior to installing SP3. This is recommended if you are auditing sensitive files on a system that has multiple user access.

<http://www.microsoft.com/windows2000/techinfo/reskit/tools/new/hlscan-o.asp>  
<http://www.microsoft.com/windows2000/techinfo/reskit/tools/new/hlscan-o.asp>

Recommendations:

Apply the fix. There really is no workaround to the problem. All audit logs created before installing the fix are suspect.

If you are auditing sensitive files on a system that has multiple users (with different access settings) you should search for all hard links that exist on your system prior to installing the patch.

### ADDITIONAL INFORMATION

The information has been provided by <<mailto:cwysopal@atstake.com>> Chris Wysopal of <<mailto:advisoriesl@atstake.com>> @stake Advisories.

=====

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:

Securiteam: [NT] NTFS Hard Links Subvert Auditing

[list-unsubscribe@securiteam.com](mailto:list-unsubscribe@securiteam.com)

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

---

- ***Previous message:*** [support@securiteam.com](mailto:support@securiteam.com): "[\[NT\] Flaw in Network Connection Manager Could Enable Privilege Elevation](#)"
- ***Messages sorted by:*** [\[ date \]](#) [\[ thread \]](#) [\[ subject \]](#) [\[ author \]](#) [\[ attachment \]](#)