

[NT] Windows Animated Cursor Stack Overflow Vulnerability (0-Day)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2007-04/msg00017.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 4 Apr 2007 16:41:05 +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

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.
<http://www.securiteam.com/maillinglist.html>

Windows Animated Cursor Stack Overflow Vulnerability (0-Day)

SUMMARY

Determina Security Research has discovered a vulnerability in the USER32.DLL code responsible for loading animated cursor (.ANI) files. This vulnerability can be exploited by a malicious web page or HTML email message and results in remote code execution with the privileges of the logged-in user. The vulnerable code is present in all versions of Windows up to and including Windows Vista. All applications that use the standard Windows API for loading cursors and icons are affected. This includes Windows Explorer, Internet Explorer, Mozilla Firefox, Outlook and others.

Microsoft fixed a closely related vulnerability with the MS05-002 security update, but their fix was incomplete. Determina Security Research was able to bypass the MS05-002 patch and develop a proof-of-concept exploit that works on fully-patched Windows systems.

This vulnerability was independently disclosed and has been actively exploited since Mar 28, prompting the release of this advisory. Exploitation details have been omitted from the advisory until a vendor patch is available.

[NT] Windows Animated Cursor Stack Overflow Vulnerability (0-Day)

DETAILS

Vulnerable Systems:

- * Microsoft Windows 2000 Service Pack 4
- * Microsoft Windows XP Service Pack 2
- * Microsoft Windows XP 64-Bit Edition Version 2003 (Itanium)
- * Microsoft Windows XP Professional x64 Edition
- * Microsoft Windows Server 2003
- * Microsoft Windows Server 2003 for Itanium-based Systems
- * Microsoft Windows Server 2003 Service Pack 1
- * Microsoft Windows Server 2003 Service Pack 2
- * Microsoft Windows Server 2003 with SP1 for Itanium-based Systems and Microsoft Windows Server 2003 with SP2 for Itanium-based Systems
- * Microsoft Windows Server 2003 with SP2 for Itanium-based Systems
- * Microsoft Windows Server 2003 x64 Edition
- * Microsoft Windows Server 2003 x64 Edition Service Pack 2
- * Microsoft Windows Vista

The ANI file format is used for storing animated cursors. The format is based on the RIFF multimedia file format and consists of a series of tagged chunks containing variable sized data. Each chunk starts with a 4 byte ASCII tag, followed by a dword specifying the size of the data contained in the chunk.

```
struct ANIChunk
{
    char tag[4]; // ASCII tag
    DWORD size; // length of data in bytes
    char data[size]; // variable sized data
}
```

One of the chunks in an ANI file is the anih chunk, which contains a 36-byte animation header structure. The buffer overflow fixed in <http://www.microsoft.com/technet/security/Bulletin/MS05-002.msp> was in the LoadCursorIconFromFileMap function. The vulnerable code did not validate the length of the anih chunk before reading the chunk data into fixed size buffer on the stack.

The pseudo code of the vulnerable function is given below:

```
int LoadCursorIconFromFileMap(struct MappedFile* file, ...)
{
    struct ANIChunk chunk;
    struct ANIHeader header; // 36 byte structure
    ...

    // read the first 8 bytes of the chunk
    ReadTag(file, &chunk);

    if (chunk.tag == 'anih') {
```

[NT] Windows Animated Cursor Stack Overflow Vulnerability (0–Day)

```
+ if (chunk.size != 36) // added in MS05–002
+ return 0;
```

```
// read chunk.size bytes of data into the header struct
ReadChunk(file, &chunk, &header);
```

For more information about the

<http://www.microsoft.com/technet/security/Bulletin/MS05–002.msp>

MS05–002 vulnerability, please refer to the

<http://research.eeye.com/html/advisories/published/AD20050111.html> eEye advisory that describes the issue in great detail.

If the animation header is valid, LoadCursorIconFromFileMap will call the LoadAniIcon function to process the rest of the chunks in the ANI file. LoadAniIcon uses the same ReadTag and ReadChunk functions as LoadCursorIconFromFileMap and contains the same vulnerability in the code that reads the anih header. This vulnerability was left unpatched in the MS05–002 security update.

```
int LoadAniIcon(struct MappedFile* file, ...)
{
    struct ANIChunk chunk;
    struct ANIHeader header; // 36 byte structure

    ...

    while (1) {
        // read the first 8 bytes of the chunk
        ReadTag(file, &chunk);

        switch (chunk.tag) {
            case 'seq':
                ...

            case 'LIST':
                ...

            case 'rate':
                ...

            case 'anih':
                // read chunk.size bytes of data into the header struct
                ReadChunk(file, &chunk, &header);
```

This code relies on the check in LoadCursorIconFromFileMap to catch the malformed anih chunk before it reaches the LoadAniIcon function. Unfortunately, LoadCursorIconFromFileMap validates only the first anih chunk, but LoadAniIcon processes all chunks in the file. By creating a file with two anih chunks, one valid and one malformed, it is possible to reach the vulnerable code in LoadAniIcon.

[NT] Windows Animated Cursor Stack Overflow Vulnerability (0-Day)

Reading the second anih chunk with the ReadChunk function will result in a classic buffer overflow, overwriting the return address of LoadAniChunk and allowing the attacker to take control of the code execution.

Exploitation:

The exploitation of this vulnerability is interesting in light of the protection features built in the latest versions of Windows XP, 2003 and Vista. It is a stack overflow and should be detected by the /GS security check. Unfortunately, the Visual Studio compiler adds the /GS check only to functions that contain certain types of arrays, assuming that most buffer overflows are a result of out-of-bounds array access. The LoadAniIcon function uses a structure as a destination buffer for the data it reads and as a result its return address is not protected by the /GS stack check. This allows an attacker to overwrite the return address and take control of the program execution on Windows XP SP2, 2003 and Vista in the same way as on Windows 2000.

In addition to the missing /GS check, the vulnerable code in USER32.DLL is wrapped in an exception handler that can recover from access violations. If the exploit is unsuccessful, for example due to the Vista ASLR, the process will not terminate and the attacker can simply try again. This gives the attacker an easy way to bypass the ASLR protection and increase the reliability of the exploit.

Solution:

The call to ReadChunk in the LoadAniIcon function should be preceded by a check similar to the one introduced in MS05-002:

```
if (chunk.size != 36)
return 0;
```

```
ReadChunk(file, &chunk, &header);
```

ADDITIONAL INFORMATION

The information has been provided by Determina Security Research.

The original article can be found at:

<http://www.determina.com/security.research/vulnerabilities/ani-header.html>
<http://www.determina.com/security.research/vulnerabilities/ani-header.html>

=====

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@xxxxxxxxxxxxxxxx

In order to subscribe to the mailing list, simply forward this email to: list-subscribe@xxxxxxxxxxxxxxxx

[NT] Windows Animated Cursor Stack Overflow Vulnerability (0-Day)

=====
=====

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.