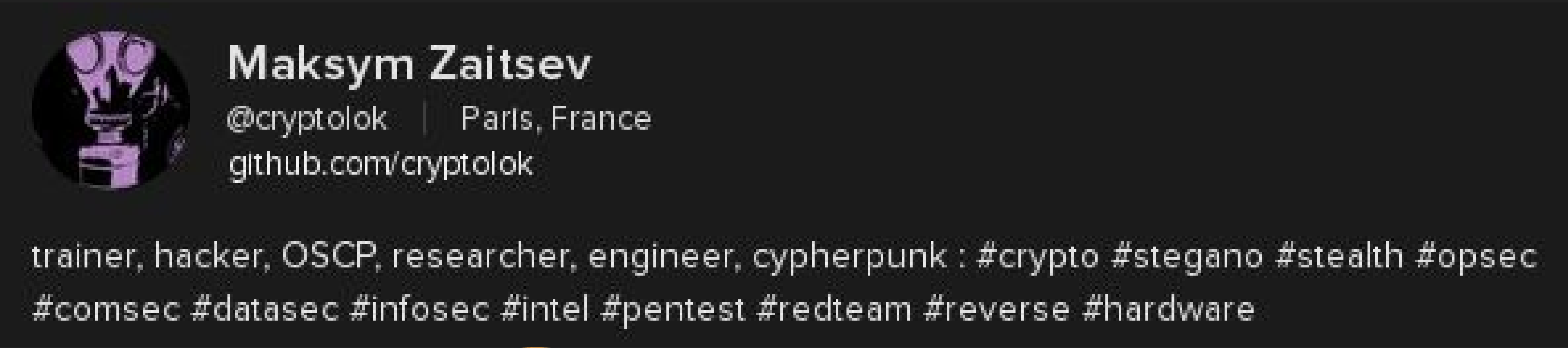
Meta-Polymorphic Evasive ShellCodes

MorphAES

id $USER



Maksym Zaitsev

@cryptolok | Paris, France

github.com/cryptolok

trainer, hacker, OSCP, researcher, engineer, cypherpunk : #crypto #stegano #stealth #opsec #comsec #datasec #infosec #intel #pentest #redteam #reverse #hardware

ENKI

ÆNEAS

ShellCodes

Historically:

Code that returns a Shell

Currently:

Any malicious payload

Not just any code!

```
<?php system($_GET['cmd']); ?>
```

An OPCODE!

```
\x31\xc0\x48…\x3b\x0f\x05
```

Opcodes

Opcode ⇔ OS-specific AND CPU-specific assembly instruction in hexa

- **Android system call ARM**
  - svc ⇔ \xef
- **Windows system call x86**
  - int ⇔ \xcd
- **Linux System call x64**
  - syscall ⇔ \x0f\x05
- **Linux syscall number =/= Windows =/= BSD !**

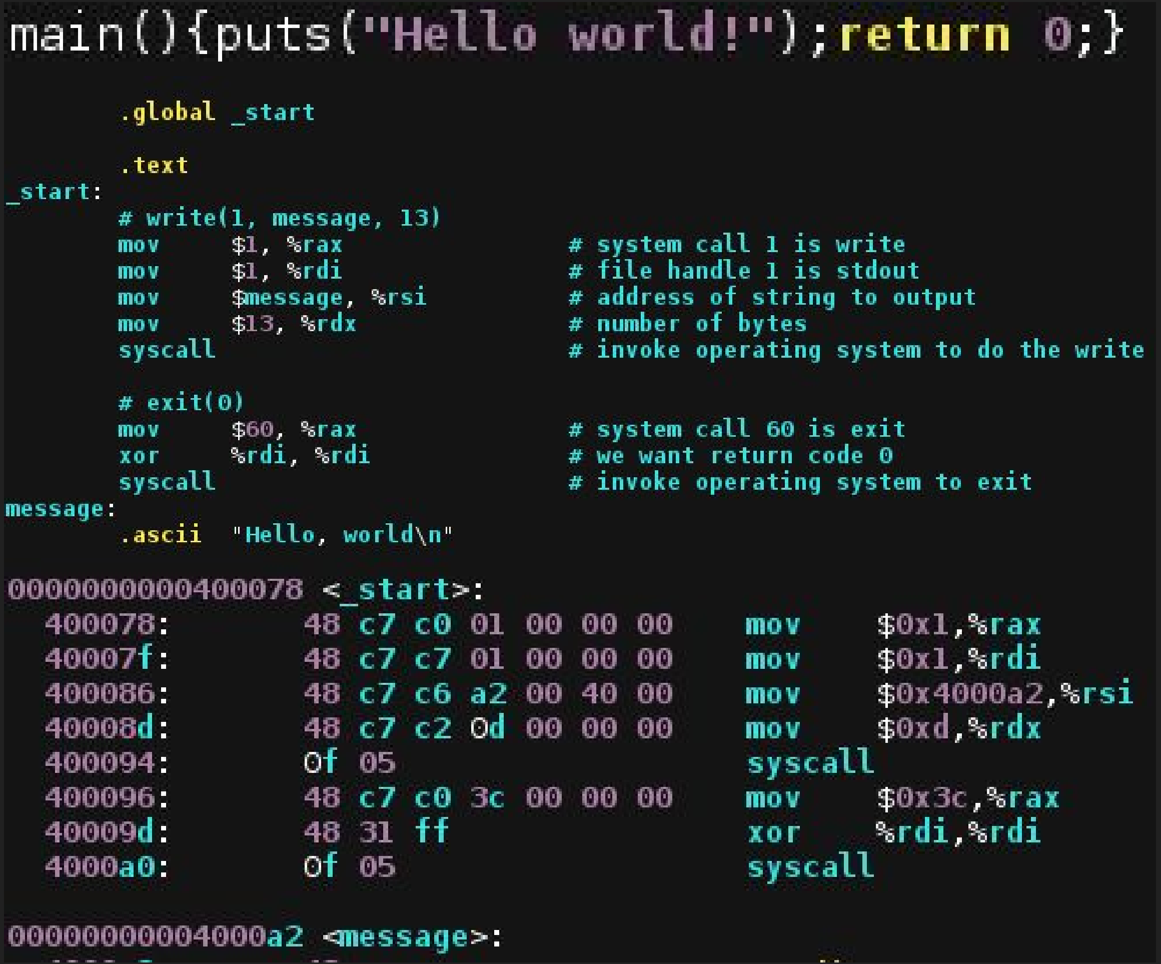
Assembly

Low-level

CPU instructions

Direct commands

Hardcore!



```
main(){puts("Hello world!");return 0;}
```

```
        .global _start

        .text
_start:
        # write(1, message, 13)
        mov     $1, %rax                # system call 1 is write
        mov     $1, %rdi                # file handle 1 is stdout
        mov     $message, %rsi          # address of string to output
        mov     $13, %rdx               # number of bytes
        syscall                         # invoke operating system to do the write

        # exit(0)
        mov     $60, %rax               # system call 60 is exit
        xor     %rdi, %rdi              # we want return code 0
        syscall                         # invoke operating system to exit
message:
        .ascii  "Hello, world\n"
```

```
0000000000400078 <_start>:
  400078:       48 c7 c0 01 00 00 00    mov    $0x1,%rax
  40007f:       48 c7 c7 01 00 00 00    mov    $0x1,%rdi
  400086:       48 c7 c6 a2 00 40 00    mov    $0x4000a2,%rsi
  40008d:       48 c7 c2 0d 00 00 00    mov    $0xd,%rdx
  400094:       0f 05                   syscall
  400096:       48 c7 c0 3c 00 00 00    mov    $0x3c,%rax
  40009d:       48 31 ff                xor    %rdi,%rdi
  4000a0:       0f 05                   syscall

00000000004000a2 <message>:
```

Stack Buffer Overflow Demo

Was too lazy to make ppt, duh...

Mitigations

Decrease buffer length?

Put shellcode after stack IP :)

Force size

0 day :)

WAF/IDPS/NGFW/UTM/AV

Morphism

DEP/NX/CANARY/ASLR

ROP/libc/leak/NOPs

SandBox/Emulation???

Stealthing

NOP (\x90, inc/dec, mov) obfuscation - IDPS/AV

XOR polymorphism (shikataganai) - IDPS/AV

AES polymorphism - heuristic/dependable/huge

Mutation metamorphism - sandbox

Splinter Cell: Chaos Theory developer Ubisoft Montreal publisher Ubisoft
© GameWallpapers.com hosted by JTLnet.com

IDEA

ShellCode morpher

Independent AES ASM polymorphism

Metamorphism

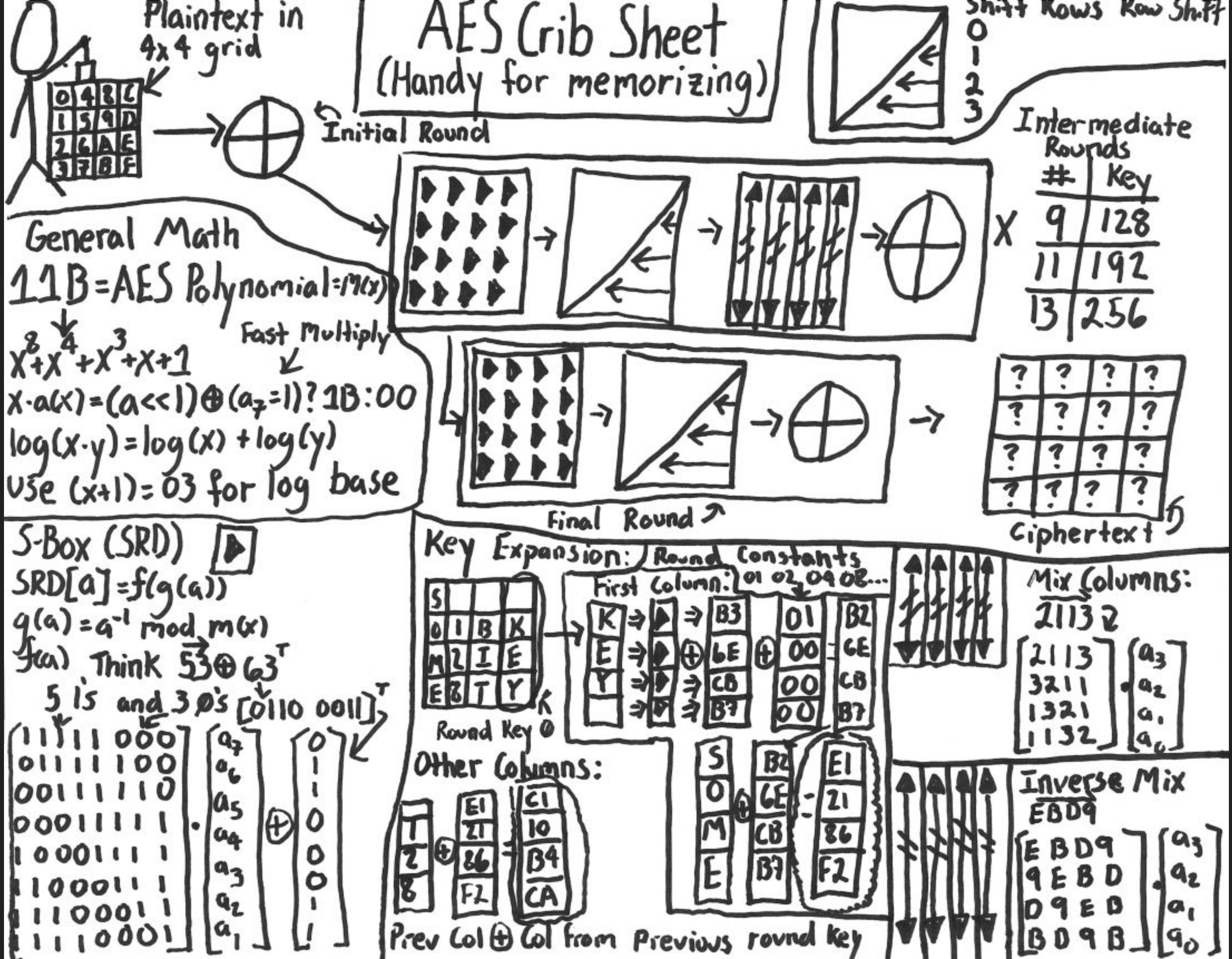
Anti-SandBox (evasion)

No bad characters

Cross-platform

Unique NOP sled

AES



AES Crib Sheet
(Handy for memorizing)

Plaintext in 4x4 grid

0	4	8	C
1	5	9	D
2	6	A	E
3	7	B	F

Intermediate Rounds

#	Key
9	128
11	192
13	256

?	?	?	?
?	?	?	?
?	?	?	?
?	?	?	?

General Math

11B = AES Polynomial = m(x)

$x^8+x^4+x^3+x+1$

Fast Multiply

$x \cdot a(x) = (a \ll 1) \oplus (a_7 = 1)\,?\,1B : 00$

$\log(x \cdot y) = \log(x) + \log(y)$

Use $(x+1) = 03$ for log base

S-Box (SRD)

$SRD[a] = f(g(a))$

$g(a) = a^{-1} \bmod m(x)$

$f(a)$ Think $\vec{53} \oplus 63^T$

5 1's and 3 0's $[0110\ 0011]^T$

$$\begin{bmatrix} 1&1&1&1&1&0&0&0\\ 0&1&1&1&1&1&0&0\\ 0&0&1&1&1&1&1&0\\ 0&0&0&1&1&1&1&1\\ 1&0&0&0&1&1&1&1\\ 1&1&0&0&0&1&1&1\\ 1&1&1&0&0&0&1&1\\ 1&1&1&1&0&0&0&1 \end{bmatrix} \cdot \begin{bmatrix} a_7\\ a_6\\ a_5\\ a_4\\ a_3\\ a_2\\ a_1 \end{bmatrix} \oplus \begin{bmatrix} 0\\ 1\\ 1\\ 0\\ 0\\ 0\\ 1\\ 1 \end{bmatrix}$$

Key Expansion:

Round Constants 01 02 04 08...

S			
O	1	B	K
M	2	I	E
E	8	T	Y

Round Key 0

First Column:

$$\begin{bmatrix} K\\ E\\ Y\\ \ \end{bmatrix} \Rightarrow \Rightarrow \begin{bmatrix} B3\\ 6E\\ CB\\ B7 \end{bmatrix} \oplus \begin{bmatrix} 01\\ 00\\ 00\\ 00 \end{bmatrix} = \begin{bmatrix} B2\\ 6E\\ CB\\ B7 \end{bmatrix}$$

$$\begin{bmatrix} S\\ O\\ M\\ E \end{bmatrix} \oplus \begin{bmatrix} B2\\ 6E\\ CB\\ B7 \end{bmatrix} = \begin{bmatrix} E1\\ 21\\ 86\\ F2 \end{bmatrix}$$

Other Columns:

$$\begin{bmatrix} \ \\ 1\\ 2\\ 8 \end{bmatrix} \oplus \begin{bmatrix} E1\\ 21\\ 86\\ F2 \end{bmatrix} = \begin{bmatrix} C1\\ 10\\ B4\\ CA \end{bmatrix}$$

Prev Col ⊕ Col from Previous round key

Mix Columns:

2113

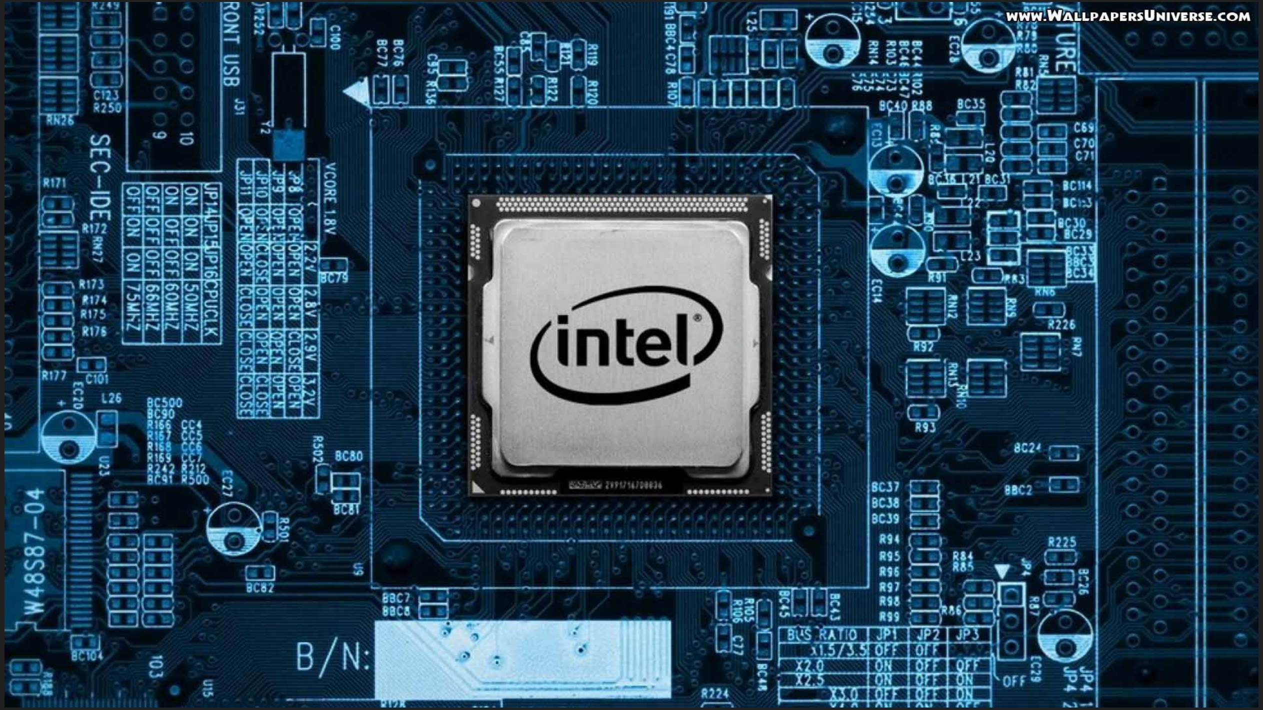
$$\begin{bmatrix} 2&1&1&3\\ 3&2&1&1\\ 1&3&2&1\\ 1&1&3&2 \end{bmatrix} \cdot \begin{bmatrix} a_3\\ a_2\\ a_1\\ a_0 \end{bmatrix}$$

Inverse Mix

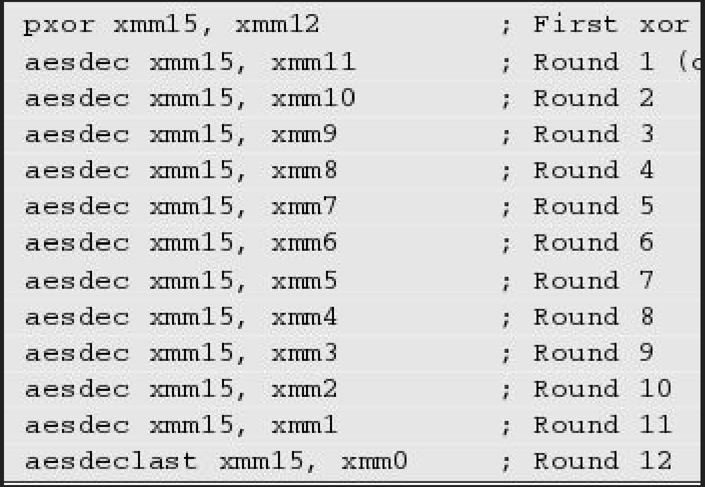
EBD9

$$\begin{bmatrix} E&B&D&9\\ 9&E&B&D\\ D&9&E&B\\ B&D&9&B \end{bmatrix} \cdot \begin{bmatrix} a_3\\ a_2\\ a_1\\ a_0 \end{bmatrix}$$

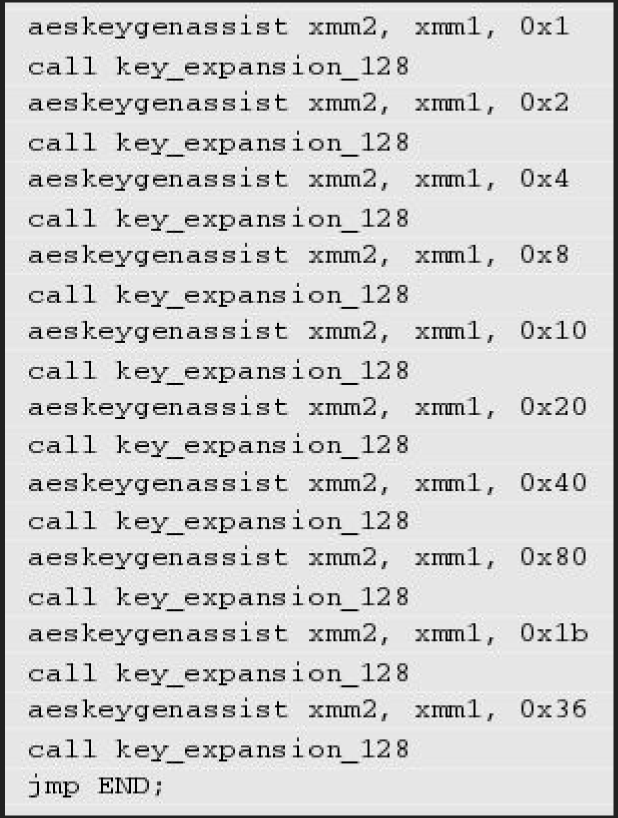
Intel AES-NI & XMM



AES-128-ECB



```
pxor xmm15, xmm12              ; First xor
aesdec xmm15, xmm11            ; Round 1 (c
aesdec xmm15, xmm10            ; Round 2
aesdec xmm15, xmm9             ; Round 3
aesdec xmm15, xmm8             ; Round 4
aesdec xmm15, xmm7             ; Round 5
aesdec xmm15, xmm6             ; Round 6
aesdec xmm15, xmm5             ; Round 7
aesdec xmm15, xmm4             ; Round 8
aesdec xmm15, xmm3             ; Round 9
aesdec xmm15, xmm2             ; Round 10
aesdec xmm15, xmm1             ; Round 11
aesdeclast xmm15, xmm0         ; Round 12
```



```
aeskeygenassist xmm2, xmm1, 0x1
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x2
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x4
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x8
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x10
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x20
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x40
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x80
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x1b
call key_expansion_128
aeskeygenassist xmm2, xmm1, 0x36
call key_expansion_128
jmp END;
```

Challenges

Key storage in 128 from 64

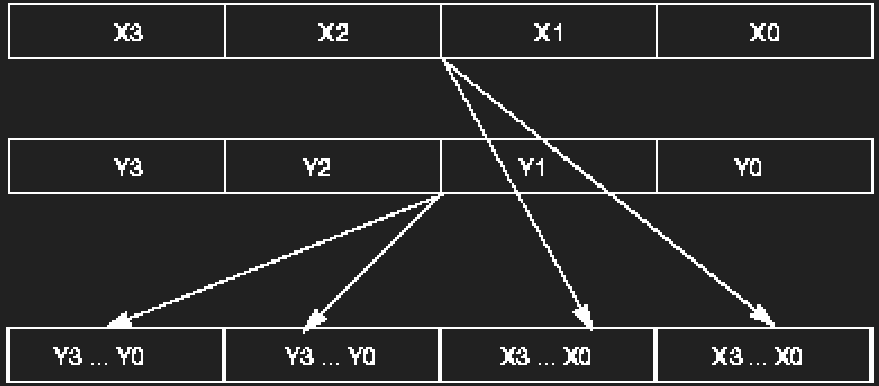
Instructions morphism

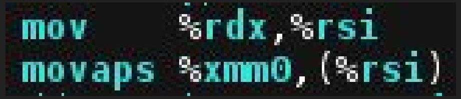
Bad characters

Arbitrary length storage

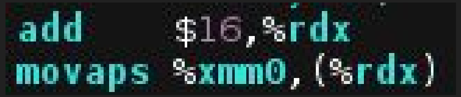
NOP sled

No execution in Qemu => Cuckoo

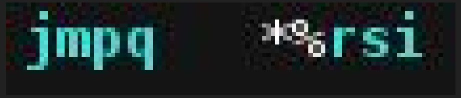




```
mov     %rdx,%rsi
movaps %xmm0,(%rsi)
```



```
add     $16,%rdx
movaps %xmm0,(%rdx)
```



```
jmpq    *%rsi
```

DEMO

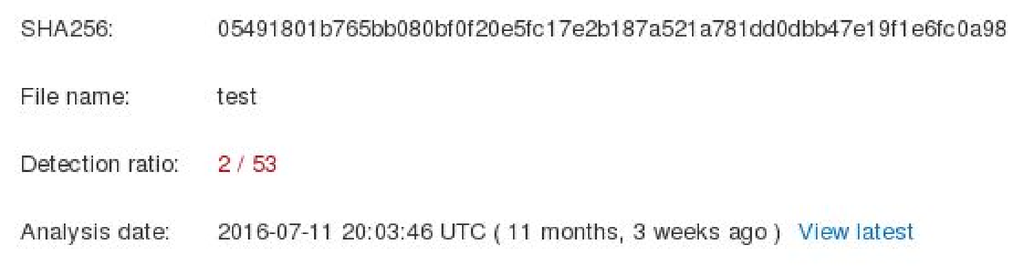
Analysis

SSDEEP

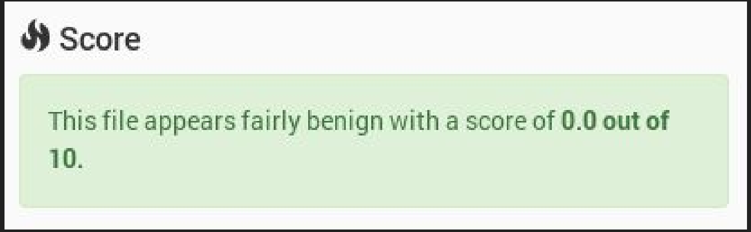
Hard Reverse

6:Cq8bnJYn4Xkm3qECaADATyEnT8snTiETiTCfhUaAP6mYGexCKdKZzX+rqVCKdKTc:xuJ0Zp2xRZof79G/KVyk/KTbA,

6:vrg+T1RfLEQD/zD1DZzDJ3zDBfjDcDRJDULUwzWq0Cgk3g4zE/Yq0Cgk3gy12Ots:vLjjEszWCp3w/YCp3Nts,



SHA256:	05491801b765bb080bf0f20e5fc17e2b187a521a781dd0dbb47e19f1e6fc0a98
File name:	test
Detection ratio:	2 / 53
Analysis date:	2016-07-11 20:03:46 UTC (11 months, 3 weeks ago) View latest



Score

This file appears fairly benign with a score of **0.0 out of 10.**

Limitations

Pseudo-metamorphism

Unicode

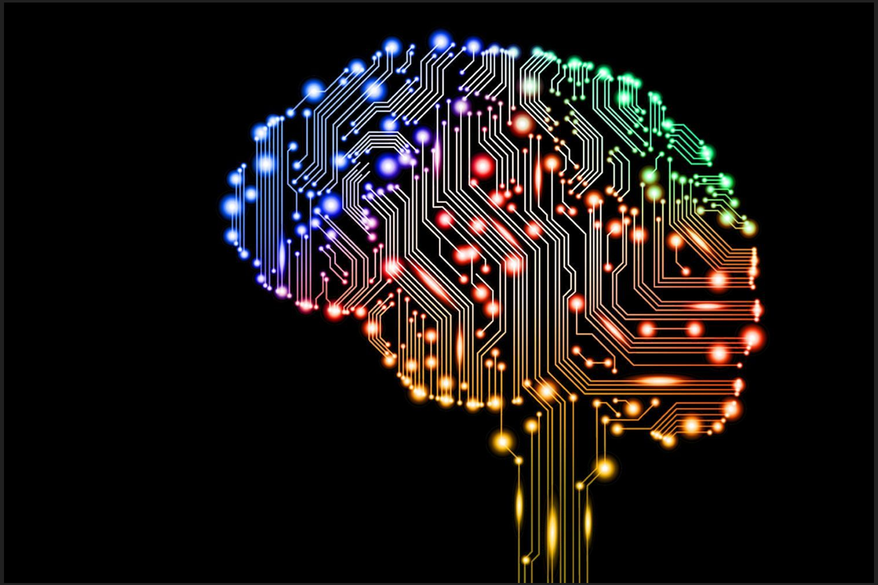
Intel x64 AES-NI

Malware-abuse prevention

Linux tool only (for now)

Mitigations

AI



Sandboxing

QUESTIONS/NOTES?

Maksym Zaitsev
@cryptolok