The Remote Metamorphic Engine

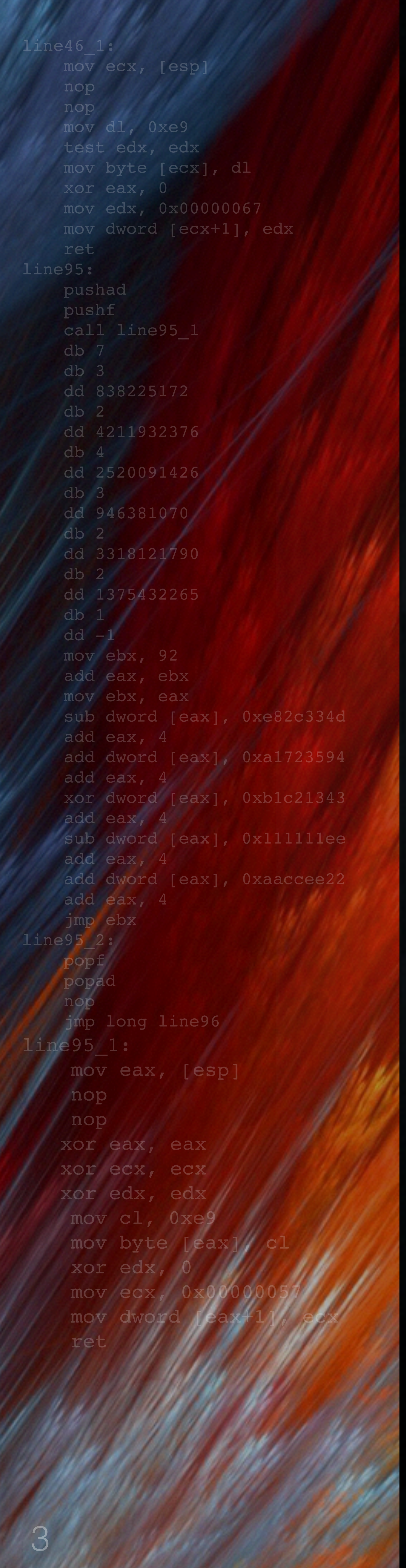
Detecting, Evading, Attacking the AI and Reverse Engineering

Amro Abdelgawad / DEFCON 24

The Remote Metamorphic Engine

- Security as undefined expression
- Flux binary mutation
- Resisting Reverse Engineering
- Evading AI machine learning
- Artificial Immunity

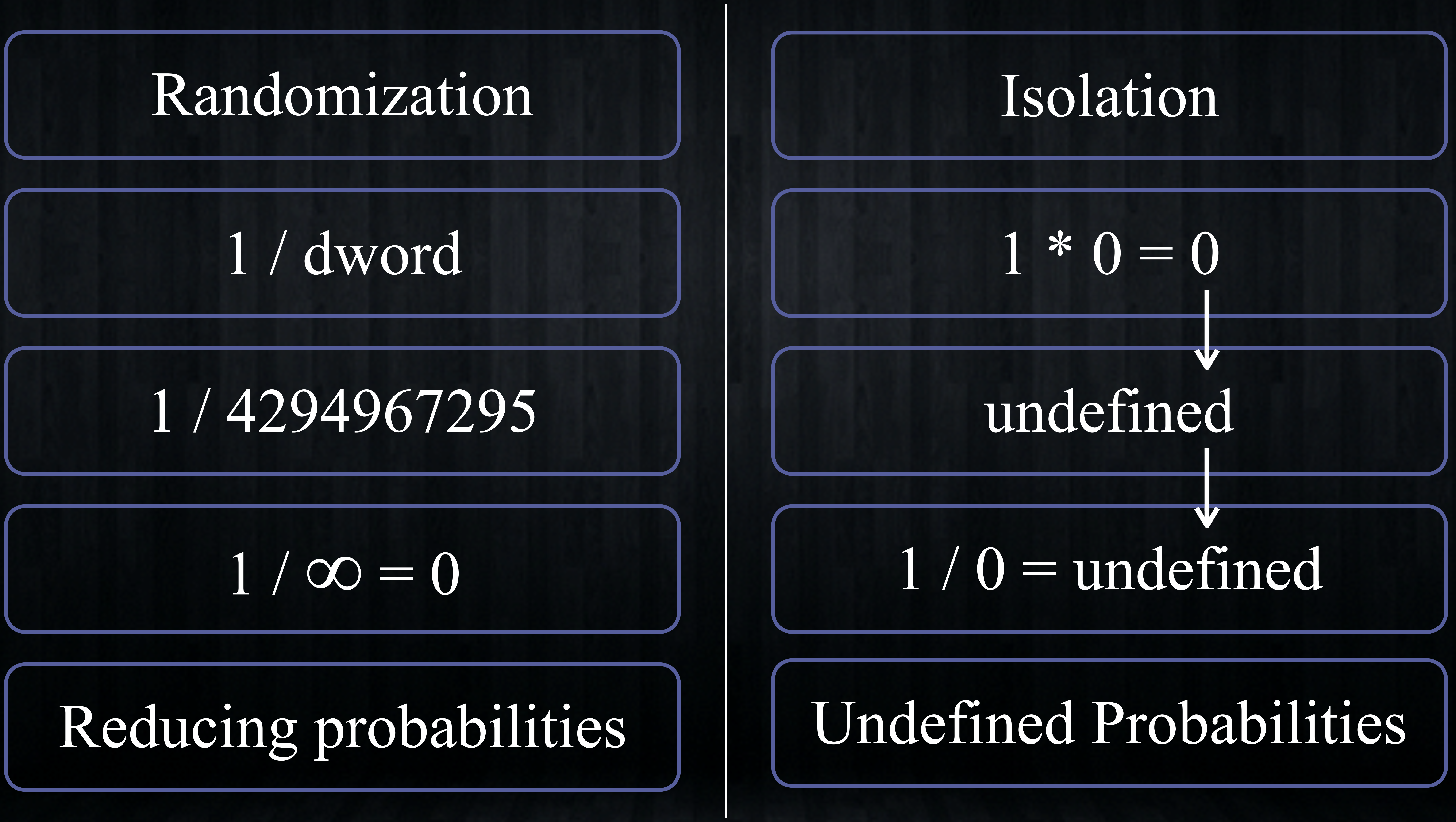
```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

{ Security Patterns

Division by Infinity | Division by Zero }



Randomization

1 / dword

1 / 4294967295

1 / ∞ = 0

Reducing probabilities

Isolation

1 * 0 = 0

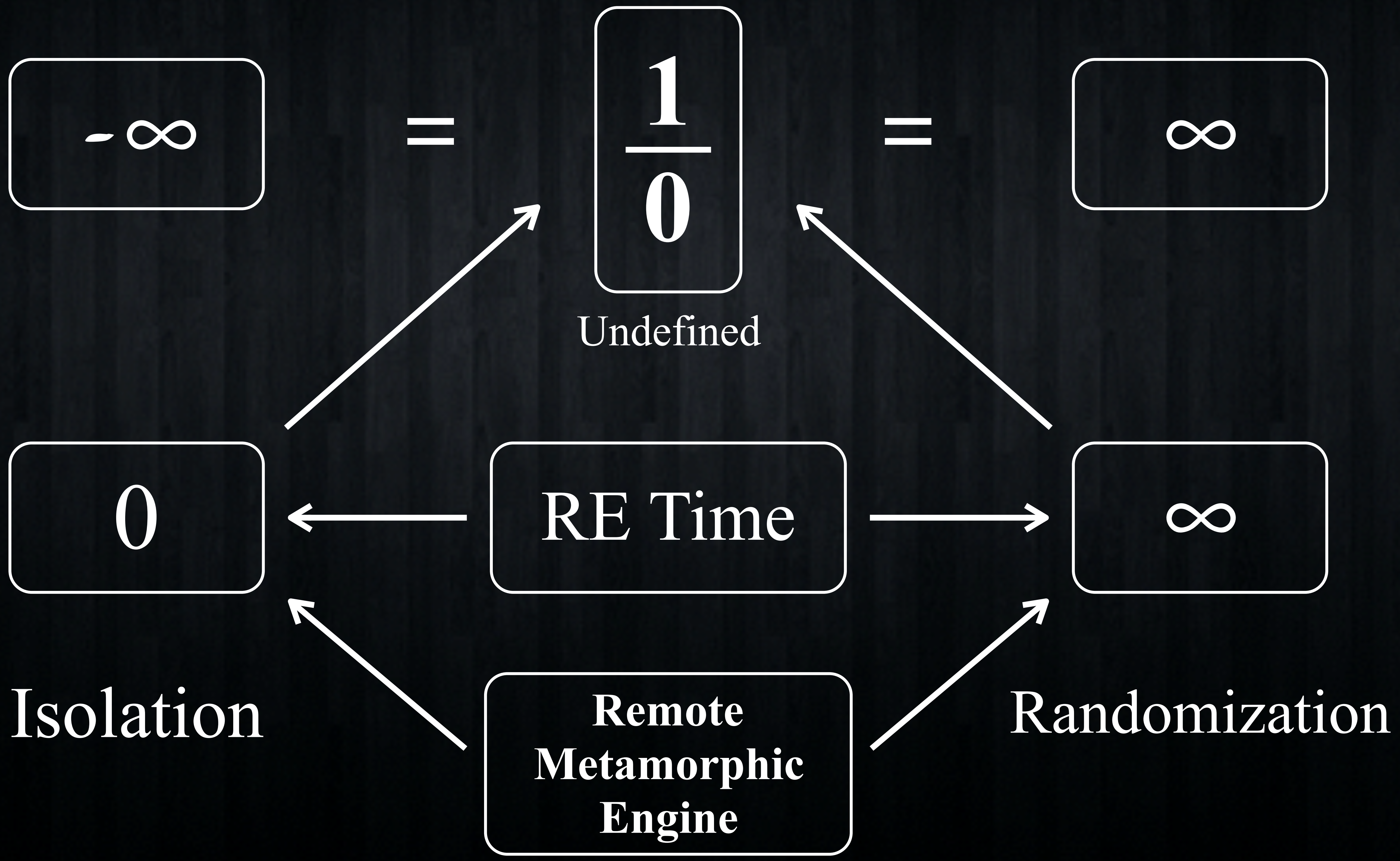
undefined

1 / 0 = undefined

Undefined Probabilities

The Undefined Expression

Security as Undefined & indeterminate expression



$-\infty = \frac{1}{0} = \infty$

Undefined

0

RE Time

∞

Isolation

Remote Metamorphic Engine

Randomization

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

The Unbreakable Code

Unpredictable
un·pre·dict·a·ble
adjective:/ˌənprəˈdiktəb(ə)l/

Likely to change suddenly and without reason
and therefore not able to be predicted
(= expected before it happens)

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

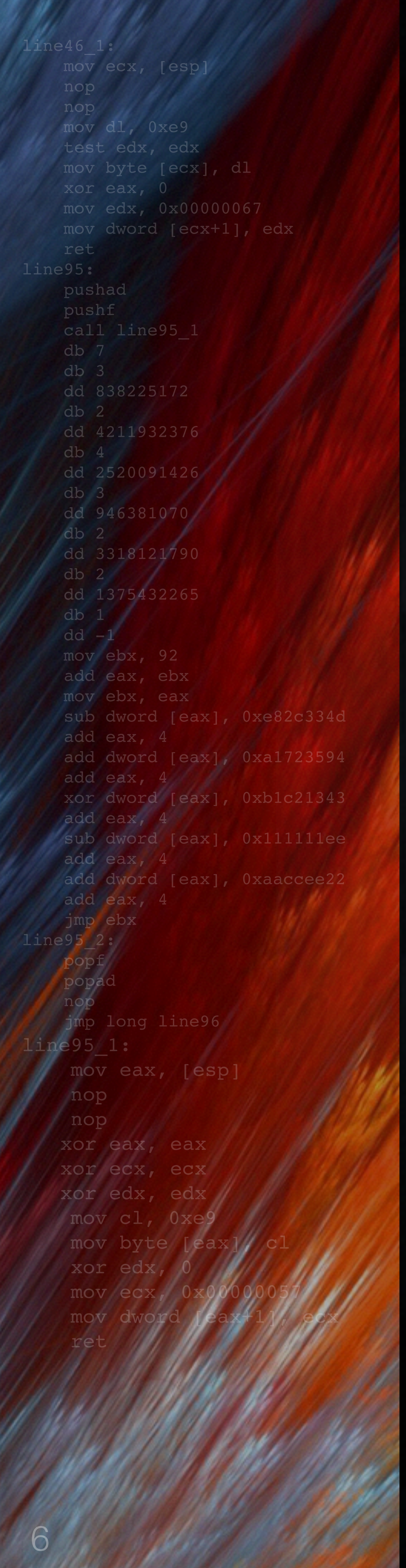
The Breakable Code

The Fixed Static Code Problem

Static Code Dynamic Data

Core security weakness in all today's software

Enables all sorts of replicable software security exploits



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Unpredictable Code Evolution

Dynamic Code Dynamic Data

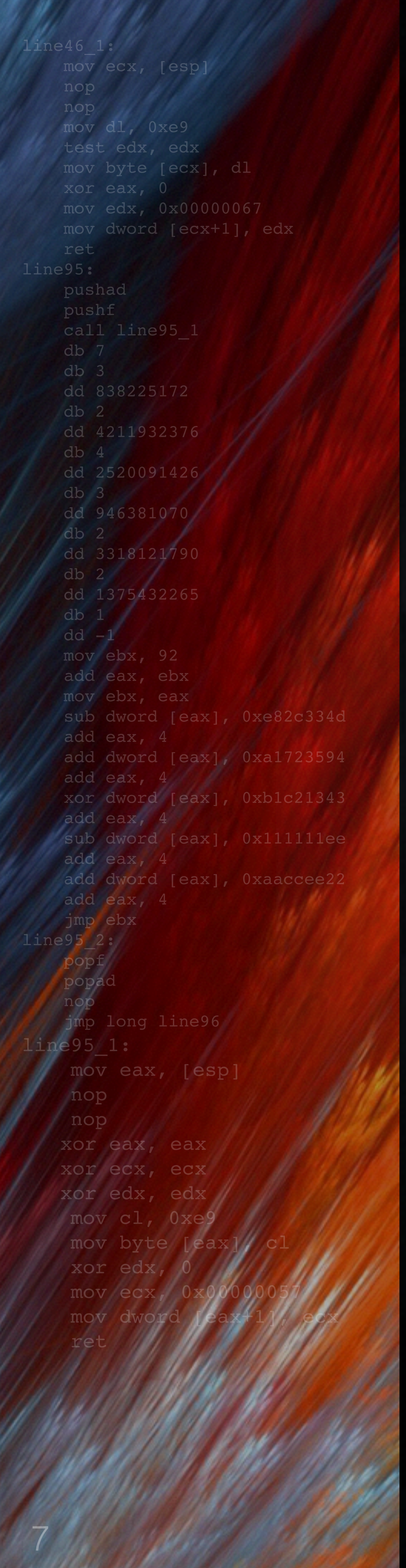
Code evolution across time

Functionality evolution across location

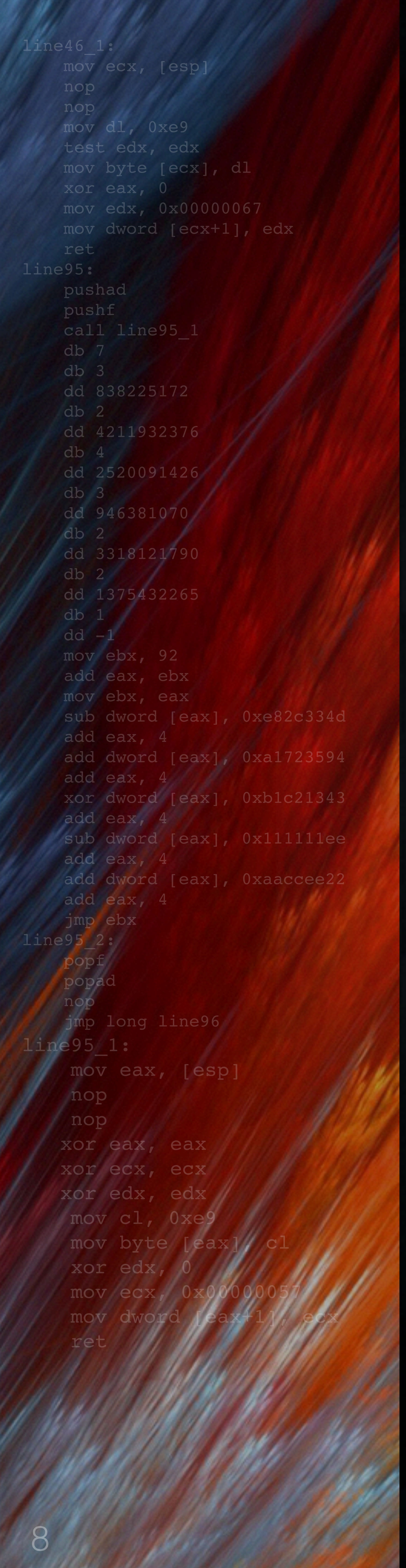
Self contained autonomous code

Unpredictable

Self aware



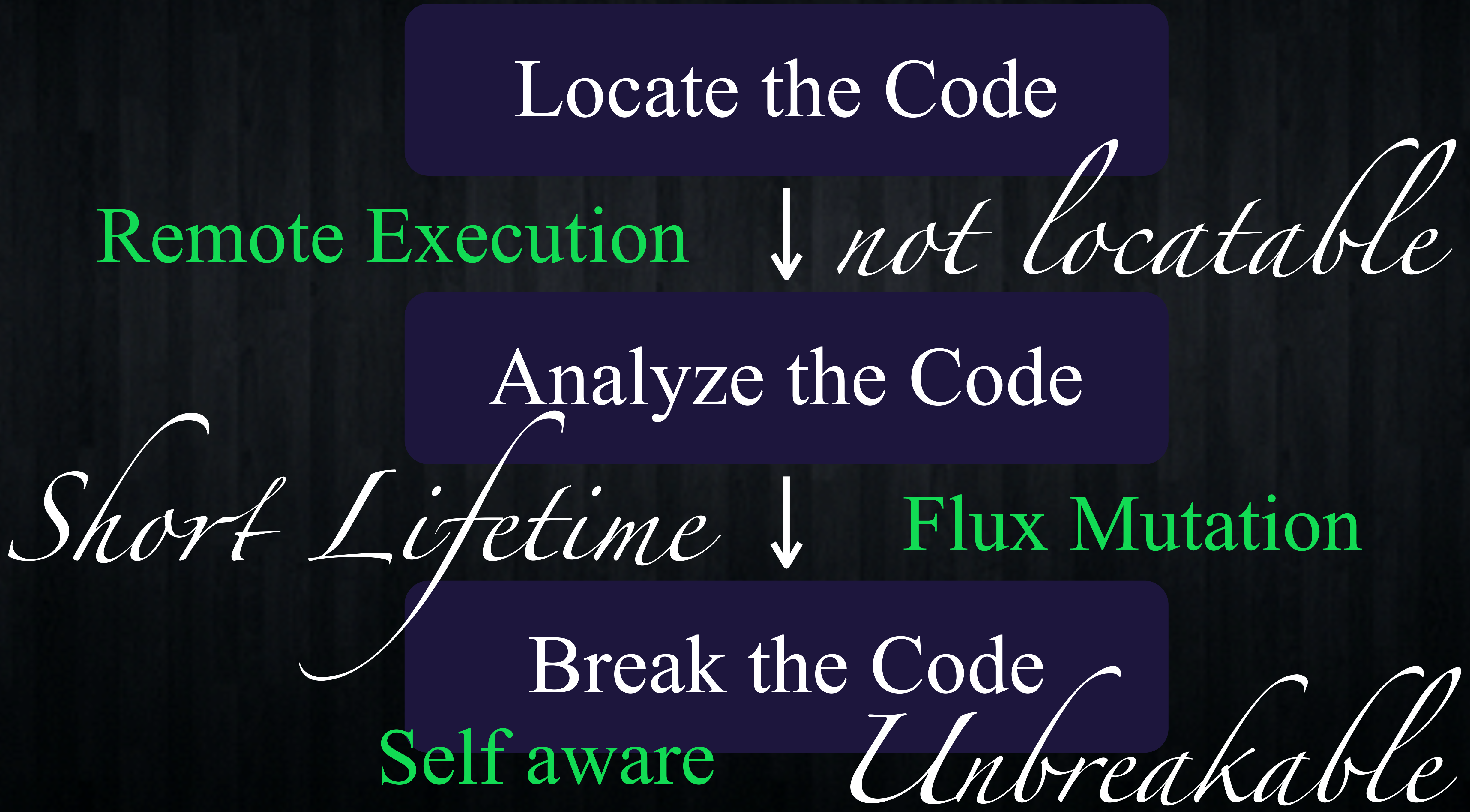
```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

Code Evolution

Resisting Reverse Engineering



Locate the Code

Remote Execution ↓ *not locatable*

Analyze the Code

Short Lifetime ↓ Flux Mutation

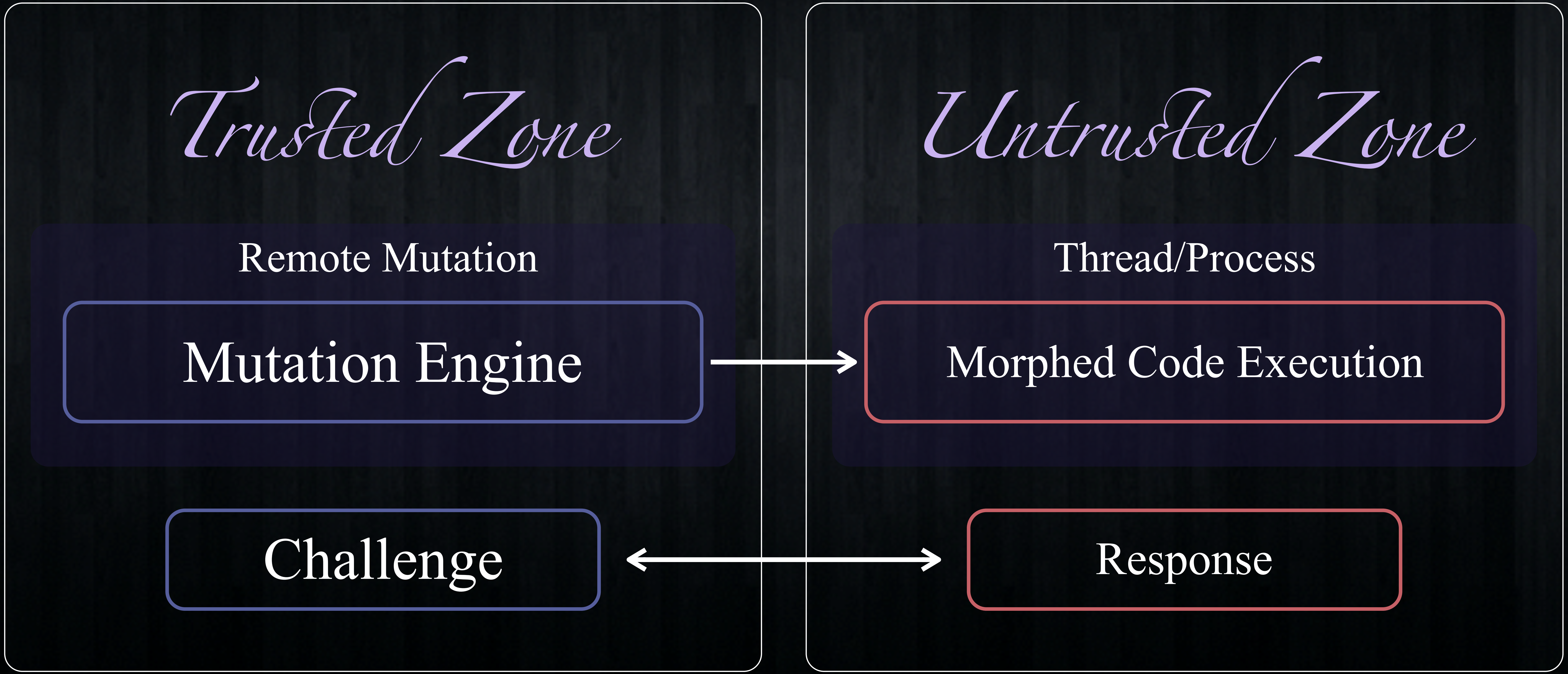
Break the Code

Self aware *Unbreakable*

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

The Remote Metamorphic Engine

Remote Flux Mutation

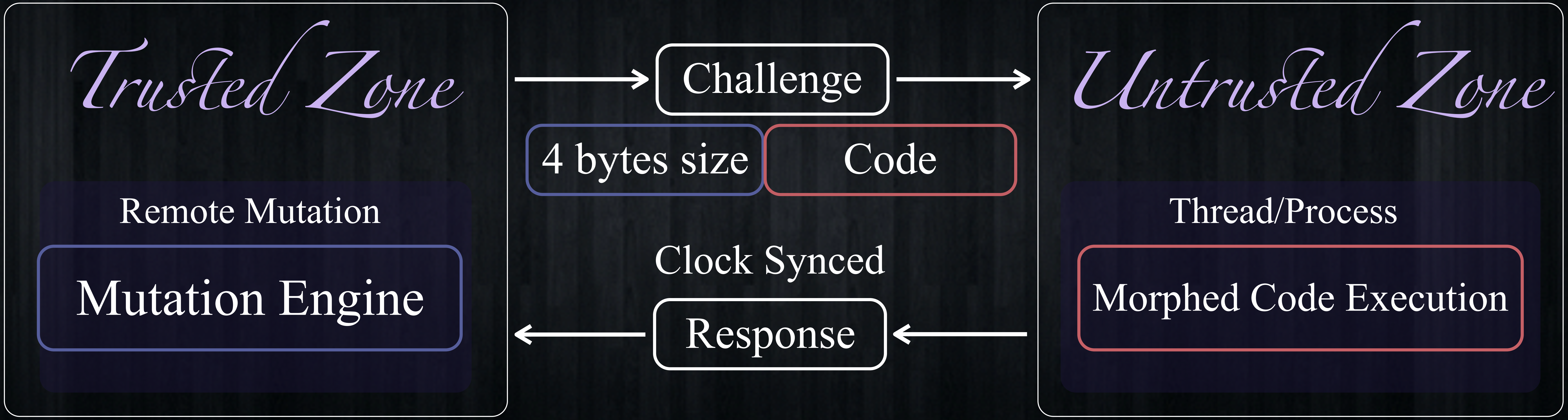


```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Why Remote?

The Remote Metamorphic Engine

Challenge Response Metamorphic Protocol



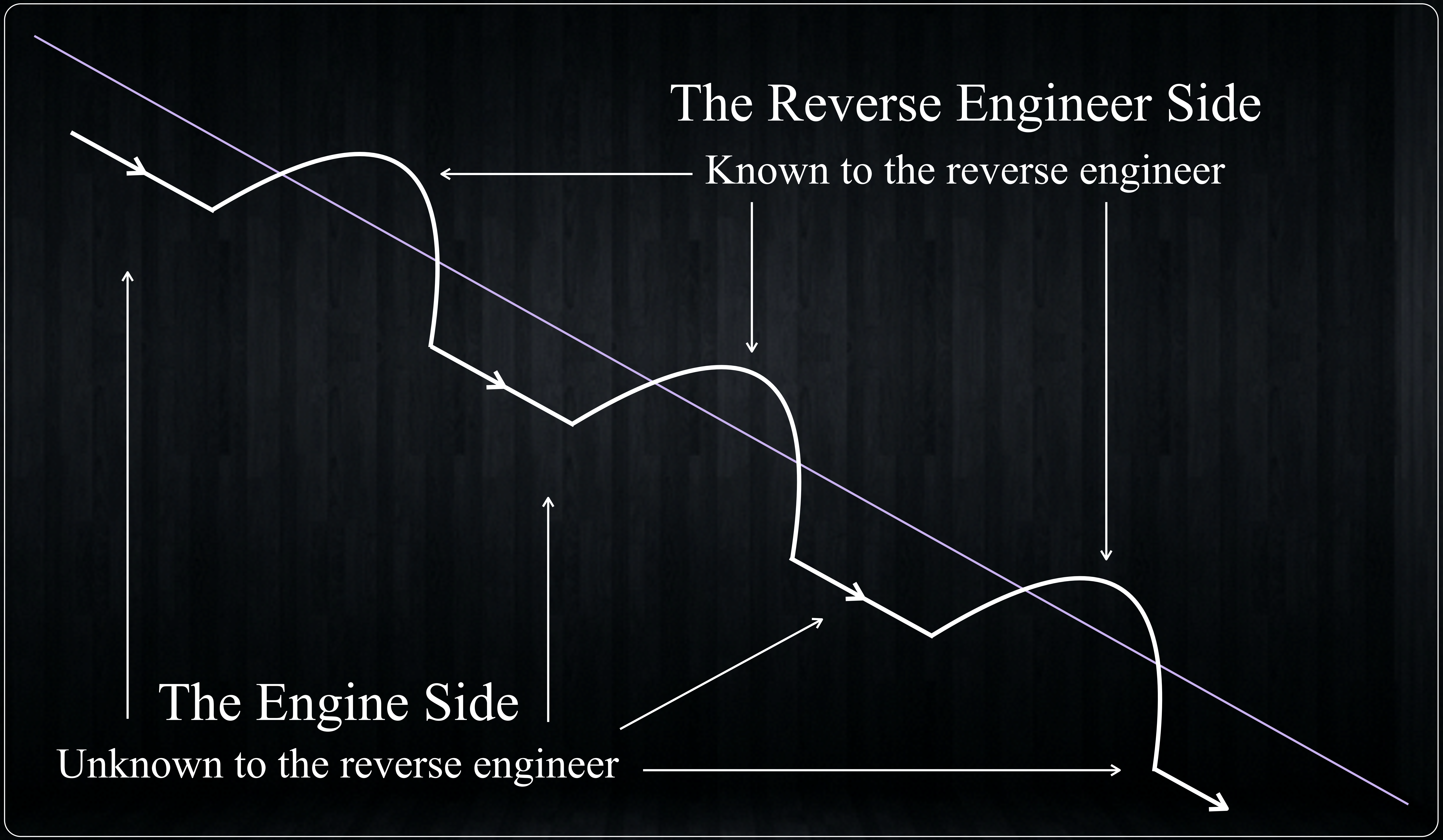
Communication protocol made of morphed clock synchronized machine code rather than data

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

The Remote Metamorphic Engine

Remote Code Slicing



The Reverse Engineer Side

Known to the reverse engineer

The Engine Side

Unknown to the reverse engineer

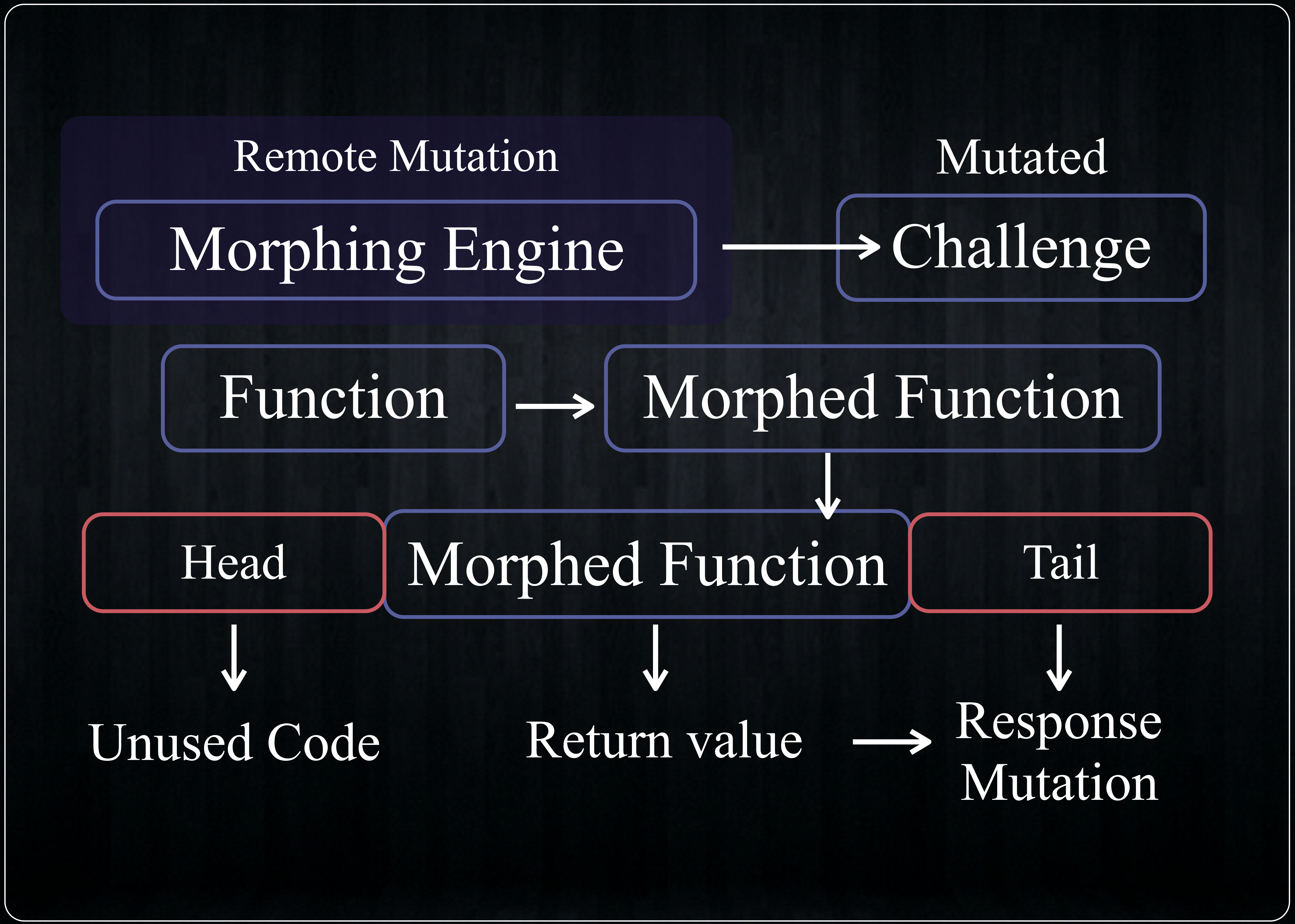
RME Challenges

- In memory code integrity check
- Execution environment integrity check
- In memory APIs code integrity check
- Detect hooks
- Clock synchronization
- Detect debuggers
- Detect Virtual Machines
- Collect & Validate Machine IDs
- Validate Hardware

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Trusted Mutation

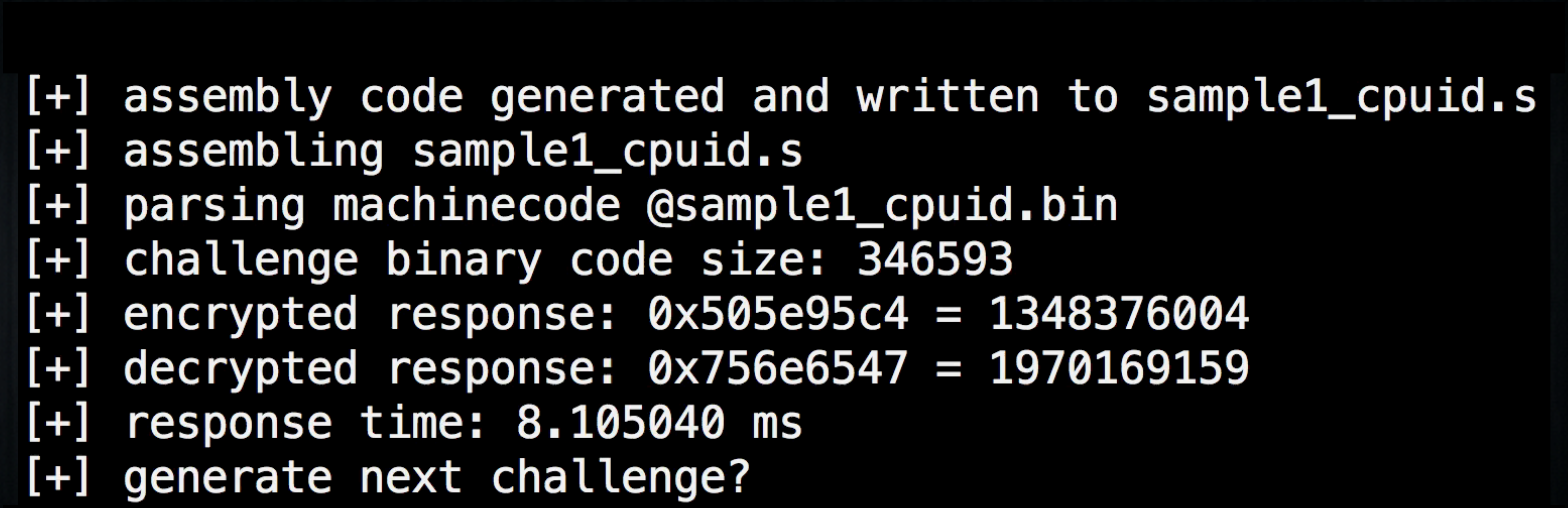
Trusted Challenge Response Mutation



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

Response Mutation

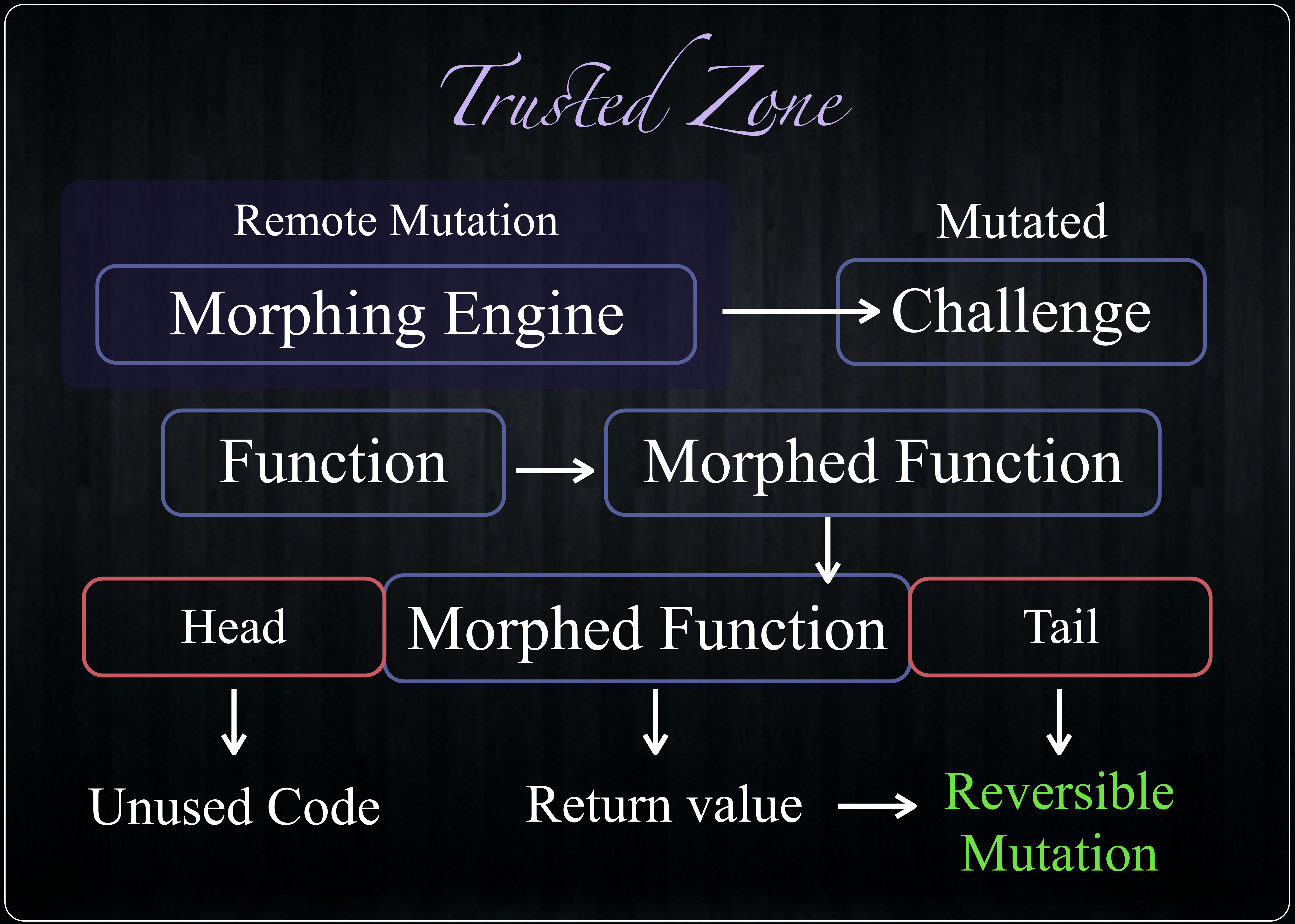
Mutated Encryption/Decryption



```
[+] assembly code generated and written to sample1_cpuid.s
[+] assembling sample1_cpuid.s
[+] parsing machinecode @sample1_cpuid.bin
[+] challenge binary code size: 346593
[+] encrypted response: 0x505e95c4 = 1348376004
[+] decrypted response: 0x756e6547 = 1970169159
[+] response time: 8.105040 ms
[+] generate next challenge?
```

Decoupled Reversible Mutation

Response Mutation



Trusted Zone

Remote Mutation

Morphing Engine

Mutated

Challenge

Function

Morphed Function

Head

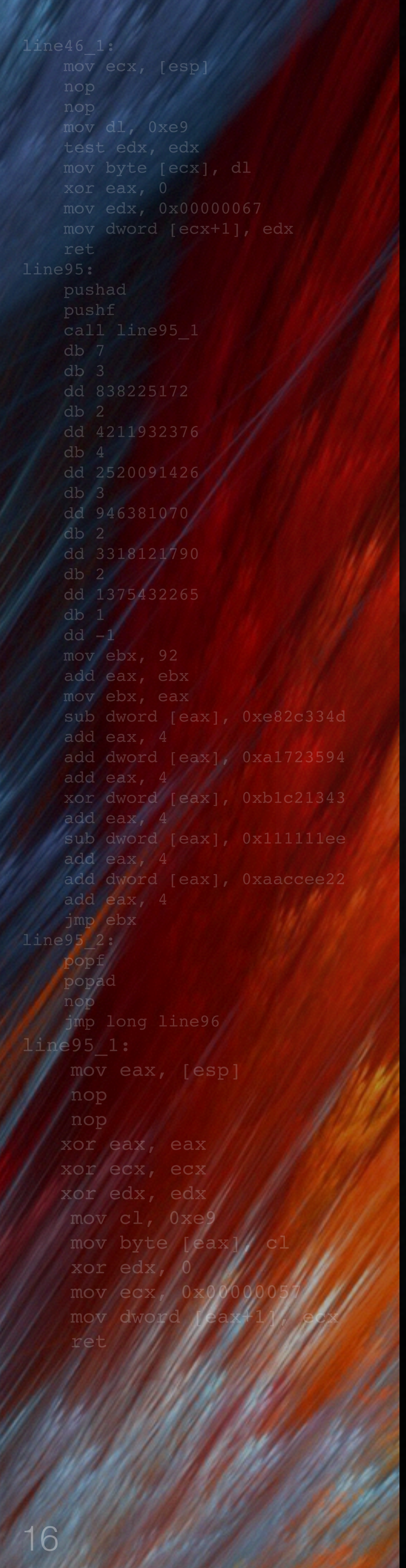
Morphed Function

Tail

Unused Code

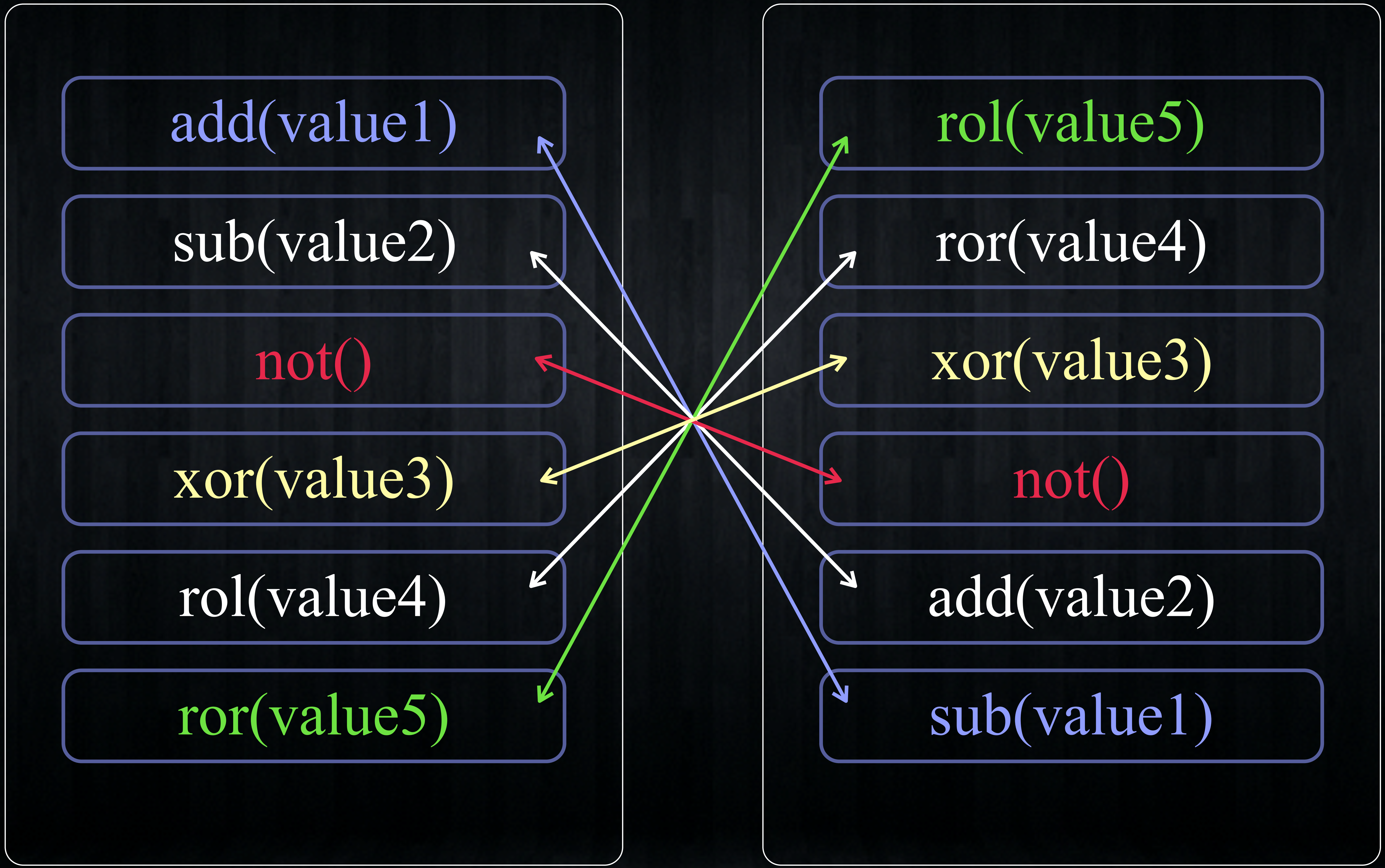
Return value

Reversible Mutation



Decoupled Reversible Mutation

Reversible Instructions



add(value1)

sub(value2)

not()

xor(value3)

rol(value4)

ror(value5)

rol(value5)

ror(value4)

xor(value3)

not()

add(value2)

sub(value1)

Reversible Instructions | Response Mutation

```
add eax, 0xe0d9780c
not eax
sub eax, 0xbcf3e676
not eax
xor eax, 0xfb7e9fdd
sub eax, 0x695e3adf
add eax, 0x3e731a34
xor eax, 0xa0b50d13
xor eax, 0x39034b8d
ror eax, 0xf
sub eax, 0xfb824ebb
xor eax, 0xd1e6a7ec
xor eax, 0xbb5202f7
ror eax, 4
xor eax, 0x9ce66186
sub eax, 0x4ec067b8
not eax
sub eax, 0xc98775b4
xor eax, 0xbdc52b4f
ror eax, 2
sub eax, 0xd925192c
ror eax, 3
```

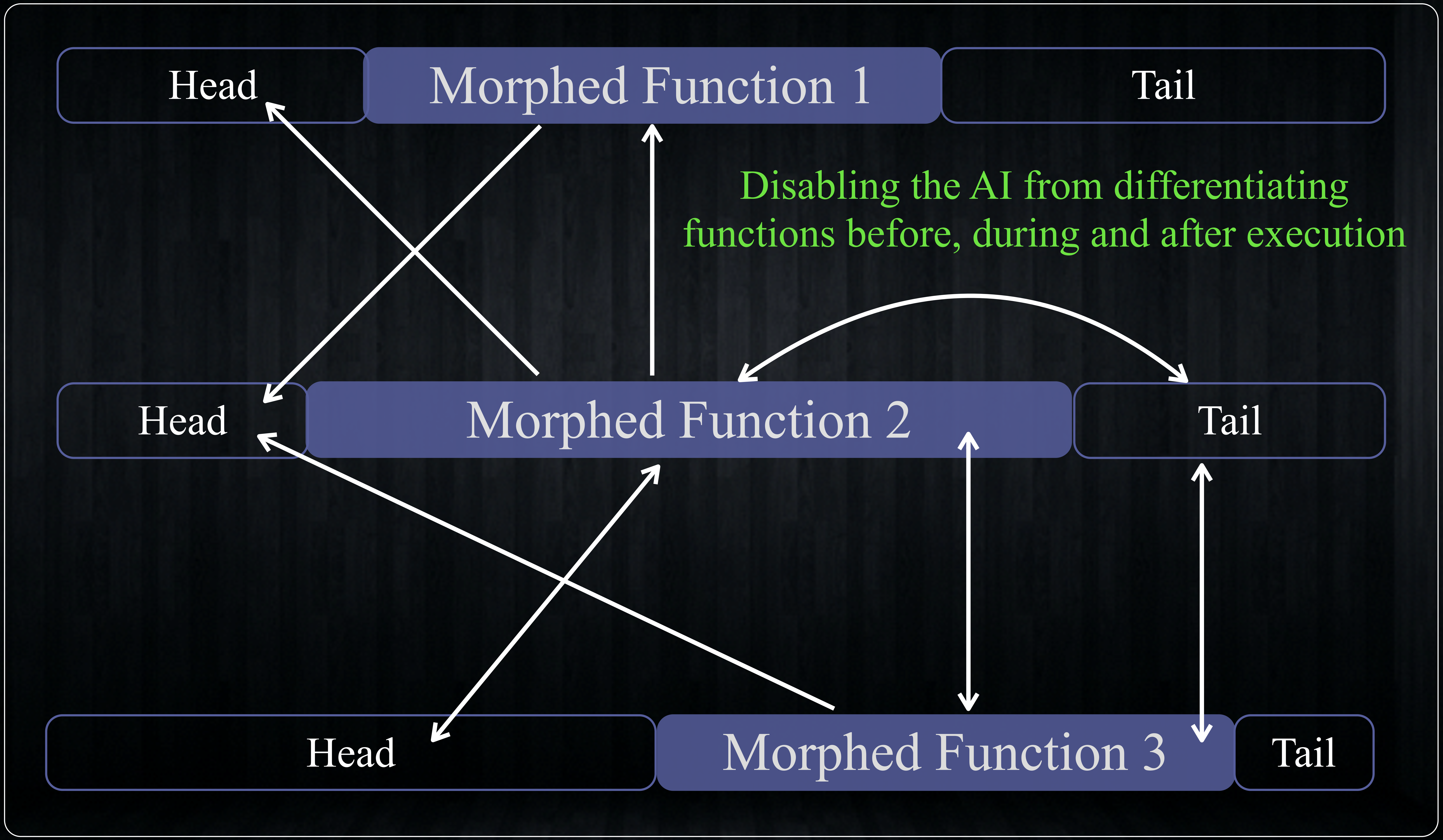
```
add eax, 0x48fa27f1
sub eax, 0xd353c205
sub eax, 0xa888b8b2
xor eax, 0xe017f6fa
ror eax, 0xd
sub eax, 0x247dab96
add eax, 0xf6696155
sub eax, 0xbeaeaad5
add eax, 0xd6c7b4ee
add eax, 0x120d5924
add eax, 0x9a0be9b9
sub eax, 0xbfe386c3
ror eax, 0x17
add eax, 0x14c58836
ror eax, 5
xor eax, 0x1984a5de
not eax
sub eax, 0x4d956430
sub eax, 0x9c9df86
add eax, 0xd88904bc
xor eax, 0xf5bcc022
xor eax, 0x205c4a75
add eax, 0xbcbb2b45
sub eax, 0xdb0a2bc0
ror eax, 0xd
add eax, 0x529eba0f
ror eax, 0x1c
add eax, 0x8150605
sub eax, 0xd8fe0628
add eax, 0xad81052c
ror eax, 5
```

```
add eax, 0x762e0f15
not eax
sub eax, 0x75707780
add eax, 0xe3265fc4
xor eax, 0x22952628
add eax, 0x231a8655
ror eax, 2
not eax
sub eax, 0x2c75569a
sub eax, 0x88ad3417
not eax
ror eax, 0x19
add eax, 0xe7634a71
not eax
xor eax, 0x500026f6
add eax, 0xad1a2fd2
sub eax, 0x937ead1b
not eax
add eax, 0x2f112a91
sub eax, 0x801608e8
xor eax, 0x9cb2998b
xor eax, 0xe626a2be
add eax, 0x3185e741
xor eax, 0x197e9520
xor eax, 0x5665148d
sub eax, 0xc739155d
```

```
add eax, 0x58f934ef
sub eax, 0xa623710f
xor eax, 0x8051cbca
ror eax, 0x1d
ror eax, 0xc
ror eax, 0x1c
xor eax, 0xa96f3357
ror eax, 0xa
xor eax, 0xf13d8c20
not eax
xor eax, 0xfb42f152
add eax, 0xb813492a
sub eax, 0x4f8728ef
add eax, 0xee0e75bc
```

Evading AI Machine Learning

Mixing Morphed Blocks



Head

Morphed Function 1

Tail

Head

Morphed Function 2

Tail

Head

Morphed Function 3

Tail

Disabling the AI from differentiating functions before, during and after execution

Mutation Engines

AV Signature Evasion

Polymorphic Engines

morphed body encryption

Metamorphic Engines

body polymorphic

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Signature Evasion

Morphing Techniques Evading Signature

Instruction reordering	Code Permutation
Subroutine permutation	Instruction Substitution
Subroutine Inlining	Dead Code Insertion
Subroutine Outlining	Changing Control Flow
Expansion	Transposition

Can not resist reverse engineering

Remote Code Evolution

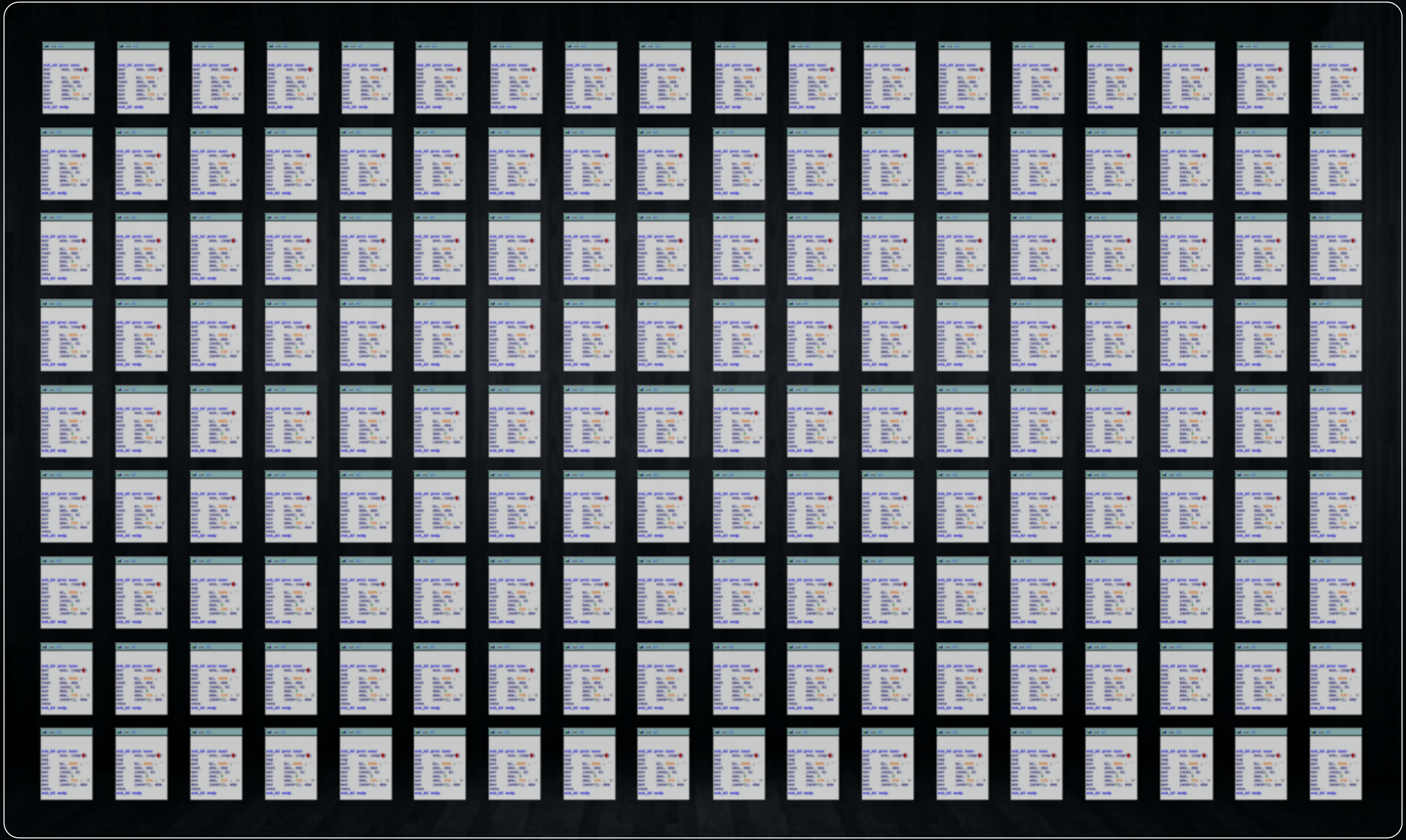
Flux Mutation Goals

- Extend Trust
- Ensure Trusted Remote Execution
- Evade Signature
- Evade AI Machine Learning
- Detect & Evade RE
- Detect Tampering Attempts

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Structure Obfuscation

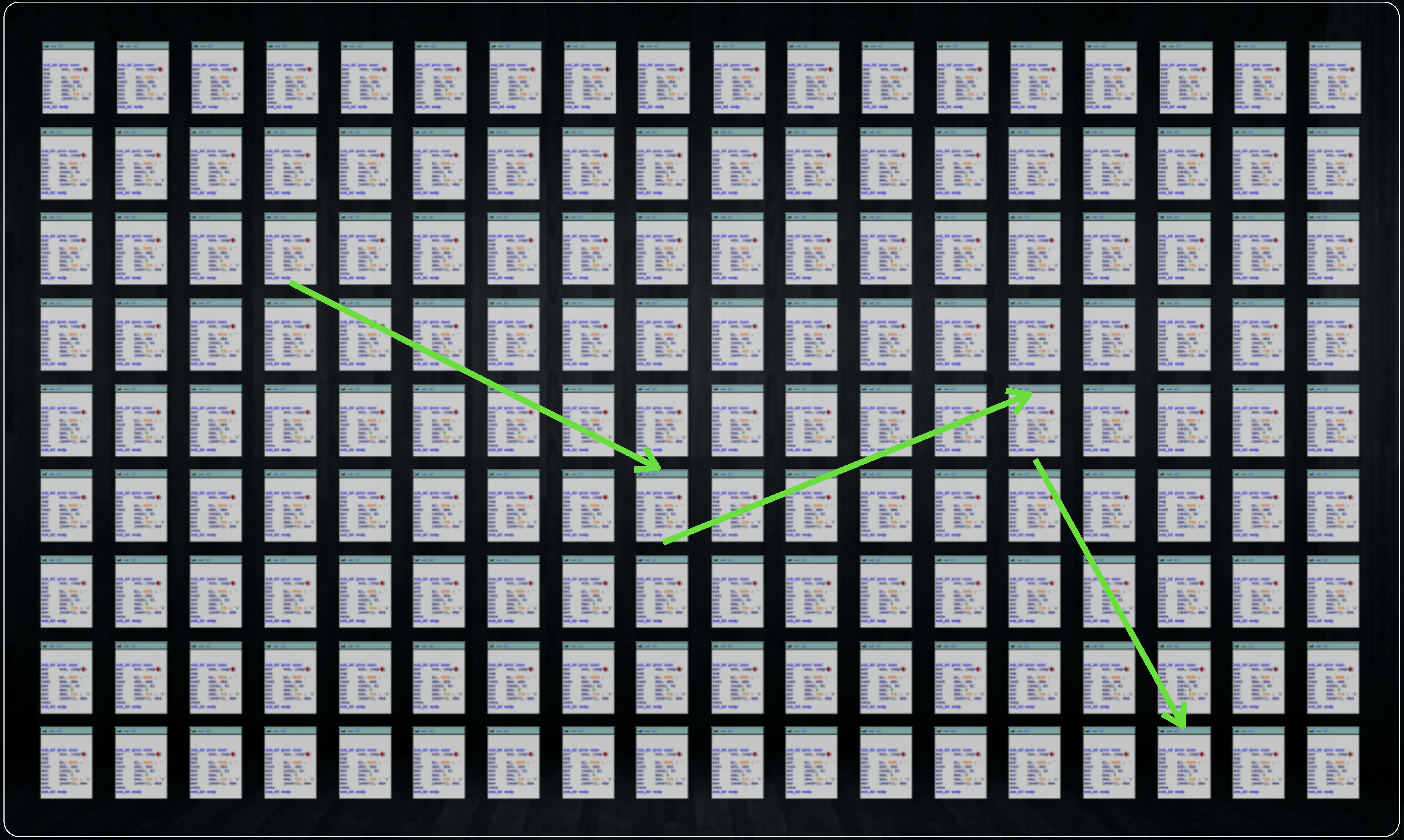
All functions look the same before and during execution



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

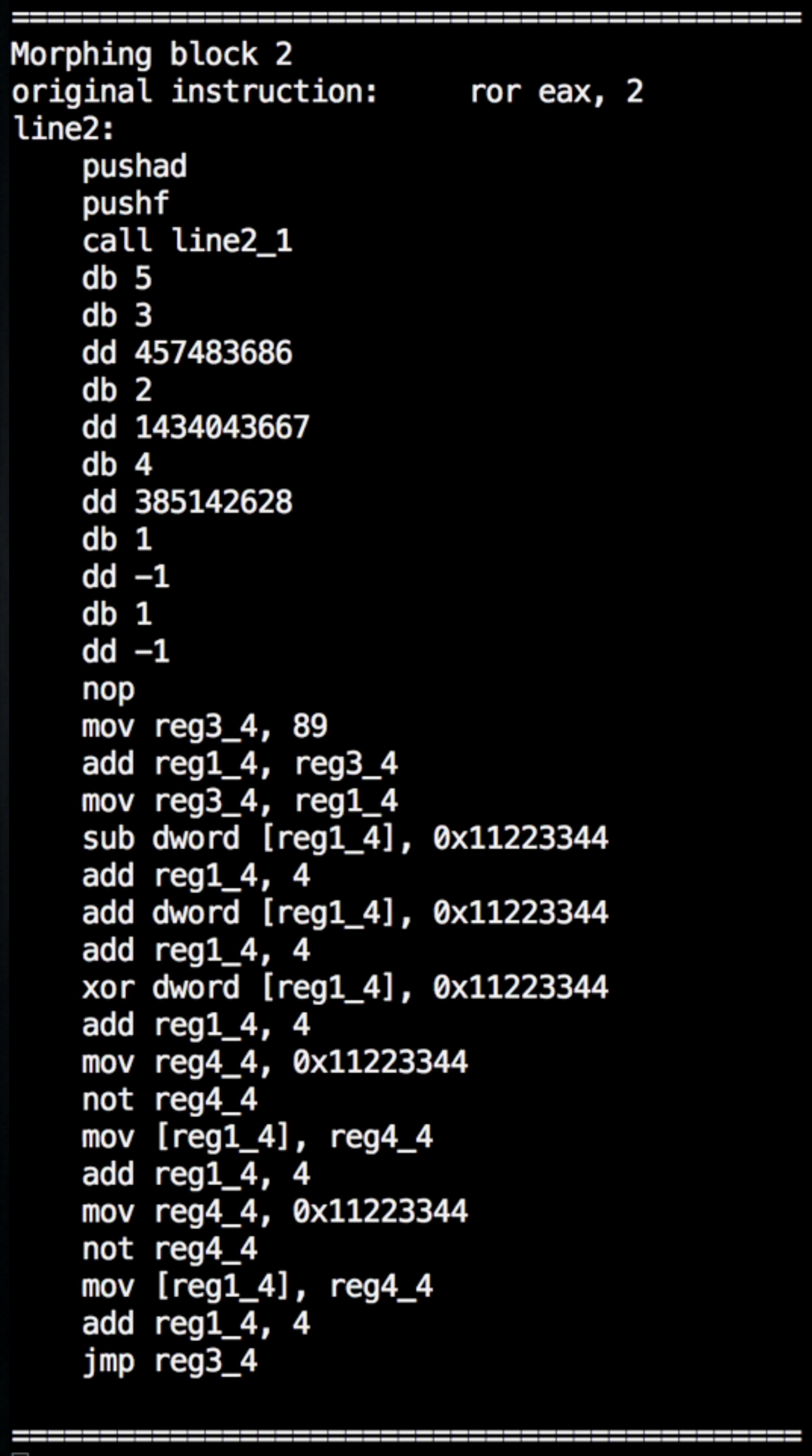
Structure Obfuscation

Self modifying basic block Edges



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```



```
=====================================================================
Morphing block 2
original instruction:     ror eax, 2
line2:
    pushad
    pushf
    call line2_1
    db 5
    db 3
    dd 457483686
    db 2
    dd 1434043667
    db 4
    dd 385142628
    db 1
    dd -1
    db 1
    dd -1
    nop
    mov reg3_4, 89
    add reg1_4, reg3_4
    mov reg3_4, reg1_4
    sub dword [reg1_4], 0x11223344
    add reg1_4, 4
    add dword [reg1_4], 0x11223344
    add reg1_4, 4
    xor dword [reg1_4], 0x11223344
    add reg1_4, 4
    mov reg4_4, 0x11223344
    not reg4_4
    mov [reg1_4], reg4_4
    add reg1_4, 4
    mov reg4_4, 0x11223344
    not reg4_4
    mov [reg1_4], reg4_4
    add reg1_4, 4
    jmp reg3_4

=====================================================================
```

```
db 1
dd -1
nop
mov reg3_4, 89
add reg1_4, reg3_4
mov reg3_4, reg1_4
sub dword [reg1_4], 0x11223344
add reg1_4, 4
add dword [reg1_4], 0x11223344
add reg1_4, 4
xor dword [reg1_4], 0x11223344
add reg1_4, 4
mov reg4_4, 0x11223344
not reg4_4
mov [reg1_4], reg4_4
add reg1_4, 4
mov reg4_4, 0x11223344
not reg4_4
mov [reg1_4], reg4_4
add reg1_4, 4
jmp reg3_4
```

RE Evasion

Morphing Techniques

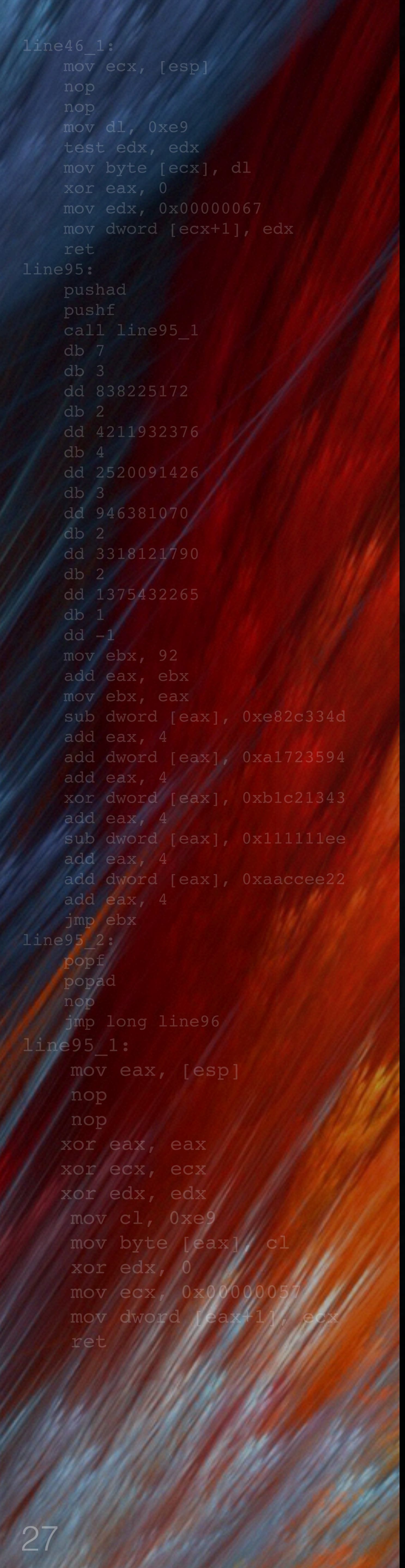
- Metamorphic + Polymorphic
- Self modifying mutation
- Code structure obfuscation
- Clock synchronized execution
- Challenge-Response Mutation
- Functionality Mutation
- Decoupled Reversible Mutation
- Slices Permutation
- Code size magnification

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

Remote Code Evolution

Morphing Techniques

```
_start:
    push 0
    pushad
    mov reg1, [fs:dword 0x30]
    movzx reg2, byte [reg1+2]
    mov dword [esp+32], reg2
    popad
    pop eax
    ret
end:
```



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

Remote Code Evolution

Morphing Techniques

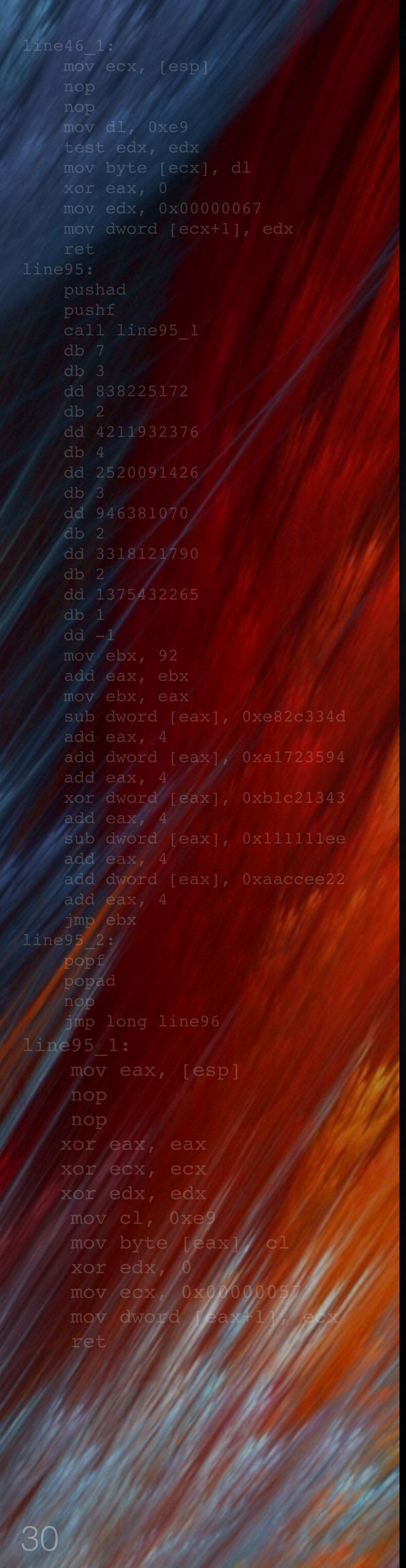
```
          _start:
           { xor reg1, reg1
    push 0 { push reg1
             pushad
             mov reg1, [fs:dword 0x30]
             movzx reg2, byte [reg1+2]
             mov dword [esp+32], reg2
             popad
             pop eax
             ret
          end:
```

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Remote Code Evolution

Morphing Techniques

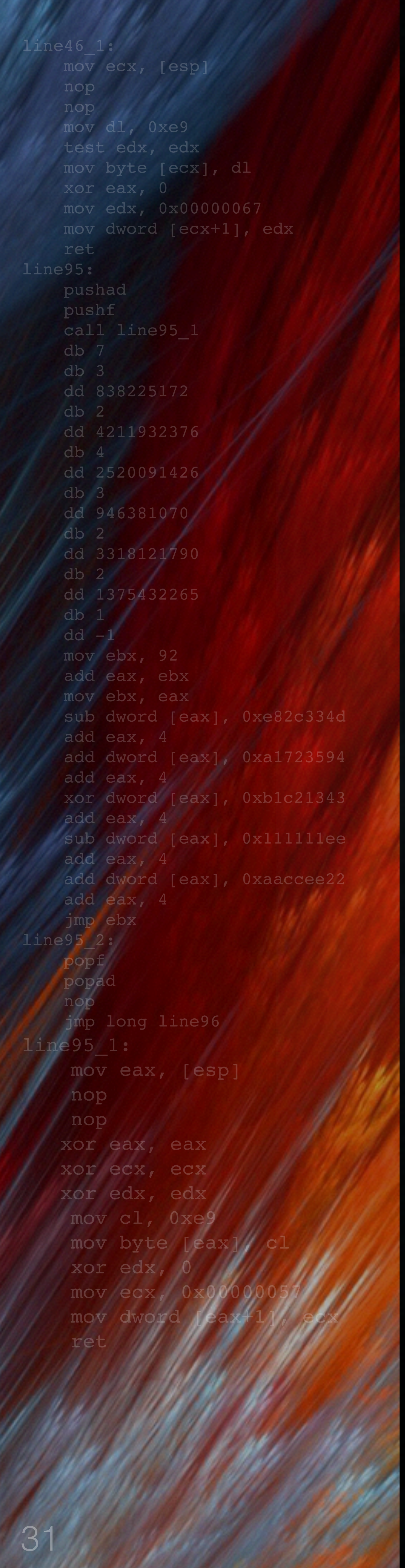
```
           _start:
push 0 {     xor reg1, reg1
             push reg1
             pushad
Insertion⟶   sub reg1, reg1
             mov reg1, [fs:dword 0x30]
             movzx reg2, byte [reg1+2]
             mov dword [esp+32], reg2
             popad
             pop eax
             ret
           end:
```



Remote Code Evolution

Morphing Techniques

```
                 _start:
        push 0 {     xor reg1, reg1
                     push reg1
                     pushad
     Insertion ⟶     sub reg1, reg1
                     mov reg1, [fs:dword 0x30]
     Insertion ⟶     add reg2, reg2
                     movzx reg2, byte [reg1+2]
                     mov dword [esp+32], reg2
                     popad
                     pop eax
                     ret
                 end:
```



Remote Code Evolution

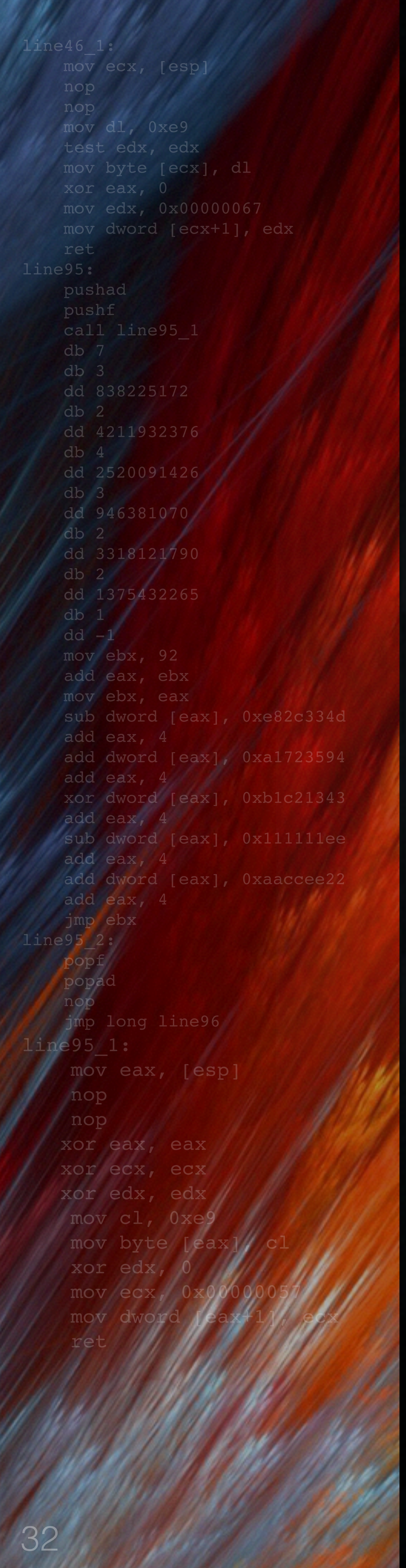
Morphing Techniques

```
                 _start:
       push 0  { xor reg1, reg1
               { push reg1
                 pushad
   Insertion --> sub reg1, reg1
                 mov reg1, [fs:dword 0x30]
   Insertion --> add reg2, reg2
                 movzx reg2, byte [reg1+2]
   Insertion --> mov reg3, reg4
                 mov dword [esp+32], reg2
                 popad
                 pop eax
                 ret
                 end:
```

Remote Code Evolution

Morphing Techniques

```
          _start:
            ⎧ xor reg1, reg1
    push 0  ⎨
            ⎩ push reg1
              pushad
Insertion --> sub reg1, reg1
              mov reg1, [fs:dword 0x30]
Insertion --> add reg2, reg2
              movzx reg2, byte [reg1+2]
Insertion --> mov reg3, reg4
              mov dword [esp+32], reg2
         ↗    popad
n*nop  -->    pop eax
         ↘    ret
          end:
```

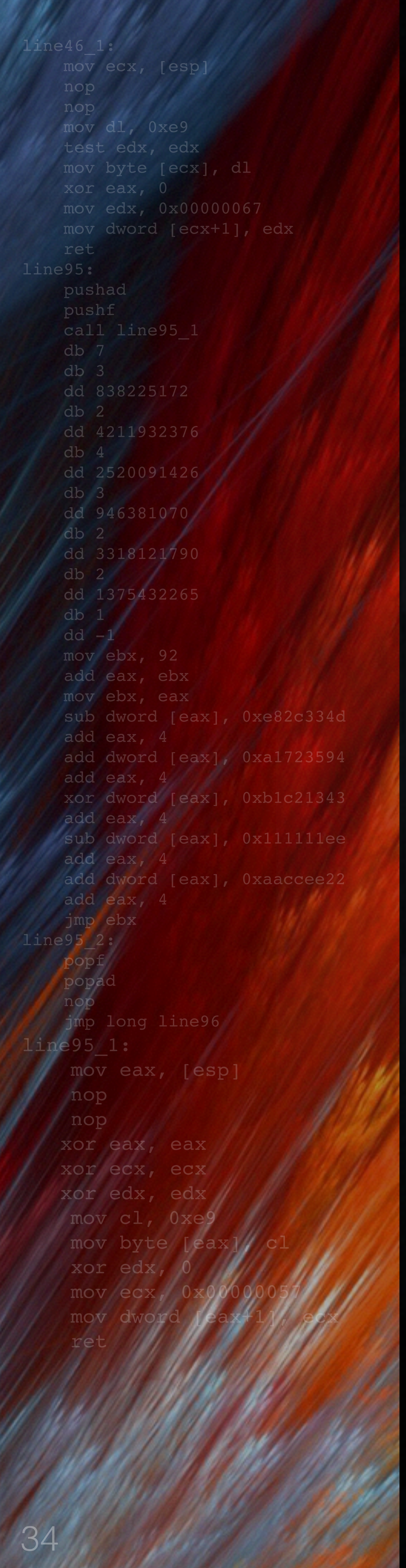


Remote Code Evolution

Morphing Techniques

```
             _start:
              { xor reg1, reg1
     push 0   { push reg1
                pushad
Insertion ⟶    sub reg1, reg1
                mov reg1, [fs:dword 0x30]
Insertion ⟶    add reg2, reg2
                movzx reg2, byte [reg1+2]
Insertion ⟶    mov reg3, reg4
                mov dword [esp+32], reg2
              ↗                            ↓
                popad
  n*nop     ⟶  pop eax      add esp,36
              ↘ ret          push reg2
             end:            sub esp,32
```

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Remote Code Evolution

First Morphing Stage

```
_start:
    xor reg1, reg1
    push reg1
    pushad
    sub reg1, reg1
    mov reg1, [fs:dword 0x30]
    add reg2, reg2
    movzx reg2, byte [reg1+2]
    mov reg3, reg4
    mov dword [esp+32], reg2
    popad
    nop
    pop eax
    nop
    ret
end:
```

Remote Code Evolution

Second Morphing Stage

```
line1:
    xor edi, edi
    jmp long line2
line2:
    push edi
    jmp long line3
line3:
    pushad
    jmp long line4
line4:
    sub edi, edi
    jmp long line5
line5:
    jmp long line6
```

```
line6:
    add ebx, ebx
    jmp long line7
line7:
    movzx ebx, byte [edi+2]
    jmp long line8
line8:
    mov ecx, edx
    jmp long line9
line9:
    mov dword [esp+32], ebx
    jmp long line10
line10:
    nop
    jmp long line11
```

```
line11:
    popad
    jmp long line12
line12:
    nop
    jmp long line13
line13:
    pop eax
    jmp long line14
line14:
    nop
    jmp long line15
line15:
    ret
    jmp long line16
```

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Remote Code Evolution

Third Morphing Stage

```
line1:
    xor edi, edi
    jmp long line2
line6:
    add ebx, ebx
    jmp long line7
line8:
    mov ecx, edx
    jmp long line9
line15:
    ret
    jmp long line16
line11:
    popad
    jmp long line12
```

```
line14:
    nop
    jmp long line15
line13:
    pop eax
    jmp long line14
line3:
    pushad
    jmp long line4
line4:
    sub edi, edi
    jmp long line5
line9:
    mov dword [esp+32], ebx
    jmp long line10
```

```
line12:
    nop
    jmp long line13
line5:
    jmp long line6

line7:
    movzx ebx, byte [edi+2]
    jmp long line8
line2:
    push edi
    jmp long line3
line10:
    nop
    jmp long line11
```

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

Self Modifying Body Polymorphism

Forth Morphing Stage

Random Obfuscation Keys

```
    db 5
    db 1
    dd -1
    db 0
    dd 27
    db 4
    dd 3524080526
    db 0
    dd 7
    db 2
    dd 545547056

line1:
    xor edi, edi
    jmp long line2

line1_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    mov byte [ecx], dl
    mov edx, 0x00000058
    mov dword [ecx+1], edx
    ret
```

```
line1:
    pushad
    pushf
    call line1_1
    db 5
    db 1
    dd -1
    db 0
    dd 27
    db 4
    dd 3524080526
    db 0
    dd 7
    db 2
    dd 545547056
    mov eax, 93
    add ecx, eax
    mov eax, ecx
    mov ebx, 0x11223344
    not ebx
    mov [ecx], ebx
    add ecx, 4
    mov ebx, 0x11223344
    ror ebx, 27
    mov [ecx], ebx
    add ecx, 4
    xor dword [ecx], 0x11223344
    add ecx, 4
    mov ebx, 0x11223344
    ror ebx, 7
    mov [ecx], ebx
    add ecx, 4
    add dword [ecx], 0x11223344
    add ecx, 4
    jmp eax
line1_2:
    popf
    popad
     xor edi, edi
     jmp long line2
    nop
    . . . . 20*nops
    nop
line1_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    mov byte [ecx], dl
    mov edx, 0x00000058
    mov dword [ecx+1], edx
    ret
```

Self modifying instructions

```
    mov eax, 93
    add ecx, eax
    mov eax, ecx
    mov ebx, 0x11223344
    not ebx
    mov [ecx], ebx
    add ecx, 4
    mov ebx, 0x11223344
    ror ebx, 27
    mov [ecx], ebx
    add ecx, 4
    xor dword [ecx], 0x11223344
    add ecx, 4
    mov ebx, 0x11223344
    ror ebx, 7
    mov [ecx], ebx
    add ecx, 4
    add dword [ecx], 0x11223344
    add ecx, 4
    jmp eax
```

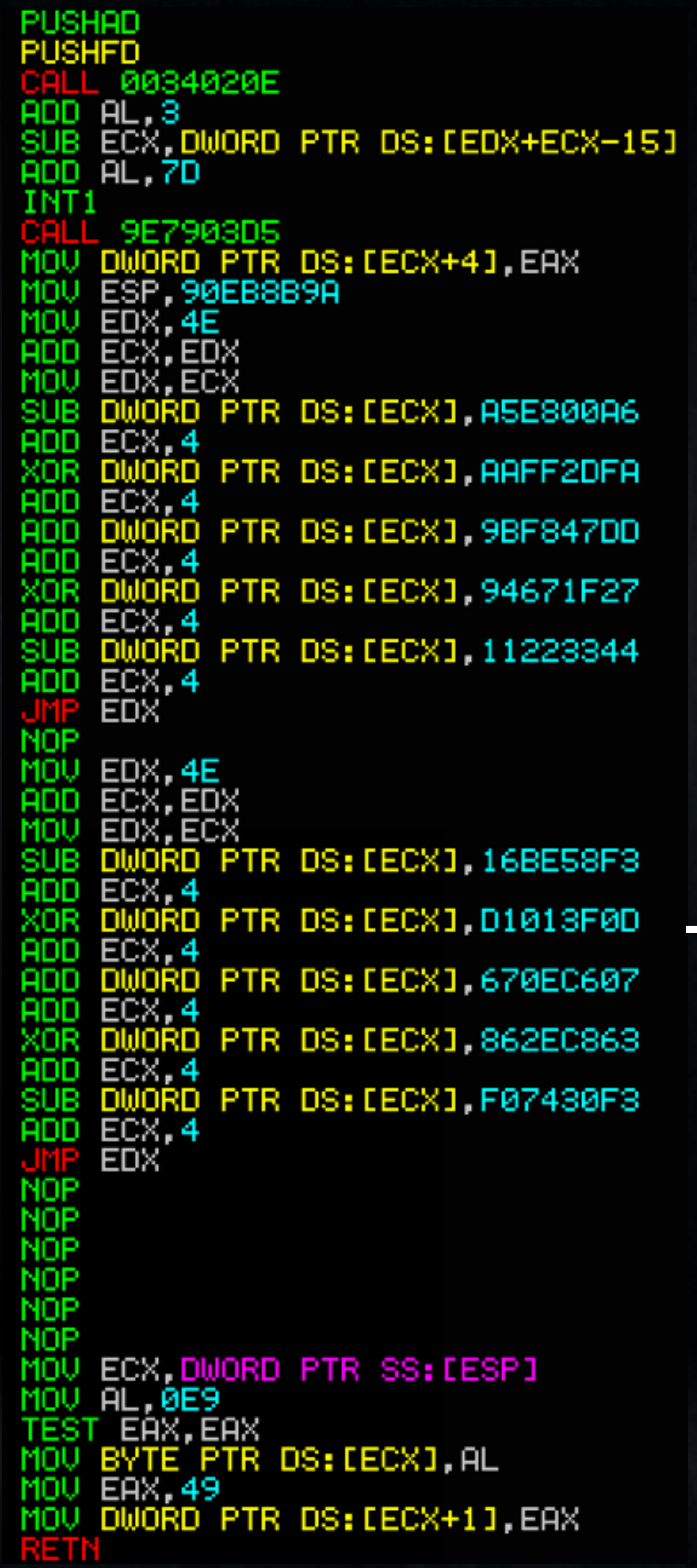
Self Modifying

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

Self Modifying Blocks

Fifth Morphing Stage

Obfuscation Keys ←



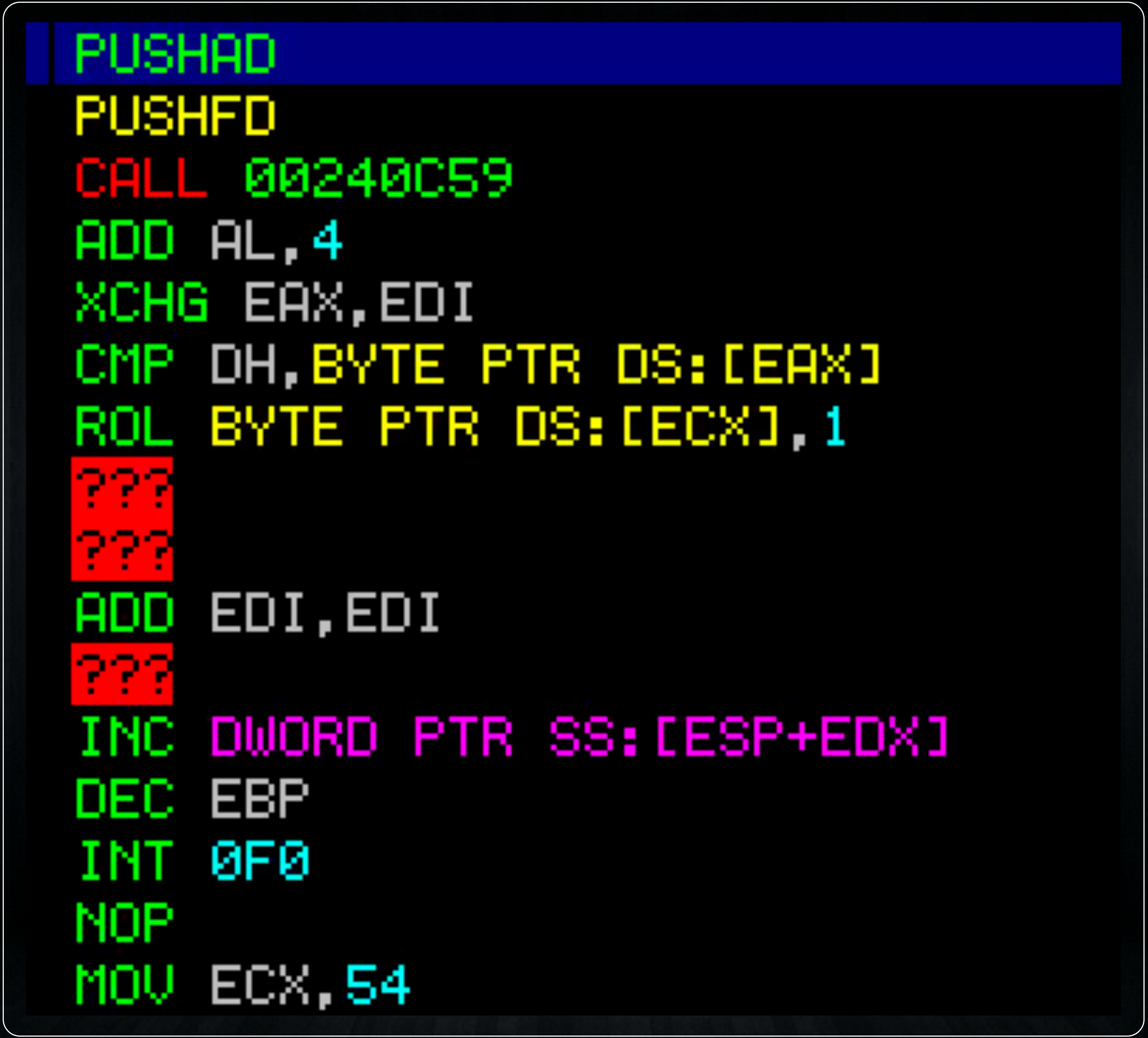
```
PUSHAD
PUSHFD
CALL 0034020E
ADD AL,3
SUB ECX,DWORD PTR DS:[EDX+ECX-15]
ADD AL,7D
INT1
CALL 9E7903D5
MOV DWORD PTR DS:[ECX+4],EAX
MOV ESP,90EB8B9A
MOV EDX,4E
ADD ECX,EDX
MOV EDX,ECX
SUB DWORD PTR DS:[ECX],A5E800A6
ADD ECX,4
XOR DWORD PTR DS:[ECX],AAFF2DFA
ADD ECX,4
ADD DWORD PTR DS:[ECX],9BF847DD
ADD ECX,4
XOR DWORD PTR DS:[ECX],94671F27
ADD ECX,4
SUB DWORD PTR DS:[ECX],11223344
ADD ECX,4
JMP EDX
NOP
MOV EDX,4E
ADD ECX,EDX
MOV EDX,ECX
SUB DWORD PTR DS:[ECX],16BE58F3
ADD ECX,4
XOR DWORD PTR DS:[ECX],D1013F0D
ADD ECX,4
ADD DWORD PTR DS:[ECX],670EC607
ADD ECX,4
XOR DWORD PTR DS:[ECX],862EC863
ADD ECX,4
SUB DWORD PTR DS:[ECX],F07430F3
ADD ECX,4
JMP EDX
NOP
NOP
NOP
NOP
NOP
NOP
MOV ECX,DWORD PTR SS:[ESP]
MOV AL,0E9
TEST EAX,EAX
MOV BYTE PTR DS:[ECX],AL
MOV EAX,49
MOV DWORD PTR DS:[ECX+1],EAX
RETN
```

All blocks have same identical structure

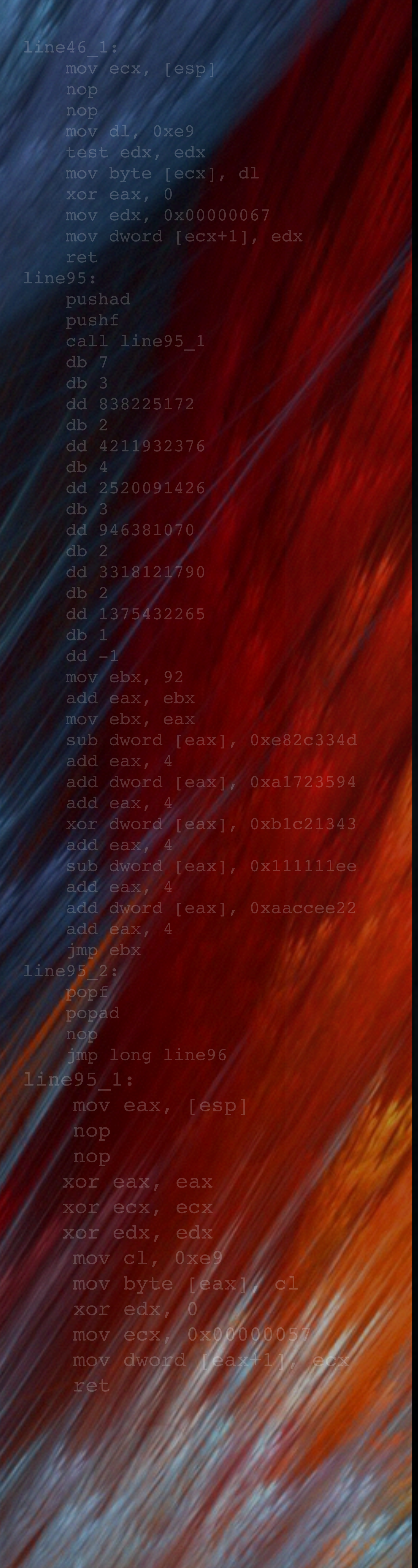
One block per morphed instruction

→ Self modifying code

Self Modifying Blocks



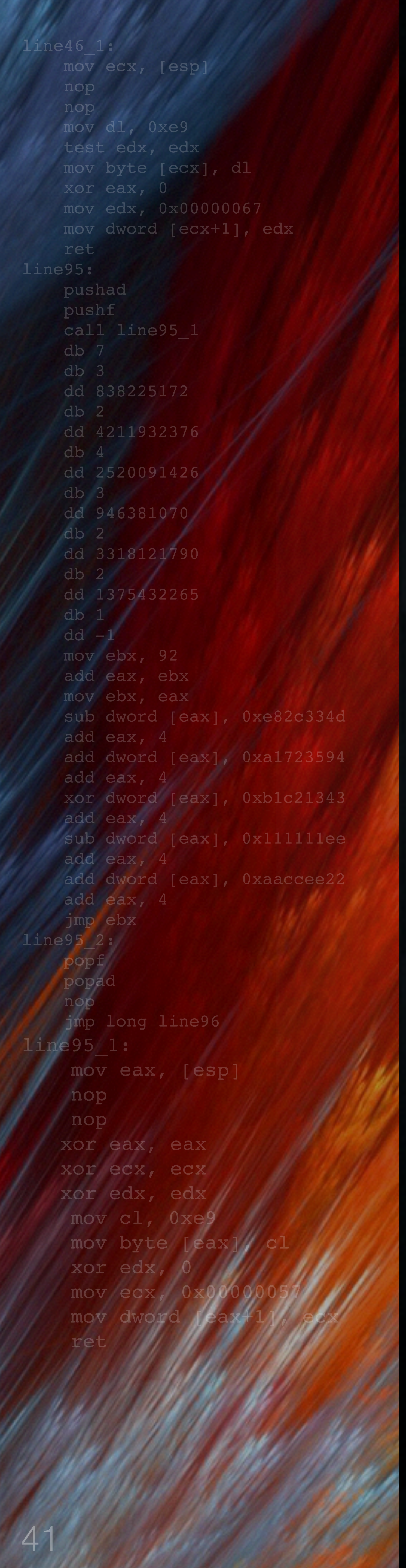
```
PUSHAD
PUSHFD
CALL 00240C59
ADD AL,4
XCHG EAX,EDI
CMP DH,BYTE PTR DS:[EAX]
ROL BYTE PTR DS:[ECX],1
???
???
ADD EDI,EDI
???
INC DWORD PTR SS:[ESP+EDX]
DEC EBP
INT 0F0
NOP
MOV ECX,54
```



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Response Time

```
[+] mutated code size: 15110 bytes
[+] encrypted response: 0x09575e31 | 156720689
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.685972 ms

[+] mutated code size: 17771 bytes
[+] encrypted response: 0x5820b6b5 | 1478538933
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.040096 ms

[+] mutated code size: 23814 bytes
[+] encrypted response: 0x5d844e9a | 1568951962
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.897926 ms

[+] mutated code size: 19768 bytes
[+] encrypted response: 0x818af8d8 | -2121598760
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.177187 ms
```



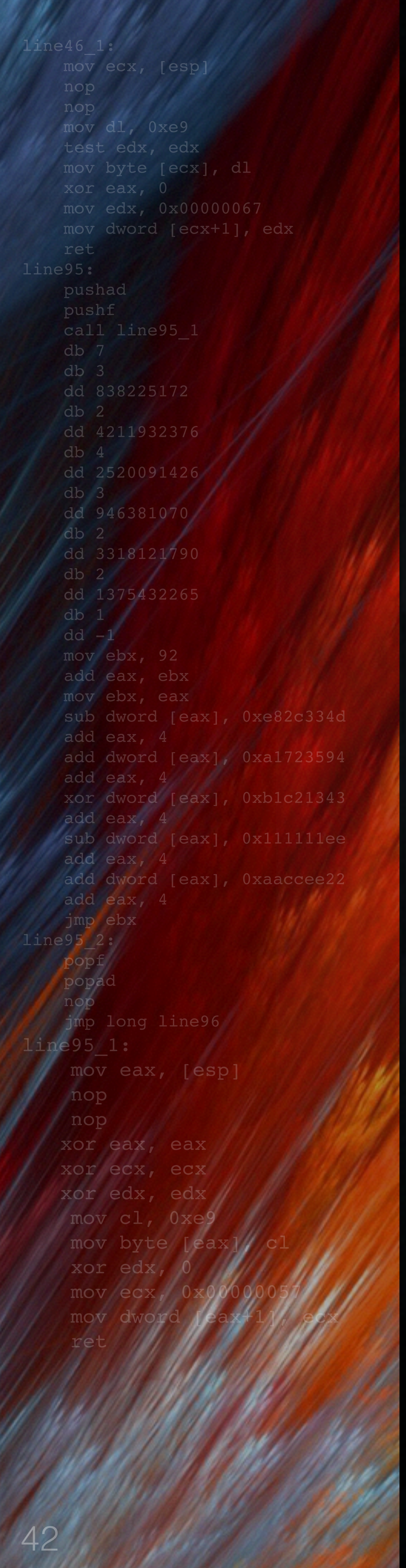
```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

Variable Code Size

```
[+] mutated code size: 15110 bytes
[+] encrypted response: 0x09575e31 | 156720689
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.685972 ms

[+] mutated code size: 17771 bytes
[+] encrypted response: 0x5820b6b5 | 1478538933
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.040096 ms

[+] mutated code size: 23814 bytes
[+] encrypted response: 0x5d844e9a | 1568951962
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.897926 ms

[+] mutated code size: 19768 bytes
[+] encrypted response: 0x818af8d8 | -2121598760
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.177187 ms
```

Response Mutation

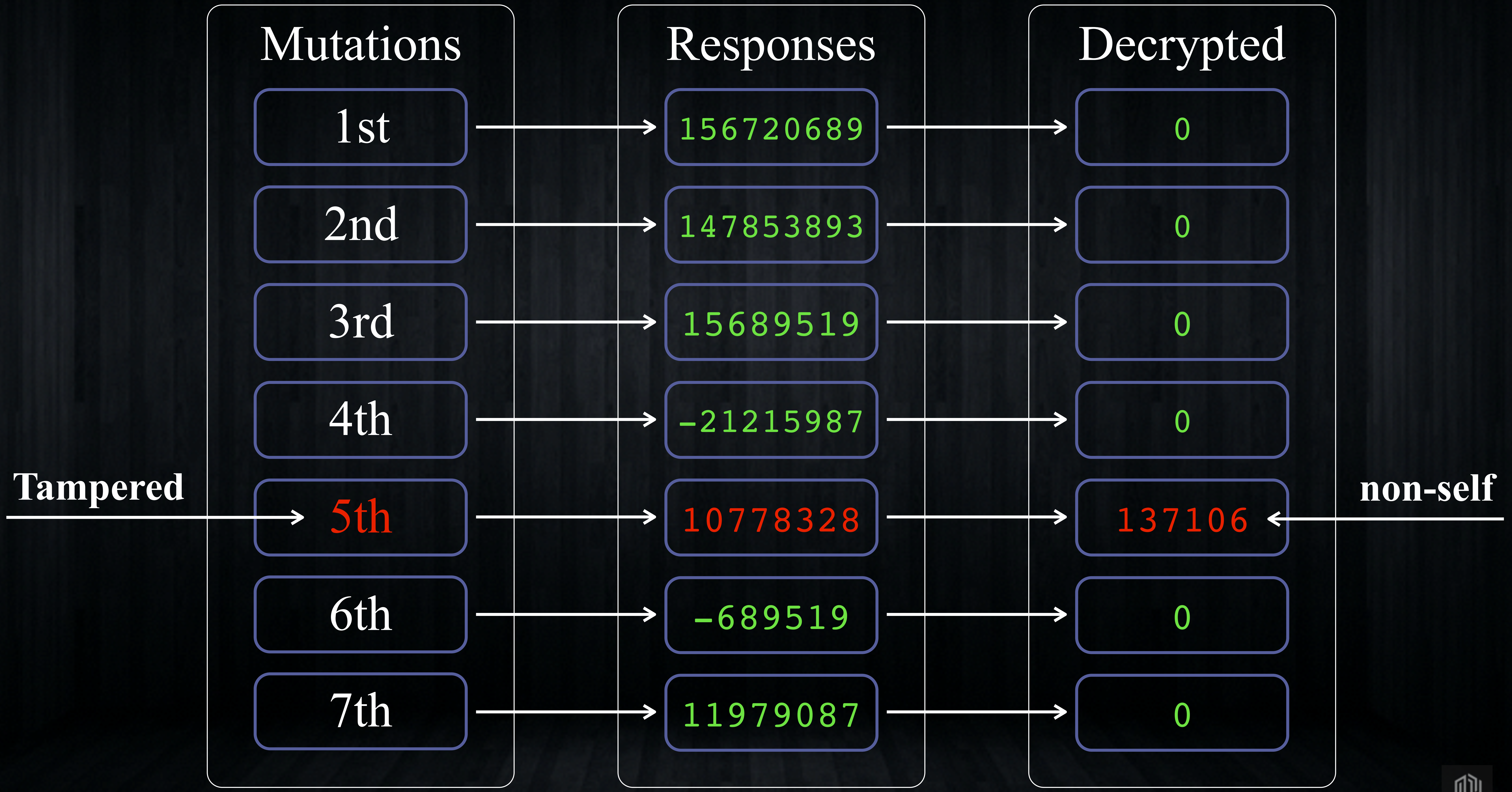
```
[+] mutated code size: 15110 bytes
[+] encrypted response: 0x09575e31 | 156720689
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.685972 ms

[+] mutated code size: 17771 bytes
[+] encrypted response: 0x5820b6b5 | 1478538933
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.040096 ms

[+] mutated code size: 23814 bytes
[+] encrypted response: 0x5d844e9a | 1568951962
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.897926 ms

[+] mutated code size: 19768 bytes
[+] encrypted response: 0x818af8d8 | -2121598760
[+] decrypted response: 0x00000001 | 1
[+] remote execution response time: 6.177187 ms
```



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

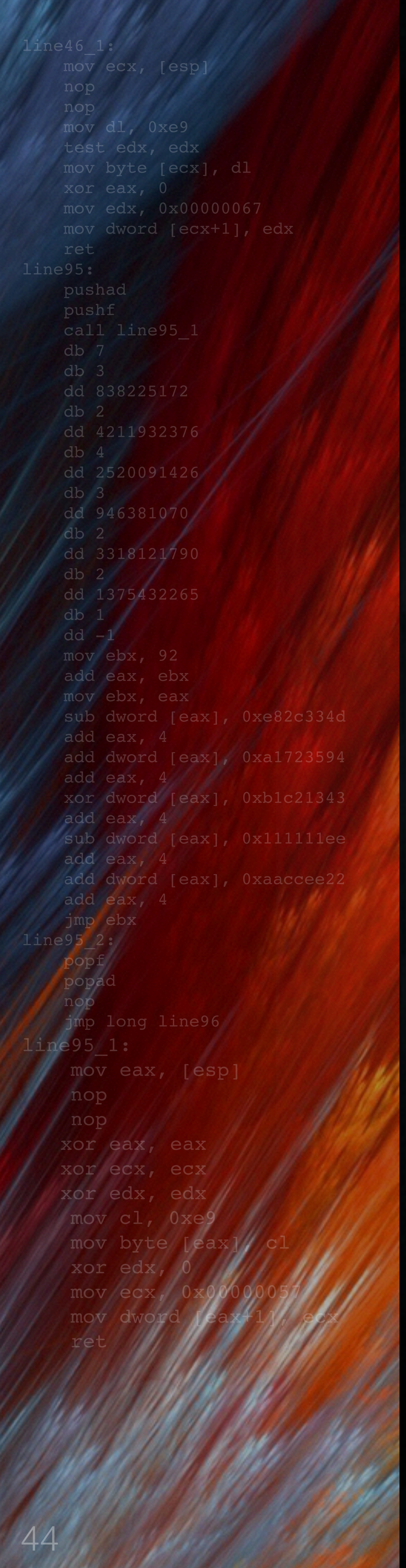
The Remote Metamorphic Engine

Artificial Immunity | Detecting the non-self



	Mutations	Responses	Decrypted	
	1st	156720689	0	
	2nd	147853893	0	
	3rd	15689519	0	
	4th	-21215987	0	
Tampered	5th	10778328	137106	non-self
	6th	-689519	0	
	7th	11979087	0	

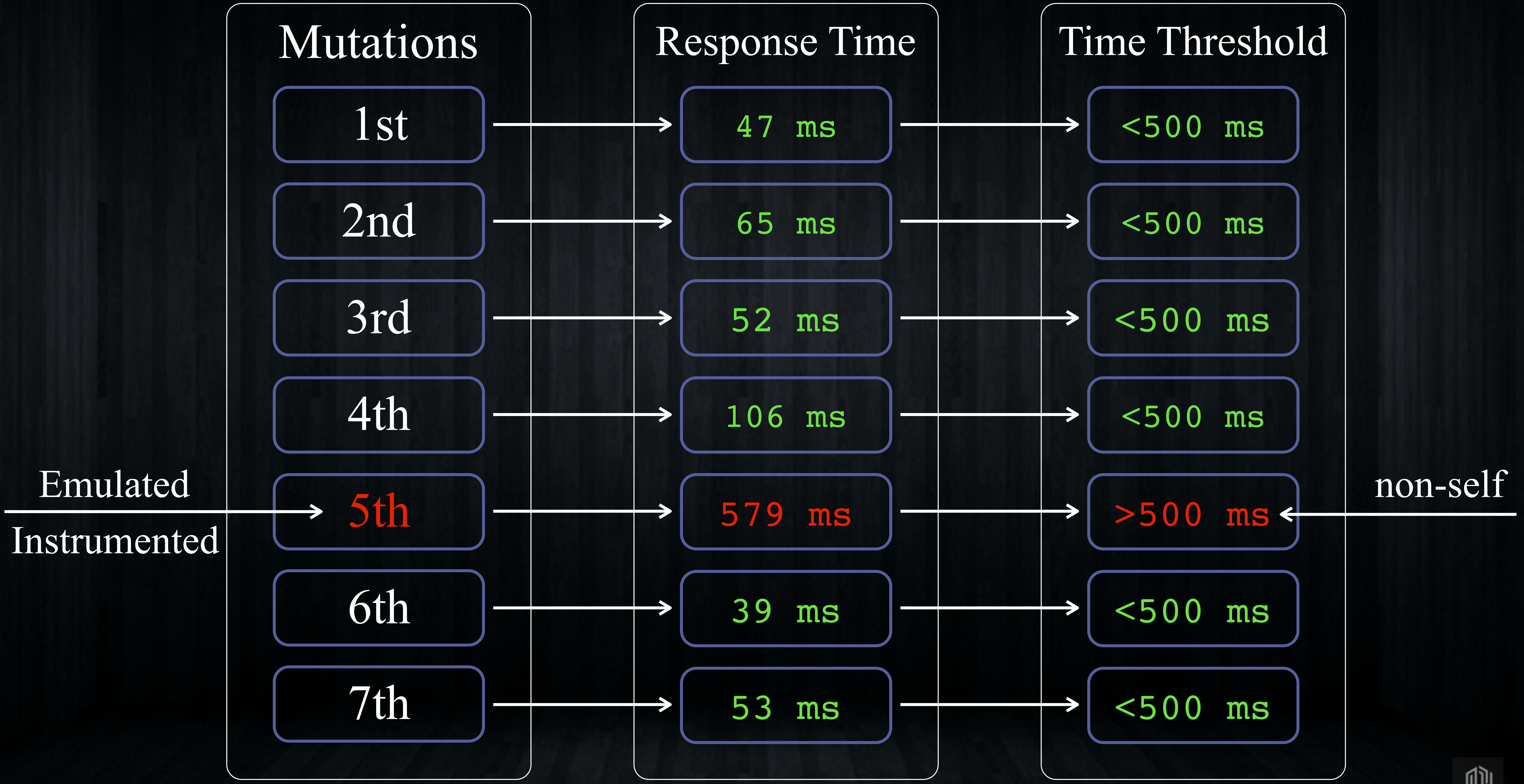
```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
   xor eax, eax
   xor ecx, ecx
   xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```

The Remote Metamorphic Engine

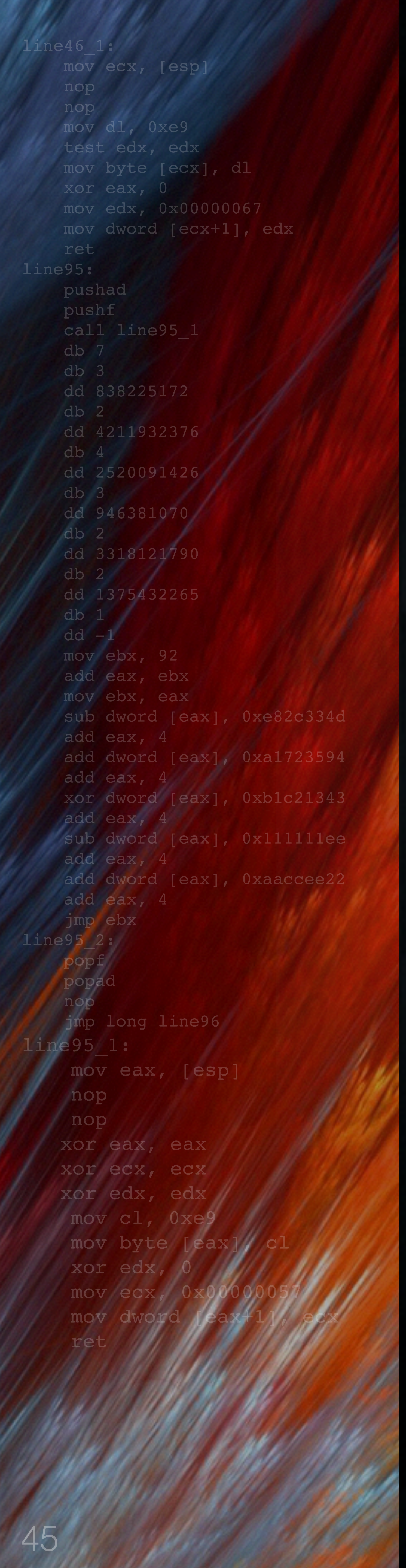
Artificial Immunity | Detecting the non-self



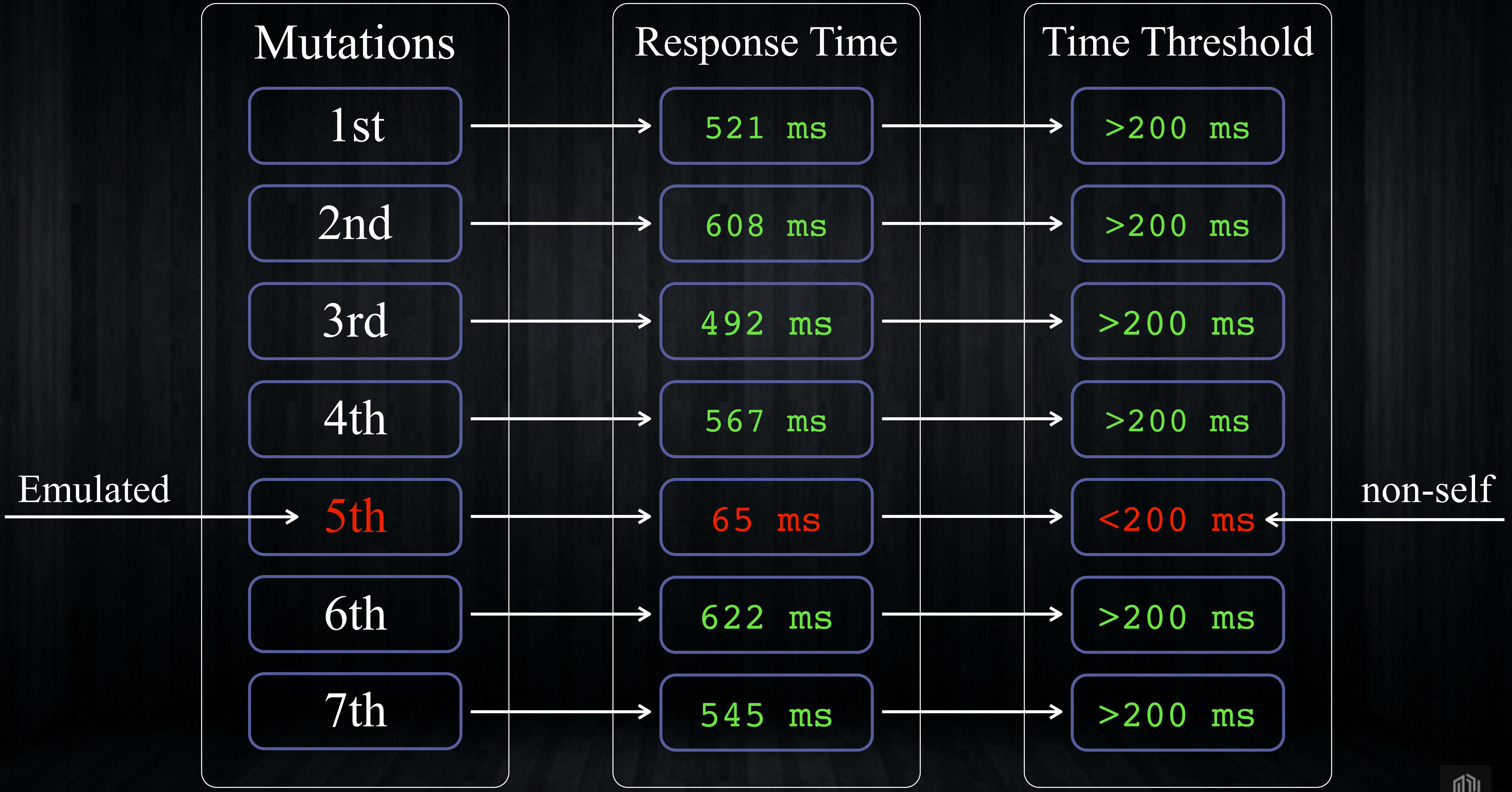
Mutations	Response Time	Time Threshold
1st	47 ms	<500 ms
2nd	65 ms	<500 ms
3rd	52 ms	<500 ms
4th	106 ms	<500 ms
5th	579 ms	>500 ms
6th	39 ms	<500 ms
7th	53 ms	<500 ms

The Remote Metamorphic Engine

Artificial Immunity | Detecting the non-self



```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
     mov eax, [esp]
     nop
     nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
     mov cl, 0xe9
     mov byte [eax], cl
     xor edx, 0
     mov ecx, 0x00000057
     mov dword [eax+1], ecx
     ret
```



Mutations	Response Time	Time Threshold
1st	521 ms	>200 ms
2nd	608 ms	>200 ms
3rd	492 ms	>200 ms
4th	567 ms	>200 ms
5th	65 ms	<200 ms
6th	622 ms	>200 ms
7th	545 ms	>200 ms

Emulated

non-self

```
line46_1:
    mov ecx, [esp]
    nop
    nop
    mov dl, 0xe9
    test edx, edx
    mov byte [ecx], dl
    xor eax, 0
    mov edx, 0x00000067
    mov dword [ecx+1], edx
    ret
line95:
    pushad
    pushf
    call line95_1
    db 7
    db 3
    dd 838225172
    db 2
    dd 4211932376
    db 4
    dd 2520091426
    db 3
    dd 946381070
    db 2
    dd 3318121790
    db 2
    dd 1375432265
    db 1
    dd -1
    mov ebx, 92
    add eax, ebx
    mov ebx, eax
    sub dword [eax], 0xe82c334d
    add eax, 4
    add dword [eax], 0xa1723594
    add eax, 4
    xor dword [eax], 0xb1c21343
    add eax, 4
    sub dword [eax], 0x111111ee
    add eax, 4
    add dword [eax], 0xaaccee22
    add eax, 4
    jmp ebx
line95_2:
    popf
    popad
    nop
    jmp long line96
line95_1:
    mov eax, [esp]
    nop
    nop
    xor eax, eax
    xor ecx, ecx
    xor edx, edx
    mov cl, 0xe9
    mov byte [eax], cl
    xor edx, 0
    mov ecx, 0x00000057
    mov dword [eax+1], ecx
    ret
```

{ Questions? }

remote.metamorphic.engine@gmail.com