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
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

---

**Remote Metamorphic Engine**

**Isolation**

**Randomization**

**RE Time**

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

$-\infty = 0$

$\infty$

**Undefined**
The Unbreakable Code

Unpredictable

adjective:/ˌʌnprɛdɪktəb(ə)l/

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

Dynamic Code Dynamic Data

Code evolution across time
Functionality evolution across location
Self contained autonomous code

Unpredictable
Self aware
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
The Remote Metamorphic Engine

Remote Flux Mutation

**Trusted Zone**
- Remote Mutation
- Mutation Engine
- Challenge

**Untrusted Zone**
- Thread/Process
- Morphed Code Execution
- Response

The Remote Metamorphic Engine

Remote Flux Mutation
Why Remote?
The Remote Metamorphic Engine

Challenge Response Metamorphic Protocol

**Trusted Zone**

Remote Mutation

Mutation Engine

**Challenge**

4 bytes size

Code

**Clock Synced**

Response

**Untrusted Zone**

Thread/Process

Morphed Code Execution

Communication protocol made of morphed clock synchronized machine code rather than data.
The Remote Metamorphic Engine

Remote Code Slicing

The Engine Side
Unknown to the reverse engineer

The Reverse Engineer Side
Known to the reverse engineer

12
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
Trusted Mutation

Trusted Challenge Response Mutation

Remote Mutation

Morphing Engine

Function

Head

Unused Code

Morphed Function

Return value

Mutated Challenge

Morphed Function

Tail

Response Mutation
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

Remote Mutation

Morphing Engine

Function

Morphed Function

Head

Morphed Function

Tail

Unused Code

Return value

Reversible Mutation

Trusted Zone

Mutated Challenge

Maturated Function

Morphed Function

Head

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)
Evading AI Machine Learning

Mixing Morphed Blocks

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

Mutation Engines

AV Signature Evasion

Polymorphic Engines
morphed body encryption

Metamorphic Engines
body polymorphic
Signature Evasion

Morphing Techniques Evading Signature

<table>
<thead>
<tr>
<th>Instruction reordering</th>
<th>Code Permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subroutine permutation</td>
<td>Instruction Substitution</td>
</tr>
<tr>
<td>Subroutine Inlining</td>
<td>Dead Code Insertion</td>
</tr>
<tr>
<td>Subroutine Outlining</td>
<td>Changing Control Flow</td>
</tr>
<tr>
<td>Expansion</td>
<td>Transposition</td>
</tr>
</tbody>
</table>

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
Structure Obfuscation

All functions look the same before and during execution
Structure Obfuscation
Self modifying basic block Edges
```assembly
Morphing block 2
original instruction:      ror eax, 2
lined:
  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                       
  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               
  xor reg4_4, 0x11223344       
  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               
  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
  xor reg4_4, 0x11223344
  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
Remote Code Evolution

Morphing Techniques

```plaintext
_start:
    push 0
    pushad
    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

```assembly
_start:
push 0
{xor reg1, reg1
push reg1
pushad
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

```assembly
_start:
push 0 {  
xor reg1, reg1
  push reg1
  pushad
  Insertion→sub reg1, reg1
  mov reg1, [fs:dword 0x30]
  movzx reg2, byte [reg1+2]
  mov reg2, byte [reg1+2]
  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

```assembly
_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

    n*nop
    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

n*nop

popad
pop ecx
ret

end:
add esp,36
push reg2
sub esp,32
Remote Code Evolution

First Morphing Stage

```asm
_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

line 1:
  xor edi, edi
  jmp long line 2

line 2:
  push edi
  jmp long line 3

line 3:
  pushad
  jmp long line 4

line 4:
  sub edi, edi
  jmp long line 5

line 5:
  jmp long line 6

line 6:
  add ebx, ebx
  jmp long line 7

line 7:
  movzx ebx, byte [edi+2]
  jmp long line 8

line 8:
  mov ecx, edx
  jmp long line 9

line 9:
  mov dword [esp+32], ebx
  jmp long line 10

line 10:
  nop
  jmp long line 11

line 11:
  popad
  jmp long line 12

line 12:
  nop
  jmp long line 13

line 13:
  pop eax
  jmp long line 14

line 14:
  nop
  jmp long line 15

line 15:
  ret
  jmp long line 16
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
```
Self Modifying Body Polymorphism

Forth Morphing Stage

Random Obfuscation Keys

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

Self modifying instructions

```
  mov eax, 93
  add ecx, eax
  mov eax, ecx
  mov ebx, 0x11223344
  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
Self Modifying Blocks

Fifth Morphing Stage

Obfuscation Keys

One block per morphed instruction

All blocks have same identical structure

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
```
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
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
The Remote Metamorphic Engine

Artificial Immunity | Detecting the non-self

<table>
<thead>
<tr>
<th>Mutations</th>
<th>Responses</th>
<th>Decrypted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>156720689</td>
<td>0</td>
</tr>
<tr>
<td>2nd</td>
<td>147853893</td>
<td>0</td>
</tr>
<tr>
<td>3rd</td>
<td>15689519</td>
<td>0</td>
</tr>
<tr>
<td>4th</td>
<td>-21215987</td>
<td>0</td>
</tr>
<tr>
<td>5th</td>
<td>10778328</td>
<td>137106</td>
</tr>
<tr>
<td>6th</td>
<td>-689519</td>
<td>0</td>
</tr>
<tr>
<td>7th</td>
<td>11979087</td>
<td>0</td>
</tr>
</tbody>
</table>
### The Remote Metamorphic Engine

**Artificial Immunity | Detecting the non-self**

<table>
<thead>
<tr>
<th>Mutations</th>
<th>Response Time</th>
<th>Time Threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st</td>
<td>47 ms</td>
<td>&lt;500 ms</td>
</tr>
<tr>
<td>2nd</td>
<td>65 ms</td>
<td>&lt;500 ms</td>
</tr>
<tr>
<td>3rd</td>
<td>52 ms</td>
<td>&lt;500 ms</td>
</tr>
<tr>
<td>4th</td>
<td>106 ms</td>
<td>&lt;500 ms</td>
</tr>
<tr>
<td><strong>5th</strong></td>
<td><strong>579 ms</strong></td>
<td>&gt;500 ms</td>
</tr>
<tr>
<td>6th</td>
<td>39 ms</td>
<td>&lt;500 ms</td>
</tr>
<tr>
<td>7th</td>
<td>53 ms</td>
<td>&lt;500 ms</td>
</tr>
</tbody>
</table>

**Mutations**
- 1st
- 2nd
- 3rd
- 4th
- **5th**
- 6th
- 7th

**Response Time**
- 47 ms
- 65 ms
- 52 ms
- 106 ms
- **579 ms**
- 39 ms
- 53 ms

**Time Threshold**
- <500 ms
- <500 ms
- <500 ms
- <500 ms
- >500 ms
- <500 ms
- <500 ms

**Comparison**
- **Emulated**
- **Instrumented**
- non-self
The Remote Metamorphic Engine

Artificial Immunity | Detecting the non-self

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
Questions?

remote.metamorphic.engine@gmail.com