A SCAFFOLDED METAMORPHIC CTF FOR REVERSE ENGINEERING

Motivation

Scaffolded

Metamorphic

Extensible

Levels being added via internships and course projects
- Distributed programming template
- Script provides random data per user
- Combined with program template to produce unique program per user

Deployable

Integrated website
- Distributing binaries
- Submission and validation of per-student solutions
- Bitbucket repository for source and website for instructors

Evaluation

Status and future

25 levels at http://midware.asegmc.org --- gridX
MidwareCTF for web security (based on gridX)
MidwareCTF for CS 321 (Bryant & O'Halloran) (held)
CTF for high school classes and camps
- Saturday Academy/CyberAcademy (youth, seniors)
- 15 levels of n00bs, 8 levels of movement
- Diverse-themed CTF and Urban Race (seniors only)
A SCAFFOLDED METAMORPHIC CTF FOR REVERSE ENGINEERING

Motivation

Scaffolded

Metamorphic

Extensible

Levels being added via internships and course projects
Build script and program template
- Script produces random data per user
- Combined with program template to produce unique program per user

Deployable

Integrated website
- Distributing binaries
- Submission and validation of per-student solutions
- Bitbucket repository for source and website for instructors

Evaluation

Table of Evaluation

<table>
<thead>
<tr>
<th>Function</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>80%</td>
</tr>
<tr>
<td>Task 2</td>
<td>75%</td>
</tr>
<tr>
<td>Task 3</td>
<td>85%</td>
</tr>
<tr>
<td>Task 4</td>
<td>90%</td>
</tr>
</tbody>
</table>

Status and future

25 interns at http://secquote.org
MetacTF for web security based on virtual
MetacTF for CS 214 (Hryant & O'Halloran 2018)
CTF for high school classes and camps
- Saturday Academy CyberAcademy (junior/senior)
- 15 levels of tasks, 6 levels of tournament
- Diverse-themed CTF and urban race (seniormores)
Motivation

Capture-the-Flag (CTF) competitions
Increasingly popular vehicle for sharpening security skills
- ICTF, PiaidCTF, CSAW, DEFCON

Goal is to evaluate rather than teach
- Challenges often open-ended, unguided, and esoteric
- Limited pedagogy
- Can be frustrating for beginners

CTFs for Instruction
Goal is to teach rather than just evaluate
- Develop skills, competence, and confidence rapidly

Examples
- Integrated into courses
- Scaffolded CTFs
  - picoCTF (PHP, Python Eval, ROP, Overflow)
  - natas @ overthewire.org (web exploitation)
  - microcorruption.com (memory exploitation)

MetaCTF
Jeopardy-style CTF for reverse engineering
- Scaffolded for quick progression and skill development
- Metamorphic to reduce cheating and allow reuse
- Extensible and configurable to support customization
- Easily deployed
Capture-the-Flag (CTF) competitions

Increasingly popular vehicle for sharpening security skills
- iCTF, PlaidCTF, CSAW, DEFCON

Goal is to evaluate rather than teach
- Challenges often open-ended, unguided, and esoteric
- Limited pedagogy
- Can be frustrating for beginners
CTFs for Instruction

Goal is to teach rather than just evaluate
  • Develop skills, competence, and confidence rapidly

Examples
  • Integrated into courses
  • Scaffolded CTFs
    • picoCTF (PHP, Python Eval, ROP, Overflow)
    • natas @ overthewire.org (web exploitation)
    • microcorruption.com (memory exploitation)
MetaCTF

Jeopardy-style CTF for reverse engineering
- Scaffolded for quick progression and skill development
- Metamorphic to reduce cheating and allow reuse
- Extensible and configurable to support customization
- Easily deployed
A SCAFFOLDED METAMORPHIC CTF FOR REVERSE ENGINEERING

Motivation

Scaffolded

Metamorphic

Extensible

Levels being added via internships and course projects
- Build levels and program template
- Script probes random data per user
- Combined with program template to produce unique program per user

Deployable

Integrated web site
- Distributing binaries
- Submission and validation of per-student solutions
- Bitbucket repository for source and website for instructors

Evaluation

Status and future

25 level at http://makeware.engineering.org -- go!X
- MetacTF for web security (based on nss)
- MetacTF for CS 3201 (Bryant & O'Halloran guided)
- CTFs for high school classes and camps
- Saturday Academy Cyberacademy (junior, senior)
- 15 levels of nss, 8 levels of mor UNDER
- Divergent-themed CTF and Urban Race (sophomores)
Scaffolded

Level Design
- Integrated with textbook
  - Focus on a specific topic
- Guided, direct instruction approach
- Uniform level operation
  - Find password to force binary to print "Good job"

Basic static and dynamic analysis
- Find password stored in read-only data section
  - readelf, objdump
- Find password via library tracing
  - ltrace

Advanced static analysis
- Disassembly
  - Decode embedded ASCII
  - Decode jump table for switch
- Disable debuggers

Advanced dynamic analysis
- Code statically compiled
- Password procedurally generated
- Use breakpoints to find password

Malware functionality
- Reverse techniques malware uses
  - Follow process spawning
  - Undo simple encoding (Base64, XOR)
- Employ techniques malware does
  - Hijacking dynamic library loading
  - Hijacking import address tables (procedure link tables)

Bypassing adversarial protections
- Anti-disassembly
  - Obfuscated control-flow instructions
  - Fake conditionals, impossible disassembly
- Anti-debugging
  - Debugger detection (ptrace, INT 3, timing)
  - Debugger trolling (SIGTRAP trap, entanglement)
- Packers
  - Dynamic unpacking and dumping
Level Design

Integrated with textbook
  • Focus on a specific topic

Guided, direct instruction approach

Uniform level operation
  • Find password to force binary to print “Good Job.”
Basic static and dynamic analysis

Find password stored in read-only data section
  • readelf, objdump

Find password via library tracing
  • ltrace
Advanced static analysis

Disassembly

- Decode embedded ASCII
  
  80484b4: movb $0x1,0x804a11d  
  804851e: movb $0x31,0x14(%esp)  
  8048523: movb $0x4e,0x15(%esp)  
  8048528: movb $0x54,0x16(%esp)  
  804852d: movb $0x49,0x17(%esp)  
  8048532: movb $0x77,0x18(%esp)  
  8048537: movb $0xe4,0x19(%esp)  
  804853c: movb $0x6a,0x1a(%esp)  
  8048541: movb $0x42,0x1b(%esp)

- Decode jump table for switch
  
  movl 28(%esp), %eax .L5:  
  subl $10647, %eax .long .L4  
  cmpl $4, %eax .long .L6  
  ja .L3 .long .L7  
  movl .L5(,%eax,4), %eax .long .L6  
  jmp *%eax .long .L4

- Disable debuggers
  
  void detectTrace(void) __attribute__((constructor));
  void detectTrace (void) {
      if(ptrace(PTRACE_TRACEME, 0, 1, 0) < 0) {
          printf("Sorry, we have disallowed debuggers on this assignment.\n");
          exit(1);
      }
  };
Advanced dynamic analysis

Debugging

- Code statically compiled
- Password procedurally generated
- Use breakpoints to find password

```
0x80488c8 <main+168>       lea   0x24(%esp),%eax
0x80488c4 <main+172>       mov   %eax,%esp
0x80488ce <main+175>       call  0x8061fd0 <strncpy>
0x8048ef4 <main+180>       test  %eax,%eax

(gdb) break strncpy
Breakpoint 3 at 0x8061fd0
(gdb) c
Continuing.

Breakpoint 3, 0x8061fd0 in strncpy ()
(gdb) x/4xw $esp
0xffffffffc: 0x80488c8 0xffffffffd 0xffffffffd
(gdb) x/s $0xffffffffd
0xffffffffd: "TNIwNjBi"
```
Malware functionality

Reverse techniques malware uses
• Follow process spawning
• Undo simple encoding (Base64, XOR)

Employ techniques malware does
• Hijacking dynamic library loading
• Hijacking import address tables (procedure link tables)
mask_output[cnt] = enc_table[(mod+rand())%64];
...
printf("%d %d %d\n",rand(),rand(),rand());
printf("Hint: %s\n",mask_output);

mashimaro ~> % export LD_PRELOAD=rand.so
mashimaro ~> % ./Ch11MalBeh_LdPreload
...
Enter the password: foo
0 0 0
Hint: i2abIun48
Try again.
mashimaro ~> % export LD_PRELOAD=
mashimaro ~> % ./Ch11MalBeh_LdPreload
...
Enter the password: i2abIun48
1350490027 1025202362 1189641421
Hint: J8DOZtxkl
Good Job.
mashimaro ~> %
Malware functionality

Reverse techniques malware uses
- Follow process spawning
- Undo simple encoding (Base64, XOR)

Employ techniques malware does
- Hijacking dynamic library loading
- Hijacking import address tables (procedure link tables)
```c
void print_good() {
    printf("Good Job.\n");
    exit(0);
}

main() {
    ...
    *ip = i;
    printf("Address %x will contain %x\n", ip, i);
    sleep(1);
    printf("Try again.\n");
    ...
}
```

```
(gdb) disassemble 0x80483f0
Dump of assembler code for function sleep@plt:
    0x080483f0 <+0>:   jmp   *0x4e548014
    0x080483f6 <+6>:   push   $0x10
    0x080483fb <+11>:  jmp   0x80483c0

End of assembler dump.
(gdb) p (void *) &print_good
$1 = (void *) 0x4e54686d <print_good>
```

```
... Enter the password: 4e548014 04e54686d
Address 4e548014 will contain 4e54686d
Good Job.
mashimaro <~> %
```
Malware functionality

Reverse techniques malware uses
- Follow process spawning
- Undo simple encoding (Base64, XOR)

Employ techniques malware does
- Hijacking dynamic library loading
- Hijacking import address tables (procedure link tables)
Bypassing adversarial protections

Anti-disassembly
- Obfuscated control-flow instructions
- Fake conditionals, impossible disassembly

Anti-debugging
- Debugger detection (ptrace, INT 3, timing)
- Debugger trolling (SIGTRAP trap, entanglement)

Packers
- Dynamic unpacking and dumping
main:
push ebp
mov ebp, esp
and esp, 0FFFFFF0h
sub esp, 20h
call print_msg
push eax
cmp eax, eax
jz short near ptr loc_804859C+1

loc_804859C:
addps xmm0, xmm7
inc esp
and al, 1Ch
in eax, dx
retn

main:
push ebp
mov ebp, esp
and esp, 0FFFFFF0h
sub esp, 20h
call print_msg
push eax
cmp eax, eax
jz short loc_804859D

loc_804859D:
pop eax
mov dword ptr [esp+1Ch], 24C3EDh
mov dword ptr [esp], offset _EnterThePassword ; "Enter the password:"
Bypassing adversarial protections

Anti-disassembly
- Obfuscated control-flow instructions
- Fake conditionals, impossible disassembly

Anti-debugging
- Debugger detection (ptrace, INT 3, timing)
- Debugger trolling (SIGTRAP trap, entanglement)

Packers
- Dynamic unpacking and dumping
cmp [ebp+var_10], 1
jle short loc_804852F

loc_80484EA:
mov edx, [ebp+var_14]
mov eax, [ebp+var_C]
add eax, edx
movzx eax, byte ptr [eax]
cmp al, 00Ch
jnz short loc_80484FD

mov eax, [ebp+var_10]
mov [esp+4], eax
mov dword ptr [esp], offset format ; "Found 0xcc %d times. You must be debugg"

Printf
mov dword ptr [esp], 0 ; status
exit
Bypassing adversarial protections

Anti-disassembly
- Obfuscated control-flow instructions
- Fake conditionals, impossible disassembly

Anti-debugging
- Debugger detection (ptrace, INT 3, timing)
- Debugger trolling (SIGTRAP trap, entanglement)

Packers
- Dynamic unpacking and dumping
mov     word ptr [eax-6], 3344h
mov     dword ptr [esp], offset format ; "Enter the password:"
call    _printf
lea     eax, [esp+28h]
mov     [esp+4], eax
mov     dword ptr [esp], offset a19s ; "%19s"
call    __isoc99_scansf
call    sigtrap
test    eax, eax
jnz     short loc_80486E6

mov     byte ptr [esp+1Ch], 0

loc_80486E6:
lea     eax, [esp+14h]
mov     [esp], eax ; s
call    _strlen
...
Bypassing adversarial protections

Anti-disassembly
- Obfuscated control-flow instructions
- Fake conditionals, impossible disassembly

Anti-debugging
- Debugger detection (ptrace, INT 3, timing)
- Debugger trolling (SIGTRAP trap, entanglement)

Packers
- Dynamic unpacking and dumping
Metamorphic

Unique per-student binaries
- Reduce cheating
- Allow re-use across schools
- Allow re-use over multiple offerings

Data

Code

Code location
Unique per-student binaries

Data and code of binaries polymorphic and metamorphic
- Reduce cheating
- Allow re-use across schools
- Allow re-use over multiple offerings
Data

Ch01StatA_Readelf

```
mashimaro <wuchang@pdx.edu> % readelf -x 15 1.01a_readelf
Hex dump of section '.rodata':
  0x08048658 03000000 01000200 25730045 6e746572 ...........s.Enter
  0x08048668 20746865 20706173 73776f72 643a2000 the password: 
  0x08048678 25387300 5a546631 4d7a6468 00536f72 %s.ZTkIMzdhs.
  0x08048688 72792e20 20547279 20616761 696e0047 ry. Try again.G
  0x08048698 6f6f6420 4a6f6200 ood Job.

mashimaro <bsull2@pdx.edu> 1:40PM % readelf -x 15 1.01a_readelf
Hex dump of section '.rodata':
  0x08048658 03000000 01000200 25730045 6e746572 ...........s.Enter
  0x08048668 20746865 20706173 73776f72 643a2000 the password: 
  0x08048678 25387300 4e475a6b 4e574531 00536f72 %s.NGZkNWE1.sor
  0x08048688 72792e20 20547279 20616761 696e0047 ry. Try again.G
  0x08048698 6f6f6420 4a6f6200 ood Job.
```
Ch15AntiDis_FakeMetaConds

80485c2:    call    804852d <print_msg>
80485c7:    stc
80485c8:    jb    80485cb <main+0x12>
80485ca:    (bad)
80485cc:    inc    %esp
80485cd:    and    $0x1c,%al
80485cf:    adc    0x4c700cc(%edi),%ebx
80485d5:    and    $0x20,%al
80485d7:    xchg   %eax,(%eax,%ecx,1)
80485da:    call    80483b0 <printf@plt>
80485df:    lea    0x18(%esp),%eax
80485e3:    mov    %eax,0x4(%esp)
80485e7:    movl    $0x8048735,(%esp)

80485c2:    call    804852d <print_msg>
80485c7:    clc
80485c8:    jae    80485cb <main+0x12>
80485ca:    (bad)
80485cc:    inc    %esp
80485cd:    and    $0x1c,%al
80485cf:    out    %eax,$0x11
80485d1:    lods   %ds:(%esi),%al
80485d2:    add    %al,%bh
80485d4:    add    $0x24,%al
80485d6:    and    %al,-0x2e17f7fc(%edi)
80485dc:    std
80485dd:    (bad)
80485de:    decl   -0x76e7dbbc(%ebp)
80485e4:    inc    %esp
80485e5:    and    $0x4,%al
80485e7:    movl    $0x8048735,(%esp)
Code location

Ch11MalBeh_HijackPLT

4e7a476d <print_good>:
push %ebp
4e7a476d:   mov %esp,%ebp
4e7a4770:   sub $0x18,%esp
4e7a4773:   movl $0x4e7a4960,(%esp)

4e545a6d <print_good>:
push %ebp
4e545a6d:   mov %esp,%ebp
4e545a70:   sub $0x18,%esp
4e545a73:   movl $0x4e545c60,(%esp)

080483f0 <sleep@plt>:
jmp *0x4e7a6014

080483f0 <sleep@plt>:
jmp *0x4e547014
Extensible

Levels being added via internships and course projects

Build script and program template
- Script produces random data per user
- Combined with program template to produce unique program per-user
Deployable

Integrated web site
  • Distributing binaries
  • Submission and validation of per-student solutions
BitBucket repository for source and website for instructors
A SCAFFOLDED METAMORPHIC CTF FOR REVERSE ENGINEERING

Motivation

Scaffolded

Metamorphic

Extensible

Levels being added via internships and course projects
- Shells that can be used to create new levels
- Script provides random data per user
- Combined with program template to produce unique program per user

Deployable

Integrated website
- Distribution binaries
- Submission and validation of per-student solutions
- GitHub repository for source and website for instructors

Evaluation

<table>
<thead>
<tr>
<th>CTF Size</th>
<th>Difficulty</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000 MB</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>2 GB</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>4 GB</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

Status and Future

25 levels at http://metamorphic.engineering -- goes
MetacTF for web security (based on ncc
MetacTF for CS 321 (Bryant & O'Halloran 1st ed)
CTFs for high school classes and camps
- Saturday Academy Cyberacademy (juniors, seniors)
- 15 levels of rats, 8 levels of micromutation
- Divergent-themed CTFs and Urban Race (Sophomores)
Evaluation

CS 492/592: Malware

Q13. Quality and usefulness of homework assignments

<table>
<thead>
<tr>
<th>Term</th>
<th>Respondents</th>
<th>Mean rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring 2010</td>
<td>12</td>
<td>4.08 ± 0.67</td>
</tr>
<tr>
<td>Winter 2011</td>
<td>9</td>
<td>4.11 ± 0.60</td>
</tr>
<tr>
<td>Winter 2012</td>
<td>7</td>
<td>3.67 ± 1.2</td>
</tr>
<tr>
<td>Winter 2013</td>
<td>8</td>
<td>4.25 ± 0.71</td>
</tr>
<tr>
<td>Winter 2014</td>
<td>8</td>
<td>4.20 ± 1.1</td>
</tr>
<tr>
<td>Winter 2015</td>
<td>6</td>
<td>4.67 ± 0.82</td>
</tr>
</tbody>
</table>

- Sikorski adopted, simple binaries
- Expanded binaries (17)
Status and future

25 levels at http://malware.oregonctf.org => gseX

MetaCTF for web security (based on natas)

MetaCTF for CS 201 (Bryant & O’Halloran’s 3rd ed)

CTFs for high-school classes and camps
  • Saturday Academy CyberAcademy (juniors, seniors)
    • 15 levels of natas, 8 levels of microcorruption
  • Divergent-themed CTF and Urban Race (sophomores)
A SCAFFOLDED METAMORPHIC CTF FOR REVERSE ENGINEERING

Motivation

Scaffolded

Metamorphic

Extensible

Deployable

Levels being added via internships and course projects
- Related tool and program template
- Script produces random data per user
- Combined with program template to produce unique program per user
- Integrated website
- Distributing binaries
- Submission and validation of per student solutions
- Bitbucket repository for source and website for instructors

Evaluation

Status and future

25 levels at http://metamorph.bergenfed.org — goX
MetacTF for web security (based on real)
MetacTF for CS 201 (Bryant & O’Hara; the Jiff)
CTF for high school classes and camps
- Saturday Academy CyberAcademy (junior, senior)
- 15 levels of rats, 8 levels of microcontrol
- Dvergent-themed CTF and Urban Race (sophomores)