Fixing Bugs in Binaries

Luis Miras
lmiras <o> gmail.com
Agenda

- Introduction
- Feasibility of binary fixes
- Current third party patches
- Strategy
- Working Example
- Future of third party patches
- Questions

RECON 2006
Introduction

- “Bah another bug !@#$$%”
- Is the src available?
- What dependencies do we have?

**Don't wait for a vendor to get around to it, fix vendor bugs yourself!**
Feasibility of binary fixes

- Consider !/$
- Complexity of fix relative to the bug.
- Fixing an “off by one” is easy
- Adding class members can get complicated.
Current third party patches

- The most common 3rd party patches are “cracks”.
- Ilfak’s WMF patch
- Determina’s and Eeye’s CreateTextRange() patch
Strategy

Mantra: “least amount of change that gets the job done”
Strategy

1. Find the root cause of the bug
2. Locate the problem code in disassembly
3. Make a fix
4. Test, test, and test
5. Refine/Refactor the fix
6. Test, test, and test
Applying fixes

- Patching the file on disk/storage
- Patching at runtime. – injecting code, dll, hooks, etc.
- Hybrid – Loaders (packers)
Working Example

Verizon xv6700

RECON 2006
xv6700 specs

- Windows Mobile 2005
- 416Mhz pxa270 ARM Processor
- 1.3 Mega pixel Camera
- EVDO
- WIFI
- Bluetooth
- USB ActiveSync Access
WM2005 Intro

- Part of the WinCE 5.0 family
- WinCE 5.0
- Windows Mobile 2005 Pocket PC
- Windows Mobile 2005 Smartphone
- Windows Mobile 2005 Pocket PC with Phone
WM2005 Intro

- Little Endian
- Very close to Win32
- Portable Executables
- Same style hooking
- Debugging with VS2005 and IDA ARM debugger
The Bug

- Reported on forums, pictures can’t be mailed through Gmail pop3.
- Google is performing file validation.
- JPEG corruption
Image Analysis

Virtual G – PPC imaging software

RECON 2006
Image Analysis

- Images opened and saved from Virtual G pass the Gmail test.
- File changes only in Exif section.
JPEG and Exif use markers for data.

ex. FFxx

<table>
<thead>
<tr>
<th>SOI Marker</th>
<th>APP1 Marker</th>
<th>APP1 Data</th>
<th>Other Marker</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFD8</td>
<td>FFE1</td>
<td>SSSS 457869660000 TTTT......</td>
<td>FFXX SSSS DDDD......</td>
</tr>
</tbody>
</table>

E1 – Start of the Exif Header

SSSS – Size field (includes 2 bytes for size field)

45786966 – ASCII for ‘Exif’
Image Analysis

RECON 2006
Image Analysis

RECON 2006
Image Analysis

RECON 2006
Image Analysis

- Exif header is too large.
- “Off by one”
- Manually changing the size field passes the Gmail file check.
- The bug is identified!
ARM assembly intro

- RISC 32 bit
- Little/Big Endian
- 16 bit Thumb code
- Load/Store Architecture

http://blogs.msdn.com/windowsmobile/archive/2005/05/05/10/ArmTutorial.aspx
Registers

16 Registers

R0-R3  General, functions args
R4-R11, R12  General
SP  Stack Pointer
PC  Program Counter
LR  Link Register
Instructions

Assembly mostly reads right to left like Intel

```
mov reg, reg
mov reg, #0x00
ldrb reg, address
strb reg, address
```
Camera.exe Analysis

- Copy it to PC using ActiveSync connection
- IDA identifies file as an ARM PE
- Flirt recognizes WinCE libs
Bug in Disassembly

- Need to locate the bug in disassembly
- Locate construction of Exif header
- “Exif” not found in strings list
Bug in Disassembly

• Look for individual letters being written out.

\[ \text{mov register, } \#0x45; \text{ ‘E’} \]
\[ \text{strb register, buffer} \]

... Success at function 0x05BCC0
### Bug in Disassembly

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
<th>Operands</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0005BD38</td>
<td>MOV R1, #0x45</td>
<td>; E</td>
<td></td>
</tr>
<tr>
<td>0005BD3C</td>
<td>STRB R1, [R2,#2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[ snip ]</td>
<td></td>
</tr>
<tr>
<td>0005BD58</td>
<td>MOV R0, #0x78</td>
<td>; x</td>
<td></td>
</tr>
<tr>
<td>0005BD5C</td>
<td>STRB R0, [R2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[ snip ]</td>
<td></td>
</tr>
<tr>
<td>0005BD78</td>
<td>MOV R0, #0x69</td>
<td>; i</td>
<td></td>
</tr>
<tr>
<td>0005BD7C</td>
<td>STRB R0, [R2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[ snip ]</td>
<td></td>
</tr>
<tr>
<td>0005BD98</td>
<td>MOV R0, #0x66</td>
<td>; f</td>
<td></td>
</tr>
<tr>
<td>0005BD9C</td>
<td>STRB R0, [R2]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RECON 2006**
Runtime Analysis

- Combining runtime and static analysis speeds up the process
- Visual Studio 2005 debugger can connect to devices over ActiveSync


RECON 2006
Runtime Analysis

- Set break points on function 0x05BCC0 and xref function (parent)
- ‘E’ gets written to a buffer identifying the location of size
- “Watch” for the size being written
Runtime Analysis

Size writing identified:

05BF7C ADD R0, R3, #0xFF00
05BF80 ADD R1, R0, #0xFE
05BF84 MOV R2, R1, LSL#16
05BF88 MOV R0, R2, LSR#16
05BF8C MOV R1, R0, LSR#8
05BF90 ADD R0, R3, #0xFE
05BF94 STRB R1, [R4] ; size write
05BF98 AND R2, R1, #0xFF
05BF9C STRB R0, [R4, #1] ; size write
The fix can divert execution before the last write (lsb).

A branch is put in, but first space to put in code must be found.
Where to patch?

- Code caves between functions
- Extending sections
- New sections
Extending a Section

RECON 2006
Extending a Section

.text section virtual size = 0x89BB4

.text section raw size = 0x89C00

Virtual size can be increased to 0x89C00
Producing 0x4C of extra space
The Fix

05BF9C  STRB  R0, [R4, #1]         B  loc_9ABC0
05BFA0  AND  R1, R0, #0xFF        MOV  R9, R9
05BFA4  ADD  R0, SP, #0x10+arg_4   MOV  R9, R9
05BFA8  BL  sub_48400              MOV  R9, R9
05BFAC  MOV  R0, #1                MOV  R9, R9
05BFB0  LDMFD SP, {R4, R5, SP, PC} MOV  R9, R9

09ABC0  SUB  R11, R0, #1
09ABC4  STRB  R11, [R4, #1]
09ABC8  AND  R1, R0, #0xFF
09ABCC  ADD  R0, SP, #0x10+arg_4
09ABD0  BL  sub_48400
09ABD4  MOV  R0, #1
09ABD8  LDMFD SP, {R4, R5, SP, PC}

RECON 2006
Signed Code Error

RECON 2006

File Explorer

Camera.exe

The file 'Camera' cannot be opened. Either it is not signed with a trusted certificate, or one of its components cannot be found. You might need to reinstall or restore this file.
Signed Code Fix

Signed code is pointed to by the Security Data Directory in the PE header.

Set RVA and size to NULL.
Signed Code Fix

<table>
<thead>
<tr>
<th>Member</th>
<th>Offset</th>
<th>Size</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exe Rva</td>
<td>00000180</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Exe Size</td>
<td>00000184</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Import Rva</td>
<td>00000188</td>
<td>Dwond</td>
<td>00093628</td>
</tr>
<tr>
<td>Import Size</td>
<td>0000018C</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Resource Rva</td>
<td>00000190</td>
<td>Dwond</td>
<td>000A6D00</td>
</tr>
<tr>
<td>Resource Size</td>
<td>00000194</td>
<td>Dwond</td>
<td>00013A14</td>
</tr>
<tr>
<td>Exception Rva</td>
<td>00000198</td>
<td>Dwond</td>
<td>000A3000</td>
</tr>
<tr>
<td>Exception Size</td>
<td>0000019C</td>
<td>Dwond</td>
<td>00012DFD</td>
</tr>
<tr>
<td>Security Rva</td>
<td>000001A0</td>
<td>Dwond</td>
<td>0000372400</td>
</tr>
<tr>
<td>Security Size</td>
<td>000001A4</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Relocation Rva</td>
<td>000001A8</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Relocation Size</td>
<td>000001AC</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Xdhin Rva</td>
<td>000001B0</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Xdhin Size</td>
<td>000001B4</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Architecture Rva</td>
<td>000001B8</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Architecture Size</td>
<td>000001BC</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Reserved</td>
<td>000001C0</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Reserved</td>
<td>000001C4</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>TLS Rva</td>
<td>000001C8</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>TLS Size</td>
<td>000001CC</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
<tr>
<td>Configuration Rva</td>
<td>000001D0</td>
<td>Dwond</td>
<td>00000000</td>
</tr>
</tbody>
</table>

**RECON 2006**
Results

RECON 2006
Bug in the Fix

The fix contains a bug if the size is:
0xXX00

For example:
0x0100 becomes 0x01FF

RECON 2006
A Better Fix

A better fix can be done without extending a section.

Changing only 2 bytes
A Better Fix

05BF80    ADD    R1, R0, #0xFE ; 0xFD
05BF84    MOV    R2, R1, LSL#16
05BF88    MOV    R0, R2, LSR#16
05BF8C    MOV    R1, R0, LSR#8
05BF90    ADD    R0, R3, #0xFE ; 0xFD
05BF94    STRB   R1, [ R4]
05BF98    AND    R2, R1, #0xFF
05BF9C    STRB   R0, [ R4, #1]
A Better Fix

On the file patch:

Offset 0x4B380: 0xFE -> 0xFD
Offset 0x4B390: 0xFE -> 0xFD
Bonus Patch

Exif header contains various fields:

Date/time
Camera Vendor
Camera Model
Artist Name aka your name in every image

RECON 2006
Bonus Patch

- Artist name comes from the Owner name stored on the phone.
- Camera.exe accesses name through the registry.
- From IDA string list: "ControlPanel\Owner\Owner"
Bonus Patch

Within the disassembly:

000481AC  LDR  R0, =aControlpanelOw
000481B0  MOV  R3, #0x80000001
000481B4  MOV  R2, #0x280
000481B8  MOV  R1, R5
000481BC  BL  get_reg_key2
000481C0  CMP  R0, #0
000481C4  BEQ  done_with_regkeys
Bonus Patch

get_reg_keys2 – returns registry value or Null if not found

Change:

000481C0  00 00 50  E3    CMP  R0,   #0
To
000481C0  00 00 50  E1    CMP  R0,  R0;  always equal

000481C4  BEQ  done_with_regkeys

RECON 2006
Bonus Patch

On the file patch:

Offset 0x0375C3: 0xE3 -> 0xE1
Bonus Patch

RECON 2006
Future of 3\textsuperscript{rd} party patches

- Continued release of 3\textsuperscript{rd} party security patches
- Advisories released with binary fix diffs
- Vulnerability market consisting of both 0day exploit and 0day patches
Questions ?
Shameless Self-Promotion

Automating Exploit Detection: Cutting-edge Tools and Techniques
Matt Hargett & Luis Miras
Blackhat Training USA

Bridging the Gap between Static & Dynamic Reversing
Defcon 14

RECON 2006