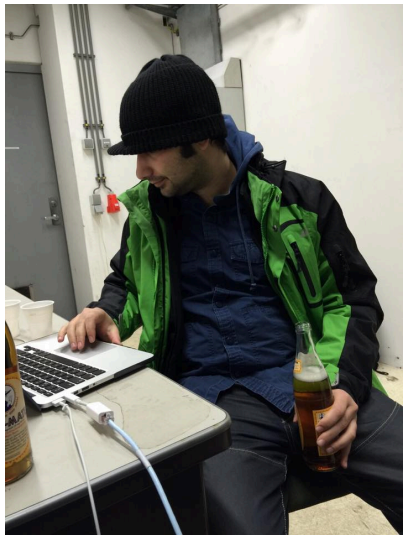
iOS Hacking: Advanced Pentest & Forensic Techniques

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 Explotation, CTF, Rfid, SDR

kpn

Outline

- 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

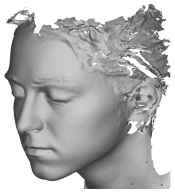
kpn

- Questions ?

Motivations

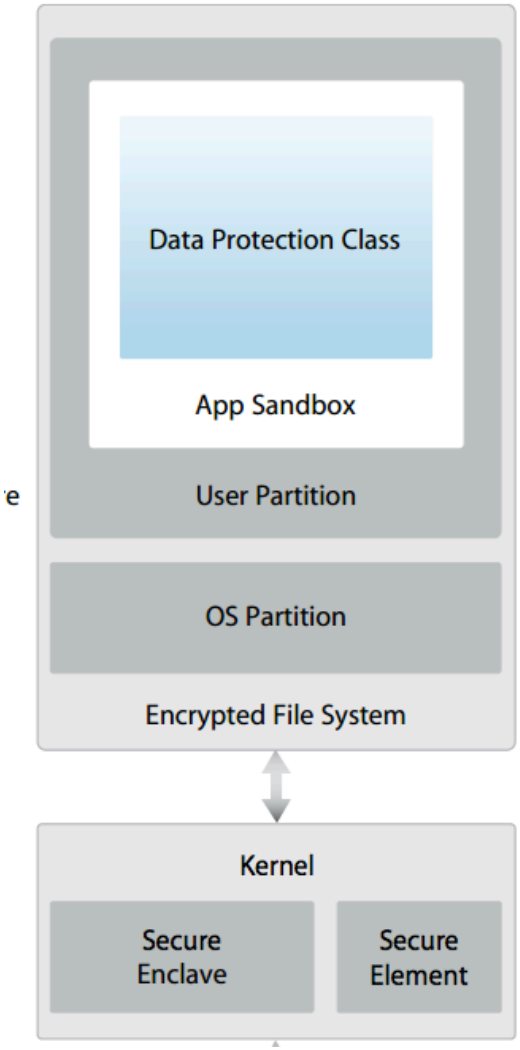
- Analyze existing security mechanism on iOS platform and circumvention techniques
- Automate and speed up mobile penetration tests
- Surveillance implants shifted focus to mobile devices
- Mobile applications are evolving and tied to monetary: iOS Mobile Payments, Paypal SDK etc.
- iOS Rootkits are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

kpn



iOS Security Architecture

5

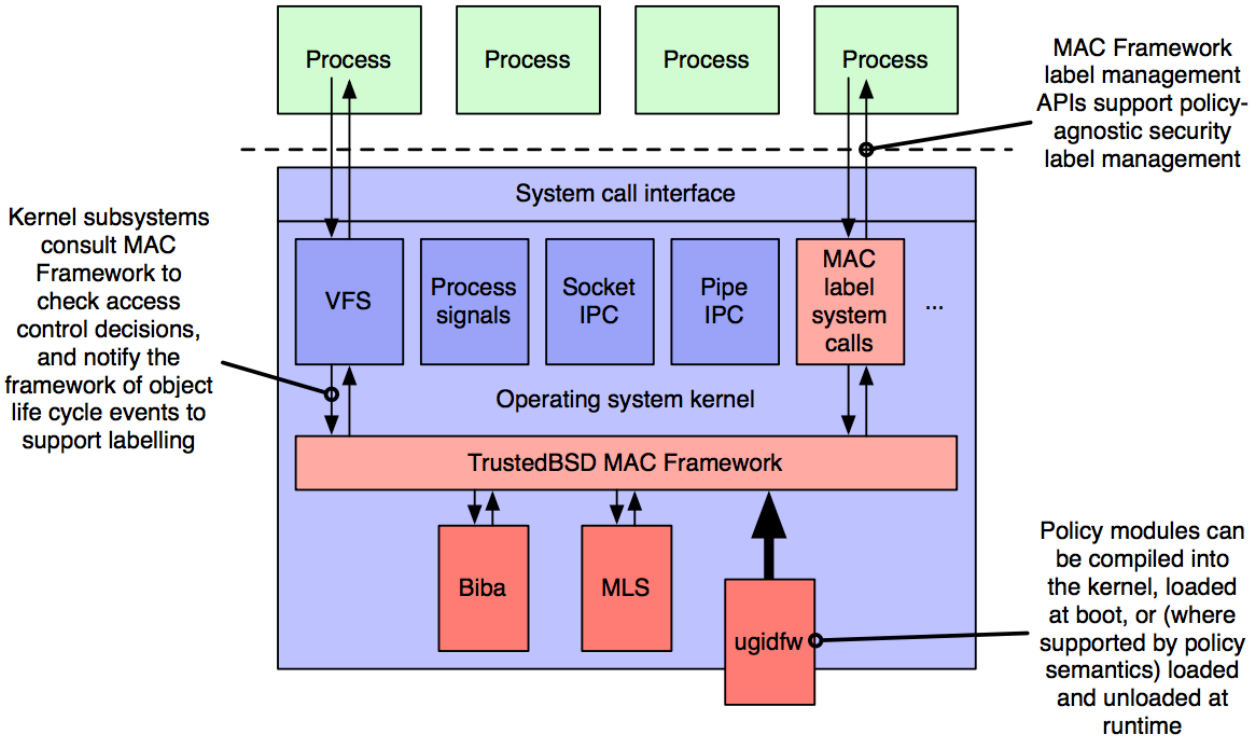


- Every app on iOS requires signing information
- Signature information within LC_CODE_SIGNATURE
- SHA1 signature verification (memory pages)

- 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*

kpn

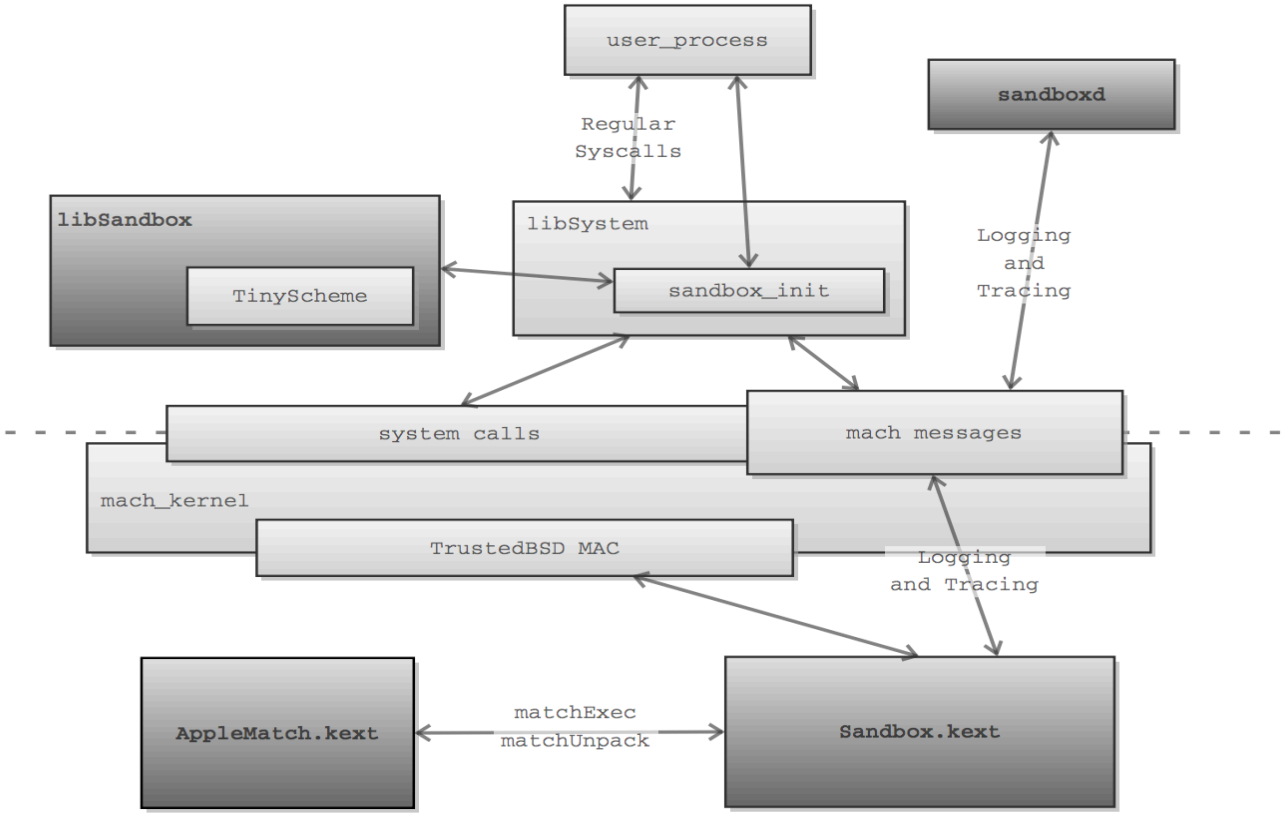
iOS Security Architecture



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

kpn

How does iOS SandBox Work?



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

kpn

How does iOS SandBox Work?

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

kpn

How does iOS SandBox Work?

iOS Sandbox Profiles (Documented)
kSBXProfileNoInternet
kSBXProfileNoNetwork
kSBXProfileNoWrite
kSBXProfileNoWriteExceptTemporary
kSBXProfilePureComputation

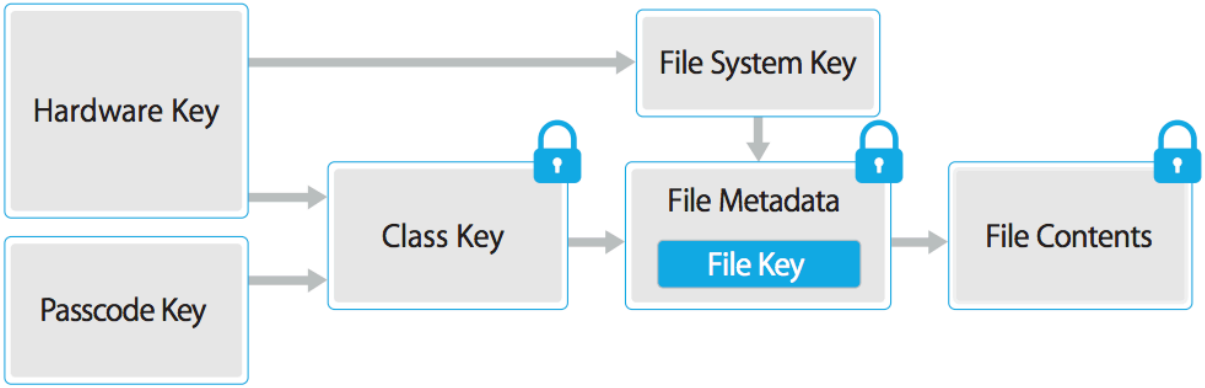
iOS Sandbox Profiles (Undocumented)
sandbox-compilerd
mDNSResponder
apsd
AppleDiags
PasteBoard
Container
MobileSafari
MobileMail
MobileMaps

Sample SandBox Usage:

```
#include <sandbox.h>

char* errbuf;
int errcode = sandbox_init("profile", SANDBOX_NAMED, &errbuf);
```

kpn

iOS : File System Encryption



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

kpn

iOS : File System Encryption (cont'd)

File API Class
NsFileProtectionNone
NsFileProtectionComplete
NsFileProtectionComplete UnlessOpen
NsFileProtectionComplete UntilFirstUserAuthentication

Security Attributes
kSecAttrAccessibleWhenUnlocked
kSecAttrAccessibleAfterFirstUnlock
kSecAttrAccessibleAlways
kSecAttrAccessibleWhenUnlocked ThisDeviceOnly
kSecAttrAccessibleAfterFirstUnlock ThisDeviceOnly
kSecAttrAccessibleAlwaysThisDeviceOnly

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]];
```

kpn

iOS : File System Encryption (cont'd)

Escrow KeyBag Location

/private/var/db/lockdown/

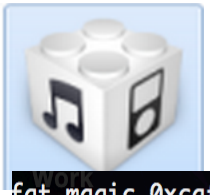
iTunes Backup Location

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

```
spammeanddie@PentestBox ~> ls /private/var/db/lockdown/
08585324a881a384dad1d491545a3c9302c198f8.plist b6b4afc0a96f6a0fea582298f6ea4a58dd5fc46d.plist
SystemConfiguration.plist                      da62180710753a1f587d8f78395e7f55da0fcb2b.plist
a7c67207b5335d821e4e6e8213ffbd5ca1e41f96.plist ec37cc3779c8925f34046ec88db234f36203f86d.plist
acf6e206d35fb8154845701591e4a8a401c889ad.plist
spammeanddie@PentestBox ~> ls ~/Library/Application\ Support/MobileSync/Backup/
08585324a881a384dad1d491545a3c9302c198f8 acf6e206d35fb8154845701591e4a8a401c889ad
a7c67207b5335d821e4e6e8213ffbd5ca1e41f96 da62180710753a1f587d8f78395e7f55da0fcb2b
spammeanddie@PentestBox ~>
```

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

kpn



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)
[redacted]:
(__DATA,__data) section
0000000010017a428        00000000 00000000 00109faa 00000001
0000000010017a438        00000000 00000000 00136b30 00000001
0000000010017a448        00000000 00000000 00136d00 00000001
0000000010017a458        00000000 00000000 00136d20 00000001
0000000010017a468        00000050 00000000 00136d68 00000001
0000000010017a478        00000000 00000000 00109f9d 00000001
0000000010017a488        00136e08 00000001 00136e20 00000001
0000000010017a498        00000000 00000000 00136e40 00000001
0000000010017a4a8        00000000 00000000 00136e90 00000001
0000000010017a4b8        00000050 00000000 00136ec8 00000001
0000000010017a4c8        00000000 00000000 00109ffa 00000001
0000000010017a4d8        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; };

https://developer.apple.com/library/mac/documentation/DeveloperTools/Conceptual/MachORuntime/index.html

kpn

Decrypting Binaries (32-bit)

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

kpn

Decrypting Binaries (64-bit)

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

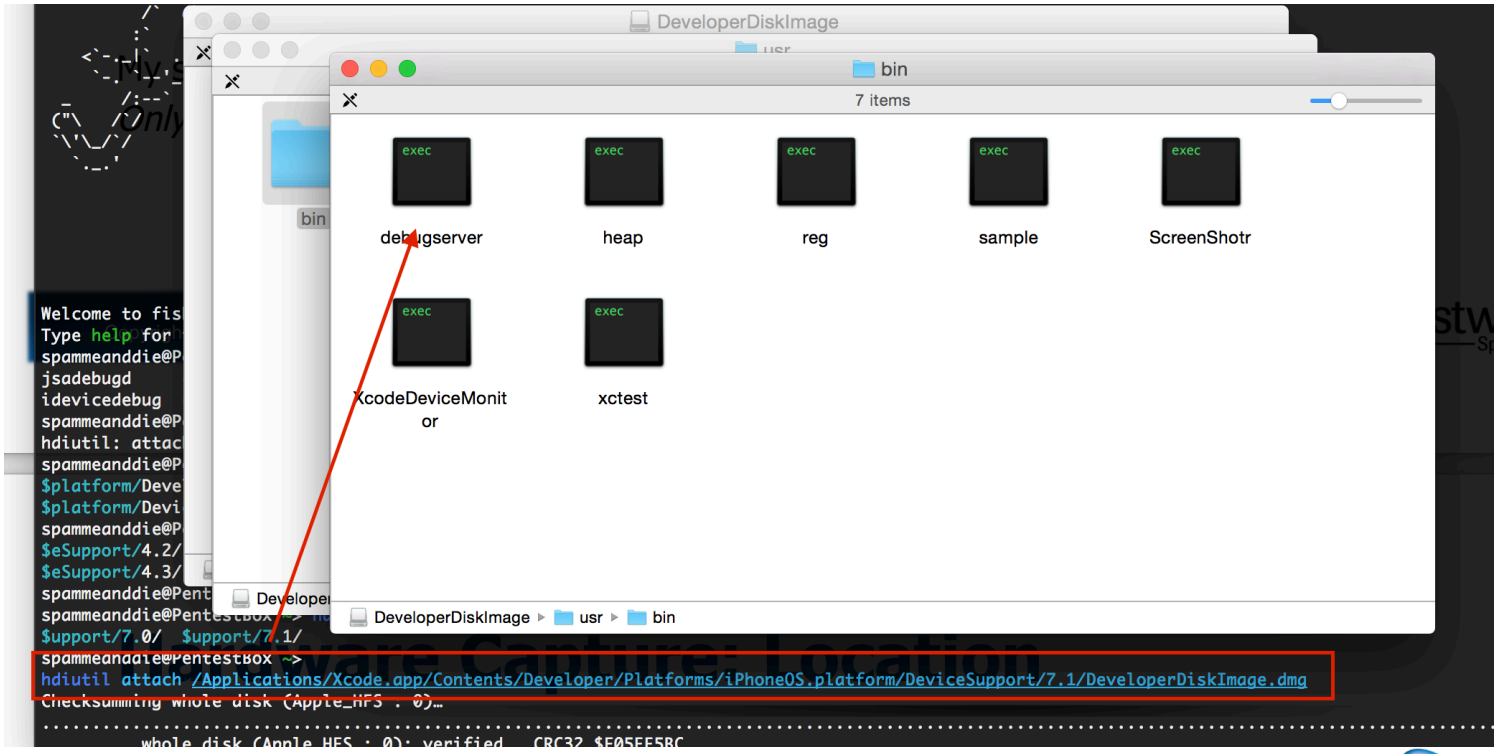
Remote debugging : Running debugserver on iOS – running LLDB on Mac

kpn

Getting the Debugger running

All you need are stored under the Xcode IDE directories

Obtain the debug server binary



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

Getting the Debugger running

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

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

Stopped thread list available if debuggger 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'
  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'
```

kpn

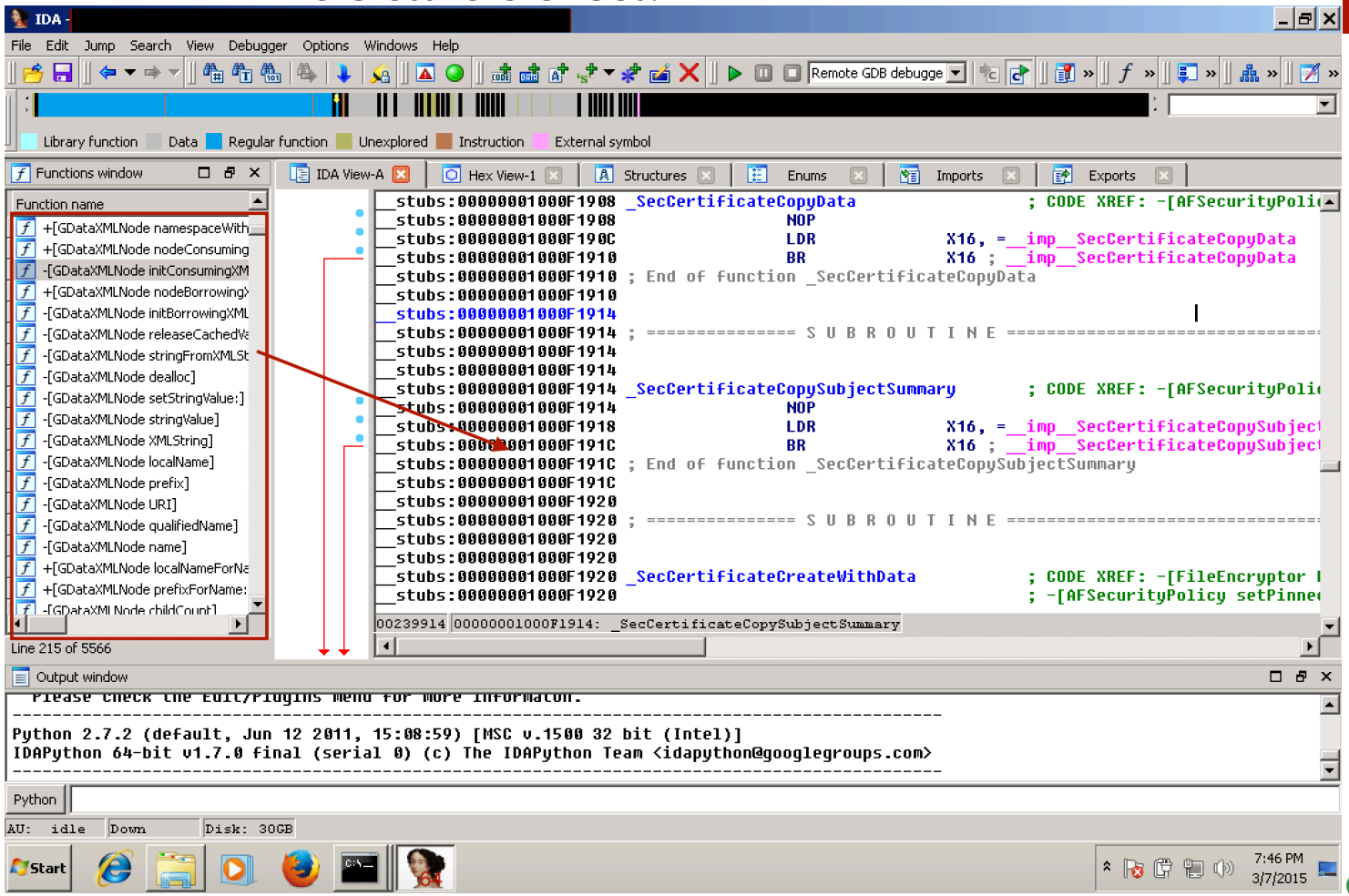
Reversing iOS Apps

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

20

Reversing iOS Apps: Sainte Ida de Louvain

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



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

kpn

Reversing iOS Apps: Dealing with Crpyto

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

22

Functions window | IDA View-A | Imports | Hex View-1 | Structures

Function name
-[ASIAuthenticationDialog show]
-[ASIAuthenticationDialog cancelAuthentica
-[ASIAuthenticationDialog requestsRequirir
-[ASIAuthenticationDialog presentNextDial
-[ASIAuthenticationDialog loginWithCreden
-[ASIAuthenticationDialog numberOfSectio
-[ASIAuthenticationDialog tableView:height
-[ASIAuthenticationDialog tableView:height
-[ASIAuthenticationDialog tableView:titleFc
-[ASIAuthenticationDialog tableView:cellFo
-[ASIAuthenticationDialog tableView:numbe
-[ASIAuthenticationDialog tableView:titleFc
-[ASIAuthenticationDialog request]
-[ASIAuthenticationDialog setRequest:]
-[ASIAuthenticationDialog type]
-[ASIAuthenticationDialog setType:]
-[ASIAuthenticationDialog tableView]
-[ASIAuthenticationDialog setTableView:]
-[ASIAuthenticationDialog didEnableRotatic

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...		_SSLWrite	/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...

Line 215 of 5566 | Line 356 of 673

Output window

kpn

Reversing iOS Apps: Dealing with Crypto

It seems the application evaluates the certificate here.

```
00F195C _SecTrustCopyPublicKey                  ; CODE XREF: -[FileEncryptor RSAEncryptData:withDERpublicKey:]+B4↑p
00F195C                                         ; -[AFSecurityPolicy setPinnedCertificates:]+1E4↑p ...
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
00F1968 _SecTrustCreateWithCertificates         ; CODE XREF: -[FileEncryptor RSAEncryptData:withDERpublicKey:]+54↑p
00F1968                                         ; -[AFSecurityPolicy setPinnedCertificates:]+1CC↑p ...
00F1968                 NOP
```

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

kpn

Reversing iOS Apps: Dealing with Crypto

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

```
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:000000010014D350↓o
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 0000000010005CE00: -[FileEncryptor RSAEncryptData:withDERpublicKey:]
```

Calling Convention : C++	Calling Convention : Objective C
ObjectPointer->Function(parameters)	[ObjectPointer Function:parameters]

kpn

Reversing iOS Apps: Hunting for Public Key

The following function evaluates the certificate .

```
00010005CEA0
00010005CEA0 evaluate_certificate                ; CODE XREF: -[FileEncryptor RSAEncryptData:withDERp
00010005CEA0                    LDR              X0, [SP,#0x70+var_48]
00010005CEA4                    ADD              X1, SP, #0x70+var_50+4
00010005CEA8                    BL               _SecTrustEvaluate
00010005CEAC                    CBNZ             W0, loc_10005CE78
00010005CEB0                    LDR              X0, [SP,#0x70+var_48]
00010005CEB4                    BL               _SecTrustCopyPublicKey
00010005CEB8                    MOV              X21, X0
00010005CEBC                    CBZ              X21, certificate_copy
00010005CEC0                    MOV              X0, X21
00010005CEC4                    BL               _SecKeyGetBlockSize
00010005CEC8                    MOV              X20, X0
00010005CECC                    MOV              X0, X21
00010005CED0                    BL               _SecKeyGetBlockSize
00010005CED4                    STR              X0, [SP,#0x70+var_58]
00010005CED8                    ADRP             X8, #selRef_length@PAGE
00010005CEDC                    NOP
00010005CEE0                    LDR              X22, [X8,#selRef_length@PAGEOFF]
```

Check the function prototypes and the definition on Apple Dev.

SecTrustEvaluate

Evaluates trust for the specified certificate and policies.

Declaration

SWIFT

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

OBJECTIVE-C

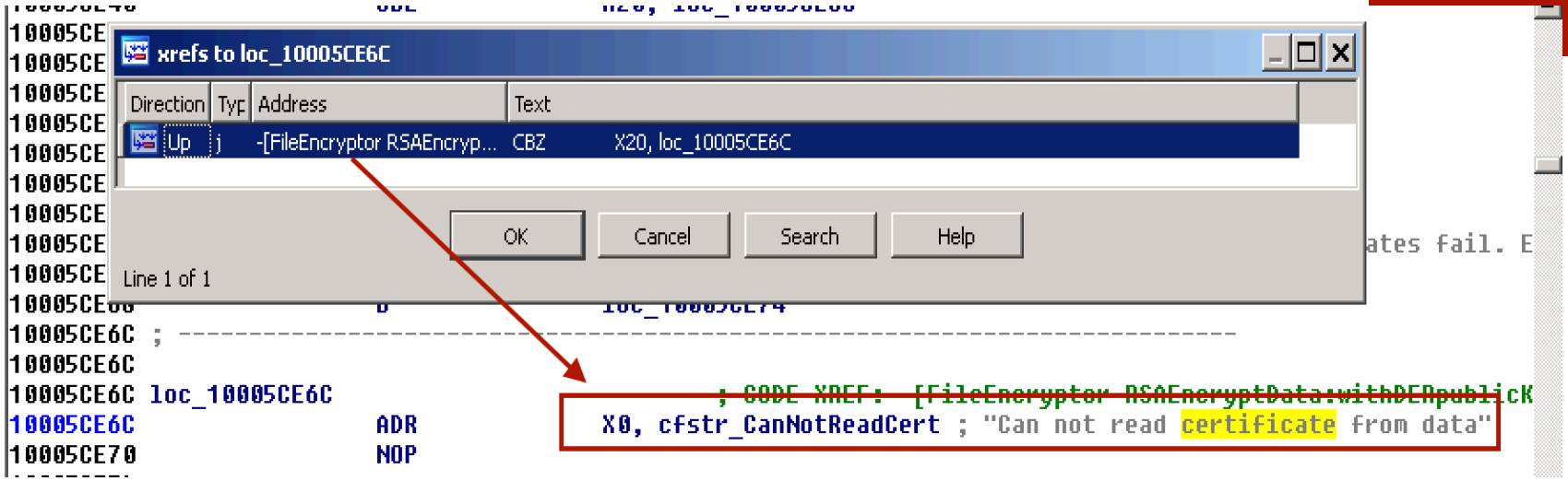
```
OSStatus SecTrustEvaluate ( SecTrustRef trust, SecTrustResultType *result );
```

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

kpn

Reversing iOS Apps: Hunting for Public Key

Cross-references definitely help.



So do the constants and the debug strings. ☺

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

kpn

Reversing iOS Apps: Hunting for Public Key

Preparation for file encryption is literally being done here.

27

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

kpn

Reversing iOS Apps: Hunting for Public Key

28

Short cheat sheet on LLDB for GDB junkies.

GDB Command	LLDB Command
(gdb) dump memory /tmp/mem.bin 0x1000 0x2000	**(lldb)** memory read --outfile /tmp/ mem.bin --binary 0x1000 0x2000
(gdb) disassemble	**(lldb)** disassemble --frame **(lldb)** di -f
(gdb) x/20i 0x1eb8	**(lldb)** disassemble --start-address 0x1eb8 --count 20
(gdb) info shared	**(lldb)** image list

kpn

Reversing iOS Apps: Hunting for Public Key

Preparation for file encryption is literally being done here.

29

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

kpn

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

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

Function name

-[FileEncryptor encryptFile:]
-[FileEncryptor randomDataOf
-[FileEncryptor generateAES2
-[FileEncryptor convertPEMtoI
-[FileEncryptor RSAEncryptDa
-[FileEncryptor AESEncrypt:ke
-[FrequentnessController initV
-[FrequentnessController viev
-[FrequentnessController reve
-[FrequentnessController reve
-[FrequentnessController che
sub_10005D494
nullsub_21
sub_10005D5A4
sub_10005D5AC
sub_10005D5B4
-[FrequentnessController viev
-[FrequentnessController viev
-[FrequentnessController didF
-[FrequentnessController viev
-[FrequentnessController logc
-[FrequentnessController setL

```
68C ; =============== S U B R O U T I N E =======================================
68C
68C ; FileEncryptor - (id)encryptFile:(id)
68C
68C ; id __cdecl -[FileEncryptor encryptFile:](struct FileEncryptor *self, SEL, id)
68C __FileEncryptor_encryptFile__           ; DATA XREF: __objc_const:0000000100014D2F0↓o
68C
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
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
```

kpn

How to Reversing on iOS Env?

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

kpn

Reversing iOS Apps: Hunting for Public Key

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 CERTI
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  ECBMDTlJXMQ8wDQY
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  GRlcjEuMCwGCSqGS
000100   49 62 33 44 51 45 4A 41  52 59 66 61 6D 56 68 62  Ib3DQEJARYfamVhb
```

kpn

iOS Apps Penetration Testing

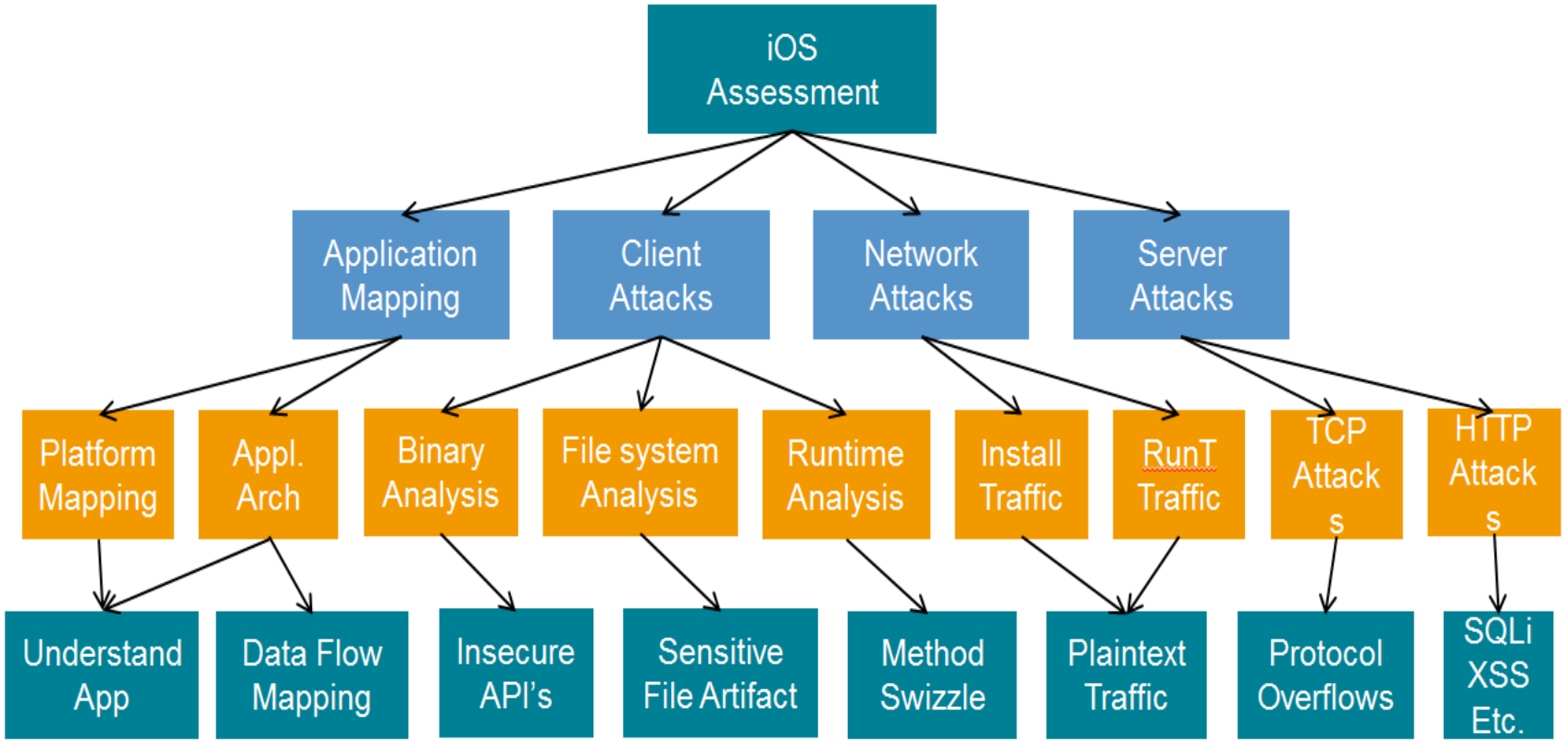
OWASP Mobile Top 10 Risks

- 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

kpn

iOS Apps Penetration Testing



https://www.owasp.org/index.php/IOS_Application_Security_Testing_Cheat_Sheet

kpn

iOS Apps Penetration Testing: Network Traffic Analysis


```
ssh ~
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 by
tes
```

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	Lengtl	Info
64	13.37149300(	192.168.2.100	192.168.2.101	TCP	66	us-cli > 49511 [ACK] Seq=201 Ack
65	13.37186600(	192.168.2.100	192.168.2.101	TCP	66	us-cli > 49511 [ACK] Seq=201 Ack
66	14.86451800(	192.168.2.100	192.168.2.101	TLSv1.2	1175	Application Data
67	14.86455000(	192.168.2.100	192.168.2.101	TLSv1.2	119	Encrypted Alert
68	14.86488000(	192.168.2.101	192.168.2.100	TCP	66	49511 > us-cli [ACK] Seq=777 Ack
69	14.86492700(	192.168.2.101	192.168.2.100	TCP	66	49511 > us-cli [ACK] Seq=777 Ack
70	14.86531300(	192.168.2.101	192.168.2.100	TLSv1.2	119	Encrypted Alert
71	14.86748300(	192.168.2.100	192.168.2.101	TCP	66	[TCP Out-Of-Order] us-cli > 4951

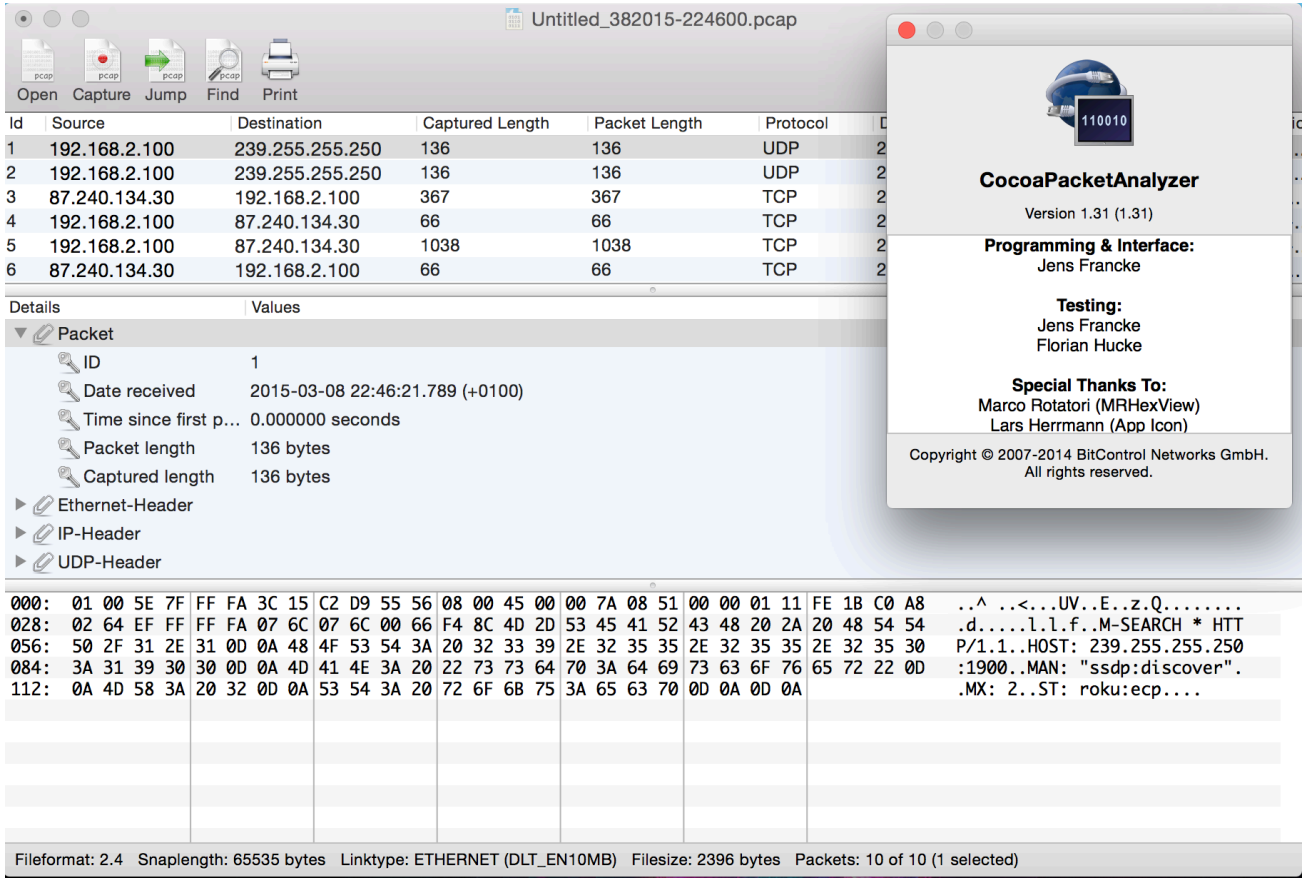
Standard input: <live capture in pro… Packets: 78 · Displayed: 78 (1… Profile: Default

https://www.wireshark.org/

kpn

iOS Apps Penetration Testing: Network Traffic Analysis

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



Cacoa Packet Analyzer:

www.tastycocoabytes.com/cpa/

kpn

SSL Interception: Function Hooks

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

context	An SSL session context reference.
data	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 `dataLength` parameter.
dataLength	The amount of data you would like to read.
processed	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

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

context	An SSL session context reference.
data	A pointer to the buffer of data to write.
dataLength	The amount, in bytes, of data to write.
processed	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 envrionment looks like ?

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

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

kpn

SSL Interception: Function Hooks

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*          PreferenceLoader.plist  libstatusbar.dylib*
ActionMenu.plist   DeviceInfoInit.dylib*  Insomnia.plist           RocketBootstrap.dylib@  libstatusbar.plist
Activator.dylib@   DeviceInfoInit.plist   MobileSafety.dylib*      RocketBootstrap.plist   samplehook.dylib*
Activator.plist    Flipswitch.dylib@      MobileSafety.plist       iSpy.dylib*             samplehook.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#
```

kpn

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 …*

kpn

SSL Interception: Function Hooks


```
GNU nano 2.2.6                File: com.samplehook.ssl_logz.txt

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=beu6WenUpxsNHyqyV98l50UOwqLGm-4Gr9jLZHZIn_OBuECu4RRk76Z0GOOHX1hN7VwRYXNl187lqMvS27pmP2rN5ItGbPxCe3E5M-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$
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,$
div{outline:none;-webkit-tap-highlight-color:rgba(0,0,0,0)}.msb{position:relative}.msfo{padding-right:38px}.ms$
!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$
```

kpn

SSL Interception: Function Hooks

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

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="dj0yJmk9QzhqS1FIMHFDalNqJmQ9WVdrOVVGbDNPVWs0TmpRbWNHbzlNVEUyTXpjNE9UQTJNZy0tJnM9Y29uc3VtZX$
oauth_signature="AmHD8w8mVWEM%2Bhsj3qsl2Bm06i8%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
```

kpn

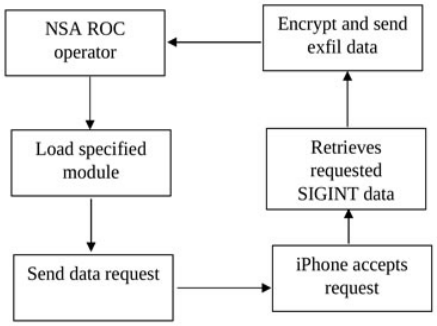
TOP SECRET//COMINT//REL TO USA, FVEY

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

TOP SECRET//COMINT//REL TO USA, FVEY

44

This is beyond the conspiracy theories: for real!

kpn

Iphone Rootkit CookBook

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/

kpn

Iphone Rootkit CookBook (cont'd)

A Sample hook for enabling iPhone Microphone.

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

    //id ct = CTTelephonyCenterGetDefault();
    //CTTelephonyCenterAddObserver(ct, NULL,callback2, NULL, NULL, CFNotificationSuspensionBehaviorHold);

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

Source Code:Tripware:

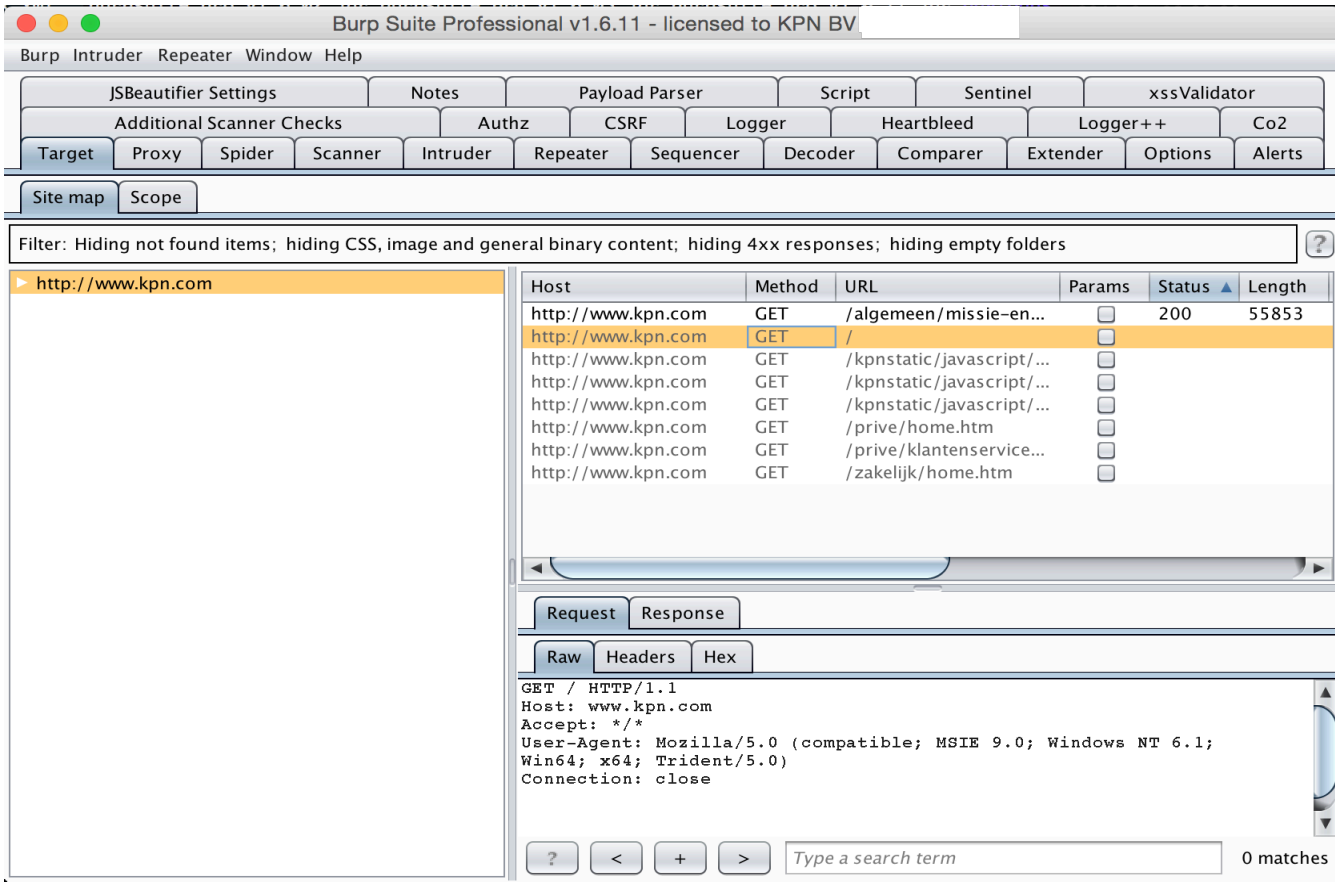
http://www.tripwire.com/state-of-security/vulnerability-management/creating-iphone-rootkits-and-like-the-nsas-dropout-jeep/

kpn

Burp Suite: Atomize Everything

47

More than standard application communication interception.

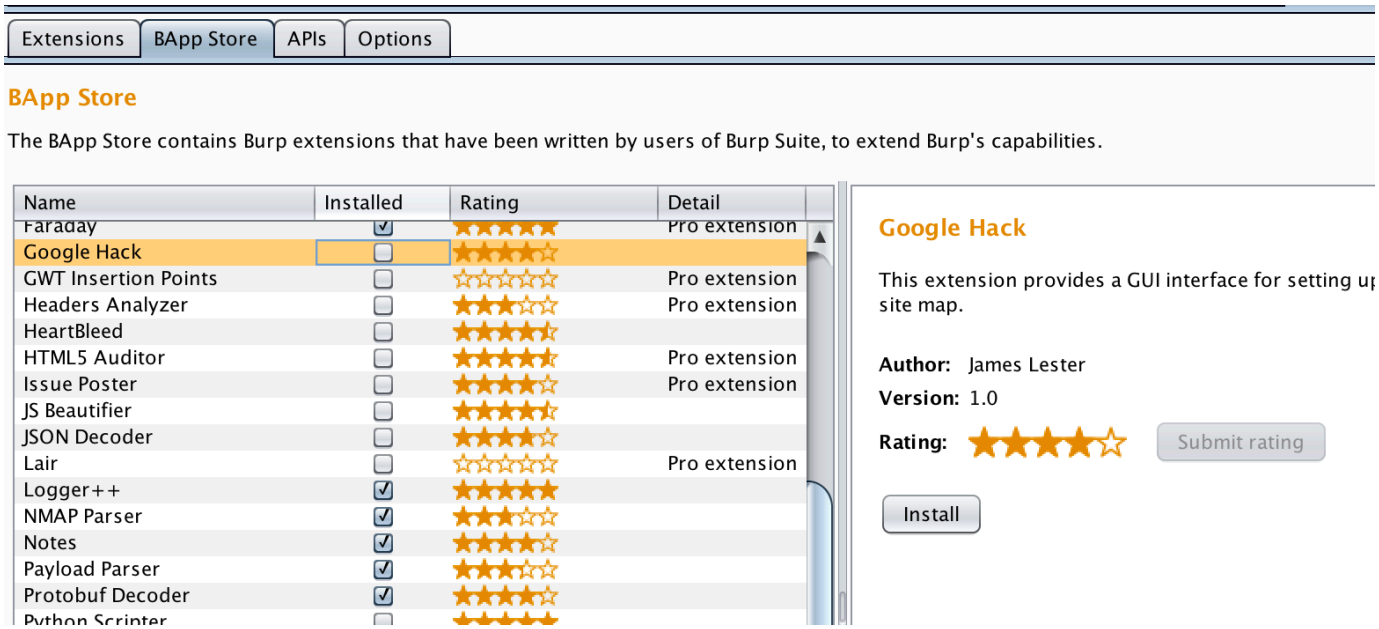


kpn

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

Burp Extensions: Installation

- 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 apllication (e.g. PhantomJS, Radamsa etc)



How Extensions Work (cont'd)

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

kpn

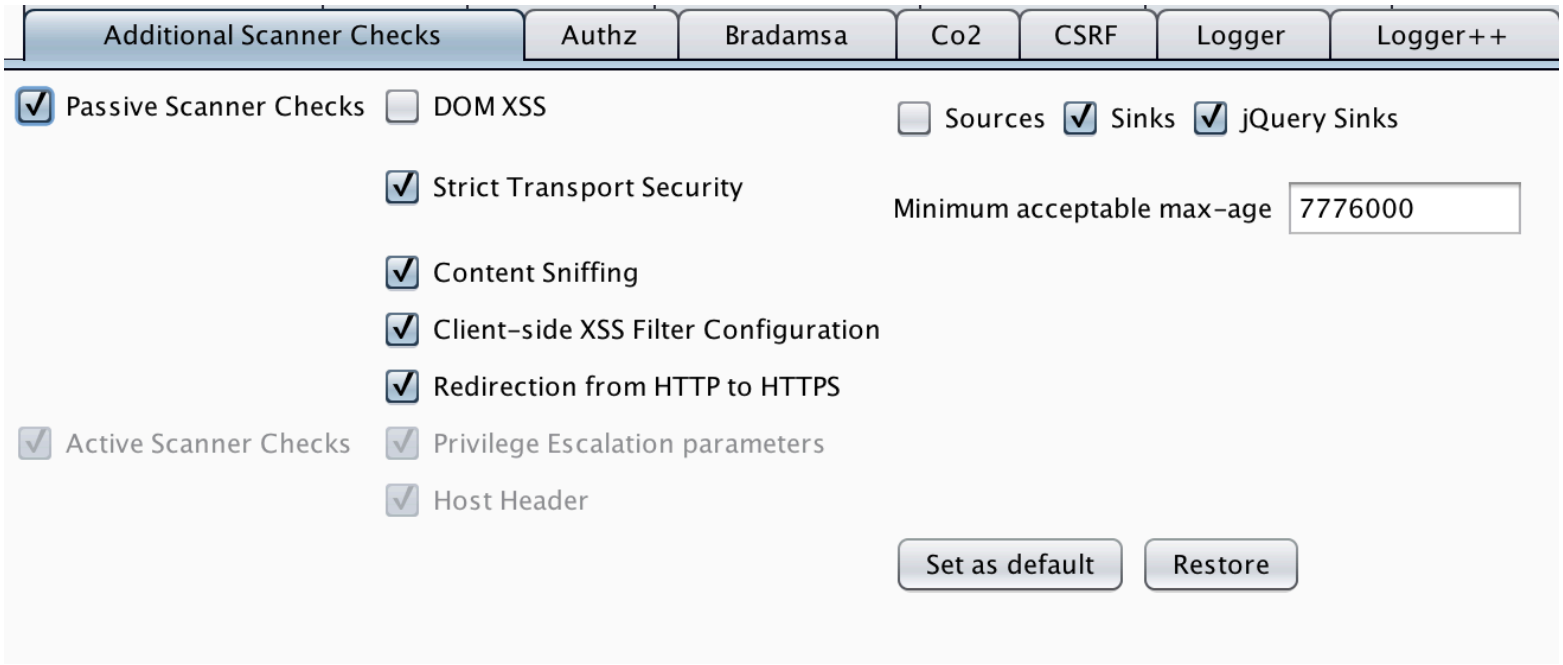
Burp Extensions in a NutShell

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

kpn

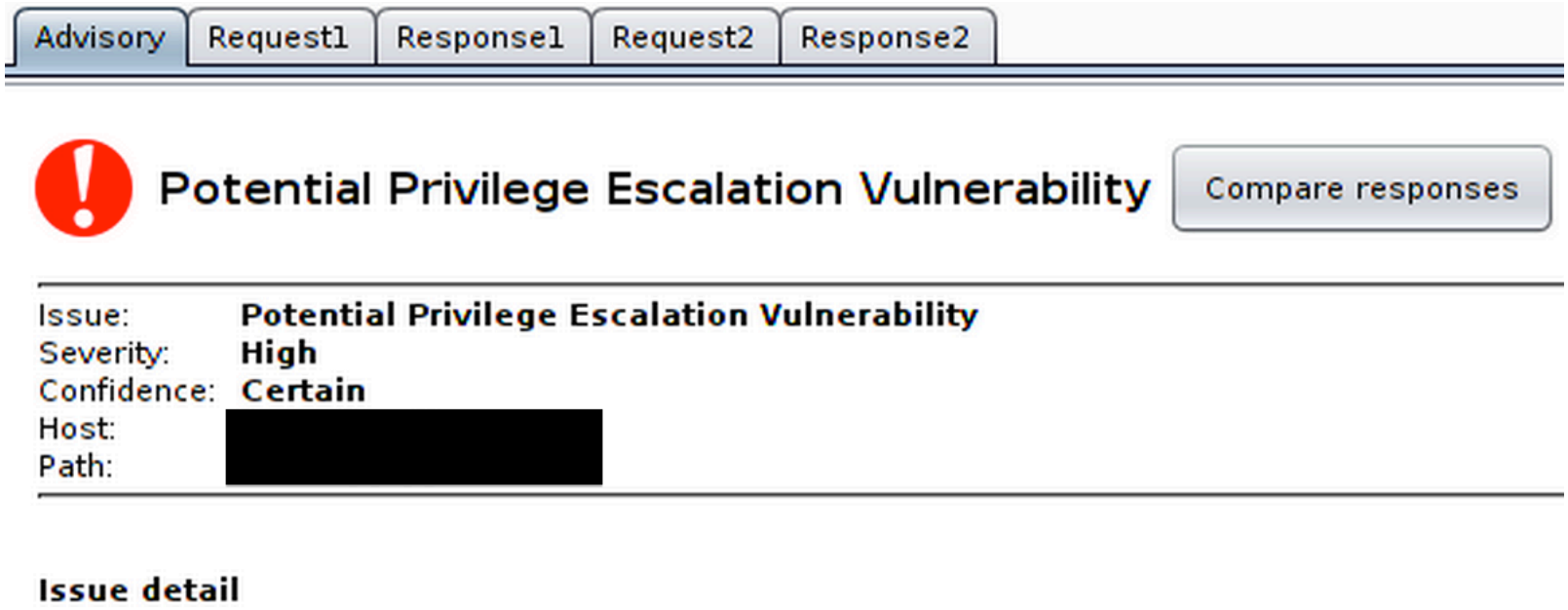
Burp Extensions : Additional Scanner Checks

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



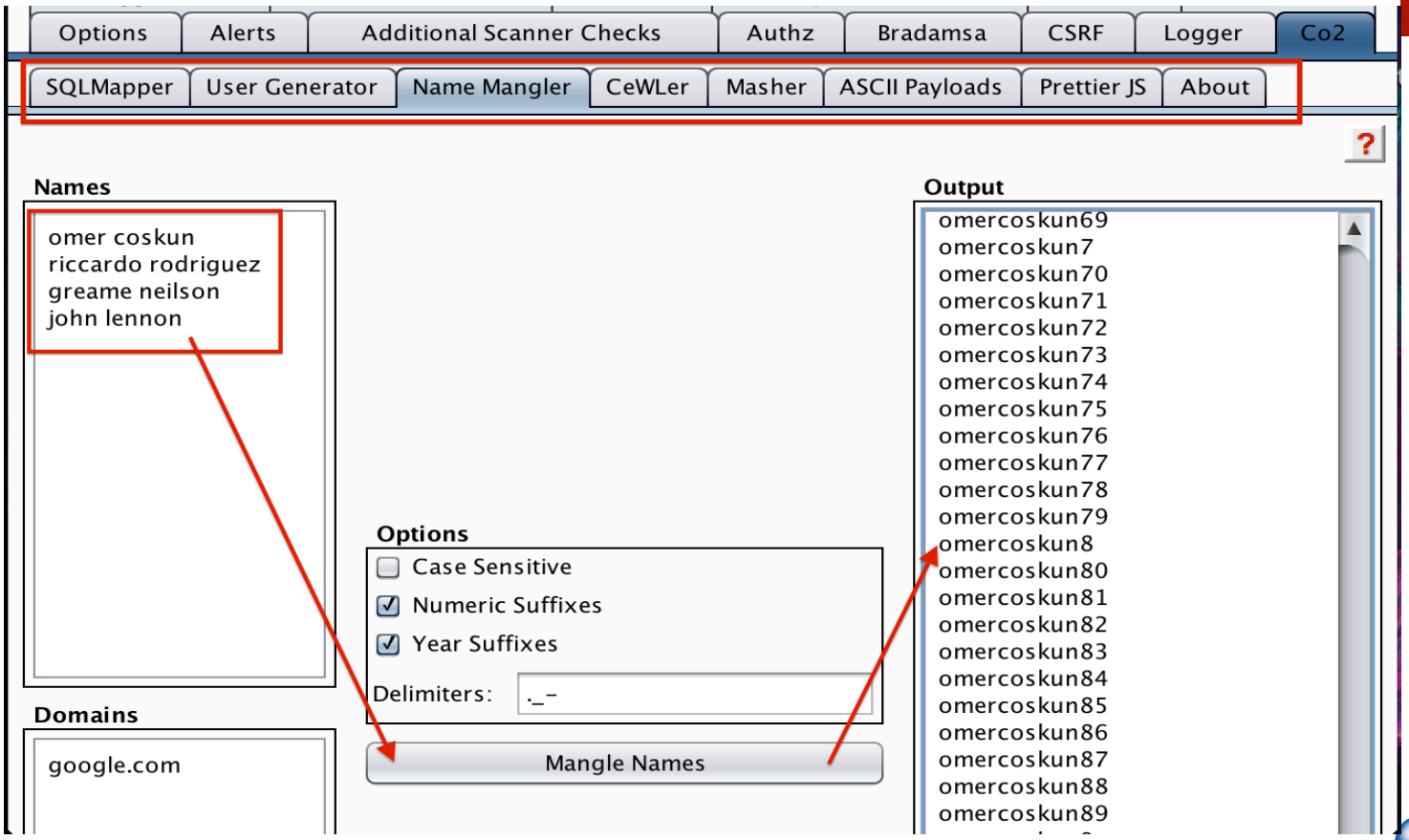
Burp Extensions : Session Auth

- To Identify authentication privilege escalation vulnerabilities.



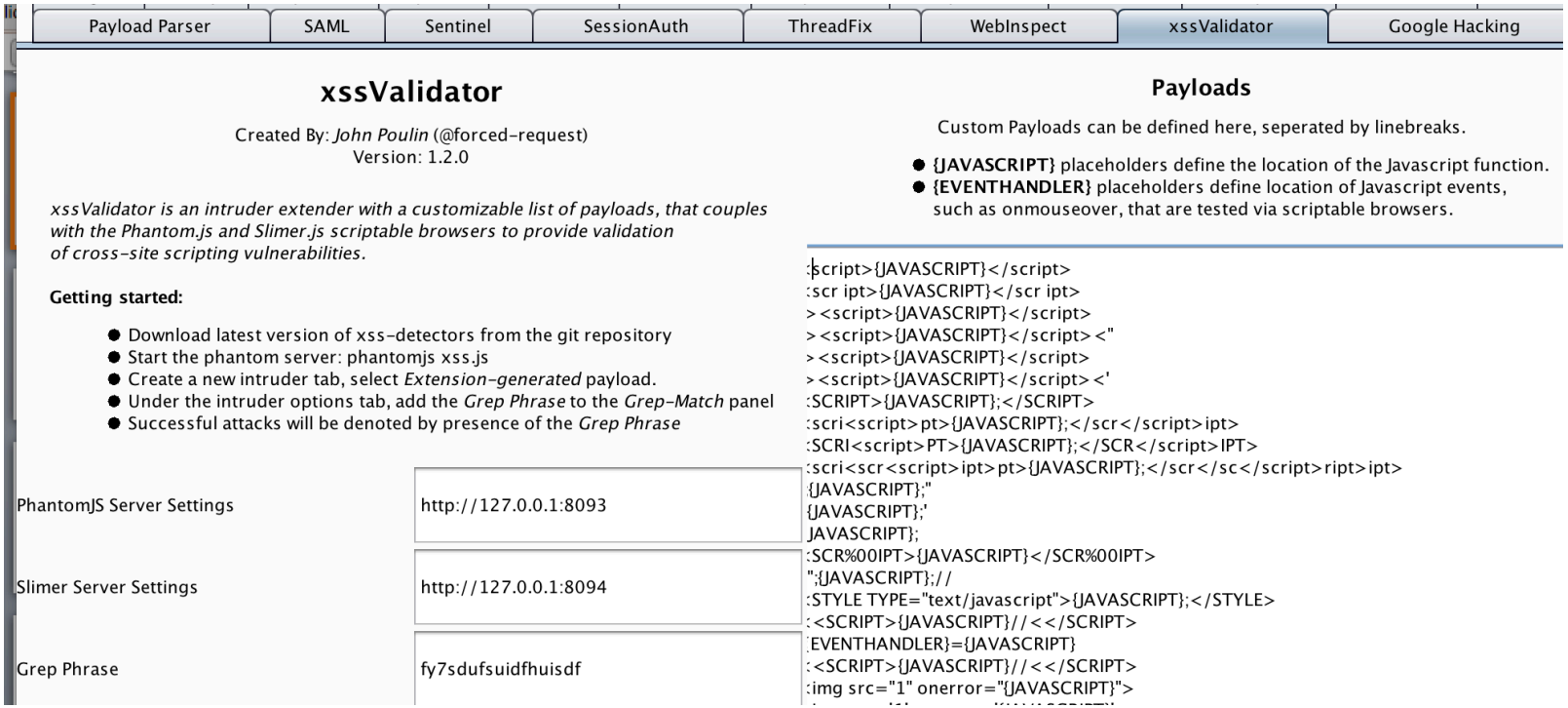
Burp Extensions : CO2

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



Fully Automated XSS Verification

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

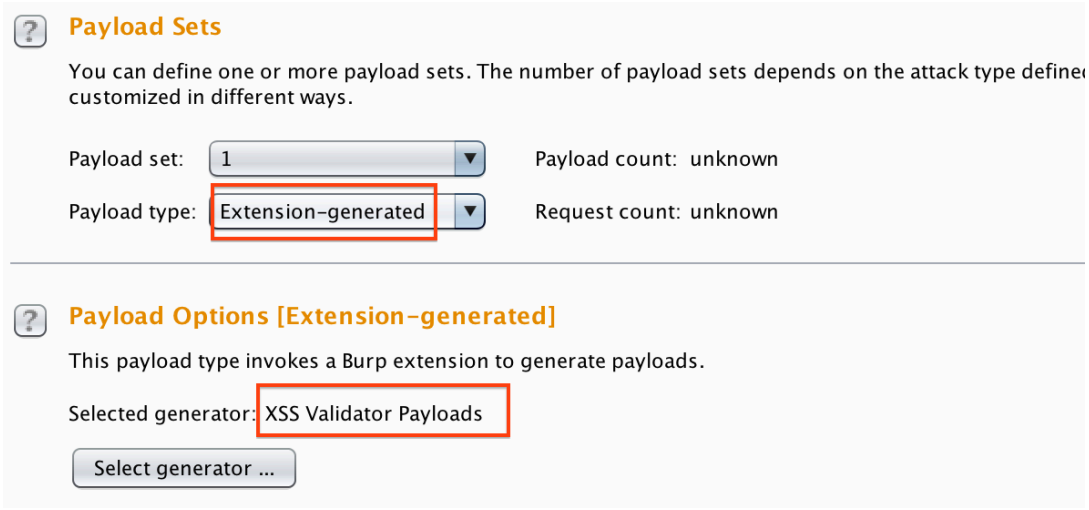


Fully Automated XSS Verification

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

```
spammeanddie@PentestBox ~/D/s/appz-algo> 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 ~/D/s/appz-algo> locate xss.js
/Users/spammeanddie/BurpSuitePro/bapps/98275a25394a417c9480f58740c1d981/xss-detector/xss.js
spammeanddie@PentestBox ~/D/s/appz-algo> phantomjs /Users/spammeanddie/BurpSuitePro/bapps/98275a25394a417c9480f58740c1d981/xss-detector/xss.js
```

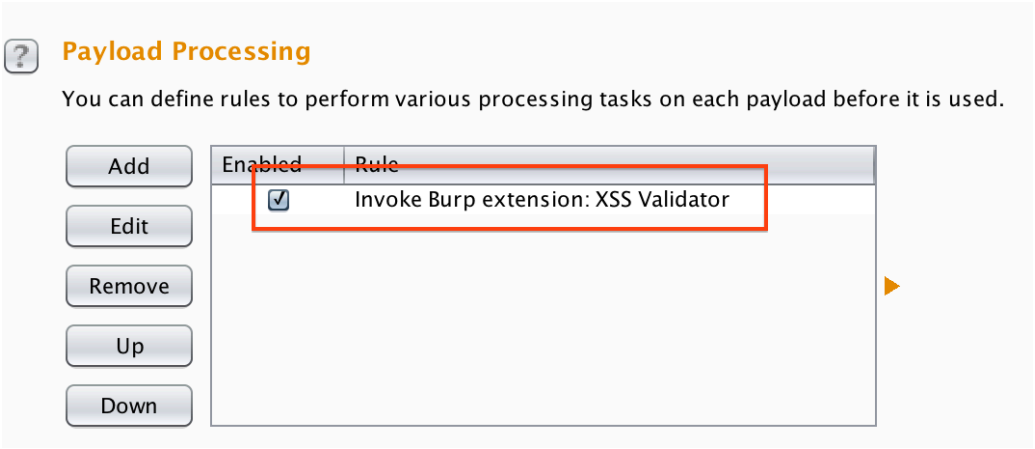
- Enable the payload extension after running wrapper.



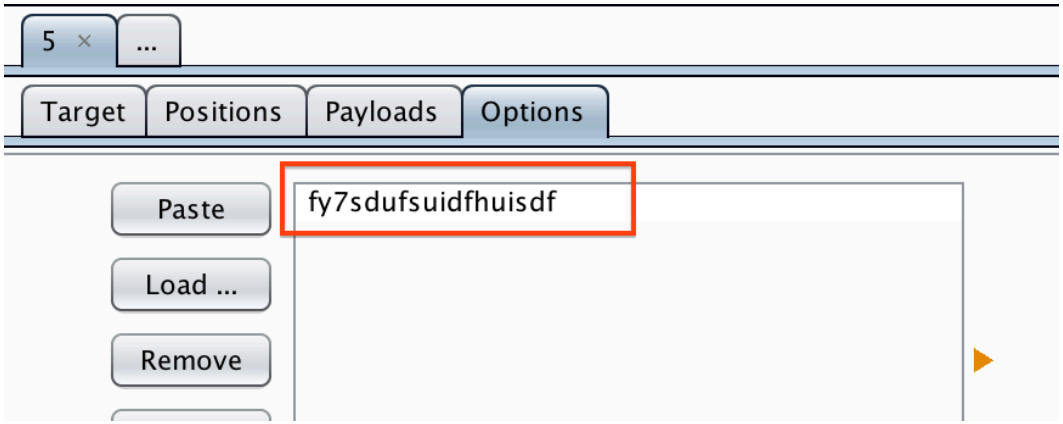
kpn

Fully Automated XSS Verification

- Enable payload processing unit for xssVerifier.



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



kpn

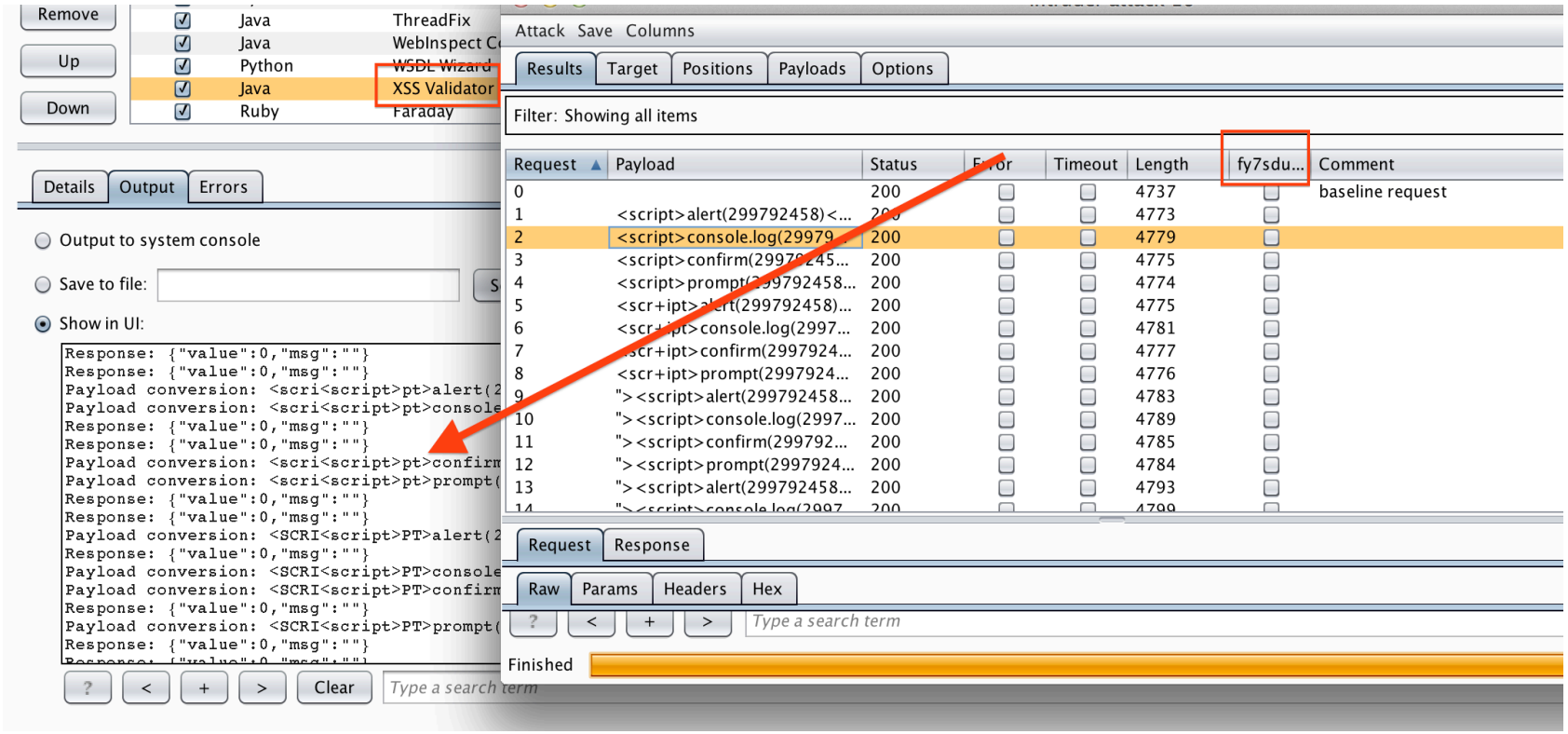
Fully Automated XSS Verification

- Content of xss.js

```
*/
reInitializeWebPage = function() {
    wp = new WebPage();
    xss = new Object();
    xss.value = 0;
    xss.msg = "";

    // web page settings necessary to adequately detect XSS
    wp.settings = {
        loadImages: true,
        localToRemoteUrlAccessEnabled: true,
        javascriptEnabled: true,
        webSecurityEnabled: false,
        XSSAuditingEnabled: false
    };

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

        xss.value = 1;
        xss.msg += 'XSS found: alert(' + msg + ')';
    };

    wp.onConsoleMessage = function(msg) {
        console.log("On console.log: " + msg);

        xss.value = 1;
        xss.msg += 'XSS found: console.log(' + msg + ')';
    };

    wp.onConfirm = function(msg) {
        console.log("On confirm: " + msg);
```

Fully Automated XSS Verification

- Let the fun begin ☺



Questions ?

Thank you very much for your attention