

Security Overview

New security capabilities in iOS 5 and Lion

Session 202

Alex Radocea

Product Security

These are confidential sessions—please refrain from streaming, blogging, or taking pictures

Agenda

- Assets and attackers
- Security capabilities
 - Existing
 - What is new

Securing What?

Data

Credentials

Privacy

Financials

Business

Resources

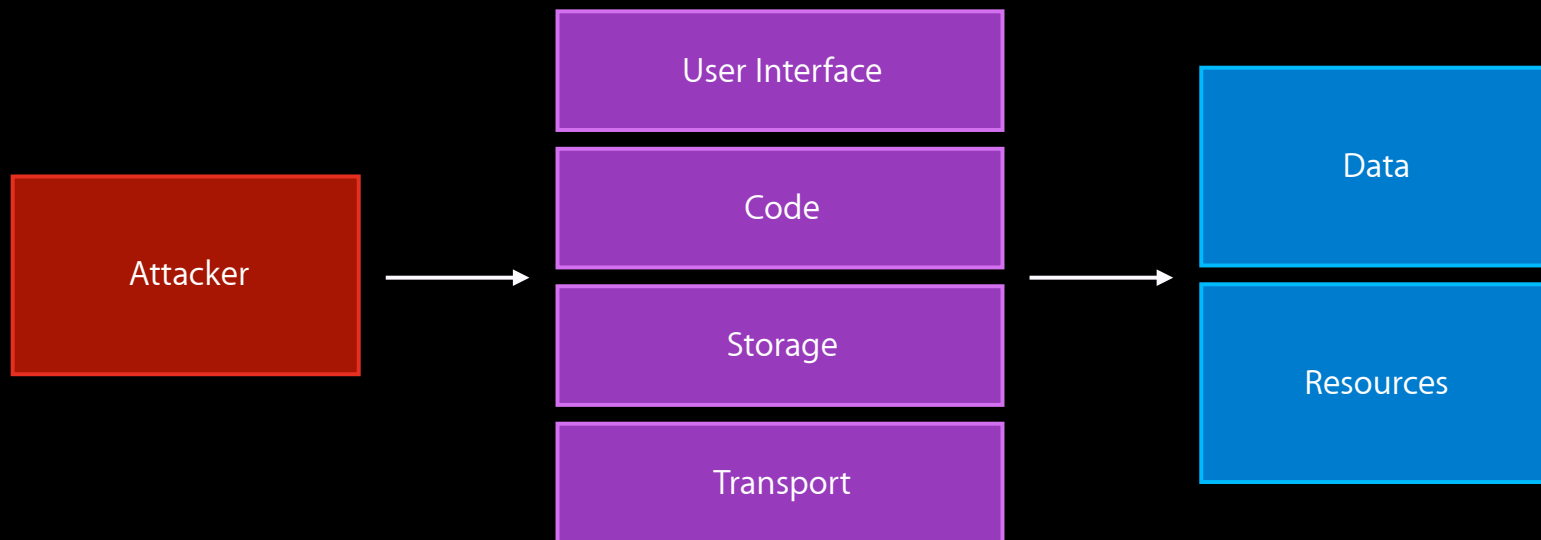
Network

Phone

Computation

Storage

How Attackers Work





Some Example Vulnerabilities

- User interface

- Trojans
- Phishing
- Spoofing

- Code

- Input validation flaws
- Memory mismanagement

- Storage

- Physical theft
- Cross application leaks
- Incorrect file permissions

- Transport

- Passive observation
- Man in the middle
- Replay attacks



Defenses

- User interface

- Quarantine
- Authorization

- Code

- Exploit mitigation
- Code hardening
- Sandboxing

- Storage

- Filesystem permissions
- Keychain
- FileVault
- Data protection

- Transport

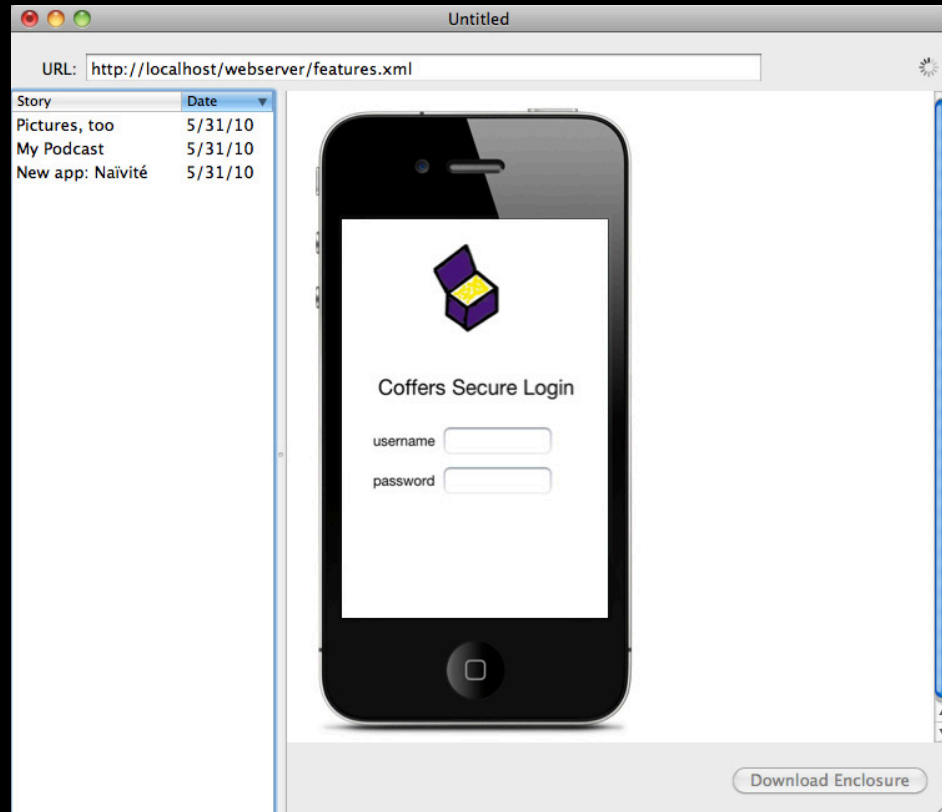
- SecurityFramework
- SecureTransport

Roadmap

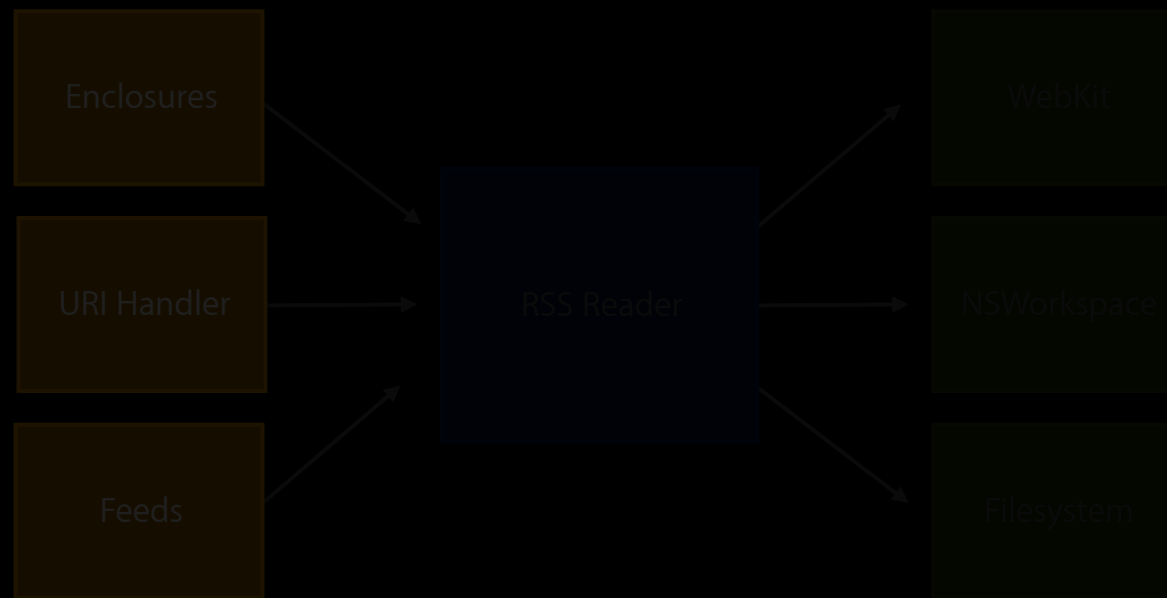
- Two applications
 - RSS Reader
 - Financial application
- Vulnerabilities and Mitigations
 - UI
 - Code
 - Storage
 - Transport

RSS Reader

- HTML5
- Rich media
- Network enabled



Attack Model for Breaking into RSS Reader



User Interface Attacks

- Trojans
- Phishing
 - Spoofing



User Interface Defenses

- Quarantine
- Authorization



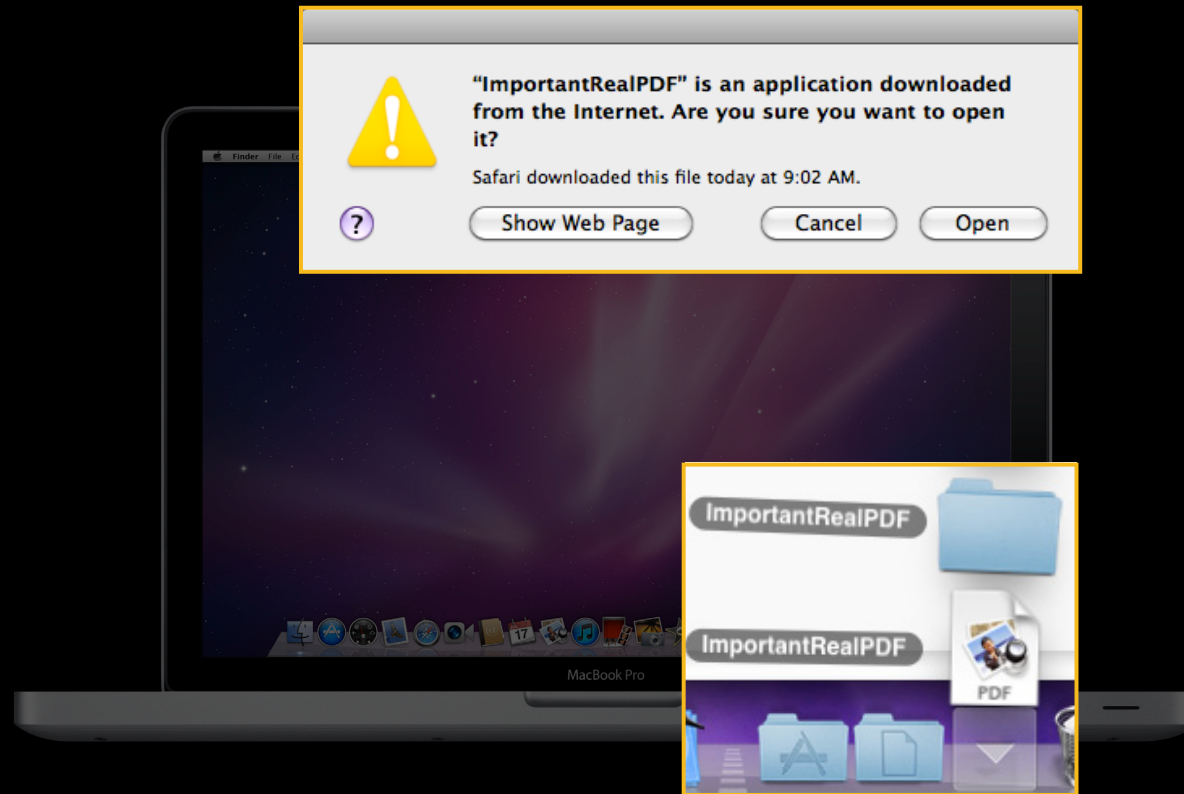
Is This Really a PDF?

Only on
Mac OS



Quarantine Stops Trojan Horses

Only on
Mac OS



Using Quarantine

Only on
Mac OS

- Info.plist key: `LSFileQuarantineEnabled`

- Automatic attributes

```
kLSQuarantineAgentNameKey;
```

```
kLSQuarantineAgentBundleIdentifierKey;
```

```
kLSQuarantineTimeStampKey;
```

- Manual attributes

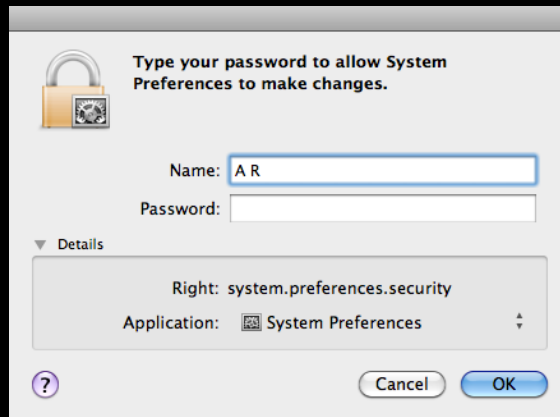
```
kLSQuarantineTypeKey;
```

```
kLSQuarantineOriginURLKey;
```

```
kLSQuarantineDataURLKey;
```


Authorization

Only on
Mac OS




Type your password to allow System Preferences to make changes.

Name:

Password:

Details

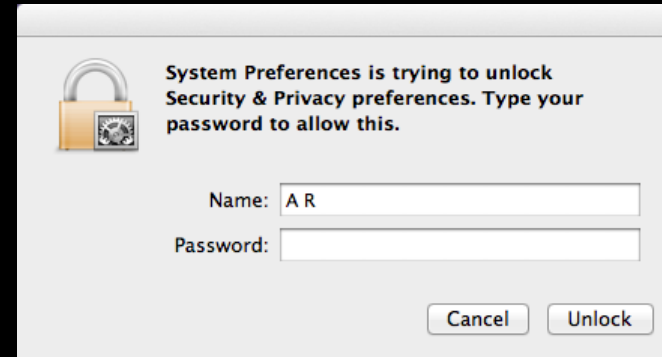
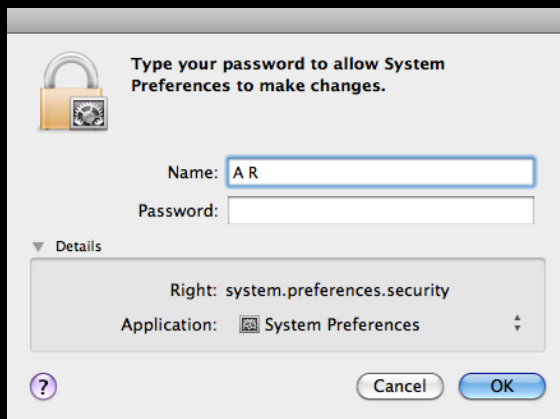
Right: system.preferences.security

Application:  System Preferences

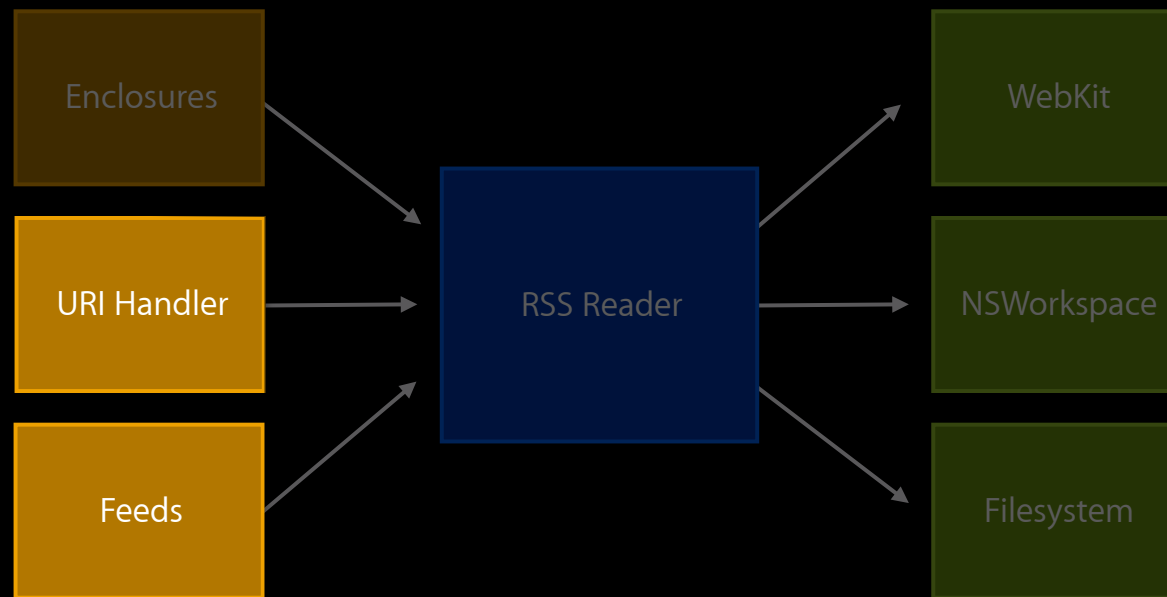
?

The image shows a standard Mac OS authorization dialog box. It features a lock icon in the top left corner. The main text asks the user to type their password to allow System Preferences to make changes. Below this, there are two input fields: 'Name' with the value 'A R' and 'Password'. A 'Details' section is expanded, showing the 'Right' as 'system.preferences.security' and the 'Application' as 'System Preferences' with its icon. At the bottom, there are 'Cancel' and 'OK' buttons, along with a help icon (question mark) and another set of 'Cancel' and 'OK' buttons.

Authorization Improvements



Attack Model for Breaking into RSS Reader



Code Vulnerabilities

- Input validation errors
- Memory mismanagement
- Logic flaws

RSS Reader Fail

```
NSLog(urlString)  
reader://%4$@
```

RSS Reader Improved

```
NSLog(@"%@", urlString)
```

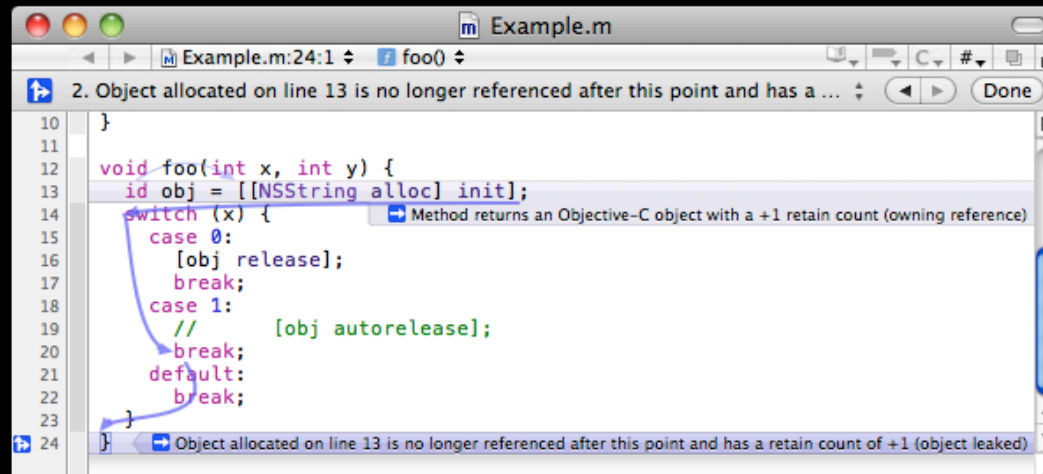
Code Defenses

- Code hardening
- Exploit mitigation
- Sandboxing



Code Hardening

- Clang static analyzer



The screenshot shows a code editor window titled "Example.m" with a Clang static analyzer warning. The warning message is: "2. Object allocated on line 13 is no longer referenced after this point and has a retain count of +1 (object leaked)". The code in the editor is as follows:

```
10 }
11
12 void foo(int x, int y) {
13     id obj = [[NSString alloc] init];
14     switch (x) {
15         case 0:
16             [obj release];
17             break;
18         case 1:
19             // [obj autorelease];
20             break;
21         default:
22             break;
23     }
24 }
```

The warning points to line 13, where the object is allocated. A tooltip for the `[[NSString alloc] init]` expression states: "Method returns an Objective-C object with a +1 retain count (owning reference)".

Code Hardening

Only on
Mac OS

- Fortify source

```
char buf[256];  
strcpy(buf, input);
```

```
char buf[256];  
strncpy(buf, input, sizeof(buf));
```

Code Hardening

Only on
Mac OS

- Stack protectors

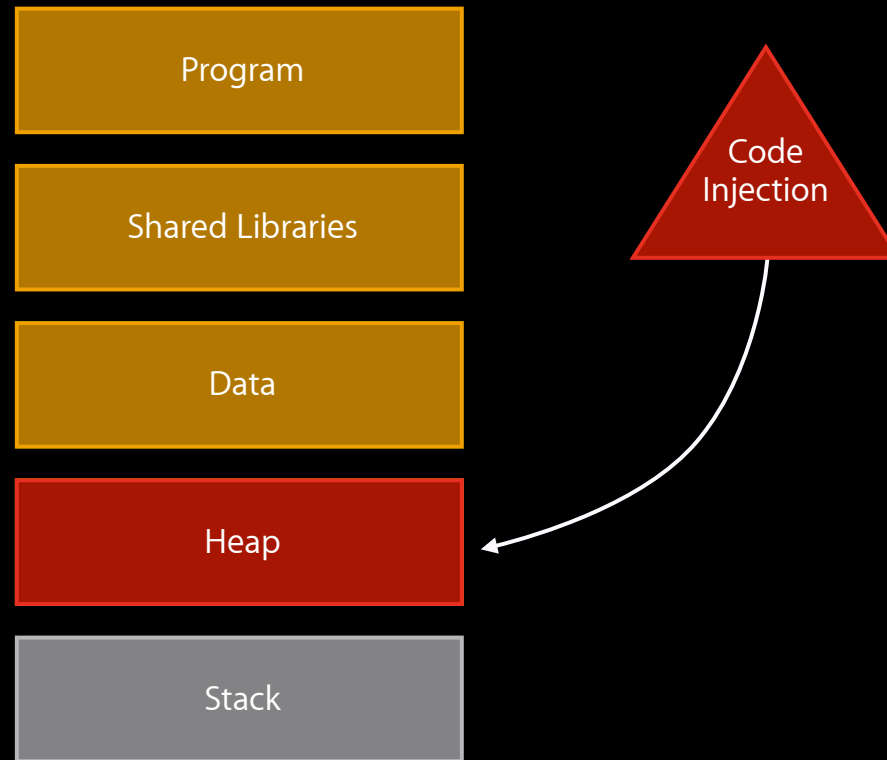
```
0x00000000100000eeb <main+103>: xor    (%rdx),%rcx
0x00000000100000eee <main+106>: je     0x100000ef5 <main+113>
0x00000000100000ef0 <main+108>: callq 0x100000ef8 <dyld_stub___stack_chk_fail>
0x00000000100000ef5 <main+113>: leaveq
0x00000000100000ef6 <main+114>: retq
```

Code Defenses

- Exploit mitigation



Code Injection Can Be Easy



Making Injection More Difficult

Improved
in Lion

Program

Shared Libraries

Data

Heap

Stack

Bypassing NX Data

- Return oriented programming

```
build_tif(base, ldmia_r4_r0);           // set stack base and initial jump

stack.Add(Node(0, Node::PTR));         // r0 = "/var/root/Media"
stack.Add(Node(1, Node::PTR));         // r1 = "/var/root/Oldmedia"
stack.Add(Node(20, Node::BYTES));      // r2,r3,r5,r6,r12
stack.Add(Node(12, Node::STACK));      // sp -> offset 12
stack.Add(ldmia_sp_r4);                 // lr = load r4,r7,pc from sp
stack.Add(rename);                     // pc = rename(r0, r1)

stack.Add(Node(12, Node::STACK));      // r4 = sp -> offset 12
stack.Add(Node(4, Node::BYTES));       // r7 = unused
stack.Add(ldmia_r4_r0);                 // pc = load r0...lr from r4

stack.Add(Node(2, Node::PTR));         // r0 = "/"
stack.Add(Node(0, Node::PTR));         // r1 = "/var/root/Media"
stack.Add(Node(20, Node::BYTES));      // r2,r3,r5,r6,r12
stack.Add(Node(12, Node::STACK));      // sp -> offset 12
stack.Add(ldmia_sp_r0);                 // lr = load from r0..pc from sp
stack.Add(symlink);                     // pc = symlink(r0, r1)

stack.Add(Node(3, Node::PTR));         // r0 = "hfs"
stack.Add(Node(2, Node::PTR));         // r1 = "/"
stack.Add(Node(0x00050000, Node::VAL)); // r2 = MNT_RELOAD | MNT_UPDATE
stack.Add(Node(8, Node::STACK));       // r3 = **data
stack.Add(mount);                       // pc = mount(r0, r1, r2, r3)
stack.Add(Node(4, Node::PTR));         // data = "/dev/disk0s1"

stack.Write();
```

http://www.toc2rta.com/files/itiff_exploit.cpp

Address Space Randomization

Aedsdrs Scpae Rzaimotinaodn

Aedsdrs Scpae Rzaimotinaodn

4.3

New

0x?????000	Program
0x?????000	Shared Libraries
0x2?????000	Data
0x3?????000	Heap
0x3?????000	Stack

Build Differences

Only on
Mac OS

Hardening Measure	Snow Leopard	Lion
ASLR (PIE)	Default off	Default on
NX Data	Default 64-bit only	Default on 32/64
MallocCorruptionAbort	Default 64-bit only	Default on 32/64
Char buf stack protectors	Default on	Default on
All stack cookies	-fstack-protector-all	-fstack-protector-all

Checking for Randomization

```
otool -arch x86_64 -hvr ExampleProgram.app/Contents/MacOS/ExampleProgram
```

Mach header

magic	cpu	cpu	caps	filetype	ncmds	sizeofcmds	flags
MH_MAGIC_64	X86_64	ALL	LIB64	EXECUTE	16	2008	NOUNDEFS DYLDLINK TWOLEVEL PIE



PIE

Code Defenses

- Sandboxing



Sandboxing

- Seatbelt
- Sandbox profiles
 - `kSBXProfileNoInternet`
 - `kSBXProfilePureComputation`

App Sandbox

Benefits

- Easy
- Transparency and conformance to user intent
- Fine grained access control
- Seamless protection in line with user experience



Related Sessions

Introducing App Sandbox

Nob Hill
Tuesday 2:00-3:00PM

App Sandbox and Mac App Store

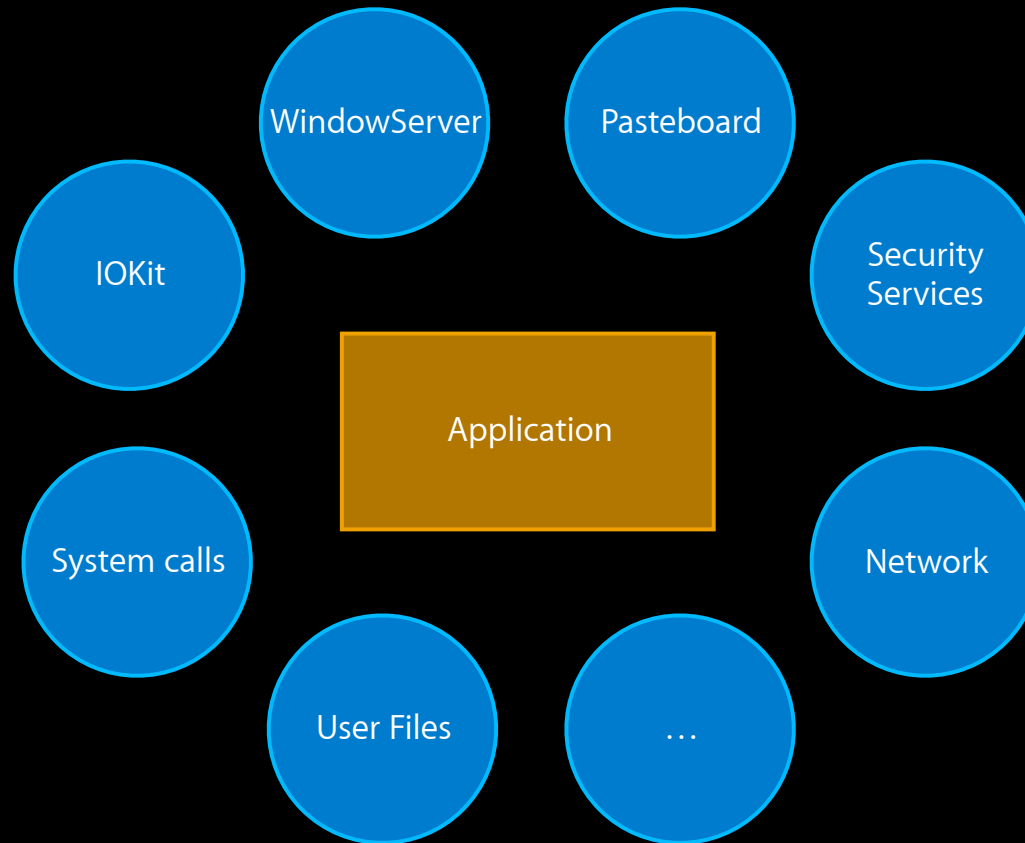
Nob Hill
Tuesday 3:15-4:15PM

Labs

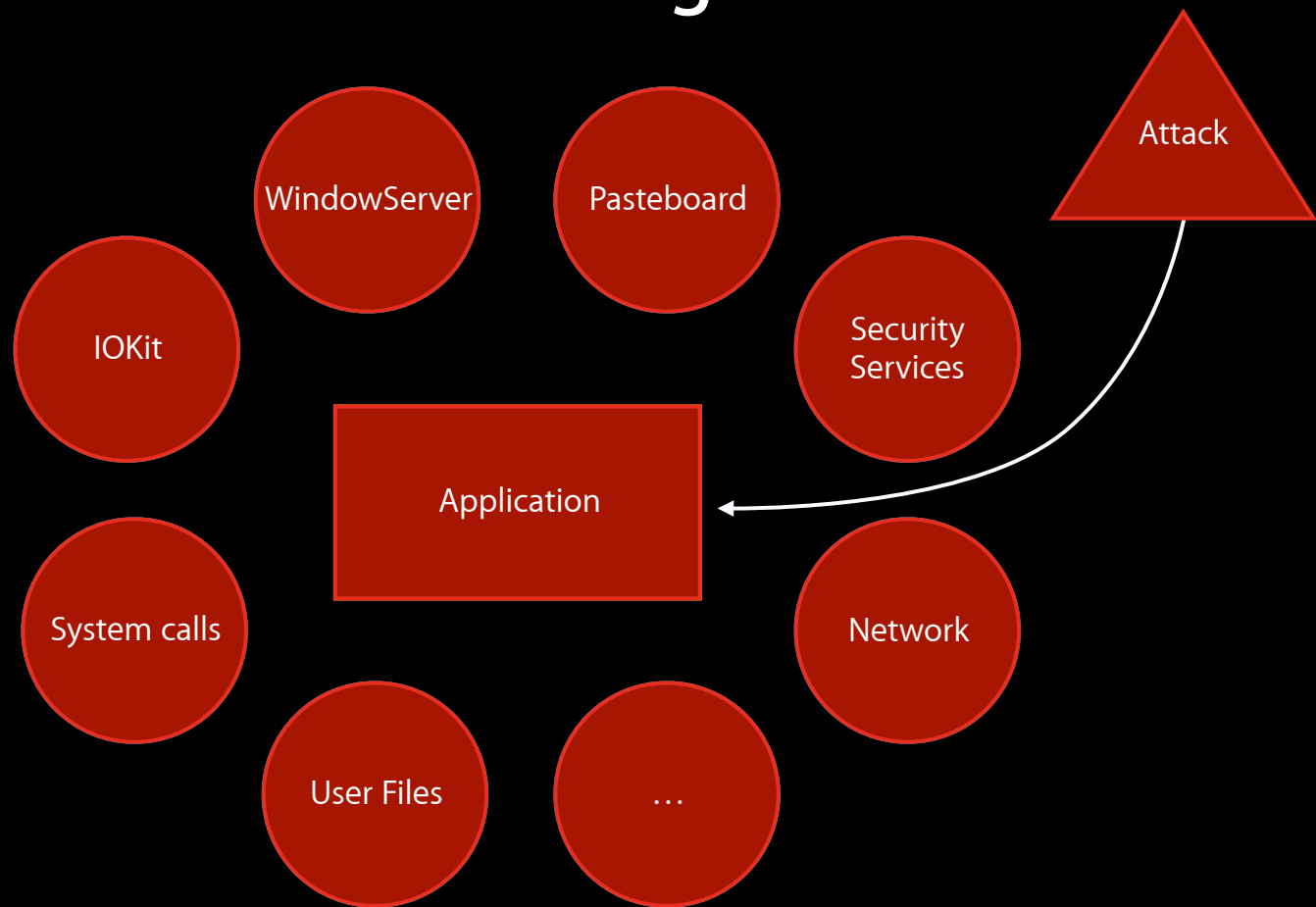
App Sandbox and Mac App Store

Location
Wednesday 9:00-12:30PM

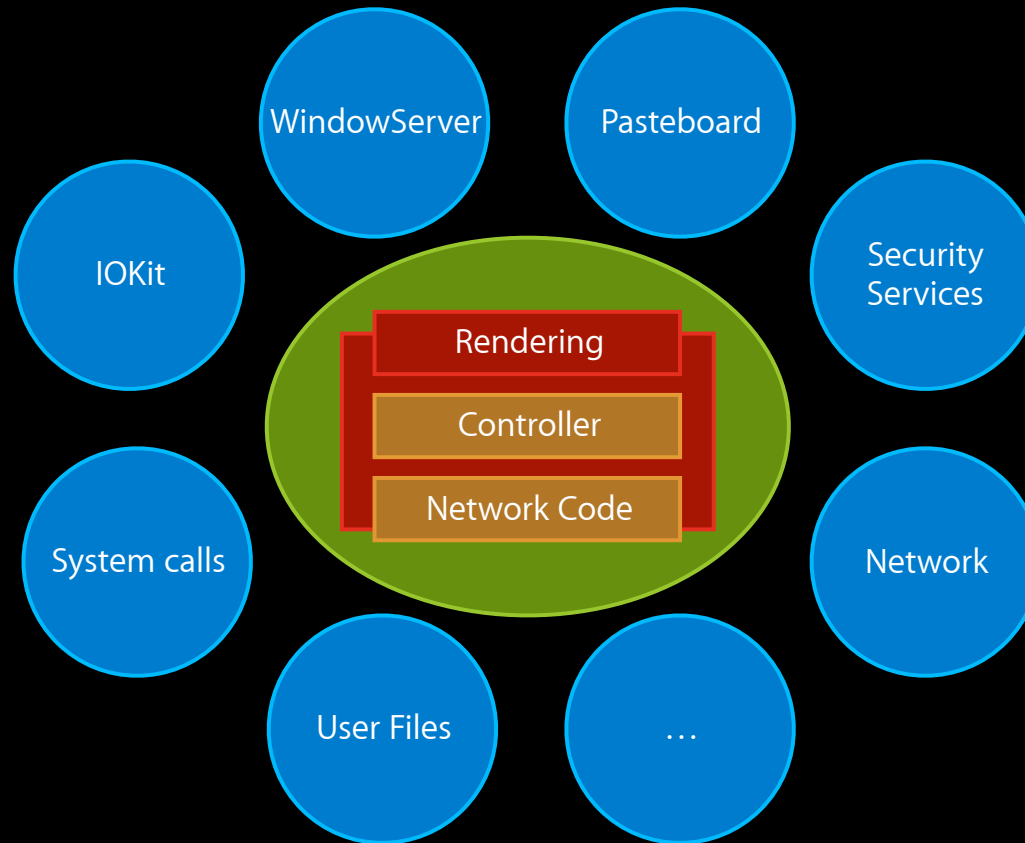
Sandboxing



Sandboxing



Sandboxing



App Sandbox



- Provides damage control
- Designed for interactive applications
- Entitlements

Understanding Entitlements

- Default deny
- Opt into capabilities

Example Entitlements



- Opening a connection to another machine
- Writing to Downloads folder
- User initiated read
- User initiated write
- Camera
- Printing

Financial Application

- Credentials
- Locally stores statements

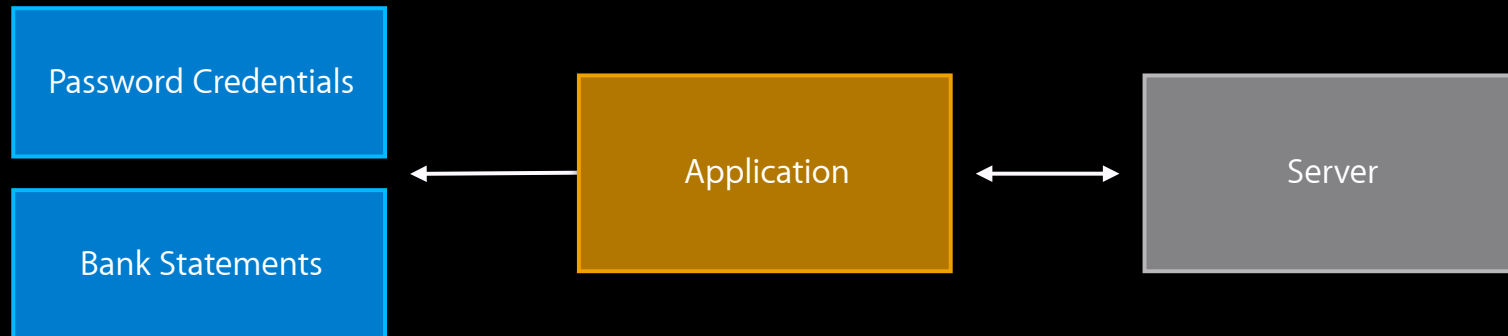


Storage Security Flaws

- Physical theft
- Incorrect file permissions
- Cross application leaks



Financial Application Attack Model



Storage Security Measures

- Keychain
- FileVault
- Data Protection



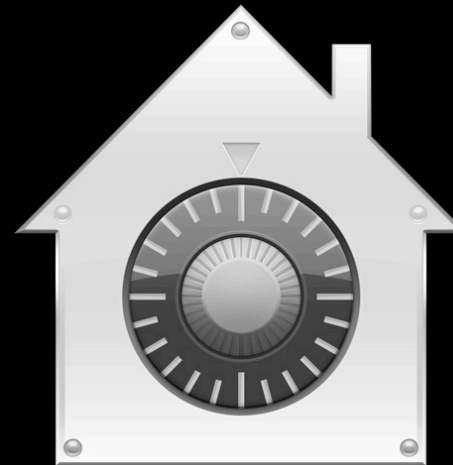
Keychain

- Secure credential storage



FileVault

- Encrypted home directories



FileVault

New

- Full disk encryption



Data Protection

Only on
iOS

- Protect files in case of compromise
- User implicitly manages availability with passcode
- Different protection classes available



The Protection Classes

Only on
iOS

File Availability

NSFiles

Keychain

Always

`NSFileProtectionNone`

`kSecAttrAccessibleAlways`

When device unlocked

`NSProtectionComplete`

`kSecAttrAccessibleWhenUnlockedunlocked`

When device unlocked or file open

After first device unlock

`kSecAttrAccessibleAfterFirstUnlock`

Use Cases

- Personal information
 - Notes, pictures, financial information
- Keychain
 - Credentials

The Protection Classes



File Availability

NSFiles

Keychain

Always

`NSFileProtectionNone`

`kSecAttrAccessibleAlways`

When device unlocked

`NSProtectionComplete`

`kSecAttrAccessibleWhenUnlockedunlocked`

When device unlocked or file open

`NS...CompleteUnlessOpen`

After first device unlock

`NS...CompleteUnlessFirstUserAuthentication`

`kSecAttrAccessibleAfterFirstUnlock`

Use cases

- For background applications
- Finishing large downloads

`NSFileProtectionCompleteUnlessOpen`

- Notifications and streaming

`NSFileProtectionAfterFirstUnlock`

File Permissions

- Temporary directories
- Incorrect permissions
 - World writeable

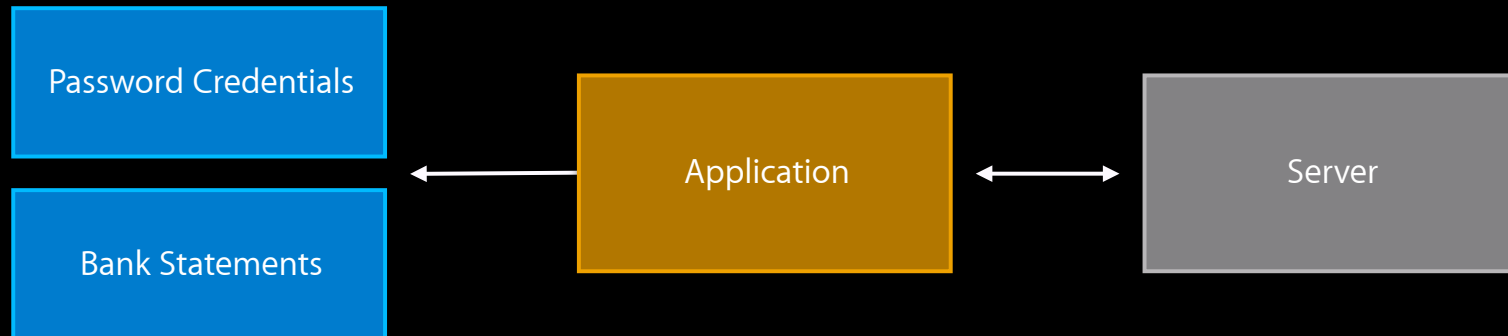
Stronger File Permissions



- No longer writeable by admin
- Exceptions
 - /Library/Caches
 - Role-owned directories

Directory	10.6	10.7
/	root:admin 0775	root:wheel 0755
/Applications/Utilities/	root:admin 0775	root:admin 0755
/Library/	root:admin 0775	root:wheel 0755

Financial Application Attack Model



Transport Security Attacks

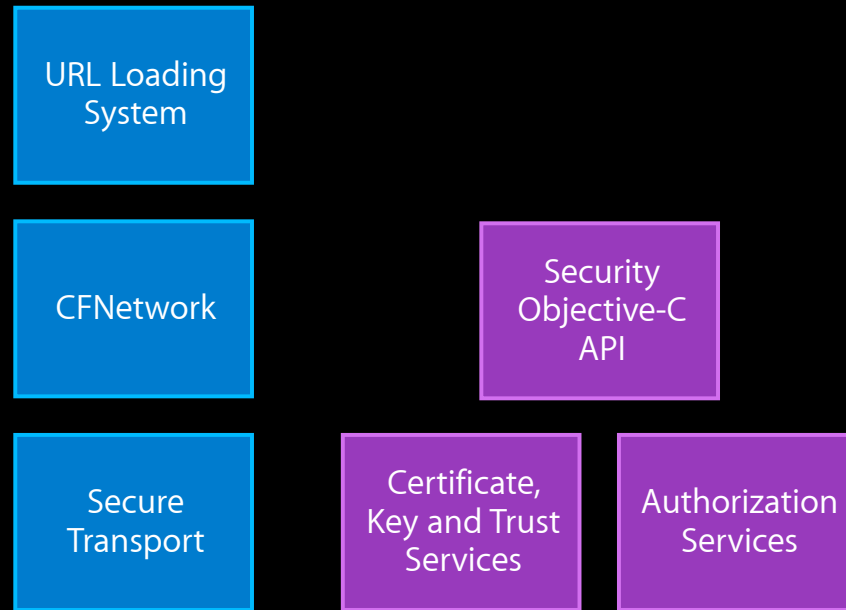
- Passive observation
- Man in the middle
- Replay attacks

```
07:36:19.796304 IP 192.168.0.103.56842 > 50.18.36.6.80: Flags [P.], seq 502:547, ack 1, win 65535, options
[nop,nop,TS val 875409880 ecr 1066924843], length 45
Maxim 0x0000: 0017 9a4a 8cc3 6033 4b25 4a2d 0800 4500 ...J...`3K%J-..E.
0x0010: 0061 4ea5 4000 4006 d4ca c0a8 0067 3212 .aN.@.....g2.
Mon Jul 0x0020: 2406 de0a 0050 80b3 afe3 fd8c 076a 8018 $....P.....j...
0x0030: ffff e6ad 0000 0101 080a 342d b1d8 3f97 .....4-...?.
0x0040: fb2b 7573 6572 6e61 6d65 3d74 2b6d 6f6e .+username=t+mon
nano 0x0050: 6579 6261 6773 2670 6173 7377 6f72 643d eybags&password=
0x0060: 7333 6372 3374 7034 7373 3477 3072 64 s3cr3tp4ss4w0rd
07:36:19.828312 IP 192.168.0.103.56842 > 50.18.36.6.80: Flags [.], ack 267, win 65535, options [nop,nop,TS
val 875409880 ecr 1066924850], length 0
0x0000: 0017 9a4a 8cc3 6033 4b25 4a2d 0800 4500 ...J...`3K%J-..E.
0x0010: 0034 c4d4 4000 4006 5ec8 c0a8 0067 3212 .4..@.^.....g2.
0x0020: 2406 de0a 0050 80b3 b010 fd8c 0874 8010 $....P.....t...
0x0030: ffff 29a6 0000 0101 080a 342d b1d8 3f97 ..).....4-...?.
0x0040: fb32 .....4-...?.
07:36:19.828352 IP 192.168.0.103.56842 > 50.18.36.6.80: Flags [.], ack 268, win 65535, options [nop,nop,TS
val 875409880 ecr 1066924850], length 0
0x0000: 0017 9a4a 8cc3 6033 4b25 4a2d 0800 4500 ...J...`3K%J-..E.
0x0010: 0034 a2dd 4000 4006 80bf c0a8 0067 3212 .4..@.....g2.
0x0020: 2406 de0a 0050 80b3 b010 fd8c 0875 8010 $....P.....u...
0x0030: ffff 29a5 0000 0101 080a 342d b1d8 3f97 ..).....4-...?.
0x0040: fb32 .....4-...?.
07:36:19.828627 IP 192.168.0.103.56842 > 50.18.36.6.80: Flags [F.], seq 547, ack 268, win 65535, options [
nop,nop,TS val 875409880 ecr 1066924850], length 0
0x0000: 0017 9a4a 8cc3 6033 4b25 4a2d 0800 4500 ...J...`3K%J-..E.
0x0010: 0034 e7e0 4000 4006 3bbc c0a8 0067 3212 .4..@.....g2.
0x0020: 2406 de0a 0050 80b3 b010 fd8c 0875 8011 $....P.....u...
0x0030: ffff 29a4 0000 0101 080a 342d b1d8 3f97 ..).....4-...?.
0x0040: fb32 .....4-...?.
```

Transport Security Solutions

- Security Framework
- SecureTransport

Security APIs



Mac OS X APIs



- Transforms
 - Easier, less code, better performance
- New Security APIs
 - CommonCrypto
 - Keychain

New iOS APIs



- Kerberos GSS API
- SecureTransport now available
 - TLS 1.1, TLS 1.2, DTLS

Summary

- User Interface defenses
 - Quarantine
 - Authorization prompts
- Code hardening features
- Built in exploit mitigation technology
- Sandboxing
 - App Sandbox
- Transport security
 - NextGen Crypto
 - SecureTransport on iOS
- Data storage security
 - FileVault
 - DataProtection

Useful References

Security Introduction

http://developer.apple.com/library/mac/#referencelibrary/GettingStarted/GS_Security/

Security Overview

http://developer.apple.com/library/mac/#documentation/Security/Conceptual/Security_Overview/

Secure Coding Guide

<http://developer.apple.com/library/mac/#documentation/Security/Conceptual/SecureCodingGuide/>

Launch Services (Quarantine notes)

http://developer.apple.com/library/mac/#releasenotes/Carbon/RN-LaunchServices/_index.html

Related Sessions

Security Overview	Nob Hill Tuesday 11:30-12:30PM
Introducing App Sandbox	Nob Hill Tuesday 2:00-3:00PM
Mac App Store	Nob Hill Tuesday 3:15-4:15PM
Securing Application Data	Nob Hill Thursday 9:00-10:00AM

Labs

Sandbox Lab

Core OS Lab B
Wednesday 9:00-11:15AM

Security Lab

Core OS Lab B
Thursday 11:30-1:30PM

Security Lab

Core OS Lab B
Friday 11:30-12:30PM



