Laufzeitanalyse & Manipulation von Apple iOS Apps

Andreas Kurtz

NESO Security Labs GmbH
Universität Erlangen-Nürnberg

mail@andreas-kurtz.de

OWASP
07.11.2012

The OWASP Foundation
http://www.owasp.org

Agenda

- Einführung in die Objective-C Runtime
  - Hintergründe & Techniken zur App-Manipulation
- Anwendungsfälle und Auswirkungen
- Maßnahmen zum Schutz der Runtime

Objective-C Runtime

EINFÜHRUNG

Objective-C

- Strikte Obermenge der Programmiersprache C

- Syntax und Konzeption an Smalltalk angelehnt
  - Nachrichtenbasiert
  - Dynamische Typisierung
  - Reflection

Objective-C

- Quelltextbeispiel:

```
HelloWorld *hello = [[HelloWorld alloc] init];
[hello sayHello:@"OWASP"];
```

```
- (void) sayHello: (NSString *) string {
   printf("Hello %s!", [string UTF8String]);
}
```

Objective-C

- Quelltextbeispiel:

```
HelloWorld *hello = [[HelloWorld alloc] init];
[hello sayHello:@"OWASP"];
```

- Im Hintergrund:

```
Class class = objc_getClass("HelloWorld");
id receiver = [[class alloc] init];
SEL selector = NSSelectorFromString(@"sayHello:");

objc_msgSend(receiver,selector,@"OWASP");
```

Objective-C

- Quelltextbeispiel:

```
HelloWorld *hello = [[HelloWorld alloc] init];
[hello sayHello:@"OWASP"];
```

Pointer auf das Objekt der Klasse *HelloWorld*

- Im Hintergrund:

```
Class class = objc_getClass("HelloWorld");
id receiver = [[class alloc] init];
SEL selector = NSSelectorFromString(@"sayHello:");

objc_msgSend(receiver,selector,@"OWASP");
```

Objective-C

- Quelltextbeispiel:

```
HelloWorld *hello = [[HelloWorld alloc] init];
[hello sayHello:@"OWASP"];
```

- Im Hintergrund:

Name der aufzurufenden Methode

```
Class class = objc_getClass("HelloWorld");
id receiver = [[class alloc] init];
SEL selector = NSSelectorFromString(@"sayHello:");
```



```
objc_msgSend(receiver,selector,@"OWASP");
```

Objective-C

- Quelltextbeispiel:

```
HelloWorld *hello = [[HelloWorld alloc] init];
[hello sayHello:@"OWASP"];
```

- Im Hintergrund:

Parameter an die Methode

```
Class class = objc_getClass("HelloWorld");
id receiver = [[class alloc] init];
SEL selector = NSSelectorFromString(@"sayHello:");

objc_msgSend(receiver,selector,@"OWASP");
```

Objective-C

- Statische Analyse wird erschwert
  - objc_msgSend

- Aber: Eigenschaften der Sprache ermöglichen eine umfassende dynamische Analyse

Technik	Verwendung
▪ Abfangen von Nachrichten	▪ Tracing interner Abläufe
▪ Senden beliebiger Nachrichten an bestehende Objekte ▪ Überschreiben von Methodenimplementierungen	▪ Manipulation interner Zustände und Abläufe

Hintergründe & Techniken

APP-MANIPULATION

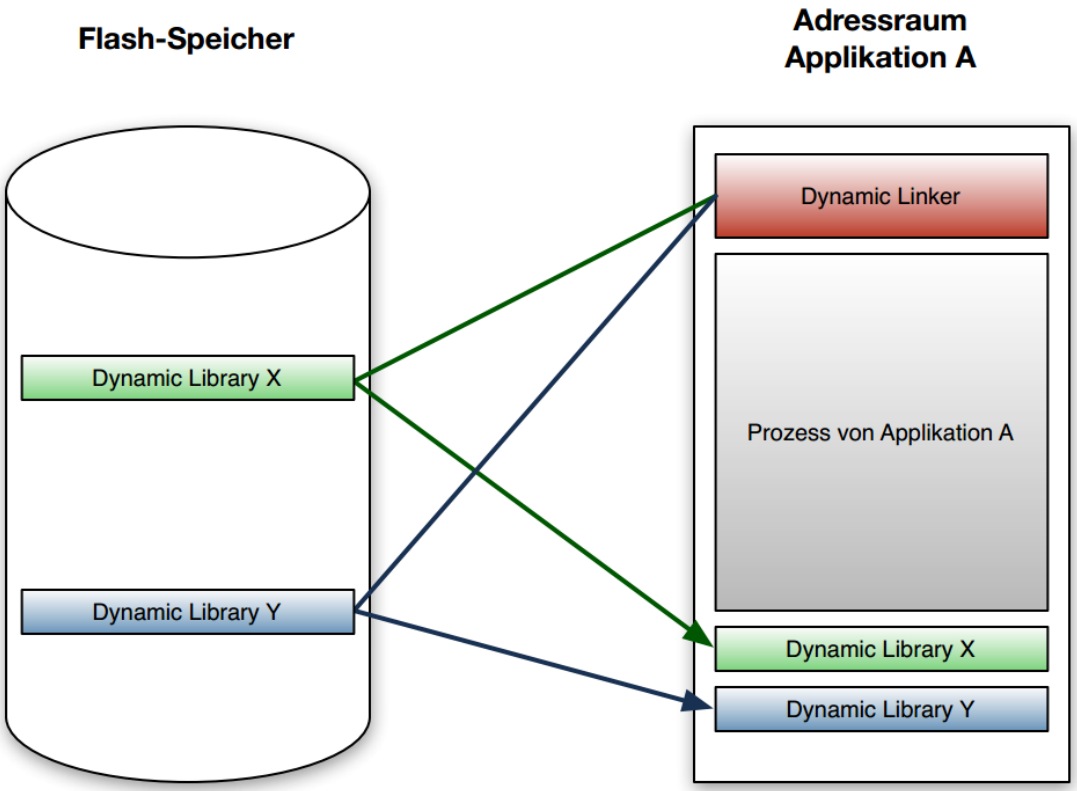
Laufzeitmanipulation

- Objective-C Runtime [1] bietet zahlreiche Möglichkeiten zur Laufzeitmanipulation

- Zwei unterschiedliche Herangehensweisen
  - Einbringen einer statischen Bibliothek mit neuen Funktionalitäten
  - Einbringen eines Interpreters für spontane Manipulationen

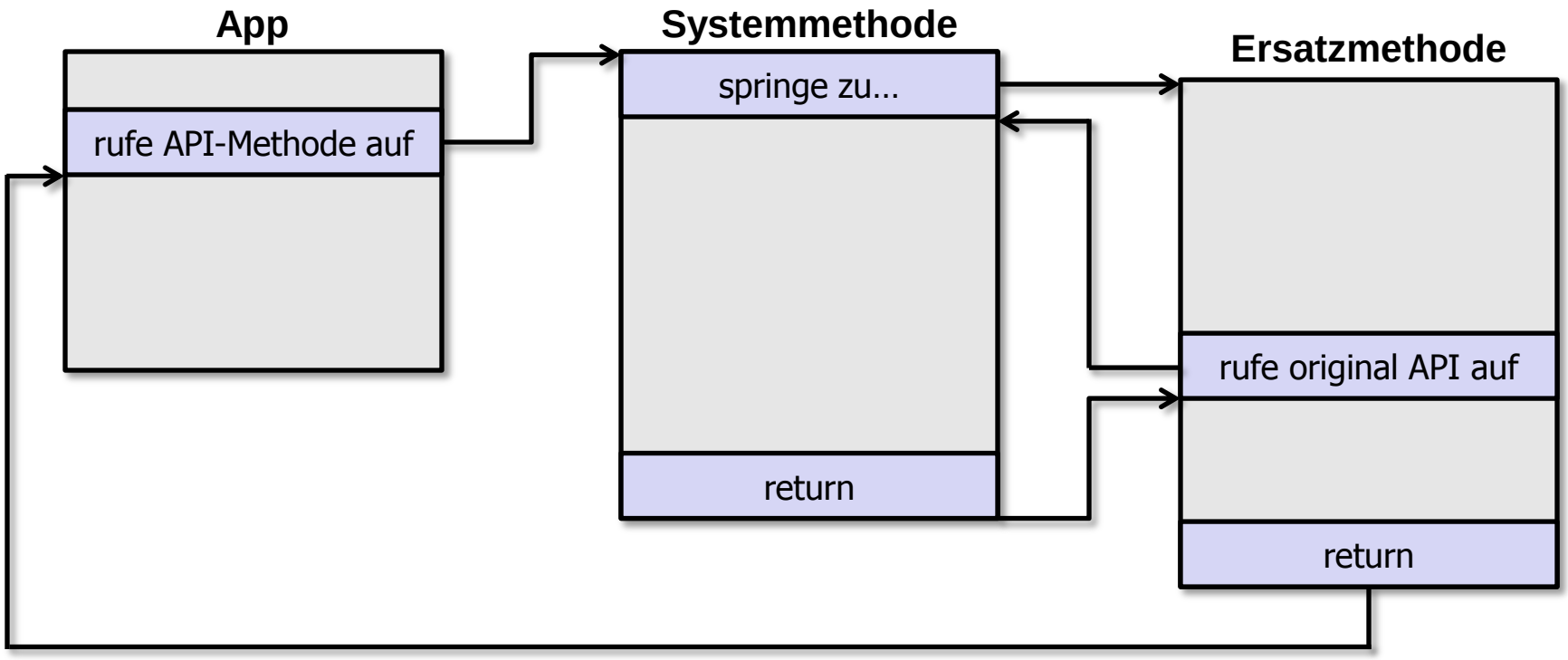
Dynamic Library Injection

- Nachladen einer Bibliothek [2] über die Umgebungsvariable *DYLD_INSERT_LIBRARIES*



Detour Runtime Patching

- Überschreiben von Methoden- bzw. Funktionsaufrufen



Hooking in der Praxis

- Mobile Substrate [3]
  - MobileLoader: Nachladen von Bibliotheken beim Start einer App
  - MobileHooker: Verantwortlich für das eigentliche Hooking, ersetzt Methoden und Funktionen

```
IMP MSHookMessage(Class class, SEL selector, IMP
replacement, const char* prefix);

void MSHookFunction(void* function, void* replacement,
void** p_original);
```

- Entwicklungs-Framework Theos (Logos) [4] erleichtert die Erstellung von Bibliotheken

Beispiel: Fälschen von Gerätemerkmalen

```
#include "substrate.h"
#import <Foundation/Foundation.h>

NSString *replaced_UIDevice_uniqueIdentifier() {
    return @"German OWASP Day";
}

__attribute__((constructor))
static void initialize() {
    MSHookMessage(objc_getClass("UIDevice"),
                  @selector(uniqueIdentifier),
                  (IMP)replaced_UIDevice_uniqueIdentifier, NULL);
}
```

Laufzeitmanipulation

- Objective-C Runtime [1] bietet zahlreiche Möglichkeiten zur Laufzeitmanipulation

- Zwei unterschiedliche Herangehensweisen
  - Einbringen einer statischen Bibliothek mit neuen Funktionalitäten ✓
  - Einbringen eines Interpreters für spontane Manipulationen

Cycript: Objective-JavaScript [5]

"A programming language designed to blend the barrier between Objective-C and JavaScript."

- Erweitert Apps um einen JavaScript-Interpreter
  - Basierend auf MobileSubstrate
- Ermöglicht spontane Manipulationen zur Laufzeit auf flexible Weise [6], [7]

Beispiel: Fälschen von Gerätemerkmalen

- **Schritt 1:** An die App “dranhängen”

```
# cycript -p <PID>
```

- **Schritt 2:** Auslesen der Geräte-ID

```
cy# [[UIDevice currentDevice] uniqueIdentifier];
@"768f0c93a69276d190b6…"
```

Beispiel: Fälschen von Gerätemerkmalen

- **Schritt 3:** Überschreiben der API-Methode:

```
cy# UIDevice.messages['uniqueIdentifier'] =
    function() { return @“German OWASP Day”; }
```

- **Schritt 4:** Nochmaliges Abfragen der nun gefälschten Geräte-ID:

```
cy# [[UIDevice currentDevice] uniqueIdentifier];
@"German OWASP Day"
```

Beispiel: Fälschen von Gerätemerkmalen

Weitere

ANWENDUNGSFÄLLE

Vorteile der Laufzeitmanipulation

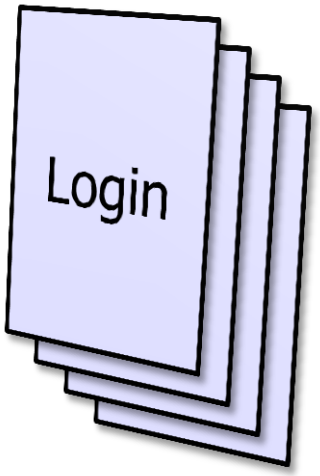
- Mittels der gezeigten Techniken lassen sich beliebige Zustände einer App manipulieren

- Unterstützung bei der dynamischen Analyse
  - Vereinfacht das Aufdecken von Schwachstellen
  - Ermöglicht das Umgehen von Sicherheitsmechanismen auf Client-seite

- Unterschiedliche Ansatzpunkte
  - Manipulation der Applikationslogik
  - Manipulation von Systemaufrufen

Beispiel: Vorgelagerte Anmeldemasken

- Sicherheitsmechanismen auf Client-seite
  - **Beispiel:** Vorgelagerte Anmeldemasken verhindern den Zugriff auf die eigentliche App



Umgehen einer Passcode-Sperre [8]

DEMO

Beispiel: Vorgelagerte Anmeldemasken

- **Schritt 1:** An die App “dranhängen”

```
# cycript -p SpringBoard
```

- **Schritt 2:** Prüfen, ob ein Passcode gesetzt ist

```
cy# [SBAwayController.sharedAwayController
                                isPasswordProtected]
1
```

Beispiel: Vorgelagerte Anmeldemasken

- **Schritt 3:** Überschreiben der App-Methode

```
cy# SBAwayController.messages['isPasswordProtected']
                              = function() { return NO; }
```

- **Schritt 4:** Aufruf der Unlock-Methode

```
cy# [SBAwayController.sharedAwayController
                              unlockWithSound:YES ]
```

Weitere Beispiele

- Übertragbar auf zahlreiche weitere Beispiele:
  - Nachweis schwacher Verschlüsselung
  - Umgehen von Beschränkungen auf Client-seite
  - Ausführen von verstecktem Code, der nicht über die GUI zugänglich ist
  - Freischalten zusätzlicher App-Funktionalität
  - Kopieren von „geschützten Inhalten“
  - Etc.

Vorteile der Laufzeitmanipulation

- Mittels der gezeigten Techniken lassen sich beliebige Zustände einer App manipulieren

- Unterstützung bei der dynamischen Analyse
  - Vereinfacht das Aufdecken von Schwachstellen
  - Ermöglicht das Umgehen von Sicherheitsmechanismen auf Client-seite

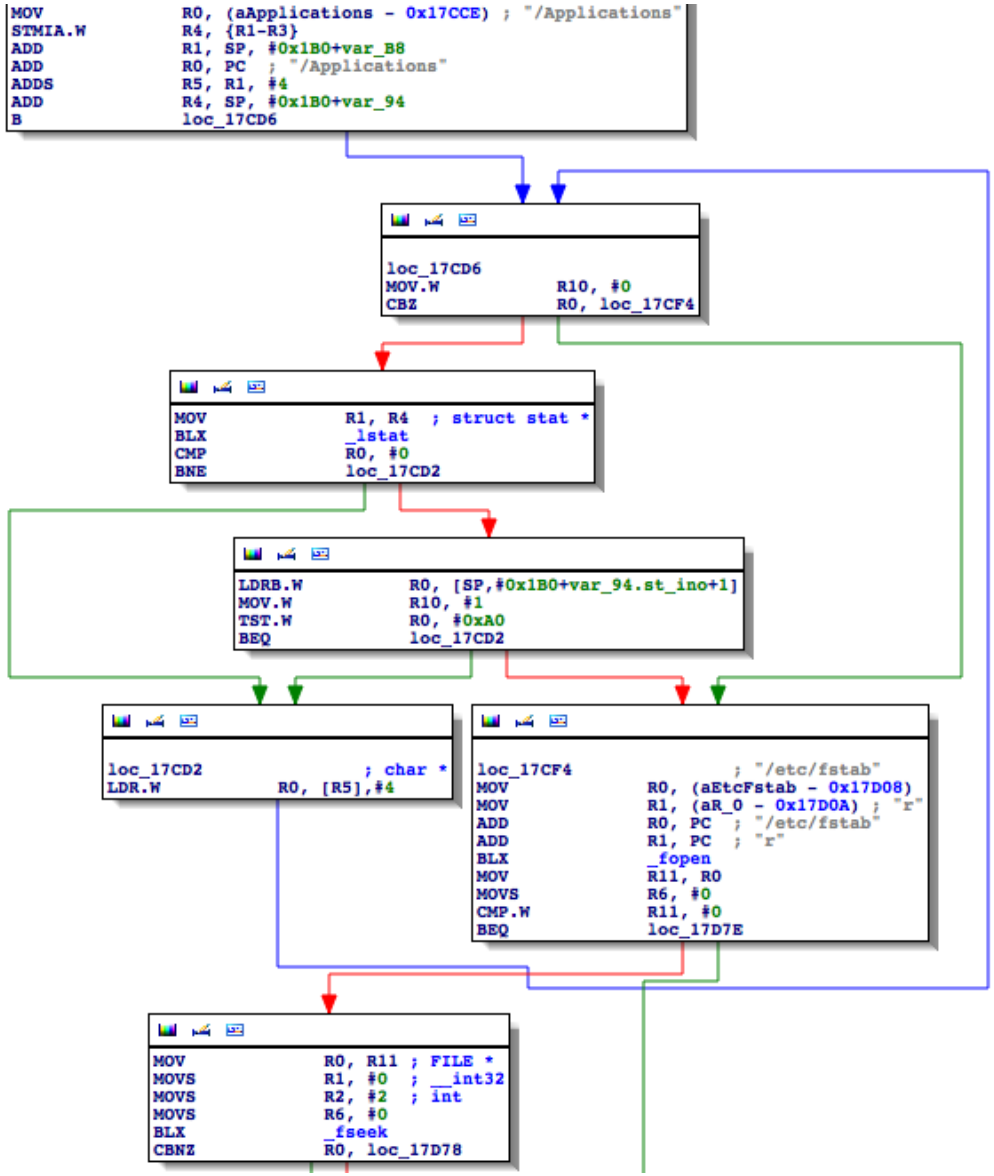
- Unterschiedliche Ansatzpunkte
  - Manipulation der Applikationslogik ✓
  - Manipulation von Systemaufrufen

Beispiel: Jailbreak-Erkennung

- **Ziel:** Sicherstellung der Plattformintegrität

- Häufig verwendete Prüfungen
  - Verdächtige Dateien und Verzeichnisse
  - Dateisystemberechtigungen
  - Mount-Optionen
  - Symbolische Links
  - Dynamisch geladene Bibliotheken
  - SSH Loopback
  - Erzeugen von Kindprozessen (Fork)
  - Etc.

Beispiel: Jailbreak-Erkennung



Beispiel: Jailbreak-Erkennung

- Um eine App prüfen zu können, muss zunächst die Erkennung umgangen werden
  - „Herauspatchen“ der Prüfroutine aus dem App Binary oder zur Laufzeit (spezifisch, zeitaufwendig)

```
Delegate.messages['isJailbroken'] =
                    function() { return NO; }
```

  - Abfangen von Systemaufrufen und Vortäuschen einer intakten Ausführungsumgebung (generisch)

Beispiel: Jailbreak-Erkennung

```
static int (*original_lstat)(const char *path, struct stat *buf);

int replaced_lstat(const char *path, struct stat *buf) {
 …
 NSArray *jailbrokenPath = [NSArray arrayWithObjects:
                                @"/Applications/Cydia.app", … ,nil];

    for(NSString *string in jailbrokenPath)
        if([originalPath isEqualToString:string])
            NSLog(@"[+] Block lstat access to %@", string);
            return -1;

    int ret = original_lstat (path, buf);
} …
MSHookFunction((void*)lstat,(void*)replaced_lstat,(void**)&original_lstat);
```

Beispiel: Jailbreak-Erkennung

```
static int (*original_lstat)(const char *path, struct stat *buf);
```

```
int replaced_lstat(const char *path, struct stat *buf) {
 …
 NSArray *jailbrokenPath = [NSArray arrayWithObjects:
                                @"/Applications/Cydia.app", … ,nil];

    for(NSString *string in jailbrokenPath)
        if([originalPath isEqualToString:string])
            NSLog(@"[+] Block lstat access to %@", string);
            return -1;

    int ret = original_lstat (path, buf);
} …
MSHookFunction((void*)lstat,(void*)replaced_lstat,(void**)&original_lstat);
```

Pointer auf die Originalfunktion

Beispiel: Jailbreak-Erkennung

```
static int (*original_lstat)(const char *path, struct stat *buf);

int replaced_lstat(const char *path, struct stat *buf) {
 …
 NSArray *jailbrokenPath = [NSArray arrayWithObjects:
                                @"/Applications/Cydia.app", … ,nil];

    for(NSString *string in jailbrokenPath)
        if([originalPath isEqualToString:string])
            NSLog(@"[+] Block lstat access to %@", string);
            return -1;

    int ret = original_lstat (path, buf);
} …
MSHookFunction((void*)lstat,(void*)replaced_lstat,(void**)&original_lstat);
```

Beginn der Ersatzfunktion

Beispiel: Jailbreak-Erkennung

```
static int (*original_lstat)(const char *path, struct stat *buf);

int replaced_lstat(const char *path, struct stat *buf) {
 …
 NSArray *jailbrokenPath = [NSArray arrayWithObjects:
                            @"/Applications/Cydia.app", … , nil];

    for(NSString *string in jailbrokenPath)
        if([originalPath isEqualToString:string])
            NSLog(@"[+] Block lstat access to %@", string);
            return -1;

    int ret = original_lstat (path, buf);
} …
MSHookFunction((void*)lstat,(void*)replaced_lstat,(void**)&original_lstat);
```

Verdächtige Jailbreak-Pfade

Beispiel: Jailbreak-Erkennung

```
static int (*original_lstat)(const char *path, struct stat *buf);

int replaced_lstat(const char *path, struct stat *buf) {
 …
 NSArray *jailbrokenPath = [NSArray arrayWithObjects:
                                @"/Applications/Cydia.app", … ,nil];

    for(NSString *string in jailbrokenPath)
        if([originalPath isEqualToString:string])
            NSLog(@"[+] Block lstat access to %@", string);
            return -1;

    int ret = original_lstat (path, buf);
} …
MSHookFunction((void*)lstat,(void*)replaced_lstat,(void**)&original_lstat);
```

Zugriff auf verdächtige Pfade unterbinden

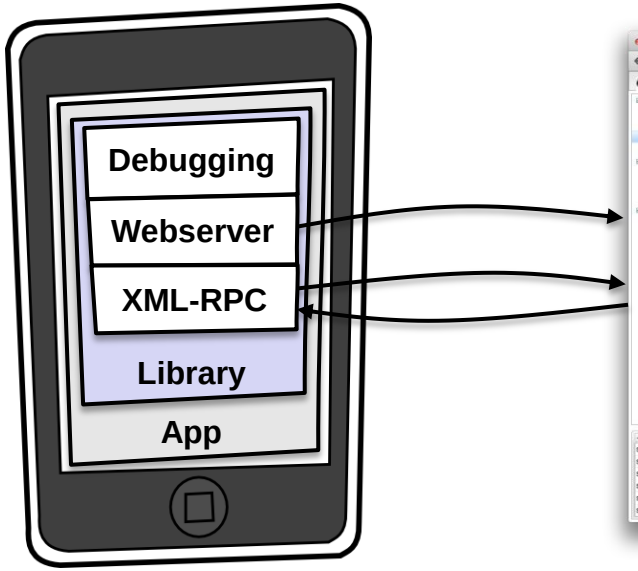
Zwischenfazit

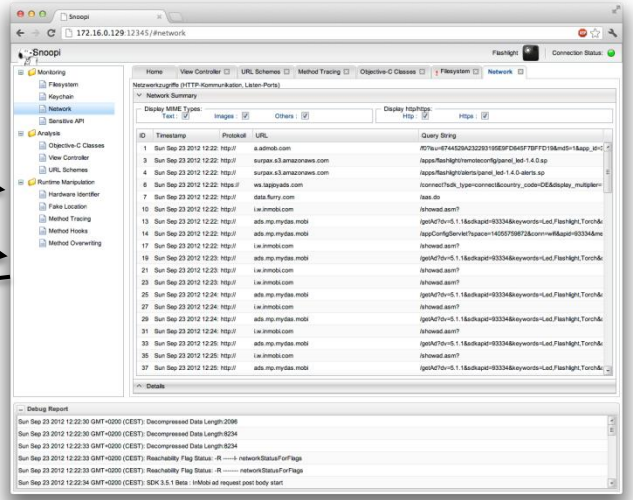
- Die Möglichkeit zur Laufzeitmanipulation vergrößert die Angriffsfläche mobiler Apps
  - Sicherheitsüberprüfungen werden erleichtert bzw. neue Angriffswege werden ermöglicht
- Kaum Werkzeuge zur dynamischen Analyse von Apps vorhanden

Neues Werkzeug

- Analyse und Manipulation von Apps zur Laufzeit über eine einfach zu bedienende Weboberfläche

- Veröffentlichung Ende 2012





GWT GUI

Sneak Preview

DEMO

Maßnahmen

SCHUTZ DER RUNTIME

Schutz vor Laufzeitmanipulation

- Möglichst wenig Daten/Logik auf Client-seite
- Bevorzugte Verwendung von C für sicherheitskritische Vorgänge
  - Inline Functions
  - Obfuscation
- Erweiterte Jailbreak-Erkennung
- Integritätsprüfung der Runtime (*dladdr()* [9])

Quellen

[1] Objective C Runtime Reference
http://developer.apple.com/library/mac/#documentation/Cocoa/Reference/ObjCRuntimeRef/Reference/reference.html

[2] dyld - the dynamic link editor (DYLD_INSERT_LIBRARIES)
http://developer.apple.com/library/mac/#documentation/Darwin/Reference/Manpages/man1/dyld.1.html

[3] Mobile Substrate
http://iphonedevwiki.net/index.php/MobileSubstrate

[4] Theos
http://iphonedevwiki.net/index.php/Theos

[5] Cycript
http://www.cycript.org

Quellen

[6] Übersicht zu Cycript
http://iphonedevwiki.net/index.php/Cycript

[7] Tipps zu Cycript
http://iphonedevwiki.net/index.php/Cycript_Tricks

[8] iOS Runtime Injection Example #1
http://www.andreas-kurtz.de/2012/02/ios-runtime-injection-example-1.html

[9] *dladdr* - find the image containing a given address
http://developer.apple.com/library/Mac/#documentation/Darwin/Reference/ManPages/man3/dladdr.3.html

Vielen Dank!

Andreas Kurtz

NESO Security Labs GmbH
Weipertstr. 8-10
74076 Heilbronn

info@nesolabs.de
http://www.nesolabs.de

OWASP

The OWASP Foundation
http://www.owasp.org