Sicherheit mobiler Apps

Andreas Kurtz

NESO Security Labs GmbH
Universität Erlangen-Nürnberg

mail@andreas-kurtz.de

OWASP
17.11.2011

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

Agenda

- OWASP Mobile Security Project
  - OWASP TOP 10 Mobile Risks
- Beispiele und Erfahrungen aus der Praxis
  - Fokus auf Defizite bei Authentisierung/Autorisierung
- Zusammenfassung

OWASP Mobile Security Project

SICHERHEIT MOBILER APPS

OWASP Mobile Security Project

- Seit 2010 [1]
- Zielsetzung
  - Entwicklung sicherer mobiler Apps
- Inhalte
  - Threat Model
  - Controls & Design Principles
  - Risks
  - Tools, Methodologies
  - Etc.

OWASP Top 10 Mobile Risks

1. Insecure Data Storage
2. Weak Server Side Controls
3. Insufficient Transport Layer Protection
4. Client Side Injection
5. Poor Authorization and Authentication
6. Improper Session Handling
7. Security Decisions Via Untrusted Inputs
8. Side Channel Data Leakage
9. Broken Cryptography
10. Sensitive Information Disclosure

Release Candidate v1.0

OWASP Top 10 Mobile Risks

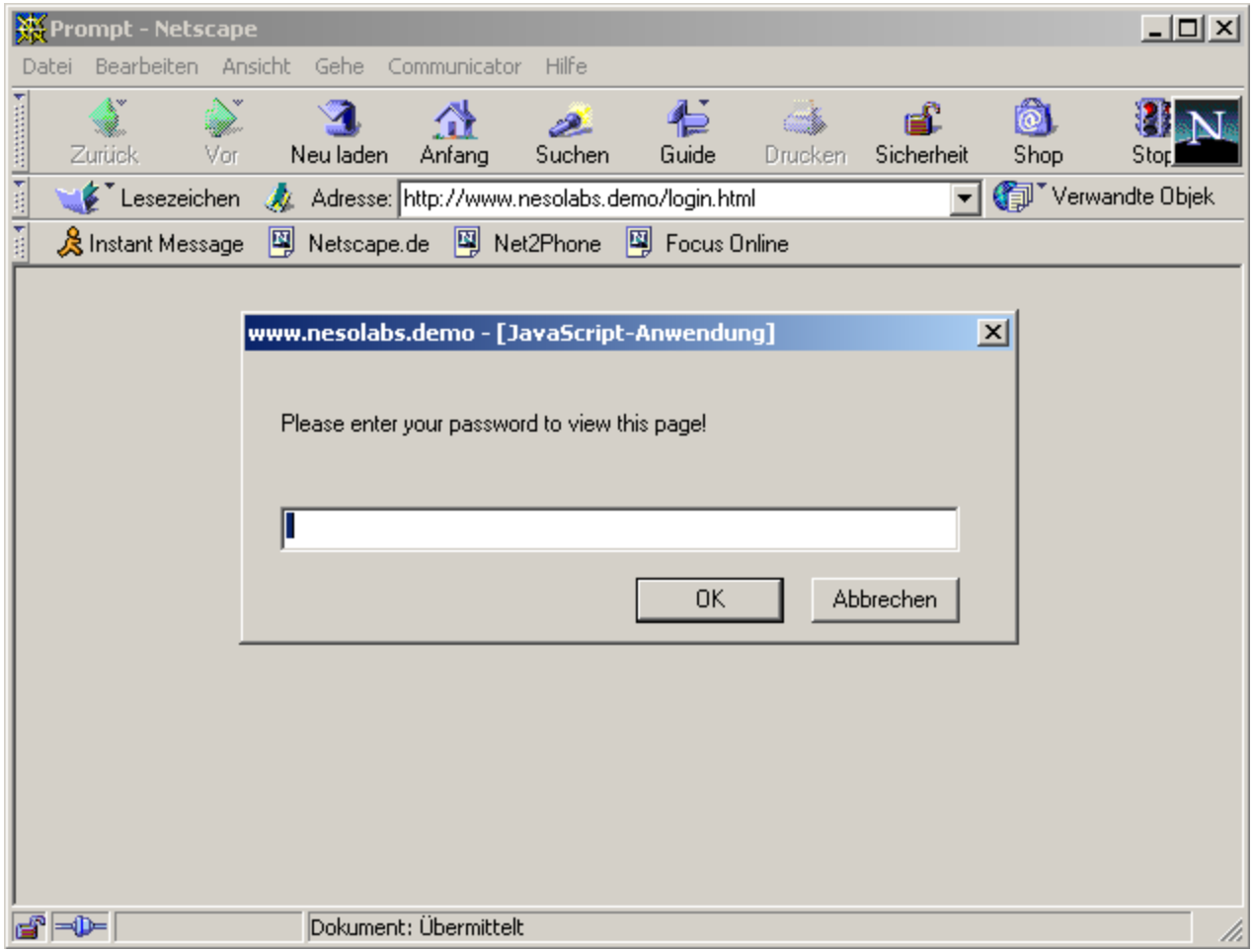
1. Insecure Data Storage
2. Weak Server Side Controls
3. Insufficient Transport Layer Protection
4. Client Side Injection
5. **Poor Authorization and Authentication**
6. Improper Session Handling
7. Security Decisions Via Untrusted Inputs
8. Side Channel Data Leakage
9. Broken Cryptography
10. Sensitive Information Disclosure

Release Candidate v1.0

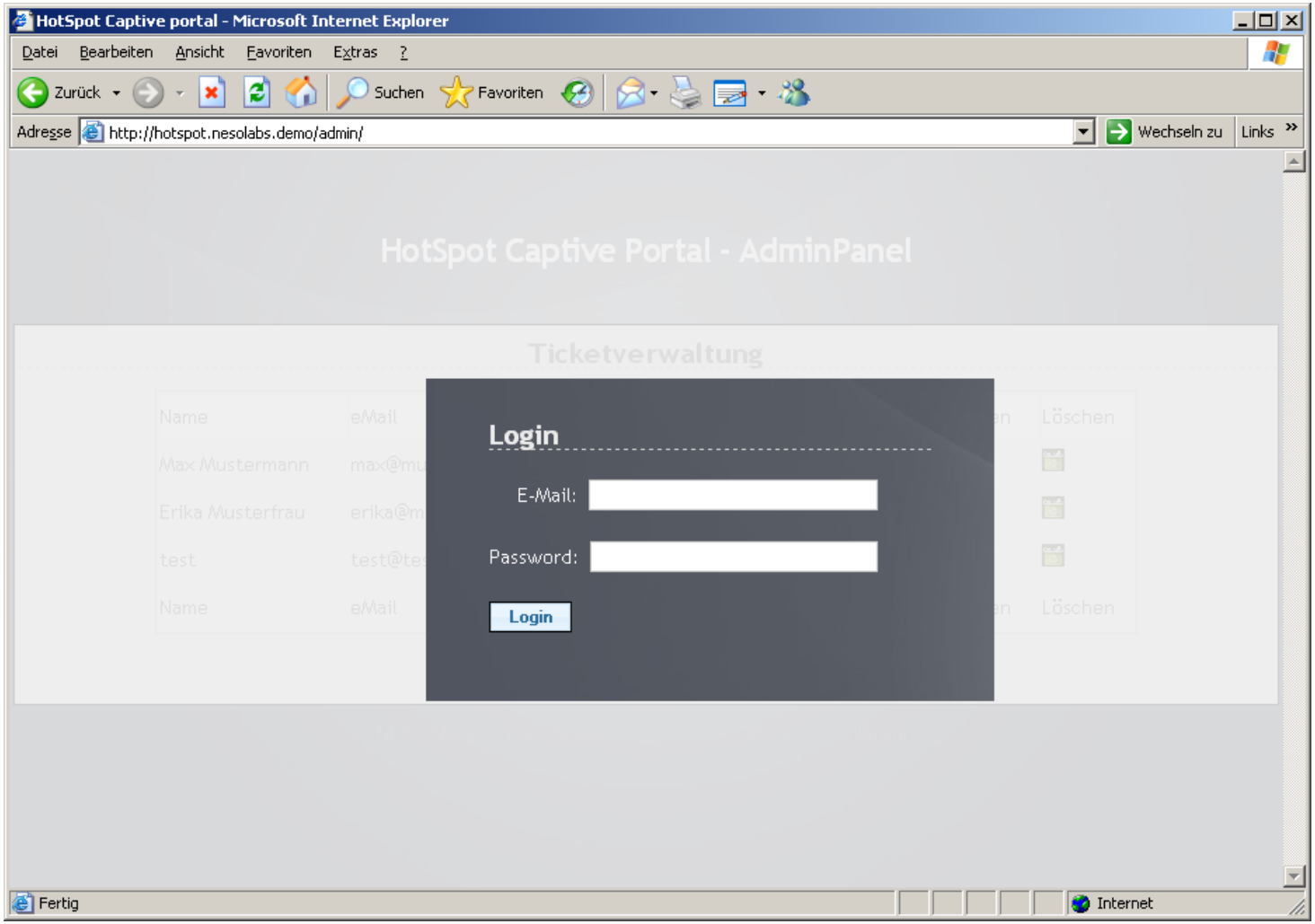
Defizite bei Authentisierung/Autorisierung

SICHERHEIT MOBILER APPS

1995


Prompt - Netscape
Datei Bearbeiten Ansicht Gehe Communicator Hilfe
Zurück Vor Neu laden Anfang Suchen Guide Drucken Sicherheit Shop Stop
Lesezeichen Adresse: http://www.nesolabs.demo/login.html Verwandte Objek
Instant Message Netscape.de Net2Phone Focus Online
www.nesolabs.demo - [JavaScript-Anwendung]
Please enter your password to view this page!
OK
Abbrechen
Dokument: Übermittelt

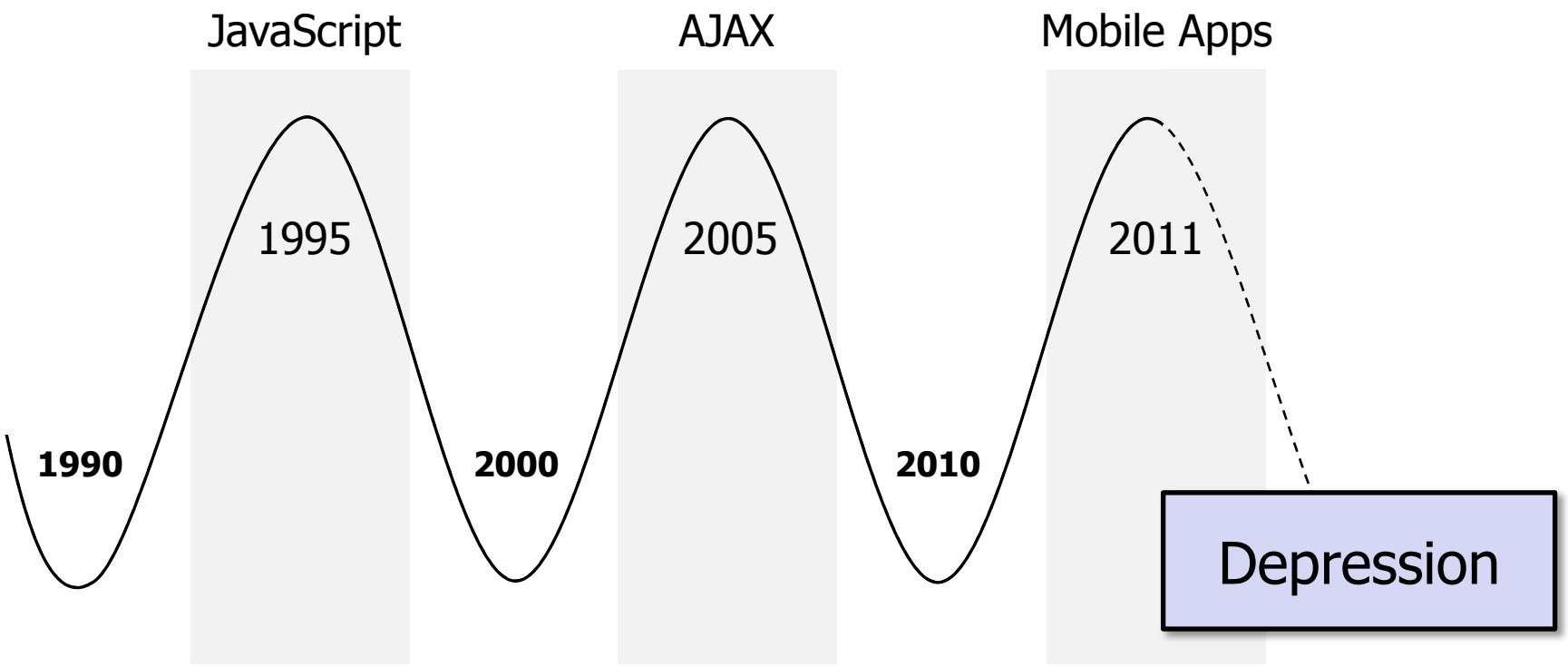
2005


HotSpot Captive portal - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück
Suchen
Favoriten
Adresse
http://hotspot.nesolabs.demo/admin/
Wechseln zu
Links
HotSpot Captive Portal - AdminPanel
Ticketverwaltung
Name
eMail
Löschen
Max Mustermann
Erika Musterfrau
test
Login
E-Mail:
Password:
Login
Fertig
Internet

2011


No SIM
12:41
+49 151 43
Done
Please confirm your country code and enter your phone number.
+49
Germany
your number without the country code:
151 43
1
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
0

Zyklische Entwicklung



Häufige Designfehler – Beispiele

- Authentisierung/Autorisierung anhand statischer Merkmale
  - UDID, IMEI, IMSI etc.
- Authentisierung/Autorisierung innerhalb der App
  - Manipulation des Transportwegs
  - Manipulation interner App-Zustände

Beispiel: Authentisierung anhand der UDID

- Registrierung eines Chat-Accounts

```
GET /client/iphone/xmpp_reg.php?cc=49&me=<VictimPhoneNumber> ↲
→ &udid=<AttackerUDID>&sms=1 HTTP/1.1
[…]
User-Agent: ChatApp/1.0 iPhone_OS/4.3.3 Device/iPhone_4
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
Proxy-Connection: keep-alive
```

Beispiel: Authentisierung anhand der UDID

- Registrierung eines Chat-Accounts

```
GET /client/iphone/xmpp_reg.php?cc=49&me=<VictimPhoneNumber> ↲
→ &udid=<AttackerUDID>&sms=1 HTTP/1.1
[…]
User-Agent: ChatApp/1.0 iPhone_OS/4.3.3 Device/iPhone_4
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
Proxy-Connection: keep-alive
```

Telefonnummer des Opfers (Benutzername)

Beispiel: Authentisierung anhand der UDID

- Registrierung eines Chat-Accounts

```
GET /client/iphone/xmpp_reg.php?cc=49&me=<VictimPhoneNumber> ↲
→ &udid=<AttackerUDID>&sms=1 HTTP/1.1
[…]
User-Agent: ChatApp/1.0 iPhone_OS/4.3.3 Device/iPhone_4
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
Proxy-Connection: keep-alive
```

Geräte-ID des Angreifers (Passwort)

Wo lag der Fehler?

- Authentisierung bzw. Session anhand statischer, kompromittierter Merkmale (z.B. Geräte-ID)
  - Geräte-ID wird häufig für Statistikzwecke an Dienstleister übermittelt (kein Geheimnis)
  - Geräte-ID sowie beliebige andere Hardware-Merkmale können mittels Software manipuliert werden

Unique Device Identifier (UDID)

*"**Important:** Never store user information based solely on the UDID. **Always use a combination of UDID and application-specific user ID.** A combined ID ensures that if a user passes a device on to another user, the new user will not have access to the original user's data."*

Quelle: iOS Developer Library
http://developer.apple.com/library/ios/#documentation/uikit/re

Deprecated in iOS 5.0

Forschungsprojekt „Gorilla“

- Erweitert das iOS Betriebssystem um verschiedene Sicherheits- und Kontrollfunktionen
- Absicherung des Systems sowie eine bessere Überwachung installierter Apps

Forschungsprojekt „Gorilla“

- Kostenlos verfügbar in Cydia [2]
  (ab Anfang 2012 auch für iOS 5)

Häufige Designfehler – Beispiele

- Authentisierung/Autorisierung anhand statischer Merkmale
  - UDID, IMEI, IMSI etc.

- Authentisierung/Autorisierung innerhalb der App
  - Manipulation des Transportwegs
  - Manipulation interner App-Zustände

Beispiel: Authentisierung via SMS

- Authentisierung eines Nutzers anhand dessen Telefonnummer
  - **Schritt 1:** Benutzer gibt seine Telefonnummer ein
  - **Schritt 2:** Telefonnummer wird ans Backend übermittelt
  - **Schritt 3:** Backend schickt SMS mit PIN an die angegebene Nummer
  - **Schritt 4:** Benutzer beendet die Verifikation durch Eingabe der PIN

Schritt 1: Eingabe der Telefonnummer

Schritt 2: Übermitteln der Nummer

- HTTPS-Anfrage an den Server:

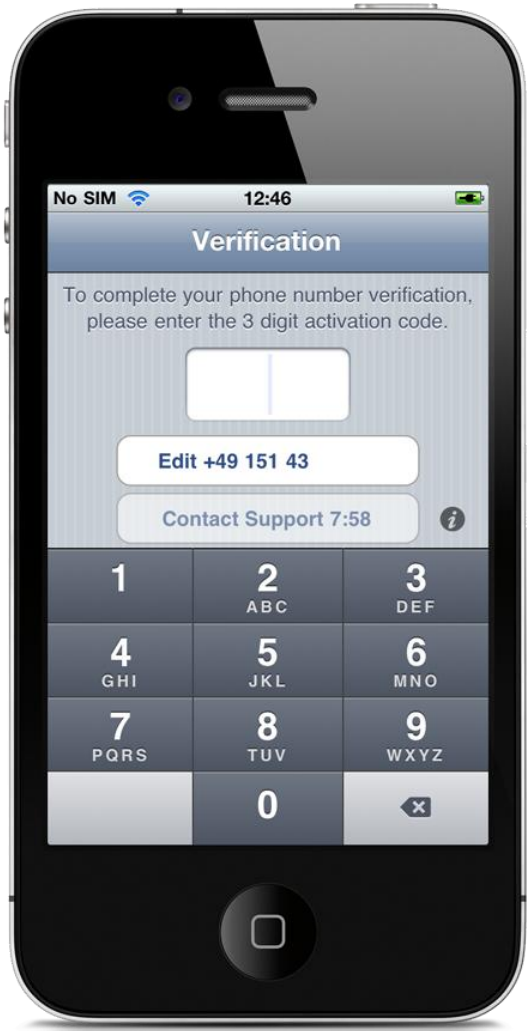
```
GET /client/iphone/smsproxy.php?to=4915143[..]&auth=569 ↲
→ &in=15143[..]&code=49&udid=b33f6a2df975532f846ca[…] HTTP/1.1
[…]
User-Agent: Messenger/2.6.4 iPhone_OS/4.3.3 Device/iPhone_4
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Schritt 2: Übermitteln der Nummer

- Server-Antwort:

```
HTTP/1.1 200 OK
X-Powered-By: PHP/5.2.11
Content-type: text/html
Connection: close
Date: Sat, 18 Jun 2011 13:48:13 GMT
Server: lighttpd/1.4.24
Content-Length: 60
ID: 1308404892 #Message Receive correctly ORDERID=18542673
```

Schritt 3/4: Eingabe & Verifikation der PIN



Wo lag der Fehler?

- Verifikation der PIN innerhalb der App [3]

```
GET /client/iphone/smsproxy.php?to=4915143[..]&auth=569 ↲
→ &in=15143[..]&code=49&udid=b33f6a2df975532f846ca[…] HTTP/1.1
[…]
User-Agent: Messenger/2.6.4 iPhone_OS/4.3.3 Device/iPhone_4
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Wo lag der Fehler?

- Verifikation der PIN innerhalb der App [3]

```
GET /client/iphone/smsproxy.php?to=4915143[..]&auth=569 ↲
→ &in=15143[..]&code=49&udid=b33f6a2df975532f846ca[…] HTTP/1.1
[…]
User-Agent: Messenger/2.6.4 iPhone_OS/4.3.3 Device/iPhone_4
Accept: */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

PIN wird bereits innerhalb der App generiert und zum SMS-Versand an das Backend übermittelt

Auswirkungen

- Übernahme einer beliebigen Identität
  - Eingabe der Telefonnummer des Opfers
  - „Abfangen“ der HTTP-Anfrage und auslesen der PIN
  - Vortäuschen einer korrekten Server-Antwort
  - Eingabe der PIN

- SMS-Proxy ☺

SMS-Proxy

Eingabevalidierung?

Anmerkung: Analyse von HTTPS

"Just out of curiosity - did you setup ssl proxy with fake ssl certificate? didn't think it was possible."

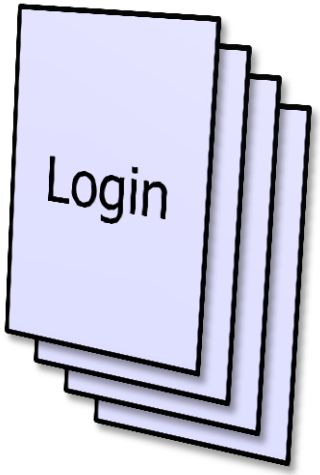
- Transportverschlüsselung stellt Vertraulichkeit und Integrität der übertragenen Daten sicher
- Schützt nicht die Backend-Schnittstelle vor Angriffen (Obscurity)

Häufige Designfehler – Beispiele

- Authentisierung/Autorisierung anhand statischer Merkmale
  - UDID, IMEI, IMSI etc. ✓
- Authentisierung/Autorisierung innerhalb der App
  - Manipulation des Transportwegs ✓
  - Manipulation interner App-Zustände

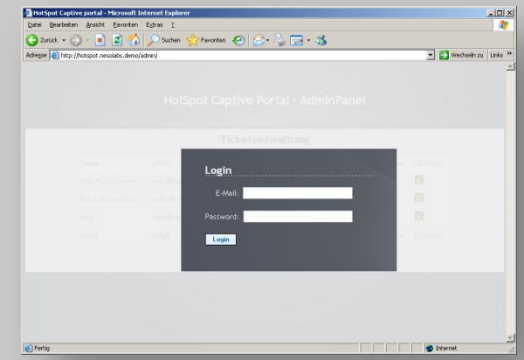
Häufige Probleme

- Sicherheitsmaßnahmen auf Client-Seite
  - **Beispiel:** Vorgelagerte Oberflächen zur Benutzeranmeldung verhindern Zugriff auf App-Funktionen



- App-Zustände können zur Laufzeit beliebig manipuliert werden

Manipulation interner App-Zustände

	AJAX	Mobile App (iOS)
Sprache	HTML / JavaScript	C / Objective-C
Werkzeuge	Firebug, Chrome Developer Tools etc.	gdb, Objective-C Runtime etc.
		

Manipulation interner App-Zustände

- Umfangreiche Objective-C Runtime [4] ermöglicht Manipulation von Apps zur Laufzeit
- Einbringen einer Bibliothek mit Debugging-Funktionalität in den Prozess der App (Library Injection)

Demonstration

Ausblenden einer Anmeldemaske
durch Manipulation interner App-Zustände

Manipulation interner App-Zustände

- Wo lag der Fehler?
  - Statischer „AppToken“ zur Kommunikation mit dem Backend
- Übertragbar auf andere Beispiele
  - Apps zur PIN/Passwortverwaltung
  - Dokumentenverwaltung
  - Etc.
- Erweiterbar auf beliebige client-seitige Logik
  - „Freischalten“ versteckter App-Funktionalität
  - Zugriff auf Code aus der Entwicklung
  - Etc.

Zusammenfassung

- Mobile Apps als alternatives Frontend
- Viele Probleme (und Lösungen!) aus der Browser-basierten Welt sind auf mobile Apps übertragbar
- Häufig noch Defizite in der Umsetzung

Quellen

[1] OWASP Mobile Security Project
https://www.owasp.org/index.php/OWASP_Mobile_Security_Project

[2] Gorilla, App zur Absicherung von Apple iOS
http://www.nesolabs.de/software/gorilla/

[3] Shooting the Messenger, Blogeintrag
http://www.andreas-kurtz.de/2011/09/shooting-messenger.html

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