

**Macoun**

# Sicher ganz einfach

Klaus M. Rodewig

*@cocoanehead*



# Letzte Tür vor der Autobahn

"PowerPoint slides are like children:  
No matter how ugly they are,  
you'll think they're beautiful if they're yours."

*Quelle: Anonym*

# Is -Ia

- Typische Fallstricke
- Methodik
- Implementierungsbeispiele



- ~~IT Security Analyst~~
- (iOS)-Entwickler
- Dezember: 2x Auflage 4



345 Seiten - für Programmierneinsteiger



1172 Seiten - für Fortgeschrittene

# Typische Fallstricke



# Wir werden alle störrben!!^



Quelle: Mediatest Digital

## Die TÜV Rheinland Datenschutzprüfung „Check your App“

Etwa 40 Prozent aller Apps lesen nach unseren Untersuchungen die Daten von mobilen Endgeräten aus, ohne dass der User bewusst zustimmt oder es überhaupt erfährt. Betroffen sind zum Beispiel Standortdaten, Passwörter, Kontaktdaten, Bilder. Unser Internetportal bietet privaten Nutzern eine kostenfreie Suchfunktion als App-Auswahlhilfe und den App-Anbietern eine Plattform um sich von Datendiebstahl zu distanzieren.

App-Auswahlhilfe und den App-Anbietern eine Plattform um sich von Datendiebstahl zu distanzieren.

etwa 40 Prozent aller Apps lesen nach unseren Untersuchungen die Daten von mobilen Endgeräten aus, ohne dass der User bewusst zustimmt oder es überhaupt erfährt. Betroffen sind zum Beispiel Standortdaten, Passwörter, Kontaktdaten, Bilder. Unser Internetportal bietet privaten Nutzern eine kostenfreie Suchfunktion als App-Auswahlhilfe und den App-Anbietern eine Plattform um sich von Datendiebstahl zu distanzieren.

Quelle: TÜV Rheinland

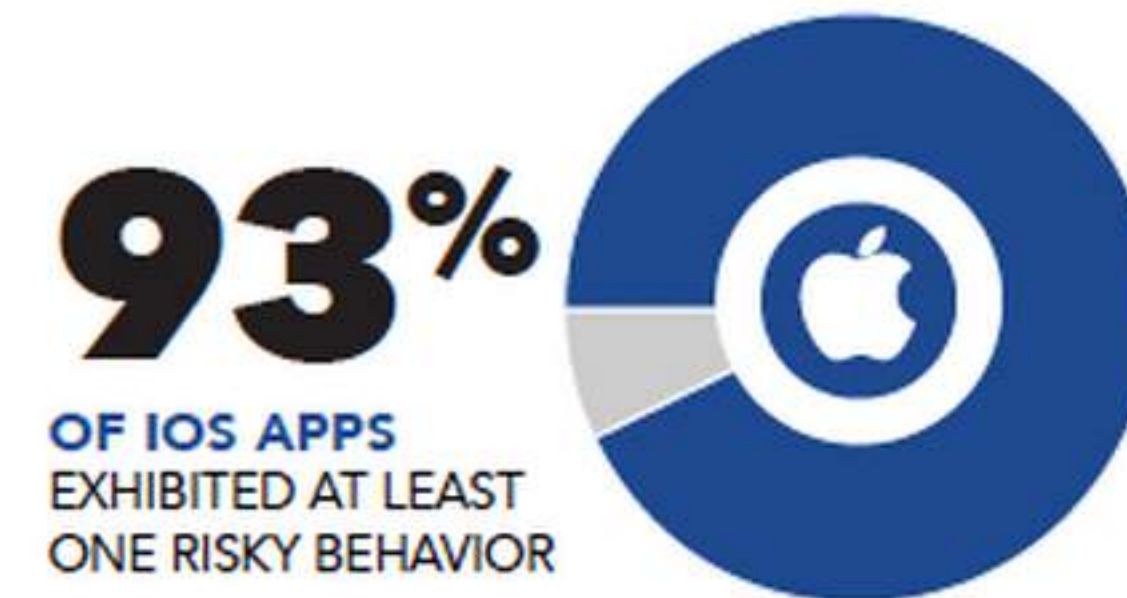
60 Prozent der beliebtesten iOS-Apps sind nicht für den Unternehmenseinsatz geeignet. Das haben Wissenschaftler des Fraunhofer-Instituts für Sichere Informationstechnologie im Rahmen einer Testreihe herausgefunden. Die Forscher prüften die beliebtesten kostenlosen Apps aller Kategorien aus Apples App Store und fanden teils gravierende Sicherheitslücken in der Programmierung: Bei rund 25 Prozent der Apps verzichteten die Entwickler absichtlich auf Schutzfunktionen und 12,5 Prozent der Apps verschicken Daten an mehr als fünf Unternehmen, die mit der eigentlichen App-Funktion nichts zu tun haben. Bei fünf Unternehmen, die mit der eigentlichen App-Funktion nichts zu tun haben. Bei absichtlich auf Schutzfunktionen und 12,5 Prozent der Apps verschicken Daten an mehr als fünf Unternehmen, die mit der eigentlichen App-Funktion nichts zu tun haben.

Quelle: Fraunhofer SIT

Für Detlev Henze, Geschäftsführer des Sicherheitsspezialisten TÜV TRUST IT GmbH, sind dies keineswegs Einzelfälle. Rund 45 Prozent der über 1.000 von seinem Unternehmen getesteten Apps übertragen Handy-Daten an spezielle Werbenetzwerke und Datensammler. Nach Angaben des Webservice flurry.com ist deren Analysefunktion heute bereits in rund 350.000 Apps auf über einer Milliarde Endgeräten implementiert. Seine Schlussfolgerung aus diesen Analysen ist eine generelle Kategorisierung der mobilen Anwendungen: „Apps sind entweder gut, hinterhältig oder ungewollt-gefährlich.“ Unter guten mobilen Anwendungen

worin sie zusammengefasst: „Apps sind entweder gut, hinterhältig oder ungewollt-gefährlich.“ Unter guten mobilen Anwendungen einer Milliarde Endgeräten implementiert. Seine Schlussfolgerung aus diesen Analysen ist eine generelle Kategorisierung der mobilen Anwendungen: „Apps sind entweder gut, hinterhältig oder ungewollt-gefährlich.“ Unter guten mobilen Anwendungen

Quelle: TÜV Austria



Quelle: Appthority

ONE RISKY BEHAVIOR  
EXHIBITED AT LEAST  
OF IOS APPS

22.09.2013 21:14

« Vorige | Nächste »

## Apples Touch ID des iPhone 5S schon gehackt UPDATE

Nur drei Tage nach dem Erscheinen von Apples iPhone 5S mit Fingerabdruck-Scanner gelang es dem Biometrie-Spezialisten "Starbug" ein iPhone mit nachgemachten Fingerabdrücken freizuschalten.

Fingerabdrücken freizuschalten.

gelang es dem Biometrie-Spezialisten "Starbug" ein iPhone mit nachgemachten

Quelle: Heise-Verlag



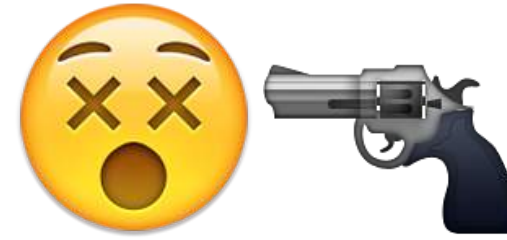
# Gründe

- User-Tracking
- Fehlende oder falsche Verschlüsselung
- falscher Umgang mit Berechtigungen
- sonstige Programmier- oder Hardwarefehler (aka “Diverses”)



# Ja und?

- Dateiverschlüsselung: physischer Zugriff auf Gerät notwendig
  - Code setzen oder TouchID verwenden
- Transportverschlüsselung: Zugriff auf Transportstrecke notwendig
  - fremde WLANs meiden
- User-Tracking: i. d. Regel wirtschaftlich notwendig



eHow » Internet » Web Design & Development » Web Development » How to Encrypt Credit Card Information in an SQL Database

# How to Encrypt Credit Card Information in an SQL Database

By Cristina Puno, eHow Contributor , last updated August 29, 2014

- 4 Design a script to encrypt the credit card information. Using an encryption algorithm of your choice, create a routine that converts the information into encrypted data. Programming languages such as PHP have built in functions that can encrypt. An example is the base 64 encryption function that can be used by invoking the following code:

```
base64_encode($credit_card_number);
```

Save this encrypted data in a separate variable.



Documentation — Security Starting Point for iOS

Q security

Security Starting Point

# Security Starting Point

Application security is about protecting users' information from being read, stolen, or destroyed by malicious people and processes. Security cannot be added to code as an afterthought; it must be built in. To keep your users' information secure, your iOS application must be resistant to attack and you must keep your users' data in a secure environment.

iOS security features are implemented at the Core OS level and its security APIs are at the Core Services level in the system architecture.

Cocoa Touch

Media

Core Services

Core OS

Security APIs

iOS security features

**Figure 1–1**Security APIs and system architecture

**Contents:**

- [Get Up and Running](#)
- [Become Proficient](#)
- [Download or Send Data Securely](#)

## Get Up and Running

For sample code that shows how to use the keychain to store passwords and other secrets, and how to share keychain items between applications, see [GenericKeychain](#).

For sample code that shows the use of the cryptographic functions found in the Security framework, see [CryptoExercise](#).

## Become Proficient

If you want to learn why and how to write secure code, read [Secure Coding Guide](#). That document explains the sources of security vulnerabilities in code and provides programming suggestions to help you write an application that will be resistant to attack. Then you can read [Security Overview](#) to learn about all the security APIs and features available in iOS.

Following that, read [Keychain Services Programming Guide](#) and [Certificate, Key, and Trust Services Programming Guide](#) to see more sample code and learn

Feedback

Documentation — NSN

Q pbkdf2

▼ NSNull Class Reference

Overview

▼ Tasks

▼ Obtaining an instance of NSNull

null

NSNull

Overview

Tasks

Obtaining an instance of NSNull

null

No Results

Following that, read [Keychain Services Programming Guide](#) and [Certificate, Key, and Trust Services Programming Guide](#) to see more sample code and learn

the security APIs and features available in iOS.

provides programming suggestions to help you write an application that will be resistant to attack. Then

## Download or Send Data Securely

To learn how to download data from a secure URL using the HTTPS protocol, or to send data securely over a network using a Secure Sockets Layer (SSL) or Transport Layer Security (TLS) data stream, see [CFNetwork Programming Guide](#).

To learn about secure data streams (TLS) data streams, see [CFNetwork Programming Guide](#).

# CommonCryptor.h

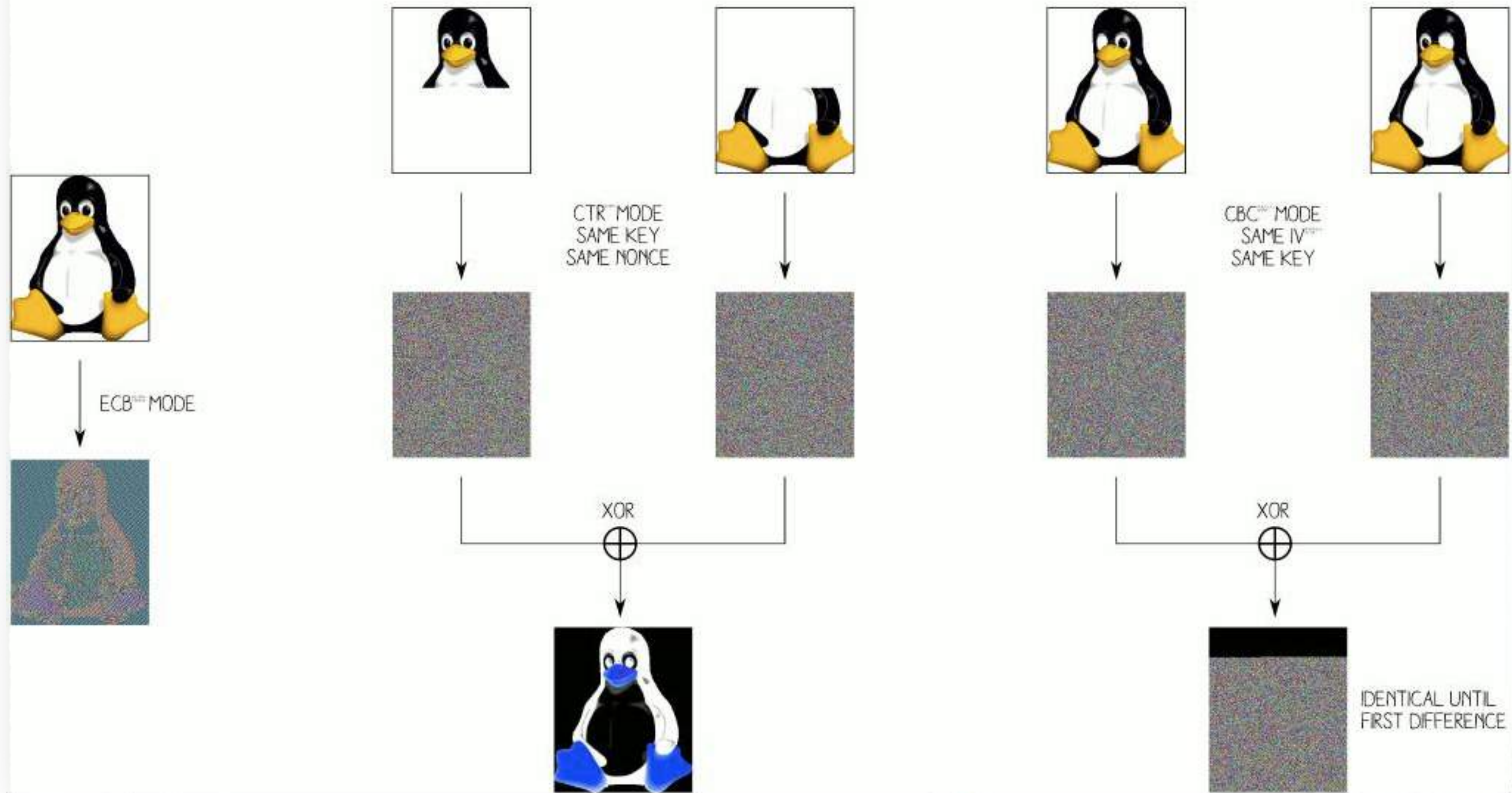
„Another option for block ciphers is Cipher Block Chaining, known as CBC mode. When using CBC mode, an Initialization Vector (IV) is provided along with the key when starting an encrypt or decrypt operation. If CBC mode is selected and no IV is provided, an IV of all zeroes will be used.,,

Quelle: CommonCryptor.h (Apple)



# Modes of operation's failures

ANGE ALBERTINI - CORKAMIC.COM  
JEAN-PIERRE AUMASSON  
VERSION 1.00  
2014/01/01





# OWASP Top 10

A1	Injection	SQL, LDAP, XML, OS-Commands
A2	XSS	Stored (persistent) XSS, reflected XSS, DOM-based XSS
A3	Broken authentication & session management	Predictable Session-IDs, password reset, password change, timeouts, etc.
A4	Insecure direct object references	z.B. Dateinamen als Parameter
A5	CSRF	Gleicher oder vorhersehbarer Token für alle User und Requests
A6	Security misconfigurations	Patchmanagement, Systemhärtung, Default-Passwörter, etc.
A7	Insecure cryptographic storage	Fehlende oder falsche Verwendung von Kryptographie
A8	Failure to restrict URL access	Fehlender Schutz vor unbefugtem Zugriff
A9	Insufficient transport layer protection	Verwendung von SSL, Secure Cookies, offizielles Zertifikat
A10	Unvalidated redirecty & forwards	Manipulation von Redirect-Zielen durch den Benutzer möglich

# CWE/SANS Top 25

ID	Description
<a href="#">M1</a>	Establish and maintain control over all of your inputs.
<a href="#">M2</a>	Establish and maintain control over all of your outputs.
<a href="#">M3</a>	Lock down your environment.
<a href="#">M4</a>	Assume that external components can be subverted, and your code can be read by anyone.
<a href="#">M5</a>	Use industry-accepted security features instead of inventing your own.
<a href="#">GP1</a>	(general) Use libraries and frameworks that make it easier to avoid introducing weaknesses.
<a href="#">GP2</a>	(general) Integrate security into the entire software development lifecycle.
<a href="#">GP3</a>	(general) Use a broad mix of methods to comprehensively find and prevent weaknesses.
<a href="#">GP4</a>	(general) Allow locked-down clients to interact with your software.

Quelle: <http://cwe.mitre.org/top25/>

<a href="#">GP4</a>	(general) Allow locked-down clients to interact with your software.
<a href="#">GP3</a>	(general) Use a broad mix of methods to comprehensively find and prevent weaknesses.

# Verschlüsselung

- Daten auf dem Endgerät gegen unbefugten Zugriff durch Dritte schützen?
- Daten auf dem Endgerät gegen unbefugten Zugriff durch den Benutzer schützen?

# Passwörter

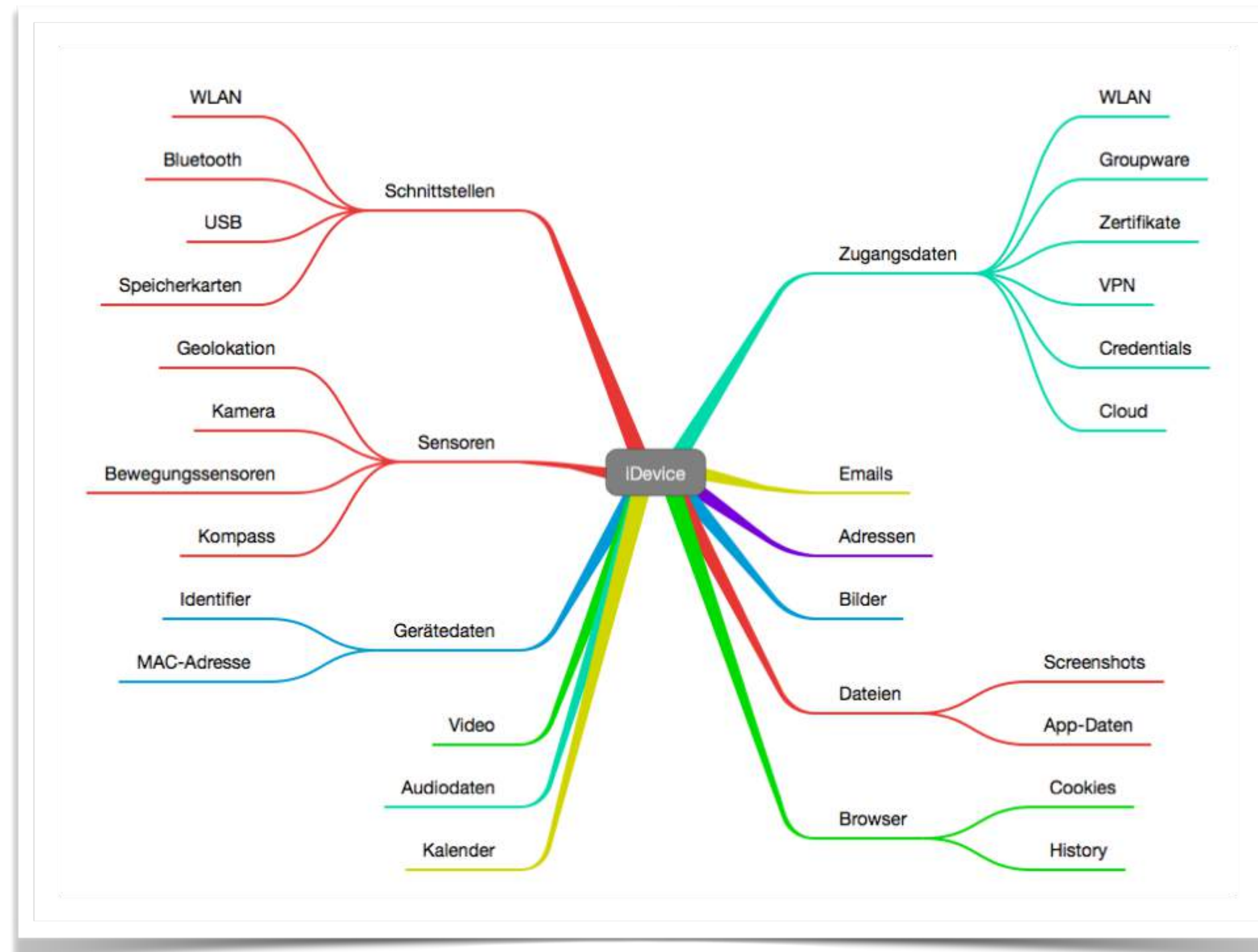
- Passwort für lokale Authentisierung?
- Passwort für Authentisierung am Server?

# Methodik

Warum?

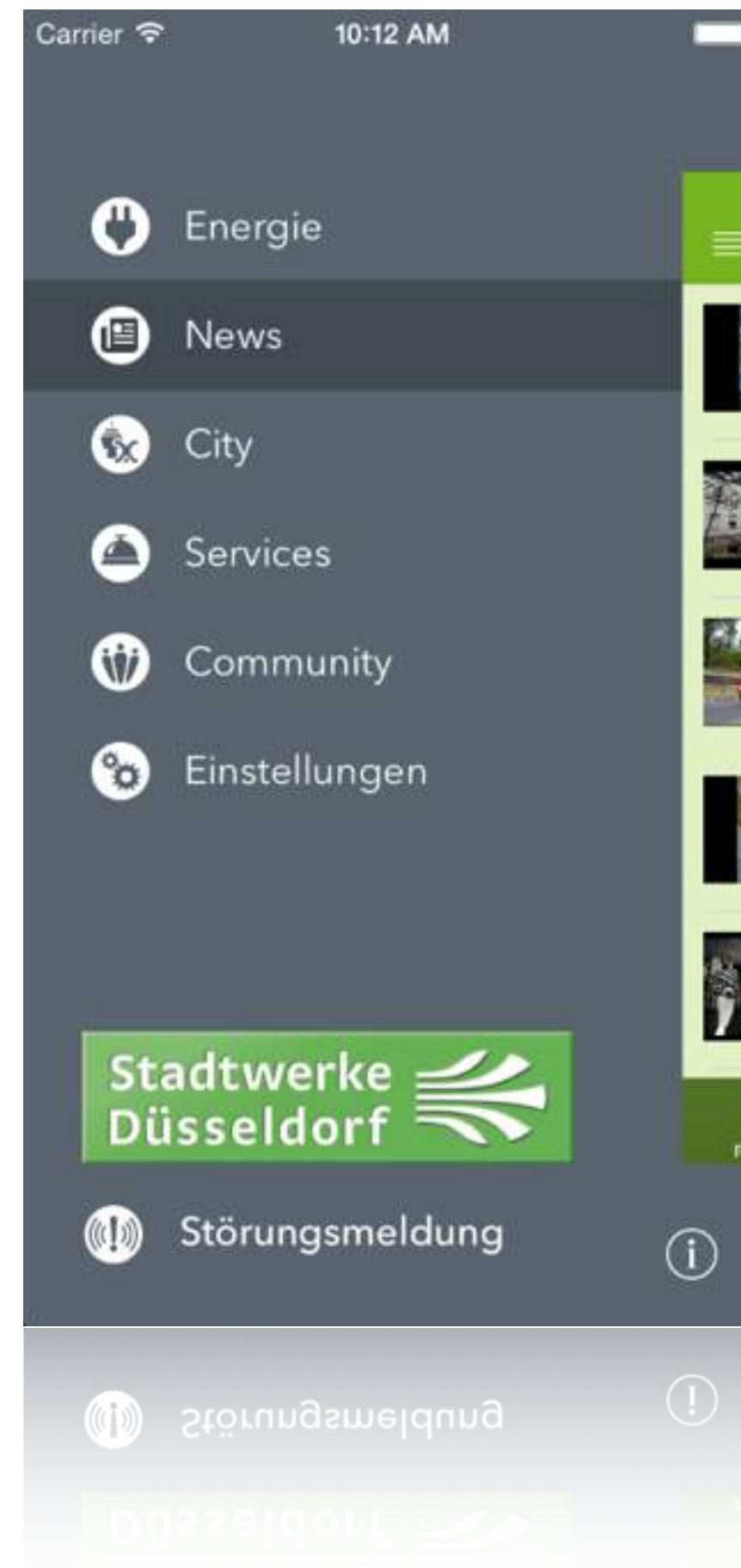


# Daten auf dem Smartphone



# Daten klassifizieren

- Nachrichten
- Vertragsverwaltung
- Zählerstandserfassung
- Smart Home
- Gewinnspiele
- Störungsmeldungen
- ...



anonyme Daten

personenbezogene Daten

sensible Daten



# STRIDE

Kategorie	B	Maßnahme
<u>S</u> poofing	▶ Vortäuschen	A
<u>T</u> ampering		
<u>R</u> epudiation		
<u>I</u> nformation disclosure		
<u>D</u> enial of service	▶ Störung	verschlüsselung
<u>E</u> levation of privileges	▶ Unbefugt	Robustheit
		Rollenmodell

23.09.2014 00:07

## Microsoft zerschlägt sein Sicherheits-Team

Im Zuge der vor einigen Monaten bekannt gemachten Stellenstreichungen will das Unternehmen seine Trustworthy Computing Group auflösen. Die verbleibenden Mitarbeiter wandern in andere Abteilungen.

« Vorige | Nächste »

# STRIDE with C

Kategorie	Beispiel	Maßnahme
<u>S</u> poofing	▶ Vortäuschen einer Identität	Authentisierung
<u>T</u> ampering	▶ Ändern von Daten	Integrität
<u>R</u> epudiation	▶ Abstreiten von Aktionen	Logging
<u>I</u> nformation disclosure	▶ Preisgabe von Informationen	Verschlüsselung
<u>D</u> enial of service	▶ Störung der Verfügbarkeit	Robustheit
<u>E</u> levation of privileges	▶ Unbefugte Rechteerweiterung	Rollenmodell
<u>C</u> ompliance	▶ Verstoß gegen Vorgaben	BDSG







# Bedrohungen (Auszug)

Kategorie	Beispiel	Maßnahme
<u>S</u> poofing	▶ Unbefugter Zugriff auf App	Benutzerauthentisierung / DS-Erklärung
<u>T</u> ampering	▶ Verändern der Datenbank	Eingabevalidierung App und Datenbank
<u>R</u> epudiation	▶ Brute-Force auf Login (huhu, 🍏)	Gestaffelte BF-Sperre (sonst DoS)
<u>I</u> nformation disclosure	▶ Verschlüsselung von Dateien	Verschlüsselung App / Datenbank
<u>D</u> enial of service	▶ Absturz durch “falsche” Daten	Fehlerbehandlung im JSON- und XML-Parser
<u>E</u> levation of privileges	▶ XSS in Community	Eingabe- und Ausgabevalidierung (App + DB)
<u>C</u> ompliance	▶ Backup in iCloud	personenbezogene Daten ausschließen

# Implementierung



# Datenschutzerklärung

15.00 Uhr

Thomas Biedorf

## Datenschutz und Apps

Die Vorgaben des BDSG und Telemediengesetz dürften den meisten eher unbekannt sein. Ich zeige, welche Paragraphen relevant sind und wie die Aufsichtsbehörden das ein oder andere bewerten. Da zur Zeit bußgeldbewehrte Bescheide der Aufsichtsbehörden verschickt werden, sicherlich nicht ganz unwichtig.

Quelle: [www.macoun.de](http://www.macoun.de)

werden, sicherlich nicht ganz unwichtig.  
Bescheide der Aufsichtsbehörden verschickt

- Welche Daten?
- Wofür?
- Wohin?
- Wie lange?
- Wer?

# SQL Injection

```
NSString *user = userName.text;  
NSString *pass = userPasswort.text;  
NSString *query = [NSString stringWithFormat:  
    @"select * from tbl.user where user='%@'and pass='%@'", user, pass];
```

# SQL Injection

```
user = @"foo";  
pass = @"bar";
```

```
—
```

```
user = @"foo'; OR 1=1; --";  
user = @"foo'; DROP TABLE members; --";
```



# Maßnahmen

- Benutzereingaben bereinigen
- Prepared Statements verwenden
- Least privilege bei der Rechtevergabe (Defense in depth)
- Daten in Datenbank sicher speichern (z.B. Passwort-Hashes)

# XSS

```
NSString *user = @"foo";
```

—

```
NSString *user = @"<script>alert(23)</script>";
```

# Maßnahme

```
NSString *sanitizedString =[user  
stringByAddingPercentEscapesUsingEncoding:NSUTF8StringEncoding];
```

Bei Eingabe und Ausgabe!

# Dateisystem

```
...  
[theData writeToFile:  
[[NSFileManager defaultManager] preferencesFileNameWithExtension:@"json"]  
options:NSDataWritingFileProtectionComplete  
error:&theError])  
...
```

# Backup

- BDSG
- iCloud != EU
- NSFileProtection?



# !(Backup)

Deployment-Target  
beachten!

```
...
NSURL *thePreferencesFile = [NSURL fileURLWithPath:
[[NSFileManager defaultManager] preferencesFileNameWithExtension:@"json"]];

NSError *theAttributeError;

BOOL setBackupAttribute = [thePreferencesFile
    setResourceValue:[NSNumber numberWithBool:YES]
    forKey:NSURLIsExcludedFromBackupKey error:&theAttributeError];
...
```

# Backend-Kommunikation

Session 3, Samstag

Alex von Below, Pepi Zawodsky

## NSURLConnection: Safety First!

Diese Session soll die Grundlagen der Transport Layer Security im Allgemeinen und NSURLConnection im besonderen auffrischen, und fortgeschrittene Techniken bei der Verwendung von NSURLConnection zeigen. Dazu werden mögliche Fehler und ihre Folgen durch einen live Man-In-The-Middle Attack verdeutlicht.

Quelle: [www.macoun.de](http://www.macoun.de)

In-The-Middle Attack verdeutlicht.

zeigen. Dazu werden mögliche Fehler und ihre Folgen durch einen live Man-In-The-Middle Attack verdeutlicht.

- TLS
- Certificate Pinning

# NSURLSession

```
if ([[challenge protectionSpace] authenticationMethod]
isEqualToString:NSURLAuthenticationMethodServerTrust]) {
    SecTrustRef serverTrust = [[challenge protectionSpace] serverTrust];
    BOOL trustedCert = NO;
    NSData *theData = [NSData new];
    if(serverTrust != NULL) {
        CFIndex theCertCount = SecTrustGetCertificateCount(serverTrust);
        for(CFIndex theCertIndex = 0; theCertIndex < theCertCount; theCertIndex++) {
            SecCertificateRef theCert =
                SecTrustGetCertificateAtIndex(serverTrust, theCertIndex);
            theData = (__bridge_transfer NSData *)SecCertificateCopyData(theCert);
            if([kEAPSecurityCloudHash caseInsensitiveCompare:[self sha1HexDigest:theData]]
== NSOrderedSame){
                trustedCert = YES;
                break;
            } else {
                trustedCert = NO;
            }
        }
    }
}
```



# Reverse Engineering

§ 69e UrhG: “Die Zustimmung des Rechtsinhabers ist nicht erforderlich, wenn ...”

- ... vom Urheber ermächtigt,
- ... Informationen nicht zugänglich,
- ... auf funktional und vom Umfang Interoperabilität beschränkt.



• • •

Bei Handlungen nach Abs. 1 gewonnene Informationen dürfen nicht

1. zu anderen Zwecken als zur Herstellung der Interoperabilität des unabhängig geschaffenen Programms verwendet werden,

2. an Dritte weitergegeben werden, es sei denn, dass dies für die Interoperabilität des unabhängig geschaffenen Programms notwendig ist

...



“Beim Reverse Engineering, ohne hierbei die Zustimmung des Urhebers zu haben und ohne in den Anwendungsbereich des § 69e UrhG zu fallen, kann der Urheber auf Unterlassung und Schadensersatz in Anspruch nehmen.”

# User-Tracking

- Flurry
- Google Analytics
- Loggr
- ...
- BDSG
- sehr invasiv

# Anforderung User-Tracking

- Anzahl Installationen
- tatsächliche Nutzung der App
- Nutzung von Funktionen
- SHA256-Hash des IDFV
- Start der App
- Beenden der App
- jede Unterfunktion
- keine Aggregation



```

-(NSString *)generateSHA256:(NSString *)inputString{
    NSMutableString *theHash = [NSMutableString stringWithCapacity:CC_SHA256_DIGEST_LENGTH];
    unsigned char passwordChars[CC_SHA256_DIGEST_LENGTH];
    CC_SHA256([inputString UTF8String],
    (int)[inputString lengthOfBytesUsingEncoding:NSUTF8StringEncoding], passwordChars);
    for(int i=0; i< CC_SHA256_DIGEST_LENGTH; i++){
        [theHash appendString:[NSString stringWithFormat:@"%02x", passwordChars[i]]];
    }
    return theHash;
}

--

self.identifierForVendor = [[[UIDevice currentDevice] identifierForVendor] UUIDString];
NSString *theIdentifierHash = [self generateSHA256:self.identifierForVendor];

--

theIDFVURLString = [NSString stringWithFormat:@"%@@?idfv=%@&status=%@&bundleVersion=%@&bundleShortVersion=
%@&bundleIdentifier=%@&os=%@", kURLEapIDFVCloud, theIdentifierHash, inStatus, bundleVersion,
bundleShortVersion, bundleIdentifier, @"iOS"];

```

# Passwörter

```
#define CUSTOMER_ID      @"lolwut"  
#ifdef STAGING  
#define API_PWD          @"lolwut2012"
```

- statisches Passwort
- Passwort im Binary



```
#define CUSTOMER_ID      @"lolwut"
#ifdef STAGING
//#define API_PWD        @"lolwut2012"
#define API_PWD_ENC      [NSArray arrayWithObjects:
                           [NSNumber numberWithInt:109],
                           [NSNumber numberWithInt:112],
                           [NSNumber numberWithInt:109],
                           [NSNumber numberWithInt:120],
                           [NSNumber numberWithInt:118],
                           [NSNumber numberWithInt:117],
                           [NSNumber numberWithInt:51],
                           [NSNumber numberWithInt:49],
                           [NSNumber numberWithInt:50],
                           [NSNumber numberWithInt:52],
                           nil]
#define API_PWD          [MSSecurityUtils
mswagDecryptString:API_PWD_ENC]
```

# Generische Maßnahmen

- Compiler-Flags
- Diverses
- Kryptographie



# Compiler-Flags

## ▼ Apple LLVM 6.0 – Custom Compiler Flags

Setting

 EAP

**Other Warning Flags**

**-Werror -Wall -Wunreachable-code**

Other Warning Flags

-Werror -Wall -Wunreachable-code

“To tear down the *Wall* would be a *Werror*.”  
(Amin Negm-Awad)

# Diverses

```
int foo = INT_MAX;  
foo++;
```

```
--
```

```
NSNumber *foobar = [NSNumber numberWithInt:257];  
NSLog(@"foobar: %i", [foobar charValue]);
```

# Kryptographie

- NSFileProtection + kSecAttr\* + Geräteverschlüsselung
  - AES 256
  - PAL
- Technische Richtlinie *BSI TR-02 / 02-1 / -2*  
“Kryptographische Verfahren: Empfehlungen und Schlüssellängen”

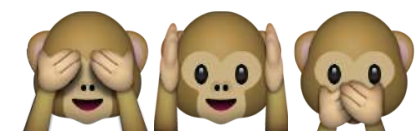
# Snakeoil

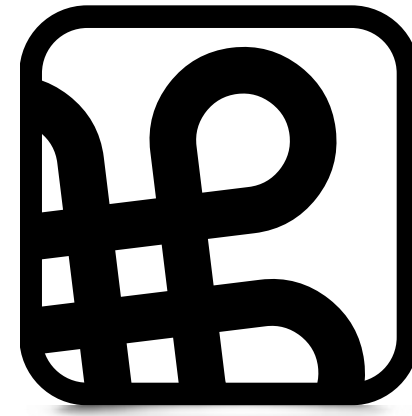
- App-Siegel
- Bandbreite von einmalig 1.000,- Euro bis 10.000,- Euro p.a.
- Prüfmethoden: Selbstauskunft, Laufzeitanalyse, Code Audit
- Snakeoil kann durchaus helfen



# Fragen?

No animals were harmed for this presentation.





**Macoun**