Einführung in die iOS Sicherheitswelt

Dresdner OWASP-Stammtisch 17.10.2013

Johannes Greiner

1. Programmieren im iApfel-Land

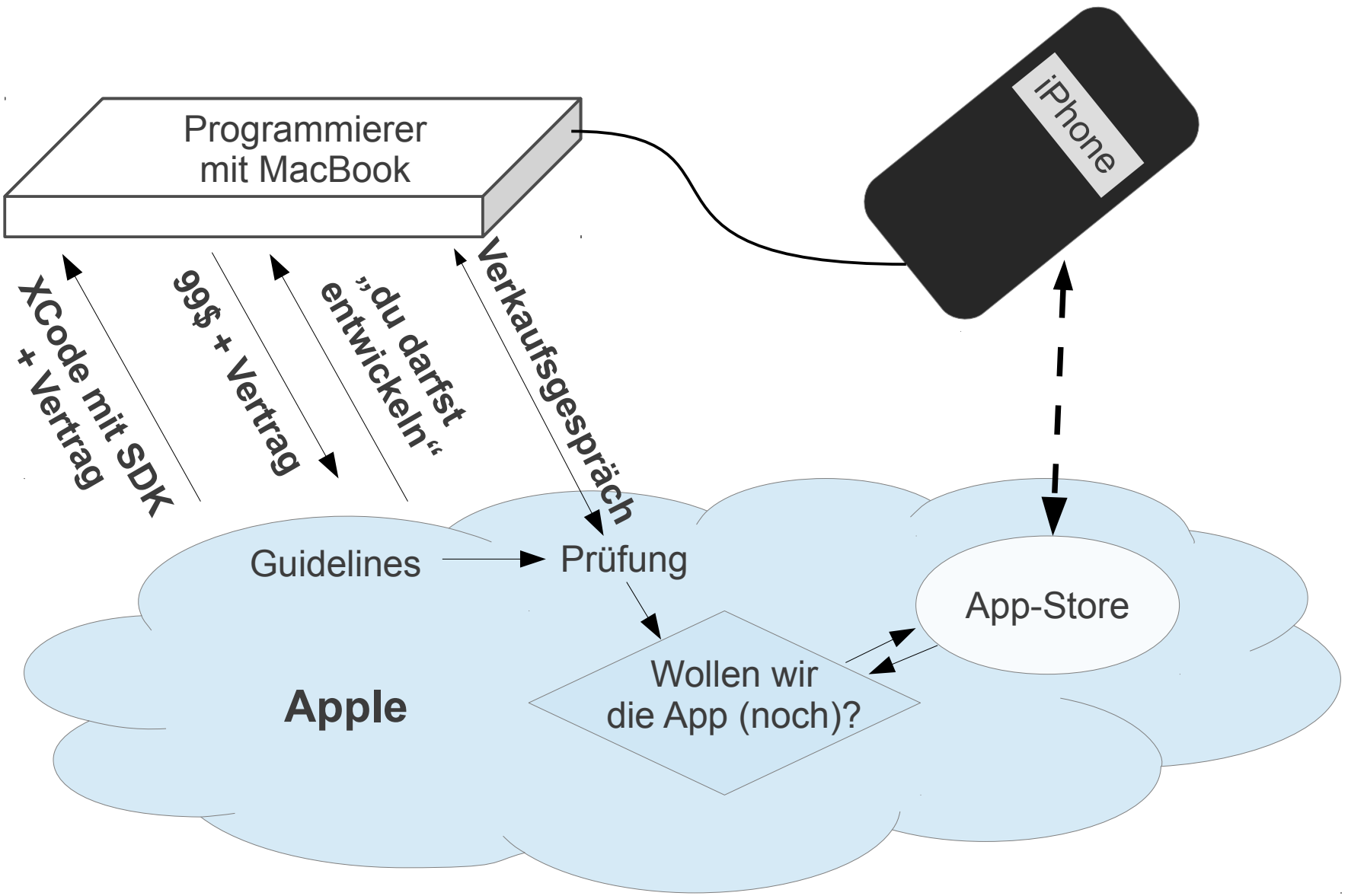
2. Sicherheitsmechanismen von iOS

3. Penetration-Testing von iOS

4. Kleine Nachspeise

1. Programmieren im iApfel-Land

Wenn ich auch ein Stück vom Apfelkuchen will …



Leitgedanken

- Apps dürfen nur was Apple erlaubt. Beispiele:
  - nur in engen Grenzen mit anderen Apps oder dem System interagieren
  - möglichst nichts im Hintergrund tun
  - einheitliche Design-Elemente
  - möglichst viel über System-Schnittstellen abwickeln
- Entwicklertools und Deployment komplett in Apples Hand. „Nutzer werden von Apple vor dem Programmierer geschützt.“
- Ausnahmen wenn Apple es will
- Dokumentation von Features wenn Apple es will

2. Sicherheitsmechanismen von iOS

Überblick

- UNIX > **BSD** > NeXTStep > Darwin > **Mac OS X** > iOS
- Nutzer `mobile` und `root`
- Code Signing
- Keychain
- Boot Architektur
- verschlüsseltes Dateisystem
- Sandboxing

Sandboxing

- jedes Programm hat eigenen virtuellen Adressraum
- direktes Schreiben von Dateien nur im eigenen Verzeichnis
- Zugriff auf Systemressourcen nur über APIs
- Nutzerabfrage bei Zugriff auf APIs
- genaue, individuelle Zugriffssteuerung theoretisch möglich

Keychain

- secure storage container für sensitive Strings
- Zugriff nur auf eigene Daten (bzw. AccessGroup)
- für Nutzer unsichtbar
- speichert Daten mit Attributen
- auch im Backup verschlüsselt
- Entschlüsselungszeitpunkt für jeden Eintrag separat einstellbar

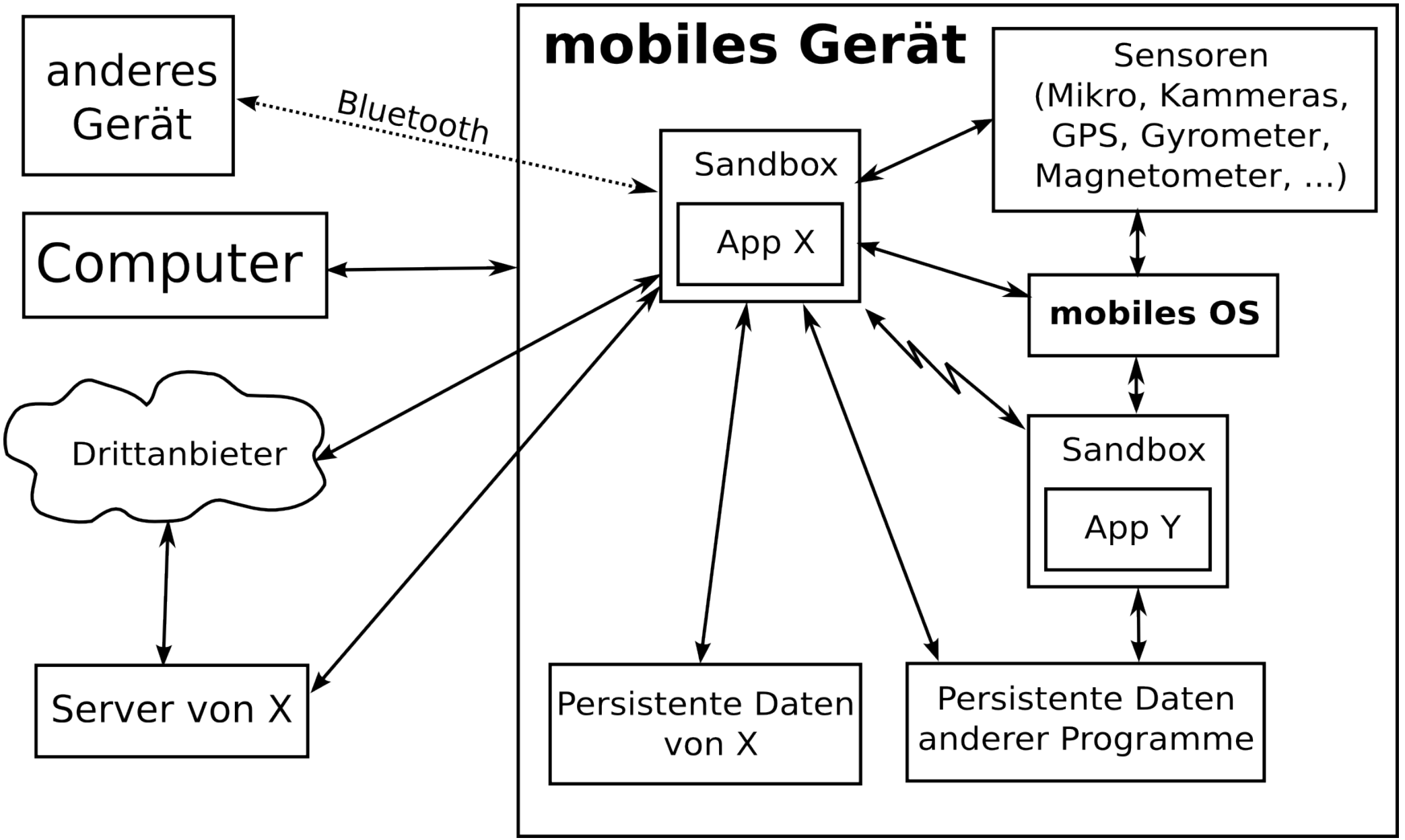
```
kSecAttrAccessibleWhenUnlockedThisDeviceOnly
kSecAttrAccessibleAfterFirstUnlock
kSecAttrAccessibleAlways
```

3. Penetration-Testing von iOS

Was kann denn da überhaupt noch was schief gehen?

Beispiele:

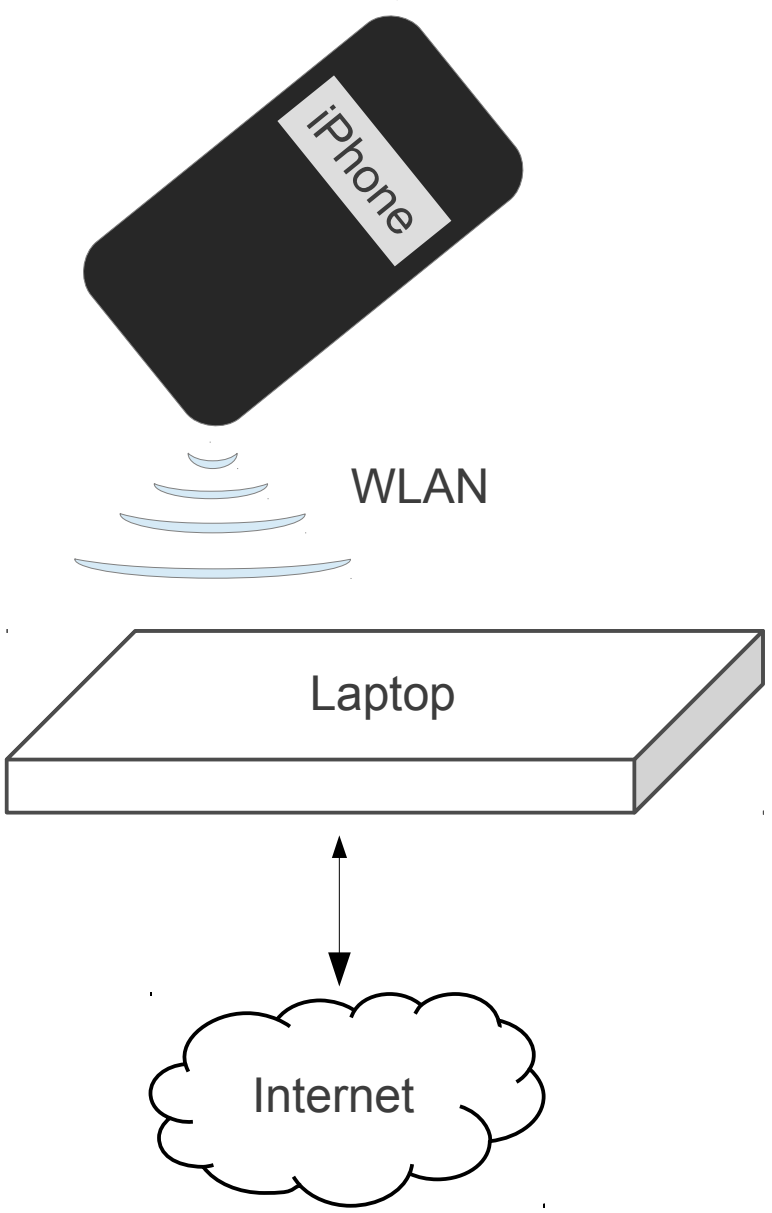
- Datenübertragung beliebig unsicher sein
- Ausspähen des Nutzers (z.B. Mikrofon – Nutzer wird nicht informiert!)
- unverschlüsseltes Ablegen von Daten (z.B. Cookies, Keychain nicht benutzen, ...)
- Abfluss sensitiver Daten über die Autovervollständigung
- Snapshots von Bildschirm bevor die App in den Hintergrund geht
- bewusstes Austricksen von Apples Sicherheitscheck damit die App auch bei falschem Zertifikat funktioniert


anderes Gerät
Bluetooth
mobiles Gerät
Sensoren (Mikro, Kammeras, GPS, Gyrometer, Magnetometer, ...)
Sandbox
App X
Computer
mobiles OS
Drittanbieter
Sandbox
App Y
Server von X
Persistente Daten von X
Persistente Daten anderer Programme

Generelles zum Testing

- Tools auf OWASP iOS CheatSheet
- sehr viel händisches und tiefgreifendes Testen
- Jailbreak ist Grundvoraussetzung für fast alles was die Interna der App betrifft
- rechtliche Fragen nicht eindeutig

Beispielsetup



Jailbreak + SSH + Apps

+ Testtools

Netzwerk Proxy + SSH
Konsole + Zugriffs-Tools

Beispieltool `Cycript`

Spaß mit MobileNotes:

1. App öffnen und Prozess-ID bestimmen

```
ps aux | grep Notes
```

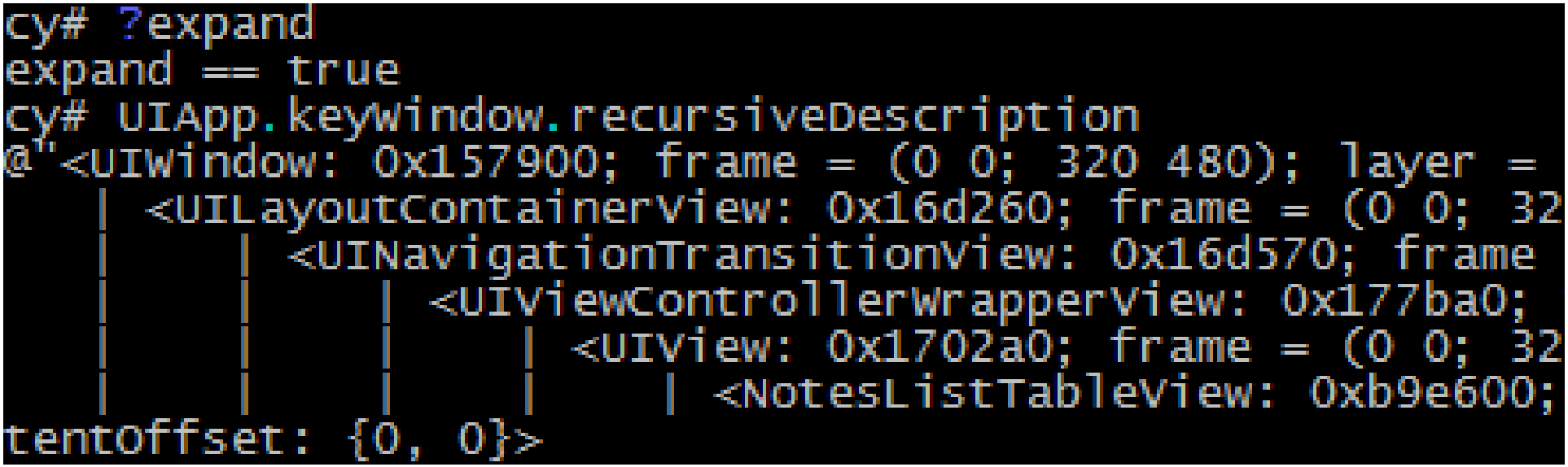
2. cycript einhängen

```
cycript -p 488
```

3. Komposition des Fensters ansehen

```
?expand
UIApp.keyWindow.recursiveDescription
```

```
cy# ?expand
expand == true
cy# UIApp.keyWindow.recursiveDescription
@"<UIWindow: 0x157900; frame = (0 0; 320 480); layer = <CALayer: 0x1579e0>>
   | <UILayoutContainerView: 0x16d260; frame = (0 0; 320 480); autoresize = W+H; layer = <CALayer: 0x16d300>>
   |    | <UINavigationTransitionView: 0x16d570; frame = (0 0; 320 480); clipsToBounds = YES; autoresize = W+H; layer = <CALayer: 0x16d610>>
   |    |    | <UIViewControllerWrapperView: 0x177ba0; frame = (0 64; 320 416); autoresize = W+H; layer = <CALayer: 0x178790>>
   |    |    |    | <UIView: 0x1702a0; frame = (0 0; 320 416); autoresize = W+H; layer = <CALayer: 0x132d50>>
   |    |    |    |    | <NotesListTableView: 0xb9e600; baseClass = UITableView; frame = (0 0; 320 416); clipsToBounds = YES; opaque = NO; autoresize = W+H; layer = <CALayer: 0x170d70>; con
tentOffset: {0, 0}>
   |    |    |    |    |    | <UIImageView: 0x172190; frame = (0 44; 320 372); opaque = NO; autoresize = W+H; userInteractionEnabled = NO; layer = <CALayer: 0x1725c0>>
   |    |    |    |    |    | <NoteCell: 0x188740; baseClass = UITableViewCell; frame = (0 99; 320 44); opaque = NO; autoresize = W; layer = <CALayer: 0x188890>>
   |    |    |    |    |    |    | <UITableViewCellContentView: 0x1888c0; frame = (0 0; 290 43); layer = <CALayer: 0x1888f0>>
   |    |    |    |    |    |    |    | <UILabel: 0x188920; frame = (10 6; 224 28); text = 'Fthgfg'; clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x17f5b
0>>
   |    |    |    |    |    |    |    | <UILabel: 0x188a00; frame = (244 12; 36 18); text = '15:31'; clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x17f5e
0>>
   |    |    |    |    |    |    |    | <UIView: 0x188c00; frame = (290 0; 30 44); layer = <CALayer: 0x188ae0>>
   |    |    |    |    |    |    |    |    | <UIImageView: 0x188b10; frame = (9 16; 9 13); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x188b40>>
   |    |    |    |    |    |    |    | <UIView: 0x189210; frame = (0 43; 320 1); autoresize = W+TM; layer = <CALayer: 0x189240>>
   |    |    |    |    |    | <NoteCell: 0x17ebe0; baseClass = UITableViewCell; frame = (0 44; 320 55); opaque = NO; autoresize = W; layer = <CALayer: 0x17ed50>>
   |    |    |    |    |    |    | <UITableViewCellContentView: 0x17ef80; frame = (0 0; 290 54); layer = <CALayer: 0x17f140>>
   |    |    |    |    |    |    |    | <UILabel: 0x17f7e0; frame = (10 17; 224 28); text = 'Buffosuc'; clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x17
f6d0>>
   |    |    |    |    |    |    |    | <UILabel: 0x1820d0; frame = (244 23; 36 18); text = '15:59'; clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x18221
0>>
   |    |    |    |    |    |    |    | <UIView: 0x182140; frame = (290 0; 30 55); layer = <CALayer: 0x1821c0>>
   |    |    |    |    |    |    |    |    | <UIImageView: 0x182240; frame = (9 27; 9 13); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x182700>>
   |    |    |    |    |    |    |    | <UIView: 0x189180; frame = (0 54; 320 1); autoresize = W+TM; layer = <CALayer: 0x1891b0>>
   |    |    |    |    |    | <UIImageView: 0x136800; frame = (0 44; 320 11); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x172df0>>
   |    |    |    |    |    | <UIImageView: 0x173b40; frame = (0 411; 320 5); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x173f20>>
   |    |    |    |    |    | <UILabel: 0x174c80; frame = (93 142; 134 24); text = 'Keine Notizen'; clipsToBounds = YES; alpha = 0; opaque = NO; userInteractionEnabled = NO; layer = <CALaye
r: 0x174d40>>
   |    |    |    |    |    | <NotesSearchBar: 0x1754f0; baseClass = UISearchBar; frame = (0 0; 320 44); autoresize = W; layer = <CALayer: 0x1755c0>>
   |    |    |    |    |    |    | <UISearchBarBackground: 0x178960; frame = (0 0; 320 44); layer = <CALayer: 0x1789f0>>
   |    |    |    |    |    |    | <UISearchBarTextField: 0x175a90; frame = (5 7; 310 31); clipsToBounds = YES; opaque = NO; layer = <CALayer: 0x175c10>>
   |    |    |    |    |    |    |    | <UITextFieldBorderView: 0x188570; frame = (0 0; 310 31); opaque = NO; layer = <CALayer: 0x183510>>
   |    |    |    |    |    |    |    | <UIImageView: 0x175850; frame = (10 8; 15 15); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x175880>>
   |    |    |    |    |    |    |    | <UITextFieldLabel: 0x178a50; frame = (32 7; 246 18); text = 'Suchen'; clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer = <CALayer
: 0x175ec0>>
   |    |    |    |    |    |    | <UIImageView: 0x178d90; frame = (0 -1; 320 1); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x175260>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x17de80; frame = (0 186; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x17e480>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x17dd50; frame = (0 230; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x16e530>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x132d00; frame = (0 274; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x17db90>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x17e430; frame = (0 318; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x133220>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x17e4e0; frame = (0 362; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x17e520>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x179b70; frame = (0 406; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x17d470>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x17a5e0; frame = (0 450; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x17a620>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x1356c0; frame = (0 494; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x135700>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x135730; frame = (0 538; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x135770>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x17e620; frame = (0 582; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x1357a0>>
   |    |    |    |    |    | <_UITableViewSeparatorView: 0x17e660; frame = (0 626; 320 1); opaque = NO; autoresize = W; layer = <CALayer: 0x17e6a0>>
   |    |    | <NotesNavigationBar: 0x159f10; baseClass = UINavigationBar; frame = (0 20; 320 44); autoresize = W; layer = <CALayer: 0x15a010>>
   |    |    |    | <UINavigationItemView: 0x16f9d0; frame = (108 8; 105 27); opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x16fa70>>
   |    |    |    | <UINavigationButton: 0x17a3e0; frame = (282 7; 33 30); opaque = NO; layer = <CALayer: 0x17a520>>
   |    |    |    |    | <UIImageView: 0x17e1a0; frame = (0 0; 33 30); clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x17e1d0>>
   |    |    |    |    | <UIImageView: 0x17e250; frame = (10 8; 13 14); clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x17e280>>
   |    |    |    |    | <UIButtonLabel: 0x17a550; frame = (0 0; 0 0); clipsToBounds = YES; hidden = YES; opaque = NO; userInteractionEnabled = NO; layer = <CALayer: 0x17a4c0>>"
```

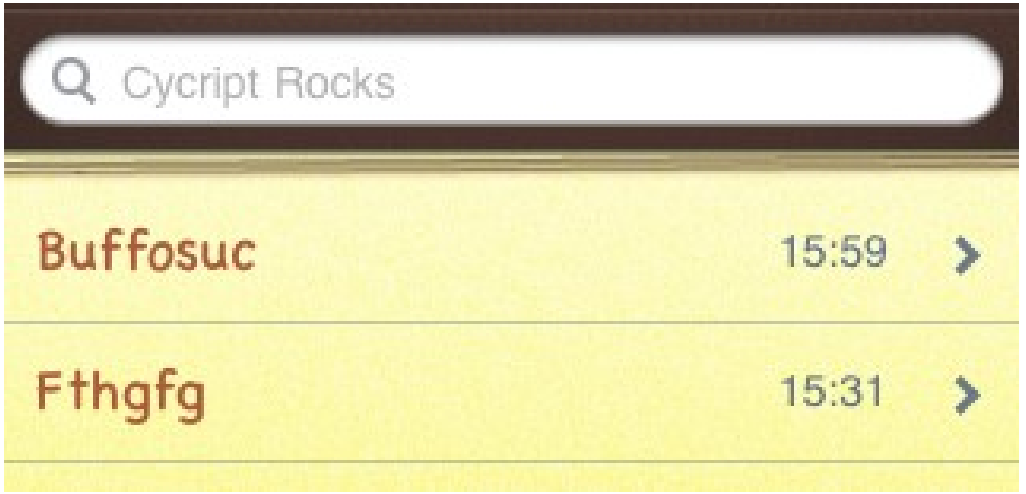


4. Suche NotesSearchBar -> UITextFieldLabel

```
<NotesSearchBar: 0x1754f0; baseClass = UISearchBar; frame = (0 0; 320 44); auto
   | <UISearchBarBackground: 0x178960; frame = (0 0; 320 44); layer = <CALayer:
   | <UISearchBarTextField: 0x175a90; frame = (5 7; 310 31); clipsToBounds = YE
   |    | <UITextFieldBorderView: 0x188570; frame = (0 0; 310 31); opaque = NO;
   |    | <UIImageView: 0x175850; frame = (10 8; 15 15); opaque = NO; userInter
   |    | <UITextFieldLabel: 0x178a50; frame = (32 7; 246 18); text = 'Suchen';
```

5. Manipuliere Textfeld-Eigenschaft

```
var st = new Instance(0x178a50)
st.text=@"Cycript Rocks"
```



6. Methoden der NoteCell Klasse ausgeben

`printMethods("NoteCell")`

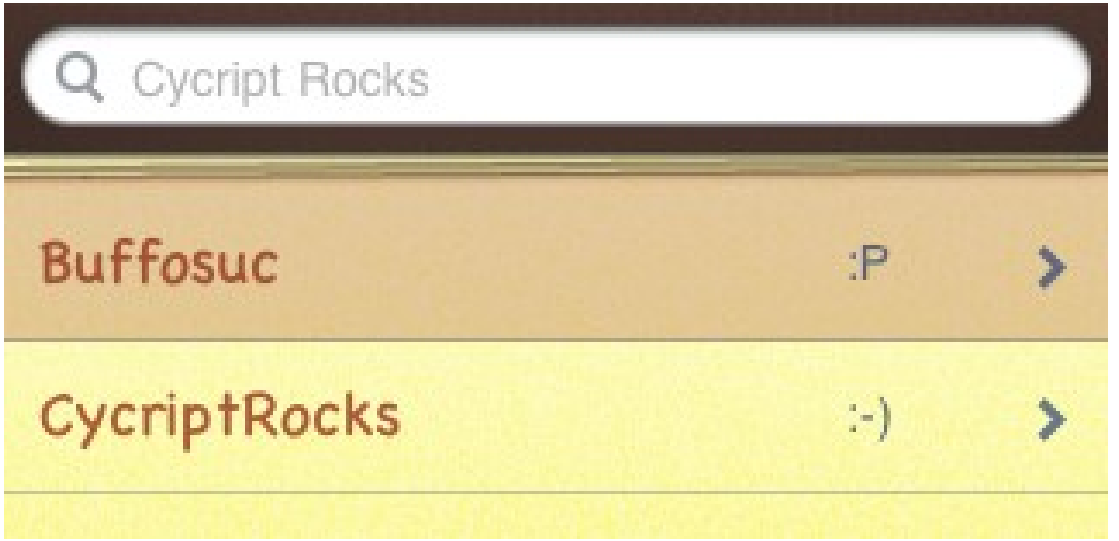
```
[{selector:@selector(setUseAlternateTextColor:),implementation:0x1996
8},
{selector:@selector(hideDate:),implementation:0x19938},
{selector:@selector(updateTitleFont),implementation:0x19cf8},
{selector:@selector(useAlternateTextColor),implementation:0x198bc},
{selector:@selector(_dateTextColor:),implementation:0x19b08},
{selector:@selector(_titleTextColor:),implementation:0x19c00},
{selector:@selector(setSummary:),implementation:0x199ec},
{selector:@selector(setContainsCJK:),implementation:0x199b0},
{selector:@selector(layoutSubviews),implementation:0x1a2e4},
{selector:@selector(dealloc),implementation:0x19e50},
{selector:@selector(setTitle:),implementation:0x19abc},
{selector:@selector(setDate:),implementation:0x19a48},
{selector:@selector(_automationID),implementation:0x198d0},
{selector:@selector(_automationValue),implementation:0x198a8},
{selector:@selector(initWithStyle:reuseIdentifier:),implementation:0x
19f20}]
```

7. Methode `setTitle:` benutzen

```
var nc1 = new Instance(0x188740)
[nc1 setTitle:@"CycriptRocks"]
```

8. Auch die Uhrzeit ist nur ein UILabel

```
var uil = new Instance(0x1820d0)
uil.text=@":-)"
```



9. Noch mehr Unsinn

```
printMethods("UIView")
[...] setRotationDegrees:duration: [...]

var uiv = new Instance(0x179dc0)
[uiv setRotationDegrees:-45 duration:1]
```

4. Kleine Nachspeise

Evasi0n 6.1 JB - Getting In

- nutze Bug im Backup-System um Zugriff auf Zeitzonen-Datei zu bekommen
- Füge symlink auf Socket ein, welcher für die Kommunikation mit `launchd` zuständig ist (`launchd` hat root-Rechte)
- nutze `launchd` und eine manipulierte leere App um mit `remount` das ganze Dateisystem schreibbar zu machen
- verändere `launchd.conf` um Veränderung persistent zu machen

- verändere `MobileFileIntegrity` um Code-Signing zu umgehen (einschleusen einer Bibliothek, die bekannte Methoden unter anderem Namen exportiert und sonst leer ist)
- ASLR umgehen:
  - finde ARM Exception Vector (architekturbasiert schwer zu verstecken)
  - provoziere Fehler im Kernel um im Exception Vector die Adresse des Kernels im Speicher herauszubekommen
- nutze weiteren USB-Bug, um auf eine vom Nutzer (zurück)gegebene Speicheradresse zu schreiben (Kernel ade!)

Quellen zum Vortrag

- http://www.forbes.com/sites/andygreenberg/2013/02/05/inside-evasi0n-the-most-elaborate-jailbreak-to-ever-hack-your-iphone/
- http://blog.accuvantlabs.com/blog/bthomas/evasi0n-jailbreaks-userland-component
- https://www.owasp.org/index.php/IOS_Application_Security_Testing_Cheat_Sheet
- https://www.isecpartners.com/media/12985/secure_development_on_ios.pdf
- https://developer.apple.com/library/ios/documentation/iphone/conceptual/iphoneosprogrammingguide/TheiOSEnvironment/TheiOSEnvir
- http://securityevaluators.com/files/papers/apple-sandbox.pdf
- https://developer.apple.com/library/ios/documentation/Security/Reference/keychainservices/Reference/reference.html#//apple_ref/doc/u
- https://developer.apple.com/library/ios/documentation/Security/Conceptual/keychainServConcepts/02concepts/concepts.html#//apple_re
- https://developer.apple.com/library/ios/documentation/Security/Conceptual/Security_Overview/SeeAlso/SeeAlso.html#//apple_ref/doc/ui
- https://www.dfn.de/fileadmin/3Beratung/Betriebstagungen/bt54/forum-mobileit-heider.pdf
- http://www.cycript.org/
- http://iphonedevwiki.net/index.php/Cycript_Tricks