SecurusGlobal

Hacking Mobile Applications

Industry Case Studies

SecurusGlobal

Introduction

- **About Me**
  - Michael Gianarakis
  - Senior Security Consultant at Securus Global
  - Working in application security for six years
  - Focus on mobile application security

About Securus Global

Provide information security assessment, advisory and assurance services including security strategy, security products, penetration testing, PCI DSS assessments and compliance audits.

SecurusGlobal

Overview

Mobile platforms have presented many **opportunities** for businesses….

SecurusGlobal

Overview

Mobile platforms have presented many **opportunities** for businesses….

BUT

SecurusGlobal

Overview

Mobile platforms have presented many **opportunities** for businesses….

BUT

It's also created a lucrative **target** for hackers.

SecurusGlobal

Key Takeaways

Understand

Identify

Defend

SecurusGlobal

Key Takeaways

Understand

Identify

Defend

SecurusGlobal

Key Takeaways

Understand

Identify

Defend

SecurusGlobal

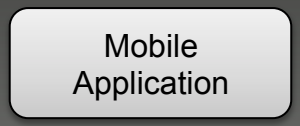
Before we get started….

SecurusGlobal

Understanding the Mobile Threat Model

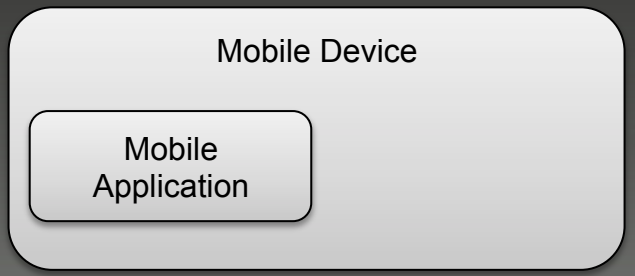
SecurusGlobal

Mobile Threat Model



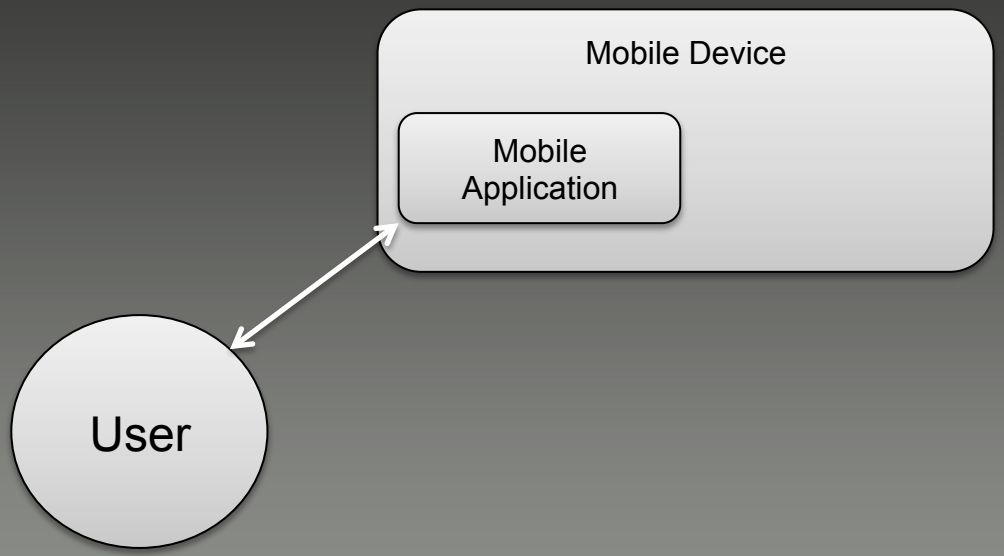
SecurusGlobal

Mobile Threat Model



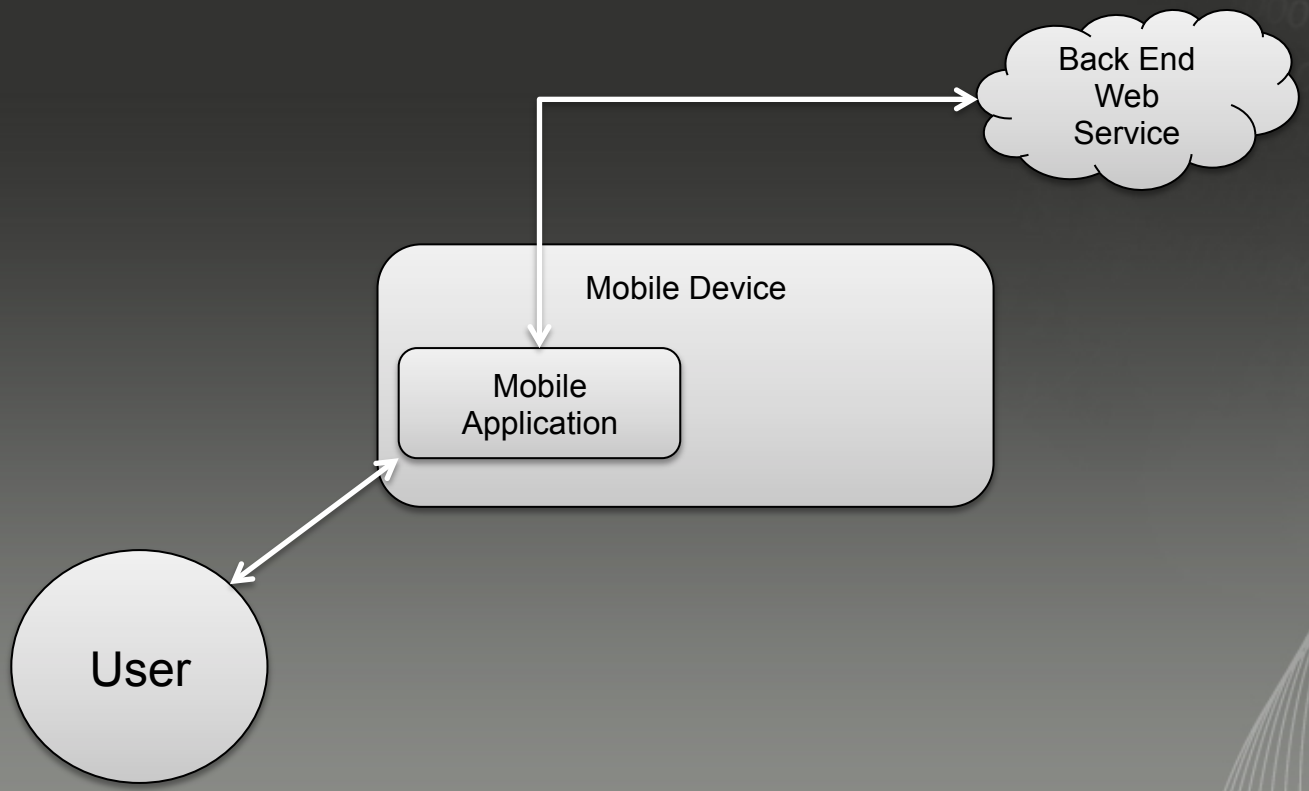
SecurusGlobal

Mobile Threat Model



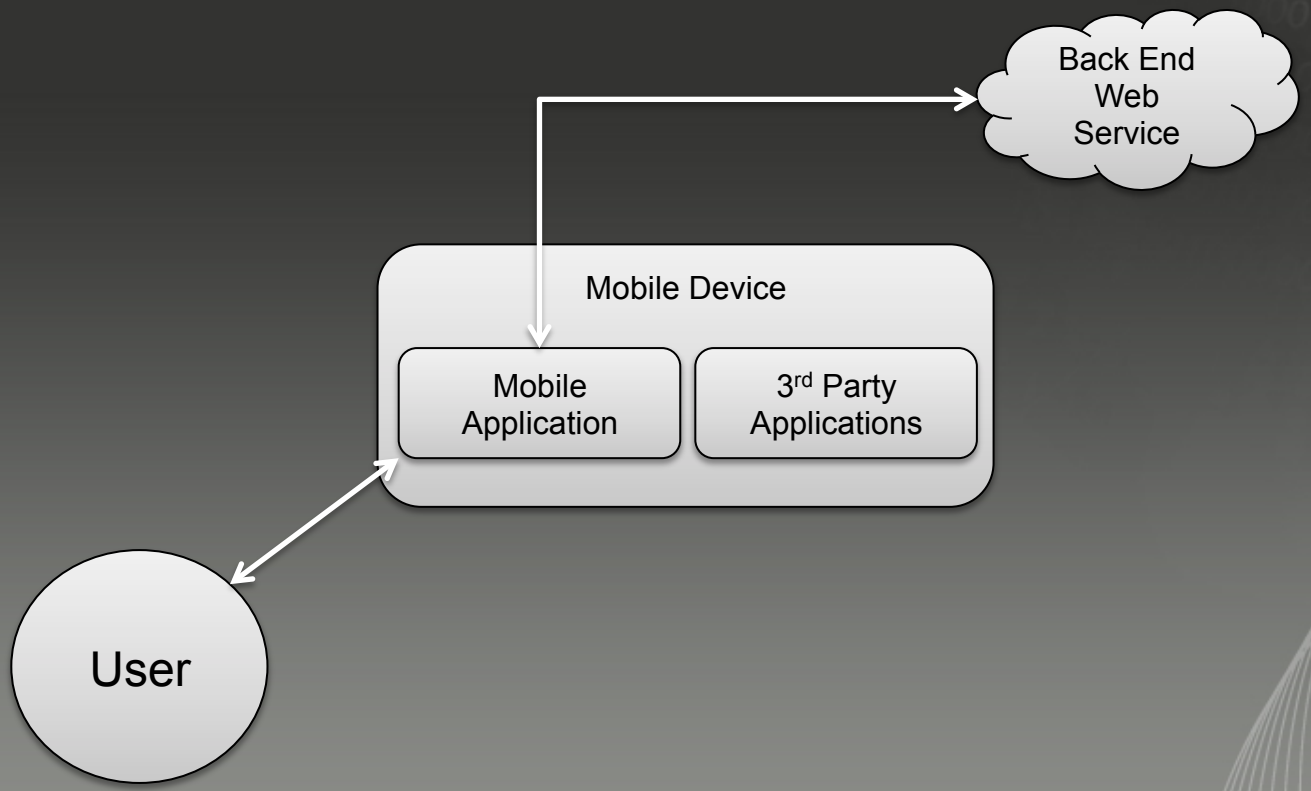
SecurusGlobal

Mobile Threat Model



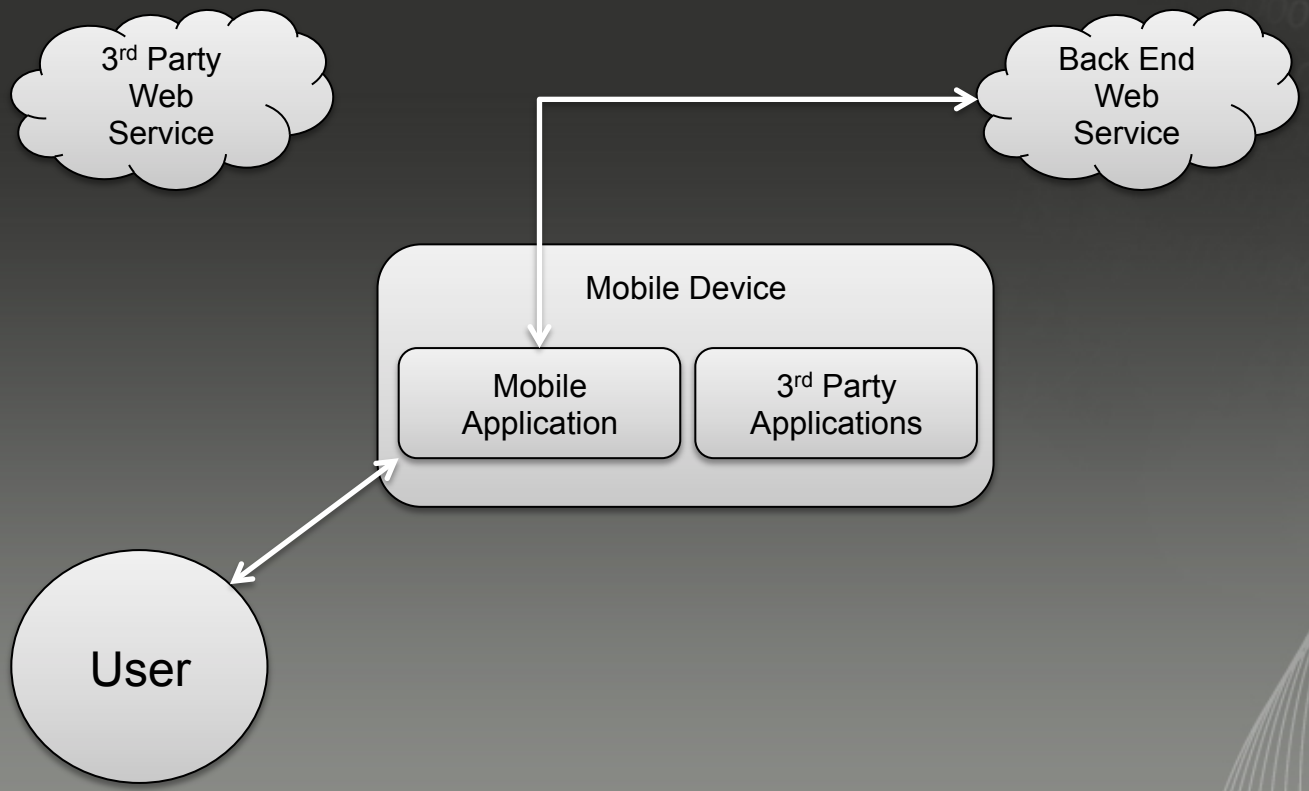
SecurusGlobal

Mobile Threat Model



SecurusGlobal

Mobile Threat Model

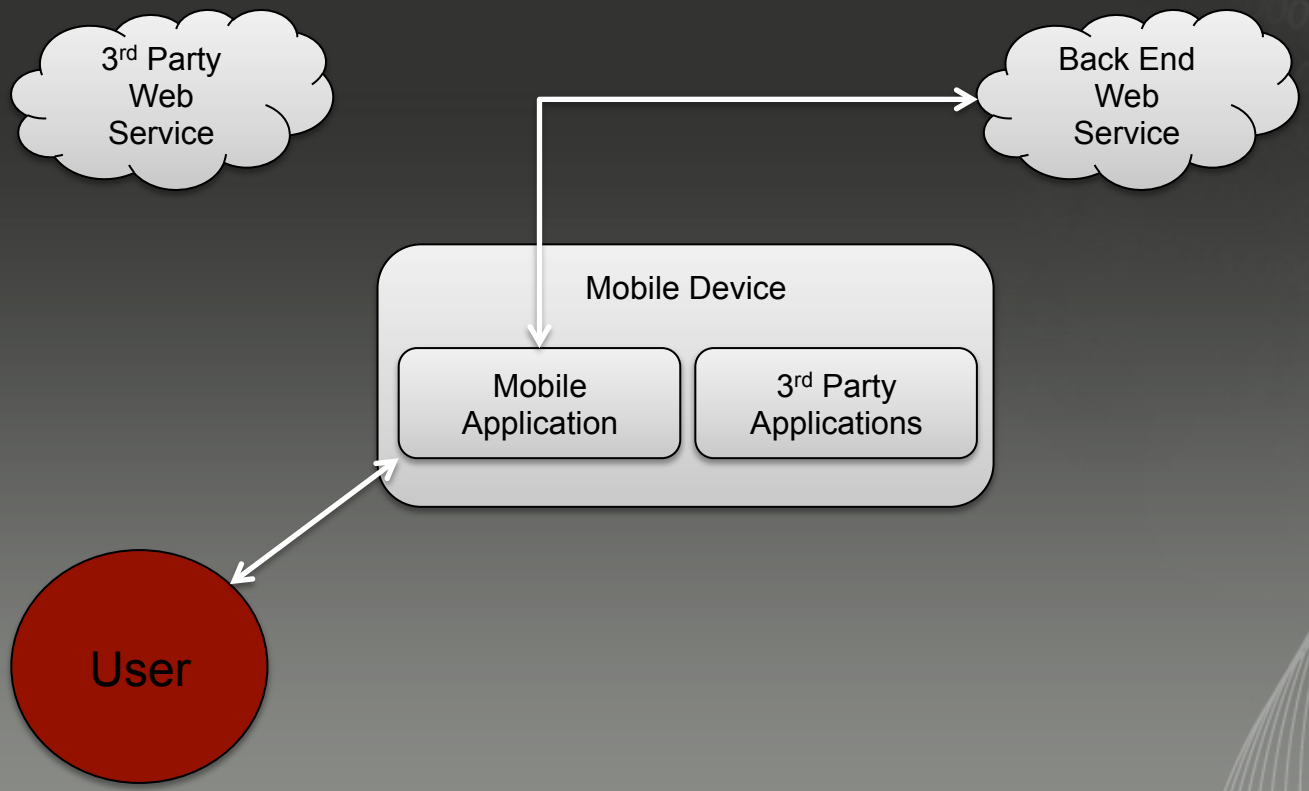


SecurusGlobal

So what are the threats….

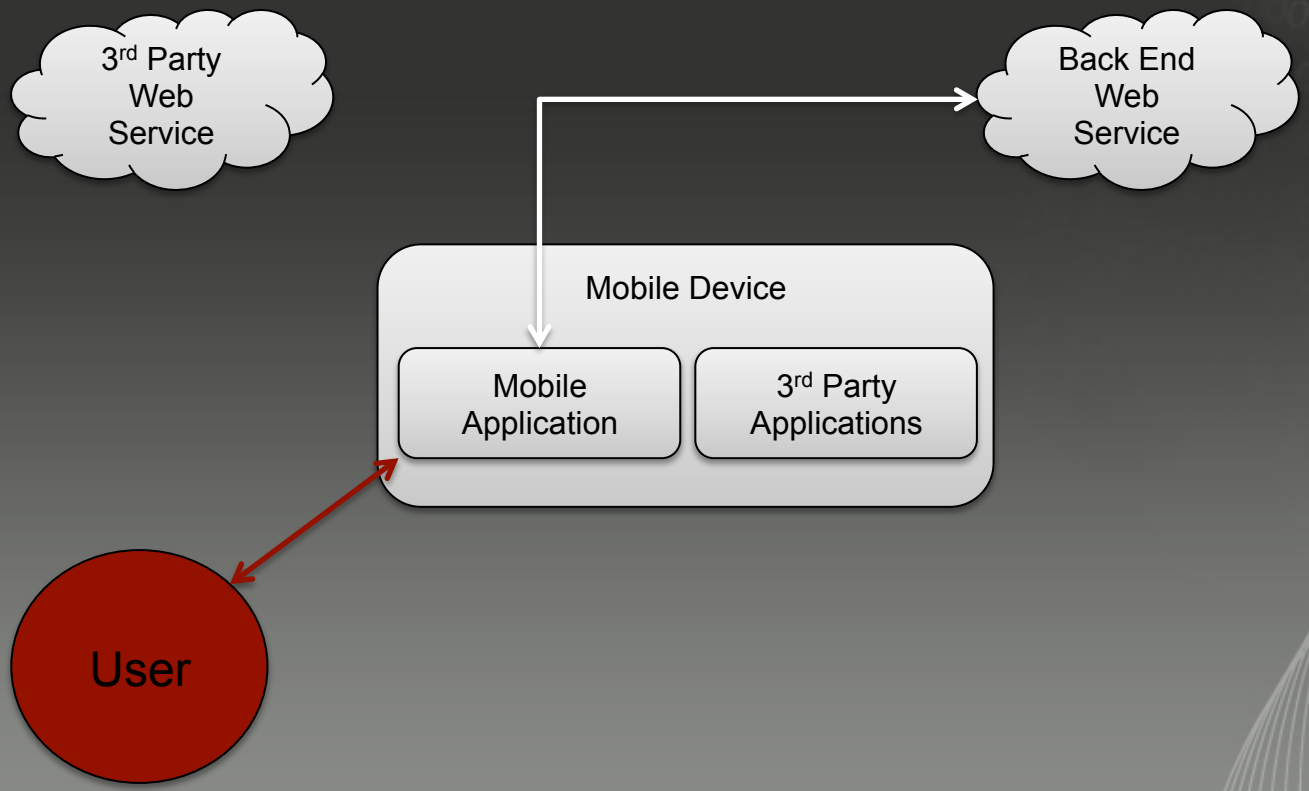
SecurusGlobal

Mobile Threat Model



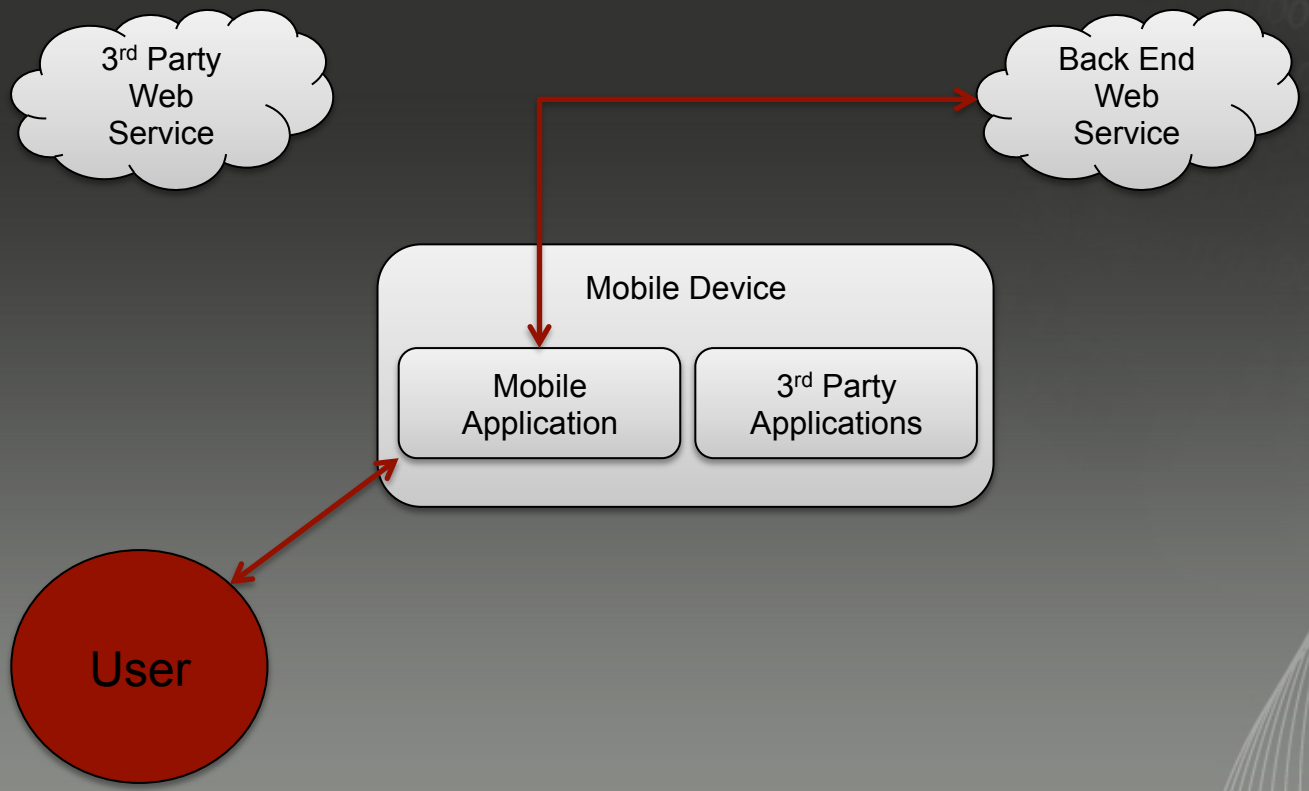
SecurusGlobal

Mobile Threat Model



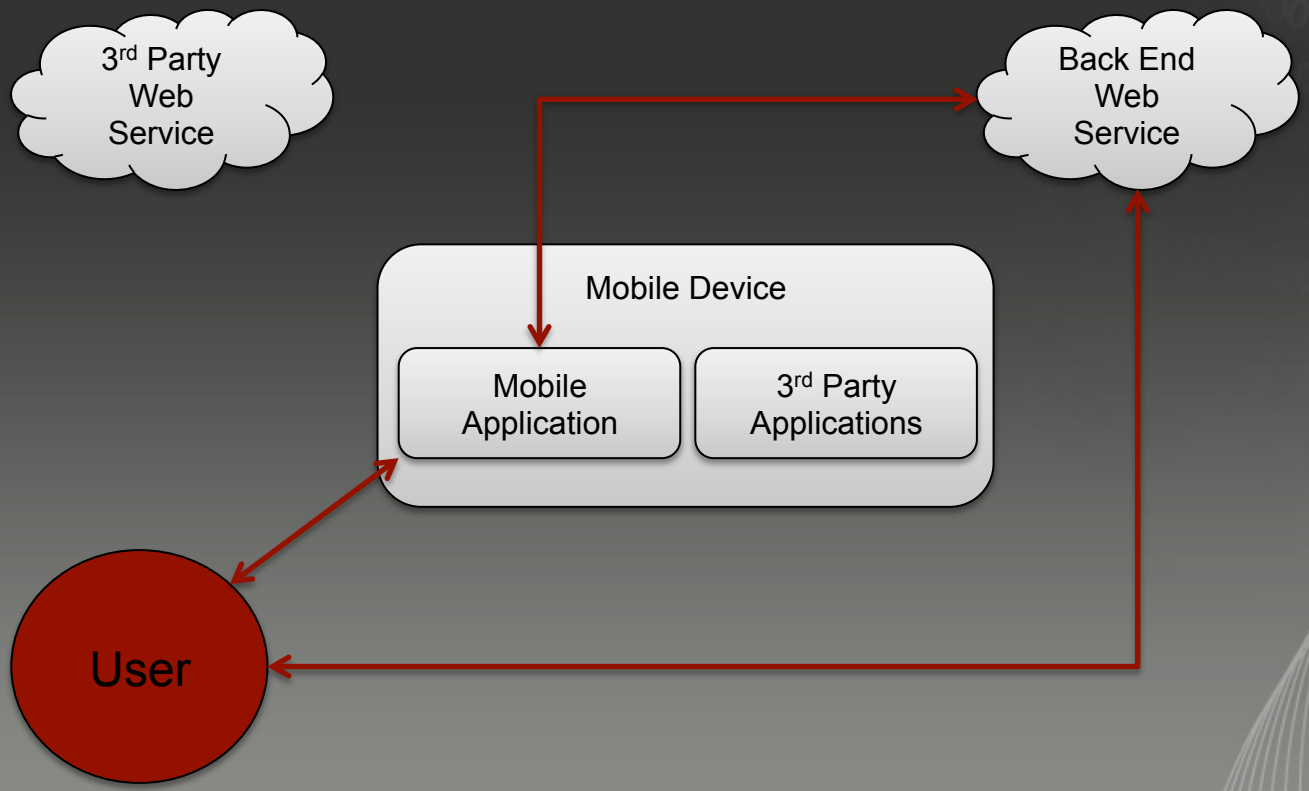
SecurusGlobal

Mobile Threat Model



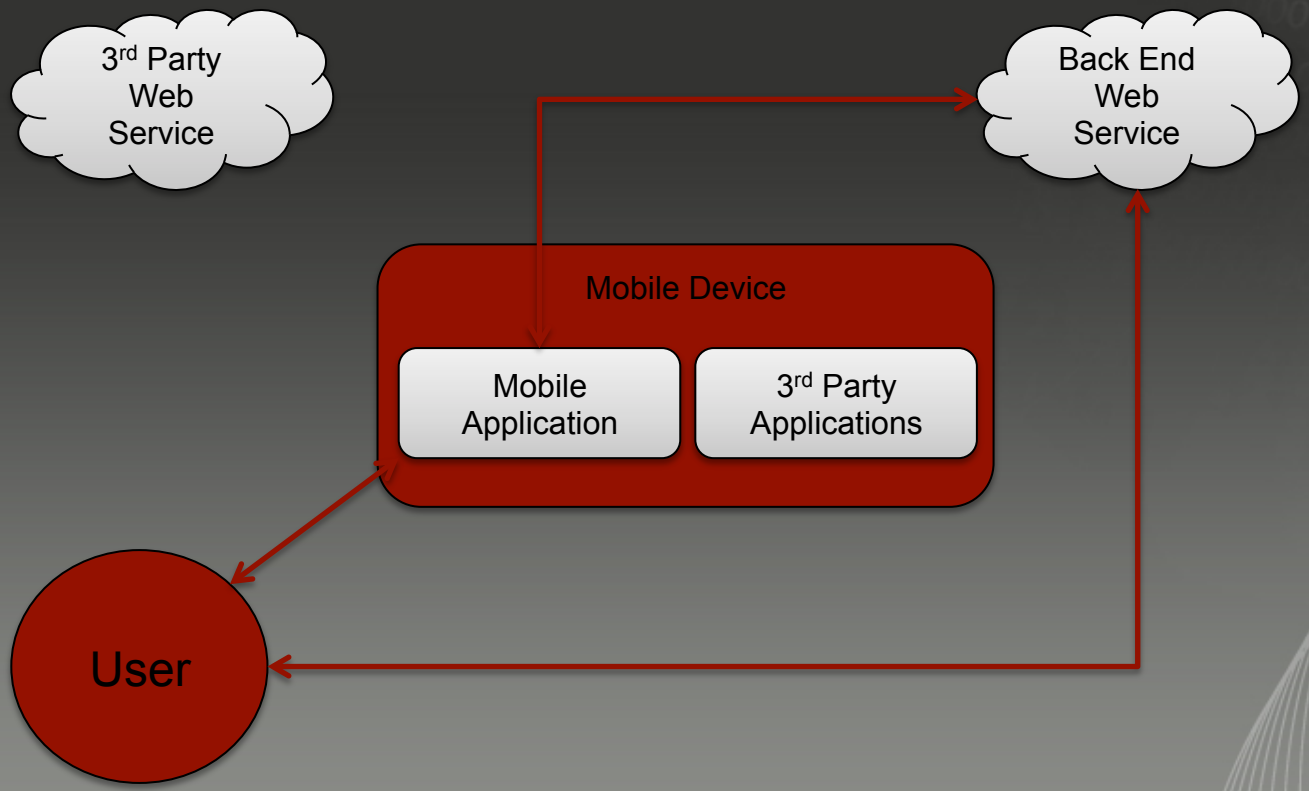
SecurusGlobal

Mobile Threat Model



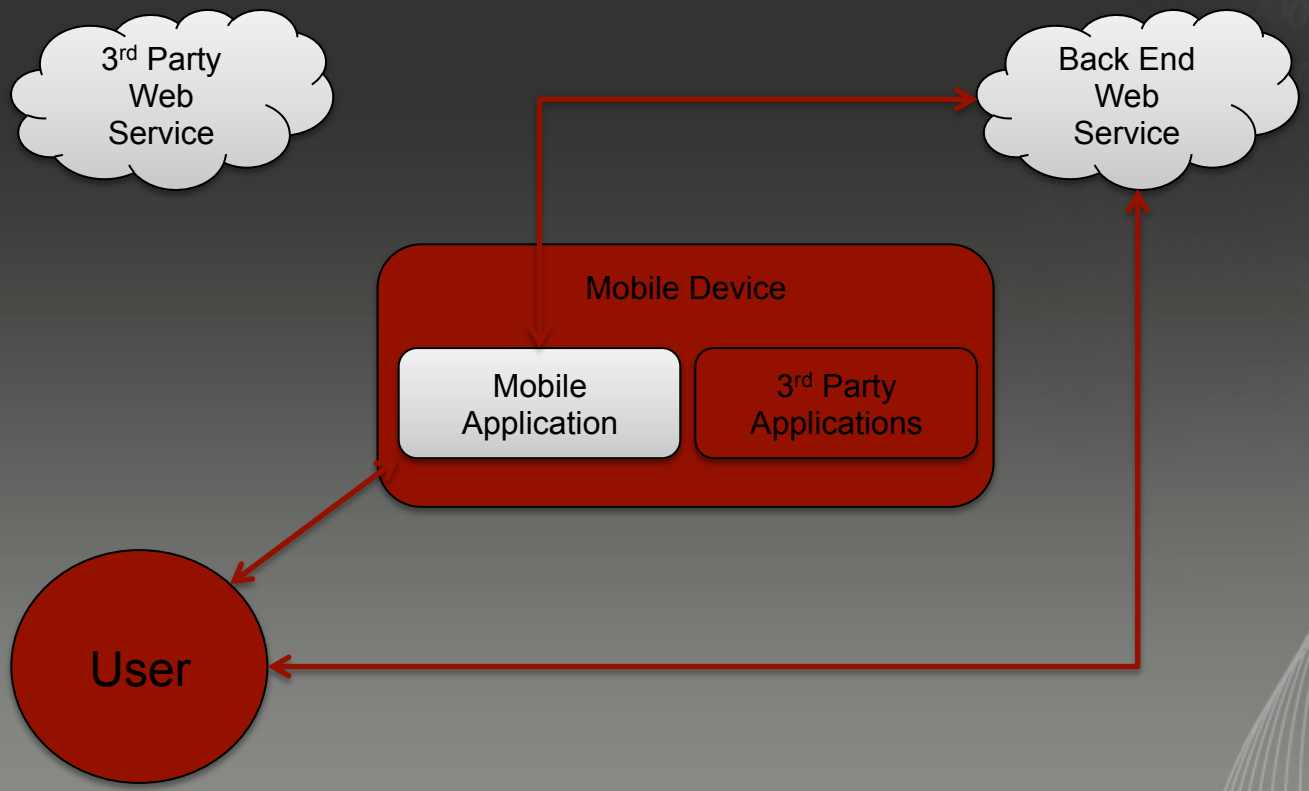
SecurusGlobal

Mobile Threat Model



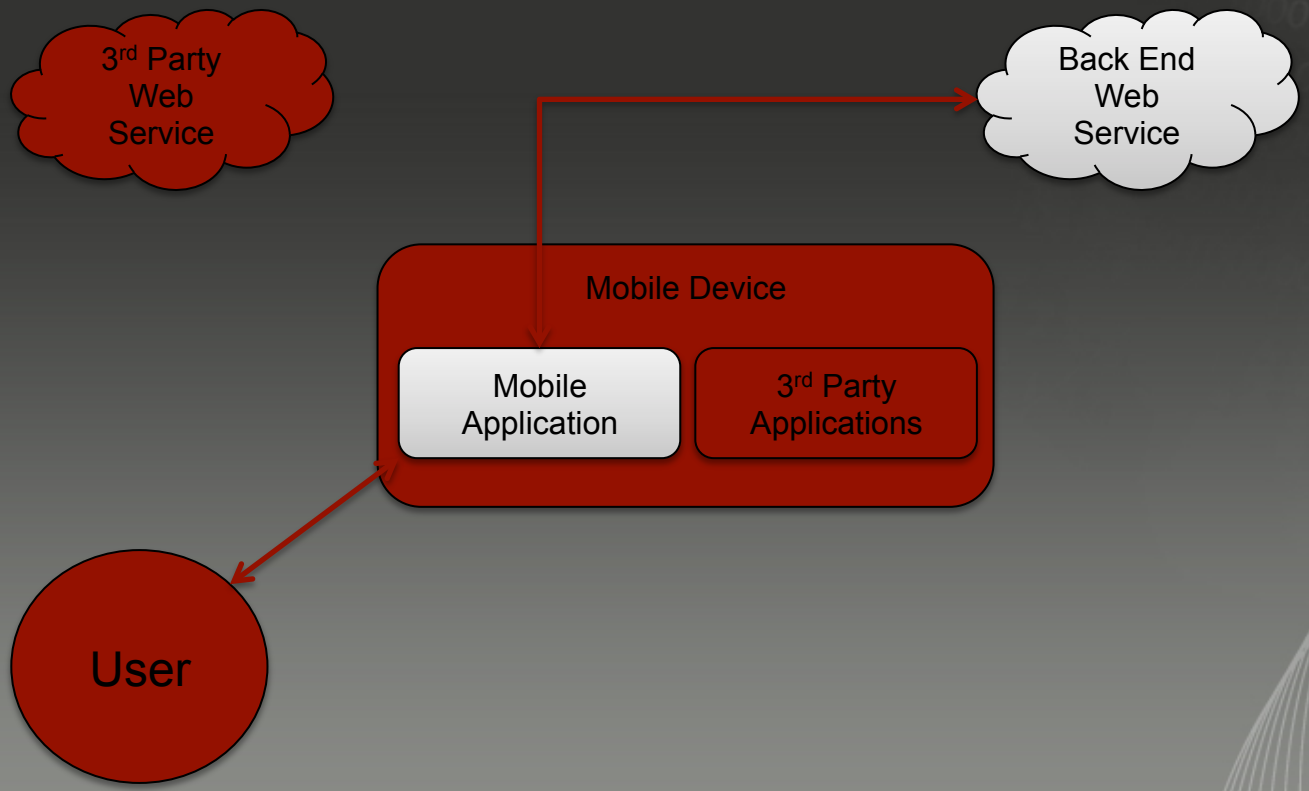
SecurusGlobal

Mobile Threat Model



SecurusGlobal

Mobile Threat Model

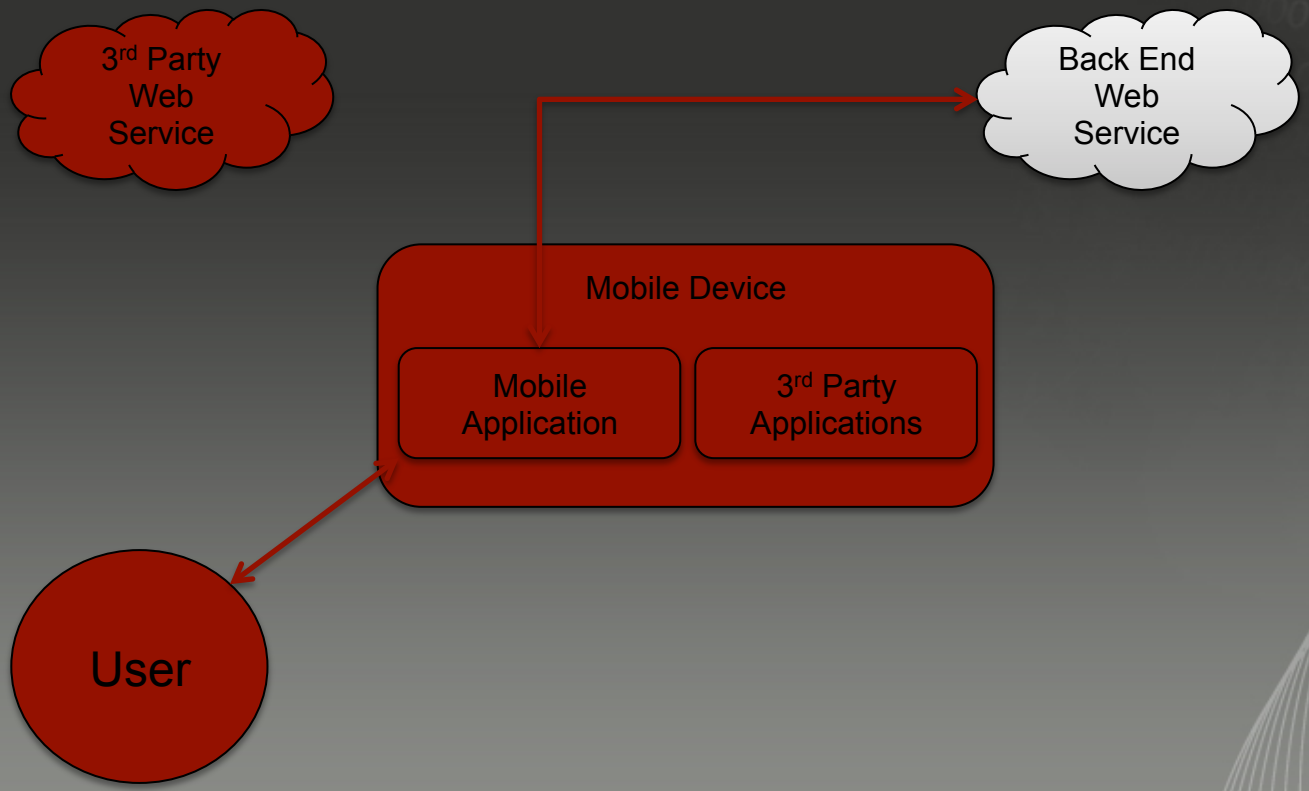


SecurusGlobal

That’s a lot of red…

SecurusGlobal

Mobile Threat Model



SecurusGlobal

Understanding the Risks

SecurusGlobal

OWASP Mobile Top 10 Risks

M1 – Weak Server Side Controls

M2 – Insecure Data Storage

M3 – Insufficient Transport Layer Protection

M4 – Unintended Data Leakage

M5 – Poor Authorisation and Authentication

M6 – Broken Cryptography

M7 – Client Side Injection

M8 – Security Decisions via Untrusted Inputs

M9 – Improper Session Handling

M10 – Lack of Binary Protections

SecurusGlobal

OWASP Mobile Top 10 Risks

M1 – Weak Server Side Controls	M2 – Insecure Data Storage	M3 – Insufficient Transport Layer Protection	M4 – Unintended Data Leakage
M5 – Poor Authorisation and Authentication	M6 – Broken Cryptography	M7 – Client Side Injection	M8 – Security Decisions via Untrusted Inputs
	M9 – Improper Session Handling	M10 – Lack of Binary Protections	

SecurusGlobal

OWASP Mobile Top 10 Risks

M1 – Weak Server Side Controls

M2 – Insecure Data Storage

M3 – Insufficient Transport Layer Protection

M4 – Unintended Data Leakage

M5 – Poor Authorisation and Authentication

M6 – Broken Cryptography

M7 – Client Side Injection

M8 – Security Decisions via Untrusted Inputs

M9 – Improper Session Handling

M10 – Lack of Binary Protections

SecurusGlobal

OWASP Mobile Top 10 Risks

M1 – Weak Server Side Controls

M2 – Insecure Data Storage

M3 – Insufficient Transport Layer Protection

M4 – Unintended Data Leakage

M5 – Poor Authorisation and Authentication

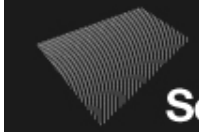
M6 – Broken Cryptography

M7 – Client Side Injection

M8 – Security Decisions via Untrusted Inputs

M9 – Improper Session Handling

M10 – Lack of Binary Protections



SecurusGlobal

OWASP Mobile Top 10 Risks

M1 – Weak Server Side Controls

M2 – Insecure Data Storage

M3 – Insufficient Transport Layer Protection

M4 – Unintended Data Leakage

M5 – Poor Authorisation and Authentication

M6 – Broken Cryptography

M7 – Client Side Injection

M8 – Security Decisions via Untrusted Inputs

M9 – Improper Session Handling

M10 – Lack of Binary Protections

SecurusGlobal

OWASP Mobile Top 10 Risks

M1 – Weak Server Side Controls

M2 – Insecure Data Storage

M3 – Insufficient Transport Layer Protection

M4 – Unintended Data Leakage

M5 – Poor Authorisation and Authentication

M6 – Broken Cryptography

M7 – Client Side Injection

M8 – Security Decisions via Untrusted Inputs

M9 – Improper Session Handling

M10 – Lack of Binary Protections

SecurusGlobal

In all of the mobile apps that we have tested for clients, these five risk are the most prevalent

This is surprisingly consistent across all of the apps that we have tested:

- Banking and trading applications
- Secure document reader applications
- Internal applications
- Commercial applications
- Brochureware

SecurusGlobal

We can simplify these risks by grouping them into three core areas:

1. Data Security
2. Runtime Security
3. Transport Security

SecurusGlobal

Identifying the Vulnerabilities

SecurusGlobal

Data Security

Improperly secured data stored on the mobile device is unfortunately very common.

SecurusGlobal

Data Security

I have come across all kinds of sensitive information stored in clear text including:

- Usernames and passwords
- Encryption keys
- Personal information
- Location data

Data Security

There are two main types of data security issues:

1. Insecure data storage (M2)
2. Unintended data leakage (M4)

SecurusGlobal

Data Security

Insecure data storage is when sensitive data is stored on the device that was not appropriately secured by the developer

Unintended data leakage typically occurs when sensitive information is stored automatically by the mobile operating system

Data Security

Sensitive data not appropriate secured by the developer includes:

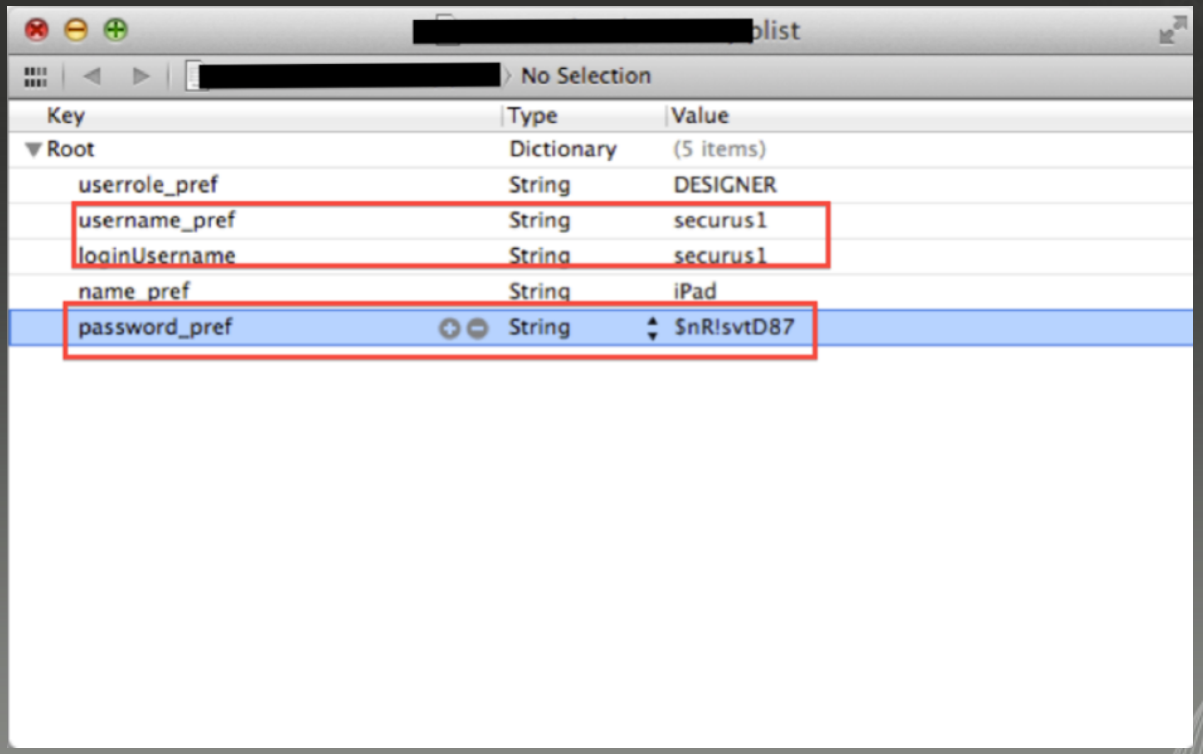
- Unencrypted databases
- Storing sensitive information in preference files
- Encrypting the data but storing the encryption key in a clear text file
- Improper use of system stores such as the Keychain
- Logging sensitive information to the device logs

SecurusGlobal

Examples

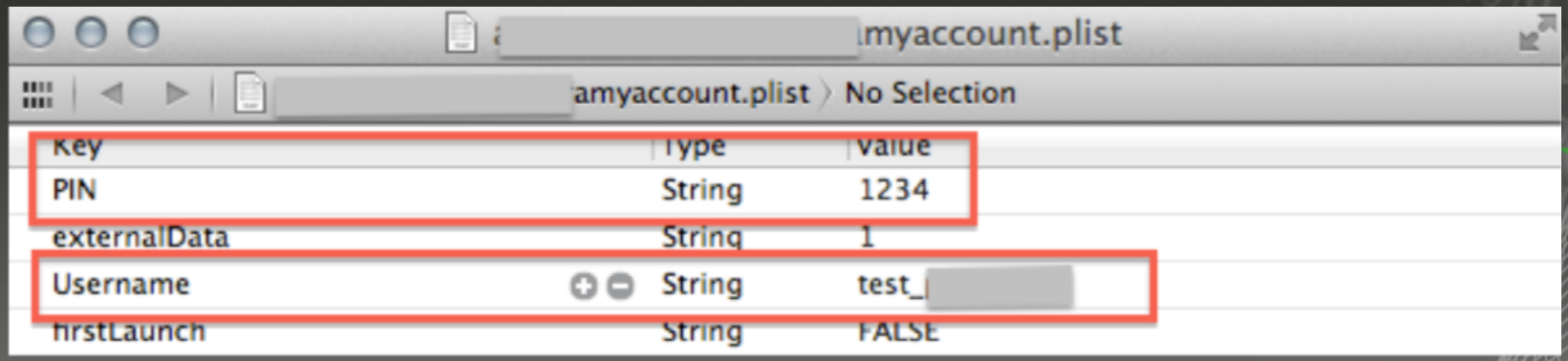
SecurusGlobal

Data Security



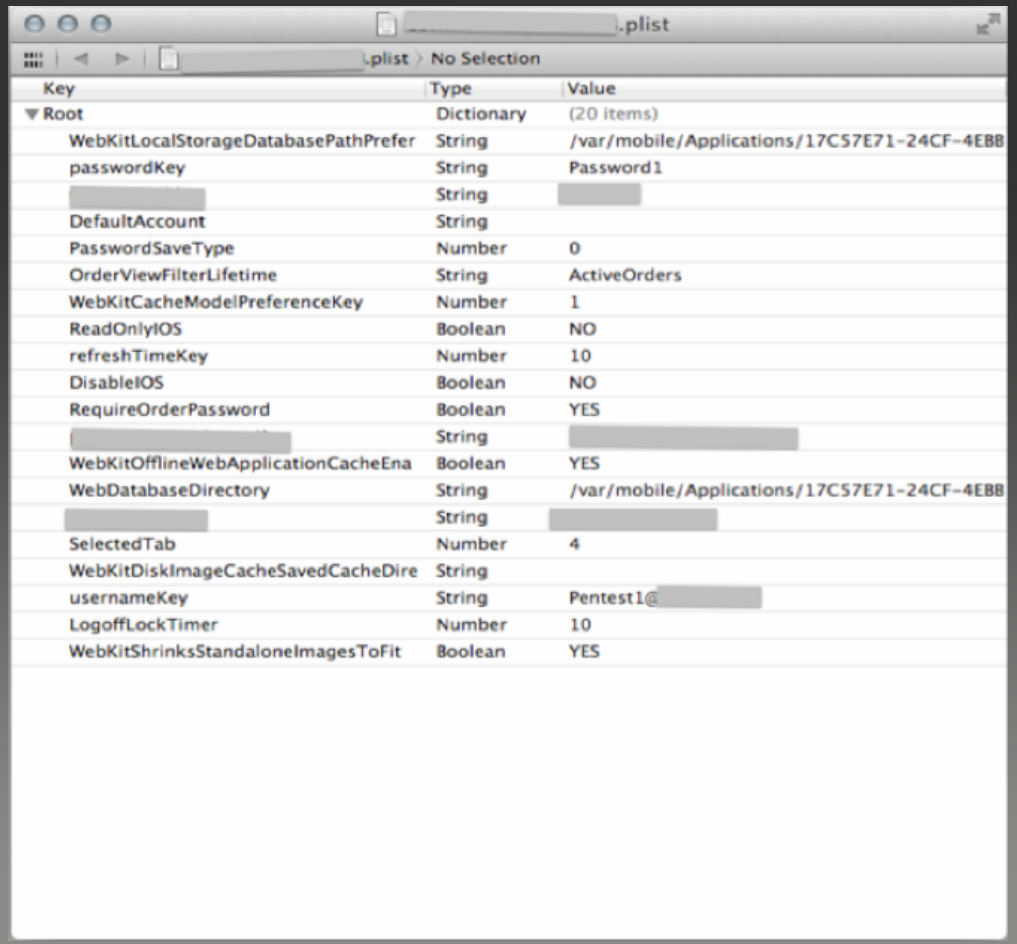
SecurusGlobal

Data Security



SecurusGlobal

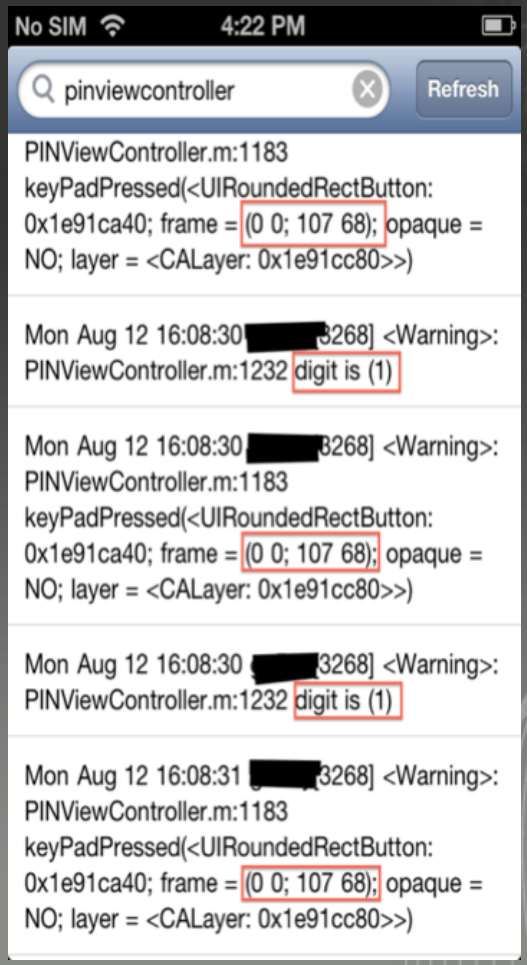
Data Security



SecurusGlobal

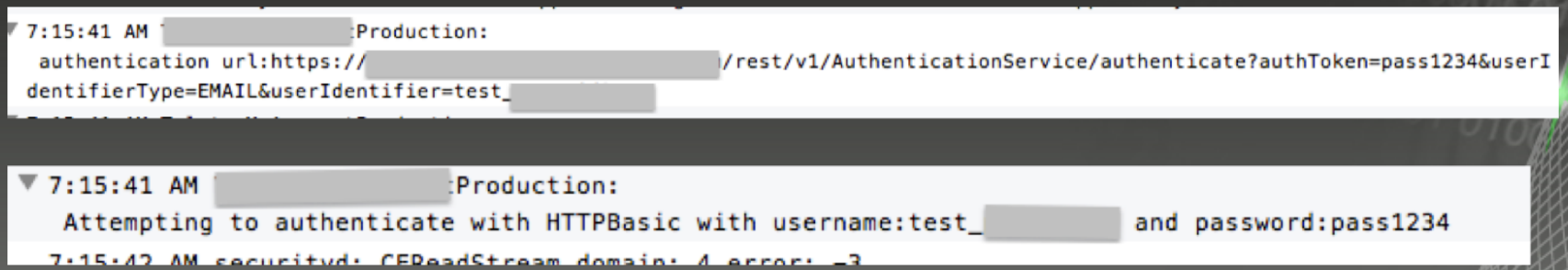
Data Security

```
Service/ServiceContracts/2009/08"><SOAP-
ENV:Body><ns1:Process><ns1:request
xsi:type="ns11:LoginRequestV4">
<ns2:SessionID
xsi:type="xsd:string">25451d7140894f4fa72
ee29ece0cf30f</ns2:SessionID>
<ns2:RequestID
xsi:type="xsd:string">1376286253</ns2:Req
uestID><ns2:AdditionalData
xsi:type="ns2:ArrayOfNameValuePair">
<ns2:NameValuePair
xsi:type="ns2:NameValuePair"><ns2:Name
xsi:type="xsd:string">DeviceUDID</ns2:Name>
<ns2:Value xsi:type="xsd:string">69D29AA2-
F3B0-4D43-83B4-04EF72B0D593</ns2:Value>
</ns2:NameValuePair></ns2:AdditionalData>
<ns2:Password
xsi:type="xsd:string">1185</ns2:Password>
<ns2:TimeZone
```



SecurusGlobal

Data Security



Data Security

Sensitive data stored by the mobile operating system includes:

- Background screenshots (iOS)
- Caches including browser caches and autocorrect
- System pasteboard

SecurusGlobal

Data Security

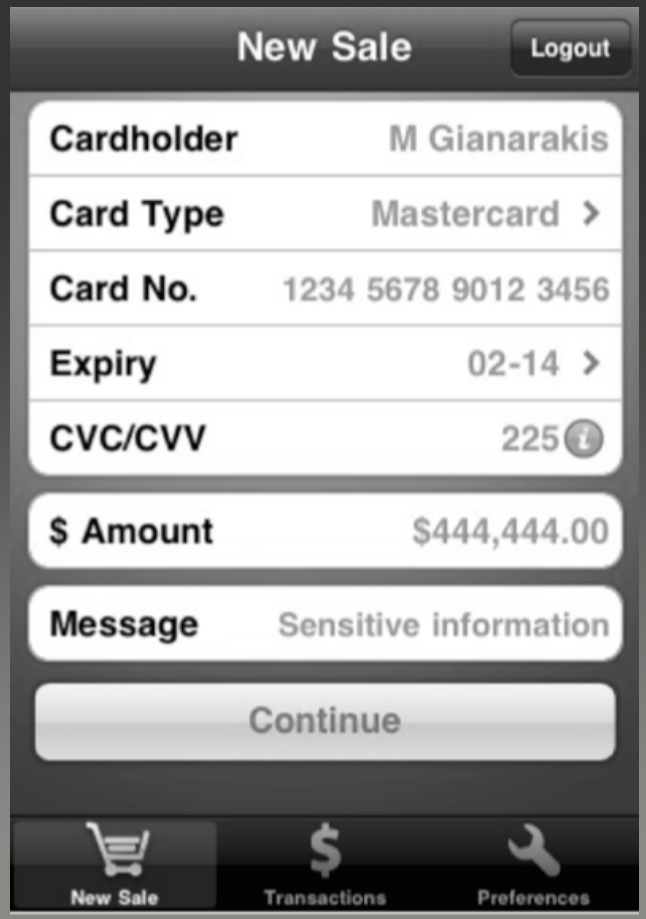
Unfortunately developers do not often realise that the mobile operating system is storing this information

SecurusGlobal

Examples

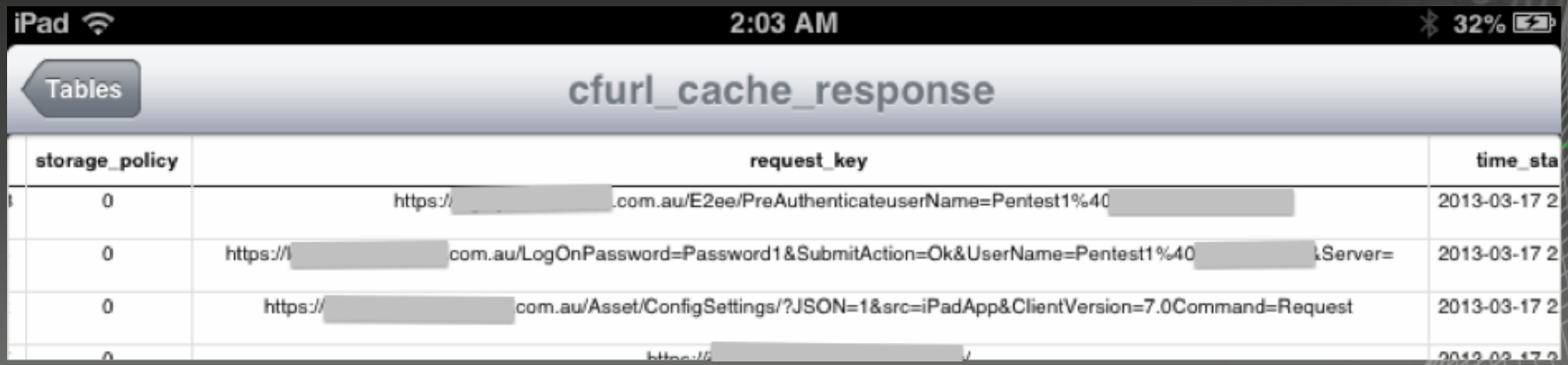
SecurusGlobal

Data Security



SecurusGlobal

Data Security



SecurusGlobal

Runtime Security

On mobile devices the execution environment is not controlled by us.

Once the security of this environment is compromised, either through malware and remote exploits or intentionally through jailbreaking or rooting, all bets are off.

SecurusGlobal

Runtime Security

Attackers can modify an application's behaviour at runtime allowing them to completely modify how the application is run.

It is possible create and call ad hoc methods as well as create ad hoc classes and methods on the fly.

SecurusGlobal

Runtime Security

These properties allows attackers to manipulate and abuse the runtime of the application.

Typically you can manipulate the runtime to bypass security locks, break logic checks, escalate privilege or steal information from memory.

SecurusGlobal

Examples

SecurusGlobal

Runtime Security

```
iPhone:~ root# ps aux | grep [redacted]
mobile     2722   2.2 29.6   555048 153188   ??  Ss    3:05PM   0:17.06 /var/mobile/Applications/188E2BF8-81B5-4781-98B9-AAA92932B0
6E/[redacted]
root       2732   0.0  0.0   273928      0 s003  R+    3:11PM   0:00.00 grep[redacted]
iPhone:~ root# cycript -p 2722
cy# UIApp.delegate.logonPIN
@"2580"
cy# UIApp.delegate.deviceToken
@"jdTbSh6l8Y"
cy# UIApp.delegate.AESKey
@"<58b3cffb 73e24bcf ae9310d8 8dc239a6 c11f460a ecae45ce b540a3cc 3e647b5f>"
cy# UIApp.delegate.mobileNumber
@"0432384204"
cy# UIApp.delegate.customerName
@"[redacted]"
cy# [AccountCache sharedInstance]
@"<AccountCache: 0x1e94cd20>"
cy# accountTest = new Instance(0x1e94cd20)
@"<AccountCache: 0x1e94cd20>"
cy# accountTest.accountList
cy# accountTest.primaryAccount
null
```

```
cy# *account_test
{isa:@"[redacted]ccount",alerts:null,usage:@"1111110",permissionBitMask:327743,accountName:@"[redacted]",displayAccountName:@"[redacted]
[redacted]",accountNumber:@"[redacted]-451732569",displayAccountNumber:@"[redacted]451732569",accountType:@"[redacted]",defaultBalance:@"<Money: 0x1d972d30>",otherBa
lance:@"<Money: 0x1d972f50>",everydayAccount:0,creditCardAccount:0,vali[redacted]nt:0,accountUserProperty:@"<AccountUserProperty: 0x1d972d00>",subA
ccountType:@"I3",originalDisplayAccountName:null,preferredAccount:1}
```

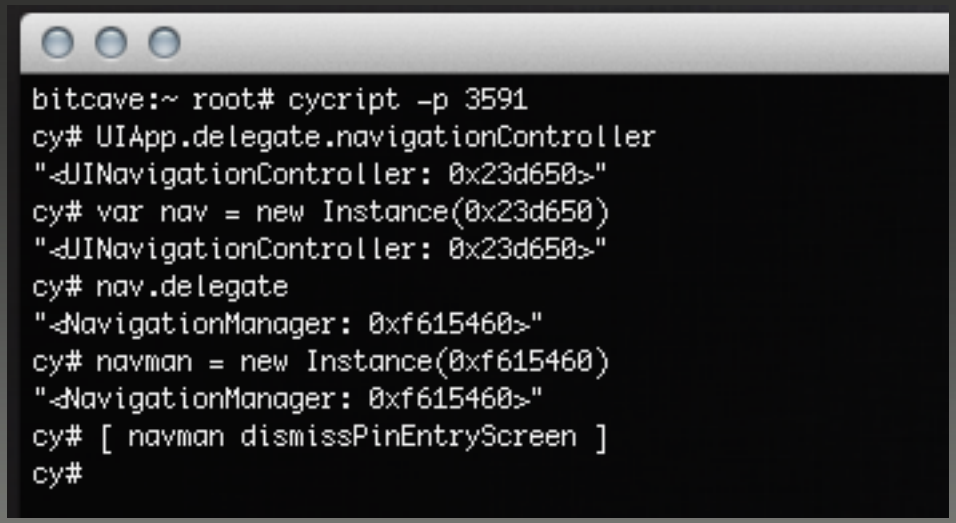
SecurusGlobal

Runtime Security

```
➜ _Securus  ssh -p 2222 root@127.0.0.1
root@127.0.0.1's password:
iPhonetest:~ root# ps aux | grep [redacted]
mobile    2736   0.0 15.1   448296  78336   ??  Ss    2:42PM   0:23.44 /var/mobile/Applications/9383A69D
-8D94-4BCC-9C7C-BF655A5EB020/[redacted]
root      2875   0.0  0.0   273928      0 s000  R+    3:23PM   0:00.00 grep [redacted]
iPhonetest:~ root# cycript -p 2736
cy# UIApp.keyWindow.rootViewController.childModalViewController.pinInputView
@"<[redacted]PinInputView: 0x202bf4e0; frame = (0 88; 320 372); autoresize = LM+RM; layer = <CALayer: 0x202f9
420>>"
cy# var pin_input_view = new Instance(0x202bf4e0)
@"<[redacted]PinInputView: 0x202bf4e0; frame = (0 88; 320 372); autoresize = LM+RM; layer = <CALayer: 0x202f9
420>>"
cy# pin_input_view.currentPin
@"2580"
cy# [[redacted]ryptoHelper sharedSecretKey]
@"MIIBPAIBAAJBAL+/xqlTLwY8WYa08AJ8bqgzfz/tlw4nsnJ+hyfnxu+onvmv7/JRsxZZOt7cTlRY4q06DsABznPZDNPdLbDcZwECAw
EAAQJBAKY37VOc61eai7ztAqAVE6u1MGrPX7GCkfsUWb8q+VWHu1Y/E/TnQntRqH0PwP9qxA8MOCFngSYohlbd5VIPjh0CIQDgyKZzqn
/gI8lNTAKvh8NQET1tImoGQ7mDQyMfPnC4qwIhANpgsSHmtxeK1ZgiJ3kfT0bdVd+zKQwkPbkazc0mLcDAiEAya/fgfrioAccEMnSwu
tqtvpLBkgRn6tWocs0kU0iMN8CIHVTAubeN3wUgRENOHvgkGtYOEDbxQJpfVjH+7pbLbkVAiEAmr//w2Yma1UggcRMulRRNRu2mo8zl0
p90gO56kBgnsk="
[redacted]SessionData: 0x201605d0>"
cy# function ivars(a){ var x=[]; for(i in *a){ try{ x[i] = (*a)[i]; } catch(e){} } return x; }
cy# session_data = new Instance(0x201605d0)
@"<[redacted]SessionData: 0x201605d0>"
cy# ivars(session_data)
{isa:@"[redacted]SessionData",_existingPin:null,_numberOfLoginAttemptsRemaining:0,_delegate:null,_reSendActiv
ationCodeSMSAfterFailedAttempts:0,_reSendActivationCodeWhenRequested:0,_isRegistered:0,_registeredDevice
s:null,_pin:null,_accountHolder:@"<[redacted]AccountHolder: 0x2013f5b0>"}
cy# ivars(session_data.accountHolder)
{isa:@"[redacted]AccountHolder",_mobilePhoneNumber:@"0432384204",_etradeMessageNumber:null,[redacted]:@"881870070"
_eTradeUserID:null}
cy#
```

SecurusGlobal

Runtime Security



```
bitcave:~ root# cycript -p 3591
cy# UIApp.delegate.navigationController
"<UINavigationController: 0x23d650>"
cy# var nav = new Instance(0x23d650)
"<UINavigationController: 0x23d650>"
cy# nav.delegate
"<NavigationManager: 0xf615460>"
cy# navman = new Instance(0xf615460)
"<NavigationManager: 0xf615460>"
cy# [ navman dismissPinEntryScreen ]
cy#
```

SecurusGlobal

Runtime Security

```
iPhonetest:~ root# ps aux | grep [redacted]
root      3227   0.0  0.0   273928      0 s000  R+   11:56AM   0:00.00 grep [redacted]
mobile    3221   0.0  7.2   392716  37044   ??  Ss   11:50AM   0:10.03 /var/mobile/Applications/9383A690
-8D94-4BCC-9C7C-BF655A5EB020/[redacted]
iPhonetest:~ root# cycript -p 3221
cy# // Jailbreak Detection Working as Normal
cy# [[redacted]JailbreakDetector sharedInstance]
@"<[redacted]ilbreakDetector: 0x1d503750>"
cy# var jb = new Instance(0x1d503750)
@"<[redacted]JailbreakDetector: 0x1d503750>"
cy# [jb isDeviceJailBroken]
1
cy# // Swizzling the isDeviceJailBroken Method]
cy# [redacted]JailbreakDetector.messages['isDeviceJailBroken'] = function() {return false;}
function () {return false;}
cy# [jb isDeviceJailBroken]
0
cy# [UIApp.delegate warnUserIfDeviceIsJailbroken] // No longer works
```

SecurusGlobal

Transport Security

Transport security is important for mobile applications as most users will connect their devices to untrusted networks

Transport Security

It is common to find insecurely implemented transport security:

- Lack of SSL certificate validation
- Unencrypted communications (although this is less common)

SecurusGlobal

Transport Security

Often times SSL validation will be a toggle in the application.

This is usually a holdover from development where the development environment does not have valid certs and is typically encapsulated in a variable somewhere.

Transport Security

Sometimes developers will forget to set SSL validation in the final build.

Also, as we saw before, variables are easy to manipulate.

SecurusGlobal

Examples

SecurusGlobal

Transport Security

```
@interface [redacted]AppConfiguration : XXUnknownSuperclass <[redacted]INCMSHTTPClientConfiguration> {
@private
    BOOL _allowInsecureCookieTransport;
    NSURL* _companyLogoEndpointURL;
    NSURL* _[redacted]UserServicesEndpointURL;
    NSURL* _contentServerEndpointURL;
    NSURL* _refreshScopeEndpointURL;
    NSString* _secureContentBasePath;
    NSURL* _loginEndpointURL;
    NSURL* _getFileEndpointURL;
    NSURL* _[redacted]EndpointURL;
    NSURL* _[redacted]EndpointURL;
    BOOL _allowUntrustedSSLCertificates;
}
@property(retain, nonatomic) NSString* secureContentBasePath; // G=0x1f753d; S=0x1f7571;
@property(retain, nonatomic) NSURL* contentServerEndpointURL; // G=0x1f7455; S=0x1f7489;
@property(retain, nonatomic) NSURL* [redacted]erServicesEndpointURL;  // G=0x1f73e1; S=0x1f7415;
@property(retain, nonatomic) NSURL* companyLogoEndpointURL; // G=0x1f736d; S=0x1f73a1;
@property(assign, nonatomic) BOOL allowInsecureCookieTransport; // G=0x1f7301; S=0x1f7335;
@property(assign, nonatomic) BOOL allowUntrustedSSLCertificates;  // G=0x1f7781; S=0x1f77b5;
@property(retain, nonatomic) NSURL* getFileEndpointURL; // G=0x1f7625; S=0x1f7659;
@property(retain, nonatomic) NSURL* refreshScopeEndpointURL;  // G=0x1f74c9; S=0x1f74fd;
@property(retain, nonatomic) NSURL* loginEndpointURL; // G=0x1f75b1; S=0x1f75e5;
@property(retain, nonatomic) NSURL* [redacted]EndpointURL; // G=0x1f7699; S=0x1f76cd;
@property(retain, nonatomic) NSURL* [redacted]EndpointURL; // G=0x1f770d; S=0x1f7741;
+(id)defaultConfiguration;  // 0x1f6949
-(void).cxx_destruct; // 0x1f77ed
-(id)debugDescription;  // 0x1f6d41
-(BOOL)allowsInsecureTransport; // 0x2da09
-(BOOL)allowsUntrustedCertificates; // 0x2d9cd
-(id)secureCookiePath;  // 0x2d995
-(id)baseURL; // 0x2d95d
@end
```

SecurusGlobal

Transport Security

```
iPhonetest:~ root# cycript -p 1053
cy# function ivars(a){ var x={}; for(i in *a){ try{ x[i] = (*a)[i]; } catch(e){} } return x; }
cy# [    AppConfiguration defaultConfiguration]
@"<    AppConfiguration: 0x1ed069b0>"
cy# config = new Instance(0x1ed069b0)
@"<    AppConfiguration: 0x1ed069b0>"
cy# ivars(config)
{isa:@"    AppConfiguration", allowInsecureCookieTransport:0,_    ompanyLogoEndpointURL:@"
                                                                                    /UserService
s/UserServices.svc",_contentServerEndpointURL:@                                    ",_refreshScopeEndpointURL:@
"                                                  ",_secureContentBasePath:@"/content/secure",_loginEn
dpointURL:@                                                .svc",_getFileEndpointURL:@"https
                  /Service.svc/GetFile/",_    rEndpointURL:@"ht                    ",_    EndpointU
RL:@                                    _allowUntrustedSSLCertificates:0}
cy# [config allowsInsecureTransport]
0
cy# [config allowsUntrustedCertificates]
0
cy# //Set the allowInsecureCookieTransport and allowUnstrustedSSLCertificates variables to 1
cy# config.allowInsecureCookieTransport = 1
1
cy# config.allowUntrustedSSLCertificates = 1
1
cy# // Transport security controls now disabled
cy# ivars(config)
{isa:@"    AppConfiguration",_allowInsecureCookieTransport:1,_companyLogoEndpointURL:@"
                                                                                    /UserService
s/UserServices.svc",_contentServerEndpointURL:@                                    ",_refreshScopeEndpointURL:@
"                                  ScopeService.svc",_secureContentBasePath:@"/content/secure",_loginEn
dpointURL:@                                    oginService.svc",_getFileEndpointURL:@
                                  ile/"    EndpointURL:@"                    ",    lEndpointU
RL:@"h                                    ,_allowUntrustedSSLCertificates:1}
cy# [config allowsInsecureTransport]
1
cy# [config allowsUntrustedCertificates]
1
cy#
```

SecurusGlobal

Defending Mobile Applications

SecurusGlobal

Defending Mobile Applications

Challenges:

- Devices are easily lost, stolen or compromised
- There are multiple attack vectors outside of your control
- Once the security of the device is compromised all best are off
- Platforms are constantly evolving
- Customers expect rapid iteration
- Developer inexperience with platforms (although this is improving)

SecurusGlobal

Defending Mobile Applications

First of all focus on the basics:

- Define the risk profile of the application
- Secure development practices
- Thorough security testing
- Monitoring and review

SecurusGlobal

Defending Mobile Applications

Effective information security is a process for managing business risk, not a product. Beware of “silver bullet” solutions.

SecurusGlobal

Defending Mobile Applications

The security of your application is **your** responsibility - not Apple, or Google or Microsoft

SecurusGlobal

Defending Mobile Applications

Mobile application security design principles:

- Assume the client is compromised
- Assume the application will connect to untrusted networks
- Assume the underlying operating system is compromised

SecurusGlobal

Defending Mobile Applications

Assume the client is compromised:

- Do not store sensitive information on the device
- Do not implement sensitive functionality in the client – always implement on the server
- Do not trust user input

Defending Mobile Applications

Assume the application will connect to untrusted networks:

- Do not transmit sensitive information unencrypted
- Do not use weak encryption
- Establish and validate certificate chain

SecurusGlobal

Defending Mobile Applications

Assume the underlying operating system is compromised:

- Genuine users will jailbreak their devices
- Attackers will jailbreak target devices
- Do not assume that physical access to the device is necessary for an attacker to compromise the device

SecurusGlobal

Defending Mobile Applications

Assume the underlying operating system is compromised

- Genuine users will jailbreak their devices
- Attackers will jailbreak target devices
- Do not assume that physical access to the device is necessary for an attacker to compromise the device

Defending Mobile Applications

Data Security

- Preference is to not store sensitive data
- Be realistic about requirements to actually store data (remember these devices are always connected)
- Be conscious of inadvertent data leakage by the operating system
- If storing sensitive data – encrypt but be aware of key management difficulties

SecurusGlobal

Defending Mobile Applications

Transport Security

- Encrypt all traffic
- ALWAYS validate certificates
- Certificate pinning
- Be aware of lax controls in development environments filtering through to production
- Do not use weak protocols (SSLv2, BEAST, CRIME etc)

SecurusGlobal

Defending Mobile Applications

Runtime Security

- Don’t trust user input
- Although hard to implement consider runtime security mechanisms
  - Anti-debugging
  - Jailbreak detection
  - Tamper response

SecurusGlobal

Defending Mobile Applications

Unfortunately it is impossible to completely secure mobile applications

- Anybody with a copy of the application and a debugger can compromise the security of the application

The aim is to make it significantly harder for the attacker such that the economic benefits of attacking the application are outweighed by the difficulty of the attack.

SecurusGlobal

Questions?