RSAConference2015

Singapore | 22-24 July | Marina Bay Sands

CHANGE

Challenge today's security thinking

SESSION ID: MBS-F01

Security Considerations for Mobile Payment Devices: Trends, Risks and Countermeasures

Wouter Veugelen

Director– Cyber Security
PwC Australia
@veugelenw

Suhas Desai

Associate Director – Cyber Security
PwC India
@desai_suhas

#RSAC

Agenda

- Trends in mobile payments
- Security risks in mobile payments applications and devices
- Mitigation strategy through secure SDLC
- Mobile security best practices

Disclaimer:
This session is intended to educate about security best practices for payment applications and devices.
Indicative security risks mitigation practices are given to address it during SDLC.

RSAConference2015

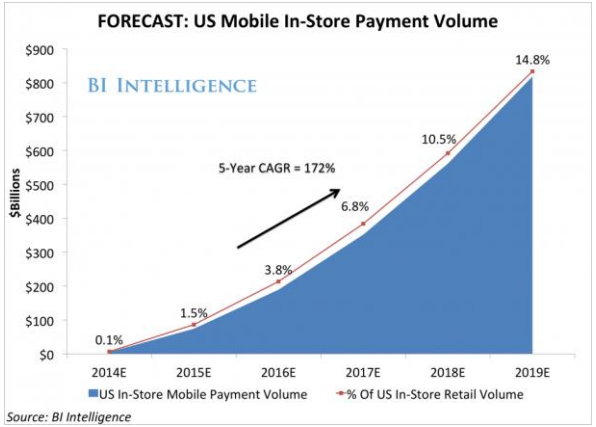
Singapore | 22-24 July | Marina Bay Sands

#RSAC

Trends in Mobile Payments

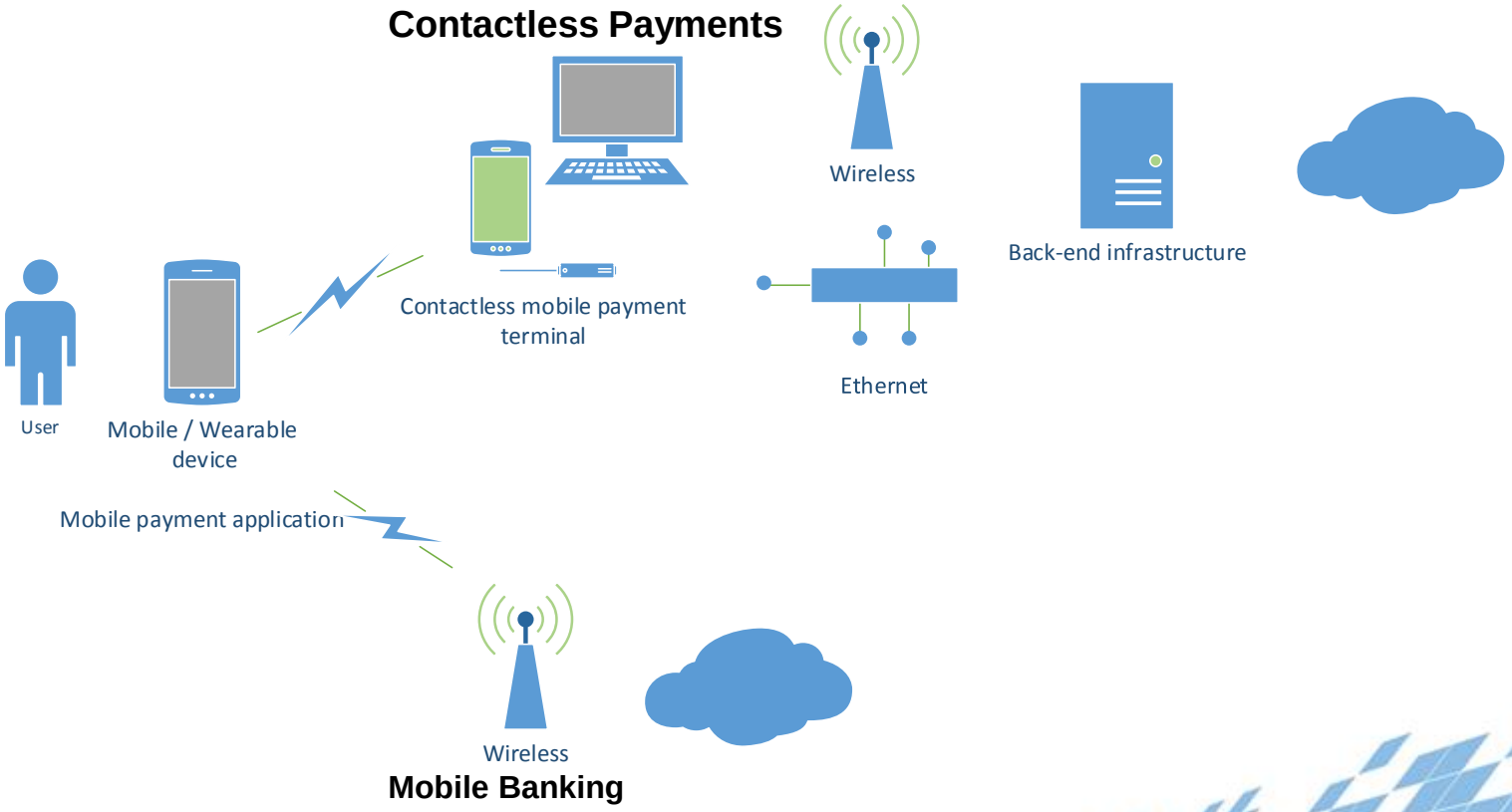
Trends in Mobile Payments

- Mobile payments in US alone predicted to reach $37 billion in 2015 and $808 billion by 2019[1]
- Technology companies currently dominate
- Payment methods:
  - Contactless payments
  - Cardless cash withdrawal
  - QR code payments
  - Wearable payments



1: Business Insider, BI Intelligence, US in-store mobile payments volume, June 2015

Mobile Payments Architecture

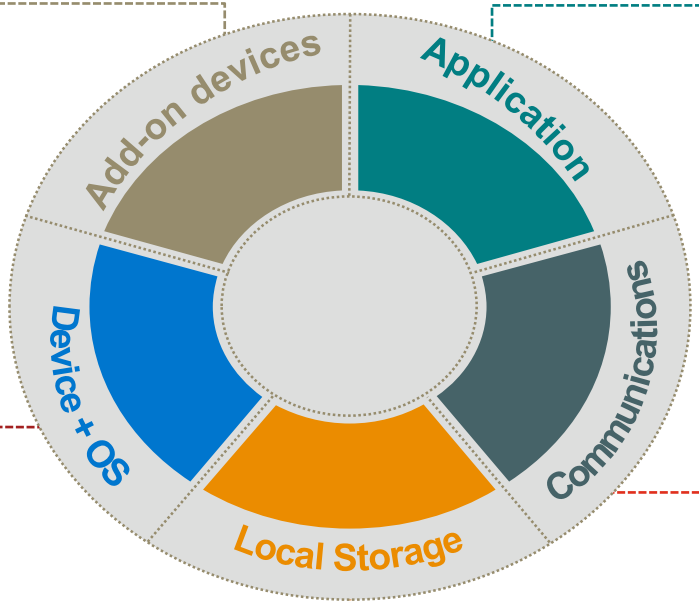


Attack Vectors against Mobile Payment Devices

- **Compromise of the mobile payment device**
  - Phishing
  - Attacks directed through physical interfaces (USB)
  - Attacks directed through wireless interfaces (e.g. Wifi, Bluetooth, NFC,…)
- **Compromise of the mobile payment application**
  - Mobile application security vulnerabilities
  - Local device storage vulnerabilities
  - Server side web application or web services security vulnerabilities
- **Compromise of the payment terminal infrastructure**
  - Direct attacks against wireless or physical interfaces
  - Infrastructure security vulnerabilities
  - Embedded device security vulnerabilities
  - Application security vulnerabilities
- **Compromise of the back-end infrastructure**

Common Security Vulnerabilities

- Insecure fingerprint/card/iris data in storage / transit - captured through add-on devices
- Insecure Connection with Parent Device



- Weak access control to protect application functionality
- Weak authentication or session management
- Weak control to prevent intercept ion and manipulation of message traffic
- Injection vulnerabilities due to improper coding practices

- Weak authentication and authorisation controls
- Missing OS patches
- Insufficient OS configuration hardening

- Weak encryption / clear text or encoded payloads
- Insufficient message integrity checks
- Lack of replay protection
- Missing transaction authentication checks

- Insecure storage of critical information
- UI impersonation though local data storage manipulation
- Insecure cryptography key storage and usage
- Data logging information disclosure

RSAConference2015

Singapore | 22-24 July | Marina Bay Sands

Security risks in Mobile Payments Apps, Devices , Communication Channels & Add-on Devices

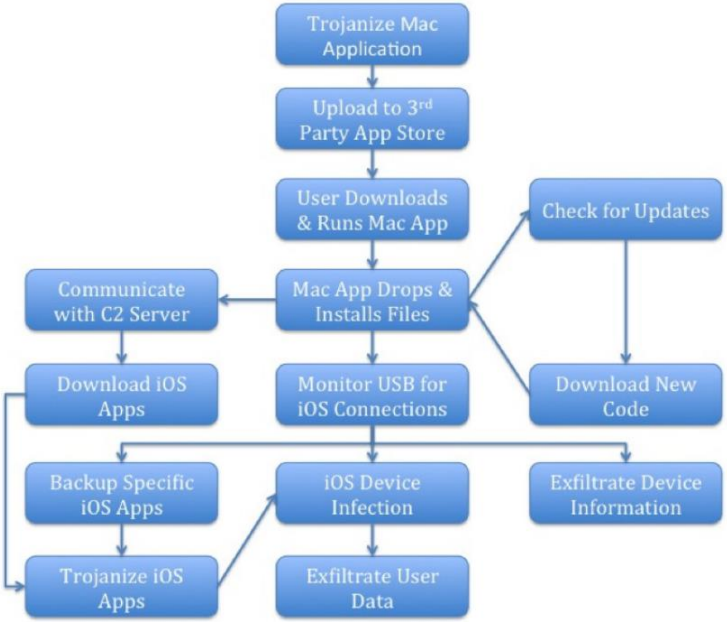
#RSAC

Mobile Device Risks

I. Direct attack via physical interfaces (USB)

Wirelurker

- Malware that monitors for, and attacks iOS devices via USB on OSX
- Installs third-party applications or automatically generated malicious applications onto the device
- On non-jail broken iOS devices through enterprise provisioning. creation of enterprise provisioning profiles on their non-jail broken iPhones and iPads. A user would then need to manually launch the installed app, then tap "Trust“



Mobile Device Risks

II. Direct attack against wireless interfaces

Mobile Pwn2own

- Every year at Mobile Pwn2own security researchers aim to obtain remote privileged access to the latest generation of smart phones, either through a browser based exploit, or through an attack against the wireless interfaces (e.g. Wifi, Bluetooth, NFC).
- Last event mobile Pwn2own event outcome:
  - Android: Privileged access from NFC and Wifi
  - iPhone: Privileged access from Wifi/Browser

Bluetooth LE

- Used by wearable devices
- Key exchange can be intercepted and traffic decrypted

Proxmark3

Ubertooth One

Major risks in Mobile Applications

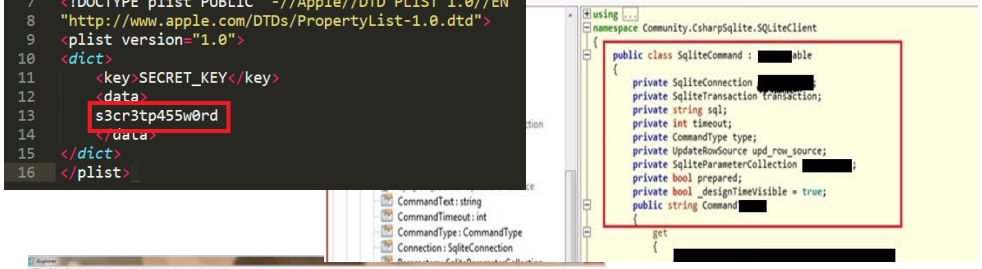
I. Insecure data storage

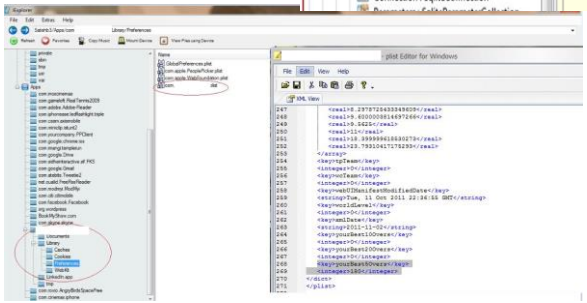
- Important data is not stored encrypted by the application (e.g. application configuration files, back-end database)
- Reliance on operating system security controls alone (e.g. keychain)
- Allowed privilege escalation within the application
- Allowed unauthorised access to PII

Authentication credentials

```
Service: MyCoolApp
Account: Encryption Root
Entitlement Group: apple
Label:
Generic Field:
Keychain Data: <?xml version"1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>SECRET_KEY</key>
    <data>
    s3cr3tp455w0rd
    </data>
</dict>
</plist>
```

SQL database credentials





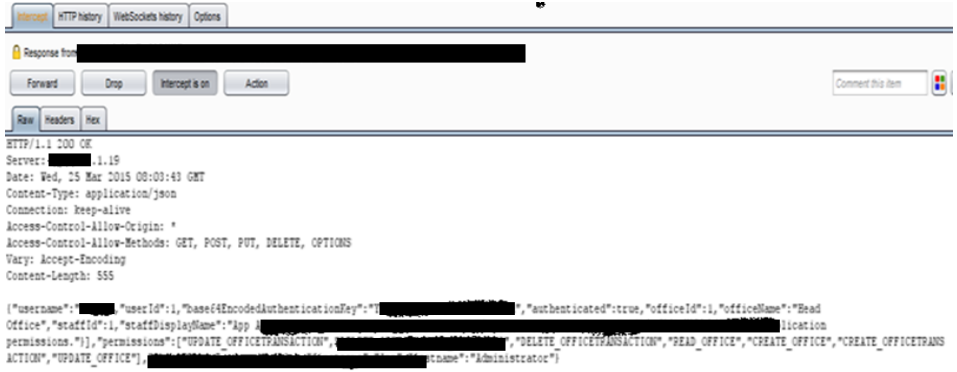
Plaintext application configuration settings

Major risks in Mobile Applications

II. Insecure payload

- Access to payment devices settings, transaction information that can be tampered with in transit

Use of basic encoding methods



Major risks in Mobile Applications

III. Reverse engineering

- Payment device root/jailbreak
- Extract mobile application from the payment device
- Use decompilers and disassemblers to application source code and understand business logic
- Sensitive data disclosure
  - e.g.: Extract certificates to bypass authentication
- Patch the application
  - e.g.: Always accept biometrical authentication (e.g. Touch ID)
  - e.g.: Disable jailbreak detection, application integrity checks, certificate pinning, debugger detection

Authentication Bypass by patching the application

```
; Attributes: bp-based frame

sub_2643EC
PUSH            {R4-R7,LR}
ADD             R7, SP, #0xC
SUB             SP, SP, #0x18
MOV             R5, R0
MOV             R0, R1
BLX             _objc_retain
MOV             R4, R0
CBZ             R4, loc_26442E
```

```
; Attributes: bp-based frame

sub_2643EC
PUSH            {R4-R7,LR}
ADD             R7, SP, #0xC
SUB             SP, SP, #0x18
MOV             R5, R0
MOVS            R0, #0
BLX             _objc_retain
MOV             R4, R0
CBZ             R4, loc_26442E
```

```
002643F4  08 46 C4 F2 42 ED 04 46  BC B1 41 F6 5A 30 C0 F2
00264404  51 00 78 44 06 68 20 46  31 46 C4 F2 0E ED 10 F1
```

```
002643F4  00 20 C4 F2 42 ED 04 46  BC B1 41 F6 5A 30 C0 F2
00264404  51 00 78 44 06 68 20 46  31 46 C4 F2 0E ED 10 F1
```

Embedded Cryptographic keys

```
data:100290C4 pCipher         dd offset Cipher        ; DATA XREF: SickEncryption+14↑r
data:100290C4                                         ; SickEncryption+3E↑r
data:100290C4                                         ; "%(@*^#(*@&&@()$@)[redacted]
```

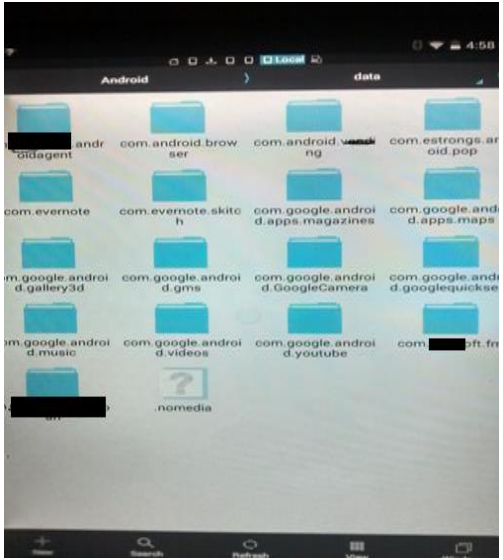

RSAConference2015

Major risks in Payment Devices

I. Unrestricted access to setting

- No kiosk mode
- Access to device settings
- Root this device
- View change file structure , settings, database
- Access mobile payment application binary
- Access logs

Payment application

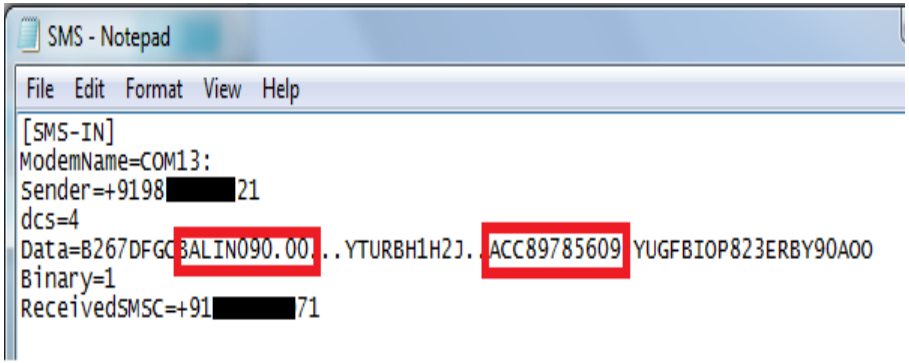


Major risks in Payment Devices

II. Communication Channels – Application SMS

- Application SMS
- Configure GSM Modem with Valid MSISDN
- Capture the SMS
- View /Modify SMS
- This is restricted scenario – Need SMSC, Application SMS generation and Older SMPP protocol Support

Clear text parameters



Major risks in Payment Devices

III. Communication Channels – USSD

- USSD Aggregators
- USSD is secure over GSM channel
- Review USSD data at aggregators
- Review data in transit from aggregators to payment gateways
- Attempt XML injections

Analyse XML payload and perform XML injection

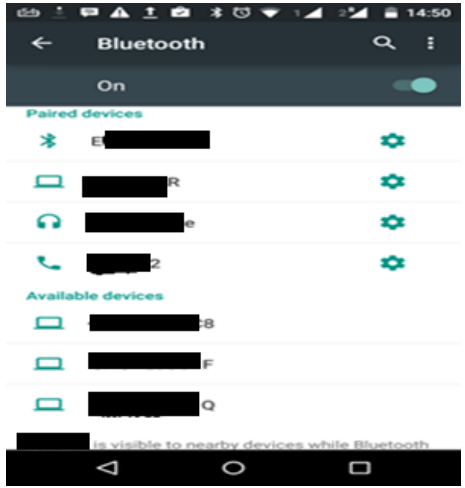
```
<?xml version= 1.0 ?>
<!DOCTYPE COMMAND PUBLIC "-//Ocam//DTD XML Command
<COMMAND>
<TYPE>[redacted]Q<TYPE>
<MSISDN>+9198[redacted]1</MSISDN>
<SRCACC>67[redacted]9</SRCACC>
<AMOUNT>150</AMOUNT>
<MPIN>1[redacted]</PIN>
<LANGUAGE>1</LANUAGE1>
```

Major risks in Payment Devices

IV. Communication Channels – Bluetooth

- Bluetooth device is integrated on the payment device
- Verify pairing mechanism
- Verify discovery
- Verify auto-connect

Pairing without PIN CODE



Major risks in Payment Devices

v. Add-on devices – Fingerprint Scanner & Printers

- Fingerprint Validations
- Printer discovery and connections
- Fingerprint data storage
- Fingerprint data in transit

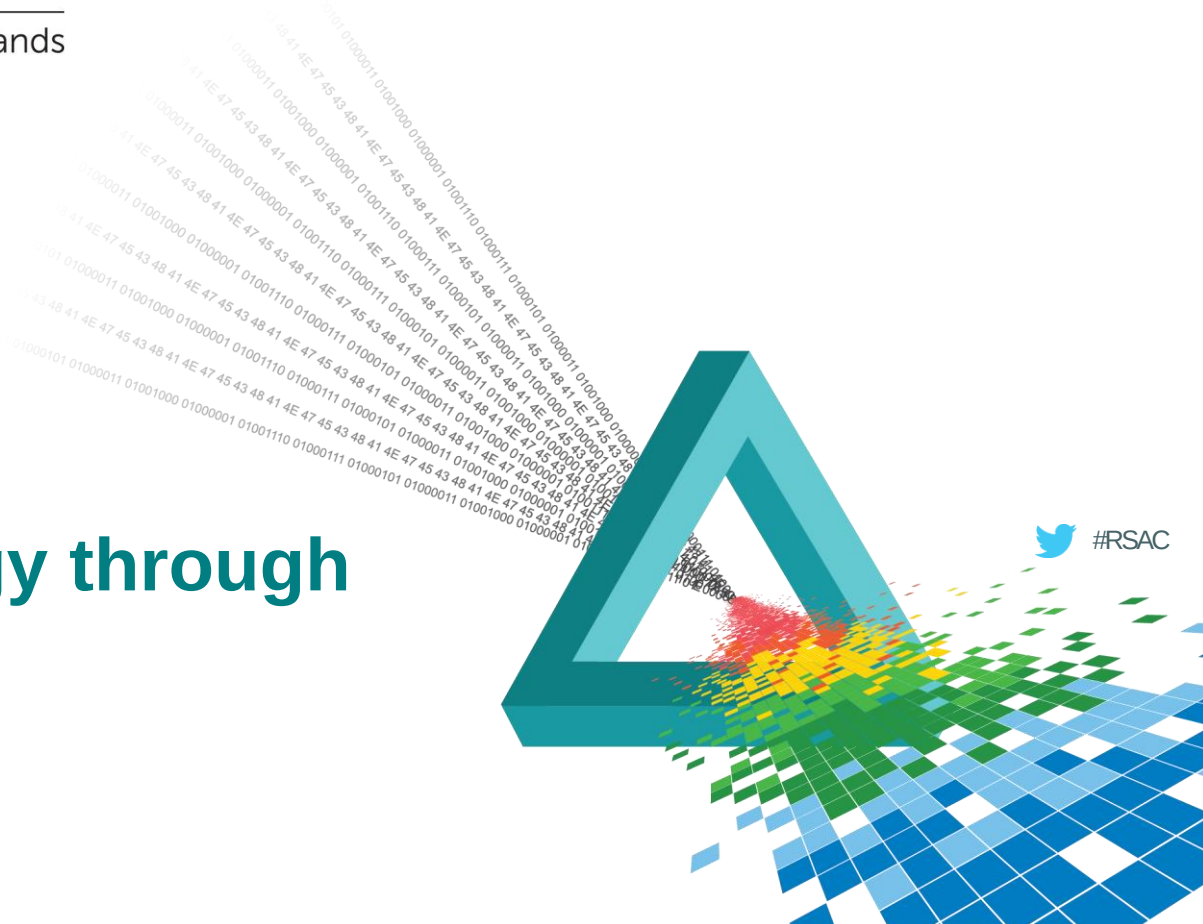
Printer Pairing without PIN / CODE and invalid receipt print

Fingerprint data storage and data transfer

RSAConference2015

Singapore | 22-24 July | Marina Bay Sands

Mitigation Strategy through Secure SDLC



#RSAC

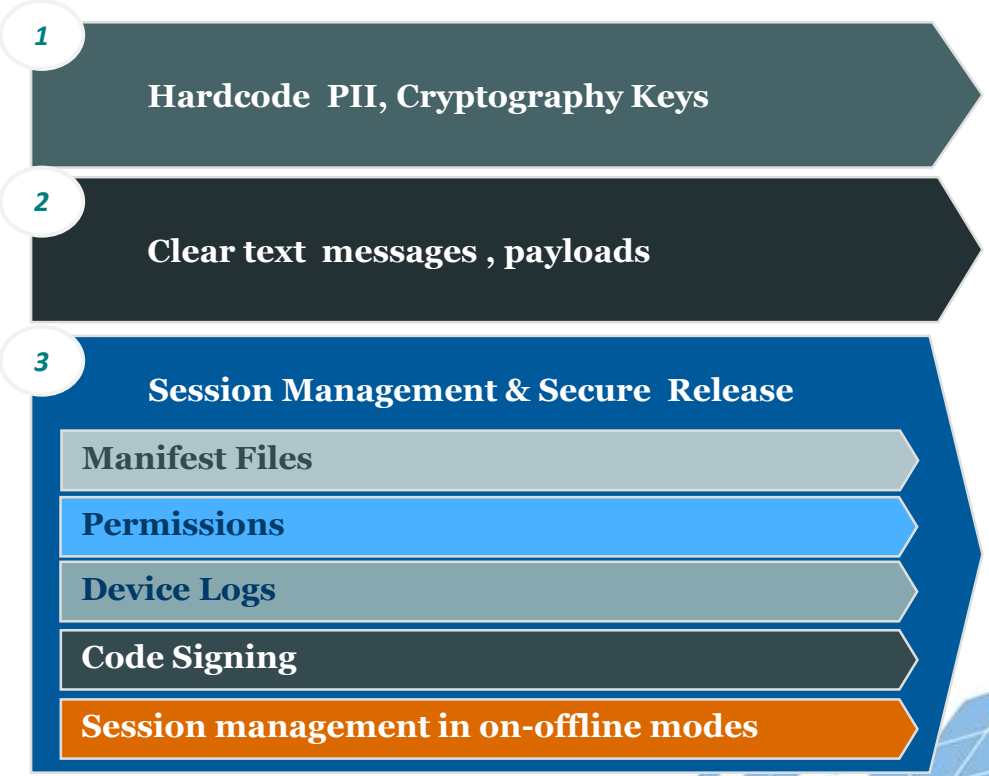
Program!

```
main()
{
int i=7;
printf(“%d”,i++*i++);
}
```

Common mistakes in source code

During payment application's source code development, below are common mistakes :

1. Hardcoded sensitive data
2. Cryptography usage
3. Exception & Error Handling
4. Logging
5. Improper Code Signing
6. Permissions
7. Configuration Files
8. Session Management



Secure SDLC Approach

Define Security Principals → ***Secure Development*** → ***Assess , Remediate and Secure Release***

Secure Design

Understanding
Understand mobile payment eco-system , business logic and customer usage

Design and Architecture
Define security principals and incorporate it in design documents and architect

Mobile Apps
Design local storage, communication channels and Platforms security

Secure Code

Payment Devices
Design device OS and App security

Source Code
Apply OWASP Mobile Security guidelines in source code development

Assessment & Secure release

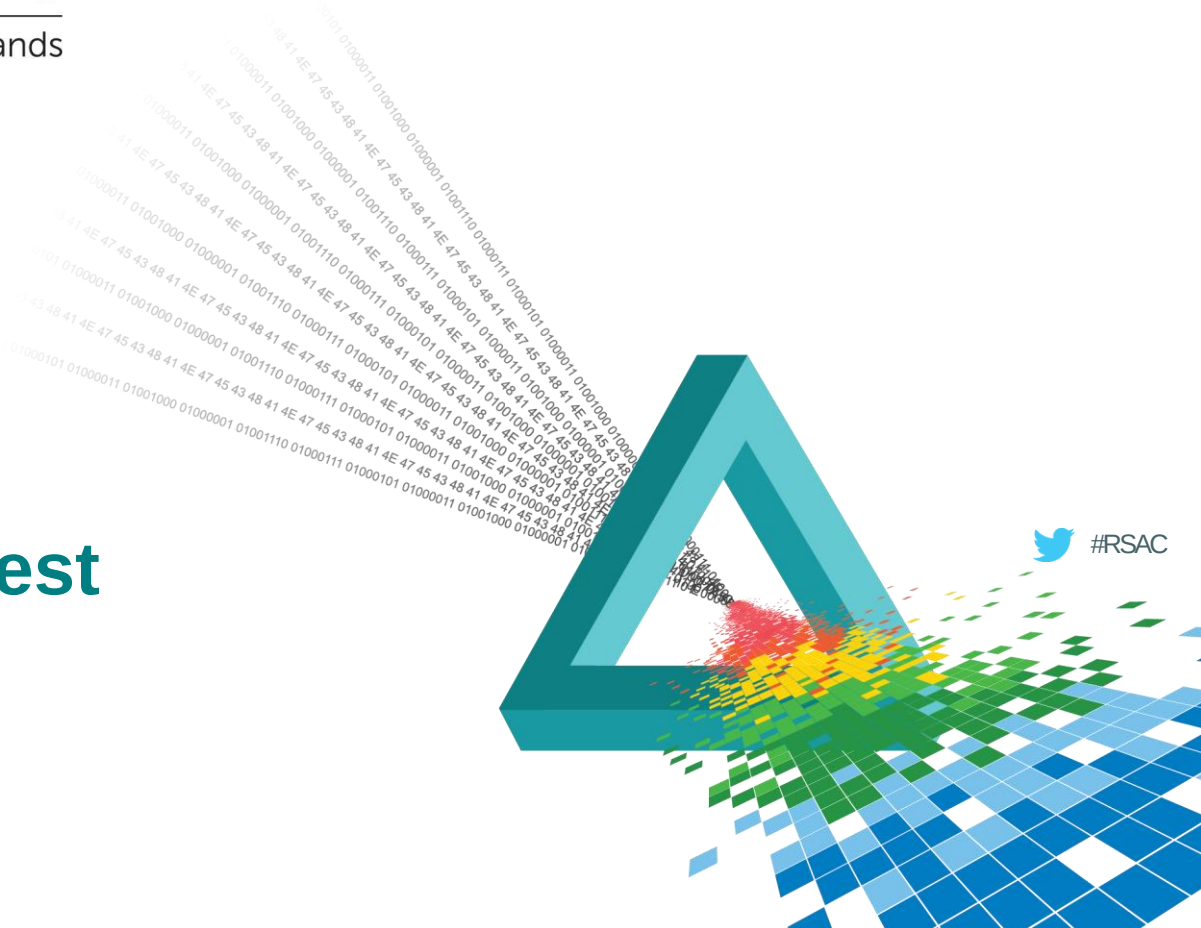
Assessment & Remediation
Assess Device and Apps and remediate security findings. Implement reusable components

Secure Release
Ensure code signing, server security, device security controls and incidence management

RSAConference2015

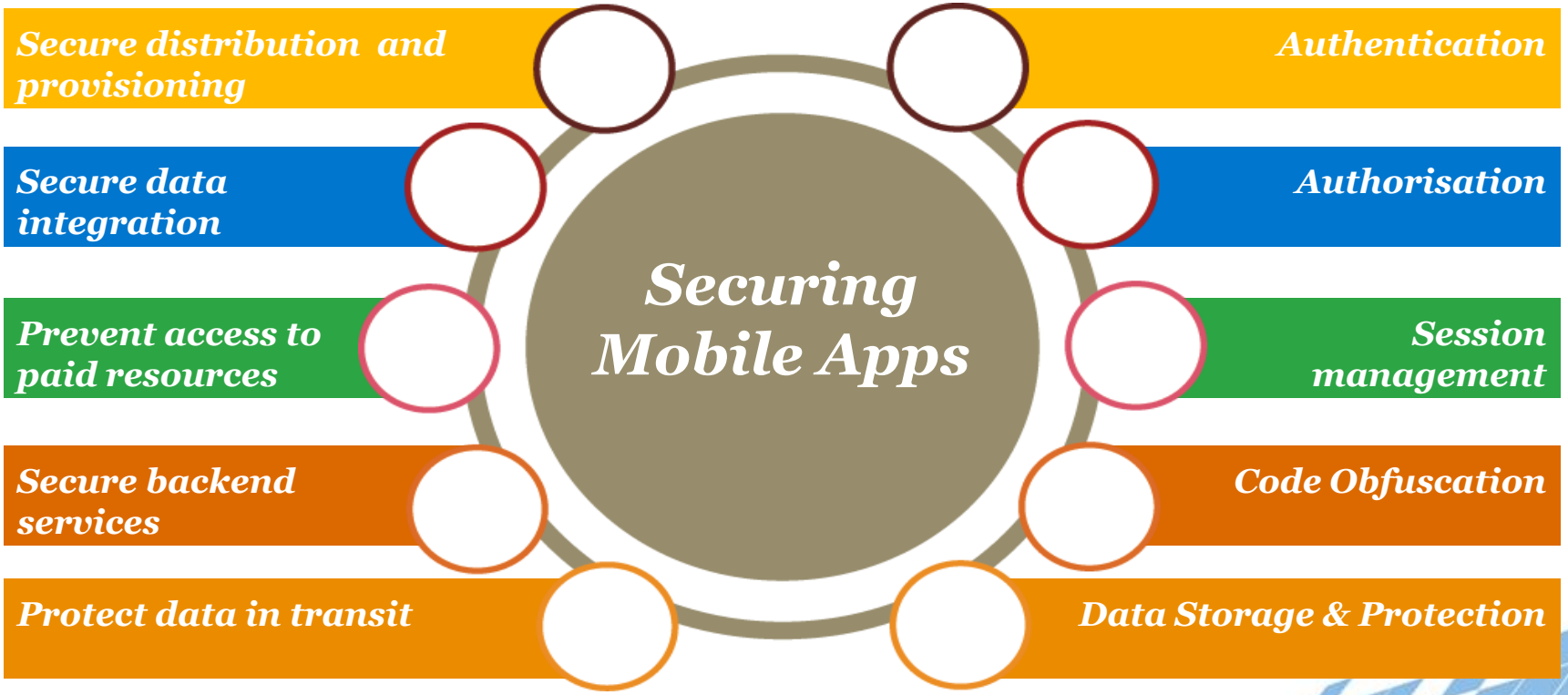
Singapore | 22-24 July | Marina Bay Sands

Mobile Security best practices



#RSAC

Mobile Security – Best Practices



Mobile Security – Best Practices

- **Authentication , Authorization and Session Management**
  - Strong **password policy**
  - Validate password and sessions if application needs to work in **offline mode**
  - Use **salted password**
  - **CAPTCHA** during registration
  - **Unique session tokens** to form valid and unique message payloads
  - NIST approved **encryption/hashing** algorithms
  - **Two factor authentication** (in case of financial transactions to be performed.)
  - **Lower timeout** for inactive session
  - Validate all messages/payloads received at backend / **mobile application server** and prevent message replay attacks. These messages/payloads should be encrypted and should have combination of padding elements, session identifiers and timestamps.

Mobile Security – Best Practices

- **Code Obfuscation**
  - **Obfuscate** all sensitive application code using either 3rd party commercial software or open source solutions where feasible
  - Implement **anti-debugging techniques**
  - Ensure **logging** is disabled as logs may be interrogated other applications with read logs
  - Hide executable code using **Address Space Layout Randomization** (ASLR)

Mobile Security – Best Practices

- **Data Storage and Protection**
  - Implement **data encryption/hashing** on the device and server.
    - Server side stored passcode to encrypt local keychain entries. Encrypted keychain values are send to the server
  - Use **NIST** approved encryption standard algorithms to encrypt the sensitive data
  - **Encryption keys** shall never be in RAM. Instead, keys should be generated real time for encryption/decryption as needed and discarded each time.
  - No sensitive data (e.g. passwords, keys etc.) in **cache or logs**
  - Use **remote wipe** APIs .
  - Do not reveal **UDID, MSISDN, IMEI** and PII

Mobile Security – Best Practices

- **Protect Data in Transit**
  - Use secure communication channels
  - Use CA provided Certificates
  - Do not disable or ignore SSL chain validation
  - Verify communication channels (USSD, SMS, GPRS, IVRS) security (e.g. Secure SMP protocol usages in case of SMS)

Mobile Security – Best Practices

- **Secure backend services and the platform**
  - Implement **Secure Backend API'S** or services
  - **Secure data transfer** between the mobile device and web-server back- ends and other external interfaces
  - Server and infrastructure **hardening**
  - Maintain and monitor **application server logs**
  - **Access control** for mobile platform

Mobile Security – Best Practices

- **Prevent unauthorised access to paid-for resources**
  - Restrict use of internal APIs (premium rate phone calls, roaming data , NFC payments) to have **privileged access** on the user's device
  - Ensure that wallet API call backs do not pass **clear text** account/pricing/ billing/item information.
  - **Logs** shall be protected from unauthorized access.
  - Check for **anomalous usage patterns** in paid-for resource usage and trigger re-authentication

Mobile Security – Best Practices

- **Secure data integration with third party services and applications**
  - Validate the third party code/libraries integration
  - Consent mechanism during application install, data transit and opt-out functionalities.

Mobile Security – Best Practices

- **Secure distribution and provisioning of mobile applications**
  - Provide applicable security updates, code fixes regularly
  - Distribute properly signed apps through authorized download centres only

Top 5 Practical Mitigations

- Server-side authentication for all sensitive information.
- Sensitive local data storage encrypted with user secret.
- Rolling key authentication linked to authorised device id.
- Code obfuscation
- Jailbreak detection

Thank You!