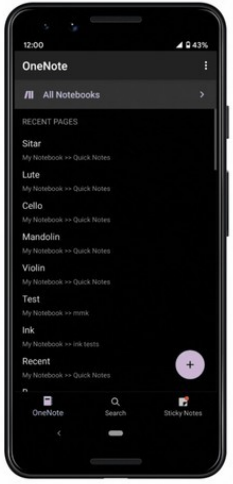
“Do Mobile Applications Mishandle Payment Information?”

A presentation based on:

"**Cardpliance: PCI-DSS Compliance of Android Applications”**

Original research by Samin Yaseer Mahmud, Akhil Acharya, Benjamin Andow, William Enck, Bradley Reaves.

Background



How can we trust businesses? How can businesses trust themselves?

Standard Practice

With no trust, consumers would be less inclined to spend money with a business.

We need a standard that helps businesses protect consumer data, and thus consumers trust businesses.

Payment Card Industry Data Security Standard (PCI-DSS)

Are We Safe?

If businesses follow the standards, we can assume relative security of our payment card data.

Samin Yaseer Mahmud and colleagues wanted to find out if this compliance is what is seen in real applications.

They conducted research to find out the how compliant a sample of android applications are with the PCI-DSS standards.

PCI-DSS

Introduced in 2004, updated as required.

Provides standards regarding safe handling and storage of sensitive card data.

Applies to different sets of data in different ways.

Applies to different businesses and business models in different ways.

Regulations

Applications must limit storage and retention time of Cardholder Data (CHD).

Includes:

- 16 digit number - 1234 5678 9012 3456
- Cardholder Name - Lucas Betts
- Expiry date – 05/21
- Service code – (relates to the magnetic stripe, not CVV)

This CHD **can** be stored, if done appropriately.

Regulations

Sensitive account data (SAD) cannot be stored. After authorisation, this data must be rendered unrecoverable.

Includes:

- Full magnetic track data
- CVV/CVV2 etc.
- PIN

This SAD **cannot** be stored in a recoverable state.

Regulations

Primary account number has special handling and display requirements.

Only first 6 or last 4 digits can be displayed, and only to those with a legitimate need to view them.

E.g. 1234 56XX XXXX XXXX or XXXX XXXX XXXX 3456

Must be stored appropriately. Recommends encryption or hashing.

Cannot be sent via SMS or other messaging services.

Regulations

Comunicating any SAD or CHD must be done securely.

Communications must take place in industry-best practice for encryption, authentication and message transmission.

Cardpliance

Samin et al. created a tool to study PCI-DSS compliance, Cardpliance.

Studied 358 applications which require payment card data.

Mixture of UI analysis and static program analysis to flag potential PCI-DSS violations.

All Cardpliance results verified manually to ensure no false positives.

Cardpliance – Difficulties

Payment card data usage is implementation specific.

PCI-DSS is context sensitive.

No common API’s to analyse calls.

PCI-DSS is imprecise.

Cardpliance – Methodology

Uses a tool called Amandroid to perform context-sensitive static analysis.

Generates “Data Dependance Graphs” to analyse how data flows through the app.

Assessed applications against 6 tests, based on the condensed regulations given previously.

Cardpliance – Results

Do Mobile Applications Mishandle Payment Information?

Overall, ~1.67% of the applications assessed were verified to be non-compliant.

Only one application was flagged for more than 3 of the 6 tests.

Still – are any violations acceptable?

Solution #1: External Providers

Application ecosystems (Google Play Store, Apple App Store) have their own payment handling systems.

Other third-party payment handling systems exist, e.g. PayPal, Windcave (formerly Payment Express), Stripe.

Reduces the liability of the business, and the expertise required by the developers.

Solution #2: Protect CHD/SAD

Both CHD and SAD should never be stored in plaintext.

Cardpliance discovered applications that log sensitive data in plaintext, or store it in plantext on the device.

This data, if PCI-DSS has provisions for storage, should be stored encrypted or hashed only.

Solution #3: Leave SSL Alone

Another finding of Cardpliance is that applications sometimes modify security API's, such as SSLSocketFactory and TrustManager.

Third party libraries also sometimes change the behaviour.

These should be left alone, or only modified if there is a reasonable use-case and security is a high priority.

Solution #4: Explicit Intent

Payment card information should be shared via explicitly addressed intent messages.

Implicitly addressed intents may cause unintentional information access. Even if not malicious, this still violates the regulations.

Explicit intents start an activity of a specific class.

Implicit intents ask the system to find an activity with a registered handler to handle the request.

Sources:

PCI-DSS version 3.2.1
https://www.pcisecuritystandards.org/documents/PCI_DSS_v3-2-1.pdf?agreement=true&time=1621249541954

Cardpliance: PCI-DSS Compliance of Android Applications
https://www.usenix.org/system/files/sec20-mahmud.pdf

Questions?

Please ask any questions now.