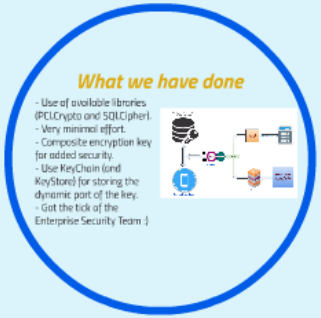
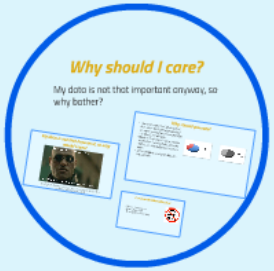
1

85% of security breaches occur due to security holes in the application, while only 15% of the security budget spent on the software dev.



2

You could secure your app (and users' data) by possibly a couple of lines of code.



Let's Hack It

Securing data on the mobile. The why, how, and what

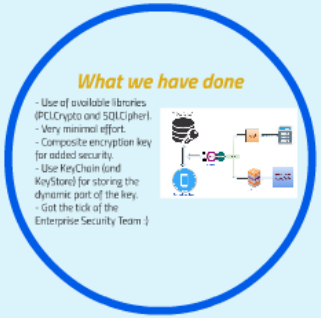
3

Attackers are well equipped with a high computing power that's cheaper than ever before. Plus, you could be breaching the law.

Prepared by: Has AlTaiar
http://www.hasaltaiar.com.au
6th May 2015

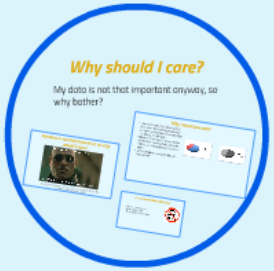
1

85% of security breaches occur due to security holes in the application, while only 15% of the security budget spent on the software dev.



2

You could secure your app (and users' data) by possibly a couple of lines of code.



Let's Hack It

Securing data on the mobile. The why, how, and what

3

Attackers are well equipped with a high computing power that's cheaper than ever before. Plus, you could be breaching the law.

Prepared by: Has AlTaiar
http://www.hasaltaiar.com.au
6th May 2015

Agenda

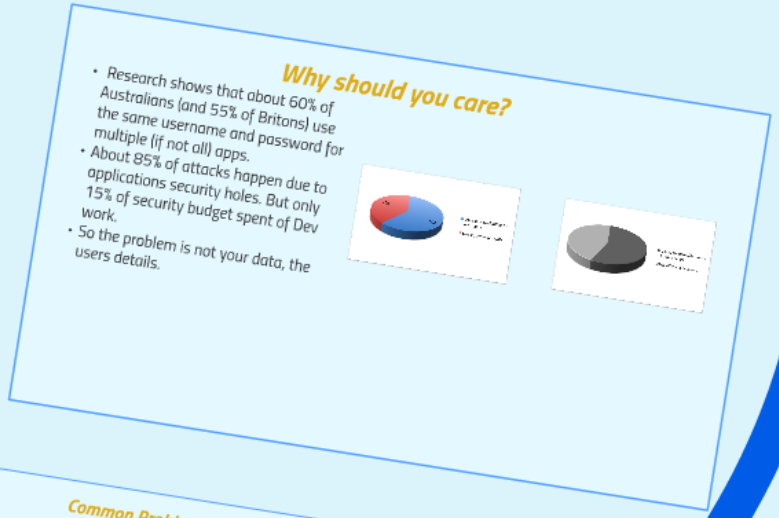
- Why should I care?
- I can fix it
- Common Flaws
- What we have done
- Reusable Blocks
- Current Hardware state
- Key Takeaways

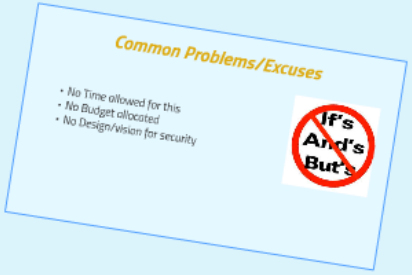
Disclaimer

I am not any security guru, and certainly do not claim to protect anybody's data.
This is just to share my experience from recent projects. No responsibility is taken...

Why should I care?

My data is not that important anyway, so why bother?



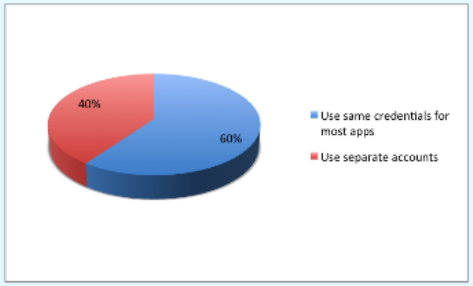


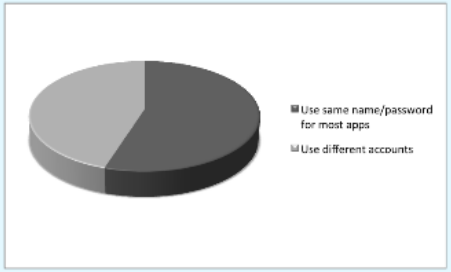
My data is not that important, so why would I care?

According to the Australian privacy law, you could fined up to $1.7M ($1.7M for organisations and $340K for individuals)

Why should you care?

- Research shows that about 60% of Australians (and 55% of Britons) use the same username and password for multiple (if not all) apps.
- About 85% of attacks happen due to applications security holes. But only 15% of security budget spent of Dev work.
- So the problem is not your data, the users details.



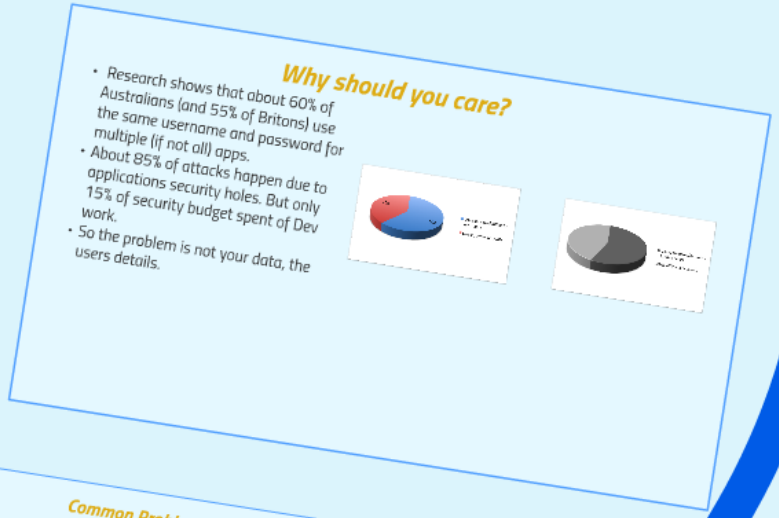


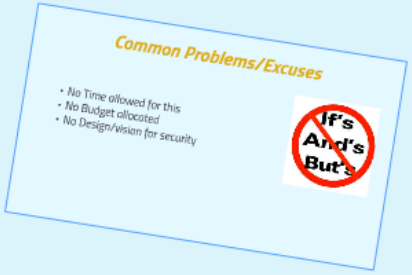
Common Problems/Excuses

- No Time allowed for this
- No Budget allocated
- No Design/vision for security

Why should I care?

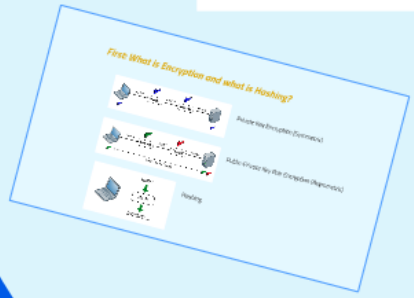
My data is not that important anyway, so why bother?



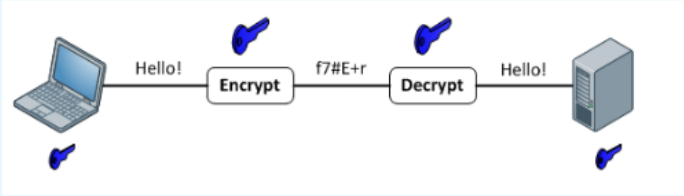


I can fix it :)

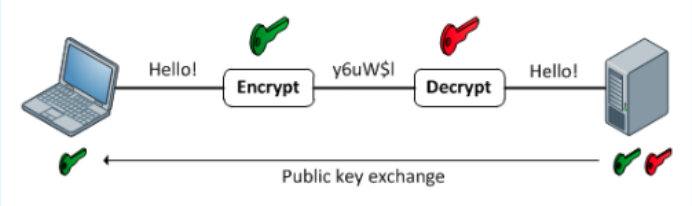
Just Encrypt/Hash it



First: What is Encryption and what is Hashing?



Private Key Encryption (Symmetric)



Public-Private Key Pair Encryption (Asymmetric)

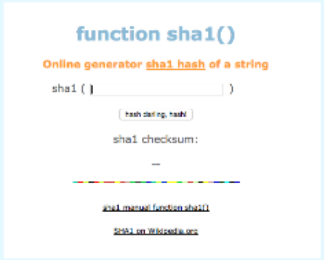
Hashing

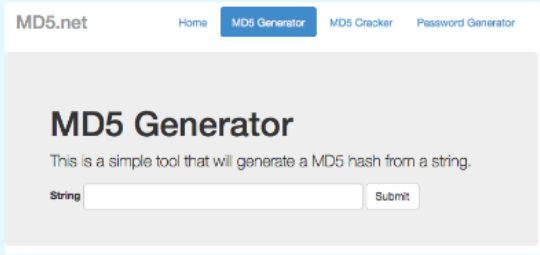
Common Flaws

- Using weak encryption or hashing algorithms (MD5 lookup SHA1,2 lookups, crackstation.net)
- Storing keys/passwords in iOS KeyChain/Android KeyStore (Not very safe, you can view contents on jailbroken devices, other apps can read other apps keys, I have done it myself :))
- Hard coded encryption keys (Xamarin.Auth)? Ooops!
- Using in-house developed algorithms.
- Using outdated protocols or libraries.
- Only Hashing Data (need to add some salt).
- Too Tight implementation, pushes people to share passwords and email keys.

Feel good statements

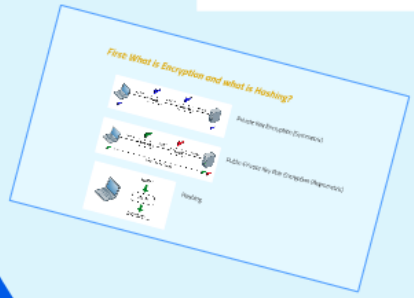
- More than 70% of Australians think that it is very hard to guess their passwords.
- I hashed the passwords, so they are safe. Or Are they?
- I have encrypted the sensitive data, so it's all good.
- I have salted the hashed values





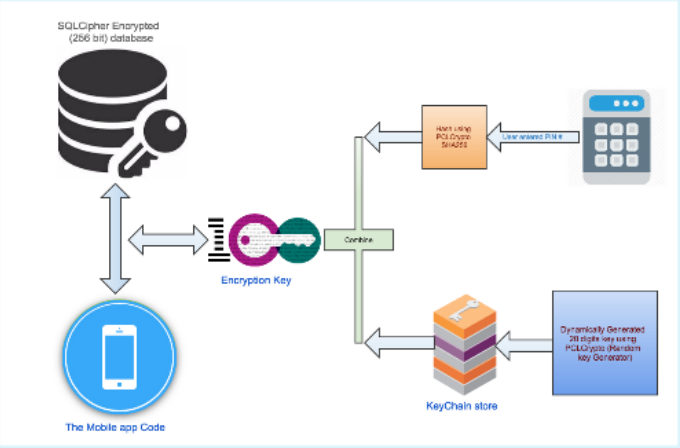
CrackStation uses massive pre-computed lookup tables to crack password hashes. These tables store a mapping between the hash of a password, and the correct password for that hash. The hash values are indexed so that it is possible to quickly search the database for a given hash. If the hash is present in the database, the password can be recovered in a fraction of a second

Just Encrypt/Hash it



What we have done

- Use of available libraries (PCLCrypto and SQLCipher).
- Very minimal effort.
- Composite encryption key for added security.
- Use KeyChain (and KeyStore) for storing the dynamic part of the key.
- Got the tick of the Enterprise Security Team :)



Reusable Blocks

PCL Crypto

- Open Source
- Available on so many platforms.
- Provide implementation of good encryption/ hashing algorithms.
- Uses same API that we are familiar with on .NET and Windows.
- Available for PCL libraries too (partial functionality).
- Many features (Cryptographically strong random number generator, Symmetric and asymmetric encryption and signatures, Key derivation, Native crypto performance (2-100X faster).

```
PM> Install-Package PCLCrypto
```

Reusable Blocks

KeyChain.NET

- Unified API for iOS and Android.
- Simple and easy-to-use.
- Comes with Android implementation for storing, accessing, and deleting keys.
- Available as a nuget package, just install and use it.
- highly customizable.

```
PM> Install-Package KeyChain.Net
```

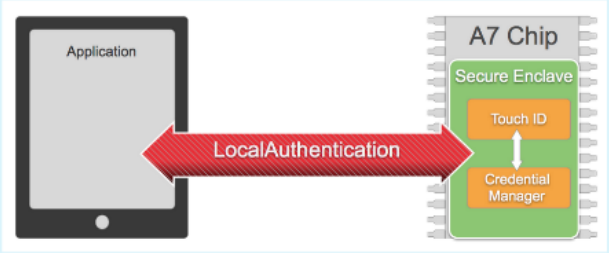
Reusable Blocks

SQLCipher

- ADO.NET and sqlite-net compatible APIs
- Well Validated and Tested.
- Available as a component on Xamarin store (no further dev needed).
- Uses sophisticated encryption libraries (256-bits AES).
- Optimized for better performance.
- Same API on iOS and Android
- Simple configuration
- 100% of database is encrypted
- Low overhead encryption, often as low as 5-15%

Hardware State

iOS
- Devices that supports Touch ID has a specialised A7 Chip.
- iOS 8 KeyStore added ACL Lists.
- iOS 8 introduced restrictions to accessing keys.
- Still need to support older devices



Hardware State

Android
- Some Vendors (Samsung starting from Galaxy S5) added biometric sensors to the device for authentication.
- Highly fragmented, multiple APIs from vendors and from Google Android team.
- The implementation of the KeyStore on Android leaves it to developers to implement, encrypt and name the file for their keys.
- Need to consider devices that do not have bio-sensors

Key takeaways

- Never store users' data without properly protecting it.
- Review your encryption/hashing algorithms, how strong they are? how long will it take to break them?
- Enforce strong questions and verifications for resetting passwords, security questions? strike a balance (convenient vs secure)
- Use VALID Certificates on the server side for encryption.
- Experiment with the so many available open source libraries for encryption/hashing and other common tasks.
- Be proactive and do the minimum when you do not have time.

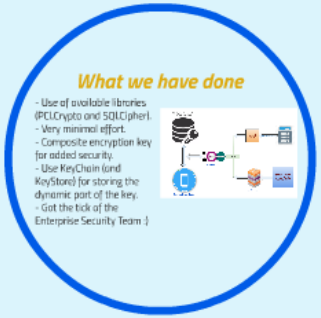
References

- http://www.troyhunt.com/2011/12/free-ebook-owasp-top-10-for-net.html
- http://www.biometricupdate.com/tag/android
- https://www.paypal-media.com/assets/pdf/fact_sheet/cis_paypal_whitepaper_final.pdf
- https://nakedsecurity.sophos.com/2013/04/23/users-same-password-most-websites/
- https://github.com/aarnott/pclcrypto
- https://components.xamarin.com/view/sqlcipher-for-xamarin-ios
- https://crackstation.net/
- http://www.shred-x.com.au/privacy-act/privacy-law-reform/
- http://www.hasaltaiar.com.au/announcing-keychain-net-a-unified-api-for-using-keychain-on-many-platforms/


iPad
9:41 AM
OWASP Top 10 for .NET developers
Troy Hunt
Microsoft
Most Valuable
Professional

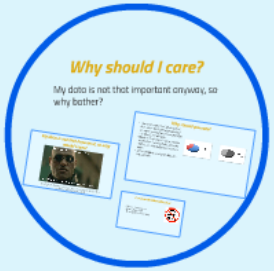
1

85% of security breaches occur due to security holes in the application, while only 15% of the security budget spent on the software dev.



2

You could secure your app (and users' data) by possibly a couple of lines of code.



Let's Hack It

Securing data on the mobile. The why, how, and what

3

Attackers are well equipped with a high computing power that's cheaper than ever before. Plus, you could be breaching the law.

Prepared by: Has AlTaiar
http://www.hasaltaiar.com.au
6th May 2015