CREATING ZERO-KNOWLEDGE SAAS APPLICATIONS

WHAT IT IS, WHY YOU WOULD WANT IT

Created by Chris Foster / @chrisfosterelli

WHO AM I?

- Software Developer
- TRU BCS Student
- TRUSU Computer Science Club
- Startup Weekend Kamloops
- Kamloops Innovation Centre fanatic
- Security enthusiast

THE PLAN

Things we will be covering:

- What is Zero Knowledge?
- How does it work? What are the benefits?
- Case studies of Zero Knowledge
- Implementing your own Zero Knowledge
- My experience
- Questions

WHAT IS ZERO KNOWLEDGE?

- Buzzword
- Data encrypted/decrypted by the client
- Server stores encrypted data
- Encryption key is derived from the user

WHY IS ZERO KNOWLEDGE GOOD?

- Doesn't know anything about your data
- Can't reveal anything about your data
- Can never access your data

ZERO KNOWLEDGE VS. TYPICAL ENCRYPTION

- Who holds the keys?
- In Zero Knowledge: you do
- In 'typical' encryption: your SaaS provider does

ASYMMETRIC & SYMMETRIC

- Symmetric encryption: user is accessing their own data
- Asymmetric encryption: users are sharing data (hard)

DOWNSIDES

- Users loose keys more often than you
- Debugging can be *much* more difficult
- Development requires more work

DETERMINING ZERO KNOWLEDGE

- Typically - you can't
- Requires decent cryptography skills
- Usually best to rely upon experts
- Use evaluated products
- "Who holds the keys?"

CASE STUDY: DROPBOX

- Dropbox encrypts your data on their servers
- Dropbox encrypts your data in transit
- Dropbox is not Zero Knowledge

CASE STUDY: SPIDEROAK

- Spideroak encrypts your data on their servers
- Spideroak encrypts your data in transit
- Spideroak is Zero Knowledge *most* the time

CASE STUDY: LASTPASS

- Lastpass gets critized often
- Actually *extremely* well done
- Javascript app manages all encryption

CREATING YOUR OWN

- You **must** have a client application
- There are very flexible options

BEING A 'STUPID' DATA STORE

- Very simple API
- Simply inserts and removes data from the database

USING JAVASCRIPT

- Handling this on the web can be done with Javascript
- This has potential pitfalls and many security guys dislike this
- However, this is better than nothing

SIGNUP

- A user must generate an encryption key on signup
- The encryption key can be encrypted with their password
- The encryption key can be their password
- The user pushes this key to the server

PROVING AUTHENTICATION

- How can we identify a user without a password?
- Ask them to prove a secret to us!
- This typically makes asymmetric encryption more attractive

CASE STUDY: MY STUFF

- At signup, generate two asymmetric key pairs
- One for phone, one for web interface
- Both key pairs are encrypted with the password
- Both key pairs are stored on the server

PHONE -> WEB INTERFACE

1. Phone encrypts message with the web interface's public key
2. Phone signs the message with its private key
3. Phone posts the message to the REST API
4. Web interface receives via socket.io
5. Web interface decrypts with its private key
6. Web interface verifies it is signed from the phone
7. Display message

WEB INTERFACE -> PHONE

1. Web interface encrypts message with the phone's public key
2. Web interface signs the message with its private key
3. Web interface posts the message to the REST API
4. Phone receives via Google Messaging
5. Phone decrypts with its private key
6. Phone verifies it is signed from the web interface
7. Phone displays message

IMPLEMENTATION TIPS

- **Do not** roll your own encryption, *ever*
- Look into password recovery options
- Make the process transparent to the user
- Open source when you can

THE WORLD NEEDS A SUPER HERO

- The problem with Zero Knowledge: it is geeky
- Zero Knowledge companies don't know how to market
- Zero Knowledge shouldn't be geeky, it should be necessary
- Users *will not* bother if it requires extra work
- We can fix this

END

@chrisfosterelli

http://fosterelli.co/