Securing web applications with minimal resources

A journey of risk taking 😇

Disclaimer

Might contain errors, no guarantees

Opinions expressed are solely those of the presenter

Presenter is not liable for any damage or loss occurring by following any advice in this presentation

About me

Software developer

Both client and server

Web products

Open source/Free software 👍

Security on a budget 

Sigh...

Security

“Generic” user authentication

APIs

VPCs, IP restrictions

Dependencies

Why on a budget?

“Small” products

Competition

Lack of resources

Lack of time

Any other valid excuse 🥺

1. *Public* vs *Private* sectors

Understanding constraints

1.1. Innovation

Public not ready *yet*

Quick adoption in *Private*

1.2. Failure

Eg: Verify - going *private* 2020 April

1.3. The looks UI/UX

Bottom of priorities

Lack of skills

1.4. Compatibility

Out of date OS’es

Older browser versions

Customer is the bottleneck

Global stats do not help

1.5. Alternate factors

Do they have phones?

Notifications out of hours?

1.6. Bring your identity

Which one?

Did you say Facebook?

1.7. Integrating

Slow pace

Old, undocumented

Restricted access

Run!

1.8. Sensitive data

GDPR helps

Forms contain a lot of it

1.9. Getting sued

Did you do everything in your power to prevent … ?

2. Adding authentication

Website needs it again

2.1. It’s just a prototype

Auth flows

Usually user/pass + social login

2.2. Gotcha, no specs

Could IT be involved?

2.3. Bike-shedding (Law of triviality)

Auth is one of first steps in the flow

Can it be “better”?

Reset password link does not work

2.4. Choice of lang/framework/plugin

Has to work with the prototype 😭

Not the time to experiment

Accommodating random integration paths

Battle-tested

2.5. Storing auth details

Transactionality

Password hashing + random salt

SQL injections

2.6.Isolating DB

VPCs

Connections from outside?

Connections from application servers

2.7. Secrets in application

Embed in CI/CD? - encryption necessary

Env variables? - who/how?

Storing in code? - where is code?

Are you watching logs?

2.8. Session

Stateless servers

Client vs Server

Token based auth

2.9. Data in the client

Controversial

Necessary in offline scenarios

People do not understand it

Synchronisation is complicated

2.10. Almost done?

We’re just getting started

3. Modifications

Client needs some changes

3.1. New requirements

Integrating with another API

Merge user details

3.2. Improvements

2FA

Exponential backoff

OTP

3.3. Are we done yet?

nope

4. It’s still great. No!

Some stuff left to do

4.1. Assessing known risks

Review each part

What could be wrong?

4.2. Automated vuln. checks

Those dependencies

4.3. Scanning code for bugs

IDEs

Automating in CI

4.4. Leverage shared responsibility

Cloud providers have documentation

4.5. Rotating keys

You do that don’t you 😜

4.6. Planning for the worst

Cloud provider has some useful docs

Prepare playbooks/scripts

4.7. Tests

Hire penetration testers

Should pay off if you go to court

5. Could we do better that that?

Third party auth (TPA) - sirens are singing

5.1. Third party auth (TPA)

Ticking all the boxes

Known risks vs unknown ones

5.2. Your users over there

Outsource user tables

Client uses API anyway

5.3. Great dev resources

Are they selling to devs?

5.4. Freemium

How likely your app will have 10k users?

5.5. Great UI

Consistent views

Easy branding

Click and go

5.6. Perks

Anomaly detection - it was not in our list but sounds great

DDoS protection

Other stuff in whitepapers

5.7. How much?

Who knows

6. New risks

What could go wrong with TPA?

6.1. Vendor lock-in

But I’m already using XYZ

6.2. Limitations of extensibility

Once upon a time there was a system.

It still is.

You need to integrate with it.

6.3. Complicated domain model

Surrogate keys

Synchronisation - are we in eventual consistency land?

6.4. Where to store *other* details?

Do I store sexual preferences or mental health history in TPA database as well?

6.5. TPA API is a bottleneck

Synchronisation over HTTP API

Merging records on the fly

6.6. Access control through UI

Who has access to that UI?

Surely passwords are rotated for that 😉

6.7. Opaque pricing

Is it cost effective?

7. New vs old issues

How many problems do TPAs solve?

7.1. Feature matrix

No space, you have to imagine it

Old issues + risks

New issues + new risks

7.2. What is it next time?

Invest in your devs 🧠

Automate/observe more □

Limit third parties 🚫

Hold hands with security experts ❤️ 💳

Appendix

Me - www.ivarprudnikov.com or twitter.com/ivarprudnikov or github.com/ivarprudnikov

Slides: