KF4009 Web Technologies
Information security issues — part 1

The Royal Academy of Engineering funded a Visiting Professorship in Practical Cybersecurity Insights at Northumbria University, 2019–2022. These slides are a slightly modified version of those delivered, intended to be available after the project has ended

Web `https://green-pike.co.uk/nvp`
Email p.brooke@northumbria.ac.uk (until it stops working. . .)
phil@green-pike.co.uk

Why this lecture?

. . . because I spend a lot of my time dealing with web technologies
There are lots of problems. . .

Why me?

An "information security professional" with experience in academia, the public sector, consultancy. . .
Generalist computer scientist and software engineer

This lecture

- Web apps overview
- Database issues
- Simple validation and host security

Web apps

Application areas?

What sorts of things are web apps used for? Make a list of examples of applications!

Technologies?

Which technologies are used in web apps? Again, make a list!

. . . and move onto the next video after you've thought of some examples

Web apps

Example application areas

Banking, ecommerce, forums & social media, entertainment, encyclopedias & dictionaries

Common technologies

- Web browser, JS, CSS, HTML
- Network
- Web server, possibly load-balanced
- Business logic, *e.g.*, PHP, Python, ...
- Database(s)
- Underlying OS
- "Bare metal", possibly with hypervisor or container

Typical introduction to security properties

Confidentiality disclosure of information and unauthorised reads (*e.g.*, medical records)

Integrity unauthorised writes or destruction (*e.g.*, modifying a cheque from 'pay £20' to 'pay £2000')

Availability access to information systems when it is required (*e.g.*, DoS)

Why do we care?

- Confidentiality
- Integrity
- Availability

Failure in one or more of these areas has an *impact*

That impact may be economic, cause physical harm to people or damage property, damage reputation or cause embarrassment

More motivations for studying security

Protecting customer privacy

- Important from the customer's viewpoint
- Not always from an organisation's (*cf.* Schneier's concept of 'externality')
- GDPR in 2018 put reputational harm on the executive agenda

Profits!

- Cheating in games is often security-related. If not dealt with, are players going to spend money?
- E-commerce: risk of fraud reduces customer confidence $\Rightarrow$ reduced sales

A functioning society

Vote-counting, ID cards, freedom of speech, "chilling effects"...

More motivations

Think about all the systems you are interacting with already today, *e.g.*, Blackboard — a huge web app

Partial list of a tiny organisation's web app systems

- Two different instances of Firefox Send
- Redmine issue tracker
- Two different forums (Rocket.Chat and Discourse)
- Multiple customer systems (mostly Python and Flask)

. . . plus reverse proxy web servers, *etc.*
Huge amounts of complexity — lots to go wrong!

Rest of this lecture

- Databases — why we care about securing them
- Simple validation

Impacts of database compromise

(Assuming you know you've been compromised. . .)

- GDPR reporting obligations (end-users, data subjects, ICO)
- Reputational harm, loss of income, . . .
- Cost of remediation, rebuild

Impacts of database compromise — passwords

You should not store plaintext passwords (store salted hashed passwords instead)

Why?

Impacts of database compromise — passwords

You should not store plaintext passwords (store salted hashed passwords instead)

- (Obvious) discloses passwords for the compromised system
- *Internal malfeasors* could abuse the passwords
- *Users often re-use passwords*
  $\Rightarrow$ Means we might need users to reset passwords across *many* systems
- Databases of compromised usernames / emails and passwords are available via multiple "black" forums/sites, including from "old" breaches

Impacts of database compromise — passwords

Examples:

- https://nakedsecurity.sophos.com/2013/11/20/serious-security-how-to-store-your-users-passwords-safely/
- https://cloud.google.com/blog/products/g-suite/notifying-administrators-about-unhashed-password-storage
- https://krebsonsecurity.com/2019/03/facebook-stored-hundreds-of-millions-of-user-passwords-in-plain-text-for-years/

Validation

Input *validation* and *sanitisation* is critical

- Prevent database (SQL) injection
- Prevent tampering / injection into later page generation
  e.g., cross-site request forgery (CSRF)
  cross-site scripting (XSS)
  pop-ups
- (XKCD Bobby Tables — "Exploits of a Mom" — `https://xkcd.com/327/`)

Where to validate?

1. Should you validate input on the client-side? (Why?)
2. Should you validate input on the server-side? (Why? What if you already validate on client-side?)

Classic host / web server security

Simple starting point: protect the network perimeter by closing everything then start opening ports

- Close all unneeded ports
- Check that you closed all unneeded ports...
- Use HTTPS
  - Use a good certificate authority (no self-signed certificates!)
- Redirect HTTP to HTTPS (or reject HTTP)
- For some types of processing, consider refusing downgrade of SSL cipher suites, *i.e.*, *require* a good standard to access the system at all

Other perimeters

- People interact with computers and other people
  Enables *social engineering*
- Physical security of computers
  Where are the servers? In a reputable company's environment? In a cheap'n'cheerful shed? Overseas in a "hostile state actor's" control?
  Could I just steal them... or the backups...

Next time

- HTTP horrors
- Corporate security irritations
- More on validation and character sets
- Packaged web apps
- Attacker viewpoint

The end

Web	`https://green-pike.co.uk/nvp`
Email	p.brooke@northumbria.ac.uk (until it stops working. . .)
	phil@green-pike.co.uk

Phil Brooke
A *Royal Academy of Engineering Visiting Professor* in Practical Cybersecurity Insights
Northumbria University