Introduction to Java Web Security

Michael Dowden
@mrdowden

Goals

- ➔ Know some Java security tools
- ➔ Understand some common attacks
- ➔ Prepared to implement basic software security
- ➔ Able to research security topics

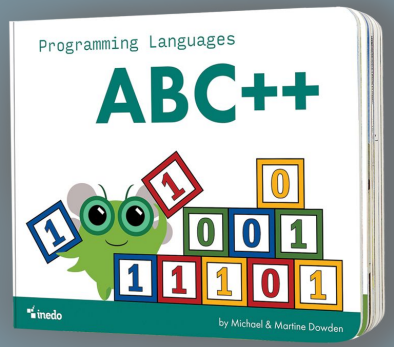
Overview

- ➔ Java Security Tools
- ➔ Attack Vectors
- ➔ Security Principles & Terminology
- ➔ Common Attacks
- ➔ Implementation Examples

Michael Dowden

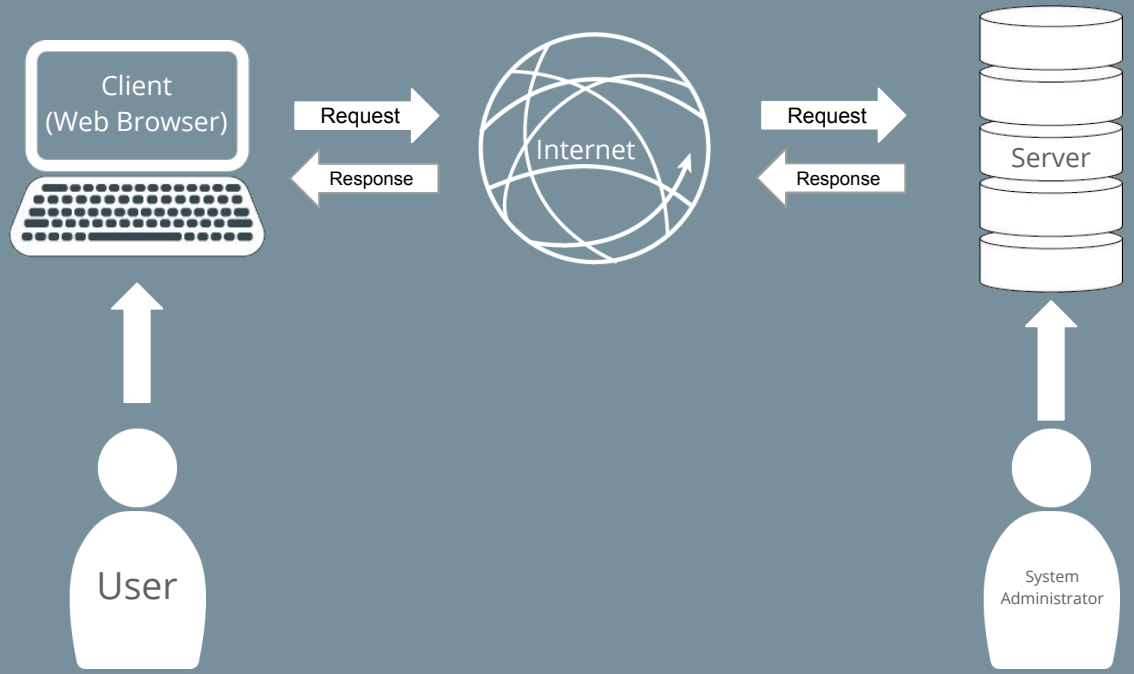
Senior Principal Consultant & Software Architecture Lead @ CSpring

- Education
  - BS Computer Science
  - MBA Entrepreneurship
- Experience
  - Software Development and IT since 1992
  - 12+ years software security
  - Full Stack - Hardware to User Interface
  - Worked with 60+ organizations in multiple industries



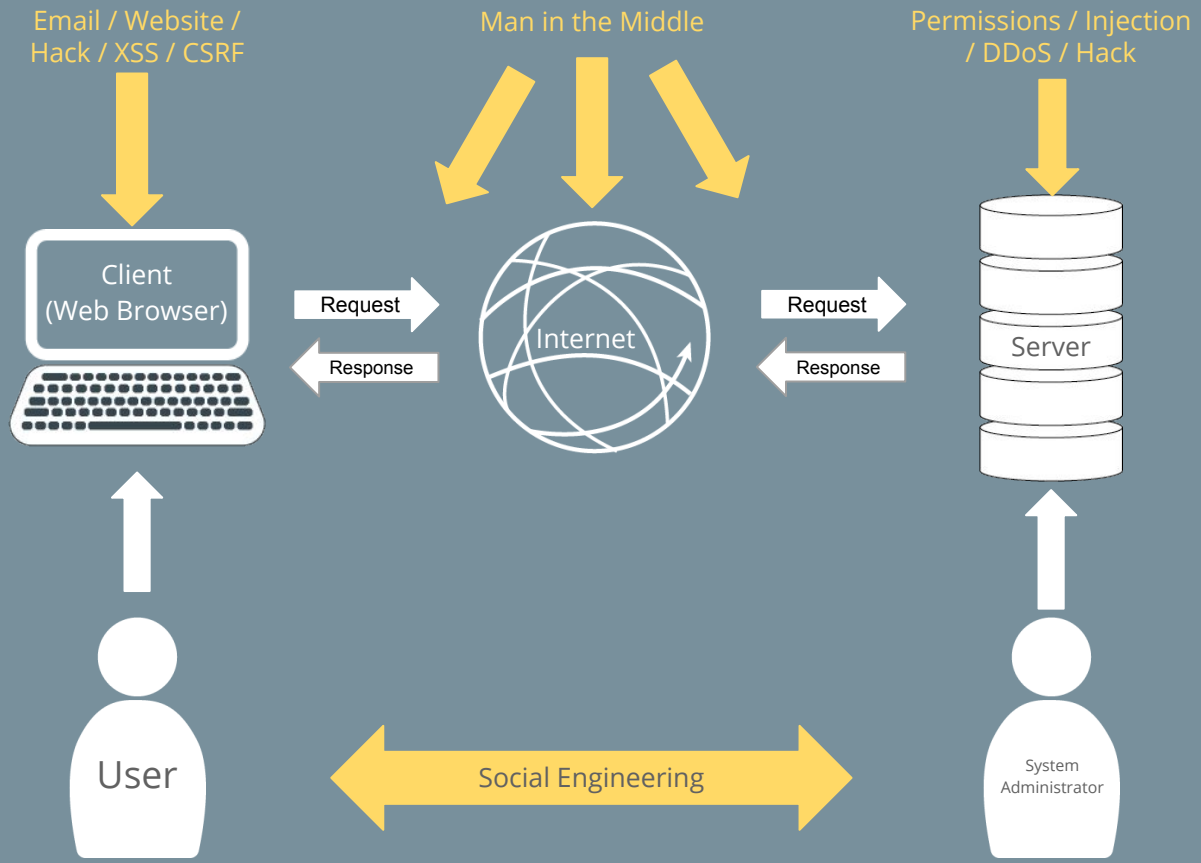
Internet Security

HTTP

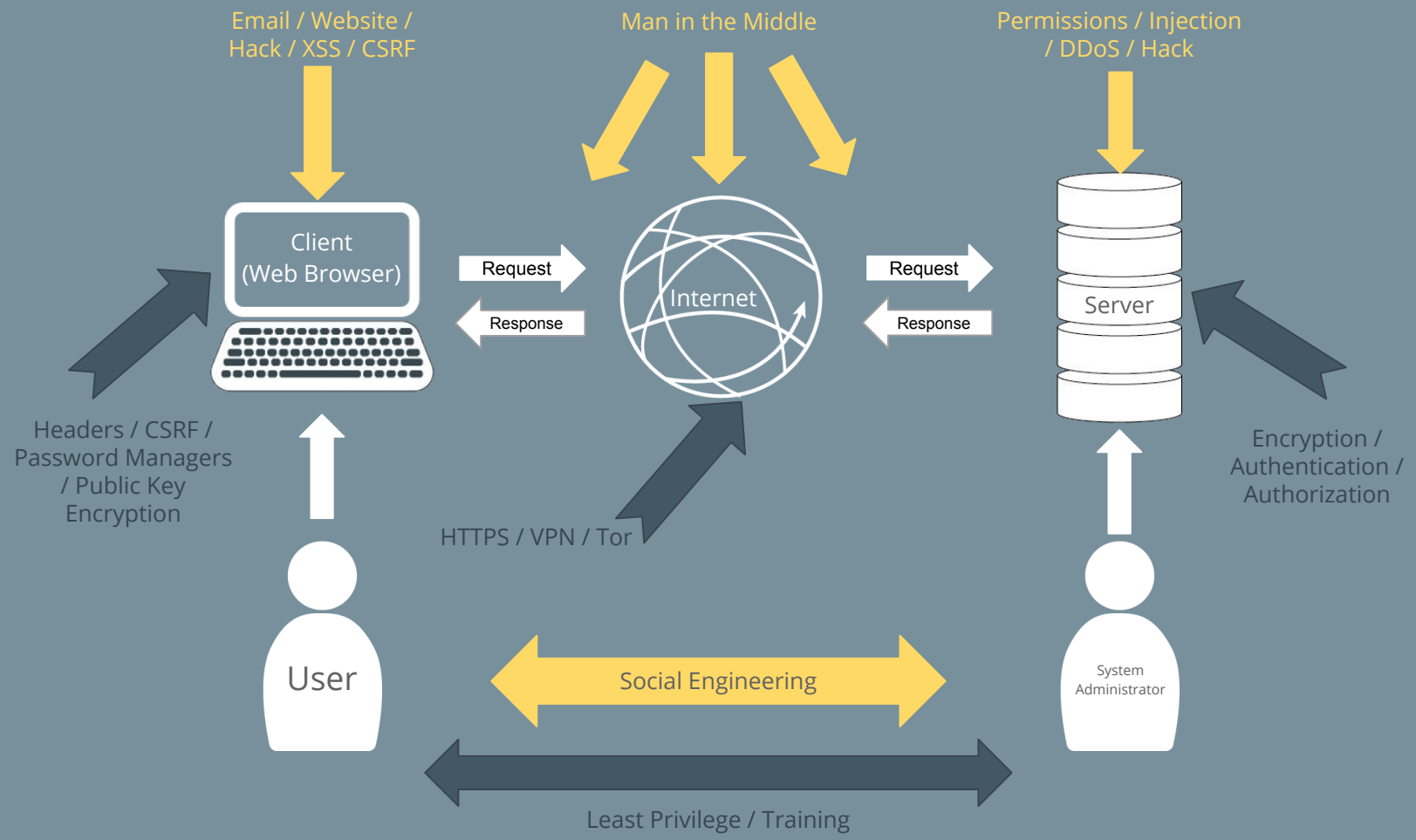

Client
(Web Browser)
Request
Response
Internet
Request
Response
Server
User
System
Administrator

Attack Vectors

...and modes of delivery



Software Mitigation



Security Topics

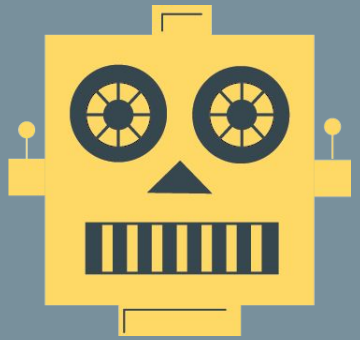
Java Security Tools

- JCA / JCE - https://goo.gl/qhlxLn
  http://docs.oracle.com/javase/8/docs/technotes/guides/security/crypto/CryptoSpec.html
- Spring Security - https://projects.spring.io/spring-security/
- Apache Shiro - https://shiro.apache.org/
- Bouncy Castle - https://www.bouncycastle.org/
- Jasypt - http://www.jasypt.org/

Key Objectives of Security

- ➔ Ensure users are who they claim to be...with every request
- ➔ Users can do what they need...but no more
- ➔ Data is kept safe
- ➔ Communication is kept private

Attack Goals



Auth & Session

Authentication

- ➔ Identity
- ➔ Something you Know (password)
- ➔ Something you Are (biometrics)
- ➔ Something you Have (security key)

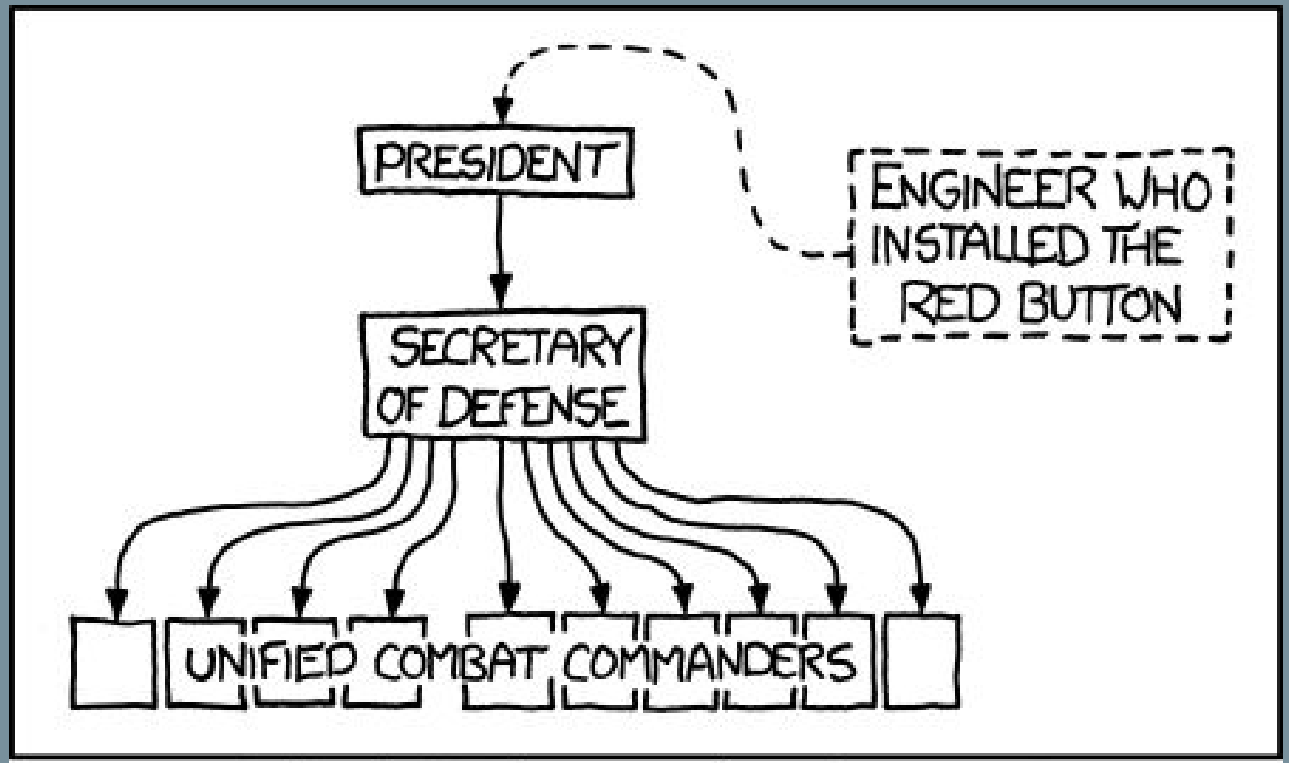
Natalie Curtiss : Grandmother? (https://flic.kr/p/7VqQPa)

Authorization

- Restrict access to specific data
- Access levels:
  - View
  - Change
  - Delete
- Rules applied based upon ID trust

Least Privilege

https://xkcd.com/898/



US NUCLEAR CHAIN OF COMMAND

Hijacking

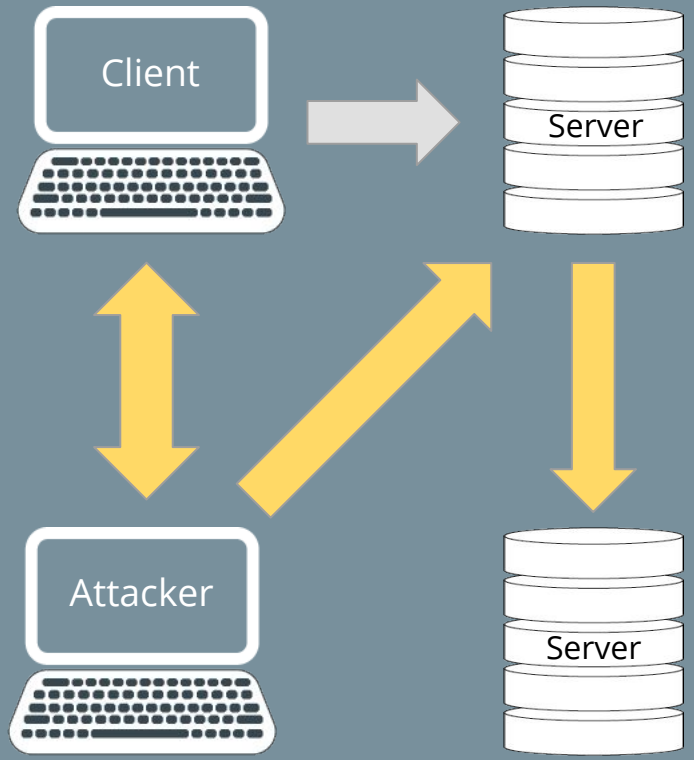
- Broken Authentication & Session Management
- Used to:
  - Gain account access
  - Impersonate users
- Protection:
  - Security frameworks such as Spring or Shiro
  - Session timeouts and fixation prevention

Broken Access Control

- Changing parameter grants unintended access
- Used To:
  - Access data
  - Perform functions
- Protection:
  - Check access / permissions with each request
  - Use indirect object references
  - Both URL and Function protection with Spring or Shiro

CSRF (Cross-Site Request Forgery)

- ➔ Impersonate user to the server
- ➔ Used to:
  - ◆ Coerce user action
  - ◆ Transfer control or resources
- ➔ Protection:
  - ◆ Unpredictable token in each request
  - ◆ Use framework built-in defenses
  - ◆ SameSite=strict flag on cookies

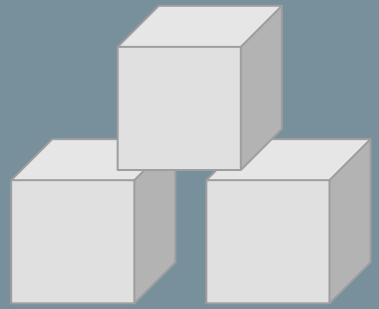



5C9MWWHYCWWN0GZ7S
N8CKGXRPQOQW6ZSY8
PJDN26AC5U4LQYSHZ
WSVIN9HLKPZF0CYXP
PW0WAGDM4SOI7MVBI
0JZS8D44KA2XJR7GS
BR6O4GF01Z5TZLM5L

Crypto

Obscurity

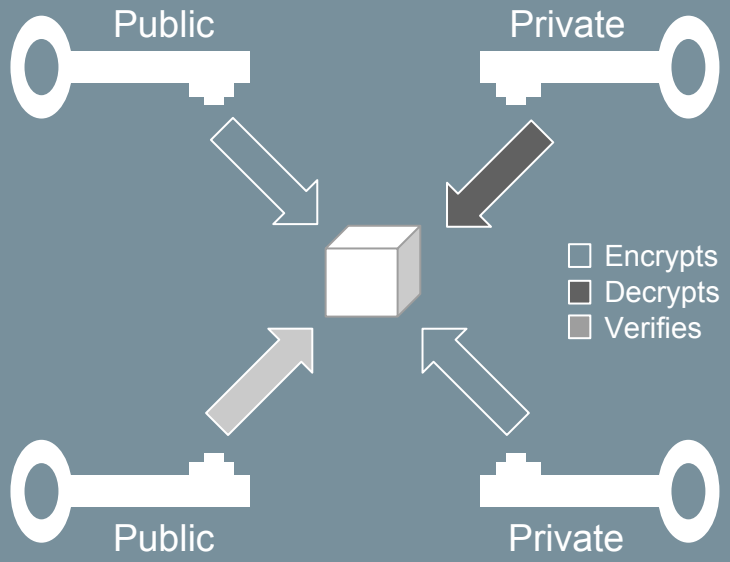
- ➔ Can’t put the cat back in the bag
- ➔ Security requires shared algorithms
- ➔ Implementation accuracy requires public review
- ➔ Unpredictable level of risk



Which box holds the prize?

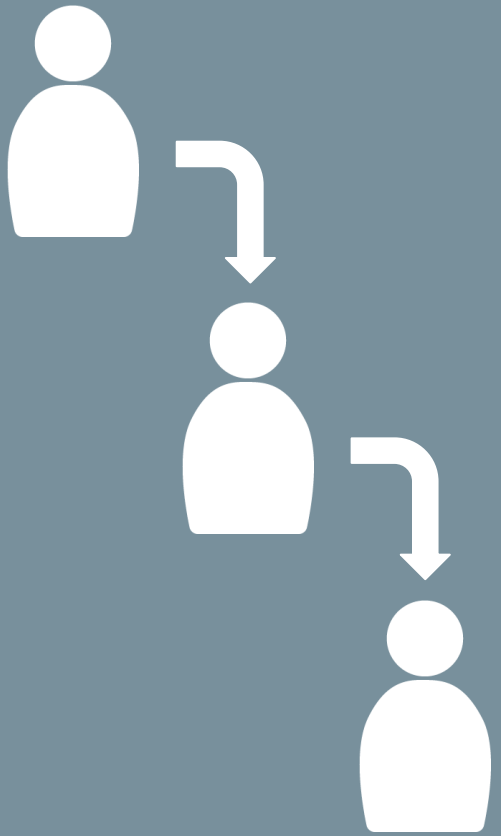
Cryptography

- ➔ Mathematically provable complexity
- ➔ Cryptographic hash
- ➔ Symmetric encryption
- ➔ Public-key encryption
- ➔ Transport Layer Security (https)



Chain of Trust

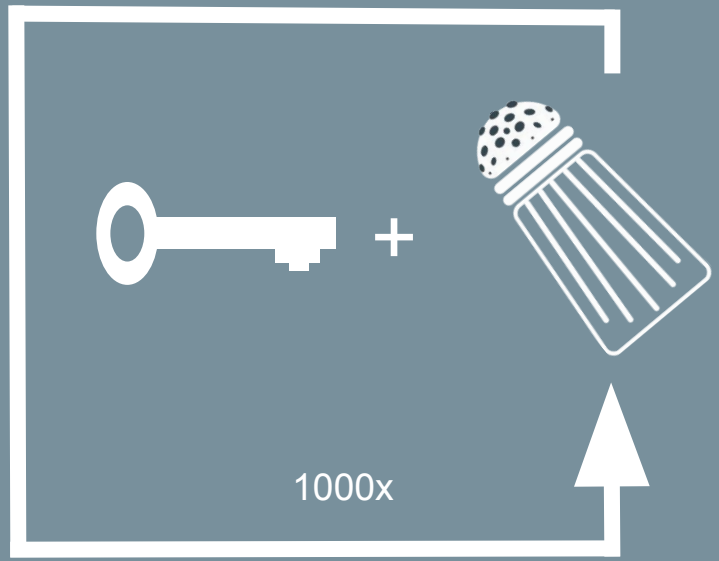
- ➔ Digital Signatures
- ➔ Certificates
- ➔ Only sign certificates you know
- ➔ Only accept certificates you trust



Passwords

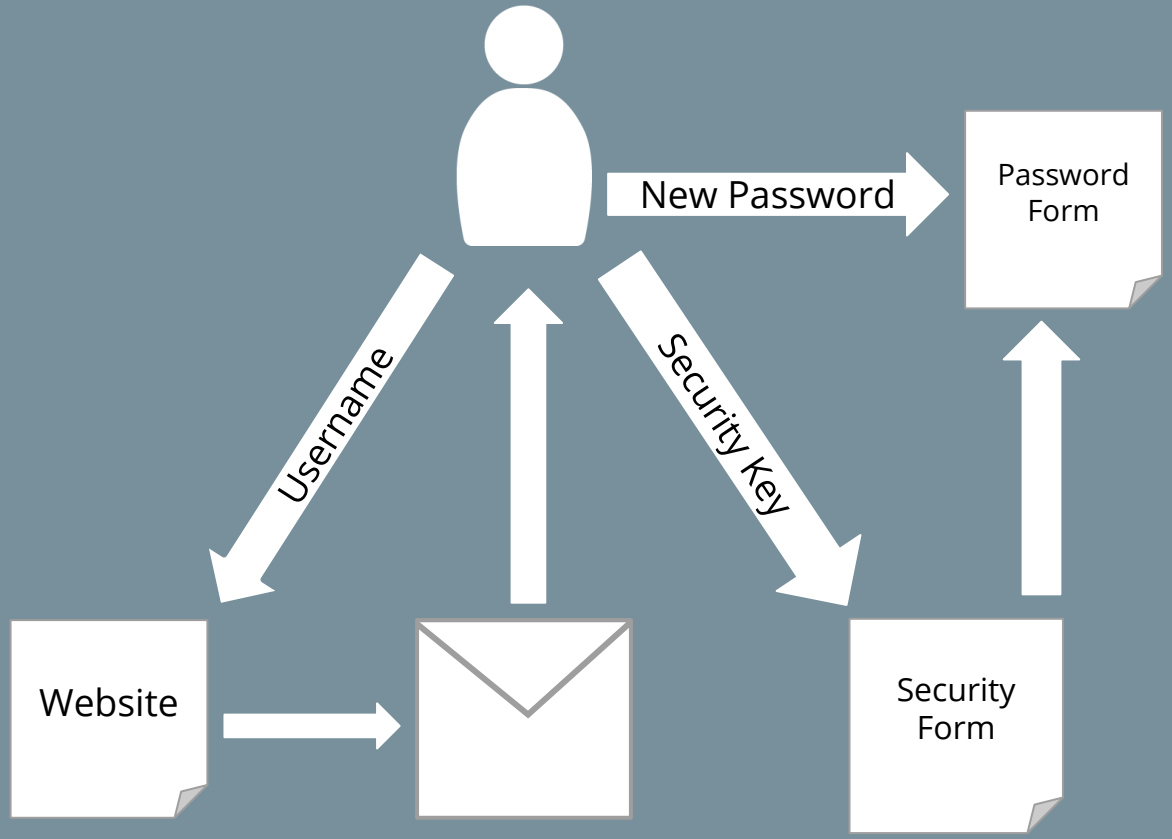
Password Protection

- Hash, don't encrypt
  - Secure algorithm (PBKDF2 with SHA512, bcrypt, scrypt)
- Salt
  - Two salts - row and app
- Iterate
  - Key derivation
- Go slow!
  - Faster hashing means faster cracking



Change Password

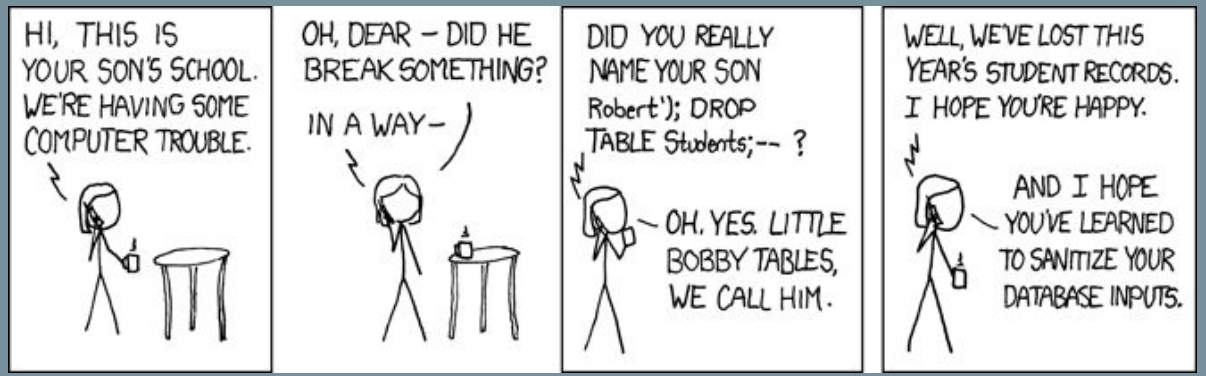
1. Click “forgot password”
2. Enter identification
3. Receive email
4. Click link
5. Enter security key(s)
6. Enter new password



Injection

(SQL) Injection

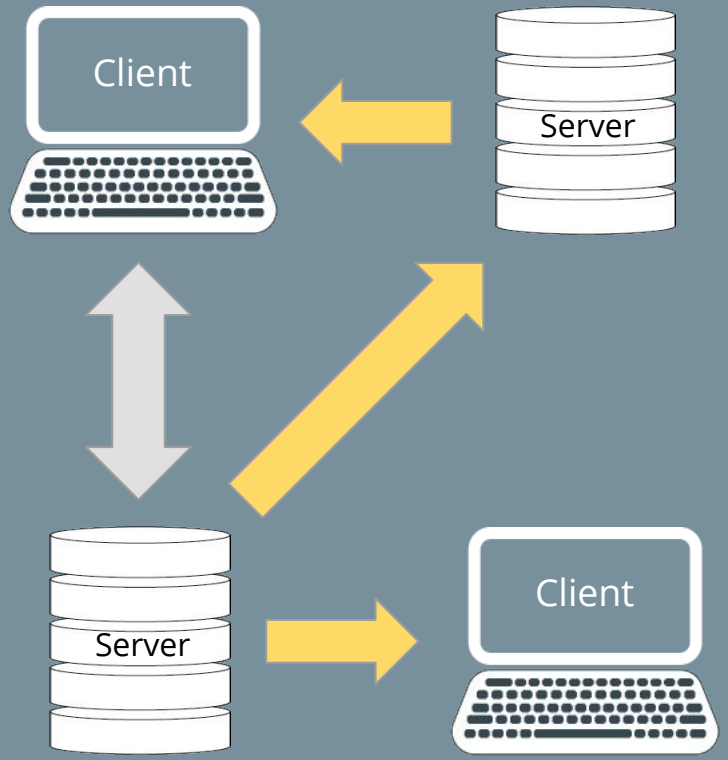
- Verbatim user-submitted content in query
- Used to:
  - Steal data
  - Corrupt data
- Protection:
  - Prepared statements
  - Escape user input
  - OWASP Java Encoder



https://xkcd.com/327/

XSS (Cross-Site Scripting)

- ➔ Verbatim display of user-submitted content
- ➔ Used to:
  - ◆ Hijack sessions or Install Trojans
  - ◆ Redirect to foreign sites
- ➔ Protection:
  - ◆ Encode all user-provided data
  - ◆ Use safe JavaScript APIs (never eval)
  - ◆ CSP Headers
  - ◆ OWASP Java HTML Sanitizer



Other Vulnerabilities

Stale Dependencies

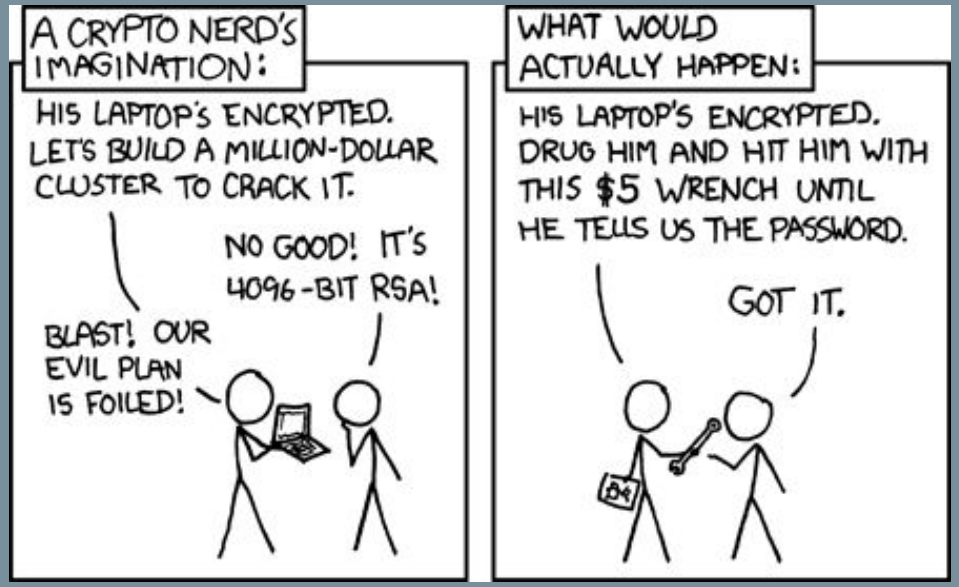
- Using components with known vulnerabilities
- Used to:
  - Compromise systems
  - Execute application code
- Protection:
  - Automated management with Ant+Ivy, Maven, or Gradle
  - OWASP DependencyCheck
  - National Vulnerability Database - https://nvd.nist.gov/

Underprotected APIs

- Insufficient protections for REST and SOAP APIs
- Used to:
  - Steal data
  - Corrupt data / deface websites
- Protection:
  - Client code doesn't contain keys
  - Strong authentication

Social Engineering

- ➔ Simply ask someone for their credentials
- ➔ Used to:
  - ◆ Obtain credentials
  - ◆ Access secure systems
- ➔ Protection:
  - ◆ Training
  - ◆ Never tell anyone your passwords
  - ◆ Always verify callers



https://xkcd.com/538/

Discussion

Minimum Developer Responsibility

- HTTPS
- Password Protection
  - Hashing for Auth
  - AES for System Logins
- OWASP Top 10 - https://owasp.org

How does online security help people?

- ➔ Restrict access to financial assets
- ➔ Protect your identity and personal information
- ➔ Defend against device takeover
- ➔ Shelter citizens from oppressive governments
- ➔ Preserve 1st, 4th, and 5th amendment rights

Security decisions

- What are we protecting?
- What is the likelihood of attack?
- What are the risks of security failure?
- What are the probable attack vectors?
- How will we detect and report breaches?
- Don’t forget the ethics!

Security Resources

- Troy Hunt
  https://www.troyhunt.com/
- Brian Krebs
  https://krebsonsecurity.com/
- Pluralsight
  https://pluralsight.com/browse/information-cyber-security
- OWASP
  https://www.owasp.org/

Michael Dowden

 @mrdowden

 linkedin.com/in/mdowden

 plus.google.com/+MichaelDowden

 lanyrd.com/profile/mrdowden/

 michael@dowden.us