COMMON WEB SECURITY THREATS

… and what to do about them

Eoin Woods
@eoinwoodz
Endava

Introductions

Eoin Woods

- CTO at Endava
- Career has spanned products and applications
  - Architecture and software engineering
  - Bull, Sybase, InterTrust
  - BGI (Barclays) and UBS
- Long time security dabbler
- Increasingly concerned at cyber threat for “normal” systems

Content

- Introducing Web Security Threats & OWASP
- The OWASP Web Vulnerabilities List
- Useful Tools to Know About
- Reviewing Defences
- Summary

Introducing Web Security Threats

Web Security Threats

- We need systems that are **dependable** in the face of
  - Malice
  - Error
  - Mischance

- People are sometimes **bad**, **stupid** or just **unlucky**

- System security aims to **mitigate** these situations

Web Security Threats

- System threats are similar to **real-world threats**:
  - Theft
  - Fraud
  - Destruction
  - Disruption

- Anything of **value** may attract unwelcome attention

"I rob banks because that's where the money is" – Willie Sutton

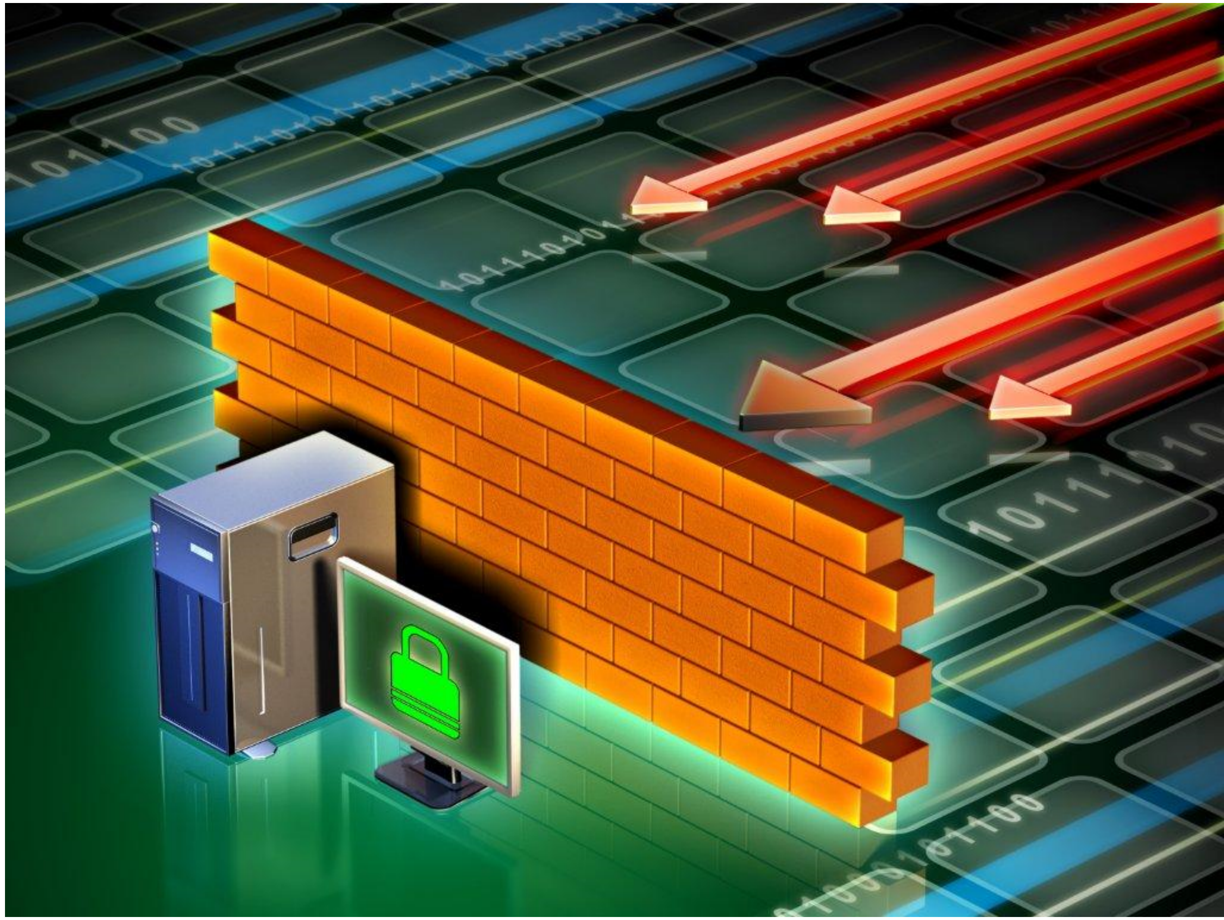
Web Security Threats

- Why do we **care** about these threats?
  - A threat is a **risk of a loss** of some sort
- Common types of **loss** are:
  - Time
  - Money
  - Privacy
  - Reputation
  - Advantage

Web Security Threats

- **Digital channels** need security
- **APIs** on the **Internet**
- **Introspection** of APIs
- Attacks being “**weaponised**”
- Today’s internal app is tomorrow’s “digital channel”

Security today mitigates tomorrow’s threat



Who are OWASP?

- The Open Web Application Security Project
  - Largely volunteer organisation, largely online
- Exists to improve the state of software security
  - Research, tools, guidance, standards
  - Runs local chapters for face to face meetings
- “OWASP Top 10” project lists top application security risks
  - Data-driven list of most significant threats to webapps
  - Referenced widely by MITRE, PCI DSS and similar
  - Updated as threats change (2003, 2004, 2007, 2010, 2013, 2017)

Other Important Security Organisations

- MITRE Corporation
  - Common Vulnerabilities and Exposures (CVE)
  - Common Weaknesses Enumeration (CWE)
- SAFECode
  - Fundamental Practices for Secure Software Development
  - Training

There are a lot of others too (CPNI, CERT, CIS, ISSA, …)

OWASP Web Vulnerabilities List

How was the 2017 List Produced?

- Project of the OWASP organisation
  - Group of ~75 volunteers create it
- Data set analysis
  - Data from 24 firms including Aspect Security, Checkmarx, MicroFocus, NCCST, Synopsis, TCS, Vantage Point, Veracode, …
  - Data represents ~114,000 applications
  - https://github.com/OWASP/Top10/2017/datacall
- Survey analysis
  - ~500 participants from the OWASP Top 10 mailing list

OWASP Top 10 - 2017

#1 Injection Attacks

#2 Broken Authentication

#3 Sensitive Data Exposure

#4 XML External Entities (XXE)

#5 Broken Access Control

#6 Security Misconfiguration

#7 Cross Site Scripting (XSS)

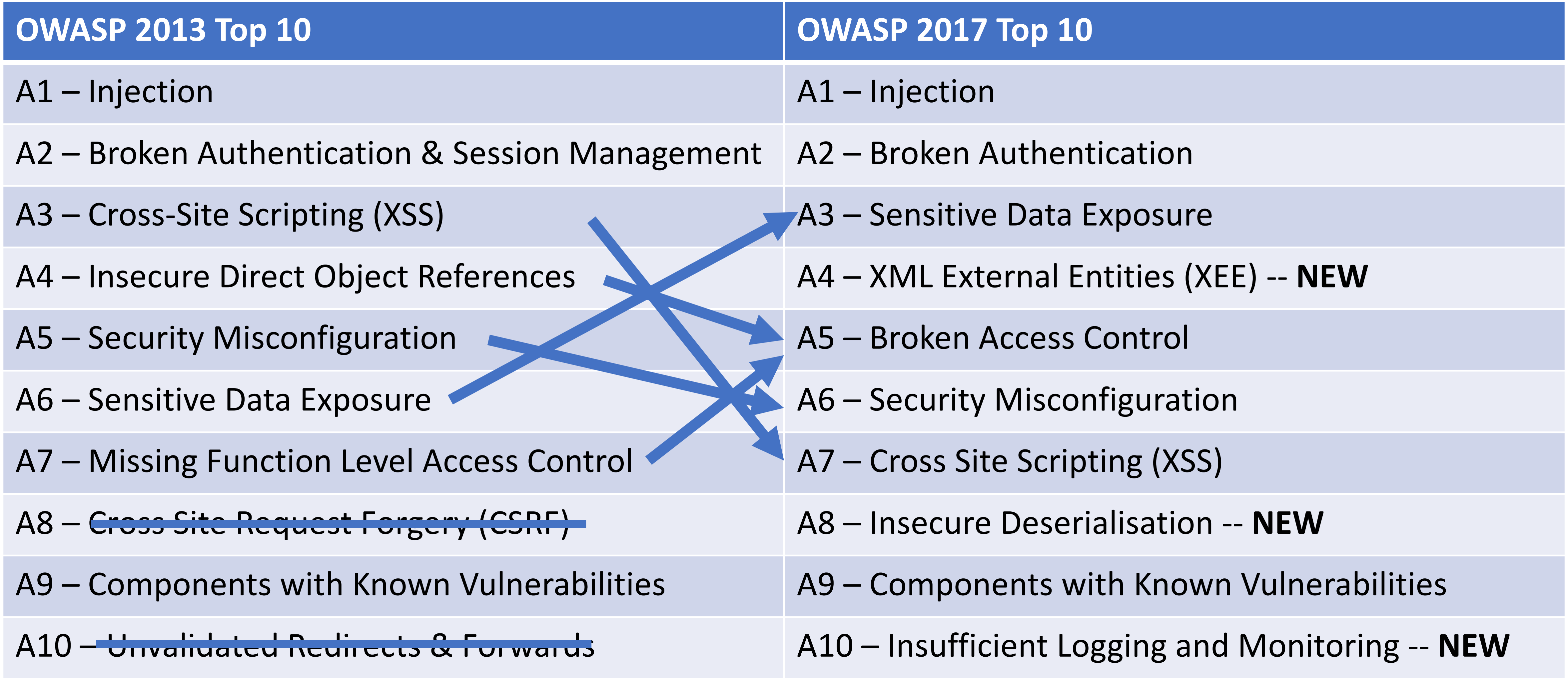
#8 Insecure Deserialisation

#9 Component Vulnerabilities

#10 Insufficient Logging and Monitoring

Some may look “obvious” but appear on the list year after year, based on real vulnerability data!

What Changed from 2013 to 2017?



OWASP 2013 Top 10	OWASP 2017 Top 10
A1 – Injection	A1 – Injection
A2 – Broken Authentication & Session Management	A2 – Broken Authentication
A3 – Cross-Site Scripting (XSS)	A3 – Sensitive Data Exposure
A4 – Insecure Direct Object References	A4 – XML External Entities (XEE) -- **NEW**
A5 – Security Misconfiguration	A5 – Broken Access Control
A6 – Sensitive Data Exposure	A6 – Security Misconfiguration
A7 – Missing Function Level Access Control	A7 – Cross Site Scripting (XSS)
A8 – ~~Cross Site Request Forgery (CSRF)~~	A8 – Insecure Deserialisation -- **NEW**
A9 – Components with Known Vulnerabilities	A9 – Components with Known Vulnerabilities
A10 – ~~Unvalidated Redirects & Forwards~~	A10 – Insufficient Logging and Monitoring -- **NEW**

#1 Injection Attacks

Exploitability	🟥 3
Prevalence	🟨 2
Detectability	🟥 3
Tech Impact	🟥 3

- Unvalidated input passed to any interpreter
  - Operating system and SQL are most common
  - Configuration injection often overlooked

```
SELECT * from table1 where name = ’%1’
```

Set ‘%1’ to **‘ OR 1=1 --** **… this results in this query:**

```
SELECT * FROM table1 WHERE name = ’’ OR 1=1 --
```

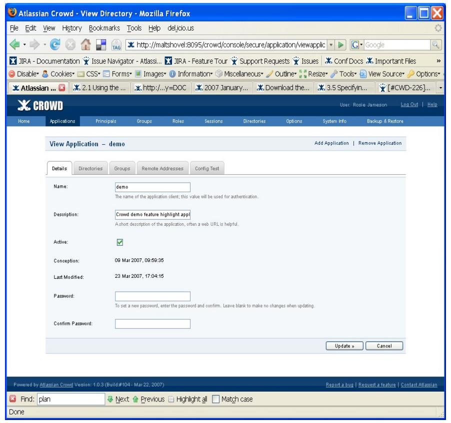
- Defences include “escaping” inputs, bind variables, using white lists, …

(See also #4 – XML External Entities …)

#2 Broken Authentication

Exploitability	**3**
Prevalence	**2**
Detectability	**2**
Tech Impact	**3**

- Credential Stuffing - millions of usernames and passwords available
- Well known credentials often present
- Unprotected session IDs
- Session IDs not rotated after login or invalidated after use
- Mitigations include strong authentication and session management controls



Atlassian Crowd - View Directory - Mozilla Firefox

http://maltshovel:8095/crowd/console/secure/application/viewapplic

User: Rosie Jameson

View Application – demo

Name: demo

The name of the application client; this value will be used for authentication.

Description: Crowd demo feature highlight appl

A short description of the application, often a web URL is helpful.

Active: ☑

Conception: 09 Mar 2007, 09:59:35

Last Modified: 23 Mar 2007, 17:04:15

Password:

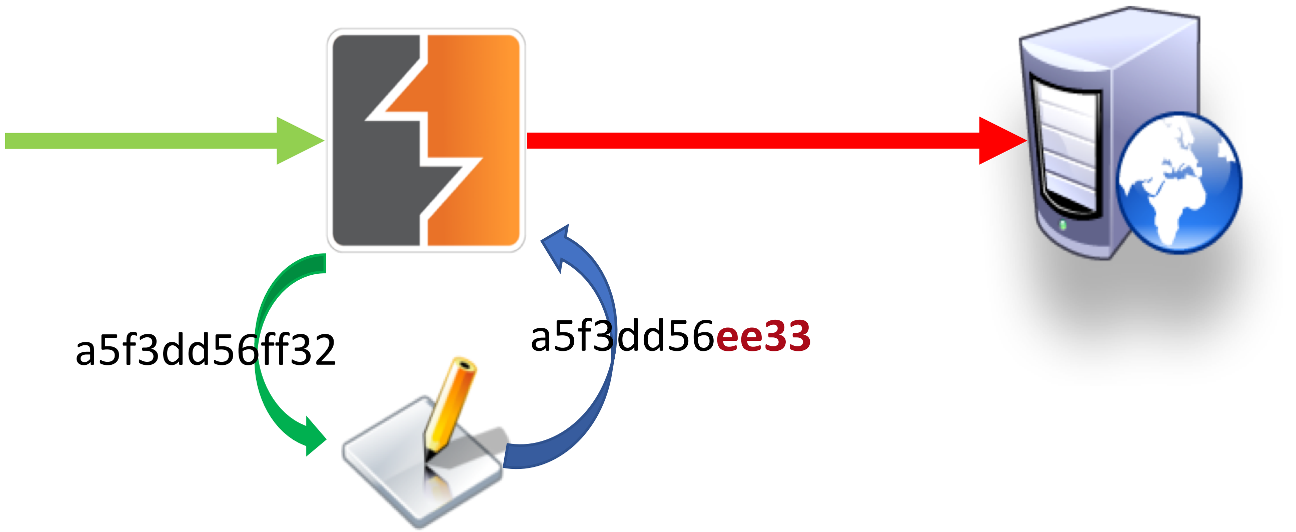
To set a new password, enter the password and confirm. Leave blank to make no changes when updating.

Confirm Password:

Powered by Atlassian Crowd Version: 1.0.3 (Build:#104 - Mar 22, 2007)

Find: plan

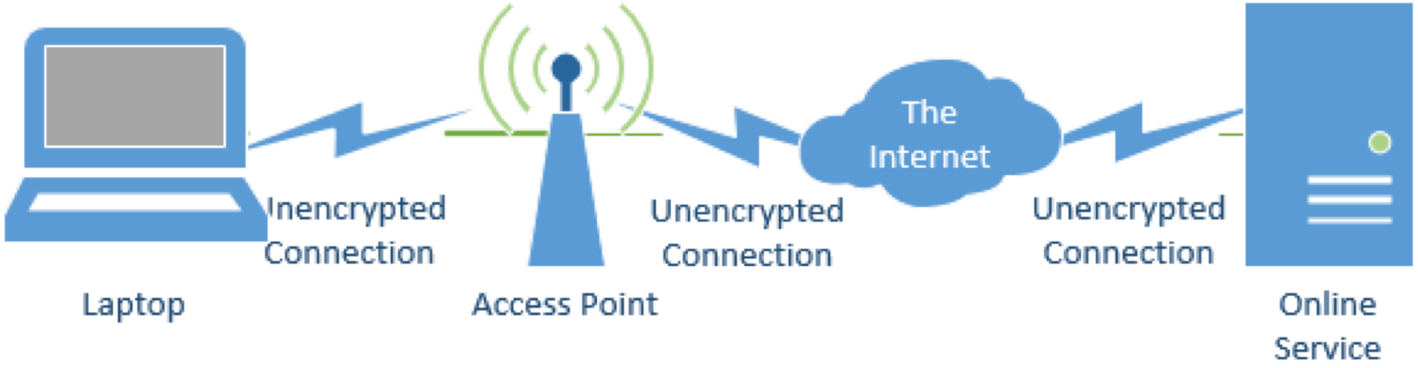
Done



#3 Sensitive Data Exposure

Exploitability	**2**
Prevalence	**3**
Detectability	**2**
Tech Impact	**3**

- Is sensitive data secured in transit?
  - TLS, message encryption
- Is sensitive data secured at rest?
  - Encryption, tokenisation, separation
- Impact can include loss of data or spoofing attacks
- Mitigation via threat analysis, encryption, limiting scope, crypto standardisation



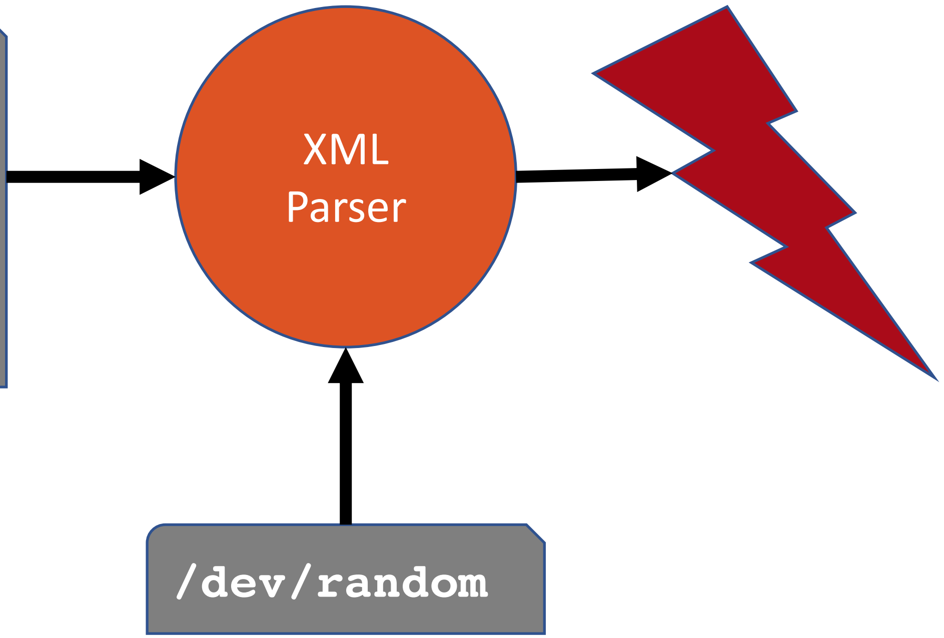
https://askleo.com

#4 XML External Entities (XXE)

Exploitability	**2**
Prevalence	**2**
Detectability	**3**
Tech Impact	**3**

- XML “external” entities cause XML parsers to retrieve external data
- Many XML parsers enable this by default (including Java’s standard library)

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



- Can expose sensitive data or provide DoS attack vector
- Mitigate by disabling external entities or removing XML

#5 Broken Access Control

Exploitability	**2**
Prevalence	**2**
Detectability	**2**
Tech Impact	**3**

- Directly referencing IDs in requests (filenames, accounts, …)
  - Not authenticating access to each on the server
  - Client can modify request and gain access to other objects
- Relying on UI or other client side code for access control
  - e.g. UI removing ”update” option & not validating action on the server

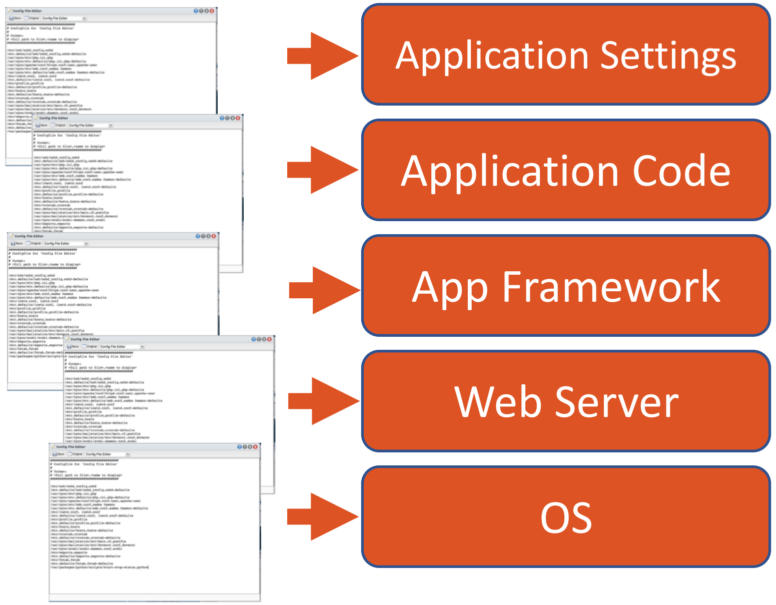
```
http://www.example.com/gettxn?txnid=4567
  → http://www.example.com/updttxn?tid=4567&value=100.00
```

- Not checking for tampering or replaying security meta data (e.g. JWT tokens)
- Mitigation through entity level access control, deny by default, strong and standardised authorisation technology and patterns, hide metadata

#6 Security Misconfiguration

Exploitability	3
Prevalence	3
Detectability	3
Tech Impact	2

- Security configuration is often complicated
  - Many different places to put it, complex & varying semantics
  - Layers from OS to application all need to be consistent
- It is easy to accidentally miss an important part
  - OS file permissions?
  - .htaccess files?
  - Shared credentials in test and production?
- Allows accidental access to or modification of resources
- Mitigation via scanning, standardisation, simplicity and automation



Application Settings

Application Code

App Framework

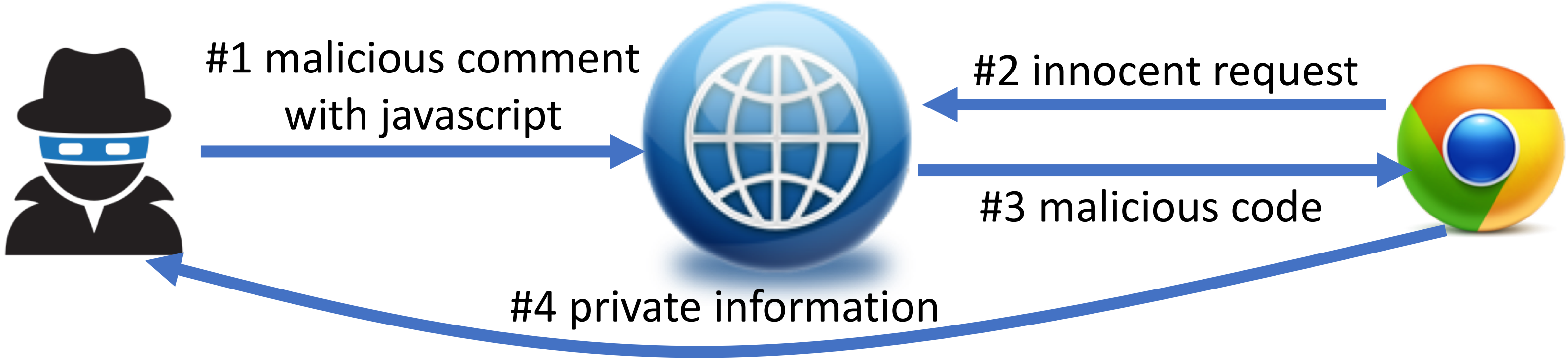
Web Server

OS

#7 Cross Site Scripting

Exploitability	3
Prevalence	3
Detectability	3
Tech Impact	2

- Occurs when script is injected into a user's web page
  - Reflected XSS attack – crafted link in email …
  - Stored XSS attack - database records, site postings, activity listings
  - DOM XSS attack - data inserted into the browser dom
- Allows redirection, session data stealing, page corruption, …

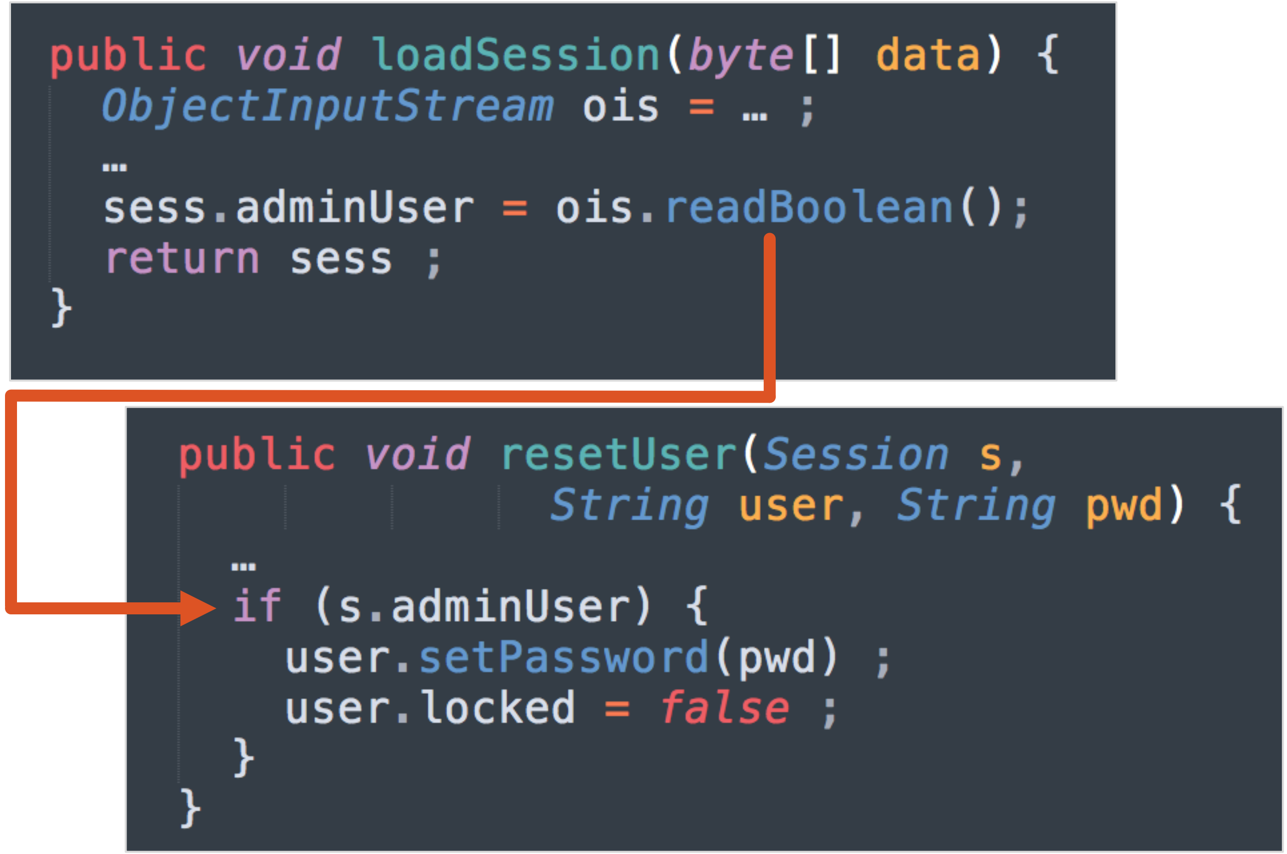


- Mitigations include validation and escaping data on the server-side

Exploitability	**1**
Prevalence	**2**
Detectability	**2**
Tech Impact	**3**

#8 Insecure Deserialisation

- Subverting de-serialisation mechanism
  - e.g. Java “gadgets” vulnerable to abuse with tampered objects
- De-serialising hostile code
  - e.g. serialised code that causes de-serialisation method to loop
- Mitigations include
  - only de-serialising from trusted sources
  - avoiding binary serialisation formats
  - signed serialisation data
  - whitelists of classes
  - platform security managers

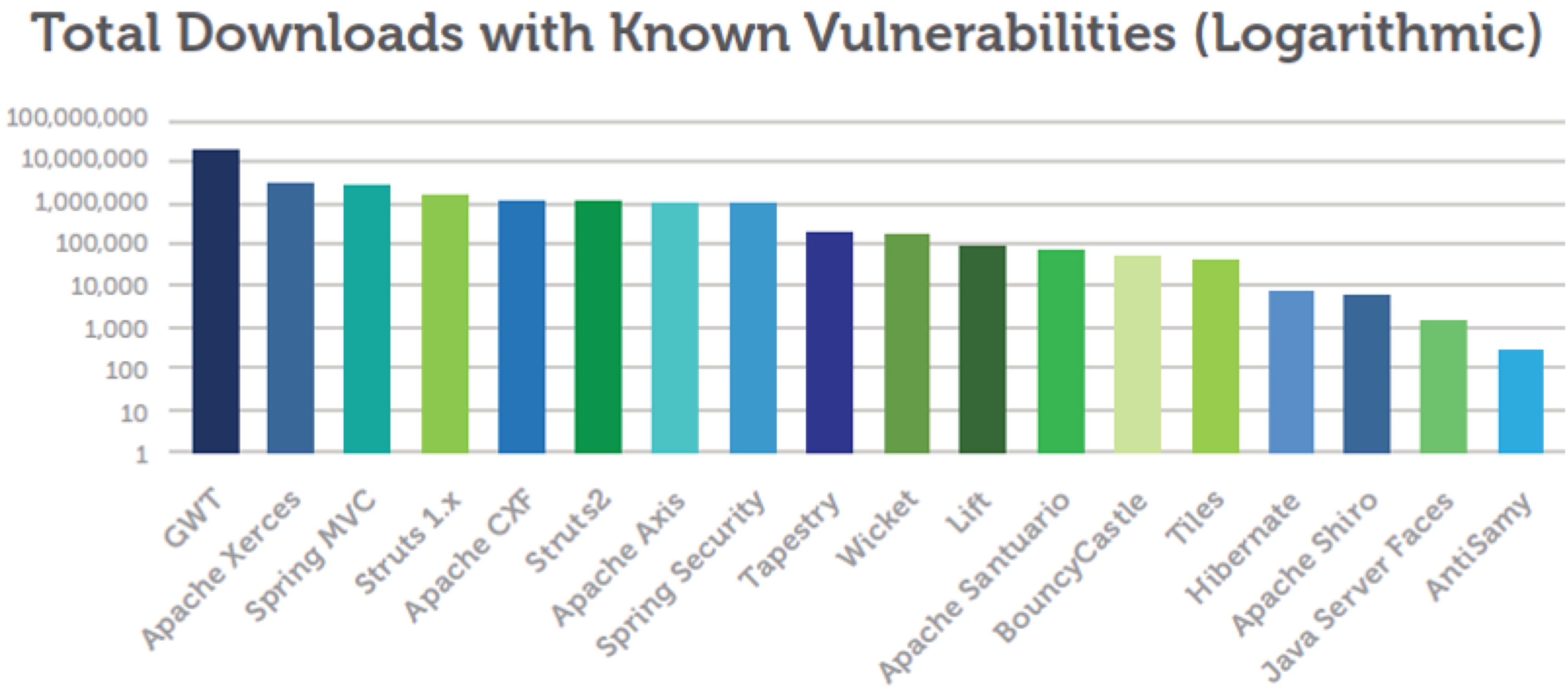


```
public void loadSession(byte[] data) {
  ObjectInputStream ois = … ;
  …
  sess.adminUser = ois.readBoolean();
  return sess ;
}
```

```
public void resetUser(Session s,
                      String user, String pwd) {
  …
  if (s.adminUser) {
    user.setPassword(pwd) ;
    user.locked = false ;
  }
}
```

#9 Known Vulnerable Components

Exploitability	**2**
Prevalence	**3**
Detectability	**2**
Tech Impact	**2**

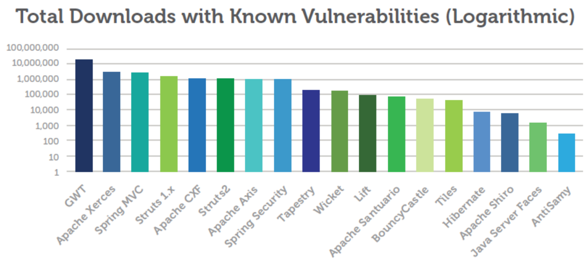


Source: "The Unfortunate Reality of Insecure Libraries", Aspect Security & Sonartype, 2012

#9 Known Vulnerable Components

Exploitability	**2**
Prevalence	**3**
Detectability	**2**
Tech Impact	**2**

- Many commonly used components have vulnerabilities
  - See weekly US-CERT list for a frightening reality check!
  - Much OSS doesn't have well researched vulnerabilities



- Few teams consider security of their 3rd party components
  - And keeping everything up to date is disruptive

- Mitigations include automated scanning of 3rd party components, actively review vulnerability lists, keep components patched

#10 Insufficient Logging and Monitoring

Exploitability	2
Prevalence	3
Detectability	1
Tech Impact	2

- Poor logging and monitoring underpins many major exploits
- Common problems:
  - Not logging key events (failed login, high value transaction, …)
  - Poor messages, no actionable statements
  - Lack of log analysis
- Centralise logging to provide better view and security of logs
  - Identify expected and unexpected log patterns (e.g OWASP coreruleset.org)
  - Know what to do when logs indicate unexpected situation
- Good test is to use OWASP ZAP, SQLMap and check for alerts
- Mitigations include standard log formats, key event logging, centralised logs, incident response plans, intrusion detection and SIEM systems

http://logio.org

OWASP
ModSecurity
Core Rule Set
THE 1ST LINE OF DEFENSE

Summary of Main Vulnerability Types

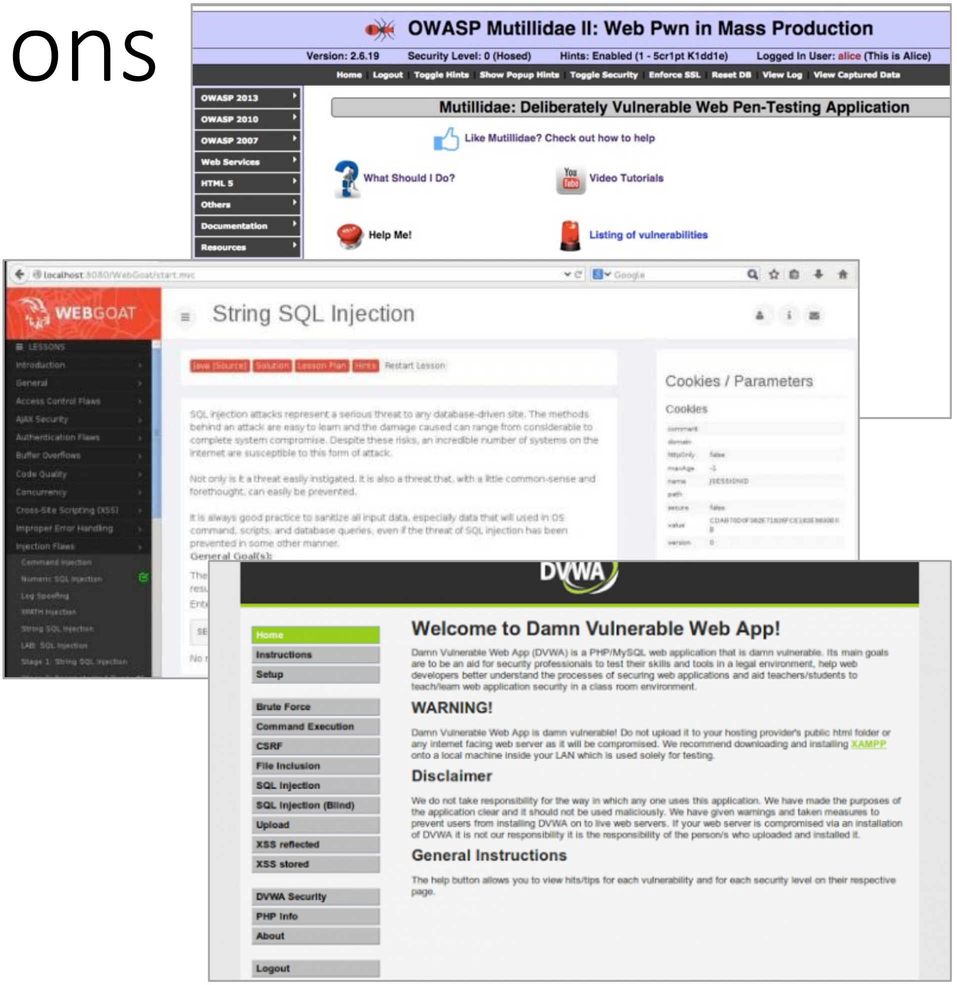
- Interpreter and page injections
  - Operating System, SQL, XML, deserialization, XSS, …
- Lack of validation
  - trusting client side restrictions
  - allowing session IDs and cookies to be reused
  - not escaping and validating input data
  - parameter values directly in pages and links
- Missing data protection
  - Sensitive data exposure, deserialisation, configuration showing metadata, …
- Complexity
  - Misconfiguration, deserialization, XXE, known vulnerabilities

OWASP
Top 10

Useful Tools

Deliberately Vulnerable Applications

- Deliberately insecure webapps
  - So run in a VM!
- OWASP Top 10 in action
  - Mutillidae & DVWA in PHP
  - WebGoat in Java



SQL injection attacks represent a serious threat to any database-driven site. The methods behind an attack are easy to learn and the damage caused can range from considerable to complete system compromise. Despite these risks, an incredible number of systems on the internet are susceptible to this form of attack.

Not only is it a threat easily instigated. It is also a threat that, with a little common-sense and forethought, can easily be prevented.

It is always good practice to sanitize all input data, especially data that will used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.

General Goal(s):

comment	
domain	
httpOnly	false
maxAge	-1
name	JSESSIONID
path	
secure	false
value	[illegible]
version	0

Welcome to Damn Vulnerable Web App!

Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications and aid teachers/students to teach/learn web application security in a class room environment.

WARNING!

Damn Vulnerable Web App is damn vulnerable! Do not upload it to your hosting provider's public html folder or any internet facing web server as it will be compromised. We recommend downloading and installing XAMPP onto a local machine inside your LAN which is used solely for testing.

Disclaimer

We do not take responsibility for the way in which any one uses this application. We have made the purposes of the application clear and it should not be used maliciously. We have given warnings and taken measures to prevent users from installing DVWA on to live web servers. If your web server is compromised via an installation of DVWA it is not our responsibility it is the responsibility of the person/s who uploaded and installed it.

General Instructions

The help button allows you to view hits/tips for each vulnerability and for each security level on their respective page.

http://sourceforge.net/projects/mutillidae/

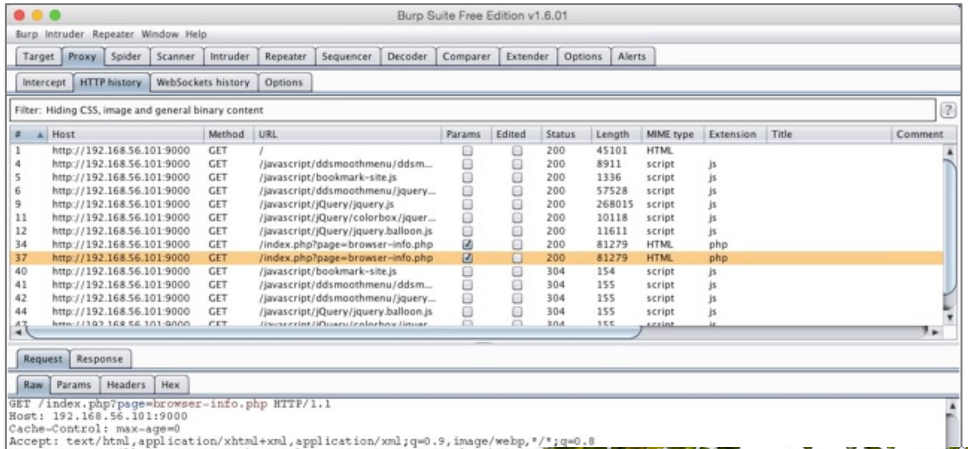
http://www.dvwa.co.uk/

https://github.com/WebGoat/WebGoat/wiki

https://github.com/eystsen/pentestlab

BurpSuite

- Proxy, scanning, pentest tool
- Very capable free version
- Fuller commercial version available
- Inspect traffic, manipulate headers and content, replay, spider, …
- Made in Knutsford!



Burp Suite Free Edition v1.6.01
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Intercept HTTP history WebSockets history Options
Request Response
Raw Params Headers Hex
? < + >
Type a search term

Filter: Hiding CSS, image and general binary content

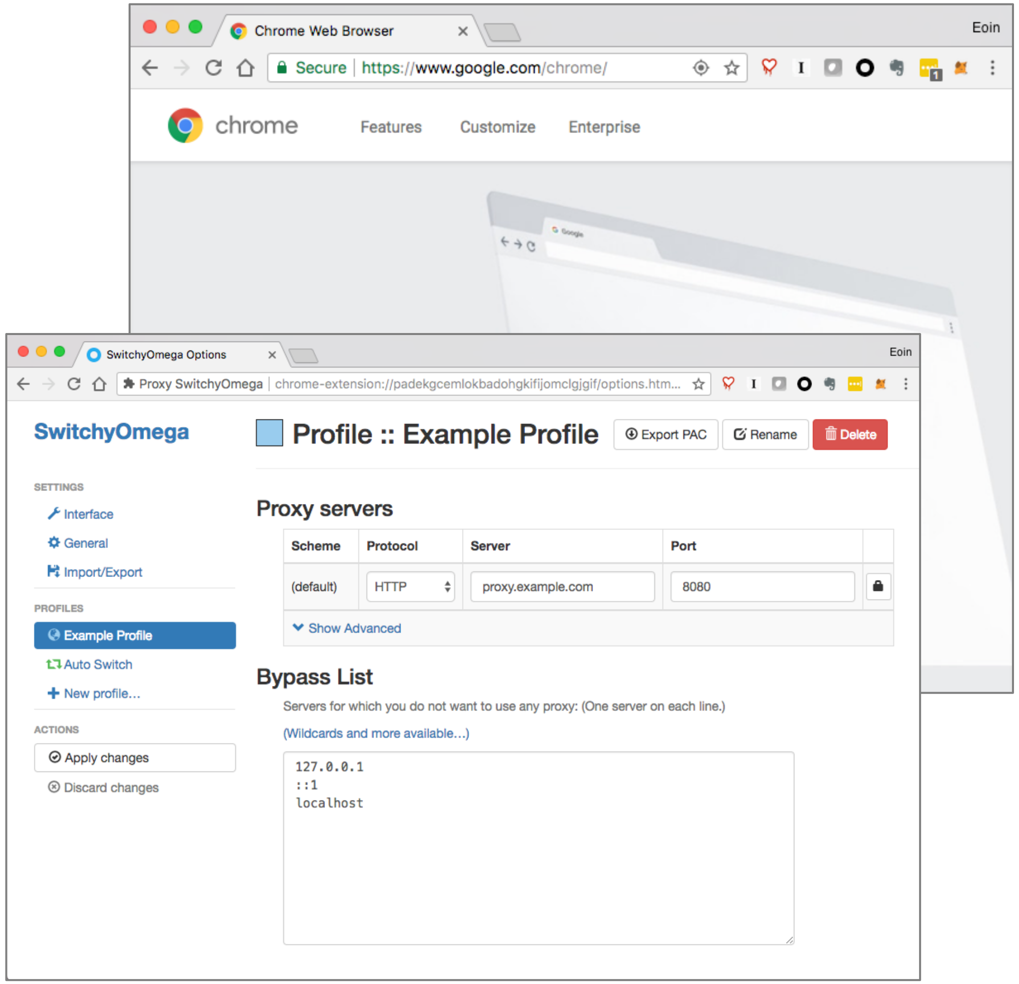
# ▲	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title	Comment
1	http://192.168.56.101:9000	GET	/	☐	☐	200	45101	HTML			
4	http://192.168.56.101:9000	GET	/javascript/ddsmoothmenu/ddsm...	☐	☐	200	8911	script	js		
5	http://192.168.56.101:9000	GET	/javascript/bookmark-site.js	☐	☐	200	1336	script	js		
6	http://192.168.56.101:9000	GET	/javascript/ddsmoothmenu/jquery...	☐	☐	200	57528	script	js		
9	http://192.168.56.101:9000	GET	/javascript/jQuery/jquery.js	☐	☐	200	268015	script	js		
11	http://192.168.56.101:9000	GET	/javascript/jQuery/colorbox/jquer...	☐	☐	200	10118	script	js		
12	http://192.168.56.101:9000	GET	/javascript/jQuery/jquery.balloon.js	☐	☐	200	11611	script	js		
34	http://192.168.56.101:9000	GET	/index.php?page=browser-info.php	☑	☐	200	81279	HTML	php		
37	http://192.168.56.101:9000	GET	/index.php?page=browser-info.php	☑	☐	200	81279	HTML	php		
40	http://192.168.56.101:9000	GET	/javascript/bookmark-site.js	☐	☐	304	154	script	js		
41	http://192.168.56.101:9000	GET	/javascript/ddsmoothmenu/ddsm...	☐	☐	304	155	script	js		
42	http://192.168.56.101:9000	GET	/javascript/ddsmoothmenu/jquery...	☐	☐	304	155	script	js		
44	http://192.168.56.101:9000	GET	/javascript/jQuery/jquery.balloon.js	☐	☐	304	155	script	js		
47	http://192.168.56.101:9000	GET	/javascript/jQuery/colorbox/jquer...	☐	☐	304	155	script	js		

```
GET /index.php?page=browser-info.php HTTP/1.1
Host: 192.168.56.101:9000
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10_3) AppleWebKit/53
Safari/537.36
DNT: 1
Referer: http://192.168.56.101:9000/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-GB,en;q=0.8,en-US;q=0.6,fr;q=0.4
Cookie: username=alice; uid=24; PHPSESSID=ncsvt213frj0l1k7mpm6g2d5s7; show
```

`http://portswigger.net/burp`

Browser and Proxy Switcher

- Chrome and Switchy Omega or other similar pairing
- Allows easy switching of proxy server to BurpSuite



Profile :: Example Profile

Proxy servers

Scheme	Protocol	Server	Port
(default)	HTTP	proxy.example.com	8080

Bypass List

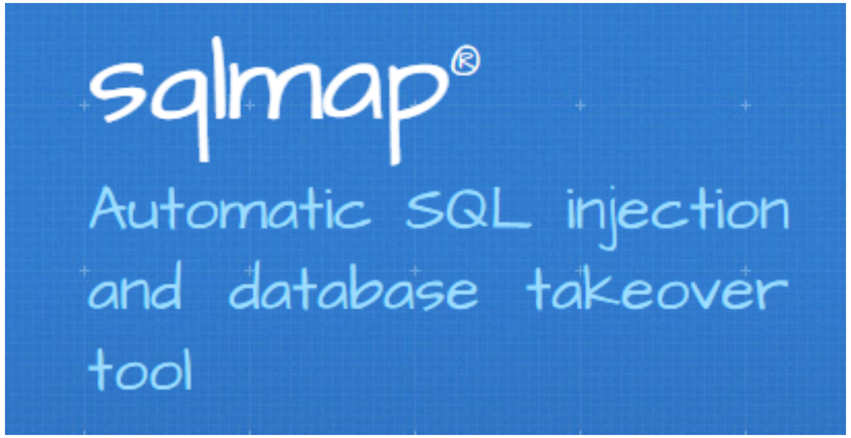
Servers for which you do not want to use any proxy: (One server on each line.)

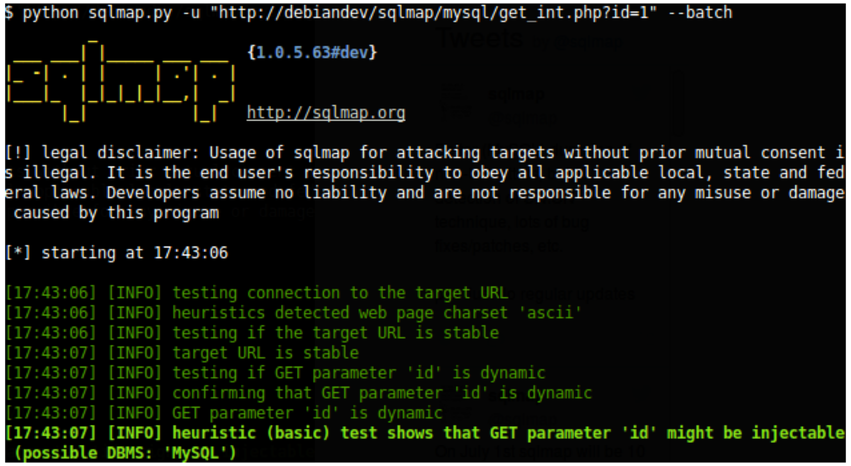
(Wildcards and more available...)

```
127.0.0.1
::1
localhost
```

sqlmap

- Automated SQL injection and database pentesting
- Open source Python command line tool
- Frighteningly effective!



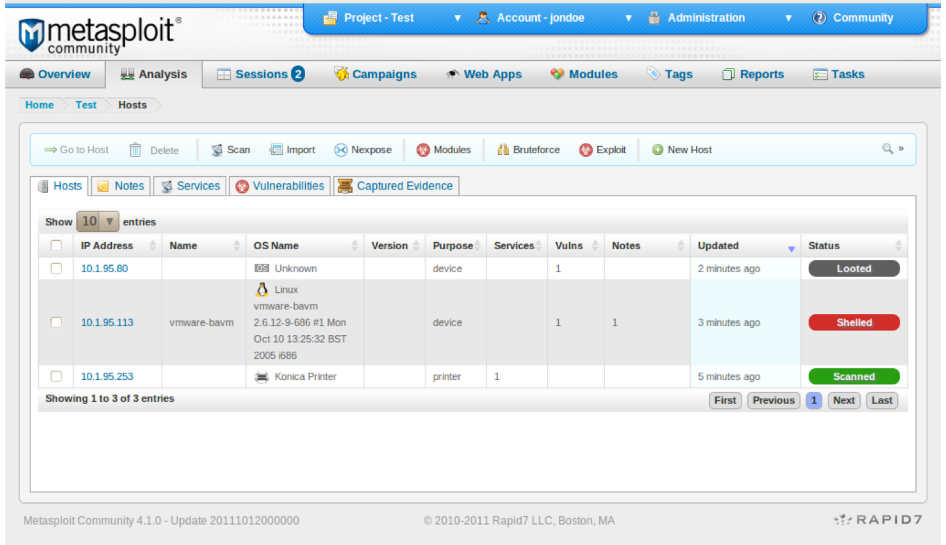


```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch
         _
 ___ ___| |_____ ___ ___  {1.0.5.63#dev}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent i
s illegal. It is the end user's responsibility to obey all applicable local, state and fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
 caused by this program

[*] starting at 17:43:06

[17:43:06] [INFO] testing connection to the target URL
[17:43:06] [INFO] heuristics detected web page charset 'ascii'
[17:43:06] [INFO] testing if the target URL is stable
[17:43:07] [INFO] target URL is stable
[17:43:07] [INFO] testing if GET parameter 'id' is dynamic
[17:43:07] [INFO] confirming that GET parameter 'id' is dynamic
[17:43:07] [INFO] GET parameter 'id' is dynamic
[17:43:07] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
 (possible DBMS: 'MySQL')
```

http://sqlmap.org

Metasploit

- The pentester’s ”standard” tool
- Very wide range of capabilities
- Commercial version available



□	IP Address	Name	OS Name	Version	Purpose	Services	Vulns	Notes	Updated	Status
□	10.1.95.80		Unknown		device		1		2 minutes ago	Looted
□	10.1.95.113	vmware-bavm	Linux vmware-bavm 2.6.12-9-686 #1 Mon Oct 10 13:25:32 BST 2005 i686		device		1	1	3 minutes ago	Shelled
□	10.1.95.253		Konica Printer		printer	1			5 minutes ago	Scanned

Showing 1 to 3 of 3 entries

Metasploit Community 4.1.0 - Update 20111012000000

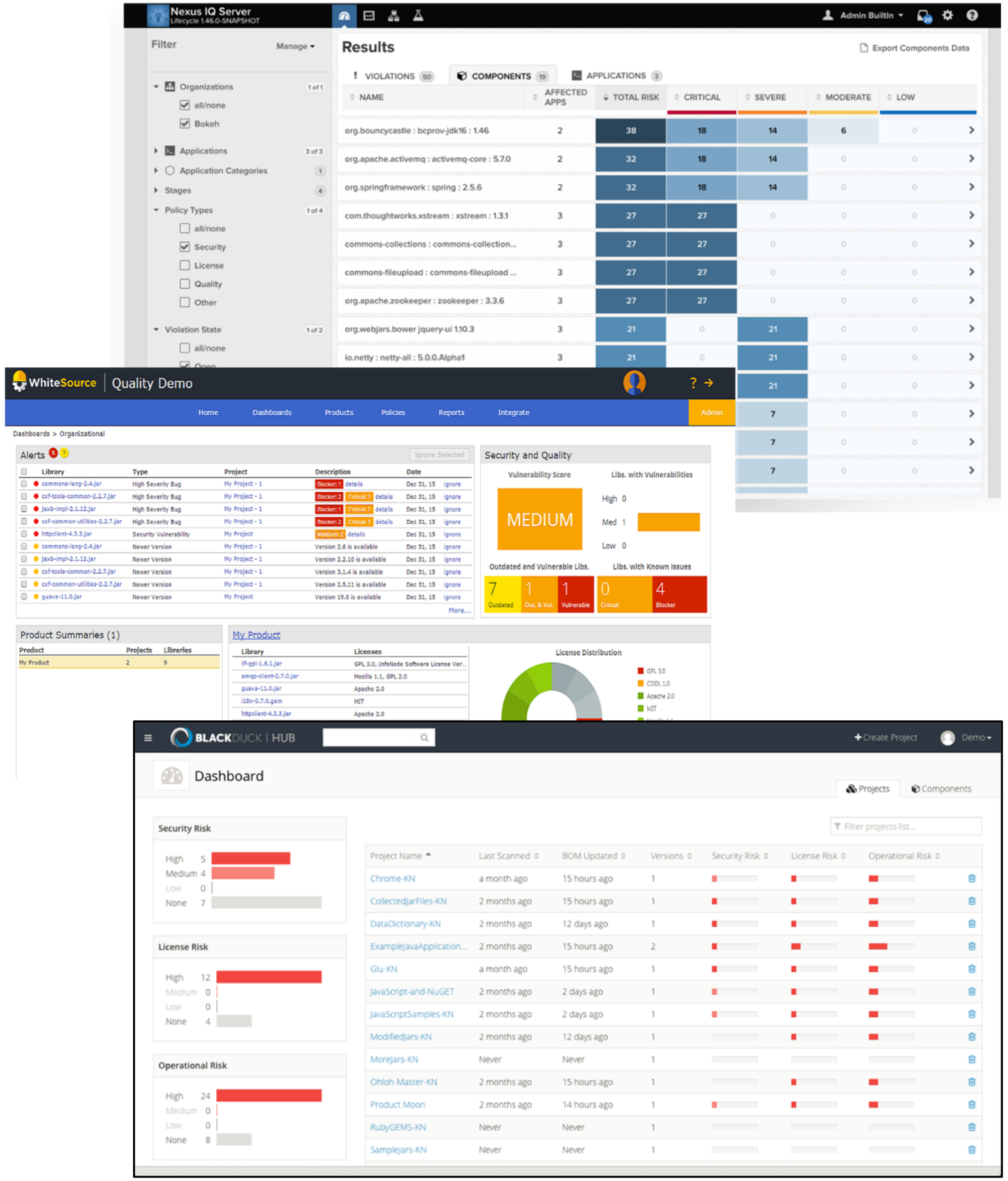
© 2010-2011 Rapid7 LLC, Boston, MA

https://www.metasploit.com

Open Source Scanning

- Example commercial tools for open source security, audit & compliance:
  - BlackDuck
  - Whitesource
  - Sonatype LCM
- Scan builds identifying open source
- Checks for known vulnerabilities
- Alerts and dashboards for monitoring

www.blackduck.com
www.whitesourcesoftware.com
www.sonatype.com/nexus-lifecycle



NAME	AFFECTED APPS	TOTAL RISK	CRITICAL	SEVERE	MODERATE	LOW
org.bouncycastle : bcprov-jdk16 : 1.46	2	38	18	14	6	0
org.apache.activemq : activemq-core : 5.7.0	2	32	18	14	0	0
org.springframework : spring : 2.5.6	2	32	18	14	0	0
com.thoughtworks.xstream : xstream : 1.3.1	3	27	27	0	0	0
commons-collections : commons-collection...	3	27	27	0	0	0
commons-fileupload : commons-fileupload ...	3	27	27	0	0	0
org.apache.zookeeper : zookeeper : 3.3.6	3	27	27	0	0	0
org.webjars.bower jquery-ui 1.10.3	3	21	0	21	0	0
io.netty : netty-all : 5.0.0.Alpha1	3	21	0	21	0	0

Library	Type	Project	Description	Date	
commons-lang-2.4.jar	High Severity Bug	My Project - 1	Blocker: 1 details	Dec 31, 15	ignore
cxf-tools-common-2.2.7.jar	High Severity Bug	My Project - 1	Blocker: 2 Critical: 3 details	Dec 31, 15	ignore
jaxb-impl-2.1.12.jar	High Severity Bug	My Project - 1	Blocker: 1 Critical: 2 details	Dec 31, 15	ignore
cxf-common-utilities-2.2.7.jar	High Severity Bug	My Project - 1	Blocker: 2 Critical: 3 details	Dec 31, 15	ignore
httpclient-4.3.3.jar	Security Vulnerability	My Project	Medium: 2 details	Dec 31, 15	ignore
commons-lang-2.4.jar	Newer Version	My Project - 1	Version 2.6 is available	Dec 31, 15	ignore
jaxb-impl-2.1.12.jar	Newer Version	My Project - 1	Version 2.2.10 is available	Dec 31, 15	ignore
cxf-tools-common-2.2.7.jar	Newer Version	My Project - 1	Version 3.1.4 is available	Dec 31, 15	ignore
cxf-common-utilities-2.2.7.jar	Newer Version	My Project - 1	Version 2.5.11 is available	Dec 31, 15	ignore
guava-11.0.jar	Newer Version	My Project	Version 19.0 is available	Dec 31, 15	ignore

More...

Security and Quality

Vulnerability Score: MEDIUM

Libs. with Vulnerabilities: High 0, Med 1, Low 0

Outdated and Vulnerable Libs.: 7 Outdated, 1 Out. & Vul., 1 Vulnerable

Libs. with Known Issues: 0 Critical, 4 Blocker

Product Summaries (1)

Product	Projects	Libraries
My Product	2	9

My Product

Library	Licenses
ilf-gpl-1.6.1.jar	GPL 3.0, InfoNode Software License Ver...
amqp-client-2.7.0.jar	Mozilla 1.1, GPL 2.0
guava-11.0.jar	Apache 2.0
i18n-0.7.0.gem	MIT
httpclient-4.3.3.jar	Apache 2.0

Security Risk: High 5, Medium 4, Low 0, None 7

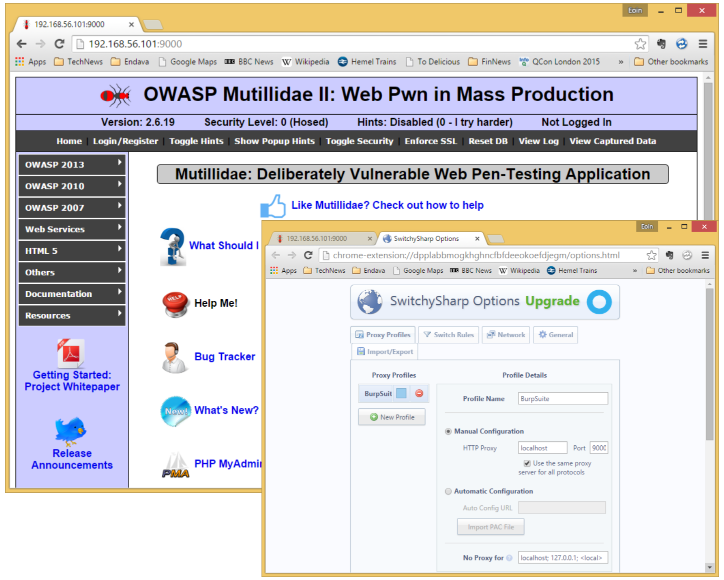
License Risk: High 12, Medium 0, Low 0, None 4

Operational Risk: High 24, Medium 0, Low 0, None 8

Project Name	Last Scanned	BOM Updated	Versions
Chrome-KN	a month ago	15 hours ago	1
CollectedJarFiles-KN	2 months ago	15 hours ago	1
DataDictionary-KN	2 months ago	12 days ago	1
ExampleJavaApplication...	2 months ago	15 hours ago	2
Glu-KN	a month ago	15 hours ago	1
JavaScript-and-NuGET	2 months ago	2 days ago	1
JavaScriptSamples-KN	2 months ago	2 days ago	1
ModifiedJars-KN	2 months ago	12 days ago	1
MoreJars-KN	Never	Never	1
Ohloh-Master-KN	2 months ago	15 hours ago	1
Product Moon	2 months ago	14 hours ago	1
RubyGEMS-KN	Never	Never	1
SampleJars-KN	Never	Never	1

Demonstrations

Mutillidae



192.168.56.101:9000

OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Mutillidae: Deliberately Vulnerable Web Pen-Testing Application

Like Mutillidae? Check out how to help

chrome-extension://dpplabbmogkhghncfbfdeeokoefdjegm/options.html

SwitchySharp Options

Proxy Profiles: BurpSuit

Profile Details

Profile Name: BurpSuite

● Manual Configuration

HTTP Proxy: localhost Port: 9000

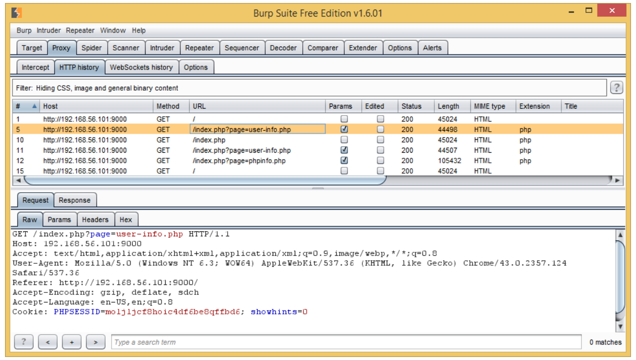
☑ Use the same proxy server for all protocols

○ Automatic Configuration

Auto Config URL:

No Proxy for: localhost; 127.0.0.1; <local>

Browser with proxy plugin



Burp Suite Free Edition v1.6.01

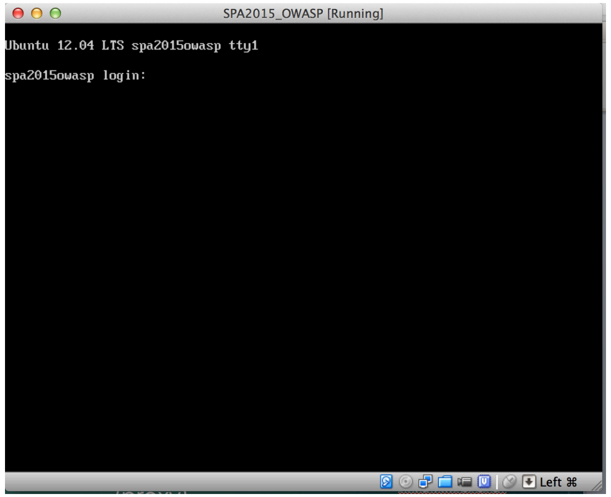
Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length	MIME type	Extension	Title
1	http://192.168.56.101:9000	GET	/	□	□	200	45024	HTML		
5	http://192.168.56.101:9000	GET	/index.php?page=user-info.php	☑	□	200	44498	HTML	php	
10	http://192.168.56.101:9000	GET	/index.php	□	□	200	45024	HTML	php	
11	http://192.168.56.101:9000	GET	/index.php?page=user-info.php	☑	□	200	44507	HTML	php	
12	http://192.168.56.101:9000	GET	/index.php?page=phpinfo.php	☑	□	200	105432	HTML	php	
15	http://192.168.56.101:9000	GET	/	□	□	200	45024	HTML		

```
GET /index.php?page=user-info.php HTTP/1.1
Host: 192.168.56.101:9000
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/43.0.2357.124
Safari/537.36
Referer: http://192.168.56.101:9000/
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Cookie: PHPSESSID=moljljcf8hoic4df6be8qffbd6; showhints=0
```

0 matches

BurpSuite (proxy)



SPA2015_OWASP [Running]

```
Ubuntu 12.04 LTS spa2015owasp tty1

spa2015owasp login:
```

Mutillidae

An Example Multi-Step Attack - Impersonation

Attacks rarely use just one vulnerability

1. SQL Injection

User list obtained

2. Plant XSS in blog

Persistent XSS achieved

3. Send link to Admin

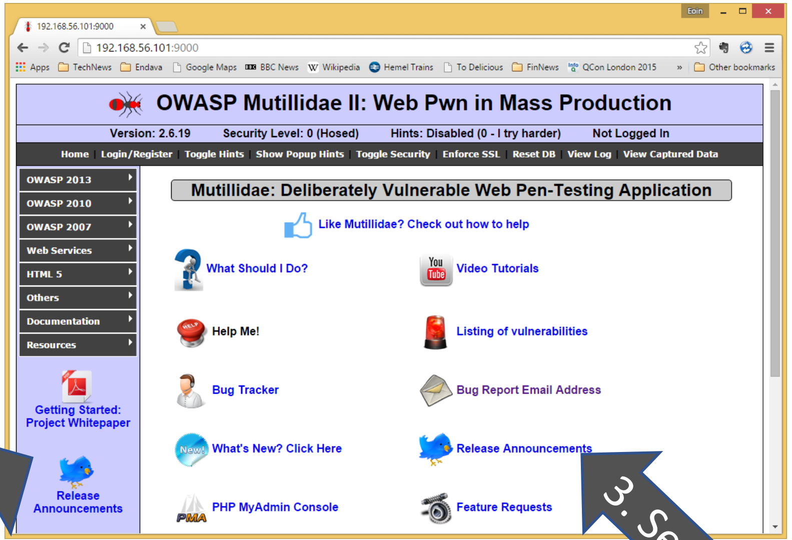
XSS Script executed

4. Steal browser state

Sessions etc. saved

5. Impersonation

Goal Achieved!



192.168.56.101:9000

Apps | TechNews | Endava | Google Maps | BBC News | Wikipedia | Hemel Trains | To Delicious | FinNews | QCon London 2015 | Other bookmarks

OWASP Mutillidae II: Web Pwn in Mass Production

Version: 2.6.19 Security Level: 0 (Hosed) Hints: Disabled (0 - I try harder) Not Logged In

Home | Login/Register | Toggle Hints | Show Popup Hints | Toggle Security | Enforce SSL | Reset DB | View Log | View Captured Data

OWASP 2013
OWASP 2010
OWASP 2007
Web Services
HTML 5
Others
Documentation
Resources

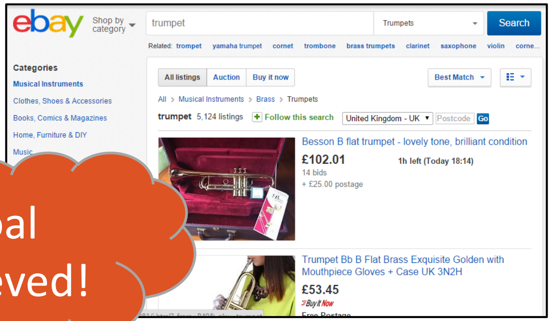
Getting Started: Project Whitepaper

Release Announcements

Mutillidae: Deliberately Vulnerable Web Pen-Testing Application

Like Mutillidae? Check out how to help

What Should I Do? | Video Tutorials
Help Me! | Listing of vulnerabilities
Bug Tracker | Bug Report Email Address
What's New? Click Here | Release Announcements
PHP MyAdmin Console | Feature Requests



Shop by category | trumpet | Trumpets | Search

Related: trompet yamaha trumpet cornet trombone brass trumpets clarinet saxophone violin corne...

Categories
Musical Instruments
Clothes, Shoes & Accessories
Books, Comics & Magazines
Home, Furniture & DIY
Music

All listings | Auction | Buy it now | Best Match

All > Musical Instruments > Brass > Trumpets

trumpet 5,124 listings + Follow this search United Kingdom - UK Postcode Go

Besson B flat trumpet - lovely tone, brilliant condition
£102.01 1h left (Today 18:14)
14 bids
+ £25.00 postage

Trumpet Bb B Flat Brass Exquisite Golden with Mouthpiece Gloves + Case UK 3N2H
£53.45
Buy It Now

Defences

Key Web Vulnerability Defences

- Don’t trust clients (browsers)
  - Validation, authorisation, …
- Identify “interpreters”, escape inputs, use bind variables, …
  - Command lines, web pages, database queries, …
- Protect valuable information at rest and in transit
  - Use encryption judiciously
- Simplicity
  - Verify configuration and correctness
- Standardise and Automate
  - Force consistency, avoid configuration errors

Don’t Trust Clients

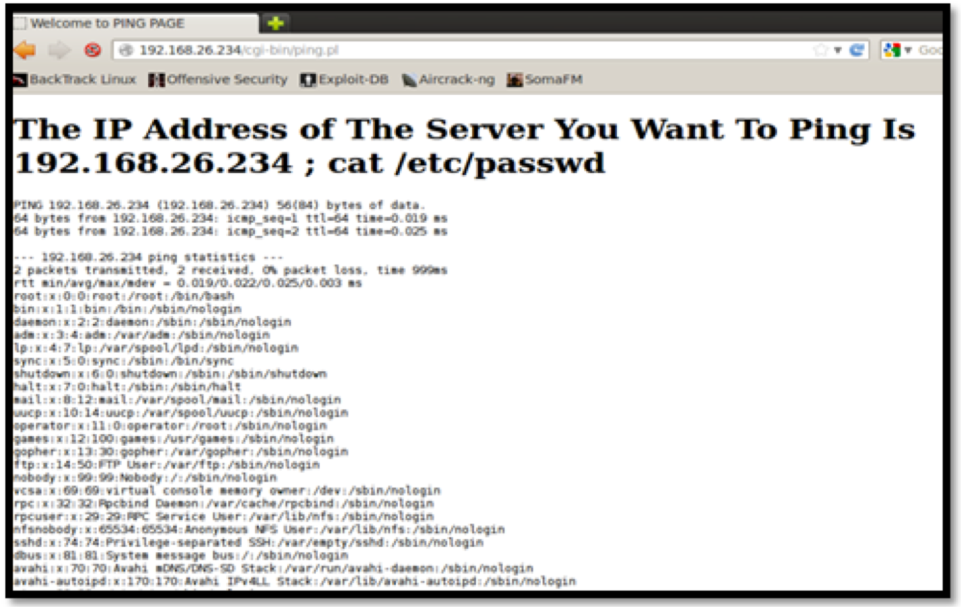
- Be wary when trusting anything from a browser
  - You don’t control it
  - Sophisticated code execution (& injection) platform
  - Output can be manipulated

- Assume or prevent tampering
  - TLS connections to avoid 3rd party interception
  - Short lived sessions
  - Reauthenticate regularly & before sensitive operations
  - Consider multi-factor authentication
  - Use opaque tokens not real object references for params
  - Validate everything

Watch out for injection

- Many pieces of software act as interpreters
  - Browser for HTML and JavaScript
  - Operating system shells – system(“mv $1 $2”)
  - Databases – query languages
  - Configuration files
  - XML parsers

- Assume that someone will work it out!
  - Avoid creating commands using string manipulation
    - Use libraries and bind variables
  - Escape all strings being passed to an “interpreter”
    - Use a third party “escaping” library (e.g. OWASP)
  - Reject excessively long strings (e.g. username > 30 char)



The IP Address of The Server You Want To Ping Is 192.168.26.234 ; cat /etc/passwd

```
PING 192.168.26.234 (192.168.26.234) 56(84) bytes of data.
64 bytes from 192.168.26.234: icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from 192.168.26.234: icmp_seq=2 ttl=64 time=0.025 ms

--- 192.168.26.234 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.019/0.022/0.025/0.003 ms
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/cache/rpcbind:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
avahi-autoipd:x:170:170:Avahi IPv4LL Stack:/var/lib/avahi-autoipd:/sbin/nologin
```

Protect Valuable Information

- Defence in depth – assume perimeter breach
  - Encrypt messaging as standard
  - Consider database encryption
  - Consider file or filesystem encryption

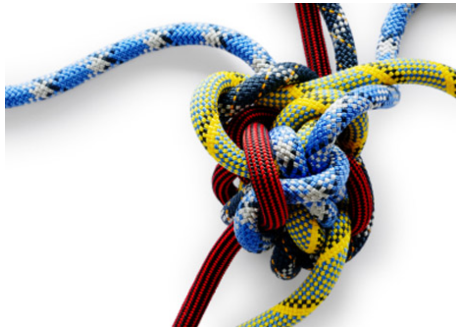
http://getacoder.com

- However encryption complicates using the data
  - Slows everything down
  - Can you query while encrypted? (Homomorphic encryption?)
  - Message routing on sensitive fields (in headers)
  - Managing and rotating the keys
  - What about restore on disaster recovery?

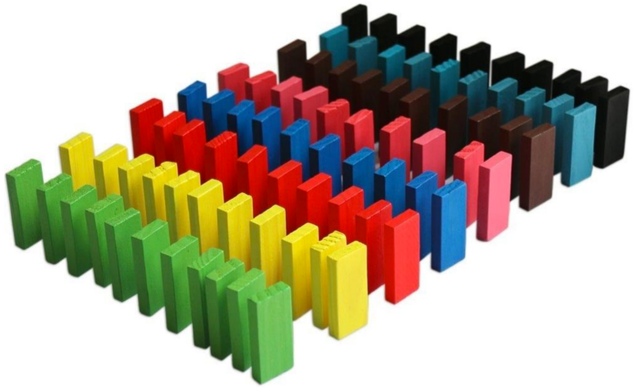
http://slate.com

Simplicity & Standardisation

- Complexity is the enemy of security
  - *“You can’t secure what you don’t understand”* - Schneier
  - Special cases will be forgotten
- Simplify, Standardise and Automate
  - Simpler things are easier to check and secure
  - Standardising an approach means there are no special cases to forget to handle
  - Automation eliminates human inconsistencies from the process so avoiding a type of risk



http://innovationmanagement.se/



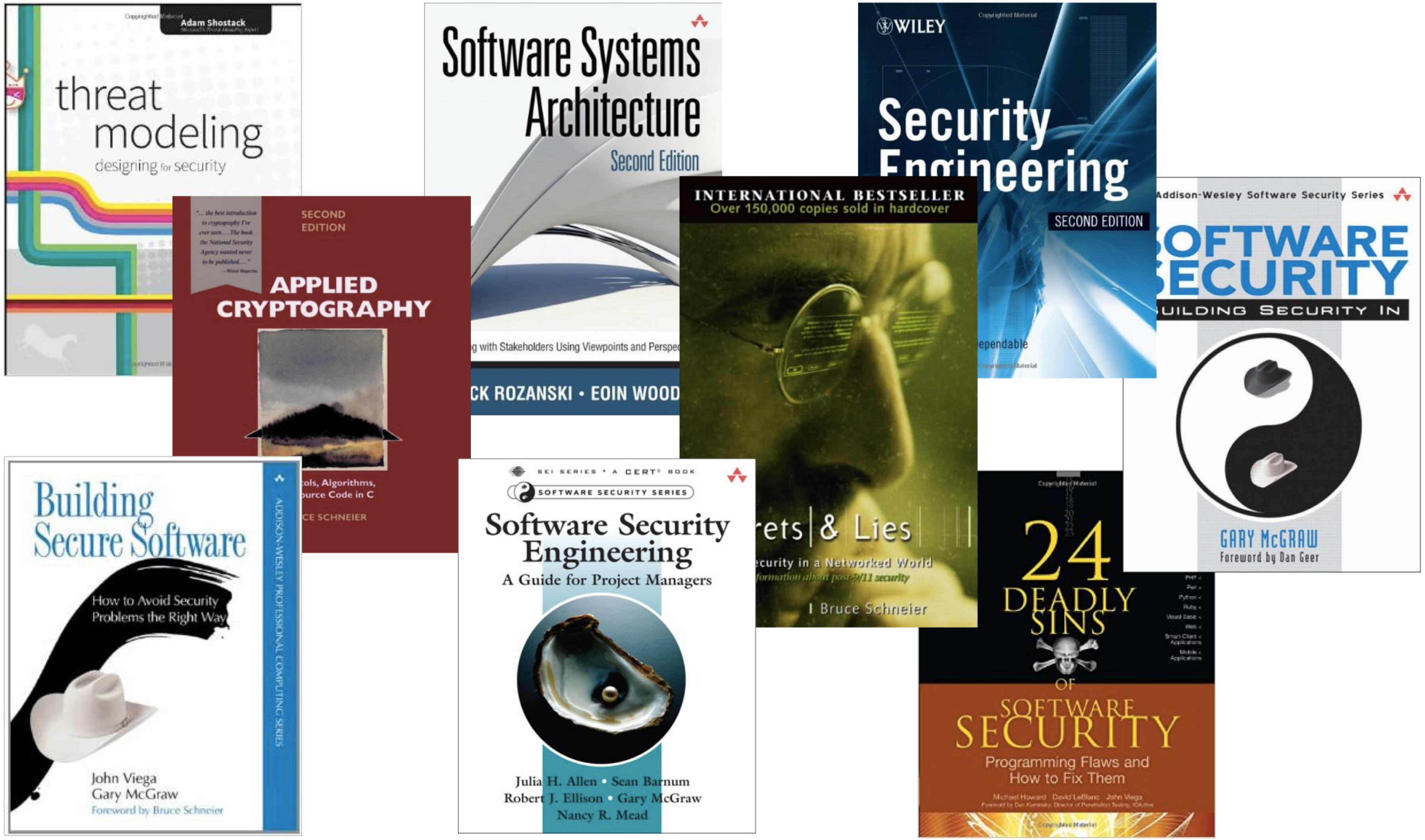
https://www.aliexpress.com

Summary

Summary

- Much of the technology we use is inherently insecure
  - Mitigation needs to be part of application development
- Attacking systems is becoming industrialised
  - Digital transformation is providing more valuable, insecure targets
- Fundamental attack vectors appear again and again
  - Injection, interception, page manipulation, validation, configuration, …
- Most real attacks exploit a series of vulnerabilities
  - Each vulnerability may not look serious, the combination is
- Most mitigations not difficult but need to be applied consistently
  - … and may conflict with other desirable qualities

Books


Adam Shostack
threat modeling
designing for security
SECOND EDITION
APPLIED CRYPTOGRAPHY
ols, Algorithms,
urce Code in C
CE SCHNEIER
Software Systems Architecture
Second Edition
g with Stakeholders Using Viewpoints and Perspe
CK ROZANSKI • EOIN WOOD
WILEY
Security Engineering
SECOND EDITION
ependable
INTERNATIONAL BESTSELLER
Over 150,000 copies sold in hardcover
rets & Lies
ecurity in a Networked World
formation about post-9/11 security
Bruce Schneier
Addison-Wesley Software Security Series
SOFTWARE SECURITY
BUILDING SECURITY IN
GARY McGRAW
Foreword by Dan Geer
Building Secure Software
How to Avoid Security Problems the Right Way
John Viega
Gary McGraw
Foreword by Bruce Schneier
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES
SEI SERIES • A CERT BOOK
SOFTWARE SECURITY SERIES
Software Security Engineering
A Guide for Project Managers
Julia H. Allen • Sean Barnum
Robert J. Ellison • Gary McGraw
Nancy R. Mead
24 DEADLY SINS OF SOFTWARE SECURITY
Programming Flaws and How to Fix Them
Michael Howard David LeBlanc John Viega

Thank You

Eoin Woods
Endava
@eoinwoodz
eoin.woods@endava.com