dionach

REAL SECURITY IN A VIRTUAL WORLD

How to break in

Tecniche avanzate di pen testing in ambito
Web Application, Internal Network and Social Engineering

ISO 27001 CERTIFICATION EUROPE™

PCI Security Standards Council™ QUALIFIED SECURITY ASSESSOR

CREST

Agenda

Time	Agenda Item
9:30 – 10:00	Introduction
10:00 – 10:45	Web Application Penetration Test
10:45 – 11:00	15min break
11:00 – 11:45	Social Engineering
11:45 – 12:00	15min break
12:00 – 13:00	Windows Challenge
13:00 – 15:00	Lunch
15:00 – 16:00	Internal Penetration Test
16:00 – 16:15	15min break
16:15– 17:00	Physical Social Engineering Test
17:00 – 18:30	Pi Challenge

dionach
REAL SECURITY IN A VIRTUAL WORLD

UNICAM
Università di Camerino
1336

./Whoami

- Mike Manzotti, Lead Consultant @ Dionach UK
- CISSP, CRT, CCT Inf, OSCP, OPST, CCNA, CCNA Security
- ~5 years of experience
- Penetration testing:
  - Web applications
  - Mobile apps
  - Internal & External Networks
  - Social Engineering
  - Red Teaming
- Security Audit:
  - ISO 27001

UNICAM Università di Camerino 1336

dionach
REAL SECURITY IN A VIRTUAL WORLD

Penetration Test ?

- is an attack on a computer system with the intention of finding security weaknesses, potentially gaining access to it, its functionality and data (*Wikipedia*)
- Commonly associated to Hacking… but yeah it's a real job ☺
- is a process which involves the following phases:
  - Information gathering
  - Identifying vulnerabilities
  - Manual testing and verification of false positives
  - Reporting
- Who needs a pen test?

UNICAM Università di Camerino 1336

dionach REAL SECURITY IN A VIRTUAL WORLD

dionach

REAL SECURITY IN A VIRTUAL WORLD

Web Application Penetration Test

ISO 27001 CERTIFICATION EUROPE™

PCI Security Standards Council™ QUALIFIED SECURITY ASSESSOR

CREST

Introduction

- The Open Web Application Security Project
- https://www.owasp.org/

- “OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted”

Introduction

- OWASP support several projects to increase web application security

- This presentation focuses on the OWASP Top 10 Project, which aims to increase awareness of web application security and to provide a broad consensus of what the most critical web applications security flaws are

OWASP Top 10 2013

1. Injection
2. Broken Authentication & Session Management
3. Cross-Site Scripting (XSS)
4. Insecure Direct Object References
5. Security Misconfiguration

OWASP Top 10 2013 (cont.)

6. Sensitive Data Exposure
7. Missing Function Level Access Control
8. Cross-Site Request Forgery (CSRF)
9. Using Known Vulnerable Components
10. Unvalidated Redirects and Forwards

A1: Injection

- Injection attacks occur when user supplied data are not sanitised or encoded prior to submitting to an interpreter. These typically allow an attacker to access data or execute commands they are not authorised to.

- SQL queries, LDAP queries, XPath queries, OS commands, program arguments, etc.

- This is a critical risk that could impact confidentiality integrity and availability. This could allow an attacker to gain access to the entire database or execute system commands which could lead to a full system compromise.

A1: Injection

- SQL Injection:
- http://192.168.163.126/staff.php?id=2'#
- http://192.168.163.126/staff.php?id=2%27+union+select+1,@@version,3,4,5%23

- Code Injection:
- https://x.x.x.x/bugtracker/manage_proj_page.php?sort=']);}error_reporting(0);system("cat /etc/*-release");%23

A2: Broken Authentication and Session Management

- Broken authentication and session management could allow an attacker to compromise passwords, keys, session cookies, or exploit other implementation flaws to assume other users' identities.

- These vulnerabilities could affect areas such as logout, password management, timeouts, remember me, secret question, account update, etc.

- The vulnerability could be caused by:
  1) Weak password complexity - 2) No account lockout
  3) Predictable session cookie - 4) Session cookie not marked secure or HTTP only
  5) Session fixation - 6) Plain text passwords in DB
  7) Change account details of other users - 7) Plain text HTTP instead of HTTPS

- Risk ranges from Low to Critical. Successful exploitation could allow an attacker to compromise users accounts, thus accessing their data or escalating their privileges. This could have reputational, financial and compliance impact.

UNICAM Università di Camerino 1336

dionach REAL SECURITY IN A VIRTUAL WORLD

A2: Broken Authentication and Session Management

- The following example shows a vulnerability discovered in an application that allowed an attacker to reset any user's password. The attacker only needs to submit the username and their chosen password:

POST http://X.X.X.X/STG/servlet/BrowserServlet

```
command=repeatpassword&requestType=UTILITY.ROUTINE&routineName=OS.PASSWORD
&routineArgs=PROCESS.REPEAT%3APENTGLOB1%3ADionach20%3ADionach20
```

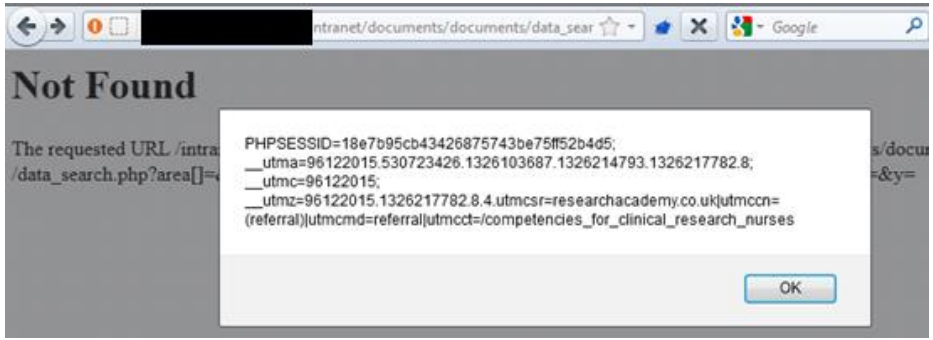
A3: Cross-Site Scripting (XSS)

- XSS attacks occur when user supplied data are not sanitised or encoded prior to displaying them in the users browser. Three types:
    1. Stored XSS
    2. Reflected XSS
    3. Dom based XSS

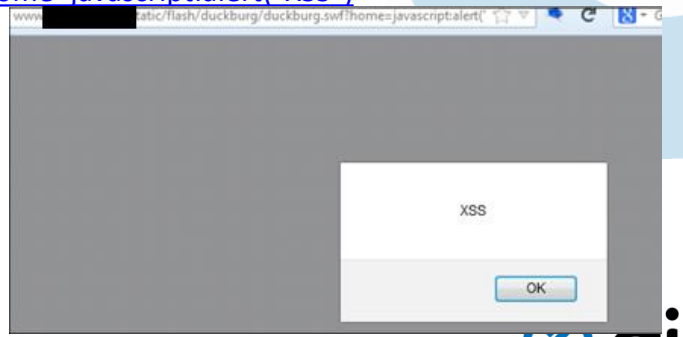
- This could be either a critical or a high risk. It could allow an attacker to hijack users session and access their data or escalate their privileges. An attacker could also use it to take control of the victims browser in order to perform drive by downloads or redirect the user to a malicious website.
- Stored XSS could also allow attacker to deface the website.
- Reputational damage.

A3: Cross-Site Scripting (XSS)

- http://XXXX.XXX/intranet/documents/documents/data_search.php?area[]=&in_docs=&in_doc_folders=&in_pages=&in_pub_folders=&in_forum=&q=&x=0&y=<script>alert(document.cookie)</script>#



- http://www.xxxx.xxx/static/flash/duckburg/duckburg.swf?home=javascript:alert("XSS")



UNICAM Università di Camerino 1336

dionach
REAL SECURITY IN A VIRTUAL WORLD

A4: Insecure Direct Object References

- The vulnerability allows an attacker to change direct object references and access data they are not authorised to access.
- Objects could be files, directories or database keys.
- This is a high risk and could have an impact on confidentiality and integrity. This could have reputational, financial and compliance impact.
- http://XXXX.org.uk/intranet/people/photos/***5.jpg***
- http://example.com/app/accountInfo?acct=***NOTMYACCT***

A5: Security Misconfiguration

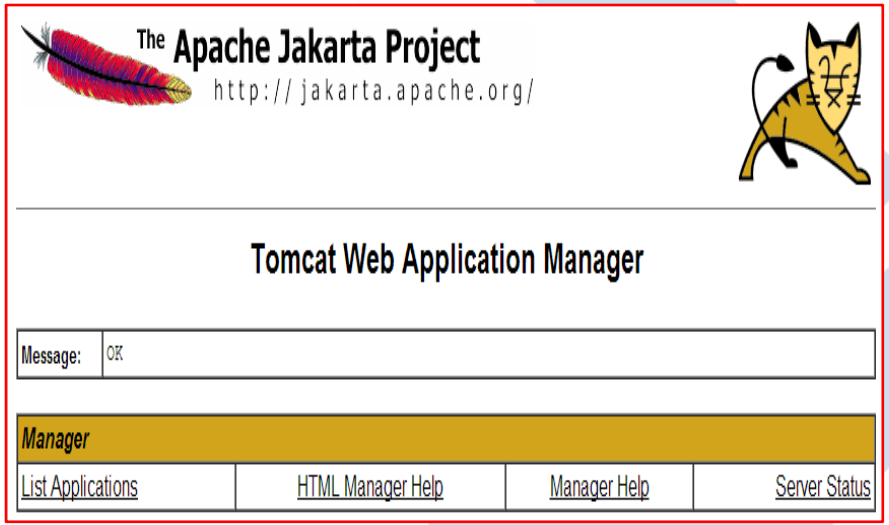
- Security Misconfiguration also includes hardening

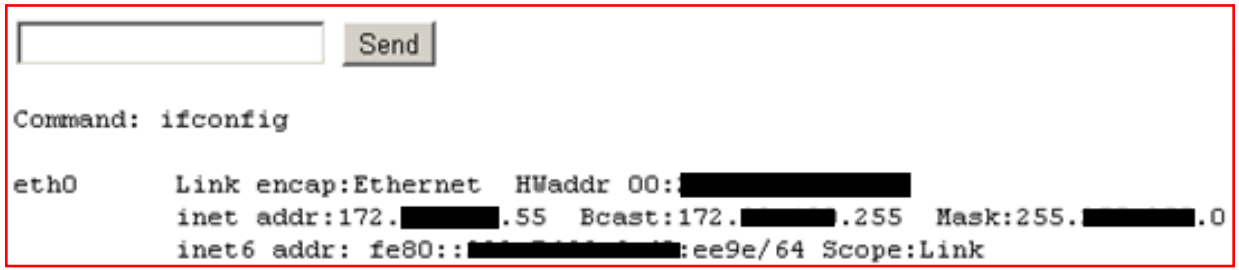
- Issues can include:
  - Old versions of software
  - Unnecessary services
  - Default passwords
  - Detailed error messages
  - Default settings

- Risk ranges from low to critical. For instance, detailed error messages are low risk as the information included is limited but a default administration service left enabled could be critical.

A5: Security Misconfiguration

- Example 1 Default Administration service with default password - By default the ‘Tomcat’ web server has a management console which uses a well known default username and password. If this is not disabled or changed an attacker could log in and upload a special ‘command shell’ program, allowing them to run commands on the web server.





A6: Sensitive Data Exposure

Storage or Transport. First, Storage

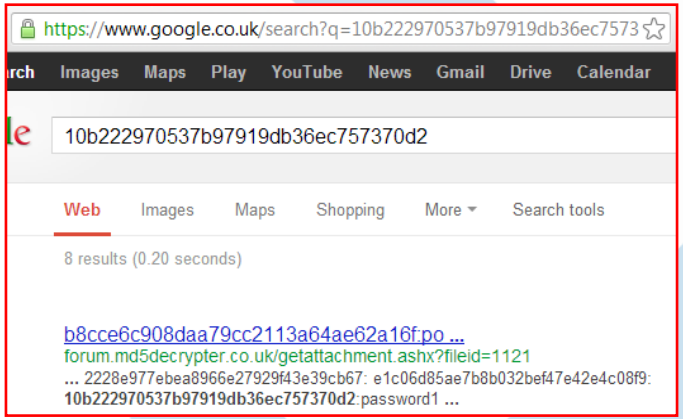
- Web applications should store sensitive information, such as passwords, in an encrypted format
- Many types of encryption are available, and while many are very strong others have known weaknesses or can easily be broken and therefore should not be used
- Data must be unencrypted at some point in order to be displayed. Attackers may attempt to exploit the application at this point to access the data in clear text
- Data that should be encrypted, such as credit card details, may not be encrypted at all
- If data can be obtained in clear text then the risk would most likely be high or critical depending on the type of information stored.

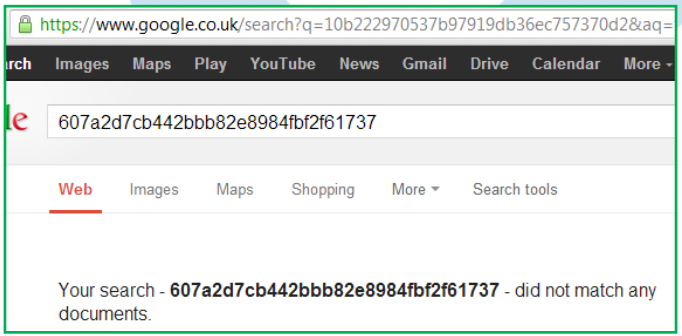
A6: Sensitive Data Exposure

- Example 1 Weak Encryption – A web application stores passwords using an unsalted hash
- If an attacker could obtain the list of passwords, the list could be brute forced (guess every combination until the correct one is found) in a matter of days compared to the years (or thousands of years) it would take to brute force salted hashes.
- In the first screenshot I hash ‘password1’ and search for the result, the third result in Google shows the clear text password in the description
- In the second search I hash the same password but use the salt ‘salt1’ and Google shows no results

---

- *(A ‘hash’ is a type of encryption that is designed to work one way. Clear text can be easily hashed but it is hard to impossible to use a hash to calculate the clear text)*
- *(A ‘salt’ is a length of random data added to the clear text when making the hash. This changes the value of the hash and makes it harder to guess the password from looking up recognised hashes)*





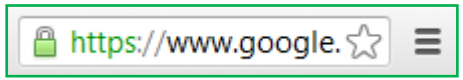
UNICAM Università di Camerino 1336

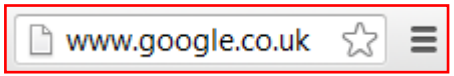
dionach REAL SECURITY IN A VIRTUAL WORLD

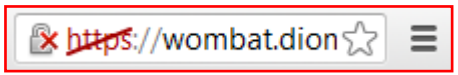
A6: Sensitive Data Exposure

Storage or Transport. Second, Transport

- This allows an attacker to listen in on communication between the web browser and the server.
- Issues commonly arise when websites do not encrypt sensitive data using SSL connections (HTTPS://) or when SSL has been configured incorrectly.
- If encryption isn't used, or a misconfiguration can be exploited, then an attacker could intercept all traffic between the user and the web server.
- Risk varies from low to medium. Many issues rely on other vulnerabilities and normally require an attacker to be on the same local network, such as an open wireless network.





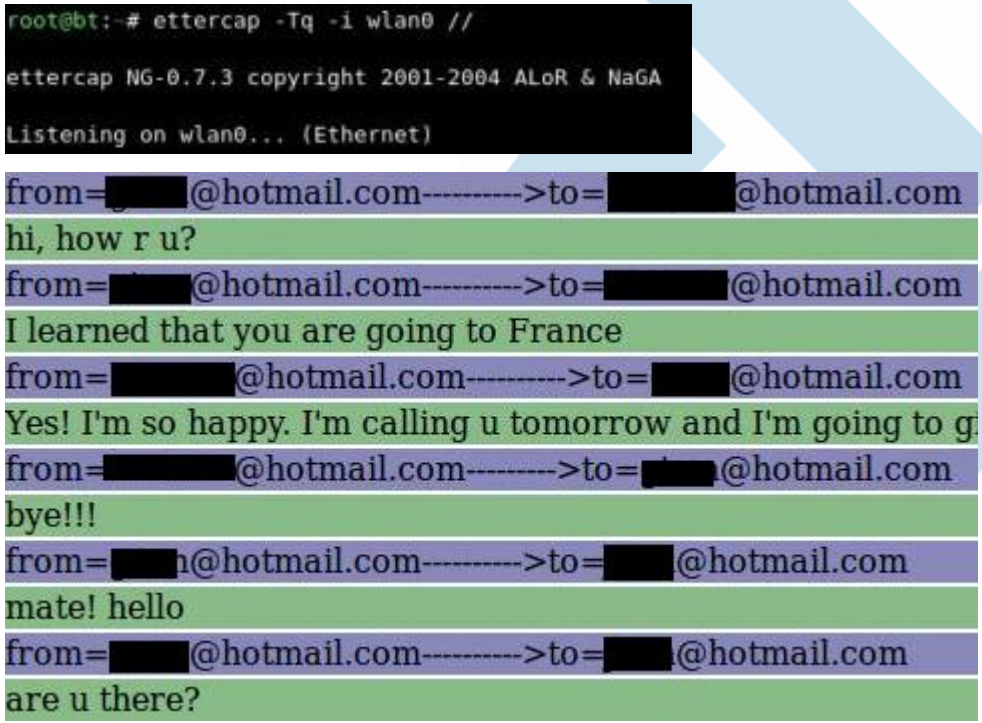


UNICAM Università di Camerino 1336

dionach
REAL SECURITY IN A VIRTUAL WORLD

A6: Sensitive Data Exposure

- Example 1 No SSL – An attacker uses the open wireless in an coffee shop
- The attacker uses the program ‘ettercap’ to monitor traffic from other users on the network and automatically capture conversations to an online instant messaging service which does not encrypt traffic with SSL.



UNICAM Università di Camerino 1336

dionach REAL SECURITY IN A VIRTUAL WORLD

A6: Sensitive Data Exposure

- Example 2 Misconfiguration – Facebook used to log users in using HTTPS but then used HTTP for the site content

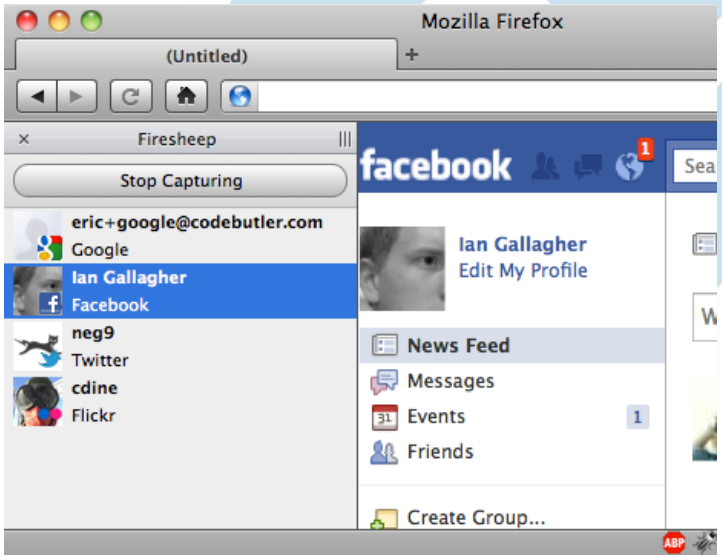
- This meant that although the users password was sent encrypted the session cookie (a small token used to keep the user logged in) was not. This could be easily exploited using the Firefox ‘Firesheep’ plugin, which provided a list of users on Facebook on the same network and allowed you to select a user in order to access Facebook as that user.

```
HTTPS Login Request

POST /login HTTP/1.1
HOST: facebook.com
user=test&password=1234
```

```
HTTP Profile Request

GET /myprofile HTTP/1.1
HOST: facebook.com
COOKIE: SESSION=ASDF22AW2F5
```



UNICAM Università di Camerino 1336

dionach
REAL SECURITY IN A VIRTUAL WORLD

A7: Missing Function Level Access Control

- This vulnerability allows users to access pages that they are not supposed to have permission to. For example, an anonymous user may be able to access profile pages and a normal user may be able to access administration pages.

- This occurs when a web app restricts access to pages by not displaying links to the page unless the user has permission but does not restrict access within the page itself. Therefore, an attacker could manually browse to the page in order to access it.

- Risk could range from low to critical depending on the nature of the pages accessible and the functions that can be performed on them.

A7: Missing Function Level Access Control

- Example - Bob and Jack both use a blog. Bob can edit his own post by using the URL '/#/edit'. He does not have a link to edit Jack's post but, as this page does not restrict the URL access, Bob can manually type the URL, using the number for Jack's post, in order to edit another user's post.

dionach
REAL SECURITY IN A VIRTUAL WORLD

A8: Cross-Site Request Forgery (CSRF)

- A CSRF vulnerability allows an attacker to force the victim's browser to perform actions the user did not intend to perform which the application thinks are legitimate requests from the victim.
- Affects HTTP GET and POST requests
- Victim would need to be logged in to the vulnerable application and then visit a website under the attackers control. Attackers could use XSS, image tags or various other techniques.
- This is a medium risk vulnerability. Successful exploitation could affects data integrity.
- This could have reputational impact.

A8: Cross-Site Request Forgery (CSRF)

- Legitimate request:
- http://example.com/app/transferFunds?amount=1500&destinationAccount=4673243243

- Attack website:

-

A9: Using Known Vulnerable Components

Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover. Applications using components with known vulnerabilities may undermine application defences and enable a range of possible attacks and impacts.

A9: Using Known Vulnerable Components

Example

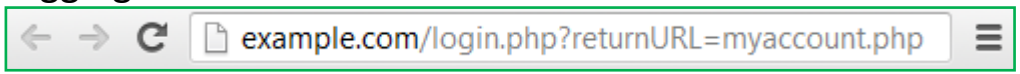
- jQuery is a very popular JavaScript library/component
- http://www.cvedetails.com/cve/CVE-2011-4969/
  - XSS vulnerability
- Not your own code, but your website is vulnerable if you use this old version

A10: Unvalidated Redirects and Forwards

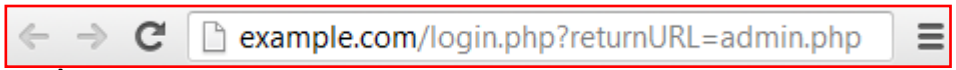
- Websites use redirects and forwards to send users to another page, often after performing another action, such as logging in.
- If a website does not properly validate the address of the redirect or forward an attacker could exploit this to send the browser to a page of their choosing.
- Ranges from low to critical depending on the nature of the site and what the redirect is being used for.

A10: Unvalidated Redirects and Forwards

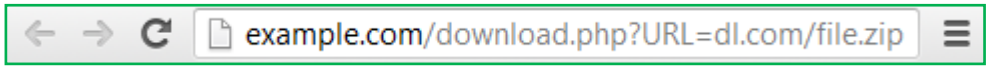
- Example 1
  - A website uses the ‘returnURL’ parameter to send a user to their account page after logging in



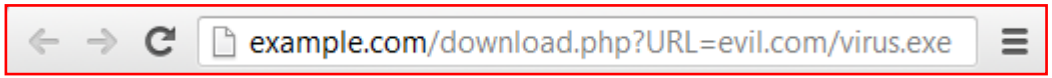
  - An attacker could alter this to request a page to which he would not normally have access



- Example 2
  - A website has links various downloads, they use a parameter to redirect to the pages



  - An attacker could change the parameter and use the link in a phishing email. Users will be sent to the real webpage before the download opens, adding credibility to the attack



UNICAM Università di Camerino 1336

dionach REAL SECURITY IN A VIRTUAL WORLD

A10: Unvalidated Redirects and Forwards

- A real world example of this is the Outlook Web Access 2003 redirect vulnerability
- When clicking a link in an email OWA 2003 used a redirect to send the user to the right URL
- If the user was not logged in when clicking the link they would need to enter their username and password
- An attacker could easily make a fake login page, which the real login page redirects to, showing a fake ‘Incorrect password’ error, most users would then entered their password again, unaware that they are using the fake login page

https://webmail.local/owa/redir.aspx?C=asdf&URL=http://evil.com/fakeloginpage.aspx

UNICAM Università di Camerino 1336

dionach REAL SECURITY IN A VIRTUAL WORLD

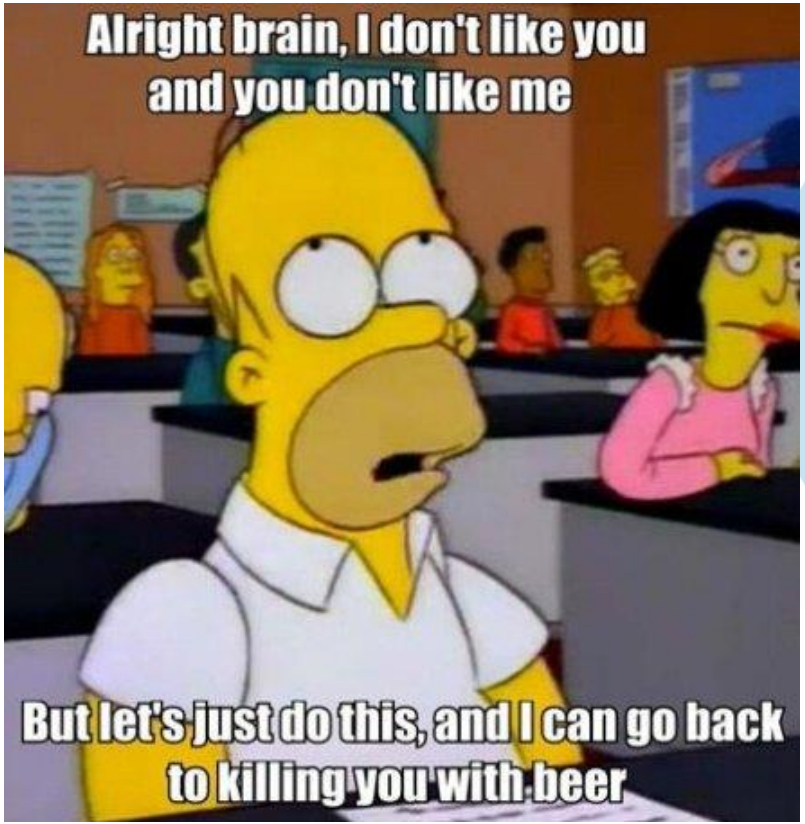
Tools

- Nmap (-p- TCP, common UDP)
- Dirbuster
- Nikto
- Nessus
- SQLmap
- CMSmap
- Burpsuite Pro
- Public resources: exploit-db, inj3ct0r, security focus, github…

Demo

- SQL Injection (SQLi)
- Insufficient Access Control
- Arbitrary File Upload
- Remote Command Execution (RCE)
- Reflected, Stored DOM Cross-site scripting (RXSS, SXSS, DXSS)
- Cross-site request forgery (CSRF)

Demo



Any Questions

UNICAM
Università di Camerino
1336

dionach
REAL SECURITY IN A VIRTUAL WORLD