Fuente: IA Stable Diffusion

Laboratorio 2. Descubrimiento de información y OSINT

DEPARTMENTO DE INFORMÁTICA. UNIVERSIDAD DE OVIEDO

Seguridad de Sistemas Informáticos | 2022 – 2023 (V3.1 *"S-81 Isaac Peral"*)

CONTENIDO

Infraestructura de este laboratorio 3
Bloque 1: Exploración de contenidos web 5
Fichero *robots.txt* 6
Metadatos de documentos 6
Bloque 2: buscando "oro" 8
Google Hacking 9
Shodan 9
Censys 10
The Wayback Machine 10
Bloque 3: Los DNS al sol 13
Descubrimiento de subdominios 14
Bloque 4: Inspección de IPs 16
Analizar un rango de IPs 17
Escaneos globales con *zmap* 17
Escaneos en LAN con *nmap* 19
Bloque 5. Es algo personal: privacidad, seguridad de navegación 20
Blacklight: Inspector de privacidad de la web 21
Crear un entorno de navegación seguro y verdaderamente privado en tu máquina virtual 21
Have I been pwnd? 23
Geolocalización de un remitente de correo electrónico (o de cualquier IP) 24
Insignias y Autoevaluación 26

AVISO

Este documento forma parte de la asignatura "*Seguridad de Sistemas Informáticos*", impartida en la *Escuela de Ingeniería Informática* de la *Universidad de Oviedo*. Es fruto del trabajo continuado de elaboración, soporte, mejora, actualización y revisión del siguiente equipo de profesores desde el año 2019

- Enrique Juan de Andrés Galiana
- Fernando Cano Espinosa
- Miguel Riesco Albizu
- José Manuel Redondo López
- Luís Vinuesa Martínez

Te pedimos por favor que **NO lo compartas públicamente en Internet**. No obstante, entendemos que puedas considerar este material interesante para otras personas. Por ese motivo, hemos creado una versión de este adaptada para que pueda cursarse de forma online, disponible gratuitamente para todo el mundo y que puedes encontrar en esta dirección: https://ocw.uniovi.es/course/view.php?id=109

A diferencia de esta versión, **la versión libre puedes promocionarla todo lo que quieras**, que para eso está 😊

GRACIAS POR TU COLABORACIÓN

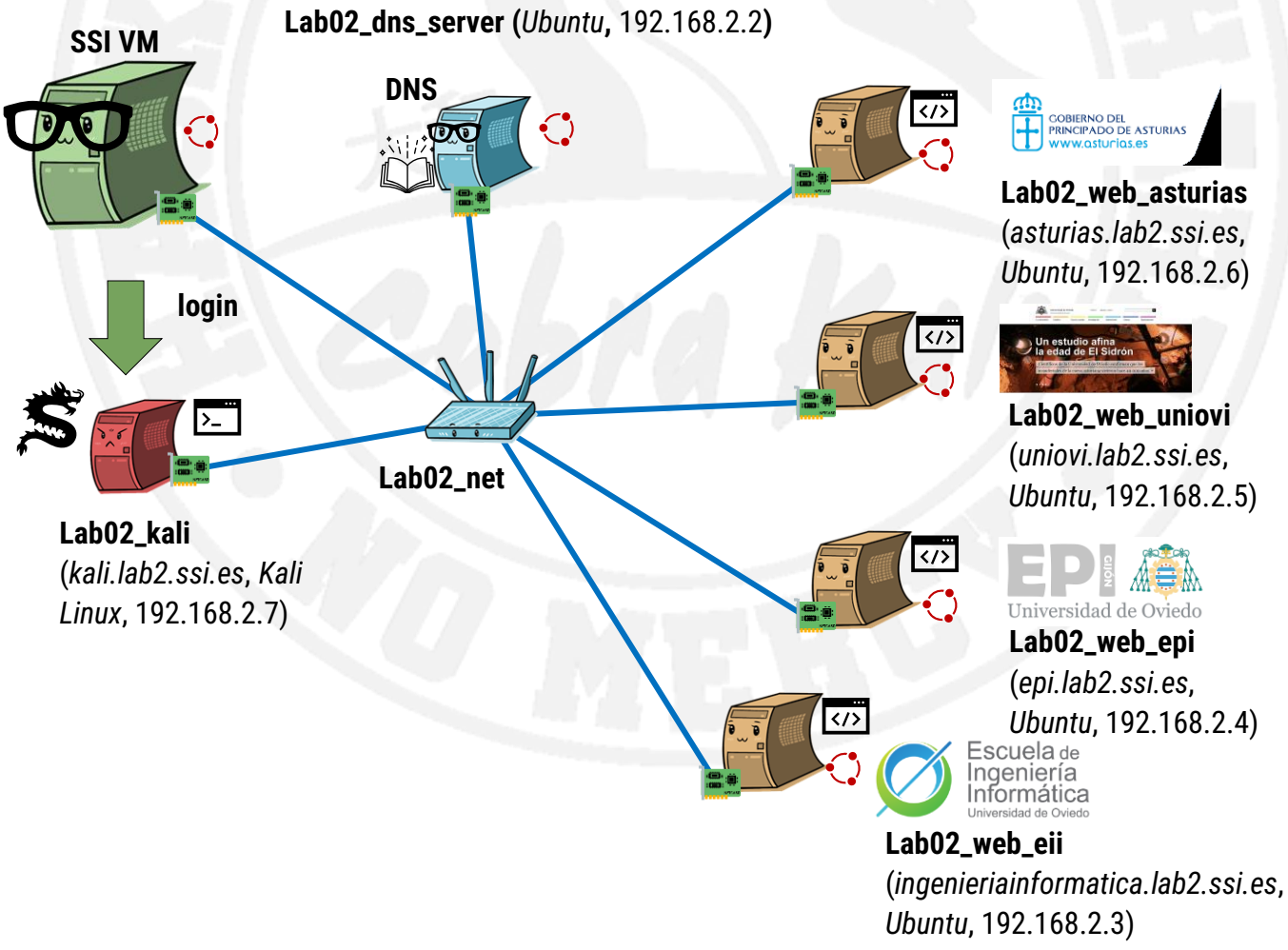
INFRAESTRUCTURA DE ESTE LABORATORIO

NOTA: Antes de empezar, por favor asegúrate de la máquina virtual de la asignatura tiene una serie de paquetes instalados para que los laboratorios funcionen correctamente. Para ello **se necesita ejecutar lo siguiente**: `sudo apt install gnome-keyring gnupg2 pass`

En primer lugar, asegúrate de tener la estructura de carpetas `ssi_labs` que os damos dentro de tu máquina virtual. Este laboratorio (`lab_02`) consiste en una infraestructura compuesta por varios contenedores. No todas las actividades requieren el uso de todos ellos. Consulta el siguiente diagrama para entender el diseño de la infraestructura:

- **Una LAN interna**: (*lab02_net*, `192.168.2.0/24`)
- **4 servidores web distintos**
- **Un DNS**: proporciona nombres para cada servidor web en el dominio `lab2.ssi.es.`
- **Una máquina Kali** (`lab2_kali`) con varias herramientas instaladas solo para fines de **enumeración** que se describen en este documento. Esto también implica que no tienes que instalarlas. Esta máquina *Kali* es el punto de partida del laboratorio.

IMPORTANTE: Asegúrate de haber leído y entendido las instrucciones del documento "*Infraestructuras automatizadas de SSI*" para entender cómo trabajar con este tipo de infraestructuras de laboratorios automatizados.



BLOQUE 1: EXPLORACIÓN DE CONTENIDOS WEB

Contenido Actual

Fichero *robots.txt*

Aplicación práctica: *Quieres saber si una web está revelando rutas "secreta" en este fichero público*

Parte del ***Robot Exclusion Standard*,** este archivo se usa para indicar a los motores de búsqueda que siguen ese estándar que no indexen las URL correspondientes a ciertas entradas que se encuentran bajo la palabra clave `Disallow` en un archivo que se llama `robots.txt`. Estos archivos siempre son accesibles usando este patrón: `<URL de la web>/robots.txt`. Algunos desarrolladores lo usan para que *Google* u otros motores de búsqueda no indexen contenido "sensible". El problema es que este archivo es público y accesible, por lo que los motores pueden leer su contenido y, por tanto, ¡tú también puedes! 😊. Puedes encontrar más información aquí: https://www.synopsys.com/blogs/software-security/robots-txt/

Resultados esperados: Para completar esta actividad es necesario elegir cualquier web que prefieras y analizar su archivo `robots.txt`, determinando si puedes encontrar algo interesante o que comprometa la seguridad (y por qué).

Metadatos de documentos

Aplicación práctica: *Necesitas saber si los documentos, imágenes, etc. de un sitio web están revelando demasiada información acerca de sus propietarios o autores*

Los metadatos son datos adjuntos a un archivo, **pero no como parte de su contenido, sino normalmente en sus cabeceras de archivo.** Los metadatos se pueden adjuntar a un archivo como texto sin cifrar o cifrado. Mucha gente ignora que los archivos pueden tener metadatos. Peor aún, **mucha gente tampoco sabe que muchos programas ponen automáticamente metadatos en los archivos que generan sin su consentimiento, o sin ser conscientes de ello**. La importancia de estos (meta)datos puede variar desde información trivial a verdaderamente comprometedora (¡díselo a los usuarios de *Parler!)*. Hoy en día, muchos servicios eliminan automáticamente los metadatos de los archivos que subes (las redes sociales, excepto *Parler,* lo hacen), pero no debes confiar en que lo hagan el 100% del tiempo (¡pueden tener errores!). También pueden conservar los metadatos originales para su uso particular (*marketing*), aunque los eliminen de los archivos que se muestran al público. En resumen, **debes eliminar los metadatos de los archivos que cargues** usando los programas adecuados por si acaso 😊.

FOCA es un programa de *Windows* que extrae y analiza los metadatos de diferentes tipos de archivos, que también implementa búsqueda en dominios, servidores DNS, URL y documentos publicados. FOCA se puede descargar desde aquí: https://github.com/ElevenPaths/FOCA. Hay una versión gratuita de código abierto y una versión comercial. Desafortunadamente, requiere configurar una *base de datos SQL* (*SQLExpress* o *SQL Server)* que lleva tiempo y esfuerzo instalar. Esto no forma parte de los objetivos de este curso, por lo que, en lugar de esta herramienta, vamos a utilizar una en línea más simple del mismo fabricante, que analiza los metadatos de documentos individuales y otros tipos de archivos: ***Metashield Clean-up Online*** (https://metashieldclean-up.elevenpaths.com/).

Esta página web permite subir cualquier archivo de una lista de tipos de archivo aceptados y extrae sus metadatos para ver si hay algo útil o sospechoso en ellos. No obstante, **es común que dicha página esté sobrecargada** y no pueda examinar un archivo que subamos en un momento dado. Si la web no está disponible, podemos usar una herramienta local llamada `exiftool` que veremos en el laboratorio 4 para otra cosa. En este

laboratorio la usaremos simplemente para **leer los metadatos completos de un archivo** que nos hayamos descargado de la web (admite zips, imágenes de muchas clases, documentos *Office*, PDF...) haciendo uso de sus opciones: https://manpages.ubuntu.com/manpages/trusty/en/man1/exiftool.1p.html. Por favor instálala si no está ya en la MV. Concretamente, la opción más útil para leer metadatos es `exiftool -a -u -g1 <nombre del fichero>`

Resultados esperados: Esta actividad se completará cuando elijas cualquier archivo local o descargado de los tipos aceptados y lo analices con la web o la herramienta `exiftool` para ver si encuentra algo interesante.

BLOQUE 2: BUSCANDO "ORO" 😊

Google Hacking

Aplicación práctica: *Puedes usar el motor de búsqueda de Google para encontrar información comprometedora de cualquier objetivo en Internet*

Google Hacking es la capacidad de realizar ciertas operaciones de búsqueda manuales en *Google,* normalmente combinadas con el operador `site:`. También se aplica sobre otros buscadores, aunque cada motor de búsqueda usa sus propios operadores. Estas operaciones de búsqueda tienen como objetivo detectar vulnerabilidades, problemas de configuración o información que puedan provocar un ataque. Hay muchas operaciones de búsqueda predefinidas que cubren diferentes tipos de ataques / procedimientos de recopilación de información en la ***Google Hacking Database*** **(GHDB):** https://www.exploit-db.com/google-hacking-database. El procedimiento para usar esta base de datos es:

- Elegir una categoría de búsqueda y una consulta
- Realizar la consulta sobre un objetivo concreto (utilizando el operador de búsqueda `site:`, https://support.google.com/websearch/answer/2466433?hl=es) y ver si el resultado coincide con el esperado, según la búsqueda elegida.

Hay una imagen con los pasos que requiere este procedimiento en el **primer seminario** del curso. Hay las siguientes categorías de búsquedas predefinidas

Footholds	Files Containing Juicy Info
Files Containing Usernames	Files Containing Passwords
Sensitive Directories	Sensitive Online Shopping Info
Web Server Detection	Network or Vulnerability Data
Vulnerable Files	Pages Containing Login Portals
Vulnerable Servers	Various Online Devices
Error Messages	Advisories and Vulnerabilities

También puedes encontrar direcciones con más ejemplos en el **primer seminario.**

Resultados esperados: Esta actividad se completará cuando elijas un sitio web y realices algunas operaciones de búsqueda con la base de datos de *Google Hacking* para familiarizarte con el proceso, analizando si encuentras algo sospechoso. NOTA: Es más que probable que *Google* te pida verificar tu identidad tras usar 2 o 3 búsquedas de este tipo, para comprobar que no eres un bot.

Shodan

Aplicación práctica: *Puedes usar el motor de búsqueda de máquinas Shodan para averiguar información acerca de cualquier dispositivo en Internet*

Shodan (http://www.shodanhq.com/) es un buscador capaz de encontrar dispositivos en lugar de páginas web: *routers*, servidores, cámaras de grabación CCTV, semáforos, ... Se pueden encontrar más detalles sobre su funcionalidad y uso en el primer **seminario** (también incluye URLs con ejemplos de uso).

Resultados esperados: Esta actividad se completará cuando puedas hacer una petición en *Shodan* sobre una sola URL o IP, analizando la salida que se obtiene y por qué crees que su información puede ser un peligro para el objetivo. Tienes más libertad si creas una cuenta gratuita, pero no es necesaria para hacer este ejercicio si te limitas a IPs individuales.

Censys

Aplicación práctica: *Puedes usar el motor de búsqueda de máquinas Censys para averiguar información acerca de cualquier dispositivo público en Internet*

Censys es como *Shodan,* pero presenta la información de una manera más estructurada. Permite hacer operaciones de búsqueda basadas en los datos que obtiene de los servidores que analiza en todo Internet. **El primer seminario** ofrece más detalles al respecto. Puedes encontrar ejemplos de uso aquí:

- https://search.censys.io/search/examples?resource=hosts
- https://support.censys.io/hc/en-us/articles/360059720271-Search-2-0-Example-Host-Queries

Resultados esperados: **Esta actividad se completará cuando puedas usar *Censys* para escanear un rango de IP o red (por ejemplo, un rango que pertenezca a Uniovi). A continuación, se debe analizar la salida que proporciona y la información que se muestra sobre cada host. Analizar la información para determinar por qué podría ser peligrosa desde el punto de vista de un ataque.**

The Wayback Machine

Aplicación práctica: *Puedes usar el motor de búsqueda Internet Archive (aka "Wayback Machine") para averiguar información acerca de sitios web publicados en el pasado en muchos dominios de Internet*

*(**NOTA**: La Wayback Machine es masiva, y por tanto es bastante lenta muchas veces. No es tu conexión o tu ISP, es que es así, y no puedes hacer nada para evitarlo, lo sentimos 🥰)*

La mayoría de las personas (¡incluyendo una cantidad considerable de administradores web!) están acostumbradas a buscar contenido comprometedor en las páginas web para eliminarlo, pero no incluyen el contenido comprometedor que **ESTABA** allí, pero ya no, o eliminan el contenido sin tener en cuenta que puede quedar archivado en la *Wayback Machine*. Este contenido puede resultar muy útil para los atacantes, y esta sección de laboratorio le enseñará cómo lidiar con él.

Hay una página web de referencia para conocer la "historia" de cualquier sitio web en Internet y su nombre es "***The Wayback Machine***" (también conocido como *The Internet Archive):* https://archive.org/web/. Este sitio web es un archivo masivo de contenidos pasados ordenandos por fecha que fueron mostrados por cualquier sitio web público que estuvo activo en Internet, pero también tiene usos "hacker". Por ejemplo, puedes inspeccionar TODAS las URL que se han extraído de un dominio en particular utilizando una consulta como esta: http://web.archive.org/web/*/<URL>/* (por ejemplo. http://web.archive.org/web/*/http://www.uniovi.es/*). Ten en cuenta que la carga de URLs es asíncrona, por lo que puede tardar un tiempo. No te preocupes si inicialmente te encuentras una tabla vacía (para Uniovi, generalmente tarda 30s en cargarse).

Una vez que tenemos la lista histórica de URL de un sitio, podemos utilizar esta información para obtener información muy interesante sobre el mismo. Puedes encontrar más detalles aquí: https://www.elladodelmal.com/2013/04/hacking-con-archivecom-wayback-machine.html

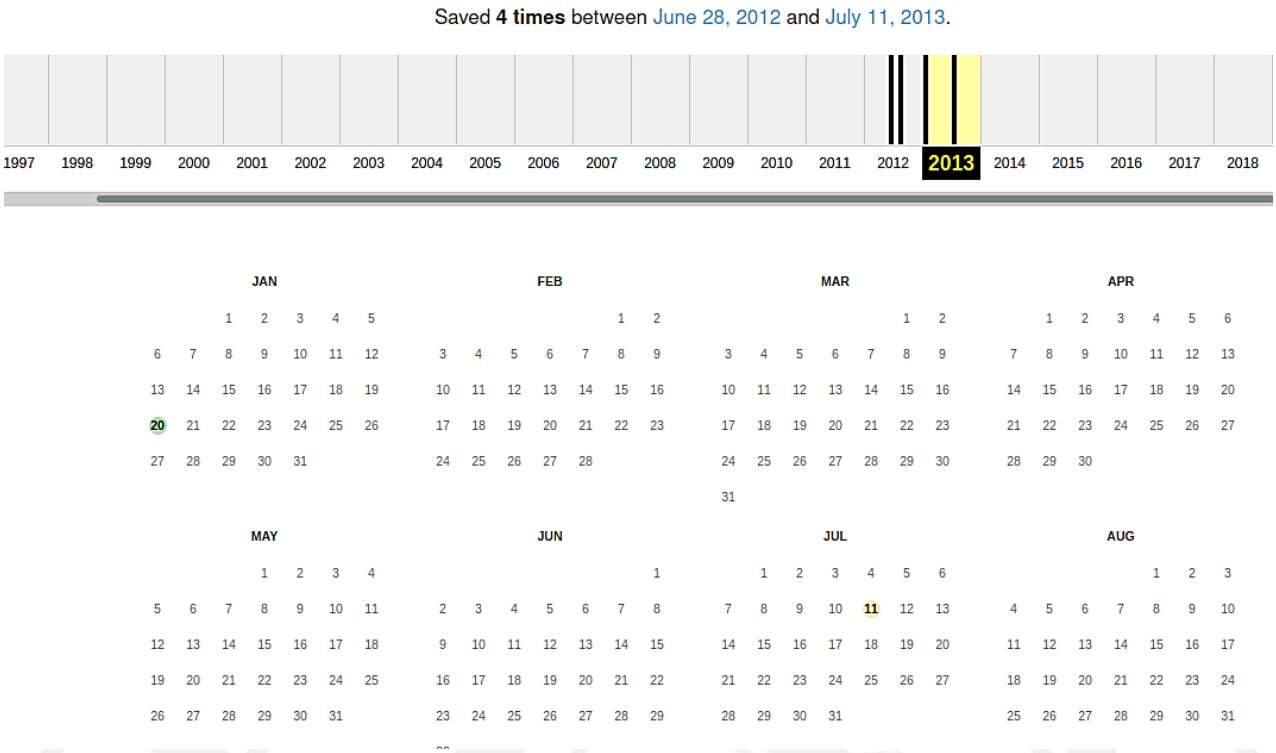
- La tabla de URL se puede procesar con código para extraer todas las URL, o las **URL de un determinado tipo** que pueden ser interesantes para un determinado propósito. Por ejemplo, puedes extraer imágenes o documentos para inspeccionar su contenido (¡o metadatos! 😉 😉 . Esto también se puede hacer en la página web *de Wayback Machine,* ya que tiene un cuadro de texto de filtrado.

100,000 URLs have been captured for this domain.

Filter results (i.e. '.txt'): pdf

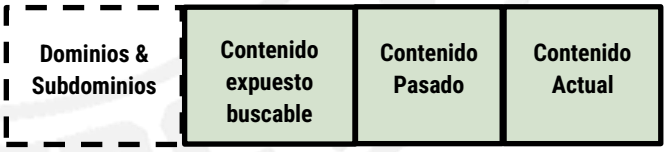
URL	MIME TYPE	FROM	TO	CAPTURES	DUPLICATES	UNIQUES
http://www.uniovi.es/-/defecto.pdf	text/html	Jun 16, 2013	Jul 15, 2013	2	0	2
http://www.uniovi.es/-/file.pdf	text/html	Jun 16, 2013	Jul 15, 2013	2	0	2
http://www.uniovi.es/-/preistabelle.pdf	text/html	Jun 16, 2013	Jul 15, 2013	2	0	2
http://www.uniovi.es/-/presitabelle.pdf	text/html	Jun 16, 2013	Jul 15, 2013	2	0	2
http://www.uniovi.es/aal/archivos_pdf/neutro_materia.pdf	text/html	Jun 6, 2011	Dec 15, 2018	7	6	1
http://www.uniovi.es/accesoyayudas/estudios/373/Oficio+de+Adjunto+Plantilla+Solicitud.pdf/5412a9a3-9730-40a4-8	text/plain	Sep 4, 2014	Sep 4, 2014	1	0	1
http://www.uniovi.es/accesoyayudas/tramites/tramite/-/asset_publisher/v9UAvM9qJpFc/content/wwww.uniovi.es/documents/31582/243104/INSTANCIA+VICERRECTOR+ESTUDIANTES_131028.pdf/d7cdc975-d56c-404f-b5a0-ef50be637791	text/html	Sep 8, 2014	Sep 8, 2014	1	0	1
http://www.uniovi.es/Alfonso_Garcia_Leal/1993478.pdf	text/html	Jan 19, 2012	Apr 12, 2012	2	1	1
http://www.uniovi.es/Areas/Mecanica.Fluidos/becasempleo/investigacionaerodinamica_paniaguaSMALL.pdf	unk	Nov 11, 2014	Nov 11, 2014	1	0	1
http://www.uniovi.es/Areas/Mecanica.Fluidos/docencia/_asignaturas/maquinas_de_fluidos/Lecc6_r1.pdf	unk	Apr 12, 2017	Apr 12, 2017	1	0	1
http://www.uniovi.es/Areas/Mecanica.Fluidos/docencia/_asignaturas/maquinas_de_fluidos/Presenta_Leccion3.pdf	unk	Apr 12, 2017	Apr 12, 2017	1	0	1
http://www.uniovi.es/Areas/Mecanica.Fluidos/docencia/_asignaturas/mecanica_de_fluidos/05_06/8.%20FLUJO_CONDUCTOS.pdf	unk	Jul 29, 2015	Jul 29, 2015	1	0	1

- También puedes obtener el contenido de **archivos interesantes** de una página web. Por ejemplo, el archivo `robots.txt`
- **Historial de cualquier archivo del sitio web**: La columna de la tabla "URL" muestra el número de copias diferentes que cualquier archivo ha tenido a lo largo del tiempo. Esto significa que podemos localizar cualquier versión de cualquier archivo que haya sido indexado. Puedes iterar a través de ellos en la vista de calendario una vez que hagas clic en el archivo



Resultados esperados: **Esta actividad se completará cuando puedas usar la *Wayback Machine* para localizar la lista de URL históricas de cualquier sitio web que elijas, cualquier archivo interesante que prefieras y las diferentes versiones históricas de cualquier documento o página web que estuviera en el sitio. Una vez que pueda hacer eso, también debes pensar en lo que esto podría significar desde el punto de vista de la seguridad (las preguntas al final del laboratorio pueden ayudarte a analizarlo).**

BLOQUE 3: LOS DNS AL SOL 😊



Descubrimiento de subdominios

Aplicación práctica: *Necesitas saber si un dominio concreto tiene subdominios registrados*

Los nombres de dominio son muy curiosos porque normalmente la mayoría de la gente conoce el principal pero, la mayoría de las veces, hay subdominios secundarios que no son tan populares, solo conocidos por algunas personas, o simplemente olvidados. En cualquier caso, suponen más vectores de enumeración, y ahí reside el potencial de descubrir más cosas sobre un objetivo. Hay muchas maneras de descubrir esta información automáticamente. Estos son algunos de ellos:

- `knockpy` (https://github.com/guelfoweb/knock). *Herramienta Python* diseñada para enumerar subdominios en un dominio de destino a través de una lista de palabras. En *Ubuntu* puedes instalarlo con `sudo apt install knockpy` y ejecutar un escaneo básico con `./knockpy <objetivo>`. En el `README` de su repositorio puedes encontrar instrucciones de uso detalladas.
- `dnsenum` (https://github.com/fwaeytens/dnsenum): Por ejemplo. `dnsenum uniovi.es`. Si no se proporciona un parámetro `-f`, de forma predeterminada será `/usr/share/dnsenum/dns.txt` o un archivo `dns.txt` en el mismo directorio que `dnsenum.pl`

NOTA: Algunos estudiantes han informado que `dnsenum` puede congelar sus laboratorios cuando se arranca. Si es tu caso, cancela rápidamente la operación (CRTL+C) y usa otra herramienta. Esto se debe a un problema de instalación de *VirtualBox*, pero no hemos podido averiguar su origen exacto, ya que no os pasa a todos.

dnsenum 1.2.6 Cheatsheet (ingenieriainformatica.uniovi.es)
multithreaded perl script to enumerate DNS information of a domain
https://github.com/fwaeytens/dnsenum

GENERAL USAGE

```
dnsenum [Options] <domain>
```

NOTES

If no -f tag supplied will default to /usr/share/dnsenum/dns.txt or the dns.txt file in the same directory as dnsenum.pl

OPTIONS

GENERAL OPTIONS

- **--dnsserver <server>**: Use this DNS server for A, NS and MX queries.
- **--enum**: Shortcut option equivalent to --threads 5 -s 15 -w.
- **-h, --help**: Print this help message.
- **--nocolor**: Disable ANSIColor output.
- **--noreverse**: Skip the reverse lookup operations.
- **--private**: Show and save private ips at the end of the file domain_ips.txt.
- **--subfile <file>**: Write all valid subdomains to this file.
- **-t, --timeout <value>**: The tcp and udp timeout values in seconds (default: 10s).
- **--threads <value>**: The number of threads that will perform different queries.
- **-v, --verbose**: Be verbose: show all the progress and all the error messages.

GOOGLE SCRAPING OPTIONS

- **-p, --pages <value>**: The number of google search pages to process when scraping names, the default is 5 pages, the -s switch must be specified.
- **-s, --scrap <value>**: The maximum number of subdomains that will be scraped from Google (default 15).

BRUTE FORCE OPTIONS

- **-f, --file <file>**: Read subdomains from this file to perform brute force. (Takes priority over default dns.txt)
- **-r, --recursion**: Recursion on subdomains, brute force all discovered subdomains that have an NS record.
- **-u, --update <a|g|r|z>**: Update the file specified with the -f switch with valid subdomains.
  - a (all) — Update using all results.
  - g — Update using only google scraping results.
  - r — Update using only reverse lookup results.
  - z — Update using only zonetransfer results.

```
operario@kali:~$ dnsenum uniovi.es
dnsenum VERSION:1.2.6

-----   uniovi.es   -----

Host's addresses:
__________________

uniovi.es.                               1216     IN    A        156.35.233.105

Name Servers:
_____________

enol.si.uniovi.es.                       172800   IN    A        156.35.14.2
chico.rediris.es.                        6186     IN    A        162.219.54.2
zeus.etsimo.uniovi.es.                   1800     IN    A        156.35.23.24
sun.rediris.es.                          3781     IN    A        199.184.182.1
solid.net.uniovi.es.                     1800     IN    A        156.35.11.170

Mail (MX) Servers:
__________________

mx02.puc.rediris.es.                     30       IN    A        130.206.19.162
mx02.puc.rediris.es.                     30       IN    A        130.206.19.130
mx01.puc.rediris.es.                     30       IN    A        130.206.19.162
mx01.puc.rediris.es.                     30       IN    A        130.206.19.130

Trying Zone Transfers and getting Bind Versions:
________________________________________________


Trying Zone Transfer for uniovi.es on enol.si.uniovi.es ...
AXFR record query failed: REFUSED

Trying Zone Transfer for uniovi.es on chico.rediris.es ...
AXFR record query failed: REFUSED
```

WHOIS NETRANGE OPTIONS:

- **-d, --delay <value>**: The maximum value of seconds to wait between whois queries, the value is defined randomly, default: 3s.
- **-w, --whois**: Perform the whois queries on c class network ranges. **Warning**: this can generate very large netranges and it will take lot of time to perform reverse lookups.

REVERSE LOOKUP OPTIONS:

- **-e, --exclude <regexp>**: Exclude PTR records that match the regexp expression from reverse lookup results, useful on invalid hostnames.

OUTPUT OPTIONS:

- **-o --output <file>**: Output in XML format. Can be imported in MagicTree (www.gremwell.com)

- https://dnsdumpster.com/: Sitio web que nos permite localizar subdominios de un dominio determinado.
- https://searchdns.netcraft.com/: Otra web que nos permite localizar subdominios de un dominio determinado.
- https://www.virustotal.com: Si vamos a "Buscar", damos un nombre DNS (por ejemplo, uniovi.es) y realizamos un escaneo de detección de URLs como el que vimos anteriormente, la pestaña "Detalles" también muestra sus subdominios asociados.
- https://crt.sh/: Darle un nombre de dominio (por ejemplo, uniovi.es) también nos da subdominios asociados.

Por último, es posible que algunos de estos dominios puedan estar, como decíamos, **abandonados.** En este caso tenemos dos opciones:

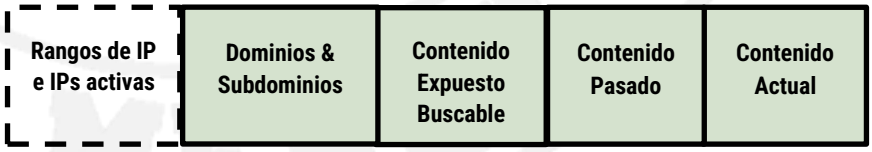
- Un dominio **verdaderamente abandonado** (sin contenido)
- Un **dominio no mantenido** (contenidos potencialmente vulnerables, versiones antiguas de los servicios, sin parchear, sin fijar...).

Si tenemos muchos dominios para analizar, distinguir los verdaderamente vacíos de los que tienen contenidos puede llevar mucho tiempo. Podemos ahorrar tiempo si inspeccionamos visualmente los contenidos de cualquiera de los dominios que encontramos para ver a qué tipo de contenidos sirven. La herramienta `eyewitness` (https://github.com/ChrisTruncer/EyeWitness) acepta un archivo con una lista de subdominios, y devuelve un informe con capturas de pantalla de cada subdominio. Por ejemplo, `eyewitness.py -f subdominios.txt`. No obstante, no la usaremos en este laboratorio.

Estas no son las únicas herramientas que pueden hacer este trabajo, pero cubren la mayoría de los casos de uso. `Knockpy` y `dnsenum` requieren instalación, pero están disponibles en los paquetes estándar de *Kali Linux*. Puedes encontrarlos instalados en el contenedor *Kali* de la infraestructura del **Laboratorio 2**.

Resultados esperados: Esta actividad se completará cuando puedas enumerar todos los subdominios de `uniovi.es` con cualquiera de las herramientas proporcionadas basadas en web. Las herramientas de línea de comandos deben usarse en el dominio incluido en la infraestructura de este laboratorio, `lab02.ssi.es`. Recomendamos utilizar al menos una de las herramientas de línea de comandos y uno de los sitios web proporcionados para obtener una mejor comprensión de cómo realizar esta actividad. NO ES NECESARIO probar todas las herramientas mencionadas.

BLOQUE 4: INSPECCIÓN DE IPS



Analizar un rango de IPs

Aplicación práctica: *Necesitas saber el rango de IPs asignado a cualquier dominio de Internet*

Una vez que conocemos los dominios DNS y subdominios de cualquier organización, podemos buscar en sus rangos de IP públicas asignadas para saber cuáles son las posibles IPs a inspeccionar, en busca de máquinas o servicios expuestos pertenecientes a esta organización. Para sitios `.com`, podemos consultar https://whois.arin.net. Para ello, podemos introducir el nombre de la organización (por ejemplo, "Microsoft", "Yahoo") en el cuadro de texto *SEARCH WHOIS-RWS* y desplazarnos hacia abajo para ver los rangos de IP que esta organización tiene asignados.

Para dominios .es podemos utilizar www.nic.es para el mismo fin. Pulsando en los detalles de un dominio (*"Ver datos")* nos muestran los servidores DNS que sirven a las diferentes redes asignadas a la organización, de las que podemos extraer fácilmente IPs y rangos IP.

Resultados esperados: Esta actividad se completará cuando pueda verificar el rango de IP asignado a uniovi.es o cualquier organización .com que desee.

Escaneos globales con *zmap*

Aplicación práctica: *Puedes hacer escaneos de servicios en un puerto concreto en todo Internet*

`zmap` (https://zmap.io/) es un escáner de red rápido *single-packet* optimizado para escaneos de red en toda Internet. En un ordenador con conexión gigabit, *ZMap* puede escanear todo el espacio de **direcciones IPv4 público en menos de 45 minutos.** Conexiones más rápidas significan menos tiempo de escaneo.

`Zmap` no se utilizará para escanear todo Internet, ya que es excesivo y no tiene un propósito real en este laboratorio. Sin embargo, se puede utilizar para encontrar servicios escuchando en un puerto concreto de forma rápida y sencilla en una red o rango de IPs conocido (que se puede obtener de la actividad anterior). Por ejemplo, si sabemos que la red `88.151.16.0/24` pertenece al dominio *asturias.es* de la actividad anterior, es muy fácil localizar servidores web activos en esta red (HTTP, puerto 80) con `zmap -p80 88.151.16.0/24`. Como la salida puede ser grande, se recomienda almacenarla en un archivo con la opción `-o`.

El siguiente *cheatsheet* hace una descripción general de todas las opciones de `zmap`. La **infraestructura del Laboratorio 2** tiene un *Kali Linux* con `zmap` instalado. Puedes instalar `zmap` en *Kali* y *Ubuntu* con `sudo apt install zmap`.

Resultados esperados: Esta actividad se completará cuando puedas comprobar los servidores web activos (o cualquier otro servicio) en una red, por ejemplo, *asturias.es* o *uniovi.es* (por defecto Zmap no funciona en redes locales).

Zmap 2.1.1 Cheatsheet (ingenieriainformatica.uniovi.es)

Fast internet-wide scanner

https://zmap.io/

```
operario@kali:~$ sudo zmap -p 80 88.151.16.0/24 -o asturias.txt
Oct 29 18:33:29.149 [WARN] blacklist: ZMap is currently using the default blacklist located at /etc/zmap/blacklist.
conf. By default, this blacklist excludes locally scoped networks (e.g. 10.0.0.0/8, 127.0.0.1/8, and 192.168.0.0/16
). If you are trying to scan local networks, you can change the default blacklist by editing the default ZMap confi
guration at /etc/zmap/zmap.conf.
Oct 29 18:33:29.153 [INFO] zmap: output module: csv
 0:00 0%; send: 10 0 p/s (261 p/s avg); recv: 0 0 p/s (0 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 0.00%
 0:01 13%; send: 256 done (6.23 Kp/s avg); recv: 17 16 p/s (16 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 6.64%
 0:02 26%; send: 256 done (6.23 Kp/s avg); recv: 76 56 p/s (36 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 29.69%
 0:03 38%; send: 256 done (6.23 Kp/s avg); recv: 76 0 p/s (24 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 29.69%
 0:04 51%; send: 256 done (6.23 Kp/s avg); recv: 76 0 p/s (18 p/s avg); drops: 0 p/s (0 p/s avg); hitrate: 29.69%
 0:05 64% (3s left); send: 256 done (6.23 Kp/s avg); recv: 76 0 p/s (14 p/s avg); drops: 0 p/s (0 p/s avg); hitrate
: 29.69%
 0:06 77% (2s left); send: 256 done (6.23 Kp/s avg); recv: 76 0 p/s (12 p/s avg); drops: 0 p/s (0 p/s avg); hitrate
: 29.69%
 0:07 89% (1s left); send: 256 done (6.23 Kp/s avg); recv: 76 0 p/s (10 p/s avg); drops: 0 p/s (0 p/s avg); hitrate
: 29.69%
Oct 29 18:33:37.339 [INFO] zmap: completed
```

GENERAL USAGE

```
zmap [OPTION]... [SUBNETS]…
```

NOTES

Probe-module (tcp_synscan): Probe module that sends a TCP SYN packet to a specific port. Possible classifications are: synack and rst. A SYN-ACK packet is considered a success and a reset packet is considered a failed response.
Output-module (csv): By default, ZMap prints out unique, successfulIP addresses (e.g., SYN-ACK from a TCP SYN scan) in ASCII form (e.g., 192.168.1.5) to stdout or the specified output file. Internally this is handled by the "csv" output module and is
equivalent to running zmap --output-module=csv --output-fields=saddr --output-filter="success = 1 && repeat = 0".

OPTIONS

BASIC ARGUMENTS

- **-b, --blacklist-file=path**: File of subnets to exclude, in CIDR notation, e.g. 192.168.0.0/16
- **-o, --output-file=name**: Output file
- **-p, --target-port=port**: port number to scan (for TCP and UDP scans)
- **-w, --whitelist-file=path**: File of subnets to constrain scan to, in CIDR notation, e.g. 192.168.0.0/16

SCAN OPTIONS

- **--retries=n**: Max number of times to try to send packet if send fails (default=`10')
- **--shard=n**: Set which shard this scan is (0 indexed) (default=`0')
- **--shards=N**: Set the total number of shards (default=`1')
- **-B, --bandwidth=bps**: Set send rate in bits/second (supports suffixes G, M and K)
- **-c, --cooldown-time=secs**: How long to continue receiving after sending last probe (default=`8')
- **-d, --dryrun**: Don't actually send packets
- **-e, --seed=n**: Seed used to select address permutation
- **-N, --max-results=n**: Cap number of results to return
- **-n, --max-targets=n**: Cap number of targets to probe (as a number or a percentage of the address space)
- **-P, --probes=n**: Number of probes to send to each IP (default=`1')
- **-r, --rate=pps**: Set send rate in packets/sec
- **-t, --max-runtime=ses**: Cap length of time for sending packets

ADDITIONAL OPTIONS

- **--cores=STRING**: Comma-separated list of cores to pin to
- **--ignore-invalid-hosts**: Ignore invalid hosts in whitelist/blacklist file
- **--max-sendto-failures=n**: Maximum NIC sendto failures before scan is aborted (default=`-1')
- **--min-hitrate=n**: Minimum hitrate that scan can hit before scan is aborted (default=`0.0')
- **-C, --config=filename**: Read a configuration file, which can specify any of these options (default=`/etc/zmap/zmap.conf')
- **-h, --help**: Print help and exit
- **-T, --sender-threads=n**: Threads used to send packets (default=`1')
- **-V, --version**: Print version and exit

EXAMPLES

zmap -p 80 (scan the Internet for hosts on tcp/80 and output to stdout)
zmap -N 5 -B 10M -p 80 (find 5 HTTP servers, scanning at 10 Mb/s)
zmap -p 80 10.0.0.0/8 192.168.0.0/16 -o (scan both subnets on tcp/80)
zmap -p 80 1.2.3.4 10.0.0.3 (scan 1.2.3.4, 10.0.0.3 on tcp/80)

NETWORK OPTIONS

- **--source-mac=addr**: Source MAC address
- **-G, --gateway-mac=addr**: Specify gateway MAC address
- **-i, --interface=name**: Specify network interface to use
- **-S, --source-ip=ip|range**: Source address(es) for scan packets
- **-s, --source-port=port|range**: Source port(s) for scan packets
- **-X, --vpn**: Sends IP packets instead of Ethernet (for VPNs)

PROBE MODULES

- **--list-probe-modules**: List available probe modules
- **--probe-args=args**: Arguments to pass to probe module
- **-M, --probe-module=name**: Select probe module (default=`tcp_synscan')

DATA OUTPUT

- **--list-output-fields**: List all fields that can be output by selected probe module
- **--list-output-modules**: List available output modules
- **--output-args=args**: Arguments to pass to output module
- **--output-filter=filter**: Specify a filter over the response fields to limit what responses get sent to the output module
- **-f, --output-fields=fields**: Fields that should be output in result set
- **-O, --output-module=name**: Select output module (default=`default')

LOGGING AND METADATA

- **--disable-syslog**: Disables logging messages to syslog
- **--notes=notes**: Inject user-specified notes into scan metadata
- **--user-metadata=json**: Inject user-specified JSON metadata into scan metadata
- **-L, --log-directory=directory**: Write log entries to a timestamped file in this directory
- **-l, --log-file=name**: Write log entries to file
- **-m, --metadata-file=name**: Output file for scan metadata (JSON)
- **-q, --quiet**: Do not print status updates
- **-u, --status-updates-file=name**: Write scan progress updates to CSV file
- **-v, --verbosity=n**: Level of log detail (0-5) (default=`3')

Escaneos en LAN con *nmap*

Aplicación práctica: *Necesitas inspeccionar una red de área local (LAN) en busca de máquinas "vivas"*

El propósito de esta actividad es localizar todas las máquinas activas en una red LAN (¡no en Internet!). Para ello la **infraestructura del laboratorio 2** tiene una red LAN construida. Esto extiende la actividad del laboratorio 1, que solo determina si un servidor está "vivo". Usa un *list scan* de este *cheatsheet* para ver si también se obtienen los nombres DNS de los hosts vivos.

Nmap 7.80 Cheatsheet series (ingenieriainformatica.uniovi.es)

Part 1: Reconnaissance (Basic)

https://nmap.org/

GENERAL USAGE

```
nmap [Scan Type(s)] [Options] {target specification}
```

NOTES

Target specifications can be host names, IP addresses, ranges, networks, etc. (scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254)

Use these options to locate "alive machines" (sometimes only returns that, sometimes they also return some port / service information)

TARGET SPECIFICATION

- **-iL <inputfilename>**: Input from from file a list of hosts/networks
- **-iR <num hosts>**: Choose random targets
- **--exclude <host1[,host2][,host3],...>**: Exclude hosts/networks
- **--excludefile <exclude_file>**: Exclude list from file

RECONOISSANCE EXAMPLES

- *nmap -sn scanme.nmap.org*
- *sudo nmap -PS22,80 scanme.nmap.org*
- *sudo nmap --traceroute scanme.nmap.org*

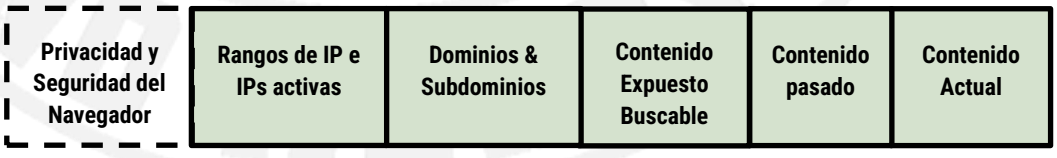
```
operario@kali:~$ nmap -sn 192.168.20.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-21 18:38 CEST
Nmap scan report for 192.168.20.10
Host is up (0.00085s latency).
Nmap done: 1 IP address (1 host up) scanned in 13.01 seconds
operario@kali:~$ sudo nmap -PS22,80 192.168.20.10
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-21 18:42 CEST
Nmap scan report for 192.168.20.10
Host is up (0.00012s latency).
Not shown: 999 closed ports
PORT   STATE SERVICE
80/tcp open  http
MAC Address: 08:00:27:67:7A:EF (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 13.30 seconds
```

HOST DISCOVERY OPTIONS (WAYS TO CHECK "ALIVE" MACHINES)

- **--dns-servers <serv1[,serv2],...>**: Specify custom DNS servers
- **-n/-R**: Never do DNS resolution/Always resolve [default: sometimes]
- **-PE/PP/PM**: ICMP echo, timestamp, and netmask request discovery probes
- **-Pn**: Treat all provided hosts as online -- skip host discovery
- **-PO[protocol list]**: IP Protocol Ping
- **-PS/PA/PU/PY[portlist]**: TCP SYN/ACK, UDP or SCTP discovery to given ports
- **-sL**: List Scan - simply list targets to scan
- **-sn**: Ping Scan - disable port scan
- **--system-dns**: Use OS's DNS resolver
- **--traceroute**: Trace hop path to each host

Resultados esperados: Esta actividad se completará cuando puedas localizar hosts activos en la LAN mencionada y sus nombres DNS.

BLOQUE 5. ES ALGO PERSONAL: PRIVACIDAD, SEGURIDAD DE NAVEGACIÓN



Blacklight: Inspector de privacidad de la web

Aplicación práctica: *Necesitas saber cómo cualquier sitio de Internet usa tus datos de navegación para su propio provecho*

Blacklight (https://themarkup.org/blacklight) escanea y descubre tecnologías específicas de seguimiento de usuarios (*tracking*) de cualquier sitio web. Esto significa que revela quién está obteniendo sus datos al navegar por un sitio y, por tanto, quien los usará para luego servirte anuncios basados (supuestamente) en tus gustos.

Resultados esperados: Esta actividad se completará cuando puedas usar *Blacklight* para inspeccionar cómo cualquier sitio que elijas está recopilando tus datos y analizar los resultados.

Crear un entorno de navegación seguro y verdaderamente privado en tu máquina virtual

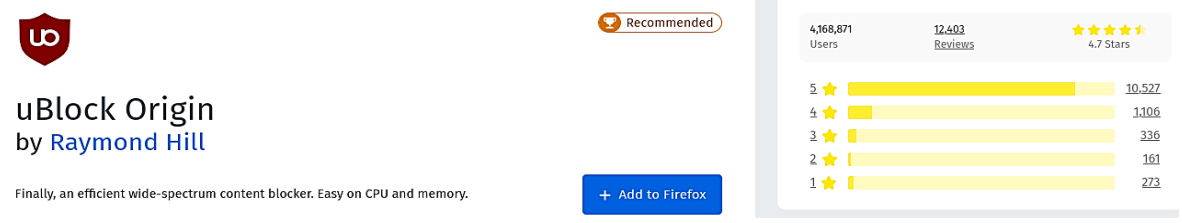
Aplicación práctica: *Puedes crear un entorno de navegación web más seguro para tu día a día*

*(**NOTA:** Seguro no significa privado. Tus intentos de conexión seguirán siendo registrados por los servidores de destino, y también registrarán tu IP. Para lograr un cierto grado de* privacidad de *navegación necesitas otras herramientas (como ToR), pero están más allá del alcance de la asignatura)*

El propósito de esta sección es luchar contra las técnicas que has descubierto con *Blacklight,* ya que la cantidad de datos recopilados por un sitio web puede ser problemática en ciertos escenarios de uso. Aparte de eso, es muy útil tener un entorno de navegación más seguro en caso de que accidentalmente navegues por una página web maliciosa o troyanizada, con anuncios maliciosos u otras amenazas web. De esta manera, puedes navegar de una manera más segura **si usas los siguientes complementos del navegador**. Al combinarlo con usar una "máquina virtual para navegación por Internet", obtienes un entorno de navegación notablemente más seguro (especialmente si es *Linux*). Usaremos *Firefox* como ejemplo de navegador, pero los mismos complementos también están disponibles en otros navegadores (también en versiones móviles). Esto también conecta con el tema de introducción de la teoría.

Para tratar con extensiones de navegador, debes considerar las cosas que se están permitiendo:

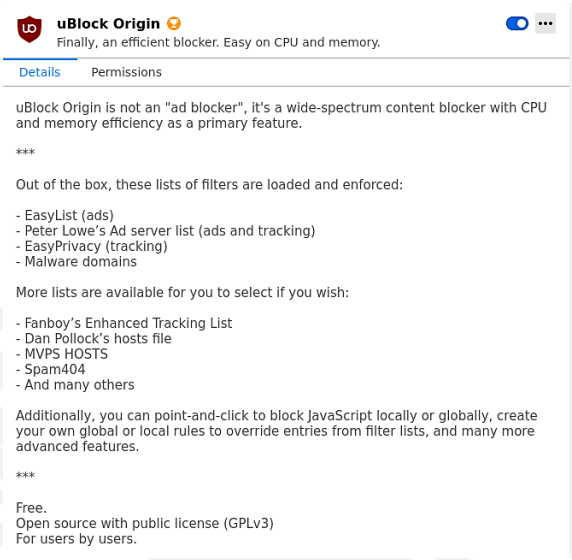
- **Hay extensiones maliciosas**: para evitar problemas, instala solo las *recomendadas* (inspeccionadas por el fabricante del navegador), con alta puntuación y muchas recomendaciones de usuario.



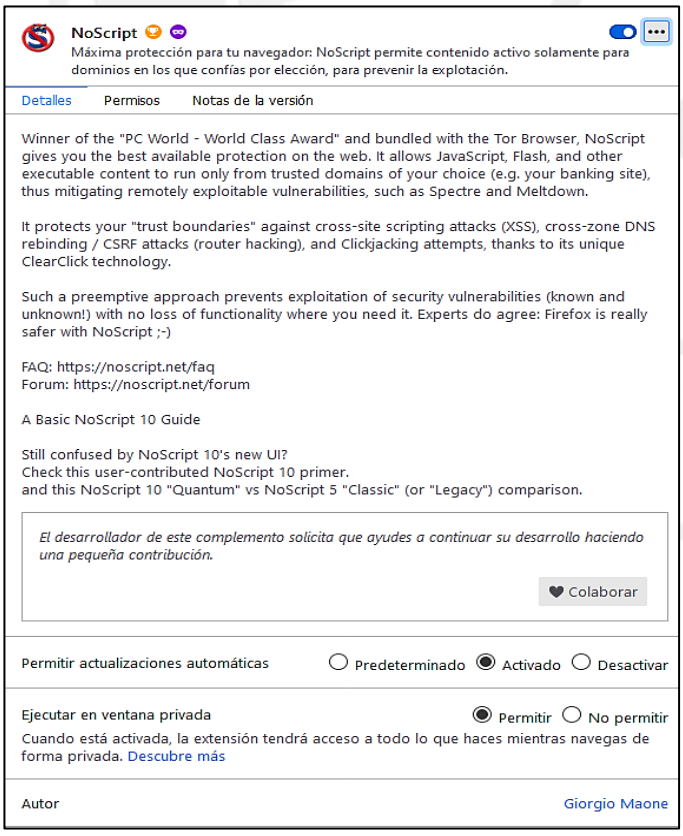
- Ten cuidado con las extensiones que piden demasiados permisos o permisos no razonables.
- Las extensiones normalmente se instalan utilizando una opción específica del navegador *(Complementos)* o un *market* (*Google Chrome Store),* no de una web externa.

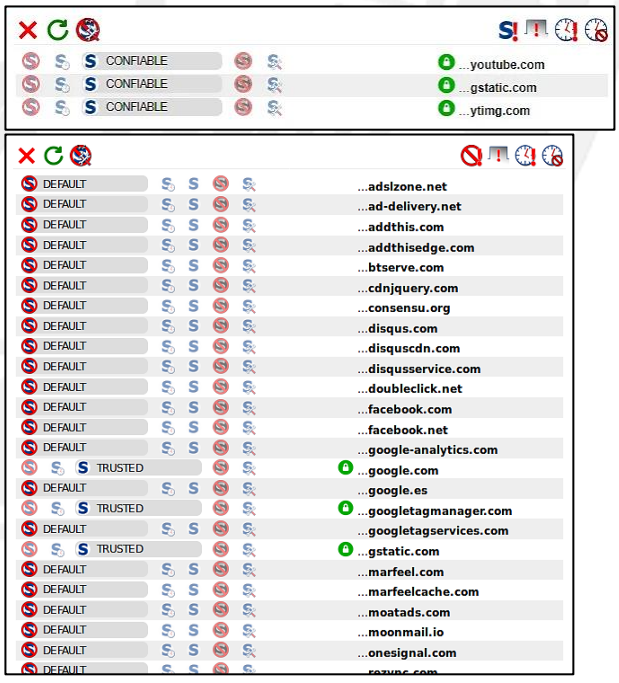
Una vez dicho esto, se recomienda instalar extensiones que se ocupen de:

- **Anuncios:** ya que pueden ser molestos y/o maliciosos, incluso siendo una fuente de ingresos legal para los propietarios de páginas. *uBlock Origin* es el complemento recomendado.

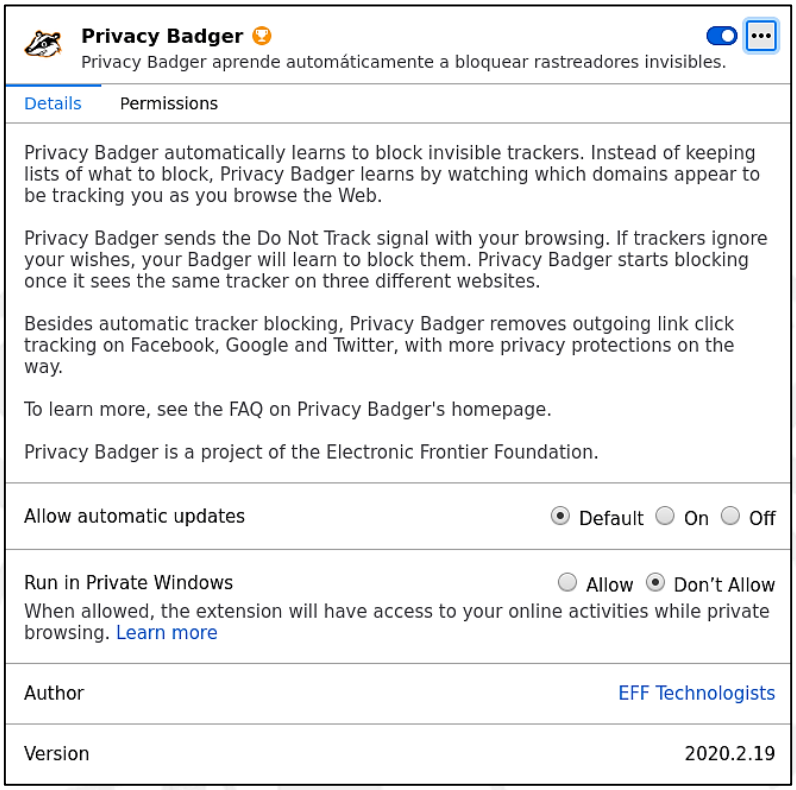


- **Bloqueadores de *scripts***: Usar *JavaScript* en páginas web es común y legítimo, pero a veces los *scripts* se usan con fines maliciosos. Para evitar esto, se recomienda la extensión *NoScript*. No solo bloquea *scripts*, sino que también identifica el uso de *scripts* maliciosos en páginas web. Sin embargo, su uso requiere "entrenamiento", ya que inicialmente bloquea la mayoría de los *scripts* en una página web, quedando muchas veces inutilizable (o haciendo que su contenido sea invisible). Sin embargo, puedes desbloquear origines de scripts hasta que la página sea lo suficientemente utilizable para lo que quieres hacer. ¡No desbloquees anuncios o dominios maliciosos obvios! Normalmente, debes comenzar con orígenes de *scripts* que tengan el mismo nombre DNS que la página que estás viendo. Este tipo de "entrenamiento" se almacena entre sesiones, por lo que solo necesitas hacerlo una vez por página web.





- **Privacidad:** *Privacy Badger* es la extensión recomendada, ya que evita que las páginas web rastreen tu historial de navegación y creen un perfil de tus comportamientos de navegación. También bloquea redes sociales populares, los intentos de seguimiento de sitios web *(Facebook, Google, Twitter...)* e implementa otras medidas que protegen la privacidad del usuario.



Resultados esperados: Esta actividad se completará cuando tu navegador seguro esté operativo y puedas navegar a páginas web con todos los complementos mencionados activos.

Have I been pwnd?

Aplicación práctica: *Necesitas saber si la clave de cualquier de tus cuentas de un servicio de Internet ha sido potencialmente filtrada en alguno de las muchas filtraciones de datos que ocurren hoy día*

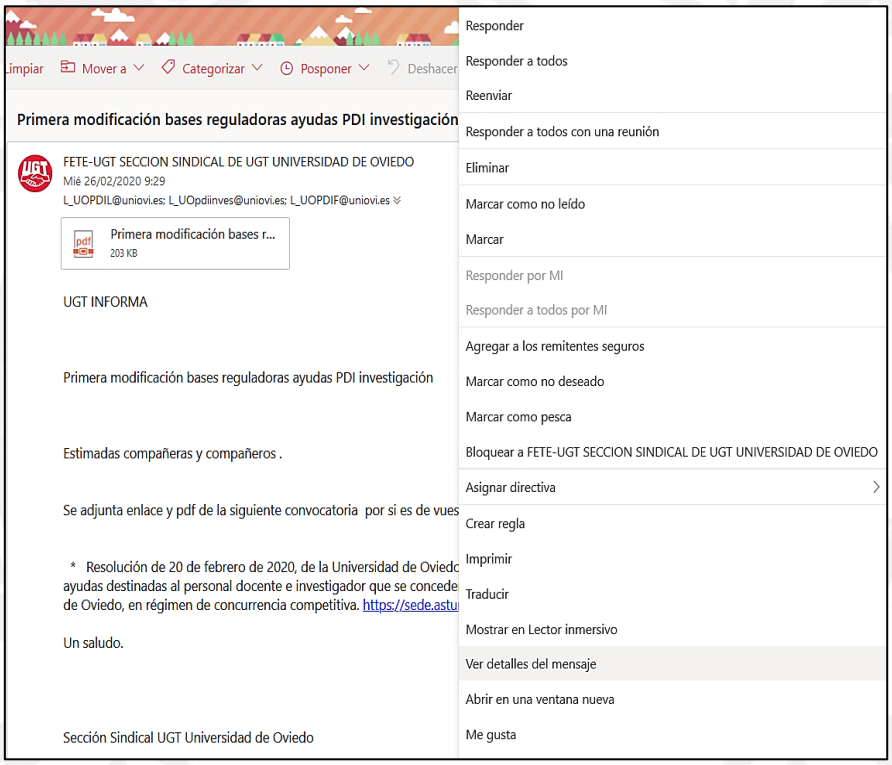
Have I been pwnd? (https://haveibeenpwned.com/) es un sitio web que **almacena y permite consultar fugas de datos de usuarios**. Acepta nombres de cuentas de correo electrónico, e indica si este nombre de cuenta se ha filtrado en alguna de las bases de datos de usuarios conocidas que han sido robadas. Por tanto, si la cuenta de correo electrónico consultada está dentro de una de las múltiples bases de datos de usuarios robadas que maneja, te indicará el sitio cuyos usuarios han sido robados, cuándo y otros detalles sobre la fuga de datos asociada. Debido a la probabilidad de usar la misma contraseña y correo electrónico en varios sitios, es muy importante verificar esto para evitar que un robo en un sitio facilite también intrusiones en otros sitios.

Resultados esperados: Esta actividad se completará cuando inspecciones cualquier cuenta de correo electrónico que quieras saber si es parte de una fuga de datos conocida.

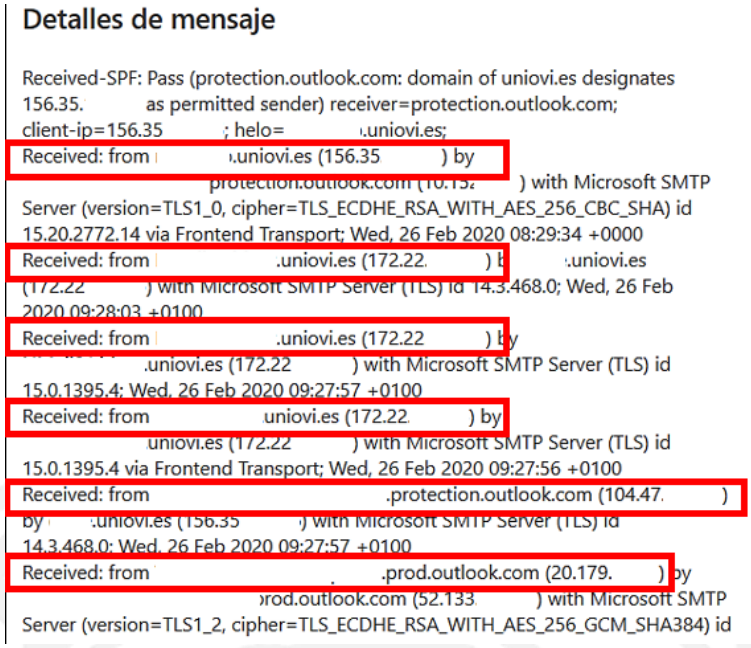
Geolocalización de un remitente de correo electrónico (o de cualquier IP)

Aplicación práctica: *Necesitas geolocalizar cualquier IP, incluidas las que forman parte de las cabeceras de un email*

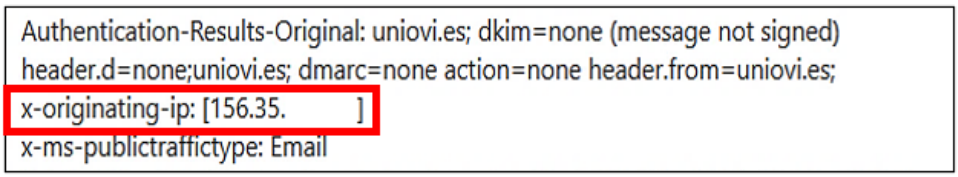
Geolocalizar una dirección IP es posible, aunque la precisión que podemos obtener (sin tener permisos especiales dados por un juez) se limita a áreas de código de distrito, no a coordenadas particulares. Cualquier IP puede ser geolocalizada utilizando servicios online como https://www.iplocation.net/. Esto nos da unas coordenadas que luego pueden ser geolocalizadas en aplicaciones populares como *Google Earth*, *Google Maps*, o similares. Una aplicación especial de esta técnica se puede hacer con correos electrónicos. Algunos proveedores de correo electrónico **conservan todos los encabezados del correo electrónico** en los mensajes que administran. Puedes ver los encabezados de correo electrónico utilizando las opciones avanzadas del cliente de correo electrónico. En el *webmail* de Uniovi la opción es esta:



Hay muchos encabezados con diferentes significados, pero no son importantes para nuestros fines. Un encabezado especial es `Received: from`, que indica los servidores de correo electrónico que han procesado el correo electrónico. Por lo general, un correo electrónico "rebota" a través de múltiples servidores de correo electrónico hasta llegar a su destino, y puedes geolocalizar estos servidores utilizando las IPs en estos encabezados (si no se eliminan).



Sin embargo, el encabezado interesante no es este, ya que solo obtendrás la geolocalización de centros de datos de servidores y no la del origen real del correo electrónico. El importante es `x-originating-ip` que, de nuevo, dependiendo del proveedor de correo electrónico, podría haberse eliminado. Sin embargo, si está presente, nos proporcionará las coordenadas aproximadas (nivel de distrito) del remitente del correo electrónico.



Resultados esperados: Esta actividad se completará cuando seas capaz de saber la geolocalización de cualquier IP y, más concretamente, de la IP del autor de un correo electrónico que recibiste, si esta información se proporciona en los encabezados de correo electrónico.

INSIGNIAS Y AUTOEVALUACIÓN

NOTA: Tienes una versión de esta tabla de insignias en formato editable disponible en el *Campus Virtual*. Puedes usar este archivo para crear un documento en el formato que desees y tomar notas extendidas de tus actividades para crear un log de lo que has hecho. Recuerda que el material que elabores se puede llevar a los exámenes de laboratorio.

Nivel de insignia	Desbloqueado cuando	¿Desbloqueado?
	Puedes encontrar y analizar cualquier archivo `robots.txt`	
	Responde a esta pregunta: *¿Cuáles crees que serán los problemas de seguridad que un `robots.txt` mal escrito podría provocar?*	
	Puedes analizar los metadatos de cualquier archivo individual de un tipo adecuado que encuentres. Puedes responder a esta pregunta: *¿Por qué los metadatos pueden ser una amenaza para la seguridad?*	
	Puedes analizar una única URL o IP utilizando *Shodan* e interpretar los resultados. También puedes responder a esta pregunta: *¿qué parte de los resultados enlaza con algo que vimos en el laboratorio anterior, y puede indicar un problema grave con la página web?*	
	Puedes buscar todas las direcciones URL históricas de un dominio.	
	Puedes responder a esta pregunta: *¿qué problemas de seguridad pueden aparecer si tenemos un archivo histórico de los diferentes contenidos de un archivo `robots.txt`?*	
	Puedes responder a esta pregunta: *¿por qué los subdominios son interesantes desde el punto de vista de la seguridad? ¿Qué podría suceder si un subdominio (y su contenido) ha sido olvidado durante mucho tiempo?*	
	Puedes analizar cómo un sitio web recopila datos de usuario y trata la privacidad mediante *Blacklight*.	
	Puedes comprobar si una contraseña ha sido potencialmente comprometida en una fuga de datos y responder a esta pregunta: *¿qué debes hacer si encuentras tu email en una de estas intrusiones documentadas?*	
	Puedes utilizar *Censys* para analizar los hosts de una red e interpretar los resultados, tanto de una red como de cualquier host individual. También puedes responder a esta pregunta: *¿puedes encontrar los servicios que se ejecutan en los hosts? ¿Cómo puedes encontrar si algunos son vulnerables?*	
	Sabes cómo utilizar la base de datos de *Google Hacking* contra un objetivo concreto.	
	Responde a esta pregunta: *¿Qué tipo de problema de seguridad crees que encontrar contenido que estaba alojado en el pasado podría suponer en un sitio web?*	
	Puedes localizar subdominios de cualquier dominio mediante una de las herramientas proporcionadas y analizar la información del dominio que muestran.	
	Puedes localizar las redes IP de cualquier dominio *.com* o *.es*	
	Puede localizar servicios activos en rangos de redes públicas utilizando `zmap`	

	Puedes localizar hosts activos en redes LAN	
	Puedes configurar un entorno de navegación privado y seguro	
	Puedes geolocalizar una IP (y, en particular, el remitente de un correo electrónico). Puedes responder a esta pregunta: *después de ver la precisión de la ubicación IP que tiene, ¿cuál es, en tu opinión, la utilidad de la geolocalización IP?*	
	Puedes responder a esta pregunta: *¿Qué sucede si en lugar de un correo electrónico tuyo, decides investigar correos electrónicos de otras personas? ¿Qué sucede si el correo electrónico se notifica como comprometido?*	
	Puedes responder a esta pregunta: Si soy capaz de encontrar cualquier imagen o documento que está (o estaba) en cualquier sitio web, *¿qué podemos hacer con estos archivos que pueda revelarme más información? ¿Qué sucede si puedo encontrar el historial de versiones de cualquiera de estos archivos? ¿Qué tipo de información puede proporcionar?*	
	Puedes responder a esta pregunta: Si soy capaz de encontrar cualquier versión de cualquier página web de que una vez fue parte de un sitio web, *¿qué podemos hacer con estos archivos que pueda revelarme más información? ¿Crees que esto podría permitir localizar posibles vulnerabilidades?*	
	Puedes responder a esta pregunta: Imagina que por error dejo en una página web un comentario comprometedor (por ejemplo, indicando un nombre de producto y / o versión, parte del código de servidor en un comentario ...). *¿Qué debemos hacer para eliminar realmente el problema de seguridad que esto podría suponer?*	
	Puedes responder a esta pregunta: Es un hecho que los escaneos UDP son muy lentos, especialmente en WAN / Internet. También se sabe que `nmap` generalmente escanea por defecto sólo los puertos más utilizados en todo el mundo (una lista finita de puertos conocidos). Sabiendo esto, si quieres crear un servicio UDP "oculto" que sea muy difícil de localizar, *¿qué harías?*	
	**He visto cosas que no creerías**: Conoces múltiples técnicas de enumeración y OSINT para extraer mucha información sobre máquinas, empresas y personas que puedes utilizar para investigarlas.	