Grado en Ingeniería Informática

Departamento de Ingeniería Informática

Seguridad en los Sistemas Informáticos

Práctica 3: Recolección de información en fuentes abiertas - Parte I

Autores:
Juan Boubeta Puig,
Manuel Lara Romera,
Jesús Rosa Bilbao,
Pedro José Navas Pérez y
Jesús Lagares Galán

Fecha:
14 de octubre de 2024

Índice

Índice de figuras

1. Objetivo

En esta práctica aprenderemos a recopilar información de fuentes públicas. Es un procedimiento pasivo, conocido en inglés como *Open Source Intelligence* (OSINT), a través del cual el atacante puede recopilar información para comprender mejor el objetivo.

Para lograrlo, podrán usarse buscadores comunes como Google, buscadores especializados como Shodan, y herramientas con las que podemos llevar a cabo esta tarea como, por ejemplo, Wappalyzer.

Esta práctica está dividida en 2 sesiones. En la primera sesión se informará sobre herramientas y procedimientos para la realización de recolección en fuentes abiertas. Reforzando dicho aprendizaje teórico con ejercicios prácticos propuestos en los que se utilizarán Google Dorks, Shodan y WaybackMachine. En la segunda sesión se realizarán ejercicios complementarios y se aprenderán técnicas para el reconocimiento especializado.

Más específicamente, en esta primera parte de la práctica conoceremos y utilizaremos herramientas, buscadores y técnicas especializadas para el reconocimiento de información y accesibles a cualquier usuario.

2. Google Dorks

Google es el motor de búsqueda más utilizado en la actualidad, gran parte de la información que se comparte es accesible a través de este buscador. Este buscador nos provee de la opción **búsqueda avanzada** con la que podemos filtrar información a través de diferentes filtros conocidos como *dorks* (véase un ejemplo en la Figura 1).

Utilizando estos filtros y nuestro ingenio podemos obtener agujeros de seguridad en la configuración de sitios webs e información que, normalmente, no es accesible a cualquier usuario, como por ejemplo documentos internos de la organización o *feeds* de cámaras.

Algunos de estos operadores de búsqueda avanzada son:

- Operador (`-`): Se suprimirá ese argumento en la búsqueda. Ej.: `Macarrones -boloñesa`, buscará macarrones pero excluyendo las búsquedas que contengan boloñesa.
- Operador (|): Equivale al OR lógico, permitiéndonos hacer una búsqueda de más de un término. Ej.: `Sevilla | Betis`, nos devolverá resultados que contengan Sevilla o Betis en su contenido.

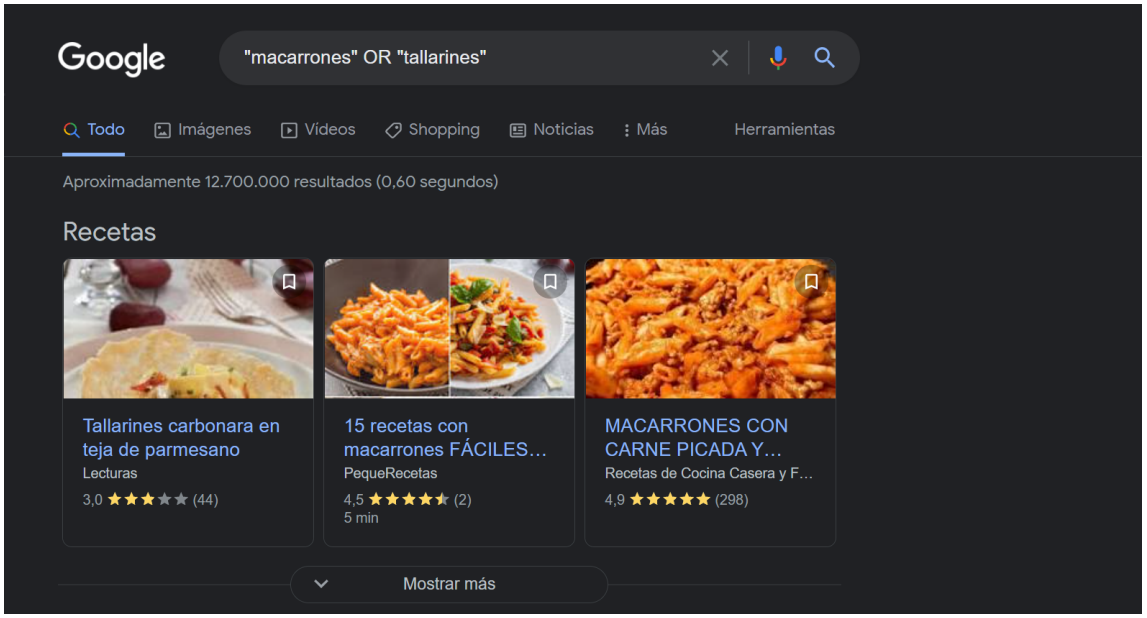

Figura 1: Utilizando la búsqueda avanzada de Google

- Operador (`site:dominio.com`): Nos permite filtrar la búsqueda para que solo se busque en esa web. Ej.: `site:uca.es`, nos devolverá búsquedas dentro del sitio especificado.
- Operador (`filetype:extension`): Nos permite filtrar la búsqueda para buscar solo archivos de esa extensión. Ej.: `filetype:pdf`, filtra la búsqueda para devolvernos documentos PDF que contengan lo buscado.
- Operador (`Intitle:texto`): Este operador busca texto, dentro del contenido del título de una web. Ej.: `Intitle:‘‘Cryptography and Network Security’’`, filtra la búsqueda para mostrarnos resultados cuyo título coincida con lo indicado.
- Operador (`Inurl:texto`): Este operador busca texto en la URL. Funciona igual que el anterior, pero relacionando lo indicado con el contenido en sí de la web.
- Operador (`Author:texto`): Busca artículos o noticias escritos por el nombre o la dirección indicada.

Una vez conocido estos operadores, sería interesante probarlos, sobre todo juntando `site` y `filetype`. A continuación, presentamos operadores un poco especiales que no se pueden usar junto a otros:

- Operador (`Allintext:texto`): Este operador busca únicamente el texto especificado dentro del contenido de webs.
- Operador (`Allintitle:texto`): Este operador devuelve las páginas webs que tienen el texto indicado en su título.
- Operador (`Allinurl:texto`): Este operador busca texto solo en URLs.
- Operador (`Cache:dominio.com`): Con este operador accedemos a la versión de la página web que Google tiene en su caché.
- Operador (`Link:dominio.com`): Este operador busca páginas webs que tienen enlaces a la página especificada.
- Operador (`Related:dominio.com`): Este operador busca páginas web que son *similares* a la proporcionada.

Si nos fijamos en los operadores (`Intitle:texto`) y (`Allintitle:texto`), pueden parecer similares, aunque no lo son. Por ejemplo, hagamos la siguiente búsqueda con los dos operadores: 'Asignatura SSI'.

- `Allintitle: Asignatura SSI`: Esta búsqueda devolverá resultados con **las dos palabras** en el título (coincidencia exacta).
- `Intitle: Asignatura SSI`: Esta búsqueda devolverá resultados con **la primera palabra** en el título, en este caso 'Asignatura'. La palabra 'SSI' podrá no aparecer en ninguna parte.

Por lo tanto, tenemos el siguiente comportamiento con estos operadores:

[Allintitle:Asignatura SSI] == [Intitle:Asignatura Intitle:SSI]

Sabiendo cómo funcionan los operadores descritos anteriormente y con un poco de destreza veremos que podemos realizar búsquedas mucho más concretas. Para ello, podemos acceder e indagar en el siguiente sitio web: `https://www.exploit-db.com/google-hacking-database` Aquí nos encontramos muchos de los operadores funcionando juntos, en instrucciones ya hechas por otras personas, que nos servirán para un propósito específico. Como podemos observar, los *dorks* se encuentran organizados en categorías, que se describen a continuación.

2.1. Usuarios

Realizando una búsqueda de este tipo en Google podemos encontrar desde una lista de usuarios y contraseñas, hasta una página web a la que podamos acceder como administradores. Un ejemplo sería utilizar `filetype:xls ‘‘username | password’’`, esto nos devolverá hojas de cálculo Excel con usuarios y contraseñas.

2.2. Contraseñas

En esta ocasión, buscaremos ficheros con contraseñas realizando búsquedas más complejas mediante:

```
inurl: ‘‘passes’’ OR inurl:‘‘password’’ OR inurl:‘‘credentials’’ -search
-download -techsupt -git -games -gz -bypass -exe filetype:txt
```

En esta búsqueda, con la instrucción *inurl* buscamos archivos que contengan esos términos que hemos especificado. Por ejemplo, podríamos añadir `inurl:‘‘usernames’’` para buscar también nombres de usuarios. Con el operador `OR` establecemos un OR lógico (también podríamos establecer un `AND`) entre los términos especificados. Con el operador `-` eliminamos resultados que no nos interesan y, por último, buscaremos archivos .txt.

2.3. Detección de servidores

Aunque para encontrar servidores el buscador más adecuado es Shodan [6], del que hablaremos posteriormente, gracias a Google también podremos hacer búsquedas para detectar servidores. Un ejemplo para conseguir estas búsquedas sería buscar en el título de la página algún reporte del estado de Apache como: `intitle: ‘‘Apache Status’’ ‘‘Apache Server Status for’’.`

Podemos encontrar dorks en la categoría *Web Server Detection*, o en [1].

2.4. Mensaje de error

El propósito de realizar este tipo de búsqueda es encontrar archivos con mensajes de error que nos lleven a deducir muchas otras cosas. Si logramos encontrar este tipo de información sobre una web, seguramente estos mensajes de error apunten a brechas de seguridad que podamos aprovechar. Para realizar este tipo de búsqueda podemos utilizar:

```
‘‘Warning: mysql.query()’’ ‘‘invalid query’’ -foro -help -ayuda -como.
```

Podemos encontrar dorks en la categoría en [2].

2.5. Escaneo de servidores vulnerables

Dentro de esta categoría podemos encontrar todos los servidores que presenten puertas traseras y otras vulnerabilidades. Podemos encontrar dorks sobre esta categoría en [3].

2.6. Búsqueda de información sensible

En esta categoría se engloba la búsqueda de todo lo que podría resultar información sensible (DNIs, documentos no públicos del gobierno, o cualquier tipo de información cuyo dueño no imaginaba que sería pública). Para encontrar este tipo de información debemos realizar una búsqueda del siguiente estilo:

```
‘‘not for public release’’ inurl:gob OR inurl:edu OR inurl:mil -.com -.net
-.es
```

Podemos encontrar dorks en la categoría en [4].

3. Buscadores especializados

Además de utilizar Google como motor de búsqueda, también podemos utilizar otros motores de búsquedas especializados en proveernos de otro tipo de información para la recolección. Estos buscadores restringen la información ofrecida en función de una serie de requisitos, por ejemplo, los metadatos que devuelven al servidor. Algunos ejemplos de estos motores son Shodan y WaybackMachine.

3.1. Shodan

Shodan [6] es un motor que tiene como objetivo ubicar en una misma herramienta todo tipo de dispositivos conectados a Internet, desde *routers* hasta cámaras de seguridad, pasando por todo lo que compete el Internet de las cosas o *Internet of Things* (IoT) [5].

Gracias a este buscador podemos recibir información muy útil en poco tiempo a fin de investigar nuevas vulnerabilidades o conocer nueva información sobre dispositivos en general.

Para utilizar Shodan accedemos a [6]; con el fin poder utilizar los filtros deberemos crearnos una cuenta previamente. Una vez hayamos creado la cuenta, escribimos la búsqueda en su barra de búsqueda, o bien indagamos los diferentes menús que Shodan pone a nuestra disposición. En este caso, haremos clic en el botón de la esquina superior izquierda denominado *Explore* (véase Figura 2). Una vez dentro de la sección, accedemos a cualquier categoría, por ejemplo, a la de *Webcam*. Dentro de la categoría seleccionada podemos ver un enorme listado de todos los dispositivos que Shodan ha encontrado (véase Figura 3). Si hacemos clic en cualquier de los dispositivos, Shodan nos permitirá acceder a todo tipo de información sobre él: IP, puertos abiertos, país al que pertenece, red de la organización, tecnología web utilizada, CVE (*Common Vulnerabilities and Exposures*) descubiertos, etc. A modo

de ejemplo, véase Figura 4. Una vez que hemos recolectado toda esta información, ya queda en nuestras manos con qué fin utilizarla.

Además de estas búsquedas simples, dentro de Shodan también existe la opción de utilizar filtros o *dorks*, por lo que el buscador adquiere un gran potencial. Gracias a estos filtros, con una simple búsqueda podríamos obtener todos los dispositivos que tienen la contraseña por defecto en Estados Unidos ``Default password country:US''.

Algunos ejemplos de estos filtros son los siguientes:

- `country`. Utilizando este filtro podemos obtener resultados específicos sobre un país determinado. Ejemplo: `country: ES`.
- `os`. Gracias a este filtro podremos filtrar los resultados por sistemas operativos, así como sus versiones. Por ejemplo: `os:``Windows XP''`.
- `port`. Utilizando este filtro podremos obtener resultados específicos sobre los puertos que queramos. Ejemplo: `port: 80`.
- `isp`. En caso de querer filtrar las búsquedas por *Internet Service Provider* (ISP) podemos utilizar este filtro. Ejemplo: `isp: telefonica`.
- `hostname`. Si lo que queremos es buscar por nombre de dominio podemos utilizar este filtro. Ejemplo: `hostname:server1.computer.com`.

3.2. WaybackMachine

WaybackMachine es un motor de búsqueda/base de datos que contiene copias de una gran cantidad de páginas de Internet. Esta herramienta aparece con el objetivo de recolectar la información efímera que aparece en Internet, como la web de un periódico, para que nunca desaparezca. En la actualidad recoge más de 330 mil millones de páginas webs y otros documentos. A WaybackMachine se le suele denominar la hemeroteca digital.

Para acceder a ella debemos entrar en [7]. Una vez dentro introducimos una web a buscar en el buscador, por ejemplo `www.uca.es`, como podemos ver en la Figura 5. Se cargará la búsqueda y se nos mostrará todo el historial de *snaphots* que el buscador tiene de la web, como vemos en la Figura 6. Una *snapshot* no es más que una copia instantánea que se tomó de la web en un momento determinado. Si ahora hacemos clic en uno de los años que aparecen en la línea de tiempo, en una burbuja (independientemente de su color) y en la hora en la que se tomó la *snapshot*, como vemos en la Figura 7, podemos comprobar cómo era la web en ese tiempo y navegar por ella (véase la Figura 8).

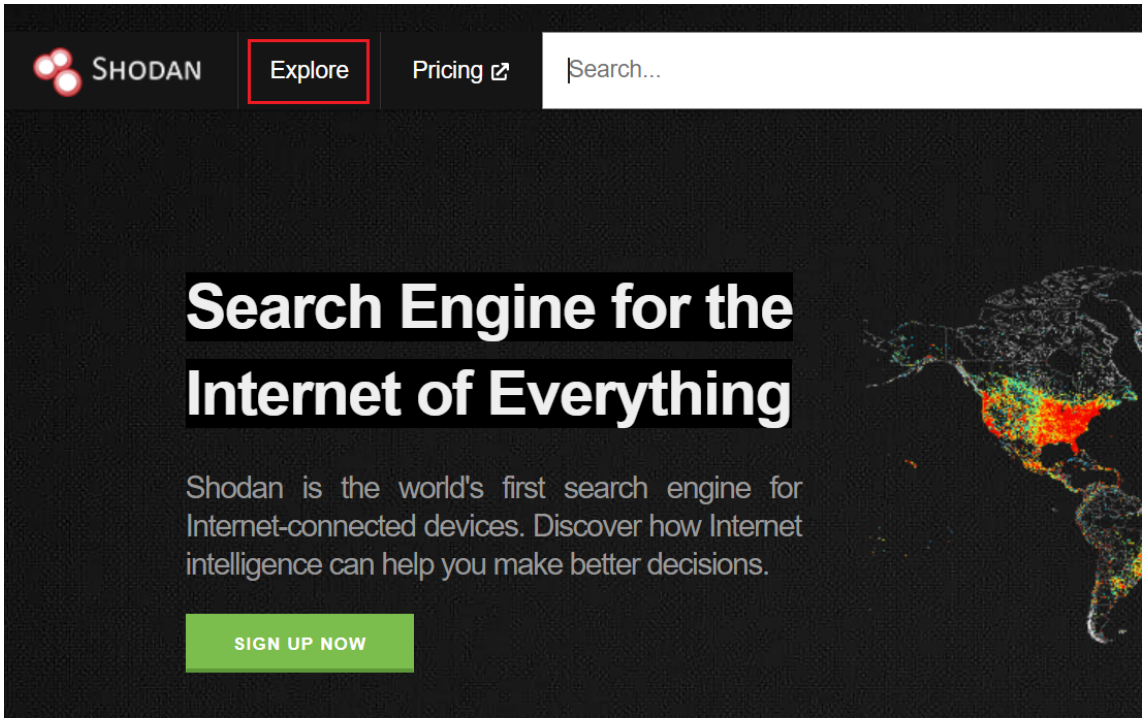

Figura 2: Pulsamos en *Explore* dentro de Shodan

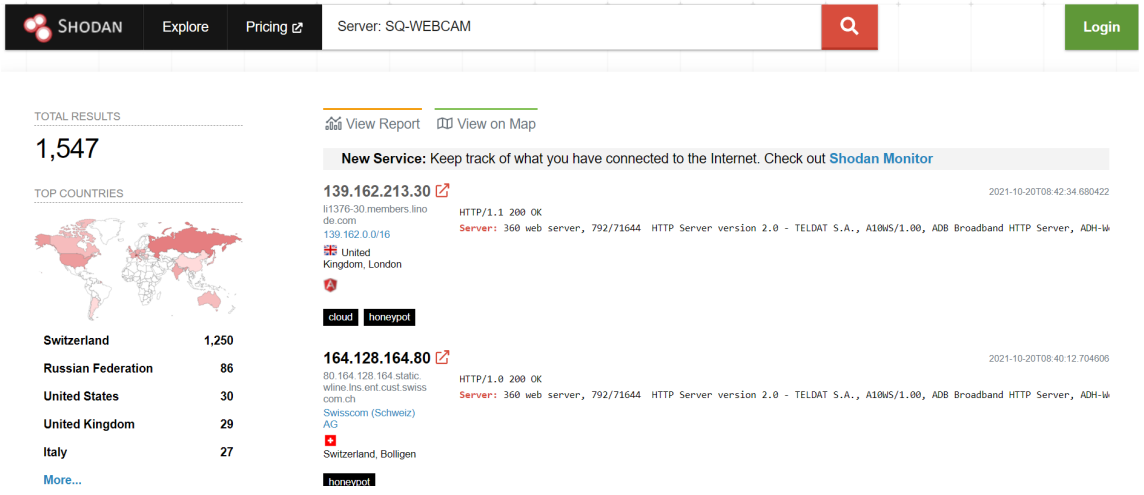

Figura 3: Lista de dispositivos que Shodan nos muestra

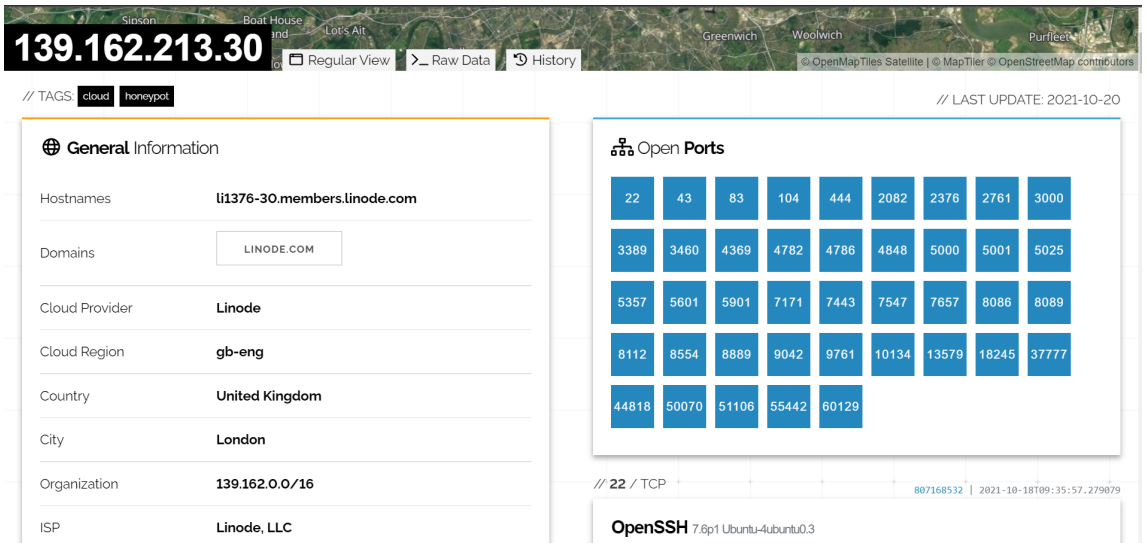

Figura 4: Información brindada por Shodan sobre el dispositivo seleccionado

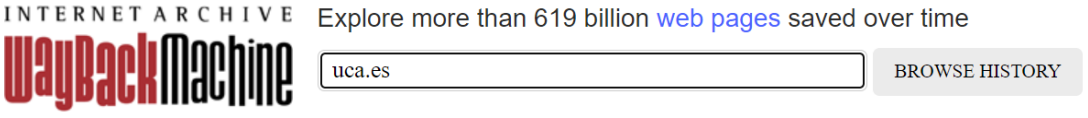

Figura 5: Introducimos una web en el buscador de WaybackMachine

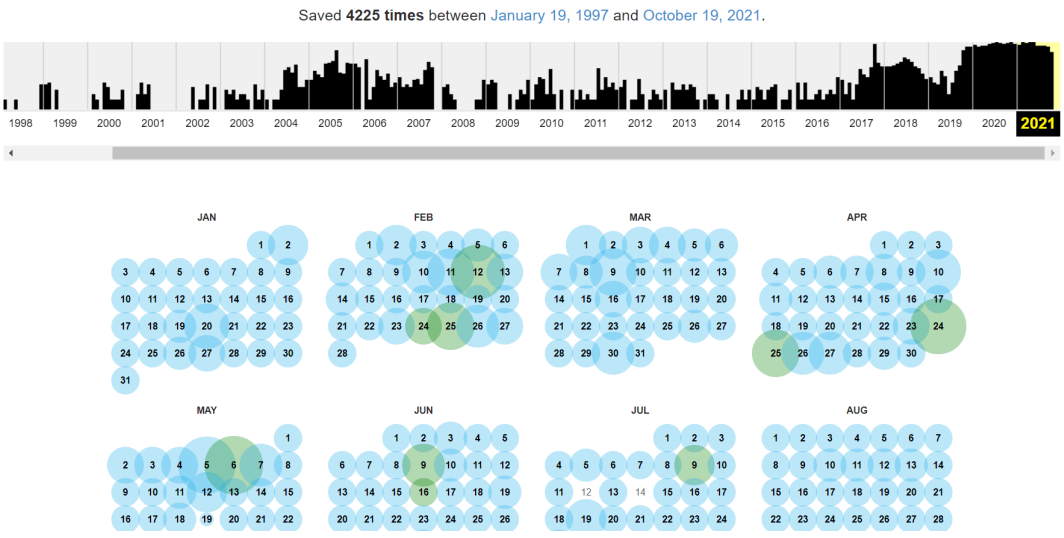

Figura 6: Resultado devuelto por WaybackMachine sobre la web de la UCA

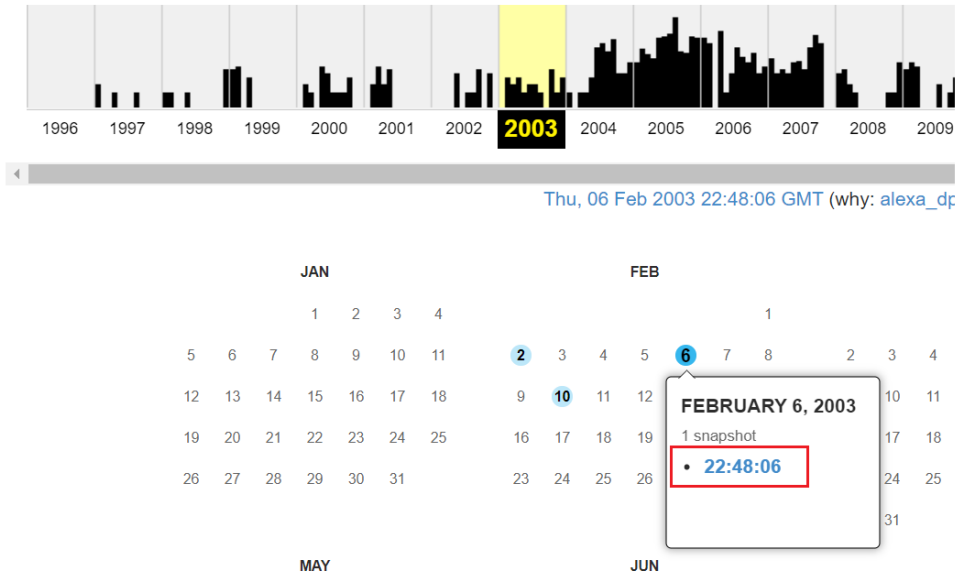

Figura 7: Hacemos clic para obtener los datos de la *snapshot*

Figura 8: Web de la UCA el 6 de febrero de 2003

4. Ejercicios

A continuación se describen los ejercicios a realizar en esta práctica.

4.1. Ejercicio 1

En primer lugar, debemos acceder al enlace `https://goo.gl/pzwPqs`. Dicho enlace nos abre una base de datos de dorks que nos permitirán realizar búsquedas para obtener nombres de usuarios de ciertos sistemas.

Comenzaremos accediendo al enlace cuyo dork es: `‘‘authentication failure; logname=’’ ext:log`. Esto nos redireccionará hacia una búsqueda en Google. Accederemos al enlace cuya dirección es `http://www.cs.fsu.edu/~langley/CIS4385-2015-1/2015-01-logs-test/auth.log`.

Conteste a las siguientes preguntas:

- ¿A qué tipo de archivo estamos accediendo, y para qué sirve en los sistemas?
- ¿Qué información útil podríamos obtener de este archivo si nuestra finalidad fuera maliciosa?

A continuación, accederemos al dork que se titula:
`filetype:conf inurl:proftpd.conf -sample`.
Una vez redireccionados a la búsqueda, accederemos al enlace `http://ftp.rinotel.com/Software/Linux/proftpd.conf`.

Conteste a las siguientes preguntas:

- ¿A qué tipo de archivo estamos accediendo, y para que sirve en los sistemas?
- ¿Qué información útil podríamos obtener de este archivo si nuestra finalidad fuera maliciosa?

Por último, accedemos al dork con título: `filetype:log username putty`. Haremos clic en el enlace que aparece, cuya URL es `http://asanovich.free.fr/COMPASS/4-1.log`.

Conteste a las siguientes preguntas:

- ¿A qué tipo de archivo estamos accediendo, y para qué sirve en los sistemas?
- ¿Qué información útil podríamos obtener de este archivo si nuestra finalidad fuera maliciosa?

4.2. Ejercicio 2

Accedemos a la página de dorks para archivos con contraseña `https://goo.gl/qcJGDR`.

A continuación, hacemos clic en el dork con título: `filetype:xml config.xml passwordHash Jenkins`. Esto nos dará una búsqueda, de la cual accederemos al enlace cuya URL es `https://open-bitbucket.nrao.edu/projects/CASA/repos/casa-pkg/browse/configuration/jenkins/warp/users/ville/config.xml?at=67d6e09964459b`

Conteste a las siguientes preguntas:

- ¿Qué tipo de archivo nos devuelve el uso de este dork?
- ¿Qué información útil podríamos obtener de este archivo si nuestra finalidad fuera maliciosa?

A continuación, usaremos el dork cuyo título es: `inurl:proftpdpasswd`. Esto nos dará como resultado una serie de URLs, de las cuales accederemos a `https://github.com/aptorres27/AnnalizaTorresPersonalWebsite/blob/master/proftpdpasswd`.

Conteste a las siguientes preguntas:

- ¿Qué información contienen estos archivos y para qué podría sernos útil dicha información si tuviéramos una intención maliciosa?

4.3. Ejercicio 3

Seleccionamos la categoría "Web Server Detection" en Google Hacking Database, y buscamos el dork cuyo título es `intext:‘‘Powered by phpSQLiteCMS’’ | intitle:‘‘phpSQLiteCMS - A simple & lightweight CMS’’`.

Al hacer esto llegaremos a un resultado de Google y accedemos a varios enlaces devueltos por la búsqueda para detectar qué tipo de información nos devuelve la búsqueda realizada.

Conteste a las siguientes preguntas:

- ¿Qué información importante estamos obteniendo a raíz del uso de este dork?

A continuación, usaremos el dork con título `‘‘PHP Credits’’ ‘‘Configuration’’ ‘‘PHP Core’’ ext:php inurl:info`. Haremos clic sobre el enlace con dirección `http://61.216.3.97/info.php`.

Conteste a las siguientes preguntas:

- ¿A qué estamos accediendo exactamente?

- ¿Qué información útil podemos obtener de aquí si nuestras intenciones fueran maliciosas?

Finalmente, usaremos el dork cuyo título es: `inurl:phpsysinfo/index.php?disp=dynamic`. Esto nos devolverá un resultado y accederemos al enlace: `http://phpsysinfo.sourceforge.net/phpsysinfo/index.php?disp=dynamic`.

Conteste a las siguientes preguntas:

- ¿A qué estamos accediendo exactamente?
- ¿Qué información útil podemos obtener de aquí si nuestras intenciones fueran maliciosas?

4.4. Ejercicio 4

Seleccionamos la categoría "Error Messages" en Google Hacking Database, y buscamos el dork cuyo título es `inurl:index of driver.php?id=` y en el resultado accederemos al enlace cuya URL es `https://uftm.edu.br/proplan/index.php?option=com_content&view=article&id=110:gtm1-divulgacao-da-uftm&catid=16:cev`.

Conteste a las siguientes preguntas:

- ¿A qué tipo de información estamos accediendo?
- ¿Qué información importante obtendríamos de este resultado, si nuestra intención fuera maliciosa?

A continuación, haremos uso del dork cuyo título es:
`inurl:/siteminderagent/forms/smpwservices.fcc`, y accederemos al enlace con URL `https://eportal.pwc.ca/siteminderagent/forms/smpwservices.fcc`.

Conteste a las siguientes preguntas:

- ¿Qué información estamos obteniendo exactamente?
- Con esta información, ¿qué tipo de ataque podríamos realizar si nuestras intenciones fueran maliciosas?

4.5. Ejercicio 5

Debido a que para la completa exploración y uso de los dorks de esta categoría, necesitaríamos unos conocimientos bastante más avanzados de los que se supone tenemos

al momento de la realización de esta práctica, en este apartado solo analizaremos el uso de un dork.

Haremos click en el dork cuyo título es ‘‘dirLIST - PHP Directory Lister’’ ‘‘Banned files: php | php3 | php4 | php5 | htaccess | htpasswd | asp | aspx’’ ‘‘index of’’ ext:php y en el resultado accederemos al enlace cuya URL es http://www.jeeptelevision.com/fotoeventi/index.php?folder=c2ljaWxpYQ==.

Conteste a las siguientes preguntas:

- ¿A qué nos da acceso dicho enlace?
- ¿Qué acciones podemos realizar usando este enlace?
- Desde el punto de vista de un ataque malicioso, ¿cómo podríamos sacar partido de este enlace?

4.6. Ejercicio 6

De los dorks que encontramos en el enlace usaremos dos, el primero será el que tiene como título allinurl: drive.Google.com/open?id=.

Con este tipo de dork podríamos acceder, por ejemplo, al enlace con URL: https://drive.google.com/file/d/0ByOO2CwASGN0d0hNbWM4OEc5ZmM/view?resourcekey=0-OrCfpNOoo9q7fbXmsMIHQQ .

Conteste a las siguientes preguntas:

- Exactamente, ¿a qué estamos accediendo?
- ¿Qué utilidad podemos encontrarle a este dork?

A continuación, usaremos el dork cuyo título es filetype:txt ‘‘gmail’’ | ‘‘hotmail’’ | ‘‘yahoo’’ -robots site:gov | site:us.

Accederemos al enlace: https://www.sec.gov/Archives/edgar/data/921669/000092847508000248/dfan14a070708.txt.

Conteste a las siguientes preguntas:

- ¿Qué estamos viendo?
- ¿Qué información útil podemos obtener de este enlace?
- ¿Por qué podríamos considerar esta información sensible?

4.7. Ejercicio 7

Haciendo uso de dorks de Shodan, encuentre las IP correspondientes a 2 equipos con servicio MQTT (*Message Queuing Telemetry Transport*) en la ciudad de Sevilla de la organización "arsys.es".

4.8. Ejercicio 8

Haciendo uso de Shodan, encuentre el número de equipos del *Nuclear Physics Institute* de Moscu con el puerto Telnet a la escucha.

4.9. Ejercicio 9

Haciendo uso de Shodan, encuentre el nombre de la organización que más servidores de Minecraft hostea en la actualidad. Una vez hecho esto encuentre la IP de los servidores de este conocido juego en la provincia de Sevilla.

4.10. Ejercicio 10

Usando Waybackmachine, encuentre información sobre *mars pathfinder* divulgada en 1996 por la NASA en su dominio nasa.gov.

Referencias

[1] OffSec Services Limited 2024: *Google Hacking Database* . https://www.exploit-db.com/google-hacking-database?category=4. [Última consulta: 2024-10-14].

[2] OffSec Services Limited 2024: *Google Hacking Database* . https://www.exploit-db.com/google-hacking-database?category=7. [Última consulta: 2024-10-14].

[3] OffSec Services Limited 2024: *Google Hacking Database* . https://www.exploit-db.com/google-hacking-database?category=6. [Última consulta: 2024-10-14].

[4] OffSec Services Limited 2024: *Google Hacking Database* . https://www.exploit-db.com/google-hacking-database?category=8. [Última consulta: 2024-10-14].

[5] Jesús Rosa-Bilbao, Juan Boubeta-Puig y Adrian Rutle: *CEPEDALoCo: An event-driven architecture for integrating complex event processing and blockchain through low-code*. Internet of Things, 22(100802):1–16, Julio 2023. https://doi.org/10.1016/j.iot.2023.100802.

[6] Shodan: *Shodan Search Engine*. https://www.shodan.io/. [Última consulta: 2024-10-14].

[7] Internet Achieve Waybackmachine: *Internet Achieve* . https://web.archive.org/. [Última consulta: 2024-10-14].