



GRADO EN INGENIERÍA INFORMÁTICA

DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

SEGURIDAD EN LOS SISTEMAS INFORMÁTICOS

---

## Práctica 5.1: Recolección de información en fuentes abiertas con buscadores web genéricos

---

**Autores:**

Juan Boubeta Puig,  
Manuel Lara Romera y  
Jesús Rosa Bilbao

**Fecha:**

4 de noviembre de 2019

# Índice

<b>1. Objetivo</b>	<b>3</b>
<b>2. Herramientas necesarias</b>	<b>3</b>
<b>3. Google Dorks</b>	<b>4</b>
3.1. Usuarios . . . . .	4
3.1.1. Ejercicio 1 . . . . .	5
3.2. Contraseñas . . . . .	9
3.2.1. Ejercicio 2 . . . . .	9
3.3. Detección de Servidores . . . . .	12
3.3.1. Ejercicio 3 . . . . .	12
3.4. Mensaje de Error . . . . .	15
3.4.1. Ejercicio 4 . . . . .	16
3.5. Escaneo de Servidores Vulnerables . . . . .	19
3.5.1. Ejercicio 5 . . . . .	19
3.6. Búsqueda de Información Sensible . . . . .	19
3.6.1. Ejercicio 6 . . . . .	20

## Índice de figuras

1.	Base de datos de Google dorks con nombres de usuarios. . . . .	5
2.	Búsqueda en Google: 'authentication failure; logname' ext:log. . . . .	6
3.	Búsqueda en Google: filetype:conf inurl:proftpd.conf -sample. . . . .	7
4.	Búsqueda en Google: filetype:log username putty. . . . .	8
5.	Base de datos de Google dorks con contraseñas. . . . .	9
6.	Búsqueda Google: filetype:xml config.xml passwordHash Jenkins . . . . .	10
7.	Búsqueda en Google: inurl:proftpdpasswd. . . . .	11
8.	Base de datos de Google dorks de detección de servidores. . . . .	12
9.	Búsqueda en Google: intext:"Powered by phpSQLiteCMS"  intitle: 'phpSQLiteCMS - A simple & lightweight CMS'. . . . .	13
10.	Búsqueda en Google: 'PHP Credits' 'Configuration' 'PHP Core' ext:php inurl:info. . . . .	14
11.	Búsqueda en Google: inurl:phpsysinfo/index.php?disp=dynamic. . . . .	15
12.	Base de datos de Google dorks para mensajes de error. . . . .	16
13.	Búsqueda en Google: inurl:index of driver.php?id= . . . . .	17
14.	Búsqueda en Google: inurl:/smpwservices.fcc   '/lm.private /CkeSetter.aspx' . . . . .	18
15.	Base de datos de Google dorks para servidores vulnerables . . . . .	20
16.	Búsqueda en Google: 'dirLIST - PHP Directory Lister' 'Banned files: php   php3   php4   php5   htaccess   httpasswd   asp   aspx' 'index of' ext:php . . . . .	21
17.	Base de datos de Google dorks para información jugosa . . . . .	22
18.	Búsqueda en Google: allinurl: drive.Google.com/open?id= . . . . .	23
19.	Búsqueda en Google: filetype:txt 'gmail'   'hotmail'   'yahoo' -robots site:gov   site:us . . . . .	24
20.	Búsqueda en Google: 'not for public release' inurl:gob OR inurl:edu OR inurl:mil -.com -.net -.es . . . . .	25

## 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 (Google, Yahoo, Bing, startpage) y buscadores especializados. En particular, en esta práctica aprenderemos a usar el buscador Google de una manera más precisa y cómo con el uso de los Google dorks [1, 2] se pueden encontrar ficheros sensibles en el espacio web. Esto nos hará tomar conciencia de las rutas o directorios donde no deberíamos almacenar ficheros sensibles, así como del peligro de almacenarlos sin ninguna protección.

## 2. Herramientas necesarias

Para la correcta realización de esta práctica, lo primero será familiarizarnos con los operadores que podemos usar en Google. A continuación, listamos los más habituales junto con su funcionalidad:

- Operador (-): Se suprime 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.
- 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.

### 3. Google Dorks

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.

#### 3.1. 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 usar **filetype:xls** **‘‘username | password’’**, esto nos devolvería hojas de cálculo Excel con usuarios y contraseñas.

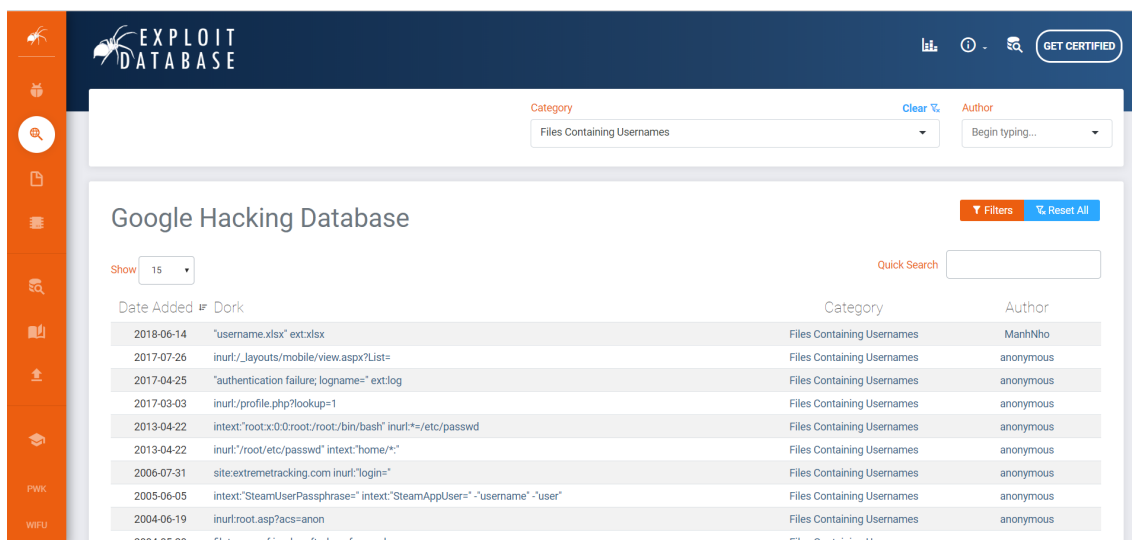


Figura 1: Base de datos de Google dorks con nombres de usuarios.

### 3.1.1. Ejercicio 1

Para realizar los ejercicios debemos acceder al enlace <https://goo.gl/pzwPqs>. Esto nos abre una base de datos de dorks (véase Figura 1) que nos permitirán realizar búsquedas para obtener nombres de usuarios de ciertos sistemas.

Para comenzar accederemos al tercer enlace para hacer uso del dork cuyo título 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>, que se muestra en la Figura 2.

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 décimo dork que se titula:

`filetype:conf inurl:proftpd.conf -sample`.

Una vez redireccionados a la búsqueda (véase Figura 3), accederemos al enlace <ftp://www.eeng.dcu.ie/pub/ee454/cygwin/etc/proftpd.conf>.

Conteste a las siguientes preguntas:

- ¿A qué tipo de archivo estamos accediendo, y para que sirve en los sistemas?

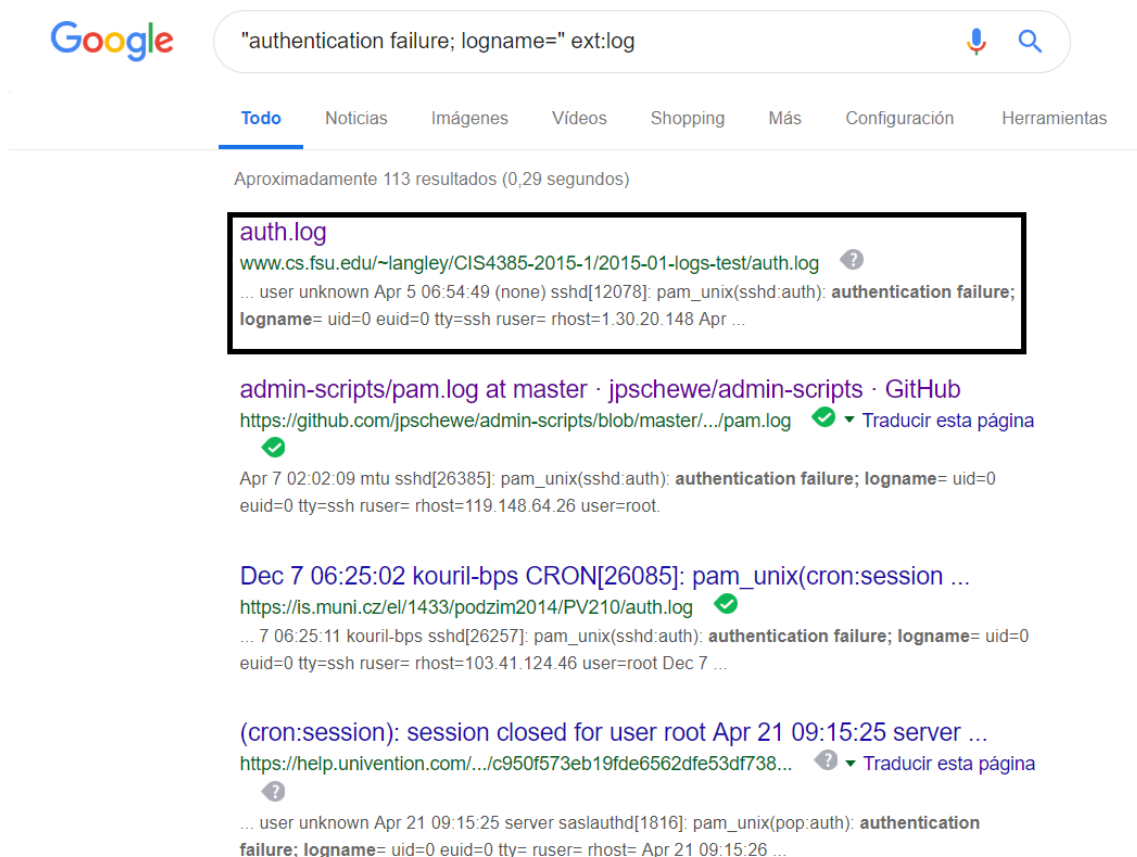


Figura 2: Búsqueda en Google: 'authentication failure; logname' ext:log.

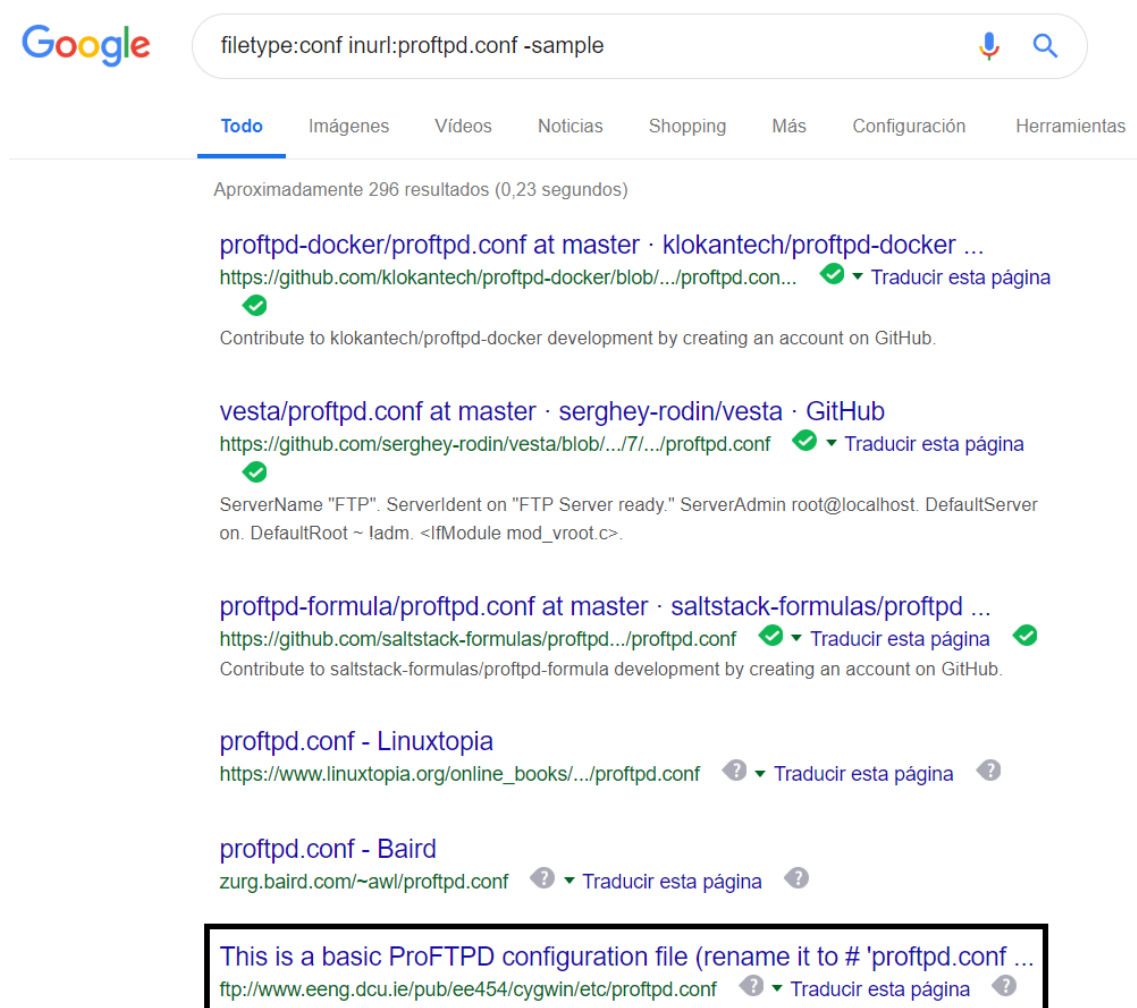


Figura 3: Búsqueda en Google: filetype:conf inurl:proftpd.conf -sample.



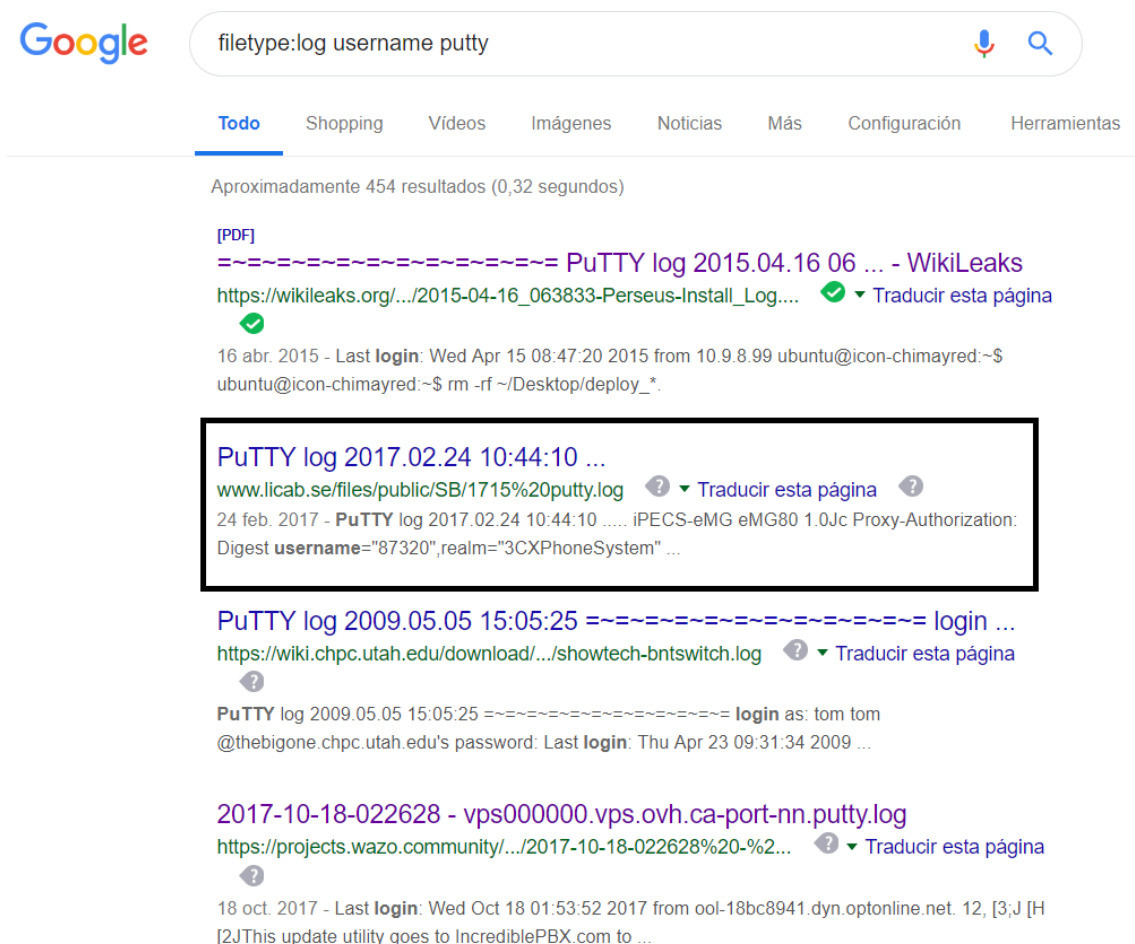


Figura 4: Búsqueda en Google: `filetype:log username putty`.

- ¿Qué información útil podríamos obtener de este archivo si nuestra finalidad fuera maliciosa?

Por último, accedemos al undécimo dork con título: `filetype:log username putty`. Haremos clic en el enlace que aparece en la Figura 4, cuya URL es `http://www.licab.se/files/public/SB/1715%20putty.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?

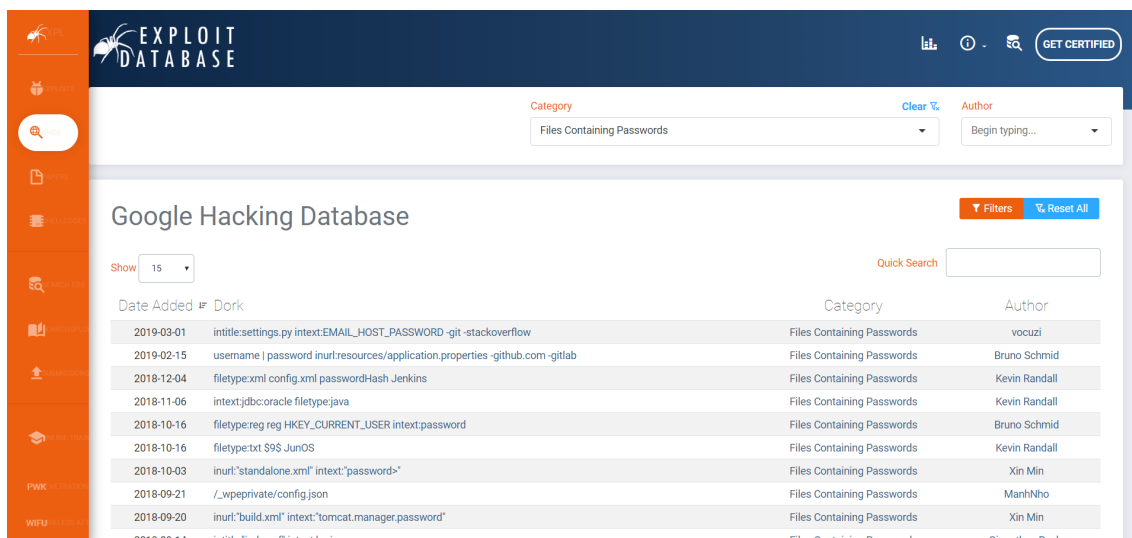


Figura 5: Base de datos de Google dorks con contraseñas.

## 3.2. Contraseñas

En esta ocasión, para comprobar la potencia de los dorks, buscaremos ficheros con contraseñas realizando búsquedas más complejas mediante:

```
inurl:'passes' OR inurl:'passwords' 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; podríamos añadir `inurl:'usernames'` para buscar también nombres de usuarios. Con el operador `OR` establecemos un `OR` lógico entre los términos especificados. Con el operador `-` eliminamos resultados que no nos interesan y, por último, buscaremos archivos `.txt`.

Con estos criterios de búsqueda seguramente nos aparezcan muchas páginas de *phishing*.

### 3.2.1. Ejercicio 2

En primer lugar, accedemos a la página de dorks para archivos con contraseña (véase Figura 5): <https://goo.gl/qcJGDR>.

A continuación, hacemos clic en el tercer dork con título: `filetype:xml config.xml passwordHash Jenkins`. Esto nos dará una búsqueda, de la cual accederemos al enlace que aparece en Figura 6, cuya URL es <https://github.com/mozilla/mozplatformqa-jenkins-config/blob/master/users/>

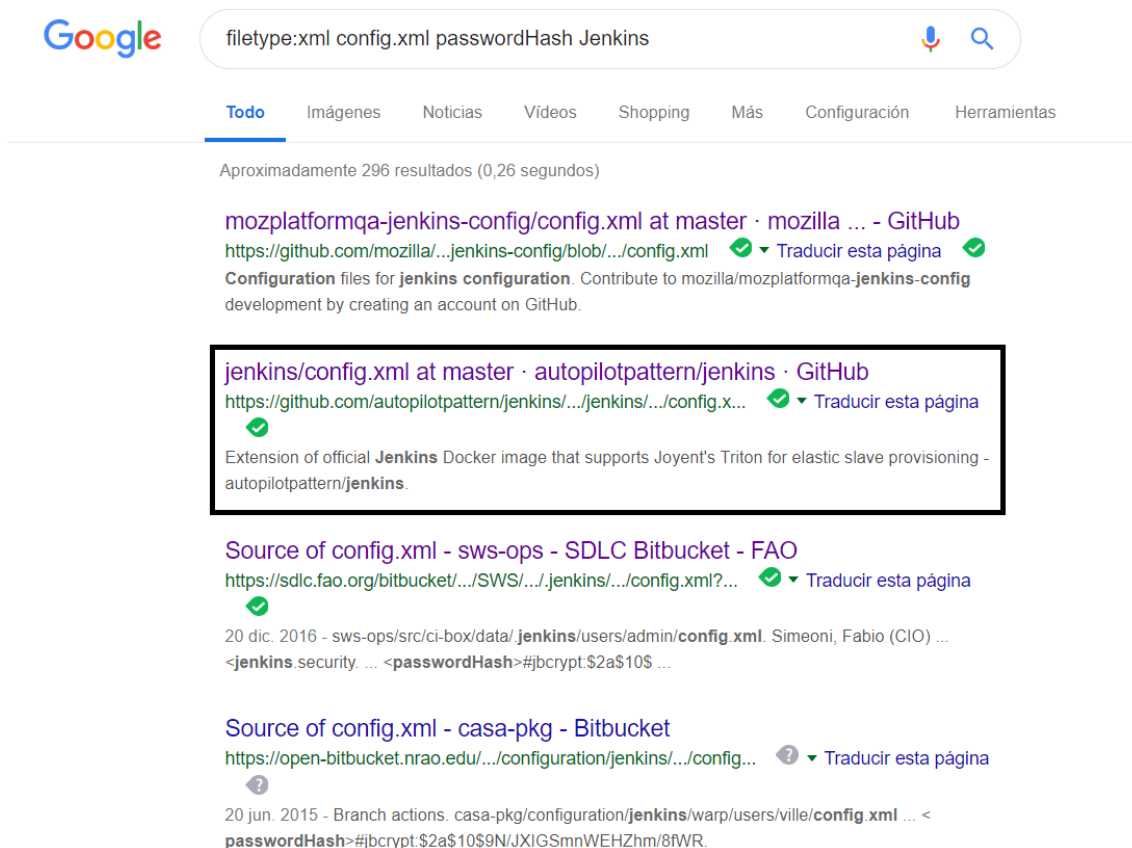


Figura 6: Búsqueda Google: filetype:xml config.xml passwordHash Jenkins

sydpolk/config.xml.

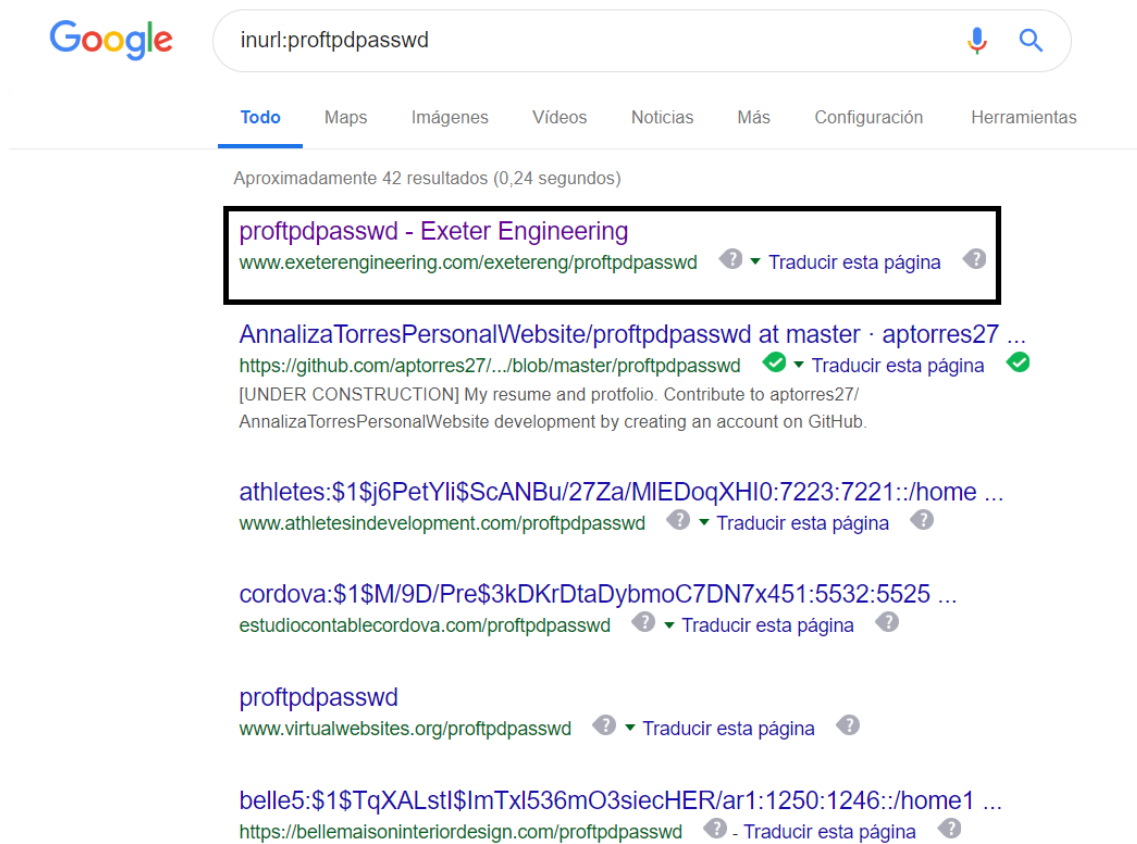
Conteste a las siguientes preguntas:

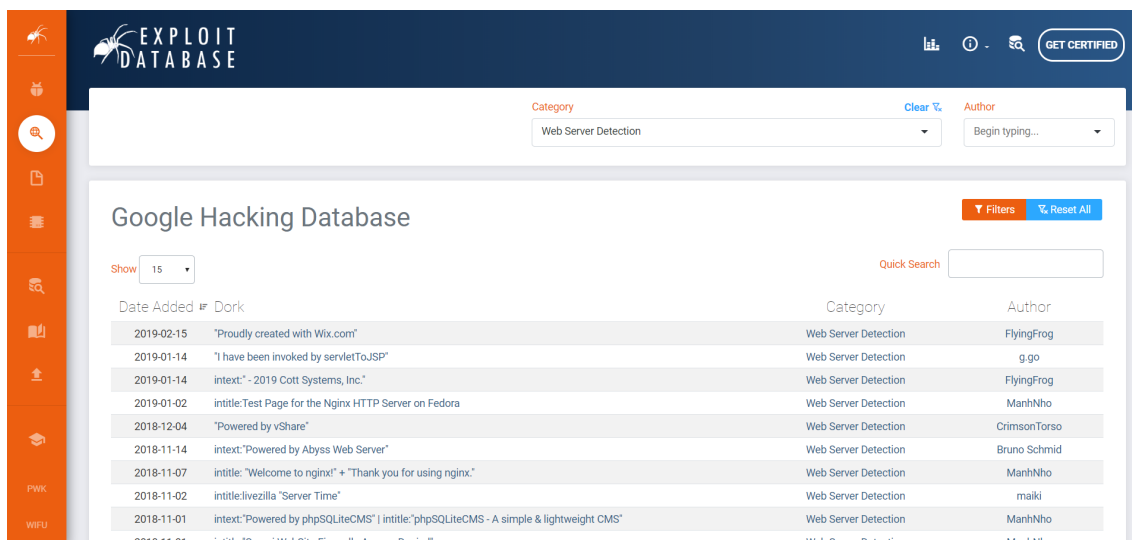
- ¿Qué tipo de archivo nos devuelve el uso de este dork?
- ¿Qué información útil podríamos obtener de este archivo si nuestra finalidad fuera maliciosa?

Por último, usaremos el dork cuyo título es: `inurl:proftpdpasswd` (situado en la página 6). Esto nos dará como resultado una serie de URLs, de las cuales accedemos a `http://www.exeterengineering.com/exetereng/proftpdpasswd` (véase Figura 7).

Conteste a las siguientes preguntas:

- ¿Qué información contienen estos archivos y para qué podría sernos útil dicha

Figura 7: Búsqueda en Google: `inurl:proftpdpasswd`.



The screenshot shows the Exploit Database website interface. At the top, there's a navigation bar with the 'EXPLOIT DATABASE' logo and a 'GET CERTIFIED' button. Below this, there's a search bar with a 'Category' dropdown set to 'Web Server Detection' and an 'Author' dropdown. The main content area is titled 'Google Hacking Database' and features a 'Show' dropdown set to '15' and a 'Quick Search' input field. A table of results is displayed with columns for 'Date Added', 'Dork', 'Category', and 'Author'.

Date Added	Dork	Category	Author
2019-02-15	"Proudly created with Wix.com"	Web Server Detection	FlyingFrog
2019-01-14	"I have been invoked by servletToJSP"	Web Server Detection	g.go
2019-01-14	intext:" - 2019 Cott Systems, Inc."	Web Server Detection	FlyingFrog
2019-01-02	intitle:Test Page for the Nginx HTTP Server on Fedora	Web Server Detection	ManhNho
2018-12-04	"Powered by vShare"	Web Server Detection	CrimsonTorso
2018-11-14	intext:"Powered by Abyss Web Server"	Web Server Detection	Bruno Schmid
2018-11-07	intitle:"Welcome to nginx!" + "Thank you for using nginx."	Web Server Detection	ManhNho
2018-11-02	intitle:livezilla "Server Time"	Web Server Detection	maiki
2018-11-01	intext:"Powered by phpSQLiteCMS"   intitle:"phpSQLiteCMS - A simple & lightweight CMS"	Web Server Detection	ManhNho
2018-11-01	intitle:"Sucuri WebSite Firewall - Access Denied"	Web Server Detection	ManhNho

Figura 8: Base de datos de Google dorks de detección de servidores.

información si tuviéramos una intención maliciosa?

### 3.3. Detección de Servidores

Aunque para encontrar servidores el buscador más adecuado es Shodan (<https://www.shodan.io/>), con Google también podremos hacer búsquedas para lograrlo. Un ejemplo para conseguir estas búsquedas sería buscar en el título de la página algún reporte de estado de Apache: `intitle: "Apache Status" "Apache Server Status for"`.

Podemos encontrar dorks sobre esta categoría en el siguiente enlace: <https://google.com/search?q=intitle:Apache+Server+Status+for&btnG=Buscar>.

#### 3.3.1. Ejercicio 3

Accedemos a la URL indicada anteriormente, y comenzaremos con el undécimo dork cuyo título es `intext:"Powered by phpSQLiteCMS" | intitle:"phpSQLiteCMS - A simple & lightweight CMS"`, de los que aparecen en la Figura 8.

Al hacer esto llegaremos a un resultado de Google como el que aparece en la Figura 9, accederemos al segundo enlace.

Conteste a las siguientes preguntas:

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

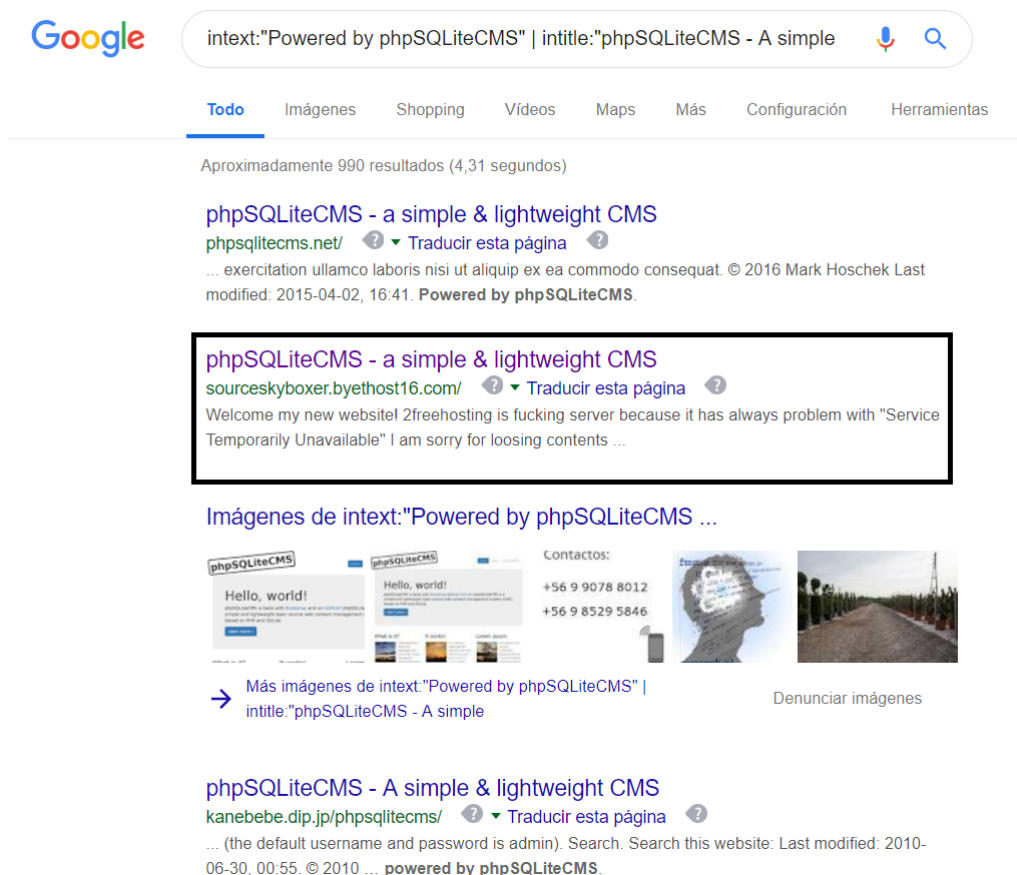


Figura 9: Búsqueda en Google: intext:"Powered by phpSQLiteCMS" | intitle: 'phpSQLiteCMS - A simple & lightweight CMS'.

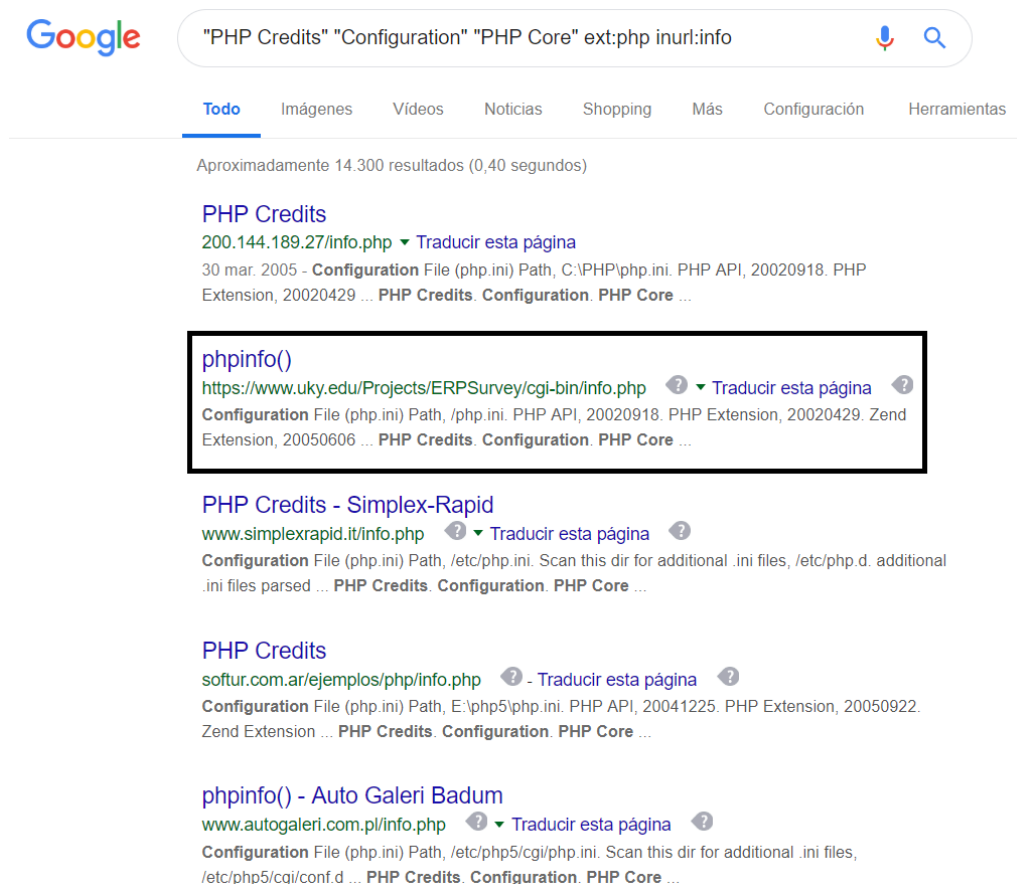


Figura 10: Búsqueda en Google: ‘‘PHP Credits’’ ‘‘Configuration’’ ‘‘PHP Core’’ ext:php inurl:info.

A continuación, usaremos el dork con título ‘‘PHP Credits’’ ‘‘Configuration’’ ‘‘PHP Core’’ ext:php inurl:info (situado en la página 3). Haremos clic sobre el enlace que aparece en la Figura 10 con dirección <http://www.uky.edu/Projects/ERPSurvey/cgi-bin/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](http://inurl:phpsysinfo/index.php?disp=dynamic) (situado en la página 3). Esto nos devolverá un resultado como el de la Figura 11, accederemos al enlace: [http:](http://inurl:phpsysinfo/index.php?disp=dynamic)

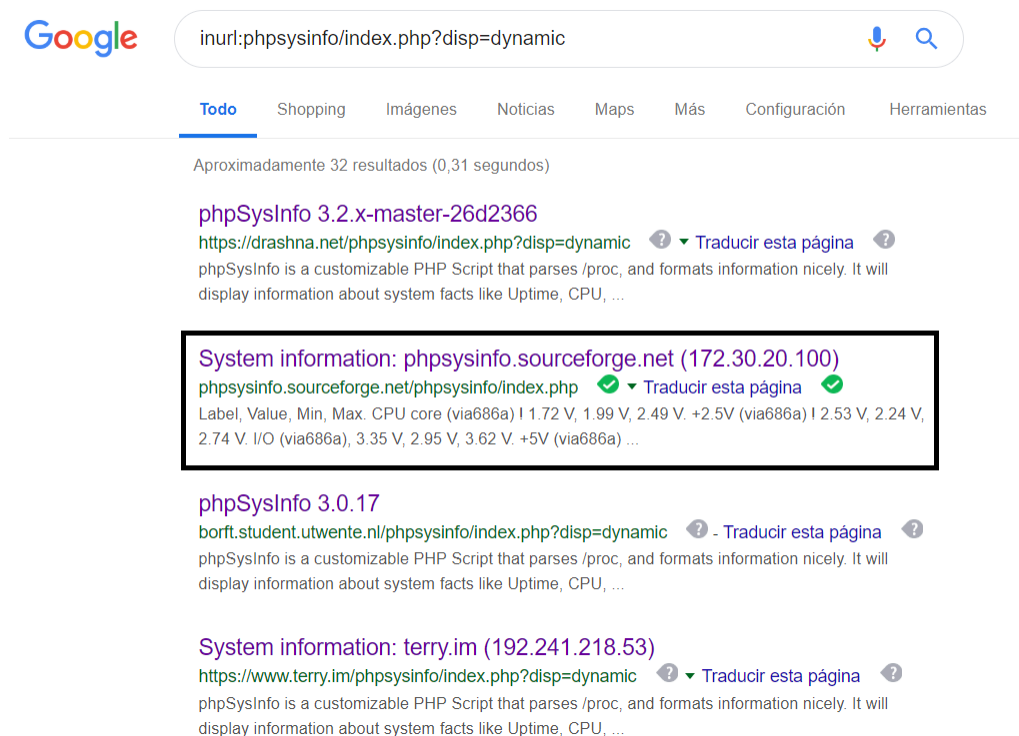


Figura 11: Búsqueda en Google: `inurl:phpsysinfo/index.php?disp=dynamic`.

`//phpsysinfo.sourceforge.net/phpsysinfo/index.php`.

Conteste a las siguientes preguntas:

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

### 3.4. Mensaje de Error

El propósito es encontrar archivos con mensajes de error de los cuales se puedan deducir muchas más cosas. Si encontramos este tipo de información sobre una web, seguramente esa web tenga muchos más fallos de seguridad. Para realizar este tipo de búsquedas:

`“Warning: mysql_query()” “invalid query” -foro -help -ayuda -como.`

Es muy importante eliminar esos términos de la búsqueda para que no nos salgan foros sobre cómo arreglar esos fallos.



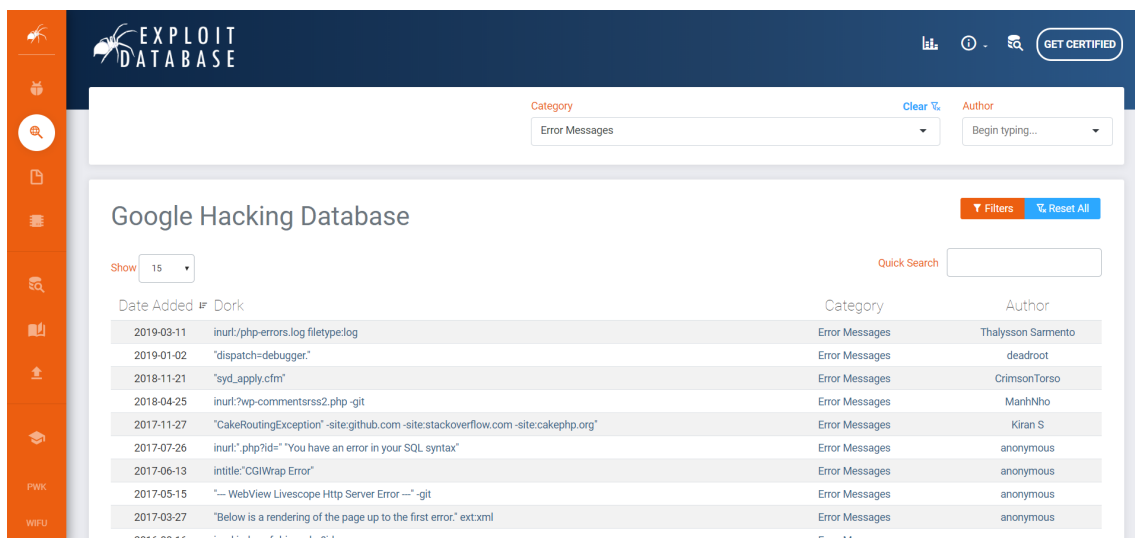


Figura 12: Base de datos de Google dorks para mensajes de error.

Podemos encontrar dorks sobre esta categoría en el siguiente enlace: <https://goo.gl/K48Tcw>.

### 3.4.1. Ejercicio 4

Accedemos al enlace citado anteriormente, que nos llevará al índice de dorks sobre esta temática. Puesto que el uso de los distintos dorks de esta categoría nos conducen a resultados similares, usaremos en este caso solo dos de ellos.

Haremos clic en el décimo dork cuyo título es `inurl:index of driver.php?id=` y en el resultado accederemos al enlace que aparece en la Figura 13, cuya URL es <http://www.yaokyo.net/index.php?id=40>.

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:/smpwservices.fcc | '/lm_private/CkeSetter.aspx'` (situado en la página 2), y accederemos al enlace de la Figura 14 con URL <https://eportal.pwc.ca/siteminderagent/forms/smpwservices.fcc>.

Conteste a las siguientes preguntas:

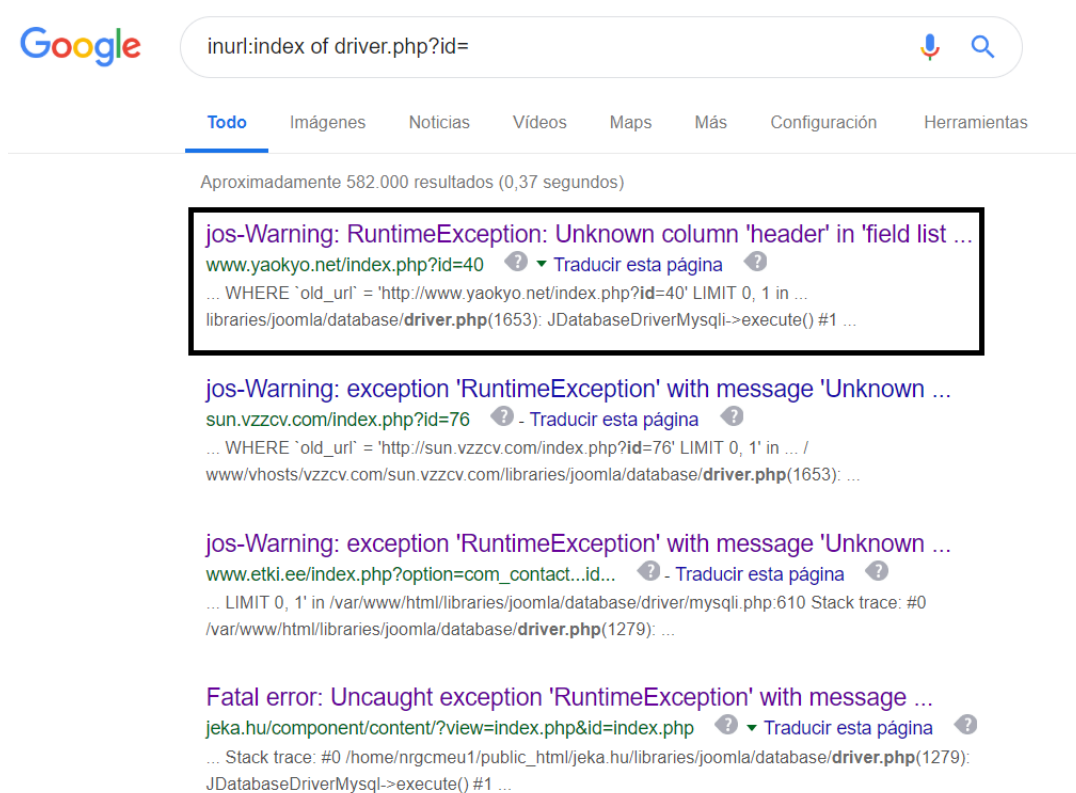


Figura 13: Búsqueda en Google: inurl:index of driver.php?id=

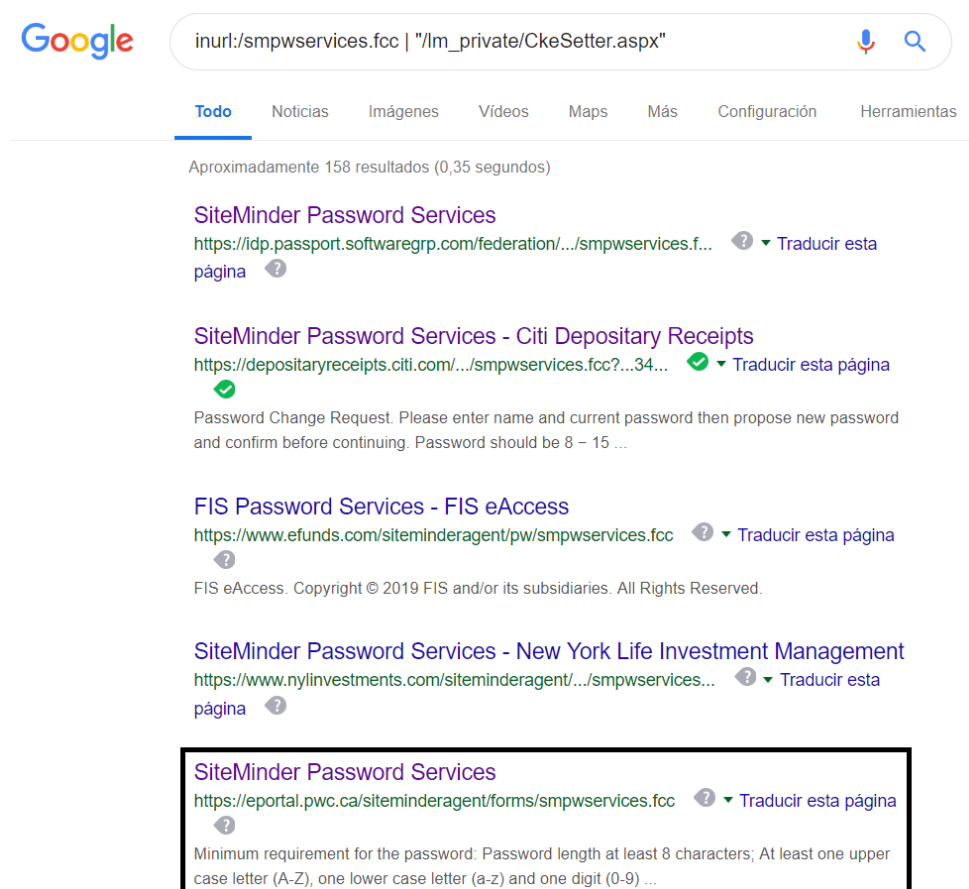


Figura 14: Búsqueda en Google: `inurl:/smpwservices.fcc | "/code>`

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

### 3.5. Escaneo de Servidores Vulnerables

En esta categoría se encuentran todos los servidores con *backdoors* y otras vulnerabilidades. Lo curioso es que al encontrar cualquier servidor con este tipo de vulnerabilidad nos permitirá subir un nc y arrancar nuestra propia shell.

Podemos encontrar dorks sobre esta categoría en el siguiente enlace: <https://google.com/search?q=+intitle:indexof+inurl:.gov>

### 3.5.1. 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.

Tras acceder al enlace anteriormente citado llegaremos a la página mostrada en la Figura 15.

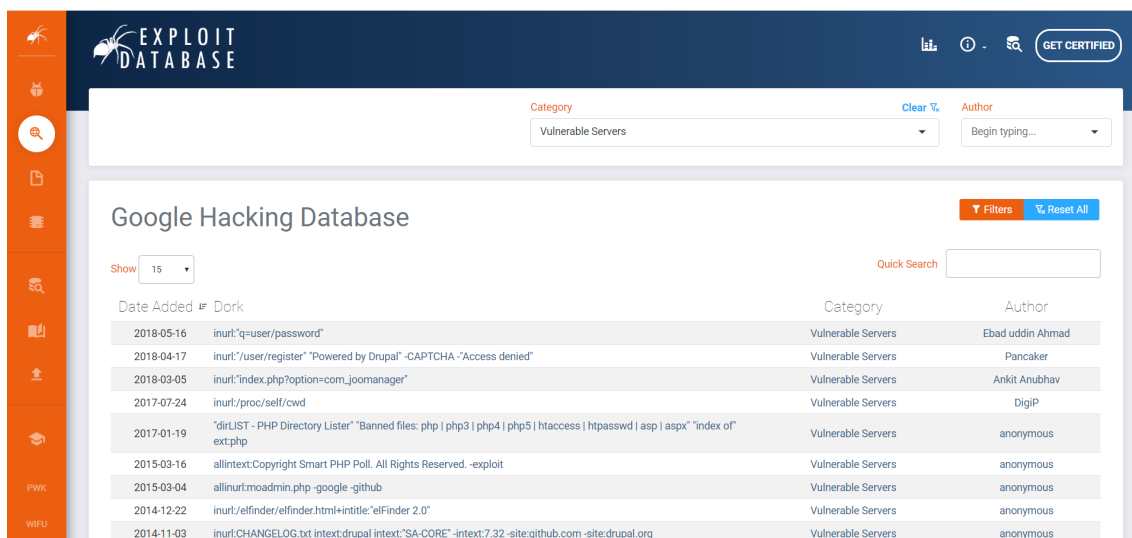
Haremos click en el quinto dork cuyo título es ‘‘dirLIST - PHP Directory Lister’’, ‘‘Banned files: php | php3 | php4 | php5 | htaccess | httpasswd | asp | aspx’’, ‘‘index of’’ ext:php y en el resultado accederemos al enlace que aparece en la Figura 16, cuya URL es <http://www.jeeptelelevision.com/fotoeventi/index.php?folder=c21jaWxpYQ==>.

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?

### 3.6. Búsqueda de Información Sensible

En esta categoría englobamos información que podría resultar sensible como DNIs, documentos no públicos del gobierno, o información cuyo dueño no imaginaba que sería pública. Para esto, debemos realizar una búsqueda como la siguiente:



The screenshot shows the Exploit Database interface. At the top, there's a navigation bar with the Exploit Database logo and a 'GET CERTIFIED' button. Below it, a search bar is set to 'Vulnerable Servers'. The main content area is titled 'Google Hacking Database' and shows a list of search results. The results are displayed in a table with columns for Date Added, Dork, Category, and Author.

Date Added	Dork	Category	Author
2018-05-16	inurl:'q=user/password'	Vulnerable Servers	Ebad uddin Ahmad
2018-04-17	inurl:'/user/register' "Powered by Drupal" -CAPTCHA -Access denied"	Vulnerable Servers	Pancaker
2018-03-05	inurl:'index.php?option=com_joomanager'	Vulnerable Servers	Ankit Anubhav
2017-07-24	inurl:/proc/self/cwd	Vulnerable Servers	DigiP
2017-01-19	"dirLIST - PHP Directory Lister" "Banned files: php   php3   php4   php5   htaccess   httpasswd   asp   aspx" "index of" ext:php	Vulnerable Servers	anonymous
2015-03-16	allintext:Copyright Smart PHP Poll. All Rights Reserved. -exploit	Vulnerable Servers	anonymous
2015-03-04	allinurl:moadmin.php -google -github	Vulnerable Servers	anonymous
2014-12-22	inurl:/elfinder/elfinder.html+intitle:"elFinder 2.0"	Vulnerable Servers	anonymous
2014-11-03	inurl:CHANGELOG.txt intext:drupal intext:"SA-CORE" -intext:7.32 -site:github.com -site:drupal.org	Vulnerable Servers	anonymous

Figura 15: Base de datos de Google dorks para servidores vulnerables

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

Podemos encontrar dorks sobre esta categoría en el siguiente enlace: <https://goo.gl/1crsVb>.

### 3.6.1. Ejercicio 6

Tras acceder al enlace citado anteriormente nos mostrará el resultado que vemos en Figura 17.

De los dorks que encontramos en el enlace usaremos dos, el primero será el que tiene como título `allinurl: drive.Google.com/open?id=` (situado en la página 15), lo que nos llevará a los siguientes resultados, que podemos ver en la Figura 18.

Con este tipo de dork podríamos acceder, por ejemplo, al enlace con título Ejercicios Oposición y URL <https://goo.gl/Cswghd>.

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` (situado en la página 16), lo que nos

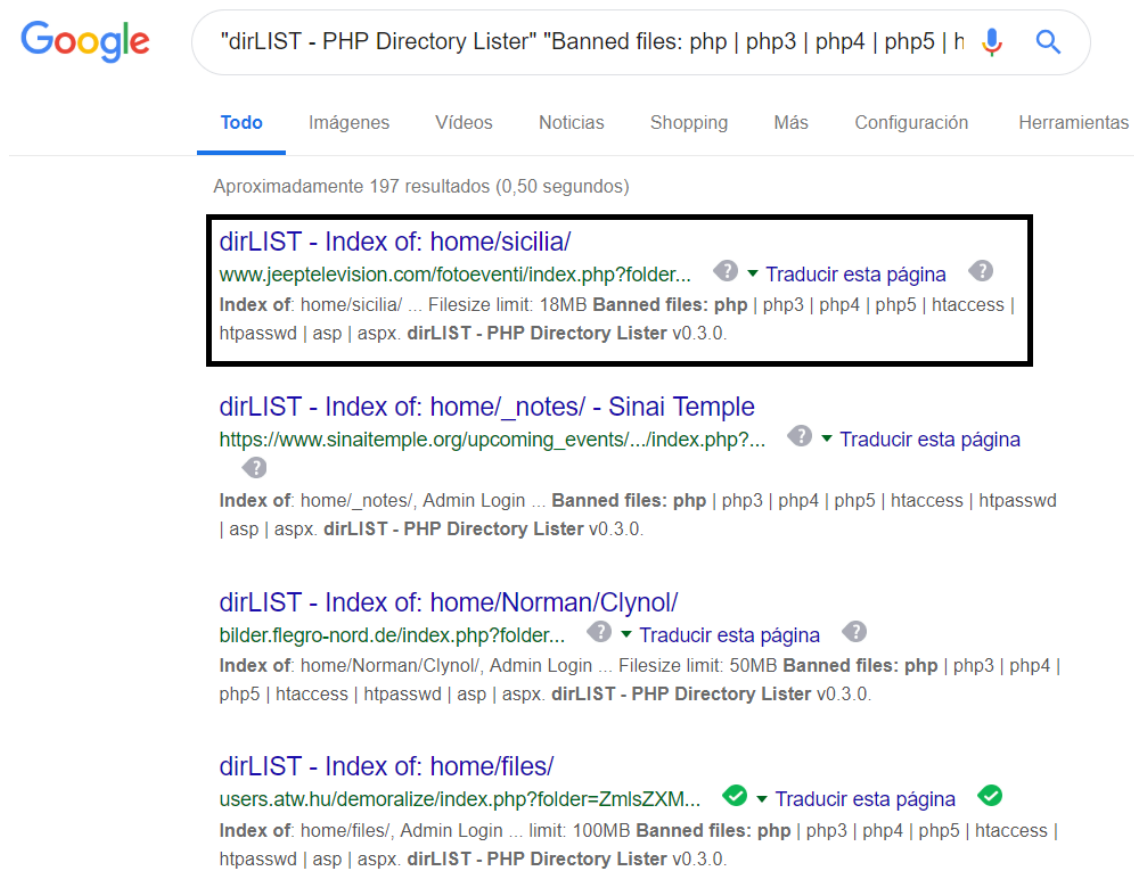


Figura 16: Búsqueda en Google: “dirLIST – PHP Directory Lister” “Banned files: php | php3 | php4 | php5 | htaccess | httpasswd | asp | aspx” “index of” ext:php

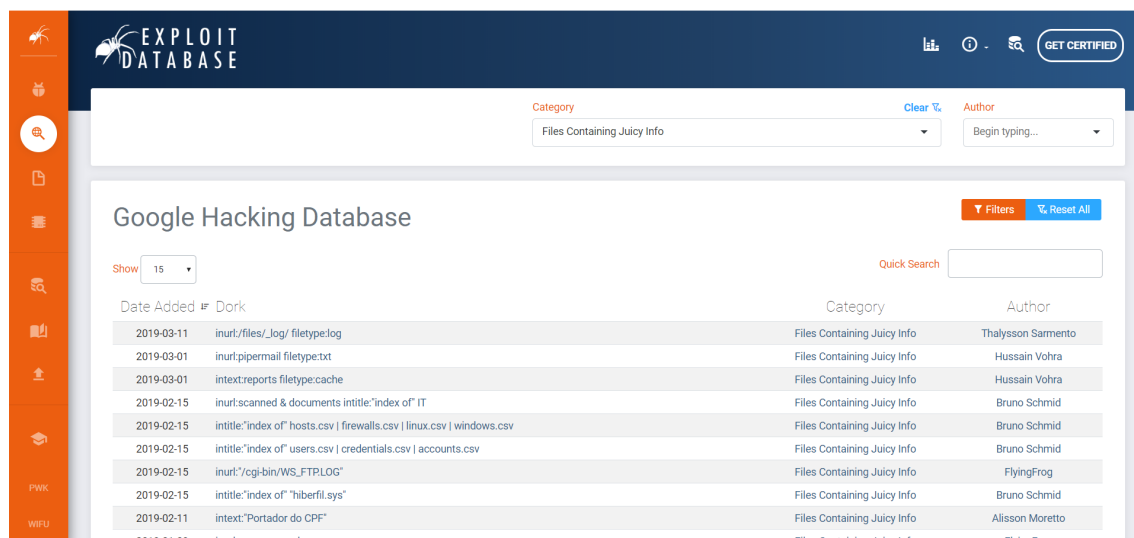


Figura 17: Base de datos de Google dorks para información jugosa

dará un resultado como el de la Figura 19. Accederemos al enlace <https://goo.gl/xynyvM>.

Conteste a las siguientes preguntas:

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

Por último, introduciremos manualmente el ejemplo descrito arriba en el cuadro de búsqueda de Google. Esto nos llevará a un resultado como el de la Figura 20.

Conteste a las siguientes preguntas:

- ¿A qué estamos accediendo?

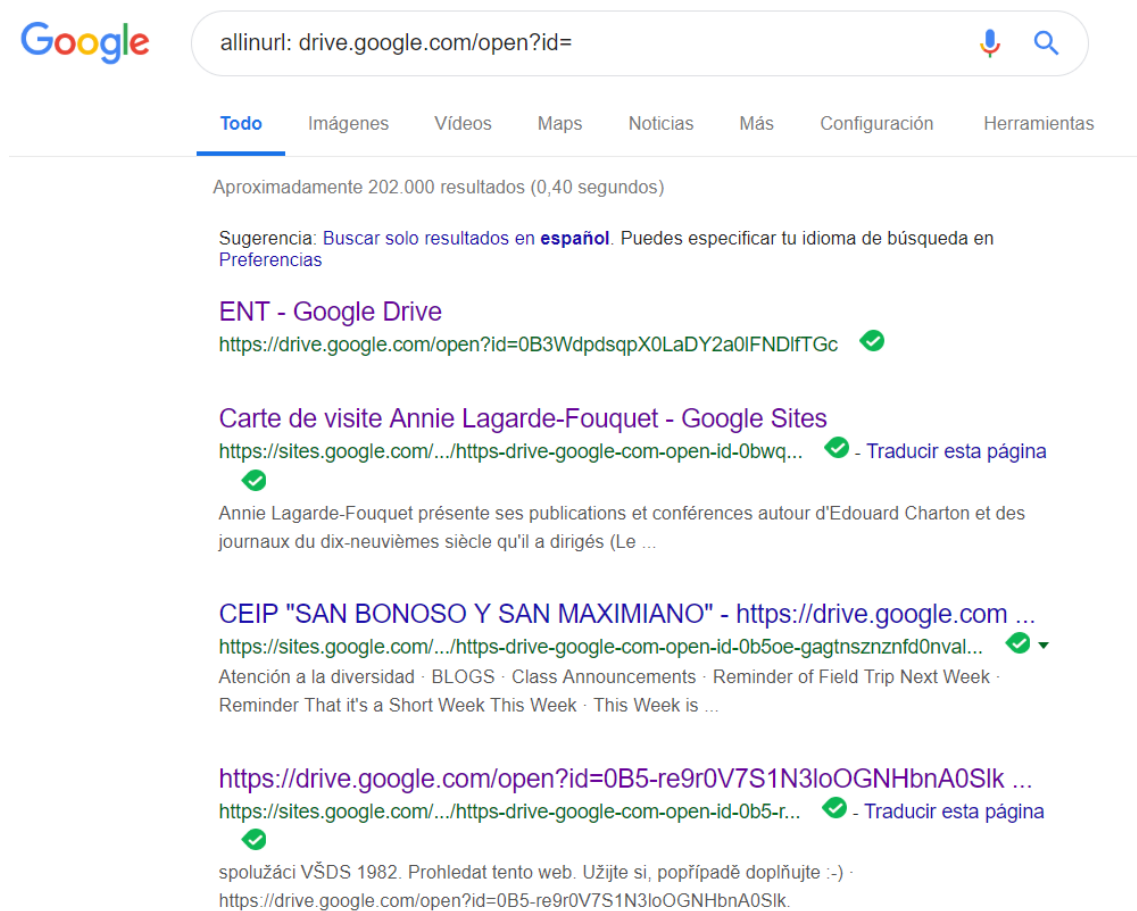


Figura 18: Búsqueda en Google: allinurl: drive.Google.com/open?id=



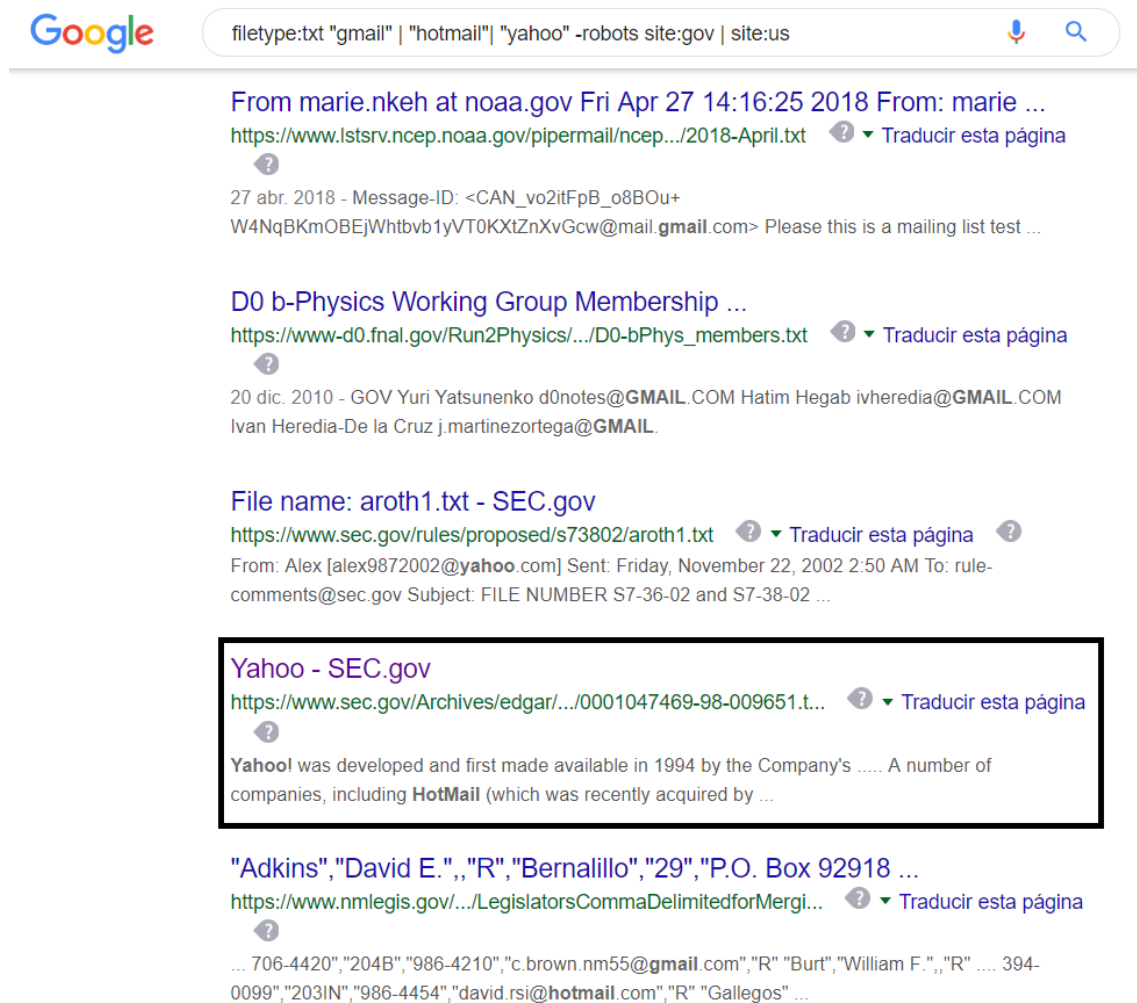


Figura 19: Búsqueda en Google: `filetype:txt "gmail" | "hotmail" | "yahoo" -robots site:gov | site:us`

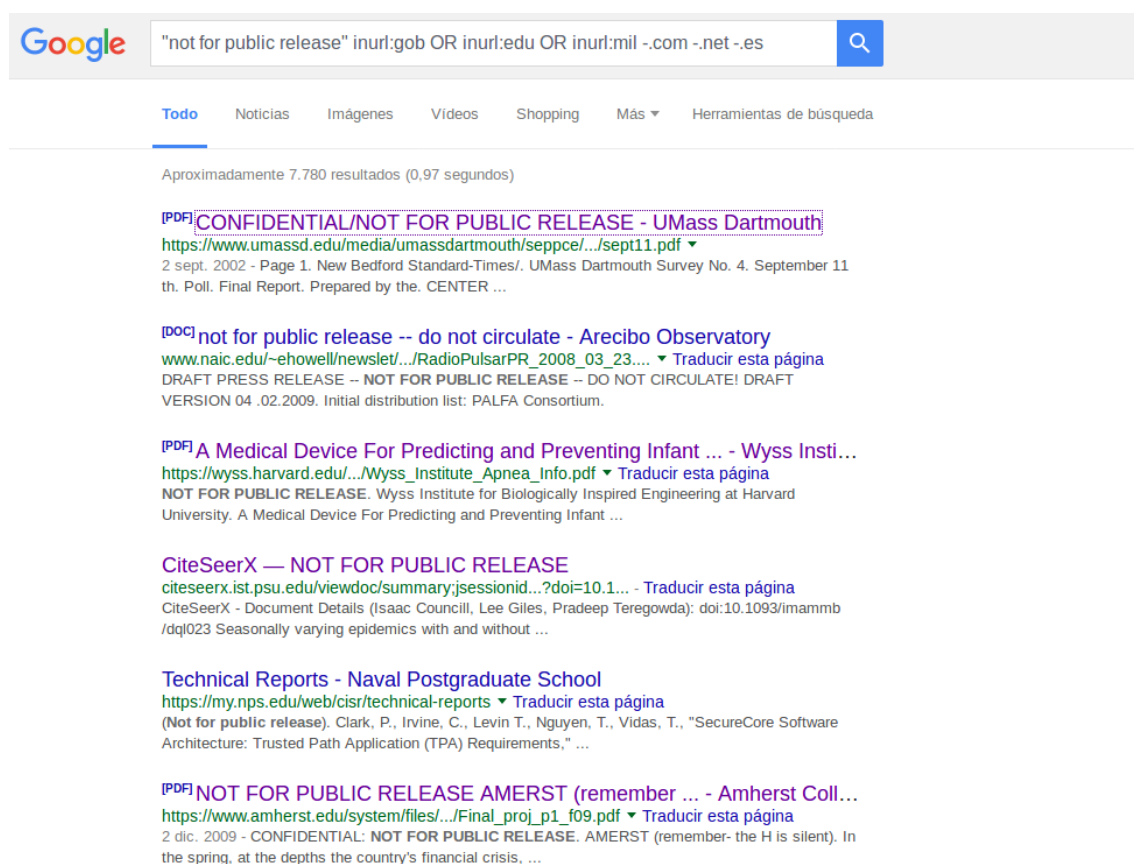


Figura 20: Búsqueda en Google: ‘‘not for public release’’ inurl:gob OR inurl:edu OR inurl:mil -.com -.net -.es

## Referencias

- [1] Pablo González Pérez, Germán Sánchez Garcés y Jose Miguel Soriano de la Cámara: *Pentesting con Kali 2.0*. 0xWORD, 2015, ISBN 978-84-608-3207-2.
- [2] Enrique Rando: *Hacking con buscadores: Google, Bing and Shodan + Robtex*. 0xWORD, 3ª edición, 2015, ISBN 978-84-616-7589-0.