Informe de la quinta actividad GSI

@ALVPI

Contents

1 Objetivo general

Hemos sido contratados por *"Teletalk"* para realizar una auditoría sobre la siguiente URL y sobre un posible acceso a su BDMS(Sistema de administración de Bases de Datos) empleado en la URL.
Para realizar esta labor, vamos a emplear la metodología **OWASP**.

2 Identificación de vulnerabilidades

Comenzamos intentando acceder directamente a la URL:

> **You IP 157.88.139.133 is blocked for trying to login with invalid username more than 3 times.Contact with Teletalk Admin through proper channel.**

Figure 1: Presencia de un filtro por IP.

Cambiamos de eduroam a nuestros datos móviles y volvemos a intentar acceder a la URL, encontrándonos con la interfaz de la página web.

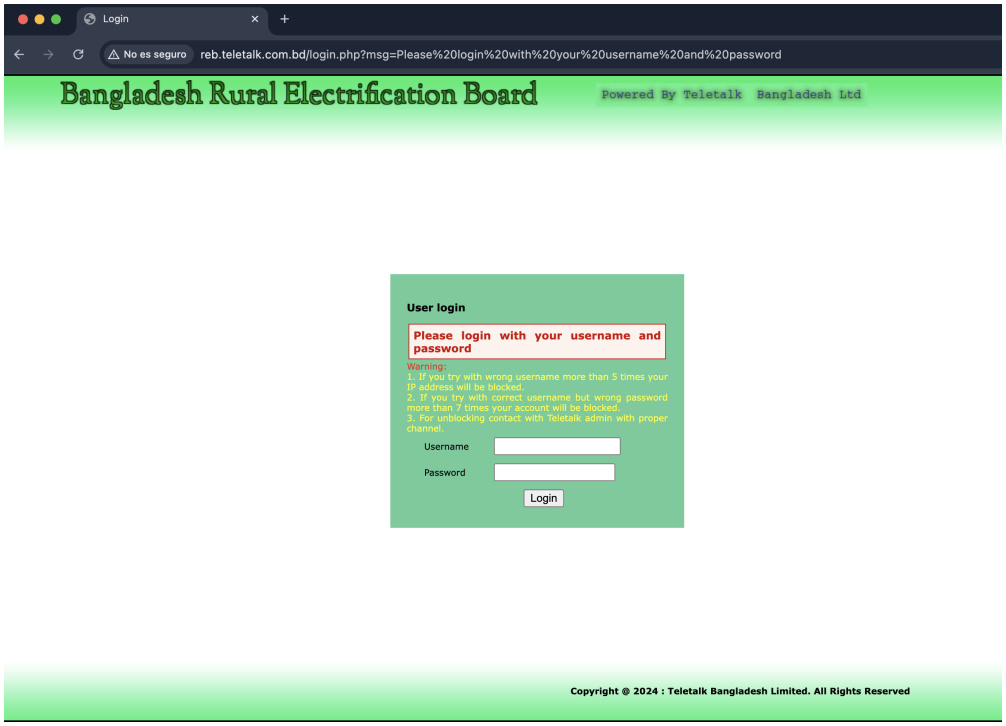


Figure 2: Interfaz del objetivo del pentesting.

Nos encontramos con un cuadro de texto donde se nos solicita un usuario y contraseña para acceder al sistema y que además nos indica el número de intentos máximos que tenemos antes de que nuestra ip sea baneada, esto nos indica que **exite una medida de protección contra ataques de fuerza bruta.**

2.1 NMAP

Lo siguiente que debemos hacer es realizar un escaneo de nmap para saber qué servicios están activos en nuestro objetivo.
En el caso de no tener nmap instalado, tendremos que utilizar el gestor de paquetes del sistema para instalarlo, en mi caso: `brew install nmap`

```
~/Desktop git:(master)±121 (2m 16.69s)
nmap -Pn -sV reb.teletalk.com.bd

Starting Nmap 7.95 ( https://nmap.org ) at 2024-12-12 16:16 CET
Nmap scan report for reb.teletalk.com.bd (103.230.104.201)
Host is up (0.24s latency).
Not shown: 985 closed tcp ports (conn-refused)
PORT     STATE    SERVICE         VERSION
21/tcp   open     ftp             vsftpd 2.2.2
80/tcp   open     http            Apache httpd 2.2.15 ((Red Hat))
89/tcp   open     http            Apache httpd 2.2.15 ((Red Hat))
90/tcp   open     http            Apache httpd 2.2.15 ((Red Hat))
139/tcp  filtered netbios-ssn
443/tcp  open     ssl/https?
646/tcp  filtered ldp
5666/tcp open     tcpwrapped
5989/tcp open     ssl/wbem-https?
8080/tcp open     http            Apache httpd 2.2.15 ((Red Hat))
8081/tcp open     http            Apache httpd 2.2.15 ((Red Hat))
8083/tcp open     http            Apache httpd 2.2.15 ((Red Hat))
8443/tcp open     ssl/https-alt?
8888/tcp open     http            Apache httpd 2.2.15 ((Red Hat))
9999/tcp open     http            Apache httpd 2.2.15 ((Red Hat))
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 136.58 seconds

~/Desktop git:(master)±121
```

Figure 3: Resultados de emplear nmap sobre la url

Si nos fijamos, nos damos cuenta de que en el puerto *443/tcp* que emplea https, nos encontramos con que está abierto pero en servicio aparece *ssl/https?* lo que indica que la web no está preparada para recibir peticiones de este tipo.
Podemos comprobar esto, forzando a que se use https para la conexión escribiendo en la url `https://reb.teletalk.com.bd/`

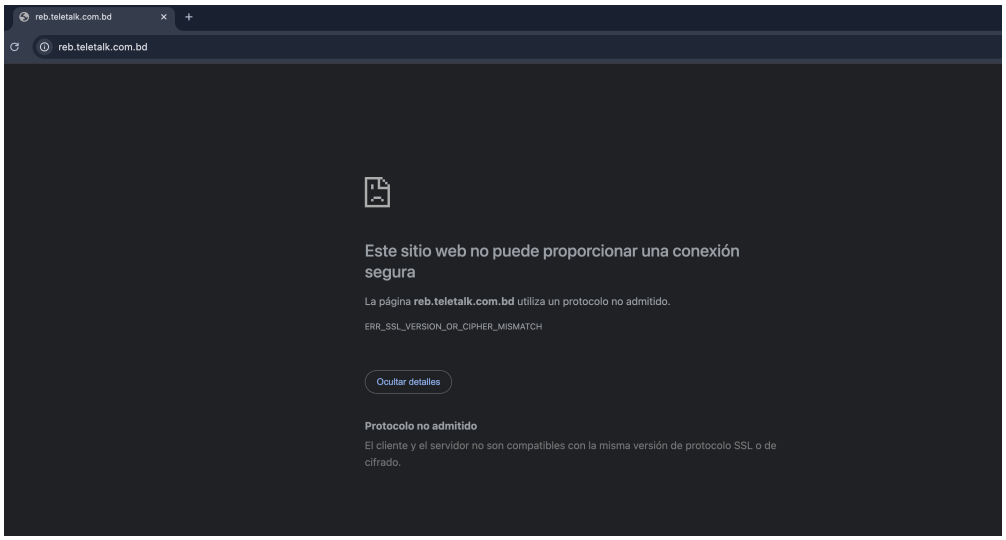


Figure 4: No soporta el protocolo https

Esto constituye la primera vulnerabilidad del sitio web, puesto que al no emplear https, las credenciales o información que se realice entre cliente y servidor va a ir en texto plano, por lo que cualquier atacante puede aprovechar esto para conseguir información sensible.
También nos podríamos haber dado cuenta de que el navegador (en mi caso Chrome) clasifica la conexión con la página como no segura (puesto que usa http).
Si nos seguimos fijando en la captura de nmap, nos damos cuenta de que la versión que está utilizando de Apache es la 2.2.15, si hacemos una consulta rápida en ExploidDataBase.

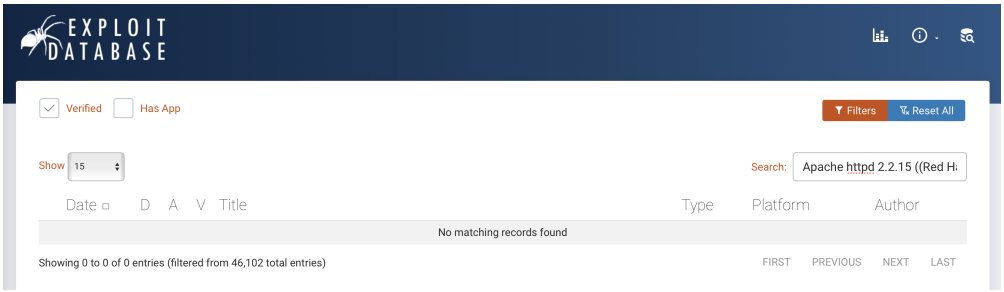


Figure 5: Listado de vulnerabilidades verificadas reportadas

Por lo que la captura de nmap ya no nos ofrece más información relevante.

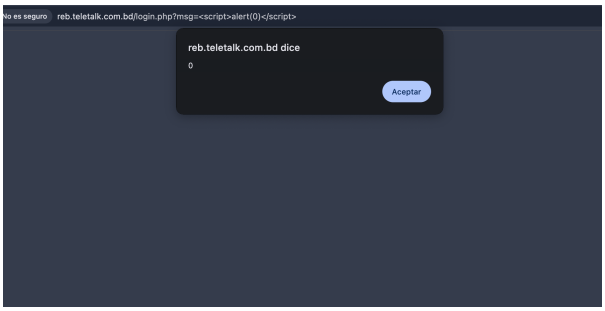
2.2 Inyecciones

Al revisar la url que se genera cuando nos conectamos a la página web, nos damos cuenta de que casualmente, el texto que aparece al final de la url es el que aparece en el recuadro que se encuentra justo encima de la sección de autenticación. Vamos a ver qué ocurre si lo modificamos:

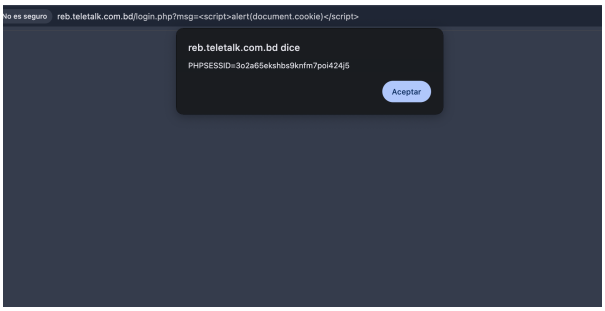
Esto es un indicador que nos permite afirmar que podemos inyectar código en la página web.
Hay que tener en cuenta que este no es un entorno controlado, lo que supone que podemos llegar a tener problemas serios, por lo que vamos a descartar inyecciones que puedan incurrir en delito y no llevaremos a cabo ningún ataque para verificar hasta dónde podemos llegar con las mismas.

2.2.1 Cross-Site Scripting(XSS)

Estamos comprobando si el sitio web nos permite generar y ejecutar código *JavaScript*



(a) alert(0) genera un mensaje de error en el navegador al querer acceder al sitio web.

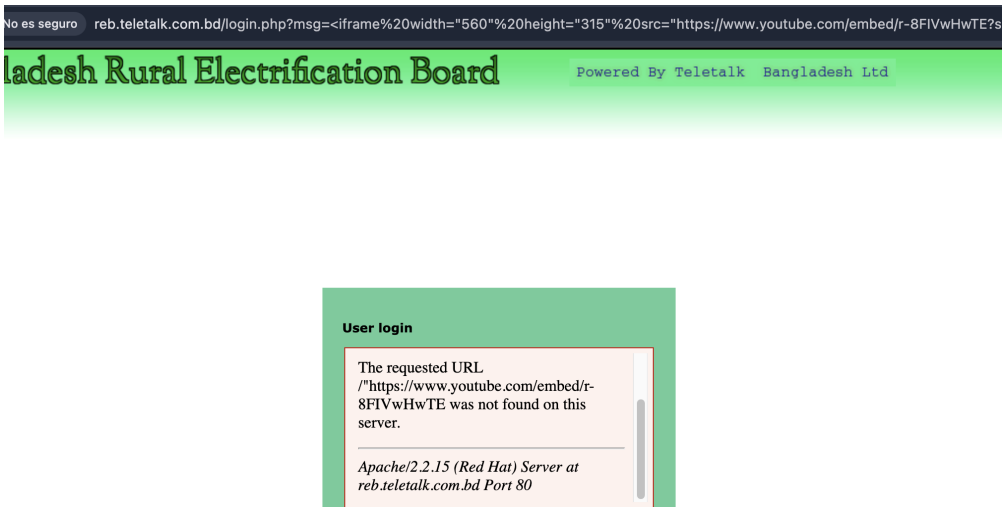


(b) Muestra el contenido de las cookies que están siendo empleadas por el sitio web.

Figure 6: Inyecciones intentadas

Al saltarnos estos popups, demostramos que esta vulnerabilidad está presente.

2.2.2 Inyección HTML



En este caso, lo que estamos haciendo es inyectar código html, y nos damos cuenta de que efectivamente el vídeo no se muestra, pero a la vez conseguimos información de la versión de apache que está utilizando la página web a la vez que sabemos que tiene algún tipo de protección contra los iframes, es decir, las incrustación de documentos dentro de la página.
Pero no quita que otro tipo de inyección HTML no surta efecto.

2.3 Vulnerabilidades encontradas

Nombre vulnerabilidad	Breve explicación
Fallo criptográfico	La web emplea un protocolo inseguro puesto que la información viaja a través de la red como texto plano
Inyecciones	A través de la modificación de la url podemos ejecutar código malicioso.

Table 1: Resumen de las vulnerabilidades encontradas, teniendo en cuenta que no es un entorno de práctica.

3 Justificación técnica

El objetivo de la práctica es hacer un pentesting empleando las técnicas recomendadas por la OWASP (Open Web Application Security Project), la cual es una organización sin fines de lucro dedicada a mejorar la seguridad del software y las aplicaciones web, que se encarga de concienciar y mejorar la seguridad de las webs.
La cual presenta un ranking sobre los principales riesgos que sufren las Webs, lo cual es una guía muy útil a la hora de saber qué ataques intentar sobre las mismas para realizar el pentesting.
En concreto hemos consultado el top 10 de riesgos.

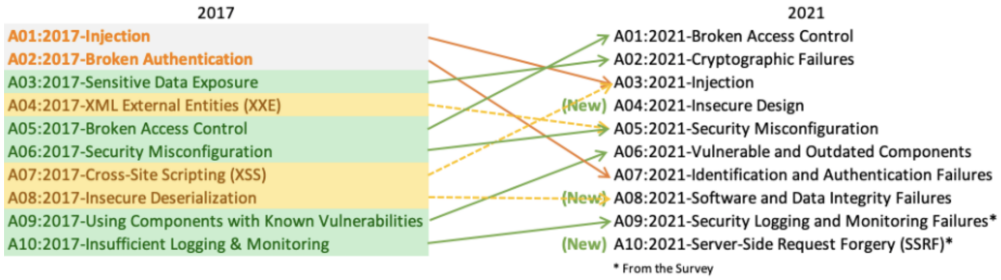


Figure 7: Los 10 ataques con mayor probabilidad de éxito.

Hemos decidido utilizar el top 2 y 3 de la lista.
De este listado, debido a que no es un sitio controlado, la gran mayoría de estas vulnerabilidades no se pueden comprobar sin incurrir en un delito, puesto que es necesario vulnerar la web para poder comprobarlas.

4 Análisis de acceso al DBMS

En este apartado, lo que nos están pidiendo es ver si el servidor de base de datos tiene un acceso directo con internet, lo que incurre en una vulnerabilidad muy grave.
Al no saber exactamente el tipo de servidor que se está utilizando en esta máquina, lo que haremos será realizar un nmap sobre los puertos estándar de los principales servidores.

Servidor de bases de datos	Puerto por defecto
MySQL	3306
Oracle Database	1521
Microsoft SQL Server	1433
PostgreSQL	5432

Table 2: Listado con los principales puertos donde están alojados los servicios de bases de datos.
Realizamos el escaneo de puertos con nmap:

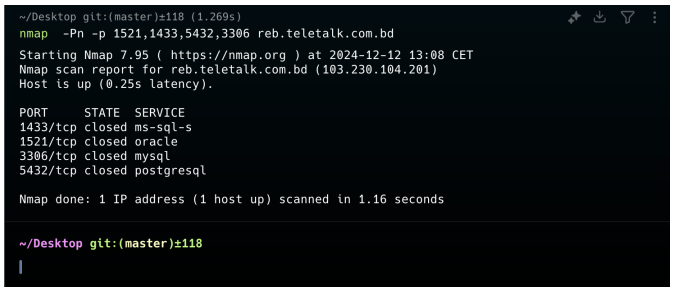


Figure 8: Estado de los principales servidores de bases de datos.

En el caso de que la captura de nmap muestre que alguno de esos puertos se encuentra abierto (esto puede ocurrir si se está empleando una vpn) no quiere decir que haya un acceso directo, puesto que puede ser una "trampa" la cual cumple una doble función, hacer perder el tiempo al atacante y recolectar información tanto del mismo como de las técnicas que está empleando para intentar vulnerar la integridad del servicio en dicho puerto.
Con esta captura podemos determinar que no existe un acceso directo a la base de datos de la web.