

**INFORME: EJECUTIVO Y TÉCNICO**

**Resultados obtenidos explotando vulnerabilidades**

**Sistema Fuzzing**

* Fecha: 24 de julio de 2024
* Cliente: Reto 7
* Consultora de Ciberseguridad: The Bridge - Accelerator
* Control de Cambios

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Versión | Documento | Fecha | Cambios | Autor | revisor | visto bueno |
| 1.1 | Informe de resultados | 03/07/2024 | Informe inicial | Victor Martínez | Ángel  Cardiel | Javier Tomás |

**Índice de Contenidos**

1. Introducción ----------------------------------------------------------------------------------3
2. Informe Ejecutivo ----------------------------------------------------------------------------3

* Introducción ------------------------------------------------------------------------------3
* Alcance ------------------------------------------------------------------------------------5
* Resumen de Actuaciones Practicadas --------------------------------------------5
* Recomendaciones generales -------------------------------------------------------6
* Normativa aplicable y sanciones ---------------------------------------------------7

1. Informe Técnico: ----------------------------------------------------------------------------9

* Explotación del sistema y vulnerabilidades --------------------------------------9
* Conclusiones ---------------------------------------------------------------------------19

1. Bibliografía ----------------------------------------------------------------------------------20
2. Anexos ---------------------------------------------------------------------------------------21

1. **INTRODUCCIÓN**

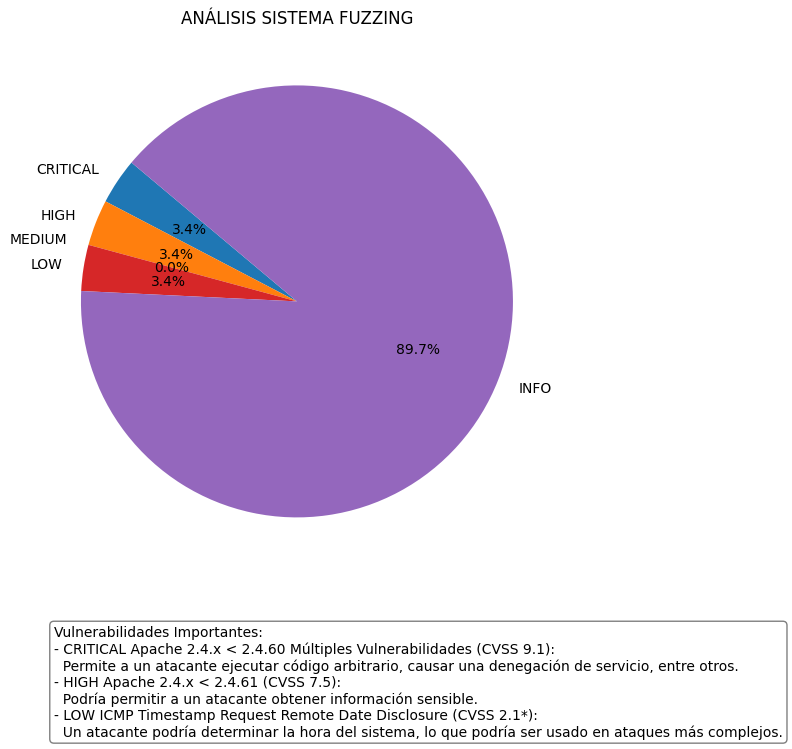
El presente informe está conformado por 2 partes: un informe ejecutivo, menos técnico y dirigido a informar a los latos cargos o ejecutivos de la compañía, y un informe técnico dirigido a los analistas de ciberseguridad y programadores que tengan que ejecutar las tareas para mitigar las vulnerabilidades explotadas, para mejorar los manuales de estrategia de la compañía para la detección, contención y respuesta ante incidentes críticos en su sistema.

2. **INFORME EJECUTIVO**

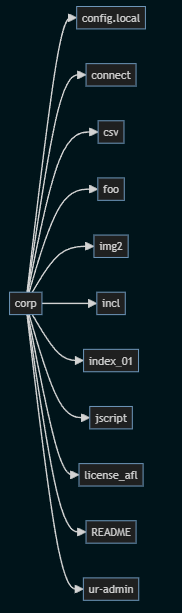
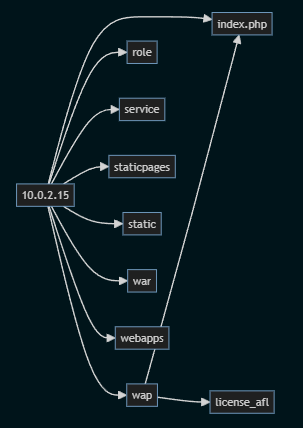
1. Introducción. – Este informe tiene como objetivo presentar los resultados de las vulnerabilidades detectadas y explotadas en el equipo Fuzzing, de acuerdo con el contrato firmado entre ambas partes, en el que permiten la explotación del sistema con la finalidad de conseguir la autenticación por atacantes externos con usuarios con privilegios root. El equipo no tiene entorno grafico y apara acceder en línea de comandos hace falta una clave y contraseña que no aportan y se han usado para su explotación diversas herramientas de ciberseguridad, destacando alguna de ellas:

* **Nessus Essentials.** - Herramienta de escaneo de vulnerabilidades más populares y completas en el ámbito de la seguridad informática, que se utiliza para identificar vulnerabilidades en sistemas y redes, detectar configuraciones incorrectas y posibles puntos de entrada para ataques en una amplia gama de plataformas, clasificando estas en críticas, altas, medias e info.

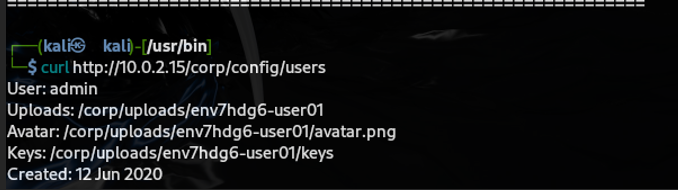
En este ataque no ha sido necesario explotar ninguna de las vulnerabilidades detectadas, siendo las más importantes, las que figuran al pie de la siguiente figura, donde se pueden observar los porcentajes de las vulnerabilidades encontradas, siendo los resultados muy buenos en general.

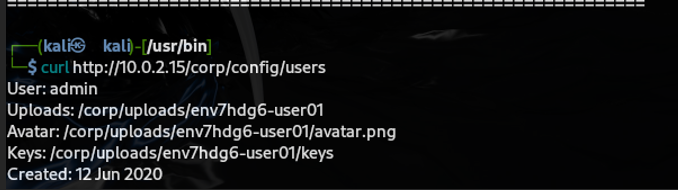


* **Dirb. -** Herramienta de seguridad y hacking web, comúnmente utilizada durante las fases de reconocimiento en pruebas de penetración y auditorías de seguridad.



Como se puede observar hay un directorio con amplia información del sistema, llamado *“Corp”,* donde se muestra la configuración local del sistema, donde se extraen credenciales de usuarios con privilegios root, causando un riesgo critico al sistema, en el caso que, atacantes maliciosos consiguieran acceder al sistema por esta vía, cómo se puede observar en las siguientes imágenes:





1. Alcance. - El alcance se ha centrado en identificar y evaluar las debilidades de seguridad en el sistema, para lograr las finalidades expuestas en el contrato, explotando algunas de las vulnerabilidades encontradas, que pueden causar daños en el sistema, así como comprometer la integridad, confidencialidad y disponibilidad de los datos del mismo.
2. Resumen de actuaciones practicadas. – Se han realizado numerosas actuaciones, explotando ciertas debilidades / vulnerabilidades detectadas, algunas de las cuales han sido comentadas anteriormente, consiguiendo finalmente el objeto del contrato, es decir, la autenticación con usuario con privilegios root en el sistema.
3. Recomendaciones generales

En el análisis efectuado de vulnerabilidades con el programa Nessus, se han encontrado muy pocas vulnerabilidades importantes, por lo que podría estar dentro de los riesgos permitidos dentro de las políticas de seguridad de ciertas empresas. No obstante, se recomienda actualizar, si es el caso, dicha política al modelo “Zero Trust”[[1]](#footnote-1).

Por otro lado, tener archivos (robots.txt) que incluyan directorios que contengan información confidencial, como credenciales de usuarios, representa un riesgo significativo de seguridad para la empresa.

¿Por qué es peligroso?

* **Acceso a información confidencial:** Los robots de búsqueda o crawlers[[2]](#footnote-2), los cuales, se encargan de recopilar información de las páginas web, pueden acceder a directorios que contiene información confidencial, como es el caso del archivo *“robots.txt”.* Este archivo funciona como una directiva para los rastreadores indicando las partes indexables de un sitio web, sin embargo, a nivel práctico, algunos robots podrían tener acceso a estos directorios bloqueados (crawlers maliciosos o ilegales), pudiendo llegar a acceder a esa información, exponerla públicamente con diferentes fines y, como el caso concreto de este informe, para explotación de las vulnerabilidades de un sistema.
* **Escalada de privilegios:** Si las credenciales de usuario con privilegios de root se almacenan en archivos dentro de directorios bloqueados por *“robots.txt”,* un atacante podría obtener acceso a esos archivos y obtener acceso de root al sistema, obteniendo un control total sobre el sitio web y el servidor, lo que podría tener consecuencias devastadoras.

Para evitar los riesgos asociados de tener información confidencial en un archivo robots.txt, se recomienda seguir estas prácticas:

* No almacenar nunca información confidencial en directorios que estén bloqueados por archivos *“robots.txt”.*
* Utilizar métodos más seguros para proteger la información confidencial con un cifrado adecuado.
* Realizar auditorías de seguridad periódicas para identificar y corregir vulnerabilidades.
* Mantener el software y los sistemas actualizados.
* Realizar cursos de capacitación a los empleados en materia de seguridad de la información.

1. Normativa aplicable y sanciones

Existen diversas normativas que regulan la protección de datos y la seguridad de la información, y que podrían ser aplicables en este caso:

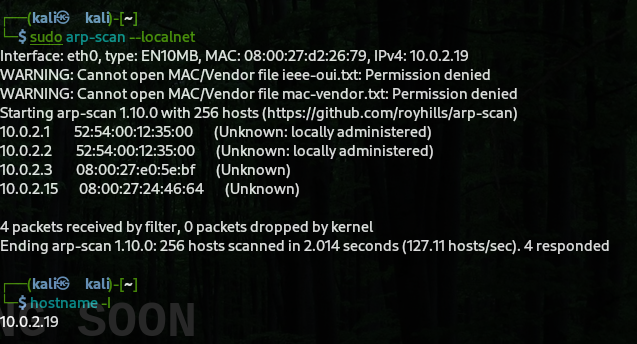
* **Reglamento General de Protección de Datos (RGPD)[[3]](#footnote-3) y la Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD)[[4]](#footnote-4)**. - Si la información confidencial que se encuentra en los directorios bloqueados, incluye datos personales, su incumplimiento podría acarrear sanciones importantes para la empresa.
* Ley de Servicios de la Sociedad de la Información y el Comercio Electrónico (LSSI**)[[5]](#footnote-5). -** Los prestadores de servicios (corporaciones, empresas, etc) deben adoptar las medidas técnicas y organizativas necesarias para garantizar la seguridad de los datos de los usuarios, pudiendo su incumplimiento acarrear sanciones para la empresa.

Las sanciones por el incumplimiento de las normativas de protección de datos y seguridad de la información pueden ser de elevado valor, por ejemplo, en el caso del RGPD, las multas pueden ascender hasta el 4% del volumen de negocio mundial anual de la empresa o 20 millones de euros, lo que sea mayor y en el caso de la LOPDGDD, las multas pueden ascender hasta 300.000 euros.

**3.- INFORME TÉCNICO**

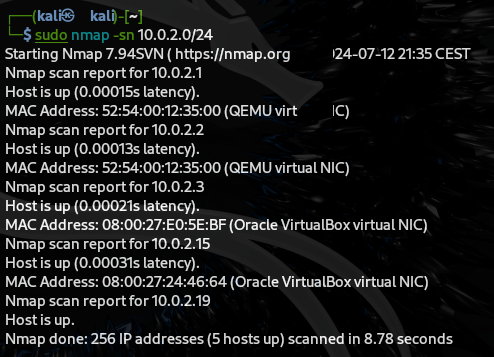
1. Explotación del sistema y vulnerabilidades. – Para conseguir el objetivo fijado en el contrato, se ha seguido la siguiente línea de investigación:

* El Equipo ha sido entregado con un sistema Linux /Debian en un entorno CLI, sin aportar credenciales de inicio de sesión de la maquina Fuzzing, por lo que el análisis y explotación será realizado en caja negra.
* Para esta explotación se ha usado como maquina atacante, un sistema Kali Linux virtualizado, en su versión .2 2024, conectando mediante Red NAT con la maquina objeto del presente.
* En primer lugar, se procede a consultar mediante el comando “*arp-scan”* la IP de nuestra maquina Fuzzing, siendo esta: 10.0.2.15 y la de la maquina atacante: 10.0.2.19.



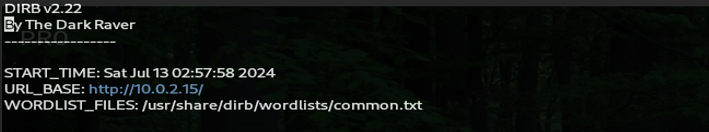
* Mediante nmap (Network Mapper), es una herramienta de seguridad para el descubrimiento de redes y usado en auditorías de seguridad, permitiendo escanear redes y hosts en busca de puertos abiertos, servicios, sistemas operativos y tipos de dispositivos, así como para identificar vulnerabilidades y agujeros de seguridad en las redes.

La herramienta funciona enviando paquetes a los hosts, que pueden variar según el tipo de escaneo (TCP SYN, UDP y ICMP) y analiza las respuestas del servidor para determinar qué puertos están abiertos, qué servicios se están ejecutando y qué sistema operativo se está ejecutando en el host, entre otras. En el caso concreto de la maquina Fuzzing, se ha utilizado con el *flag* “-sn”, el cual no realiza un escaneo de puertos normal, sino que busca dispositivos activos en la red enviando paquetes ICMP (PING), cuyo resultado nos aporta además de las IPs, las MAC de cada dispositivo.



* Mediante el uso de *Dirb*, siendo una herramienta de seguridad y hacking web, comúnmente utilizada durante las fases de reconocimiento en pruebas de penetración, que usa para descubrir objetos y directorios ocultos o no indexados en un servidor web.

La herramienta realiza peticiones HTTP GET a una lista de URLs predefinidas, con el objetivo de encontrar recursos accesibles en el servidor que no están visibles desde las páginas de la aplicación web o que no deberían ser accesibles públicamente, utilizado un conjunto de listas de palabras (wordlists) para realizar ataques de “fuerza bruta” contra el servidor y descubrir estos archivos o directorios ocultos basándose en las respuestas comunes de los servidores web. En nuestro caso:

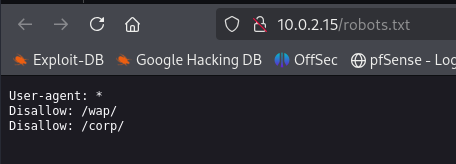


Este programa puede ser muy útil para encontrar información sensible que ha sido mal configurada o expuesta por los administradores del sitio web (directorios de administración, scripts de instalación, archivos de copia de seguridad, etc). En el caso que nos ocupa:

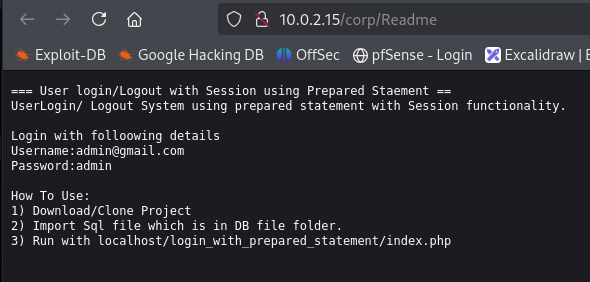


* Desde el mismo navegador web de la maquina Kali, se puede acceder al archivo *“robots.txt*”, siendo éste usado por los administradores de sistemas para los crawlers no procedan a indexar ciertos directorios, que no desean que sean visibles públicamente.

Estos archivos, pueden aportar pistas a los atacantes para saber en qué archivos indagar, y si no está bien configurado, como vemos en este caso, se puede acceder con cierta facilidad y si, además, los directorios guardan información confidencial se puede usar como vía de entrada para validarse mediante escalada de privilegios:

* Se accede, de la misma manera, desde el mismo navegador de nuestro equipo atacante al servidor Fuzzing, concretamente al directorio “*corp/readme*”, donde se puede observar el usuario admin y la contraseña. Se prueba la misma pero no llega a funcionar, pudiendo haber sido cambiada la contraseña.

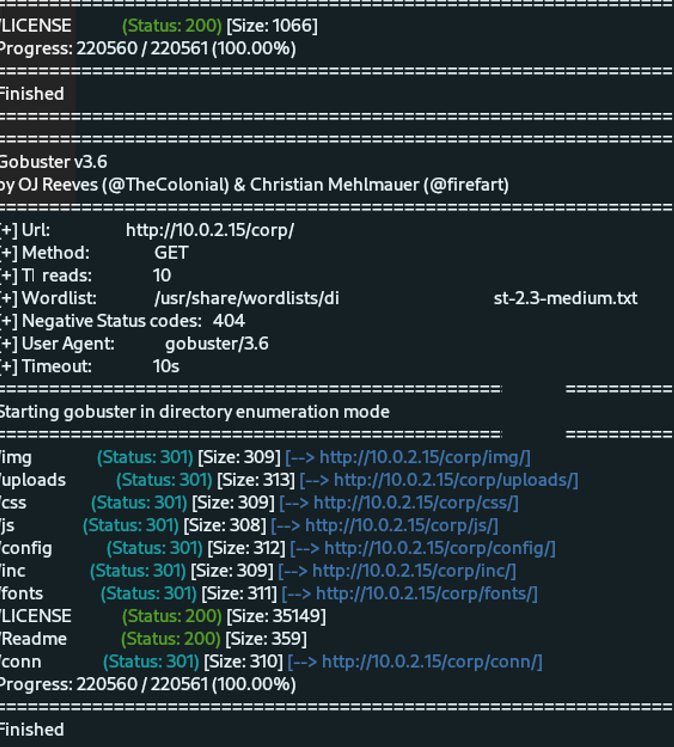


* Con la herramienta de seguridad “Gobuster”, utilizada para el “*brute-forcing[[6]](#footnote-6)*” de URLs (directorios y nombres de archivos) en servidores web y para la enumeración de subdominios, utilizada durante las fases de pruebas de penetración y auditorías de seguridad.

La herramienta realiza solicitudes a direcciones web con diferentes nombres, basándose en listas de palabras (wordlists), para descubrir recursos ocultos que no están enlazados en las páginas accesibles del sitio web, como directorios, archivos específicos y subdominios, siendo apreciada por su velocidad y eficiencia.



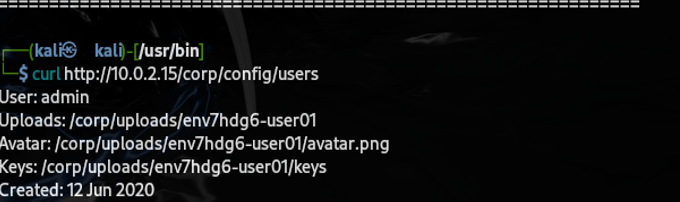
* En nuestro caso, se procede analizar el servidor en la IP 10.0.2.15, aportando todos los directorios disponibles dentro de “corp”:

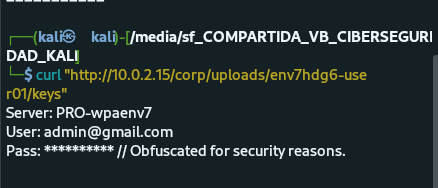


* Como se puede observar, hay 2 archivos con el código 200, es decir que son válidos, y otros con el código 301, significando que los archivos han sido removidos de manera permanente a otra ubicación, pero permitiendo el acceso a dichos directorios.
* Se accede a la ruta del directorio *“corp/config.local”,* donde se pueden observar 2 directorios ocultos, a los cuales no se tiene los permisos necesarios para su acceso y otro, llamado “/users”, el cual aparece como valido y accesible, en la cual, por su nombre, puede contener información de usuarios y claves del sistema.



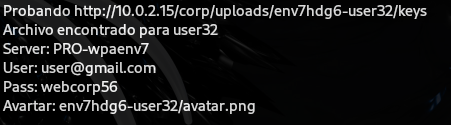
* Se procede al acceso al directorio “users”, conteniendo el nombre de un usuario y varias carpetas de las que se puede destacar:
* La carpeta Uploads, al final aparece “…-user01”, lo que sugiere que pueda haber más user (2,3,4…)
* La carpeta Keys, parece que guarda las claves del usuario 01



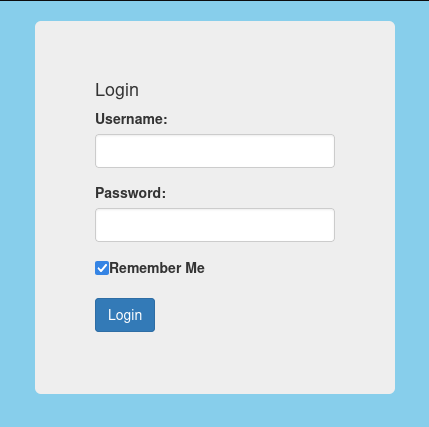
* Si accedemos al directorio “*Keys”,* nos aportan las credenciales del servidor, el usuario administrador del sistema(user01) y la contraseña ofuscada, por lo que se deduce que ésta se encuentra en otro directorio diferente. 
* Para la búsqueda de más usuarios, se ha procedido a realizar un Sprint con Python, para que compruebe de manera automática si existen más usuarios (user01 – 100) y muestre los archivos correspondientes:

**

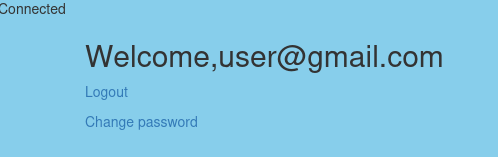
* Como resultado del script anterior, además del user01, nos aporta información del user32, con su usuario y contraseña:



* Ahora, que tenemos un usuario y una contraseña procedemos a intentar loquearnos en la web del servidor, la cual se encuentra protegida*:*

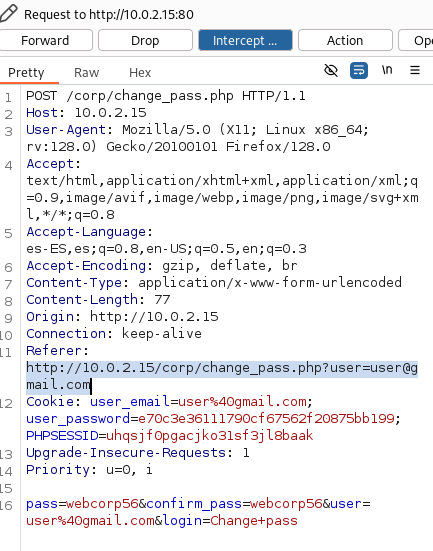


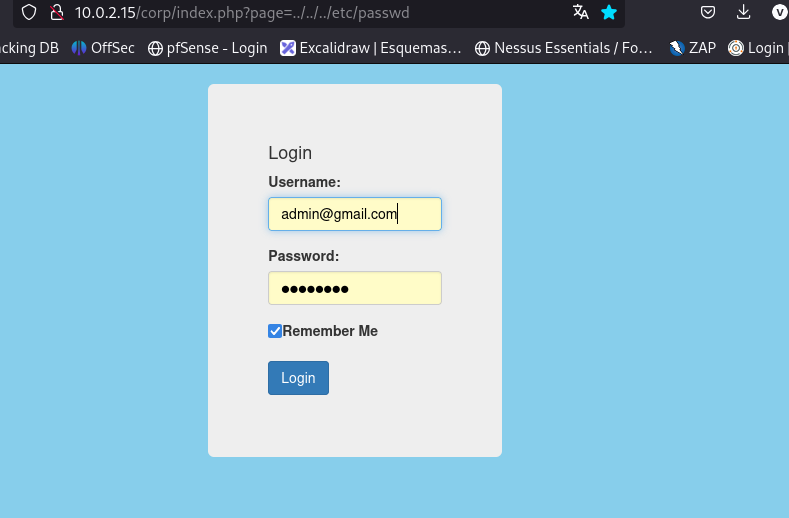
La conexión del user32 se establece de forma satisfactoria, mostrando dos pestañas, una para desconectarse y otra para cambiar la contraseña:



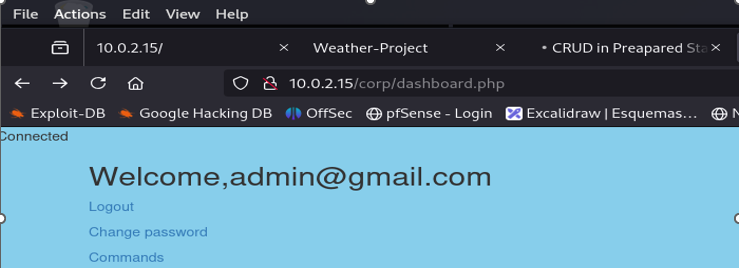
* En este punto, se utiliza la herramienta Burp Suite Community, la cual es ampliamente utilizada por profesionales de la seguridad para realizar pruebas de penetración y auditorías de seguridad en aplicaciones web, ofreciendo un conjunto de herramientas que trabajan juntas para apoyar todo el proceso de pruebas de seguridad, desde el mapeo inicial y el análisis de la superficie de ataque de una aplicación, hasta la identificación y explotación de vulnerabilidades de seguridad que ofrece un conjunto de herramientas que trabajan juntas para apoyar todo el proceso de pruebas de seguridad, desde el mapeo inicial y el análisis de la superficie de ataque de una aplicación, hasta la identificación y explotación de vulnerabilidades de seguridad.

Para nuestro caso, se ha aprovechado el intento de cambio de clave en el user32, procediendo a su escaneo con esta aplicación, interceptando el código que ejecuta el servidor al realizar esta acción. Una vez en Burp Suite, se procede a cambiar en línea de código el [user@gmail.com](mailto:user@gmail.com) por [admin@gmail.com](mailto:admin@gmail.com) , enviando el código modificado de vuelta al servidor.

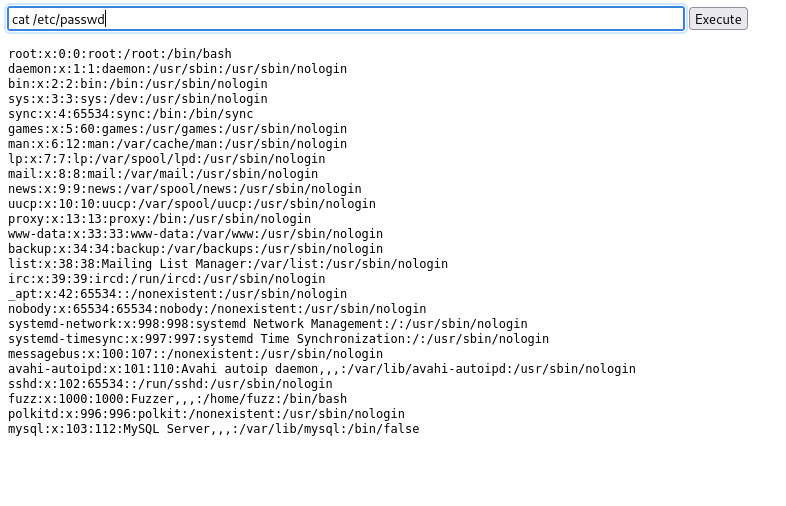




* Finalmente se procede al cambio de contraseña del usuario [admin@gmail.com](mailto:admin@gmail.com) con privilegios root.



* Si entramos en la pestaña, que no aparecía al estar loqueado con [user@gmail.com](mailto:user@gmail.com): Commans, y ahí intentamos acceder a la ruta *etc/passwd* es positiva, pero no a la ruta *etc/shadow*, deduciéndose que es un usuario con privilegios de root, pero no es el usuario root.



2.- Conclusiones

Una vez terminado el análisis de vulnerabilidades y su explotación en la maquina Fuzzing, se recomienda implantar, por ser necesario y en la medida de lo posible, las recomendaciones indicadas anteriormente y otras de carácter más técnico como:

* 1. Tener información confidencial en un archivo *robots.txt* es una práctica peligrosa que puede exponer la información a actores maliciosos y acarrear sanciones importantes para la empresa, siendo crucial tomar medidas para proteger la información confidencial y garantizar el cumplimiento de las normativas de protección de datos y seguridad de la información.
  2. Para protegerse del uso por curiosos o atacantes de herramientas de enumeración de fuerza bruta en servidores web (Brute-Forcing URL), se proponen una serie de medidas
     + Uso de contraseñas seguras y complejas
     + Habilitar la autenticación de dos factores o multifactor (2FA o MFA).
     + Limitar el número de intentos de inicio de sesión
     + Utilizar un firewall de aplicaciones web (WAF)
     + Uso de CAPTCHA
  3. En relación a las vulnerabilidades encontradas mediante Nessus, los resultados están dentro de los limites de riesgo asumible, solo encontrando 1 Critica, 1 alta y 1 baja, pero apareciendo como INFO, algunas que han sido aprovechadas en esta explotación para conseguir el resultado final, las cuales son:
     + La vulnerabilidad crítica y alta se refieren a la versión apache que usa el sistema debiendo ser actualizada a la versión 2.4.61 o superior.
     + La vulnerabilidad baja permite a un atacante conocer la hora y fecha del sistema, que podría usarse en ataques más complejos (canal lateral, MITM, etc), por lo que debería corregirse.
     + En nuestro caso, hemos usado la vulnerabilidad INFO 10302, la cual perite acceder al archivo robots.txt del servidor, pero además se deberían corregir otras de este tipo como, las relacionadas con el protocolo SSH, MySQL, entre otras. Se adjunta en Anexos el informe Nessus en imágenes.



**4.- BIBLIOGRAFÍA**

<https://www.nist.gov/publications/zero-trust-architecture>

<https://commission.europa.eu/law/law-topic/data-protection/reform/rules-business-and-organisations/enforcement-and-sanctions/sanctions/what-if-my-companyorganisation-fails-comply-data-protection-rules_es>

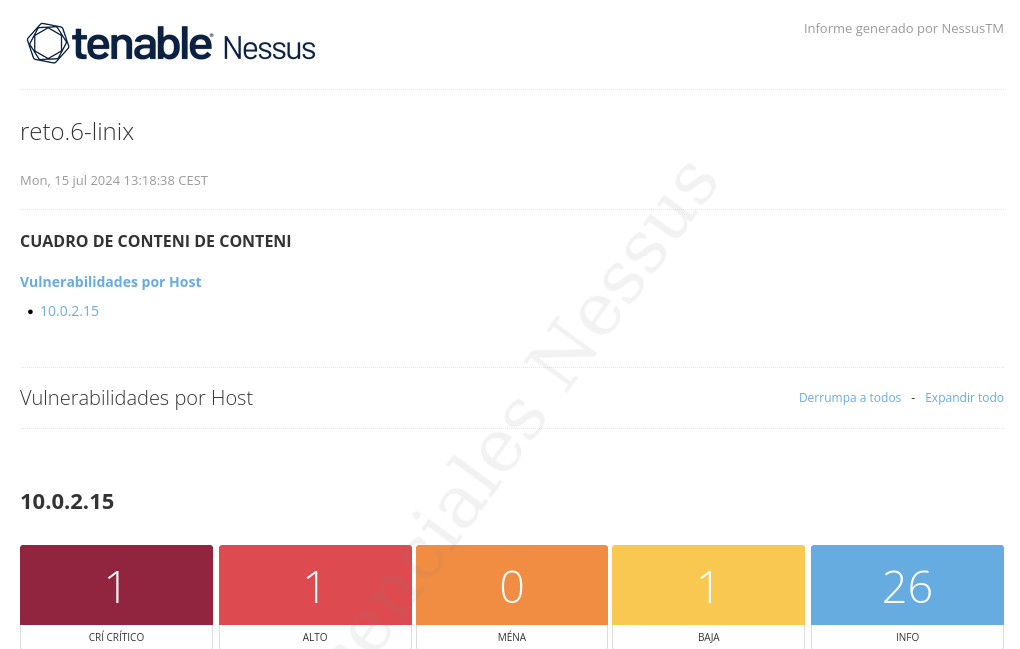
<https://ayudaleyprotecciondatos.es/2019/02/19/sanciones-rgpd-lopd-2019/>

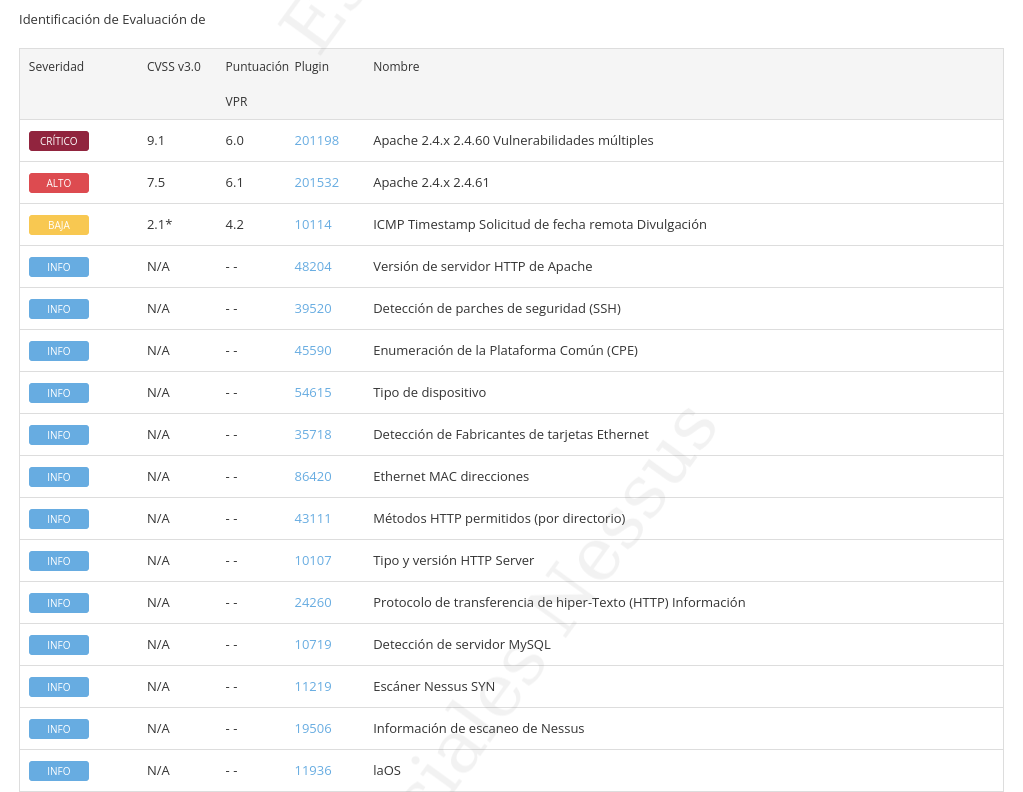
<https://nmap.org/man/es/index.html>

<https://owasp.org/www-community/attacks/Brute_force_attack>

<https://keepcoding.io/blog/que-es-burp-suite/>

**5.- ANEXOS**

****





1. Zero Trust, parte de la premisa de no confiar en ningún usuario, dispositivo o sistema dentro o fuera de la red organizacional y se basa en los siguientes principios clave:

   * **Verificación continua:** La identidad y la autorización de cada usuario y dispositivo se verifican constantemente.
   * **Principio de Menos privilegios:** Los usuarios y dispositivos solo reciben acceso a los recursos que necesitan para realizar su trabajo.
   * **Segmentación:** La red se segmenta en zonas para limitar el acceso, contención de amenazas y evitar el movimiento lateral de las mismas
   * **Protección de datos:** Los datos se protegen con cifrado adecuado y otras medidas de seguridad.
   * **Monitoreo y respuesta:** La actividad de la red se monitorea constantemente para detectar y responder a las amenazas.

   [↑](#footnote-ref-1)
2. Conocido como **rastreador**, **araña web** o **robot**, es un programa informático automático que navega por la web y recopila información de las páginas web [↑](#footnote-ref-2)
3. El RGPD es un reglamento de la Unión Europea que establece normas estrictas para la protección de datos personales [↑](#footnote-ref-3)
4. La LOPDGDD es ley española que desarrolla el RGPD y que establece normas específicas para la protección de datos personales en España [↑](#footnote-ref-4)
5. La LSSI es una legislación española que regula la prestación de servicios de la sociedad de la información y el comercio electrónico, estableciendo una serie de obligaciones a las empresas e infracciones en caso de su incumpliento, [↑](#footnote-ref-5)
6. Es un tipo de **ataque de fuerza bruta** que se utiliza para intentar acceder a contenido web protegido mediante la prueba de todas las combinaciones posibles de URL hasta encontrar una válida, usado para acceder a contenido restringido, como páginas web con inicio de sesión, directorios privados o archivos confidenciales. [↑](#footnote-ref-6)