

**INFORME: EJECUTIVO Y TÉCNICO**

**PENTESTING EN SISTEMAS “PIVOTING-AD”**

* Fecha: 7 de noviembre de 2024
* Cliente: Reto 20 – Team Challenge
* Consultora de Ciberseguridad: The Bridge - Accelerator
* Control de Cambios

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Versión | Documento | Fecha | Cambios | Autor | revisor | visto bueno |
| 1.1 | Informe de resultados | 13/11/2024 | 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

1. 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

1. Bibliografía ---------------------------------------------------------------------25
2. **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 sistema formado por 3 equipos pertenecientes a la red *“Pivoting-AD”*, 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,** ademásde **movimientos laterales entre los equipos y persistencia** en el sistema explotado.

**-** La red objetivo, consta de un equipo con S.O **Linux conectada** a **2** ordenadores con S.O. **Windows** “server 2019” en ***“Active Directory”***, necesitando para acceder a ellos credenciales 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 interconexión entre los dispositivos de su sistema, de manera remota y segura, y las conexiones HTTP de los servicios web de su empresa, respectivamente.
* En el servicio FTP, se ha detectado una grave vulnerabilidad, permitiendo la**autenticación** mediante **credenciales *“Anonymous”: “230”,*** a la **vista** de **cualquier atacante**,pudiendo acceder a la información sensible que pudiera encontrase en sus servidores , además de poder ser una posible vía de entrada no autorizada a sus sistemas.
* Además, se ha podido observar **sin autenticarse** en el mismo que existe un archivo llamado ***“CALL.html”***.
* Mediante herramientas de escaneo de servidores web para descubrir **directorios, archivos, subdominios** y otros puntos de entrada ocultos y menos evidentes, se ha **escasa información** para permitir conocer la estructura web de su organización, pero si la necesaria para poder acceder de forma no autorizada a su sistema mediante la ejecución de código malicioso, pudiendo exponer información sensible y representar un **riesgo de seguridad significativo** ( usuarios del sistema, etc)

**-** En resumen, se han conseguido explotar varias vulnerabilidades graves, debido a una **falta de actualización** en las tecnologías o aplicaciones, así como **accesos a zonas sensibles**, que no deberían estar abiertas al público que iremos desarrollando en el próximo punto.



1. 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 **FTP*,*** el cual presenta una grave vulnerabilidad al tener habilitado la autenticación mediante usuario ***Anonymous y contraseña 230.*** Una vez dentro del servicio, observando poseer permisos de lectura y escritura en los archivos, ***permitiendo inyectar código*** ***malicioso*** para conseguir un acceso no autorizado vía “CLI” en la primera máquina “Linux-User”, con un ***usuario con bajos privilegios*.**
* Una vez dentro de la máquina Linux, se precede a investigar por diferentes directorios, encontrando información valiosa, que nos ha permitido encontrar las credenciales de otro usuario con más privilegios en el sistema **“Shrek”: “onion”,** habiendo tenido que usar técnicas de desencriptado sencillo para ello.
* Además, junto a las mencionadas credenciales, se ha hallado lo que **parecen** unas **credenciales de Windows**, debido a la barra invertida usada **“\”** y la estructura de la misma, pudiendo ser de una de las máquinas Windows a explotar: **EXAMPLE\testing:2021!Query.** En caso que esto fuera cierto, estaríamos ante una grave vulnerabilidad, por tener **credenciales** de una máquina de su red en **texto plano**.
* Siguiendo con el análisis, se ha detectado otra **vulnerabilidad crítica significativa**, ya que, el sistema permite obtener **permisos root al ejecutar el programa python 3.5**, sin necesitar contraseña, por lo que se procede a i**nyectar código malicioso en python,** permitiendo la conexión en modo **root** y máximos privilegios en la maquina “Linux-User”.
* Finalmente, las credenciales encontradas de Windows nos permiten acceder a la máquina **“Windows-User”** con el **usuario “testing”.**
* Mediante aplicaciones específicas para obtención de información para ataques en sistemas Kerberos, se obtiene un **servicio principal** **“iis\_service - (HTTP/WINDOWS)”**, obteniendo su hash, siendo “**hacheado”** mediante aplicaciones para ello, obteniendo descifrar la contraseña en plano: “**LaRosalia2021”**.
* Una vez obtenidas el usuario y la contraseña del servicio en un entorno de Kerberos, y con el uso de otras herramientas, se han obtenido los **hashes** de la bases de datos local del entorno “active Directory” **(SAM[[1]](#footnote-1))** , consiguiendo los **hashes NTLM[[2]](#footnote-2)** de los usuarios del dominio y aprovechando que el sistema Kerberos permite autenticarse con el mismo, se realiza un **“ataque Pass the hash”** con la contraseña del Administrator con resultado positivo, consiguiendo una conexión vía CLI a la maquina **Windows-DC con máximos privilegios** en el sistema.



1. 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 máximos** en la máquina **Windows-DC**, mediante la realización de **movimientos laterales** sobre las otras máquinas de la **red Pivoting-AD**, aportando detalles más técnicos más adelante.



1. RECOMENDACIONES GENERALES.- En el **análisis y explotación** reciente de seguridad de su **red Pivoting-AD**, 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 vulnerabilidades y debilidades identificadas**, junto a las correspondientes **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. **DETECCION DE PROBLEMAS:**

* **Puertos abiertos** al exterior detectados:
* Puerto 21 (FTP) - Permite la transferencia de archivos.
* Puerto 22 (SSH) - Permite conexiones remotas seguras.
* Puerto 80 (HTTP) - Habilita el tráfico web no cifrado.

### **- Análisis de Vulnerabilidades:**

* **FTP** (Puerto 21): Este puerto presenta un riesgo alto debido a la **autenticación** configurada como **"anonymous:230"**, lo que permite **acceso sin credenciales** específicas. Esta configuración facilita la posibilidad de acceso no autorizado y modificación de archivos sensibles, abriendo una **vía potencial para intrusiones** en el sistema.
* **SSH** (Puerto 22): El servicio SSH es crítico para conexiones seguras, pero una configuración incorrecta o una **versión desactualizada** puede exponer el sistema a ataques de fuerza bruta o explotación de vulnerabilidades conocidas.
* **HTTP** (Puerto 80): El **tráfico web sin cifrar** puede **permitir** a un atacante **interceptar comunicaciones** o manipular contenido entre el cliente y el servidor.

**- Recomendaciones para Mitigación:**

* Cerrar el puerto FTP o, **preferiblemente,** reemplazarlo por **SFTP**, que asegura la transferencia de archivos mediante el cifrado SSH.
* **Actualizar SSH** a la última versión y aplicar configuraciones de seguridad avanzadas.
* Implementar **autenticación con claves públicas** para SSH en lugar de contraseñas.
* Considerar el uso de **HTTPS (Puerto 443)** en lugar de HTTP (Puerto 80) para asegurar el tráfico con cifrado SSL/TLS, p**rotegiendo la confidencialidad e integridad de los datos** en tránsito.

1. **DETECCION DE PROBLEMAS:**

* **Directorio web abierto ("Index of")**: Se detectó que el servidor web ha permitido acceso a un **directorio crítico “/files”** a través de una página de índice, lo que expone a los usuarios o atacantes a visualizar la estructura interna del servidor.
* **Riesgo de exposición:** Esta configuración puede hacer que archivos sensibles, como configuraciones, registros de errores o respaldos, sean accesibles públicamente, así como a la **ejecución de archivos maliciosos desde la web**.

**- Análisis de Vulnerabilidades:**

* **Riesgo de enumeración de archivos:** Un atacante puede acceder y enumerar archivos y directorios expuestos a través de la página "Index of", permitiéndole **comprender mejor la estructura interna del sistema**.
* **Posibilidad de acceso a archivos sensibles:** Si existen archivos **mal configurados** o que **contienen información confidencial** (como bases de datos, credenciales, o archivos de configuración), estos podrían ser leídos o descargados, facilitando posibles **ataques de escalamiento o robo de información.**
* **Incremento de la superficie de ataque:** La visibilidad de los archivos facilita el descubrimiento de vulnerabilidades o errores de configuración en el sistema, abriendo una puerta para futuros intentos de explotación.

**- Recomendaciones para Mitigación:**

* **Deshabilitar la opción de listado de directorios[[3]](#footnote-3):** Configurar el servidor web para que **no** permita el acceso público a la página **"Index of".**
* **Restringir permisos de acceso a directorios sensibles:** Asegurarse de que **sólo el personal autorizado** tenga acceso a directorios sensibles y que estos no sean accesibles públicamente.
* **Mover archivos confidenciales fuera de la raíz pública del servidor web:** Mantener archivos de configuración, registros y respaldos fuera de la raíz web y asegurarse de que solo sean accesibles internamente o a través de conexiones seguras

1. **DETECCION DE PROBLEMAS:**

* **Archivos detectados con credenciales inseguras:** Se han identificado archivos en el sistema que almacenan credenciales, algunos de los cuales usan hashes débiles (MD5) y otros contienen contraseñas en texto plano.
* **Riesgo de exposición:** La presencia de credenciales en texto plano y hashes inseguros aumenta el riesgo de que un atacante pueda descifrar o acceder directamente a estas contraseñas.

**- Análisis de Vulnerabilidades:**

* **Credenciales en texto plano:** Las contraseñas almacenadas en texto plano son accesibles en caso de que un atacante obtenga acceso al sistema de archivos, permitiéndole **obtener acceso a cuentas o servicios sin necesidad de descifrado**.
* **Hashes débiles:** Si las credenciales están almacenadas con algoritmos de **hash inseguros** (como MD5 o SHA-1), son **vulnerables a ataques de fuerza bruta** o ataques de diccionario, ya que estos algoritmos son rápidos de procesar y existen bases de datos públicas de hashes precalculados.
* **Facilidad para escalar privilegios y movimientos laterales:** La presencia de credenciales vulnerables facilita que un atacante, tras obtener acceso inicial, escale privilegios en el sistema o comprometa otras cuentas, servicios u otros equipos del sistema.

- **Recomendaciones para Mitigación:**

* **Reemplazar almacenamiento de contraseñas en texto plano:** Modificar las configuraciones para que las contraseñas no se guarden en texto plano. Implementar el almacenamiento de contraseñas cifradas con un estándar seguro.
* **Usar algoritmos de “hashing” seguros:** Reemplazar los hashes débiles por **algoritmos robustos** como bcrypt[[4]](#footnote-4), Argon2[[5]](#footnote-5), o PBKDF2[[6]](#footnote-6), que incorporan mecanismos de "salting" y son resistentes a ataques de fuerza bruta.
* **Restringir el acceso a archivos de credenciales:** Limitar los permisos de acceso para que solo usuarios o procesos autorizados puedan ver estos archivos, minimizando el riesgo de exposición.

1. **DETECCION DE PROBLEMAS:**

* **Permiso “NOPASSWD” en Python3.5**: Se detectó que el usuario vulnerado “Shrek” tiene permisos sudo configurados con la opción **NOPASSWD** para ejecutar **/usr/bin/python3.5** como root, sin requerir contraseña.
* **Riesgo de abuso**: Esta configuración permite ejecutar comandos arbitrarios con privilegios elevados, ya que Python puede ser utilizado para acceder al sistema de archivos, ejecutar comandos y inyectar código malicioso en el sistema.

**- Análisis de Vulnerabilidades:**

* **Ejecución de comandos como root**: Dado que Python permite ejecutar código arbitrario, un atacante que obtenga acceso a este usuario puede utilizar sudo /usr/bin/python3.5 para ejecutar comandos o scripts con permisos de root.
* **Riesgo de escalamiento de privilegios**: Esta configuración facilita el escalamiento de privilegios, ya que el atacante no necesita autenticarse para ejecutar Python con permisos elevados, permitiendo acceder a archivos de sistema, modificar configuraciones, y comprometer otros usuarios.
* **Impacto crítico en la seguridad**: La capacidad de ejecutar Python como root sin contraseña compromete completamente la seguridad del sistema, dado que Python tiene acceso ilimitado al entorno y recursos del sistema.

**- Recomendaciones para Mitigación:**

* **Eliminar el permiso NOPASSWD en /usr/bin/python3.5**: Revocar el acceso a sudo sin contraseña para Python. Esto evitará que cualquier usuario ejecute Python como root sin autenticación.
* **Restringir el acceso a Python con privilegios elevados**: Si es necesario que ciertos scripts se ejecuten como root, configurar sudo para que solo el script necesario tenga permisos y no el intérprete de Python completo.
* **Implementar políticas de seguridad de sudo estrictas**: Revisar y minimizar los comandos permitidos con sudo, especialmente con la opción NOPASSWD, para reducir el riesgo de abuso.

1. **DETECCION DE PROBLEMAS:**

* **Enumeración de SPNs en el Dominio Kerberos:** Utilizando el script **PowerView.ps1**, se realizó una enumeración de los **Service Principal Names (SPN)** del controlador de dominio del entorno Active Directory, lo que **permitió identificar cuentas de servicio** configuradas para Kerberos. Estas cuentas pueden ser un objetivo, ya que suelen estar asociadas con servicios y podrían tener permisos elevados o credenciales útiles.
* **Credenciales obtenidas:** Se descifró un hash de SPN del servicio **iis\_service (HTTP/WINDOWS)**, logrando **acceder** a la cuenta **con privilegios** de servicio web.

**- Análisis de Vulnerabilidades:**

* **Acceso mediante SPN:** La existencia de **SPNs** configurados para Kerberos expone cuentas de servicio que pueden ser **vulnerables a ataques de fuerza bruta** o de descifrado de hash, como el **ataque “Kerberoasting”**.
* **Privilegios del servicio web:** La cuenta del servicio **“iis\_service”** otorgó acceso a la red interna y a recursos adicionales que, con credenciales válidas, pudieron ser aprovechados para avanzar en el ataque.

- **Recomendaciones para Mitigación:**

* **Limitar el número de cuentas de servicio con SPNs:** Solo las cuentas estrictamente necesarias deben tener configurados SPNs. Eliminar SPNs innecesarios **reduce la superficie de ataque.**
* **Rotar y fortalecer contraseñas de cuentas de servicio:** Usar contraseñas **robustas y configurar una política de rotación** regular de contraseñas para cuentas de servicio.
* **Auditar accesos y configuraciones de Kerberos:** Revisar y auditar frecuentemente la configuración de Kerberos para detectar SPNs expuestos y cuentas vulnerables.

1. **DETECCION DE PROBLEMAS:**

* **Acceso a la SAM y NTLM Hashes:** A través del acceso a la cuenta de servicio obtenida, se **empleó** la colección de herramientas **Impacket[[7]](#footnote-7)** para extraer los hashes NTLM de la SAM (Security Account Manager) del sistema.

Esta acción **permitió acceder a hashes** de otras cuentas, incluidas cuentas de **privilegio más alto.**

### **- Análisis de Vulnerabilidades:**

* **Riesgo de acceso no autorizado a Hashes NTLM en la SAM:** La extracción de hashes permite que un atacante realice ataques de Pass-the-Hash o intente descifrar estos hashes para obtener credenciales en texto claro.
* **Escalabilidad del ataque:** La posibilidad de extraer estos hashes incrementa la probabilidad de un ataque de **escalamiento de privilegios o de movimiento lateral en la red.**

- **Recomendaciones para Mitigación:**

* **Asegurar el Archivo SAM y Hashes NTLM:** Limitar el acceso a la SAM y evitar que cuentas de servicio puedan acceder a los hashes NTLM almacenados, implementando **políticas de restricción de acceso más robustas.**
* **Deshabilitar LM y NTLMv1 en el dominio**: Configurar la seguridad de la red para deshabilitar los hashes LM y NTLMv1, lo cual evita que se almacenen hashes inseguros en la SAM.
* **Implementar protección contra ataques de Pass-the-Hash**: Usar tecnologías como **Credential Guard**[[8]](#footnote-8) en entornos Windows para proteger las credenciales y mitigar el riesgo de explotación de Pass-the-Hash.

1. **DETECCION DE PROBLEMAS:**

* **Pass-the-Hash para Escalamiento de Privilegios:** Utilizando los hashes NTLM obtenidos, se realizó unPass-the-Hash en otra máquina Windows, **logrando acceso con máximos privilegios**, lo que facilitó el **control total sobre el sistema**.

### **- Análisis de Vulnerabilidades:**

* **Pass-the-Hash para Escalamiento de Privilegios:** La vulnerabilidad de Pass-the-Hash permite a un atacante **autenticarse con privilegios elevados** **utilizando un hash NTLM en lugar de la contraseña**, lo que facilita el acceso total sin necesidad de autenticación adicional.
* **Compromiso total del sistema:** Este método otorga al atacante control completo sobre la máquina de destino, comprometiendo así toda su seguridad y acceso a datos críticos.

- **Recomendaciones para Mitigación:**

* **Implementar autenticación multifactor (MFA):** Agregar un segundo factor de autenticación puede evitar que el uso de hashes NTLM sea suficiente para acceder a sistemas críticos.
* **Segregar cuentas de administrador y usuarios estándar:** Minimizar el uso de cuentas con privilegios administrativos y usar cuentas con privilegios limitados para tareas rutinarias.
* **Auditar y monitorear el uso de credenciales de alto privilegio:** Implementar un sistema de auditoría y monitorización para identificar intentos de Pass-the-Hash y alertar sobre posibles actividades sospechosas.

**-**  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”**[[9]](#footnote-9). 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.

1. 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 cuales, **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.

1. NORMATIVA APLICABLE Y SANCIONES

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

* **Reglamento General de Protección de Datos** (RGPD)[[10]](#footnote-10) **y** la **Ley Orgánica de Protección de Datos Personales y garantía de los derechos digitales** (LOPDGDD)[[11]](#footnote-11). - Si la información confidencial que se encuentra en el sistema no se encuentra debidamente custodiada, su incumplimiento podría acarrear sanciones importantes para la empresa.
* **Ley de Servicios de la Sociedad de la Información y el Comercio Electrónico** (LSSI)[[12]](#footnote-12). - 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[[13]](#footnote-13).** - 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[[14]](#footnote-14)**.- Estándar internacional que ayuda a las empresas a identificar, gestionar y mitigar riesgos de ciberseguridad, estableciendo los requisitos de un SGSI[[15]](#footnote-15), el cual proporciona el marco de protección para la triada CIA, asegurando que la organización cumple con los requisitos legales y normativos vigentes.
* Además,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.
* **NIST - CIBERSECURITY FRAMEWORK[[16]](#footnote-16).-** 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.

1. **- INFORME TÉCNICO**

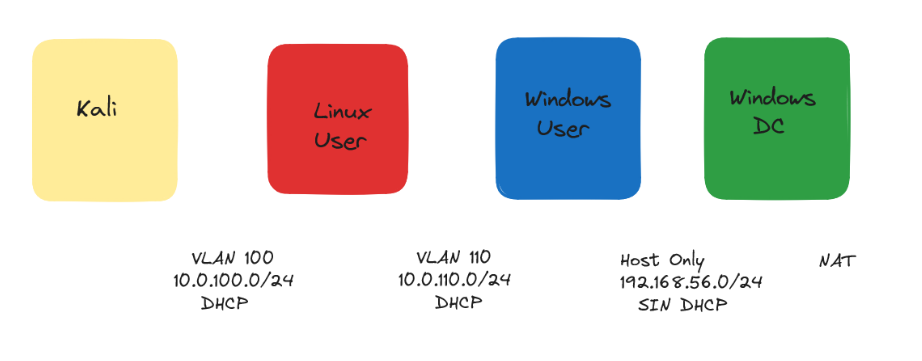


1. **PRESENTACIÓN. –** Para conseguir el objetivo fijado en el contrato, se han seguido la siguiente línea de investigación:

El laboratorio de pruebas ha sido entregado y montado en red por la empresa contratante, formado por tres equipos:

* Máquina Linux-User
* Máquina Windows-User
* Máquina Windows-DC

A este laboratorio se le ha conectado una máquina Kali desde la que se han realizado las explotaciones en cada una de las máquinas del laboratorio hasta conseguir el objetivo del contrato: **obtención de privilegios máximos en la maquina Windows-DC.**



Los Equipos han sido entregado con los siguientes S.O: un sistema **Linux** 3.2 - 4.9 - generic i686*,* sin aportar credenciales de inicio de sesión y **dos máquinas Windows** con los sistemas Windows server 2019 en un **entorno Active Directory,** sin credenciales de acceso, por lo que, el análisis y explotación será realizado sin acceso a información interna de la organización.

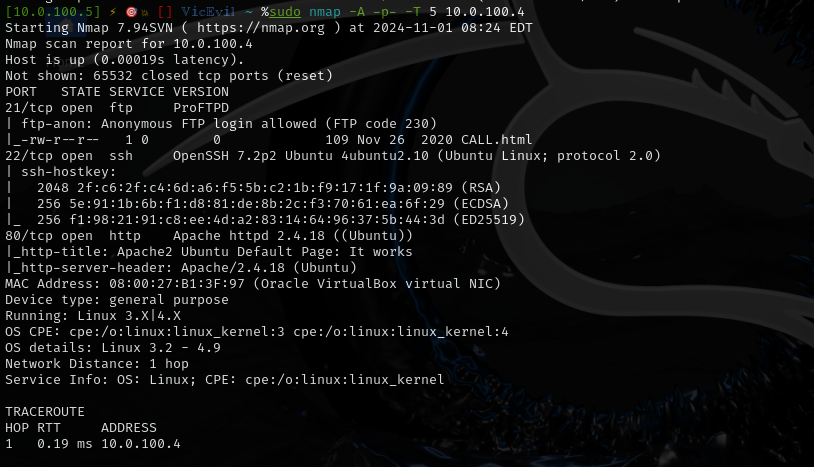


Imagen 1.- Información completa los puertos y servicios de la máquina “Linux-Usr”

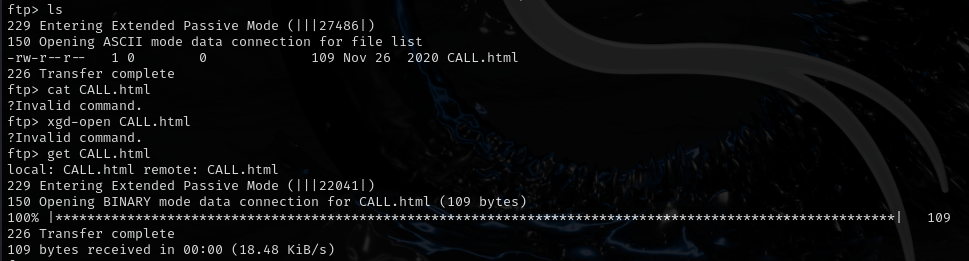
* INFORMACIÓN INICIAL. - Se procede a consultar mediante **Nmap***,* herramienta de código abierto utilizada para explorar y auditar la seguridad de redes y sistemas*,* el rango de IPs donde se encuentran ambas maquinas*,* siendo la **de Linux-User: 10.0.100.4** y de la máquina atacante: 10.0.100.5. Además, la maquina objetivo tiene un total de **3 puertos** abiertos:

|  |  |  |  |
| --- | --- | --- | --- |
| **Puerto** | **21** | **22** | **80** |
| **Servicio** | FTP | SSH | HTTP |
| **Versión** | ProFTP | OpenSSH 7.2p2 -Ubuntu | Apache 2.4.18(Ubuntu) |

* Además, en el servicio FTP, se confirma tener **activado la autenticación** mediante **usuario anónimo** con las credenciales: **Anonymous:230.**
* Se ha seguido la siguiente la **línea de explotación del sistema**, que más adelante se irá desarrollando: el **servicio FTP**, consiguiendo autenticación de con las credenciales anónimas, consiguiendo **inyectar código malicioso**, permitiendo acceso no autorizado al sistema, donde se han explotado otras vulnerabilidades encontradas, consiguiendo **pivotar** desde esta máquina a las maquinas del **entorno Kerberos**, finalizando con **ataque “Pass-the-hash”(PTH**) consiguiendo máximos privilegios en la maquina Windows -DC.



1. **FASE DE EXPLORACIÓN - EXPLOTACIÓN**
2. ACCESO SERVICIO FTP.-Mediante las credenciales obtenidas mediante la aplicación Nmap, se procede a la conexión al servicio, encontrando en su interior un **archivo** llamado **“CALL.html”**, procediendo a su lectura y transferencia a la máquina Kali:



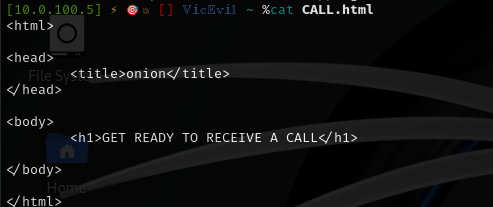


Imagen 2.- Detalle de la conexión al servidor FTP, la transferencia y apertura del archivo “CALL”

Una vez transferido a la Kali, se procede a su apertura destacando que en el head del archivo “html”, tiene un título llamado “onion” y una frase en inglés “Get ready to receive a call”.

1. USO APLICACIONES DE ESCANER DE SERVIDORES WEB.- Se han usado **varias herramientas de “Fuzzing”**, comúnmente utilizadas 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, siendo utilizada sobre el **puerto 80**, apareciendo en el análisis:

**-** Algunas de las aplicaciones usadas han sido Gobuster, dirsearch, Dirb, entre otras, así como con distintos diccionarios, encontrando únicamente un directorio interesante para esta explotación: **/files**

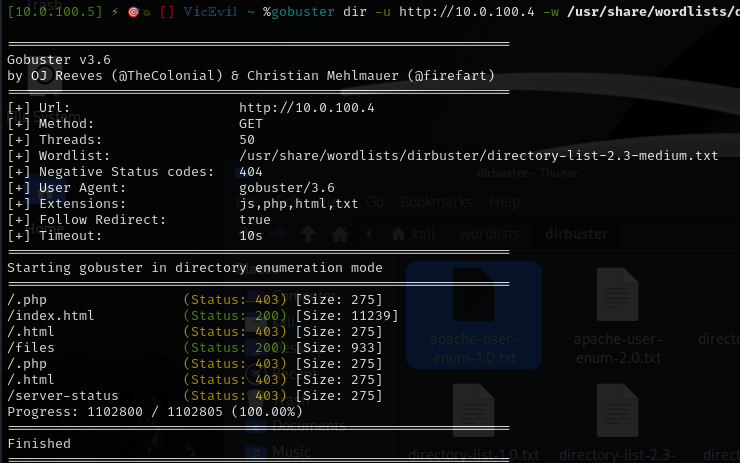
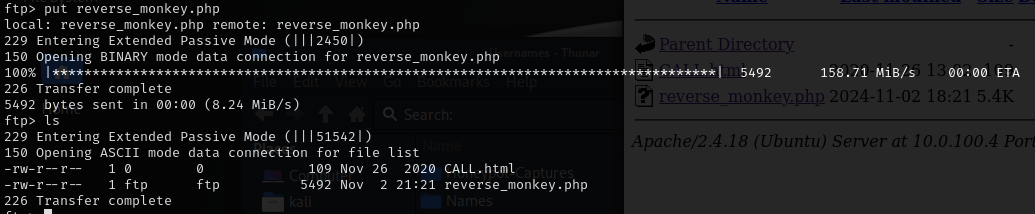


Imagen 3.- Resultado de la búsqueda de directorios con una de las aplicaciones usadas.

- **Con esta nueva información**, se comprueba que al **subir archivos al servidor FTP**, estos se muestran en el **directorio web: /Files**, por lo que se procede a **subir una shell maliciosa** , siendo ejecutada desde el navegador con resultado positivo, **consiguiendo un acceso no autorizado** a la máquina **Linux-User** con escasos privilegios:



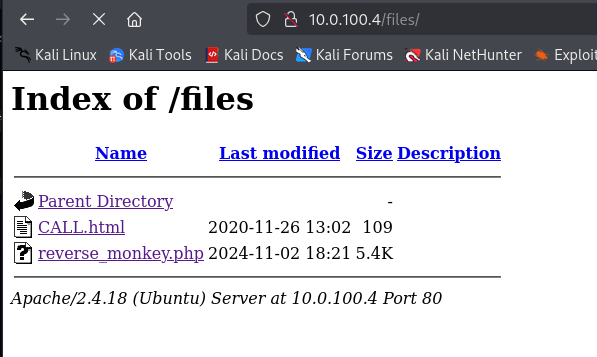
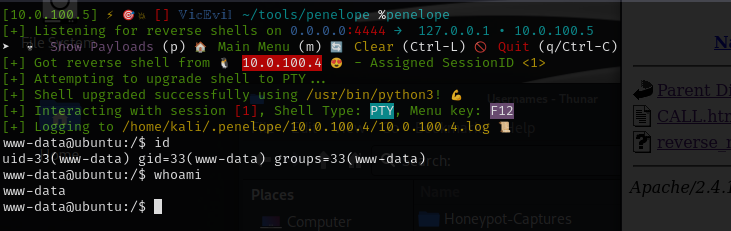
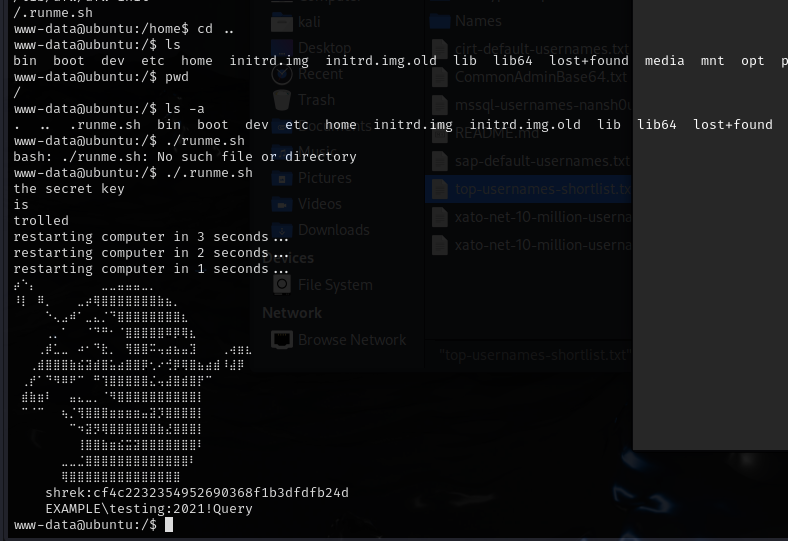


Imagen 4.- Subida de archivos a través de FTP y ejecución de la Reverse Shell en la web, siendo recepcionada en la Kali mediante la herramienta “Penélope”

**-** Se procede a **investigar** por diferentes **directorios** de la máquina **“Linux-User”,** encontrando información sobre un **archivo “runme.sh”**, realizando la búsqueda del mismo mediante el comando **“find”**, siendo encontrado al final de una larga lista, el cual, estaba oculto. Una vez en el directorio donde se halla el archivo, se ejecuta consiguiendo **información importante**:

* **Shrek:cf4c2232354952690368f1b3dfdfb24d**, siendo este último un hash MD5, procediendo a su descifrado correspondiendo a **“onion” (título archivo HTML)**.
* **EXAMPLE\testing:2021!Query.-** Parece **corresponder** a un dominio, usuario y una contraseña de **Windows**, por la **estructura y la barra “\”.**



- Se procede a ejecutar el comando **“sudo -l”**, donde se observa que podemos **ejecutar** la aplicación **python3.5 en modo root** **sin contraseña**, procediendo a su **ejecución** junto un **shell de python**, consiguiendo **acceso root**, procediendo al cambio de contraseña:

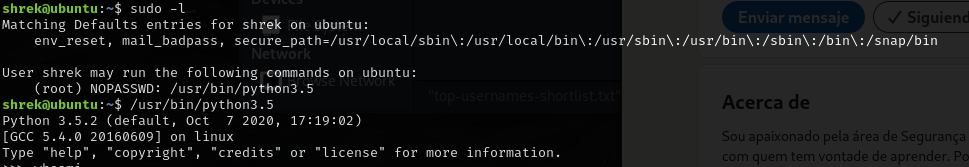
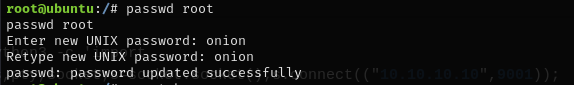
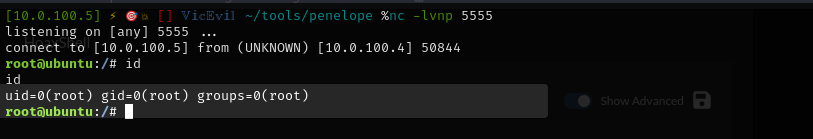
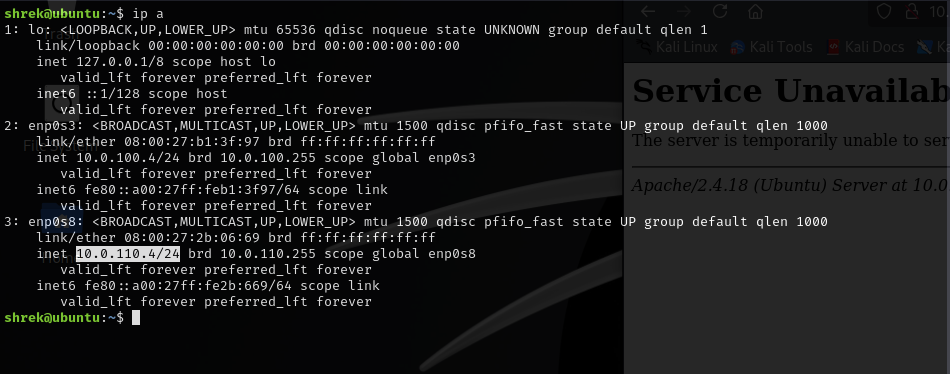
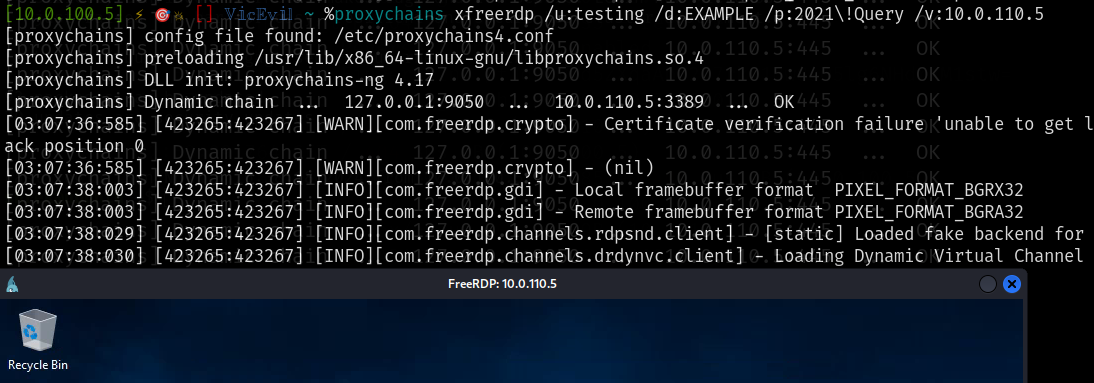


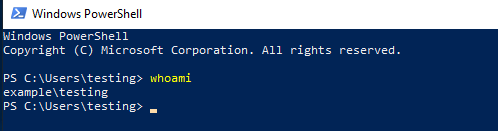
Imagen 5.- Ejecución de Shell explotando un archivo mal configurado consiguiendo acceso root al sistema

- Se realiza una conexión mediante un túnel dinámico a través el usuario shrek, consiguiendo la IP del **segundo adaptador de red** de la maquina **Linux User: 10.110.0.4**., con la **finalidad** de **conocer y pivotar** sobre las **maquinas Windows,** establecidas por este lado del adaptador de red de Linux-User.



- Con **la contraseña** extraída anteriormente **de Windows**, y a través de un **túnel local** establecido a través de la **maquina “Linux User”** por el **puerto 3389**, accedemos a la maquina **Windows User** mediante **“RDP**”, con resultado positivo.





- Una vez **dentro** de la maquina **“Windows User”,** se consigue la **IP del segundo adaptador de red** y mediante el comando **“arp -a”,** **la IP del adaptador** **de red** de la máquina **“Windows DC”:**

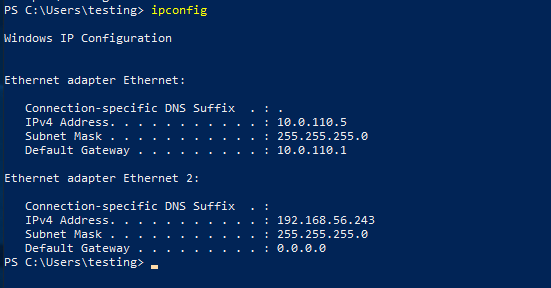


Imagen 6.- IP del segundo adaptador de red de la máquina “Windows-User”



Imagen 7.- Uso del comando “arp” desde Windows-User para obtener la IP de Windows-User

- Se realiza **“ping”** desde la **Kali** para comprobar que **funcionan** las **técnicas de ssh-tuneling** utilizadas , obteniendo **conexión** con las máquinas **Windows.**

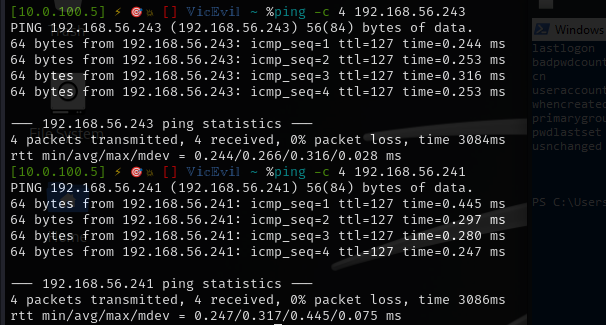


Imagen 8.- Resultados de las distintas comprobaciones de conexión desde Kali con cada una de las máquinas Windows.

- Mediante la herramienta **“PowerView[[17]](#footnote-17)”**, se consultan los usuarios del dominio y su nombre principal en el entorno **Active Directory**, con la finalidad de **ejecutar** un **ticket TGS al KDC** y poder **autenticarnos** en alguno de los **servicios**.

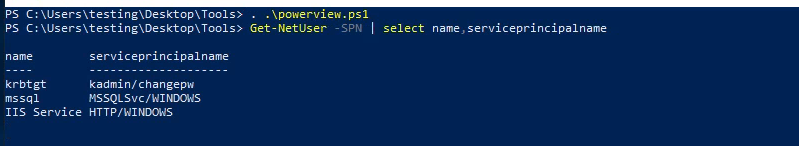


Imagen 9.- Los tres servicios principales obtenidos del entorno Kerberos

- Se verifican los **tres servicios** anteriores, extrayendo cada uno de sus hashes Kerberos para intentar descifrarlo mediante la **herramienta hashcat**, **obteniendo** resultado positivo **únicamente** en el servicio **“iis\_service” (HTTP/WINDOWS)**, consiguiendo su contraseña en texto plano : **LaRosalia2021**



Imagen 10.- Hash Kerberos del servicio “HTTP\WINDOWS” usando PShell

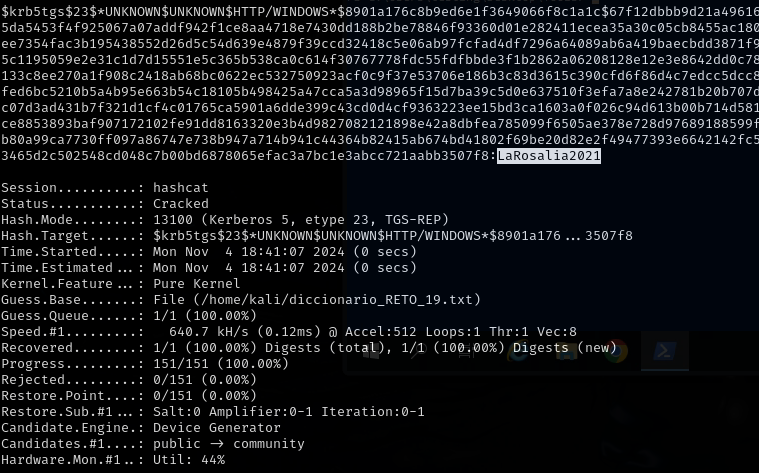
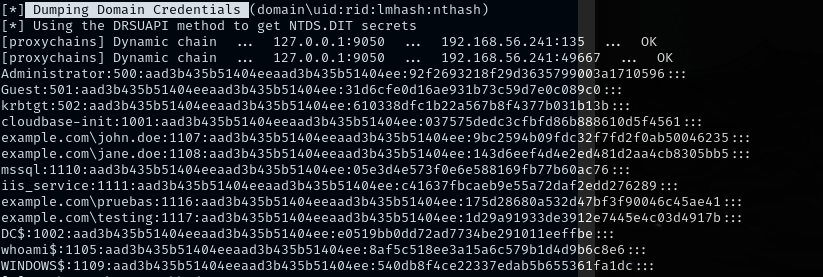


Imagen 11.- resultado del descifrado con la herramienta Hashcat

- Mediante el uso de aplicaciones pertenecientes a la colección de herramientas de Python llamada **Impacket[[18]](#footnote-18),** concretamente: **secretsdump.py**[[19]](#footnote-19) y **wmiexec.py**[[20]](#footnote-20), se han realizado las siguientes acciones:

* Se ha procedido a la búsqueda de **hashes de contraseñas y secretos** del sistema Windows, ejecutando, primeramente, la herramienta **secretsdump.py**, aportando muchísima información, destacando para esta explotación:
* Área **“Dumping Domain Credentials**”, donde se muestran los hashes de NTLM y claves de Kerberos para cada usuario del controlador de dominio del AD.



* Con la información obtenida, se procede a realizar un **ataque pass-the-hash (PTH)**[[21]](#footnote-21) , usando para ello la parte del **hash NT** del hash NTLM del **administrador**[[22]](#footnote-22), a través de la aplicación **“wmiexec.py”** , que permite autenticarse usando la técnica Pass-the-Hash, **obteniendo una conexión remota** no autorizada al sistema objetivo, consiguiendo **autenticarnos** de la **maquina “Windows DC”** con **máximos privilegios** (example\administrator), consiguiendo la finalidad del presente contrato:

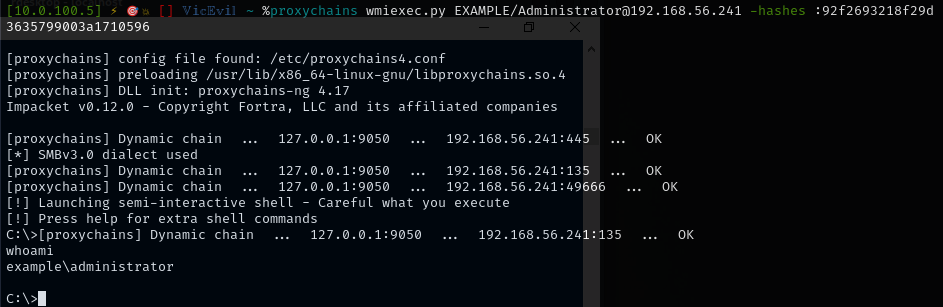


Imagen 12.- Shell remota de la máquina Windows User con privilegios de Administrador

1. **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.

1. RECOMENDACIONES CRÍTICAS:

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

1. **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.

1. **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.

1. **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.

1. 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.***

1. 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.

1. 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[[23]](#footnote-23))***, 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

1. **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.

1. 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.**

1. 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>

1. Security Account Manager) es una base de datos en sistemas Windows que almacena información de cuentas de usuario, incluyendo los hashes de contraseñas, siendo crucial para autenticación en sistemas locales (con hashes LM y NTLM). [↑](#footnote-ref-1)
2. tipo de hash de contraseñas usado en sistemas Windows, basado en el algoritmo MD4 y sin utilizar un "salt", lo que hace que dos contraseñas iguales generen el mismo hash, facilitando ataques de fuerza bruta y de diccionario, siendo relativamente fácil de romper con herramientas como Hashcat o John the Ripper. [↑](#footnote-ref-2)
3. Esto se puede hacer en servidores como Apache usando la directiva Options -Indexes o en Nginx eliminando la opción de autoindex. [↑](#footnote-ref-3)
4. Función de hashing diseñada para ser computacionalmente intensiva, lo que dificulta los ataques de fuerza bruta. Además, bcrypt incorpora salting, un mecanismo que añade datos aleatorios al proceso de hashing, lo que hace que contraseñas idénticas generen hashes diferentes. [↑](#footnote-ref-4)
5. Función de hashing diseñadas igual que la anterior, pero a la que le incorporan para una mayor seguridad el uso de la memoria y numero de núcleos CPU, haciéndolo más resistente a ataques de fuerza bruta y a ataques con hardware especializado. [↑](#footnote-ref-5)
6. función de derivación de clave utilizada para proteger contraseñas que utiliza una función hash repetida muchas veces sobre la contraseña combinada con un valor salt (aleatorio) para generar un hash seguro. La cantidad de repeticiones es configurable, aumentando la resistencia contra ataques de fuerza bruta. [↑](#footnote-ref-6)
7. conjunto de herramientas y bibliotecas en Python que facilita la creación y ejecución de scripts de red para atacar y manipular protocolos de red, especialmente en entornos de Active Directory (AD), capaz de interactuar con varios protocolos de red como SMB, LDAP, y RPC, que son esenciales en entornos Windows y AD. [↑](#footnote-ref-7)
8. característica de seguridad en Windows que protege las credenciales de los usuarios (como hashes NTLM y tickets Kerberos) mediante la virtualización. [↑](#footnote-ref-8)
9. 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. [↑](#footnote-ref-9)
10. El RGPD es un reglamento de la Unión Europea que establece normas estrictas para la protección de datos personales [↑](#footnote-ref-10)
11. La LOPDGDD es ley española que desarrolla el RGPD y que establece normas específicas para la protección de datos personales en España [↑](#footnote-ref-11)
12. 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., [↑](#footnote-ref-12)
13. 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. [↑](#footnote-ref-13)
14. 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. [↑](#footnote-ref-14)
15. 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. [↑](#footnote-ref-15)
16. 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. [↑](#footnote-ref-16)
17. herramienta cuya principal función es **enumerar y mapear entornos de Active Directory (AD)**, proporcionando información detallada sobre usuarios, grupos, permisos, políticas y configuraciones del dominio. [↑](#footnote-ref-17)
18. permite realizar diversas tareas pentesting en redes Windows, pudiendo interactuar con protocolos de red como SMB, RDP, LDAP, Kerberos, entre otros, facilitando acciones como la extracción de hashes, ejecución remota de comandos, y ataques basados en autenticación. [↑](#footnote-ref-18)
19. **secretsdump.py**.- Extrae hashes de contraseñas del sistema desde SAM, NTDS.dit y LSA, siendo ideal para obtener credenciales de usuarios y hashes de NTLM. [↑](#footnote-ref-19)
20. **wmiexec.py.**- Permite ejecutar comandos en sistemas remotos a través de WMI usando credenciales o hashes, siendo útil para ejecutar comandos en máquinas con privilegios administrativos. [↑](#footnote-ref-20)
21. **Pass-the-Hash (PtH)** es una técnica de ataque que permite a un atacante autenticarse en un sistema Windows remoto, usando el hash NTLM de la contraseña en lugar de la contraseña en texto claro. [↑](#footnote-ref-21)
22. “Administrator:500:aad3b435b51404eeaad3b435b51404ee:92f2693218f29d3635799003a1710596:::” [↑](#footnote-ref-22)
23. 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. [↑](#footnote-ref-23)