

Guía para Validar Configuración Servidor SSH acorde al marco de Ciberseguridad NIST.



**SEGURIDAD Y
PRIVACIDAD DE
LA INFORMACIÓN**

HISTORIAL DE CAMBIOS

Versión	Fecha	Descripción
1.0.0	29/10/2023	Versión inicial del documento
2.0.0	15/11/2023	Versión final

Índice

Índice.....	3
Lista ilustraciones.....	5
1. DERECHOS DE AUTOR	6
2. PÚBLICO OBJETIVO.....	7
3. INTRODUCCIÓN.....	8
4. ALCANCE.....	10
5. DEFINICIONES.....	11
6. VALIDACIÓN ESCENARIO CON MARCO NIST.....	12
6.1. ESCENARIO 1 - CONFIGURACIÓN PREDETERMINADA	12
6.1.1. CONFIGURACIÓN ENCRIPCIÓN CANAL EN ESCENARIO	12
6.1.1.1. ENCRIPCIÓN CANAL	12
6.1.1.2. ESTANDAR SOBRE CANAL ENCRIPADO	12
6.1.1.3. PRUEBA CANAL ENCRIPADO	12
6.1.1.4. RESULTADO ESPERADO DEL CANAL ENCRIPADO	13
6.1.1.5. RESULTADO OBTENIDO DEL CANAL ENCRIPADO	13
6.1.2. CONFIGURACIÓN EJECUCIÓN COMANDOS (cmd / terminal)	14
6.1.2.1. EJECUCIÓN COMANDOS.....	14
6.1.2.2. ESTANDAR SOBRE EJECUCIÓN COMANDOS	14
6.1.2.3. PRUEBA EJECUCIÓN COMANDOS	15
6.1.2.4. RESULTADO ESPERADO DE EJECUCIÓN COMANDOS	15
6.1.2.5. RESULTADO OBTENIDO DE EJECUCIÓN COMANDOS	15
6.1.3. CONFIGURACIÓN INTERCAMBIO DE CLAVES.....	18
6.1.3.1. INTERCAMBIO DE CLAVES.....	18
6.1.3.2. ESTANDAR SOBRE INTERCAMBIO DE CLAVES.....	18
6.1.3.3. PRUEBA INTERCAMBIO DE CLAVES.....	18
6.1.3.4. RESULTADO ESPERADO DE INTERCAMBIO DE CLAVES	19
6.1.3.5. RESULTADO OBTENIDO DE INTERCAMBIO DE CLAVES.....	19
6.1.4. CONFIGURACIÓN TRANSFERENCIA DE ARCHIVOS SFTP	19
6.1.4.1 TRANSFERENCIA DE ARCHIVOS SFTP	19
6.1.4.2. ESTANDAR DE ARCHIVOS SFTP	20
6.1.4.3. PRUEBA DE TRANSFERENCIA DE ARCHIVOS SFTP	20
6.1.4.4. RESULTADO ESPERADO	20
6.1.4.5. RESULTADO OBTENIDO	20
6.1.5. CONFIGURACIÓN SOPORTE CAMBIO DE CLAVES	21
6.1.5.1. ENCRIPCIÓN CAMBIO DE CLAVES	21

6.1.5.2. ESTANDAR SOBRE CAMBIO DE CLAVES	22
6.1.5.3. PRUEBA CAMBIO DE CLAVES	22
6.1.5.4. RESULTADO ESPERADO DE CAMBIO DE CLAVES.....	22
6.1.5.5. RESULTADO OBTENIDO DE CAMBIO DE CLAVES	22
6.1.6. CONFIGURACIÓN SISTEMA AUTENTICACIÓN FUERTE	24
6.1.6.1. ENCRIPCIÓN FUERTE O DSA.....	24
6.1.6.2. ESTANDAR SOBRE ENCRIPCIÓN FUERTE O DSA.....	24
6.1.6.3. PRUEBA ENCRIPCIÓN FUERTE O DSA.....	24
6.1.6.4. RESULTADO ESPERADO DE ENCRIPCIÓN FUERTE O DSA.....	24
6.1.6.5. RESULTADO OBTENIDO DE ENCRIPCIÓN FUERTE O DSA.....	24
6.1.7. CONFIGURACIÓN DEL ATAQUE DE HOMBRE EN EL MEDIO.....	25
6.1.7.1. ENCRIPCIÓN DEL HOMBRE EN EL MEDIO	25
6.1.7.2. ESTANDAR SOBRE HOMBRE EN EL MEDIO	25
6.1.7.3. PRUEBA SOBRE HOMBRE EN EL MEDIO.....	25
6.1.7.5. RESULTADO OBTENIDO DE HOMBRE EN EL MEDIO	26
6.2. ESCENARIO 2 - CONFIGURACIÓN ALTO NIVEL NIST	30
7. PREGUNTAS FRECUENTES.....	31
¿Qué es necesario para poder llevar a cabo un laboratorio semejante?	31
¿Se puede usar una máquina de ubuntu desktop y luego volverla servidor?	31
¿Desde cuál perspectiva de atacante, cliente o servidor se mira la práctica de laboratorio?	31
¿Por qué se utiliza la clave encriptada entre el cliente y el servidor?	31
¿Cuáles herramientas se recomienda para analizar el tráfico y paquetes enviados?	31
¿Qué sucede que al realizar una conexión desde la terminal por medio de plink de putty muestra set de caracteres indeseados como cntrl, backsp, [01:34,Public entre otros?	32
¿Cómo puedo cambiar la configuración del servidor SSH?	32
¿Cómo ver los usuarios conectados y comando para desconectarlos remotamente?	32
8. INFORMACIÓN DE LOS AUTORES.....	33
LUZ EUGENIA MUÑOZ LONDOÑO.....	33
JULIÁN ANDRÉS PEÑA RÚA	33
MARIA EUGENIA GONZÁLEZ PÉREZ	33
HÉCTOR FERNANDO VARGAS MONTOYA	33

Lista ilustraciones

Ilustración 1. Captura intercambio de claves con aplicativo monitoreo redes WireShark. Elaboración propia.	13
Ilustración 2. Paquete respuesta del servidor al cliente con información de intercambio de claves Diffie-Hellman.	14
Ilustración 3. Archivo lista de contraseñas y usuarios posibles. Fuente propia.	15
Ilustración 4. Captura mensaje de inicio ejecución Hydra. Fuente propia.	16
Ilustración 5. Captura mensaje finalización ejecución Hydra. Fuente propia.	16
Ilustración 6. Captura mensaje de ejecución Hydra. Fuente propia.	16
Ilustración 7. Conexión remota a servidor SSH por medio de Putty. Elaboración propia.	17
Ilustración 8. Captura de comando update en terminal. Elaboración propia.	17
Ilustración 9. Ejecución de comando apt list –upgradable para ver lista de actualizaciones pendientes. Elaboración propia.	18
Ilustración 10. Captura de monitoreo de redes paquete cliente. Elaboración propia.	19
Ilustración 11. Transferencia de archivos – SFTP con WinSCP	21
Ilustración 12. Captura monitoreo tráfico sftp con wireshark. Elaboración propia.	21
Ilustración 13. Muestra dos tipos de autenticación de clave pública y contraseña. Elaboración propia.	23
Ilustración 14. Captura entrega paquetes con llaves. Elaboración propia.	25
Ilustración 15. Muestra si la clave publica del Servidor SSH es auténtica. Elaboración propia.	26
Ilustración 16. Ping-cliente-servidor. Elaboración propia.	26
Ilustración 17. Ping-cliente-ubuntu. Elaboración propia.	27
Ilustración 18. Ping-cliente-servidor-kali Linux. Elaboración propia.	27
Ilustración 19. Muestra Man in the middle-kali-Linux. Elaboración propia.	28
Ilustración 20. Conexión-ssh-wireshark-kali-Linux. Elaboración propia.	29

1. DERECHOS DE AUTOR

Todas las referencias a los documentos de la Configuración segura con SSHv2: Reducción riesgos de conectividad con derechos reservados por parte del Grupo de Egresados Programadores de la Institución Universitaria Salazar y Herrera (GEPI).

Para el desarrollo de esta guía, se recogieron aspectos importantes de mejores prácticas y documentos de uso libre, tomando como base los lineamientos recomendados en Norma la ISO IEC 27005 – 2009, y la ley 1712 de 2014 por medio de la cual se crea la Ley de Transparencia y del Derecho de Acceso a la Información Pública Nacional y se dictan otras disposiciones.

2. PÚBLICO OBJETIVO

Entidades públicas de orden nacional y entidades públicas del orden territorial, así como proveedores de servicios de Gobierno en Línea, y terceros que deseen adoptar el Modelo de Seguridad y Privacidad de TI en el marco de la Estrategia de Gobierno en Línea.

3. INTRODUCCIÓN

Esta guía entrega los lineamientos básicos que deben ser utilizados por los responsables de la seguridad de la información, para iniciar la verificación y rectificación en la configuración de los servidores SSH, con el fin de determinar en nivel de cumplimiento basado en la escala del marco de ciberseguridad del estándar ISO 27001 determinando si el escenario cuenta con un registro de eventos y acciones de los usuarios, con restricción de acceso a carpetas y permisos, cantidad intentos limitados para iniciar sesión, tiempo de espera tras agotar intentos limitados y otros elementos que se describirán más adelante.

La realización de una verificación y rectificación de la configuración de los servidores SSH hace parte del debido proceso que a nivel estratégico se ha definido en el Modelo de Seguridad y Privacidad de la información con respecto a la seguridad de Configuración segura con SSHv2: Reducción riesgos de conectividad de los procesos de una entidad, y cuyo objetivo es dar cumplimiento a los puntos principales descritos en la guía de controles del Anexo A del estándar ISO/IEC 27001:2013

- A.6. Organización de la seguridad de la información
- A.9. Control de acceso
- A.10. Criptografía
- A.12. Seguridad de las operaciones
- A.13. Seguridad de las comunicaciones

Adicionalmente, se tiene en cuenta una escala de medición contenida en la tabla 1.

Tabla 1. Escala de evaluación para la validación de controles del instrumento evaluación MSPI. Elaborado por MinTIC.

Descripción	Calificación	Criterio
No Aplica	N/A	No aplica.
Inexistente	0	Total falta de cualquier proceso reconocible. La Organización ni siquiera ha reconocido que hay un problema a tratar. No se aplican controles.
Inicial	20	1) Hay una evidencia de que la Organización ha reconocido que existe un problema y que hay que tratarlo. No hay procesos estandarizados. La implementación de un control depende de cada individuo y es principalmente reactiva. 2) Se cuenta con procedimientos documentados, pero no son conocidos y/o no se aplican.
Repetible	40	Los procesos y los controles siguen un patrón regular. Los procesos se han desarrollado hasta el punto en que diferentes procedimientos son seguidos por diferentes personas. No hay formación ni comunicación formal sobre los procedimientos y estándares. Hay un alto grado de confianza en los conocimientos de cada persona, por eso hay probabilidad de errores.
Efectivo	60	Los procesos y los controles se documentan y se comunican. Los controles son efectivos y se aplican casi siempre . Sin embargo, es poco probable la detección de desviaciones, cuando el control no se aplica oportunamente o la forma de aplicarlo no es la indicada.
Gestionado	80	Los controles se monitorean y se miden. Es posible monitorear y medir el cumplimiento de los procedimientos y tomar medidas de acción donde los procesos no estén funcionando eficientemente.
Optimizado	100	Las buenas prácticas se siguen y automatizan . Los procesos han sido redefinidos hasta el nivel de mejores prácticas , basándose en los resultados de una mejora continua .

4. ALCANCE

La verificación y rectificación de la configuración del cliente- servidor se realizó a través de pruebas pentesting, buscando conocer si cumplan controles de procesos en las actividades de protección de datos alineada a la norma ISO 27001:2013, NIST 800-53, RFC 4253 y la guía de controles del modelo de seguridad y privacidad de la información.

5. DEFINICIONES

Peritaje informático: Se conoce como peritaje informático a los estudios e investigaciones orientados a la obtención de una prueba informática de aplicación en un asunto judicial para que sirva a un juez para decidir sobre la culpabilidad o inocencia de una de las partes.

NIST: (National Institute of Standards and Technology), es una agencia no reguladora que promueve la innovación mediante el fomento de la ciencia, los estándares y la tecnología de la medición. El marco de ciberseguridad de la NIST consta de estándares, pautas y mejores prácticas que ayudan a las organizaciones a mejorar su gestión de riesgos de ciberseguridad.

6. VALIDACIÓN ESCENARIO CON MARCO NIST

6.1. ESCENARIO 1 - CONFIGURACIÓN PREDETERMINADA

Para una prueba de peritaje en un escenario SSH donde se utiliza la configuración predeterminada, la cual es aquella que se encuentra luego de instalar OpenSSH y que no modifica el archivo de configuración, este escenario posibilita ataques para obtener la siguiente información:

- SERVIDOR Conocer la IP
- Puerto utilizado (predeterminado el 22)
- Servicios abiertos (SFTP para Web, Correo, Bases datos)

CLIENTE

- Conocer la IP
- Conocer nombre usuario
- Conocer contraseña

6.1.1. CONFIGURACIÓN ENCRIPCIÓN CANAL EN ESCENARIO

6.1.1.1. ENCRIPCIÓN CANAL

Enmascarar el canal de transmisión de datos entre el cliente y el servidor es crucial para garantizar la confidencialidad e integridad de la información transmitida. El cifrado protege los datos de posibles interceptaciones por parte de terceros malintencionados, lo que ayuda a prevenir la fuga de información sensible y reduce el riesgo de manipulación de datos durante la transmisión. En el contexto de SSH, esto es esencial para asegurar que las credenciales de inicio de sesión y cualquier otra información sensible no sean vulnerables a ataques de intermediarios o escuchas no autorizadas. Por defecto, luego de instalar OpenSSH, el canal es encriptado al establecer una conexión remota entre Cliente-Servidor.

6.1.1.2. ESTANDAR SOBRE CANAL ENCRIPTADO

Acorde a la NIST y la ISO 27001 A.13.2.1 "Se debe contar con políticas, procedimientos y controles de transferencia formales para proteger la transferencia de información mediante el uso de todo tipo de instalaciones de comunicación". Donde las cuales se tienen en cuenta algunas directrices como:

- a) definir los procedimientos diseñados para proteger la información transferida contra interceptación, copiado, modificación, enrutado y destrucción;
- c) definir los procedimientos para proteger información electrónica sensible comunicada que están como adjuntos;
- f) establecer el uso de técnicas criptográficas, (proteger la confidencialidad, la integridad y la autenticidad de la información).
- j) no dejar mensajes que contengan información confidencial, en las máquinas contestadoras, ya que éstos pueden ser escuchados por personas no autorizadas, almacenados en sistemas comunales o almacenados incorrectamente como resultado de una marcación incorrecta;

6.1.1.3. PRUEBA CANAL ENCRIPTADO

Ataques de "hombre en el medio" (MITM): un atacante intenta interceptar la conexión SSH para robar información confidencial. La encriptación del canal de conexión SSH proporciona una protección contra este tipo de ataque porque durante la sesión todos los paquetes son encriptados de acuerdo al algoritmo de cifrado acordado.

6.1.1.4. RESULTADO ESPERADO DEL CANAL ENCRYPTADO

En un ataque MITM contra SSH con configuración predeterminada, el atacante podría interceptar la comunicación entre el cliente y el servidor SSH sin ser detectado. Esto permitiría al atacante capturar las credenciales de inicio de sesión, contraseñas y cualquier otra información transmitida a través de la conexión SSH. Posteriormente, el atacante podría usar estas credenciales para acceder al servidor de manera no autorizada, una ventaja al ser enviados los paquetes de información desde el protocolo SSH es que estos viajarán en un canal cifrado con un algoritmo específico y los paquetes de datos también se encontrarán encriptados.

Se espera en este resultado que tanto el servidor como el cliente establezcan un acuerdo de algoritmos criptográficos los cuales les permitan enviar y recibir datos, evidenciándose a través del acuerdo de intercambio de claves Diffie-Hellman. En un aplicativo de supervisión y monitoreo de redes se podrá notar en sus logs información alusiva al intercambio de claves disponibles y el acuerdo Diffie-Hellman y su respectivo algoritmo de cifrado.

6.1.1.5. RESULTADO OBTENIDO DEL CANAL ENCRYPTADO

Se ejecutó la prueba del hombre del medio donde se pudo monitorear satisfactoriamente todo el proceso de conexión al servidor SSH. Siendo el cliente la IP 192.168.214.1 y el servidor la IP 192.168.214.128.

En la ilustración 1 Se puede observar 3 filas que se encuentran sombreadas con color AZUL y cuyo protocolo mencionado es SSHv2, en este se muestran 3 momentos donde se intercambian claves primero el cliente, luego servidor y por último el acuerdo Diffie Hellman.

Source	Destination	Protocol	Length	Info
192.168.214.1	192.168.214.128	TCP	1514	51486 → 22 [ACK] Seq=29 Ack=42 Win=131328 Len=1460 [
192.168.214.1	192.168.214.128	SSHv2	90	Client: Key Exchange Init
192.168.214.1	192.168.214.128	TCP	1514	[TCP Retransmission] 51486 → 22 [ACK] Seq=29 Ack=42
192.168.214.1	192.168.214.128	TCP	90	[TCP Retransmission] 51486 → 22 [PSH, ACK] Seq=1489
192.168.214.1	192.168.214.128	TCP	1514	[TCP Retransmission] 51486 → 22 [PSH, ACK] Seq=65 Ac
192.168.214.1	192.168.214.128	TCP	1514	[TCP Retransmission] 51486 → 22 [PSH, ACK] Seq=65 Ac
192.168.214.128	192.168.214.1	TCP	66	22 → 51486 [ACK] Seq=42 Ack=1525 Win=64000 Len=0 SLE
192.168.214.128	192.168.214.1	TCP	66	[TCP Dup ACK 448#1] 22 → 51486 [ACK] Seq=42 Ack=1525
192.168.214.128	192.168.214.1	SSHv2	1134	Server: Key Exchange Init
192.168.214.128	192.168.214.1	TCP	1134	[TCP Retransmission] 22 → 51486 [PSH, ACK] Seq=42 Ac
192.168.214.1	192.168.214.128	SSHv2	1262	Client: Diffie-Hellman Key Exchange Init
192.168.214.1	192.168.214.128	TCP	1262	[TCP Retransmission] 51486 → 22 [PSH, ACK] Seq=1525
192.168.214.128	192.168.214.1	TCP	60	22 → 51486 [ACK] Seq=1122 Ack=2733 Win=64128 Len=0

Ilustración 1. Captura intercambio de claves con aplicativo monitoreo redes Wireshark. Elaboración propia.

Si ampliamos la información obtenida entre los paquetes espiados por medio del ataque del hombre en el medio y supervisando con el aplicativo de monitoreo Wireshark podremos ver que existen varios mensajes entre el cliente y servidor pertenecientes al Diffie-Hellman donde se comparten las llaves algorítmicas acordadas.

*eth0						
Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda						
ip.addr -- 192.168.214.1						
No.	Time	Source	Destination	Protocol	Length	Info
453	22:22:36.213785531	192.168.214.1	192.168.214.128	TCP	1262	[TCP Retransmission]
454	22:22:36.214087659	192.168.214.128	192.168.214.1	TCP	60	22 → 51486 [ACK] Seq
455	22:22:36.223255749	192.168.214.128	192.168.214.1	SSHv2	1514	Server: Diffie-Hellman
456	22:22:36.223255789	192.168.214.128	192.168.214.1	SSHv2	178	Server: Encrypted pa
457	22:22:36.225699763	192.168.214.128	192.168.214.1	TCP	54	22 → 51486 [ACK] Seq
458	22:22:36.225788223	192.168.214.128	192.168.214.1	TCP	1514	[TCP Out-Of-Order] 2
459	22:22:36.225876772	192.168.214.128	192.168.214.1	TCP	178	[TCP Retransmission]
460	22:22:36.225956456	192.168.214.1	192.168.214.128	TCP	60	51486 → 22 [ACK] Seq
461	22:22:36.232724277	192.168.214.1	192.168.214.128	SSHv2	134	Client: New Keys, En
462	22:22:36.233649281	192.168.214.1	192.168.214.128	TCP	54	51486 → 22 [ACK] Seq
463	22:22:36.233702881	192.168.214.1	192.168.214.128	TCP	134	[TCP Retransmission]
464	22:22:36.234089068	192.168.214.128	192.168.214.1	SSHv2	118	Server: Encrypted pa
465	22:22:36.245715822	192.168.214.128	192.168.214.1	TCP	118	[TCP Retransmission]
TCP payload (1460 bytes)						
SSH Protocol						
SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)						
Packet Length: 1228						
Padding Length: 9						
Key Exchange (method:sntrup761x25519-sha512@openssh.com)						
Message Code: Diffie-Hellman Key Exchange Reply (31)						
KEX host key (type: ssh-ed25519)						
Multi Precision Integer Length: 1071						
DH server f: ac0e64aa52372652e8124fa453fab86160da95152b64c390a499209e757f9bd2eb66c163...						
KEX H signature length: 83						
KEX H signature: 0000000b7373682d6564323535313900000040513b4ab0b9810b9163bedfb9472e69cafe...						
Padding String: 00000000000000000000						
SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)						
Packet Length: 12						
Padding Length: 10						
Key Exchange (method:sntrup761x25519-sha512@openssh.com)						
Message Code: New Keys (21)						
Padding String: 00000000000000000000						
SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)						
Packet Length (encrypted): a5bd7e23						
Encrypted Packet: 9b2674b9b9cd7cd626461a7dc29cc6eb188e93e58d3e45b0abe9edba78990bcc93fbf529...						
MAC: 83717257f56d1599d542a1f0017f77caaa42452dc37993aef1d11f92e53aaab0						
[Direction: server-to-client]						
0020	d6 01 00 16 c9 1e 97 f3	c6 a5 37 e8 13 4f 50 10	...	...	...	...
0030	01 f5 99 60 00 00 00 00	04 cc 09 1f 00 00 00 33	...	...	...	...
0040	00 00 00 0b 73 73 68 2d	65 64 32 35 35 31 39 00	...	ssh-	...	...
0050	00 00 20 1d 33 9e 80 43	ce 60 12 35 09 e4 9c cd	...	3..C	...	...
0060	36 39 70 01 3c 76 1a 08	f5 13 b1 2b 76 26 00 f1	69p	<v..	...	...
0070	c8 12 97 00 00 04 2f ac	0e 64 aa 52 37 26 52 e8	...	...	...	...
Destination Port (tcp.dstport), 2 byte(s)						

Ilustración 2. Paquete respuesta del servidor al cliente con información de intercambio de claves Diffie-Hellman.

En este punto se puede evidenciar que tanto el cliente como el servidor están compartiendo mensajes que contienen el tipo de encriptación (aes256-ctr, mac:hmac-sha2-256) y que no se encuentra comprimida. Adicionalmente podemos también notar que comparte 2 paquetes, uno donde menciona que se emite la respuesta al Diffie-Hellman y otra donde comparte las nuevas llaves.

6.1.2. CONFIGURACIÓN EJECUCIÓN COMANDOS (cmd / terminal)

6.1.2.1. EJECUCIÓN COMANDOS

Es la capacidad de realizar acciones como iniciar programas o eliminar archivos desde la terminal.

6.1.2.2. ESTANDAR SOBRE EJECUCIÓN COMANDOS

En el ISO 27001 Anexo 12 “Seguridad de las operaciones”, menciona para asegurar las operaciones correctas y seguras de las instalaciones de procesamiento de información debe revisar los procedimientos de operación con instrucciones operacionales, que incluye instalar y configurar sistemas. Adicionalmente debe tener en cuenta la separación de los ambientes de desarrollo, pruebas y operación para establecer que el software de desarrollo y de operaciones debe funcionar en diferentes sistemas o procesadores de computador y en diferentes dominios o directorios. Establecer que los usuarios deben usar diferentes perfiles de usuario para sistemas operacionales y de pruebas, y los menús deben desplegar mensajes de identificación apropiados para reducir el riesgo de error.

Revisar las directrices para control de software operacional como actualizar el software operacional, aplicaciones y bibliotecas de programas tarea que solo la debe llevar a cabo administradores entrenados, con autorización apropiada de la dirección.

Adicionalmente, la NIST recomienda que para la autenticación existan controles tales como:

- Cantidad intentos
- Tiempo de espera

Esto con el fin de dificultar los diversos ataques de fuerza bruta dado que imposibilita al atacante realizar una gran cantidad de ataques en poco tiempo vulnerando así el servidor rápidamente.

6.1.2.3. PRUEBA EJECUCIÓN COMANDOS

Para revisar si un servidor SSH permite la ejecución de comandos primero deberemos de contar con una cuenta de usuario la cual se pueda utilizar y luego comprobar si tiene este habilitado. En este caso puede realizar alguna de las siguientes prácticas:

- Ataque de Fuerza Bruta: Un atacante intenta adivinar la contraseña o la clave SSH probando repetidamente combinaciones de credenciales hasta que tenga éxito.
- Ataque de Fuerza Bruta por Diccionario: Similar al ataque de fuerza bruta, pero en lugar de probar todas las combinaciones posibles, el atacante utiliza una lista de palabras comunes (diccionario) para adivinar las credenciales.

6.1.2.4. RESULTADO ESPERADO DE EJECUCIÓN COMANDOS

En una configuración predeterminada luego de instalar OpenSSH, es permitida la ejecución de comandos para instalar software operacional, realizar acciones con ficheros como crear, eliminar, buscar o actualizar, listar archivos, entre otras acciones.

6.1.2.5. RESULTADO OBTENIDO DE EJECUCIÓN COMANDOS

Se realiza un ataque de fuerza bruta por diccionario, creando dos ficheros, uno con los posibles usuarios y otro con las posibles contraseñas.

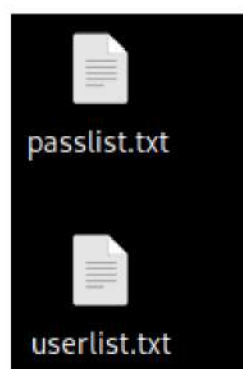


Ilustración 3. Archivo lista de contraseñas y usuarios posibles. Fuente propia.

Para ello se apoya del sistema operativo Kali Linux y se utiliza el programa Hydra en su versión gráfica. Luego de configurar la IP del servidor 192.168.214.128, el puerto y tipo de conexión se inicia el servicio el cual toma menos de 2 minutos en realizar todas las posibles combinaciones, esto se puede evidenciar al comparar la ilustración 4 y 5.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-11-14 13:41:59
[DATA] max 16 tasks per 1 server, overall 16 tasks, 152 login tries (l:19/p:8), ~10 tries per task
[DATA] attacking ssh://192.168.214.128:22/
```

Ilustración 4. Captura mensaje de inicio ejecución Hydra. Fuente propia.

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-14 13:42:32  
<finished>
```

Ilustración 5. Captura mensaje finalización ejecución Hydra. Fuente propia.

Se realizan 152 combinaciones posibles para poder encontrar un usuario válido que permitiera el ingreso al sistema y probar si sus credenciales y permisos permiten la ejecución de comandos de terminal.

```
xHydra  
Salir  
Target Passwords Tuning Specific Start  
[ATTEMPT] target 192.168.214.128 - login "bichoj" - pass "1234" - 130 of 152 [child 14] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichoj" - pass "123456789" - 130 of 152 [child 4] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichoj" - pass "123456" - 131 of 152 [child 11] (0/0)  
[22][ssh] host: 192.168.214.128 login: bichoj password: 123456789  
[ATTEMPT] target 192.168.214.128 - login "bichopena" - pass "abcd" - 142 of 152 [child 5] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichopena" - pass "asdfgh" - 142 of 152 [child 0] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichopena" - pass "abcd" - 142 of 152 [child 5] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichopena" - pass "abcd" - 142 of 152 [child 5] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichopena" - pass "abcdef" - 143 of 152 [child 12] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichopena" - pass "abcdef" - 143 of 152 [child 12] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichopena" - pass "@bcd3" - 144 of 152 [child 10] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "123456789" - 145 of 152 [child 7] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "1234" - 146 of 152 [child 0] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "123456789" - 146 of 152 [child 7] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "1234" - 146 of 152 [child 0] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "123456" - 147 of 152 [child 14] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "asdfgh" - 148 of 152 [child 5] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "asdf" - 149 of 152 [child 6] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "asdf" - 149 of 152 [child 6] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "abcd" - 150 of 152 [child 15] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "abcdef" - 151 of 152 [child 8] (0/0)  
[RE-ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "abcd" - 151 of 152 [child 15] (0/0)  
[ATTEMPT] target 192.168.214.128 - login "bichojulian" - pass "@bcd3" - 152 of 152 [child 12] (0/0)  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-11-14 13:42:32  
<finished>
```

Ilustración 6. Captura mensaje de ejecución Hydra. Fuente propia.

En la ilustración 6 se puede apreciar los intentos (ATTEMPT) y los reintentos (RE-ATTEMPT) de acceso al servidor y sus posibles combinaciones. Al final, antes del mensaje de finalizado y la estampa de hora, se puede apreciar un mensaje donde dice que se ha encontrado exitosamente una contraseña válida. La contraseña válida es aquella que se encuentra con el texto en negrilla, es decir, las credenciales login: bichoj y password: 123456789 en el servidor ssh puerto 22.

```
bicho@bichojserv: ~  
login as: bicho  
bicho@192.168.214.128's password:  
Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 6.2.0-36-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
Expanded Security Maintenance for Applications is not enabled.  
  
17 updates can be applied immediately.  
2 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
2 updates could not be installed automatically. For more details,  
see /var/log/unattended-upgrades/unattended-upgrades.log  
Last login: Tue Nov 14 14:29:47 2023 from 192.168.214.130  
bicho@bichojserv:~$
```

Ilustración 7. Conexión remota a servidor SSH por medio de Putty. Elaboración propia.

Al utilizar credenciales para una conexión remota a través de Putty evidenciaremos si realmente dichos credenciales funcionan. En la ilustración 7 se puede apreciar que la conexión se realizó con éxito. Luego procedemos a probar algún comando como actualizar el sistema, ejecutar alguna aplicación o crear archivos.

```
bicho@bichojserv: ~  
bicho@bichojserv:~$ sudo apt update  
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease  
Hit:2 http://us.archive.ubuntu.com/ubuntu jammy InRelease  
Hit:3 http://us.archive.ubuntu.com/ubuntu jammy-updates InRelease  
Hit:4 http://us.archive.ubuntu.com/ubuntu jammy-backports InRelease  
Hit:5 https://ppa.launchpadcontent.net/wireshark-dev/stable/ubuntu jammy InRelease  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
19 packages can be upgraded. Run 'apt list --upgradable' to see them.  
bicho@bichojserv:~$
```

Ilustración 8. Captura de comando update en terminal. Elaboración propia.

Como se observa en la ilustración 8 se ha ejecutado un comando update con credenciales de administrador (sudo) que luego de realizar 5 intentos (Hit) finaliza con un mensaje el cual menciona que tiene 19 paquetes que pueden ser actualizados y ejecutando el comando “apt list –upgradable” podrá verlos.

Se ejecuta el comando indicado y se puede apreciar cada uno de los elementos que pueden ser actualizados del servidor Ubuntu (ver ilustración 9).

```

bicho@bichojserv:~$ apt list --upgradable
Listing... Done
bind9-dnsutils/jammy-updates 1:9.18.18-0ubuntu0.22.04.1 amd64 [upgradable from: 1:9.18.12-0ubuntu0.22.04.1]
bind9-host/jammy-updates 1:9.18.18-0ubuntu0.22.04.1 amd64 [upgradable from: 1:9.18.12-0ubuntu0.22.04.1]
bind9-libs/jammy-updates 1:9.18.18-0ubuntu0.22.04.1 amd64 [upgradable from: 1:9.18.12-0ubuntu0.22.04.1]
distro-info-data/jammy-updates,jammy-updates 0.52ubuntu0.5 all [upgradable from: 0.52ubuntu0.4]
firmware-sof-signed/jammy-updates,jammy-updates 2.0-1ubuntu4.2 all [upgradable from: 2.0-1ubuntu4.1]
gjs/jammy-updates 1.72.4-0ubuntu0.22.04.1 amd64 [upgradable from: 1.72.2-0ubuntu1]
libgjs0g/jammy-updates 1.72.4-0ubuntu0.22.04.1 amd64 [upgradable from: 1.72.2-0ubuntu1]
libnss-systemd/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
libpam-systemd/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
libprocps8/jammy-updates,jammy-security 2:3.3.17-6ubuntu2.1 amd64 [upgradable from: 2:3.3.17-6ubuntu2]
libsystemd0/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
libsystemd0/jammy-updates 249.11-0ubuntu3.11 i386 [upgradable from: 249.11-0ubuntu3.10]
libudev1/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
procps/jammy-updates,jammy-security 2:3.3.17-6ubuntu2.1 amd64 [upgradable from: 2:3.3.17-6ubuntu2]
systemd-oomd/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
systemd-sysv/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
systemd-timesyncd/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
systemd/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
udev/jammy-updates 249.11-0ubuntu3.11 amd64 [upgradable from: 249.11-0ubuntu3.10]
bicho@bichojserv:~$

```

Ilustración 9. Ejecución de comando `apt list --upgradable` para ver lista de actualizaciones pendientes. Elaboración propia.

Esto lleva a la conclusión de que al tener una configuración predeterminada al no impedir o dificultar accesos de fuerza bruta se obtienen las credenciales de un usuario, al ser permitida la ejecución de comandos se logra revisar actualizaciones pendientes y realizar actualizaciones pudiendo gestionar estas desde repositorios fraudulentos o con fines maliciosos al modificar las librerías del sistema. Acorde a los estándares tenidos en cuenta se considera que como calificación que hay una evidencia de que la Organización ha reconocido que existe un problema y que hay que tratarlo, pero no hay procesos estandarizados.

6.1.3. CONFIGURACIÓN INTERCAMBIO DE CLAVES

6.1.3.1. INTERCAMBIO DE CLAVES

Se refiere al proceso mediante el cual el cliente y el servidor establecen una clave compartida de forma segura para cifrar la comunicación subsiguiente. Esto se realiza para proteger la confidencialidad e integridad de los datos durante la transmisión.

Es crucial que este intercambio de claves esté cifrado porque, de lo contrario, existe el riesgo de que un atacante intercepte las claves durante el proceso y pueda comprometer la seguridad de la conexión.

6.1.3.2. ESTÁNDAR SOBRE INTERCAMBIO DE CLAVES

El estándar que se usa para este intercambio es el Diffie-Hellman, que es un protocolo de intercambio de claves que permite a dos partes acordar de manera segura una clave de sesión compartida. Es utilizado para generar claves públicas y privadas, hay que tener muy presente la longitud de las claves. La más indicada según el estándar RFC 4253 (SSH Transport Layer Protocol), el cual se centra en la capa de transporte SSH, para este intercambio es de 2048 Bits y haciendo más difícil descifrar las claves que fueron enviadas entre el cliente y servidor. Adicionalmente recomienda evitar utilizar algoritmos de intercambio de claves débiles o desactualizados.

6.1.3.3. PRUEBA INTERCAMBIO DE CLAVES

Ataque de hombre en el medio: En un ataque del hombre en el medio se puede ver a dos máquinas que interactúan entre sí haciendo la petición de autenticación para la conexión SSH, la cual incluye el proceso de realizar el intercambio de claves de algorítmicas entre el servidor y el cliente que será usada durante toda la sesión. Este intercambio de claves algorítmicas garantiza que ambas partes saben con quien están intercambiando.

6.1.3.4. RESULTADO ESPERADO DE INTERCAMBIO DE CLAVES

Cuando hay una petición de lado del cliente, el sistema comprueba a través de un intercambio de claves que exista una comunicación transversal entre ambas máquinas. Si las dos claves algorítmicas coinciden, el servidor SSH permite establecer el canal para la conexión sin ningún problema.

6.1.3.5. RESULTADO OBTENIDO DE INTERCAMBIO DE CLAVES

Se realiza un ataque del hombre en el medio utilizando Kali Linux buscando supervisar el tráfico de red que tienen el cliente y el servidor analizando paquete por paquete enviado buscando identificar el intercambio de claves y cumplimiento del estándar Diffie-Hellman.

The image shows a network packet capture analysis. The top section displays a list of network packets with columns for time, source IP, destination IP, protocol, and details. Packet 442 is highlighted, showing a TCP ACK segment. Below this, the packet details for the SSH protocol are expanded, showing the SSH Version 2 key exchange process. The details include the packet length, padding, and the key exchange method (sntrup761x25519-sha512@openssh.com). The algorithms section lists the cookie, key exchange algorithms, server host key algorithms, encryption algorithms, and MAC algorithms. The bottom section shows the raw packet data in hexadecimal and ASCII format, with a frame size of 90 bytes and a reassembled TCP size of 1496 bytes.

Ilustración 10. Captura de monitoreo de redes paquete cliente. Elaboración propia.

En la ilustración 10 se aprecia alguno de los paquetes obtenidos a través del monitoreo de redes, este paquete de cliente envía algoritmos de cifrado disponibles para acuerdo Diffie-Hellman. Se puede observar el protocolo que fue usado que en este caso es SSH, también se observa el canal encriptado y su algoritmo que es el aes256.

Al implementar este protocolo, es importante tener en cuenta las recomendaciones de seguridad y elegir el tamaño de clave adecuado para garantizar una comunicación segura.

6.1.4. CONFIGURACIÓN TRANSFERENCIA DE ARCHIVOS SFTP

6.1.4.1 TRANSFERENCIA DE ARCHIVOS SFTP

Este protocolo permite transferir datos cifrados entre tu ordenador local y el espacio web del que dispone en un host a través de Secure Shell (SSH).

6.1.4.2. ESTANDAR DE ARCHIVOS SFTP

En este escenario predeterminado según la capa de transporte SSH RFC 4253, el cual ha sido diseñado para ser simple y flexible, debe permitir negociación de parámetros y minimizar el número de viajes de ida y vuelta. A continuación, se menciona algunas instrucciones:

- a). El algoritmo de autenticación de mensajes y el algoritmo hash son todo negociado.
- b). Se espera que, en la mayoría de los entornos, se necesitarán 2 viajes de ida y vuelta para el intercambio completo de claves.
- c). El peor de los casos son 3 viajes de ida y vuelta.

Acorde con la ISO 27001 en el anexo 13. "Transferencia de información" buscando mantener la seguridad de la información transferida dentro de una organización y con cualquier entidad externa. Con la cual se debe contar con políticas, procedimientos y controles de transferencia para proteger el envío y captura de información mediante el uso de todo tipo de instalación de comunicaciones.

Se recomienda definir los procedimientos diseñados para proteger la información transferida contra interceptación, copiado, modificación, enrutado y destrucción; establecer el uso de técnicas criptográficas, (proteger la confidencialidad, la integridad y la autenticidad de la información).

6.1.4.3. PRUEBA DE TRANSFERENCIA DE ARCHIVOS SFTP

Ataques de "hombre en el medio" (MITM): un atacante intenta interceptar la conexión SSH para robar información confidencial. La encriptación de los paquetes enviados a través de la conexión SSH proporciona una protección contra este tipo de ataque.

6.1.4.4. RESULTADO ESPERADO

Validar una transferencia de archivos SFTP desde la máquina atacante (kali linux), utilizando un software motor o supervisor de redes (wireshark), capturando los paquetes enviados en la conexión remota los cuales son altamente susceptibles a ser capturados y leídos fácilmente si su algoritmo de cifrado es débil. El cliente realiza una conexión desde un programa como FileZilla o WinSCP. La conexión se realiza desde el puerto 22 que está habilitado en el servidor SSH y realiza transferencia de archivos subiendo y descargando.

El resultado esperado en la transferencia de archivos SFTP es una transferencia de archivos segura y exitosa entre cliente - servidor; se comprueba desde un supervisor de tráfico de redes si los archivos compartidos cumplen con el estándar.

6.1.4.5. RESULTADO OBTENIDO

En Las siguientes figuras se pueden observar cómo se realizaron las diferentes pruebas para hacer una transferencia de archivos SFTP.

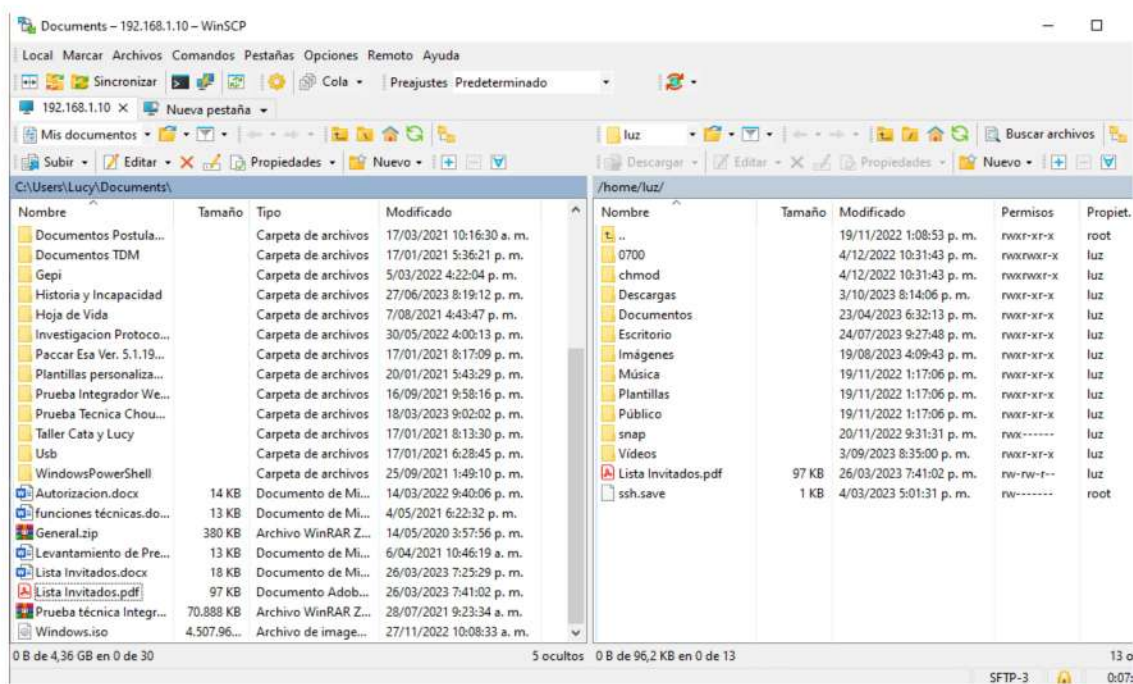


Ilustración 11. Transferencia de archivos – SFTP con WinSCP

La ilustración 11 muestra una conexión al servidor con IP 192.168.1.10 desde un gestor de archivos remotos (WinSCP), adicionalmente evidencia que se descarga un archivo, “Lista invitados.pdf” al cliente.

192.168.214.1	192.168.214.128	SSHv2	1514	Client: Encrypted packet (len=1460)
192.168.214.1	192.168.214.128	SSHv2	1514	Client: Encrypted packet (len=1460)
192.168.214.1	192.168.214.128	SSHv2	1514	Client: Encrypted packet (len=1460)
192.168.214.1	192.168.214.128	SSHv2	1514	Client: Encrypted packet (len=1460)
192.168.214.1	192.168.214.128	SSHv2	1514	Client: Encrypted packet (len=1460)
192.168.214.1	192.168.214.128	SSHv2	1514	Client: Encrypted packet (len=1460)

Ilustración 12. Captura monitoreo tráfico sftp con wireshark. Elaboración propia.

Detrás de las conexiones para gestionar los archivos remotos podremos ver un comportamiento semejante a la ilustración 12 la cual nos muestra que el servidor está compartiendo datos encriptados con el cliente. En este caso el cliente (IP 192.168.214.1) subió un archivo al servidor (IP 192.168.214.128), en los mensajes se aprecian varios paquetes encriptados porque el tamaño del archivo excedía el límite de paquete por lo que este es fragmentado.

6.1.5. CONFIGURACIÓN SOPORTE CAMBIO DE CLAVES

6.1.5.1. ENCRIPCIÓN CAMBIO DE CLAVES

El soporte de cambio de claves (key change support) en el contexto del protocolo SSH se refiere a la capacidad de la comunicación entre el cliente y el servidor para adaptarse a cambios en las claves de cifrado o autenticación durante la sesión. Esto es importante para mantener la seguridad de la conexión en caso de que las claves originales se vean comprometidas o necesiten ser actualizadas por cualquier motivo.

En una conexión SSH, las claves se utilizan para establecer la autenticación y cifrado. Si por alguna razón es necesario cambiar las claves durante la sesión, el soporte de cambio de claves asegura que la comunicación continúe de manera segura. Este proceso se realiza de manera transparente para el usuario, permitiendo la renovación de las claves sin interrupciones en la conexión.

En un escenario donde el servidor SSH cuenta con la configuración predeterminada, es decir, no se ha manipulado el archivo de configuración; el cambio de clave durante una sesión debe de ser encriptada o cifrada al enviar y recibir paquetes durante la sesión remota del cliente-servidor.

6.1.5.2. ESTANDAR SOBRE CAMBIO DE CLAVES

Acorde a la NIST y la ISO 27001 A.13.2.1 "Se debe contar con políticas, procedimientos y controles de transferencia formales para proteger la transferencia de información mediante el uso de todo tipo de instalaciones de comunicación". Donde las cuales se tienen en cuenta algunas directrices como:

- a) definir los procedimientos diseñados para proteger la información transferida contra interceptación, copiado, modificación, enrutado y destrucción;
- b) definir los procedimientos para la detección de software malicioso y protección contra éste, que puede ser transmitido mediante el uso de comunicaciones electrónicas;
- e) definir las responsabilidades del personal, las partes externas y cualquier otro usuario no comprometen a la organización, (por difamación, acoso, suplantación, envío de cadenas, compras no autorizadas, etc.);
- f). Establecer el uso de técnicas criptográficas, (proteger la confidencialidad, la integridad y la autenticidad de la información).
- h) definir los controles y restricciones asociadas con las instalaciones de comunicación, (el reenvío automático de correo electrónico a direcciones de correo externas);
- j) no dejar mensajes que contengan información confidencial, en las máquinas contestadoras, ya que éstos pueden ser escuchados por personas no autorizadas, almacenados en sistemas comunales o almacenados incorrectamente como resultado de una marcación incorrecta;

6.1.5.3. PRUEBA CAMBIO DE CLAVES

Ataque de Fuerza Bruta: Los ataques de fuerza bruta en las pruebas de penetración son simulaciones de ataques informáticos para evaluar la seguridad de un sistema en sus parámetros para autenticar. Al permitir que los usuarios cambien sus claves con frecuencia, se hace más difícil que los probadores de penetración obtengan acceso no autorizado a la cuenta del usuario.

6.1.5.4. RESULTADO ESPERADO DE CAMBIO DE CLAVES

Con este resultado se espera que los usuarios puedan cambiar la clave durante una sesión ssh, para que al atacante se les haga más difícil adivinar la contraseña y se evite comprometer la información.

6.1.5.5. RESULTADO OBTENIDO DE CAMBIO DE CLAVES

Para verificar si un servidor SSH Linux soporta cambios de clave, puede seguir los siguientes pasos:

Conectarse al servidor SSH Linux utilizando un cliente SSH como PuTTY o OpenSSH.

Una vez conectado al servidor, abra una sesión de terminal y escriba el siguiente comando:

- `ssh -v usuario@servidor`

Este comando le permitirá conectarse al servidor utilizando el usuario "usuario" y la dirección IP del servidor "servidor".

A medida que se establece la conexión, se mostrará información detallada en la sesión de terminal. Busque la siguiente línea en el registro de conexión:

- `debug1: Authentications that can continue: publickey, password`

Esta línea indica que se permiten dos tipos de autenticación: clave pública y contraseña.

Intente cambiar la clave de usuario en el servidor. Para hacer esto, escriba el siguiente comando en la sesión de terminal:

- password

Si se le solicita una contraseña actual, escríbala y luego escriba la nueva contraseña dos veces. Si el cambio de clave es exitoso, verá un mensaje que indica que la clave ha sido actualizada.

Intente conectarse al servidor nuevamente utilizando el mismo comando `ssh -v` que utilizó en el paso 2. Si la conexión es exitosa y se le permite ingresar con la nueva contraseña, entonces el servidor SSH Linux soporta cambios de clave.

```
luz@luz-VirtualBox: ~
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: algorithm: diffie-hellman-group-exchange-sha256
debug1: kex: host key algorithm: ssh-ed25519
debug1: kex: server->client cipher: aes128-ctr MAC: hmac-sha2-256 compression: none
debug1: kex: client->server cipher: aes128-ctr MAC: hmac-sha2-256 compression: none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST(2048<8192<8192) sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: SSH2_MSG_KEX_DH_GEX_GROUP received
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: SSH2_MSG_KEX_DH_GEX_REPLY received
debug1: Server host key: ssh-ed25519 SHA256:f0P1OaaBFVgIvp8MwNjmXGeDTst3NRvdiraXfTCiIRI
debug1: load_hostkeys: fopen /home/luz/.ssh/known_hosts2: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts: No such file or directory
debug1: load_hostkeys: fopen /etc/ssh/ssh_known_hosts2: No such file or directory
debug1: Host 'localhost' is known and matches the ED25519 host key.
debug1: Found key in /home/luz/.ssh/known_hosts:3
debug1: rekey out after 4294967296 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: rekey in after 4294967296 blocks
debug1: Will attempt key: /home/luz/.ssh/id_rsa RSA SHA256:SLM8x6bVA40NbqZpik0rcdCarY+uH+w+A3V4jDz51eU
debug1: Will attempt key: /home/luz/.ssh/id_ecdsa
debug1: Will attempt key: /home/luz/.ssh/id_ecdsa_sk
debug1: Will attempt key: /home/luz/.ssh/id_ed25519 ED25519 SHA256:aHuFR10XkRPHxjrjDb8JTTEOXoNN6fuirjeBUZnlFYgE
debug1: Will attempt key: /home/luz/.ssh/id_ed25519_sk
debug1: Will attempt key: /home/luz/.ssh/id_xmss
debug1: Will attempt key: /home/luz/.ssh/id_dsa
debug1: SSH2_MSG_EXT_INFO received
debug1: kex_input_ext_info: server-sig-algs=<ssh-ed25519,sk-ssh-ed25519@openssh.com,ssh-rsa,rsa-sha2-256,rsa-sha2-512,ssh-dsa-sha2-nistp521,sk-ecdsa-sha2-nistp256@openssh.com,webauthn-sk-ecdsa-sha2-nistp256@openssh.com>
debug1: kex_input_ext_info: publickey-hostbound@openssh.com=<0>
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Offering public key: /home/luz/.ssh/id_rsa RSA SHA256:SLM8x6bVA40NbqZpik0rcdCarY+uH+w+A3V4jDz51eU
debug1: Server accepts key: /home/luz/.ssh/id_rsa RSA SHA256:SLM8x6bVA40NbqZpik0rcdCarY+uH+w+A3V4jDz51eU
Authenticated to localhost ([127.0.0.1]:22) using "publickey".
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: pledge: filesystem
```

Ilustración 13. Muestra dos tipos de autenticación de clave pública y contraseña. Elaboración propia.

En la ilustración 13 se identifica como se realiza una autenticación de claves y también como se cambian las claves y así comprobar que el servidor SSH soporta el cambio de claves.

6.1.6. CONFIGURACIÓN SISTEMA AUTENTICACIÓN FUERTE

6.1.6.1. ENCRIPCIÓN FUERTE O DSA

De sus siglas en inglés Digital Signature Algorithm, es el algoritmo de firma digital utilizado en la autenticación de claves en sistemas SSH. Si el servidor brinda una llave pública luego de autenticar se puede decir que este soporta DSA, en caso contrario se puede decir que no está disponible.

6.1.6.2. ESTANDAR SOBRE ENCRIPCIÓN FUERTE O DSA

Los más comunes son el RSA (Rivest-Shamir-Adleman) y DSA (Digital Signature Algorithm), que se utiliza principalmente para firmas digitales en lugar de encriptación de datos. Algunas de las mejores prácticas de seguridad para garantizar una encriptación fuerte son:

- Longitud de clave: Uso de claves más largas para mayor seguridad. El estándar RSA propone 2048 bits o más, para DSA, se sugiere una longitud mínima de 2048 bits.
- Uso de algoritmos actualizados: Se deben evitar algoritmos obsoletos o vulnerables.

6.1.6.3. PRUEBA ENCRIPCIÓN FUERTE O DSA

En general, la autenticación fuerte o DSA en un servidor SSH ayuda a proteger contra una amplia gama de ataques, pentest o pruebas de intrusión. Sin embargo, es importante tener en cuenta que la seguridad de un sistema no depende solo de una sola medida de seguridad, sino de múltiples capas de protección y buenas prácticas de seguridad. Algunos de los ataques que se pueden considerar en esta sección:

- Ataques de suplantación de identidad: los ataques de suplantación de identidad implican que un atacante se haga pasar por un usuario legítimo. La autenticación fuerte o DSA ayuda a prevenir estos ataques mediante la verificación de la autenticidad de la clave pública del usuario.
- Ataques de man-in-the-middle: los ataques de man-in-the-middle implican que un atacante intercepte las comunicaciones entre el usuario y el servidor SSH para interceptar información confidencial. La autenticación fuerte o DSA puede prevenir estos ataques al verificar la autenticidad de las claves públicas y privadas del usuario y del servidor.

6.1.6.4. RESULTADO ESPERADO DE ENCRIPCIÓN FUERTE O DSA

En un ambiente SSH de configuración predeterminada, el atacante al encontrarse en medio de la conexión remota del cliente y servidor no podrá visualizar o leer el contenido de los paquetes porque estos se encuentran cifrados, dependiendo del algoritmo de cifrado acordado o disponible entre las máquinas podrá decirse que es fuerte.

6.1.6.5. RESULTADO OBTENIDO DE ENCRIPCIÓN FUERTE O DSA

Se realizó la prueba a través de un ataque del hombre del medio. Se supervisaron los paquetes intercambiados entre cliente y servidor SSH pero no se logró identificar con exactitud si la encriptación fue fuerte o si cumplía con DSA debido a que mostraba una lista de algoritmos de cifrado luego de establecer la autenticación (ver ilustración 14).

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda						
ip.addr == 192.168.214.1						
No.	Time	Source	Destination	Protocol	Length	Info
47	03:56:50.288380429	192.168.214.1	192.168.214.128	SSHv2	90	Client: Key Exchange Init
48	03:56:50.291115197	192.168.214.1	192.168.214.128	TCP	1514	[TCP Out-Of-Order] 55894 → 22 [ACK] Seq=22
49	03:56:50.291193966	192.168.214.1	192.168.214.128	TCP	90	[TCP Retransmission] 55894 → 22 [PSH, ACK]
50	03:56:50.319985286	192.168.214.128	192.168.214.1	SSHv2	1134	Server: Key Exchange Init
51	03:56:50.322643115	192.168.214.128	192.168.214.1	TCP	1134	[TCP Retransmission] 22 → 55894 [PSH, ACK]
52	03:56:50.333890936	192.168.214.1	192.168.214.128	SSHv2	1262	Client: Diffie-Hellman Key Exchange Init
53	03:56:50.335064847	192.168.214.1	192.168.214.128	TCP	1262	[TCP Retransmission] 55894 → 22 [PSH, ACK]
54	03:56:50.343775097	192.168.214.128	192.168.214.1	SSHv2	1514	Server: Diffie-Hellman Key Exchange Reply,
55	03:56:50.343775137	192.168.214.128	192.168.214.1	SSHv2	178	Server: Encrypted packet (len=124)
56	03:56:50.351002623	192.168.214.128	192.168.214.1	TCP	1514	[TCP Out-Of-Order] 22 → 55894 [ACK] Seq=1
57	03:56:50.351097232	192.168.214.128	192.168.214.1	TCP	178	[TCP Retransmission] 22 → 55894 [PSH, ACK]
58	03:56:50.351262710	192.168.214.1	192.168.214.128	TCP	60	55894 → 22 [ACK] Seq=2733 Ack=2706 Win=13:
59	03:56:50.358700479	192.168.214.1	192.168.214.128	SSHv2	134	Client: New Keys, Encrypted packet (len=64)
60	03:56:50.358762288	192.168.214.1	192.168.214.128	TCP	54	55894 → 22 [ACK] Seq=2733 Ack=2706 Win=13:
61	03:56:50.366638043	192.168.214.1	192.168.214.128	TCP	134	[TCP Retransmission] 55894 → 22 [PSH, ACK]
62	03:56:50.367059349	192.168.214.128	192.168.214.1	SSHv2	118	Server: Encrypted packet (len=64)
63	03:56:50.379010826	192.168.214.128	192.168.214.1	TCP	118	[TCP Retransmission] 22 → 55894 [PSH, ACK]
64	03:56:50.380925553	192.168.214.1	192.168.214.128	SSHv2	134	Client: Encrypted packet (len=80)
65	03:56:50.390777165	192.168.214.1	192.168.214.128	TCP	134	[TCP Retransmission] 55894 → 22 [PSH, ACK]
Checksum: 0x7e66 [unverified] [Checksum Status: Unverified] Urgent Pointer: 0 [SEQ/ACK analysis] [Timestamps] TCP payload (80 bytes) SSH Protocol SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none) Packet Length: 12 Padding Length: 10 Key Exchange (method:sntrup761x25519-sha512@openssh.com) Message Code: New Keys (21) Padding String: 80addacd3712b3cb4c3b SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none) Packet Length (encrypted): 3ccbbaed Encrypted Packet: 88182850af28b2057008cd80fb29de6ae83c5a7595c8707def0d2f62 MAC: af4dfa8cffe0e596f93aab4bc3dab755a91c35f1d961fb6d0f4351ffd493668f [Direction: client-to-server]						
0000 00 0c 29 f7 4f 34 00 50 56 c0 00 08 08 00 45 00 ... 04:P						
wireshark eth0QYFVE2.pcapng Paquetes: 2684 · Mostrado: 1971 (73.4%) · Perdido: 0 (0.)						

Ilustración 14. Captura entrega paquetes con llaves. Elaboración propia.

6.1.7. CONFIGURACIÓN DEL ATAQUE DE HOMBRE EN EL MEDIO

6.1.7.1. ENCRIPCIÓN DEL HOMBRE EN EL MEDIO

La encriptación para este ataque es tener tres máquinas virtuales un servidor SSH, Cliente Windows y Kali Linux con el objetivo de poder identificar con mayor facilidad un espionaje en el que el atacante se coloca en medio del cliente y el servidor.

6.1.7.2. ESTANDAR SOBRE HOMBRE EN EL MEDIO

Llevando a cabo las pruebas del hombre en el medio se recomienda tener diferentes medidas de protección y algunas acciones para minimizar el riesgo son la Seguridad de la red, Cifrado y VPN, el uso de contraseñas robustas y evitar difundir información personal.

6.1.7.3. PRUEBA SOBRE HOMBRE EN EL MEDIO

En esta prueba se tuvieron en cuenta varios aspectos y fue la configuración de las tres máquinas cada una debe de tener un usuario y contraseña también hay unos programas los cuales ya vienen por defecto instalados en la tercera maquina y ellos son Wireshark y Ettercap los dos nos van a ayudar bastante porque uno es para mirar el tráfico de red y el otro va a realizar un sniffing donde va a detectar las IP de las respectivas máquinas para luego ejecutar la prueba.

6.1.7.4. RESULTADO ESPERADO DE HOMBRE EN EL MEDIO

Con esta prueba se puede validar una conexión de tres máquinas 2 de ellas son las víctimas y la 3 es el atacante donde se comprueba con las IP de las víctimas, después se observa que el atacante está dentro del mismo host para así poder cambiar el enrutamiento de la información y robarla sin que la víctima se dé cuenta de lo que está sucediendo.

6.1.7.5. RESULTADO OBTENIDO DE HOMBRE EN EL MEDIO

A continuación, se pueden ver las diferentes figuras que se obtuvieron mediante las pruebas realizadas del "Man in the middle".

```
luz@luz-VirtualBox: ~  
debug1: /etc/ssh/ssh_config line 19: include /etc/ssh/ssh_config.d/*.conf matched no files  
debug1: /etc/ssh/ssh_config line 21: Applying options for *  
ssh: Could not resolve hostname 192.168.1.10-ssh: Name or service not known  
luz@luz-VirtualBox:~$ which scp  
/usr/bin/scp  
luz@luz-VirtualBox:~$ sudo nano /etc/ssh/sshd_config  
[sudo] contraseña para luz:  
luz@luz-VirtualBox:~$ ssh-keygen -f /etc/ssh/ssh_host_rsa_key.pub
```

Ilustración 15. Muestra si la clave publica del Servidor SSH es auténtica. Elaboración propia.

En la ilustración 15 se puede observar que en la línea 8 nos encontramos con un comando ssh-keygen -f el cual nos indica que la clave se encuentra activa y autenticada en el Servidor SSH.

```
pc. Símbolo del sistema  
Adaptador de LAN inalámbrica Wi-Fi:  
  
Sufijo DNS específico para la conexión. . . :  
Dirección IPv6 . . . . . : 2800:e2:8180:2e0:8383:1e75:c501:7163  
Vínculo: dirección IPv6 local. . . : fe80::e1ee:56e6:f68a:606c%16  
Dirección IPv4. . . . . : 192.168.1.5  
Máscara de subred . . . . . : 255.255.255.0  
Puerta de enlace predeterminada . . . . : fe80::2e00:abff:fe6e:e90f%16  
192.168.1.254  
  
Adaptador de Ethernet Conexión de red Bluetooth:  
  
Estado de los medios. . . . . : medios desconectados  
Sufijo DNS específico para la conexión. . . :  
  
C:\Users\Lucy>ping 192.168.1.10  
  
Haciendo ping a 192.168.1.10 con 32 bytes de datos:  
Respuesta desde 192.168.1.10: bytes=32 tiempo<1m TTL=64  
Respuesta desde 192.168.1.10: bytes=32 tiempo<1m TTL=64  
Respuesta desde 192.168.1.10: bytes=32 tiempo<1m TTL=64  
Respuesta desde 192.168.1.10: bytes=32 tiempo<1m TTL=64  
  
Estadísticas de ping para 192.168.1.10:  
Paquetes: enviados = 4, recibidos = 4, perdidos = 0  
(0% perdidos),  
Tiempos aproximados de ida y vuelta en milisegundos:  
Mínimo = 0ms, Máximo = 0ms, Media = 0ms  
  
C:\Users\Lucy>
```

Ilustración 16. Ping-cliente-servidor. Elaboración propia.

En la ilustración 16 se realiza un ping desde el cliente al servidor esto se hace con motivo de mirar que haya conexión con el cliente y servidor.

```
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
luz@luz-VirtualBox:~$ ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=1 ttl=128 time=11.7 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=128 time=9.57 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=128 time=15.5 ms
64 bytes from 192.168.1.5: icmp_seq=4 ttl=128 time=14.0 ms
64 bytes from 192.168.1.5: icmp_seq=5 ttl=128 time=13.2 ms
^C
--- 192.168.1.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 9.568/12.797/15.510/2.032 ms
```

Ilustración 17. Ping-cliente-ubuntu. Elaboración propia.

En la ilustración 17 se realiza un ping desde el servidor al cliente esto con el objetivo de ver que se realice una correcta conexión cliente – servidor.

```
File  Actions  Edit  View  Help
(lucy@kali)-[~]
$ ping 192.168.1.5
PING 192.168.1.5 (192.168.1.5) 56(84) bytes of data.
64 bytes from 192.168.1.5: icmp_seq=1 ttl=128 time=0.775 ms
64 bytes from 192.168.1.5: icmp_seq=2 ttl=128 time=0.921 ms
64 bytes from 192.168.1.5: icmp_seq=3 ttl=128 time=1.11 ms
^C
— 192.168.1.5 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 0.775/0.934/1.108/0.136 ms

(lucy@kali)-[~]
$ ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=0.994 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.969 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=0.789 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=64 time=1.39 ms
64 bytes from 192.168.1.10: icmp_seq=5 ttl=64 time=1.41 ms
^C
— 192.168.1.10 ping statistics —
5 packets transmitted, 5 received, 0% packet loss, time 4009ms
rtt min/avg/max/mdev = 0.789/1.111/1.409/0.247 ms

(lucy@kali)-[~]
$
```

Ilustración 18. Ping-cliente-servidor-kali Linux. Elaboración propia.

En la ilustración 18 se observa que en la tercera maquina también hay que realizar un ping al cliente y al servidor para verificar la conexión cliente – servidor.

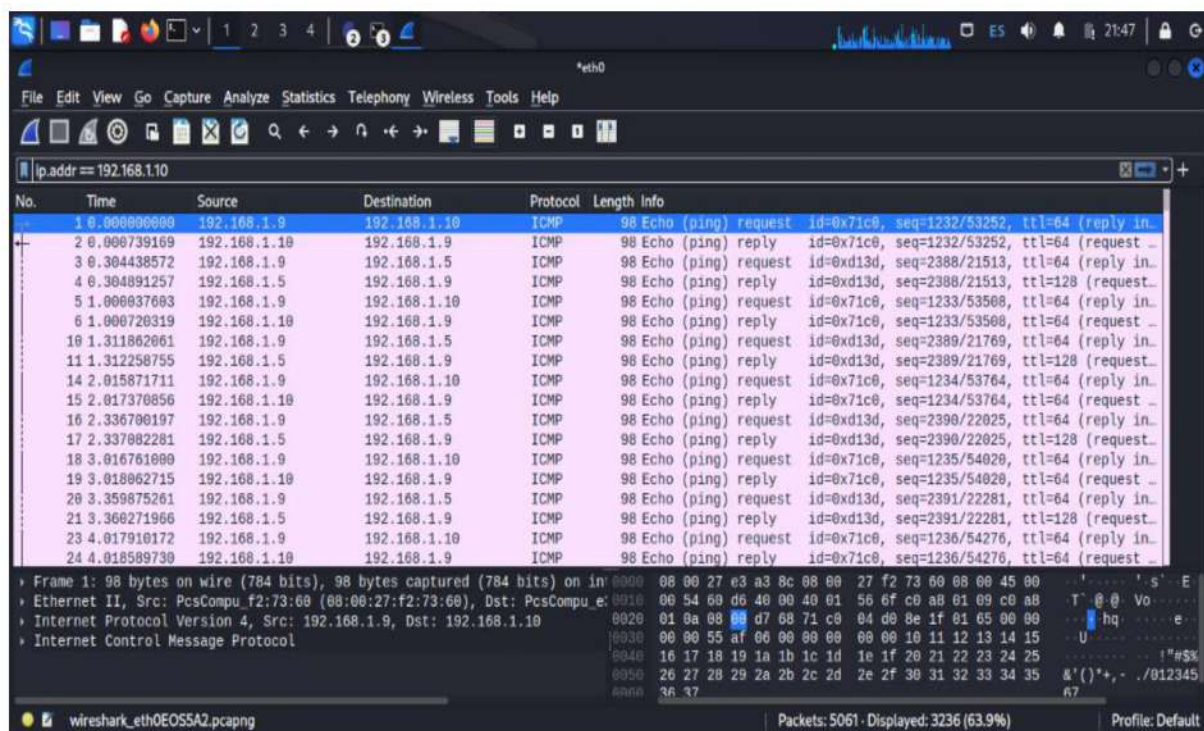


Ilustración 19. Muestra Man in the middle-kali-Linux. Elaboración propia.

En la ilustración 19 se observa que desde el programa Wireshark se realiza un tráfico de red y se puede mirar el ping de las IP del cliente – servidor.

No.	Time	Source	Destination	Protocol	Length	Info
2644	403.747352827	192.168.1.5	192.168.1.10	SSHv2	82	Client: Protocol (SSH-2.0-PuTTY_Release_0.76)
2651	403.813926895	192.168.1.10	192.168.1.5	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_8.9p1 Ubuntu-3ubun
2653	403.830062700	192.168.1.5	192.168.1.10	SSHv2	1310	Client: Key Exchange Init
2655	403.836855181	192.168.1.10	192.168.1.5	SSHv2	350	Server: Key Exchange Init
2657	403.845319598	192.168.1.5	192.168.1.10	SSHv2	78	Client: Diffie-Hellman Group Exchange Request
2659	403.865661925	192.168.1.10	192.168.1.5	SSHv2	590	Server: Diffie-Hellman Group Exchange Group
2661	403.898187629	192.168.1.5	192.168.1.10	SSHv2	582	Client: Diffie-Hellman Group Exchange Init
2663	403.933158833	192.168.1.10	192.168.1.5	SSHv2	1514	Server: Diffie-Hellman Group Exchange Reply, New Key
2664	403.933175003	192.168.1.10	192.168.1.5	SSHv2	298	Server: Encrypted packet (len=244)
2669	403.970889808	192.168.1.5	192.168.1.10	SSHv2	134	Client: New Keys, Encrypted packet (len=64)
2671	403.973464762	192.168.1.10	192.168.1.5	SSHv2	118	Server: Encrypted packet (len=64)
2695	407.400974833	192.168.1.5	192.168.1.10	SSHv2	134	Client: Encrypted packet (len=80)
2697	407.412837710	192.168.1.10	192.168.1.5	SSHv2	134	Server: Encrypted packet (len=80)
2760	416.024936935	192.168.1.5	192.168.1.10	SSHv2	326	Client: Encrypted packet (len=272)
2765	416.146602432	192.168.1.10	192.168.1.5	SSHv2	102	Server: Encrypted packet (len=48)
2767	416.149640534	192.168.1.5	192.168.1.10	SSHv2	134	Client: Encrypted packet (len=80)
2775	416.486984776	192.168.1.10	192.168.1.5	SSHv2	550	Server: Encrypted packet (len=496)
2779	416.537265155	192.168.1.10	192.168.1.5	SSHv2	118	Server: Encrypted packet (len=64)

▶ Frame 2644: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on
 ▶ Ethernet II, Src: Chongqin_45:e8:9d (5c:3a:45:45:e8:9d), Dst: PcsCompu_f:
 ▶ Internet Protocol Version 4, Src: 192.168.1.5, Dst: 192.168.1.10
 ▶ Transmission Control Protocol, Src Port: 55392, Dst Port: 22, Seq: 1, Acl
 ▶ SSH Protocol

Ilustración 20. Conexión-ssh-wireshark-kali-Linux. Elaboración propia.

En la ilustración 20 se muestra que desde la tercera maquina kali-linux se obtuvo una conexión exitosa con las dos máquinas las cuales son el cliente – servidor y se observa que se está efectuando un ataque de "Man in the middle".

6.2. ESCENARIO 2 - CONFIGURACIÓN ALTO NIVEL NIST

Para una prueba de peritaje en un escenario SSH donde se utiliza la configuración más completa del marco NIST debe tener la siguiente información:

SERVIDOR

- Conocer la IP
- Puerto utilizado (predeterminado el 22)
- Servicios abiertos (SFTP para Web, Correo / Bases datos)
- Cantidad intentos autenticación
- Tiempo espera al agotar intentos autenticación
- Jailfolder
- Restricción cifrado y algoritmos a Ciphers aes256-ctr, aes192-ctr, aes128-ctr
- Restricción de MAC e IP para cuenta Root
- Cuentas para roles administrativos deshabilitadas

CLIENTE

- Conocer la IP
- Conocer nombre usuario
- Conocer contraseña

Se plantea para un futuro llevar a cabo esta configuración buscando conocer que tanto cumple con las prácticas internacionales estandarizadas mediante pruebas de penetración en ambientes controlados virtualizados como aquellos que se realizaron en el escenario 1.

7. PREGUNTAS FRECUENTES

¿Qué es necesario para poder llevar a cabo un laboratorio semejante?

R: Se requieren de varios elementos que se detallan en la tabla siguiente:

Insumo	Detalles
Software base	Sistemas Operativos • Servidor: Ubuntu 20.4 • Cliente: Windows 10 • Sistema auditor: Kali Linux Canal SSH • OpenSSH • Putty Virtualización • Virtualbox • Visual Studio Code
Hardware	Procesador: Intel Core i5 2.5 Ghz Arquitectura: x64 RAM: 8 GB HDD: 500 GB
Datos	Copias de seguridad en tiempo real utilizando servidor en nube: • Microsoft OneDrive
Referentes	OVAL NIST RFC OpenSSH ISO

Tabla 2 para realizar PenTest o Pruebas de intrusión. Elaboración propia.

¿Se puede usar una máquina de ubuntu desktop y luego volverla servidor?

R: Si, se puede utilizar una versión Desktop e instalar el paquete de OpenSSH. Debe de registrar según configuración predeterminada y recomendaciones internacionales.

¿Desde cuál perspectiva de atacante, cliente o servidor se mira la práctica de laboratorio?

R: Desde el atacante.

¿Por qué se utiliza la clave encriptada entre el cliente y el servidor?

R: La clave se utiliza para poder desencriptar el mensaje que ha enviado. En el "handshake" se utiliza la clave para establecer el canal de comunicación entre cliente y servidor con algoritmos de cifrado comunes.

¿Cuáles herramientas se recomienda para analizar el tráfico y paquetes enviados?

R: Siendo el mejor desde el siguiente orden: Network Minner, WireShark, TcpDump y Mojo Packets.

¿Qué sucede que al realizar una conexión desde la terminal por medio de plink de putty muestra set de caracteres indeseados reemplazando teclas como cntrl, backsp por secuencias semejantes a <-[01:34?

R: Esto se debe a que falta un intérprete de **caracteres de escape** en el sistema operativo. Se recomienda instalar Ansicon.

¿Cómo puedo cambiar la configuración del servidor SSH?

R: Con la dirección puedo llamar el archivo 'sshd_config' que hay dentro de la carpeta '/etc/ssh/' en la máquina del servidor con OpenSSH. Para poder editarlo debe de contar con los permisos de edición.

¿Cómo ver los usuarios conectados y comando para desconectarlos remotamente?

R: Para ver los usuarios conectados se puede utilizar el comando 'who'.

Para desconectar a los usuarios se puede utilizar el comando 'sudo pkill -KILL -u usuario'.

8. INFORMACIÓN DE LOS AUTORES

LUZ EUGENIA MUÑOZ LONDOÑO

AFILIACIÓN E INSTITUCIÓN: Ingeniera de Sistemas IUSH, Equipo Investigación GEPI

CORREO: luz.munozl@comunidad.iush.edu.co

JULIÁN ANDRÉS PEÑA RÚA

AFILIACIÓN E INSTITUCIÓN: Ingeniero de Sistemas IUSH, Equipo Investigación GEPI

CORREO: julian.penar@comunidad.iush.edu.co

MARIA EUGENIA GONZÁLEZ PÉREZ

AFILIACIÓN E INSTITUCIÓN: Magister Ingeniería del Software UdeM, Docente de la IUSH

CORREO: maria.gonzalezp@salazaryherrera.edu.co

HÉCTOR FERNANDO VARGAS MONTOYA

AFILIACIÓN E INSTITUCIÓN: Magíster en Seguridad Informática, Docente del ITM

CORREO: hectorvargas@itm.edu.co