ISSN: 1646-9895

risti

Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

M a i o 2 0 • M a y 2 0



©AISTI 2020 http://www.aisti.eu

Nº E29

risti
Revista Ibérica de Sistemas e Tecnologias de Informação
Iberian Journal of Information Systems and Technologies

Recebido/Submission: 27/11/2019
Aceitação/Acceptance: 25/01/2020

Guía práctica para el análisis de vulnerabilidades de un entorno cliente-servidor GNU/Linux mediante una metodología de pentesting

Jesús Andrés Ovallos-Ovallos[1], Dewar Rico-Bautista[2], Yurley Medina-Cárdenas[2]

j.ovallos@unisimonbolivar.edu.co, dwricob@ufpso.edu.co, ycmedinac@ufpso.edu.co

[1] Universidad Simón Bolivar, Av 3 #13-34, Br La Playa, 540006, Cúcuta, Colombia.

[2] Universidad Francisco de Paula Santander Ocaña, Sede Algodonal Vía Acolsure, 546551, Ocaña, Colombia.

Pages: 335–350

Resumen: La seguridad de la información se ha convertido en uno de los temas centrales de toda organización en especial si esta se transmite a través de la red. Tomando en cuenta que el principal activo de estas organizaciones es la información, se hace necesario no escatimar esfuerzos para blindar a dicho elemento de eventos que busquen crear daño. En la red se encuentra todo tipo de amenazas que buscan vulnerar cualquier sistema. Generalmente son ataques dirigidos por personas cuyo objetivo es crear caos y obtener beneficios económicos. Diferentes procesos se utilizan hoy en día para mejorar el trato que se le da a la información y su seguridad, desde aplicación de estándares hasta la certificación de estos procesos. Este artículo contiene un análisis de vulnerabilidad a un entorno Cliente-Servidor con el fin de encontrar fallos de seguridad y gestionar los resultados en pro de la seguridad.

Palabras-clave: Evidencia digital; Análisis forense; seguridad informática; metodologías; delito informático.

A practical guide to analyzing vulnerabilities in a GNU/Linux client-server environment using a pentesting methodology

Abstract: Information security has become one of the central themes of any organization, especially if it is transmitted through the network. Taking into account that the main asset of these organizations is information, it is necessary to spare no effort to shield this element from events that seek to create damage. In the network there are all kinds of threats that seek to harm any system. They are generally attacks directed by people whose objective is to create chaos and obtain economic benefits. Different processes are used today to improve the treatment of information and its security, from the application of standards to the certification of these processes. This article contains a vulnerability analysis to a Client - Server environment in order to find security flaws and manage the results in favor of security.

Keywords: Digital evidence; Forensic analysis; computer security; methodologies; cybercrime.

1. Introducción

La seguridad de la información se ha convertido en uno de los temas centrales de toda organización. La información que viaja a través de las redes de comunicación diariamente se está viendo sujeta a procedimientos ilegales que buscan vulnerarla de alguna manera (Carter et al., 2014). Estas acciones son realizadas por expertos llamados intrusos informáticos a través de potentes herramientas de penetración de sistemas. Todos estos factores son riesgos inminentes para organizaciones e instituciones (Lyubimov, 2010). Teniendo en cuenta que el principal activo de estas organizaciones es la información, se hace necesario no escatimar esfuerzos para blindar a dicho elemento de eventos que busquen crear daño en él diferentes procesos se utilizan hoy en día para mejorar el trato que se le da a la información y su seguridad, desde aplicación de estándares hasta la certificación de estos procesos (Y. Medina-Cárdenas et al., 2016), (Y. C. Medina-Cárdenas & Rico-Bautista, 2016).

La mejor forma de estar preparados para posibles ataques es analizar la infraestructura de la empresa con pruebas de seguridad. Delimitando la temática hacia un entorno más técnico haciendo referencia en sí a la configuración de equipos, podemos a través de la realización análisis de vulnerabilidad periódicos identificar los puntos rojos en materia de seguridad. De ahí la importancia de que instituciones y organizaciones implementen este tipo de controles con el fin de garantizar la confidencialidad, integridad y disponibilidad de su información. La seguridad de la información es quizás el factor más importante a tomar en cuenta a la hora de implementar una red de datos o un sistema (Boynton, 2007). La información representa el eje central de funcionamiento tanto de instituciones como de organizaciones, y los respectivos análisis permiten a dichas entidades entender el comportamiento de sus compañías. A través de pruebas de laboratorio en ambiente controlado y con el apoyo de metodologías de pentesting (Lobo-Parra et al., 2019), (Rueda-Rueda et al., 2019); como lo son el NIST SP 800-55 y el OSSTMM v.3, podemos realizar un análisis de vulnerabilidad sin comprometer ningún sistema real (Bonetti et al., 2013).

El resto del artículo está dividido cuatro secciones. En la sección dos se presenta la revisión de la literatura. En la sección tres se presenta la metodología aplicada. En la sección 4 se presentan los resultados y al final las conclusiones y el trabajo futuro

2. Revisión de la literatura

La seguridad de las TI es un tema frecuente al que se enfrentan las empresas en estos últimos meses. Valdez realiza el trabajo Auditorias de Seguridad Informática y la OSSTMM3 (Valdez, 2013), (Rico-Bautista et al., 2016). Se ha hablado de fallos de seguridad incluso en medios de comunicación no especializados, por ejemplo del fallo en el protocolo DNS (que tanta repercusión ha tenido) y el del no tan conocido BGP, lo cual hace ver, la importancia que está tomando la seguridad en la sociedad actual, y sobre todo en la de las empresas. Situándonos en este escenario, el proyecto que voy a realizar se centraría en auditorías de seguridad informática (en especial, de vulnerabilidades y tests de intrusión) (Alvernia Acevedo & Rico-Bautista, 2017), tomando como referencia, la metodología descrita por ISECOM: la Open Source Security Testing Methodology Manual (OSSTMM a partir de ahora). Teniendo en cuenta, que ésta es, quizás la metodología

más extendida en el campo, siendo además abierta, me ha resultado interesante tomarla como referencia. La primera pregunta que debemos responder antes de adentrarnos en el proyecto, es: ¿Que es una auditoría de seguridad? ¿Qué es un test de intrusión? Una auditoría de seguridad, es el análisis y estudio de un sistema, para intentar conocer las limitaciones de un sistema (vulnerabilidades, debilidades, preocupaciones etc. según OSSTMM), para posteriormente poder corregirlas. Un test de intrusión, se considera una como un tipo de auditoría de seguridad, donde las vulnerabilidades que se buscan, son aquellas que permitirían el acceso al sistema de alguien no autorizado, para comprobar la resistencia que ofrece el sistema ante este tipo de ataques.

Veloz et al. (2017) plantean un enfoque metodológico para profesionales sobre Ethical Hacking, partiendo desde el valor de la estrategia refiriéndose al valor de la seguridad de la empresa, pasando por el tipo de análisis de seguridad a emplear y los tipos de reconocimiento activo y pasivo, hasta concluir con la presentación de análisis de resultados. Internet es un logro tecnológico muy importante y es utilizado en una infinidad de áreas y propósitos. Cada organización que se conecta a Internet asume un riesgo ya que no siempre los propósitos de los usuarios de Internet son buenos. Si bien al conectarse a Internet las organizaciones pueden utilizar tecnologías como el correo electrónico, la web, intranet entre otros, también ponen en riesgo su información, sus recursos y su reputación. Es por ese motivo que las organizaciones deben estar preparadas con tecnologías que permitan controlar el acceso a sus sistemas sin perjudicar el quehacer de la organización (Y. Medina-Cárdenas et al., 2019). Además de las tecnologías, se debe tener un marco de trabajo en donde se delinean las actividades que son permitidas de acuerdo al quehacer de la organización y para ello se debe establecer políticas que regulen el acceso a los servicios de red, la cuales pueden ser internas o externas (Rico-Bautista et al., 2019).

Ramos (2014) presenta las Técnicas comunes de Ataque a equipos con sistema operativo UNIX o derivados El presente trabajo trata de brindar un panorama acerca de las técnicas más comunes de obtención de información remota en forma activa y pasiva, detección de vulnerabilidades y posteriormente una introducción al ataque en entornos tipo UNIX, como Linux (Debian, Ubuntu, Red Hat, Fedora, Suse, Mandriva, etc.), BSD (FreeBSD, OpenBSD, NetBSD), Solaris, AIX, HP-UX, SCO Unix, etc.). Picouto, Lorente, García y Ramos, exponen en su libro Hacking y Seguridad en Internet, un conjunto de pasos y formas para comprometer un sistema a través de metodologías empleando herramientas de seguridad, comenzando en su obra por destacar conceptos básicos con los relacionado al protocolo TCP/IP. De la Fraga (2003) presenta la experiencia de trabajo en la seguridad de la red de la Sección de Computación del CINVESTAV. Se describirán dos soluciones para la construcción de redes seguras: una por división de redes usando puertas (gateways), cortafuegos (firewalls), lo que generan esquemas de redes militarizadas y redes desmilitarizadas; y otra por monitoreo de paquetes de la red. Los esquemas aquí presentados permitirían formalizar un esquema de pago por uso de red a usuarios con máquinas propias (laptops).

En el contexto nacional, se encuentra muy poca información científica acerca del tema, sin embargo, la página de la ACIS, encontramos varios artículos, relacionados con el tema. López et. al (2001) se presenta la seguridad informática y se muestra su terminología técnica, encontrándonos en este, propiedades de un sistema, que activos

se deben proteger, tipos de ataques, modelos de seguridad informática, entre otras cosas que complementan el estudio de seguridad en un sistemas. Hablar hoy en día de la palabra vulnerabilidad, es sinónimo de alerta para un administrador de red o de seguridad, incluso para programadores y desarrolladores. Los profesionales en el campo informático buscan reducir la brecha de inseguridad que pueda haber en sus plataformas, con el fin de impedir un ataque informático que pueda comprometer sus sistemas (Tian et al., 2014). El artículo "¿Qué son las vulnerabilidades de software?"(Hernandez, 2014), da una visión acerca del tema, donde menciona el concepto, impacto, vector y como gestionar dichas vulnerabilidades para mitigar los riesgos. En octubre de 2012 se conoció la noticia donde a varios usuarios del cliente de correos de Apple icloud, les fue hurtada su información y borrada de manera remota los datos almacenados en sus dispositivos Apple (ipad, iphone y MacBook Pro), el hecho se produjo al ingresar a través de ingeniería social a la cuenta de Amazon de los afectados y prosiguiendo a la cuenta de icloud. Los servicios que se ofrecen en internet se encuentran estructurados bajo el modelo Cliente-Servidor, este se basa en peticiones realizadas por el cliente hacia el servidor sobre algún servicio en especial (SMTP, FTP, WEB, MAIL) entre otros, realizada la petición el servidor da una respuesta al cliente, completando de esta forma la solicitud. Según la Radio Nacional de Colombia en el mes de abril de 2011 los diferentes portales gubernamentales sufrieron diversos ataques informáticos. "En tan solo 18 días tenemos registrados 28 ataques exitosos que lograron comprometer la seguridad de estos sitios y solamente en la mañana de este 18 de abril ya fueron hackeados dos sitios web gubernamentales de Colombia", informó el ingeniero experto en seguridad informática de la compañía de seguridad Kaspersky, Dmitri Beztushev.

De acuerdo a estudios realizados por el Ponemon Institute y auspiciado por HP (Ponemon Institute, 2015), los ataques cibernéticos pueden ser costosos si no se resuelven con rapidez. El tiempo promedio para la resolución de un ataque cibernético es de 24 días, pero puede tardar hasta 50 días de acuerdo con el estudio de este año. El costo promedio incurrido durante este período de 24 días fue de US$ 591.780, lo que representa un aumento de 42 por ciento con relación al costo promedio estimado del último año, de US$ 415.748, durante un período de resolución promedio de 18 días. El análisis de vulnerabilidades, hace parte del contexto de la seguridad informática, la cual tiene cada día más adeptos. Es importante que las organizaciones tengan políticas serias de seguridad de la información basada en Sistemas de Gestión de la Seguridad de la Información, como la ISO 27000, estándar de referencia mundial para las buenas prácticas en el manejo de la información y su respectiva seguridad. Las herramientas hacking de código abierto que existen en la actualidad, permiten que personas sin un amplio conocimiento en informática puedan comprometer un sistema, estas personas son conocidas como SCRIPT KIDDIE, y son causantes de gran parte de los incidentes de seguridad del mundo. Pero no solo existen herramientas de código abierto con el fin de causar daño, en la actualidad existe una distribución GNU/Linux llamada BackTrack, es una tools de auditoria para evaluar la seguridad de un sistema, este sistema operativo esta basado en su similiar Ubuntu 10.04 y es una de las opciones preferidas por los profesionales de la seguridad a la hora de realizar sus labores de auditoría.

3. Metodología aplicada

En el modelo cliente-servidor, el dispositivo que solicita información se denomina cliente y el dispositivo que responde a la solicitud se denomina servidor. Los procesos de cliente y servidor se consideran una parte de la capa de Aplicación, ver figura 1. El cliente comienza el intercambio solicitando los datos al servidor, que responde enviando uno o más streams de datos al cliente. Los protocolos de capa de Aplicación describen el formato de las solicitudes y respuestas entre clientes y servidores. Además de la transferencia real de datos, este intercambio puede requerir de información adicional, como la autenticación del usuario y la identificación de un archivo de datos a transferir. Un ejemplo de una red cliente/servidor es un entorno corporativo donde los empleados utilizan un servidor de e-mail de la empresa para enviar, recibir y almacenar e-mails. El cliente de correo electrónico en la computadora de un empleado emite una solicitud al servidor de e-mail para un mensaje no leído. El servidor responde enviando el e-mail solicitado al cliente.

Aunque los datos generalmente se describen como un flujo del servidor al cliente, algunos datos siempre fluyen del cliente al servidor. El flujo de datos puede ser el mismo en ambas direcciones o inclusive ser mayor en la dirección que va del cliente al servidor. Por ejemplo, un cliente puede transferir un archivo al servidor con fines de almacenamiento. La transferencia de datos de un cliente a un servidor se conoce como subida y la de los datos de un servidor a un cliente, descarga (Andrew, 2007). Las características básicas de una arquitectura Cliente-Servidor son: Combinación de un cliente que interactúa con el usuario, y un servidor que interactúa con los recursos compartidos. El proceso del cliente proporciona la interfaz entre el usuario y el resto del sistema. El proceso del servidor actúa como un motor de software que maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.

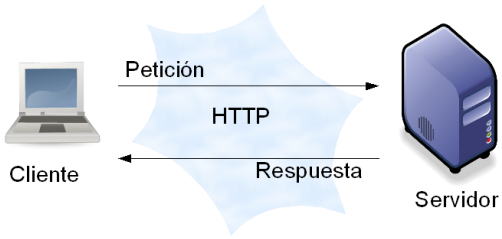


Figura 1 – Modelo Cliente/Servidor. Fuente: Adaptado de (Marini, 2012)

OSSTMM es un estándar profesional para el testeo de seguridad en cualquier entorno, desde el exterior como el interior. Incluye lineamientos de acción, la ética del tester profesional, legislación sobre testeo de seguridad y un conjunto integral de test. Lo más importante en esta metodología es que los diferentes tests son evaluados y ejecutados donde sean aplicables, hasta arribar a los resultados esperados dentro de un período

de tiempo determinado. Solo así el testeador habrá ejecutado el test en conformidad conel modelo OSSTMM, y por ello, el informe podrá ser considerado mínimamente exhaustivo. Esta metodología propone varios tipos de análisis de seguridad, para efectos del proyecto se asemeja a un tipo de análisis llamado, Vulnerability Assessment.

Vulnerability Assesment (Evaluación de Seguridad). Es un tipo de test de seguridad, este test es ideal cuando una organización no posee los recursos para realizar un análisis de seguridad exhaustivo, y además cuanto se urge de resultados con respecto al tiempo. Es un análisis y pruebas relacionadas con la identificación de puertos abiertos, servicios disponibles y vulnerabilidades conocidas en los sistemas de información objetivos; las mismas incluyen, a su vez, verificaciones manuales y automáticas de falsos positivos, lo que permite la identificación de los puntos débiles de la red y análisis profesional individualizado (Y. Medina-Cárdenas & Rico-Bautista, 2012). Es un tipo de análisis que busca identificar e informar fallas en los dispositivos y en los procesos tecnológicos. Las etapas son : Etapa de reconocimiento pasivo; Etapa de reconocimiento activo superficial; Etapa de reconocimiento activo en profundidad; Etapa de análisis de vulnerabilidades y Etapa de reporte.

4. Resultados y análisis

El desarrollo del análisis de vulnerabilidad se efectuara a través de la realización de una guía práctica de laboratorio, donde desde la óptica de un pentester se realizan maniobras de vulneración de sistemas con el fin de identificar puntos críticos o fallas en el mismo que puedan comprometer la integridad del entorno o sistemas que se encuentra en evaluación. Los software usados en los laboratorios se observan en la tablas 1.

Laboratorio 1	**Laboratorio 2**	**Laboratorio 3**
Linux Back Track 5	Ubuntu Server 12.04	Ubuntu Server 12.04
Nmap	Nmap	Nmap
Servidor BIND9	Linux Back Track 5	Linux Back Track 5
Ubuntu Server	Servidor VSFTPD	Servidor LAMP (Linux, Apache, Mysql, PHP)
Fierce	Hydra	Sqlmap
Webmin 1.620	Webmin 1.620	Damn Vulnerable Web Application 1.0.7

Tabla 1 – Software usado en los diferentes laboratorios

Laboratorio 1. Implementación de un ataque de transferencia de zona DNS entre sistemas GNU/Linux arquitectura cliente servidor.

La práctica de laboratorio se realizó en una red de área local mediante el protocolo de comunicaciones Internet Protocol 4 bajo la arquitectura cliente-servidor. El ataque del presente laboratorio tiene como finalidad extraer la zona DNS de un servidor configurado en la distribución Ubuntu Server 11.04. Los sistemas operativos que componen la red de laboratorio fueron configurados bajo los siguientes parámetros de red, ver tabla 2.

	Computador 1	Computador 2
Sistema Operativo:	*Ubuntu Server 12.04*	*Linux BackTrack 5*
Dirección IPv4:	192.168.22.100	192.168.22.88
Mascara de subred:	255.255.255.0	255.255.255.0
Dirección de red:	192.168.22.0	192.168.22.0
Dirección de broadcast:	192.168.22.255	192.168.22.255

Tabla 2 – Configuración de los equipos de laboratorio 1

Verificando resolución del servidor DNS. La finalidad es verificar que el servicio de nombres de dominio se ha configurado exitosamente. Para realizar este proceso se debe ejecutar el navegador de Internet en el cliente y en la barra de navegación se procede a digitar el dominio configurado anteriormente (www.pentesting.local).

Descubriendo servicios mediante el análisis de puertos abiertos. En primera instancia se realiza un escaneo de puertos mediante la herramienta de seguridad Nmap para descubrir que puertos abiertos se encuentran en el sistema servidor de la red. Se revisó el recurso *Service Name and Transport Protocol Port Number Registry* de la organización *Internet Assigned Numbers Authority* (IANA), en donde se establecen la asignación de puertos estándares para las comunicaciones cliente-servidor en las redes de comunicaciones. El protocolo de resolución de nombre de dominios que utiliza el puerto 53 para sus funcionalidades.

Verificando resolución de nombres del servidor DNS. Se realizó una consulta DNS mediante la utilidad de *dig* para verificar que el servidor está resolviendo las direcciones IP a dominios y viceversa. Al ejecutar *dig*, se pudo determinar que el *hostname* del servidor DNS es *serverftp*, y además, se puede observar que el dominio es *pentesting.local* y que el servidor tiene habilitado el puerto UDP 53 el cual utiliza el servicio DNS para las consultas.

Verificando resolución de nombres del servidor DNS mediante la utilidad *nslookup*. Nuevamente, se realiza una consulta DNS mediante la utilidad de *nslookup* para verificar que el servidor está resolviendo las direcciones IP a dominios y viceversa.

Estructurando el vector de ataque con la herramienta de análisis DNS – Fierce. Para acceder a la herramienta se va a BackTrack >> *Information Gathering* >> *Network Analysis* >> *DNS Analysis* >> *Fierce*.

Al estar en el directorio donde se encuentra almacenada la herramienta, se ejecuta el comando *ls* para revisar que archivos contiene el directorio. En la ruta del sistema donde hace referencia *fierce* se muestran dos archivos: el primero, llamado fierce.pl que es el ejecutable del programa en perl. Además, esta contenido el archivo hosts.txt el cual es un diccionario de datos para realizar comparaciones fuerza bruta. Para lanzar el ataque al servidor de nombres de dominio BIND instalado y configurado en la distribución de Ubuntu Server 12.04, se debe ejecutar el siguiente comando:

```
# ./fierce.pl –dns pentesting.local –w hosts.txt
```

Dónde:

./fierce.pl	Comando para ejecutar la utilidad fierce
-dns	Opción para indicar el DNS que se debe escanear
pentesting.local	Nombre de dominio al cual se debe realizar el ataque
–w	Indicar un *wordlist* o diccionario de datos a utilizar
hosts.txt	Nombre del diccionario a emplear

Si se desea guardar los resultados del ataque para analizarlos posteriormente se puede adicionar la opción > testfierce.txt para almacenar la salida de resultados en un archivo de texto de salida. Por los que la sentencia completa adicionando este comodín queda de la siguiente manera:

```
# ./fierce.pl –dns pentesting.local –w hosts.txt > testfierce.txt
```

El ataque de transferencia de zona DNS permite descubrir la debilidad de los servidores que se ejecutan sobre plataformas GNU/Linux mediante el servicio *Berkeley Internet Name Server* lo cual establece que si el servicio BIND tiene una configuración con un nivel bajo de seguridad se pueden implementar con éxito este tipo de ataques. Analizando los resultados obtenidos mediante la generación de un ataque de transferencia de zona DNS se debe tener presente que las configuraciones por defecto de los programas no son las más seguras por lo que se debe tener en cuenta agregar directrices que eleven el nivel de seguridad de los sistemas de cómputo. La recomendación para que este tipo de ataque no tenga éxito en un ambiente empresarial es agregar una directiva al archivo named.conf.local el cual inhabilita la transferencia. La utilidad allow-transfer { none; }; la cual establece que la transferencia de zona DNS no está permitida para ningún equipo, por lo tanto al generar nuevamente el ataque expuesto anteriormente no se debe obtener la transferencia del archivo del servidor BIND.

Laboratorio 2. Implementación de un ataque de autenticación FTP entre sistemas GNU/Linux arquitectura cliente servidor.

La práctica de laboratorio se realizó en una red de área local mediante el protocolo de comunicaciones Internet Protocol 4 bajo la arquitectura cliente-servidor. Además, se utilizó el protocolo de transferencia de archivos FTP. Los sistemas operativos que componen la red de laboratorio fueron configurados bajo los siguientes parámetros de red, ver tabla 3.

	Computador 1	**Computador 2**
Sistema Operativo:	*Ubuntu Server 12.04*	*Linux BackTrack 5*
Dirección IPv4:	192.168.22.100	192.168.22.88
Mascara de subred:	255.255.255.0	255.255.255.0
Dirección de red:	192.168.22.0	192.168.22.0
Dirección de broadcast:	192.168.22.255	192.168.22.255

Tabla 3 – Configuración de los equipos de laboratorio 2

Descubriendo servicios mediante el análisis de puertos abiertos. En primera instancia se realiza un escaneo de puertos mediante la herramienta de seguridad Nmap para descubrir que puertos abiertos se encuentran en el sistema servidor de la red. En el recurso *Service Name and Transport Protocol Port Number Registry* se verifica que el puerto 21 que se encuentra ejecutándose corresponde a un servidor FTP.

Verificando conexión al servidor FTP. Se realiza una petición de conexión al servidor FTP desde la Shell del equipo cliente, para determinar que el servicio se está ejecutando y se encuentra disponible para la trasferencia. El testeo de conexión se realiza mediante el comando # ftp 192.168.22.100. Se puede realizar una segunda prueba desde el navegador de Internet para confirmar la funcionalidad del servidor FTP. Para ello se ingresa ftp://192.168.22.100 en la barra de direcciones. De esta forma se comprueba que el servidor FTP se está ejecutando y que se solicita las credenciales de autenticación. Una vez autenticado el servidor permitirá acceder al usuario a los archivos almacenados en el servidor FTP.

Estructurando el vector de ataque con la herramienta Hydra. Para acceder a la herramienta se va a BackTrack >> *Privilege Escalation* >> *Password Attacks* >> *Online Attacks* >> *hydra*.

Al seleccionar la herramienta se muestra el contenido de funcionalidades que contiene la herramienta para realizar las acciones requeridas para un test de seguridad al protocolo de transferencia de archivos FTP. Al estar en el directorio donde se encuentra almacenada la herramienta, se ejecuta el comando ls para revisar que archivos contiene el directorio. En la ruta del sistema donde hace referencia Hydra se muestran dos archivos pass.txt y p.txt los cuales son diccionarios de datos para realizar comparaciones a fuerza bruta. Para lanzar el ataque al servidor de transferencia de archivos vsftpd instalado y configurado en la distribución de Ubuntu Server 12.04, se debe ejecutar el siguiente comando:

hydra –L /root/pass.txt –P /root/pass.txt ftp://192.168.22.100

Dónde:

./fierce.pl	Es el comando para ejecutar la utilidad fierce
–dns	Opción para indicar el dns que se debe escanear
pentesting.local	Nombre de dominio al cual se debe realizar el ataque
–w	Indicar un *wordlist* o diccionario de datos a utilizar
hosts.txt	Nombre del diccionario a emplear

Cuando la herramienta de seguridad *hydra* termina de realizar el proceso del ataque, muestra en la *shell* del cliente las credenciales respectivas (admonftp, pentesting, administrator) de usuarios encontrados (adminftp, userpentest, adminlab).

El ataque de autenticación a fuerza bruta mediante diccionario de datos, permite descubrir la inseguridad del servidor VSFTPD, el cual realiza el proceso de autenticación de usuarios mediante texto en claro. El cual permite a un atacante informático por medio de comparaciones forzadas descubrir nombres de usuarios y contraseñas. Para evitar accesos a cuentas de usuarios de sistema, los usuarios de los servidores FTP deben ser creados como usuarios virtuales y con su respectiva contraseña cifrada, además emplear

letras, números y caracteres especiales para evitar ser descifrada con ataques de fuerza bruta empleando diccionarios de palabras; de la misma forma los usuarios del servidor FTP pueden estar contenidos dentro de una base de datos *mysql*, para una mayor seguridad. Configurar el servicio bajo los mecanismos SSL o TLS para cifrar la sesión completa, y evitar ataques de *sniffing*, en donde se revele información de autenticación.

Laboratorio 3. Implementación de una búsqueda de vulnerabilidades en base de datos SQL entre sistemas GNU/Linux arquitectura cliente servidor.

La práctica de laboratorio se realizó en una red de área local mediante el protocolo de comunicaciones Internet Protocol 4 bajo la arquitectura cliente-servidor. Además, se utilizó el protocolo HTTP para configurar un sitio WEB, adicionalmente empleando una base de datos bajo MySQL. Los sistemas operativos que componen la red de laboratorio fueron configurados bajo los siguientes parámetros de red, ver tabla 4.

	Computador 1	**Computador 2**
Sistema Operativo:	*Ubuntu Server 12.04*	*Linux BackTrack 5*
Dirección IPv4:	192.168.22.100	192.168.22.88
Mascara de subred:	255.255.255.0	255.255.255.0
Dirección de red:	192.168.22.0	192.168.22.0
Dirección de broadcast:	192.168.22.255	192.168.22.255

Tabla 4 – Configuración de los equipos de laboratorio 3

Descubriendo servicios mediante el análisis de puertos abiertos. En primera instancia se realiza un escaneo de puertos mediante la herramienta de seguridad Nmap para descubrir que puertos abiertos se encuentran en el sistema servidor de la red. En el recurso *Service Name and Transport Protocol Port Number Registry* se verifica que el puerto 80 que se encuentra ejecutándose corresponde a un servidor http.

Verificando conexión al servidor HTTP con MySQL. Por medio de un navegador web accedemos a la ruta donde está alojada la aplicación, se ingresa las credenciales y se observa que es correcto el acceso a la aplicación.

Estructurando el vector de ataque con la herramienta de escaneo sqlmap. Para acceder a la herramienta se va a BackTrack >> *Vulnerebility Assessment* >> *Web Assessment* >> *Vulnerability Scanners* >> *sqlmap*. Para lanzar el ataque a la aplicación, se debe ejecutar el siguiente comando:

```
# python sqlmap.py –u “url_vulnerable” --cokie=“cookie de sesion”
                    -technique=BEUST --users --passwords
```

Dónde:

Python sqlmap.py	Comando para ejecutar la utilidad sqlmap
–u	Indica la URL de la aplicación con vulnerabilidad
--cookie	Cookie de sesión en donde existe la vulnerabilidad de aplicación

--technique	Indica la técnica de inyección, en este caso técnicas basadas en todas las opciones unión de columnas, unión de caracteres, basadas en error
--users	Permite enumerar los usuarios de la base de datos de mysql
--passwords	Permite obtener los *passwords* de los usuarios de mysql

Al lanzar el escaneo, la herramienta emplea las distintas técnicas de *sql injection*, para la búsqueda de información. Primero enumera los usuarios de la base de dato mysql y seguido extrae el password de forma cifrada, y compara con su diccionario de palabras si existe coincidencia de palabras y da en texto claro la correspondiente contraseña. Cuando la herramienta de seguridad sqlmap termina de realizar el proceso del ataque, muestra en la *shell* del cliente las credenciales de usuarios encontradas. El ataque con técnicas de inyecciones SQL en aplicaciones web vulnerables, permite descubrir información por medio de consultas a la base de datos; por donde se puede extraer información de las tablas correspondientes a todas las bases de datos que se encuentran instaladas en el servidor. (Costanzo & Amerini, 2014) . Emplear contraseñas con letras mayúsculas y minúsculas, números y caracteres especiales en los usuarios de administración de mysql, además de cifrar la información de las tablas de las bases de datos existentes en el servidor. En aplicaciones web validar las consultas de los datos empleando la función mysql_real_escape_string, esta función antepone barras invertidas a los caracteres \ x00, \n, \r, \, ‘, “ y \x1a, para no permitir consultas en los campos.

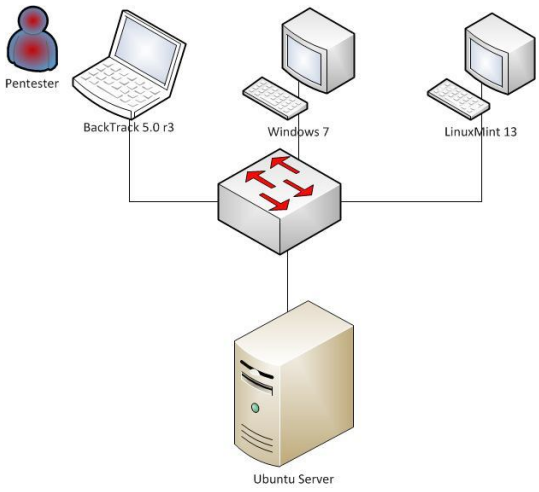


Figura 2 – Topología de trabajo

El informe de resultados contiene la información obtenida durante el proceso de investigación, es la consolidación del test de intrusión ejecutado. Se detallan las vulnerabilidades encontradas en el entorno servidor GNU/Linux, y además se dictan las respectivas recomendaciones para solucionarlas. La Aplicación del test, se

realizó en una red LAN, en ambiente controlado, conformada por equipos activos, pasivos y un dispositivo de red intermedio (Switch). El análisis de vulnerabilidad realizado se basó metodológicamente tomando como referencia apartes del "Manual de metodología abierta para el testeo de seguridad" OSSTMM, en combinación con guías de penetración. Las vulnerabilidades encontradas, se obtuvieron a través de la ejecución de una guía práctica de laboratorio, donde se realizaron pruebas de penetración por cada uno de los servicios reconocidos, además se aplicó el software de búsqueda de vulnerabilidades conocido como NESSUS, arrojando este un reporte y cotejándose el mismo con las pruebas realizadas manualmente. Se exponen las vulnerabilidades halladas con su respectiva recomendación para lograr la corrección de la falla, esta técnica es conocida como HARDENING. La topología de trabajo se encuentra representada en la figura 2.

La seguridad de los sistemas de los equipos servidores de las organizaciones es de vital importancia para que el servicio se mantenga siempre disponible. Además, las debidas y correctas configuraciones permiten que no se generen eventos que puedan causar riesgos o gran impacto en el funcionamiento de la infraestructura de TI (Yadav & Dong, 2014). Por esto se recomienda que el personal encargado de la seguridad de los diferentes sistemas informáticos, sean conocedores de la problemática, es decir estén capacitados para ejercer esa función, además la seguridad de la información implica un conocimiento de normas de seguridad en todos los ejes humanos y técnicos de la organización, en otras palabras se debe capacitar desde el celador hasta el gerente de la compañía en aspectos básicos de seguridad informática, para de esta no tener ningún tipo de inconveniente, ya que en su mayoría de casos, los grandes fallos de seguridad ocurren por descuidos y falta de conocimiento de los empleados en el manejo correcto de las aplicaciones con las que cuenta la organización. Después de realizar el análisis de vulnerabilidad, se logran identificar un conjunto de vulnerabilidades especificadas en la tabla 5.

Servicio	Software de servicio	Ataque/ Vulnerabilidad	Recomendación
DNS	**BIND9**	**Transferencia de zona**	**La recomendación para que este tipo de ataque no tenga éxito en un ambiente empresarial es agregar una directiva al archivo named.conf. local el cual inhabilita la transferencia.**
FTP	VSFTPD	Autenticación por fuerza bruta	Para evitar accesos a cuentas de usuarios de sistema, los usuarios delos servidores FTP deben ser creados como usuarios virtuales y con su respectiva contraseña cifrada. Los usuarios del servidor FTP pueden estar contenidos dentro de una base de datos mysql, para una mayor seguridad. Configurar el servicio bajo los mecanismos SSLo TLS para cifrar la sesión completa, y evitar ataques de sniffing, en donde se revele información de autenticación.

Base de datos	MySQL	Inyección de código SQL	Emplear contraseñas con letras mayúsculas y minúsculas, números y caracteres especiales en los usuarios de administración de mysql, además de cifrar la información de las tablas de las bases de datos existentes en el servidor. En aplicaciones web validar las consultas de los datos empleando la función mysql_real_escape_string, esta función antepone barras invertidas a los caracteres \ x00, \n, \r, \, ‘, “ y \x1a, para no permitir consultas en los campos

Tabla 5 – Definición de algunos estilos

El diagnóstico situacional, fue recopilado en base a documentación científica, o en su defecto de páginas electrónicas especializadas en la temática tratada, permitiendo de esta forma lograr una información coherente y actualizada. La implementación de la arquitectura Cliente-Servidor, comprendida esta por: sistema operativo y aplicaciones de servicios basadas en Linux, permiten por su licencia libre, realizar modificaciones según las necesidades de configuración de los mismos, obteniendo robustez y estabilidad.

4. Conclusiones

Los acontecimientos informáticos que suceden a diario, demuestran que la seguridad de la información debe ser tema de primer nivel dentro de una organización, empezando desde el o la recepcionista, pasando por el departamento de sistemas, seguido del oficial de seguridad de la información (CISO), y culminando en la junta directiva o gerencia de la misma organización. La aplicación de las diversas pruebas en los escenarios propuestos y la variedad de protocolos probados, demuestran la robustez de la herramienta de auditoría BackTrack 5.0 R2, cuenta con aproximadamente 300 utilidades, entre las que facilitaron las pruebas de penetración a servicios y protocolos como: DNS, FTP Y SQL, además su distribución basada en Ubuntu nos permite actualizar, agregar o eliminar paquetes y herramientas, posibilitando generar un ambiente personalizado según las necesidades del pentester.

La vulneración al protocolo DNS, a través del ataque de transferencia de zona realizado en el primer laboratorio, permitió demostrar la debilidad del protocolo y el riesgo que genera a una organización la materialización de este; ya que se logró recopilar toda la información disponible en el protocolo, como información de los subdominios y detalles de la infraestructura de TI, información valiosa en el proceso de “information gathering” , paso importante que se encuentra propuesto en el estándar OSSTMM. En el segundo laboratorio, el vector de ataque se centra en la vulneración del protocolo FTP (Protocolo de transferencia de archivos) a través de la aplicación de una técnica de fuerza bruta con diccionario, utilizando la herramienta Hydra, logrando comprometer de manera exitosa el protocolo. Esto permite concluir que es un protocolo no seguro y que debe ser sustituido por la versión segura del mismo, el protocolo seguro de transferencia de archivos, SFTP.

En ataques a bases de datos por medio de técnicas como la inyección de código, se concluye que a estas se les debe realizar todas aquellas medidas de hardering existentes, referentes a bases de datos; se recomienda la aplicación de técnicas, como el uso de contraseñas robustas con criterios alfanuméricos y el uso de minúsculas y mayúsculas, el cifrado de las tablas, y la utilización de funciones sql actuales, sin olvidar la actualización permanente de la versión del motor de la base de datos. Las distribuciones de software libre para entorno servidor, son totalmente estables, los servicios software para este tipo de plataformas cuentan con excelente soporte y documentación y la flexibilidad de las licencias libres permiten al administrador de servicios jugar con los componentes del sistema operativo acomodándolos a sus necesidades. La utilización de software libre, permite la disminución de gastos contribuyendo con la viabilidad financiera de una organización, ya que este tipo de software no requiere la cancelación en efectivo por el uso de su licencia, al ser estas totalmente gratis.

En el aspecto metodológico, se puede concluir que el estándar de referencia OSSTMM, es una guía técnica y completa, permitiendo llevar a buen término cualquier tipo de análisis de seguridad. Los análisis de seguridad son versátiles, no hay que seguir una metodología al pie de la letra, estas brindan un referente o una base, cada organización realiza sus procesos de manera diferente, por lo que cada análisis arrojara resultados diferentes a los otros, por eso la libertad y habilidad del pentester para ejecutar una metodología creada en base a muchas.

Un análisis de vulnerabilidad es un tipo de test de seguridad práctico, al aplicarse se obtuvieron resultados contundentes de manera rápida, y es una buena opción cuando se carece de tiempo para realizar una auditoría exhaustiva a un sistema y además de ser más económica que otros tipos de test existentes en el mercado. Será efectivo, solo si los resultados son gestionados de manera clara, concisa y práctica, permitiendo demostrar a todos los componentes de una organización las fallas que ocurren, para que de esta forma se han gestionados como una política de la empresa, y sea aplicada en todos sus ejes. Se puede usar este documento para la base de la edición final del artículo que se va a enviar. Durante el proceso de análisis de vulnerabilidad, se encuentra que los servicios configurados carecen de un proceso de Hardering, es recomendable que los administradores de red, o administradores de seguridad de la información, de manera constante realicen jornadas de actualización en los conocimientos de seguridad de la información.

Referencias

Alvernia Acevedo, S. A., & Rico-Bautista, D. (2017). Análisis de una red en un entorno IPV6: Una mirada desde las intrusiones de red y el modelo TCP/IP. *REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA)*, *1*(29). https://doi.org/10.24054/16927257.v29.n29.2017.2490

Andrew, M. W. (2007). Defining a process model for forensic analysis of digital devices and storage media. *Proceedings - SADFE 2007: Second International Workshop on Systematic Approaches to Digital Forensic Engineering*. https://doi.org/10.1109/SADFE.2007.8

Bonetti, G., Viglione, M., Frossi, A., Maggi, F., & Zanero, S. (2013). A comprehensive black-box methodology for testing the forensic characteristics of solid-state drives. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/2523649.2523660

Boynton, B. C. (2007). Identification of process improvement methodologies with application in information security. *InfoSecCD'07: Proceedings of the 4th Annual Conference on Information Security Curriculum Development*. https://doi.org/10.1145/1409908.1409939

Carter, K. M., Idika, N., & Streilein, W. W. (2014). Probabilistic threat propagation for network security. *IEEE Transactions on Information Forensics and Security*. https://doi.org/10.1109/TIFS.2014.2334272

Gerardo, L., & Fraga, D. (2003). *Redes Seguras Usando el Sistema GNU / Linux Resumen*. *May*.

Hernandez, J. L. (2014). ¿Qué son las vulnerabilidades del software? *Vulnerabilidad De Software*.

Lobo-Parra, L. D., Rico-Bautista, D., Medina-Cárdenas, Y., & Sanchez-Ortiz, E. A. (2019). Defining of a practical methodology for the acquisition and analysis of digital evidence in the context of a research post mortem. *RISTI: Revista Ibérica de Sistemas y Tecnologías de La Información*, *E18*, 369–384.

López, Ó., Amaya, H., León, R., & Acosta, B. (2001). Informática forense: generalidades, aspectos técnicos y herramientas. *Universidad de Los Andes*. ….

Lyubimov, A. V. (2010). Integral engineering technique for information security methodologies. *SIN'10 - Proceedings of the 3rd International Conference of Security of Information and Networks*. https://doi.org/10.1145/1854099.1854105

Marini, E. (2012). El Modelo Cliente/Servidor. In *Octubre*.

Medina-Cárdenas, Y., Areniz Arévalo, Y., & Rico-Bautista, D. (2016). Modelo estratégico para la gestión tecnológica en la organización: plan táctico de la calidad (ITIL & ISO 20000). In *Fondo editorial ITM* (Vol. 1). Instituto Tecnológico Metropolitano. https://doi.org/10.22430/9789585414006

Medina-Cárdenas, Y. C., & Rico-Bautista, D. (2016). Alineación estratégica bajo un enfoque organizacional de gestión tecnológica: ITIL & ISO 20000. *Revista Tecnura*, *20*(1), 82–94. https://doi.org/10.14483/22487638.11681

Medina-Cárdenas, Y., & Rico-Bautista, D. (2012). Mejores prácticas de gestión para la calidad de los servicios en tecnologías de información. *Gerencia Tecnológica Informática*.

Medina-Cárdenas, Y., Rico-Bautista, D., & Arevalo-Ascanio, J. G. (2019). Impact of the implementation of iso 9001 in the university: Perception auditor and leader. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*.

Ponemon Institute. (2015). 2015 Cost of Cyber Crime Study: Global. *Ponemon Institute Research Report*.

Ramos Ramos, L. J. (2014). Pruebas De Penetración O Pent Test. *Revistasbolivianas*.

Rico-Bautista, D., Guerrero, C. D., Medina-Cárdenas, Y., & García-Barreto, A. (2019). Analysis of the potential value of technology: Case of universidad francisco de paula santander Ocaña. *RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao*, *E17*, 756–774.

Rico-Bautista, D., Rojas-Osorio, J., & Medina-Cárdenas, Y. (2016). Pentesting empleando técnicas de ethical hacking en redes IPv6. *Revista Ingenio UFPSO*, *11*(1), 79–96.

Rueda-Rueda, J. S., Rico-Bautista, D., & Florez-Solano, E. (2019). Open Practice Guide for Digital Forensics on Android Devices. *RISTI (Revista Ibérica de Sistemas y Tecnologías de La Información)*, *18*, 442–457.

Tian, Z., Jiang, W., Li, Y., & Dong, L. (2014). A digital evidence fusion method in network forensics systems with Dempster-shafer theory. *China Communications*. https://doi.org/10.1109/CC.2014.6880464

Valdez, A. (2013). OSSTMM 3. *Revista de Información, Tecnología y Sociedad*.

Veloz, J., Alcivar, A., Salvatierra, G., & Silva, C. (2017). Ethical hacking, una metodología para descubrir fallas de seguridad en sistemas informáticos mediante la herramienta KALI-LINUX. *Informática y Sistemas: Revista de Tecnologías de La Informática y Las Comunicaciones*. https://doi.org/10.33936/isrtic.v1i1.194

Yadav, S. B., & Dong, T. (2014). A comprehensive method to assess work system security risk. *Communications of the Association for Information Systems*. https://doi.org/10.17705/1cais.03408