Tema 5. Implementación de pruebas

5.1 Introducción

En los capítulos anteriores hemos mencionado los fundamentos teóricos en los que basamos la realización de nuestro proyecto.

Explicamos las bases de la seguridad informática para hacer notar la relevancia y la justificación de nuestras pruebas. Posteriormente enumeramos los ataques más comunes a los que se ven expuestos los sistemas de información y las herramientas más populares para llevarlos a cabo, con la finalidad de desglosar los procesos y los efectos de las pruebas que realizaríamos y, finalmente, establecimos la estructura de nuestros planes basándonos en las metodologías de análisis de riesgo y *hacking* ético.

A continuación, presentamos la implementación real de todos los antecedentes recopilados. Detallamos nuestros planes, su cronología, las herramientas utilizadas, su ejecución, su documentación, el análisis de resultados y nuestras recomendaciones derivadas de todos los procesos.

5.2 Plan de pruebas

Plan de pruebas

Con la finalidad de evitar la ejecución de un proyecto de seguridad limitándonos a la utilización aleatoria de diferentes herramientas, presentamos este plan de pruebas metodológico con la finalidad de definir detalles acerca de qué hacer, cuándo hacerlo y cómo hacerlo.

El tipo de prueba que se plantea llevar a cabo es el de caja blanca, esto debido a que contamos con acceso a información vital de la red como diagramas, sistemas operativos y aplicaciones instaladas.

Siguiendo la teoría existente respecto a las pruebas de penetración habituales, los siguientes son los pasos a considerar:

1. Reconocimiento del *host* y escaneo. Tiene como finalidad la recolección de información acerca de la red. Cabe mencionar que la metodología empleada para tales objetivos es la de reconocimiento activo del *host*, que consiste en la utilización de herramientas técnicas para la obtención de información. Éste es un proceso que es fácilmente detectable, sin embargo, es el que arroja mejores y más precisos resultados.

   Ejemplos del tipo de información a recolectar en primera instancia son los siguientes:

   - Direcciones IP de los equipos en la red.
   - Puertos *UDP* y *TCP* accesibles en los sistemas.
   - Sistema operativo en los sistemas.

Herramientas:

NMap. Es una aplicación que corre en varias plataformas, incluyendo Linux y Windows (la versión de Linux arroja resultados más exactos), que permite el escaneo de grandes redes. Ayuda a determinar los *hosts* activos y qué servicios están ofreciendo éstos. Permite, además, implementar muchas de las diferentes técnicas de escaneo de puertos existentes.

Cuando los resultados arrojan puertos clave abiertos NMap es capaz de determinar el sistema operativo de los *hosts*, de lo contrario, puede utilizarse alguna herramienta específica de *fingerprinting*.

Xprobe2. Es una aplicación que corre en Linux y que permite la identificación del sistema operativo instalado en un objetivo. Contiene una base de datos con las diferentes firmas de los sistemas operativos más conocidos aunque también incluye una puntuación probabilística para adivinarlo.
Una vez recolectada la información anterior, puede dibujarse un diagrama de red que contenga los nombres de los *hosts*, las direcciones IP, el número de los puertos activos y los sistemas operativos instalados.

2. Penetración de la red.

Servidores y *Firewalls*

Herramientas:

Nessus. Es un escáner de vulnerabilidades de código abierto que corre en diversos sistemas operativos dentro de los que se encuentran Linux y Windows. Se encarga de verificar los servicios y vulnerabilidades que se suscitan a través de puertos estándar y no estándar. Puede mostrar sus resultados en formato *HTML* para una mejor visualización.

HTTPTunnel. Es una herramienta que utiliza la técnica de *tunneling* a través del puerto *TCP* 80. Se trata de una aplicación cliente/servidor que requiere que el segundo sea instalado en el sistema objetivo. HTTPTunnel permite redirigir puertos no abiertos en un *firewall* hacia el mencionado puerto 80, con lo que puede conseguirse la utilización de servicios que de otra manera no serían permitidos.

Switches y *Routers*

Herramientas:

WiFiSlax. Es una distribución GNU/Linux con funcionalidades *LiveCD* y *LiveUSB* que incluye una larga lista de herramientas de seguridad y auditoría listas para ser utilizadas entre las que destacan numerosos escáneres de puertos y vulnerabilidades, herramientas para creación y

diseño de *exploits*, *sniffers*, herramientas de *análisis forense* y herramientas para la auditoría inalámbrica.

Ettercap. Es una suite de herramientas disponible para sistemas Windows, Linux, entre otros. Incluye métodos de *sniffing*, inyección de caracteres, colección de contraseñas, filtrado y sustitución de paquetes, envenenamientos de la red, robo de puertos, por mencionar algunos. Muchas de las funcionalidades anteriores permiten la ejecución de ataques *man in the middle* en LANs (Local Area Network) con *switch*.

Wireshark. Es un analizador de protocolos muy utilizado para observar el tráfico de una red de datos y solucionar problemas suscitados en las mismas. Cuenta con una *interfaz* gráfica con varias opciones de filtrado y de organización de la información además de la capacidad de reconstruir el flujo completo de una sesión *TCP*.

3. Mantener acceso. Establecer un *backdoor* en el sistema para futuros ataques. A pesar de que es importante conocer acerca de virus y gusanos cuando se planea la manera de asegurar la red de una organización, no son herramientas comunes que se utilicen en las pruebas de penetración. Sin embargo, los caballos de Troya son utilizados normalmente para ganar y mantener acceso a sistemas comprometidos.

   Es ésta la manera en que realizaremos esta fase dentro de nuestra metodología.

   Herramientas:

   Back Orifice 2000 (BO2K). Es una herramienta cliente-servidor de administración remota disponible para plataformas Windows y Linux que permite actividades como la edición de registros, transferencia de archivos, creación de línea de comandos, control de procesos, apagado y reinicio remoto, captura de archivos de contraseñas, captura de pantallas, control de ratones y teclados, comunicación cifrada, entre otras.

   El *servidor* debe ser instalado en el equipo objetivo, por lo cual, es un proceso que se realiza una vez que éste ha sido comprometido. Tiene la ventaja de que puede ser ocultado del administrador de tareas de Windows aun estando en ejecución.

   La ejecución de todas las funcionalidades señaladas anteriormente dependen de la instalación de diversos *plug-ins*, agrupados en categorías como cifrado, autenticación, habilitación de servidores y de clientes, comunicaciones y misceláneo.

4. Destrucción de la evidencia. Borrar los archivos necesarios para ocultar el hecho de que la red ha sido vulnerada, y que en un posible *análisis forense* posterior, ésta no pueda ser usada en contra del atacante.

Para este efecto, la herramienta Back Orifice 2000 descrita arriba, brinda opciones útiles.

5.3 Documentación

5.3.1 Resumen

Este reporte detalla varias pruebas de intrusión recientes hechas en un laboratorio de cómputo de la UNAM, realizadas por los alumnos Luis Hugo Flores Román y Gustavo Gabriel Hernández Hernández, entre las fechas 19 de enero del 2010 y 26 de febrero de 2010. Tales pruebas tienen la finalidad de comprobar la seguridad de la red interna emulando las técnicas de un atacante malicioso. La combinación de pruebas ejecutadas contra la red incluyó el escaneo de puertos, escaneo de vulnerabilidades, ataques contra contraseñas entre otras, detalladas más adelante en el reporte.

Tras analizar los resultados de las pruebas realizadas se recomienda lo siguiente para mejorar la seguridad de la red:

- Actualizar de manera regular las versiones del *kernel* de los sistemas operativos Linux o los service pack de los sistemas operativos Windows instalados en los *hosts*.

- Actualizar las versiones de los programas que proveen los servicios instalados en los *hosts*.

- Actualizar de manera regular las definiciones de virus de los antivirus instalados en los *hosts*.

- Realizar una vigilancia más estrecha en las actividades que realizan los alumnos dentro del laboratorio.

- Actualizar las aplicaciones y servicios que corren bajo el *servidor*.

- No permitir la existencia de cuentas de cualquier servicio que funcionen con los valores por defecto.

- Cuidar la información que se publica en la página web del laboratorio.

- Aplicar un protocolo de cifrado al canal de comunicación con el *switch*.

- Uso de técnicas criptográficas en la autenticación de la red inalámbrica como, por ejemplo, *WEP*. Una mejor opción sería la utilización de claves dinámicas con *WPA* o *WPA2*.

- Desactivar la difusión del identificador de red inalámbrica o *SSID*.

- Administrar la autenticación de la red inalámbrica mediante un servidor RADIUS.

Se incluye dentro de este documento una explicación acerca del ámbito en el que el proyecto fue desarrollado, seguido de los resultados completos de las pruebas y el análisis de los resultados de las mismas.

5.3.2 Ámbito del proyecto

El tipo de *pentesting* realizado fue el de caja blanca, debido a que se tuvo acceso previo a información referente al objetivo, como diagrama de red, direcciones IP, entre otras.

La pruebas concernientes a este reporte fueron realizadas dentro de la red privada de un laboratorio de cómputo de la UNAM, con un rango de IP's comprendido entre la 172.16.1.1 a la 172.16.1.13, excluyendo únicamente la dirección 172.16.1.8. Adicionalmente, se corrieron pruebas en el *servidor* web, *router* inalámbrico y *switch*.

Se permitió el uso de troyanos y *backdoors* en el rango de IP's antes mencionado, más no en los demás dispositivos. Esto para no comprometer el desarrollo de las actividades diarias del laboratorio.

Alrededor de 19200 *exploits* fueron probados en contra del *servidor* y de la red. La mayoría de ellos pueden ser clasificados, pero no limitados, a las siguientes categorías:

- Escaneo de puertos.
- *Fingerprinting*.
- Vulnerabilidades de *servidor* web.
- Vulnerabilidades de *servidor FTP*.
- Ataques de diccionario.
- Vulnerabilidades de administración remota.
- Vulnerabilidades de *switch*.
- Vulnerabilidades de *router* inalámbrico.
- Vulnerabilidad a aplicaciones *backdoor*.

5.3.3 Análisis de resultados

A continuación presentamos un breve análisis de los resultados obtenidos, después de que se aplicaran las pruebas arriba mencionadas.

A. Reconocimiento del host y escaneo

Las pruebas llevadas a cabo en esta fase incluyeron escaneos del tipo *ARP* ping y SYN, además de la detección de sistemas operativos.

En primera instancia, se utilizó la herramienta Nmap dentro de un sistema operativo Windows (escaneo intensivo de todos los puertos TCP) aunque, para obtener resultados más exactos, posteriormente se hizo uso de la herramienta Xprobe2 dentro de un ambiente Linux para comparar los resultados de *fingerprinting*.

La información recopilada y las vulnerabilidades descubiertas se enlistan a continuación. (Véase figura 34.)

Listado de equipos

- IP: 172.16.1.1 (servidor, interna)
  Sistema operativo: Linux *Kernel* 2.6.0 – 2.6.7

  Puertos abiertos

Puerto	Servicio	Versión
21/TCP	ftp	ProFTPD 1.3.1
22/TCP	ssh	OpenSSH 3.6.1p2
80/TCP	http	Apache httpd 2.2.9
111/TCP	rpcbind	2
199/TCP	smux	Linux SNMP multiplexer
443/TCP	http	Apache httpd 2.2.9
1311/TCP		
3306/TCP	mysql	MySQL
5555/TCP		
8000/TCP		
32768/TCP	status	1
32774/TCP		

  Vulnerabilidades

Servicio	Descripción	Recomendación
FTP	Vulnerabilidad SQL injection en los módulos mod_sql_mysql y mod_sql_postgress, asociada a la versión 1.3.1 de proFTPD.	Actualización de la aplicación.
SSH	Vulnerabilidad DoS o ejecución arbitraria de código utilizando buffer_init y buffer_free en buffer.c, asociada a versiones anteriores a la 3.7.1 de OpenSSH.	Actualización de la aplicación.
HTTP	Vulnerabilidad DoS debida a un fallo en el manejo de respuestas excesivas de un servidor origen cuando se utiliza mod_proxy_http, asociada a la versión 2.2.9 de Apache httpd.	Actualización de la aplicación.

- IP: 172.16.1.2
  Sistema operativo: Linux *Kernel* 2.6.8

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 5.0

  Vulnerabilidades

Servicio	Descripción	Recomendación
Linux Kernel 2.6.8	Vulnerabilidad DoS local debida a una falla al manejar conexiones SCTP. Vulnerabilidad de la función cpuset_task_read() dentro de /kernel/cpuset.c, debida a un error underflow, que puede ser explotada para leer la memoria del kernel. Vulnerabilidad debida a una falla del kernel en el manejo de las semillas generadoras de números aleatorios, debilitando así la seguridad de las aplicaciones que confían en el debida generación de los números antes mencionados.	Actualización del Kernel instalado en el host.

- IP: 172.16.1.3
  Sistema operativo: Linux *Kernel* 2.6.9 – 2.6.11

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 4.3
111/TCP	rpcbind	2
42998/TCP	status	1

  Vulnerabilidades

Servicio	Descripción	Recomendación
SSH	Vulnerabilidad DoS de pre-autenticación debida a la posibilidad de incurrir en una condición de carrera en el manejador de señales, asociada a la versión 4.3 de OpenSSH.	Actualización de la aplicación.

- IP: 172.16.1.4
  Sistema operativo: Linux *Kernel* 2.6.9 – 2.6.11

Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 4.3
111/TCP	rpcbind	2
57951/TCP	status	1

Vulnerabilidades

Servicio	Descripción	Recomendación
SSH	Vulnerabilidad DoS de pre-autenticación debida a la posibilidad de incurrir en una condición de carrera en el manejador de señales, asociada a la versión 4.3 de OpenSSH.	Actualización de la aplicación.

- IP: 172.16.1.5
  Sistema operativo: Linux *Kernel* 2.6.9 – 2.6.11

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 4.3
111/TCP	rpcbind	2
54968/TCP	status	1

  Vulnerabilidades

Servicio	Descripción	Recomendación
SSH	Vulnerabilidad DoS de pre-autenticación debida a la posibilidad de incurrir en una condición de carrera en el manejador de señales, asociada a la versión 4.3 de OpenSSH.	Actualización de la aplicación.

- IP: 172.16.1.6
  Sistema operativo: Linux *Kernel* 2.6.8

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 5.0

Vulnerabilidades

Servicio	Descripción	Recomendación
Linux Kernel 2.6.8	Vulnerabilidad DoS local debida a una falla al manejar conexiones SCTP. Vulnerabilidad de la función cpuset_task_read() dentro de /kernel/cpuset.c, debida a un error underflow, que puede ser explotada para leer la memoria del kernel. Vulnerabilidad debida a una falla del kernel en el manejo de las semillas generadoras de números aleatorios, debilitando así la seguridad de las aplicaciones que confían en el debida generación de los números antes mencionados.	Actualización del Kernel instalado en el host.

- IP: 172.16.1.7
  Sistema operativo: Linux *Kernel* 2.6.8

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 5.0

  Vulnerabilidades

Servicio	Descripción	Recomendación
Linux Kernel 2.6.8	Vulnerabilidad DoS local debida a una falla al manejar conexiones SCTP. Vulnerabilidad de la función cpuset_task_read() dentro de /kernel/cpuset.c, debida a un error underflow, que puede ser explotada para leer la memoria del kernel. Vulnerabilidad debida a una falla del kernel en el manejo de las semillas generadoras de números aleatorios, debilitando así la seguridad de las aplicaciones que confían en el debida generación de los números antes mencionados.	Actualización del Kernel instalado en el host.

- IP: 172.16.1.9
  Sistema operativo: Microsoft Windows XP SP3

  Puertos abiertos

Puerto	Servicio	Versión
139/TCP	netbios-ssn	
445/TCP	microsoft-ds	Microsoft Windows XP microsoft-ds

Vulnerabilidades: Ninguna a destacar.

- IP: 172.16.1.10
  Sistema operativo: Linux *Kernel* 2.6.0, 2.6.2

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 5.0

  Vulnerabilidades: Ninguna a destacar.

- IP: 172.16.1.11
  Sistema operativo: Linux *Kernel* 2.6.0, 2.6.2

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 5.0

  Vulnerabilidades: Ninguna a destacar.

- IP: 172.16.1.12
  Sistema operativo: Linux *Kernel* 2.6.9 – 2.6.11

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 4.3
111/TCP	rpcbind	2
51318/TCP	status	1

  Vulnerabilidades

Servicio	Descripción	Recomendación
SSH	Vulnerabilidad DoS de pre-autenticación debida a la posibilidad de incurrir en una condición de carrera en el manejador de señales, asociada a la versión 4.3 de OpenSSH.	Actualización de la aplicación.

- IP: 172.16.1.13
  Sistema operativo: Linux *Kernel* 2.6.9 – 2.6.11

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 3.9p1
111/TCP	rpcbind	2
32769/TCP	status	1

Vulnerabilidades

Servicio	Descripción	Recomendación
SSH	Esta versión de OpenSSH (3.9p1) no registra la fuente de las conexiones rechazadas.	Actualización de la aplicación.

- IP: 180.20.20.1 (servidor, externa)
  Sistema operativo: Linux *Kernel* 2.6.9 – 2.6.11

  Puertos abiertos

Puerto	Servicio	Versión
22/TCP	ssh	OpenSSH 3.6.1p2
80/TCP	http	Apache httpd 2.2.9
443/TCP	http	Apache httpd 2.2.9

  Vulnerabilidades

Servicio	Descripción	Recomendación
SSH	Vulnerabilidad DoS debida al orden en que las claves son intentadas en una autenticación de llave pública, asociada a la versión 3.6.1p2 de OpenSSH.	Actualización de la aplicación.
HTTP	Vulnerabilidad DoS debida a un fallo en el manejo de respuestas excesivas de un servidor origen cuando se utiliza mod_proxy_http, asociada a la versión 2.2.9 de Apache httpd.	Actualización de la aplicación.

Diagrama de red

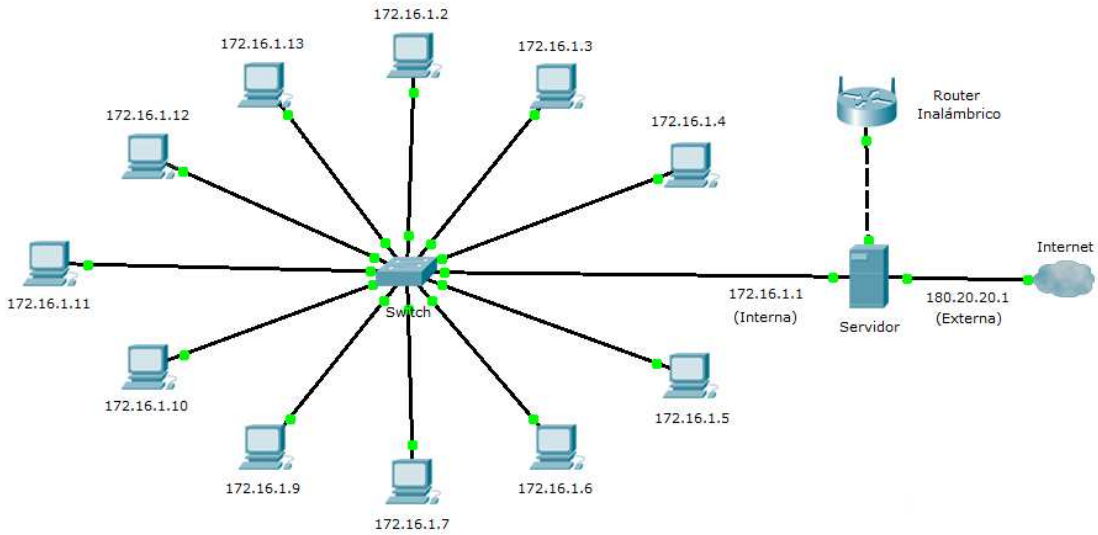

Figura 34. Diagrama de la red del laboratorio de cómputo analizado.

B. Penetración de la red

En esta fase se llevaron a cabo varios tipos de pruebas contra los diversos dispositivos del laboratorio, incluyendo escaneo de vulnerabilidades, ataques de diccionario, ataques *man in the middle*, *sniffing*, MAC *spoofing*, por mencionar algunos.

Una descripción más detallada de las pruebas realizadas en cada uno de los dispositivos y los resultados de las mismas se muestran a continuación.

Servidor

Previo a la realización de las pruebas en el *servidor* del laboratorio, se verificó que la cantidad de tráfico generado por el escáner de vulnerabilidades Nessus no fuera el suficiente para provocar un *DoS* en el dispositivo. Para lograr lo anterior, se instaló temporalmente un *servidor* (Apache 2.2.14) en uno de los *hosts* de la red interna y fue escaneado mientras se simulaba en él una carga de trabajo importante.

Una vez comprobado que la integridad de los servicios no corría riesgos, se escaneó la dirección IP externa del dispositivo, obteniendo como resultado 3 vulnerabilidades de alto riesgo, 19 de riesgo medio y 56 de bajo o riesgo nulo.

A continuación, se detallan los hallazgos más importantes. (Véase tabla 7.)

Servicio	Descripción	Riesgo	Recomendación
FTP	El servidor FTP remoto tiene configurada una cuenta con las credenciales establecidas de manera predeterminada.	Alto	Modificar la contraseña del servidor FTP remoto.
PHP	Diversas vulnerabilidades asociadas a versiones anteriores a la 5.2.7 del tipo buffer overflow en varias funciones, provocando la posibilidad de evadir restricciones de seguridad.	Alto	Actualizar la versión de PHP instalada en el servidor.
SSH	El servicio remoto soporta conexiones hechas con las versiones 1.33 o 1.5 de SSH, que ofrecen protocolos criptográficos inseguros.	Medio	Deshabilitar la compatibilidad con la versión 1 del protocolo.
HTTP	El servidor web remoto contiene un script PHP que permite el acceso a la función phpinfo(), que contiene información crítica acerca del servidor.	Medio	Remover el script PHP.
HTTP	Diversas vulnerabilidades asociadas a versiones anteriores a la 2.2.12 de Apache, por ejemplo, DoS, consumo excesivo de memoria, fallas en módulos de compresión	Medio	Asegurarse que los módulos en dónde se presenten las fallas no estén activados, o bien, actualizar la versión de

	de archivos y de flujo de datos.		Apache.
HTTP	El servidor web remoto soporta los métodos TRACE o TRACK, que son utilizados para depurar las conexiones del servidor.	Medio	Deshabilitar los métodos.
SSL	El servicio remoto soporta el uso de cifradores SSL de fuerza media o débil, es decir, aquellos con longitudes de llave de entre 56 y 112 bits o ningún cifrado en absoluto.	Medio	Reconfigurar la aplicación para evitar el uso de cifradores de fuerza media o débil.
SSL	Diversas vulnerabilidades de cifrado asociadas a versiones anteriores a la 3.0 de SSL, por ejemplo, ataques man in the middle o descifrado de comunicaciones entre los clientes y el servicio afectado.	Medio	Actualizar el servicio SSL a la versión 3.0
Dreamweaver	Dreamweaver produce archivos XML que contienen información de sincronización de archivos y directorios, lo que puede derivar en revelación de información crítica.	Medio	Desactivar la opción "Mantener información de sincronización" de la categoría "Información remota".

Tabla 7. Vulnerabilidades identificadas a través de Nessus.

Tras analizar las vulnerabilidades reportadas por Nessus, concluimos que la manera más viable de lograr un acceso no autorizado al *servidor* era mediante una conexión *FTP*.

Tras la instalación de la suite XAMPP en el dispositivo, llevada a cabo por otros compañeros tesistas, fue creada una cuenta *FTP* a la que no le fueron modificadas las credenciales predeterminadas. Lo anterior nos permitió acceder únicamente a una sección del servidor donde se alojaba la información de un proyecto de tesis y, aunque no fue posible escalar privilegios, pudimos observar los nombres de usuarios de cuentas con presumiblemente mayores facultades.

A través de la aplicación Hydra y de 5 diccionarios, buscamos *crackear* las cuentas observadas en el *servidor* y así poder alcanzar la escalación de privilegios, pero no fue posible debido a la fortaleza de las contraseñas asociadas a ellas.

Switch

En primera instancia, se empleó nuevamente el escáner de vulnerabilidades Nessus en la IP para configuración remota establecida en el dispositivo. Cabe mencionar que esta IP fue adquirida en el documento PDF de la práctica 2A del laboratorio de la materia administración de redes, por lo que es de conocimiento público.

A pesar de que los resultados arrojaron 8 vulnerabilidades de bajo o nulo riesgo, una fue muy relevante:

Vulnerabilidad	Descripción	Nivel de riesgo	Recomendaciones
Servicio de configuración remota sin cifrado	El servicio de configuración remota corre sobre un canal sin ningún método de cifrado, por tanto, los nombres de usuario, contraseñas y comandos viajan en texto claro.	Bajo o nulo.	Cifrar el canal sobre el que corre el servicio.

Conociendo lo anterior, se procedió a la utilización de ettercap y wireshark en un ambiente Linux para ejecutar los ataques *man in the middle* y *sniffing* contra el *switch* y el *host* desde el cual se realiza la configuración remota, con la intención de capturar la contraseña de administración del dispositivo.

La captura se realizó exitosamente mientras alumnos de la facultad experimentaban con la configuración del dispositivo. (Véase figura 35.)

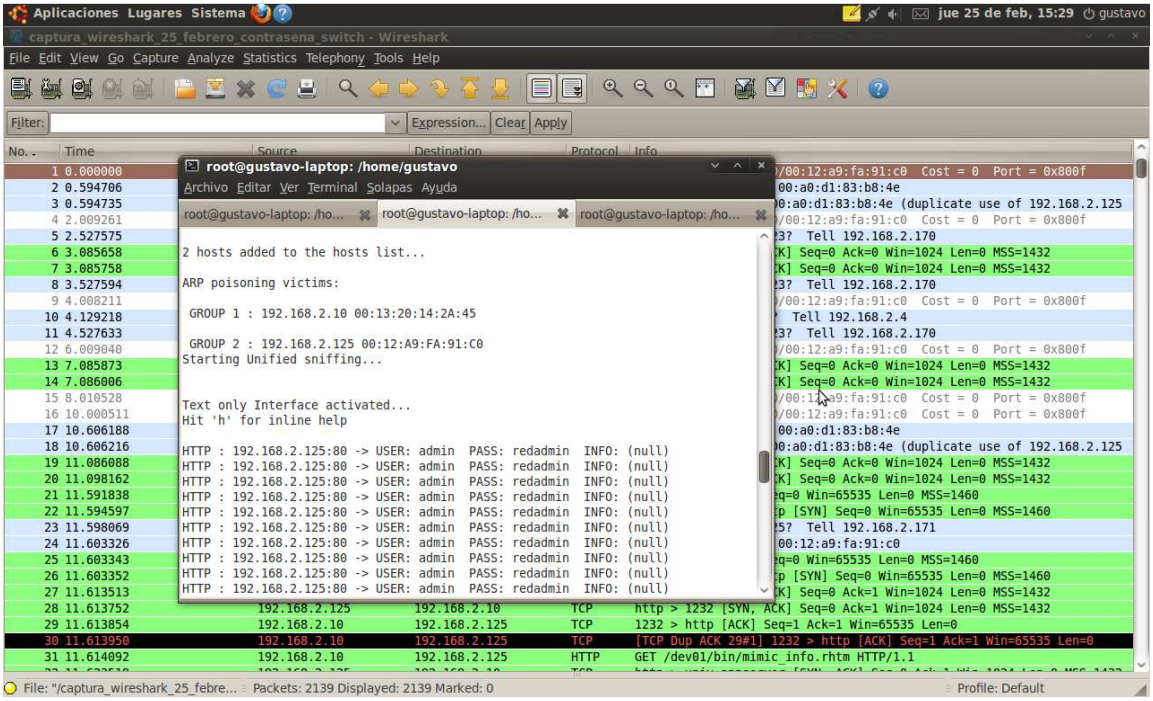

Figura 35. Captura de la contraseña de administración mediante ettercap y wireshark.

Teniendo posesión de la cuenta de administrador, cualquier otra actividad ilícita en el *switch* es fácilmente realizable.

Router

Se utilizó la distribución de Linux WifiSlax 3.1, herramienta especialmente diseñada para la auditoría de seguridad inalámbrica y lo descubierto luego de su utilización se detalla a continuación.

La red inalámbrica del laboratorio no cuenta con ningún método de cifrado en las comunicaciones, ya sea *WEP*, *WPA* o *WPA2* y se tiene habilitada la difusión del *ESSID* de la red. En cambio, sí cuenta con el método conocido como filtrado por MAC para impedir el acceso libre y no autorizado a la red del laboratorio. Sin embargo, ésta única precaución no brinda la seguridad suficiente como para que un usuario ilícito gane acceso a la red.

Haciendo uso de la distribución de Linux WifiSlax en su versión 3.1, se logró tener una amplia visión de las señales de redes inalámbricas captadas por nuestro adaptador, siendo la del laboratorio de redes y seguridad la que nos interesaba. Con la ayuda de la herramienta, pudimos obtener en cuestión de segundos el *ESSID* de la señal además de las direcciones MAC de las tarjetas inalámbricas pertenecientes al *router* y a los equipos legítimos que en ese momento se encontraban conectados a la red además de la potencia con que la señal llegaba. Estos datos no fueron los únicos proporcionados pero sí son los más significativos. Con esta información a la mano, el siguiente paso fue modificar la dirección MAC de nuestra computadora por alguna de las direcciones autorizadas mediante la aplicación para Linux llamada Macchanger. Una vez logrado esto, se pudo autenticarse en la red del laboratorio como usuario legítimo.

Vulnerabilidad	Descripción	Recomendaciones
La red no está protegida por ningún protocolo de cifrado	La utilización de protocolos de cifrado de datos para redes inalámbricas como lo son WEP y WPA son primordiales, ya que se encargan de codificar la información transmitida para proteger su confidencialidad y son proporcionados por los propios dispositivos inalámbricos (sólo los usuarios con contraseña pueden conectarse al punto de acceso).	Implementar algún medio de cifrado, siendo el de WPA o WPA2 los más recomendados.
El modo de difusión del ESSID está activado	Una característica de la conectividad inalámbrica es la capacidad de un adaptador de red inalámbrico de una computadora para buscar una red inalámbrica existente de forma automática. Desactivada la característica "Difusión ESSID", la única forma en que un ordenador puede unirse a la red es estableciendo manualmente en el SSID del ordenador el nombre específico de la red.	Desactivar el modo de difusión del ESSID.

C. Mantener Acceso

Por lo general, mantener el acceso se logra mediante la instalación de un programa *backdoor* dentro de las máquinas víctima y atacante. Dichas aplicaciones tienen dos componentes principales: el programa *servidor*, mismo que se instala en el equipo víctima, y el programa cliente, que actúa sobre la computadora del atacante. Por medio de estos programas, el atacante puede ejecutar remotamente en los sistemas infectados las mismas acciones que el administrador o usuario legítimo del equipo.

En esta parte de la prueba no se nos fue permitida la instalación de ninguna aplicación del tipo antes mencionado en ninguno de los dispositivos activos de la red, salvo en un par de computadoras que usan los alumnos para realizar sus prácticas.

Con estas limitaciones, nos dimos a la tarea de probar la aplicación llamada Back Orifice 2000 utilizando como máquina víctima al equipo con dirección IP 192.168.2.5 y como atacante al de IP 192.168.2.4. La aplicación no necesitó ser instalada en ninguna de las dos computadoras, simplemente se configuró el cliente en la máquina atacante y el *servidor* se configuró y ejecutó en la víctima.

Existen diversos *plug-ins* configurables en la aplicación y éstos pueden dividirse en las siguientes categorías:

- De cifrado.
- De autenticación.
- Propias del *servidor*.
- Propias del cliente.
- De comunicación.
- Misceláneos.

No se utilizó ni cifrado ni autenticación en esta prueba, puesto que no se consideraron necesarias. Se incluyeron todos los *plug-ins* relacionados al *servidor* ya que son estos los que permiten realizar la mayoría de acciones para el control remoto del mismo. En éstos se incluyen funciones como son el *keylogging*, la transferencia de archivos, el control de procesos, entre otros. Los *plug-ins* de comunicación son los encargados de especificar el protocolo propio de la capa de transporte del modelo *OSI* y el puerto a utilizar para la comunicación. Para nuestro caso, se utilizó el protocolo *TCP*, el puerto fue el utilizado normalmente por el programa que es el 54320. Finalmente, se encuentran los llamados misceláneos, e incluyen la utilidad BoPeep y LoveBeads. BoPeep permite realizar el secuestro del mouse y del teclado, esto es, controlarlos remotamente desde la máquina atacante además de que permite el despliegue de una pantalla en la que se muestra en tiempo real, la pantalla de la máquina en la que se está ejecutando

el *servidor*. LoveBeads se utiliza para controlar más de una computadora infectada aunque esta utilidad no fue implementada.

BO2K facilita también algunas opciones para que su funcionamiento permanezca lo más indetectable posible para la máquina infectada como, por ejemplo, activarse al iniciar sesión la computadora víctima esto con el fin de mantener el acceso al equipo. Otra opción es la de ocultar el proceso, ya que sin esto, la actividad será detectada por el *Windows Task Manager* de la computadora infectada como bo2k.exe. Con esta opción activada, el proceso simplemente no aparecerá en la ventana de actividad de Windows.

Para realizar la configuración del cliente, se incluyeron todos los *plug-ins*, a excepción de los concernientes al *servidor*; y ningún valor fue modificado.

Una vez que ambos, cliente y *servidor* fueron configurados y que el *servidor* fue ejecutado, simplemente se procedió a conectarse con la máquina víctima desde el cliente tecleando la dirección IP de la otra.

Una vez conectadas ambas, se procedió a probar los *plug-ins* instalados, pudiéndose listar los procesos que corrían bajo la computadora infectada, listar sus archivos, abrirlos y modificarlos, se probó la aplicación de *keylogger*, el secuestro del mouse y del teclado, la visualización del monitor y el desplegado de mensajes.

Una vez finalizadas las pruebas, todos los procesos involucrados fueron detenidos y los archivos borrados.

Vulnerabilidad	Descripción	Recomendaciones
Consideramos que lograr el uso de este programa no puede considerarse una vulnerabilidad, puesto que este fue consentido por el personal del laboratorio, sin embargo, un usuario mal intencionado podría fácilmente haber dejado corriendo el servidor en el equipo víctima.	Un usuario mal intencionado puede dejar corriendo el programa servidor en la máquina víctima.	Una vigilancia más estrecha por parte del personal con respecto a la actividad de los alumnos que hacen uso del laboratorio.

D. Destrucción de la evidencia

En vista de que gracias al programa BO2K se tiene pleno acceso al equipo víctima, se comprobó que el Registro de Eventos de Windows podía ser borrado con esta herramienta, y junto con él, toda la actividad ilegal producto del uso de este programa, sin embargo, esta acción no se nos fue permitida.

Vulnerabilidad	Descripción	Recomendaciones
Es posible borrar el registro de eventos de Windows mediante la herramienta BO2K	Este servicio graba cada actividad que ocurre en el equipo, como por ejemplo, la modificación o eliminación de archivos, la ejecución de aplicaciones, por mencionar algunas.	Más estrecha vigilancia de la actividad de los alumnos que hacen uso del laboratorio.

5.3.4 Resumen final

La posibilidad de recibir ataques informáticos es un problema que afecta a todo tipo de sitios y que los obliga a implementar políticas de seguridad eficientes, capaces de minimizar cualquier riesgo. El desafío es difícil, porque involucra seguridad física, lógica y capacitación en recursos humanos; difícil más no imposible.

A continuación se presenta un pequeño resumen con recomendaciones para minimizar el impacto de un ataque en cada dispositivo examinado.

A. Red interna

- Realizar actualizaciones de las aplicaciones y servicios que corren bajo los equipos que componen la red interna, instalar parches de seguridad con regularidad.

- Se recomienda realizar una vigilancia más estrecha a las actividades de los alumnos que utilizan el equipo de laboratorio así como al *software* que utilizan.

B. Servidor web

- Realizar actualizaciones de las aplicaciones y servicios que corren bajo el *servidor*, instalar parches de seguridad con regularidad.

- No permitir la existencia de cuentas de cualquier servicio que funcionen con los valores por defecto, ya sea para login, contraseña o ambas. Eliminarlas si no se utilizarán más, o modificar dichos valores.

- Tener cuidado respecto a la información que se publica en la página web del laboratorio.

- Realizar escaneos en busca de vulnerabilidades con regularidad.

C. Switch

- Aplicar un protocolo de cifrado al canal de comunicación con el *switch*.

- Cuidar la información que se publica en la página web del laboratorio.

D. Router

- Cifrar la señal propia del laboratorio, ya sea con protocolos tales como *WEP*, *WPA* o *WPA2*, siendo este último el más recomendable, implementar una contraseña robusta y fomentar la cultura entre el personal encargado del laboratorio del cambio de contraseña cada cierto periodo. Si al protocolo de cifrado se le añaden otras medidas de autenticación como por ejemplo el integrar un servidor *Radius*, sería una adición estupenda a la seguridad de la señal inalámbrica.

- Existen otras medidas deseables como por ejemplo, desactivar la difusión del *ESSID* de la señal.