Sociedad Andaluza para el Desarrollo de las Telecomunicaciones S.A.
CONSEJERÍA DE ECONOMÍA, INNOVACIÓN, CIENCIA Y EMPLEO

seguridad+ Y CONFIANZA DIGITAL | AndalucíaCERT CENTRO DE SEGURIDAD TIC

Informe de divulgación

Laboratorio para el análsis de malware (I): Introducción y arquitectura del laboratorio

Tipo de documento:	*Informe*
Autor del documento:	*AndalucíaCERT*
Código del Documento:	*CERT-IF-7478-150704*
Edición:	*0*
Categoría	*Público*
Fecha de elaboración:	*07/04/2015*
N° de Páginas	*1 de 18*

1 TABLA DE CONTENIDOS

2 OBJETIVO

Éste es el primero de una serie de documentos que describen la implementación de un laboratorio para el análisis de malware. Este laboratorio permitirá estudiar una muestra de malware en un entorno controlado y seguro, y que cuente con las herramientas necesarias para obtener gran cantidad de información. Se hablará principalmente de dos enfoques a la hora de realizar el análisis, uno usando herramientas manuales y otro con herramientas automáticas.

3 ALCANCE

Este documento va dirigido al personal de la Junta de Andalucía y al público en general. Pretende aportar las nociones necesarias para entender y tener conocimiento sobre las técnicas que se suelen usar para combatir las amenazas de malware.

4 INTRODUCCIÓN AL ANÁLISIS DE MALWARE

Antes de entrar a describir las herramientas que se pueden usar en el laboratorio es necesario definir qué se considera por software malicioso o malware, y realizar una introducción a los diferentes enfoques que se pueden seguir para después realizar un análisis de malware.

Se entiende por malware o software malicioso aquel software diseñado para llevar a cabo acciones no deseadas, y a menudo dañinas, sin el consentimiento explícito del usuario. El análisis de malware tiene como principal objetivo proporcionar la información necesaria para combatir o enfrentarse a una infección o a una intrusión de red. Proporcionará información sobre qué ha ocurrido exactamente. El análisis de una muestra puede dar información sobre lo que puede hacer ese fichero malicioso, cómo podemos detectar su presencia en la red o en una máquina, y qué medidas podemos aplicar para medir y contener el daño que puede generar.

Una vez que tenemos información de la muestra podremos generar firmas que detecten infecciones en una red de equipos, firmas a nivel de host y firmas de red.

Las firmas de host se usarán para detectar la presencia del malware en un equipo. Se basan principalmente en los cambios que ha realizado el malware en la máquina: ficheros creados o modificados, modificación o creación de nuevas claves de registro, etc. También es posible crear firmas de antivirus basándonos en las características del propio malware, normalmente se busca la presencia de cadenas de bytes en el ejecutable.

Las firmas de red se usan para detectar la presencia del código malicioso monitorizando el tráfico de red. El análisis de malware ayuda a generar firmas de red más fiables, disminuyendo la proporción de falsos positivos.

Normalmente, cuando se realiza un análisis de malware se utiliza una gran variedad de herramientas. Cada una proporcionará alguna información de la pieza de malware analizada. Estas herramientas podrán realizar un proceso automatizado, enfocado al análisis de grandes cantidades de piezas de malware, o un proceso manual, de cara a realizar un análisis más exhaustivo y personalizado de una muestra. En siguientes documentos se proporcionará una relación de herramientas de los dos tipos.

Hay dos enfoques principales a la hora de analizar malware: análisis estático y análisis dinámico.

- **El análisis estático** consiste en examinar la muestra sin ejecutarla en la máquina. Esta técnica analiza las características del fichero, el formato, información que se puede extraer y puede dar una idea de sus funcionalidades sin llegar a ejecutarlo.
- **El análisis dinámico** implica la ejecución de la muestra en la máquina y el análisis de su comportamiento, las acciones que realiza, conexiones de red, ficheros que crea o modifica, ... Esta técnica implica cierto riesgo y debe realizarse siempre después de haber realizado un análisis estático de la muestra, tener indicios de su comportamiento y en un entorno seguro.

Cuando se está realizando un análisis de malware es importante tener claras las categorías más comunes en las que se puede clasificar el malware como pueden ser: virus, troyanos, rootkits, keyloggers, etc. Esto ayudará a acelerar el análisis, permitirá intuir qué está intentando hacer el malware.

5 ARQUITECTURA DEL LABORATORIO

Un laboratorio para análisis de malware es un entorno que dispone de las herramientas y medios necesarios para realizar investigaciones, análisis, experimentos, etc, de forma controlada.

Este entorno permite asegurar que no se producen influencias no previstas que alteren el resultado de la investigación, es decir, existe un control del entorno y esto proporciona una mayor fiabilidad en los datos obtenidos. También permite repetir un experimento o medición y comprobar que se obtiene siempre el mismo resultado, de esta manera se normaliza la situación.

Un entorno seguro y controlado permitirá investigar el malware sin exponer tu propia máquina u otras máquinas de la red a un riesgo innecesario. Se podrá usar una máquina física dedicada o máquinas virtuales para estudiar el malware de forma segura.

Lo más común es usar máquinas virtuales para el análisis de malware. En las máquinas físicas suele ser más complicado limpiar completamente la instalación de un malware. Aunque se pueden usar herramientas de restauración de discos, la restauración a un estado anterior con snapshot suele facilitar bastante la tarea. La mayor desventaja que pueden presentar las máquinas virtuales es que, en algunas ocasiones, el malware pueda detectar que se está ejecutando en una máquina virtual, y comportarse de forma diferente que si estuviera ejecutándose en una máquina física.

Existen diferentes arquitecturas que se pueden utilizar con las máquinas virtuales:

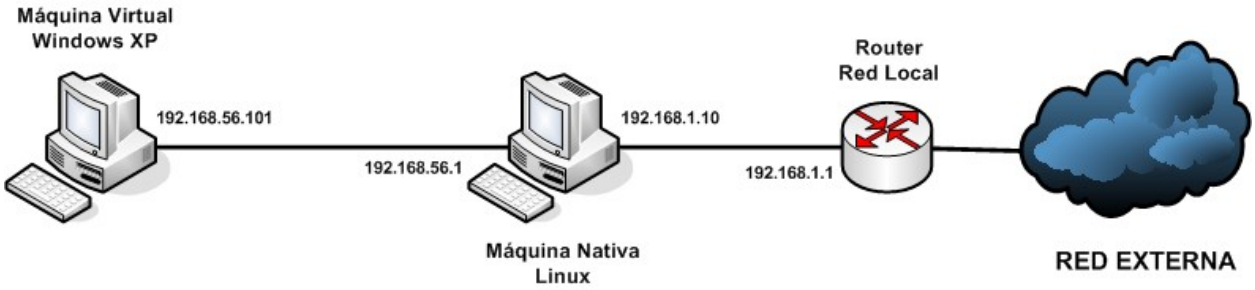
- **Sin acceso a Internet:** Permite ejecutar malware si poner en riesgo otras máquinas del entorno. Una desventaja de esta opción es que muchas piezas de malware dependen de una conexión a Internet para establecer, por ejemplo, conexión con el servidor de control. Sin conectividad de red no se podrá analizar la actividad de red maliciosa.
- **Acceso a servicios falsos:** Se pueden proporcionar servicios de red falsos, de esta manera al ejecutar el malware, éste puede pensar que está conectando con Internet cuando en realidad no es así, y mediante estos servicios monitorizar todas las peticiones que realice. Se puede, por ejemplo, implementar un servidor dns falso (con ApateDNS) y recoger las peticiones de resolución que realice la máquina virtual.
- **Acceso a Internet:** En esta opción se proporcionará acceso a Internet a la máquina virtual en la que se ejecuta el malware. Es recomendable tener controladas las conexiones que realiza la máquina virtual para evitar que pueda infectar o provocar daño a las máquinas de la red local.

Nuestra recomendación, por conllevar un menor riesgo, es no proporcionar acceso a Internet o dar acceso a servicios falsos.

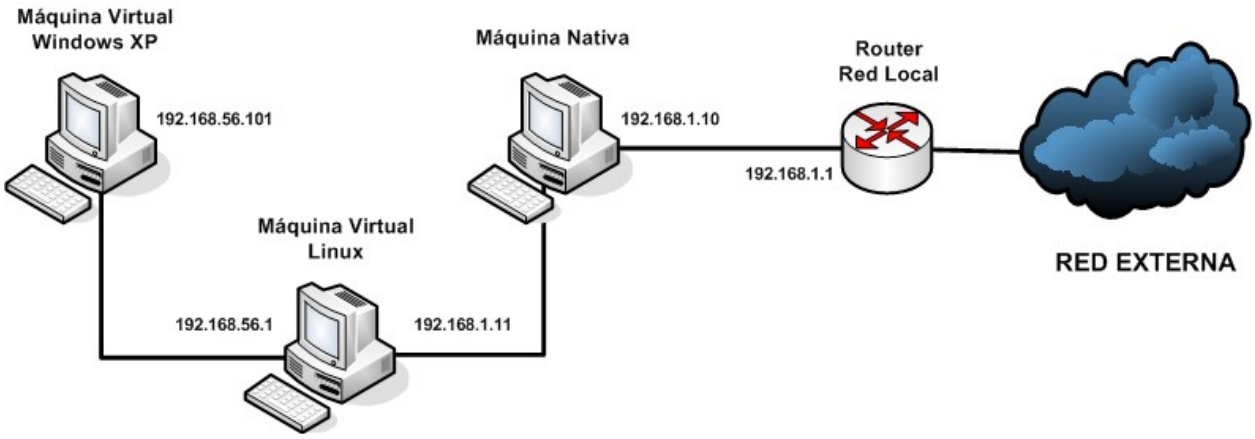
Es este documento se describirá la configuración de un laboratorio en el que las muestras se ejecutarán en máquinas virtuales, y éstas contarán con acceso a Internet. El acceso a Internet de las máquinas virtuales se proporcionará a través de una interfaz en modo sólo-anfitrión en la máquina nativa, que habrá que agregar previamente en VirtualBox. A través de esta interfaz se encaminarán las peticiones hasta la red local e Internet.

Este método permite tener un control del tráfico que genera la máquina virtual y filtrarlo mediante reglas de firewall en la máquina nativa, asegurándonos de esta manera que la máquina virtual no tiene acceso a las máquinas de la red local y protegiéndolas de las acciones del malware.

Se muestra un diagrama de red del escenario descrito:



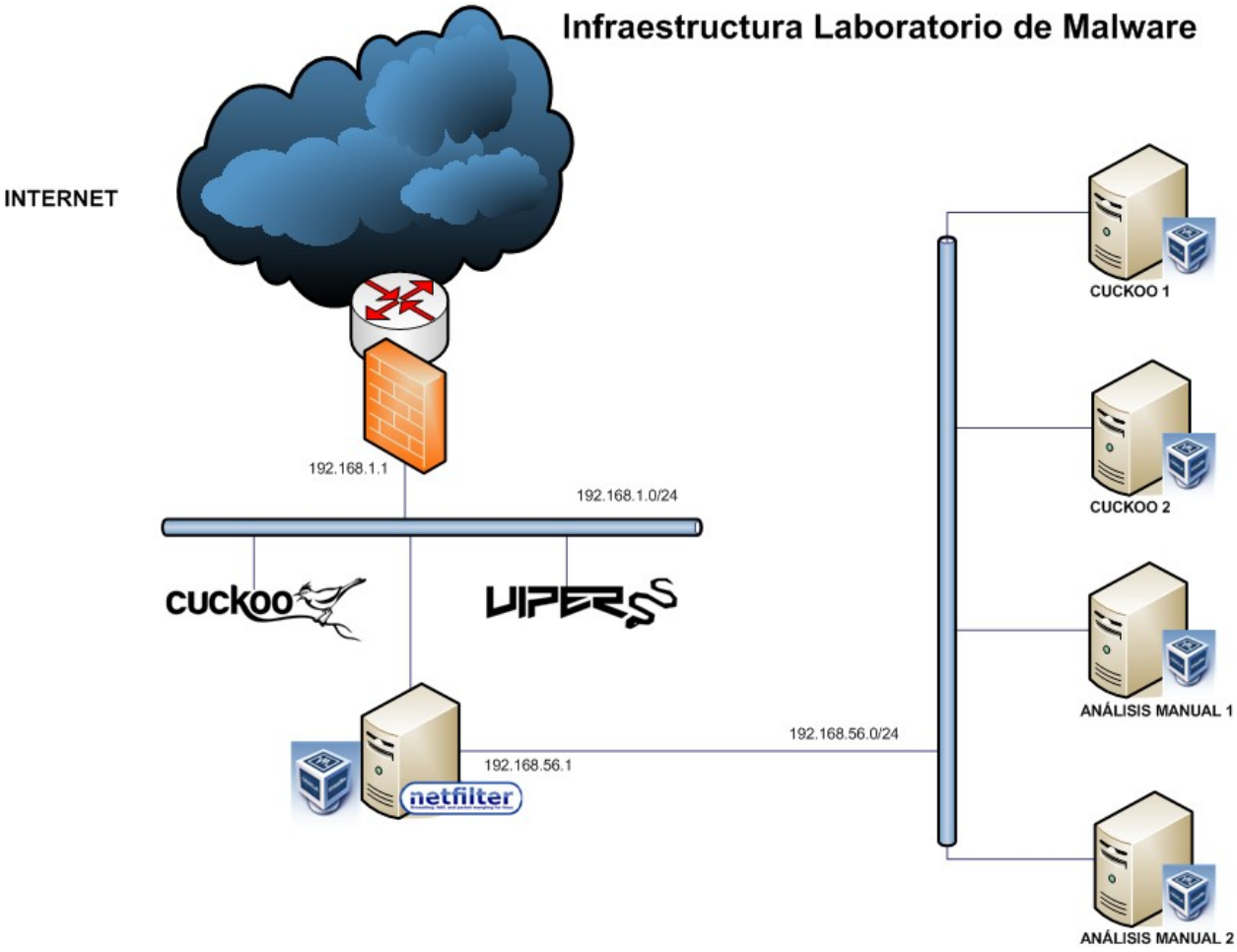
Otra forma de dar acceso a la máquina virtual Windows a Internet sería a través de otra máquina virtual que hiciera las funciones de router entre la red interna de máquinas virtuales y la red local. Esta máquina debería de tener dos interfaces de red, una en cada red, y activar, igual que en escenario anterior, el reenvío de paquetes. Con este método las conexiones de la máquina Windows de podrían filtrar y controlar en la máquina virtual Linux.



El software de virtualización que se usará es VirtualBox, y se puede descargar de la siguiente url:

- https://www.virtualbox.org/

En nuestro caso el escenario demo que se describirá será el siguiente:

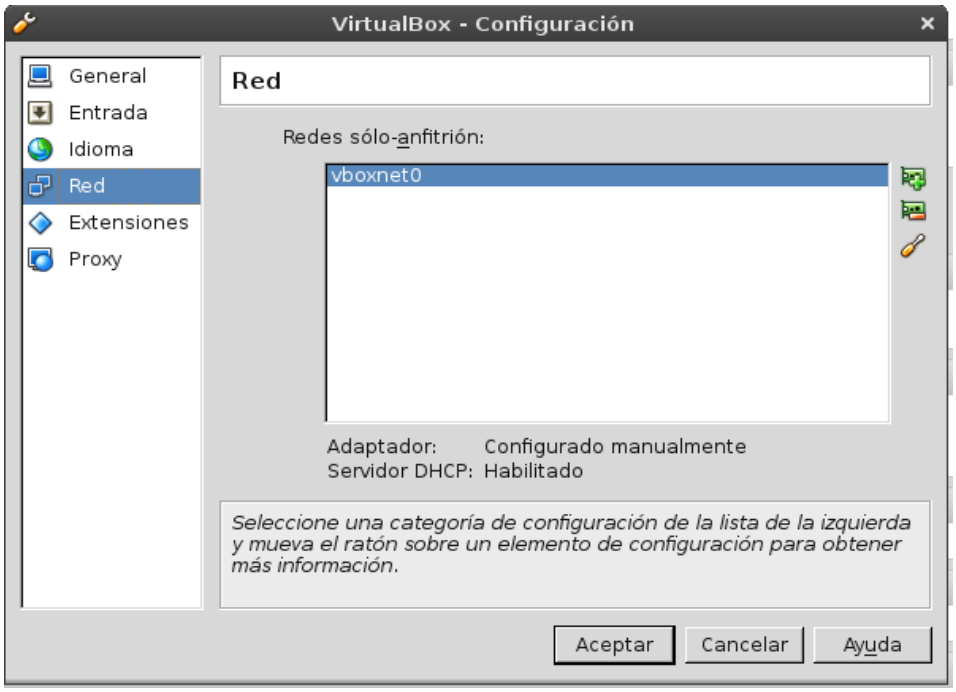


Tendremos una red de laboratorio interna en la que podremos alojar distintas máquinas virtuales para realizar análisis manuales, que tendrán instaladas las herramientas que necesitemos para ello, y máquinas para realizar análisis automáticos, usadas por la sandbox.

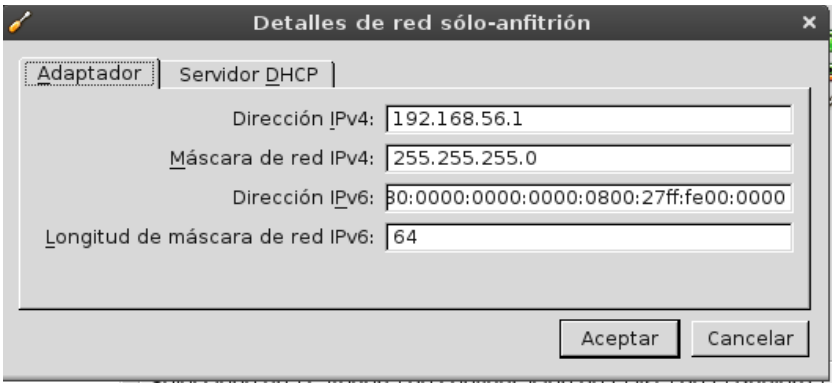
Una vez instalado hay que preparar la máquina virtual de Windows XP. Una vez creada e instalado el sistema operativo se procederá con la configuración de red y de seguridad necesaria, la cual se describe en los siguientes puntos.

5.1 CONFIGURACIÓN MÁQUINA WINDOWS XP

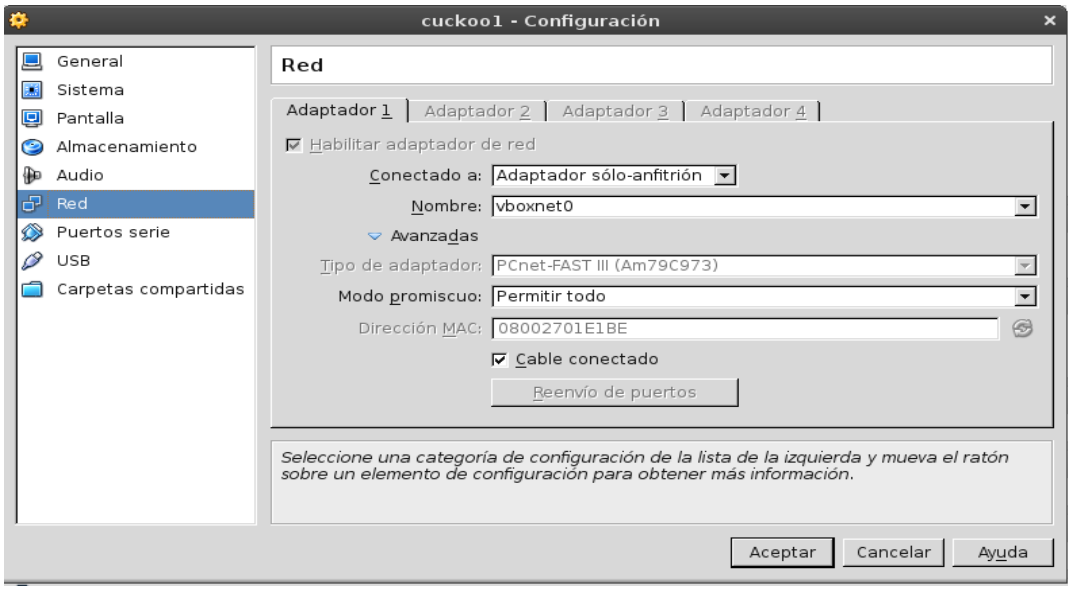
Para configurar la red en VirtualBox en la máquina Windows tendremos que crear primero una interfaz de red de tipo "Adaptador sólo-anfitrión". Pulsamos en el menú superior de VirtualBox: "Archivo → Preferencias → Red" y después en el icono arriba a la derecha con el signo "+".



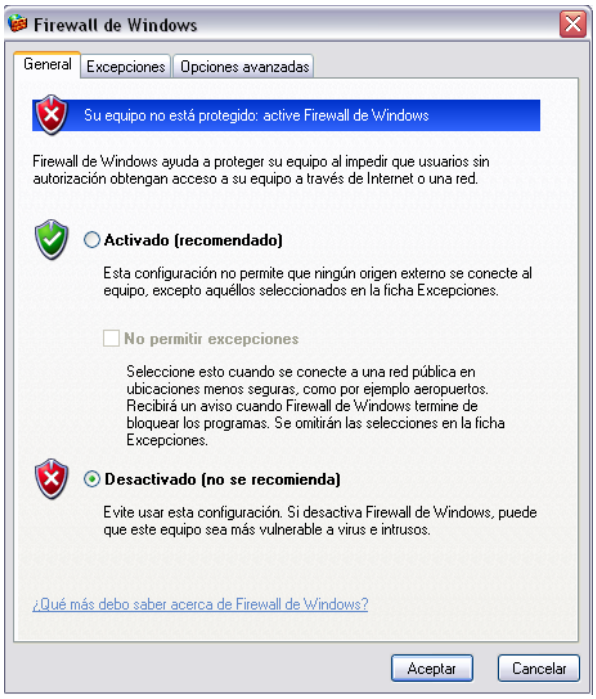
Le asignamos una dirección IP a la interfaz:

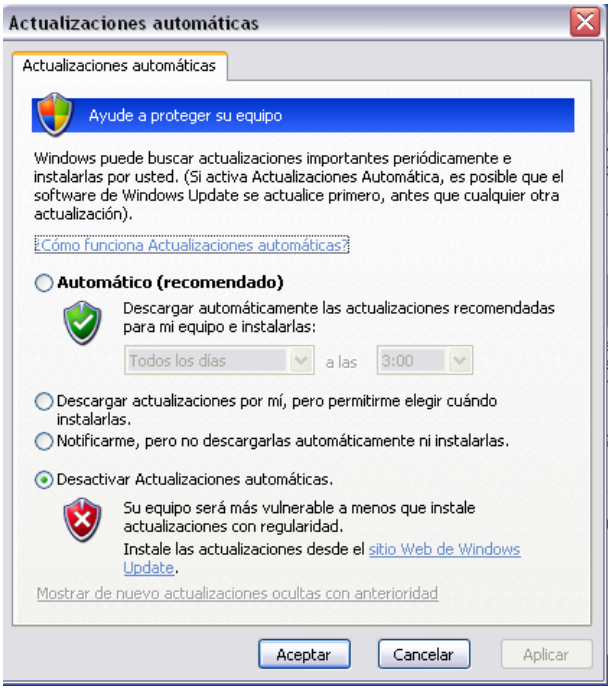


Ahora nos vamos a la configuración de red de la máquina virtual Windows, pulsamos el botón derecho sobre la máquina virtual → Configuración → Red. Asignamos el "Adaptador sólo-anfitrión" y la interfaz que hemos creado "vboxnet0". El modo promiscuo lo asignamos a "Permitir todo".

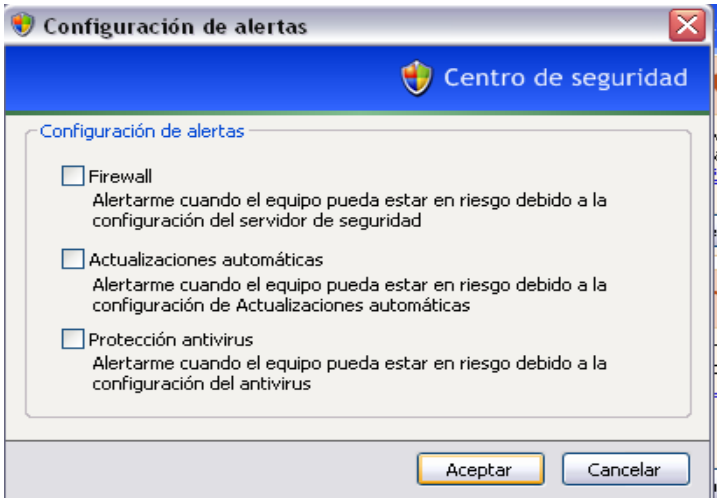


Se desactivarán las actualizaciones automáticas y el firewall de Windows XP. Nos vamos al "Panel de Control" → "Centro de Seguridad":

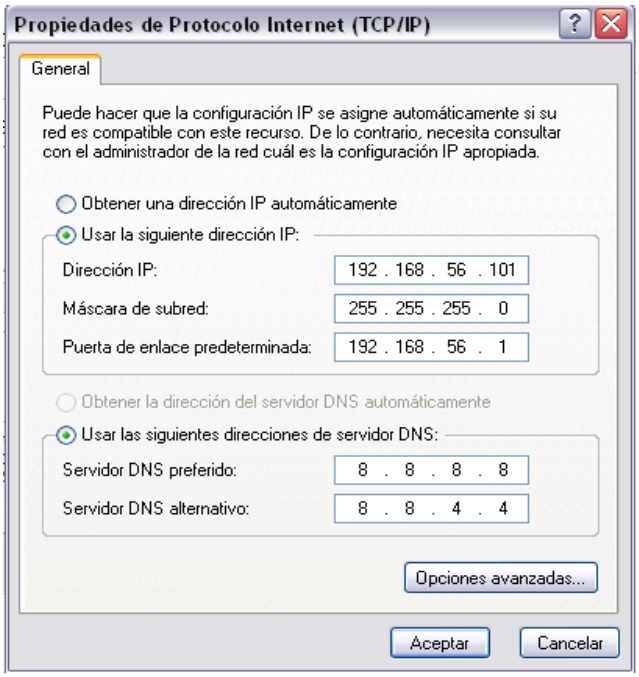




Dejamos desactivadas las alertas del Centro de Seguridad en "Cambiar la forma en que el Centro de Seguridad me alerta" en las opciones de la izquierda del "Centro de Seguridad":



Por último, configuraremos la red de la máquina virtual para que las conexiones pasen a través de la máquina nativa Linux.



5.2 CONFIGURACIÓN MÁQUINA NATIVA LINUX

La máquina Linux nativa tendrá que tener como mínimo dos interfaces de red, vboxnet0 conectada a la red de las máquinas virtuales, y la interfaz que esté conectada a la red local, en nuestro caso eth0.

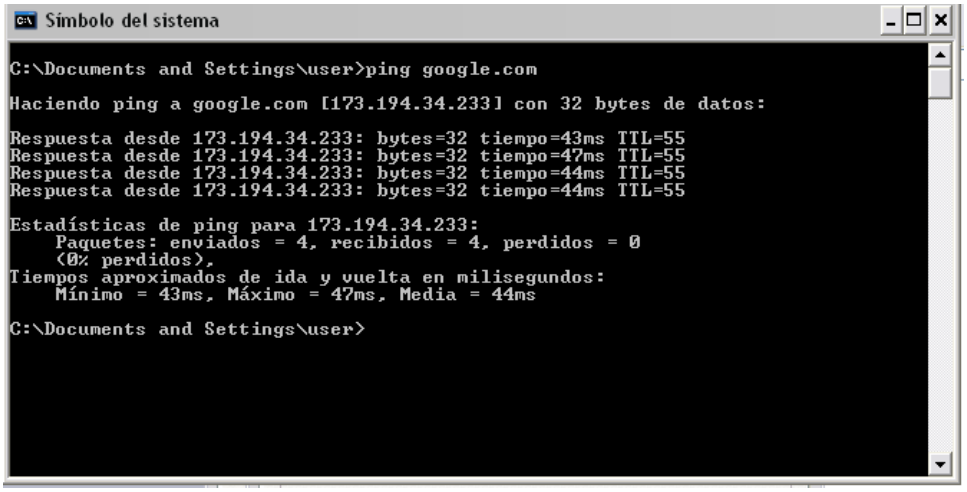
El objetivo es permitir el acceso a Internet a las máquinas Windows en las que se ejecutarán las muestras, pero no permitir conectar a la red local. De esta manera se evita que se puedan propagar infecciones a la red local.

Antes de ejecutar cualquier muestra en el laboratorio es muy importante tener claro este punto y comprobar que efectivamente no hay acceso a la red local desde las máquinas virtuales. También se cortará el acceso desde la red de laboratorio a la interfaz vboxnet0 para evitar ser objetivo de ataque desde las máquinas infectadas del laboratorio.

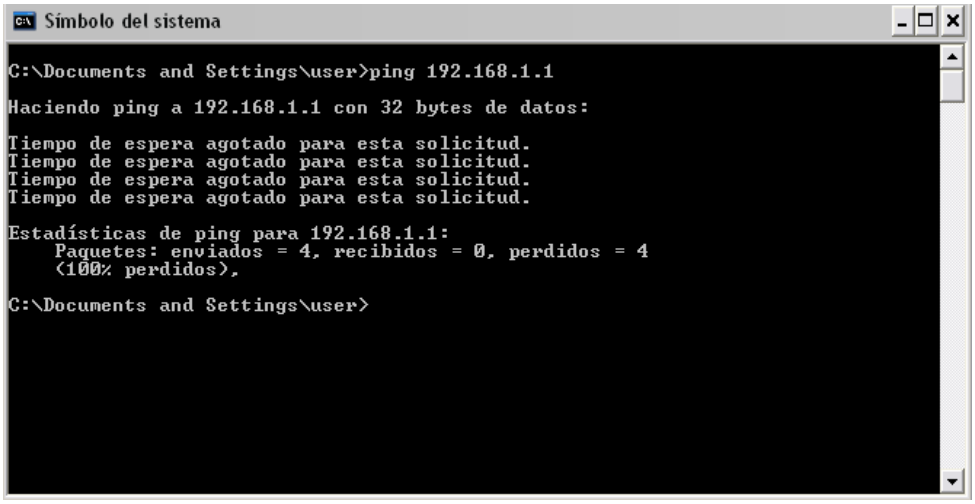
```
# ifconfig
eth0      Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:xx
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:16

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:696 errors:0 dropped:0 overruns:0 frame:0
          TX packets:696 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:52798 (51.5 KiB)  TX bytes:52798 (51.5 KiB)

vboxnet0  Link encap:Ethernet  HWaddr xx:xx:xx:xx:xx:xx
          inet addr:192.168.56.1  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::800:27ff:fe00:0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:3680 (3.5 KiB)
```

Para permitir que las peticiones de la máquina Windows pasen a través de la máquina Linux es necesario activar reenvío de paquetes (ip_forward) y crear algunas reglas en el firewall de Linux iptables.

Comprobamos que desde la máquina Windows XP tenemos conexión a Internet haciendo un ping a google.com:



Se comprueba que no hay conectividad con la red local desde la máquina Windows XP:

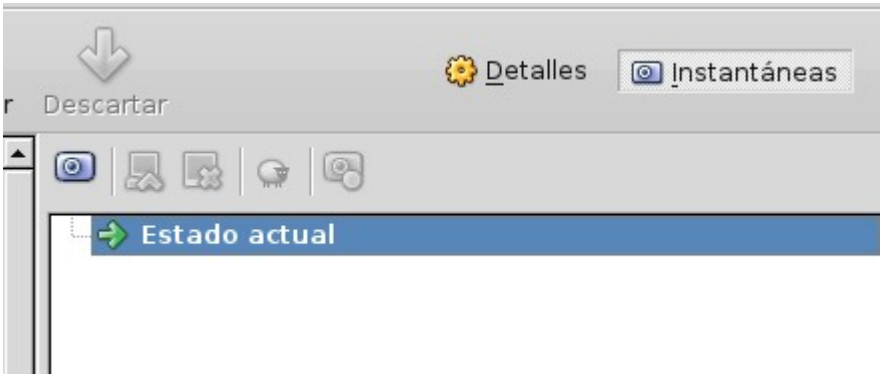


5.3 SNAPSHOTS

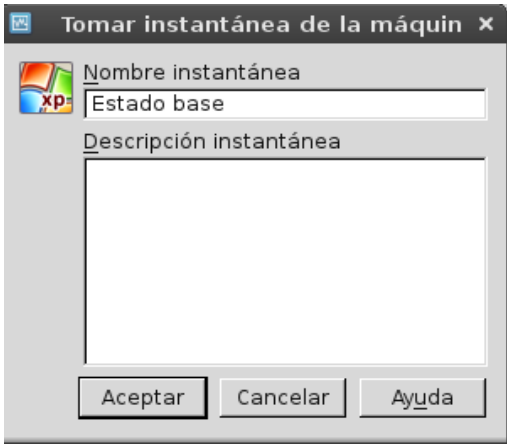
En las máquinas virtuales es posible realizar snapshots. Esto permite salvar el estado de la máquina virtual y volver a él posteriormente descartando los cambios producidos hasta ese momento. Esta característica propia de las máquinas virtuales resulta muy útil para el análisis de malware para deshacer la instalación de un malware fácilmente.

Lo primero que se hará cuando se tenga configurada completamente la máquina Windows XP es tomar un snapshot. Este snapshot se tomará como estado base. En el caso de encontrarnos a mitad de una investigación es posible hacer snapshot adicionales y volver a cualquiera de los puntos guardados.

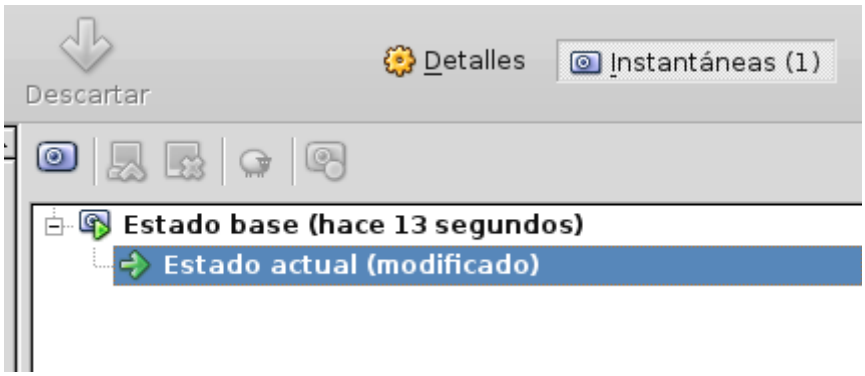
Para hacer un snapshot tendremos que pulsar sobre la máquina virtual → Pulsar en "Instantáneas" arriba a la derecha → Pulsar en "Tomar instantánea" en la imagen de la cámara de fotos.



Le asignamos un nombre al snapshot y pulsamos "Aceptar":



Veremos que se crea un nuevo snapshot con el nombre indicado:



Si quisiéramos volver al estado anterior sólo tendremos que pulsar sobre el snapshot "Estado base" y elegir la opción "Recuperar instantánea".

5.4 COMPARTICIÓN DE FICHEROS

La transferencia de ficheros hacia o desde la máquina virtual se pueden realizar creando una carpeta de red compartida. Creamos una carpeta en el Escritorio de la máquina Windows XP, por ejemplo con el nombre "Compartida". Pulsamos botón derecho sobre la carpeta → Propiedades → Compartir → "Compartir esta carpeta en red".

En la máquina nativa, si exploramos la red, veremos el contenido de la carpeta compartida desde Windows y también podremos escribir en ella.

6 REPOSITORIO DE MALWARE

Para almacenar muestras únicas de malware en un repositorio centralizado usaremos viper. Viper puede usarse en consola, ya que cuenta con un intérprete de comandos, o a través de una interfaz web:

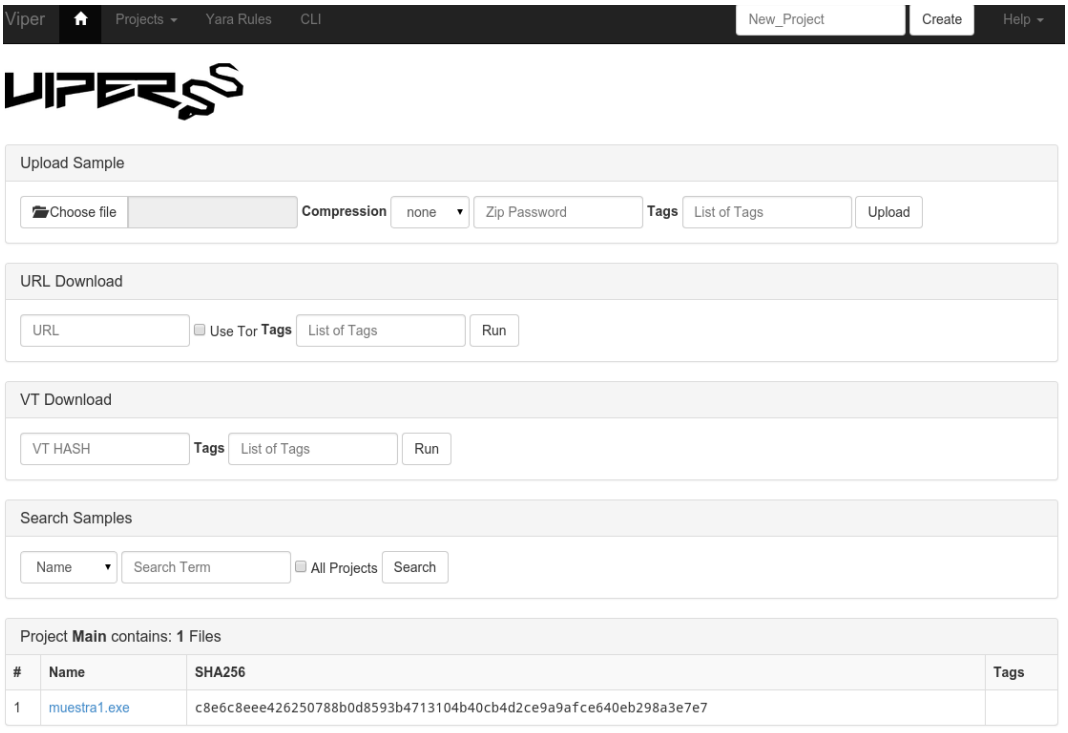
```
$ python /root/tools/viper/viper.py
        _
       (_)
  _   _ _ ____  _____  ____
 | | | | |  _ \| ___ |/ ___)
  \ V /| | |_| | ____| |
   \_/ |_|  __/|_____)_| v1.2
         |_|

You have 68 files in your default repository
viper > help
Commands:
+------------+-------------------------------------------------+
| Command    | Description                                     |
+------------+-------------------------------------------------+
| clear      | Clear the console                               |
| close      | Close the current session                       |
| delete     | Delete the opened file                          |
| exit, quit | Exit Viper                                      |
| export     | Export the current session to file or zip       |
| find       | Find a file                                     |
| help       | Show this help message                          |
| info       | Show information on the opened file             |
| notes      | View, add and edit notes on the opened file     |
| open       | Open a file                                     |
| projects   | List or switch existing projects                |
| sessions   | List or switch sessions                         |
| store      | Store the opened file to the local repository   |
| tags       | Modify tags of the opened file                  |
+------------+-------------------------------------------------+

Modules:
+--------------+------------------------------------------------+
| Command      | Description                                    |
+--------------+------------------------------------------------+
| apk          | Parse Android Applications                     |
| clamav       | Scan file from local ClamAV daemon             |
| cuckoo       | Submit the file to Cuckoo Sandbox              |
| debup        | Parse McAfee BUP Files                         |
| editdistance | Edit distance on the filenames                 |
| elf          | Extract information from ELF headers           |
| email        | Parse eml and msg email files                  |
| exif         | Extract Exif MetaData                          |
| fuzzy        | Search for similar files through fuzzy hashing |
| html         | Parse html files and extract content           |
| ida          | Start IDA Pro                                  |
| idx          | Parse Java idx files                           |
| image        | Perform analysis on images                     |
| jar          | Parse Java JAR archives                        |
| office       | Office Document Parser                         |
| pdf          | Parse and analyze PDF documents                |
| pe           | Extract information from PE32 headers          |
| r2           | Start Radare2                                  |
| rat          | Extract information from known RAT families    |
| reports      | Online Sandboxes Reports                       |
| shellcode    | Search for known shellcode patterns            |
| strings      | Extract strings from file                      |
| swf          | Parse, analyze and decompress Flash objects    |
| virustotal   | Lookup the file on VirusTotal                  |
| xor          | Search for xor Strings                         |
| yara         | Run Yara scan                                  |
+--------------+------------------------------------------------+
viper >
```

Además del almacenamiento de muestras, viper cuenta con muchas herramientas que pueden ayudar durante el análisis.

Se puede arrancar la interfaz web con el siguiente comando:

```
# python /root/tools/viper/web.py -p 80
Bottle v0.12.8 server starting up (using WSGIRefServer())...
Listening on http://127.0.0.1:80/
Hit Ctrl-C to quit.
```

Se puede acceder a través de un navegador web:



7 CONCLUSIONES

Como se ha hablado a lo largo del documento, el análisis de malware nos puede proporcionar la información necesaria para detectar infecciones en equipos y en la red, determinar el impacto que pueden ocasionar y las acciones que realiza, así como tomar medidas que puedan evitar que se produzcan incidentes de seguridad o reducir su impacto en caso de producirse.

Aunque hay que tener en cuenta que la ejecución de muestras de malware suponen un riesgo para el equipo en el que se ejecutan y para la red. Por ello debemos de tomar la precauciones necesarias para contar con un entorno que proporcione seguridad y protección. También es importante que el entorno permita revertir las modificaciones que realiza el malware analizado de manera sencilla, y nos permita sacar conclusiones fácilmente, es decir, se trate se un entorno controlado y fiable, flexible y que cuente con las herramientas necesarias para obtener información que nos aporte algún valor.

Se describen los enfoques que se puede seguir para realizar análisis de malware. Por una parte tenemos el análisis estático, mediante el cual se obtiene información del fichero sin ejecutarlo, y por otra el análisis dinámico, que obtiene información ejecutando la muestra y analiza su comportamiento. También se distinguen, en base al proceso que se sigue para la obtención de información, en análisis manual y análisis automático.

Por último se plantea la arquitectura del laboratorio que se implementará y se describen los pasos necesarios para ponerla en marcha.

8 DOCUMENTACIÓN DE REFERENCIA

- Cuckoo Sandbox: http://cuckoosandbox.org/
- Viper: http://viper.li/
- Virtualbox: https://www.virtualbox.org/