ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

"PROPUESTA DE IMPLEMENTACIÓN DE UN AMBIENTE DINÁMICO DE MALWARE BASADO EN CUCKOO SANDBOX PARA LA RED LOCAL DEL EDIFICIO DE LA FIE-ESPOCH"

Trabajo de titulación presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

AUTOR: JONATHAN FERNANDO QUEZADA HARO

TUTOR: ING. EDWIN ALTAMIRANO

Riobamba – Ecuador

2017

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN ELECTRÓNICA, TELECOMUNICACIONES Y REDES

El Tribunal del Trabajo de Titulación certifica que el proyecto técnico: PROPUESTA DE IMPLEMENTACIÓN DE UN AMBIENTE DINÁMICO DE MALWARE BASADO EN CUCKOO SANDBOX PARA LA RED LOCAL DEL EDIFICIO DE LA FIE-ESPOCH, de responsabilidad del señor Jonathan Fernando Quezada, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	**FIRMA**	**FECHA**
Ing. Washington Luna **DECANO FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**	_____	_____
Ing. Franklin Moreno **DIRECTOR DE ESCUELA DE INGENIERÍA ELECTRÓNICA TELECOMUNICACIONES Y REDES**	_____	_____
Ing. Edwin Altamirano **DIRECTOR TRABAJO DE TITULACIÓN**	_____	_____
Ing. Vinicio Ramos **MIEMBRO DEL TRIBUNAL**	_____	_____

Yo, Jonathan Fernando Quezada Haro declaro ser el autor del presente trabajo de titulación: "PROPUESTA DE IMPLEMENTACIÓN DE UN AMBIENTE DINÁMICO DE MALWARE BASADO EN CUCKOO SANDBOX PARA LA RED LOCAL DEL EDIFICIO DE LA FIE-ESPOCH", que fue elaborada en su totalidad por mí persona, bajo la dirección del Ingeniero Edwin Altamirano, haciéndome totalmente responsable por las ideas, criterios, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio de la misma pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.

Jonathan Fernando Quezada Haro

DEDICATORIA

A Dios por darme un día más de vida y brindarme las fuerzas necesarias para seguir adelante. A mi madre Angélica que ha sido mi apoyo incondicional para alcanzar este logro académico. A mi hermano Alexander que siempre ha estado pendiente de mi desempeño académico. A mi novia Mayra quien me da fortaleza para seguir adelante con nuestro proyecto de vida.

Jonathan

AGRADECIMIENTO

Agradezco a Dios y a mis padres por brindarme esa oportunidad de existir en este mundo.

A mi hermano y a mi familia por ser quienes siempre me alientan a seguir adelante.

A la Escuela Superior Politécnica de Chimborazo por brindarme la oportunidad de formarme como profesional.

De manera especial quiero agradecer al Ing. Edwin Altamirano, al Ing. Msc. Vinicio Ramos V. y al Ing. Luis Alberto Pazmiño por sus inestimables aportes en el desarrollo y culminación de mi carrera estudiantil. Sin su apoyo, hubiera sido más difícil llegar hasta donde estoy.

Y primordialmente a mi madre que siempre se ha preocupado por mí, me ha inculcado valores que no he olvidado y por ser el apoyo incondicional en todos los momentos de mi vida.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE GRÁFICOS

ÍNDICE DE ANEXOS

RESUMEN

Para el presente trabajo de titulación se diseñó e implementó un ambiente dinámico de detección de malware basado en Cuckoo Sandbox para la red local del edificio de la Facultad de Informática y Electrónica de la Escuela Superior Politécnica de Chimborazo, donde se utilizó el método Inductivo-Deductivo para la recolección y análisis de la información. Se realizó el análisis, diseño, implementación, ejecución y verificación de los resultados que muestra la herramienta Open Source "Cuckoo Sandbox" la cual permite estudiar el comportamiento del malware. Para la simulación de ataques informáticos se siguió la metodología Ec-Council Ethical Hacker, Scanning, Identificación de Vulnerabilidades, Penetración al Sistema. Para completar estos pasos, se utilizó el sistema operativo Kali Linux 2016.2, la cual está basada en Debian GNU/Linux diseñada principalmente para la auditoría y seguridad informática y cuenta con herramientas que permiten vulnerar servidores, de la misma manera que lo haría un posible atacante informático. La caja de arena cuckoo ofrece la posibilidad de conectarse cada acción realizada por el malware en la máquina virtual, para observar cual es el comportamiento y el objetivo del malware una vez que ingresa en la red. Al implementar el sistema automatizado Cuckoo Sandbox se analizó el comportamiento del malware en la red, se concluyó que implementando los mecanismos de seguridad se garantiza un nivel aceptable de seguridad en la transmisión de los datos por la red local y se recomienda ejecutar las políticas de seguridad personalizadas para la red de datos del Edificio de la FIE de la Escuela Superior Politécnica de Chimborazo.

PALABRAS CLAVES: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <REDES DE COMPUTADORES>, <SEGURIDAD INFORMÁTICA>, <MALWARE>, <CUCKOO SANDBOX (SOFTWARE)>, <KALI LINUX 2016.2>, <ATAQUES INFORMÁTICOS>, <POLÍTICAS DE SEGURIDAD>

SUMMARY

A dynamic environment System of malware detection based on Cuckoo Sandbox was designed and developed for the local network of the Building of Informatics and Electronics Faculty of Escuela Superior Politécnica de Chimborazo, where the Inductive-Deductive method was used for the collection and analysis of information. The analysis, design, implementation, execution and verification of results were made, which showed that Open Source tool "Cuckoo Sandbox" allows to study the malware behavior. The methodology Ec-Council Ethical Hacker, Scanning, Identification of Vulnerabilities, and Penetration to the System for the simulation of computer attacks followed. The Kali Linux 2016.2 operating system completed these steps, which is based on Debian GNU/Linux designed mainly for auditing and computer security and has tools that allow attacking servers, in the same way as a possible attacker computer science would do it. The cuckoo sandbox offers the possibility of connecting every action taken by the malware in the virtual machine, to observe the behavior and purpose of the malware once it enters the network. The automated system Cuckoo Sandbox analyzed the behavior of malware in the network, it conclude that implementing security mechanisms ensures an acceptable level of security in the transition of data over the local network and it recommended to implemented security polices customized for the data network of the building of Informatics and Electronics Faculty of Escuela Superior Politécnica de Chimborazo.

KEY WORDS: <TECHNOLOGY AND ENGINEERING SCIENCES>, <COMPUTER NETWORKS>, <COMPUTER SECURITY>, <MALWARE>, <CUCKOO SANDBOX (SOFTWARE)>, <KALI LINUX 2016.2>, <COMPUTER ATTACKS>, <SECURITY POLICES>

INTRODUCCIÓN

En la vida siempre ha existido la evolución, lo cual nos hace pensar que la seguridad informática es una de las más vulneradas en la actualidad con la automatización del malware, y la mayor motivación de un agente informático es la necesidad por estar siempre un paso delante de los hackers no éticos. Se ha creado organizaciones de seguridad en redes para establecer comunidades formales de profesionales de la seguridad en redes. Es importante que los profesionales de seguridad informática estén siempre al tanto de las nuevas actualizaciones y parches de los sistemas operativos. (Sanchez, 2010)

Si no sabes contra que estas combatiendo, no sabes cómo enfrentarlo, para analizar nuevos vectores de ataque y para saber que está pasando, se implementó un sistema o área de pruebas que permite analizar cualquier tipo de malware, se ejecuta la muestra en un sistema virtualizado empleando APIS que monitorizan los resultados, para lograrlo se procedió a investigar sobre la automatización del malware con un propósito y fin específico. Es verdad que la seguridad no se la posee en su totalidad, pero si se puede tener en su mayoría, ya que para una persona muy capas siempre habrá otra que este por arriba de ella.

Para lograr el objetivo planteado se utilizó la metodología inductiva–deductiva para la recolección de información, diseño e implementación de un recinto de seguridad para la evaluación de los resultados. El objetivo de este proyecto está dirigido al diseño e implementación de un sistema de análisis de comportamiento del malware y es un aporte académico dirigido a todos los estudiantes que se interesen en la confidencialidad de los datos ya que cuando la información es privada siempre existe la preocupación de que alguien pueda modificarla o robarla para utilizarla según sus intereses. (Ing. Felipe Pérez Roque, 2013)

ANTECEDENTES

En los últimos años y conforme la evolución de las tecnologías, los mecanismos utilizados para la evasión y transmisión de código malicioso por parte de los desarrolladores de malware se han perfeccionado en las últimas décadas. Ante el dinamismo en el cambio de las redes, se crean nuevas metodologías para evitar la seguridad utilizando conceptos como amenazas día Zero o algoritmos DGAs, donde nuevos métodos avanzados han sido desarrollados para eludir la detección de malware siendo esto uno de los principales problemas para los sistemas de seguridad de primera generación existentes como antivirus, firewall, antispams entre otros. (Ávila, 2012)

El concepto de "Sandbox" es utilizado en informática cuando se trata de temas relacionados con la seguridad y casi siempre se refiere a una zona restringida, en la que los elementos que se ejecutan en el entorno, se encuentran aislados de los recursos sensibles del sistema y con acceso restringido a funciones críticas. (Informática, 2014)

En este sentido existen varias herramientas de sandboxing para diferentes plataformas, como por ejemplo Windows o Linux. Para el caso en estudio corresponde estudiar a Cuckoo Sandbox que es el framework open source más utilizado para analizar y descubrir el funcionamiento de amenazas de todo tipo en un entorno controlado. (Informática, 2014)

A más de ser un sistema centralizado, donde una máquina se encarga de ejecutar los componentes "core" del sistema y por otro lado, hay máquinas virtuales aisladas que permiten la ejecución de los programas que deben ser analizados. La máquina donde corre Cuckoo se encarga de gestionar el estado de cada una de las máquinas virtuales definidas en el fichero de configuración de la herramienta y se encarga, entre otras cosas, de iniciar, detener y enviar muestras de Malware a las máquinas virtuales especificadas. (Informática, 2014)

De las investigaciones realizadas sobre antimalware la de mayor eficiencia y efectividad es Malwarebyts, detecta y elimina el malware en tiempo real con tecnología avanzada anti-malware, anti-spyware y anti-rootkit. Busca las amenazas más recientes y peligrosas automáticamente para que usted esté protegido sin tener que hacer nada. Por esta razón se propone un análisis de malware basado en una tecnología open Source. (Malwarebytes, 2017)

FORMULACIÓN DEL PROBLEMA

¿Cómo analizar el tráfico de malware utilizando el software Cuckoo Sandbox y establecer políticas de seguridad?

SISTEMATIZACIÓN DEL PROBLEMA

¿Será posible mejorar el uso del internet con la utilización del software libre?
¿Cómo analizar el tráfico de red con la implementación del software?
¿Cómo analizar los paquetes utilizando la herramienta Cuckoo-Sandbox?
¿Cómo controlar el comportamiento de los procesos maliciosos mientras se ejecuta en un entorno aislado sin interferir con los demás procesos dentro de la red?

JUSTIFICACIÓN DEL TRABAJO DE TITULACIÓN

JUSTIFICACIÓN TEÓRICA

En la Escuela Superior Politécnica de Chimborazo y en particular en la Escuela de Ingeniería de Electrónica Telecomunicaciones y Redes de la Facultad de Informática y Electrónica, se propone realizar un análisis del comportamiento del malware en la red que brinda a los estudiantes y docentes el servicio de internet.

- **Malware**: es la abreviatura de "Malicious software" (software malicioso), término que engloba a todo tipo de programa o código de computadora cuya función es dañar un sistema o causar un mal funcionamiento. Dentro de este grupo se encontró términos como: Virus, Trojan (Caballo de Troya), Gusano (Worm), Dialers, Spyware, Adware, Rootkits, Hijackers, Keyloggers, Rogues, entre otros. (Rivero, 2017)

- **Virus**: son sencillamente programas creados para infectar sistemas y otros programas creándoles modificaciones y daños que hacen que estos funcionen incorrectamente y así interferir en el funcionamiento general del equipo, registrar, dañar o eliminar datos, o bien propagarse por otros equipos y a través de Internet. (Galas, 2015)

- Es decir, la diferencia entre un malware y un virus es sencillamente que un malware es algo más genérico, es un término que engloba a los virus, pero puede representar muchas otras cosas peligrosas. (Galas, 2015)

- Por tanto, una vez estudiados estos conceptos, un antimalware y un antivirus no son lo mismo ya que el primero está orientado a "todo tipo de malware" (tradicionalmente a todos menos los virus ya que para eso estaban los antivirus, aunque eso está cambiando y los antimalware ya empiezan a detectar "casi todo") y los antivirus están orientados a los virus en todas sus formas, pero no se especializan tanto en el resto del malware (como Spyware y Adware por poner un ejemplo). (Galas, 2005)

- Como futuros ingenieros Electrónicos en Telecomunicaciones esta problemática sobre el mal uso de tecnologías locales involucra directamente la participación en el análisis diagnóstico y solución de los problemas que conciernen a la seguridad en las redes de datos.

- Una vez finalizado el proyecto se pretende mejorar el nivel de seguridad con políticas de seguridad establecidas, comprobando su funcionalidad y cualificando su operación, para de esta manera establecer su impacto a corto y largo plazo.

JUSTIFICACIÓN APLICATIVA

En la red de la Facultad de Informática y Electrónica se implementará un software que analizará el comportamiento del malware, que permitirá un tráfico seguro para los usuarios, manteniendo la integridad en los datos y seguridad en la red.

Por lo tanto, se hace necesario hacer un estudio en el cual se dará a conocer los tipos de malware que afectan el normal funcionamiento de la red, así como la integridad en sus datos.

La adaptabilidad con los laboratorios de la FIE es uno de los recursos fundamentales para el trabajo de investigación a realizarse y de esta forma poder dar una solución inmediata cuando se presente algún inconveniente de tipo malicioso en la red.

Se utilizó la distribución de GNU/Linux como la plataforma para la configuración de este análisis, solución y disminución de los problemas que se encuentren en la red de datos y prevenir ataques que pongan en peligro la integridad de los datos, siendo una herramienta de software libre los gastos que involucran el proyecto no son altos.

OBJETIVOS

OBJETIVOS GENERALES

- Implementar un ambiente dinámico de malware basado en Cuckoo Sandbox para la red local del edificio de la FIE.

OBJETIVOS ESPECÍFICOS

- Estudiar el contexto tecnológico de redes actuales.
- Implementar una red de prueba para simulación de ataques y técnicas de malware, utilizando tecnología de virtualización.
- Analizar el tráfico de malware en la red local del edificio de la FIE mediante el software Cuckoo Sandbox.
- Medir la seguridad contra ataques a la red de malware.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. Seguridad de redes

La seguridad de redes son las políticas para advertir, informar y monitorear accesos no autorizados, además permite que la seguridad sea garantizada y que el funcionamiento de todas las máquinas de una red sea eficiente, incluyendo evitar que los usuarios que no están autorizados intercedan en el sistema y realicen operaciones involuntarias que lo dañen, previniendo así la interrupción del servicio. El objetivo principal de la seguridad en redes es mantener la integridad, disponibilidad, privacidad, control u autenticidad de la información manejada por computada, mediante procedimientos basados en una política de seguridad tal que permita el control de lo adecuado. (Bustamante, 2014)

1.2. Estándar de seguridad

La norma 802.1 describe la interrelación entre las partes del documento y su relación con el Modelo de Referencia OSI. También contiene información sobre normas de gestión de red e interconexión de redes. Establece los estándares de interconexión relacionados con la gestión de redes. (Matinez, 2011)

- Define algunas cosas útiles, como el formato de la dirección LAN, el protocolo de SNAP, la "EtherTypes Parque infantil", y los arcos OID registro
- La "interfaz de capa superior" grupo de trabajo en 802
- Define la seguridad y la Reducción de "pegamento" que une las LAN definido por los 802 grupos de MAC. (Matinez, 2011)

1.2.1. El estándar 802.1Q

IEEE 802.1Q (también conocido como direccionamiento de VLAN) fue un proyecto en el proceso de estándares IEEE 802 para desarrollar un mecanismo que permita múltiples redes puenteadas para compartir la misma conexión de red física sin que la información pase a otra red (i.e. trunking). IEEE 802.1q es además el nombre del estándar usado por el proceso, y el mismo nombre para el protocolo de encapsulamiento usado para implementar este mecanismo sobre redes Ethernet. (Ledesma, 2008)

El protocolo IEEE 802.1Q, también conocido como dot1Q, fue un proyecto del grupo de trabajo 802 de la IEEE para desarrollar un mecanismo que permita a múltiples redes compartir de forma transparente el mismo medio físico, sin problemas de interferencia entre ellas (Trunking) o enlace troncal. Es también el nombre actual del estándar establecido en este proyecto y se usa para definir el protocolo de encapsulamiento usado para implementar este mecanismo en redes Ethernet. Todos los dispositivos de interconexión que soportan VLAN deben seguir la norma IEEE 802.1Q que especifica con detalle el funcionamiento y administración de redes virtuales. (Payan, 2013)

1.3.Kali Linux

Es una gran suite de seguridad informática, conocido que el ecosistema GNU/Linux en general tiene gran versatilidad y se puede adaptar a numerosos trabajos, desde uso doméstico a trabajos extremadamente complejos en supercomputadoras. (Martinez, 2015)

Entre tantas y tantas distribuciones de GNU/Linux, todas interesantes y útiles dependiendo de los gustos y necesidades de cada persona, hay algunas más especializadas, que tienen funciones más avanzadas en determinados campos y tareas más específicas. Estas distribuciones, si bien no son adecuadas, o simplemente útiles para todos usuarios, ofrecen funcionalidades muy potentes en sus respectivos campos de trabajo. La distribución que en concreto, está centrada en el campo de la seguridad informática.

Kali está basada en Debian, y fue diseñada principalmente para la auditoria y seguridad informática en general. Actualmente es mantenida por Offensive Security Ltd. que desarrolló la distribución a partir de la re-escritura de BackTrack (también desarrollada por ellos), una distribución predecesora a Kali, y que gozó de mucho éxito entre las personas que se dedicaban a esta actividad. (Martinez, 2015)

Kali Linux trae preinstalados una gran cantidad de programas relacionados con el tema de la seguridad informática (más de 600 programas), siendo algunas de las más conocidas Nmap (un escáner de puertos), Wireshark (un sniffer), John the Ripper (Un crackeador de passwords) y la suite Aircrack-ng (Software para pruebas de seguridad en redes inalámbricas), además del inigualable Metasploit, la gran suite de explotación de vulnerabilidades. (Martinez, 2015)

1.3.1. Características de Kali Linux

Kali Linux es una completa reconstrucción de BackTrack Linux, y se adhiere completamente a los estándares de desarrollo de Debian. Se ha puesto en funcionamiento toda una nueva infraestructura, todas las herramientas han sido revisadas y empaquetadas, y se utiliza ahora Git para el CVS. (Quezada, 2015)

- Más de 300 herramientas de Pruebas de Penetración
- Es Libre y siempre lo será
- Árbol Git Open Source
- Cumple con FHS (Filesystem Hierarchy Standart)
- Amplio soporte para dispositivos inalámbricos
- Parches al Kernel para inyección.
- Entorno de desarrollo seguro
- Paquetes y repositorios firmados con GPG
- Varios lenguajes
- Completamente personalizable
- Soporte ARMEL y ARMHF. (Quezada, 2015)

1.3.2. Política de Código Abierto en Kali Linux

Kali Linux es una distribución que agrega miles de paquetes de software libre en su sección principal. Como derivado de Debian, todo el software en sí, cumple con las Guías de Software Libre de Debian.

Como una excepción a lo anterior, Kali Linux no-libre contiene varias secciones con herramientas que no son de código abierto, pero que son permitidas para su distribución por Offensive Security a través de licencias específicas o determinadas en acuerdo con los vendedores. Si tú quieres construir un derivado de Kali, deberías revisar la licencia de cada paquete no-libre de Kali(especifico) antes de incluirlo en tu distribución (paquetes no-libres los cuales son importados de Debian son seguros para redistribuir). (Kali Linux, 2017)

Más importante aún, todos los desarrollos específicos de Kali hechos para su infraestructura o para integrar el software suministrado han sido puestos bajo la licencia GNU GPL. Si requiere de más información acerca de la licencia o cualquier pieza de software, puede chequear el paquete de código en debian/copyright o /usr/share/doc/*package*/copyright para un paquete que ya tenga instalado. (Kali Linux, 2017)

1.4. Malware

1.4.1. Definición

El término Malware proviene de la asociación de dos palabras de la lengua inglesa: Malicious Software. Las cuales se asocian a todo aquel software que tiene el propósito de causar un daño. Los objetivos de éste software malicioso van desde una simple recolección de información personal de usuario hasta el uso de recursos de forma remota o bien dañar la estructura del sistema operativo afectado. El Malware es todo código malévolo instalado en una computadora y puede dar al atacante un grado verdaderamente alarmante de control sobre el sistema, red o datos, del usuario sin su conocimiento (Oscar & Reyes, 2009)

1.4.2. Evolución

El desarrollo de programas dañinos se originó a través de la creación de virus informáticos y, aunque inicialmente sus fines se destinaban estrictamente a la investigación, con el tiempo su objetivo derivó en la obtención de reconocimiento por parte de sus autores y en la actualidad con fines lucrativos. La evolución de los malware comenzó en 1949 cuando Von Neumann estableció la idea de un programa almacenado y expuso La Teoría y Organización de Autómatas Complejos, donde presentaba por primera vez la posibilidad de desarrollar pequeños programas replicantes y capaces de tomar el control de otros programas de similar estructura. (Cardozo & García, 2015)

En 1972 Robert Thomas Morris creó el primer virus: el Creeper era capaz de infectar máquinas IBM 360 de la red ARPANET (la precedente de Internet) y emitía un mensaje en pantalla que decía “Soy una enredadera (Creeper), atrápame si puedes”. Para eliminarlo, se creó otro virus llamado Reaper (segadora) que estaba programado para buscarlo y eliminarlo. Este es el origen de los actuales antivirus. A principio de este siglo se considera la época de las grandes epidemias masivas que tuvieron su punto álgido en el 2004, donde utilizando técnicas de ingeniería social, se infectaba a los usuarios por medio del correo electrónico siendo el gusano I LOVE YOU el de mayor repercusión mediática. (Cardozo & García, 2015)

El malware se dio a conocer a muchos usuarios de equipos a través de las infecciones generalizadas causadas por Melissa (en 1999) y LoveLetter (en 2000). Ambos se basaban en el correo electrónico, y LoveLetter que se propagó a través de datos adjuntos de correo electrónico infectados. Cuando se abría el documento adjunto, el malware sobrescribía una serie de tipos de archivos diferentes en el equipo del usuario y se enviaba a sí mismo por correo electrónico a otras

personas de la libreta de direcciones del usuario. LoveLetter se convirtió rápidamente en el incidente más costoso de su clase en ese momento. (Microsoft, 2012)

A pesar de los daños causados por Melissa y LoveLetter, podría aducirse que estas infecciones tuvieron tres efectos positivos: hicieron que el malware informático fuera sometido a controles más estrictos; aumentaron la sensibilización social sobre el malware informático y recalcaron la importancia de las copias de seguridad. (Microsoft, 2012)

1.4.3. Clasificación de los Malwares

Los Malware pueden ser clasificados según sus efectos y características dela siguiente manera:

1.4.3.1 Virus

Un virus es un programa introducido subrepticiamente en la memoria de un ordenador que, al activarse, destruye total o parcialmente la información almacenada, es decir un virus informático es un programa que se copia automáticamente ya sea por medios de almacenamiento o por Internet, y que tiene por objeto alterar el normal funcionamiento del ordenador. Un virus puede ser o no, muy peligroso, pero independientemente de dicho grado, si el sistema a comprometer es crítico, un virus de bajo grado de peligrosidad podrá causar graves daños. Si por el contrario dicho virus es muy peligroso y afecta a una computadora familiar sus daños serán mínimos. Por ello desde el punto de vista de una empresa o gran corporación, un virus sea cual sea, debe ser considerado siempre como peligroso. (Prieto Álvarez, 2014)

1.4.3.2. Rootkits

Es un software que modifica el sistema operativo de la computadora, y permite que el malware permanezca oculto al usuario, evitando que el proceso malicioso sea visible en el sistema.
Originalmente el término Rootkit proviene de sistemas Unix y hacía referencia a pequeñas utilidades y herramientas que permitían acceso como "root" de esos sistemas. El término ha evolucionado y actualmente es un conjunto de herramientas utilizadas en cualquier sistema para conseguir acceder ilícitamente al mismo. Generalmente se los utiliza para ocultar procesos y programas que permiten acceso al sistema atacado, incluso tomar control de parte del mismo. (Segu.Info, 2015)

1.4.3.3. Troyanos

El término troyano proviene de la leyenda del caballo de Troya, ya que su objetivo inicial es el de engañar a los usuarios para que los ejecuten simulando ser archivos normales e indefensos, como juegos, programas, animaciones etc. Es decir, es un software malicioso que permite la administración remota de una computadora de forma oculta y sin el consentimiento del propietario. Generalmente están disfrazados como algo atractivo o inocuo que invitan al usuario a ejecutarlo. Pueden tener un efecto inmediato y tener consecuencias como el borrado de archivos del usuario e instalar más programas maliciosos. Son usados para empezar la propagación de un gusano, inyectándolo de forma local dentro del usuario. (Lara, 2015)

1.4.3.4. Gusanos

Los Gusanos Informáticos son programas que realizan copias de sí mismos, alojándolas en diferentes ubicaciones del ordenador. El objetivo de este malware suele ser colapsar los ordenadores y las redes informáticas, impidiendo así el trabajo a los usuarios. A diferencia de los virus, los gusanos no infectan archivos. Su principal objetivo es propagarse y afectar al mayor número de ordenadores posible. Los gusanos suelen utilizar técnicas de ingeniería social para conseguir mayor efectividad, para ello los creadores de malware seleccionan un tema con para camuflar el archivo malicioso. Los temas más recurrentes son los relacionados con el sexo, famosos, temas morbosos, temas de actualidad o software pirata. (Cardozo & García, 2015)

1.4.3.5. Backdoors

Una puerta trasera o Backdoor es un software que permite el acceso al sistema de la computadora ignorando los procedimientos normales de autenticación. Una puerta trasera es un programa que permite a un atacante brincar los controles normales de seguridad, como son cortafuegos y filtros de acceso de ip. Tienen la característica de permitirle al atacante conectarse remotamente al equipo infectado, luego de que el atacante accede al ordenador del usuario, los usos que puede hacer del mismo son variados, según las herramientas que utilice: enviar correos masivos, eliminar o modificar archivos, ejecución de archivos, reiniciar el equipo o usos más complejos como instalar aplicaciones para uso malicioso. (Ramírez & Reyes, 2009)

1.4.3.6. Spywares

Según (Ramírez & Reyes, 2009) los spyware son aplicaciones que recopilan información sobre una persona u organización sin su conocimiento. La función más común que tienen estos programas es la de recopilar información sobre el usuario y distribuirlo a empresas publicitarias u otras organizaciones interesadas, pero también se han empleado en círculos legales para recopilar información contra sospechosos de delitos, como en el caso de la piratería de software.

Además, pueden servir para enviar a los usuarios a sitios de Internet que tienen la imagen corporativa de otros, con el objetivo de obtener información importante. Puede tener acceso por ejemplo a: correo electrónico y contraseña; dirección IP y DNS; teléfono, país; páginas que se visitan, qué tiempo se está en ellas y con qué frecuencia se regresa; qué software está instalado en el equipo y cuál se descarga; qué compras se hacen por Internet; tarjeta de crédito y cuentas de banco.

1.4.3.7. Keyloggers

Los Keyloggers monitorizan todas las pulsaciones del teclado y las almacenan para realizar operaciones fraudulentas como son pagos desde cuentas de banco o tarjetas de crédito. La mayoría de estos sistemas son usados para recopilar contraseñas de acceso, espiar conversaciones de chat u otros fines. (Lara, 2015). Los Keyloggers existen forma de hardware y software.

Los Keyloggers de hardware son pequeños dispositivos que se instalan entre nuestra computadora y el teclado. Son difíciles de identificar para un usuario inexperto, pero si se presta atención es posible reconocerlos a simple vista. Tienen distintas capacidades de almacenamiento, son comprados en cualquier casa especializada y generalmente son instalados por empresas que desean controlar a ciertos empleados, mientras que los Keyloggers por software, actualmente son los más comunes, muy utilizados por el malware orientado a robar datos confidenciales o privados del usuario. Como es de imaginar, la información obtenida es todo lo que el usuario ingrese en su teclado como por ejemplo documentos, nombres de usuarios, contraseñas, números de tarjetas, Pines, etc. (Segu.Info, 2015)

1.4.3.8. Ransomware

Según (Lara, 2015) los Ransomware son también llamados criptovirus o secuestradores y son programas que cifran los archivos importantes para el usuario, haciéndolos inaccesibles, y piden que se pague un “rescate” para poder recibir la contraseña que permite recuperar los archivos.

1.4.4. Vulnerabilidades usadas por el malware

Existen factores que hacen un sistema más vulnerable al malware por ejemplo la homogeneidad, errores de software, código sin confirmar, sobre-privilegios de usuario y sobre-privilegios de código. Una causa de la vulnerabilidad de redes, es la homogeneidad del software multiusuario. Es decir, cuando todos los ordenadores de una red funcionan con el mismo sistema operativo, si se puede comprometer ese sistema, se podría afectar a cualquier ordenador que lo use. Los

creadores de malware pueden infectar una gran cantidad de computadoras sin tener que adaptar el software malicioso a diferentes sistemas operativos. . (EtapaNet, 2010)

1.4.5. Captura de malware con Dionaea

En los últimos años se ha podido observar la evolución acelerada de las tecnologías de la información y comunicación a tal grado que al día de hoy conoce y disfruta los servicios que nos permiten acceder a la información que se genera en el mundo de manera rápida y sencilla. Sin embargo, este mismo desarrollo ha generado diversas amenazas que, si bien no son nuevas, se han adaptado perfectamente al tiempo y a las necesidades actuales, de tal manera que representan serios problemas para usuarios finales, para pequeñas y medianas empresas, grandes organizaciones, universidades y gobiernos. (Santillan, 2015)

Ante este panorama, los profesionales de la seguridad de la información se han visto en la necesidad de desarrollar habilidades y conocimiento que les permita estudiar y analizar el comportamiento, la estructura, los riesgos y el impacto que estas amenazas pueden provocar.
El término que engloba al conjunto de elementos al que se refiere el malware, acrónimo de malicious software, dentro del cual se incluyen virus, gusanos, troyanos, puertas traseras, bots, rootkits, spyware, rogues, ransomware, downloaders, droppers, entre otros. (Santillan, 2015)

1.4.5.1. Protocolos Dionaea para captura del malware

A continuación, trampas para malware de Dionaea:

- **Server Message Block (SMB)** - SMB es el protocolo principal ofrecido por Dionaea. SMB tiene una historia digna de errores explotables remotas, y es un objetivo muy popular para los gusanos.

- **Protocolo de transferencia de hipertexto (HTTP)** - Dionaea soporta HTTP en el puerto 80, así como **HTTPS**. Un certificado SSL autofirmado se crea en el arranque para HTTPS.

- **Protocolo de transferencia de archivos (FTP)** - Dionaea proporciona un servidor FTP en el puerto de base 21. Se permite la creación de directorios, y la carga y descarga de archivos.

- **Trivial File Transfer Protocol (TFTP)** - Dionaea proporciona un servidor TFTP en el puerto 60 que se puede utilizar para servir archivos.

- **Microsoft SQL Server (MSSQL)** - Dionaea implementa el protocolo de secuencia de datos tabular que es utilizado por Microsoft SQL Server. Escucha de TCP / 1433 y permite a los clientes acceder, se puede decodificar las consultas se ejecutan en la base de datos.

- **Voz sobre IP (VoIP)** - Desarrollado como parte de GSoC 2011 por PHIBO, el protocolo VoIP utiliza en Dionaea es el protocolo inicial de Sesión (SIP). Este módulo no se conecta a un registrador de VoIP / servidor externo; simplemente espera a que los mensajes SIP entrantes, registra todos los datos como incidentes y / o vertederos de datos binarios, y reacciona en consecuencia. (Tan, 2014)

1.4.5.2. LibEmu

Dionaea utiliza LibEmu para detectar y evaluar cargas enviadas por los atacantes con el fin de obtener una copia del malware.

LibEmu se utiliza detectar, medir, y si es necesario, ejecute el código shell. Shellcode medición perfiles se llevan a cabo mediante la ejecución del código shell en LibEmu VM, y el registro de llamadas a la API y argumentos. Esto es suficiente para la mayoría de los perfiles de shellcodes; pero no para shellcodes de múltiples etapas. Además de la grabación de llamadas a la API y argumentos, permitir shellcodes a tomar medidas. (Tan, 2014)

Una vez que se obtuvo la carga útil y su perfil, tener que actuar sobre él con el fin de adquirir una copia del malware. (Tan, 2014)

Se presenta técnicas utilizadas por los atacantes informaticos, y cómo Dionaea actuar sobre ellos:

- **Shell Encuadernación / conectarse de nuevo, Exec** - Dionaea ofrece emulación de la cáscara de la carga útil que ofrece una cáscara al atacante.

- **URLDownToFile API** -Otra vez, Dionaea ofrece emulación de concha y actúa sobre shellcodes que utiliza URLDownloadToFile llamada a la API para recuperar archivos a través de HTTP y ejecutar archivos recuperados después.

- **Las cargas útiles de varias etapas** - Nunca sabra qué esperar de la etapa posterior; LibEmu se utiliza para ejecutar el código shell en el LibEmu VM. (Tan, 2014)

1.4.6. Ejecución de un Malware

1.4.6.1. Frutas RAT

Frutas RAT es un troyano muy difícil de detectar que entra en el interior del equipo, por la obsolescencia del sistema operativo y aprovecha la vulnerabilidad del nivel bajo de seguridad con el que cuenta el equipo. Frutas RAT toma el control del equipo cuando los usuarios visitan sitios web hackeadas o descargan archivos infectados. Sin que se dé cuenta el usuario se infecta con un rootkit, Frutas RAT no sólo puede tomar ventaja de las lagunas del sistema para atacar el sistema, sino que también puede evitar la mayor parte de detección de seguridad.

Frutas RAT fue desarrollado en Java por Adwind y cada versión recibe el nombre de una fruta. Por ejemplo, Frutas RAT v 0.9 de llama Nuez, la v0.7 Durazno y la v0.6 Manzana; al estar desarrollado en Java, se hace ideal para ejecutar en cualquier ambiente ya que el servidor generado por el troyano en un archivo JAR, la infección puede realizarse sobre cualquier sistema operativo, Windows, Linux o MacOS. (Borghello, 2014)

El troyano Frutas RAT, es un archivo .JAR que, al momento de ejecutarse en el host de la víctima, crea un servidor el cual deja abierta una puerta trasera denominada Backdoor engañando un fichero de configuración con dirección IP y puerto.

Esta puerta de acceso permite a los atacantes informáticos llevar a cabo un sinfín de acciones, como, por ejemplo, finalizar procesos, explorar los archivos del sistema hacer aparecer un pop-up, descargar y ejecutar ficheros, una vez instalado Frutas RAT crea muchos archivos maliciosos e infecciosos en el disco duro del sistema.

1.5. Análisis de Malware

El análisis dinámico de códigos maliciosos permite conocer de una manera rápida y efectiva qué acciones realiza una amenaza en el sistema. De esta forma se puede obtener información acerca de los archivos creados, conexiones de red, modificaciones en el registro, etc. Para lograr este fin existe una gran cantidad de recursos y herramientas que brindan la posibilidad de analizar una

amenaza a través de diferentes enfoques. En el post de hoy compartir con ustedes algunos conceptos y herramientas a tener en cuenta al momento de conocer qué es lo que hace el código malicioso que desea analizar. (Ramos, 2011)

Siempre es necesario tener en cuenta cuál es la amenaza que se analizó y en base a ello decidir si es necesario realizar el trabajo en una máquina física o si un entorno virtual será lo mejor. Esta elección depende de los gustos del investigador y también del código malicioso a analizar; si el malware tiene protección contra máquinas virtuales lo mejor será hacerlo directamente en una máquina física. (Ramos, 2011)

1.6. Identificación de Amenazas

La identificación de amenazas requiere conocer los tipos de ataques, el tipo de acceso, la forma operacional y los objetivos del atacante. Desde 1990 hasta nuestros días, el CERT viene desarrollando una serie de estadísticas que demuestran que cada día se registran más ataques informáticos, y estos son cada vez más sofisticados, automáticos y difíciles de rastrear. (Borghello, 2014)

La figura 1- 1 detalla el tipo de atacante, las herramientas utilizadas, en qué fase se realiza el ataque, los tipos de procesos atacados, los resultados esperados y/u obtenidos y los objetivos perseguidos por los intrusos. (Borghello, 2014)

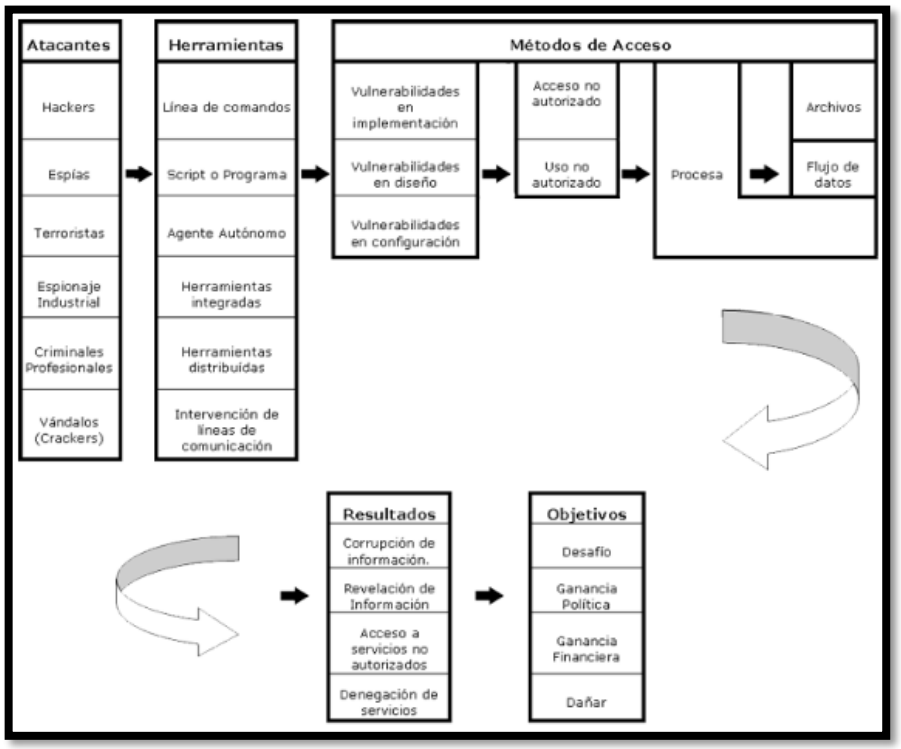


Figura 1 - 1: Identificación de amenazas
Realizado por: (Borghello, 2009)

1.7. Análisis de la red

1.7.1. SPAN LOCAL

SPAN local admite una sesión SPAN totalmente dentro de un switch; Todos los puertos de origen o VLAN de origen y los puertos de destino se encuentran en el mismo conmutador o pila de conmutadores. SPAN local copia tráfico de uno o más puertos de origen en cualquier VLAN o de una o más VLAN a un puerto de destino para el análisis. Por ejemplo, en la Figura 2-1, todo el tráfico en el puerto 5 (el puerto de origen) se refleja en el puerto 10 (el puerto de destino). Un analizador de red en el puerto 10 recibe todo el tráfico de red desde el puerto 5 sin estar físicamente conectado al puerto 5. (Cisco, 2014)

Las sesiones SPAN (locales o remotas) le permiten supervisar el tráfico en uno o más puertos o una o más VLAN y enviar el tráfico supervisado a uno o más puertos de destino. (Cisco, 2014)
Una sesión SPAN local es una asociación de un puerto de destino con puertos de origen o VLAN de origen, todo ello en un solo dispositivo de red. SPAN local no tiene sesiones de origen y de destino separadas. Las sesiones SPAN locales recopilan un conjunto de paquetes de entrada y salida especificados por el usuario y los forman en un flujo de datos SPAN, que se dirige al puerto de destino. (Cisco, 2014)

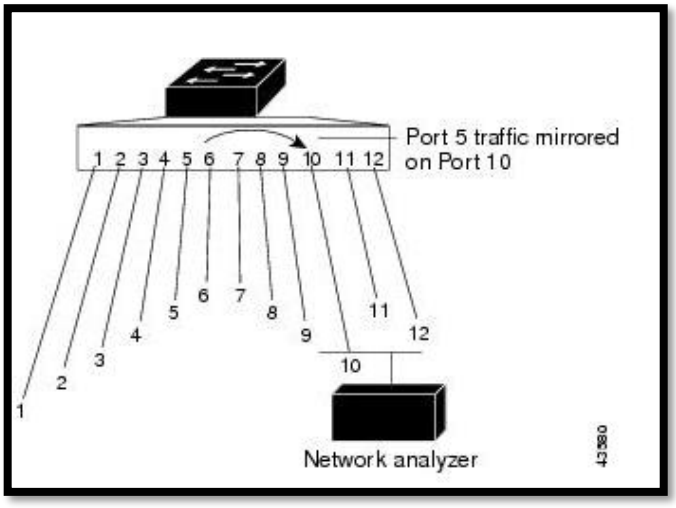


Figura 2-1:Span Local
Realizado por: (Cisco, 2014)

El monitoreo del tráfico en una sesión SPAN tiene las siguientes restricciones:

- Las fuentes pueden ser puertos o VLAN, pero no se pueden mezclar puertos de origen y VLAN de origen en la misma sesión.

- El conmutador admite hasta dos sesiones SPAN o RSPAN locales.

- Puede tener varios puertos de destino en una sesión SPAN, pero no más de 64 puertos de destino por pila de conmutadores.

- Las sesiones SPAN no interfieren con el funcionamiento normal del interruptor. Sin embargo, un destino SPAN de sobre suscritos, por ejemplo, un puerto de 10 Mb / s que supervisa un puerto de 100 Mb / s, puede resultar en paquetes perdidos o perdidos. (Cisco, 2014)

1.7.2. Trafico supervisado

Las sesiones SPAN pueden supervisar estos Tipos:

- Recepción (Rx) SPAN-El objetivo de recibir (o ingresar) SPAN es monitorear tanto como sea posible todos los paquetes recibidos por la interfaz de origen o VLAN antes de que cualquier modificación o procesamiento sea realizado por el switch. Una copia de cada paquete recibido por la fuente se envía al puerto de destino para esa sesión SPAN. (Cisco, 2014)

- Transmitir (Tx) SPAN-El objetivo de transmitir (o salida) SPAN es monitorear tanto como sea posible todos los paquetes enviados por la interfaz de origen después de que la modificación y el procesamiento sea realizado por el conmutador. Una copia de cada paquete enviado por el origen se envía al puerto de destino para esa sesión SPAN. La copia se proporciona después de que se modifica el paquete. (Cisco, 2014)

1.7.3. Puertos de origen

Un puerto de origen (también llamado puerto supervisado) es un puerto conmutado o enrutado que supervisa para el análisis de tráfico de red. En una sesión SPAN local o una sesión de origen RSPAN, puede supervisar los puertos de origen o las VLAN para el tráfico en una o ambas

direcciones. El conmutador admite cualquier número de puertos de origen (hasta el número máximo de puertos disponibles en el conmutador) y cualquier número de VLAN de origen (hasta el número máximo de VLAN admitidas). (Cisco, 2014)

1.7.4. Configuración de un SPAN local

1.7.4.1. Pautas de configuración SPAN

- El puerto de destino no puede ser un puerto de origen; Un puerto de origen no puede ser un puerto de destino.

- No puede tener dos sesiones SPAN con el mismo puerto de destino.

- Cuando configura un puerto de conmutador como un puerto de destino SPAN, ya no es un puerto de switch normal; Sólo el tráfico supervisado pasa a través del puerto de destino SPAN.

- La introducción de los comandos de configuración SPAN no elimina los parámetros SPAN previamente configurados. Debe ingresar la sesión de no monitor {session_number | Todos | Local | Remoto} comando de configuración global para eliminar los parámetros SPAN configurados.

- Para SPAN local, los paquetes salientes a través del puerto de destino SPAN llevan los encabezados de encapsulación originales, sin etiquetar, ISL o IEEE 802.1Q, si se especifican las palabras clave de replicación de encapsulamiento. Si las palabras clave no se especifican, los paquetes se envían en forma nativa.

- Puede limitar el tráfico SPAN a VLANs específicas mediante la palabra clave vlan del filtro. Si se está supervisando un puerto troncal, sólo se controla el tráfico en las VLAN especificadas con esta palabra clave. De forma predeterminada, todas las VLAN se supervisan en un puerto troncal.

- No puede mezclar VLAN de origen y VLAN de filtro en una sola sesión SPAN. (Cisco, 2014)

1.7.4.2. Creación de una sesión SPAN local

Como se observa en la tabla 1 -1, seguir los pasos para crear una sesión SPAN y especifique los puertos de origen (supervisados) o las VLAN y los puertos de destino (de supervisión): (Cisco, 2014)

Tabla 1-1: Creación de una sesión SPAN Local

PASOS	MANDO	PROPÓSITO
Paso 1	Configurar terminal	➢ Ingrese al modo de configuración global.
Paso 2	Ninguna sesión de monitor { session_number | Todos | Local | Remoto }	➢ Elimine cualquier configuración SPAN existente para la sesión. ➢ Para session_number, el rango es de 1 a 66. ➢ Especifique todo para eliminar todas las sesiones SPAN, local para eliminar todas las sesiones locales o remota para eliminar todas las sesiones SPAN remotas.
Paso 3	Monitor session session_number fuente { interfaz interfaz-id | Vlan vlan-id } [, | -] [ambos | Rx | Tx]	➢ Especifique la sesión SPAN y el puerto de origen (puerto supervisado). ➢ Para session_number, el rango es de 1 a 66. ➢ Para id de interfaz, especifique el puerto de origen o la VLAN de origen que se va a supervisar. ➢ Para la interfaz de origen id, especifique el puerto de origen a monitorizar. Las interfaces válidas incluyen interfaces físicas e interfaces lógicas de puerto-canal (puerto-canal, número-canal-puerto) . Los números de canal de puerto válidos son 1 a 48. ➢ Para vlan-id, especifique la VLAN de origen a supervisar. El rango es de 1 a 4094 (excluyendo la VLAN RSPAN).
	Monitor session session_number destino { interface interface-id [, | -] [replicación de encapsulación]}	➢ Especifique la sesión SPAN y el puerto de destino (puerto de supervisión). ➢ Para session_number, especifique el número de sesión ingresado en el paso 3. ➢ Nota Para SPAN local, debe utilizar el mismo número de

		sesión para las interfaces de origen y de destino. ➢ Para interface-id, especifique el puerto de destino. La interfaz de destino debe ser un puerto físico; No puede ser un EtherChannel, y no puede ser una VLAN.
	Fin	➢ Regresar al modo EXEC privilegiado.
	Show monitor [session_number de sesión] Show running-config	➢ Verifique la configuración.
	Copiar running-config startup-config	➢ (Opcional) Guarde la configuración en el archivo de configuración.

Realizado por: (Cisco, 2014)

1.8. Oracle VM Virtual Box

VirtualBox es un programa de virtualización capaz de instalarse en un ordenador de cualquier sistema operativo de manera virtual, está herramienta ayuda a conocer y analizar los diferentes sistemas operativos, con el propósito de probar y crear aplicaciones de software sin alterar el ordenador, esta aplicación puede ser descargada de su web oficial, es decir www.virtualbox.org, tomando en cuenta la versión que sea compatible para el sistema operativo en el cual se esté trabajando. A continuación, se detalla las características importantes de esta aplicación. (Sanz, 2016)

- Puede ser instalado en diversos Sistemas Operativos de 32 y 64 bits de Windows, GNU/Linux, Mac OS X y Solaris.

- Virtual Box permite que el usuario organice las MMVV individualmente y colectivamente.

- Es software libre, trabaja con licencia GPLv2.

- Virtual Box es muy parecido en todas las plataformas donde se puede ejecutar ya que se pueden portar MMVV entre ellas, es decir se puede crear una Máquina Virtual en Windows y luego ejecutarla en GNU/Linux. (Gómez, 2016)

1.9. CUCKOO SANDBOX

Cuckoo es un sistema automatizado de análisis de malware de código abierto. Se utiliza para ejecutar y analizar automáticamente archivos y reunir resultados de análisis exhaustivos de malware. Funciona mientras se ejecuta dentro de un sistema operativo Windows aislado. Puede recuperar el siguiente tipo de resultados: (Carlos, 2017)

- Trazas de llamadas de la API de win32 realizadas por todos los procesos generados por el malware.

- Archivos que están siendo creados, eliminados y descargados por el malware durante su ejecución.

- Capturas de pantalla del escritorio de Windows tomadas durante la ejecución del malware. (Carlos, 2017)

Cuckoo Sandbox se compone de un software de gestión central que maneja la ejecución de la muestra y el análisis, cada análisis se lanza en una máquina virtual fresca y aislada. La infraestructura de Cuckoo está compuesta por un anfitrión (El software de gestión) y una serie de máquinas Guest (máquinas virtuales para el análisis). (Carlos, 2017)

El host ejecuta el componente principal del Sandbox que gestiona todo el proceso de análisis, mientras que los invitados son los entornos aislados donde el malware se ejecuta y se analiza con seguridad. (Sandbox, 2016). En la figura 222 se muestra la arquitectura de Cuckoo Sandbox.

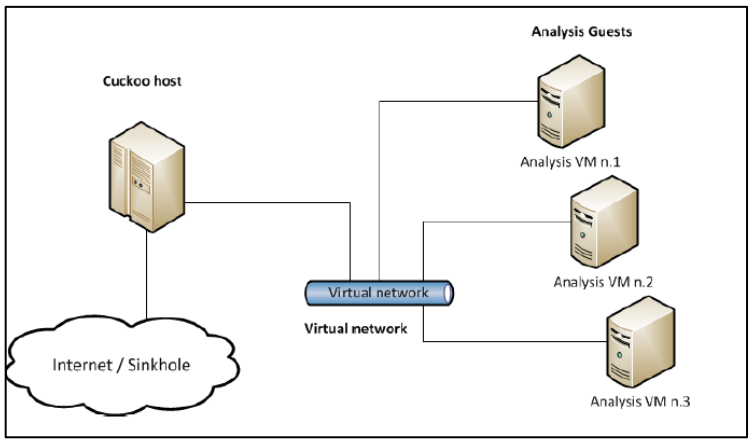


Figura 3-1: Arquitectura de Cuckoo Sandbox
Fuente: Sandbox 2016

- **Cuckoo Host:** Es el responsable de la gestión de invitados y análisis. Además, carga el tráfico y genera los informes.

- **Analysis Guests:** Genera un ambiente limpio cuando se ejecuta una muestra.

- **Virtual network:** Es una red insolada donde se ejecutan las máquinas virtuales de análisis.

Cuckoo Sandbox se puede descargar desde el sitio web oficial, donde se distribuyen las versiones estables y empaquetadas. Para su instalación es recomendable configurar en GNU / Linux (Debian o Ubuntu), además en Mac OS X como host. (Carlos, 2017)

1.7.1. Instalación de Cuckoo Sandbox

Para realizar la configuración de Cuckoo se necesitará instalar softwares y bibliotecas requeridos. A continuación, se detalla los pasos para realizar una correcta instalación:

- Instalar python en la versión 2.7 como se muestra en la figura 4-1.

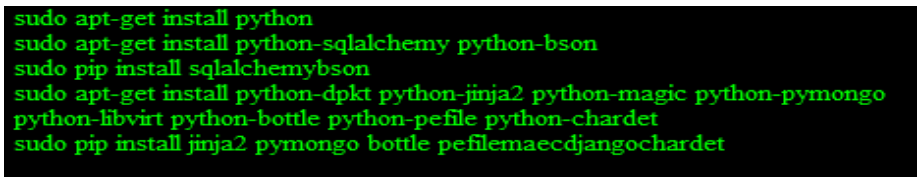


Figura 4-1: Instalando dependencias de cuckoo
Realizado por: Jonathan Quezada.2017

- Como se observa en la figura 5-1 se instala Tcpdump para analizar las conexiones que realiza el host infectado con el malware.

Figura 5-1: Instalación Tcpdump
Realizado por: Jonathan Quezada.2017

- Instalar yara, pycripto y distorm para poder instalar distorm como se observa en la figura 6-1.

```
wget http://distorm.googlecode.com/files/distorm3-1.0.zip
unzip distorm3-1.0.zip
cd distorm3
python setup.py build
python setup.py build install
```

Figura 6-1:Instalación distorm
Realizado por: Jonathan Quezada.2017

- Se instala yara como se observa en la figura 7-1.

```
wget http://yara-project.googlecode.com/files/yara-python-1.4a.tar.gz
tar -xvzf yara-python-1.4a.tar.gz
cd yara-python-1.4a
python setup.py build
python setup.py build install
```

Figura 7-1: Instalación yara
Realizado por: Jonathan Quezada.2017

- En la figura 8-1 se instaló pycripto.

```
wget http://ftp.dlitz.net/pub/dlitz/crypto/pycrypto/pycrypto-2.6.1.tar.gz
tar -xvzf pycrypto-2.6.1.tar.gz
cd pycrypto-2.6.1/
python setup.py build
python setup.py build install
```

Figura 8-1: Instalación pycripto
Realizado por: Jonathan Quezada.2017

- Instalar volatility como se muestra en la figura 9-1.

```
wget https://code.google.com/p/volatility/downloads/detail?name=volatility-2.3.1.tar.gz
tar -xvzf volatility-2.3.1.tar.gz
cd  volatility-2.3.1/
python setup.py install
```

Figura 9-1: Instalación volatility
Realizado por: Jonathan Quezada.2017

- Crear un usuario cuckoo y lo añade al grupo de VirtualBox, deber tener en cuenta que trabaja con el kernel de VirtualBox, al momento de crear las máquinas virtuales nos asegura que el usuario cuckoo debe pertenecer al grupo para así al momento de ejecutar cuckoo pueda tener acceso a estas máquinas como se observa en la figura 10-1.

```
adduser cuckoo
sudousermod -G vboxusers cuckoo
```

Figura 10-1: Crear un usuario cuckoo
Realizado por: Jonathan Quezada.2017

- Instalar Msql-server como se observa en la figura 11-1.

```
sudo apt-get install mysql-server python-mysqldb -y
mysql -u root -p
create database cuckoo;
grant all privileges on cuckoo.* to cuckoo@localhost identified by 'passW0rd' ;
flushprivileges;
quit;
```

Figura 11-1: Instalar base de datos
Realizado por: Jonathan Quezada.2017

- Finalmente, se clona o instala cuckoo, como se observa en la figura 12-1.

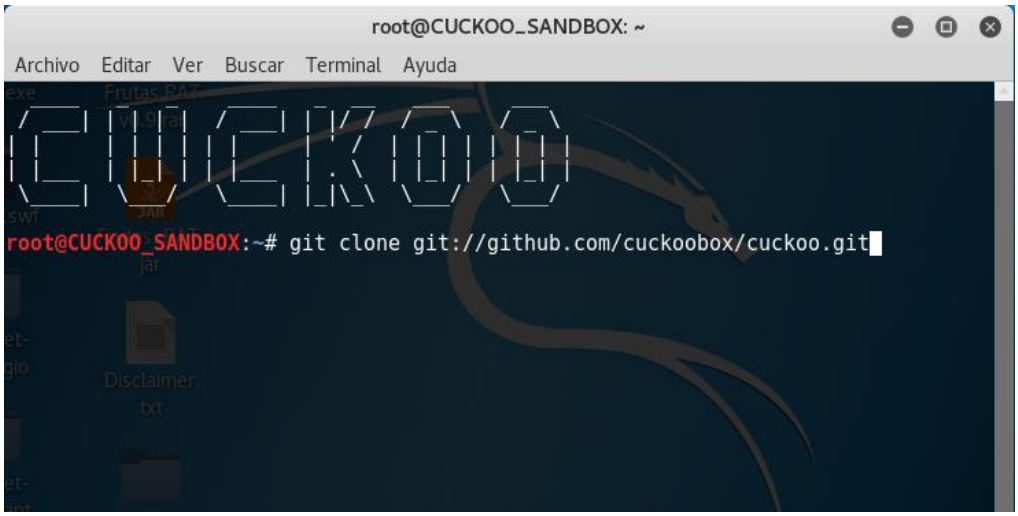


Figura 12-1: Instalación cuckoo
Realizado por: Jonathan Quezada.2017

- En la Figura 13-1 se observa como Cuckoo se carga correctamente, tomando en cuenta que se tiene que configurar la máquina virtual.

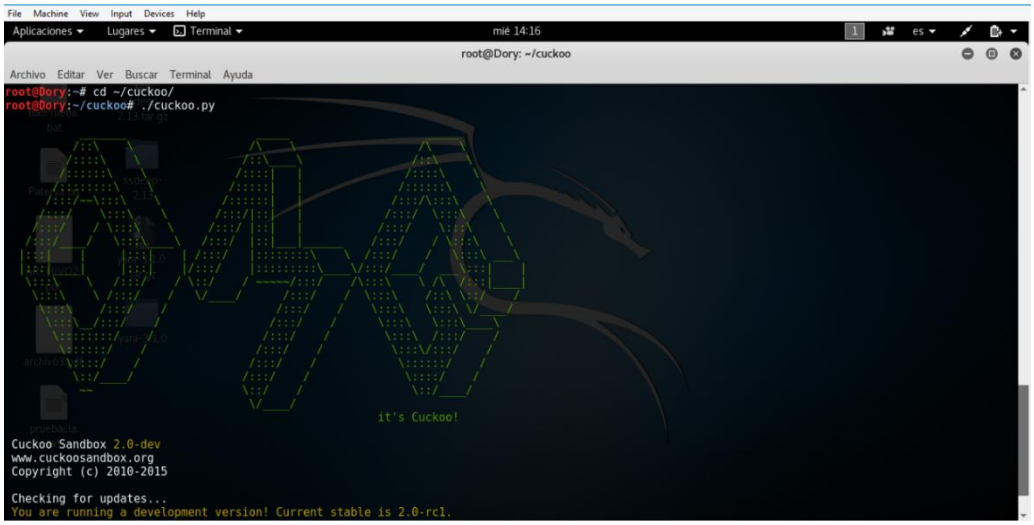


Figura 13-1: Cuckoo en escucha
Realizado por: Jonathan Quezada.2017

- Instalación y configuración de la máquina virtual:

Como se observa en la Figura 14-1 deber instalar una máquina virtual en nuestro caso trabajar con VirtualBox, al host lo llamó cuckoo, se configura la red de la maquina como solo anfitrión, por lo tanto, crear una interfaz virtual, de manera gráfica o por medio de linea de consola en el terminal de Kali Linux.

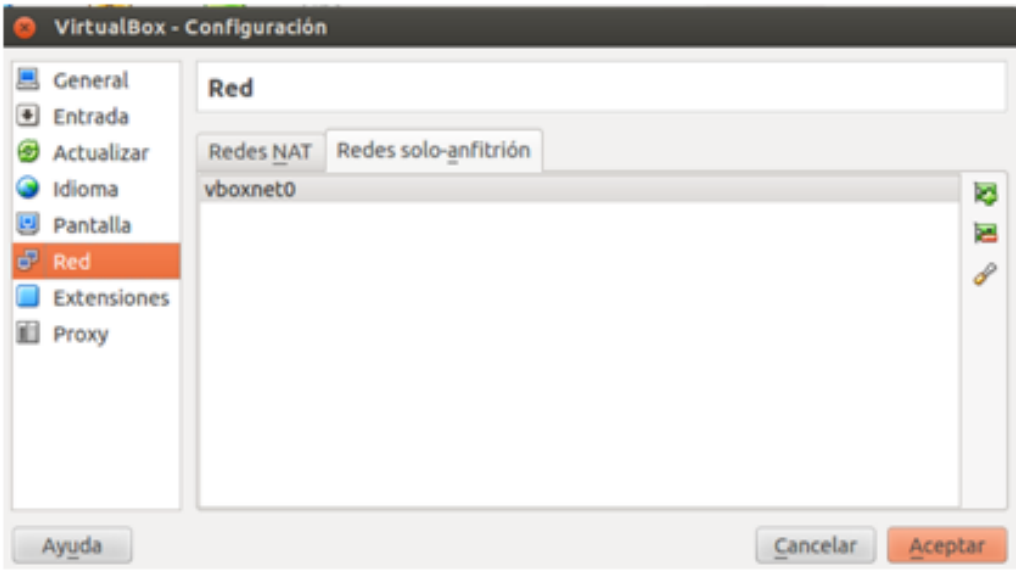


Figura 14-1: Configuración del equipo
Realizado por: Jonathan Quezada.2017

El equipo debe estar en la red que creo en la Figura 15-1. Se determina que para la configuración de nuestra maquina víctima es muy importante que se encuentre en la misma red para realizar él envió de malware a el equipo víctima.

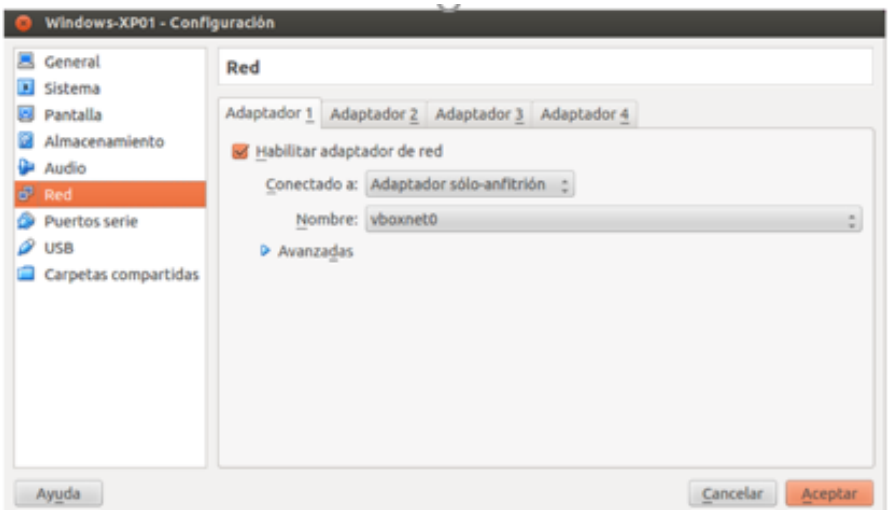


Figura 15-1: Configurando la red del host
Realizado por: Jonathan Quezada.2017

Se procede a configurar el host, previamente debe instalar python y además se recomienda instalar las Guestadditions de VirtualBox. Copiar el agente.py que se encuentra en la ruta /cuckoo/agent y se deberá configurar para que arranque junto con el inicio de Windows para que puedan tener conectividad entre cuckoo y el host de Windows. Finalmente tomar una snapshot de la máquina anfitrión, sobre esta snapshot se ejecutarán todos los ficheros que le pasarán a Cuckoo. (Amaya, 2013)

- Configuración de Cuckoo Sandbox

- Configuración de la máquina Host

Antes de poner a funcionar el sistema automático de análisis de muestras de malware debe tener presente que cuckoo se inicia a través de diferentes archivos de configuración, por lo que debe configurarlos previos al inicio del sistema. (Amaya, 2013)

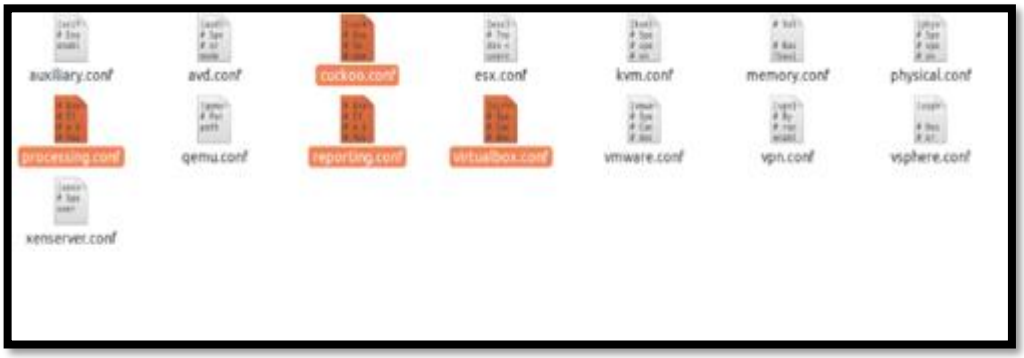


Figura 16-1: Configuración de parámetros
Realizado por: Jonathan Quezada.2017

En el archivo cuckoo.conf este archivo se definen los parámetros de conectividad de la máquina que servirá como HOST y la interfaz de red a través de la cual se van a conectar la máquina virtual y la máquina HOST. Además, se debe configurar la conectividad a la base de datos donde Cuckoo va a registrar toda la información de los análisis. Por defecto viene para trabajar con una base de datos SQLite, pero es posible configurar una variedad de motores de bases de datos. (Amaya, 2013)

Una de las ventajas que tienen los ficheros de configuración de cuckoo es que vienen muy bien explicados y leyendo la descripción de cada propiedad, es sencillo saber cuáles son los valores que se deben utilizar o si con el valor por defecto es suficiente. (Adastra, 2017)
Machinery define el software de virtualización que se utilizará para arrancar las máquinas virtuales que actuarán como "guest". El valor por defecto es "virtualbox" y es método recomendado para hacer las pruebas. (Adastra, 2017)

Es importante resaltar que dentro de este archivo se configuran los temporizadores para regular los tiempos que duraran los análisis, parámetros muy importantes para que el proceso de análisis de muestras no entre en ciclos infinitos.

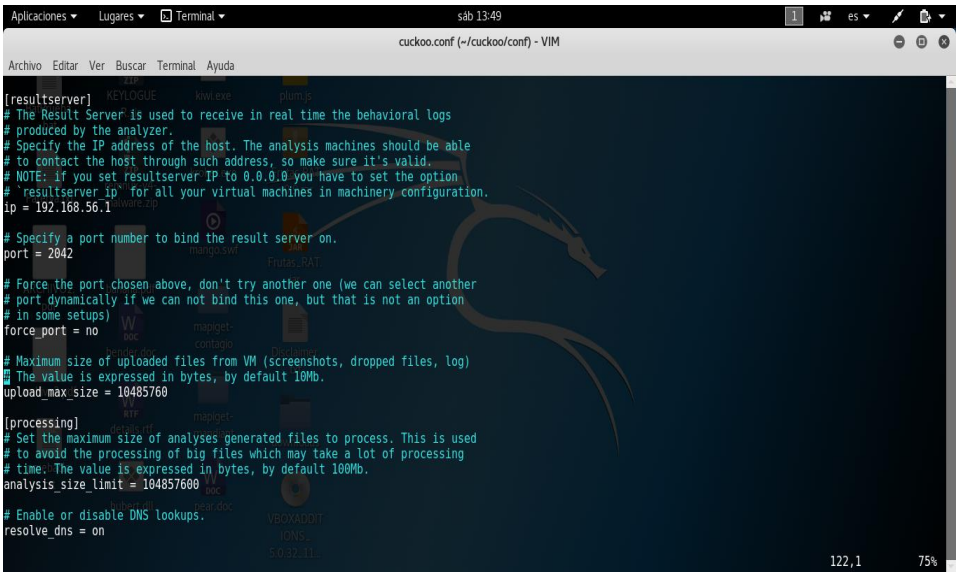


Figura 17-1: Configuración cuckoo.conf
Realizado por: Jonathan Quezada.2017

Otro archivo importante es el que está relacionado con la máquina virtual (GUEST). En este caso, con VirtualBox, por lo tanto, el archivo asociado es virtualbox.conf. de forma análoga al caso del HOST, debe configurar los parámetros de conectividad y las características de nuestra máquina virtual en este punto es muy importante garantizar que la etiqueta que se asigna en este archivo coincida con el nombre que tiene la máquina virtual. (Amaya, 2013)

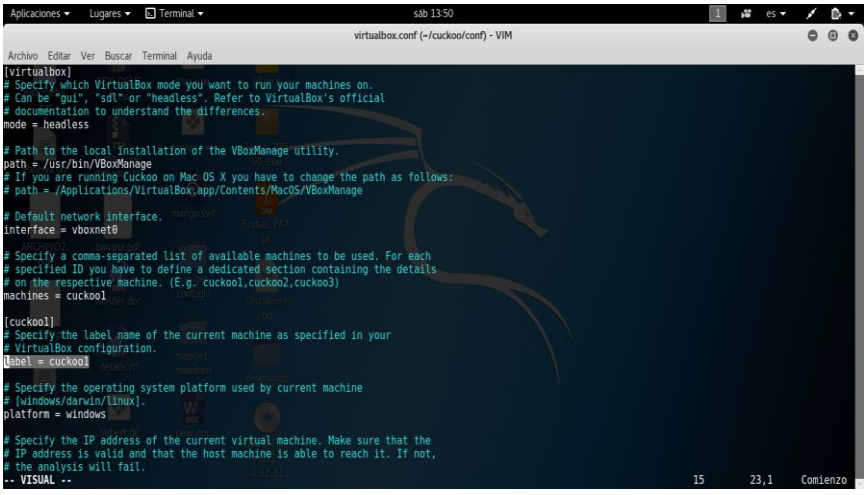


Figura 18-1: Configuración virtualbox.conf
Realizado por: Jonathan Quezada.2017

1.9.2. Configurando la máquina virtual (GUEST)

La configuración de la máquina virtual es un poco más sencilla. Simplemente es necesario instalar Python y copiar el archivo agent.py para que se ejecute cada vez que reinicia la máquina virtual. Una vez que instalada la máquina virtual conviene registrarla en VirtualBox para poder manejarla por línea de comandos. (Amaya, 2013)

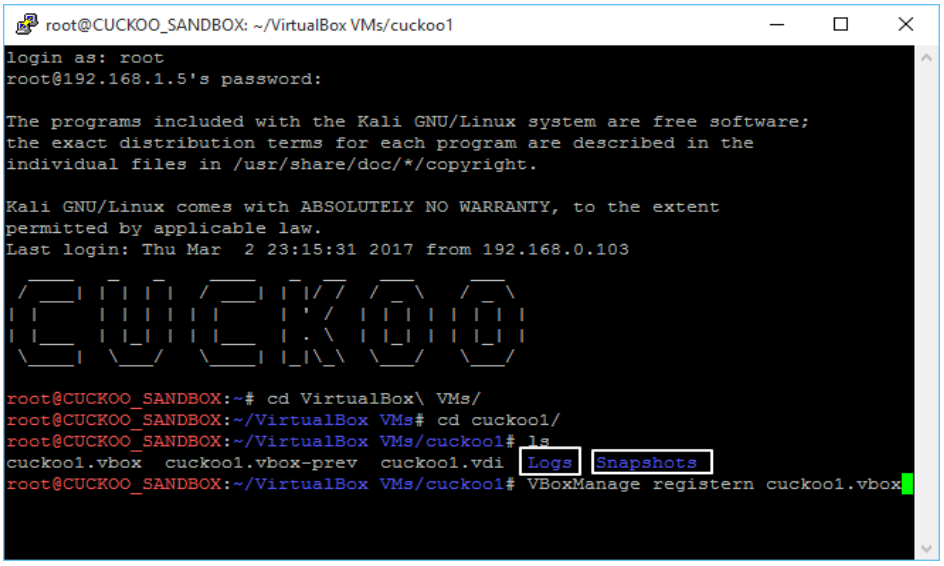


Figura 19-1: Registrar host
Realizado por: Jonathan Quezada.2017

1.9.3. Conectividad entre HOST Y GUEST

Una vez configurado lo relacionado con Cuckoo, solamente resta garantizar la comunicación entre la máquina virtual y nuestra máquina host. Para esto, utilizando la interfaz de la máquina virtual hace un direccionamiento IP de tal forma que ambas máquinas queden en la misma red. (Amaya, 2013)

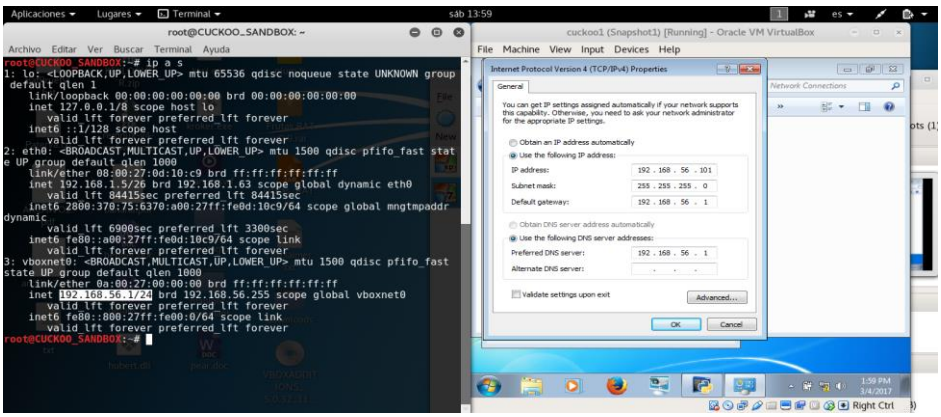


Figura 20-1: Configuración conectividad
Realizado por: Jonathan Quezada.2017

Luego modificar en nuestro sistema Linux el firewall con iptables. Para esto se utilizó la siguiente configuración: (Amaya, 2013)

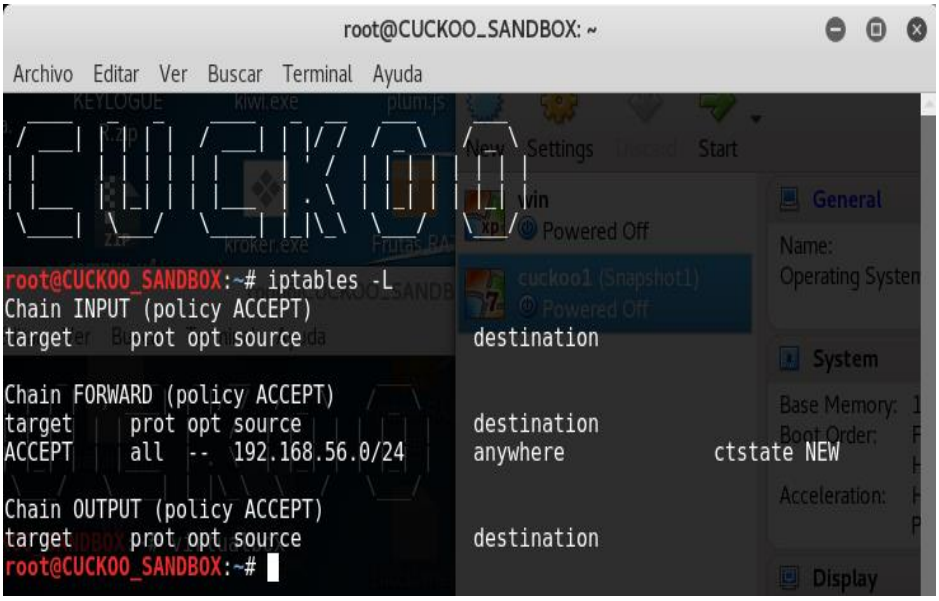


Figura 21-1: Políticas de iptables
Realizado por: Jonathan Quezada.2017

Establecida la conectividad entre el host y la máquina virtual, proceder a ejecutar el servicio de cuckoo sobre nuestro sistema Kali Linux, como se muestra en la figura 22-1.

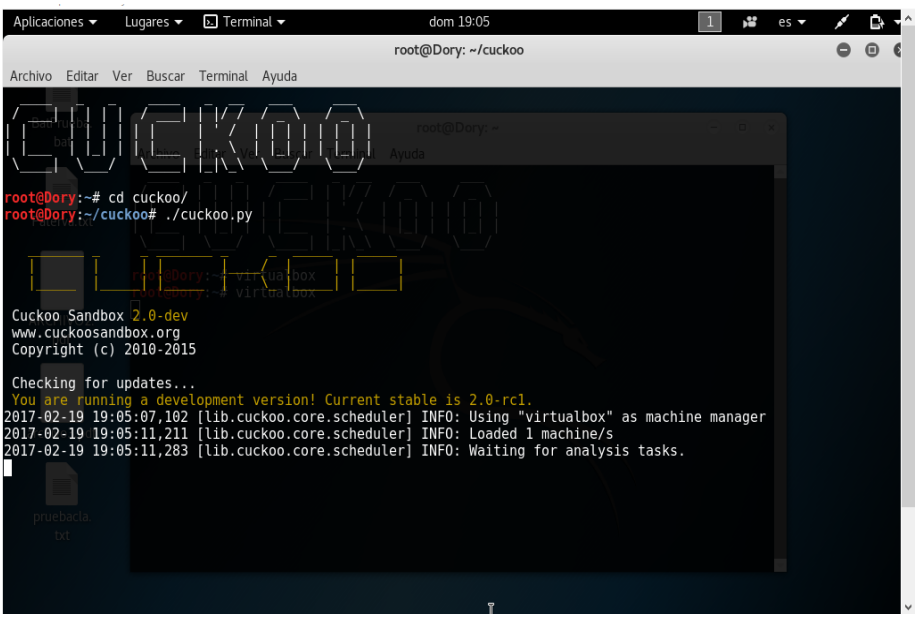


Figura 22-1: Inicio de cuckoo
Realizado por: Jonathan Quezada.2017

CAPÍTULO II

2. CONFIGURACIÓN E IMPLEMENTACIÓN DEL SOFTWARE PARA ANALIZAR EL COMPORTAMIENTO DEL MALWARE EN LOS LABORATORIOS DEL EDIFICIO DE LA FIE.

2.1. Introducción

En el presente capítulo se presenta el diseño de una red de prueba para realizar los ataques de malware y observar las vulnerabilidades de los equipos utilizados para configurar los todos los parámetros de CUCKOO SANDBOX para analizar el comportamiento del malware al momento de infectar los hosts a continuación, se detalla todos los pasos que se debe seguir para la adecuada configuración del software.

2.4. Análisis de la Infraestructura de red de la FIE.

La red de la Facultad de Informática y Electrónica presenta un diseño jerárquico a través de la utilización de Vlans, el que se detalla a continuación en la tabla 2-2.

Tabla 2-2: Ancho de Banda

ANCHO DE BANDA	
Bajada	**2.82 Mbs**
Subida	**2.47 Mbs**

Realizado por: Jonathan Quezada, 2017

2.5. Infraestructura y Equipamiento

Tabla 3-2: Equipamiento FIE

EDIFICIO DE LA FIE	
El edificio de electrónica está estructurado:	• Por 24 puntos de red • 1 Rac • 1 Switch Cisco 2930 • Un Access Point Cisco.
Servicios Web	• Sistema Académico (Oasis) • Educación virtual (elearning) • Evaluación Institucional

	• Matricula • Bibliotecas Virtuales • Servicio Medico • Bienestar Politécnico • Bolsa de Empleo • WebMail

Realizado por: Ing. William Sanchez, 2015

2.6. Infraestructura de la red

Tener presente que la red principal está en la Escuela de Medicina la misma que se conecta al DTIC para luego llegar al RAC que se encuentra ubicado en el tercer piso en el Edificio de la Facultad de Informática y Electrónica como se muestra en la figura 23-2.

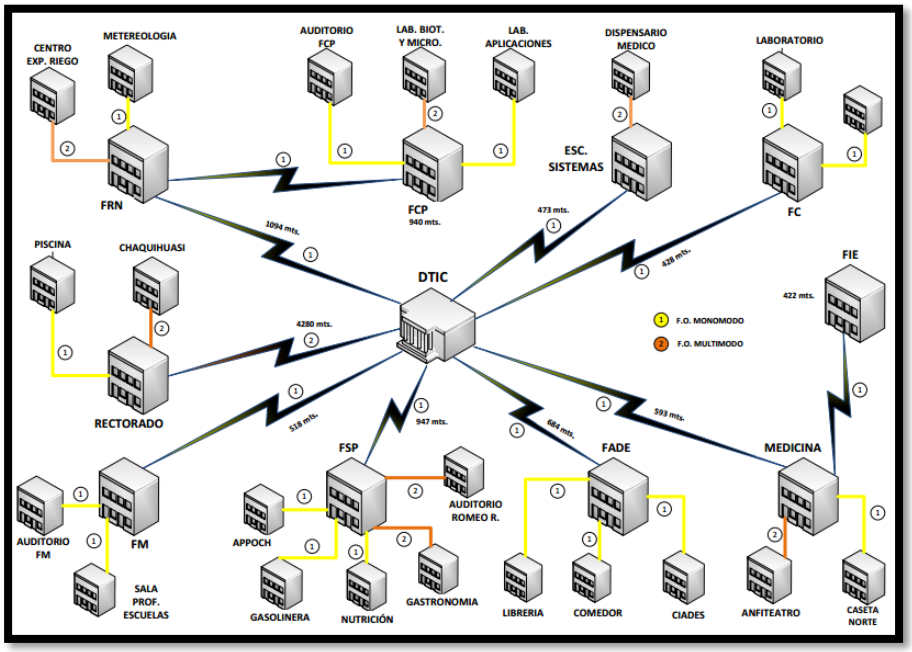


Figura 23-2: Esquema Físico de Red
(FIE - ESPOCH, 2015)

La escuela de Ingeniería Electrónica de la Escuela Superior Politécnica de Chimborazo funciona en dos edificios y sus equipos de red están conectados mediante fibra óptica que llega a un RAC que se encuentra en el tercer piso del edificio de la FIE, para que se distribuya en cada piso, la

misma que cuenta con cuenta con un switch 25650 para proveer de internet a los laboratorios de la Facultad, como se muestra en la Figura 24-2.

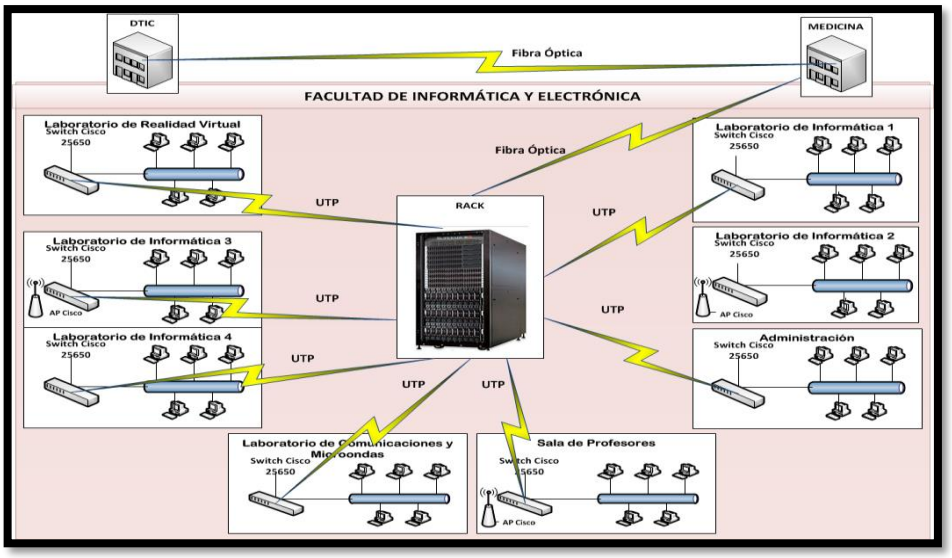


Figura 24-2: Diseño Lógico De La Red Fie - Espoch, 2015
Realizado por: Ing. Juan Carlos Silva

2.7. Dispositivos conectados al Rack de la Facultad de Informática y Electrónica

Las siguientes tablas muestran una descripción de los equipos y la ubicación de los mismos con los que cuenta la FIE.

Tabla 4-2: Equipos FIE

EQUIPOS	UBICACIÓN
1 Router Cisco 3560g	Sala de servidores Edificio de la FIE.
8 switch Cisco 2960 48 puertos	En cada laboratorio
1 switch Cisco 2960, 48 puertos.	En la sala de profesores

Realizado por: Ing. William Sánchez, 2015

2.7.1. Dispositivos en la Zona WIFI

Tabla 5-2: Dispositivos Wireless

1 Access Point Cisco	**AIR-BR1310G-A-k9**
3 Access Point Cisco	**Modelo WRT320N**

Realizado por: Ing. William Sanchez, 20151

2.7.2. Equipos existentes en los Laboratorios de la FIE.

Tabla 6-2: Equipos de los laboratorios

Laboratorios FIE	EQUIPAMIENTO	PUNTOS DE RED
LAB.1	Proyector, pantalla eléctrica 33 PCS I7, 4G RAM , 500gb en disco duro.	**33**
LAB.2	Proyector, pantalla eléctrica 33 PCS I7, 4G RAM, 500gb en disco duro.	**33**
LAB.3	Proyector, pantalla eléctrica 33 PCS I7, 4G RAM , 500gb en disco duro.	**33**
LAB. 4	Proyector, pantalla eléctrica 33 PCS I7, 4G RAM , 500gb en disco duro.	**33**
LAB. REALIDAD VIRTUAL	Proyector, pantalla eléctrica 33 PCS I7, 4G RAM , 500gb en disco duro.	**33**
LAB. COMUNICACIONES Y MICROONDAS	Proyector, pantalla eléctrica 19 PCS I7, 4G RAM , 500gb en disco duro.	**19**

Realizado por: Jose luis Silva, 2015

2.8. Identificación de Vulnerabilidades en los equipos

Para la identificación de vulnerabilidades se utilizó la herramienta Nessus en su versión Home libre. Se utilizó los hosts del laboratorio de la FIE. Para identificar los puertos que se encuentren abiertos, para dirigir los ataques de infección de malware a los equipos de prueba.

En la figura 25-2 se muestra la pantalla de logueo de la herramienta Nessus.

Figura 25-2: Loguin NESSUS
Realizado por: Jonathan Quezada, 2017

Una vez logueados ingresar a Basic Network Scan como se muestra en la figura 26-2

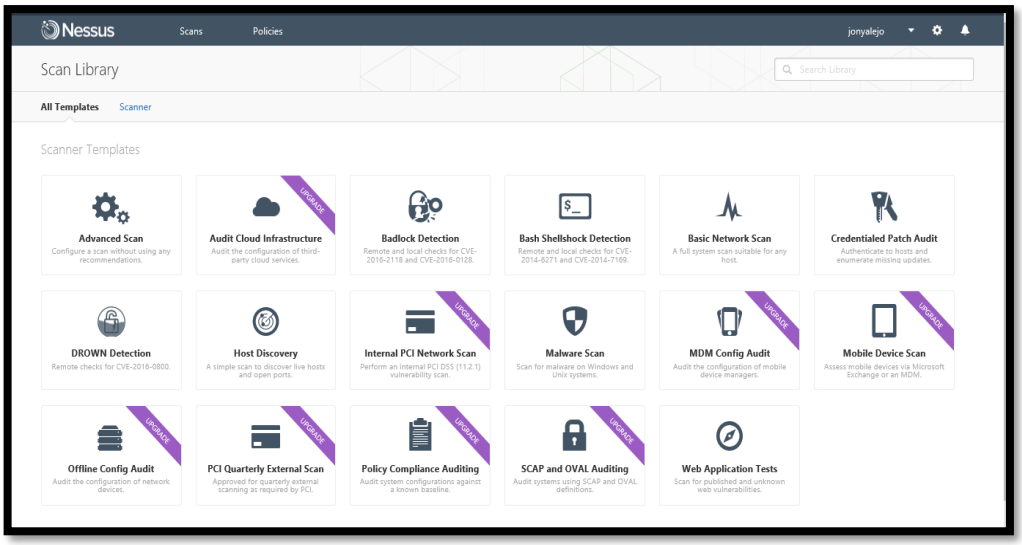


Figura 26-2: Escaneo de host
Realizado por: Jonathan Quezada, 2017

A continuación, se configura los parámetros necesarios para iniciar el escaneo de los hosts que se encuentren conectados a la red local del edificio de la FIE, y así escanear la red completa como se muestra en la figura 27-2.

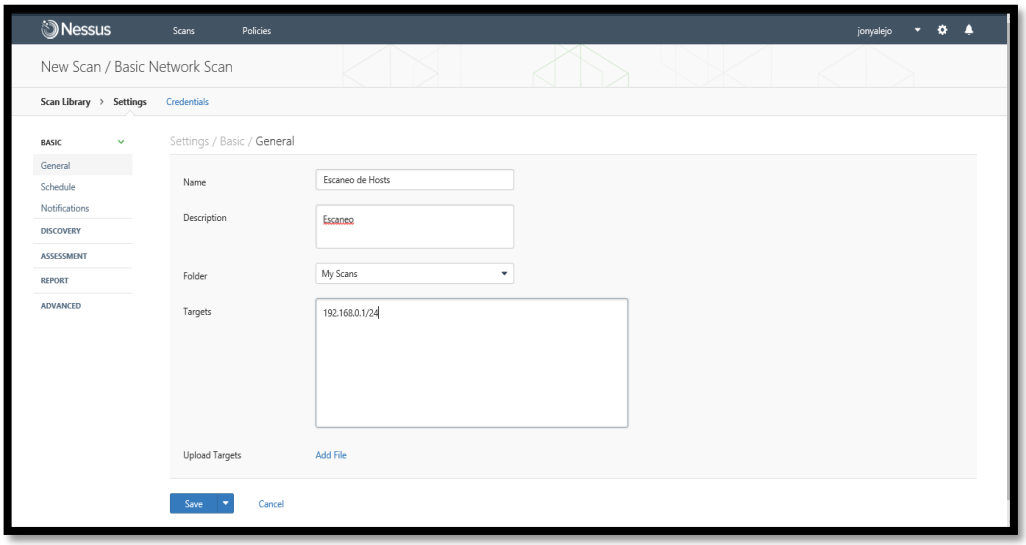


Figura 27-2: Identificar el nombre del análisis
Realizado por: Jonathan Quezada, 2017

Una vez analizada las vulnerabilidades se muestran los resultados obtenidos mediante el escaneo como se muestra en la figura 28-2.

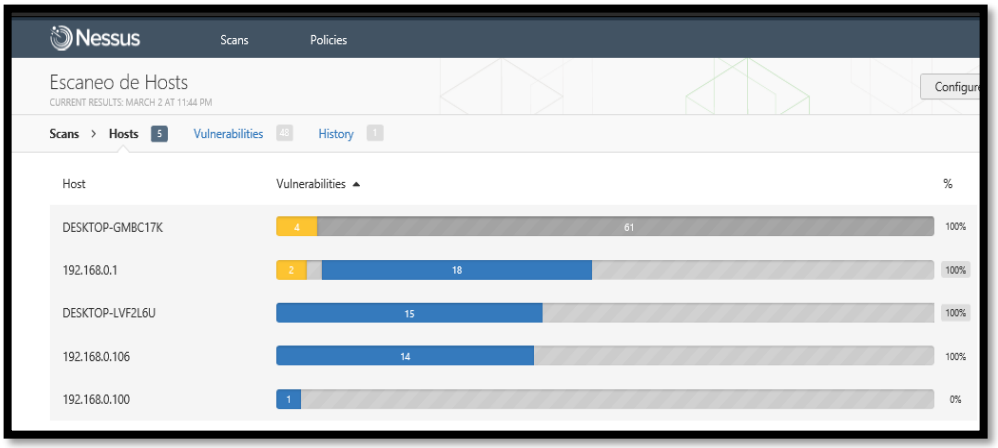


Figura 28-2: Analizando de vulnerabilidades
Realizado por: Jonathan Quezada, 2017

Una vez finalizado el análisis se observa el resumen de las vulnerabilidades encontradas y se puede obtener un informe técnico y ejecutivo como se muestra en la figura 29-2. Los resultados de color rojo representan las amenazas de nivel alto, los resultados de color amarillo representan

amenazas de nivel medio, los de color verde representan las amenazas de nivel bajo, y la de color azul son de carácter informativo.

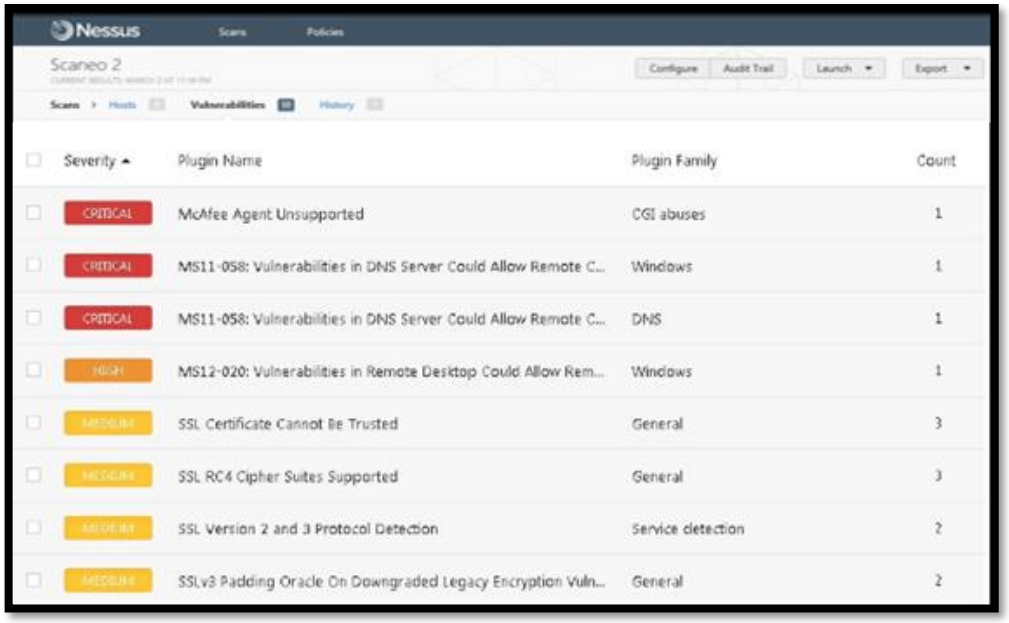


Figura 29-2: Vulnerabilidades encontradas
Realizado por: Jonathan Quezada, 2017

Se encontró un total de 98 vulnerabilidades en la red como se observa en la figura 30-2.

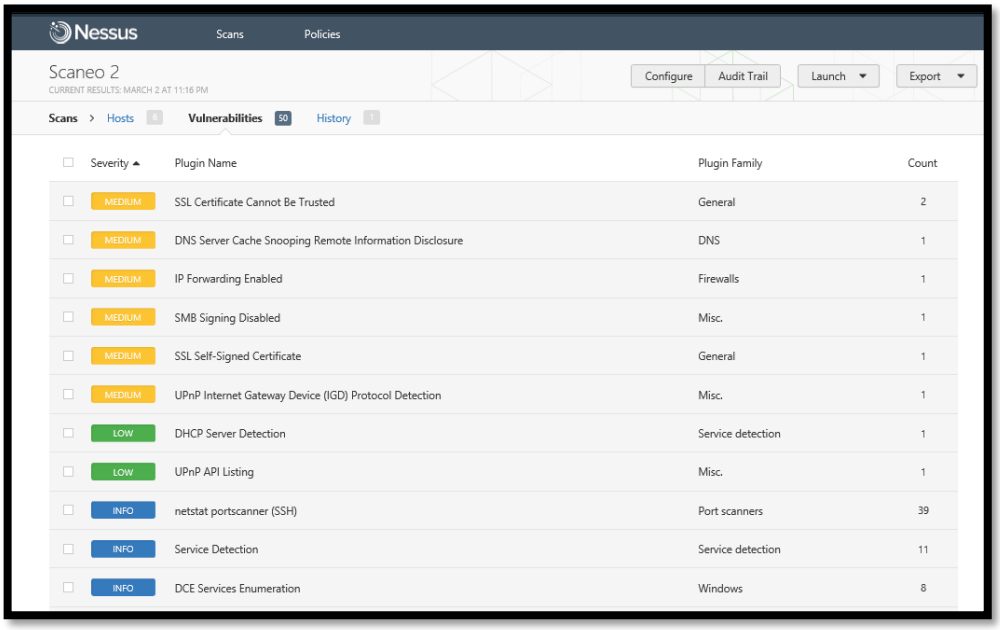


Figura 30-2: Detalles de Vulnerabilidades
Realizado por: Jonathan Quezada, 2017

En la figura 31-2 se muestra el detalle del número de escaneo que se realiza a la red en donde se visualiza detalles como tiempo de inicio y fin de escaneo tiempo de ejecución del escaneo la ip de la red entre otros.

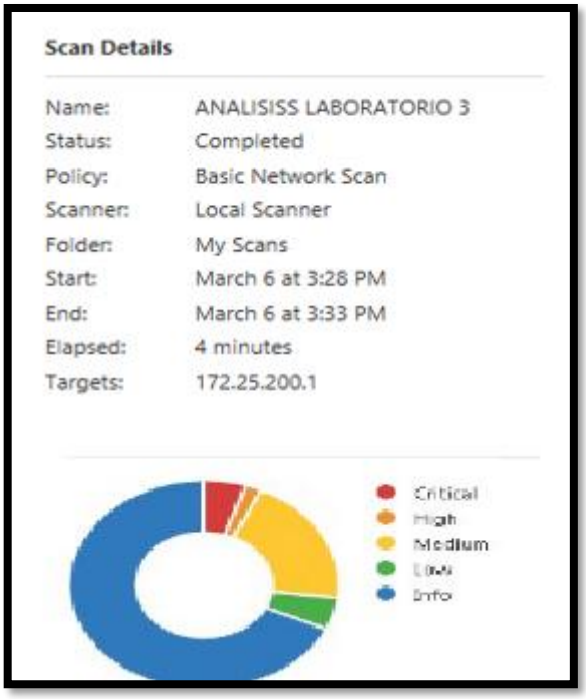


Figura 31-2: Informe de vulnerabilidades
Realizado por: Jonathan Quezada, 2017

La siguiente pantalla muestra la descripción de la solución que debe realizar el usuario para contrarrestar la vulnerabilidad del host, como se muestra en la figura 32-2.

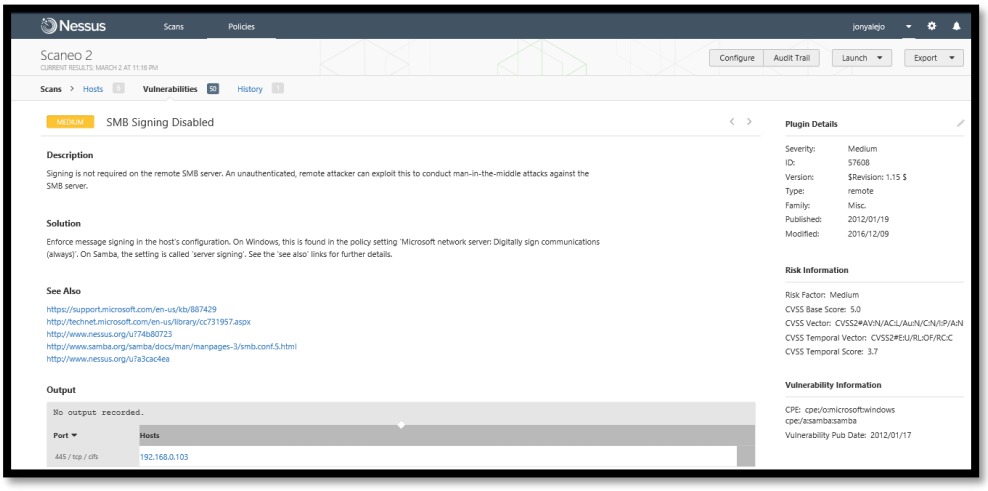


Figura 32-2: Detalles de la amenaza
Realizado por: Jonathan Quezada, 2017

Finalmente, una vez realizado el análisis se observa que la mayoría de vulnerabilidades encontradas son del servicio SMB de Windows. Es muy importante analizar los resultados que nos arroja el sistema para elegir el tipo de exploit(malware) que se va a ejecutar. Para guardar

el reporte generado por la herramienta Nessus se exporta como archivo tipo HTML. El contenido de este archivo se visualizará el resultado final tal cual y como se mostró en la figura X-2. Para realizar este proceso en la opción report se debe elegir la opción Executive Summary como se muestra en la figura 33-2.

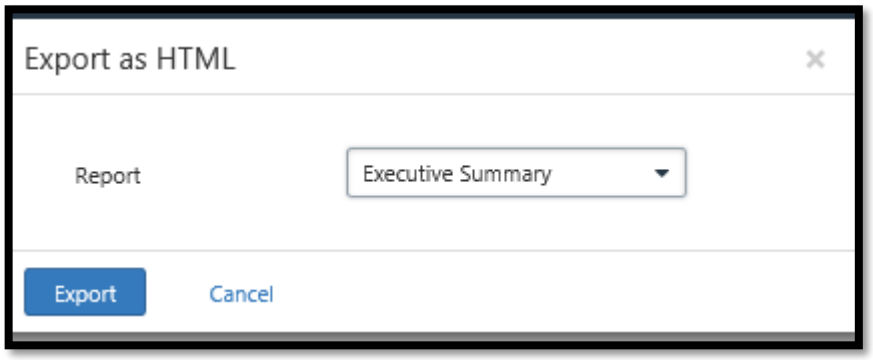


Figura 33-2: Reporte final
Realizado por: Jonathan Quezada, 2017

2.9. Diseño e implementación de la red de prueba para realizar los ataques informáticos.

Se procedió a implementar la red de prueba para realizar ataques informáticos y medir el nivel de vulnerabilidad de los equipos, para lograr una mejor comprensión sobre la infección de malwares y de esta manera identificar las amenazas lógicas que se pueden dar en la red.

Como se observa a continuación el esquema de red de prueba no existen ningún tipo de software que nos brinden algún tipo de seguridad para proteger la integridad tanto de los docentes como de los estudiantes, como se muestra en la figura 34-2.

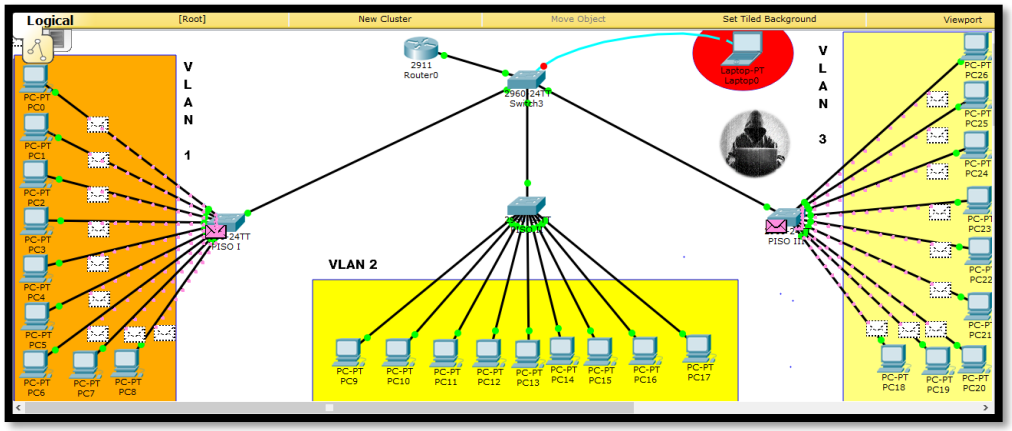


Figura 34-2: Esquema red de prueba
Realizado por: Jonathan Quezada, 2017

Para lo cual se probó los siguientes comandos:

SSH: Protocolo que permite que otro usuario inicie sesión interactiva en un equipo remoto para ingresar comandos por medio de una conexión segura como se visualiza en la figura 35-2.

Figura 35-2: Configurando la red de prueba
Realizado por: Jonathan Quezada, 2017

Se prueba contraseñas para conectarse por medio de ssh y establecer una conexión simulando ser el administrador de la red como se observa en la figura 36-2.

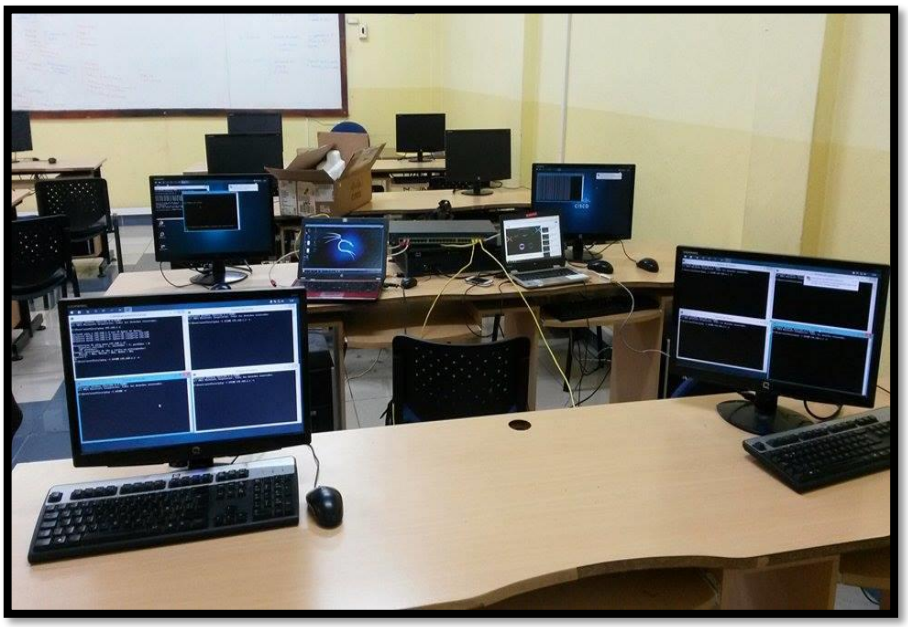


Figura 36-2: Probando las vulnerabilidades de los equipos
Realizado por: Jonathan Quezada, 2017

Se probó posibles combinaciones de contraseñas creadas por el administrador de la red para crear un diccionario que realice un ataque de fuerza bruta para aprovechar que el puerto 22 se encuentra abierto, como se muestra en la figura 37-2.

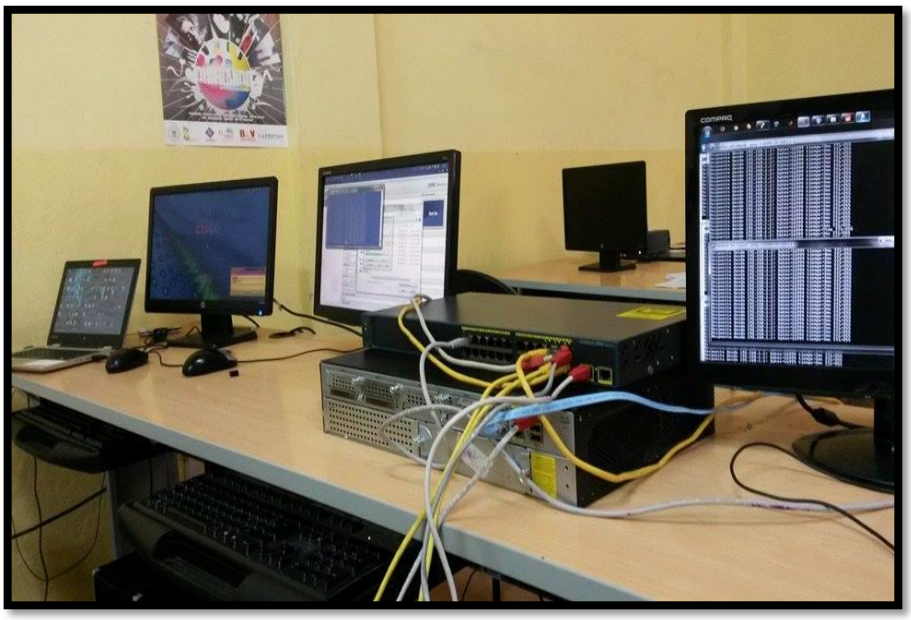


Figura 37-2: Ataques DDOS
Realizado por: Jonathan Quezada, 2017

Se procedió a realizar la conexión ssh y se observa que este canal de es más seguro que telnet, como se observa en la figura 38-2.

Figura 38-2: Probando ssh
Realizado por: Jonathan Quezada, 2017

Finalmente, se consiguió el objetivo planteado que es infiltrarnos en la victima y con el comando ls –l se observó los archivos y directorios además se visualiza los permisos de cada archivo y así se modifica según nuestras intenciones como se observa en la figura 39-2.

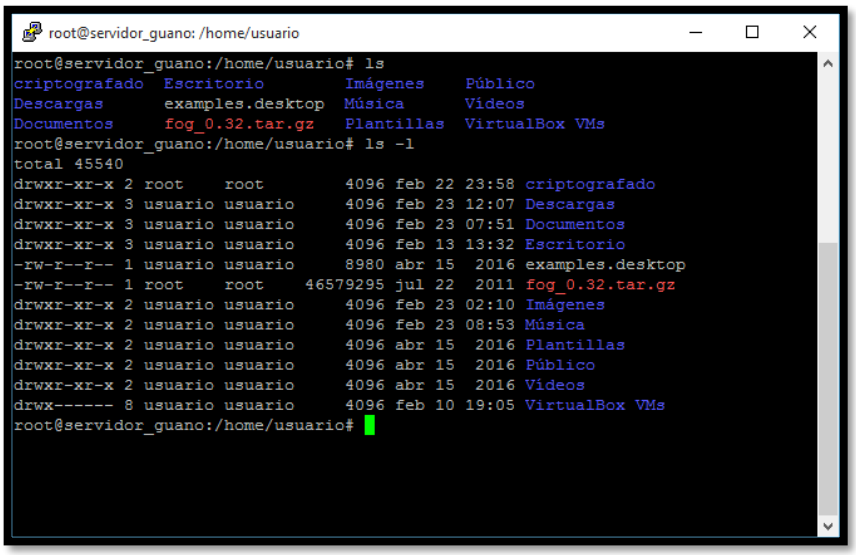


Figura 39-2: Visualización de la información del equipo vulnerado
Realizado por: Jonathan Quezada, 2017

El siguiente paso es crear una carpeta con el nombre HACKEADO que es alerta para el usuario en donde puede contener la información correspondiente de que su equipo ha sido vulnerado como se muestra en la figura 40-2.

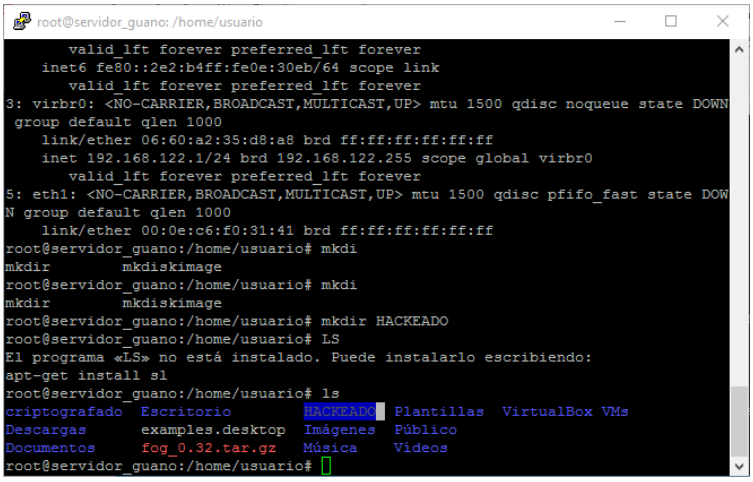


Figura 40-2: Crear la carpeta de alerta
Realizado por: Jonathan Quezada, 2017

Queda en potestad del atacante informático realizar las pruebas o ataques maliciosos que traten de dañar o robar información importante que comprometa la integridad de la persona dueña del host vulnerado.

2.10. Análisis de tráfico en la red local

Para el análisis se configuró previamente el puerto SPAN local que soporta una sesión SPAN enteramente dentro de un switch; todos los puertos de origen o VLAN y los puertos de destino están en el mismo conmutador de la pila. El tráfico local SPAN es copia de uno o más puertos de origen en cualquier VLAN o de una o más VLAN a un puerto de destino para su análisis. Por ejemplo, en la Figura 40-2 , todo el tráfico de los puertos 1,2,3 (puertos de origen) se refleja al puerto 5 (el puerto de destino). Un analizador de red en el puerto 5 recibe todo el tráfico de red desde los puertos 1,2,3 sin la necesidad de estar conectado físicamente a los puertos, como se muestra en la figura 41-2.

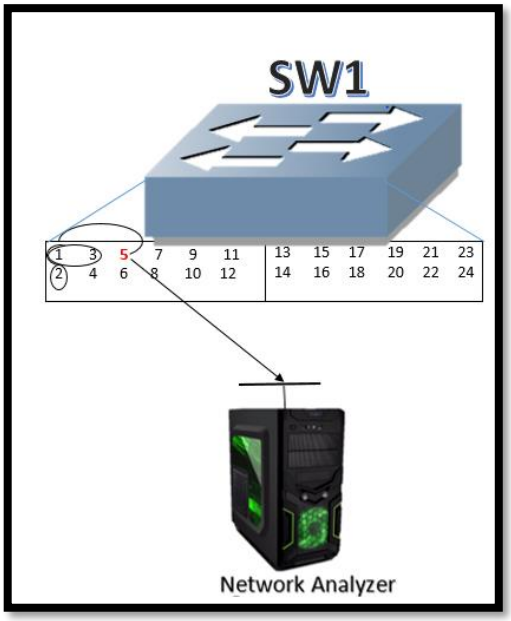


Figura 41-2: Span Local
Realizado por: Jonathan Quezada, 2017

2.11. Análisis de tráfico en la red con WIRESHARK

En primera instancia, para realizar un análisis dinámico de un código malicioso se procede a infectar un sistema en un entorno controlado. Por lo general, se recurre a una máquina virtual. De esta forma, es posible ejecutar Wireshark y seleccionar la interfaz de red de la máquina virtual para comenzar a capturar los paquetes de red. A continuación, puede visualizarse en la figura 42-2 de cómo se realiza la mencionada tarea. (CATOIRA, 2013)

Se seleccionó la interfaz de red que se configuro como Span para analizar el tráfico de los laboratorios de la FIE como se presenta en la figura 42-2.

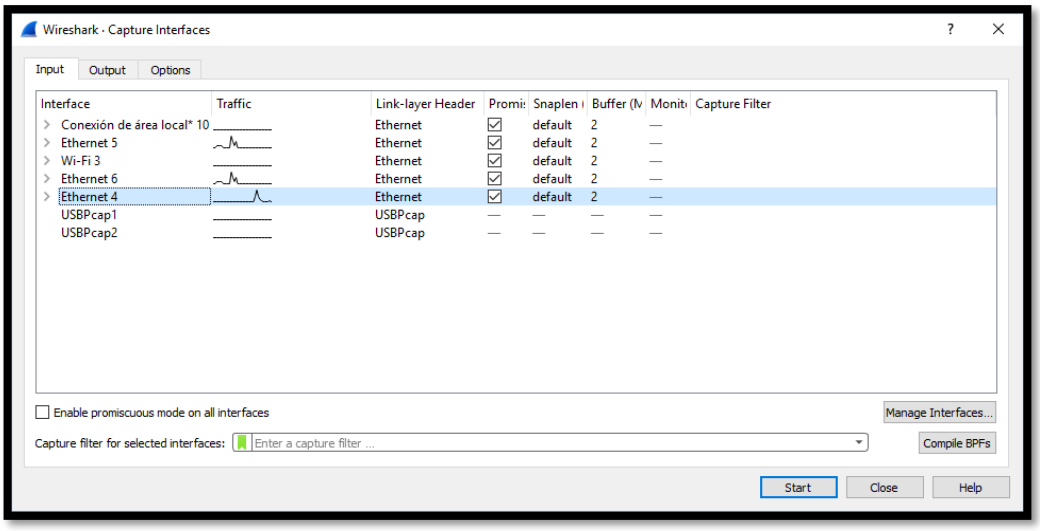


Figura 42-2: Selección interfaz de red
Realizado por: Jonathan Quezada, 2017

Se capturo paquetes específicos para el análisis del tráfico de la red local del edificio de la FIE, estudiando específicamente el protocolo SSDP ya que este protocolo presenta una vulnerabilidad en el sistema operativo que puede ser utilizado por los hackers para ataques informáticos.

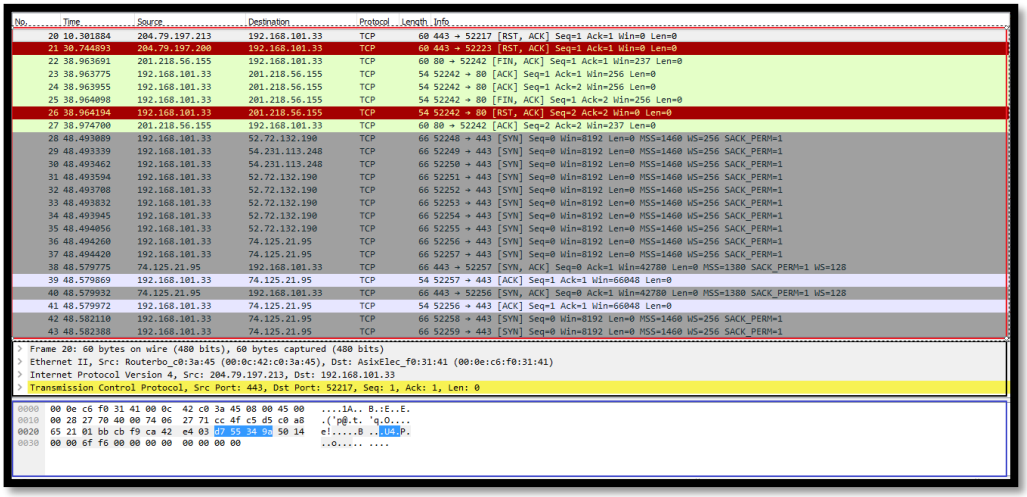


Figura 43-2: Capturando Paquetes
Realizado por: Jonathan Quezada, 2017

Se filtró los protocolos que nos interesan para nuestro análisis de tráfico para el caso es el protocolo SSDP, como se observa en la figura 44-2.

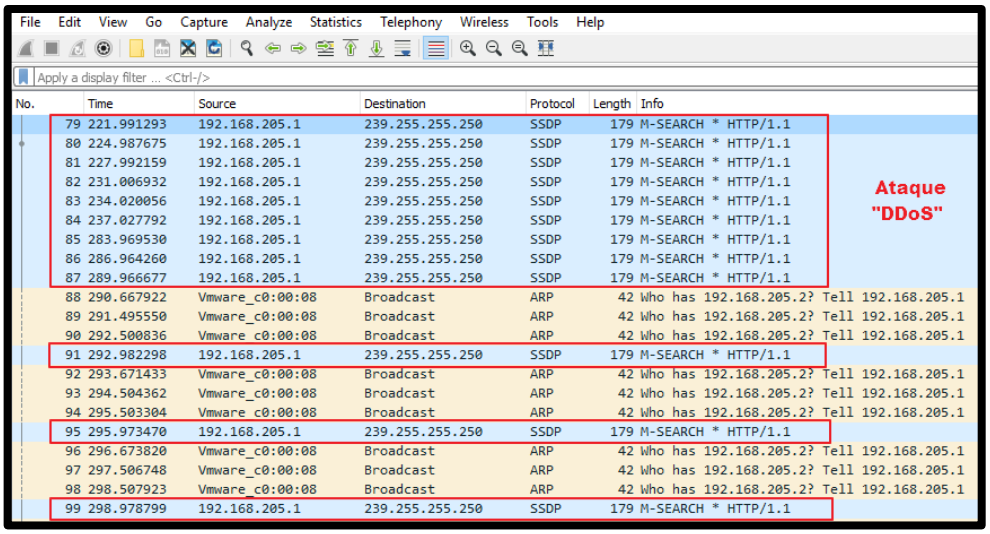


Figura 44-2: Análisis de protocolos específicos
Realizado por: Jonathan Quezada, 2017

2.12. Análisis de vulnerabilidades utilizando hacking ético.

Para el análisis de vulnerabilidades por medio de hacking ético, se realizó a través de fases de ingreso a la red, los mismos que se detallan a continuación.

2.10.1. FOOTPRINTING

1) Cambiar la MAC-ADDRESS para evitar cualquier rastreo de nuestra verdadera dirección física como se observo en la figura 45-2.

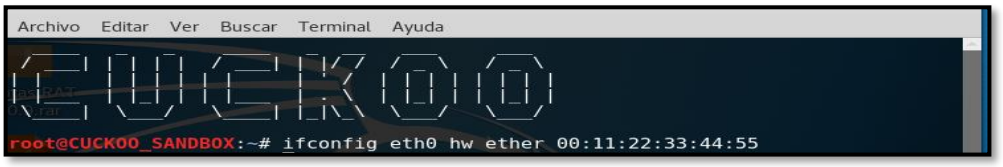


Figura 45-2: : Cambio de Mac Address
Realizado por: Jonathan Quezada, 2017

2) Verificar que se realizó el cambio de MAC ADDRESS, como se visualiza en la figura 46-2.

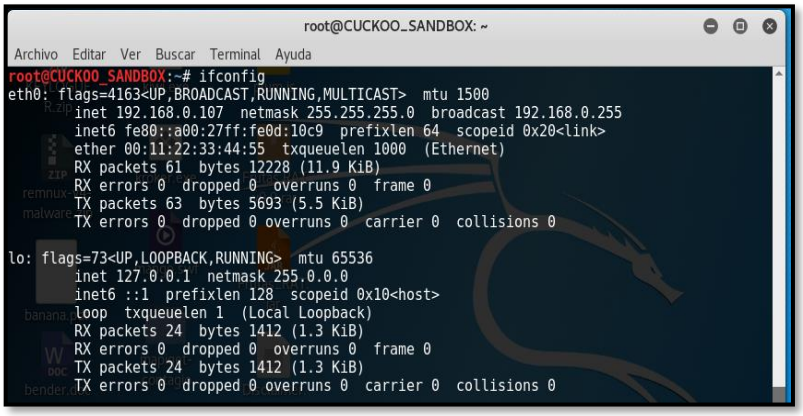


Figura 46-2: Comprobar el cambio de Mac Address
Realizado por: Jonathan Quezada, 2017

2.12.2. Conectividad con las direcciones Gateway de las Vlans.

Para visualizar la conectividad con las direcciones Gateway de las vlans se ejecuta el comando ping más la ip como se muestra en la figura 47-2, tanto a la dirección de la puerta de enlace como a la dirección física del ordenador.

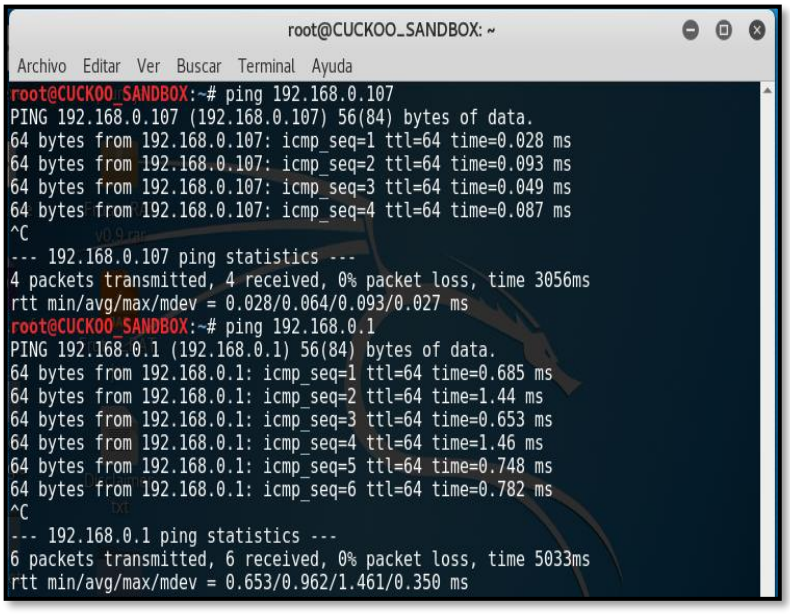


Figura 47-2: Prueba ICMP
Realizado por: Jonathan Quezada, 2017

3) Se ejecuta el comando traceroute mas la dirección web que se quiere atacar como por ejemplo www.elerning.edu.ec, como se observa en la figura 48-2.

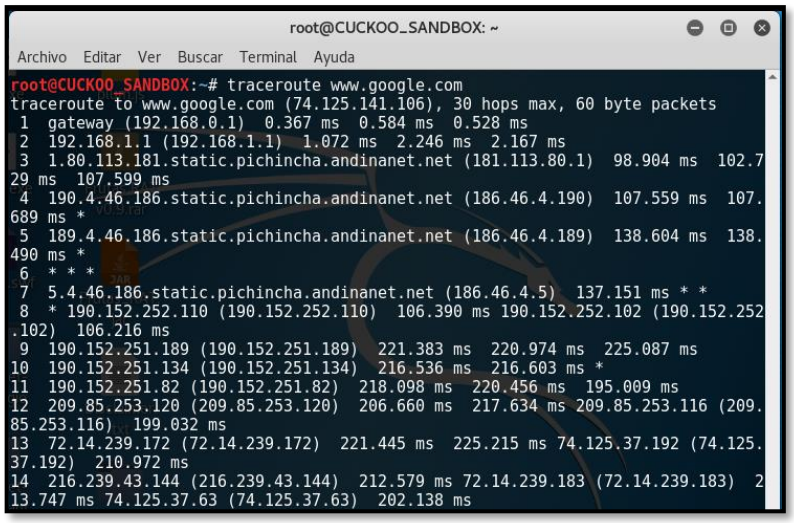


Figura 48-2: Prueba de traceroute
Realizado por: Jonathan Quezada, 2017

2.12.3. Scanning

Se utiliza el programa nmap, para visualizar los equipos que están activos en la red, así como sus direcciones físicas y los puertos que se encuentran abiertos. Las instrucciones que se necesitan para ejecutar se muestran en la figura 49-2.

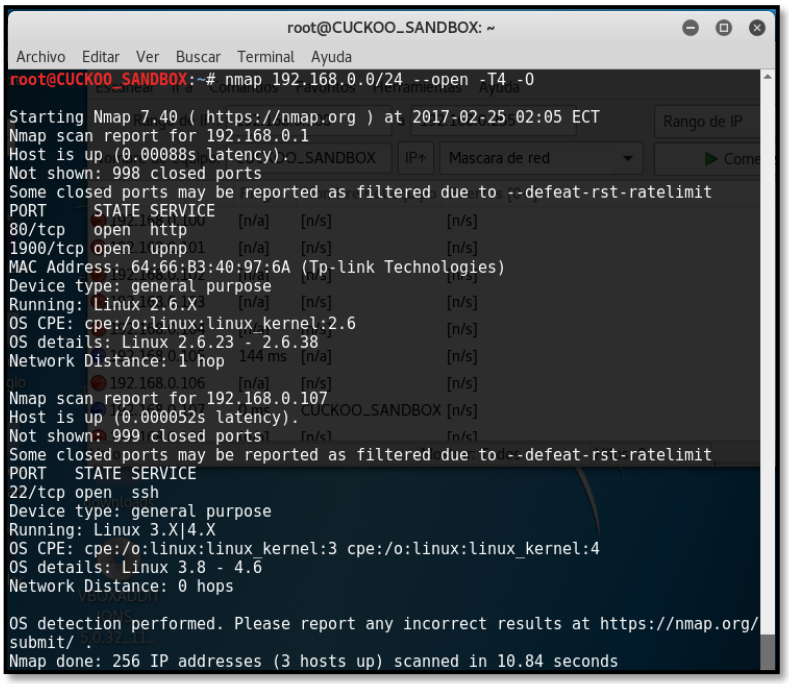


Figura 49-2: Puertos abiertos de los hosts
Realizado por: Jonathan Quezada, 2017

2.13. Instalación del malware.

El malware que se utilizó para este ataque de infección de malware es el denominado Frutas_RAT.jar, como se muestra en la figura 50-2.

Figura 50-2: Inicio de frutas Rat
Realizado por: Jonathan Quezada, 2017

2.14. Configurar los parámetros del malware.

Una vez ubicados en la interfaz gráfica de Frutas_RAT se crea un nuevo servidor con la ip del atacante, para que se re direccione todo lo que realiza la victima a la ip de Kali Linux, como se observa en la figura 51-2.

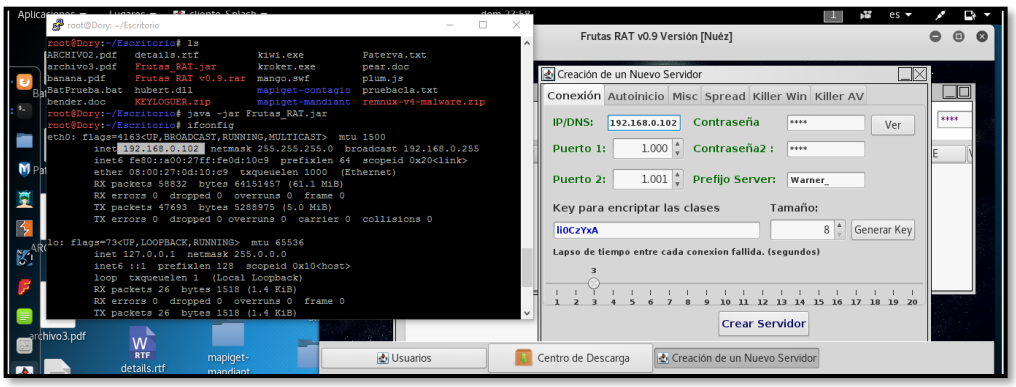


Figura 51-2: Verificando la ip del atacante
Realizado por: Jonathan Quezada, 2017

Para configurar el malware, se debe realizar una ingeniería social a la víctima para asegurar que el archivo se ejecute y luego proceder a propagar a la víctima y así obtener nuestro objetivo que es observar lo que hace en el transcurso de la utilización del equipo logrando utilizar la información para atacar y vulnerar el equipo. Para lo cual se siguen los siguientes pasos:

1) En esta ventana activar todas las casillas y asignar un nombre, este nombre será el del archivo que se envió a la víctima de ataque con la extensión .jar en este caso el nuestro fue PRUEBAS DE RUTEO II.jar, como se observa en la figura 52-2.

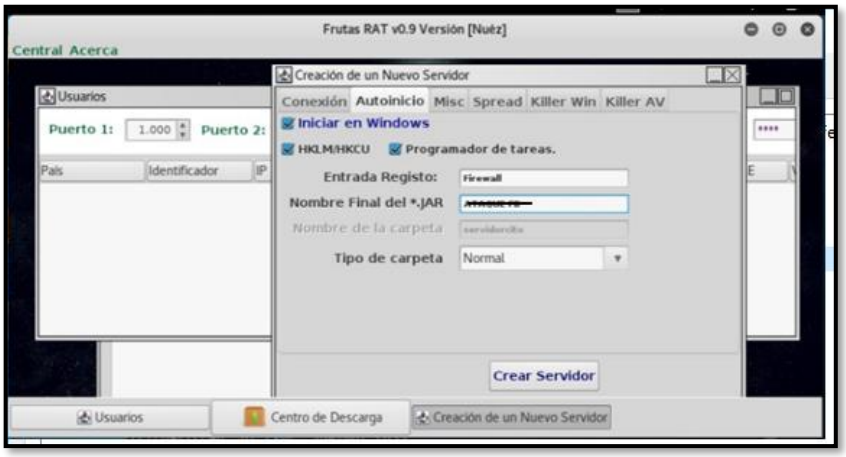


Figura 52-2: Creación del malware
Realizado por: Jonathan Quezada, 2017

2) En esta opción marcar las tareas que se van a bloquear para que la ejecución de nuestro malware proceda con éxito, como se observa se la figura 53-2.

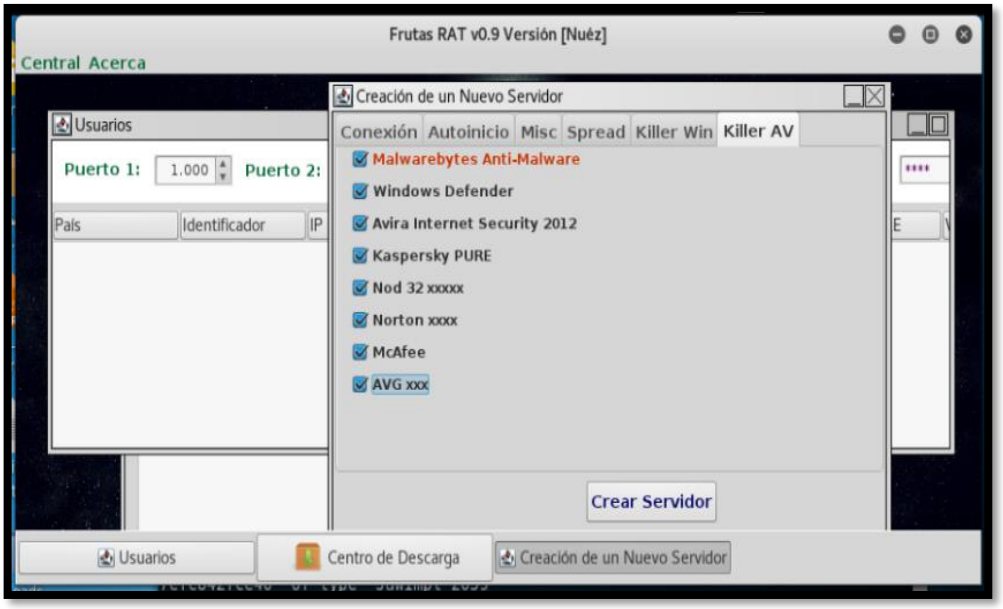


Figura 53-2: Deshabilitar todos los antivirus posibles de la victima
Realizado por: Jonathan Quezada, 2017

2.15. Propagación del Malware.

Utilizando ingeniería social enviar el virus y persuadir a nuestros compañeros de la escuela de Telecomunicaciones para que se descarguen el archivo, en este caso se utilizó el correo

electrónico como se muestra en la figura 54-2, para propagar el malware así cierto grupo de compañeros accedan al archivo.

Figura 54-2: Propagar el malware
Realizado por: Jonathan Quezada, 2017

2.16. Infección del Malware

Para realizar el ataque el usuario debe ejecutar el archivo denominado PRUEBAS RUTEO II logrando así vulnerar el equipo, como se observa en la figura, como se observa en la figura 55-2.

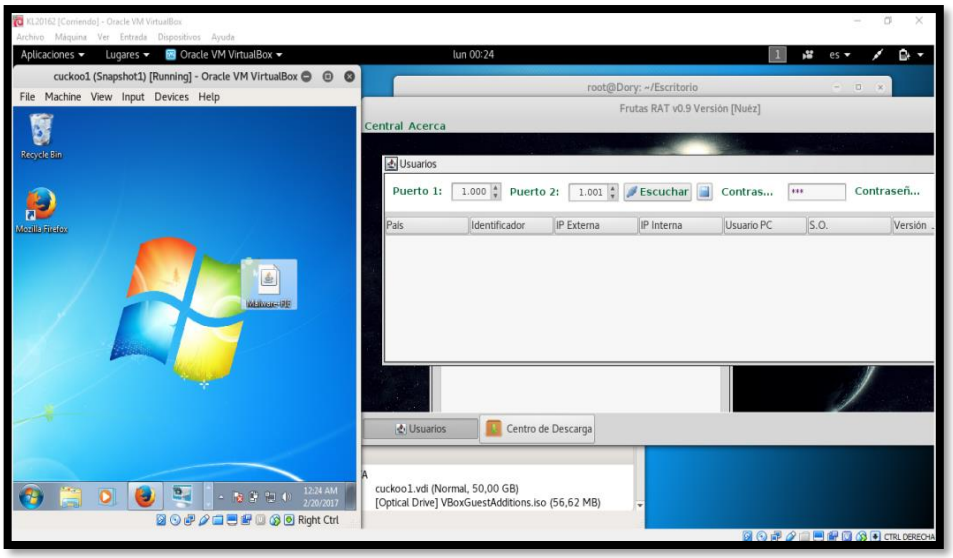


Figura 55-2: Ejecutar el malware en la maquina victima
Realizado por: Jonathan Quezada, 2017

3) Se reporta el número de descargas del malware que se propago utilizando ingeniería social los hosts de las víctimas, como se observa en la figura 56-2.

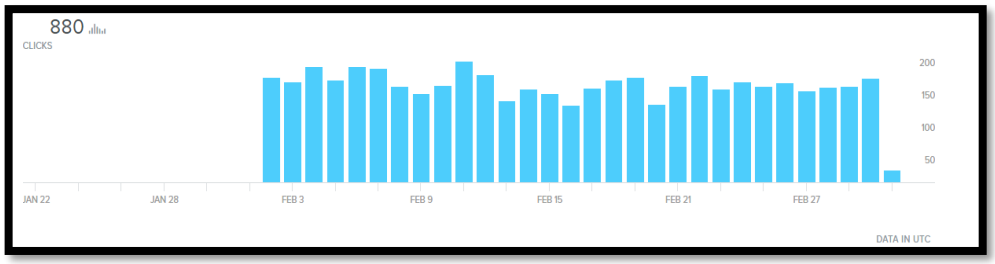


Figura 56-2: Numero de descargas de las victimas
Realizado por: Jonathan Quezada, 2017

4) La figura 56-2 muestra los porcentajes de las descargas realizadas por las víctimas se observa que el 46.5% de personas infectadas que realizaron la descarga por medio de la aplicación Messenger de Facebook y el 40% lo realizaron directamente desde su ordenador desde la plataforma Facebook, y el 7,4% y 3,5% respectivamente ejecutaron la descarga, como se observa en la figura 57-2.

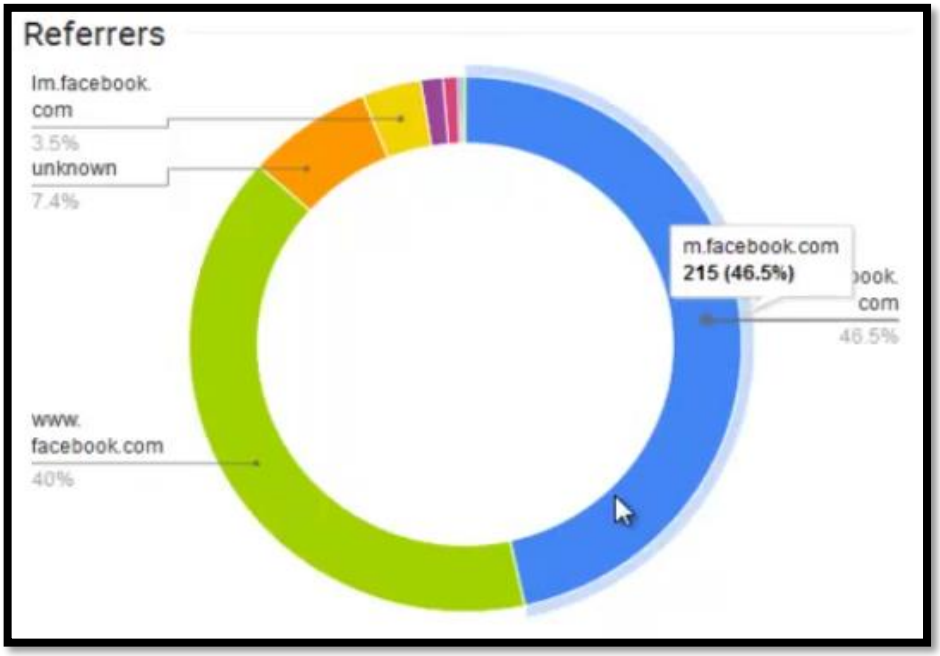


Figura 57-2: Lugares desde donde accedieron
Realizado por: Jonathan Quezada, 2017

5) Tipo de navegadores de donde accedieron a la descarga del malware, como se observa en la figura 58-2.

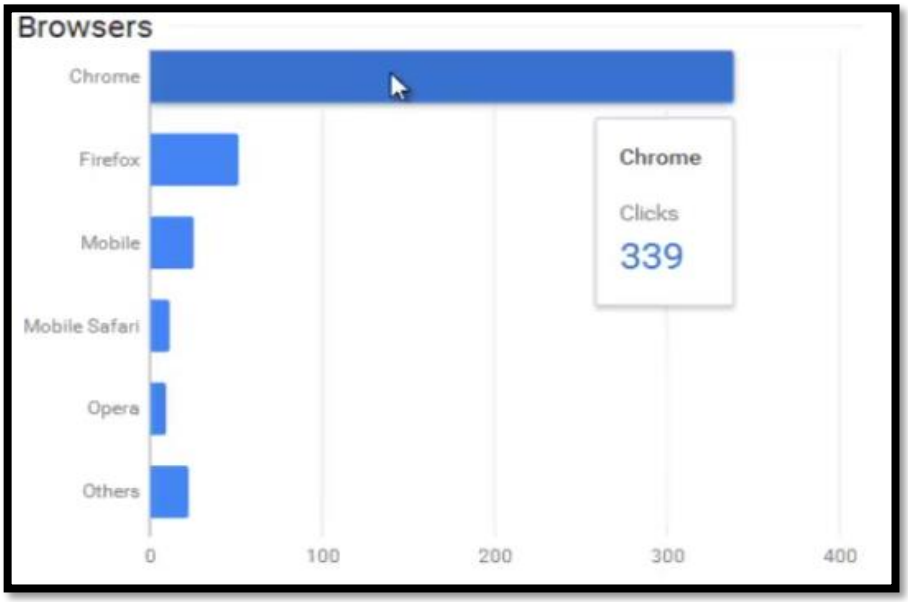


Figura 58-2: Navegadores de donde accedieron
Realizado por: Jonathan Quezada, 2017

2.11. Evaluación del ataque

Una vez ejecutado el malware se puede elegir los múltiples ataques al equipo vulnerado, entre los cuales tener: Mensajes de Texto, abrir URL, ataque D. o. S, entre otros para el caso de estudio se utilizó capturas de pantalla como se muestra en la figura 59-2.

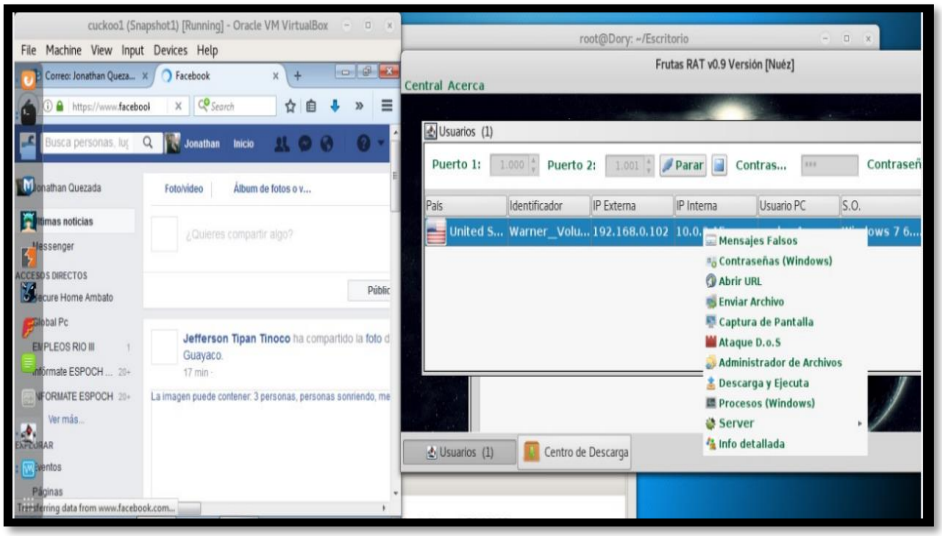


Figura 59-2: Opciones de ataque
Realizado por: Jonathan Quezada, 2017

Una vez que se tiene el control de todo el equipo víctima es posible refinar nuestro ataque capturando las pantallas del equipo infectado y observar lo que la víctima realiza en su computadora e incluso tomando el control total del usuario infectado, para el caso de estudio observar que la víctima ingreso a Facebook donde se pudo visualizar todos los procesos y movimientos que realizo, como se observa en la figura 60-2.

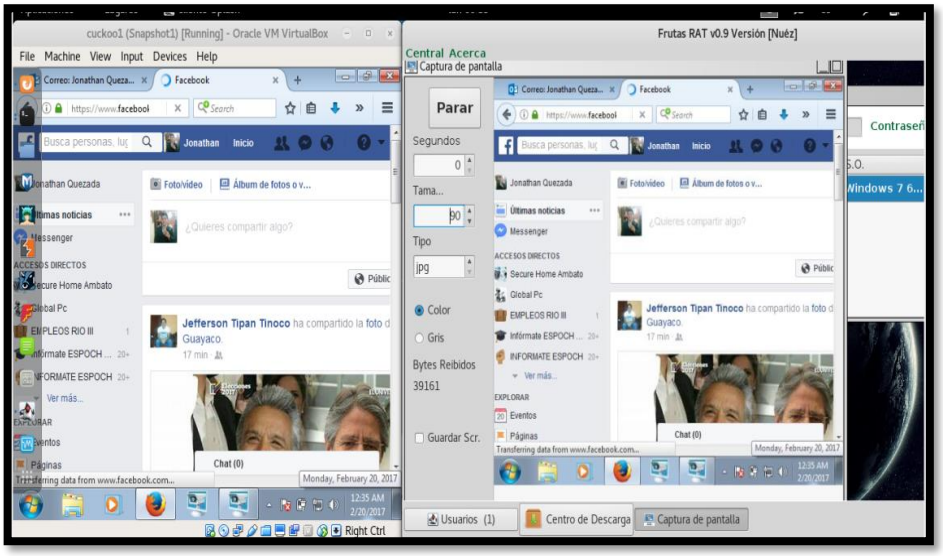


Figura 60-2: Espiando el comportamiento de la víctima
Realizado por: Jonathan Quezada, 2017

CAPÍTULO III

3. EVALUACIÓN DE RESULTADOS

3.1. Introducción

El siguiente capítulo está constituido en dos partes, en la primera se va a tomar en cuenta las políticas de seguridad para optimizar la confidencialidad de los datos tanto de los docentes como de los estudiantes de la facultad y la segunda determina la capacidad de comparar el comportamiento de los diferentes tipos de malware y analizar los informes generados en el sistema de análisis.

3.2. Implementación de la topología de red simulada.

Una vez realizado el análisis del estado de la red actual, se procede a tomar las muestras de malware más relevantes que serán analizadas en la siguiente topología de red simulada para llevar a cabo el análisis de comportamiento del malware, como se observa en el gráfico 1-3.

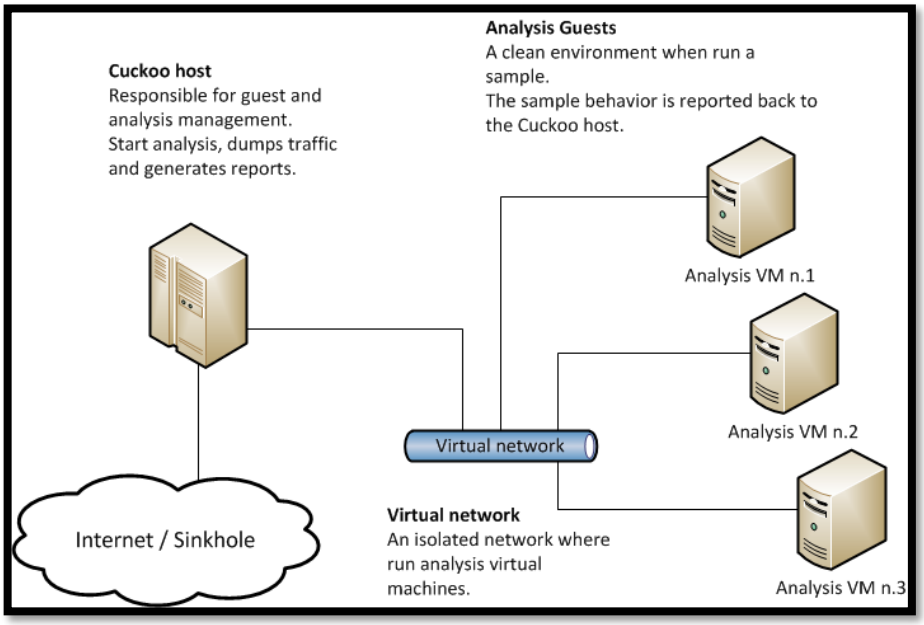


Gráfico 1-3: Cuckoo Sandbox
Realizado por: (Foundation, 2015)

En la tabla 9-2 se encuentran las especificaciones de los equipos virtuales a utilizarse tanto como atacante y víctima.

Tabla 7-2: Referencia de los equipos virtuales

Sistema Operativo	Disco Duro	Memoria
Windows Professional	40 GB	1536 MB
Windows Ultimate	40 GB	1536 MB
Kali Linux 2016.2	60 GB	4419 MB

Realizado por: Jonathan Quezada, 2017

3.2.1. Resultados obtenidos con la herramienta Cuckoo Sandbox

Se implementó la herramienta de software libre Cuckoo Sandbox para analizar el comportamiento del malware en una red local e implementar políticas de seguridad que mitiguen los riesgos. Se enviará un archivo .rar que contiene un keylogger para que se instale en la máquina víctima, y así poder observar el comportamiento del malware por medio de Cuckoo Sandbox.

Las características del equipo Windows que se utilizó como víctima en un entorno virtualizado son exactamente las mismas que poseen las máquinas de los laboratorios de la FIE. Para lo cual se siguió los siguientes pasos.

1) Instalar la herramienta Kali y luego configurar el analizador de comportamiento de malware como se muestra en el gráfico 2-3.

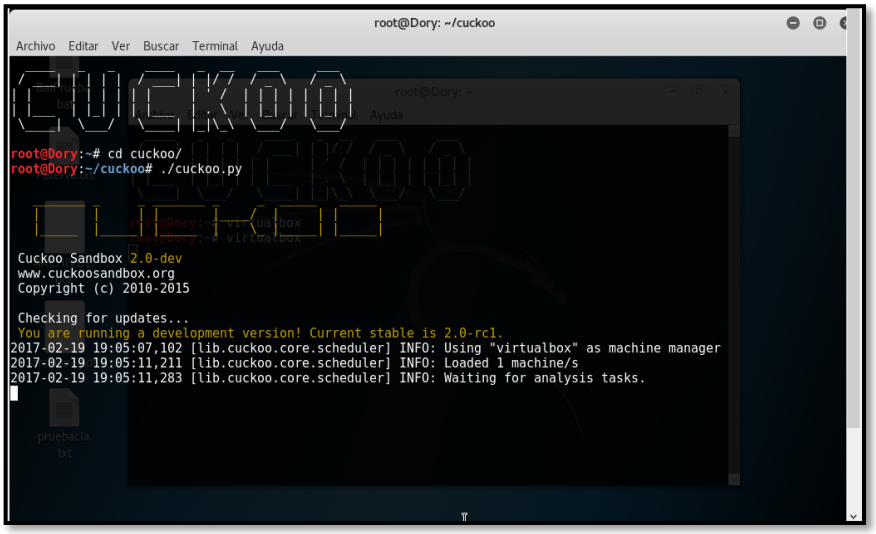


Gráfico 2-3: Iniciando el análisis
Realizado por: Jonathan Quezada, 2017

2) Declarar políticas de seguridad de acuerdo a los requerimientos de la host víctima, en la figura 57-3 muestra un reporte de que todos los antivirus conocidos no está instalados en los equipos donde se realiza el análisis de malware, como se observa en el gráfico 3-3.

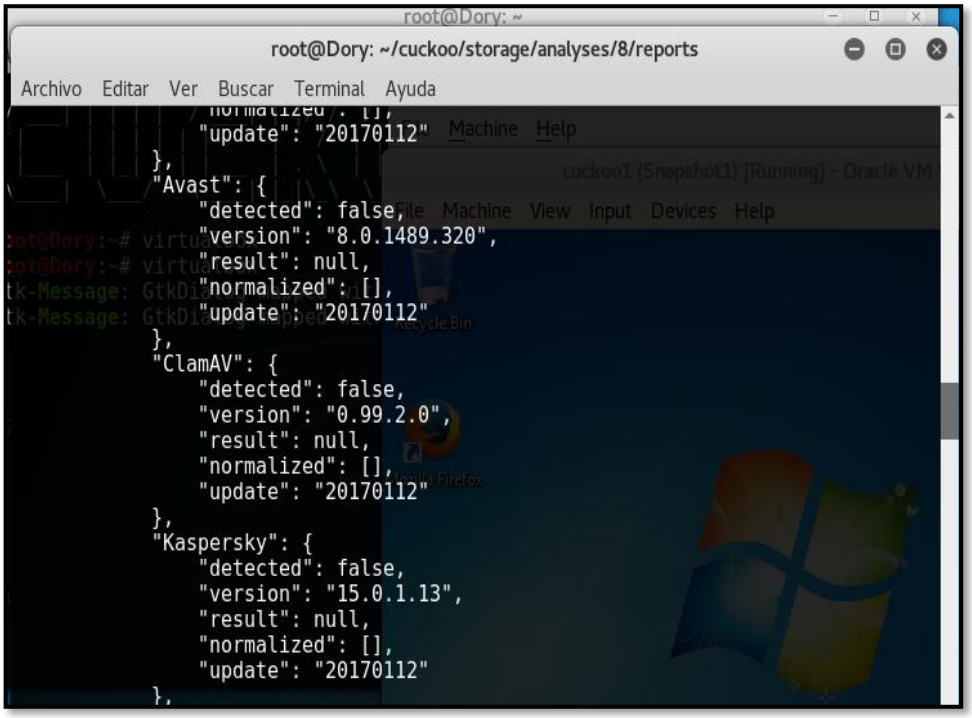


Gráfico 3-3: Verificando antivirus del equipo víctima
Realizado por: Jonathan Quezada, 2017

3) Declarar las políticas de seguridad:

- Implementar la herramienta Malwarebyts para mitigar la propagación de malwares en los equipos de los laboratorios de la FIE.

- Crear dos usuarios con diferentes privilegios en los equipos para minimizar riesgos.

- Cerrar el puerto 21 para evitar un ataque de metasploit.

- Mantener los sistemas operativos actualizados, y no es recomendable tener desactivadas las actualizaciones automáticas.

4) Se escribe la siguiente ruta **/cd/root/cuckoo/storage/analyses/8** para observa el comportamiento del malware al momento de la propagación en los hosts simulados como se muestra en el gráfico 4-3.

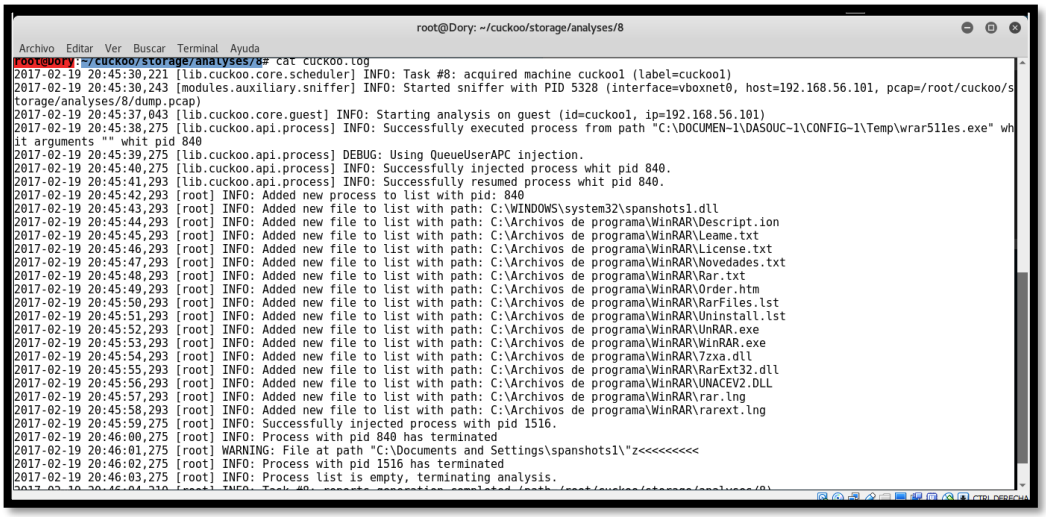


Gráfico 4-3: Análisis del comportamiento del malware en la maquina víctima
Realizado por: Jonathan Quezada, 2017

3.2.1. Dionaea

Es un honeypot de baja interacción (emula servicios), sucesor del proyecto Nepenthes, escrito en C pero que además incorpora Python como lenguaje de scripting, utiliza la biblioteca Libemu para emular la ejecución de instrucciones Intel x86 y detectar shellcodes. Además, cuenta con soporte para IPv6 y protocolo TLS, que fue utilizado en el proyecto para recolectar muestras de malware y posterior análisis.
Instalación de la herramienta Dionaea.

1) El primer paso consiste en instalar la herramienta dionea mediante una instrucción de comandos como se muestra a continuación.

```
# git clone git://git.carnivore.it/dionaea.git dionaea \
```

2) Con el siguiente comando se instaló el servicio de Dionaea.

```
# /opt/dionaea/bin/dionaea -D -r /opt/dionaea -w /opt/dionaea
  -p /opt/dionaea/var/dionaea.pid -l all,-debug -L '*'
```

3) Finalmente visualiza nos en la siguiente ruta donde se van almacenar todas las muestras capturadas.

```
# /opt/dionaea/var/dionaea/binaries/
```

3.3. Análisis inicial de la Red.

Tabla 8-3: Evaluación inicial

Situación inicial RED FIE				
	Validación de la licencia	Activo	Tiempo de evasión del sistema de seguridad	Nivel de Seguridad
ANTIVIRUS	No	No	O segundos	Ninguno
DEPP-FREE	Activo	Si	1 mes	Medio
FIREWALL	Open Source	Si	25 minutos	Bajo

Realizado por: Jonathan Quezada, 2017

El gráfico 5-3 muestra los datos de validación de seguridad inicial que cuentan los equipos de los laboratorios

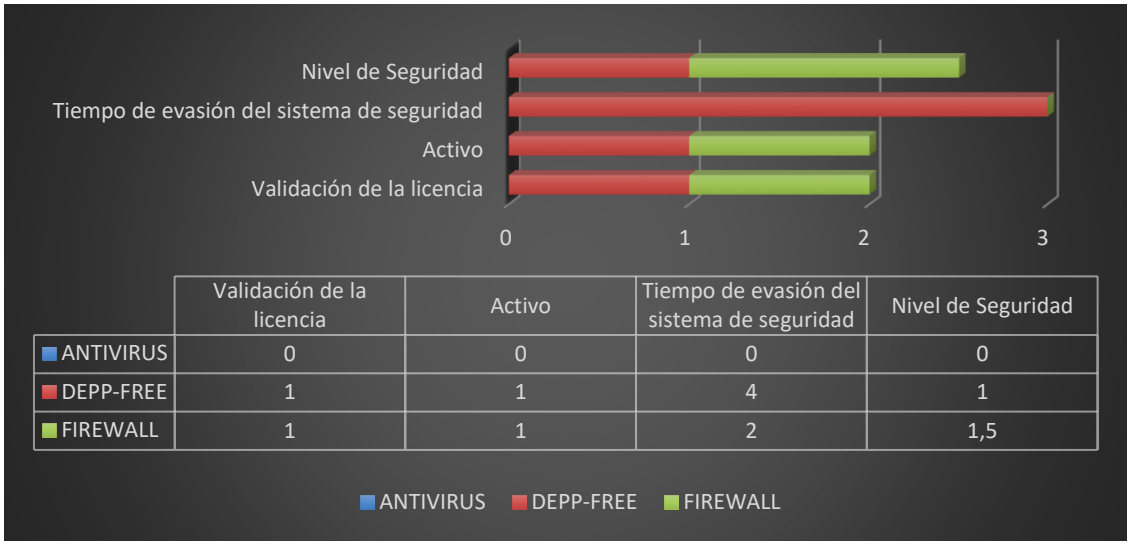


Gráfico 5-3: Seguridad Inicial
Realizado por: Jonathan Quezada, 2017

3.2.1. Vulnerabilidades de la red.

Los principales incidentes que se detectaron en el año 2016 fueron los Escaners y malwares. En la gráfica se observó que los Malware alcanzaron el 40% de incidentes durante el periodo académico Abril – Agosto analizados en el sistema académico elerning 2016 y que los últimos

lugares ocupan los gusanos con tan solo un incidente en el periodo académico que corresponde al 6% en ataque

Tabla 9-3: Diferentes ataques

Incidentes encontrados		
Reporte	Ataques(%)	Dificultad
Malware	40	3
Escaner	32	2
DDoS	25	1
Gusano	6	1
Otros	15	0

Realizado por: Jonathan Quezada, 2017

El gráfico 6-3 muestra los tipos de malware con el nivel de dificultad en los equipos.

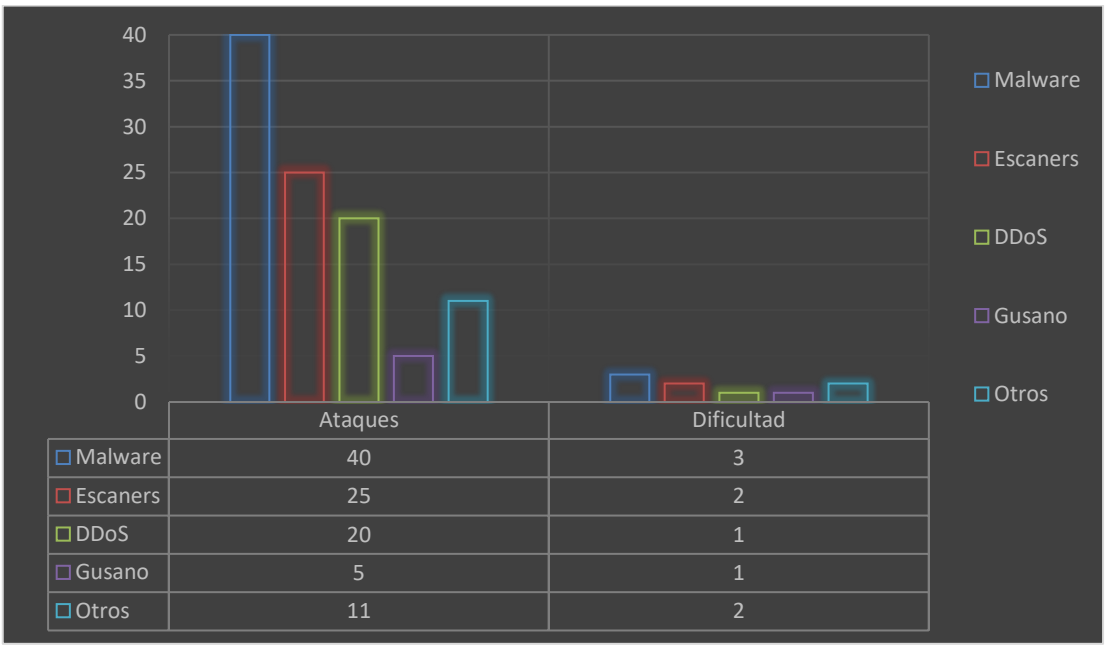


Gráfico 6-3: Tipos de Malwares
Realizado por: Jonathan Quezada, 2017

3.4. Efectividad de los ataques

Se determinó que la efectividad de la infección a los diferentes hosts de los laboratorios de la FIE, fueron de 80% ya que nos facilitó el trabajo que no tengan un antimalware instalado para que detecte los archivos maliciosos y los bloquee como se observa en la tabla 10-3.

Tabla 10-3: Efectividad de ataques

	Frutas Rat	Keylogger	Ataque Man in The Middle
Posibilidad	7	9	8
Dificultad	6	2	3
Daño	8	9	8

Realizado por: Jonathan Quezada, 2017

El gráfico 7-3 se muestra los tipos de malware con el nivel de dificultad en los equipos.

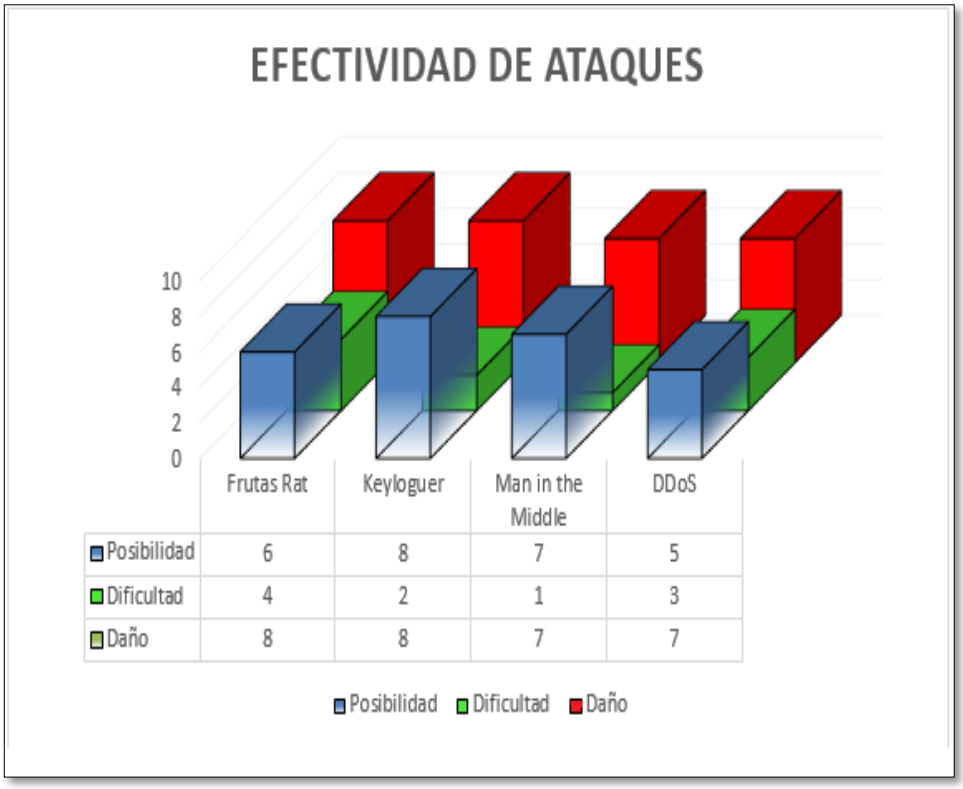


Gráfico 7-3: Efectividad de ataques
Realizado por: Jonathan Quezada, 2017

3.5. Implementación un antimalware en los equipos de la FIE

El tiempo en el que se puede detectar un malware depende específicamente de la cantidad de información almacenada en los equipos, además se clasifica a los malwares por su peligrosidad y

se los establece en un rango del 1 – 10 utilizando las recomendaciones del estándar de seguridad 802.1Q, como se muestra en la tabla 11-3.

Tabla 11-3: Análisis Malwarebytes

	AMENAZA	TIPO	ACCIÓN	NIVEL DE RIESGO
Malwarebytes	PUP.Optional.MindSpark	Clave de registro	Modifica registro del sistema	Baja
	Trojan.BHO	Software Malicioso	Seguimiento de las pulsaciones del teclado	Alta
	Backdoor.0Access	Software Malicioso	Control remoto del host	Medio
	Unknown Rootkit Driver	Software Malicioso	Corrompe el funcionamiento del Sistema Operativo	Alta

Realizado por: Jonathan Quezada, 2017

En la figura 67-2 se muestra los niveles de peligrosidad de los malwares más comunes encontrados en el equipo

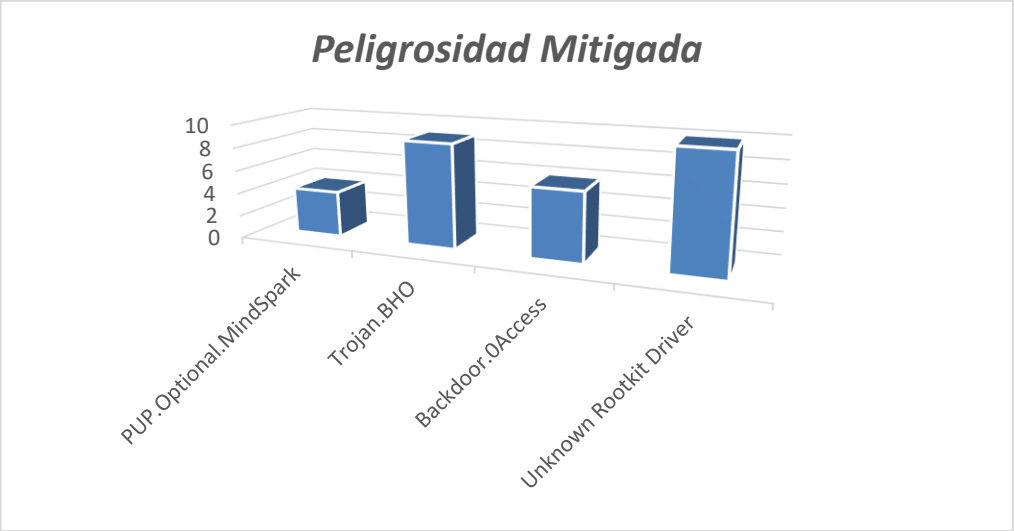


Gráfico 8-3: Peligrosidad de los Malwares
Realizado por: Jonathan Quezada, 2017

3.6. Situación Actual

Una vez realizadas las pruebas y ataques informáticos en los laboratorios de la FIE, se evaluó los resultados obtenidos de la siguiente manera:

- Las computadoras del laboratorio no detectan los malware debido a que no tienen instalados un antivirus con licencia siendo más vulnerables a ataques informáticos y maliciosos. En la tabla 12-3 se muestra un análisis comparativo de la situación inicial

actual realizados en los equipos de la FIE. Se determina un rango de seguridad de 1-10 establecido según el estándar de seguridad 802.1Q.

Tabla 12-3: Comparativas Sistemas de Seguridad

Herramientas de Seguridad	Rango de seguridad establecido	Nivel de Seguridad inicial	Nivel de Seguridad actual
Sin Antimalware	1	1	10
Deep Freeze	5	5	5
Malwarebytes	0	0	10

Realizado por: Jonathan Quezada, 2017

Alto= 10 Medio=5 Bajo=1

En gráfico 9-3 se visualiza los niveles de seguridad de las herramientas que se encuentran instalas en los equipos de la FIE.

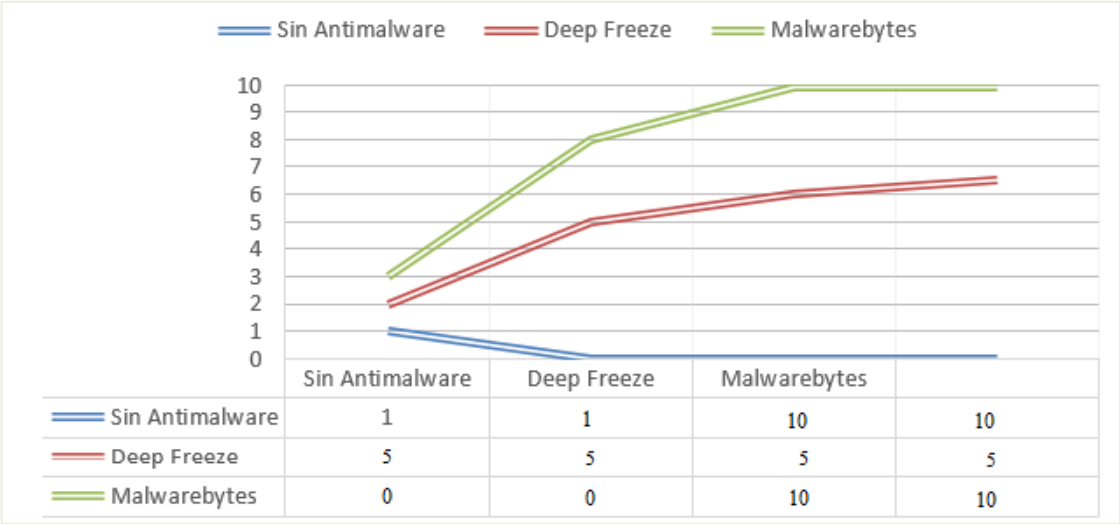


Gráfico 9-3: Niveles de seguridad

Realizado por: Jonathan Quezada, 2017

La confidencialidad y autenticidad de los datos tanto para los estudiantes como para los docentes son personales y para evitar posibles ataques en la red de la facultad se establece una autenticación MAC ADDRESS para cada punto de red de los laboratorios. Implementado esta política es más sencillo detectar desde que punto si realiza el ataque informático.

CONCLUSIONES

1 El estudio de las redes actuales ayudó a visualizar, detectar y realizar trazabilidad del malware en la red, ante nuevos vectores de ataque introducidos por los desarrolladores de malware que cada vez usan algoritmos más sofisticados para poder introducirse en las redes actuales evadiendo la seguridad.

2 Al implementar la red de prueba existe un nivel alto de vulnerabilidad en los equipos de los laboratorios de la FIE, ya que al momento que se realizó la infección de malware con Frutas RAT se obtuvo el 90% de efectividad en el ataque y el 10% fue la dificultad de usar técnicas para la evasión del firewall.

3 Se determinó el nivel de peligro es alto, mediante un ambiente dinámico de análisis de comportamiento de malware basado en Cuckoo Sandbox, al estar expuestos los usuarios cuando navegan por una computadora sin un nivel adecuado de seguridad para que sea cada vez más difícil de determinar su objetivo una vez infiltrado en la red.

4 El sistema funciona correctamente, con un nivel alto de seguridad, evitando el peligro que puede ser tener un equipo sin protección de un antimalware licenciado y no tener a las maquinas actualizadas con los nuevos parches lanzados por las empresas de seguridad informática.

RECOMENDACIONES

1 Debido a que las pruebas se realizan en un entorno virtualizado se recomienda que el disco duro del equipo en la que se implementó Cuckoo Sandbox posea una capacidad alta de procesamiento de 32 Gb en RAM y almacenamiento total del equipo mínimo 1Tera.

2 Se recomienda la instalación de un antimalware en los equipos de los laboratorios de la FIE para proteger la confidencialidad e integridad de los datos de los usuarios (docentes como de los estudiantes) que utilizan la red local del edificio de la FIE.

3 Se recomienda que las máquinas de los laboratorios de la FIE se apaguen diariamente al finalizar la jornada académica diaria, para evitar que los usuarios sean víctimas de malwares específicos encargados de recolectar información valiosa de la posible víctima.

4 Se recomienda la utilización del software Cuckoo Sandbox para un estudio avanzado del comportamiento del malware en la red.

5 Después de obtener los resultados de las vulnerabilidades de los equipos, se recomienda la ejecución de una política de seguridad que permita detectar, identificar y reparar las vulnerabilidades informáticas encontradas en los laboratorios del edifico de la Facultad de Informática y Electrónica mitigando de forma óptima las posibles vulnerabilidades que se pueden dar en la red.

BIBLIOGRAFÍA

- **Adastra**. *Cuckoo Sandbox y detección de malware.* [En linea] 2017. [Citado el: 20 de marzo 2017.] Disponible en: https://thehackerway.com/2014/11/18/cuckoo-sandbox-y-deteccion-de-maleware/.

- **Amaya, Camilo Gutiérrez.** Welivesecurity. *Analizando muestras con Cuckoo: Instalación y configuración (1/5).* [en línea] 24 de diciembre, 2013. [Citado el: 7 de Marzo 2017.] Disponible en: http://www.welivesecurity.com/la-es/2013/12/24/analizando-muestras-cuckoo-instalacion-configuracion-1/

- **Ávila, Ing. Mario.** *Detección de Malware Avanzado En Redes Organizacionales y Corporativas.* [En linea] 2012. pp. 289 - 295 [Citado el: 6 de Marzo 2017.] Disponible en: http://bibliotecadigital.econ.uba.ar/download/tpos/1502-0569_AvilaRodriguezMR.pdf.

- **Borghello, Cristian.** *Amenazas Lógicas - Tipos de Ataques.* [En linea] 2009. [Citado el: 6 de Marzo 2017.] Disponible en: http://www.segu-info.com.ar/ataques/ataques.htm.

- **Borghello, Lic. Cristian.** *Noticias sobre seguridad de la información.* [En linea] 12 de mayo 2014. [Citado el: 6 de Marzo 2017.] Disponible en: http://blog.segu-info.com.ar/2014/05/infeccion-de-frutas-rat-y-descarga.html.

- **Bustamante, Rubén.** "*Seguridad de Redes"*. Guatemala - Honduras, pp. 8-43.

- **Cardozo & García, Luis Fran Bladimiro.** "*Clasificación del Malware"*. Guadalajara-México, pp. 289-295.

- **Carlos, Ingeniero Juan.** *Seguridad Informatica - Informatica Forensec.* [En linea] 26 de febrero 2017. [Citado el: 6 de Marzo 2017.] Disponible en: http://computoforensec.blogspot.com/2017/02/cuckoo-es-un-sistema-automatizado-de.html.

- **Cisco.** *Catalyst 3750-X and 3560-X Switch Software Configuration Guide, Release 12.2(55)SE. Configuring SPAN and RSPAN.* [En linea] 10 de Noviembre 2014. [Citado el: 6 de Marzo 2017.] Disponible en: http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/12-2_55_se/configuration/guide/3750xscg/swspan.html.

- **Cisco.** *Catalyst 3750-X y 3560-X Guía de configuración del interruptor de software, versión 12.2 (55) SE.* [En linea] 10 de Noviembre 2014. [Citado el: 6 de Marzo 2017.] Disponible en: http://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3750x_3560x/software/release/12-2_55_se/configuration/guide/3750xscg/swspan.html#37143.

- **EtapaNet.** *Malware*. [En linea] 2010. [Citado el: 7 de Marzo 2017.] Disponible en: http://www.etapa.net.ec/Portals/0/Productos%20y%20Servicios/Malware.pdf.

- **López Ferreras, F. & Saturnino Maldonado, R. M. "***Análisis de Circuitos Lineales"*. 3raed. México, 2011, pp. 77-84.

- **Galas, Cleto.** *Qué son los virus informáticos.* [En linea] 2 de abril 2015. [Citado el: 6 de marzo 2017.] Disponible en: http://documentslide.com/documents/-que-son-los-virus-informaticos-los-virus-informaticos-son-sencillamente-programas-creados-para-infectar-sistemas-y-a-otros-programas-creandoles.html.

- **Gómez, Prof. Francisco Periañez.** *Características de VirtualBox.* [En linea] 8 de septiembre 2016. [Citado el: 6 de Marzo 2017.] Disponible en: http://fpg.x10host.com/VirtualBox/caractersticas_de_virtualbox.html.

- **Informática, Profesionales de la seguridad.** *Cuckoo Sandbox y detección de malware*. [En linea] 18 de Noviembre 18 2014. [Citado el: 6 de Marzo 2017.] Disponible en: https://thehackerway.com/tag/cuckoo-sandbox/.

- **Ing. Felipe Pérez Roque, Dr. Enrique Valdés Zaldívar, Dra. Olimpia Arias de Fuentes.** Sistema de Adquisición de Datos con comunicación inalámbrica. *SciELO*. [En linea] Septiembre 2013. [Citado el: 6 de Marzo 2017.] Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1815-59282013000300007.

- **Juan, C. V.** *Servicios de seguridad de la informacion.* [En linea] [Citado el : 6 de Marzo 2017.] Disponible en: http://contact.orben.com/seguridad2/?gclid=CjwKEAiArvTFBRCLq5-7-MSJ0jMSJABHBvp0YrfoIZ3JL1ta-pZ0-XEjA9ZrJliy5_1gqFVsgnlUhoCXQLw_wcB.

- **Kali Linux.** *Política de Código Abierto en Kali Linux.* [En linea] 2017. [Citado el: 10 de Marzo 2017.] Disponible en: http://es.docs.kali.org/kali-policy-es/politica-de-codigo-abierto-en-kali-linux.

- **Ledesma, Rodolfo.** *All Networking.* [En linea] 19 de marzo 2008. [Citado el: 10 de marzo 2017.] Disponible en: http://allnetworking.blogspot.com/2008/03/8021q.html.

- **Ortega, C. V. "***Introducción a la Seguridad Informática".* Madrid - España, pp. 12 – 45.

- **Oscar & Reyes, Ramírez Omar.** "*Implementación de un laboratorio de Análisis de malware"* México, pp. 12-34.

- **Triviño, Roberto,** *Implementación de un ambiente de malware CITIC.* (Tesis Maestría). Escuela Politécnica del Ejército, Departamento de Eléctrica y Electrónica, Carrera de Ingeniería en Electrónica y Telecomunicaciones. Ecuador, Sangolquí. 2006, pp. 45-70. [Citado el: 8 de Octubre de 2016.]. Disponible en: http://repositorio.espe.edu.ec

ANEXOS

ANEXO A

Equipamiento por sedes o extensiones donde se impartirá la carrera

INVENTARIO DE LOS EQUIPOS EXISTENTES EN LA FIE.

	Laboratorios y/o talleres			
Nombre	**Equipamiento**		**Metros cuadrados**	**Puestos de trabajos**
Laboratorio de Automatización Industrial	1 1 1 1 1 1 1 1	Estación de Evaluación Estación de Musculo Neumático Estación de Separación Estación de Giro Estación de Clasificación Estación de Robot Estación de Pick and Place Sopladora de Plástico	66,86 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
Laboratorio de Redes Industriales	1	Módulo de Ensamblaje Base y Tapa	69,12 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
	1	Módulo de Evaluación		
	1	Estación de Clasificación		
	1	Field Point		
	1	PLC WAGO		
Laboratorio de Hidráulica y Neumática	1	Equipo de hidráulica TP601	67,81 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
	2	Válvula limitadora de presión		
	2	Regulador de caudal de 2 vías		
	2	Regulador de flujo unidireccional		
	2	Válvula de antirretorno, desbloqueable		
	2	Válvula de antirretorno, 0,6 MPA de presión de apertura		
	2	Válvula de 4/2 vías, accionada manualmente		
	2	Válvula de 4/3 vías, manual, posición central a descarga (ab –> t)		
	2	Válvula de 4/3 vías, manual, con centro cerrado y enclavamiento		
	2	Válvula de cierre		
	2	Cilindro diferencial 16/10/200 con cubierta		
	2	Peso de 9 kg para cilindro		
	2	Motor hidráulico		
	4	Conector en t		
	6	Distribuidor de 4 vías con manómetro		
	2	Manómetro		
	2	Sensor de caudal		

	6	Mangueras con acoplamientos rápidos 1000 mm		
	14	Mangueras con acoplamientos rápidos 600 mm		
	14	Mangueras con acoplamientos rápidos 1500 mm		
Laboratorio de Control y Máquinas	5	Tableros Didácticos para Laboratorio de Maquinas	67,81 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
	3	Motores Trifásico 3HP		
	3	Motores 3 ~ ABB 1.5HP		
	3	Motores de Inducción Monofásica 1/2HP		
	2	Motores BALDOR Current Direct		
	2	Motores de Inducción Monofasica VOGES 1HP		
	1	Motor Monofasico CARPANELLI 1HP		
	10	Intel GALILEO		
	22	Temporizadores		
	4	Guardamotores		
	25	Arrancadores		
	27	Relé		
	22	Relés Térmicos		
	6	Electronic Timer		
	25	Contactores ABB		
	4	Contactores LS Industrial		
Laboratorio de Electrónica	9	Osciloscopios Digitales	68,44 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
	9	Osciloscopios Analógicos		
	10	Generadores de Señal		
	4	Laboratorios Digitales		
	10	Laboratorios Experimentales		
	4	Fuentes Regulables		
	10	Multímetros		
	3	Tarjetas FPGA		
	10	NI ELVIS		
	8	Legos Mindstorms		
	2	Brazos Robóticos		
	15	Project Board		
	1	Fluke 435 Power Quality Analizer		
	8	Entrenadores PIC		
	8	Cautines		
	2	ACS350 Convertidores de Frecuencia		
	9	Logos 23ORCE		
	1	DAQ		
	2	My DAQ		
		CIRCUITOS ELÉCTRICOS		
	3	Fuente de poder dc		
	3	Fuente de poder ac		

	3	Generador de señales		
	3	Generador de funciones		
	3	Testing and display		
	3	Basic device module		
	3	Basic electricity experiment module		
	3	Two sensor module		
	3	Diode, clipper and clamper module		
	3	Rectifier, differentiator integrator circuit module		
	3	Transistor amplifier circuit module		
	3	Multi-stage amplifier circuit module		
	3	Fet circuit experiment module		
	3	Five op amplifier circuit module		
	3	Four combination logic circuit experiment module		
	3	Two sequential logic circuit experiment Module		
		EQUIPOS DE ELECTRÓNICA		
	4	Fuente de poder DC		
	4	Fuente de poder AC		
	4	Generador de funciones		
	4	Testing and display		
	4	Variable resistors		
	4	Clipping & clamping circuits		
	4	Rectifier, differential & integrator circuits		
	4	Transistor amplification circuits		
	4	Multistage amplification circuits		
	4	Otl amplifier circuit		
	4	Ocl amplifier & feedback circuit		
	4	Two oscillator circuits		
	4	Voltage regulator circuits		
	4	Voltage regulator & amplitude modulation (am) circuits		
	4	Frequency modulation (fm) & op amplifier circuits		
	4	Five op amplifier circuits		
		EQUIPOS DE ELECTROMAGNETÍSMO		
	1	Fuente de poder DC		
	1	Fuente de poder AC		
	1	Generador de funciones		
	1	Testing and display		
	1	Variable resistors		
	1	Basic electricity experiments module		
	1	Magnetism element introduction module		

	1	Magnetic field module		
	1	Ampere's rule module		
	1	Fleming's rule module		
	1	Electromagnetic induction		
	1	Electronic circuit fundamental experiments module		
		Two basic electronic circuit experiments		
	1	Special electronic components experiments module		
	1	Oscillator experiments and applications module		
Laboratorio de Robótica	1	IMPRESORA 3D		31 puestos de trabajo(30 estudiantes y 1 docente)
	5	MakerBot Dark Sanguine Red ABS Filament (1kg Spool)		
	5	**MakerBot True White ABS Filament (1kg Spool)**		
	5	**MakerBot True Yellow ABS Filament (1kg Spool)**		
	5	**MakerBot True Blue ABS Filament (1kg Spool)**		
	5	**MakerBot True Orange ABS Filament (1kg Spool)**		
	5	**SINGLE BOARD ROBOT COMPUTERN STARTER KIT**		
	5	**ADVANCED FPGA DEVELOPMENT SYSTEM**		
	4	**ROBOT HUMANOIDE**		
Laboratorio de Sistemas de Control Automático	1	Fuente de poder dc		31 puestos de trabajo(30 estudiantes y 1 docente)
	1	Summing junction		
	1	P-controller		
	1	I-controller		
	1	D-controller		
	1	Sum/dif amplifier		
	1	Integrator		
	1	Inverting amplifier- push-button r-cal.4		
	1	Inverting amplifier - push-button r-cal.5		
	1	Second order plant		
	1	Lead/lag compensator		
	1	Test signal generator		
	1	Function generator		
	1	Over range check		
	1	Analog power driver		
	1	Dc servo PWM driver		
	1	Linear VR angle/position sensor & buffer		
	1	Calibration & testing module		
	1	Data acquisition device		
	1	Dc servo motor & control unit		
Laboratorio de Electrónica de Potencia	1	DC Power Supply (15V/2A)		31 puestos de trabajo(30 estudiantes y 1 docente)
	1	DC Power Supply (0-40V/6A)		
	1	Reference Variable Generator		
	1	Differential Amplifier		

	1	Current Transducer		
	1	Three Phase Angle Controller		
	1	R.M.S. Meter		
	1	Power Meter (0.3W-30KW)		
	1	Resistor Load Unit		
	1	Resistor Load		
	1	Inductive Load Unit		
	1	Flyback Switching Power Supply		
	1	Boost Switching Power Supply		
	1	Buck Switching Power Supply		
	1	Buck-Boost Switching Power Supply Unit		
	1	Electronic Ballast Fluorescent Lamp		
	1	IGBT Drive Set		
	1	DC PWM Generator		
	1	Single Phase PWM Controller		
	1	Three Phase PWM Controller		
	1	Three Phase Rectifier & Filter		
	1	Three Phase Rectifier & Filter		
	1	Power Diode Set		
	1	Fuse Set		
	1	Thyristor (800V/10A)		
	1	SCR/TRIAC Set		
	1	MOSFET/ IGBT Set		
	1	SCR DC Chopper Set		
	1	Isolating Transformer		
	1	Magnetic Powder Brake Unit		
	1	Brake Controller		
	1	DC Permanent-Magnet Machine		
	1	Three-Phase Squirrel Cage Motor		
	1	Experimental Frame		
	1	Coupling		
	1	Connecting Leads Set		
Laboratorio de Instrumentación Industrial	1	DSP interface control board		31 puestos de trabajo(30 estudiantes y 1 docente)
	1	Control unit		
	1	Interface unit		
	1	**Function generator**		
	1	**OCL amplifier**		
	1	Audio selector		
	1	Speaker		
	1	Fixed dc power		
	1	Module fixed slide rail		
	1	Dc servo PWM control		
	1	Step motor control		
	1	Temperature control		
	1	PLC I/O interface		
	1	OCL amplifier		
	3	PSOC		
	3	I/O peripheral circuits		
Laboratorio de Sistemas Embebidos	10	FPGA		31 puestos de trabajo(30
	5	COLOR LCD TOUCHSCREEN		

y Microcontroladores	5 5 5	STEREO CAMERA MODULE VHDC BREADBOARD VHDCI MALE-TO-MALE CABLE		estudiantes y 1 docente)
Laboratorio de Informática 1	33	Marca CPU hp Monitor hp lv1911 18.5” Teclado hp ps2 Mouse óptico ps2 Procesador Intel core i7. Velocidad procesador 3.4 hz Memoria RAM 4gb Disco duro 500gb 7200 rpm Smart sata Modelo 6200 pro mt color negro	72,52 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
Laboratorio de Informática 2	33	Marca CPU hp Monitor hp lv1911 18.5” Teclado hp ps2 Mouse óptico ps2 Procesador Intel core i7. Velocidad procesador 3.4 hz Memoria RAM 4gb Disco duro 500gb 7200 rpm Smart sata Modelo 6200 pro mt color negro	72,52 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
Laboratorio de Informática 3	33	Marca CPU hp Monitor hp lv1911 18.5” Teclado hp ps2 Mouse óptico ps2 Procesador Intel core i7. Velocidad procesador 3.4 hz Memoria RAM 4gb Disco duro 500gb 7200 rpm Smart sata Modelo 6200 pro mt color negro	72,52 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
Laboratorio de Informática 4	32	Marca CPU hp Monitor hp lv1911 18.5” Teclado hp ps2 Mouse óptico ps2 Procesador Intel core i7. Velocidad procesador 3.4 hz Memoria RAM 4gb Disco duro 500gb 7200 rpm Smart sata Modelo 6200 pro mt color negro	72,52 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)
Laboratorio de Realidad Virtual	33	Marca CPU hp Monitor hp lv1911 18.5” Teclado hp ps2 Mouse óptico ps2 Procesador Intel core i7. Velocidad procesadora 3.4 hz	66,88 m^2	31 puestos de trabajo(30 estudiantes y 1 docente)

		Memoria RAM 4gb Disco duro 500gb 7200 rpm Smart sata Modelo 6200 pro mt color negro		

Bibliotecas específicas por sedes o extensiones existentes en la Biblioteca Central.

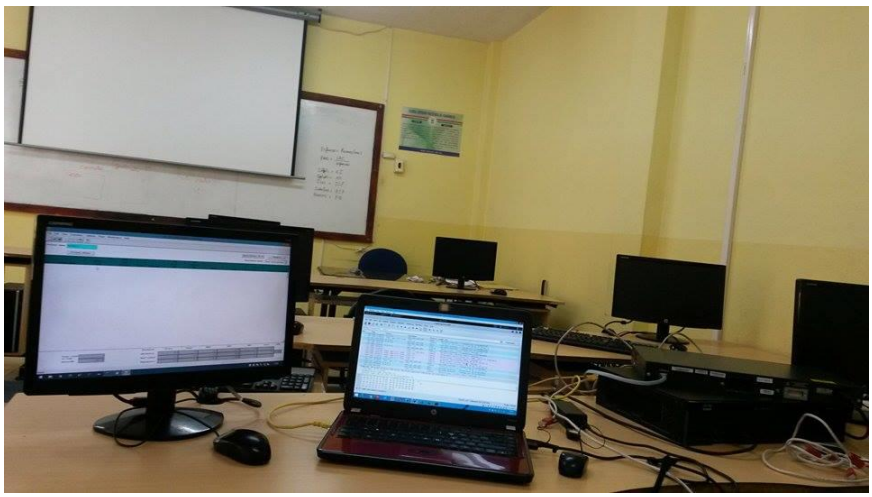
Tabla#2 Información de los encargados de las bibliotecas.

Desglose por cantidad	Número	Descripción general
Títulos	12548	Biblioteca General
Volúmenes	17660	Ejemplares impresos – Biblioteca General
Base de datos en línea	6	Biblioteca Virtual(Base de datos)
Suscripciones a revistas especializadas	3	IEEEXplore, EBSCOhost, Springer

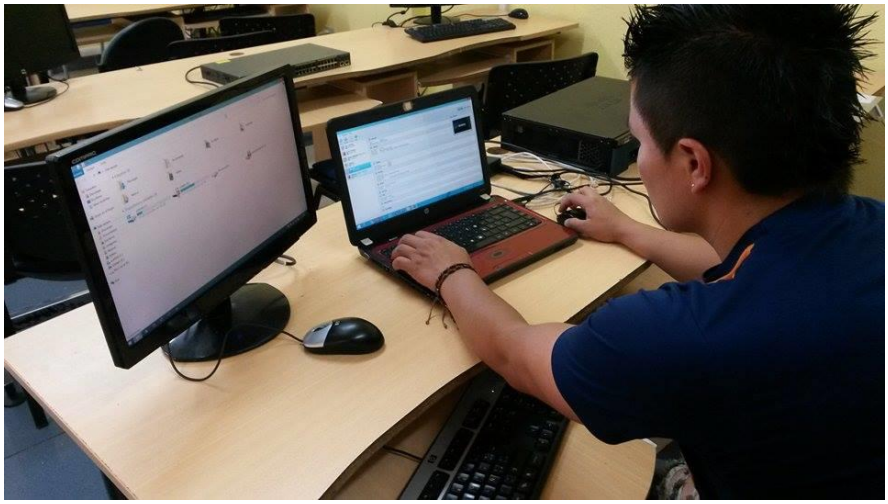
ANEXO B

- **IMPLEMENTACIÓN DE LA RED DE PRUEBA.**

1. En la red de prueba que se implementa se analizó el tráfico para observa comportamiento malicioso para identificarlo y analizarlo.



2. Utilizando la herramienta Kali Linux en su versión 2016.2, se realizó la creación y configuración de los parámetros del malware.



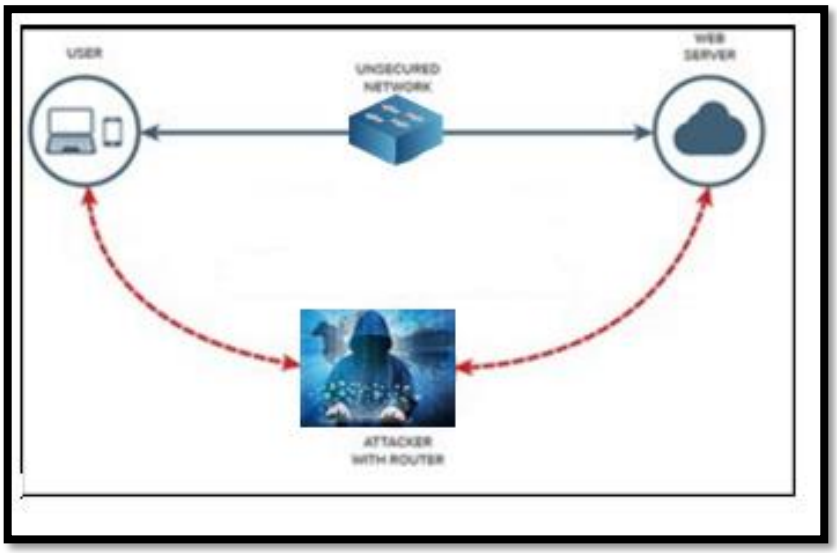
3. Se escaneo los puertos abiertos de los equipos para direccionar los ataques para vulnerar el equipo.

4. Una vez determinada la vulnerabilidad de la víctima se ejecutó la mejor opción de ataque informático aprovechando la vulnerabilidad de la víctima.

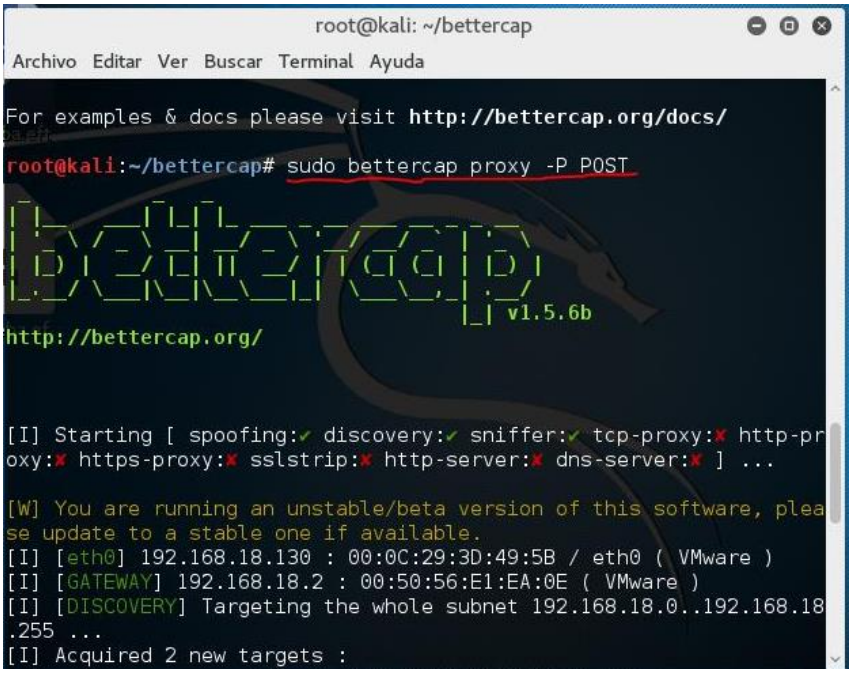
ANEXO C

- **ATAQUE MAN IN THE MIDDLE CON BETTERCAP.**

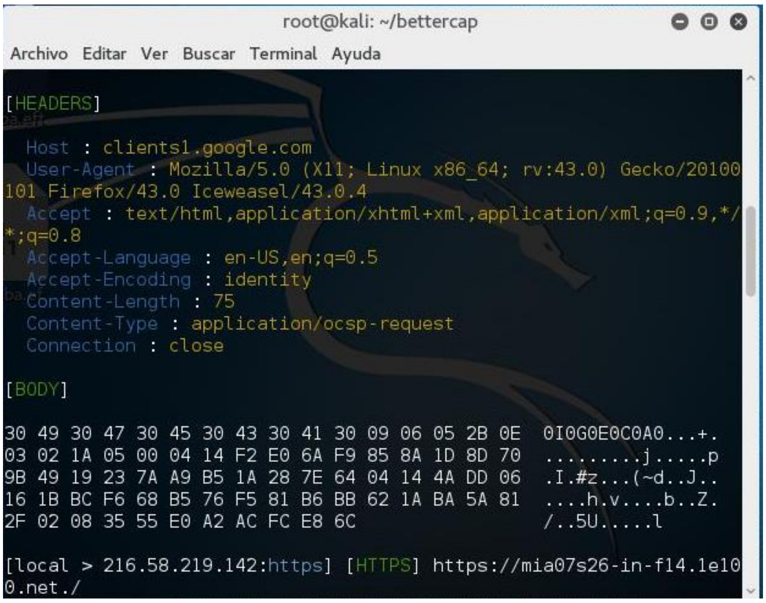
Diagrama de Bettercap para realizar ataques MITM (Man-in-the middle)



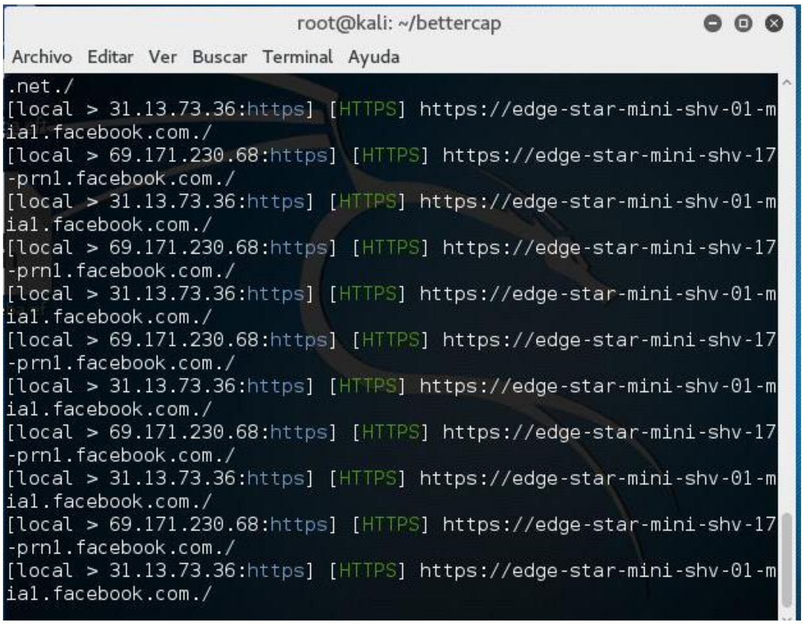
a) Se indicia en la figura el arranque del programa bettercap para realizar el ataque.



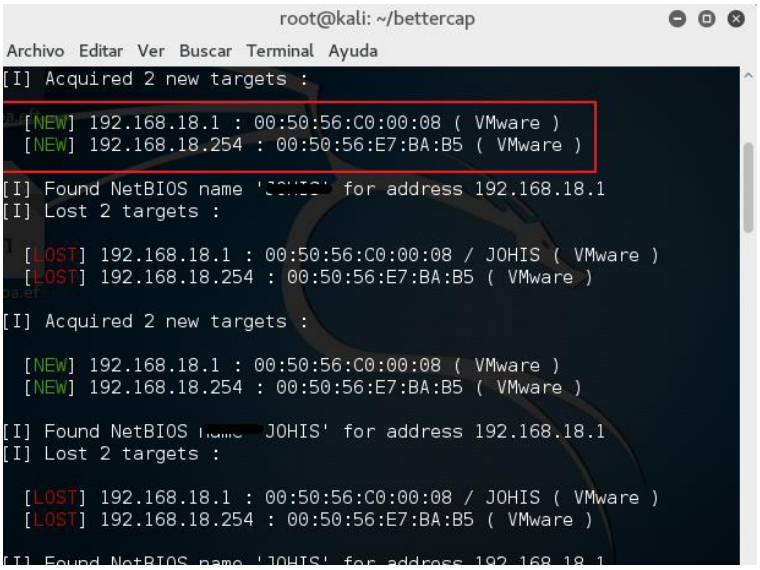
b) Empezó a escanear el comportamiento de la red.



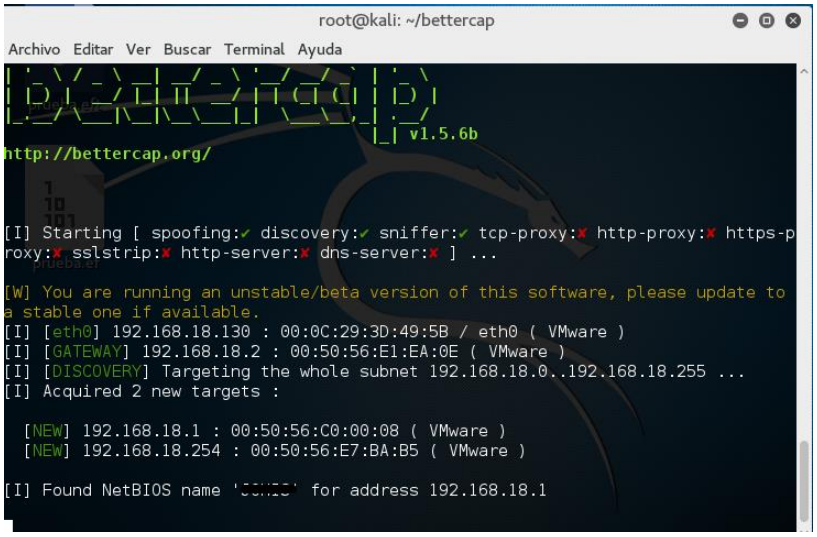
4) Se observó las ip de las víctimas y a los sitios que ingresan



5) Se analizó el tráfico para visualizar la ip de la víctima.



6) Se identificó la ip de la victima

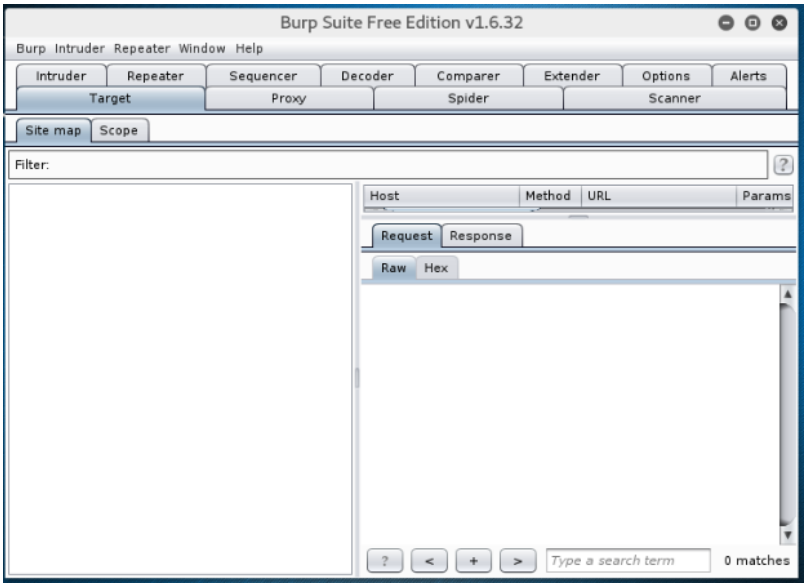


7) Una vez identificada la ip de la víctima se realiza el ataque mencionado una vez que el usuario visite un sitio web.

ANEXO D

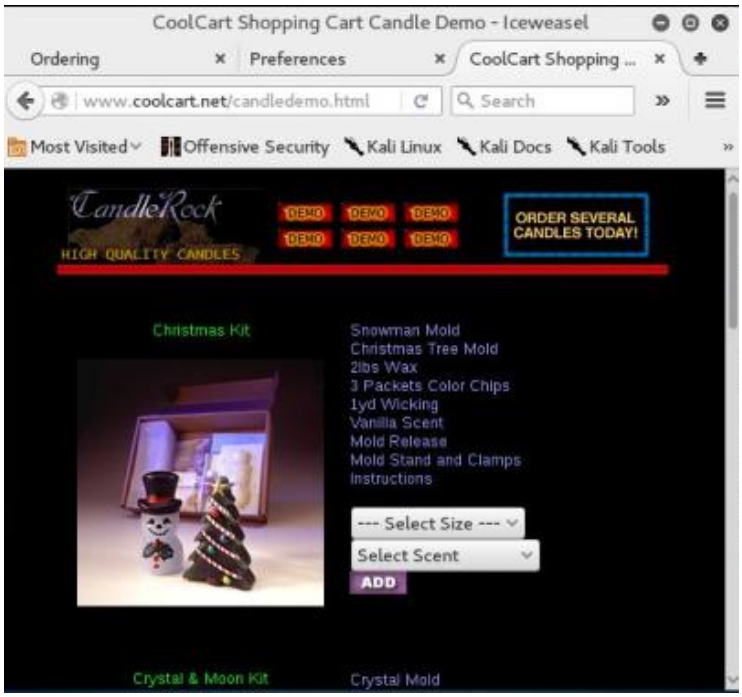
- **ATAQUE SQL INJECTION.**

1) Se utilizó la herramienta Burp Suit que permite combinar técnicas manuales para enumerar, analizar explotar y atacar aplicaciones web.

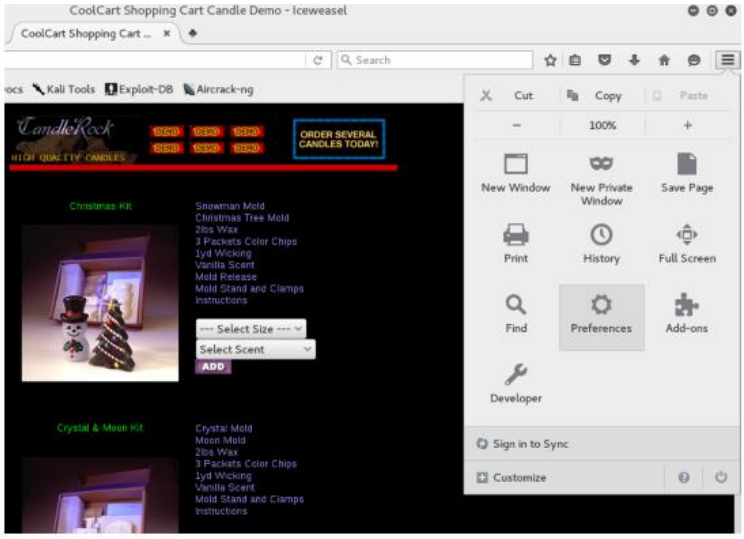


2) En el sistema Kali Linux se ejecutó la herramienta Burp Suite.

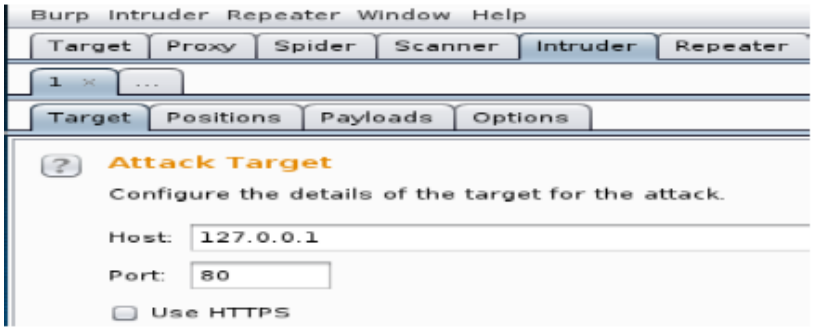
3) Seleccionar la página web a la que se desea atacar, en este caso se realizara la intrusión a una tienda online.

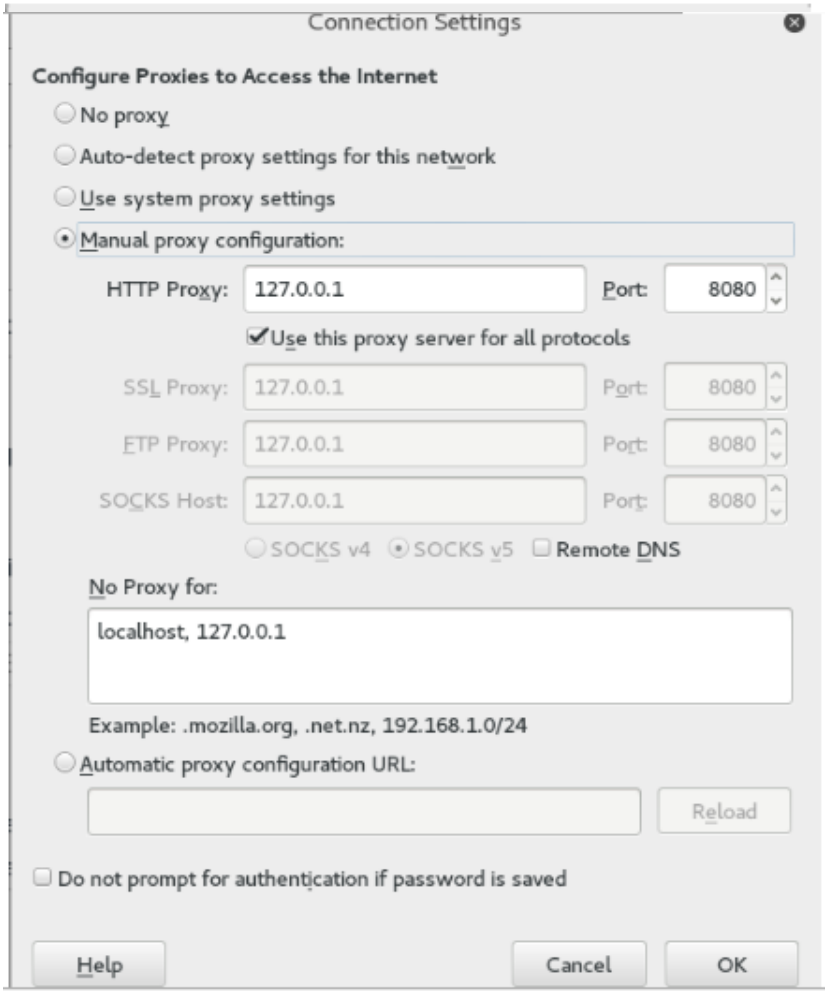


4) Se cambió la configuración del proxy en el navegador siguiendo los pasos determinados a continuación:
   a) Menú
   b) Preferencias
   c) Avanzado – Red
   d) Conexión
   e) Configuración
   f) Configuración manual proxy



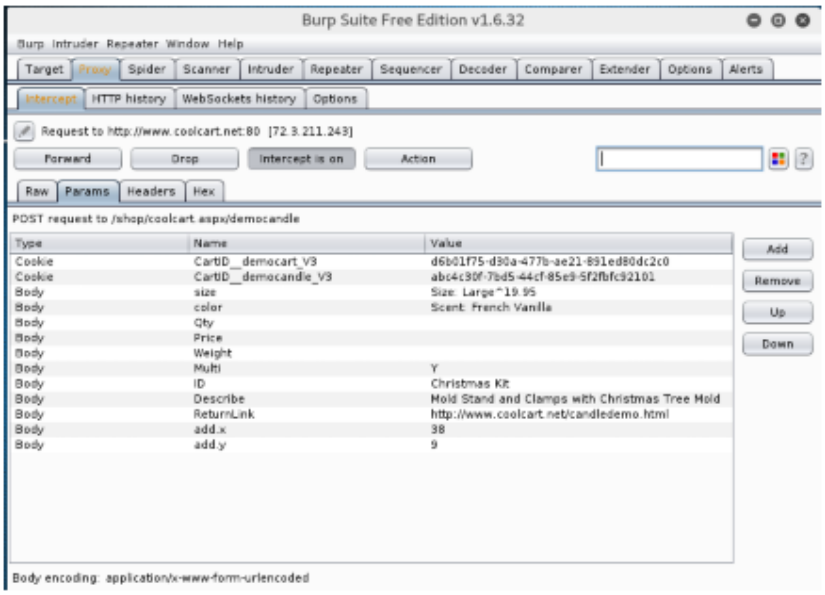
5) Se cambió la configuración del proxy manual se necesita cambiar la dirección del host y puerto según la configuración que tiene Burp suite.





6) Seleccionar los parámetros del producto que se desea comprar en la tienda online y añadir la compra.

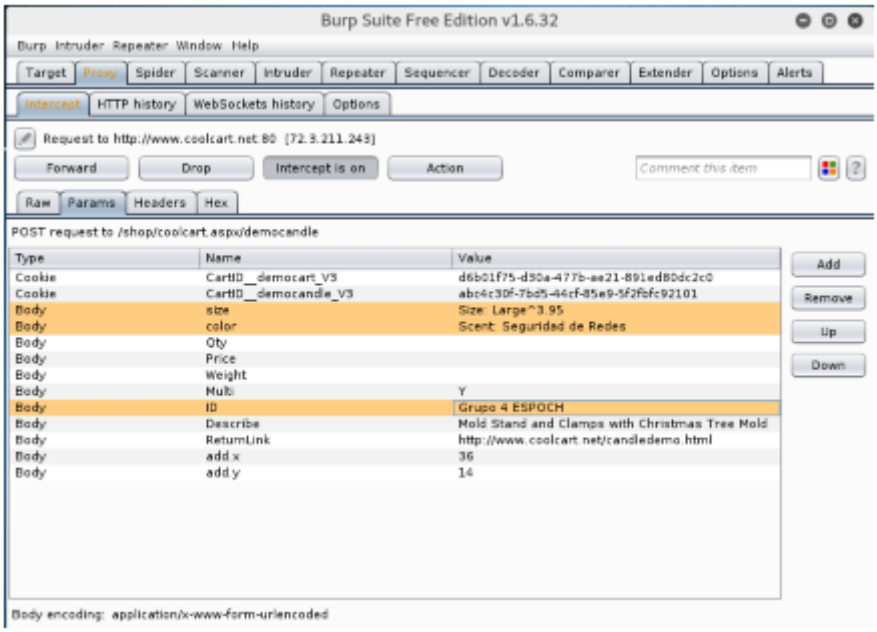
7) Una vez que se acepta la compra, Burp suite tiene acceso a la información, debido a la configuración del proxy que se realizó.



8) Modificar los parámetros de la compra, en este caso se cambia los siguientes datos:

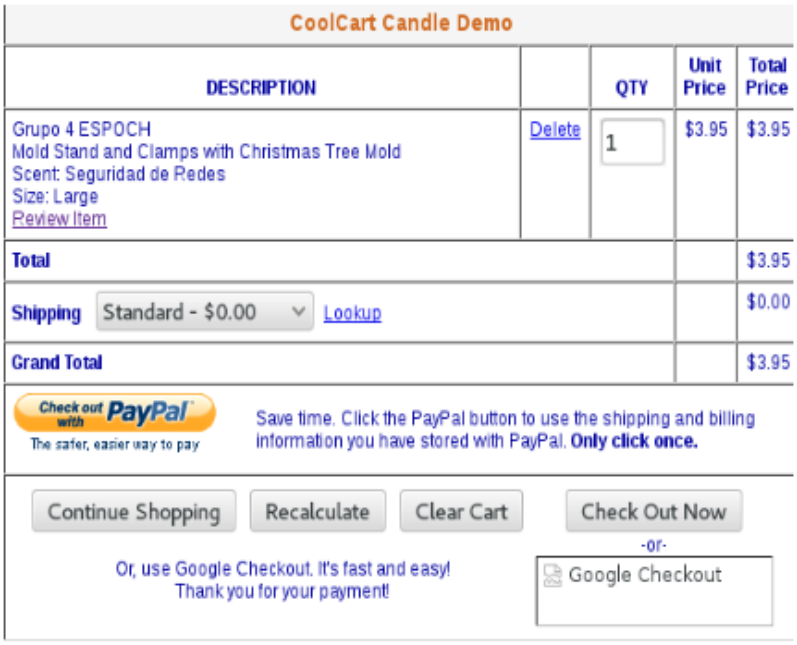
- Suize
- Color e ID

Para realizar esto se da doble click para alterar estos datos.



9) Finalizada la alteración de los datos, es necesario dar click en Forward e inmediatamente se modificarán los datos de la compra.

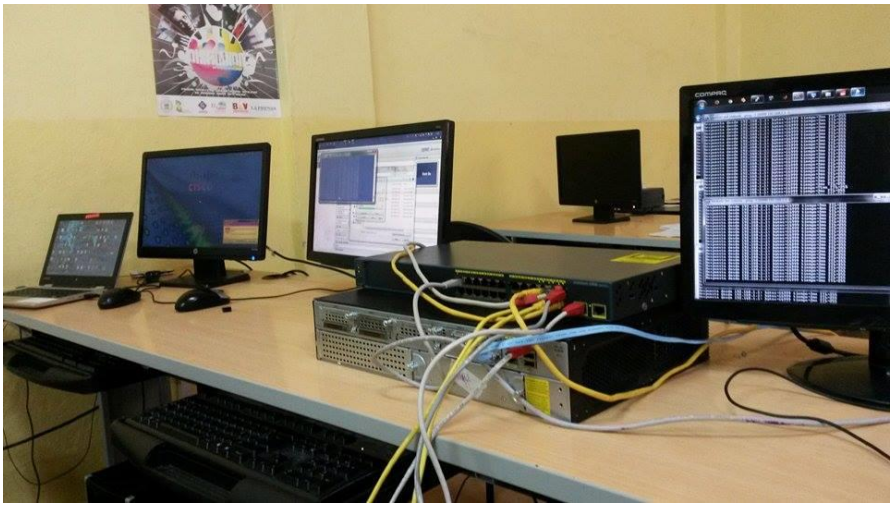
10) Se observa que los datos han sido modificados.



ANEXO E

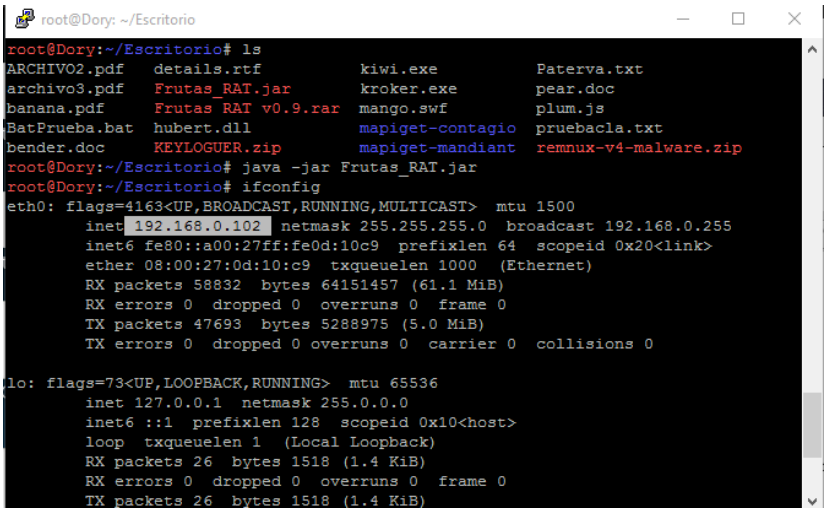
- **INFECCIÓN DE MALWARE CON FRUTAS RAT.**

1) Se implementó la red para realizar la práctica.

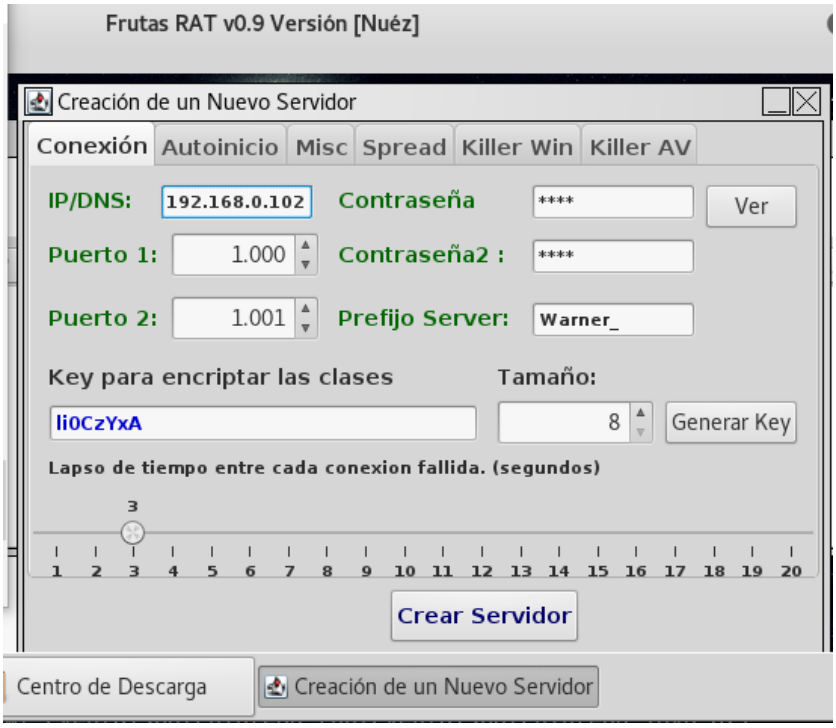


2) En el terminal de Kali se escribe: java –jar Frutas_RAT.jar para ejecutar el archivo descargado anteriormente.

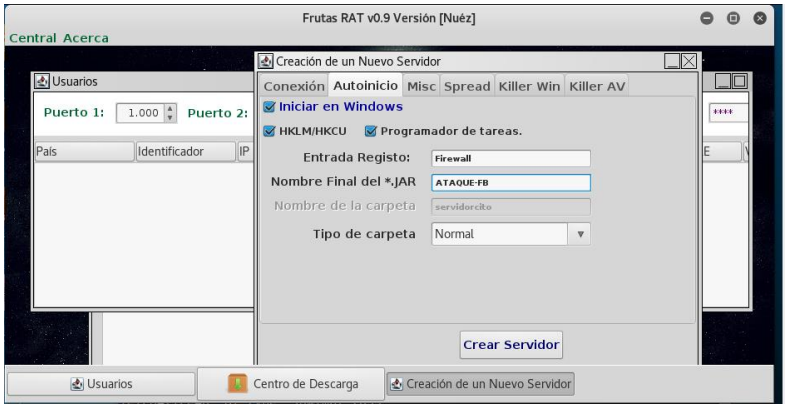
3) En un nuevo terminal con el comando ifconfig se averigua la dirección ip del atacante y



4) Una vez en Frutas_RAT se ingresa a Creación de un nuevo servidor.



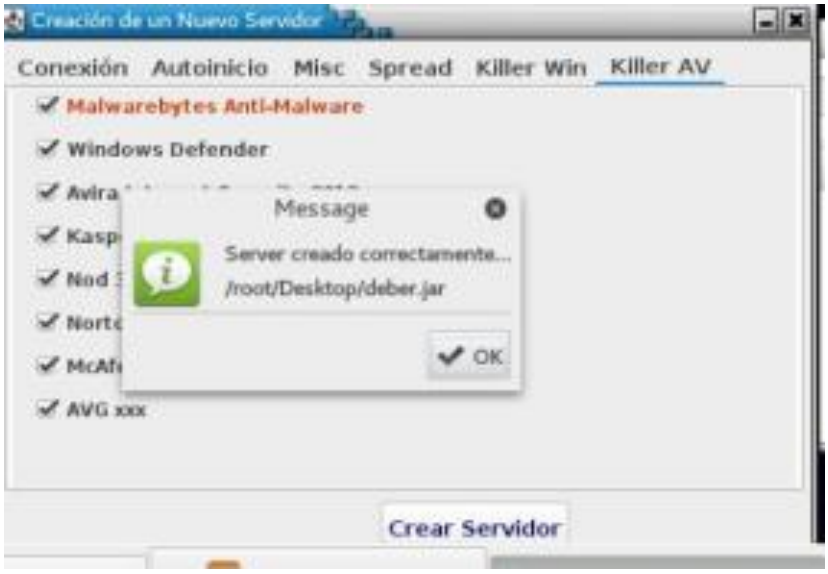
5) En esta ventana activar todas las casillas y asignar un nombre, se lo denomino ATAQUE-FB, para posterior enviar a la víctima la extensión .jar en este caso fue ATAQUE-FB.jar



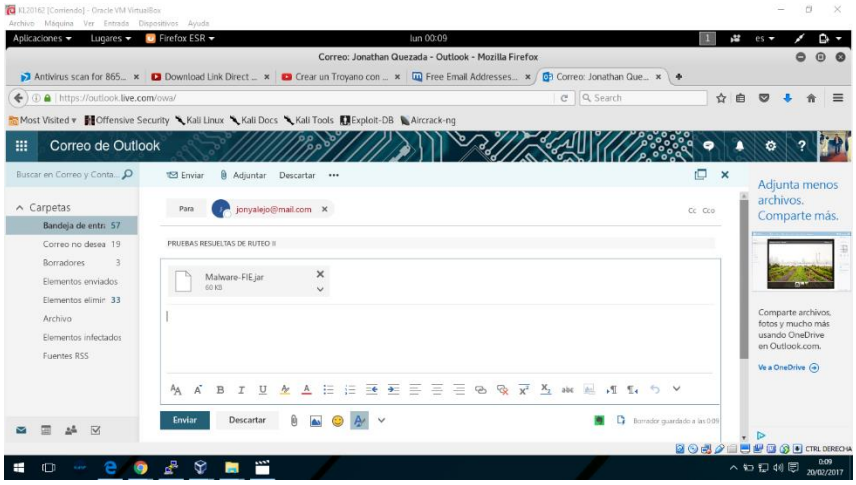
6) En la siguiente opción se marcó las tareas que van a ser bloqueadas.

7) Es necesario desactivar los antivirus para que no nos detecte como troyano.

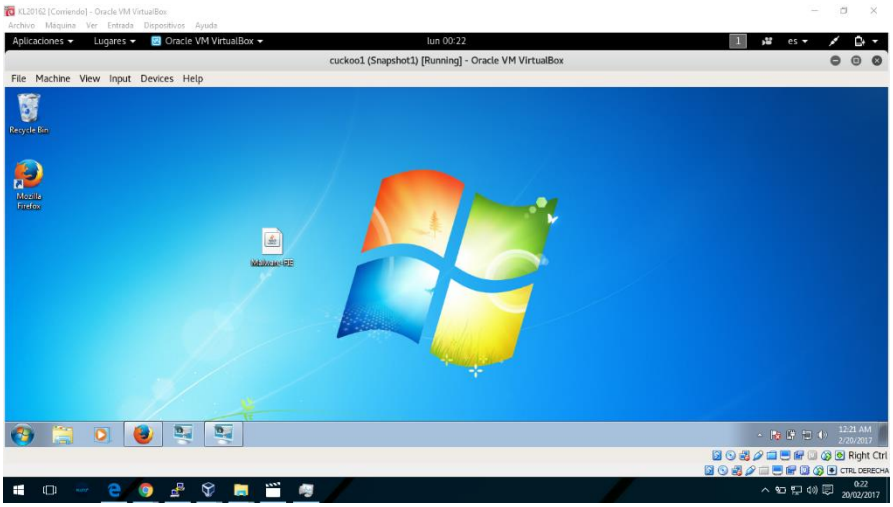
8) Una vez finalizado las configuraciones del troyano dar click en crear servidor y escoger la ruta que tendrá, en este caso, en el escritorio, posteriormente mediante un mensaje indicara que el servidor fue creado correctamente y como observar el archivo está listo para ser enviado.



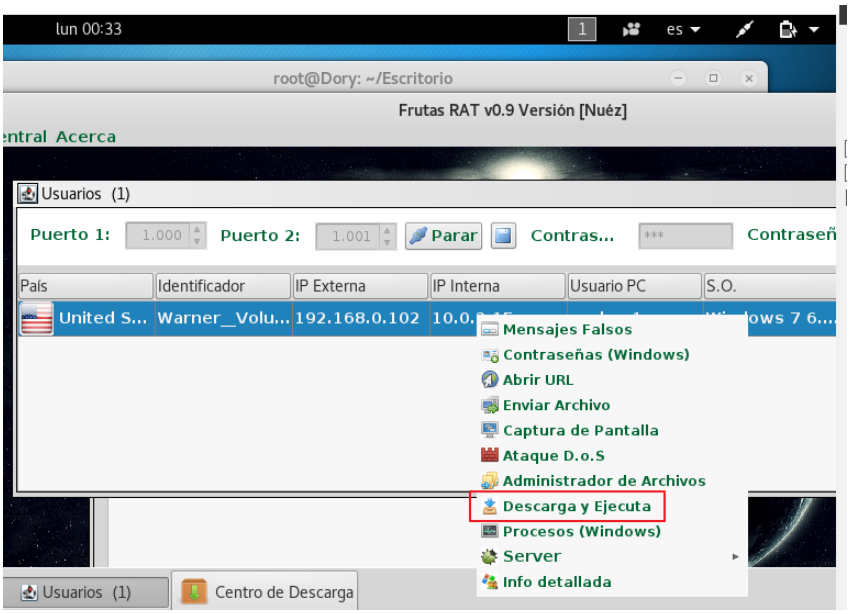
9) Se utilizó ingeniería social para enviar el virus a nuestros compañeros en la facultad y conseguir infectar la mayoría de equipos posibles.



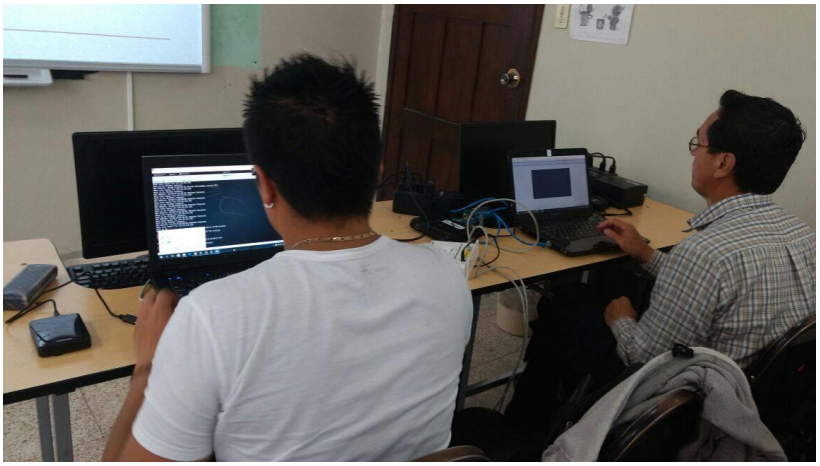
10) El malware ya se encuentra en la máquina del usuario víctima.



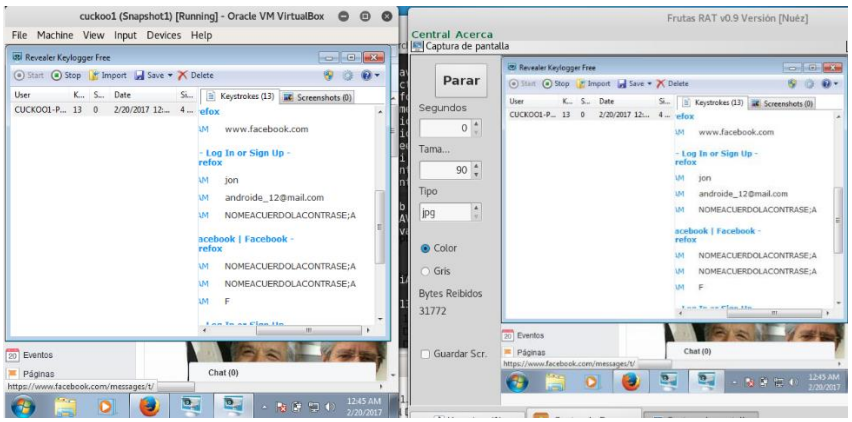
11) Cuando la víctima ejecuta el malware tomar el control total de la computadora que está utilizando.



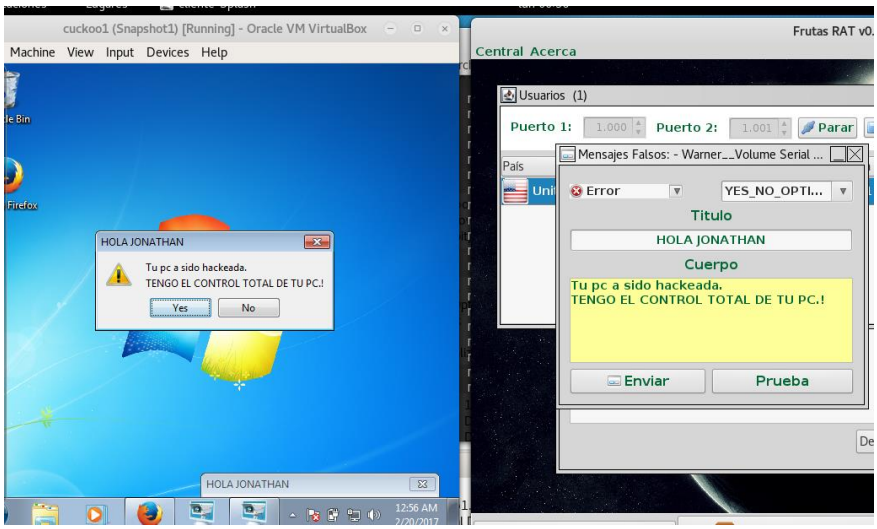
12) Se observó que la víctima a descargado el archivo que se ha propagado por medio de www.live.com.



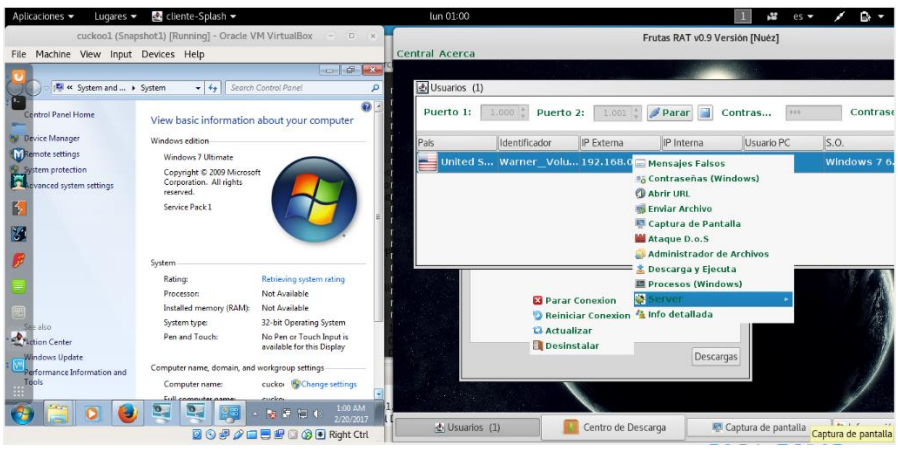
13) Se descargó e instalo un keylogger en la victima y se observa todo lo que hace el su ordenador.



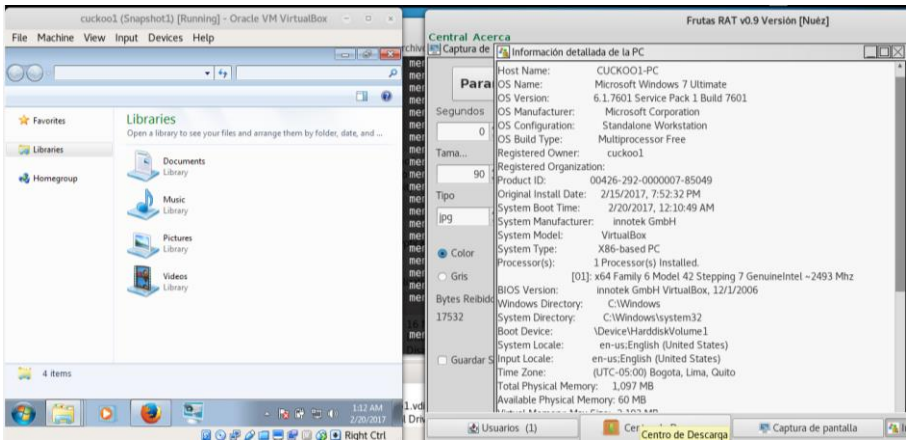
14) Se envió un mensaje a la víctima advirtiéndole que su ordenador ha sido vulnerado.



15) Se observó las características de la computadora vulnerada.



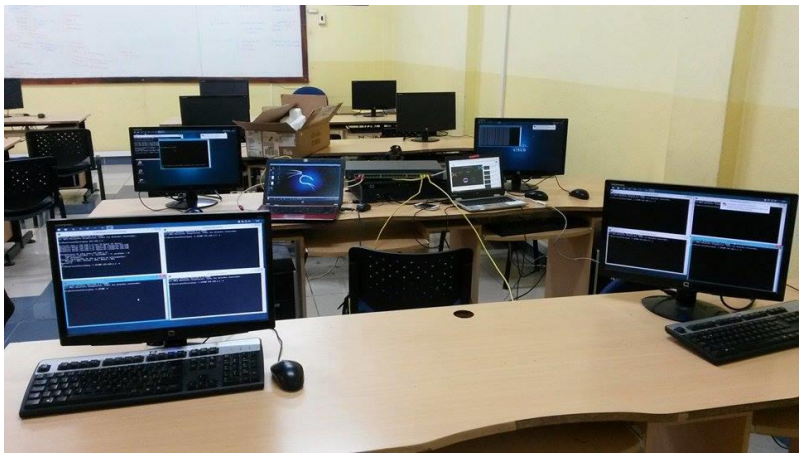
16) Finalmente se observa información específica y muy confidencial de la víctima.



ANEXO F

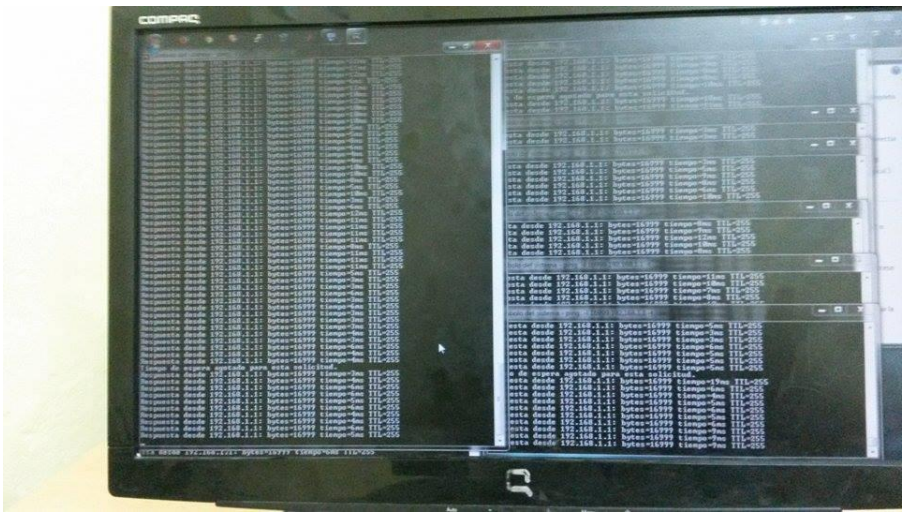
- **ATAQUE DDOS CON MÁQUINAS ZOMBIS.**

1) Ataque sincronizado a la red de prueba implementada en los laboratorios.



2) Configuración de los parámetros para iniciar el ataque DDoS.

3) Resultado del ataque DDoS.



ANEXO G

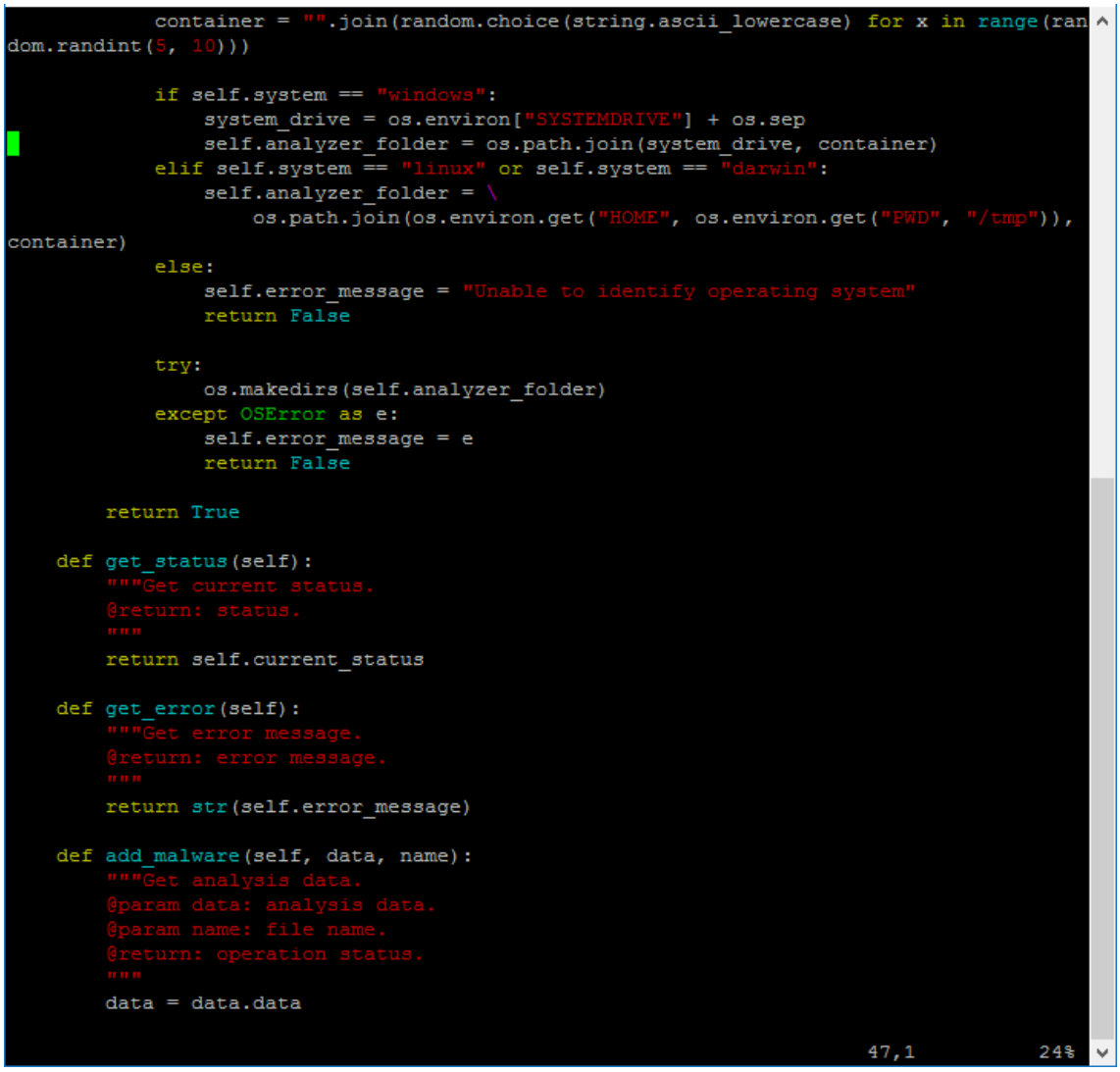
LÍNEAS DE PROGRAMACIÓN PARA CONFIGURACIÓN.

- agent.py
- cuckoo.conf
- reporting.conf
- virtualbox.conf.

Agente.py

root@CUCKOO_SANDBOX: ~/cuckoo/agent

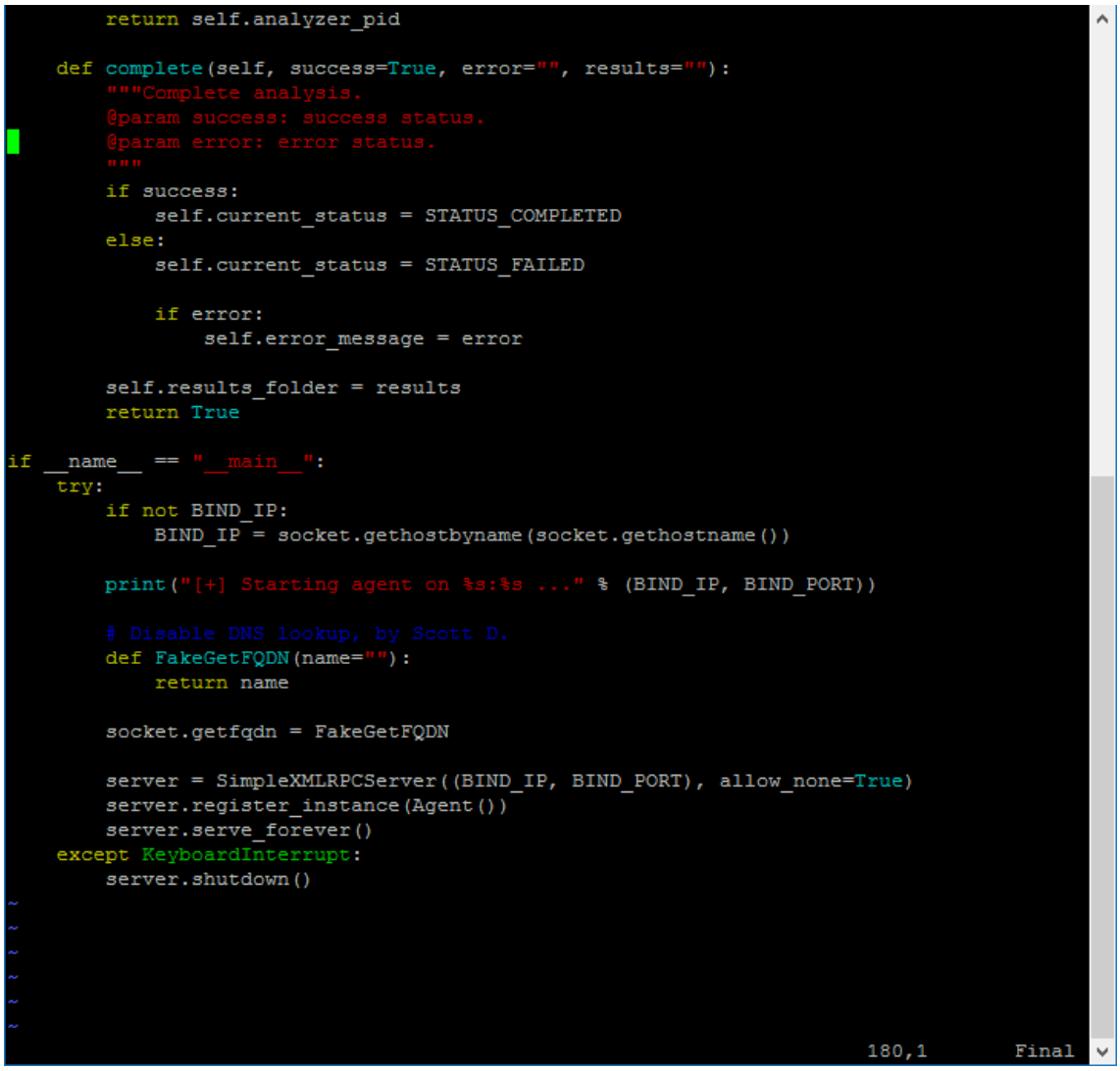
```
# Copyright (C) 2010-2013 Claudio Guarnieri.
# Copyright (C) 2014-2016 Cuckoo Foundation.
# This file is part of Cuckoo Sandbox - http://www.cuckoosandbox.org
# See the file 'docs/LICENSE' for copying permission.

import os
import sys
import time
import socket
import string
import random
import platform
import subprocess
import ConfigParser
from StringIO import StringIO
from zipfile import ZipFile
from SimpleXMLRPCServer import SimpleXMLRPCServer

BIND_IP = "0.0.0.0"
BIND_PORT = 8000

STATUS_INIT = 0x0001
STATUS_RUNNING = 0x0002
STATUS_COMPLETED = 0x0003
STATUS_FAILED = 0x0004

class Agent(object):
    """Cuckoo agent, it runs inside guest."""

    def __init__(self):
        self.system = platform.system().lower()
        self.analyzer_path = ""
        self.analyzer_pid = 0

        self.error_message = None
        self.current_status = STATUS_INIT
        self.analyzer_folder = ""
        self.results_folder = ""

    def _initialize(self):
        if not self.analyzer_folder:
            random.seed(time.time())
                                                    33,1      Comienzo
```



```
            container = "".join(random.choice(string.ascii_lowercase) for x in range(ran
dom.randint(5, 10)))

            if self.system == "windows":
                system_drive = os.environ["SYSTEMDRIVE"] + os.sep
                self.analyzer_folder = os.path.join(system_drive, container)
            elif self.system == "linux" or self.system == "darwin":
                self.analyzer_folder = \
                    os.path.join(os.environ.get("HOME", os.environ.get("PWD", "/tmp")),
container)
            else:
                self.error_message = "Unable to identify operating system"
                return False

            try:
                os.makedirs(self.analyzer_folder)
            except OSError as e:
                self.error_message = e
                return False

        return True

    def get_status(self):
        """Get current status.
        @return: status.
        """
        return self.current_status

    def get_error(self):
        """Get error message.
        @return: error message.
        """
        return str(self.error_message)

    def add_malware(self, data, name):
        """Get analysis data.
        @param data: analysis data.
        @param name: file name.
        @return: operation status.
        """
        data = data.data
                                                                47,1          24%
```

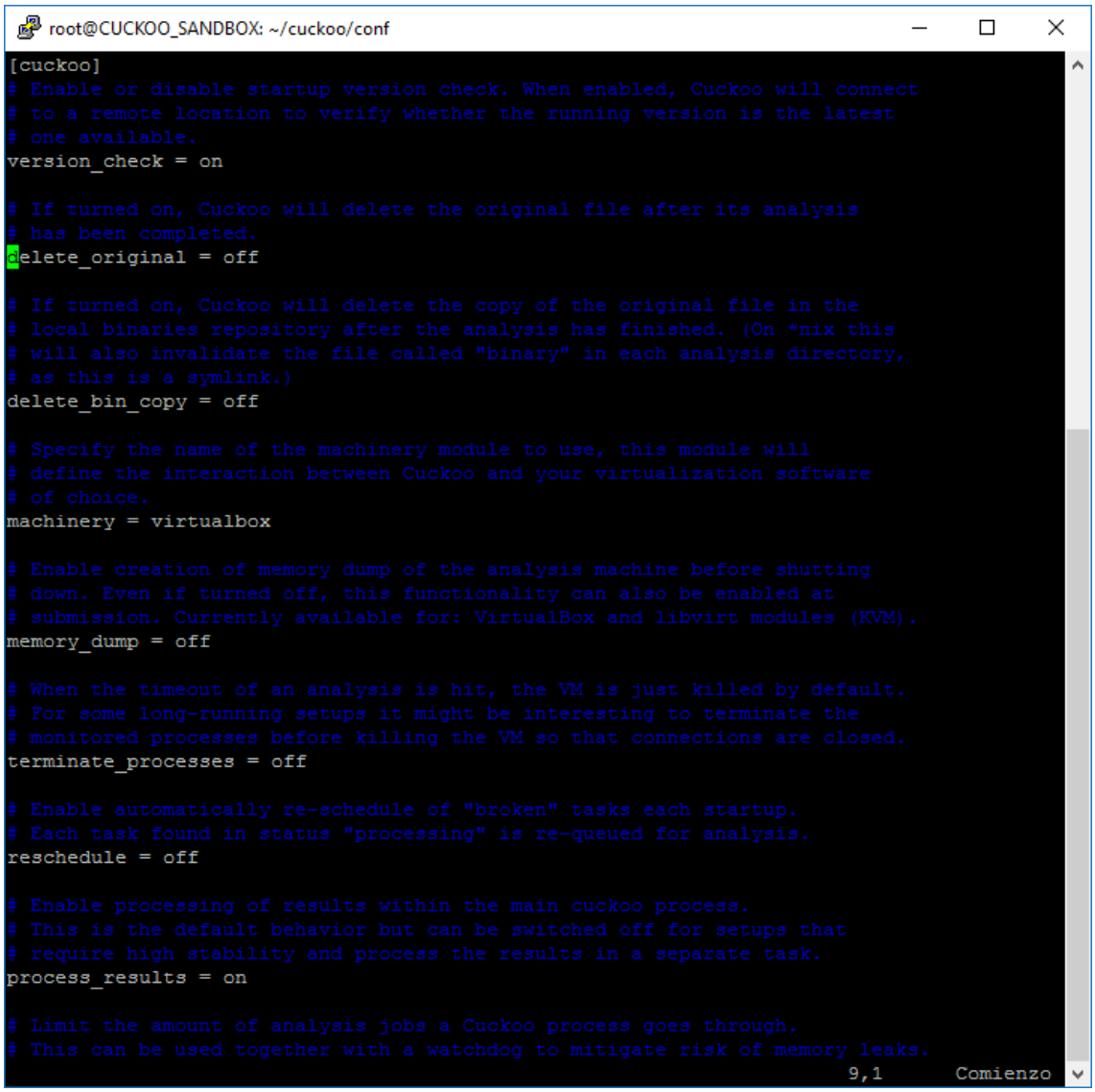
```
        if self.system == "windows":
            root = os.environ["TEMP"]
        elif self.system == "linux" or self.system == "darwin":
            root = "/tmp"
        else:
            self.error_message = \
                "Unable to write malware to disk because the operating " \
                "system could not be identified."
            return False

        file_path = os.path.join(root, name)

        try:
            with open(file_path, "wb") as sample:
                sample.write(data)
        except IOError as e:
            self.error_message = \
                "Unable to write sample to disk: {0}".format(e)
            return False

        return True

    def add_config(self, options):
        """Creates analysis.conf file from current analysis options.
        @param options: current configuration options, dict format.
        @return: operation status.
        """
        if not isinstance(options, dict):
            return False

        config = ConfigParser.RawConfigParser()
        config.add_section("analysis")

        try:
            for key, value in options.items():
                # Options can be UTF encoded.
                if isinstance(value, basestring):
                    try:
                        value = value.encode("utf-8")
                    except UnicodeEncodeError:
                        pass
                                                                    87,1          48%
```

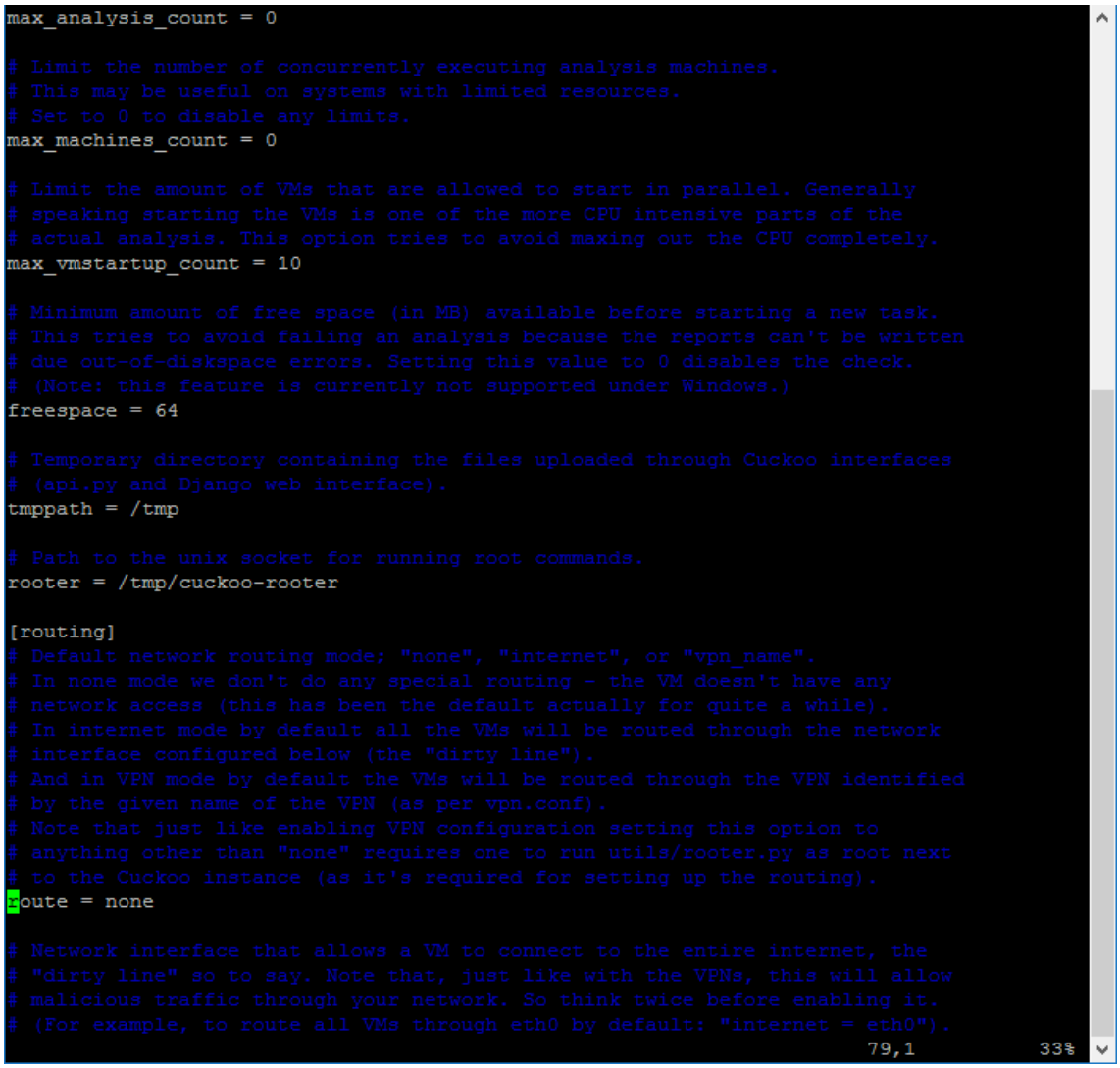
```
                config.set("analysis", key, value)

            config_path = os.path.join(self.analyzer_folder, "analysis.conf")

            with open(config_path, "wb") as config_file:
                config.write(config_file)
        except Exception as e:
            self.error_message = e
            return False

        return True

    def add_analyzer(self, data):
        """Add analyzer.
        @param data: analyzer data.
        @return: operation status.
        """
        data = data.data

        if not self._initialize():
            return False

        try:
            zip_data = StringIO()
            zip_data.write(data)

            with ZipFile(zip_data, "r") as archive:
                archive.extractall(self.analyzer_folder)
        finally:
            zip_data.close()

        self.analyzer_path = os.path.join(self.analyzer_folder, "analyzer.py")
        return True

    def execute(self):
        """Execute analysis.
        @return: analyzer PID.
        """
        if not self.analyzer_path or not os.path.exists(self.analyzer_path):
            return False

                                                              129,1         73%
```



```
        return self.analyzer_pid

    def complete(self, success=True, error="", results=""):
        """Complete analysis.
        @param success: success status.
        @param error: error status.
        """
        if success:
            self.current_status = STATUS_COMPLETED
        else:
            self.current_status = STATUS_FAILED

            if error:
                self.error_message = error

        self.results_folder = results
        return True

if __name__ == "__main__":
    try:
        if not BIND_IP:
            BIND_IP = socket.gethostbyname(socket.gethostname())

        print("[+] Starting agent on %s:%s ..." % (BIND_IP, BIND_PORT))

        # Disable DNS lookup, by Scott D.
        def FakeGetFQDN(name=""):
            return name

        socket.getfqdn = FakeGetFQDN

        server = SimpleXMLRPCServer((BIND_IP, BIND_PORT), allow_none=True)
        server.register_instance(Agent())
        server.serve_forever()
    except KeyboardInterrupt:
        server.shutdown()
~
~
~
~
~
~
                                                          180,1         Final
```

Configuracion cuckoo.conf



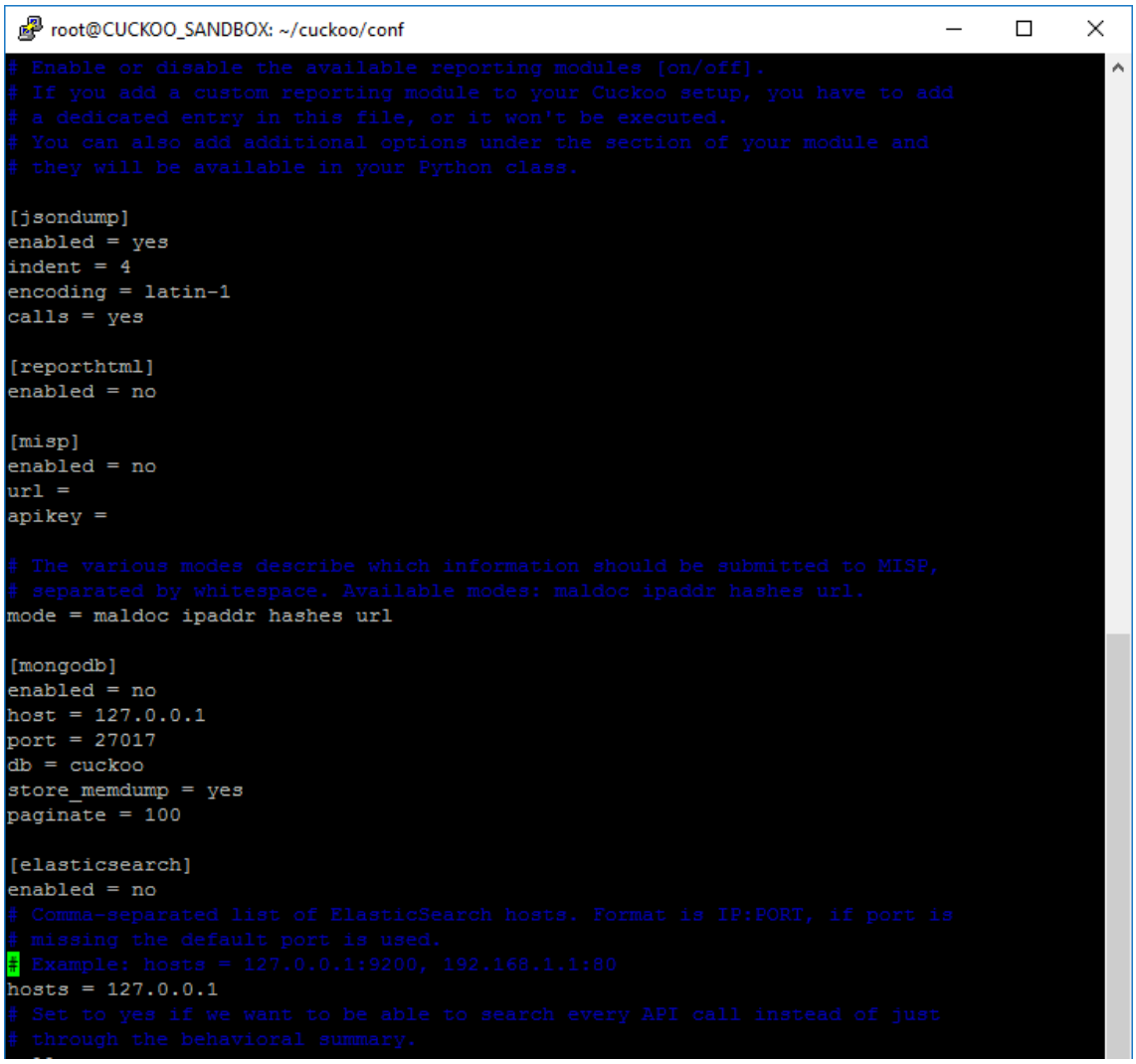


```
max_analysis_count = 0

# Limit the number of concurrently executing analysis machines.
# This may be useful on systems with limited resources.
# Set to 0 to disable any limits.
max_machines_count = 0

# Limit the amount of VMs that are allowed to start in parallel. Generally
# speaking starting the VMs is one of the more CPU intensive parts of the
# actual analysis. This option tries to avoid maxing out the CPU completely.
max_vmstartup_count = 10

# Minimum amount of free space (in MB) available before starting a new task.
# This tries to avoid failing an analysis because the reports can't be written
# due out-of-diskspace errors. Setting this value to 0 disables the check.
# (Note: this feature is currently not supported under Windows.)
freespace = 64

# Temporary directory containing the files uploaded through Cuckoo interfaces
# (api.py and Django web interface).
tmppath = /tmp

# Path to the unix socket for running root commands.
rooter = /tmp/cuckoo-rooter

[routing]
# Default network routing mode; "none", "internet", or "vpn_name".
# In none mode we don't do any special routing - the VM doesn't have any
# network access (this has been the default actually for quite a while).
# In internet mode by default all the VMs will be routed through the network
# interface configured below (the "dirty line").
# And in VPN mode by default the VMs will be routed through the VPN identified
# by the given name of the VPN (as per vpn.conf).
# Note that just like enabling VPN configuration setting this option to
# anything other than "none" requires one to run utils/rooter.py as root next
# to the Cuckoo instance (as it's required for setting up the routing).
route = none

# Network interface that allows a VM to connect to the entire internet, the
# "dirty line" so to say. Note that, just like with the VPNs, this will allow
# malicious traffic through your network. So think twice before enabling it.
# (For example, to route all VMs through eth0 by default: "internet = eth0").
                                                              79,1          33%
```

```
internet = none

# Routing table name/id for "dirty line" interface. If "dirty line" is
# also default gateway in the system you can leave "main" value. Otherwise add
# new routing table by adding "<id> <name>" line to /etc/iproute2/rt_tables
# (e.g., "200 eth0"). ID and name must be unique across the system (refer to
# /etc/iproute2/rt_tables for existing names and IDs).
rt_table = main

# To route traffic through multiple network interfaces Cuckoo uses
# Policy Routing with separate routing table for each output interface
# (VPN or "dirty line"). If this option is enabled Cuckoo on start will try
# to automatically initialise routing tables by copying routing entries from
# main routing table to the new routing tables. Depending on your network/vpn
# configuration this might not be sufficient. In such case you would need to
# initialise routing tables manually. Note that enabling this option won't
# affect main routing table.
auto_rt = yes

[resultserver]
# The Result Server is used to receive in real time the behavioral logs
# produced by the analyzer.
# Specify the IP address of the host. The analysis machines should be able
# to contact the host through such address, so make sure it's valid.
# NOTE: if you set resultserver IP to 0.0.0.0 you have to set the option
# `resultserver_ip` for all your virtual machines in machinery configuration.
ip = 192.168.56.1

# Specify a port number to bind the result server on.
port = 2042

# Force the port chosen above, don't try another one (we can select another
# port dynamically if we can not bind this one, but that is not an option
# in some setups)
force_port = no

# Maximum size of uploaded files from VM (screenshots, dropped files, log)
# The value is expressed in bytes, by default 10Mb.
upload_max_size = 10485760

[processing]
# Set the maximum size of analyses generated files to process. This is used
                                                              121,1         67%
```

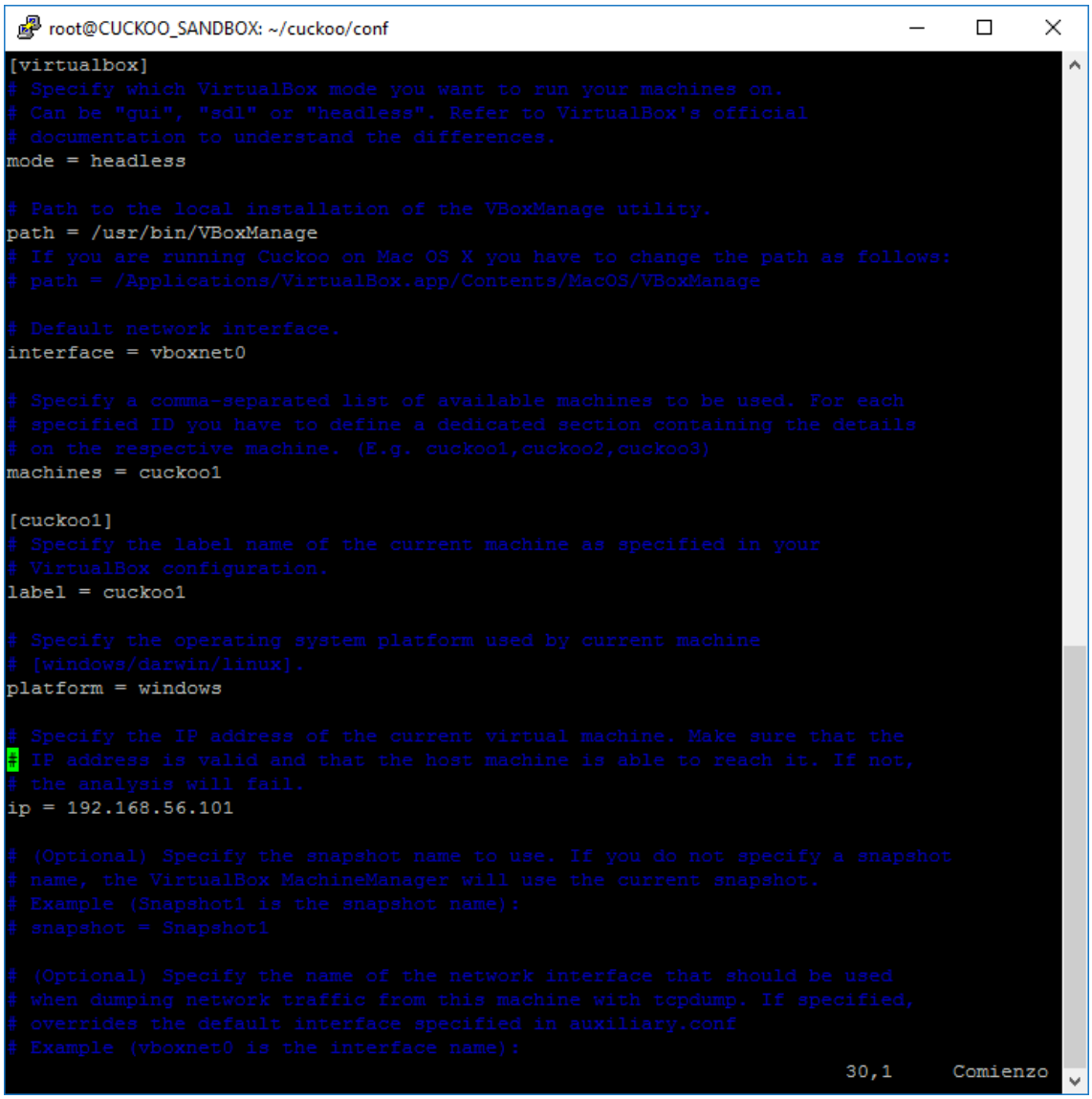
```
# Set the maximum size of analyses generated files to process. This is used
# to avoid the processing of big files which may take a lot of processing
# time. The value is expressed in bytes, by default 100Mb.
analysis_size_limit = 104857600

# Enable or disable DNS lookups.
resolve_dns = on

# Enable PCAP sorting, needed for the connection content view in the web interface.
sort_pcap = on

[database]
# Specify the database connection string.
# NOTE: If you are using a custom database (different from sqlite), you have to
# use utf-8 encoding when issuing the SQL database creation statement.
# Examples, see documentation for more:
# sqlite:///foo.db
# postgresql://foo:bar@localhost:5432/mydatabase
# mysql://foo:bar@localhost/mydatabase
# If empty, default is a SQLite in db/cuckoo.db.
connection =

# Database connection timeout in seconds.
# If empty, default is set to 60 seconds.
timeout =

[timeouts]
# Set the default analysis timeout expressed in seconds. This value will be
# used to define after how many seconds the analysis will terminate unless
# otherwise specified at submission.
default = 120

# Set the critical timeout expressed in (relative!) seconds. It will be added
# to the default timeout above and after this timeout is hit
# Cuckoo will consider the analysis failed and it will shutdown the machine
# no matter what. When this happens the analysis results will most likely
# be lost.
critical = 60

# Maximum time to wait for virtual machine status change. For example when
# shutting down a vm. Default is 60 seconds.
vm_state = 60
                                                          167,1         Final
```

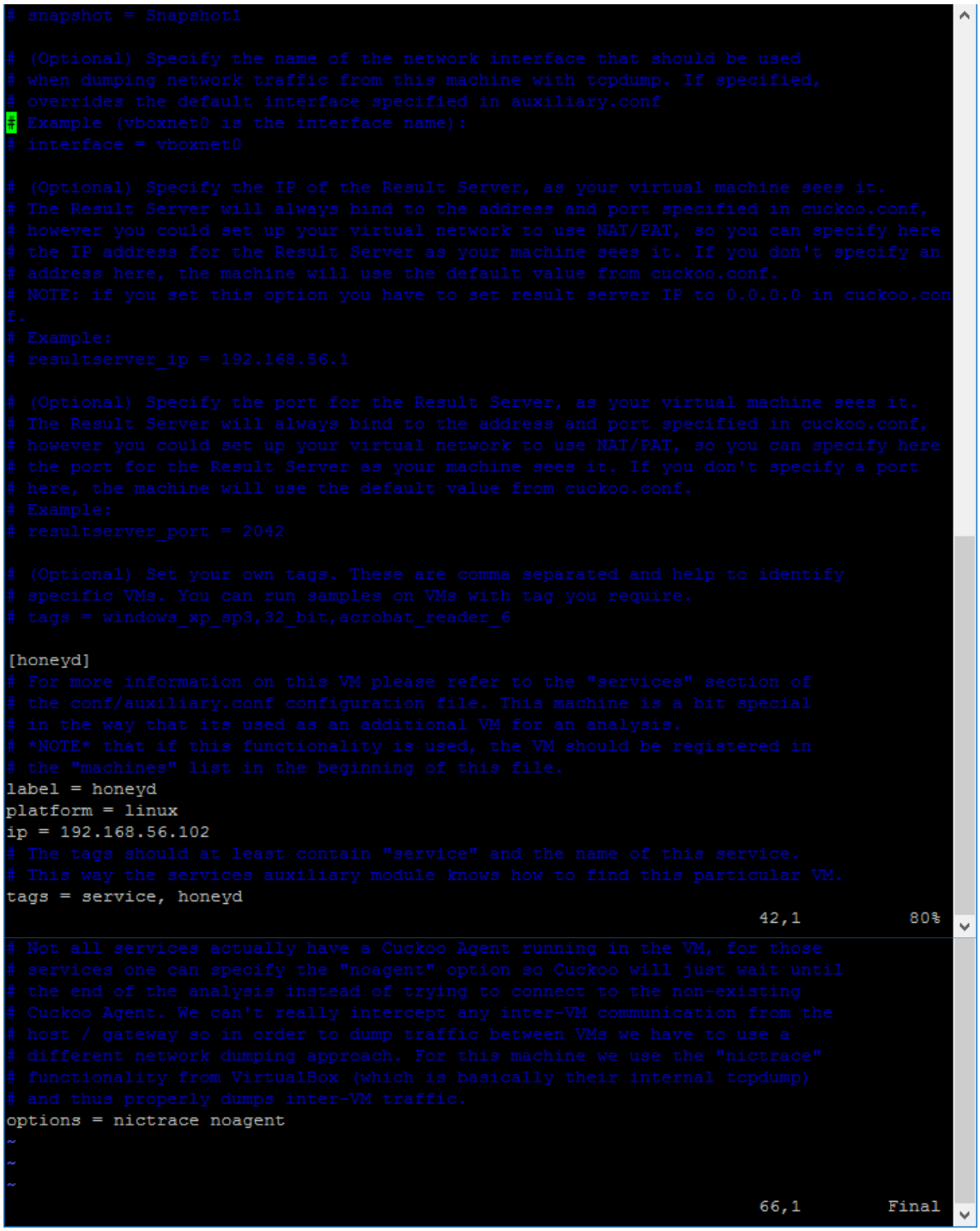
Configuración reporting.conf



```
calls = no
# Index of this Cuckoo instance. If multiple Cuckoo instances connect to the
# same ElasticSearch host then this index (in Moloch called "instance") should
# be unique for each Cuckoo instance.
#
# index = cuckoo
#
# Logging time pattern.  This sets how elasticsearch creates indexes
# by default it is yearly in most instances this will be sufficient
# valid options: yearly, monthly, daily
#
# index_time_pattern = yearly
#
# Cuckoo node name in Elasticsearch to identify reporting host. Can be useful
# for automation and while referring back to correct Cuckoo host.
#
# cuckoo_node = "cuckoo.example.host"

[moloch]
enabled = no
# If the Moloch web interface is hosted on a different IP address than the
# Cuckoo Web Interface then you'll want to override the IP address here.
# host = 127.0.0.1
#
# Following are various configurable settings. When in use of a recent version
# of Moloch there is no need to change any of the following settings as they
# represent the defaults.
#
# moloch_capture = /data/moloch/bin/moloch-capture
# conf = /data/moloch/etc/config.ini
# instance = cuckoo

[notification]
# Notification module to inform external systems that analysis is finished.
# You should consider keeping this as very last reporting module.
enabled = no

# External service URL where info will be POSTed.
# example : https://my.example.host/some/destination/url
url =
                                                              48,1          57%
# Cuckoo host identifier - can be hostname.
# for example : my.cuckoo.host
identifier =

[mattermost]
enabled = no

# Mattermost webhook URL.
# example : https://my.mattermost.host/hooks/yourveryrandomkey
# url=
#
# Cuckoo host URL to make analysis ID clickable.
# example : https://my.cuckoo.host/
# myurl=
#
# Username to show when posting message
username = cuckoo
# What kind of data to show apart from default.
# Show virustotal hits.
# show-virustotal=yes
#
# Show matched cuckoo signatures.
# show-signatures=no
#
# Show collected URL-s by signature "network_http".
# show-urls=no
#
# Hide filename and create hash of it
# hash-filename=no
```

Configuración de virtualbox.conf




snapshot = Snapshot1

(Optional) Specify the name of the network interface that should be used
when dumping network traffic from this machine with tcpdump. If specified,
overrides the default interface specified in auxiliary.conf
Example (vboxnet0 is the interface name):
interface = vboxnet0

(Optional) Specify the IP of the Result Server, as your virtual machine sees it.
The Result Server will always bind to the address and port specified in cuckoo.conf,
however you could set up your virtual network to use NAT/PAT, so you can specify here
the IP address for the Result Server as your machine sees it. If you don't specify an
address here, the machine will use the default value from cuckoo.conf.
NOTE: if you set this option you have to set result server IP to 0.0.0.0 in cuckoo.con
f.
Example:
resultserver_ip = 192.168.56.1

(Optional) Specify the port for the Result Server, as your virtual machine sees it.
The Result Server will always bind to the address and port specified in cuckoo.conf,
however you could set up your virtual network to use NAT/PAT, so you can specify here
the port for the Result Server as your machine sees it. If you don't specify a port
here, the machine will use the default value from cuckoo.conf.
Example:
resultserver_port = 2042

(Optional) Set your own tags. These are comma separated and help to identify
specific VMs. You can run samples on VMs with tag you require.
tags = windows_xp_sp3,32_bit,acrobat_reader_6

[honeyd]
# For more information on this VM please refer to the "services" section of
# the conf/auxiliary.conf configuration file. This machine is a bit special
# in the way that its used as an additional VM for an analysis.
# *NOTE* that if this functionality is used, the VM should be registered in
# the "machines" list in the beginning of this file.
label = honeyd
platform = linux
ip = 192.168.56.102
# The tags should at least contain "service" and the name of this service.
# This way the services auxiliary module knows how to find this particular VM.
tags = service, honeyd
42,1 80%
# Not all services actually have a Cuckoo Agent running in the VM, for those
# services one can specify the "noagent" option so Cuckoo will just wait until
# the end of the analysis instead of trying to connect to the non-existing
# Cuckoo Agent. We can't really intercept any inter-VM communication from the
# host / gateway so in order to dump traffic between VMs we have to use a
# different network dumping approach. For this machine we use the "nictrace"
# functionality from VirtualBox (which is basically their internal tcpdump)
# and thus properly dumps inter-VM traffic.
options = nictrace noagent
~
~
~
66,1 Final