Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

EVALUACIÓN DE HERRRAMIENTAS PARA ESCENARIOS DE GENERACIÓN DE TRÁFICO

TRABAJO FIN DE MÁSTER

Elena Rodríguez Mata

2015

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos

TRABAJO FIN DE MÁSTER

EVALUACIÓN DE HERRAMIENTAS PARA ESCENARIOS DE GENERACIÓN DE TRÁFICO

Autor
Elena Rodríguez Mata

Director
Víctor A. Villagrá González

Departamento de Ingeniería de Sistemas Telemáticos

2015

Resumen

El aumento de la penetración del uso de Internet, del uso de las redes sociales y el streaming de vídeo y audio, ha hecho necesario aumentar el ancho de banda del que disponen los usuarios finales, provocando un incremento considerable del tráfico tcp/ip que tienen que soportar los primeros niveles de agregación. A la par que se aumenta el ancho de banda de las redes, es necesario desarrollar e investigar nuevos sistemas de procesamiento de paquetes que sean capaces de trabajar a estas tasas de línea.

Generar tráfico que refleje adecuadamente diferentes condiciones y topologías de red, es crítico para realizar experimentos válidos sobre bancos de pruebas de la red, por lo que la generación de tráfico real es necesaria para poder comprobar el funcionamiento real de cualquier sistema de procesamiento de paquetes como por ejemplo routers, firewalls, sistemas IDS, etc.

La manera más fiable de generar tráfico real es mediante la reproducción de tráfico previamente capturado.

Abstract

The increasing use of the Internet, social networking and new video and audio streaming applications, has led to an increment in the end users bandwidth demand. This has meant a considerable increment in the TCP/IP traffic that the distribution network has to handle. Due to this increasing bandwidth demand, it is necessary to develop new packet processing systems that are capable of working at higher bit rates. It is critical to perform valid experiments on network testbeds, the use of a traffic generation tool that adequately reflects different network topologies and conditions.

In order to test how the packet processing systems (routers, firewalls, IDS systems, etc) would operate in a real environment, it is required to generate traffic as similar as possible to the traffic generated on a real environment. The easiest way to achieve this is by replicating previously captured real traffic.

Índice general

Índice de figuras

Siglas

ARP	Address resolution protocol
DDOS	Distributed denial of service
DHCP	Dynamic host configuration protocol
DNS	Domain name system
FTP	File transfer protocol
HTTP	Hypertext transfer protocol
ICMP	Internet control message protocol
IP	Internet protocol
IPS	Intrusion prevention system
IDS	Intrusion detection system
LOIC	Low orbit ion cannon
MAC	Media access protocol
NAT	Network address translation
SMTP	Simple mail transfer protocol
SSH	Secure shell
TCP	Transport control protocol
UDP	User datagram protocol
VLAN	Virtual local area network

1 Introducción

Hoy en día, las amenazas se han vuelto más complejas, el área que debe defenderse es más amplia que antes y, además, hay carencia de personal bien preparado en seguridad cibernética. Una oportunidad para el canal es acercar expertos a los clientes para que les ayuden a entender sus riesgos, dimensionarlos y mitigarlos.

Las amenazas presentan una actividad incremental, el perfil de la gente que se dedica a perpetrar estos ataques también ha evolucionado en los últimos años, el estilo ha cambiado, no así el objetivo: la información.

Las vulnerabilidades y amenazas en general alcanzaron su nivel más alto, ya que en octubre pasado se registró un aumento de 14% en las alertas totales acumuladas. Cualquier empresa tiene información que puede ser de valor: desde una lista de clientes, hasta propiedad intelectual, procesos y listas de precios, que es la información que buscan los hackers.

Los métodos de los atacantes incluyen robo socialmente planeado de contraseñas y acreditaciones, infiltraciones escondidas a simple vista, así como explotación de la confianza requerida para transacciones económicas, servicios de gobierno e interacciones sociales.

1.1 Objetivos del Trabajo Fin de Máster

El objetivo del presente documento es proporcionar la información necesaria para la generación de diferentes tipos de tráfico, tanto lícito como malicioso, incluyendo una comparativa de los generadores.

De esta manera se elige la herramienta más idónea para la realización de pruebas de escalabilidad, creando un escenario donde se muestra el tráfico generado diferente según sea la naturaleza del mismo.

Mediante la evaluación, se establecen resultados sobre su rendimiento en un sistema.

En base a estos efectos, se pueden aplicar variaciones en las fases de diseño y desarrollo de las herramientas, dependiendo del hardware del sistema. Adicionalmente, el trabajo proporciona una idea sobre cómo mejorar en el amplio campo de la ciberseguridad apoyándose en los resultados siempre que sean relevantes desde el punto de vista práctico, y la información obtenida se trate con la reserva oportuna.

1.2 Estructura de la memoria

El presente trabajo nos sitúa en un entorno de ciberseguridad, introduciendo en el tema aspectos importantes como las amenazas que puede sufrir un sistema, así como los fallos de seguridad y los ataques existentes.

Se explican los escenarios de generación de tráfico, tanto de simulación, formación y validación, para más adelante presentar los distintos generadores de tráfico lícito y malicioso. Aunque existen multitud de herramientas para ello, nos centramos en algunas para su evaluación

Una vez valoradas, se han sacado en conclusión distintas ideas.

2 Ciberseguridad

Para la generación de tráfico malicioso, se deben tener en cuenta tres conceptos fundamentales, como las vulnerabilidades de un sistema junto con sus fallos de seguridad, y el buen uso de los llamados exploits.

2.1 Vulnerabilidades

Existen varios factores que hacen a un sistema más vulnerable al malware: 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. Por ejemplo, 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. En particular, Microsoft Windows tiene la mayoría del mercado de los sistemas operativos, esto permite a los creadores de malware infectar una gran cantidad de computadoras sin tener que adaptar el software malicioso a diferentes sistemas operativos.

La mayoría del software y de los sistemas operativos contienen bugs que pueden ser aprovechados por el malware. Los ejemplos típicos son los desbordamiento de búfer(buffer overflow), en los cuales la estructura diseñada para almacenar datos en un área determinada de la memoria permite que sea ocupada por más datos de los que le caben, sobre escribiendo otras partes de la memoria. Esto puede ser utilizado por el malware para forzar al sistema a ejecutar su código malicioso.

Originalmente las computadoras tenían que ser booteadas con un disquete, y hasta hace poco tiempo era común que fuera el dispositivo de arranque por defecto. Esto significaba que un disquete contaminado podía dañar la computadora durante el arranque, e igual se aplica a CD y memorias USB con la función AutoRun de Windows la que ya ha sido modificada. Aunque eso es menos común ahora, sigue siendo posible olvidarse de que el equipo se inicia por defecto en un medio removible, y por seguridad normalmente no debería haber ningún disquete, CD, etc., al encender la computadora. Para solucionar este problema de seguridad basta con entrar en la BIOS del ordenador y cambiar el modo de arranque del ordenador.

En algunos sistemas, los usuarios no administradores tienen sobre-privilegios por diseño, en el sentido que se les permite modificar las estructuras internas del sistema, porque se les han concedido privilegios inadecuados de administrador o equivalente. Esta es una decisión de la configuración por defecto, en los sistemas de Microsoft Windows la configuración por defecto es sobre-privilegiar al usuario. Esta situación es debida a decisiones tomadas por Microsoft para priorizar la compatibilidad con viejos sistemas sobre la seguridad y porque las aplicaciones típicas fueron desarrollados sin tener en cuenta a los usuarios no privilegiados.

El malware, funcionando como código sobre-privilegiado, puede utilizar privilegios para modificar el funcionamiento del sistema. Casi todos los sistemas operativos populares y también muchas aplicaciones scripting permiten códigos con muchos privilegios, generalmente en el sentido que cuando un usuario ejecuta el código, el sistema no limita ese código a los derechos del usuario. Esto hace a los usuarios vulnerables al malware contenido en archivos adjuntos de correos electrónicos, que pueden o no estar disfrazados.

Dada esta situación, se advierte a los usuarios de que abran solamente archivos solicitados, y ser cuidadosos con archivos recibidos de fuentes desconocidas. Es también común que los sistemas operativos sean diseñados de modo que reconozcan dispositivos de diversos fabricantes y cuenten con drivers para estos hardwares, algunos de estos drivers pueden no ser muy confiables.

¿Qué es una vulnerabilidad?

Entre las miles de líneas de código que forman un programa, siempre hay algún fragmento que está mal diseñado o que es tan complejo que prever su comportamiento en todos los escenarios resulta una tarea casi imposible.

Cuando no afecta al funcionamiento de la aplicación, ese fragmento de código puede quedarse sin detectar durante mucho tiempo, lo que obviamente retrasa su corrección. Incluso cuando el defecto se conoce, puede que no sea arreglado por falta de recursos.

En el momento en que alguien malintencionado descubra el fallo, es posible que intente aprovecharlo con fines destructivos, convirtiéndolo en un auténtico agujero de seguridad, uno que pueda ser aprovechado mediante un ataque informático (*exploit*).

Un "ataque" puede ser activo o pasivo. Cuando es activo, el ataque daña el sistema hasta que deja de funcionar. Los ataques activos "fuerzan" las entradas del sistema, y se usan, por ejemplo, para tumbar sitios web o desactivar software de importancia vital.

Los ataques pasivos actúan como un ninja: se introducen en el sistema sin causar daño aparente. Mucho más peligrosos a largo plazo, tienen por objetivo la obtención de privilegios en los sistemas atacados, la instalación de programas dañinos (malware) y la sustracción de datos confidenciales.

Si un atacante aprovecha una vulnerabilidad recién descubierta, entonces hablamos de un ataque de día cero. Su potencial dañino es mucho mayor, puesto que inicialmente no hay defensa alguna contra el mismo; es el autor del programa quien debe actuar lo antes posible para evitar el desastre.

Figura 1. Virus de la Policía

En los casos más graves, que es cuando aparecen en aplicaciones muy populares, los ataques se convierten en puertas abiertas para virus de todo tipo, lo que provoca auténticas emergencias a nivel mundial, como la vivida con el Virus de la Policía, que aprovecha una vulnerabilidad de la máquina virtual de Java para entrar en los PC.

¿Por qué los programas tienen vulnerabilidades?

El código de un programa seguro nos da un buen rendimiento en cualquier situación.

Lo que sí puede ocurrir es que los programas sean liberados con fallos de seguridad (*bugs*) no detectados o considerados como poco importantes. O que la aplicación incluya funciones no documentadas, como utilidades de control remoto o actualización de archivos.

Finalmente, muchos agujeros no son detectados sencillamente porque nadie ha intentado encontrarlos ni ha efectuado pruebas de calidad. Pero incluso cuando se pone a prueba un programa, es posible que algunos agujeros de seguridad graves queden sin descubrir.

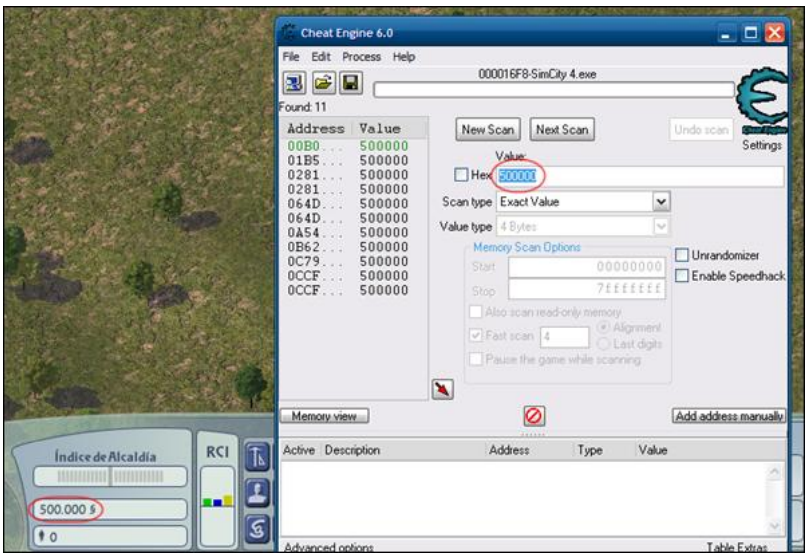


Figura 2. *Cheat Engine*

Esto pasa sobre todo con aplicaciones muy grandes; en el mundo del software, complejidad e inseguridad suelen ir de la mano. La existencia de agujeros de seguridad es inevitable.

¿Cómo consigue alguien encontrar y explotar vulnerabilidades?

Cuando un atacante comprueba un sistema en busca de vulnerabilidades es mucho más agresivo que el usuario promedio. Por ejemplo, puede intentar abrir una brecha introduciendo ejecutables en un formulario con el que solo deberían cargarse imágenes, o saturar de peticiones un programa hasta que este se reinicie o abandone toda resistencia.

Todas las vías de entrada de un programa pueden forzarse o engañarse. Y es que los programas, en el fondo, no son más que filtros por los que pasan los datos que nosotros, los usuarios, les proporcionamos: cuando lo que damos es excesivo, el filtro se atasca y rompe, dejando que pase de todo. Encontrar fallos requiere ingenio, perseverancia y, obviamente, herramientas.

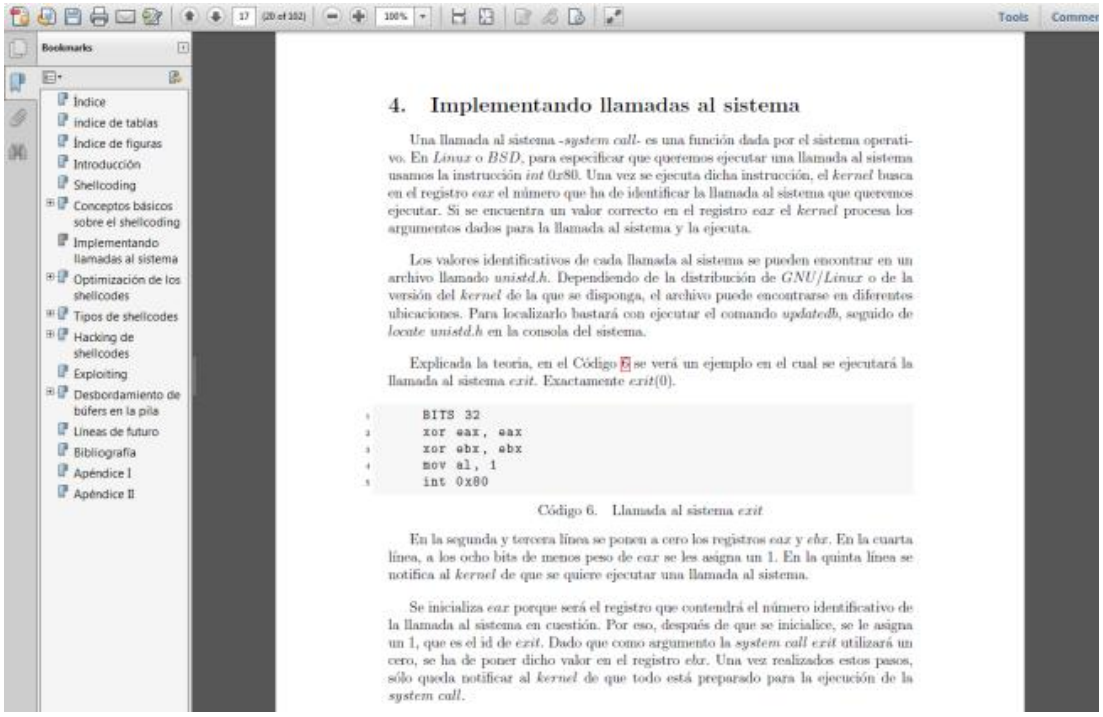


Figura 3. Captura del manual "*Introducción a la explotación de software en sistemas Linux*".

Una vez halladas las vulnerabilidades, el atacante puede crear un exploit o aplicar una técnica ya conocida. A los atacantes que usan herramientas ya existentes reciben en el despectivo apodo de *script-kiddies*.

¿Cómo se solucionan los agujeros de seguridad?

Se tapan o parchean. Los parches informáticos, llamados así en recuerdo a los parches que se aplicaban a las tarjetas perforadas y a las cintas magnéticas de los primeros ordenadores, modifican trozos de código problemáticos de una aplicación. A veces incluyen también mejoras.

Un ejemplo conocido de parche son los que Microsoft aplica a sus productos cada cierto tiempo, bien a través de Windows Update, bien bajo la forma de un *Service Pack*, que no es sino un paquete con muchos parches reunidos en un solo instalador. Otros parches vienen directamente en forma desactualización de una aplicación concreta.

Sin embargo, los parches no siempre arreglan los problemas de seguridad detectados. Ha ocurrido, por ejemplo, con Java, una plataforma que sigue padeciendo vulnerabilidades versión tras versión.

El parcheo de las aplicaciones, además, no es visto con buenos ojos por usuarios y administradores cautelosos, quienes anteponen la estabilidad ante todo lo demás.

¿Cómo evitar los agujeros de seguridad?

Lo primero es mantener navegadores y plugins siempre actualizados. Los parches periódicos de Microsoft, Adobe y Oracle son esenciales para mantener a raya el ataque de malware oportunista. Los antivirus rara vez son eficaces contra los ataques de día cero, pero, si el software está actualizado, el malware chocará contra una pared. Además, se recomienda usar un detector de vulnerabilidades.

Finalmente, el individuo debe mantenerse informado. La suscripción a boletines y blogs de seguridad permiten estar al tanto de los ataques recién descubiertos por la comunidad, que lo convierte en una acción bastante útil antes de que sea demasiado tarde.

2.2 Fallos de seguridad

La mayoría de los problemas de seguridad de la información se debe a un conjunto de fallas básicas en la implementación y desarrollo del proceso de seguridad de la información. Los fallos más comunes y más graves son:

1. La ausencia de una estructura de políticas, normas y procedimientos.

Las normativas (políticas, normas y procedimientos) existen para regir como una organización desea que los recursos informáticos sean usados. Como no existe legislación con ciertos asuntos, como por ejemplo el control de los mensajes de correo electrónico, la organización necesita informar sobre el uso a sus usuarios.

2. La gestión del control de acceso permite una identificación para el uso común.

Muchas organizaciones poseen identificaciones que no son individuales y son utilizadas por un grupo determinado de usuarios. Son casos normales de acceso a la información. Con ese acceso común es muy difícil identificar que usuario hizo determinada acción.

3. Ausencia de un administrador de la información.

Es un factor crítico para el éxito en el proceso de seguridad, el administrador de la información debe ser la persona del área de negocio o del área administrativa responsable de la información. Es quien autoriza (o no) el acceso de los demás usuarios de la empresa a determinada información.

4. Planes de continuidad.

Los planes de continuidad de negocio o planes para situaciones de contingencia debe ser un proceso actualizado. Pero, muchas empresas desarrollan sus planes y estos quedan estancados. Un plan de continuidad tiene que ser activo: actualizado constantemente, probado y en crecimiento continuo.

5. Registros de acciones realizadas.

Se recomienda que exista siempre un registro de acceso y modificaciones en los sistemas. También es importante la existencia del registro de tentativas de acceso y errores de identificación y contraseña.

6. Copias de seguridad.

Las copias de seguridad deben existir por razones legales y por la necesidad de mantener a salvo las transacciones históricas de la empresa. El procedimiento para la creación de las copias de seguridad debe mantenerse permanentemente actualizado. Es muy importante asegurarse que las copias de seguridad que se realizan puedan ser restauradas en caso de necesitarse.

7. Ausencia de un administrador del proceso de seguridad.

La seguridad de la información es una responsabilidad de todos. Sin embargo, un profesional debe ser responsable de la existencia del proceso de seguridad de la información. Pequeñas y medianas empresas suelen disponer de una persona capacitada y dedicada a esta función.

8. La falta de una gestión de riesgo.

Cuando no existe una gestión de riesgo, los análisis de riesgo y de amenazas son hechos aleatoriamente y normalmente sólo cuando se está ante un riesgo inminente. Toda organización debe poseer una gestión de riesgo continua.

9. Usuarios: poco entrenamiento y concientización.

La persona es el factor determinante para el éxito o fracaso del proceso de seguridad de la información en una organización. Cada usuario necesita ser capacitado. Necesita saber sus responsabilidades, lo que puede y lo que no puede hacer.

2.3 Exploit

Exploit (del inglés *to exploit*, "explotar" o 'aprovechar') es un fragmento de software, fragmento de datos o secuencia de comandos y/o acciones, utilizada con el fin de aprovechar una vulnerabilidad de seguridad de un sistema de información para conseguir un comportamiento no deseado del mismo. Ejemplos de comportamiento erróneo: Acceso de forma no autorizada, toma de control de un sistema de cómputo, consecución privilegios no concedidos lícitamente, consecución de ataques de denegación de servicio. Hay que observar que el término no se circunscribe a piezas de software, por ejemplo cuando lanzamos un ataque de ingeniería social, el ardid o discurso que preparamos para convencer a la víctima también se considera un *exploit*. Y poder así capturar cierta información de la víctima a través de este tipo de ataque.

Los *exploits* pueden tomar forma en distintos tipos de software, como por ejemplo *scripts*, virus informáticos o gusanos informáticos.

Tipos de exploit:

Según la forma en la que el *exploit* contacta con el software vulnerable:[2]

- *Exploit* remoto.- Si utiliza una red de comunicaciones para entrar en contacto con el sistema víctima. Por ejemplo puede usar otro equipo dentro de la misma red interna o tener acceso desde la propia Internet.
- *Exploit* local.- Si para ejecutar el exploit se necesita tener antes acceso a el sistema vulnerable. Por ejemplo el exploit puede aumentar los privilegios del que lo ejecuta. Este tipo de exploits también puede ser utilizado por un atacante remoto que ya tiene acceso a la máquina local mediante un exploit remoto.
- *Exploit ClientSide*. - Aprovechan vulnerabilidades de aplicaciones que típicamente están instaladas en gran parte de las estaciones de trabajo de las organizaciones. Ejemplos típicos de este tipo de software son aplicaciones ofimáticas (Ej. Microsoft Office, Open Office), lectores de PDF (Ej. Adobe Acrobat Reader), navegadores (Ej.Internet Explorer, Firefox, Chrome, Safari), reproductores multimedia (Ej. Windows Media Player, Winamp, iTunes). El exploit está dentro de ficheros interpretados por este tipo de aplicaciones y que llega a la máquina objetivo por distintos medios (Ej. email o pendrive). El archivo será usado por el programa y si no es detenido por ningún otro programa (Ej. firewall o antivirus) aprovechará la vulnerabilidad de seguridad.

Las peculiaridades de este tipo de ataques son:

- Requieren la intervención del usuario del lado del cliente. Por ejemplo necesitan que abra cierto archivo o que haga click en cierto link
- Es un ataque asincrónico porque el momento en que se lanza no es el mismo en que se consigue ejecutar el *exploit* (ya que necesita la acción del usuario).
- Se lanza a ciegas, no se sabe qué aplicaciones y versiones utiliza.

Según el propósito de su ataque:

- Curiosidad
- Fama personal
- Beneficio personal
- Espionaje

3 Escenarios de generación de tráfico

Dentro de este ámbito, se toman de especial importancia los escenarios de simulación, formación y validación que a continuación se describen aplicados al tráfico tanto lícito como ilícito.

3.1 Escenarios de simulación

La calidad de las simulaciones depende de la calidad de la herramienta de simulación en sí, de la calidad de los modelos utilizados en la simulación, y la calidad de los modelos de tráfico.

Se ha demostrado a través de la simulación de que la distribución del tamaño del archivo puede dar lugar al tráfico de red similar. Las mediciones de baja resolución que cubren seis días de tráfico de Internet indican que también el número de remitentes activos es auto similar y que a largo plazo las fluctuaciones auto similares en el tráfico de red puede ser causada por un número alternativo de fuentes de tráfico a pesar de que cada uno genera una traza de paquetes distribuidos exponencialmente.

Por otro lado, el análisis de trazas de paquetes detallados de conexiones TCP que cubren tres horas de tráfico indica que la observación de la auto similitud a nivel de IP no depende significativamente de la llegada y salida de activos remitentes (conexiones TCP).

Además, dada la relativa escasez de resultados de análisis para los modelos de tráfico auto similares, la evaluación del desempeño a través de estudios de simulación supone de especial importancia.

Los componentes activos en una red de comunicación son nodos y conmutadores (o routers). Los nodos son las fuentes y los sumideros de tráfico de red. Aquí es donde los flujos de paquetes se inyectan en la red y dejan la red.

Los interruptores son intermedios de reenvío de los paquetes desde un nodo fuente a un nodo de destino. Desde la red perspectiva, un paquete que llega es, básicamente, un triplete de información: el tiempo de aparición, su longitud en bytes y su dirección de destino. El modelo de tráfico define estos tres parámetros interrelacionados de una fuente de tráfico.

Ejemplos de fuentes de tráfico pueden ser conversación de voz, la descarga de archivos FTP, sesiones TCP o sesiones web interactivas.

El mismo punto de vista se puede dar por la carga de trabajo visto por un conmutador. Los paquetes que lleguen a sus puertos de entrada son caracterizados por exactamente los mismos tres parámetros.

Sin embargo, se ve la multiplexación de flujos de paquetes de diferentes modelos de tráfico, de esta manera, se ocultan las características de los flujos de paquetes individuales. Esto es apoyado por un reciente estudio de escalas de tiempo pequeños hecho en mediciones de tráfico troncal de Internet. Se encontró que las fluctuaciones de tráfico son casi sin correlación en escalas pequeñas de tiempo (1-100 ms) y hay una transición a dependencias de largo alcance que se producen entre 100 ms y un segundo, independientemente de la velocidad del enlace, la utilización y la hora. La distribución de tamaños de flujo de paquetes individuales es principalmente en la capa de aplicación dependiente y es suficiente para explicar las correlaciones a escalas de tiempo que van desde segundos a minutos.

Los flujos densos con tiempos entre llegadas predominantemente cortos causaban correlaciones incluso a escalas de tiempo pequeños. La cantidad relativa de los flujos densos también tuvo un impacto en las correlaciones de la corriente del paquete agregado. Los tiempos entre llegadas de los flujos son los resultados del cuello de botella de la velocidad de enlace de las redes subyacentes, junto con el mecanismo de control de retroalimentación de protocolos como TCP.

Se espera que la distribución del tiempo entre llegadas en el flujo agregado es constante en el intervalo de tiempo de simulación.

Las mediciones tomadas de enlaces ligeramente cargados muestran el mismo resultado y parecen implicar los siguientes supuestos:

El flujo de tráfico agregado se pueden tomar para ser estacionario, es decir, sus propiedades estadísticas no cambian significativamente con el tiempo.

Los flujos individuales que componen el flujo agregado pueden tomarse independientes.

Las medidas citadas también parecen apoyar un desacoplamiento de los tres factores al generar flujos de paquetes agregados para simulaciones de red:

El primero genera los tiempos de llegada, a continuación, las longitudes de los paquetes se extraen al azar de acuerdo a su distribución marginal y finalmente las

direcciones de destino se dibujan de nuevo al azar de la distribución de direcciones marginal.

Por otra parte, en qué medida las correlaciones tienen que ser modeladas al generar los tiempos de llegada depende de la longitud deseada de la simulación. Para simulaciones cortas de hasta 100 ms las correlaciones pueden ser en gran parte ignoradas, pero deben incluirse definitivamente para simulaciones más largos más allá de un segundo de tiempo simulado.

Es importante contar con generadores de tiempo de llegada capaces de modelar el nivel deseado de correlación de largo alcance.

Dos aspectos son particularmente importantes: cómo generar el tráfico auto similar en los modelos de simulación y la forma de estimar el grado de auto similitud en una traza de tráfico. [1]

3.2 Escenarios de formación

Tras un espectacular desarrollo tecnológico, una notable despreocupación política y una excesiva confianza de las personas acerca del poder, impacto, penetración e influencia política, social y económica de las Tecnologías de la Información y las Comunicaciones (TIC), se han comenzado a constatar las posibilidades y los riesgos que entraña el ciberespacio y proliferan las estrategias y organizaciones de ciberdefensa y ciberseguridad.

La seguridad de las TIC proporciona a estos sistemas los servicios de seguridad necesarios para garantizar los principios básicos de la seguridad de la información, evitando cualquier pérdida de confidencialidad, disponibilidad, autenticidad e integridad de la misma. Además, deberán proporcionar servicios de auditabilidad y trazabilidad de las actividades que se hayan ejecutado en el sistema. Estos servicios de seguridad deberán, entre otros, identificar y autenticar a los usuarios autorizados, controlar los accesos de estos usuarios en función del *Need-to-Know*, (según las restricciones a los datos), verificar la integridad de la información, registrar y auditar la actividad de los usuarios y controlar las conexiones desde y hacia el sistema clasificado.

Sin embargo, la evolución tecnológica y el aumento en el nivel de amenaza cibernética obligan no solo a dinamizar el proceso de acreditación de los sistemas TIC que tratan información clasificada; sino también a administrar de forma efectiva y eficiente su seguridad. Para ello es necesario realizar una importante inversión

económica en recursos humanos y técnicos para integrar los centros de gestión de seguridad de la información y los centros de explotación de los sistemas mediante interfaces modernos – a nivel de procesos, tecnología y comunicaciones – para obtener y disponer de un conocimiento preciso de ciber-situación, permitiendo con ello la ejecución de análisis de riesgos dinámicos.

Igualmente, es necesario adquirir e implementar herramientas y servicios TIC que permitan mejorar el funcionamiento y la seguridad de los sistemas TIC clasificados. En tercer lugar, es conveniente evolucionar del concepto *Need-To Know*, que describe las restricciones de datos sensibles a un modelo *Work-Related Access Model* que permita llevar a cabo una gestión más granular y precisa de los permisos de acceso a la información; y finalmente se hace necesario realizar planes de concienciación, formación y capacitación continua del personal técnico que administra y gestiona los sistemas TIC, así como de los usuarios finales de los mismos.

Y es que debemos recordar siempre que el ser humano es el eslabón más débil en la "cadena" de la seguridad de los sistemas TIC.

3.3 Escenarios de validación

Los generadores de tráfico se clasifican en cinco categorías de acuerdo con las medidas utilizadas en perspectiva de validación.

Motores de reproducción

Toman previamente tráfico capturado para enviar los paquetes fuera de la interfaz de red al mismo tiempo que son grabados.

La aplicación de reproducción de código abierto más común es tcpreplay que puede utilizar archivos libpcap como entrada. También es capaz de reescribir en las capas 2, 3 y 4 la información de cabecera para diversos fines de prueba. Aunque trabaja en cualquier plataforma UNIX, el rendimiento puede depender en gran medida del medio en el que esté instalado.

Generadores de máximo rendimiento

Los generadores de máximo rendimiento se utilizan generalmente para probar actuaciones de red de extremo a extremo. Aunque la aplicación de estas herramientas se diferencia de la categoría anterior, la validación de las técnicas también utilizan los valores de rendimiento y de IPT.

Iperf es ampliamente utilizado en la red de ingeniería para pruebas de ancho de banda, el retraso del jitter y la tasa de pérdidas, ya que es disponible en varias plataformas.

Bruto es un generador de tráfico de paquetes a nivel de kernel de Linux por lo que garantiza un comportamiento más controlable.

También es una extensión de la misma metodología para una plataforma de hardware específica (Intel IXP2400). Esta solución muestra que esta implementación proporciona valores más precisos, tanto en el rendimiento y el nivel de IPT.

Kute es otro generador de paquetes a nivel de kernel de Linux. La herramienta se puede configurar para cualquier tipo de paquete. Se muestra que tanto la tasa de paquetes y las propiedades IPT se aproximan al valor esperado mejorando a las otras herramientas del generador de tráfico.

Ostinato es un generador de tráfico muy reciente a nivel de usuario disponible para muchas plataformas. Los usuarios pueden definir varios flujos de tráfico a través de una interfaz gráfica de usuario amigable y transmitir fácilmente a la interfaz de red.

Generadores basados en modelos

Estos generadores de tráfico utilizan diferentes modelos estocásticos para la creación de trazas a nivel de paquetes. Este procedimiento plantea la cuestión de si el tráfico generado sigue las mismas estadísticas que se están estableciendo por el modelo.

Deben ser probados para diversas estadísticas que el modelo estocástico implica (IPT, distribución del tamaño del paquete y correlación). Algunos de estos generadores son Tg y Mgen.

Generadores de alto nivel y auto-configurables

Este tipo de generadores de tráfico se basan en el nivel más alto del modelo de tráfico de la red y también son capaces de automáticamente configurar sus parámetros basados en mediciones en vivo. Por lo tanto, la creación de una salida es estadísticamente similar al tráfico inicial.

Harpoon es un generador de tráfico que es capaz de producir tráfico artificial basado en diversas características de flujo.

Además, la herramienta puede analizar mediciones reales para extraer tales valores por lo que se pueden crear tráfico artificial con características que están cerca de la medida original en vivo. Para la validación se utilizan medidas de rendimiento de

bytes, la interconexión de distribución del tiempo, la distribución de tamaño de archivo, frecuencia IP, la distribución y bytes, los paquetes de volumen y el flujo de las distribuciones.

Swing es otro generador de tráfico de alto nivel que es capaz de generar tráfico en función de las características de un rastro real.

La validación de las características agregadas de tráfico utiliza la comparación cuantitativa de la media, la mediana, y los valores de rango inter-cuartil de atributos de base estadísticos.

Se presenta un procedimiento para convertir un flujo TCP para la conexión de vectores y un generador de tráfico llamado TMIX que es capaz de producir tráfico artificial basado en estos vectores. En primer lugar, se valida el modelo mediante la comparación de la Unidad de Datos de programa (ADU).

En segundo lugar, la salida de TMIX es validado por los siguientes parámetros: rendimiento, Round Trip Time (RTT) y el tamaño de los flujos de las distribuciones.

LiTGen puede reproducir tráfico de aplicación (web, correo y P2P). Se presenta la necesidad de establecer correlación entre el número de paquetes y el tiempo entre llegadas.

En las funciones del tráfico del generador D-ITG se satisfacen las condiciones dadas anteriormente. El modo de generación de paquetes artificial de la herramienta utiliza el modelo oculto de Márkov para modelar el tiempo entre paquetes (IPT) y tamaño de paquete de secuencia (PS). Así, en el papel de la distribución de los valores de IPT y PS junto con el rendimiento se comparan entre múltiples mediciones reales y la correspondiente traza generada.

Generadores de escenario especial

Estos generadores de tráfico por lo general representan un tipo específico de condiciones de la red por lo que a menudo ofrecen técnicas de medidas únicas para la situación dada. Por ejemplo, EAR describe un método para transferir una captura de nivel de paquete en una secuencia de eventos que sigue el protocolo IEEE 802.11.

Se propone unas medidas cuantitativas, que son específicas para el entorno de WLAN y por lo tanto no se puede utilizar en cualquier otro escenario de red para otras medidas.

Se valida la herramienta LiTGen por medidas específicas de la web, tales como solicitud de frecuencia o distribuciones de tamaño de documento. En caso del tráfico de YouTube, se centra en las características basadas en proxy caché. [2]

4 Herramientas de tráfico lícito

A continuación se muestran las distintas herramientas de generación de tráfico lícito, para su posterior evaluación con sus características y funciones principales.

4.1 D-ITG

D-ITG (Distributed Internet generador de tráfico) es una plataforma capaz de producir tráfico IPv4 e IPv6 con precisión mediante la replicación de la carga de trabajo de las aplicaciones actuales de Internet. Al mismo tiempo D-ITG es también una herramienta de medición de red capaz de medir las métricas de rendimiento más comunes (por ejemplo, el rendimiento, retardo, jitter, pérdida de paquetes) a nivel de paquete.

D-ITG puede generar tráfico siguiendo modelos estocásticos para el tamaño de los paquetes (PS) y el tiempo de salida entre (IDT) que imitan el comportamiento de protocolo de nivel de aplicación. Al especificar las distribuciones de variables aleatorias IDT y PS, es posible elegir diferentes procesos de renovación para la generación de paquetes: mediante el uso de la caracterización y modelización de los resultados de la literatura, la D-ITG es capaz de replicarse propiedades estadísticas de tráfico de aplicaciones diferentes conocidos (por ejemplo Telnet, VoIP - G.711, G.723, G.729, detección de actividad de voz, RTP comprimido - DNS, juegos en red).

En la capa de transporte, D-ITG actualmente soporta TCP (Transmission Control Protocol), UDP (User Datagram Protocol), SCTP (Corriente del Protocolo de control de transmisión), y DCCP–(datagramas de Control de Congestión Protocolo). También soporta ICMP (Internet Control Message Protocol). Entre las diversas características descritas a continuación, el modo pasivo FTP-como también se apoya para llevar a cabo experimentos en presencia de NATs, y es posible ajustar el TOS (DS) y los campos de cabecera IP TTL.

4.1.1 Arquitectura y componentes

La arquitectura de D-ITG comprende diferentes componentes:

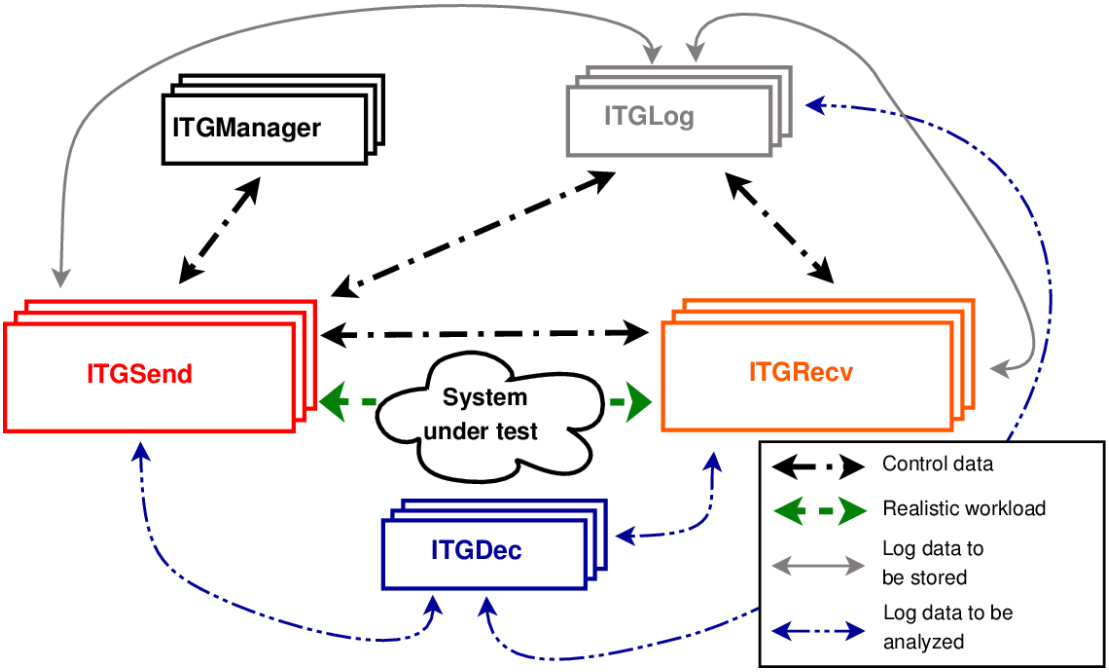


Figura 4. Arquitectura de D-ITG

Los componentes del generador son los siguientes:

- **ITGSend: Remitente de D-ITG**

El componente ITGSend es responsable de generar los flujos de tráfico y puede trabajar en tres modos diferentes:

- **Single-flujo** - leer la configuración del flujo de tráfico solo para generar hacia una sola instancia ITGRecv desde la línea de comandos;
- **Multi-flujo** - leer la configuración del tráfico de múltiples flujos de generar hacia una o más instancias ITGRecv desde un archivo de script. El guión es de una línea para cada flujo de tráfico, que incluye un conjunto de opciones de línea de comandos como en el modo de flujo único;
- **Daemon** - ejecutarse como un demonio que escucha en un socket UDP para obtener instrucciones y puede ser controlado de forma remota utilizando la API de D-ITG.

Cada flujo de tráfico generado es descrito por dos procesos estocásticos relacionados con Tamaño de paquetes (PS) y la hora de salida a Internet (IDT), a través del cual los perfiles de tráfico bien definidos se pueden generar, emulando protocolos de aplicaciones como VoIP, DNS, etc...

PS e IDT serie también se puede cargar desde un archivo para cada flujo. ITGSend puede registrar información sobre cada paquete enviado o recibido, cuando se ejecuta en *One Way* o *Ida y Vuelta* modo respectivamente (véase más adelante). En el primer caso, las marcas de tiempo (y otros datos) de los paquetes enviados se almacenan, mientras que en el segundo caso, las marcas de tiempo (y otros datos) de los paquetes enviados y recibidos se almacenan. Para cada flujo de la dirección IP de origen se puede especificar, que es útil para los hosts multitarjeta.

- **ITGRecv: Receptor de la Plataforma D-ITG**

El componente ITGRecv se encarga de recibir tráfico paralelo de múltiples flujos generados por una o más instancias ITGSend. Normalmente se ejecuta como un demonio multi-hilo que escucha en un socket TCP para peticiones de recepción del tráfico entrante. Cada vez que se recibe una petición desde la red, se crea un nuevo hilo, que realiza todas las operaciones relacionadas con la nueva solicitud (por ejemplo, recibir los paquetes del flujo). Los números de puerto en el que ITGRecv recibirán cada flujo y de cualquier actividad de registro requerido en el lado receptor puede ser controlado remotamente por ITGSend. Un protocolo de señalización específica, llamada TSP, permite ITGRecv y ITGSend configurar adecuadamente y gestionar el proceso de generación de tráfico.

- **ITGLog: Logger de D-ITG**

El componente ITGLog es responsable de recibir y almacenar información de registro enviado por ITGSend y ITGRecv. Se ejecuta como un demonio multi-hilo que escucha en un socket TCP para solicitudes de registro de entrada. Entre la información que se reciba a través de TCP o UDP en números de puerto asignados dinámicamente en el rango [9003-10003].

- **ITGDec: Decodificador de D-ITG**

El componente ITGDec es responsable de decodificar y analizar los archivos de registro almacenados durante los experimentos llevados a cabo mediante el uso de D-ITG.

ITGDec analiza los archivos de registro generados por ITGSend y ITGRecv y calcula los valores promedio de la tasa de bits, retardo y jitter en toda la duración del experimento o en intervalos de tiempo de tamaño variable.

ITGDec produce los siguientes resultados sobre cada flujo y sobre todo el conjunto de flujos:

- Informes Sintético:

    - Duración Experimento
    - Los paquetes transferidos
    - Bytes de carga útil transferidos
    - Sólo ida / retardo de ida y vuelta (mínimo, máximo, promedio, desviación estándar)
    - Bitrate promedio
    - Tasa media de paquetes
    - Paquetes perdidos
    - Paquetes duplicados
    - Eventos de pérdida
    - El tamaño promedio de pérdida de ráfaga
    - Primero último número / secuencia
- Muestreados QoS métricas time series:
    - Velocidad de bits [Kbps] (es decir goodput)
    - Sólo ida / retardo de ida y vuelta [ms]
    - Jitter [ms] (es decir, la variación del retardo)
    - La pérdida de paquetes [pps] (es decir, la pérdida de paquetes por segundo)

4.1.2 Características

D-ITG es capaz de generar múltiples flujos unidireccionales de muchos remitentes hacia muchos receptores, cada uno de ellos con las siguientes características. [3]

- Personalizable propiedades de nivel de flujo
    - duración
    - comenzará el tiempo
    - número total de paquetes
    - número total de KBytes
- Compatibles Layer-3 características
    - protocolos: IPv4, IPv6
    - campos de cabecera personalizables:
        - direcciones IP de origen y de destino
        - interfaz de origen de enlace (para dispositivos multitarjeta)
        - valor inicial TTL
        - Byte DS
    - NAT transversal: modo pasivo FTP-como
- Compatibles Layer-4 características

    - protocolos: TCP, UDP, ICMP, DCCP, SCTP
    - campos de cabecera personalizables:
        - números de puerto de origen y destino
- Compatibles Layer-7 características
    - PS estocástico predefinidas (tamaño del paquete) y de IDT (Inter Hora de salida) perfiles:
        - Telnet
        - DNS
        - Quake3
        - CounterStrike (activos e inactivos)
        - VoIP (G.711, G.729, G.723)
    - Contenido de la carga útil: al azar o leer desde el archivo
    - Procesos estocásticos compatibles tanto para PS e IDT:
        - Distribuciones soportadas son uniforme, constante, exponencial, Pareto, Cauchy, Normal, Poisson, Gamma, Weibull
        - Selección de la semilla aleatoria explícita para replicar el mismo proceso estocástico
        - Cargando PS IDT serie de archivo
- QoS a nivel de paquetes métricas
    - Bitrate
    - Tasa de paquetes
    - Una demora manera (requiere la sincronización de relojes)
    - Round Trip Time
    - Jitter
    - Paquete perdido

4.2 Ostinato

Ostinato es un generador de tráfico de código abierto y analizador con una interfaz gráfica de usuario amigable. Envía paquetes de trabajo de varios flujos con diferentes protocolos a distintas velocidades. [4]

4.2.1 Características

- Se ejecuta en Windows, Linux, BSD y Mac OS X (probablemente se ejecutará en otras plataformas también con poca o ninguna modificación, pero esto no ha sido probado)

- Abrir, editar, reproducir y guardar archivos PCAP
- Soporte para los protocolos estándar más comunes
    - Ethernet / 802.3 / SNAP LLC
    - VLAN (con QinQ)
    - ARP, IPv4, IPv6, IP-en-IP alias Túneles IP (6over4, 4over6, 4over4, 6over6)
    - TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD
    - Cualquier texto basado en el protocolo (HTTP, SIP, RTSP, NNTP, etc.)
    - Más protocolos en las obras ...
- Modificar cualquier campo de cualquier protocolo (algunos protocolos permiten cambiar los campos de paquetes con cada paquete en tiempo de ejecución, por ejemplo el cambio de direcciones IP / MAC)
- Usuario proporcionado Hex Dump - especificar algunos o todos los bytes en un paquete
- Protocolos de pila en cualquier orden arbitrario
- Crear y configurar múltiples flujos
- Configure las tasas de flujo, estalla, no. de paquetes
- Cliente individual puede controlar y configurar varios puertos en varios equipos de generación de tráfico
- Control exclusivo de un puerto para evitar que el sistema operativo desde el envío de paquetes perdidos proporciona un entorno de pruebas controlado
- Estadísticas ventana muestra el puerto en tiempo real recibir / transmitir estadísticas y tarifas
- Captura paquetes y verlos (necesita Wireshark para ver los paquetes capturados)
- Marco para agregar nuevos constructores de protocolo fácilmente.

4.3 Netniff-NG

Netsniff-ng es un analizador de redes Linux libre y kit de herramientas de redes originalmente escrito por Daniel Borkmann. Su ganancia de rendimiento que se alcanza a través de mecanismos zero-copy de los paquetes de red (RX_RING, TX_RING), por lo que el núcleo Linux no necesita copiar los paquetes desde el espacio del núcleo al espacio de usuario a través de llamadas al sistema como recvmsg ().libpcap, comenzando con la liberación 1.0.0, también es compatible con el mecanismo de copia cero en Linux para capturar (RX_RING), por lo que los programas que utilizan libpcap también utilizan ese mecanismo en Linux.

Figura 5. Logo de Netsniff-NG

Netsniff-ng fue creado inicialmente como un sniffer de red con el apoyo de la interfaz de paquetes-mmap kernel Linux para los paquetes de red, pero más tarde, se han añadido más herramientas para que sea una herramienta útil, como la suite iproute2, por ejemplo. A través de la interfaz zero-copy del kernel, el procesamiento de paquetes eficiente puede llegar incluso en hardware. Por ejemplo, se ha llegado a Gigabit Ethernet a velocidad de cable con Trafgen de netsniff-ng. El conjunto de herramientas-netsniff ng no depende de la biblioteca libpcap. Por otra parte, no se necesitan parches especiales del sistema operativo para ejecutar el kit de herramientas. netsniff-ng es software libre y ha sido puesto en libertad bajo los términos de la Licencia Pública General de GNU versión 2. [5]

El juego de herramientas se compone actualmente de un analizador de redes, capturador de paquetes y replayer, un generador de tráfico a velocidad de cable, un túnel IP multiusuario cifrada, un compilador Berkeley Packet Filter, la creación de herramientas estadísticas de redes y un trazado de ruta del sistema autónomo:

Netsniff-ng es un analizador de red rápida basada en mecanismos nmap. Se puede grabar archivos pcap a disco, reproducirlas y hacer un fuera de línea y el análisis en línea. Los archivos pcap también son compatibles con tcpdump o capturas de Wireshark.

Trafgen es un generador de tráfico de red multi-hilo basado en mecanismos nmap. Tiene su propio, de bajo nivel lenguaje de configuración de paquetes basado en macro flexible. Tiene una velocidad significativamente mayor que mausezahn y se acerca mucho a pktgen, pero se extiende desde el espacio de usuario. Las trazas pcap también se pueden convertir en una configuración de paquetes Trafgen.

mausezahn es un generador de paquetes de alto nivel que se puede ejecutar en un dispositivo de hardware-software y viene con un Cisco-como CLI. Se puede elaborar casi cada posible o imposible de paquetes. Por lo tanto, se puede utilizar, por ejemplo, para probar el comportamiento de la red en circunstancias extrañas (prueba de esfuerzo, los paquetes con formato incorrecto) o para probar los dispositivos de hardware-software para varios tipos de ataques.

bpfc es un compilador Berkeley Packet Filter (BPF) que entiende el idioma original BPF desarrollada por McCanne y Jacobson. También es compatible con extensiones de filtro de Linux. Esto puede ser especialmente útil para los filtros más complicados, que los filtros de alto nivel no apoyan.

ifpps es una herramienta que proporciona estadísticas del kernel de Linux. Recoge datos estadísticos directamente de archivos procfs y no se aplica ninguna de monitoreo de tráfico el espacio de usuario que falsificar estadísticas sobre altas tasas de paquetes. Por inalámbrica, datos sobre conectividad enlace se proporciona también.

FlowTop es una herramienta de seguimiento de conexión superior-como que se puede ejecutar en un host o router final. Es capaz de presentar TCP o UDP flujos que han sido recogidos por sistema netfilter del kernel. Se muestra la información GeoIP y máquina de estado TCP. Además, en los hosts finales FlowTop puede mostrar los PID y los nombres de aplicación que se relacionan con los flujos. Ningún usuario vigilancia del tráfico espacial se hace, por tanto, todos los datos se reunieron por el núcleo.

Curvetun es un túnel multiusuario ECDH ligero para Linux. Utiliza el interfaz TAP Linux TUN y apoyos {IPv4, IPv6} {más de IPv4, IPv6} con UDP o TCP como protocolos de transporte. Los paquetes son codificados de extremo a extremo por un cifrado de flujo simétrico (Salsa20) y autenticados por un MAC (Poly1305), donde las claves previamente se han calculado con el protocolo de acuerdo de claves ECDH (Curve25519).

Astraceroute es un sistema autónomo (AS) de utilidad de trazado de ruta. A diferencia de traceroute o tcptraceroute, no solamente la pantalla salta, sino también su AS información al que pertenecen, así como información GeoIP y otras cosas interesantes. Por defecto, se utiliza una sonda de paquetes TCP y cae de nuevo a sondas ICMP en caso de que se haya recibido ninguna respuesta ICMP.

Concluyendo, el kit de herramientas se divide en pequeñas empresas de servicios públicos, que son útiles o no necesariamente están relacionados entre sí. Cada programa por sí mismo llena un vacío como ayudante en su depuración de red.

Los paquetes específicos de distribución están disponibles para todas las principales distribuciones de sistemas operativos como Debian o Fedora Linux. También se ha añadido a la Red Forensic Toolkit de Xplico, GRML Linux, SecurityOnion, y para el Kit de herramientas de seguridad de red. El conjunto de herramientas Netsniff-ng también se utiliza en el mundo académico.

A continuación se muestra el ejemplo de ataque SYN:

```
/* Trafgen example file: TCP SYN attack
* Used when developing the iptables solution for SYNPROXY
 *     http://rhelblog.redhat.com/2014/04/11/mitigate-tcp-syn-flood-attacks-with-red-hat-enterprise-linux-7-beta/
 *
 * This file need to be run with --cpp for c-preprocessor call.
 * Command example:
 * trafgen --cpp --dev dummy0 --conf syn_attack01.trafgen --cpu 2 --verbose
 *
 * Note: dynamic elements "drnd()" make trafgen slower
 */

#define ETH_P_IP     0x0800

#define SYN          (1 << 1)
#define ACK          (1 << 4)
#define ECN          (1 << 6)

{
  /* --- Ethernet Header --- */
  /* NEED ADJUST */
  0x00, 0x12, 0xc0, 0x02, 0xac, 0x56,  # MAC Destination
  0x00, 0x12, 0xc0, 0x02, 0xac, 0x5a,  # MAC Source
  const16(ETH_P_IP),
```

```
/* IPv4 Version, IHL, TOS */
0b01000101, 0,
/* IPv4 Total Len */
const16(40),
/* IPv4 Ident */
//drnd(2),
const16(2),

/* IPv4 Flags, Frag Off */
0b01000000, 0,
/* IPv4 TTL */
64,
/* Proto TCP */
0x06,
/* IPv4 Checksum (IP header from, to) */
csumip(14, 33),

/* NEED ADJUST */
198, 18, 51, drnd(1), # Source IP
198, 18, 51, 2,       # Dest IP

/* TCP Source Port */
drnd(2),
/* TCP Dest Port */
const16(80),
```

```
  /* TCP Sequence Number */
  drnd(4),
  /* TCP Ackn. Number */
  c32(0), /* NOTICE ACK==zero with SYN packets */

  /* TCP Header length + Flags */
  //const16((0x5 << 12) | SYN | ECN)   /* TCP SYN+ECN Flag */
  //const16((0x5 << 12) | SYN | ACK)   /* TCP SYN+ACK Flag */
  const16((0x5 << 12) | SYN)           /* TCP SYN Flag */
  //const16((0x5 << 12) | ACK)         /* TCP ACK Flag */

  /* Window Size */
  const16(16),
  /* TCP Checksum (offset IP, offset TCP) */
  csumtcp(14, 34),
  const16(0), /*PAD*/
}
```

4.4 PackETH

Es una herramienta GUI y CLI generadora de Ethernet de paquetes. Te permite crear y enviar un posible paquete o secuencia de paquetes en el enlace Ethernet. Es muy fácil de usar, potente y compatible con muchos ajustes de parámetros al enviar secuencia de paquetes. [6]

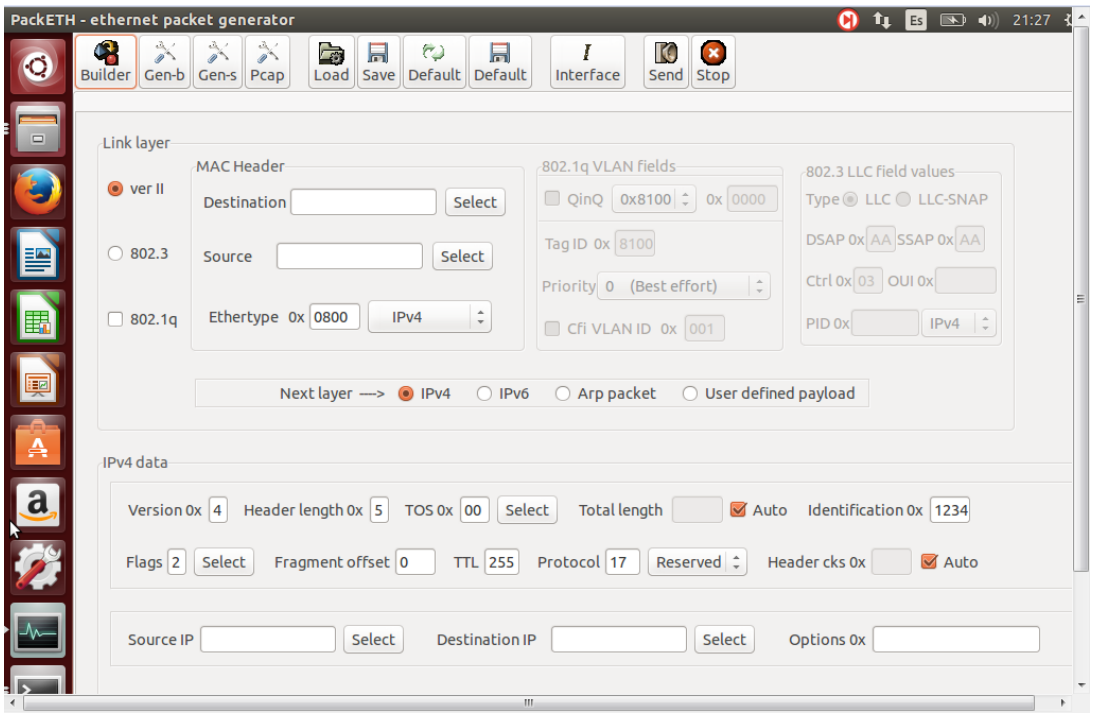


Figura 6. Interfaz de usuario de PackETH en Linux.

4.5 Tomahawk

Tomahawk es una poderosa herramienta diseñada específicamente para medir la in/seguridad de los sistema NIPS (sistemas de prevención de intrusiones basado en redes). Esto lo logra enviando ciertos paquetes que contienen "partes de los ataques" más conocidos y espera a que el sistema de "Reensamblado" de los NIPS falle.

Por otra parte también se pueden realizar valoraciones del comportamiento de los NIPS bajo "carga extrema" de ataques. Es muy importante en el mundo real, dado que la mayoría de los NIPS más conocidos no se desempeñan del todo bien ante estas

circunstancias y esto puede ser aprovechado por la mayoría de los Worms o gusanos de la Internet.

Tomahawk es una herramienta *Open Source* perfecta para comprobar los Sistemas de Detección/Prevención de Intrusiones (*IDS/IPS*).

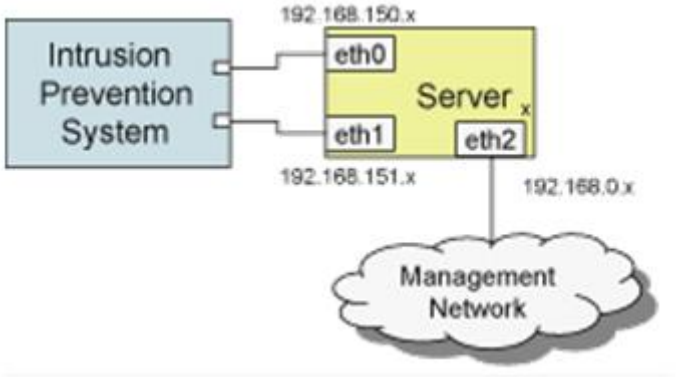


Figura 7. Configuración de Tomahawk conectada a un IPS

Se tiene un servidor Tomahawk con tres tarjetas de red (NICs): una de gestión (eth2) y dos para pruebas (eth0 y eth1). Tomahawk reproduce una o más capturas de red en formato pcap/tcpcump.

Cuando Tomahawk reproduce el pcap, los paquetes llegan a las interfaces del IPS en el mismo orden en el que hubiesen sido recibidas si la secuencia hubiese sido recibida.

El siguiente ejemplo, extraido del tutorial de Tomahawk, ilustra el funcionamiento: si tenemos un pcap consistente en los tres paquetes clásicos del establecimiento de una sesión TCP (SYN / SYN-ACK / ACK):

Paquete 1 (SYN): ip.src = 172.16.5.5 ip.dest = 172.16.5.4

Paquete 2 (SYN-ACK): ip.src = 172.16.5.4 ip.dest = 172.16.5.5

Paquete 3 (ACK): ip.src = 172.16.5.5 ip.dest = 172.16.5.4

Cuando Tomahawk lee el primer paquete encuentra por primera vez la dirección 172.16.5.5 en el campo origen y la dirección 172.16.5.4 en el campo destino, Tomahawk asigna las direcciones 172.16.5.5 con el cliente y 172.16.5.4 con el servidor. A partir de ese momento los paquetes del servidor son transmitidos por la interfaz eth1, y los paquetes cliente son transmitidos por la interfaz eth0.

Las herramientas para probar NIPS han sido limitadas en funcionalidad. Están diseñadas normalmente para probar otros productos, tales como interruptores (por ejemplo, SmartBits / IXIA), infraestructuras de servidor (por ejemplo, WebAvalanche), o firewalls y sistemas de detección de intrusiones (IDS o Firewall Informer Informer).

Tomahawk puede ser utilizado para probar el rendimiento y bloqueo de capacidades de los sistemas de prevención de intrusiones basados en red (PIN).

Las pruebas de rendimiento

El rendimiento de muchos NIPSS es altamente dependiente de la mezcla de protocolo. Un NIPS debe volver a montar e inspeccionar los datos a nivel de aplicación encapsulados en el tráfico de red. Se debe decodificar protocolos de red y nivel de aplicación. Debido a que algunos protocolos son más computacionalmente intensivas para decodificar que otros, el efecto de un NIPS tiene sobre el rendimiento de la red puede ser altamente dependiente de la mezcla de protocolo que debe fluir a través de los NIPS.

Tomahawk puede probar el rendimiento de un NIPS utilizando la combinación más realista posible de protocolos: uno obtenido tomando una muestra del tráfico de la red y repetirlo.

Tomahawk también puede probar la conexión / segunda calificación de un NIPS. Al capturar una traza de paquetes que contiene una configuración de conexión simple y desmontaje (6 paquetes: SYN, SINC_CONF, ACK, FIN_ACK, FIN_ACK, ACK) y reproducir el tráfico utilizando Tomahawk, un solo PC puede generar 25-50000 conexiones / segundo de la red tráfico. Con 3 PCs de bajo costo, a unos 90K conexiones / seg se pueden generar, lo suficiente como para poner a prueba los límites de cualquier NIPS.

Pruebas de seguridad

Además de las pruebas de throughput, Tomahawk puede probar las capacidades de bloqueo de un NIPS mediante la reproducción de ataques incrustados en trazas de paquetes. Tomahawk informa si un ataque se completa o se bloquea, lo que permite una verificación independiente del ataque bloqueando capacidades en un NIPS.

Por repitiendo el mismo ataque cientos de veces, Tomahawk también puede probar cómo fiable bloquea un NIPS un ataque. Un NIPS que bloquea un ataque sólo el 9 en 10 veces no vale mucho en un brote de gusano.

Comandos y ejemplos

Las siguientes secciones detallan algunos de los comandos con ejemplos para Tomahawk.

Reproducir un archivo

La siguiente línea reproduce el archivo outlook.pcap vez:

tomahawk -l 1 -f outlook.pcap

La salida se muestra a continuación:

Completado un bucle de outlook.pcap

Terminado un bucle de outlook.pcap: tiempo de espera: 0 Retrans: 0 Enviados: 843 RECV: 843

La primera línea se imprime cuando Tomahawk termina de cargar el pcap y comienza a transmitir.

La segunda línea:

Completado un bucle de outlook.pcap , imprime cuando completa la grabación de la pcap. Si la repetición pcap no terminó (quizás porque estaba bloqueado por un IPS), la palabra "Finalizado" se sustituye por la palabra "tiempo de espera".

El último grupo de líneas:

Terminado un bucle de outlook.pcap: tiempo de espera: 0 Retrans: 0 Enviados: 843 RECV: 843

Se dan estadísticas agregadas para todas las repeticiones de outlook.pcap. Esto incluye el número de paquetes enviados, cantidad recibida, y el número de retransmisiones.

Replicar un archivo varias veces

Para jugar este pcap cinco veces seguidas, se utilizaría la siguiente:

tomahawk -l 5 -f outlook.pcap

El parámetro "-l" controla el número de bucles. Esto genera la siguiente salida:

Completado un bucle de outlook.pcap

Completado un bucle de outlook.pcap

Completado un bucle de outlook.pcap

Completado un bucle de outlook.pcap

Completado un bucle de outlook.pcap

Terminado 5 bucles de outlook.pcap: el tiempo de espera: 0 Retrans: 0 Enviados: 4215 RECV: 4215

Las estadísticas de resumen indican que las 5 repeticiones de outlook.pcap terminaron sin ser bloqueadas.

Si un ataque se repite a través de un IPS, el IPS debe bloquear el ataque. Debido a que un IPS menudo bloquea una corriente (identificado por un anfitrión / cuádruple puerto), Tomahawk da a cada repetición del ataque su puerto único. Suponiendo que el PCAP contiene 2 direcciones, Tomahawk reescribe los paquetes de modo que la primera repetición del ataque es de 10.0.0.1 a 10.0.0.2, la segunda repetición es de 10.0.0.3 a 10.0.0.4, y así sucesivamente.

Iniciar Control de Dirección

Se puede controlar la dirección de inicio con la bandera "-a". Por ejemplo:

tomahawk -l 5 -f outlook.pcap -a 11.0.0.1

Inicia ataques de repetición en 11.0.0.1. Esta bandera es útil si está utilizando varias máquinas para generar carga. Un uso típico se encarna en el siguiente fragmento de una secuencia de comandos:

ADDR = $ (ifconfig eth0 | grep 'inet addr' | sed 's /\./ / g' | awk '{print $ 5}') tomahawk -a 10. $ ADDR.0.1...

La primera línea extrae el último octeto de la dirección IP asignada a eth0. El segundo invoca Tomahawk, dando a la máquina de su propio bloque de 16 millones de direcciones IP.

Replay paquetes en paralelo

El ejemplo anterior juega 5 copias de forma secuencial outlook.pcap. Tomahawk espera la primera repetición para completar antes de enviar el segundo. Puede usar la bandera "-n" para enviar los paquetes de repetición en paralelo. Por ejemplo:

tomahawk -n 3 -l 5 -f outlook.pcap

Este comando repeticiones outlook.pcap 5 veces, con hasta 3 versiones que se ejecutan simultáneamente. Esta función es útil para la toma de una muestra de tráfico de red capturado a velocidades relativamente bajas y "ampliación" de la red que representa. Por ejemplo, suponga que tiene un rastro de tráfico de una red de 100 Mbps con 500 hosts. Mediante el uso de la "-n 10" se puede simular una red con 5.000 hosts en una red troncal Gigabit.

También puede utilizar Tomahawk para hacer múltiples ataques simultáneamente. Por ejemplo:

tomahawk -n 3 -l 5 -f outlook.pcap -f slammer.pcap -f codered.pcap

Este comando ejecuta hasta 3 copias de Outlook, 3 copias de Slammer, y 3 copias de CodeRed simultáneamente. En términos de la herramienta, ejecuta 9 repeticiones simultáneas en total, 6 de los cuales (Slammer y CodeRed) son ataques. El número de pcaps que se pueden cargar sólo está limitado por la memoria.

Banderas Globales y Handler

Tomahawk tiene dos tipos de indicadores: indicadores globales y banderas de controlador. Indicadores globales afectan a todos los pcaps, banderas del controlador afectan pcaps subsiguientes hasta anulado. Por ejemplo, considere lo siguiente:

tomahawk -n 3 -l 5 -f outlook.pcap -n 2 -l 4 -f slammer.pcap -f codered.pcap

Hasta 7 pcaps y 4 ataques se ejecutan simultáneamente, y un total de 8 ataques se ejecutan.

Retransmitir paquetes perdidos

Como se mencionó anteriormente, Tomahawk retransmite paquetes perdidos. Los parámetros para la retransmisión se controlan con las banderas de controlador -r -t y. Por ejemplo:

tomahawk 1 -l -r -t 5 1000 -f outlook.pcap

Este comando le dice a Tomahawk esperar (al menos) 1000 milisegundos antes de declarar un paquete perdido ("-t 1000") y para retransmitir el paquete de 5 veces ("-r 5") antes de abandonar y se imprime un mensaje de tiempo de espera.

Conservar las direcciones IP sin modificaciones

De vez en cuando, la dirección IP en el paquete es una parte importante del ataque. Por ejemplo, algunos mensajes Stacheldraht establecer la dirección IP de origen a "3.3.3.3

Se utilizan los controles -A si las direcciones se modifican:

tomahawk -l 1 -A 0 -f stacheldraht.pcap

Se usa "-A 0" para evitar Tomahawk desde el cambio de direcciones IP para pcaps posteriores y "-A 1" para cambiar las direcciones IP para pcaps posteriores. Por ejemplo:

tomahawk -l 1 -A 0 -f stacheldraht.pcap -A 1 -f outlook.pcap

En este ejemplo, Tomahawk deja las direcciones IP en stacheldraht.pcap sin cambios; Considerando que modifica las direcciones IP en outlook.pcap.

Generar Tráfico Limpio

El siguiente comando genera una gran cantidad de tráfico limpio:

tomahawk -n 50 -l 10000 -f http.pcap

En la práctica, Tomahawk puede generar 70 a 500 Mbps en una máquina, dependiendo de la plataforma y pcaps utilizados. Para mayor rendimiento, TippingPoint recomienda dos Intel PRO / 1000 LAN NIC en un procesador de la familia Pentium 2.0+ GHz con una configuración de servidor. La plataforma Dell 1750 también se desempeña bien para estas necesidades.

Limitar las corrientes simultáneas

Puede cargar un gran número de archivos y limitar el número total de flujos simultáneos como una prueba. Si se supone un tiempo de espera de paquetes a 1 segundo y se permiten 5 retransmisiones, cada ataque dura 5 segundos para el tiempo de espera. Para alcanzar la tasa de ataque deseada, hay que ejecutar 50 ataques simultáneamente. Los controles –N:

tomahawk -N 50 -l 1 -f attack1.pcap -f attack2.pcap ... -f attack1000.pcap

La bandera -N limita el número total de casos pcaps que se pueden reproducir simultáneamente; mientras que, -n limita el número de instancias de cada pcap que se puedan ejecutar de forma simultánea. Por ejemplo:

tomahawk -N 50 -n 5 -l 10 -f attack1.pcap -f attack2.pcap ... -f attack1000.pcap

Este comando establece que se ejecuten 5 copias de cada uno de attack1 hasta attack1000 simultáneamente (5.000 en total). Los -N 50 nos indican que se reproducen 50 a la vez. [7]

4.6 Bit-Twist

Bit-Twist es un sencillo pero potente generador de paquetes Ethernet basado en libpcap. Está diseñado para complementar tcpdump , que por sí mismo ha hecho un gran trabajo en la captura de tráfico de red.

Con Bit-Twist, ahora se puede regenerar su tráfico capturado en una red en directo. Los paquetes se generan a partir de archivo de rastreo tcpdump (archivo.pcap). Bit-Twist también viene con un completo editor de archivo de rastreo para permitirle cambiar el contenido de un archivo de rastreo.

En general, el generador de paquetes es útil para simular el tráfico de red o escenario, las pruebas de firewall, IDS, IPS y, y solución de problemas diversos problemas en la red.

Características

Estas son sólo algunas de las características importantes que lo hacen único y se destaca como uno de los mejores generadores de paquetes Ethernet a disposición de la comunidad de código abierto. [8]

- Se ejecuta en Mac OS X (y * BSD), Linux y Windows.
- Envía múltiples archivos de rastreo a la vez.
- Envía paquetes a una velocidad específica o velocidad de línea en Mbps.
- Editor de archivo de seguimiento integral con control sobre la mayoría de los campos en Ethernet, ARP, IP, ICMP, TCP y UDP cabeceras con corrección de cabecera de comprobación automática.
- Anexa la carga útil de usuario a los paquetes existentes después de una cabecera específica.

- Selecciona un rango específico de paquetes y lo guarda en otro archivo de rastreo.
- Altamente scriptable – se puede convertir en una herramienta de generador de paquetes extremadamente flexible.

4.7 Tcpreplay

Tcpreplay es un conjunto de utilidades GPLv3 con licencia para sistemas operativos UNIX (y Win32 bajo Cygwin) para editar y reproducir el tráfico de red que fue capturado previamente por herramientas como tcpdump y Wireshark . Te permite clasificar el tráfico como cliente o servidor, reescritura de Capa 2, 3 y 4 paquetes y finalmente reproducir el tráfico de nuevo en la red ya través de otros dispositivos tales como switches, routers, cortafuegos, NIDS y de IPS. Tcpreplay soporta ambos modos NIC individuales y dobles para probar ambos oler y dispositivos en línea.

Tcpreplay es utilizado por numerosos cortafuegos, IDS, IPS, NetFlow y de otros proveedores, empresas, universidades, laboratorios y proyectos de código abierto de redes. Si su organización utiliza tcpreplay, por favor háganoslo saber quién eres y lo que se utiliza para para que podamos seguir añadiendo características que son útiles.

Tcpreplay está diseñado para trabajar con el hardware de red y normalmente no penetra más allá de la capa 2. Yazan Siam con el patrocinio de Cisco desarrolló *tcpliveplay* para reproducir archivos pcap TCP directamente a los servidores. Use esta utilidad si desea probar toda la pila de red y en la aplicación. [9]

Desde la versión 4.0, tcpreplay se ha mejorado para abordar las complejidades de las pruebas y puesta a punto IP Flow / NetFlow hardware. Las mejoras incluyen:

- Soporte para NetMap modificado controladores de red para un rendimiento 10 GigE a velocidad de cable
- Mayor precisión de velocidad de reproducción
- Mayor precisión de los resultados de la presentación de informes
- Estadísticas de flujo incluyendo flujos por segundo (fps)
- Análisis de flujo para el análisis y puesta a punto de los tiempos de espera de caducidad de flujo
- Cientos de miles de flujos por segundo (depende de los tamaños de flujo en el archivo pcap)

La suite tcpreplay incluye las siguientes herramientas:

- tcpprep - archivo pcap de múltiples pasadas pre-procesador que determina paquetes como cliente o servidor y crea archivos de caché utilizados por tcpreplay y tcprewrite
- tcprewrite - editor de archivos pcap que reescribe encabezados 2 paquetes TCP / IP y Capa
- tcpreplay - repeticiones archivos pcap a velocidades arbitrarias en la red
- tcpliveplay - Las repeticiones tráfico de red almacena en un archivo pcap en redes en vivo utilizando nuevas conexiones TCP
- tcpreplay-edit - repeticiones y ediciones PCAP archivos a velocidades arbitrarias en la red
- tcpbridge - puente de dos segmentos de red con el poder de tcprewrite
- tcpcapinfo - decodificador de archivos pcap prima y depurador

La versión 4.0 es la primera versión entregada por Fred Klassen y patrocinado por AppNeta.

Ejemplos:

```
root @ pw29: ~ #  tcpreplay -i -t eth7 -K --loop 5000 smallFlows.pcap
Caché de archivos está habilitado
Actual: 71305000 paquetes (46082655000 bytes) enviados en 38,03
segundos.
Nominal: 1201832266.1 Bps, 9614.65 Mbps, 1.859.629,17 pps
Flujos: 1209, los flujos de 31.53 fps, 71.215 millones de paquetes de
flujo, 90.000 no-flujo
Estadísticas para dispositivo de red: eth7
    Paquetes intentados: 71305000
    Paquetes exitosos: 71305000
    Paquetes perdidos: 0
    Paquetes truncados: 0
    Paquetes reintentados (ENOBUFS): 0
    Paquetes reintentados (EAGAIN): 0
```

```
root @ pw29: ~ #  tcpreplay -i eth7 --mbps = 9,500 -K --loop 5000
smallFlows.pcap
Caché de archivos está habilitado
Actual: 71305000 paquetes (46082655000 bytes) enviados en 38,08
segundos.
Nominal: 1187499244.6 Bps, 9499.99 Mbps, 1.837.451,28 pps
Flujos: 1209, los flujos de 31.15 fps, 71.215 millones de paquetes de
flujo, 90.000 no-flujo
Estadísticas para dispositivo de red: eth7
    Paquetes intentados: 71305000
    Paquetes exitosos: 71305000
    Paquetes perdidos: 0
    Paquetes truncados: 0
```

```
    Paquetes reintentados (ENOBUFS): 0
    Paquetes reintentados (EAGAIN): 0
```

```
root @ pw29: ~ #  tcpreplay -i eth7 tK --loop 50.000 --netmap --
unique-ip smallFlows.pcap
El controlador de red de la conmutación para eth7 a NetMap modo bypass
... hecho!
Caché de archivos está habilitado
Actual: 713050000 paquetes (460826550000 bytes) enviados en 385,07
segundos.
Nominal: 1194660947.8 Bps, 9557.28 Mbps, 1.848.532,79 pps
Flujos: 60450000 flujos, 156712.44 fps, 712 150 000 900 000 paquetes
de flujo, no flujo
Estadísticas para dispositivo de red: eth7
    Paquetes intentados: 713050000
    Paquetes exitosos: 713050000
    Paquetes perdidos: 0
    Paquetes truncados: 0
    Paquetes reintentados (ENOBUFS): 0
    Paquetes reintentados (EAGAIN): 0
Cambio de controlador de red para eth7 al modo normal... hecho
```

4.8 Comparativa de los generadores

Aspecto	D-ITG	PACKETH	OSTINATO	TOMAHAWK	NETSNIFF-NG	TCPREPLAY	BITTWIST
Versión evaluada	jul-13	abr-15	jul-14	abr-06	may-15	dic-14	abr-12
Última actualización	2.8.1	1.8.1	0.6	1.1	0.5.9	4.1.0	2.0
Gratis	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Open source	Sí	Sí	Sí	Sí	Sí	Sí	Sí
A/P	Activa	Activa	Activa	Activa	Activa	Activa	Activa
Privilegios	Usuario	Usuario	Usuario	Usuario	Usuario	Usuario	Usuario
Plataforma soportada	Linux/Windows	Linux, Mac OS X, Windows	Windows, Mac OS X, Linux, FreeBSD	RedHat(Linux)	Linux	Linux, Windows	Linux, Mac OS X, Windows
Protocolo de red	IPv4,IPv6,ICMPv4,ICMPv6	ARP, Ipv4, IPv6	ARP, IPv4, IPv6, IP-in-IP	IP	IPv4,IPv6	IPv4,IPv6	IPv4,IPv6
Protocolo de transporte	TCP, UDP	TCP, UDP, ICMPv4,ICMPv6	TCP, UDP, ICMPv4, ICMPv6, IGMP, MLD	TCP	TCP, UDP	TCP, UDP	TCP, UDP
Resultados reportados	Retardo, jitter y throughput	Creación de paquetes ethernet	Creación de streams	Rendimiento de los IPS	Análisis offline de pcaps	Análisis offline de pcaps	Análisis offline de pcaps
Interfaz de usuario	Consola	GUI, CLI	GUI	Consola	Consola	Consola	Consola
Sincronización requerida	No	No	No	Sí	No	No	No

Figura 8. Tabla comparativa de los generadores de tráfico

5 Herramientas de tráfico ilícito

En este capítulo, se describen las herramientas con las que hemos hecho las pruebas de cómo pueden afectar con sus ataques a un sistema.

5.1 Metasploit

Metasploit es un proyecto open source de seguridad informática que proporciona información acerca de vulnerabilidades de seguridad y ayuda en tests de penetración "Pentesting" y el desarrollo de firmas para sistemas de detección de intrusos.

Figura 9. Logo de Metasploit

Su subproyecto más conocido es el Metasploit Framework, una herramienta para desarrollar y ejecutar exploits contra una máquina remota. Otros subproyectos importantes son las bases de datos de opcodes (códigos de operación), un archivo de shellcodes, e investigación sobre seguridad.

Inicialmente fue creado utilizando el lenguaje de programación de scripting Perl, aunque actualmente el Metasploit Framework ha sido escrito de nuevo completamente en el lenguaje Ruby.

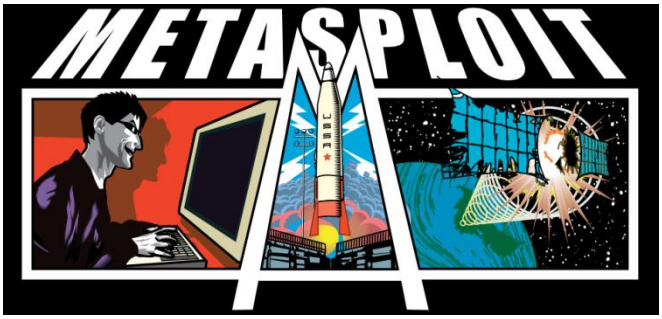


Figura 10. Metasploit

Metasploit fue creado por H.D. Moore en el 2003, como una herramienta de red portátil usando el lenguaje Perl. El 21 de octubre de 2009, el Proyecto Metasploit anuncio anunció [1] que había sido adquirida por Rapid7, una empresa de seguridad que ofrece soluciones unificadas de gestión de vulnerabilidades.

Al igual que los productos de la competencia, como Core Security Technologies y Core Impacto, Metasploit se puede utilizar para probar la vulnerabilidad de los sistemas informáticos o entrar en sistemas remotos. Al igual que muchas herramientas de seguridad informática, Metasploit se puede utilizar tanto para actividades legítimas y autorizadas. Desde la adquisición de Metasploit Framework, Rapid7 ha añadido dos Open source "Codigo abierto" llamados Metasploit Express y Metasploit Pro. [10]

M**arco/Sistema Metasploit**

Los pasos básicos para la explotación de un sistema que utiliza el Sistema incluyen:

1. La selección y configuración de un código el cual se va a explotar. El cual entra al sistema objetivo, mediante el aprovechamiento de una de bugs; Existen cerca de 900 exploits incluidos para Windows, Unix / Linux y Mac OS X;
2. Opción para comprobar si el sistema destino es susceptible a los bugs elegidos.
3. La técnica para codificar el sistema de prevención de intrusiones (IPS) e ignore la carga útil codificada;
4. Visualización a la hora de ejecutar el exploit.

Metasploit se ejecuta en Unix (incluyendo Linux y Mac OS X) y en Windows. El Sistema Metasploit se puede extender y es capaz utilizar complementos en varios idiomas.

Para elegir un exploit y la carga útil, se necesita un poco de información sobre el sistema objetivo, como la versión del sistema operativo y los servicios de red instalados. Esta información puede ser obtenida con el escaneo de puertos y "OS fingerprinting", y con herramientas como Nmap, NeXpose o Nessus. Estos programas, pueden detectar vulnerabilidades del sistema de destino.

Metasploit puede importar los datos de la exploración de vulnerabilidades y comparar las vulnerabilidades identificadas.

Metasploit 3.0 comenzó a incluir herramientas de fuzzing, utilizadas para descubrir las vulnerabilidades del software, en lugar de sólo explotar bugs conocidos. Metasploit 4.0 fue lanzado en agosto de 2011.

Interfaces de Metasploit

Hay varias interfaces para Metasploit disponibles. Las más populares son mantenidas por Rapid7 y Estratégico Ciber LLC.

Edición Metasploit

La versión gratuita contiene una interfaz de línea de comandos, la importación de terceros, la explotación manual y fuerza bruta.

Edición Community Metasploit

En octubre de 2011, Rapid7 liberado Metasploit Community Edition, una interfaz de usuario basada en la web gratuita para Metasploit. Metasploit community incluye, detección de redes, navegación por módulo y la explotación manual.

Metasploit express

En abril de 2010, Rapid7 libero Metasploit Express, una edición comercial de código abierto, para los equipos de seguridad que necesitan verificar vulnerabilidades. Ofrece una interfaz gráfica de usuario, integra nmap para el descubrimiento, y añade fuerza bruta inteligente, así como la recopilación de pruebas automatizado.

Metasploit Pro

En octubre de 2010, Rapid7 añadió Metasploit Pro, de código abierto para pruebas de penetración. Metasploit Pro incluye todas las características de Metasploit Express y añade la exploración y explotación de aplicaciones web.

Armitage

Armitage es una herramienta de gestión gráfica para ciberataques del Proyecto Metasploit, visualiza objetivos y recomienda métodos de ataque. Es una herramienta para ingenieros en seguridad web y es de código abierto. Destaca por sus contribuciones a la colaboración del equipo rojo, permitiendo sesiones compartidas, datos y comunicación a través de una única instancia Metasploit.

Cargas útiles

Metasploit ofrece muchos tipos de cargas útiles, incluyendo:

- 'Shell de comandos' permite a los usuarios ejecutar scripts de cobro o ejecutar comandos arbitrarios.
- 'Meterpreter' permite a los usuarios controlar la pantalla de un dispositivo mediante VNC y navegar, cargar y descargar archivos.
- 'Cargas dinámicas' permite a los usuarios evadir las defensas antivirus mediante la generación de cargas únicas.

Escenario de uso con metasploitable

Tenemos Metasploit en Linux:

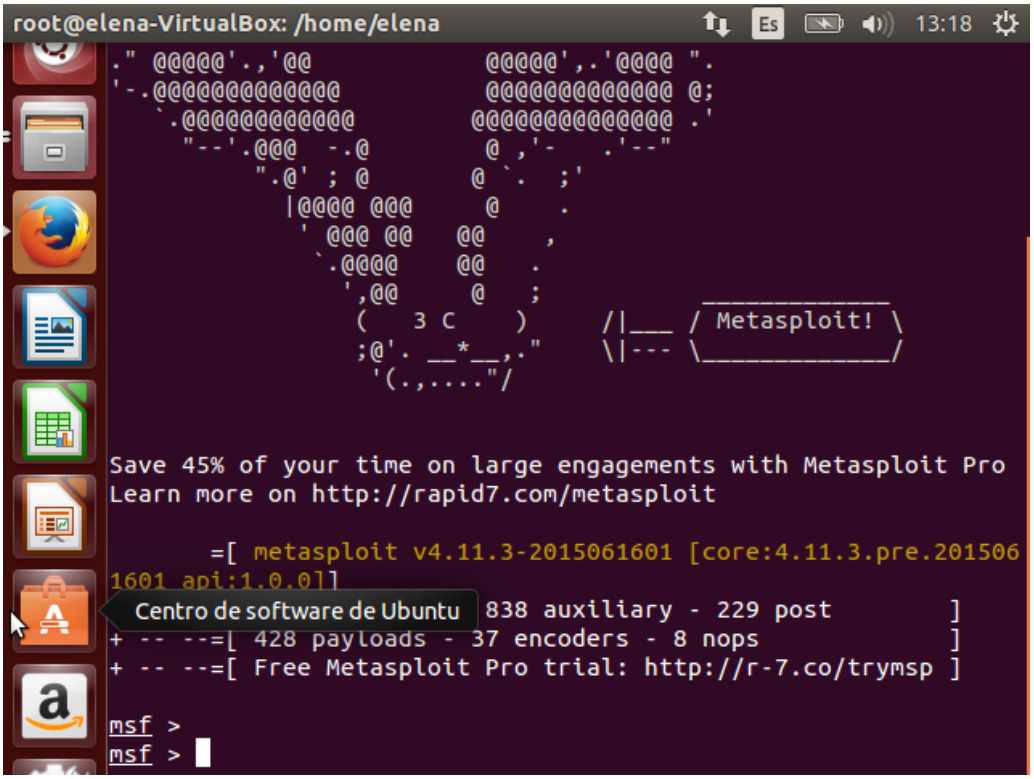


Figura 11. Captura de Metasploit en Linux

Captura con Wireshark:

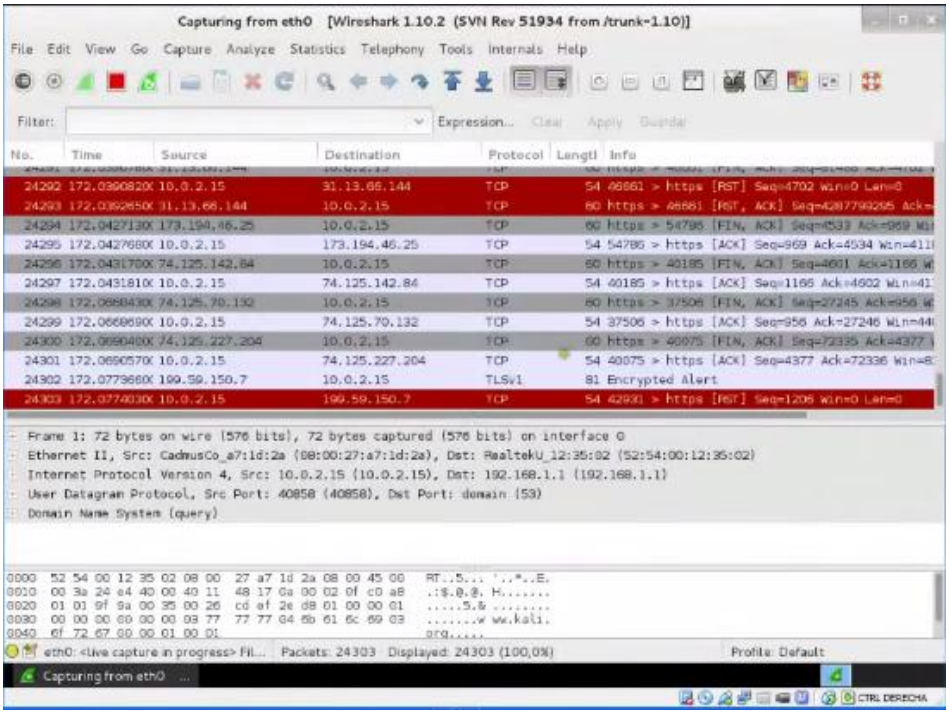


Figura 12. Captura de protocolo TCP con Metasploit

Con nmap, se escanean los puertos, y se observa que el 3790 está abierto:

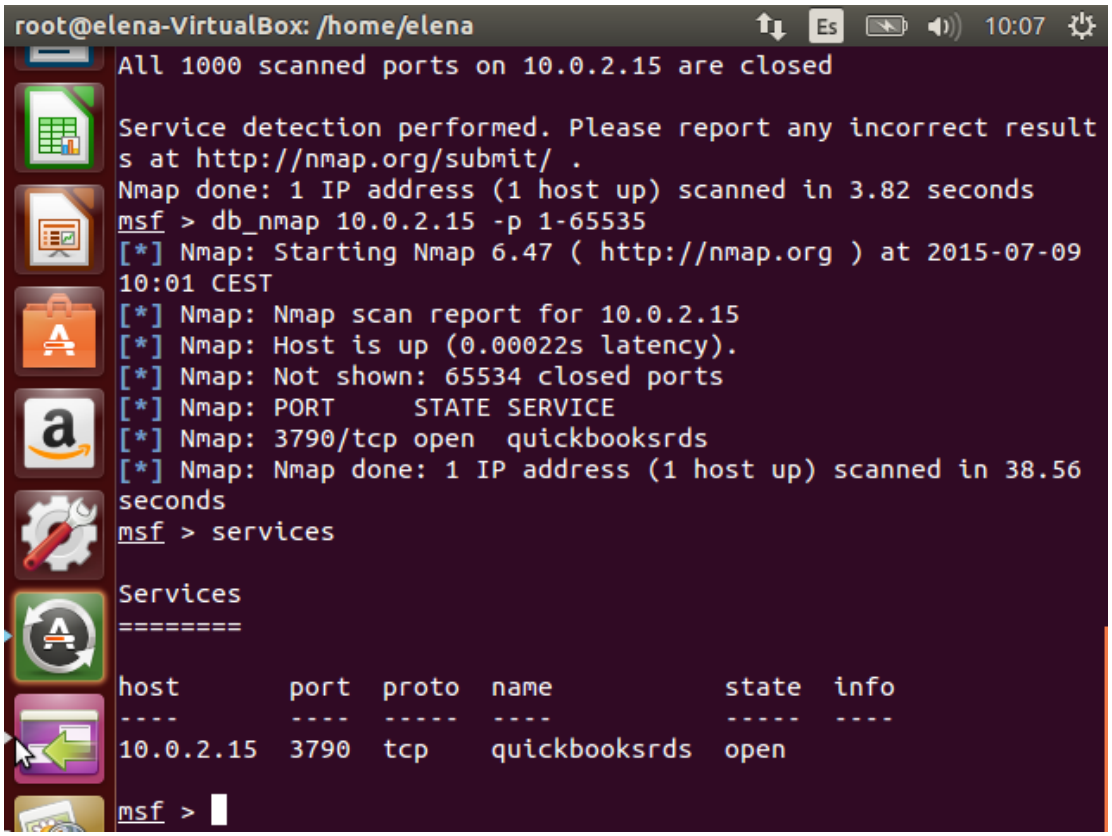


Figura 13. Captura de escaneo de puertos en Metasploit

Se buscan módulos con el comando search para el servicio ftp:

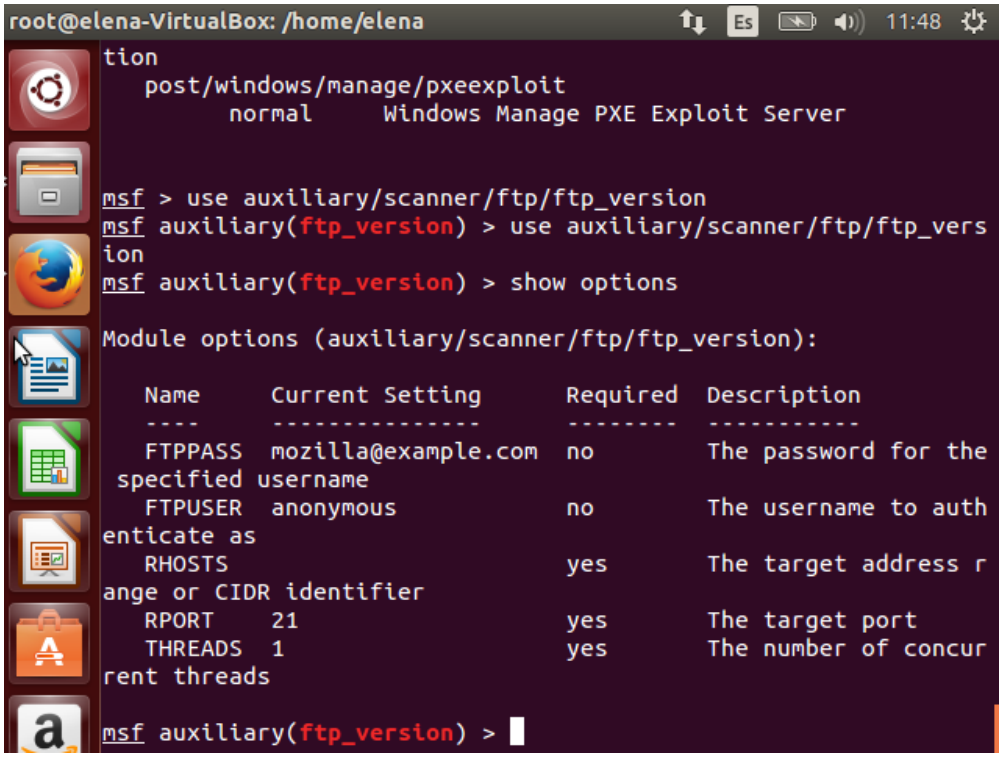


Figura 14. Captura de búsqueda de módulos en Metasploit

Se obtiene la versión del servicio y se buscan exploits que se aprovechen de vulnerabilidades.

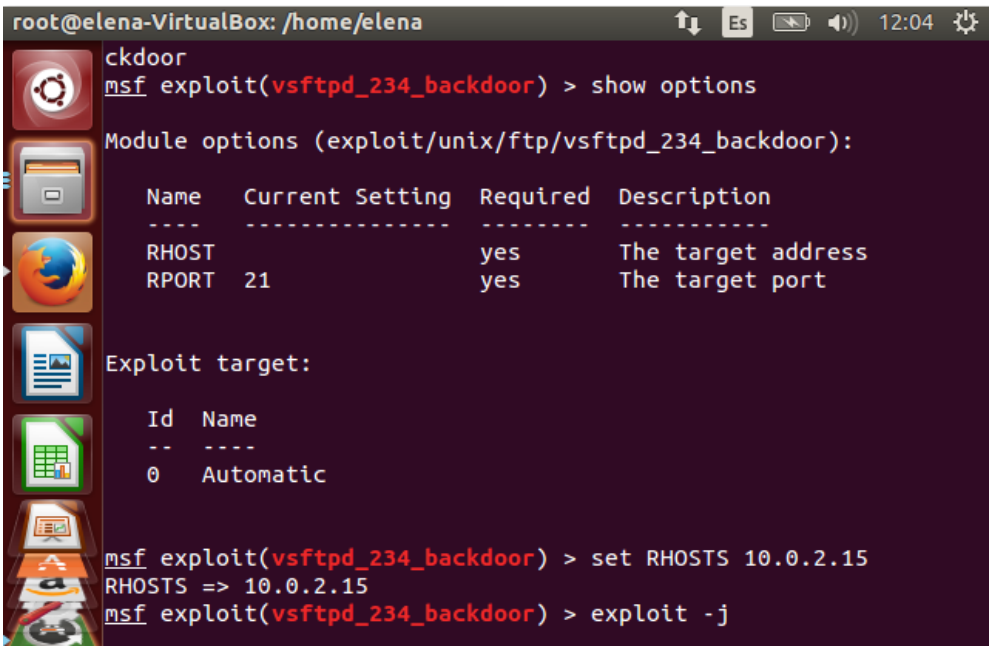


Figura 15. Captura de búsqueda de exploits en Metasploit

Se aprecia uno, y se procede a la ejecución.

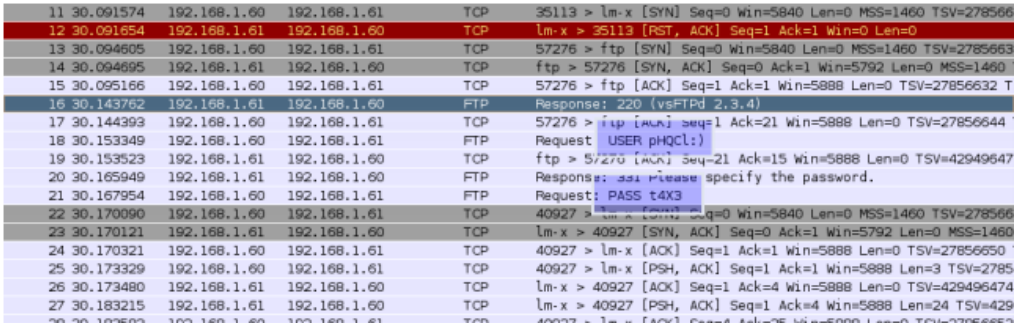


Figura 16. Captura de Wireshark con la ejecución del exploit.

5.2 LOIC

Low Orbit Ion Cannon (abreviado LOIC) es una aplicación diseñada para realizar un ataque de denegación de servicio durante el proyecto Chanology, desarrollada por «praetox» usando el lenguaje de programación C# (Existe también un fork en C++ y Qt llamado LOIQ). La aplicación realiza un ataque de denegación de servicio del objetivo enviando una gran cantidad de paquetes TCP, paquetes UDP o peticiones HTTP con objeto de determinar cuál es la cantidad de peticiones por segundo que puede resolver la red objetivo antes de dejar de funcionar. [11]

IRCLOIC

El desarrollador «NewEraCracker» actualizó LOIC para su uso en la *Operation Payback*, arreglando algunos bugs y añadiendo algunas habilidades nuevas al programa.

Desde la versión 1.1.1.3, LOIC incorpora la posibilidad de que el usuario delegue voluntariamente el control de la aplicación LOIC al operador de un canal IRC, que puede controlar de esta manera un ataque coordinado empleando todos los clientes conectados a dicho canal. Esta característica se denomina comúnmente como inteligencia de enjambre (*Hive Mind*, en inglés) y permite la organización rápida de una botnet formada por voluntarios. La versión con control remoto por IRC se denomina habitualmente IRCLOIC.

Esta actualización funciona en Windows XP y posteriores, necesitando Microsoft .NET Framework 3.5 Service Pack 1. También funciona en sistemas Linux con los paquetes Mono o Wine.

En las últimas versiones se ha añadido una opción para iniciar el programa oculto como un servicio de Windows. En todas las versiones de *LOIC* es bastante fácil conocer la dirección IP del atacante si éste no está protegido por una red Peer-to-peer anónimo.

LOIQ

LOIQ es una implementación de la funcionalidades de LOIC empleando el lenguaje C++ y la biblioteca Qt 4, lo cual permite compilarlo en cualquier plataforma que disponga de la biblioteca Qt (Linux, BSD, Unix, Windows, Windows CE, Mac OS X, etc...)

JSLOIC

lizadas en lenguaje JavaScript, lo cual permite ejecutarlo desde un navegador web sin necesidad de instalar ningún software en el ordenador. Esta implementación realiza test limitados a peticiones HTTP, no permitiendo el envío de paquetes TCP y UDP, ni delegar el control para su uso en modo colectivo.

USO DE LOIC EN LA OPERATION PAYBACK

LOIC comenzó a emplearse en la *Operation Payback* para bloquear los servidores de varias organizaciones protectoras de copyright como respuesta al ataque de denegación que ellos habían lanzado contra varios servidores de archivos *torrent*. Desde la red 4chan se hizo un llamamiento para que la gente instalase IRCLOIC en sus ordenadores y cediese su control a través del servidor "irc.skidsr.us" (ahora servidor de LulzSec), de forma que el grupo Anonymous pudiera realizar ataques coordinados de forma similar a los utilizados en el Proyecto Chanology. Dado el éxito mediático de dichos bloqueos, y la cantidad de voluntarios anónimos que cedieron sus ordenadores para participar en el bloqueo, se siguió realizando un programa de bloqueos organizados contra varias instituciones afines a la gestión de derechos de autor.

El 6 de diciembre de 2010, se hizo un nuevo llamamiento para usar el programa LOIC de forma masiva contra las empresas PostFinance y PayPal por el bloqueo de las cuentas bancarias de WikiLeaks, y posteriormente contra Mastercard por impedir que sus tarjetas fueran empleadas para la donación de fondos destinados a la fundación Wikileaks o a la defensa de Julian Assange.

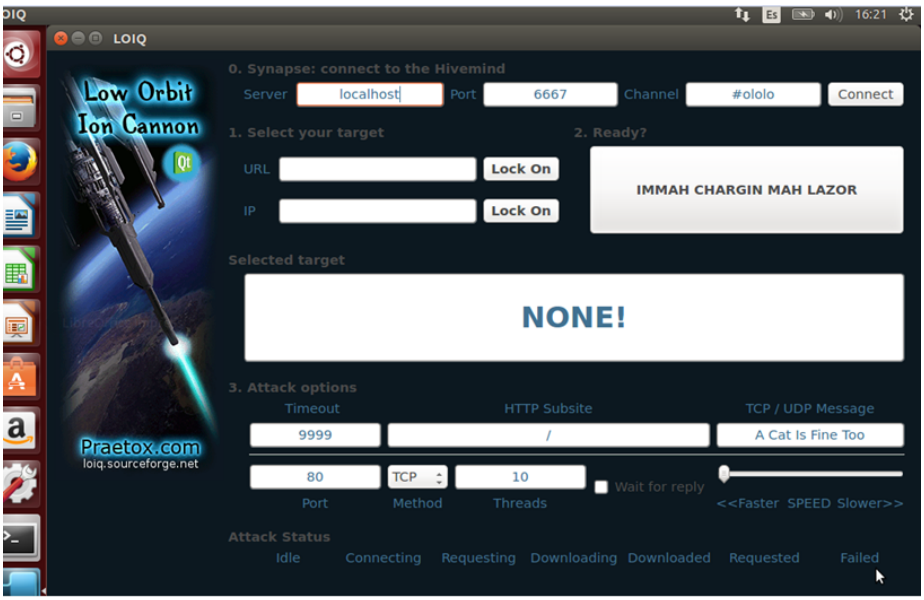


Figura 17. GUI de Loic en Linux

Peligros asociados

En el caso de los ataques de DDoS a equipos ajenos a nosotros, está de más decir que son ilegales. Además de ser fácilmente rastreable por los registros de conexión de los servidores y los proveedores de internet.

Pero existe un peligro que muchas veces no vemos, y es el de instalar una herramienta que está desarrollada para hacer daño y permite que otra persona controle nuestra PC. Básicamente, si una persona puede controlar a quién yo ataco, no solamente estoy atacando, si no que estoy dándole el control a otra persona de mis ataques. Como si esto fuera poco, este tipo de herramientas son complejas y sólo las personas con un buen entendimiento de los lenguajes de programación van a poder comprender exactamente qué es lo que hacen. Así que no se puede tener la certeza de que controlen otros parámetros de nuestras PC.

Por otro lado, varias de las versiones que se pueden descargar de Internet ya vienen compiladas, así que no podemos analizar el código fuente. Y, si bien se entrega un código fuente adjunto a las versiones compiladas, no podremos comprobar que sea el código correspondiente. Por lo tanto, la única forma que tenemos de estar seguros de qué hace esta herramienta es analizar línea por línea todo el código fuente y compilarlo nosotros mismos.

Se utiliza la herramienta LOIC para realizar un ataque a una IP 10.0.2.4 dentro de nuestro sistema de máquinas virtuales de Linux.

Se utiliza el puerto 80 y el método TCP y ejecutamos con 20 threads, que son el número de peticiones que se llevan a cabo.

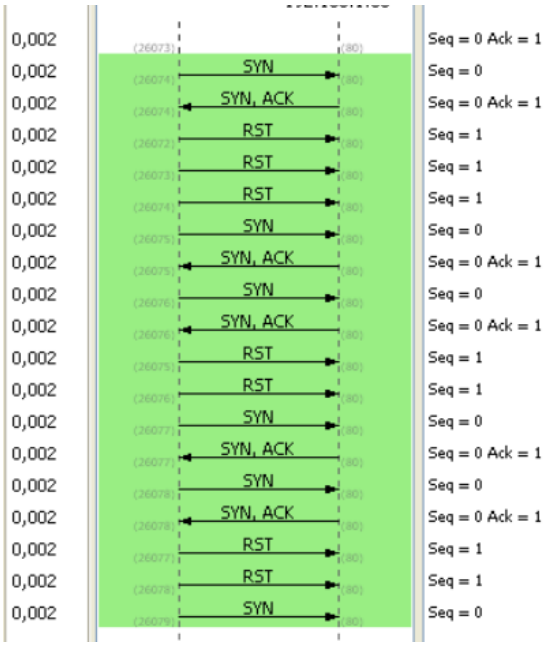


Figura 18. Captura del flujo de TCP en Wireshark con Loic

Sin la ejecución, nuestro procesador de mantiene en niveles máximos:

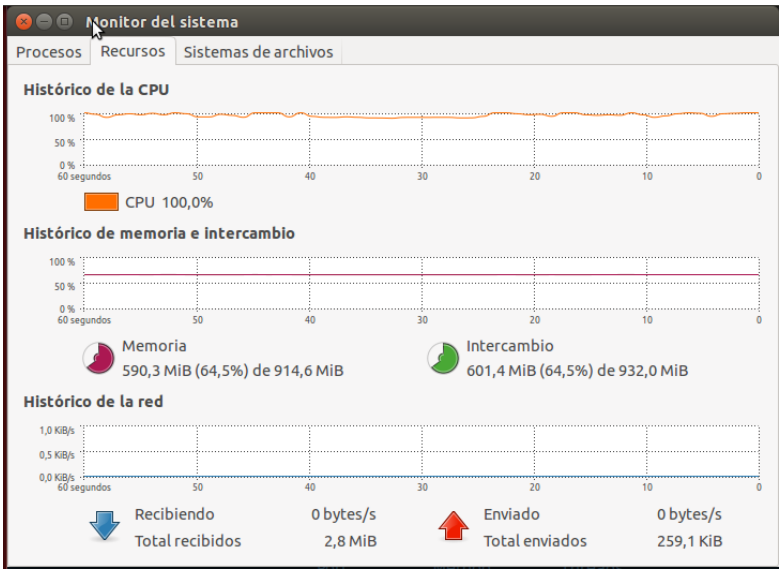
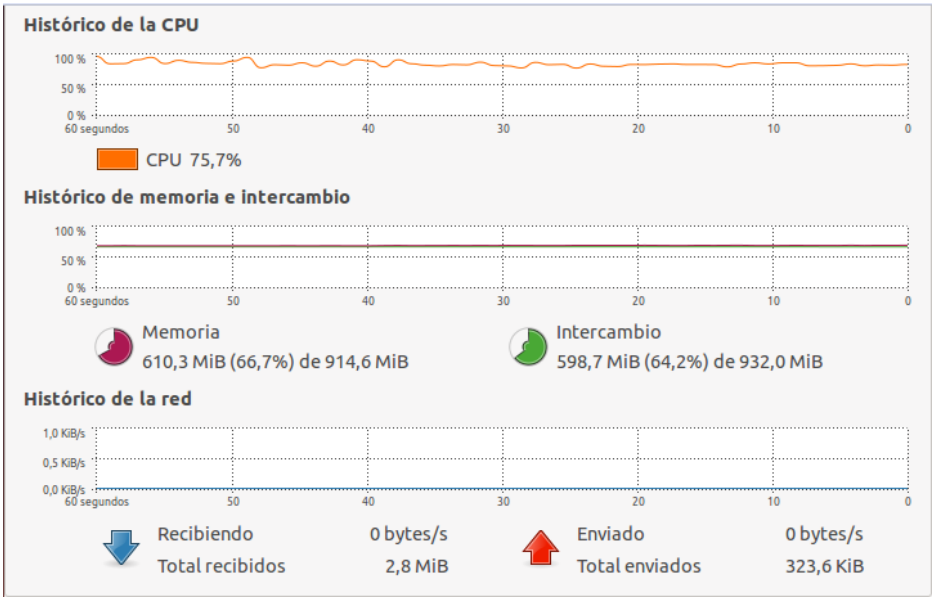


Figura 19. Captura del monitor en Linux antes del ataque

Cuando se comienza el ataque, la CPU empieza abajar:



Captura 20. Captura del monitor en Linux después del ataque

Se sigue el comportamiento de conexiones TCP, que en la captura 19 describe de forma muy intuitiva mediante flechas, el origen y destino de cada paquete, resaltando los flag activos que intervienen en cada sentido de la conexión.

En nuestro caso se observa que, en un intervalo muy corto de tiempo, existen numerosos intentos de conexión por parte de la IP 10.0.0.200 al puerto 80 de la máquina 10.0.0.101, situación algo inusual. El servidor ha tratado de resolver la MAC de la máquina cliente en numerosas ocasiones, una de ellas la podemos ver en el paquete 7852, pero, al no recibir respuesta alguna y, por tanto, al carecer de la dirección física del host, no puede enviar un ACK-SYN al mismo para continuar con el establecimiento de la conexión a tres pasos.

Esto conlleva que la pila TCP/IP de nuestro servidor tenga que esperar por cada conexión un tiempo determinado, durante el cual seguirán llegando más paquetes que irán creando nuevas conexiones. Por cada conexión que se intente establecer se creará una estructura en memoria denominada TCB (Transmission Control Block) que es usada por la pila TCP/IP del sistema operativo para identificar cada una de las conexiones (sockets local y remoto, segmento actual, punteros a buffers de envío y recepción) y que, con un número muy elevado, pueden acabar con los recursos de la máquina produciendo que el equipo deje de contestar más solicitudes de conexión.

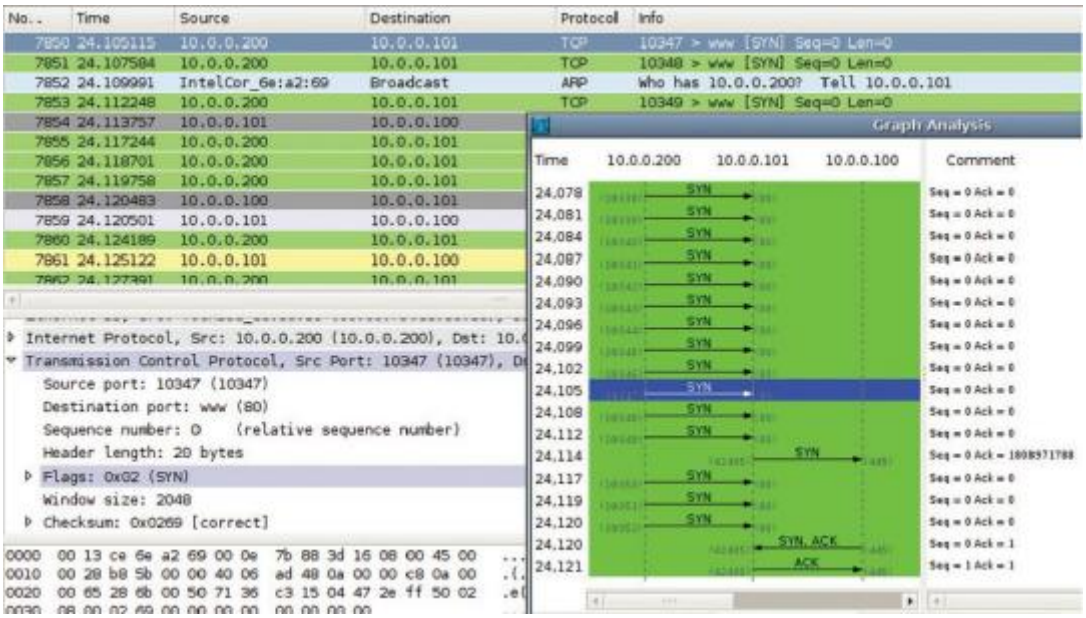


Figura 21. Captura de flujo TCP

6 Evaluación de rendimiento de las herramientas

Para evaluar el rendimiento de los replicadores de tráfico se han realizado pruebas de envío con varios ficheros de trazas distintos según el tamaño de los paquetes y su procedencia. Los paquetes enviados son capturados por un analizador de tráfico que genera un fichero de trazas de salida con los paquetes capturados. Finalmente, se comparan el fichero de trazas enviado y el generado por la capturadora para comprobar que el replicador ha enviado correctamente la traza.

En los siguientes apartados se van a presentar las pruebas realizadas y los resultados obtenidos para cada una de ellas.

Ostinato: Su principal ventaja con respecto al resto, es su flexibilidad. Permite simular gran cantidad de protocolos con diferentes tiempos entre paquetes. Además, permite añadir mediante scripts cualquier protocolo que no esté implementado en la herramienta.

Además, es sencillo crear capturas a partir de parámetros, guardarlas en PCAP y luego replicarlas con él mismo u otro programa.

En las medidas de rendimiento es el generador software que mejores tasas binarias alcanza, llegando a los 9 Gbps para el envío de paquetes de 8950 bytes.

El objetivo de estas pruebas es medir la máxima tasa binaria alcanzable para los distintos tamaños de paquete utilizados, para lo que se ha enviado con tiempo entre paquetes constante de 0 ns.

Se comienza con la captura de tráfico de HTTP. Para ello se establece el envío de 3000 paquetes y un tamaño de 60 bytes por paquete. El puerto elegido es el 80-HTTP.

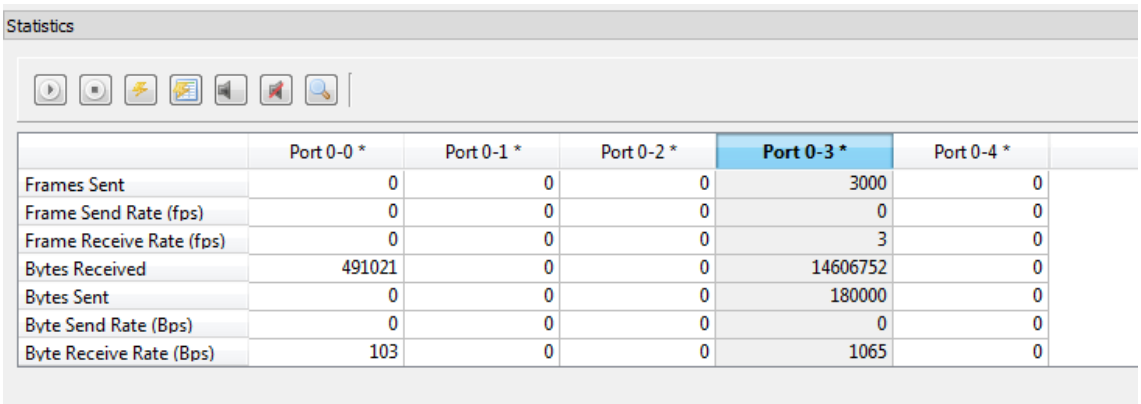


Figura 22. GUI de Ostinato

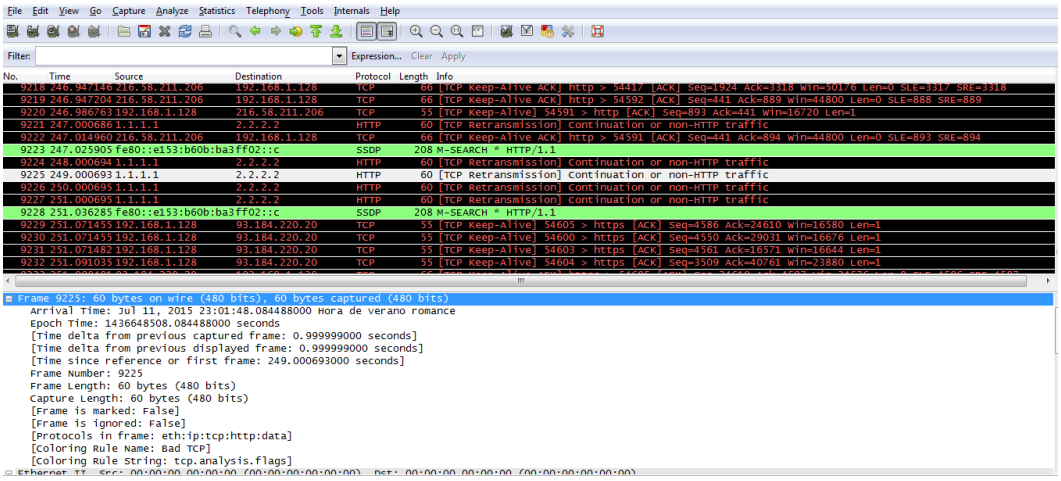


Figura 23. Captura de tráfico HTTP con Ostinato

Se continúa con trafico TCP para un paquete de 1500 bytes, con mayor ancho de banda, alcanzando una tasa binaria de 2422000 bps.

Con un paquete de 64 bytes, el rendimiento es claramente menor y debido al mayor envío de paquetes por segundo.

Valorando la escalabilidad del sistema se obtienen los resultados siguientes:

La CPU alcanza niveles mayores al igual que el uso de la memoria física aumenta considerablemente.

Al ejecutar varios flujos a la vez, el rendimiento es lineal, en este caso ha sido probado en un procesador core i5 de cuatro núcleos.

Al lanzar más flujo que núcleos, varía levemente el rendimiento de la máquina

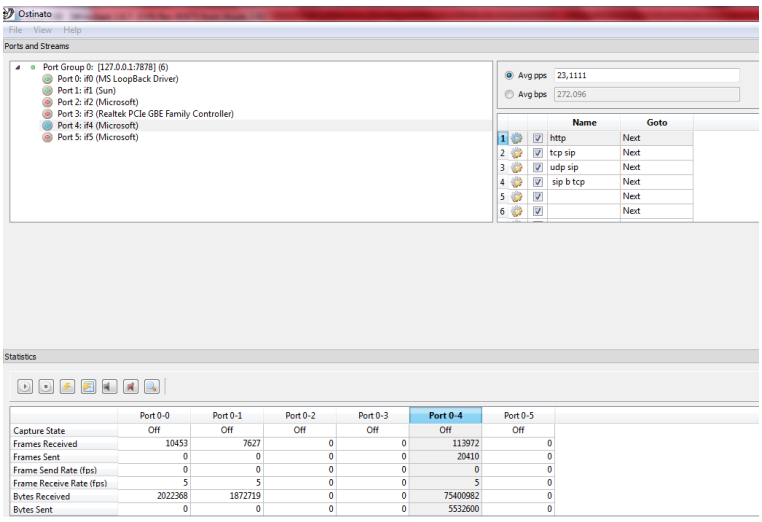


Figura 24. Interfaz de Ostinato con la elección de los puertos

No hay limitación en el tamaño de la captura a replicar.

Tcpreplay: Se replican los pcaps que se han capturado previamente con el sniffer Wireshark, obteniendo así los siguientes resultados:

Para un paquete de 64 bytes con flujo TCP:

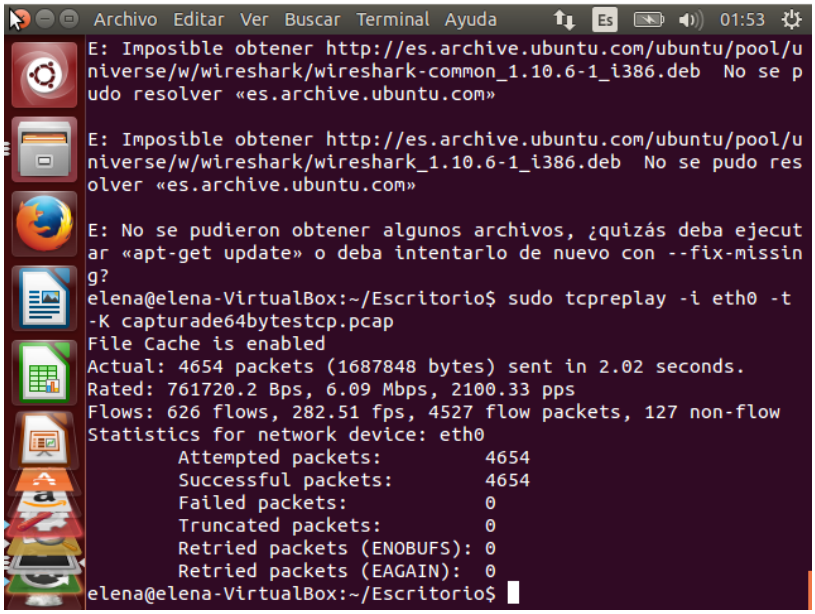


Figura 25. Captura de tráfico de Tcpreplay para un paquete de 64 bytes

Con un paquete de 1500 bytes, se obtienen aproximadamente 10Mbps.

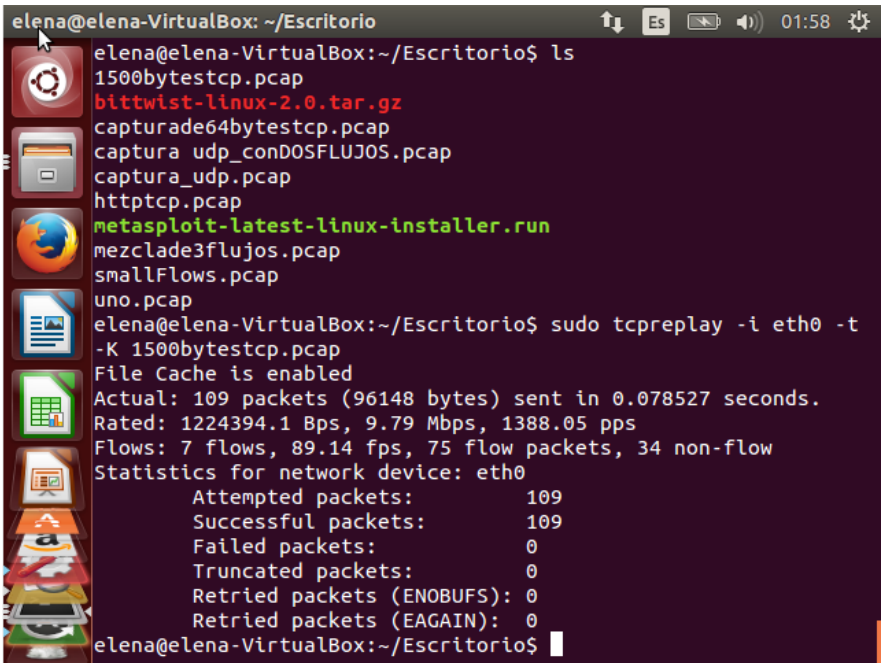


Figura 26. Captura de tráfico de Tcpreplay para un paquete de 1500 bytes

D-ITG: Se observa que en los paquetes de 1500 bytes, se alcanza un ancho de nada de 83.1 Mbps, mientras que con tamaños de paquetes más pequeños, se da una medición de 38.1 Mbps.

Cuando se procede a la replicación de paquetes de 128 bytes y 384 bytes, se observa un aumento del rendimiento de 39,1 Mbps.

A partir de tamaños de paquetes de 500 bytes, el rendimiento aumenta en menor proporción.

Tomahawk: Un único servidor Tomahawk puede generar 200-450 Mbps de tráfico. Mediante el uso de varios servidores y agregar el tráfico a través de un interruptor, 1 Gbps o más del tráfico se pueden reproducir a través de los NIPS.

Además, se puede limitar la velocidad de datos generada por Tomahawk, utilizando la bandera "-R". Por ejemplo, para generar 100 Mbps de tráfico limpio, utilice lo siguiente:

tomahawk -n 50 -l 10000 -f flujo1500bytes.pcap -R 100

El valor de "R" es un número de coma flotante. Para generar 100 Kbps de tráfico, utilizaría la siguiente:

tomahawk -n 50 -l 10000 -f flujo1500bytes.pcap -R 0.1

Trafgen (Netsniff-NG): Tiene unas mejoras del rendimiento significativas:

- Es capaz de escalar hasta alrededor de 12 Mpps, al utilizar 13 CPUs.

- Se aprecia flexibilidad a la hora de definir el contenido del paquete.

PackETH: Se obtienen resultados con la interfaz de línea de comandos:

./packETHcli -i lo -m 1 -f flujo64bytes.pcap

- Envía flujo64bytes.pcap una vez en lo

./packETHcli -i eth0 -m 2 -d 1000 -n 300 -f flujo1500bytes.pcap

- Envía flujo1500bytes.pcap 300 veces con 1000 nosotros (1 ms) entre ellos

./packETHcli -i eth0 -m 2 -d -n -1 0 -f packet2.pcap

- Envía flujo1500bytes.pcap a velocidad máxima, infinitas veces, no hay contadores

./packETHcli -i eth0 -m -d 2 0 0 -n -f packet2.pcap

- Envía flujo1500bytes.pcap a velocidad máxima, infinitas veces, con los contadores

./packETHcli -i eth1 -m -d 2 0 0 -n -s "1000 1500 100" -p 10 -f flujo1500bytes.pcap

- Envía 10 veces flujo1500bytes.pcap a velocidad máxima, comenzar con la longitud del paquete de 1000 bytes de aumentar la longitud del paquete de 100 bytes, envía otros 10 paquetes... hasta 1500 bytes.

./packETHcli -i eth0 -m 2 100 -d -n -s 0 "8500 8500" -f flujo1500bytes.pcap

- Envía flujo1500bytes.pcap infinitas veces con 300US entre ellas con el tamaño de 8500 bytes (incluso si flujo1500bytes es más largo).

7 Conclusiones

En el desarrollo del proyecto se han utilizado diferentes herramientas de trabajo que han ayudado a lograr los objetivos que queríamos alcanzar. A lo largo del documento, se examina cada uno de los generadores de tráfico, mostrando sus características y sus cometidos principales, dando una comparativa entre todos ellos, y mostrando resultados de interés para su seguimiento a mayor escala y así lograr mejorar las ventajas que ya proporcionan.

Se ha explorado acerca de los tipos de tráfico existentes y los generadores de tráfico que actualmente se encuentran disponibles. D-ITG y Ostinato fueron dos de los generadores de tráfico que más se ajustaron a las características buscadas; pero finalmente la mayoría de nuestros resultados se obtuvieron con Ostinato gracias a su sencilla interfaz y el permiso de crear diferentes tipos de tráfico a la vez, y generar reportes detallados de los eventos tiempo real.

7.1 Valoración

Este estudio sirve para todos los administradores de redes de datos a nivel mundial ya que hoy por hoy lo que se busca mediante internet es garantizar los servicios y la información que viaja por las redes en general, es por ello que es de utilidad entrar a revisar el diagnóstico que nos brindan estas herramientas a la hora de conocer el desempeño de la red y sus dispositivos.

Así se facilita la detección de errores y posibles fallos en tiempo real que suceden en todos los ámbitos profesionales.

7.2 Líneas de continuación

Se ha contemplado que en un futuro se pueda mejorar fácilmente añadiendo nuevos módulos o modificando los existentes a los generadores de tráfico evaluados.

A continuación se proponen algunas ideas para mejorar el diseño y aportar nuevas funcionalidades:

Añadir la posibilidad de capturar el tráfico recibido por la interfaz y almacenarlo en la memoria para, posteriormente, replicarlo o almacenarlo en disco en formato pcap.

Permitir que el usuario especifique la tasa binaria a la que desea enviar en vez del tiempo entre paquetes, en algunos generadores anteriormente descritos.

Añadir un módulo que recoja estadísticas del envío y que se dé la opción de generar un informe para el usuario.

Modificar el módulo que lee y escribe de la memoria para que aplique un algoritmo de compresión a los datos. De esta manera se aumentaría el tamaño máximo de la traza que se puede replicar.

Y por último comparar el rendimiento monotorizando más herramientas usando TCP y UDP con IPv6. Este trabajo adicional podría decidir qué generador de tráfico realmente proporciona los mejores resultados.

Bibliografía

[1] Horn, G., et al. "An empirical comparison of generators for self similar simulated traffic." *Performance Evaluation* 64.2 (2007): 162-190.

[2] Molnar, Stephen, Peter Megyesi, and Geza Szabo. "How to validate traffic generators." Communications Workshops (ICC), 2013 IEEE International Conference on. IEEE, 2013.

[3] http://traffic.comics.unina.it/software/ITG/

[4] http://code.google.com/p/ostinato/

[5] http://netsniff-ng.org/

[6] http://packeth.sourceforge.net/

[7]http://tomahawk.sourceforge.net/

[8] http://bittwist.sourceforge.net/

[9] http://tcpreplay.appneta.com/wiki/overview.html

[10] http://www.metasploit.com/

[11] http://sourceforge.net/projects/loic/

Anexo A

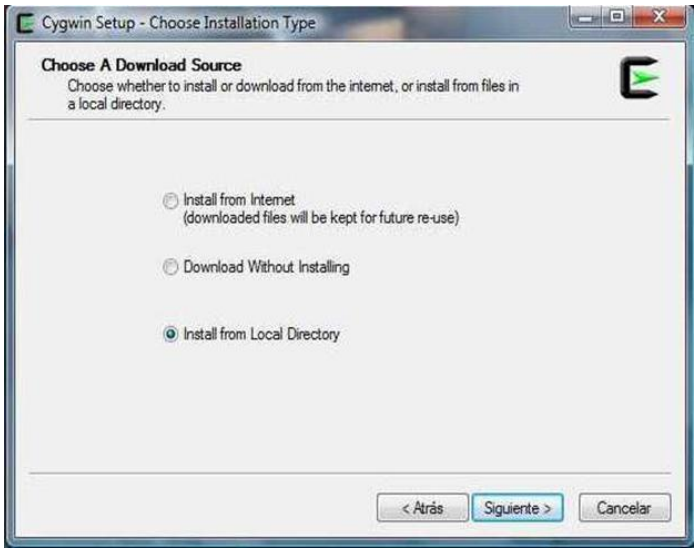
Instalar Cygwin

Se necesita para levantar el D-ITG y Bittwist en Windows, unos pasos más adelante. Hay que recordar que se usa para ejecutar los archivos con extensión *.sh

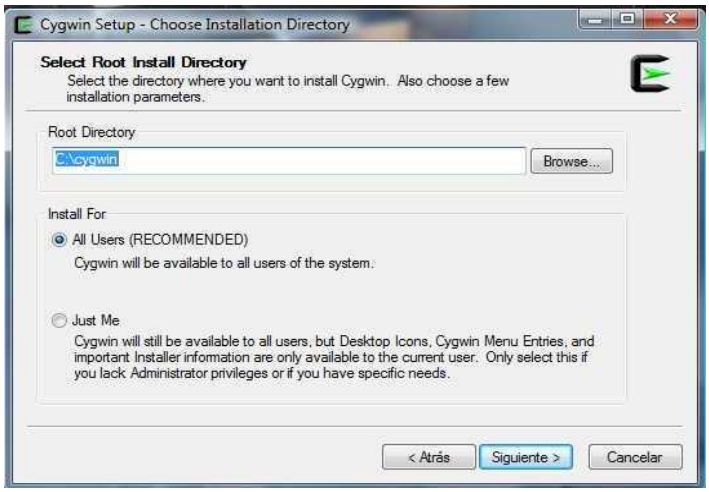
El archivo que se ha ejecutado es: Cygwin.zip. Se descomprime y se ejecuta: setup.exe

Al ejecutar sale el asistente de configuración.

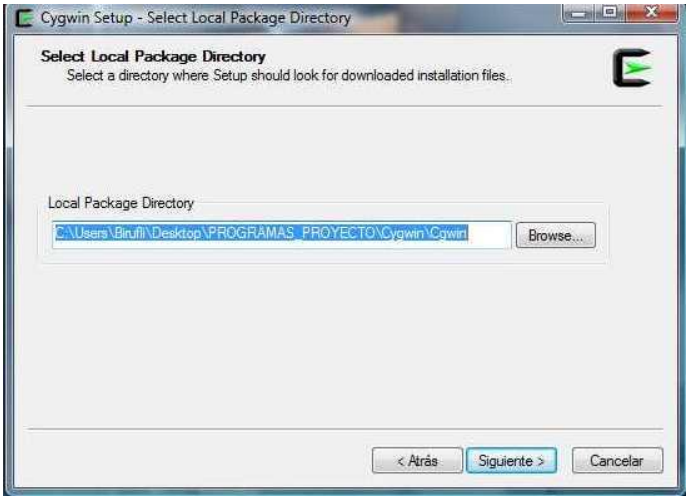
Existen tres formas distintas de instalación, en este caso se optó desde el directorio local



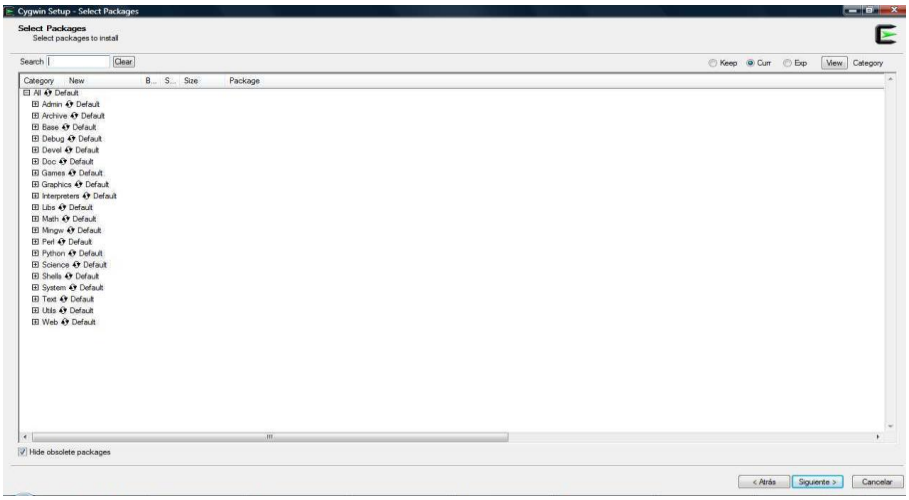
Se elige el directorio raíz de instalación. Se recomienda seleccionar todos los usuarios.



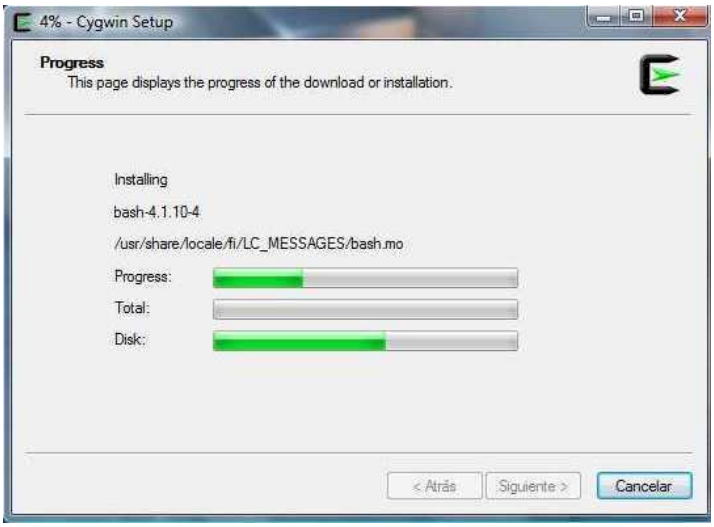
Localizar directorio para instalar.



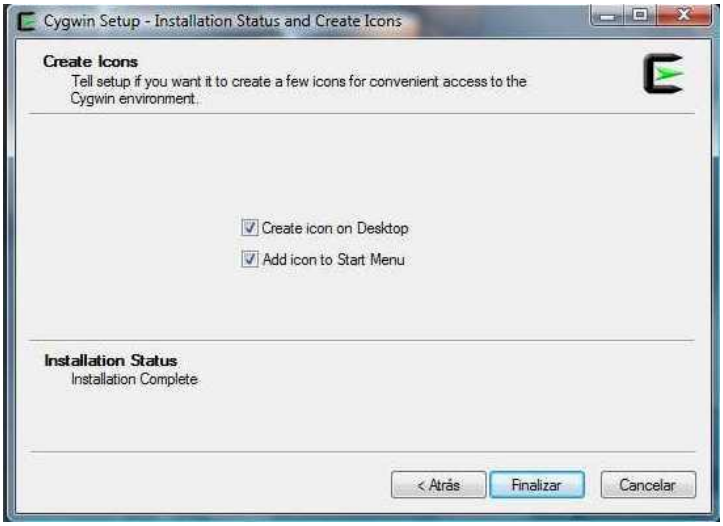
Seleccionar paquetes para instalar.



Progreso de instalación.



Elegir crear iconos. Terminar.



Terminal de Cywin.