

Proyecto Amparo:

**Diseño e implementación de una
Darknet para monitoreo de la
red en Chile – CLCERT**

Informe Final

Renata Faccilongo,
Dept. Cs. de la Computación, U. de Chile

Alejandro Hevia,
Director CLCERT & Profesor Asistente
Dept. Cs. De la Computación, U. de Chile

Sergio Miranda,
Director de Tecnología CLCERT, U. de Chile

31 de Enero 2011

Tabla de contenido

Sección 1: Introducción	3
Motivación	3
Objetivos	4
Antecedentes	4
Funcionamiento y Arquitectura de la Darknet	6
Creación de una Darknet.....	7
Qué ataques conviene monitorear	8
Sección 2: Implementación.....	14
Definición de arquitectura	14
Servidor.....	15
Software instalado	16
Instalación y captura de datos	18
Despliegue de resultados	21
Implementación de gráficos.....	22
Copia y Borrado de material	46
Visibilidad del proyecto	48
Sección 3: Conclusiones	50
Sección 4: Trabajo futuro	50
Anexo: Bibliografía	51

Sección 1: Introducción

Una darknet es una porción o espacio IP asignado y ruteado en el cual no existen servicios activos ni servidores. Tal espacio se denomina "*dark*" (oscuro) porque un observador externo tiene la impresión que no existe nada en dichas redes. Una darknet incluye de hecho al menos un servidor, configurado como un capturador de paquetes. Este servidor captura todo el tráfico y paquetes que ingresen a la darknet, lo cual puede ser utilizado para un análisis en tiempo real o análisis posterior a los eventos o forensicos. Ningún paquete legítimo debiese ingresar a la darknet, dichos paquetes pueden transitar por error o fallas en la configuración, pero la gran mayoría de dichos paquetes son enviados por malware. El malware en general realiza búsquedas activas de dispositivos, por lo que enviará paquetes hacia la darknet que permitirá su captura y análisis.

El concepto de darknet en el cual se basa el presente trabajo, fue el definido por el Team Cymru [CYMRU10]. Sin embargo existen otras aproximaciones interesantes, como el proyecto "Network Telescope" de CAIDA [CAIDA1-10,CAIDA2-10] y el "Internet Motion Sensor", de la Universidad de Michigan [MICH10]. Todos estos trabajos fueron revisados para implementar el siguiente proyecto.

Este proyecto busca obtener la mayor cantidad de antecedentes de modo de determinar el volumen y características del tráfico malicioso que circula por la red, de modo de permitir la definición de acciones en orden a mitigar el impacto que dicho tráfico representa sobre los equipos de dicha red, y a su vez extender dicho conocimiento a una red más amplia, posiblemente incluso a nivel nacional.

Motivación

Hoy en día existen pocos antecedentes confiables sobre el número y características de los ataques provenientes desde el extranjero hacia la red nacional chilena, y prácticamente no se cuenta con información que permita dimensionar el volumen de equipos locales comprometidos con alguna clase de malware y que pueden estar formando parte de *botnets*. El CLCERT, el Grupo de Respuesta a Incidentes de Seguridad Computacional de Chile, busca responder a estas preguntas en virtud de su misión: el monitorear y analizar los problemas de seguridad de los sistemas computacionales en Chile, y reducir la cantidad de incidentes de seguridad perpetrados desde y hacia éstos.

Una poderosa herramienta para cumplir con dicho objetivo es poder contar con datos estadísticos confiables sobre el malware que circula por nuestras redes. Con el fin de obtener tráfico malicioso de la manera más confiable posible, se propone implementar una darknet.

Objetivos

a. Objetivos Generales

Este proyecto tiene como objetivo el diseñar e implementar una darknet, operada por el CLCERT, de modo de estudiar el comportamiento de la red nacional e internacional con el fin de monitorear gusanos, ataques DoS, spam y compromisos en general.

Para su implementación se pretenden seguir los lineamientos del Team Cymru, montando tal red con el equipamiento necesario, y desarrollando las aplicaciones que permitan procesar los datos y desplegar estadísticas relevantes con información útil para identificar, mitigar y/o eliminar los ataques monitoreados, así como reportar y/o compartir información con otras entidades interesadas (como, por ejemplo, otros CERTs).

b. Objetivos Específicos

1. Determinar las características de la red (topología, alcances, visibilidad) más adecuada para montar la darknet en el CLCERT.
2. Definir equipamiento de hardware y el software a instalar de acuerdo a los objetivos definidos.
3. Instalación y obtención de datos de prueba desde la darknet, de acuerdo a los diversos usos que se hayan definido (recolección de flujos, detección de ataques DoS, análisis vía IDS, etc.) para su análisis.
4. Desarrollo de aplicaciones de captura, almacenamiento y procesamiento de los datos de acuerdo al resultado de las pruebas previas y al flujo de datos estimado.
5. Despliegue de estadísticas y procedimientos de manejo de la información, por ejemplo para compartir con otros grupos o el envío a subsistemas para un procesamiento más profundo.

Antecedentes

La topología de Internet está en constante evolución y ha habido cambios dramáticos en cuanto a la accesibilidad, apareciendo periódicamente nuevos métodos en que el software malicioso se propaga y se detecta. Al mismo tiempo, los firewalls y dispositivos NAT diseñados para proteger los hogares y las empresas, están empezando a ser permeables a muchas de las amenazas a las que fueron diseñados para defenderse. En particular, los usuarios móviles actúan como portadores de programas maliciosos, puntos de acceso inalámbricos proporcionan puertas traseras a muchas redes, y aplicaciones complejas tienen agujeros abiertos a través de firewalls. El resultado final ha sido una proliferación de la actividad maliciosa sin ser detectados dentro de los perímetros de la red.

Para combatir el aumento de las amenazas dentro de la red y la falta de visibilidad de las subredes, se busca construir una darknet y luego usar la información para reducir los ataques o daños provocados por malware. Las darknets tienen múltiples usos: acoger el flujo de los colectores, los detectores de backscatter, analizadores de

paquetes y cajas de IDS (Intrusion Detection System) [UNAM10] . La elegancia de la darknet es que reduce considerablemente los falsos positivos en comparación a cualquier otro dispositivo o tecnología.

Algunas organizaciones se han preocupado de implementar soluciones específicas para este problema. Tal como se mencionó anteriormente, el Team Cymru popularizó el concepto de Darknet. Pero además, existen otras organizaciones que han servido de inspiración y ayuda. Tal es el caso del laboratorio **CAIDA**, con su proyecto llamado "*Network Telescope*" o telescopio de red en Castellano [CAIDA1-10,CAIDA2-10,CAIDA3-10]. Un telescopio de red es una porción de IPs donde debería existir poco o nada de tráfico legítimo la cual es monitoreada (el tráfico entrante) a fin de obtener una visión de lo que está ocurriendo en la red. Con este dispositivo es posible hacer visible y estudiar diversos ataques, incluyendo de denegación de servicios, infección de máquinas con gusanos de Internet y escaneos de red, así como problemas asociados a malas configuraciones y pobres políticas de seguridad. Intuitivamente, un telescopio de red permite estudiar el comportamiento de máquinas a distancia (aquellas comprometidas) sin estar necesariamente cerca de ellas. Si un equipo envía paquetes a direcciones IP al azar, se deberían poder ver algunos de los paquetes si se hace un seguimiento en un determinado espacio de direcciones.

Telescopios más grandes son capaces de detectar eventos que generan menos paquetes, ya sea por la corta duración o la baja velocidad de transmisión. Además los telescopios más grandes tienen una mayor precisión en la determinación de la hora de inicio y de término de un evento, no así los pequeños. Éstos últimos pueden no ser útiles para observar eventos externos, pero puede resultar muy efectivo tener uno interno para identificar rápidamente problemas internos. Los telescopios internos, por otro lado, son muy útiles al momento de detectar máquinas internas infectadas con gusanos, malas configuraciones o equipos hackeados. Además permite la captura de datos para hosts conectándose a IPs sin asignar en un espacio de direcciones.

Los telescopios distribuidos se usan para aumentar su tamaño, esto lo hacen ocupando bloques de espacio de direcciones que no sean contiguos. Algunas ventajas son que reduce la dependencia del alcance de un único bloque, el tráfico de carga se puede propagar por distintos sitios, y puede evitar que sean saltados (omitidos accidental o intencionalmente) por algoritmos de selección. Algunas desventajas y desafíos de esta manera de organización de los telescopios son que las estadísticas pueden resultar engañosas (diferentes piezas pueden tener diferente alcance en diferentes tiempos), sincronización del tiempo, y distribución de la información. Pero afortunadamente existen esfuerzos, tanto voluntarios como comerciales, para solucionar estos temas.

Además existen los telescopios *anycast* [MOORE10], los cuales anuncian el mismo prefijo de la dirección para muchos lugares. Son similares a los telescopios distribuidos en cuanto a ventajas y desventajas pero no se recibe la diversidad de los rangos de dirección de bloque. Puede proveer rutas para los hosts finales más cortas al telescopio, lo que podría mejorar el monitoreo cuando la red está sobrecargada.

Una iniciativa similar es la desarrollada por la Universidad de Michigan. Su proyecto, el *Internet Motion Sensor* [MICH10] se basa en el mismo concepto de redes "oscuras" distribuidas, pero donde el tráfico es respondido en forma minimal (ante un intento de conexión SYN, cada IP de la darknet responde con un SYN-ACK). Con ello, en principio es posible capturar tráfico más relevante (por ej. en detección de gusanos TCP),

ciertamente a un costo mayor en la implementación y operación del sistema. Para más información al respecto visitar [MICH10].

Funcionamiento y Arquitectura de la Darknet

El tráfico malicioso generado por escaneos o la propagación de malware no utiliza nombres de dominio para localizar a sus víctimas, si no que seleccionan un rango de direcciones y lo escanean en busca de equipos vulnerables a ciertos exploits. Ésta es justamente la razón por la cual se busca monitorear el tráfico de un segmento de red "oscuro", esto es, sin servicios hacia el exterior. Sin embargo, en este tipo de monitoreo también es posible detectar tráfico correspondiente a equipos con configuración de red errónea, como por ejemplo mensajes de broadcast enviados a un segmento al cual no pertenece el emisor.

A grandes rasgos, la arquitectura de la darknet propuesta consiste en re-direccionar el tráfico de un cierto segmento de red a un servidor recolector el cual interactúa con otros equipos internos, los cuales ejecutan las aplicaciones de administración, y procesamiento, en particular, clasificación / filtrado de datos y generación de estadísticas (Figura 1).

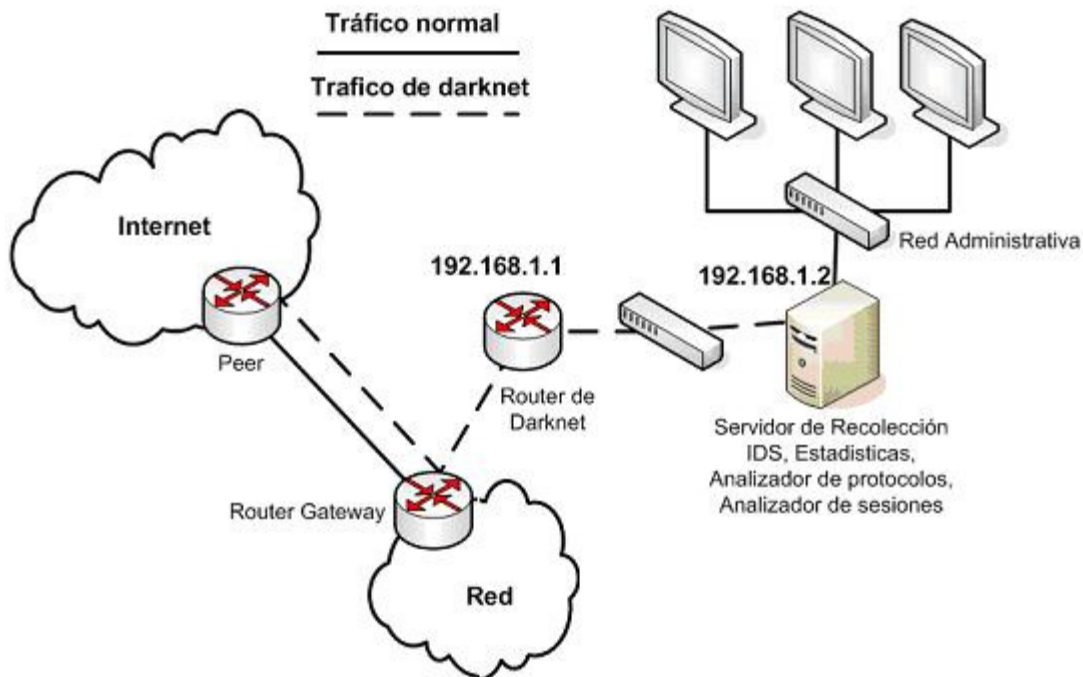


Figura 1: Arquitectura general de la darknet CLCERT propuesta.

El tamaño del rango de direcciones monitoreado no impacta considerablemente en la calidad de la información capturada, pero sí en la precisión de la actividad maliciosa en la red.

Creación de una Darknet

Para crear una darknet, se deben redireccionar paquetes y solicitudes a un computador en específico para luego hacer análisis o interactuar con el atacante para obtener resultados más fidedignos. Según [AskStu10] los pasos para implementar una darknet son:

Paso 1:

Elegir una región o segmento de direcciones IPs de la red, las cuales serán ruteadas hacia la darknet.

Paso 2:

Para configurar la arquitectura del sistema, se requiere utilizar un router o un *switch layer-3*. La idea es que transmita todo el contenido que entre a la darknet. Además se necesita un servidor que actúe como recopilador de datos y un *switch ethernet* para interconectar el servidor y los equipos de la darknet. Además sería deseable un sensor IDS o un analizador de protocolos. El router debe ser configurado para que reenvíe el tráfico destinado a la darknet al servidor de recolección. Para facilitar la administración, es importante contar con una interfaz ethernet adicional en el servidor recolector. Estos equipos deben estar cuidadosamente securizados, pues recibirán tráfico claramente malicioso en forma continua.

Paso 3:

El servidor de recolección de datos debe entender el protocolo ARP (Address Resolution Protocol) . Además, se debe configurar el router de manera que canalice todo el tráfico destinado a la darknet a una única dirección IP en la interfaz ethernet en el servidor.

Ejemplo: si se está usando un router cisco y una red /30, el punto entre el router y la interfaz de la darknet puede ser 192.168.2.0/30, el router de la interfaz Ethernet 192.168.2.1/30 y el servidor recolector 192.168.2.3/30. Con ello, se deben realizar los siguientes comandos de manera que todo el tráfico destinado a la darknet llegue a 192.168.2.2 (servidor recolector).

```
router# conf t
router (config)# ip route [rango interno de IP] [mascara subnet] [IP
recolector ]
router (config)# ^Z
router# wr
```

Paso 4:

Se debe tener una herramienta de análisis de archivo de registro sincronizada para el *logging* del firewall. Esto puesto que los firewalls pueden fallar o apagarse accidentalmente provocando que el tráfico de la darknet puede pasar sin filtrarse. Por ello, debemos considerar agregar rutas nulas. Un ejemplo de ello es:

```
route add-net 10.0.0.0 / 3 127.0.0.1-blackhole
```

Observamos que, dependiendo del hardware, de las opciones de software y del tamaño de la darknet, el logging podría reducir el rendimiento de la darknet.

Paso 5:

Ahora que ya se tiene la idea del darknet, se necesita almacenar la información en un formato que sea útil para el posterior análisis. Una buena idea sería utilizar archivos binarios con formato *pcap* ya que una gran cantidad de aplicaciones de análisis de red pueden operar con ellos. La manera más fácil de hacer esto es mediante *tcpdump* (con su característica incorporada de rotación).

Un ejemplo de un comando de *tcpdump* para cumplir con la rotación necesitada sería:

```
tcpdump -i [interfaz a escuchar] -n -w [archivo a escribir] -C125
```

En este ejemplo, el DNS está desactivado, el archivo se escribe cada 125 millones de bytes comprometidos, y *n* se incrementa para nombres de archivos únicos.

Posibles ataques a monitorear

Hoy en día en la red existen múltiples ataques tales como ataques de denegación de servicios (directos o distribuidos, DDoS), ataques de Respuesta DNS, ataques directos a routers, escaneos de puertos mediante el envío de paquetes ICMP y ataques a servicios particulares tales como Web, Mail, SSH, FTP (archivos) entre otros. La pregunta es, en general, ¿qué ataques y eventos se pueden detectar a través de una darknet o telescopio de red? Según [Mart10] éstos incluyen

- Actividad sospechosa por puertos (diferentes protocolos, TCP, UDP, ICMP, etc.)
- Tráfico relacionado con servicios específicos (SSH, WEB, DB, etc.)
- Direcciones IP y dominios en lista negra.
- Ataques comunes a equipos de la red (fuerza bruta, escaneos, etc)
- Patrones generados por malware (escaneos, tráfico excesivo, baja de servicios)
- Flujo de tráfico (gusanos, virus, exploits automatizados, etc.)
- Botnets dentro y fuera de la red.
- Posible tráfico malicioso hacia redes externas (spam, phishing, etc.)

Sin embargo, al considerar que ataques son posibles de monitorear se debe considerar el espacio y dificultad de cómputo de realizarlo. En este sentido, una vez que se ha capturado tráfico de red, éste debe ser parseado y procesado automáticamente, posiblemente mediante scripts, en tiempo real, debido a la gran cantidad de datos que se reciben diariamente. Los principales objetivos del procesamiento son:

- Clasificar la información.

- Formato de la información
- Detección de falsos positivos

Para responder la pregunta anterior, a continuación se revisan los principales ataques estudiados por los propulsores de este tipo de dispositivos (concretamente, el Team Cymru, CAIDA, y la Universidad de Michigan).

La red del telescopio de red (UCSD) y la darknet recolectan tráfico resultante de un amplio rango de eventos posibles, tales como malas configuraciones (por ejemplo el uso de direcciones IPs incorrectas), escaneo malicioso de un espacio de direcciones realizado por hackers que buscan vulnerabilidades, ataque de denegación de servicios y propagación automática de software malicioso (gusanos). En cuanto a ataques, los principales estudiados en los que las mencionadas organizaciones son los siguientes [CAIDA3-10].

a. Ataque de denegación de servicios (DoS):

El telescopio de red de CAIDA (U. California San Diego, EE.UU.) puede ser utilizado para monitorar la propagación aleatoria de malware (código malicioso) para crear denegación de servicios. Para dificultar que la víctima bloquee el ataque, el adversario usualmente utiliza una dirección IP fuente falsa y aleatoria (análogo a alterar la dirección del remitente en el correo) en cada paquete que envía a la víctima. Dado que el atacado no es capaz de distinguir entre una solicitud proveniente de un atacante o una legítima, la víctima del DoS intenta responder a todas las solicitudes que le llegan. Cuando el atacante falsifica una dirección de origen en el telescopio de red, se le hace el seguimiento de una respuesta destinada a un computador que no existe (y por lo tanto nunca se envía la consulta inicial) (Figura 2). Al monitorear estas respuestas no solicitadas, los investigadores pueden identificar a la víctima del ataque de DoS, inferir información sobre el volumen del ataque, ancho de banda de la víctima, su ubicación y los tipos de servicio a los que apunta el atacante.

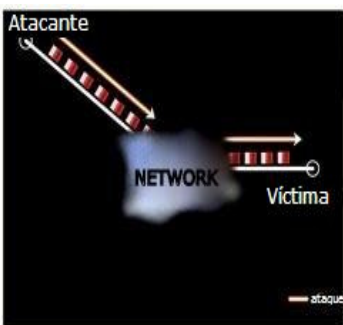


Figura 1: El atacante envía paquetes con una dirección de origen falsa a la víctima de DoS.

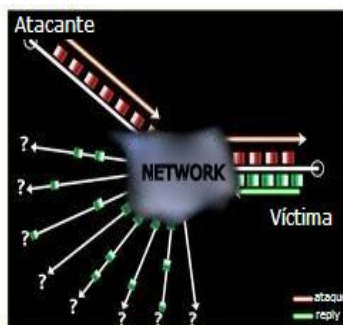


Figura 2: La víctima no puede distinguir entre solicitudes legítimas o falsas por lo tanto responde a todas las posibles.

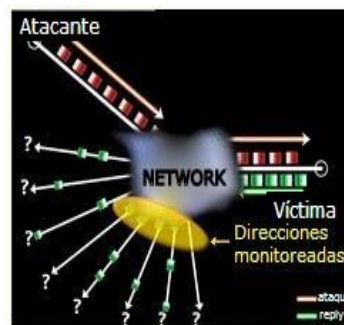


Figura 3: El telescopio está compuesto de 1/256 del espacio de direcciones IPv4, por lo tanto recibe aproximadamente 1/256 de las respuestas dadas a solicitudes con IP falsa generadas por la víctima del DoS.

Figura 2: Detección de un ataque denegación de servicio (DoS)

Algunos ataques de DoS no usan una dirección IP falsa al atacar a sus víctimas. Claramente, el telescopio no monitoreará ataques que utilicen un legítimo ataque de

Denegación de Servicios, y tampoco monitoreará ataques que no usen una IP falsa generada aleatoriamente.

Cabe recordar que el tipo de darknet utilizada por el Team Cymru es del tipo *agujero negro*, es decir, todo entra al espacio de direcciones IP pero nada se escapa, a no ser que el administrador del sistema decida ingresar y analizar algún dato en específico. Obviamente, esto trae ventajas y desventajas a nivel de monitorear ataques.

Los paquetes TCP pueden ser SYNs (intentando crear una conexión), ACK-SYN (respondiendo a un falso SYN enviado desde alguna parte) o RSTs respondiendo también a paquetes falsos que no fueron originados por la IP fuente indicada. Los paquetes UDP intentan obtener una respuesta de algún tipo, pero los paquetes ICMP nunca son enviados como respuesta de una conexión TCP o UDP puesto que son principalmente solicitudes.

En general, la utilidad de los datos recolectados va más allá de los ataques mencionados. Ellos se utilizan para el análisis forense de nuevos ataques, en contextos/sistemas diferentes o que afecten a redes distintas a la red monitoreada.

b. Gusanos

Dado que el espacio de direcciones de la darknet nunca responde (son IPs sin dueño) es inusual que un computador de la darknet sea el blanco de un ataque, a no ser que se esté frente a un gusano de gran escala, o que se propague solo. Otra posibilidad son errores de configuración humanos por parte de administradores de sistema.

Se puede utilizar Snort para buscar ataques basados en UDP o ICMP, porque ellos tienen "payloads", pero un ataque mediante TCP nunca mostrará sus "payloads". Los paquetes TCP son principalmente de reconocimiento (de sondeo para IPs con puertos de escucha) o retrodispersión de ataques de suplantación de direcciones dirigidas a otra víctima.

Muchos gusanos de Internet se transmiten a través de la generación aleatoria de una dirección IP la cual será el blanco del ataque, con la esperanza que esta dirección generada esté en uso por un equipo vulnerable. Si, por ejemplo, el segmento de red monitoreado contiene una fracción $1/256$ de todos los números IPs disponibles, entonces la red recibirá una de cada 256 de las sondas de los hosts infectados con gusanos de escaneo aleatorio (Figura 3).

Muchos gusanos no escanean realmente al azar, y los problemas de red (provocados por gusanos entre otros) pueden evitar que el telescopio reciba las sondas de todos los hosts infectados. Sin embargo, en general, en experiencias de este tipo (UCSD), el telescopio ha logrado detectar nuevos infectados transmitiendo a una velocidad lenta de como 10 paquetes por segundo, durante los 30 segundos que dura la infección.

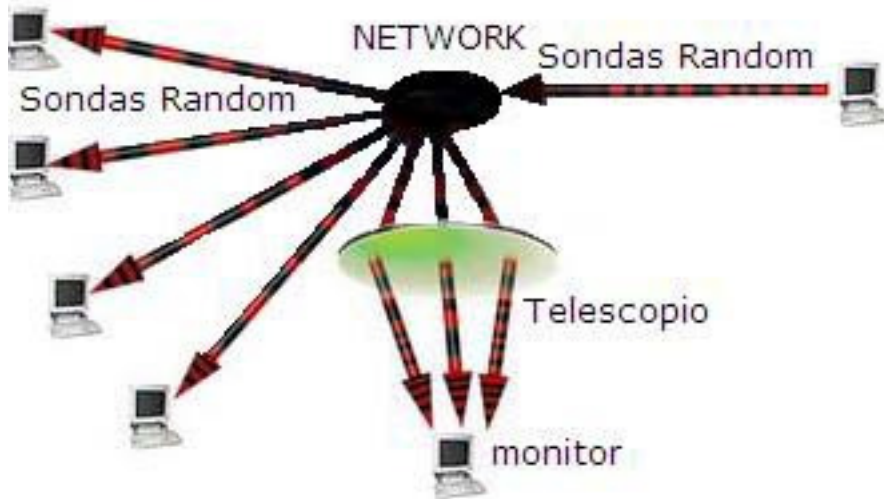


Figura 3: Detección de gusanos. Los computadores infectados de manera aleatoria intentan infectar a otros computadores vulnerables. El telescopio monitorea aproximadamente 1 de cada L intentos de ataque para un segmento de red consistente de L números IPs distintos.

Una forma en que se podría manejar la información dada por la darknet es convirtiendo los archivos de datos *tcpdump* al formato de *argus*, un formato relativamente estándar. Esto permitiría ingresar todos los registros a una base de datos para luego realizar consultas SQL sobre ella sobre patrones específicos buscados. Sin embargo, tal procesamiento es costoso computacionalmente. Cabe destacar que siempre se deben retener los *tcpdumps* porque así se tiene la posibilidad de volver a los datos originales y mirar las cabeceras de los paquetes e incluso información de la data (*payload*) de UDP o ICMP.

Toda esta información puede ser usada luego cuando se detecte un patrón específico que determine que se está frente a un gusano. Por ejemplo los paquetes del gusano Slammer pueden ser encontrados usando una expresión del tipo "libpcap" como la siguiente:

```
udp and dst port 1434 and 'udp[4:2] == 384'
while Nachi was ip[3] == 0x5c && icmp[icmptype] == icmp-echo && icmp[24:2] ==
0xaaaa
```

Por ello, un procedimiento posible para detectar gusanos buscará primero recopilar los datos de cada nuevo ataque en formato *tcpdump*, para luego buscar patrones de interés. Una búsqueda exitosa se reduce entonces a construir un script adecuado, el cual puede modificarse y perfeccionarse.

Para detectar virus o gusanos se puede utilizar Snort [Snort10], el cual implementa un motor de detección de ataques y barrido de puertos que permite registrar, alertar y responder ante cualquier anomalía previamente definida. Cuando un paquete coincide con algún patrón establecido en las reglas de configuración, se almacena en el log. Así se sabe cuándo, de dónde y cómo se produjo el ataque. Snort dispone de una base de

datos de ataques que se está actualizando constantemente y a la cual se puede añadir o actualizar online.

c. Escaneo Malicioso

Existen intentos automáticos, semiautomáticos y manuales para localizar computadores vulnerables en Internet. El escaneo difiere de otro tipo de tráfico visible a través del telescopio porque no se produce aleatoriamente. Los motivos del atacante al momento de elegir su blanco parecen ser arbitrarios desde la perspectiva de la víctima. El telescopio recibe muchos tipos de escaneos continuamente, incluyendo escaneos basados en ping por la existencia de un dispositivo en una determinada dirección IP, explotaciones secuenciales de una pequeña cantidad de puertos vulnerables secuenciales pertenecientes a un cierto rango de direcciones IP, e incluso escaneos utilizando reseteo TCP.

Probablemente el valor más subestimado de una darknet (especialmente cuando se combinan datos provenientes de múltiples darknets) es que permite determinar la línea base, es decir, ¿qué es lo normal? ¿es normal ver 5 o 5000 paquetes "sondeando" para un número de puerto? Si se ven 10 paquetes por mes durante todo un año para el espacio de la darknet, y de pronto se ven 10000 paquetes, significa que algo ha cambiado, quizás haya una nueva vulnerabilidad. Sería un buen momento entonces para comprobar si hay actualizaciones o si es que se tienen servicios en el puerto solicitado, etc.

Sin embargo, no siempre la información entregada por la darknet pudiera ser fácil de interpretar. Por ejemplo, un escaneo de puertos limitado (uno o más puertos) pudiera deberse a variados motivos (administración de red, etc.), todos legítimos. En casos que los datos entregados por la darknet frecuentemente no puedan comprobar la existencia de un ataque, una alternativa posible es considerar datos provenientes de otros hosts distribuidos geográficamente a fin de contrastar si ellos han detectado dichas pruebas/eventos o no.

La información que entrega la darknet de una red puede ser muy distinta a la que entrega la de otra red. Por ejemplo, el espacio de direcciones de un típico usuario doméstico tendrá un comportamiento muy distinto al de una red universitaria, corporativa o de gobierno. La clave para hacer uso de la información entregada por la darknet es combinarla con datos de redes ocupadas. Es interesante comprar una darknet ubicada afuera del firewall principal (que solamente ve paquetes del exterior) con una darknet ubicada dentro de la red (que no ve paquetes originados afuera). Una mostrará qué tan mal están las cosas, y lo importante de la presencia del firewall, mientras que otra mostrará los problemas originados en el interior.

Existen herramientas tales como PortSentry [BAD10] que monitorea escaneos de puertos que se hacen hacia un determinado computador. Es una aplicación muy potente, que junto con iptables puede ayudar a detectar ataques a los servidores. Por otro lado se puede contar con el Firewall el cual debe ser suficientemente bueno como para detectar pruebas de escaneo de un atacante. El Firewall debe realizar inspecciones confiables teniendo un conjunto específico de reglas (rule set). Se deben usar NIDS (Network Intrusion Detection System) [WISEDATA10] para evitar el intento

de detección de sistemas operativos usando herramientas como Nmap. Finalmente solo los puertos necesarios deben permanecer abiertos, el resto deben ser filtrados.

Los ataques desarrollados anteriormente son la principal preocupación para CAIDA y para el Team Cymru. Muchas restricciones de privacidad y seguridad están asociadas a los datos que entregan estas organizaciones. Esto es porque algunos virus, además de los daños que provocan, incluyen la instalación de puertas traseras y proveen acceso sin restricciones a computadores infectados; los datos rescatables desde el telescopio contienen características que advierten sobre estas máquinas infectadas o vulnerables. Además, si bien el origen de algunos ataques es evidente, un importante volumen del tráfico tiene un volumen desconocido. Sin identificar completamente el tráfico, no se puede evaluar la seguridad y la privacidad del impacto que produciría la filtración de los datos.

Una darknet sólo puede proporcionar un elemento más de análisis de los ataques existentes en la red. Algunos de los telescopios funcionan de diferente manera, limitando el alcance de la conexión, pero permiten la configuración y de ahí nace la posibilidad de obtener las suficientes características para determinar la intención (escaneo o explotación).

Para más información respecto a este tema ver [CAIDA1-10,CAIDA2-10,CAIDA3-10].

Sección 2: Implementación

Definición de arquitectura

Diagrama Conceptual

El diseño de la red es el recomendado para una Darknet como la que se desea implementar. Todo el tráfico dirigido hacia la red monitoreada es ruteado hacia un servidor de recolección, no se permite que haya tráfico que escape de dicha red (ver figura 4). El servidor de recolección posee dos interfaces, una para el monitoreo y otra para administración.

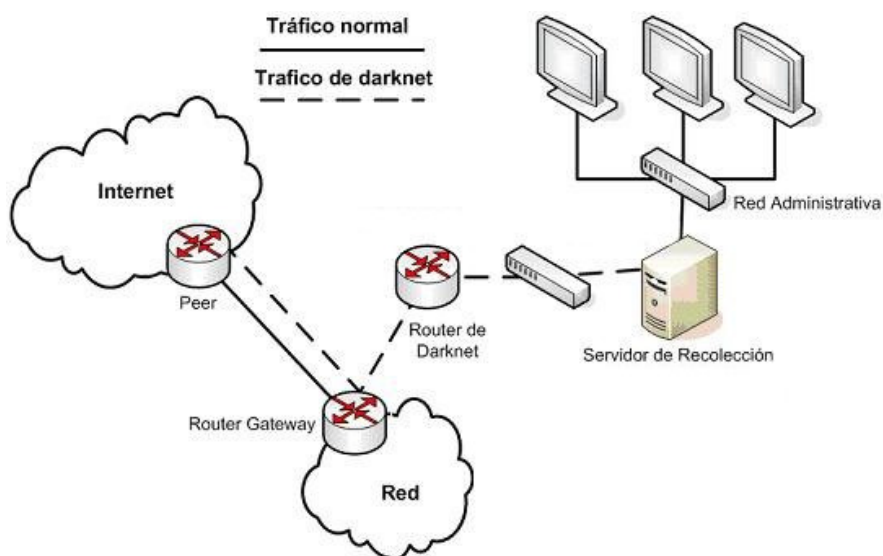


Figura 4: Diagrama conceptual de la darknet

Diagrama Físico

En la Red de monitoreo se encuentran ubicados los equipos clientes del CLCERT que administran la Darknet. El tráfico capturado que proviene de INTERNET es dirigido hacia la red Darknet, en este caso, por razones de seguridad, la enmascaremos como la red X.X.X.X/24. La conexión entre el servidor y el router se realiza a través de un red privada 192.168.66.0/24. Este tráfico llega a través de uno de los dos enlaces que posee la Facultad, a través de los router 200.9.99.130 y 172.16.38.137, este último es el router que nos conecta con los servicios centrales de la Universidad. Todo el tráfico que proviene de la red monitoreada es redirigido por los switches internos hacia el servidor de monitoreo, el 172.17.66.67.

La red interna de administración del CLCERT esta conectada directamente al switch 10.0.1.119 para acceder a la interfaz interna del servidor de monitoreo.

1. Presupuesto máximo: 2200 US\$
2. Case rackeable
3. Disco duro: 200GB
4. Memoria: 1GB
5. CPU: Intel Xeon 2.0GHz (o AMD equivalente)
6. 2 tarjetas de red Gigabit
7. 3 años de garantía en sitio

Con estas características y considerando nuestra experiencia administrando servidores y el equipamiento actualmente utilizado en otros proyectos dentro de la Universidad, las posibles marcas a cotizar fueron HP, IBM y Dell. Dell fue descartado debido a que actualmente no existe un respaldo adecuado para el sector Educación, por lo que la búsqueda en el portal se limitó a HP e IBM.

Los servidores que cumplieron con los requisitos fueron:

Modelo	Detalle	Precio US\$
HP DL120-G5	Intel(R) Xeon(R) CPU E3110 @ 3.00GHz, 1GB Ram, 2 discos 160GB	2000
IBM X3550	Intel(R) Xeon(R) CPU E3110 @ 3.00GHz, 1GB Ram, 1 disco 160 GB	2190
HP DL 160-G6	Intel(R) Xeon(R) CPU E3100 @ 3.00GHz, 1GB Ram, 1 disco 160GB	2150

De las tres alternativas se decidió adquirir el servidor HP DL120-G5 , el cual ofreció el mejor precio y el mejor rendimiento ya que permitía duplicar el espacio en disco, característica de suma importancia en el proyecto. El equipo se adquirió directamente en el portal de compras y estuvo disponible a partir de mediados de mayo para su utilización.

Software instalado

La decisión del sistema operativo a instalar se basó principalmente en la familiaridad de uso y su seguridad.

Se evaluaron los sistemas CentOS y OpenBSD, TeamCymru también recomienda Sun Solaris, el que fue descartado porque no se encuentra entre los SO desplegados en los sistemas administrados. CentOS es el más utilizado en los servicios que entrega la Facultad y está basado en RedHat Enterprise, es un sistema robusto e ideal para entregar servicios en la red. Sin embargo se optó por OpenBSD 4.2 version 64 bits, sistema operativo libre y portable, el que fue recomendado por la comunidad (proyectos encabezados por CERTBr y TeamCymru) para este tipo de proyectos por ser seguro y simple de instalar (<http://www.openbsd.org/es>).

Se realizó una instalación de los paquetes por omisión y se particionó el disco duro con el siguiente esquema:

Filesystem	Size	Mounted on
/dev/wd0a	1005M	/
/dev/wd0k	122G	/home
/dev/wd0d	3.9G	/tmp
/dev/wd0f	2.0G	/usr
/dev/wd0g	1005M	/usr/X11R6
/dev/wd0h	5.9G	/usr/local
/dev/wd0j	2.0G	/usr/obj
/dev/wd0i	2.0G	/usr/src
/dev/wd0e	5.9G	/var

Se configuró una de las interfaces de red con la dirección IP de administración (correspondiente a una dirección en la red 172.17.66.0/24) y la otra interfaz con la dirección de ruteo para la red de monitoreo de la darknet (192.168.66.2)

```
# ifconfig -a
```

```
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 33152
    priority: 0
    groups: lo
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
em0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    lladdr 00:23:7d:fd:ac:e1
    priority: 0
    groups: egress
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 172.17.66.67 netmask 0xfffff00 broadcast 172.17.66.255
    inet6 fe80::223:7dff:febd:ace1%em0 prefixlen 64 scopeid 0x1
bge0: flags=8b43<UP,BROADCAST,RUNNING,PROMISC,ALLMULTI,SIMPLEX,MULTICAST> mtu
1500
    lladdr 00:26:55:57:82:2d
    description: Monitoreo
    priority: 0
    media: Ethernet autoselect (1000baseT full-duplex)
    status: active
    inet 192.168.66.2 netmask 0xfffffff broadcast 192.168.66.3
    inet6 fe80::226:55ff:fe57:822d%bge0 prefixlen 64 scopeid 0x2
```

Se decidió instalar sólo el software básico para la captura de paquetes (tcpdump). Todo software adicional que se requiera para el despliegue se instalará en un equipo adicional.

Instalación y captura de datos

a. Red Darknet

La Universidad posee algunas redes clase C en desuso desde hace algunos años, lo cual calza muy bien con este proyecto. Se escogió utilizar la red X.X.X.X/24 la cual hace unos 3 años que fue liberada. Se configuró el router para que reenvie todo el tráfico de dicha red. Los comandos son los siguientes:

- i) Se activó la red X.X.X.X/24 en el switch principal, apuntando a una ruta nula.

```
#no ip route X.X.X.X 255.255.255.0 null 0
```

- ii) Se redirige el tráfico de esta red a la ip interna del servidor de recolección.

```
# ip route X.X.X.X 255.255.255.0 192.168.66.2
```

El servidor se habilito como router para pasar el tráfico entre las interfaces:

```
# sysctl net.inet.ip.forwarding=1
```

b. Captura de tráfico

El tráfico se captura via tcpdump desde la interfaz de captura (bge0) y se almacena en archivos diarios para su posterior análisis. Para ello se ejecuta una tarea via crontab que inicia la escritura de un nuevo archivo:

```
0 * * * * /bin/sh /var/darknet/bin/captura.sh
```

El script captura.sh es el siguiente:

```
#!/bin/sh
hoy=`date +%F`
out=salida-$hoy

# Busca el pid del proceso de captura y lo termina
pid=`ps -aux | grep tcpdump | grep -v grep | awk '{print $2}'`
kill -9 $pid
# comienza una nueva captura
cd /var/darknet/capturas
tcpdump -w $out -n -i bge0 ip or udp or icmp &
```

El comando " tcpdump -w \$out -n -i bge0 ip or udp or icmp " captura todo el tráfico de tipo IP/UDP/ICMP que entre a través de la interfaz de monitoreo.

Los archivos capturados tienen un tamaño promedio de 6Mb con 100 mil paquetes capturados por día (aproximadamente).

c. Definición de estructuras de datos y aplicaciones

Los archivos de datos capturados se almacenan en archivos binarios diarios de tipo pcap (package capture). Este formato permite guardar el paquete capturado en forma completa y existen muchísimas aplicaciones que permiten manipular dichos datos para su análisis y despliegue. La darknet genera los datos diarios y almacena los datos en archivos cuyo nombre tiene el formato salida-año-mes-día (por ejemplo salida-2011-01-02).

Esquema XML

Para poder compartir la información obtenida, se requiere definir un formato simple y estandarizado. Para ello se decidió utilizar un esquema XML que permita enviar información a terceros y alimentar subsistemas futuros.

El esquema definido es el siguiente:

```
<proto></proto>    protocolo transmitido (tcp,udp,icmp)
<IPsrc></IPsrc>    Direccion IP de origen
<PORTsrc></IPsrc>  Puerto de origen
<IPdst></IPdst>    Direccion IP de destino
<PORTdst></PORTdst> Puerto de destino
<length></length> Largo del paquete
```

Los datos requeridos pueden obtenerse desplegandolos con la sentencia:

```
tcpdump -nqtr <archivo pcap>
```

Se construyó un script que permite procesar un archivo determinado y construir el XML detallado. El script es el siguiente:

```
#!/bin/sh
case $# in
  0|1|2) echo "gen_xml dump anho mes [dia]"; exit 1;;
  3) dump=$1; anho=$2; mes=$3;;
  *) dump=$1; anho=$2; mes=$3;dia=$4;;
esac

echo "<dia-ataques>"
echo " <anho>"$anho"</anho> "
echo " <mes>"$mes"</mes>"
echo " <dia>"$dia"</dia>"
echo "</dia-ataques>"
echo " <head> Darknet </head>"
echo "<datos>"
tcpdump -nqtr $dump | sed 's/\./ /g' | sed 's:/ /' | awk -f procesa_linea.awk
echo " </datos>"
```

El código awk del proceso "procesa_linea.awk" es el siguiente:

```
{ print "<item>";

  if (NF==14) {
    print " <proto>tcp</proto>";
    print " <IPsrc>"$2"."$3"."$4"."$5"</IPsrc>";
    print " <PORTsrc>"$6"</IPsrc>";
    print " <IPdst>"$8"."$9"."$10"."$11"</IPdst>";
    print " <PORTdst>"$12"</PORTdst>";
    print " <length>"$14"</length>";
    print "</item>";
  }
  if (NF==15) {
    print " <proto>udp</proto>";
    print " <IPsrc>"$2"."$3"."$4"."$5"</IPsrc>";
    print " <PORTsrc>"$6"</IPsrc>";
    print " <IPdst>"$8"."$9"."$10"."$11"</IPdst>";
    print " <PORTdst>"$12"</PORTdst>";
    print " <length>"$15"</length>";
    print "</item>";
  }
  if (NF==19) {
    print " <proto>icmp</proto>";
    print " <IPsrc>"$2"."$3"."$4"."$5"</IPsrc>";
    print " <IPdst>"$7"."$8"."$9"."$10"</IPsrc>";
    print " <length>"$19"</length>";
    print "</item>";
  }
}
```

Búsquedas por ASN

Con el fin de proveer mecanismos futuros para compartir datos, se generaron procedimientos que permitan obtener informacion de un determinado grupo de ASNs , de modo de filtrar las IPs que pertenecen a la entidad solicitante. Para ello se hace uso de las utilidades que provee TeamCymru y que permiten obtener el ASN asociado a una direccion IP determinada. Por ejemplo:

```
tcpdump -nr $dump | awk '{print $2}' | awk -F. '{print $1"."$2"."$3"."$4}' >> ips_entrada
echo begin > ips_distintas
sort -u ips_entrada -o ips_distintas
echo end >> ips_distintas
nc whois.cymru.com 43 < ips_distintas | sort -n > ips_asns
```

El archivo ips_asns contendra lineas de la forma *ASN number | IP | descripcion* , por ejemplo:

```
27680 | 186.40.7.97 | TELEFONICA MOVIL DE CHILE S.A.
```

Con esta información es simple generar el XML filtrando por las IPs requeridas.

Despliegue de resultados

Para el despliegue de estadísticas de la información recolectada se requieren elementos adicionales, como por ejemplo, un equipo donde se instale el software requerido y ejecute los procedimientos de generación de gráficos y videos, y un servidor web para desplegar dicha información.

El equipo que se encarga de procesar los archivos pcaps de la darknet, recibe dicho archivo en forma diaria a través de una conexión ssh con llave compartida. Una vez generados los datos relevantes, estos se transmiten con el mismo procedimiento al servidor web.

En el proyecto se utiliza el siguiente equipamiento, de propiedad del CLCERT:

1) Equipo de Proceso (procesamiento de datos y generación de gráficos/videos)

Desktop de escritorio, con SO Ubuntu 10.04, con el siguiente hardware:

- Procesador Pentium D 3.0GHz
- Memoria 1GB
- Disco 80GB

El servidor contiene todo el software relevante para la generación de gráficos, captura de videos y manipulación de archivos de captura (pcap). El equipo se encuentra en la red interna del CLCERT y en una oficina de acceso restringido.

2) Servidor Web

Desktop servidor, corriendo SO CentOS 5.5, con el siguiente hardware:

- Procesador Pentium D 3.0 GHz
- Memoria 2GB
- Disco 280 GB

El servidor ejecuta como servidor web la versión Apache 2.2, contiene el host virtual que atiende los requerimientos desde el exterior. Este servidor se aloja en el *data center* de la Universidad.

El esquema simple del proceso es el siguiente:



Figura 6: Esquema de proceso generación de gráficos y videos. La flecha indica una conexión segura entre las máquinas.

Definición de estadísticas relevantes

El proyecto requiere que los datos capturados sean presentados de forma clara, de manera de identificar posibles ataques en forma oportuna. La manera más clara de presentar dicha información es a través de gráficos. Para ello se busco obtener gráficos que permitieran al menos:

- Identificar ataques de denegación de servicios
- Reconocer escaneo de puertos
- Identificar posibles atacantes y sus destinos
- Obtener información sobre los protocolos utilizados para el envío de paquetes.

Se revisaron diversas aplicaciones que permitian generar los gráficos requeridos, siendo los más adecuados los programas Afterglow, TCPstat, Moncube y EtherApe, cuyo alcance se explica en el siguiente capitulo.

Opcionalmente, se consideró incluir en la arquitectura un equipo con un sistema detector de intrusos para así capturar información de gusanos UDP e ICMP (y para un análisis posterior). Sin embargo, por razones de eficiencia (del proceso y costos), tal alternativa será implementada en el futuro.

Implementación de gráficos

Para la generación de gráficos de tráfico en este proyecto se utilizaron principalmente cuatro herramientas: Afterglow, TCPstat, Moncube y EtherApe. Cada uno de estos programas genera gráficos automáticamente a partir de un archivo que contiene la descripción de los paquetes que tienen como destino la Darknet.

El archivo de tipo pcap con el tráfico diario recibido en la Darknet es almacenado en el servidor, y es descargado en forma automática al computador donde se generarán los gráficos de forma segura, siendo almacenados en una carpeta llamada "capturas". De esta manera, siempre se tiene el archivo del día anterior alojado en el computador encargado de graficar. Luego estos gráficos creados se envían al servidor web para luego desplegar la información en Internet. Otra tarea programada se encarga de descargar el archivo del día cada 5 minutos, lo cual permitirá tener una visión en línea de lo que ocurre en la Darknet, para así mostrar videos en la página web.

El siguiente diagrama muestra lo que sucede entre los servidores:

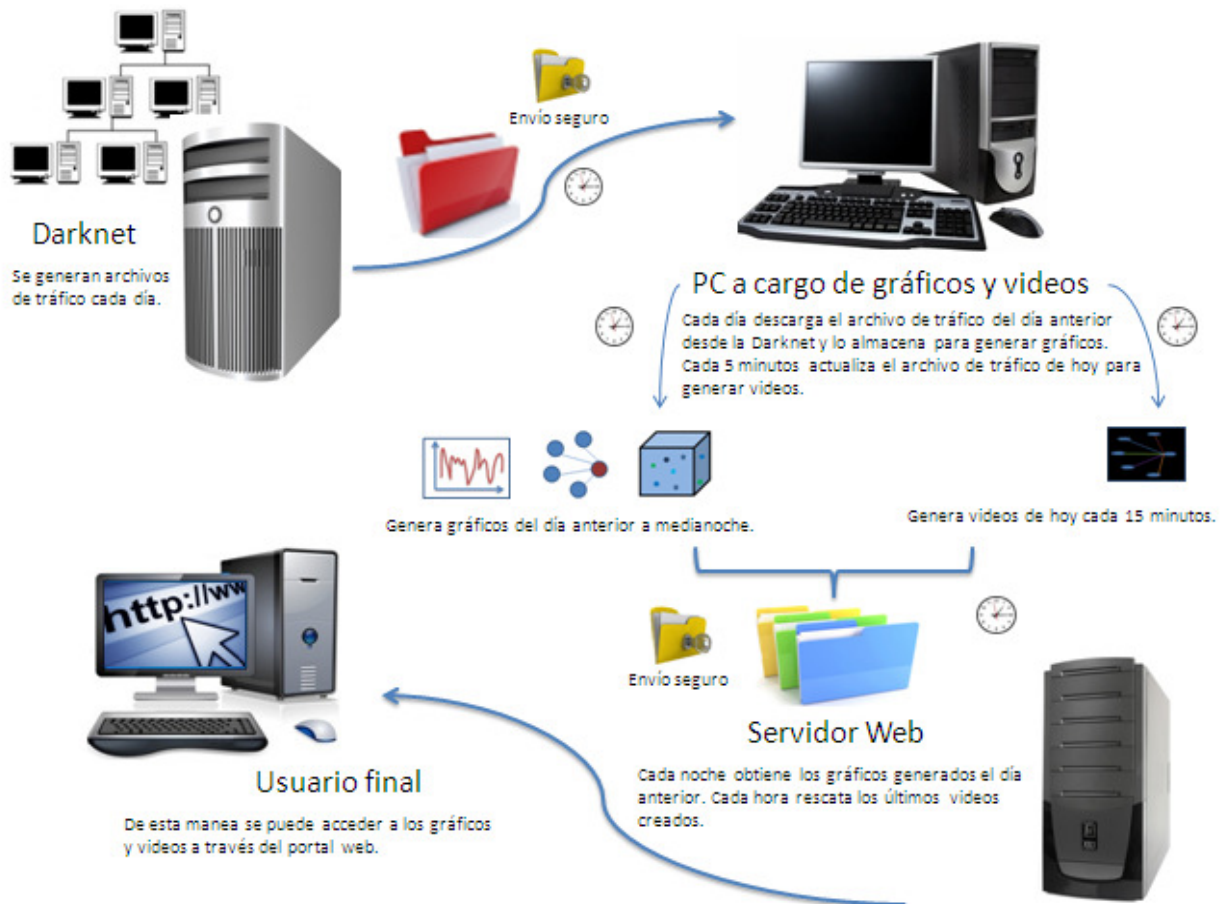


Figura 7: Representación del funcionamiento del sistema

Todas las herramientas utilizadas generan distintos gráficos a partir del archivo de tráfico del día anterior, a excepción de EtherApe cuya labor es generar un video simulando el tráfico existente en los últimos 10 minutos.

A continuación se explicarán cada una de las herramientas utilizadas y su funcionamiento.

a. AfterGlow

AfterGlow (<http://afterglow.sourceforge.net/>) es un software que realiza una malla con todas las conexiones existentes en un cierto período de tiempo. Para este proyecto se generan varias mallas por día. En cada media hora del día, se generan 3 gráficos: uno a los 10 minutos, otro a los 20 minutos y otro al final de la media hora. De esta manera se puede apreciar como va variando el estado de las conexiones existentes en la red.

Éstas son algunas de las imágenes obtenidas con AfterGlow:

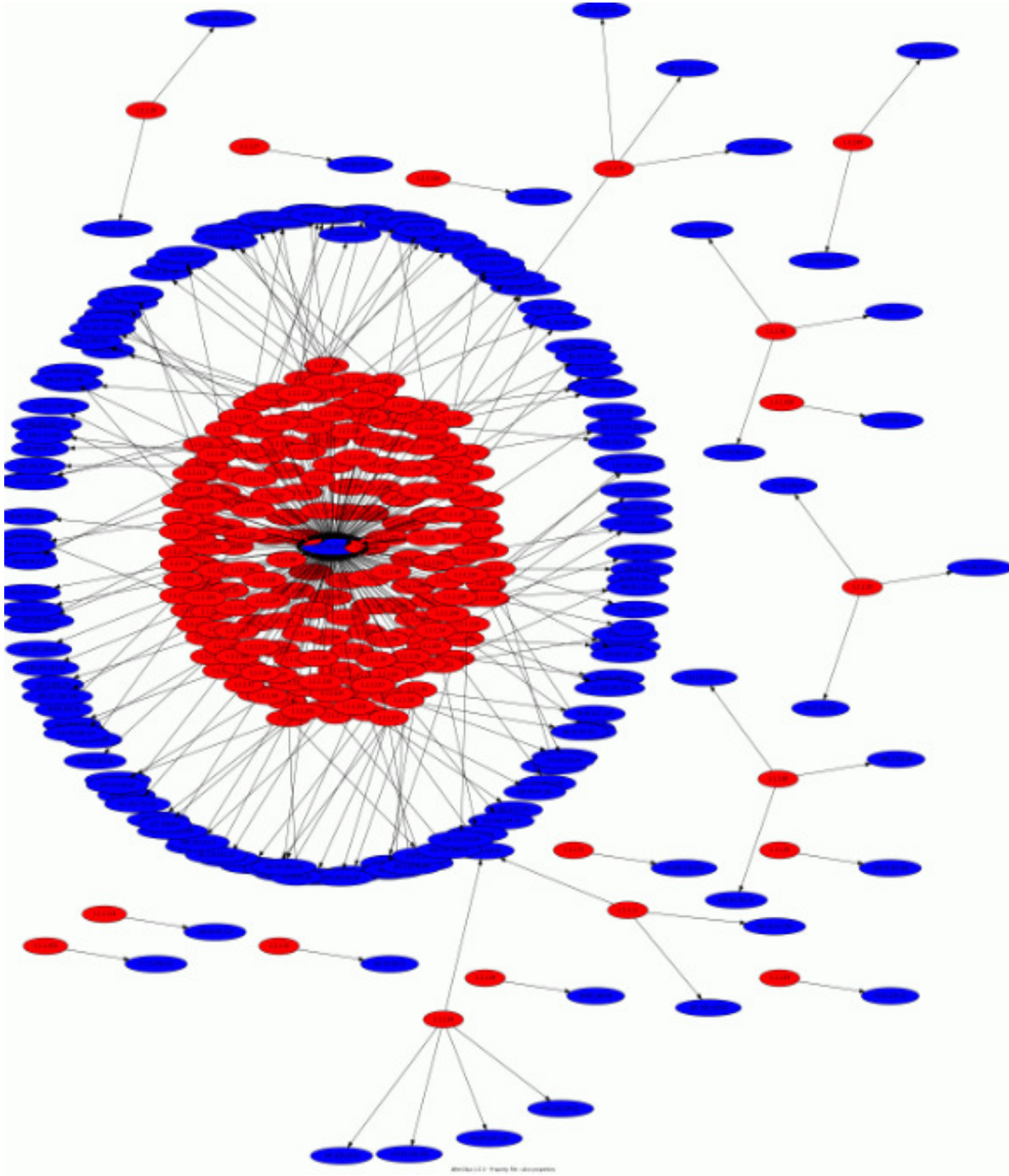


Figura 8: Tráfico de 00:00 a 00:10 horas de un día determinado de Diciembre 2010.

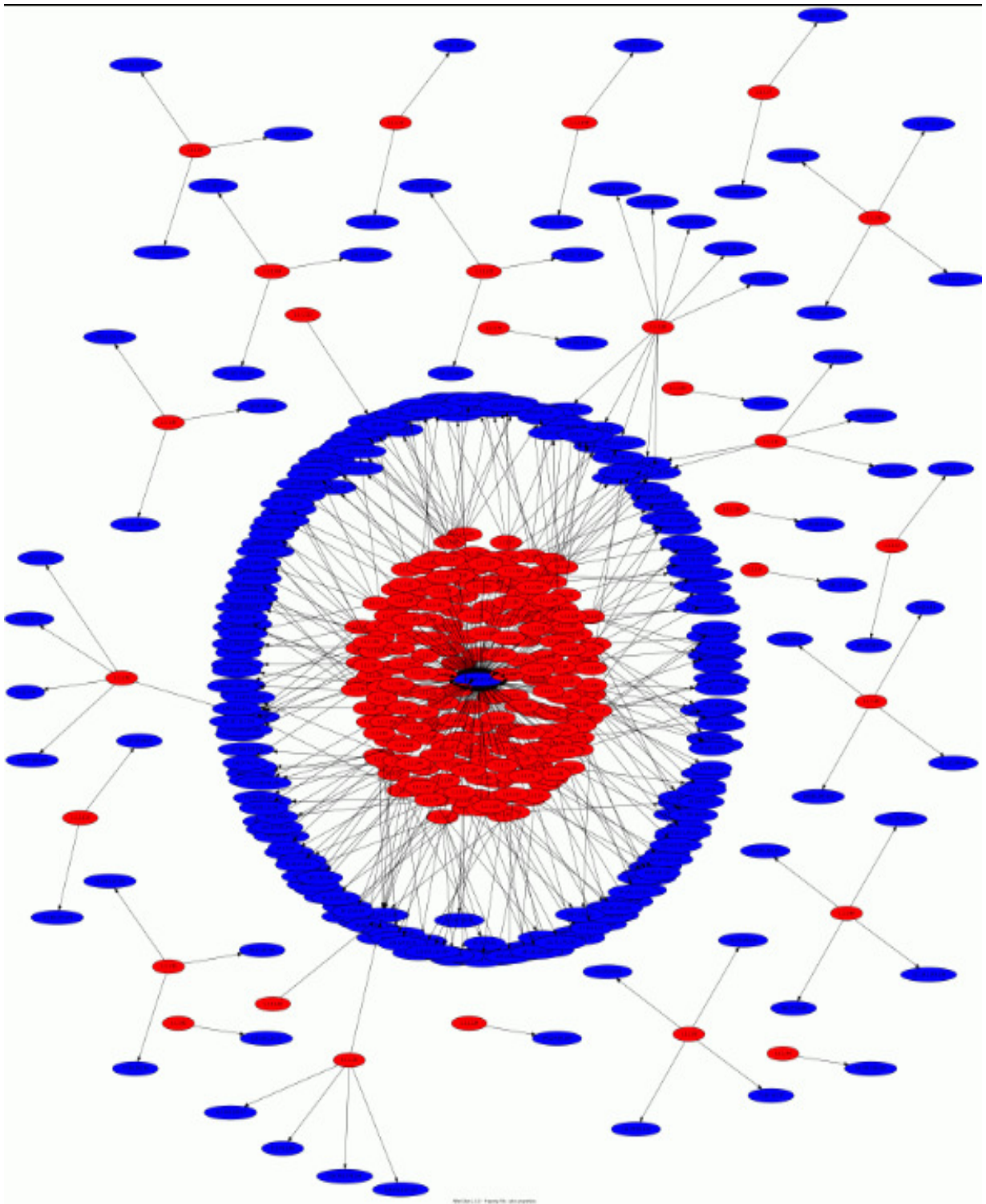


Figura 9: Tráfico de 00:00 a 00:20 horas de un determinado día de Diciembre 2010.

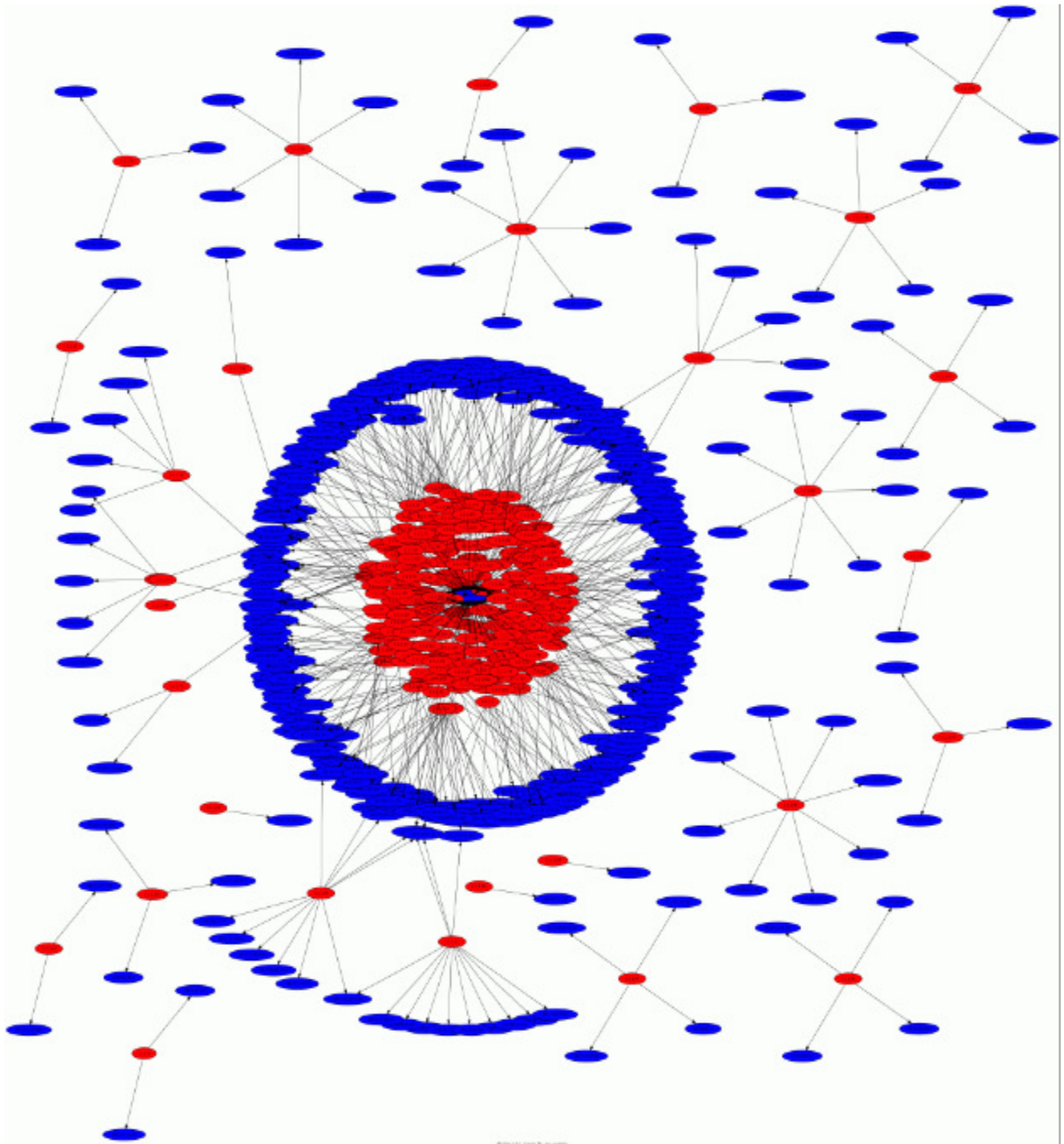


Figura 10: Tráfico de 00:00 a 00:30 hrs de un determinado día de Diciembre 2010.

Tal como se puede apreciar en las figuras 8, 9 y 10 , los nodos de color rojo son las IP de destino (en este caso, las direcciones IP de la Darknet) mientras que los nodos azules son las direcciones IP de origen, probablemente maliciosas.

Se implementó un script cuya función es calcular la fecha de ayer y de definir los cortes de tiempo necesarios para graficar, es decir, por cada media hora existente en el día se envía la orden de generar tres archivos pcap. A modo de ejemplo, si se llama "hh" a una determinada hora del día, se solicitará la generación de un archivo que

contiene el tráfico existente desde las hh:00 hasta las hh:10, otro desde las hh:00 hasta las hh:20 y otro desde las hh:00 hasta las hh:30 (ocurre lo mismo para la siguiente media hora, se generan archivos pcap desde las hh:30 hasta las hh:40, hasta las hh:50 y hasta las hh:59:59).

El código del script es el siguiente:

```
#Se calcula la fecha de ayer
#!/bin/bash
ayer=$(date --date='1 day ago')
echo "ayer era: " $ayer
year=$(date --date='1 day ago' +%Y)
month=$(date --date='1 day ago' +%m)
day=$(date --date='1 day ago' +%d)

#Se determina el nombre y ubicación del archivo de tráfico al que se #le deben hacer los cortes. Además se le hace una copia para trabajar #con él.
archivo="salida-"$year-"-$month"-"$day
salida=/home/capturas/$archivo
cp $salida /home/afterglow/src/perl/graph/$archivo

#Para cada hora del día se necesitan 6 archivos de tráfico, 3 por #cada media hora.
for H in "00" "01" "02" "03" "04" "05" "06" "07" "08" "09" "10" "11" "12" "13"
"14" "15" "16" "17" "18" "19" "20" "21" "22" "23";
do

    #Se llama al script encargado de graficar (explicado más adelante) #con el archivo de tráfico diario, y las horas en las que se quiere #hacer el corte.
    /home/afterglow/src/perl/graph/afterglow.sh $archivo ${H}h00m00s ${H}h10m00s
    /home/afterglow/src/perl/graph/afterglow.sh $archivo ${H}h00m00s ${H}h20m00s
    /home/afterglow/src/perl/graph/afterglow.sh $archivo ${H}h00m00s ${H}h30m00s
    /home/afterglow/src/perl/graph/afterglow.sh $archivo ${H}h30m00s ${H}h40m00s
    /home/afterglow/src/perl/graph/afterglow.sh $archivo ${H}h30m00s ${H}h50m00s
    /home/afterglow/src/perl/graph/afterglow.sh $archivo ${H}h30m00s ${H}h59m59s

done

#Finalmente se elimina la copia del archivo de tráfico
cd /home/afterglow/src/perl/graph
rm $archivo
```

El script que se encarga de generar los gráficos es el siguiente (afterglow.sh):

#calcula la fecha del día de ayer para utilizarlo al final para #ordenar las imágenes en carpetas diarias.

```
year=$(date --date='1 day ago' +%Y)
month=$(date --date='1 day ago' +%m)
day=$(date --date='1 day ago' +%d)
```

```
dia=$year"-"$month"-"$day
```

#ingresa al directorio y elimina todo posible gráfico existente que #tenga el mismo nombre del que se generará, para evitar copias.

```
cd /home/afterglow/src/perl/graph/
rm -rf graph_afterglow*
rm -rf tcpslice*
rm -rf grafica*
```

#tcpslice es el encargado de generar los pcap entre los tiempos #solicitados a partir del archivo de tráfico diario, el de ayer.

#Los parámetros que recibe son \$2: archivo de ayer, \$3: hora inicio #del corte, \$1: hora final del corte. Estos parámetros son definidos #tal como se explicó anteriormente por otro script. Luego genera el #archivo tcpslice_\$1_\$2_\$3.pcap que se utilizará para graficar.

```
/usr/sbin/tcpslice -w tcpslice_$1_$2_$3.pcap $2 $3 $1
```

#Para evitar la publicación de las IPs de la darknet se modifican #mediante una máscara, que se encarga de dejar solo el último octeto #de la IP sin alterar. Ahora salidaNUEVA.pcap tiene las direcciones #IP alteradas.

```
/usr/bin/tcprewrite --pnat=X.X.X.X/24:1.1.1.0/24 --infile=tcpslice_$1_$2_$3.pcap
--outfile=salidaNUEVA.pcap
```

#Se genera un archivo .dot a partir de las propiedades de color #asignadas y utilizando el software afterglow.pl considerando la IP #de destino y de origen.

```
/usr/bin/tshark -r salidaNUEVA.pcap -T fields -E separator=, -e ip.dst -e ip.src
| perl afterglow.pl -c color.properties -t > grafica_$1_$2_$3.dot
```

#Luego neato es el encargado de generar el gráfico como una imagen #.gif.

```
/usr/bin/neato -Tgif -o graph_afterglow_$1_$2_$3.GIF ./grafica_$1_$2_$3.dot
```

#Se crean una serie de directorios para ir almacenando los gráficos #por día de manera ordenada.

```
mkdir -p /home/AFTERGLOW/$dia/TCPSLICE/
mv tcpslice_$1_$2_$3.pcap /home/AFTERGLOW/$dia/TCPSLICE/tcpslice_$1_$2_$3.pcap
mkdir -p /home/AFTERGLOW/$dia/DOT/
mv grafica_$1_$2_$3.dot /home/AFTERGLOW/$dia/DOT/grafica_$1_$2_$3.dot
mkdir -p /home/AFTERGLOW/$dia/GRAHP/
```

```
mv graph_afterglow_$1_$2_$3.gif  
/home/AFTERGLOW/$dia/GRAHP/graph_afterglow_$1_$2_$3.gif
```

#Se cambia el formato de la imagen de GIF a PNG (y se borra el archivo con extensión GIF).

```
convert /home/AFTERGLOW/$dia/GRAHP/graph_afterglow_$1_$2_$3.GIF /home/  
/AFTERGLOW/$dia/GRAHP/graph_afterglow_$1_$2_$3.PNG  
rm /home/AFTERGLOW/$dia/GRAHP/graph_afterglow_$1_$2_$3.GIF
```

Mediante una tarea programada, Afterglow corre todos los días a las 00:05 de la madrugada, de manera que el archivo que contiene el tráfico del día de ayer ya se encuentre almacenado y disponible para su uso.

b. Moncube

Moncube (<http://www-moncube.cea.fr/doku.php/en:cube:cube>) es otro sistema capaz de generar gráficos pero simulando el tráfico dentro de un cubo que se puede mover, rotar, acercar y alejar entre otras propiedades. El eje rojo (eje x) se muestran las IPs de destino (darknet, simbolizadas por las IP de la forma 10.10.10.X), el eje azul (eje z) equivale a las IPs de origen, y el eje verde (eje y) corresponde a los puertos de destino. Para este proyecto se toman fotos a este cubo en distintas posiciones para visualizar de mejor manera lo que ocurre mediante imágenes.

Al igual que el software anterior, se toma el archivo de tráfico del día de ayer para generar las imágenes. Con este sistema se generan 3 imágenes diarias, todas son fotos del cubo en distintas posiciones.

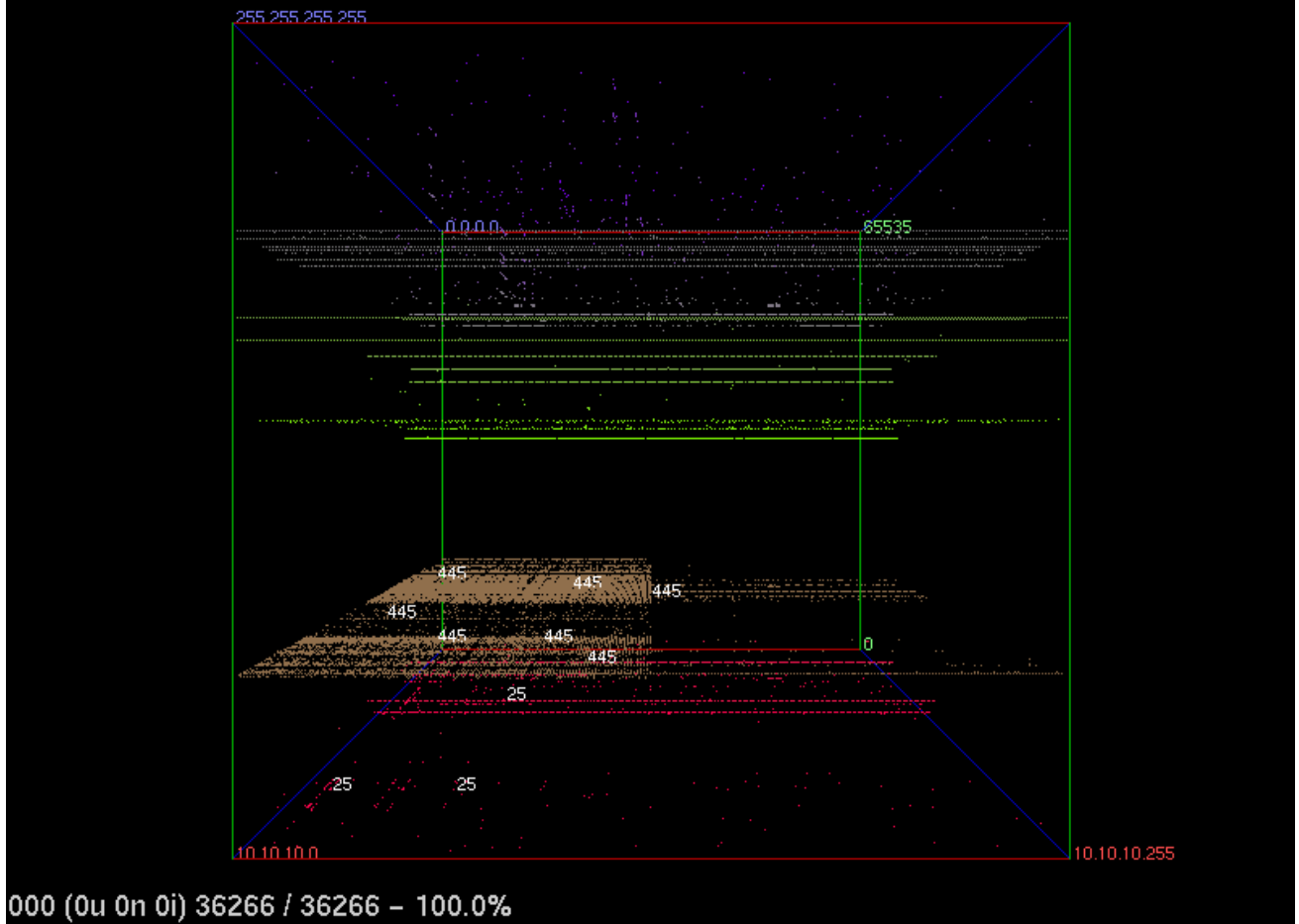


Figura 11: Cubo en primera posición.

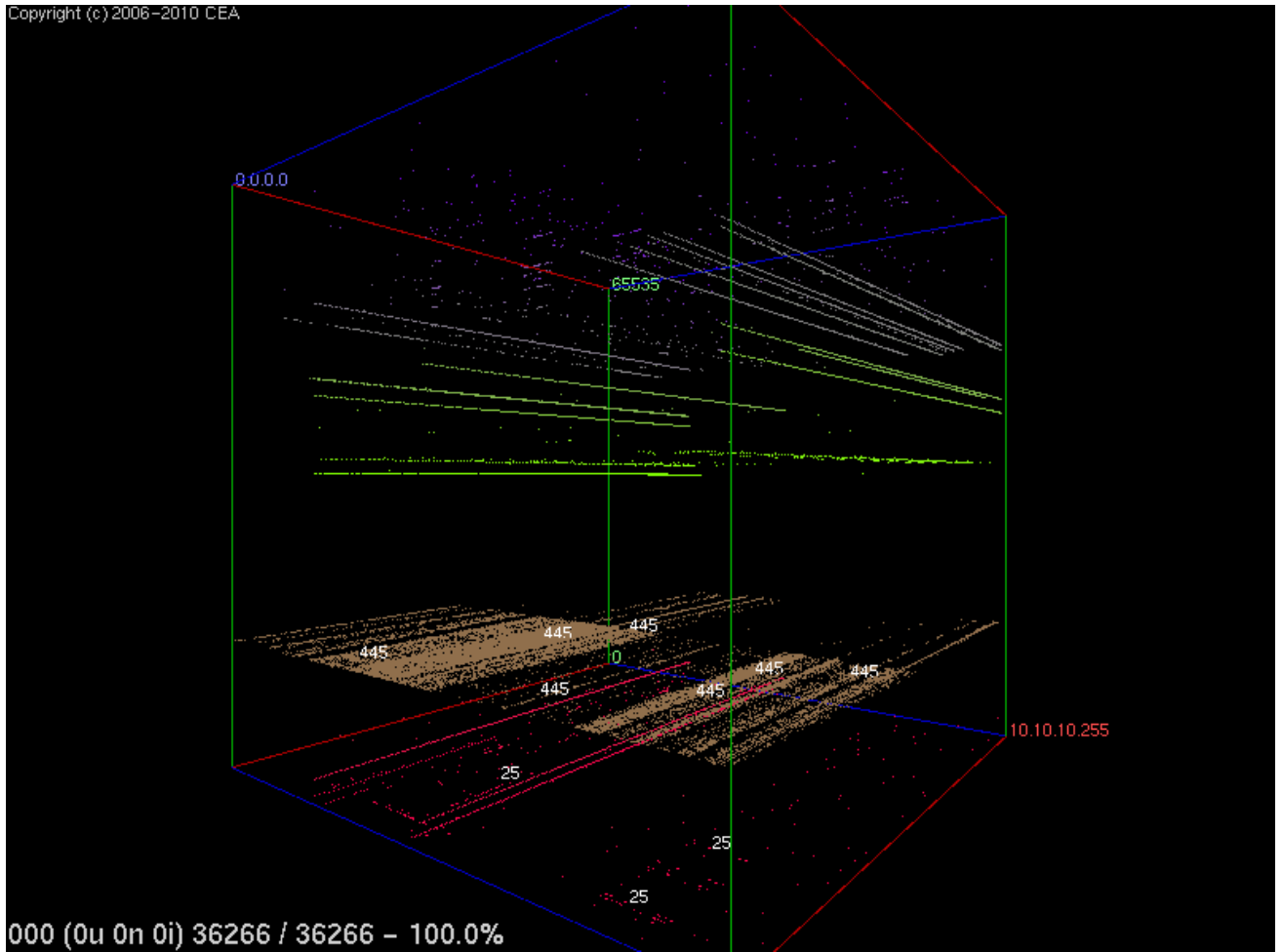


Figura 12: Cubo en segunda posición.

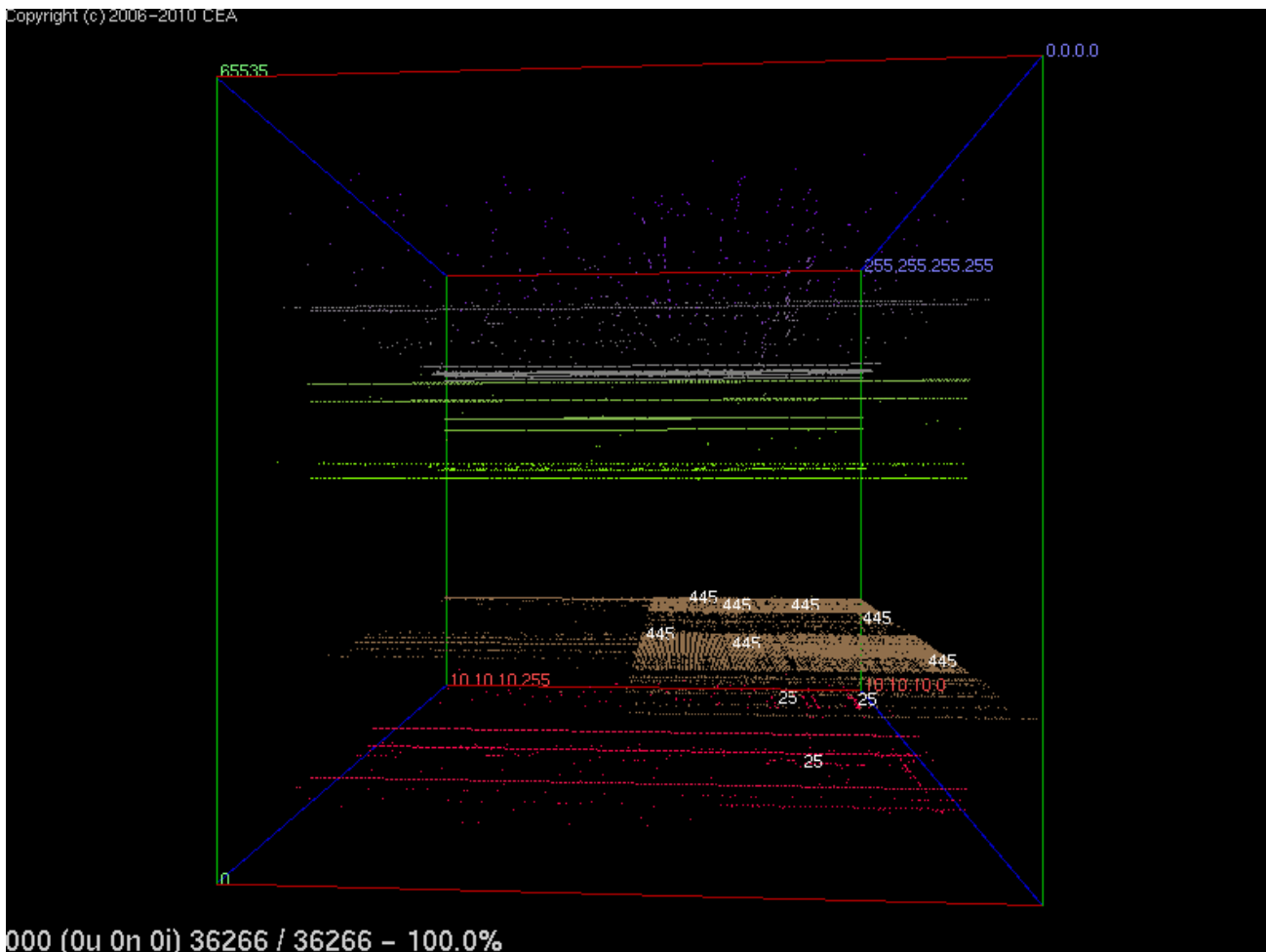


Figura 13: Cubo en tercera posición.

Para ejecutar la aplicación deben configurarse diversos parámetros tales como: el archivo de entrada a leer, el formato del archivo, el rango de IPs de destino (IPs de la Darknet), colores etc. Existe un script encargado de calcular el la fecha del día de ayer y de solicitar a otro script la generación de las imágenes.

A continuación el código del primer script:

#Se calcula la fecha de ayer y el nombre del archivo de tráfico.

```
#!/bin/bash
ayer=$(date --date='1 day ago')
year=$(date --date='1 day ago' +%Y)
month=$(date --date='1 day ago' +%m)
day=$(date --date='1 day ago' +%d)

archivo="salida-"$year-"-$month"-"$day
salida=$archivo
```

#Se llama al segundo script encargado de la ejecución de Moncube con #el archivo de tráfico del día de ayer como entrada.

```
/home/moncube-1.2.3/bin/moncube.sh /home/capturas/$salida
```

A continuación se presenta el script encargado de hacer la mayor parte del trabajo, ejecución de Moncube y captura de imágenes:

#calcula la fecha del día de ayer para utilizarlo al final para #ordenar las imágenes en carpetas diarias.

```
year=$(date --date='1 day ago' +%Y)
```

```
month=$(date --date='1 day ago' +%m)
```

```
day=$(date --date='1 day ago' +%d)
```

```
dia=$year-"-$month"-"$day
```

#Función encargada de capturar imágenes del cubo mediante la tecla #“w”, definida por Moncube para sacar fotos. También se usa la tecla #“r” para iniciar y terminar la rotación del cubo.

```
waitNow(){
```

#pasaron 5 segundos (tiempo necesario para que Moncube alcance a leer el archivo de tráfico)

```
sleep 5s
```

#se saca primera foto

```
xdotool key w
```

#empieza a rotar el cubo

```
xdotool key r
```

#pasaron 3 segundos más

```
sleep 3s
```

#se saca segunda foto

```
xdotool key w
```

#pasaron 7 segundos más

```
sleep 7s
```

#se saca la última foto

```
xdotool key w
```

#se deja de hacer rotar el cubo

```
xdotool key r
```

#se esperan 2 segundos más para que la última foto alcance a #guardarse y se cierra el programa.

```
sleep 2s
```

```
killall parsecube
```

```
}
```

#Se ingresa al directorio donde está el ejecutable y se borran imágenes o archivos con nombres que podrían crear copias.

```
cd /home
```

```
rm salida*
```

```
rm -rf cubeshot-*
```

```
rm moncube_A_$dia.bmp
rm moncube_B_$dia.bmp
rm moncube_C_$dia.bmp
```

#Se cambia la IP de la Darknet para mantenerla en secreto (IP de #destino) y se define como 10.10.10.0. Para esto se usa la #herramienta tcperewrite a #partir del archivo de tráfico de ayer, #dado como parámetro \$1.

```
/usr/bin/tcprewrite --pnat=xxx.xx.xx.x/x:10.10.10.0/24 --infile=$1 --
outfile=salidaMON
```

#A partir del archivo generado en la línea anterior, salidaMON, se #genera “salida” que contiene el formato que Moncube necesita para #poder graficar. Esto se hace mediante tcpdump.

```
/usr/sbin/tcpdump -r salidaMON -nqt > salida
```

#Se llama al ejecutable “parsecube” pero también llamando a la #función waitNow para que corra al mismo tiempo y capture las #imágenes.

```
waitNow |./parsecube
```

#Se crea los directorios necesarios para tener las imágenes bien #ordenadas junto con el archivo “salida”, contenedor del tráfico en #el formato especificado.

```
mkdir -p ~/Desktop/MONCUBE/$dia/
mv cubeshot-001.bmp /home/MONCUBE/$dia/moncube_A_$dia.bmp
mv cubeshot-002.bmp /home/MONCUBE/$dia/moncube_B_$dia.bmp
mv cubeshot-003.bmp /home/MONCUBE/$dia/moncube_C_$dia.bmp
mv salida /Desktop/MONCUBE/$dia
```

Moncube se corre a las 00:01 para que el archivo del tráfico de ayer se encuentre disponible. Este es el primer software en correr, y trabaja solo ya que utiliza más recursos que otros de los utilizados.

Lo interesante de este software es que a través de las imágenes se pueden observar ciertos patrones que pueden indicar ciertos ataques o comportamientos peligrosos. Los desarrolladores de Moncube hicieron una lista de imágenes y sus posibles significados para contribuir en el estudio de la seguridad, esta información se encuentra en el sitio especificado anteriormente bajo el link llamado “Scan patterns”.

c. TCPstat

TCPstat (<http://www.frenchfries.net/paul/tcpstat/>) es una herramienta que permite la generación de gráficos de dos dimensiones. Permite graficar cuántos paquetes por segundo llegan a la red durante un determinado tiempo. A partir de esto se generan dos tipos de gráficos, uno donde se indican los protocolos de los paquetes, y otro

donde se grafican 2 curvas: una que representa a los paquetes con cualquier flag activada y otra con solamente los paquetes cuya flag SYN-ACK está activada, en caso de que existan, ya que la llegada de este tipo de paquetes es una señal de un ataque de esa índole. Esto es porque un atacante puede enviar una solicitud (envío de paquete SYN) a un determinado equipo y poniendo como dirección de origen alguna IP perteneciente a la darknet. De esta manera el equipo vulnerado le responde a la darknet con paquetes SYN-ACK. Dado que la Darknet nunca intentó comunicarse con alguien existe la sospecha de que existen computadores siendo vulnerados mediante un ataque de denegación de servicios (envío múltiple de paquetes esperando respuesta para que se le acaben los recursos al atacado).

Estos son algunos ejemplos de los gráficos que se pueden obtener:

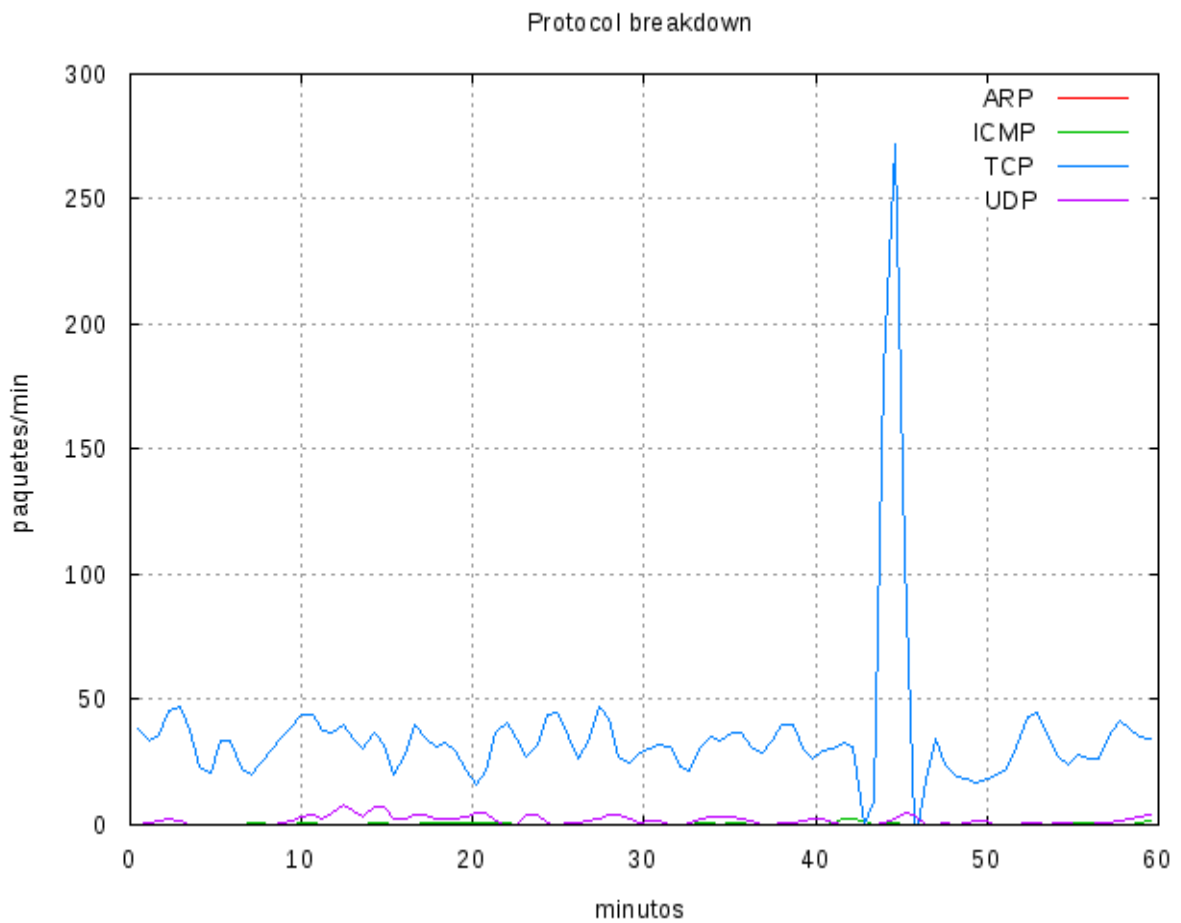


Figura 14: Paquetes por protocolo durante una hora.

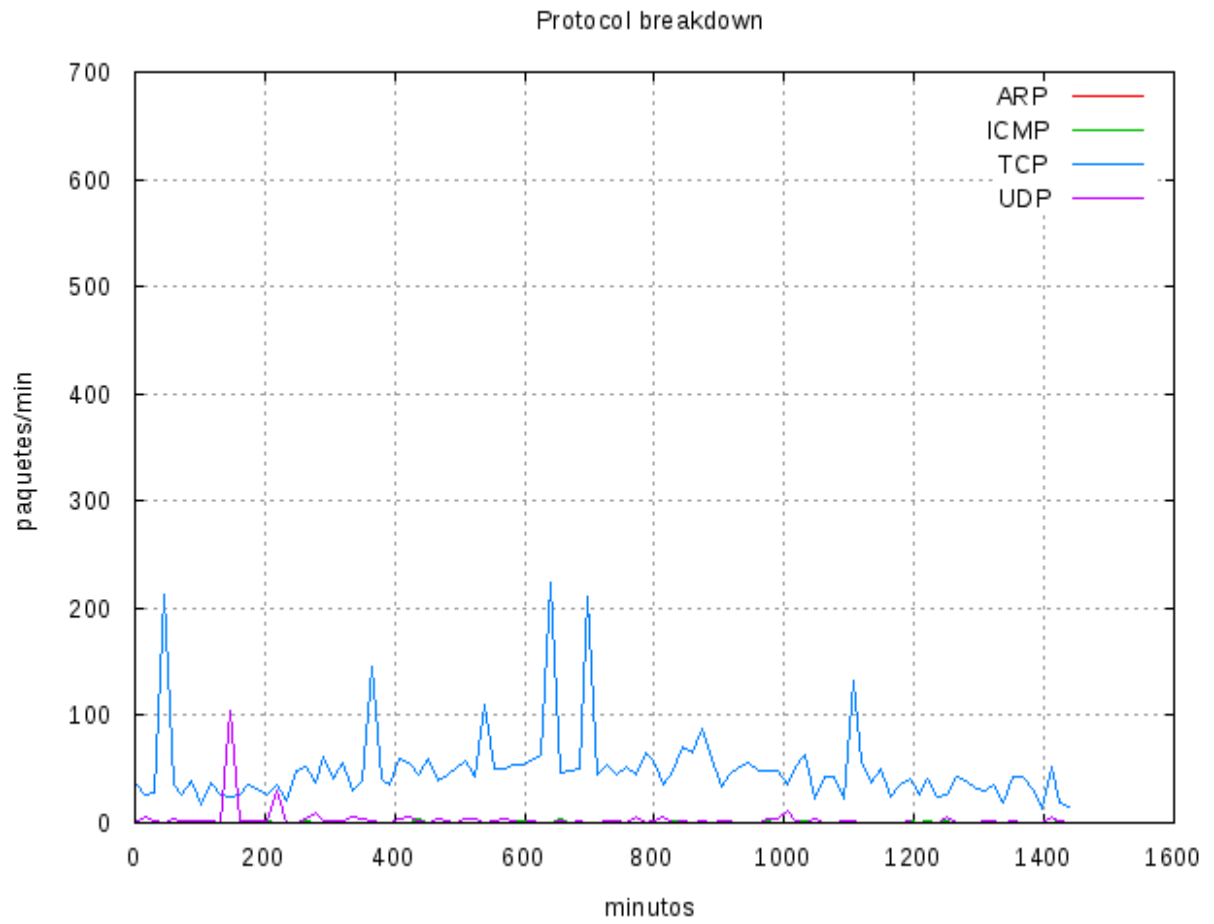


Figura 15: Paquetes por protocolo durante un día.

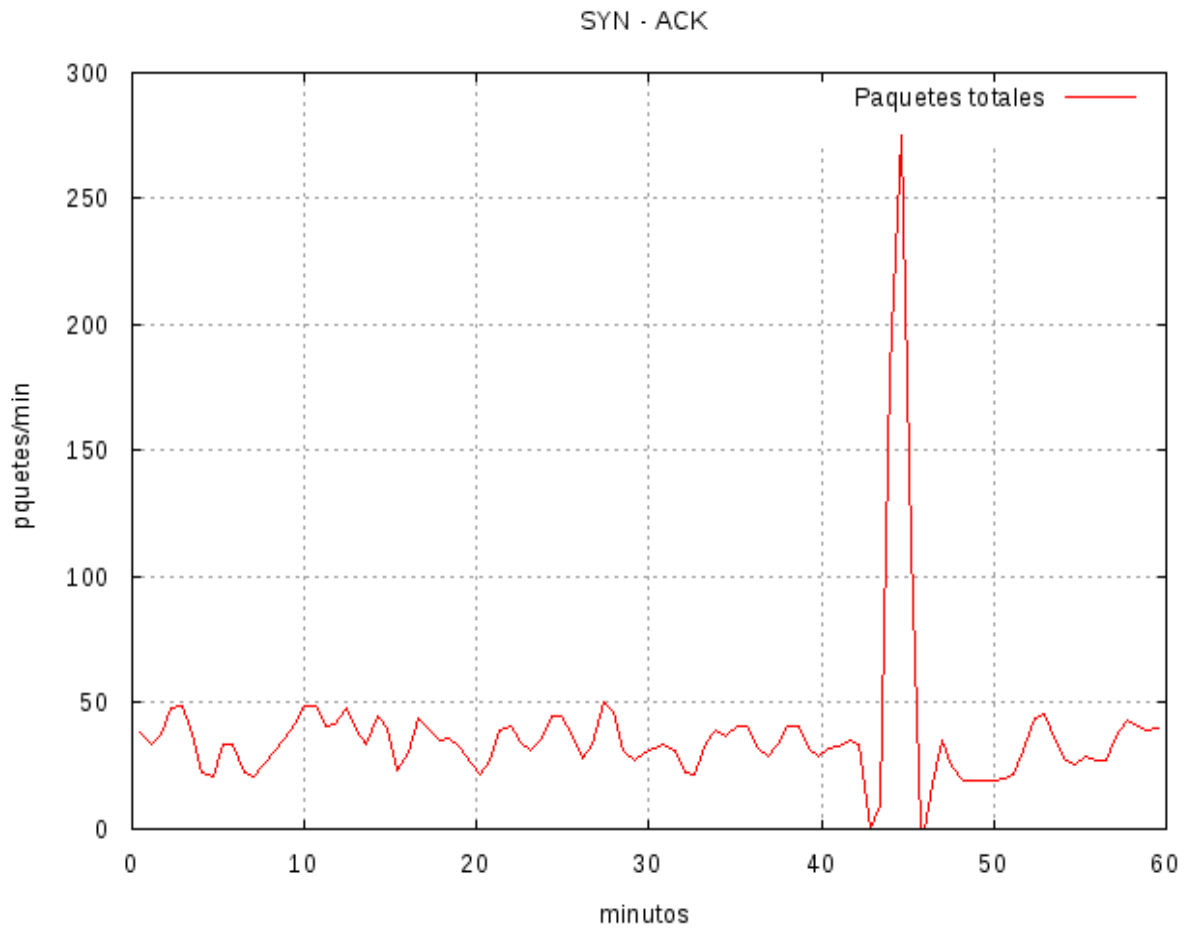


Figura 16: Paquetes totales durante una hora (no se observan paquetes con flag TCP-SYN activada).



Figura 17: Paquetes totales durante un día (no se observan paquetes con flag TCP-SYN activada).

Para la generación del gráfico de paquetes según protocolo se realizó un script que se encarga de cortar el archivo de tráfico del día de ayer en pequeños archivos que contienen el tráfico de una hora del día cada uno. Luego se llama a otro script encargado de graficar (tanto lo que ocurre cada hora como lo que ocurre en el día en su totalidad).

El script encargado de los cortes del archivo de tráfico es el siguiente:

```
# Se calcula la fecha de ayer
#!/bin/bash
ayer=$(date --date='1 day ago')
echo "ayer era: " $ayer
year=$(date --date='1 day ago' +%Y)
month=$(date --date='1 day ago' +%m)
day=$(date --date='1 day ago' +%d)
```

#Se conoce el nombre del archivo de tráfico y se copia en el #directorio de tcpstat para trabajarlo.

```
archivo="salida-"$year-"$month-"$day
salida=/home/capturas/$archivo
cp $salida /home/tcpstat-1.5/$archivo
```

#Para cada hora del día se crean un archivo de tráfico que contiene #solamente los paquetes de esa hora.

```
cd /home/tcpstat-1.5
for H in "00" "01" "02" "03" "04" "05" "06" "07" "08" "09" "10" "11" "12" "13"
"14" "15" "16" "17" "18" "19" "20" "21" "22" "23";
do
/usr/sbin/tcpslice -w ${H}h00m00s_${H}h59m59s.pcap ${H}h00m00s ${H}h59m59s
$archivo
```

#Luego se llama al siguiente script que se encargará de graficar #utilizando como entrada el tráfico de la hora asignada.

```
/home/tcpstat-1.5/TCPstats_protocol.sh ${H}h00m00s_${H}h59m59s.pcap
done
```

#Además se le entrega al script encargado de graficar una copia de #todo el tráfico del día anterior para generar un gráfico más #general.

```
/home/tcpstat-1.5/TCPstats_protocol.sh $archivo
```

A continuación se presenta el script encargado de la generación de gráficos utilizando la herramienta TCPstat:

#Se calcula la fecha del día de ayer

```
year=$(date --date='1 day ago' +%Y)
month=$(date --date='1 day ago' +%m)
day=$(date --date='1 day ago' +%d)
```

```
dia=$year-"$month-"$day
```

#Se ingresa al directorio adecuado y se eliminan archivos que #contengan nombres que podrían generar copias.

```
cd /home/tcpstat-1.5/
rm -rf graph_protocol_*
rm -rf arp.data
rm -rf icmp.data
rm -rf tcp.data
rm -rf udp.data
```

#Se llama a tcpstat con el archivo de tráfico adecuado (ya sea de #una hora o diario) y se filtran los paquetes por protocolo, sacando #un promedio de paquetes vistos en los últimos 60 segundos para cada #protocolo.

```
/usr/bin/tcpstat -r $1 -o "%R\t%A\n" 60 > arp.data
/usr/bin/tcpstat -r $1 -o "%R\t%C\n" 60 > icmp.data
```

```
/usr/bin/tcpstat -r $1 -o "%R\t%T\n" 60 > tcp.data
/usr/bin/tcpstat -r $1 -o "%R\t%U\n" 60 > udp.data
/usr/bin/gnuplot
```

#Luego se llama a gnuplot.script que se explicará a continuación #para generar el gráfico final a partir de los archivos .data creados #en el paso anterior. Se generan imágenes .png

```
/home/tcpstat-1.5/gnuplot.script > graph_protocol_$1.png
```

#Se guardan los archivos usados y creados por día en sus respectivas #carpetas.

```
mkdir -p /home/TCP_PROTOCOL/$dia/PNG/
mv graph_protocol_* /home/TCP_PROTOCOL/$dia/PNG
mkdir -p /home/TCP_PROTOCOL/$dia/DATA/
mv arp.data /home/TCP_PROTOCOL/$dia/DATA
mv icmp.data /home/TCP_PROTOCOL/$dia/DATA
mv tcp.data /home/TCP_PROTOCOL/$dia/DATA
mv udp.data /home/TCP_PROTOCOL/$dia/DATA
mkdir -p /home/TCP_PROTOCOL/$dia/PCAP/
mv *.pcap /home/TCP_PROTOCOL/$dia/PCAP
cp salida* /home/TCP_PROTOCOL/$dia/PCAP
```

El siguiente script importante es el `gnuplot.script`, que utilizando el programa `gnuplot` (<http://www.gnuplot.info>), grafica y pone etiquetas al gráfico. Este script define el título del gráfico, el nombre del eje X e Y, entre otras cosas.

Utilizando `"($1/60):2"` se logra controlar las unidades en que se quieren desplegar los datos. El `"$1/60"` significa que el primer parámetro, llamado `"$1"`, que es el tiempo (eje X) esté dividido en 60. Por defecto el tiempo está en segundos, por lo que al dividir por 60 se tendrá el tiempo en minutos. Luego el número `"2"` sin ninguna operación aplicada, como fue la división en el caso anterior, implica que en el eje Y (segundo parámetro) se tendrán los paquetes promedio vistos en el último minuto, valor por defecto que no se desea alterar para manejar todo en "minutos".

```
set term png small #FFFFFF
  set data style lines
  set grid
  set yrange [ 0 : ]
  set title "Protocol breakdown"
  set xlabel "minutes"
  set ylabel "packets/min"
  plot "arp.data" using ($1/60):2 smooth csplines title "ARP" \
    ,"icmp.data" using ($1/60):2 smooth csplines title "ICMP" \
    ,"tcp.data" using ($1/60):2 smooth csplines title "TCP" \
    ,"udp.data" using ($1/60):2 smooth csplines title "UDP"
```

Para generar gráficos a partir de paquetes que tengan la flag `"SYN ACK"` prendida se utilizan los siguientes scripts (muy similares a los anteriores).

El siguiente es el script encargado de los cortes:

```
# Se calcula la fecha de ayer
#!/bin/bash
ayer=$(date --date='1 day ago')
echo "ayer era: " $ayer
year=$(date --date='1 day ago' +%Y)
month=$(date --date='1 day ago' +%m)
day=$(date --date='1 day ago' +%d)

#Se conoce el nombre del archivo de tráfico y se copia en el #directorío de
tcpstat para trabajarlo.
archivo="salida-"$year-"-$month"-"$day
salida=/home/capturas/$archivo
cp $salida /home/tcpstat-1.5/$archivo

#Para cada hora del día se crean un archivo de tráfico que contiene #solamente
los paquetes de esa hora.
cd /home/tcpstat-1.5
for H in "00" "01" "02" "03" "04" "05" "06" "07" "08" "09" "10" "11" "12" "13"
"14" "15" "16" "17" "18" "19" "20" "21" "22" "23";
do
    /usr/sbin/tcpslice -w ${H}h00m00s_${H}h59m59s.pcap ${H}h00m00s ${H}h59m59s
    $archivo

#Luego se llama al siguiente script que se encargará de graficar #utilizando
como entrada el tráfico de la hora asignada. Nótese que el siguiente script
utilizado en este caso es TCPstats_SYNACK.sh que #tendrá una labor parecida a la
de TCPstats_protocol.sh pero no #equivalente.
    /home/tcpstat-1.5/TCPstats_SYNACK.sh ${H}h00m00s_${H}h59m59s.pcap
done

#Además se le entrega al script encargado de graficar una copia de #todo el
tráfico del día anterior para generar un gráfico más #general.
/home/tcpstat-1.5/TCPstats_SYNACK.sh $archivo
```

El siguiente script es el encargado de la generación de gráficos en sí:

```
#!/bin/bash
# ./TCPstats_SYNACK.sh salida.pcap

#Se calcula la fecha del día de ayer
year=$(date --date='1 day ago' +%Y)
month=$(date --date='1 day ago' +%m)
day=$(date --date='1 day ago' +%d)

dia=$year-"-$month"-"$day
```

#Se ingresa al directorio adecuado y se eliminan archivos que #contengan nombres que podrían generar copias.

```
cd /home/tcpstat-1.5/  
rm -rf graph_synack_*  
rm -rf total.data  
rm -rf synack.data
```

#En este caso se grafican 2 curvas, una que hace referencia a todos #los paquetes y otra que tiene solo aquellos paquetes con la flag #“SYN-ACK” prendida. Para esto se aplica un filtro.

```
/usr/bin/tcpstat -r $1 -o "%R\t%n\n" 60 > total.data
```

```
/usr/sbin/tcpdump -r $1 'tcp[13] = 18' -w synack.pcap  
/usr/bin/tcpstat -r synack.pcap -o "%R\t%n\n" 60 > synack.data
```

```
/usr/bin/gnuplot /home/tcpstat-1.5/gnuplotACK.script > graph_synack_${1}.png
```

#Se genera el .data a partir del archivo generado en la línea #anterior. Posteriormente se grafica.

```
/usr/bin/tcpstat -r archivo_nuevo.pcap -o "%R\t%T\n" 60 > tcp.data  
/usr/bin/gnuplot /home/tcpstat-1.5/gnuplotACK.script > graph_synack_${1}.png
```

#Se guardan los archivos usados y creados por día en sus respectivas #carpetas.

```
mkdir -p /home/TCP_SYNACK/$dia/PNG/  
mv graph_synack* /home/TCP_SYNACK/$dia/PNG  
mkdir -p /home/TCP_SYNACK/$dia/DATA/  
mv total.data /home/TCP_SYNACK/$dia/DATA  
mv synack.data /home/TCP_SYNACK/$dia/DATA  
mkdir -p /home/TCP_SYNACK/$dia/PCAP  
mv *.pcap /home/TCP_SYNACK/$dia/PCAP  
cp salida* /home/TCP_SYNACK/$dia/PCAP
```

Finalmente el script gnuplot SYNACK.script, que se encarga de graficar y poner etiquetas al gráfico. Es muy similar a gnuplot.script solo que ahora se grafican todos los paquetes en una curva, y los SYN-ACK en otra (en caso de que existan, si no hay la curva no aparecerá.)

```
set term png small #FFFFFF  
    set data style lines  
    set grid  
    set yrange [ 0 : ]  
    set title "SYN - ACK"  
    set xlabel "minutes"  
    set ylabel "packets/min"
```

```

    plot "total.data" using ($1/60):2 smooth csplines title "Paquetes
    totales" \
    , "synack.data" using ($1/60):2 smooth csplines title "Paquetes SYN-
    ACK"

```

Tanto los scripts encargados de generar gráficos por protocolo de paquetes en tiempo, como los encargados de graficar paquetes SYN-ACK en el tiempo, corren a las 00:02 utilizando la información del tráfico del día anterior.

d. Etherape

Este software (<http://etherape.sourceforge.net/>) permite generar un video tomando como entrada un archivo pcap de tráfico. EtherApe crea una simulación "en tiempo real" de las conexiones existentes entre el exterior y la Darknet.

Para esto se lanza EtherApe para que realice la simulación mediante un video, por otro lado se filma el sector de la pantalla donde el software está corriendo. De esta manera se logra capturar un video de la simulación.

Este software corre cada 15 minutos y crea un video de los últimos 10 minutos transcurridos, con un desfase de 5 minutos (tiempo dado para la correcta creación del video). Existe un script encargado de crear cortes en el archivo de tráfico del día actual, y generar pequeños pcap con la información de lo ocurrido en los 10 minutos mencionados:

```

#!/bin/bash

#Cálculo de la fecha actual
year=$(date +%Y)
month=$(date +%m)
day=$(date +%d)

#Se copia el archivo de tráfico en el directorio de EtherApe
archivo="salida-"$year-"-$month"-"$day

salida=/home/capturas/$archivo
cp $salida /home/etherape-0.9.9/$archivo

#Se define la hora de inicio y de fin de cada video. La de inicio #será la de
#hace 15 minutos atrás, y la de fin será la de hace 5 #minutos atrás. De esta
#manera se obtendrán simulaciones de 10 #minutos de duración. Recordar que este
#script se lanza cada 15 #minutos.
hora_inicio=$(date --date='15 minutes ago' +%H)
minuto_inicio=$(date --date='15 minutes ago' +%M)
hora_fin=$(date --date='5 minutes ago' +%H)

```

```
minuto_fin=$(date --date='5 minutes ago' +%M)
```

#Luego se realizan los cortes para tener pequeños archivos de tráfico #que contienen lo ocurrido en los 10 minutos de tráfico definidos.

```
/usr/sbin/tcpslice -w  
{hora_inicio}h${minuto_inicio}m00s_{hora_fin}h${minuto_fin}m00s.pcap  
{hora_inicio}h${minuto_inicio}m00s {hora_fin}h${minuto_fin}m00s  
/home/etherape-0.9.9/$archivo
```

#Se mueve el archivo generado al directorio de EtherApe.

```
mv {hora_inicio}h${minuto_inicio}m00s_{hora_fin}h${minuto_fin}m00s.pcap  
/home/etherape-0.9.9
```

#Se llama al siguiente script encargado de lanzar EtherApe.

```
/home/etherape-0.9.9/ethMin.sh  
{hora_inicio}h${minuto_inicio}m00s_{hora_fin}h${minuto_fin}m00s.pcap
```

El script que hace correr EtherApe y filmar el video es el siguiente:

```
#! /bin/bash
```

#Cálculo del día actual

```
year=$(date +%Y)  
month=$(date +%m)  
day=$(date +%d)
```

```
dia=$year"-"$month"-"$day
```

#Se desea capturar filmar la ventana de Etherape, entonces en caso #de que éste se cierre, no se desea que el script siga capturando #video al fondo del Escritorio, es por esto que se crea esta #variable que luego es utilizada como condición en la toma de #video.

```
CUENTA=`ps -efa | grep '[/]etherape' > /dev/null`
```

#Función encargada de filmar cuando EtherApe está corriendo. Además #se cierran los 2 programas utilizados.

```
waitNow(){  
ffmpeg -f x11grab -r 25 -s 530x328 -i :0.0+0,25 -t 600 /home/etherape-  
0.9.9/video.mpeg  
killall -9 ffmpeg  
killall -9 etherape  
}
```

#Se mueve el archivo de tráfico al directorio necesario, se entra #en él y se eliminan archivos que porían crear duplicidades.

```
mv $1 /home/etherape-0.9.9  
cd /home/etherape-0.9.9
```

```
rm video*
rm salidaDESTINO*
```

#Se crea un nuevo archivo dándole a todos los paquetes la misma #dirección de destino de la darknet, simulada como 10.10.10..10.1 #en este caso. Esto se hace a partir del archivo de entrada que #contiene tráfico de los últimos 10 minutos (con un desfase de 5 #minutos).

```
/usr/bin/tcprewrite --pnat= X.X.X.X/X:10.10.10.1 --infile=$1 --
outfile=salidaDESTINO
```

#Etherape corre junto con el método encargado de la captura de video.

```
waitNow | etherape -r salidaDESTINO
```

#Se crean los distintos directorios para organizar las imágenes, #videos y #archivos de tráfico.

```
mkdir -p /home/ETHERAPEminuto/$dia/VIDEO/
mv video* /home/ETHERAPEminuto/$dia/VIDEO/video_{$1}.mpeg
mkdir -p /home/ETHERAPEminuto/$dia/PCAP/
mv salida* /home/ETHERAPEminuto/$dia/PCAP
mv *.pcap /home/ETHERAPEminuto/$dia/PCAP
```

Este sistema corre todos los días tal como los demás, cada 15 minutos para así obtener una visión "on line" de lo que está ocurriendo en la Darknet.

Para el lanzamiento automático de todas estas tareas se utiliza "Configured Scheduled Tasks" de Gnome, que funciona como "cron" pero con la posibilidad de ejecutar comandos que necesiten interfaz gráfica, tal como es el caso de Etherape y Moncube.

Copia y Borrado de material

Cada día se copian las imágenes y los videos desde el equipo de proceso al servidor web para el despliegue de los gráficos en la página web. Esto se realiza mediante los siguientes scripts que transfieren datos en forma segura (via scp), con mecanismos de claves compartidas.

Para copiar las imágenes:

```
#!/bin/bash
#Cálculo del día anterior
year=$(date --date='1 day ago' +%Y)
month=$(date --date='1 day ago' +%m)
day=$(date --date='1 day ago' +%d)

dia=$year-"-$month"-"$day
```

#Se crea un directorio para ordenar las imágenes

```
mkdir -p /var/www/html/darknet/graficos/
```

#Se copian las imágenes desde el origen hasta destino organizando #todo con carpetas nombradas igual que el software utilizado para los #gráficos en cada caso.

```
scp -r usuario@host-proceso:/home/AFTERGLOW/$dia/GRAHP/
/var/www/html/darknet/graficos/AFTERGLOW/$dia
scp -r usuario@host-proceso:/home/MONCUBE/$dia/
/var/www/html/darknet/graficos/MONCUBE
scp -r usuario@host-proceso:/home/TCP_PROTOCOL/$dia/PNG/
/var/www/html/darknet/graficos/TCP_PROTOCOL/$dia
scp -r usuario@host-proceso:/home/TCP_SYNACK/$dia/PNG/
/var/www/html/darknet/graficos/TCP_SYNACK/$dia
```

Este script se ejecuta al inicio de cada día, una vez que las imágenes hayan sido generadas en el equipo encargado de hacerlo.

Luego para copiar los videos:

```
#!/bin/bash
#Cálculo de la fecha actual
year=$(date +%Y)
month=$(date +%m)
day=$(date +%d)

#Cálculo del día actual
dia=$year"-"$month"-"$day

#Se crea un directorio para ordenar las imágenes
mkdir -p /var/www/html/darknet/videos

#Copiado de videos en el directorio asignado.
scp -r usuario@host-proceso:/home/ETHERAPEminuto/$dia/VIDEO/
/var/www/html/darknet/videos/ETHERAPEminuto/$dia/
```

Este script se ejecuta cada una hora para copiar los últimos videos generados.

Debido a que los videos ocupan demasiado espacio en disco duro, éstos se borran cada una hora en el equipo de proceso. Entonces, cada hora los videos se copian desde el equipo de proceso hacia el servidor web, pero 30 minutos después de ocurrido esto, en el equipo de proceso se borran todos los videos generados 2 horas atrás. De esta manera se asegura que nunca se borre un video que todavía no haya sido copiado en el servidor web.

El borrado de los videos se hace de la siguiente manera:

```
#Se calcula la fecha actual
#!/bin/bash
year=$(date +%Y)
month=$(date +%m)
day=$(date +%d)

dia=$year-"-$month"-"$day

#Se calcula la hora pasada hace 2 horas
horapas=$(date --date='2 hour ago' +%k)

#Se ingresa al directorio donde se encetran los videos de hoy.
cd /home/ETHERAPEminuto/$dia/VIDEO/

#En caso de que la hora pasada sea menor a 10 se le debe concatenar #un "0" a la hora, y borrarle el espacio en blanco que trae por #defecto adelante del dígito de la hora.
if [ $horapas -lt 10 ];
then
#Se borran los archivos.
rm -rf video_0${horapas:1:1}*

else
#En caso de la que la hora no sea menor a 10, solamente se borran los #archivos.
rm -rf video_${horapas}*
echo borre video_${horapas}*
fi
```

Y se ejecuta en cada hora del día con 30 minutos, es decir, a las 00:30, 01:30, 02:30, etc.

Finalmente toda esta información se puede ver desde el sitio web, filtrando por fecha, herramienta y hora.

Visibilidad del proyecto

Para permitir darle visibilidad al proyecto se levanto un sitio web cuya direccion es <http://darknet.clcert.cl>, donde pueden consultarse los diversos gráficos y videos generados. La interfaz permite seleccionar una fecha determinada y el tipo de gráfico o video que se quiere visualizar.

Seleccione una fecha (imagenes desde 12-01-11, videos desde 13-01-11 23hrs)

Fecha :



Seleccione un tipo de gráfico

- | Nombre | Ícono |
|--|---|
| <input checked="" type="radio"/> Red de Conexiones (Afterglow) |  |
| <input type="radio"/> Cubo de Monitoreo (Moncube) |  |
| <input type="radio"/> Paquetes v/s Tiempo (TCP Stat) |  |
| <input type="radio"/> Video de Simulación (EtherApe) |  |

Procesar

Figura 18: Webpage página del proyecto.

Sección 3: Conclusiones

El objetivo general del proyecto, "diseñar e implementar una Darknet, operada por el CLCERT", se cumplió ampliamente. La darknet implementada se encuentra actualmente capturando datos y almacenándolos en archivos de datos tipo *pcap* en forma diaria. Gracias a estos datos es posible desplegar en forma automática gráficos que permiten visualizar los eventos de seguridad capturados. Se han generado además procedimientos que permitan compartir datos (archivos pcap o xml) con otros grupos cuando se requiera y además la posibilidad de utilizar dichos datos para alimentar subsistemas, como la honeynet actualmente en funcionamiento dentro del CLCERT.

Este proyecto permitió al CLCERT adquirir la experiencia necesaria para montar sistemas similares en subredes específicas, aprender del tipo de tráfico que circula en las redes nacionales y ayudar a concientizar sobre los riesgos de seguridad en la red. El proyecto pretende continuar en el tiempo y en lo posible extenderse con nuevas funcionalidades.

Sección 4: Trabajo futuro

Existen varios proyectos que implican la explotación de los datos generados y mejoras en los procedimientos de procesamiento de ellos, los cuales se espera concretar en el mediano plazo. Entre ellos se pueden destacar:

1- Una versión mejorada del sitio web del proyecto, que permita mayor flexibilidad en el acceso a los gráficos, por ejemplo definiendo rangos de tiempo en forma dinámica. También se espera estandarizar el despliegue de videos, de modo que sea compatible con la mayor parte de navegadores y sistemas operativos de los usuarios.

2- Obtención de datos desde sistemas de detección de intrusos, de modo obtener estadísticas de los tipos de ataque más frecuentes observados.

3- Redirección de un porcentaje de tráfico de la darknet hacia la honeynet local, implementando con ello un híbrido darknet/honeynet.

3- Procedimientos para compartir en forma automática datos que afecten a alguna entidad en particular (por ejemplo via ASN) usando filtros específicos que garanticen la privacidad de la información (evitar filtrar información indebida).

Anexo: Bibliografía

- [Mart10] Presentación "Aprendiendo del enemigo", Carlos M. Martinez, Anteldata CSIRT ANTEL, <http://www.csirt-antel.com.uy/main/public/aprendiendo-del-enemigo-01.pdf>, Agosto 2007.
- [MOORE10] - Presentación Network Telescopes, David Moore CAIDA, University of California, San Diego, Computer Science & Engineering Department, EE.UU. <http://www.caida.org/publications/presentations/2003/dimacs0309/>, Septiembre 2003.
- [Mich10] <http://www.eecs.umich.edu/fjgroup/>
- [AskStu10] Ask Student, <http://www.askstudent.com/security/sinkholes-in-network-security-5-easy-steps-to-deploy-a-darknet/>
- [CYMRU10] Team CYMRU (darknet project), <http://www.team-cymru.org/Services/darknets.html>
- [CAIDA1-10] CAIDA (Telescope Network), <http://www.caida.org/data/publications/bydataset/index.xml>
- [CAIDA2-10] CAIDA (Telescope Network), <http://www.caida.org/research/security/telescope/>
- [CAIDA3-10] CAIDA (Telescope Network), http://www.caida.org/data/passive/network_telescope.xml
- [UNAM10] Revista digital UNAM, Dirección General de Servicios de Cómputo Académico-UNAM. <http://www.revista.unam.mx/vol.9/num4/art21/int21.htm>
- [Snort10] <http://es.wikipedia.org/wiki/SNORT>
- [WISEDATA10] <http://www.wisedatasecurity.com/net-scanning.html>
- [BAD10] <http://bad-robot.blogspot.com/2008/08/portsentry-detecta-y-bloquea-escaneos.html>

