UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

DARKNET, MOTOR DE DETECCIÓN DE TRÁFICO MALICIOSO PARA EL TELESCOPIO DE SEGURIDAD DE LA UNAM

T E S I S

PARA OBTENER EL TÍTULO DE
INGENIERO EN COMPUTACIÓN

PRESENTA:
JAVIER ULISES SANTILLÁN ARENAS

DIRECTOR DE TESIS:
ING. RUBÉN AQUINO LUNA

CIUDAD UNIVERSITARIA, MÉXICO D.F 2011

AGRADECIMIENTOS

A Guille y Maleno, mamá y papá que siempre me han dado su apoyo y amor incondicional, consejos certeros y la guía necesaria para el logro de mis objetivos. Esto es fruto de su esfuerzo.

A mi hermano Daniel, ejemplo y guía que ha permitido definir parte de mí, tu compañía desde pequeño ha complementado mi camino en esta vida. A mi hermano Enry, que ha sido una influencia desde pequeño y me ha encaminado al lugar donde ahora estoy, este mundo de la computación, la curiosidad y las ganas de triunfar (aún recuerdo jugar Prince of Persia o "el persa", Gorilla desde Qbasic o escuchar que "se borra el sistema operativo por jugar Chess"). A toda mi familia por su constante ayuda, cuidado y cariño en todos los momentos de la vida.

A ti, Xim, por inspirarme, complementarme y cambiar mi vida llenándola de recuerdos y sentimientos inolvidables, por compartir el entusiasmo y amor por la UNAM y por darme las lecciones que han marcado mi forma de ser. "You're a skimmy, I'm a machine", "MorReckoner"…

A mis amigas y amigos, Rosa, Christian, Sergio, Julio por compartir gustos, proyectos y ambiciones personales y profesionales, así como a aquellos que durante la facultad me permitieron ser parte de exitosos equipos de lucha y desvelos.

A mi director de tesis, Rubén, por su paciencia y guía en el logro de mis objetivos profesionales, como jefe, profesor del plan de becarios y amigo.

A la Facultad de Ingeniería y al Departamento de Seguridad en Cómputo/UNAM-CERT por formarme personal y académicamente, encaminándome a la línea que he de seguir durante mi vida profesional: la seguridad en cómputo.

A mis amigos, Debian, Perl y C por sus interminables horas de lucha, desesperación y satisfacciones, exit(0);

Finalmente, a ti, mi querida y amada Universidad, cuna de conocimiento y fuente de saber, cultura y humanismo, lugar de experiencias inolvidables y espacios secretos. Tu nobleza y grandeza me inspiran a seguir creciendo y a dar lo mejor de mí para devolver a la sociedad parte de lo que por ti he logrado. Ahora, por el siguiente paso…

Hoy y siempre, puma de corazón y universitario por convicción.

POR MI RAZA HABLARÁ EL ESPÍRITU.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

A. PLANTEAMIENTO DEL PROBLEMA

El desarrollo y el incremento de las redes de datos alrededor del mundo han impulsado la creación de mecanismos para compartir, transferir o distribuir información por medios digitales. La facilidad, eficiencia y conveniencia de utilizar medios electrónicos implica, hasta cierto punto, exponer dicha información a determinadas amenazas que existen dentro de este mundo digital.

Estas amenazas potenciales como virus, gusanos, ataques dirigidos, negación de servicio (DoS), escaneos, botnets, spam, etc. si bien en concepto no son nuevas, durante los últimos años han ido evolucionando y adaptándose a los nuevos mecanismos de comunicación digital y en general al desarrollo de Internet. Algunas de sus principales características son la automatización de su distribución y la forma en que se aprovechan de las vulnerabilidades de sus objetivos. Esto significa que independientemente del daño o impacto que ocasionan, su existencia implica mayores consideraciones al momento de utilizar alguno de estos medios de comunicación digital. Tomando esto en cuenta, es entendible suponer la necesidad de poder identificar el origen de dichas amenazas con la finalidad de aplicar algún mecanismo de mitigación.

En la actualidad existen diversos mecanismos y soluciones para poder identificar anomalías en una red de datos y hasta cierto punto corregirlas. Herramientas como los sistemas de detección de intrusos (Intrusion detection system o IDS por sus siglas en inglés), sistemas de prevención de intrusos (Intrusion prevention system o IPS), firewalls, analizadores de protocolos o "sniffers", antivirus, correlacionadores de eventos (SIEM), etc. permiten monitorear y analizar la actividad del tráfico en la red, por otro lado, herramientas alternativas como las tecnologías honeypot permiten hacer un análisis y detección de tráfico de red malicioso por medio de una simulación o interacción real bajo un ambiente controlado con diversas entidades maliciosas.

Generalmente, cualquiera de estos mecanismos funciona de manera local o perimetral, es decir, pueden trabajar instalados directamente en el equipo o pueden estar monitoreando la actividad de la red por medio de un dispositivo en el perímetro de la misma. En concepto tienen muchas similitudes y la principal diferencia es solamente el alcance: actividad local o actividad en la red.

Existe otro tipo de detección de tráfico de red malicioso: las "Darknets" o "telescopios de red". En realidad es un concepto ambiguo porque no existe una definición formal que establezca sus características, sin embargo, su esquema consiste en una detección por medio de análisis de tráfico mediante diversas herramientas como IDS, honeypot, flujos, etc. desplegadas en infraestructuras de red con direcciones IP "no asignadas" o en los propios "core" de las redes. Dicha técnica no es nueva, sin embargo, a nivel mundial son escasas las implementaciones que permiten la detección de tráfico malicioso y clasificación de incidentes de seguridad en cómputo o monitoreo de actividad de Internet. Algunas de ellas, de las cuales se hablará más adelante, son CAIDA, The Darknet Project, Internet Motion Sensor, IBN, entre otros. En México no se tiene conocimiento de un proyecto similar, por lo que la implementación de la Darknet en el Telescopio de Seguridad de la UNAM representa la creación de una referencia importante en este tipo de esquemas.

La principal motivación para el desarrollo de este proyecto proviene de la necesidad de proporcionar mayor información sobre cada incidente reportado dentro de RedUNAM por parte de la Subdirección de Seguridad de la Información/UNAM-CERT. Esto representa la creación de una fuente de información y detección de incidentes de seguridad dentro y fuera de la red de la Universidad.

Implementado en entornos más pequeños y sencillos, funciona como un complemento a los sistemas de detección de intrusos, permitiendo generar estadísticas y referencias importantes de la actividad de tráfico de red malicioso.

B. OBJETIVO

El objetivo de este trabajo es presentar la propuesta de una Darknet como un motor de detección de tráfico de red malicioso para ser implementado en el Telescopio de Seguridad de la UNAM, explicando su diseño y características de funcionamiento, las ventajas que tiene respecto a otros tipos de herramientas, la comparación entre la UNAM-Darknet con otros modelos similares y finalmente, el análisis de la implementación puesta en producción.

C. ESTRUCTURA DE LA TESIS

El trabajo consta de seis capítulos fundamentales. El primer capítulo abarca el marco teórico de referencia que trata los conceptos relacionados con los sistemas de detección de tráfico malicioso, sistemas de monitoreo y sistemas de análisis de tráfico de red en general. Es necesario conocer y entender dichos tópicos debido a que el funcionamiento general del proyecto propuesto está basado en varios de ellos y con algunos otros tiene similitudes en cuanto a los objetivos que persiguen. El segundo capítulo presenta los fundamentos del diseño de una Darknet, sus características, funcionamiento, tipos, esquemas de implementación y un análisis al modelo de detección. En el tercer capítulo se presenta el diseño de un mecanismo de detección de tráfico malicioso para RedUNAM. Abarca el análisis del problema y de todas las consideraciones necesarias para la implementación en la red académica más grande de México. En el cuarto capítulo se profundiza en la estructura general de la herramienta desarrollada, explicando a detalle su esquema de funcionamiento ya implementado en RedUNAM y algunas otras características disponibles en el motor de detección. El quinto capítulo aborda el análisis de la implantación de la Darknet tomando en cuenta el aprovechamiento de la información recopilada, analizada, procesada y almacenada en el Telescopio de Seguridad de la UNAM.

Finalmente, en el sexto capítulo se presentan las conclusiones de la tesis las cuales abordan las deducciones finales de los resultados, ventajas y desventajas identificadas, limitaciones, capacidades posibles y las perspectivas sobre desarrollos futuros de la herramienta.

CAPÍTULO 1

ANTECEDENTES

Este capítulo tiene como objetivo abordar los conceptos necesarios relacionados con los sistemas de detección de tráfico malicioso, monitoreo de redes y análisis de tráfico de red en general. Éstos son necesarios para contextualizar el desarrollo del proyecto presentado en esta tesis, ya que como se verá más adelante, el esquema de funcionamiento general del motor de detección de tráfico malicioso toma como base a determinadas herramientas y técnicas vistas en este primer apartado.

1.1 MECANISMOS PARA DETECCIÓN DE TRÁFICO DE RED MALICIOSO

La actividad del flujo de datos a través de la red puede ser monitoreada y medida de diversas maneras. Este monitoreo incluye la clasificación del tipo de tráfico en cuanto a su naturaleza y es importante porque muchas de las funciones vitales de la red pueden estar en juego cuando una amenaza potencial se transforma en un incidente.

Se define como "tráfico de red malicioso" a cualquier tipo de tráfico que sea originado por algún agente o entidad maliciosa ya sea lógica (malware) o real (humanware) y que tenga la finalidad de lograr algún objetivo que atente contra alguno de los pilares de la seguridad informática: disponibilidad, integridad o confidencialidad.

A pesar de que el origen del tráfico puede no ser realmente malicioso (por ejemplo usuarios incautos), a final de cuentas se convertirá en tráfico malicioso "accidental" cuando de manera intencional o no, atente contra los pilares de la seguridad.

La importancia de poder identificar y detectar el tráfico malicioso se justifica con el hecho de que este tipo de tráfico es el que puede alterar el funcionamiento de una red o, en el peor de los casos, causar tal impacto que interrumpa por completo la actividad general del entorno. El SANS Institute[1] a través del Internet Storm Center (ISC), cuenta con un cálculo denominado "Survival Time"[2] el cual consiste en medir el tiempo promedio que tarda un equipo de cómputo en ser atacado o alcanzado por algún tipo de malware en propagación, considerando que se expone a una red pública sin

[1] Organización internacional dedicada a la investigación, capacitación y publicación de recursos relacionados con seguridad en cómputo.

[2] http://isc.sans.edu/survivaltime.html

restricciones. Si se diera el caso que el equipo no contara con los parches adecuados, entonces esta medición significaría el tiempo en que el equipo sería infectado o vulnerado. En la figura 1 se puede observar que las últimas mediciones indican que el "survival time" de un equipo Unix es de aproximadamente 3700 minutos, mientras que el de un equipo Windows es casi de 450 minutos. Esto nos da una muestra del verdadero problema con el tráfico de red malicioso.

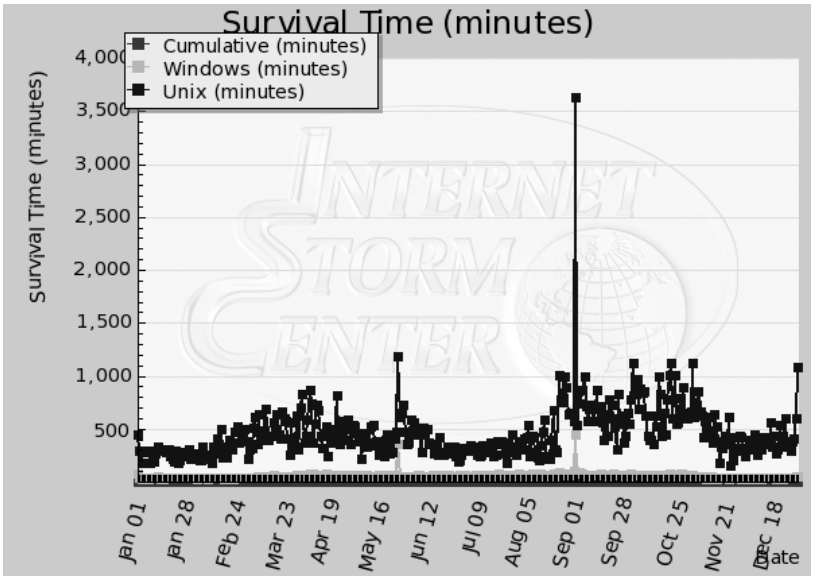


Figura 1. SANS ISC-Survival time (enero 2005 - agosto de 2010)

¿Cómo saber si determinado tráfico es anormal? o ¿cómo definir si el tráfico monitoreado es malicioso? Este tipo de cuestiones pueden ser respondidas abordando distintos puntos de vista. Por un lado se tiene la definición de políticas de una organización en las cuales se puede establecer lo que se considera como anormal y por otro lado se tiene el comportamiento general de determinadas amenazas que definen muy bien a un evento como un incidente de seguridad. Tomando en cuenta lo anterior, un ejemplo es que quizá para una organización el utilizar software P2P no representa mayor amenaza, sin embargo, las consecuencias indirectas de este hecho sí pueden ocasionar cambios en el entorno tales como afectación en la velocidad de la red, la exposición a las redes P2P potencialmente dañinas o fuga de información. Lo anterior demuestra que también es importante identificar la diferencia entre tráfico malicioso y tráfico anómalo.

El tráfico anómalo se define de acuerdo a las políticas o consideraciones de una organización, por lo que su existencia en una red será considerada de manera diferente independientemente de que pueda ser o no malicioso. Entonces, en los términos de la seguridad informática, podemos tratar de una manera más específica al tráfico malicioso relacionándolo con amenazas bien definidas que según su naturaleza puede tratarse de algún agente de malware o de una técnica o mecanismo especializado.

Algunos métodos para detectar "tráfico anormal" se basan en la comparación y análisis del comportamiento "esperado" de cierto tipo de protocolos o aplicaciones. Es decir, si se tiene bien identificada la estructura, forma y comportamiento del tráfico según su naturaleza, entonces cualquier patrón fuera de ella representa un factor para poder identificarlo como anormal. Algunos mecanismos para poder lograr dicha identificación y detección son las técnicas basadas en análisis de patrones, las herramientas especializadas y el análisis de bitácoras (log analysis). Ejemplos de algunos de ellos y que se abordarán a manera de antecedentes necesarios son:

- Firewalls
- Sistemas de detección de intrusos (IDS)
- Sistemas de prevención de intrusos (IPS)
- Analizadores de paquetes
- Analizadores de flujos
- Bitácoras de aplicaciones
- Telescopios de red o de seguridad

1.1.1 FIREWALLS

Básicamente, un firewall es un dispositivo físico o lógico que brinda la capacidad de controlar el flujo del tráfico de red a través de un sistema, es decir, funciona como un mecanismo de control de acceso a diferentes niveles. Estos dispositivos permiten controlar mediante políticas o reglas establecidas todo lo que fluye en el sistema tomando en cuenta factores como dirección, origen, destino, protocolos, aplicaciones,

etc. y brindan la capacidad de mantener aislado, hasta cierto punto, el sistema de potencial tráfico malicioso.

Fundamentalmente, un firewall necesita tener la capacidad de realizar las siguientes tareas:

- Administrar y controlar el tráfico de red
- Control de acceso
- Actuar como intermediario
- Proteger los recursos
- Registrar y reportar los eventos

La figura 2 muestra diversos tipos de firewalls a partir de los cuales se definen varias clasificaciones. Algunas de ellas son:

- Por modo de funcionamiento o tecnología
    - Firewall de filtrado de paquetes
    - Firewall de estado
    - Firewall de aplicación
    - Servidor proxy
- Por esquema de funcionamiento
    - Firewall de host
    - Firewall de red

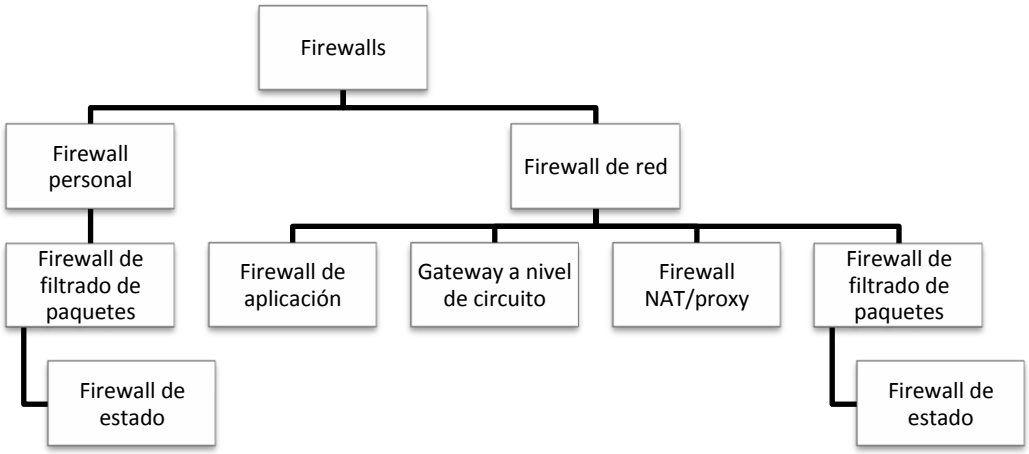


Figura 2. Tipos de firewalls

La principal diferencia entre un firewall de host y uno de red es que este último tiene la capacidad de analizar el tráfico de varios equipos en un entorno, a diferencia del de host que solo abarcará el tráfico que entra y sale de un equipo sencillo.

1.1.2 SISTEMAS DE DETECCIÓN DE INTRUSOS (IDS)

Los sistemas de detección de intrusos son dispositivos físicos o lógicos que permiten analizar el tráfico de red para identificar posibles paquetes maliciosos o anómalos. Son un mecanismo pasivo de detección ya que su tarea fundamental es alertar pero no actuar, sin embargo pueden proveer de información muy específica sobre la actividad detectada debido a que el análisis de un IDS sobre los paquetes de red es más específico que el que realiza un firewall. Esto convierte a los IDS en herramientas muy poderosas para conocer el panorama de la actividad en la red.

El funcionamiento general de los IDS se basa en detectar tráfico malicioso mediante firmas o anomalías.

La detección por firmas consiste en la definición de un patrón con características específicas las cuales comúnmente se basan en patrones de amenazas conocidas. Las firmas contienen características como tipo de tráfico, dirección de flujo, protocolo, direcciones IP, puertos o incluso el contenido de datos en el paquete.

Tabla 1. Ejemplo de firmas del IDS Snort

alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"SCAN SSH Version map attempt"; flow:to_server,established; content:"Version_Mapper";fast_pattern:only; classtype:network-scan; sid:1638; rev:6;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN"; flow:stateless; flags:SF,12; reference:arachnids,198; classtype:attempted-recon; sid:624; rev:8;)

En la Tabla 1 se pueden observar dos firmas, la primera define la alerta “SCAN SSH Version map attemp” cuando el protocolo es TCP y va hacia cualquier equipo de la red interna hacia el puerto 22, además debe establecerse la conexión y en el contenido del paquete debe contener la cadena “Version Mapper”. Cuando un paquete de red

coincida con este patrón, entonces se levantará la alerta proporcionando la información relacionada como marca de tiempo (timestamp), direcciones IP y puertos origen y destino, etc. Los desarrolladores de IDS comúnmente liberan nuevas firmas para poder detectar nuevas amenazas.

Por otro lado está la detección basada en anomalías. Este esquema funciona definiendo ciertos criterios base o *baselines* que suponen un funcionamiento normal del sistema o de la red. Cuando se detecta cierta actividad que no corresponde con el baseline en un cierto rango, entonces el IDS lo puede interpretar como una anomalía y en su caso identificarlo como tráfico malicioso. Al igual que las firmas, los baselines deben estar actualizados para poder ser más eficaces y detectar nuevas posibles amenazas.

Con los IDS existen dos problemas principales: los falsos positivos y los falsos negativos. Los primeros se refieren a todos los eventos levantados como alertas pero que en realidad no se trataban de tráfico malicioso. Esto puede ser causado por varios motivos pudiendo ser que alguna situación o aplicación llevaron a generar el paquete con las características de la firma y un aspecto muy fuerte a tomar en cuenta es que mientras más general se defina la regla, más falsos positivos se pueden tener.

El segundo problema es la generación de falsos negativos. Estos consisten en todos los eventos que a pesar de presentarse en la red no son alertados por el IDS. En realidad este tipo de omisiones puede implicar un mayor riesgo puesto que desde cierto punto de vista es más conveniente detectar algo que no existe, que no detectar algo que en verdad existe y que es malicioso, aunque debe tenerse presente que también contar con un número significativo de falsos positivos puede llegar a ser contraproducente. La detección y manejo de falsos positivos es un área de estudio específica en el campo de los IDS.

Existen varios tipos de IDS:

- IDS basados en host

Funciona monitoreando la actividad de un sistema local. Por su esquema de funcionamiento (figura 3), analiza el tráfico de red que entra y sale de dicho equipo, así como los cambios en el sistema de archivo y actividad del sistema en general.

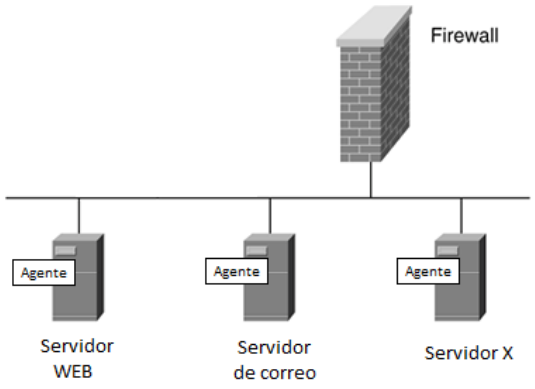


Figura 3. Esquema de un IDS de host

- IDS basados en red

Analiza el tráfico de un equipo o red. Puede instalarse en un equipo analizando sólo el tráfico que fluye a través de él, sin embargo, para que funcione ampliamente debe implementarse un esquema en donde reciba el tráfico de todos los equipos conectados a la red (figura 4), comúnmente llamado "port mirror". Generalmente se instala en el perímetro de la red o subred para poder monitorear el tráfico de entrada y salida de la misma. El éxito en su funcionamiento depende de su correcta ubicación.

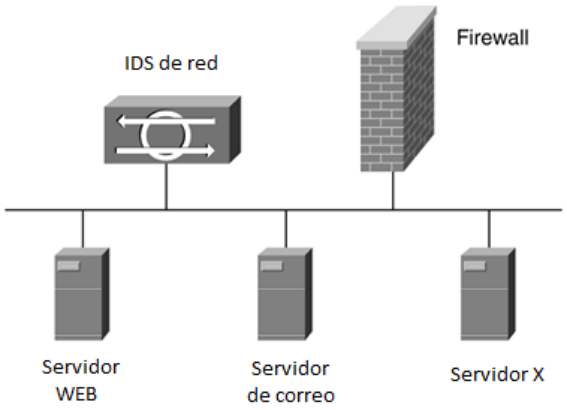


Figura 4. Esquema de un IDS de red

- IDS distribuidos

Es un esquema de varios IDS desplegados a lo largo de una red los cuales centralizan la información (figura 5). Este tipo de esquemas puede ser útil en redes de gran tamaño, sin embargo debido a la gran cantidad de información que implica, necesita un monitoreo y mantenimiento constante.

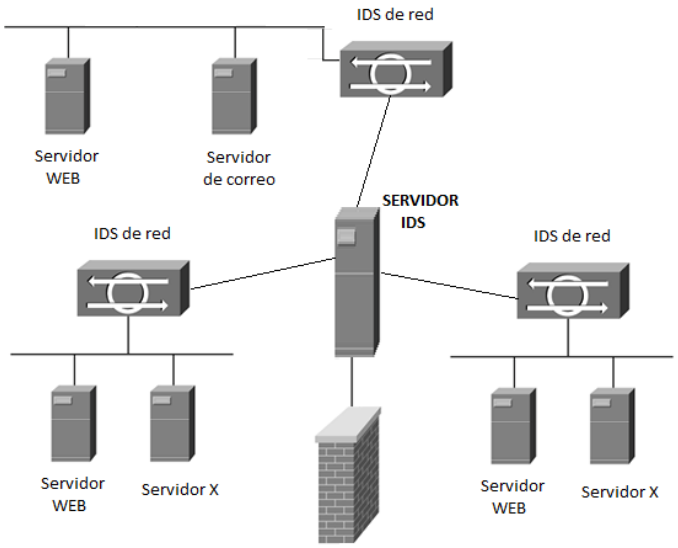


Figura 5. Esquema de un IDS distribuido

Los factores clave para el buen funcionamiento de los IDS son entonces: ubicación, buenas firmas, baselines y mantenimiento.

1.1.3 SISTEMAS DE PREVENCIÓN DE INTRUSOS

Los sistemas de prevención de intrusos o IPS son, al igual que los IDS, mecanismos físicos o lógicos para la detección de tráfico malicioso basándose en firmas o anomalías. La principal diferencia con los IDS es que los sistemas de prevención de intrusos son dispositivos activos que tienen la característica de actuar bajo demanda según las alertas detectadas, a esto se le conoce como "inline". Lo anterior significa que a partir de que un evento es detectado, el sistema puede aplicar automáticamente una medida de mitigación lo cual implica que los IPS tengan capacidades de firewall.

Por estas características, a este tipo de dispositivos también se les conoce como IDP o sistemas de detección y prevención de intrusos.

Por ejemplo, si un IDS alerta sobre un escaneo o ataque de fuerza bruta al servicio de SSH generalmente configurado en el puerto TCP/22, entonces solo alertará. Por su parte, el IPS adicionalmente creará una regla en su sistema de filtrado para bloquear los paquetes provenientes del origen del ataque o alguna otra medida configurada previamente.

Con este esquema podría pensarse que los IPS son mejores que los IDS, o incluso que podrían sustituir a los firewalls. Desde hace algunos años, firmas internacionales de seguridad y otras organizaciones de desarrollo e investigación, han abierto un gran debate sobre si los IDS han quedado obsoletos o representan una tecnología que será sustituida por los IPS. Gartner, organización mundialmente conocida dedicada a la tecnología y negocios, publicó en 2003 una declaración donde afirmaba "Intrusion Detection Systems a Market Failure"[3] mientras que otras opiniones expresaban lo contrario. El argumento era que los IPS no podría aún representar la nueva generación de dispositivos de detección debido a que aún eran inmaduros. Lo que es evidente es que aún existen consideraciones que deben tomarse en cuenta para la implementación de un IPS.

Primeramente está el problema de los falsos positivos. Con los IDS este problema implica solamente realizar un proceso de discriminación de alertas, el cual en el peor de los casos representa un trabajo adicional y excesivo para el administrador de la red. En cambio, con un IPS un falso positivo podría representar una auto-negación de servicio o problema general con la red puesto que de manera automática aplicaría las reglas en su firewall interno para poder mitigar "el ataque". Es por eso que las firmas de detección deben definirse con el criterio más acertado posible para minimizar el número de falsos positivos.

Debido a su naturaleza de mecanismo activo, un IPS debe:

[3] Gartner Information Security Hype Cycle Declares Intrusion Detection Systems a Market Failure. http://www.gartner.com/5_about/press_releases/pr11june2003c.jsp

- Dar respuesta inmediata a un evento
- Mantener el estado de las conexiones
- Tener conocimiento del protocolo o comportamiento de la aplicación

Existen dos tipos de IPS:

- NIPS – Network IPS: Monitorear tráfico de varios equipos en la red.
- HIPS – Host IPS: Monitorean el tráfico y actividad del equipo local.

1.1.4 ANALIZADORES DE PROTOCOLOS

Este tipo de herramientas también se conoce como analizadores de paquetes o "sniffers". Su principal objetivo es proveer la capacidad de capturar los paquetes que circulan por la red para poder analizarlos.

En ocasiones son considerados como "armas de doble filo" debido a que pueden ser utilizados con fines maliciosos, de manera que puedan capturar tráfico y obtener información privada atentando contra la confidencialidad de la información, sin embargo, su función consiste en un análisis a fondo del tráfico de la red.

Para que estas herramientas puedan ser utilizadas, las interfaces o dispositivos de captura son configuradas en modo "promiscuo" el cual consiste en activar la capacidad de poder recibir tráfico de red incluso si no va dirigido a dicha interfaz.

El utilizar este tipo de herramientas también implica la necesidad de mayor experiencia y conocimiento a fondo para un verdadero aprovechamiento, esto se debe a que un analizador de paquetes puede ser explotado solamente si se tienen los conocimientos adecuados de análisis de datagramas, es decir, conocer las técnicas y teoría necesaria para poder interpretar el contenido "crudo" que consiste en datos con formato hexadecimal. Entonces, la tarea fundamental es mapear la información a partir de los modelos de referencia de cada protocolo para saber qué datos del paquete corresponden a cada campo de la cabecera de determinado protocolo.

La relevancia de los analizadores de paquetes está en poder distinguir el tráfico potencialmente malicioso a partir de la definición de expresiones de filtrado o mediante procesamiento cuyo fin es agrupar información estadística e interpretarla. Por ejemplo, al capturarse todo el tráfico de una organización, puede notarse que existe tráfico excesivo hacia un determinado puerto o dirección IP, o que existe tráfico de servicios o aplicaciones no permitidas o utilizadas en la organización. El punto esencial es que el administrador de la red tenga la capacidad de interpretar estos patrones y poder indagar en algo más específico hasta poder detectar una amenaza o anomalía.

En concepto, muchos de los sistemas IDS, IPS o incluso firewalls tienen este tipo de herramientas como motor de captura de tráfico de red a un nivel más bajo, lo cual les permite analizarlo en tiempo real, sin embargo la diferencia radica en que generalmente solo almacenan los paquetes de red por los cuales se detectaron determinadas alertas, es decir, los paquetes que coinciden con las firmas o anomalías definidas por el IDS o IPS.

Varias de las herramientas más conocidas están basadas en la librería libpcap[4] y han sido portadas a varios sistemas operativos. Algunos ejemplos son Tcpdump, Windump, Wireshark, Snort, Tshark, entre otras.

Un punto muy importante a tomar en cuenta es que debido a que estas herramientas pueden capturar todo el tráfico de la red, el equipo en donde se instala debe contar con las características necesarias en cuanto a procesamiento y sobre todo, en capacidad de almacenamiento el cual depende totalmente de la cantidad de tráfico que circula en la red.

[4] Packet Capture Library http://www.tcpdump.org/pcap3_man.html

1.1.5 ANÁLISIS DE FLUJOS

Un flujo está definido como un grupo unidireccional de paquetes que comparten las siguientes características generales:

- Protocolo
- Dirección IP origen
- Puerto origen
- Dirección IP destino
- Puerto destino

El análisis de flujos es una técnica para detección de tráfico malicioso y monitoreo de red muy eficaz que contiene información de la capa de red y de transporte (modelo de referencia OSI). A diferencia del análisis de paquetes o la detección por medio de un IDS/IPS, este tipo de análisis no se basa en la definición de firmas o anomalías, sino que se encarga de estudiar los patrones en los flujos de la red tomando en cuenta métricas como las que se listan a continuación:

- Protocolo
- Direcciones IP: Origen y destino
- Marcas de tiempo (timestamp): inicio y final
- Conteo y estadísticas de paquetes
- Conteo y estadísticas de bytes
- TTL (para paquetes IP)
- Banderas TCP (para tráfico TCP/IP)
- Entre otras

En la tabla 2 se muestra un ejemplo de flujo de red donde se puede apreciar las características mencionadas y cuyos datos son suficientes para generar estadísticas importantes sobre el tráfico de red:

Tabla 2. Ejemplos de flujos de red

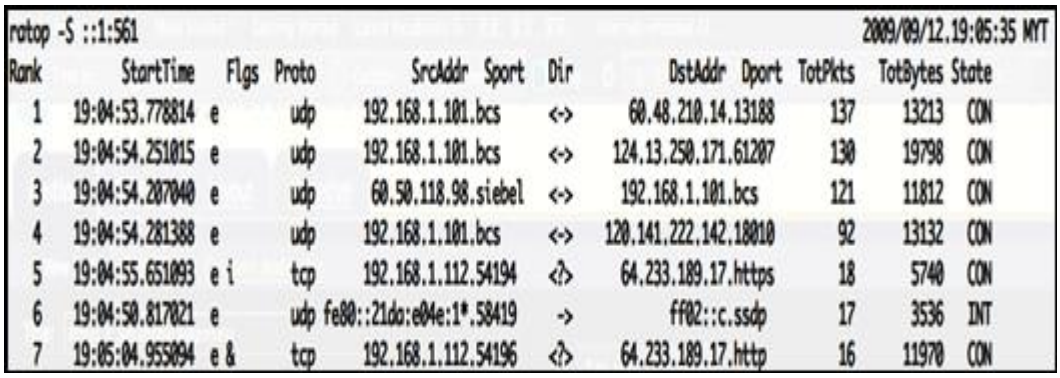
TOTAL PAQUETES	TOTAL BYTES	PROTOCOLO	IP ORIGEN	PUERTO ORIGEN	IP DESTINO	PUERTO DESTINO	TIMESTAMP
355	2312	TCP	1.1.1.1	34531	2.2.2.2	22	1282700162
1231	12223	TCP	3.3.3.3	139	4.4.4.4	139	1282700398
313	987	UDP	5.5.5.5	32212	6.6.6.6	21	1282701091

El objetivo de analizar este conjunto de datos busca lograr tareas como:

- Identificar y cuantificar las sesiones de red o conjuntos de sesiones que parezcan sospechosas.
- Obtener un monitoreo general de la actividad de la red mediante estadísticas del tráfico de red incluyendo conteos por protocolo, puertos, direcciones, estados, consumo de recursos, transferencias, etc.
- Facilitar la detección de patrones de tráfico maliciosos como accesos no autorizados, anomalías en servicios, propagación de malware, escaneos, tráfico en segmentos no utilizados, etc.
- Evitar el análisis del contenido completo del tráfico de red por lo que este procedimiento es más fácil de realizar e implica menor cantidad de recursos de almacenamiento.
- En caso de un incidente, mediante una auditoria se puede obtener una fuente importante de evidencia para reconstruir los hechos mediante logs.

Debido a que el análisis de flujos de red es más común en entornos de mediano o gran tamaño, la consideración principal es la cantidad de datos que se recolectan, no obstante, al poder colapsar en sesiones a todos los paquetes de red que compartan la información de protocolo y direccionamiento omitiendo el almacenamiento del contenido de todo el tráfico de red, lo convierte en un método muy eficaz para detección de tráfico malicioso que facilita mantener gran cantidad de información por largos periodos de tiempo haciendo muy conveniente su manejo mediante bases de datos y su explotación a través de técnicas de minería de datos. La figura 6 muestra un

ejemplo de la información de las sesiones de los paquetes de red y sus parámetros relacionados.



ratop -S ::1:561 2009/09/12.19:05:35 MYT

Rank	StartTime	Flgs	Proto	SrcAddr Sport	Dir	DstAddr Dport	TotPkts	TotBytes	State
1	19:04:53.778814	e	udp	192.168.1.101.bcs	<->	60.48.210.14.13188	137	13213	CON
2	19:04:54.251015	e	udp	192.168.1.101.bcs	<->	124.13.250.171.61207	130	19798	CON
3	19:04:54.287040	e	udp	60.50.118.98.siebel	<->	192.168.1.101.bcs	121	11812	CON
4	19:04:54.281388	e	udp	192.168.1.101.bcs	<->	120.141.222.142.18010	92	13132	CON
5	19:04:55.651093	e i	tcp	192.168.1.112.54194	<?>	64.233.189.17.https	18	5740	CON
6	19:04:50.817021	e	udp	fe80::21da:e04e:1*.58419	->	ff02::c.ssdp	17	3536	INT
7	19:05:04.955094	e &	tcp	192.168.1.112.54196	<?>	64.233.189.17.http	16	11970	CON

Figura 6. Información proporcionada por el cliente de argus (ratop).

Uno de los pioneros en el desarrollo y utilización de esta tecnología de análisis fue Cisco el cual desde hace algunos años ha agregado esta funcionalidad en sus dispositivos de enrutamiento. Sin embargo, también se han desarrollado tecnologías similares por otras compañías y a su vez, se han creado varias herramientas, formatos y protocolos que implementan el análisis de flujos. La tabla 3 muestra una relación de los diferentes formatos para análisis de flujos desarrollados por algunas compañías:

Tabla 3. Formatos de flujos de tráfico de red

COMPAÑIA	FORMATO FLUJO
Cisco Systems	NetFlow
Extreme Networks	CLEAR-Flow and sFlow
Foundry Networks	sFlow
Huawei Technology	NetStream
Juniper Networks	J-Flow, cFlowd (NetFlow v5)
Bluecoat Systems	Packeteer-2

1.1.6 ANÁLISIS DE BITÁCORAS

El análisis de bitácoras es todo un campo de estudio no solo en la seguridad informática, sino dentro de todas las tecnologías de la información. Es tal su

importancia que muchas de los nuevos mecanismos de monitoreo se han desarrollado basados en técnicas de análisis de bitácoras.

Su mayor importancia está en poder interpretar información almacenada de bitácoras de un sistema para poder identificar algún evento o situación específica. Abarca la utilización de varias herramientas como fuentes de datos y de otras para recopilación, procesamiento y explotación de la información por lo que se entiende como un mecanismo de detección pasivo posterior a la actividad detectada, compartiendo varios aspectos de un análisis forense.

Considerando el campo de la seguridad informática y específicamente la parte de detección de tráfico malicioso o monitoreo de redes, se puede obtener gran cantidad de datos a partir de fuentes como:

- Firewalls
- IDS/IPS
- Analizadores de tráfico y de flujos
- Sistemas de enrutamiento o conectividad (switches, routers, etc)
- Bitácoras del sistema operativo
- Bitácoras de aplicaciones de monitoreo
- Bitácoras de aplicaciones de propósito específico

Esto implica la utilización de algoritmos, técnicas o métodos de ordenamiento, clasificación y almacenamiento de la información para facilitar su procesamiento.

Debido a la gran cantidad de información que puede ser generada según el tamaño de la red o del nivel de detalle de las bitácoras, es conveniente dividir los datos fuente y preprocesarlos para una mayor eficiencia, sin embargo estas tareas pueden resultar demasiado complicadas si no se cuenta con las herramientas necesarias o las habilidades suficientes para poder manejar la información.

Saber si la red sufrió un ataque, conocer las estadísticas sobre el flujo de tráfico de red, detectar si un equipo o usuario específico intenta recurrentemente acceder a un

recurso no autorizado o incluso si se han logrado dichos accesos, entre otras actividades, son eventos que pueden ser detectados mediante el análisis de bitácoras.

Para definir un mecanismo de análisis de bitácoras es necesario tomar en cuenta:

- Cantidad de datos fuente
- Recursos de procesamiento (equipos, aplicaciones, etc.)
- Tiempos de respuesta

Una vez considerado esto, se debe definir de la manera más clara posible:

- Información esperada
- Formatos de salida

Para ello, existen diversas herramientas que permiten procesar datos:

- Herramientas nativas de los sistemas operativos (basados en UNIX/GNU Linux): grep, sort, awk, etc.
- Lenguajes de programación (scripting): Perl, Python, PHP, etc.
- Software especializado

De cualquier manera, una de las principales consideraciones en el análisis de logs será la cantidad de datos, puesto que implica procesamiento, lo cual a su vez se traduce en recursos y tiempo. El Dr. Anton Chuvakin[5] publicó en su blog[6] una medición trivial pero muy clara de la cantidad de recursos de almacenamiento que puede implicar un análisis de bitácoras:

> *100,000 log messages / second x 300 bytes / log message ~ 28.6 MB*
> *x 3600 seconds ~ 100.6 GB / hour*
> *x 24 hours ~ 2.35 TB / day*
> *x 365 days ~ 860.5 TB / year*
> *x 3 years ~ 2.52 PB*
>
> *Oops! Now you know what is a petabyte.*
> *And, BTW, you also now what is a trillion – of log messages.*

[5] Reconocido experto en el campo de la seguridad y análisis de logs, autor de los libros "Security Warrior", "PCI Compliance" y contribuidor de "Know Your Enemy II", "Information Security Management Handbook" y una docena de artículos sobre "log management", SIEM, seguridad y análisis de datos, administración de la seguridad, entre otros.

[6] http://chuvakin.blogspot.com/

Además, dependiendo de las necesidades de tiempo de respuesta ante un evento dado, pueden generarse problemas en la capacidad de procesar los datos cuando se llega a niveles de varios GB al día, no obstante, la correcta aplicación de técnicas y soluciones de análisis de bitácoras permitirá detectar, clasificar, almacenar y obtener interpretaciones sobre la información relacionada con la posible actividad maliciosa en el entorno de la red.

1.2 MECANISMOS ALTERNATIVOS PARA LA DETECCIÓN DE TRÁFICO MALICIOSO

Adicionalmente a los métodos comunes para detección de tráfico malicioso, desde hace varios años han surgido tecnologías con un enfoque de análisis más profundo. Algunas de estas tecnologías se basan en el concepto de emulación de servicios y redes y su objetivo principal es recibir, bajo un ambiente controlado, todo el tráfico malicioso que sea posible para ser almacenado y estudiado. Aunque existen distintas variantes de métodos alternativos de detección de tráfico malicioso, aquí se abordarán los temas relacionados con la propuesta original de la tesis.

1.2.1 TECNOLOGÍAS HONEYPOT

Este tipo de tecnologías surgieron a finales de los 90's bajo el concepto de sistemas honeypot. Su objetivo primordial es recibir el tráfico malicioso de manera intencional y poder interactuar con él. En el contexto más general, un honeypot es un equipo señuelo que se instala en un punto de la red para poder recibir, y por lo tanto detectar, tráfico malicioso o patrones relacionados. Una de las referencias más importantes en este campo es el proyecto "The Honeynet Project"[7], organización internacional fundada por Lance Spitzner en 1999 cuya actividad fundamental es la investigación en el campo de las tecnologías honeypots y las amenazas a la seguridad de la información. Fue Lance Spitzner quién introdujo por primera vez el término honeynet el cual define como una red de honeypots de alta interacción que simulan una red en

[7] Portal web del proyecto internacional http://www.honeynet.unam.mx

producción, configurada de manera que la actividad sea monitoreada, registrada y discretamente regulada.

Conforme han pasado los años, las características de los honeypots han cambiado pero el concepto inicial sigue siendo el mismo. Pueden clasificarse por su entorno y por su funcionamiento.

Por su entorno se dividen en:

- Honeypots de producción: Son equipos situados en el entorno real de la red junto con otros equipos en producción como servidores. Comúnmente son de baja interacción y funcionan como un complemento a la detección de amenazas, mitigando de alguna manera los riesgos en la actividad de la red.
- Honeypots de investigación: Son equipos instalados con el solo propósito de estudiar el comportamiento y las tendencias del tráfico malicioso causado por intrusos o ataques automatizados, situados en entornos exclusivos de prueba y generalmente con fines académicos. Tienen la característica de capturar y analizar la información por lo que su configuración y administración puede ser más compleja.

Por su modo de funcionamiento se dividen en:

- Honeypots de baja interacción: Son equipos cuya característica principal es emular los servicios de un sistema real para poder interactuar lo suficiente con los intrusos o amenazas automatizadas y recopilar datos. Al tratarse de una emulación, la efectividad y cantidad de información recopilada depende de la complejidad con la que el honeypot interactúe. Esto implica un menor riesgo pero los hace detectables más fácilmente. Son muy eficientes para la detección de patrones maliciosos, captura de malware y generación de estadísticas de tráfico en general.
- Honeypots de alta interacción: Son equipos reales, es decir, no implementan la emulación de ningún servicio sino que la interacción se da realmente por el software instalado en el sistema. Esto permite que los intrusos tengan una

interacción y control real del sistema señuelo lo cual implica un mayor riesgo y por lo tanto la necesidad de un sistema de control externo que permita monitorear, almacenar y procesar la información capturada. Este tipo de honeypot puede implementarse junto con mecanismos capaces de obtener información muy detallada de la actividad que el intruso realiza en el sistema (por ejemplo es posible monitorear los comandos que escribe en una terminal), por lo que de manera general el enfoque es más para un ámbito de investigación de las técnicas y tendencias de ataques de seguridad en cómputo.

La única diferencia entre una red de honeypots y una honeynet estrictamente se refiere a que las honeynets están formadas por uno o más honeypots de alta interacción desplegados bajo un esquema de control y análisis de tráfico de red como se observa en la figura 7.

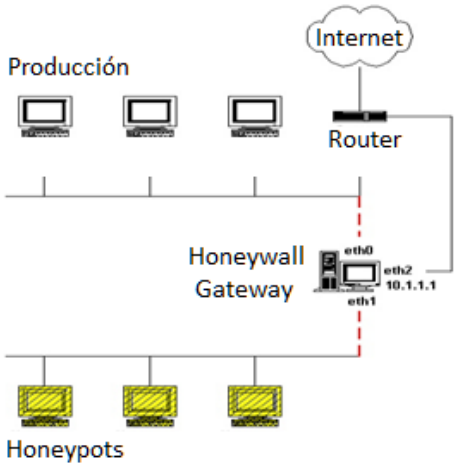


Figura 7. Esquema de conectividad de una honeynet

Las honeynets han evolucionado durante los últimos años implementando nuevas características, principalmente enfocada en el manejo y control de la información. Los tipos de honeynets que existen pueden englobarse en:

- Honeynets Generación I (1999)
    - Arquitectura inicial y básica de control y captura de datos

- Honeynets Generación II (2002)
    - Arquitectura mejorada en el control y captura de los datos: bridge capa 2, filtrado, sistema de alertas, módulo sebek v2.x.
- Honeynets Generación III (2005)
    - Arquitectura con análisis de datos
- Honeynets distribuidas
    - Honeynets desplegadas a lo largo de Internet
- Honeynets virtuales
    - Honeynets desplegadas en un mismo equipo

Cada módulo implementado en una honeynet cumple funciones específicas. La evolución de cada uno de ellos se ha dado según la necesidad de un mejor manejo y control de los datos. Las honeynets actuales permiten obtener información muy detallada de las actividades de los honeypots y de los aspectos del tráfico malicioso en ellos. La tabla 4 muestra las herramientas utilizadas en cada subsistema del Honeywall:

Tabla 4. Subsistemas del honeywall

MÓDULO	CARACTERÍSTICAS
Control de datos	Iptables – Firewall Rate-limitting – Límites de conexiones Snort-inline – Tecnologías IPS
Captura de datos	Logs de iptables – Información de firewall Alertas de IDS – Información de Snort p0f – Identificación pasiva de S.O. Sebek – Captura avanzada de datos Tcpdump – Captura de tráfico
Análisis de datos	MySQL - Almacenamiento de información en BD Argus y Hflow – Análisis de flujos Swatch – Análisis de logs de alertas del IDS y firewall Walleye – Interfaz gráfica Web

Con el avance de las características de emulación de los honeypots de baja interacción y los sistemas de administración como honeywall, la utilización de tecnologías honeypot ha permitido obtener buenos resultados y sobre todo sigue siendo un campo amplio en la investigación, tanto para fines académicos como para la industria. En la UNAM existe desde hace varios años el Proyecto Honeynet UNAM[8], el cual realiza investigaciones sobre este tipo de tecnologías y sobre mecanismos de detección y análisis de tráfico malicioso en general.

1.2.2 DARKNETS

El término de Darknet se refiere a todo el conjunto de direcciones o segmentos IP que no han sido asignados para algún servicio o dispositivo específico en una red, y que son destinadas a ser utilizados por uno o varios equipos con características especiales para identificación de posibles amenazas en la red. Partiendo de este concepto, todo el tráfico que circule por la red y que de algún modo se relacione con alguna de estas direcciones o segmentos IP, adquiere la característica de ser potencialmente anómalo o malicioso debido a que en teoría no debería existir tráfico dirigido hacia ellos.

Este concepto ha sido utilizado para poder identificar tráfico malicioso y fallas en dispositivos de red aprovechando su baja probabilidad de que se trate de un falso positivo. Posee las características adecuadas para combinarla con tecnologías honeypot y poder capturar, procesar y analizar muestras de tráfico malicioso en honeynets y redes en producción.

Debido a que es una parte fundamental en la propuesta de este trabajo de tesis, el concepto de Darknet se abordará de una manera más detallada en el capítulo 2.

[8] Proyecto dependiente de la Subdirección de Seguridad de la Información/UNAM-CERT en la Dirección General de Cómputo y de Tecnologías de la Información y Comunicación (DGTIC) y participante como UNAM-Chapter en The Honeynet Project. http://www.honeynet.unam.mx

1.2.3 TELESCOPIOS DE RED

De manera general, este concepto se refiere a un mecanismo de detección y monitoreo basado en el despliegue de sensores distribuidos a lo largo de un entorno de red o bien mediante el análisis de datos de varios dispositivos de conectividad. De cualquier forma, una de las características principales es que permite conjuntar la información recopilada por diversas fuentes de un espacio grande de monitoreo.

Su objetivo principal es detectar tráfico malicioso y monitorear la actividad general de la red combinando diversas tecnologías como las mencionadas en secciones anteriores (IDS, honeypots, Darknets, etc.), sin embargo, los telescopios de seguridad pueden ir más allá que los sistemas convencionales. Debido a que este tipo de mecanismo está enfocado a entornos de gran escala, adicionalmente de obtener un esquema de la actividad de la red, dan la posibilidad de generar muestras de información de tráfico malicioso de redes externas y dependiendo de su tamaño y complejidad, pueden llegar a ser referencias importantes para conocer las tendencias globales en la actividad del tráfico de Internet.

Algunas características de este tipo de tecnología son:

- Modelo de detección distribuido
- Modelo de administración centralizada o distribuida
- Entornos de gran escala
- Gran cantidad de información recopilada, procesada y almacenada
- Altamente demandante en recursos de hardware
- Generadores de información estadística importante
- Monitores de las tendencias de tráfico en espacios grandes de Internet
- Identificación de anomalías a nivel global

Quizá en un concepto muy general, la diferencia formal entre una Darknet y un telescopio de red es su tamaño y complejidad.

En apartados posteriores se mencionan características específicas del Telescopio de seguridad de la UNAM, cuyo principal motor de detección de tráfico de red malicioso es la Darknet-UNAM, propuesta de esta tesis.

CAPÍTULO 2

DARKNETS Y TELESCOPIOS DE RED

En este capítulo se abarca el concepto, funcionamiento y esquemas de implementación de la tecnología Darknet. Presenta también el análisis a los modelos de detección para poder tener un mejor contexto sobre el diseño y potencial de la Darknet-UNAM implementada en la red académica más grande de México, RedUNAM.

2.1 INTRODUCCIÓN A LAS DARKNETS

Como se mencionó en un apartado anterior, una Darknet es un equipo o conjunto de ellos los cuales utilizan direcciones IP o segmentos de red que no están asignados a ningún servicio o dispositivo específico dentro de un entorno. Esto quiere decir que todas las direcciones IP de la Darknet están explícitamente reservadas para no ser utilizadas en algún equipo de la red de producción lo cual implica que solo los equipos o el equipo, denominado servidor Darknet, hará uso de estas direcciones para su funcionamiento.

2.2 TRÁFICO DE RED "NO ASIGNADO"

El tráfico no asignado corresponde al relacionado con todas las direcciones IP de la Darknet. En un entorno ideal este tráfico no debería existir, sin embargo, es importante mencionar que el hecho de que exista no necesariamente significa que haya actividad maliciosa en la red ya que puede deberse también a alguna anomalía en la configuración de algún equipo o dispositivo de enrutamiento.

Las direcciones IP no asignadas representan una inversión del espacio de direcciones para obtener un mecanismo alterno de detección de tráfico malicioso y anomalías, principalmente cuando se trata de direcciones IP homologadas. El número de direcciones destinadas a la Darknet es proporcional a la cantidad de eventos detectados y dependiendo de las características y capacidades de implementación, es también proporcional a la efectividad de la información obtenida.

En organizaciones grandes pueden existir Darknets internas, es decir, con direcciones IP de segmentos privados. Por la misma razón, este tipo de Darknets serán efectivas solamente dentro del entorno de la red, siendo incapaces de detectar tráfico malicioso externo y limitando su acción a detectar eventos originados desde equipos internos. En este caso, el tamaño de la Darknet juega un papel diferente ya que aunque literalmente se asignen miles o decenas de miles de direcciones IP privadas, el campo de acción continúa limitado a la detección de tráfico generado en la red interna y posiblemente hacia redes externas.

La asignación del tipo de direcciones IP privadas o públicas depende de los objetivos de la implementación de la Darknet, pero para fines de un telescopio de red, se deben utilizar direcciones homologadas con el objetivo de poder detectar eventos desde y hacia redes externas.

La figura 8 muestra un ejemplo de una Darknet en una red en producción.

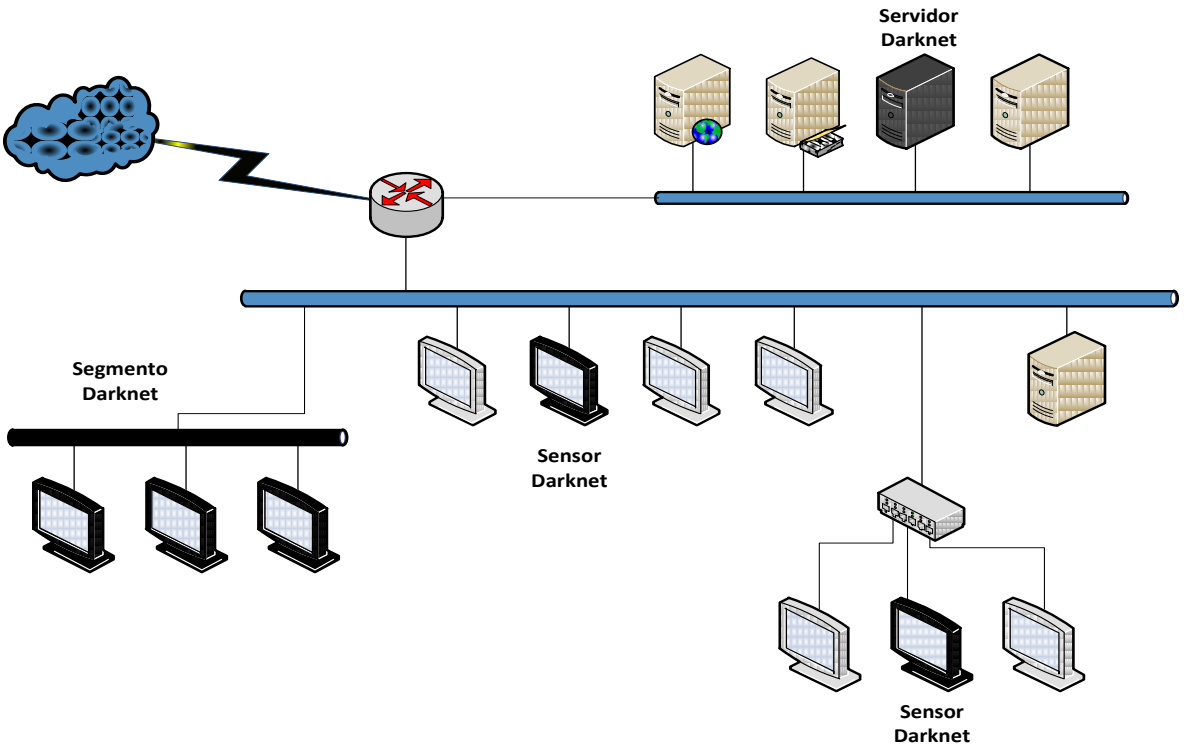


Figura 8. Esquema de una Darknet en una red en producción

2.3 CARACTERÍSTICAS DE UNA DARKNET

Independientemente de sus objetivos e implementación, de manera general una Darknet posee las siguientes características:

- Utiliza direcciones IP no asignadas.
- Todo el tráfico en la Darknet es potencialmente sospechoso.
- La cantidad de falsos positivos en la detección de anomalías es muy bajo.
- Puede detectar tráfico malicioso o anomalías en la configuración de dispositivos.
- Tiene las características para implementar tecnologías honeypot de manera muy conveniente y eficiente para obtener:
    - Muestras de tráfico malicioso
    - Muestras de malware
- Con la información obtenida, tiene la capacidad de generar información estadística importante sobre el tráfico de red.
- Inversión de direcciones IP de la red para su funcionamiento.

2.4 ESQUEMA DE FUNCIONAMIENTO Y HERRAMIENTAS RELACIONADAS

El funcionamiento específico de una Darknet depende de sus objetivos, pero el concepto general toma en cuenta aspectos como:

- Tecnologías implementadas
    - Honeypots, IDS, análisis de flujos, etc.
- Capacidad y complejidad de interacción
    - Simulación de servicios, equipos reales, etc.
- Capacidad y complejidad de análisis
- Campo de acción

El objetivo principal es que cada uno de los equipos o el servidor Darknet reciba e interactúe, bajo un ambiente controlado, con todo el tráfico dirigido hacia él. A partir de ese momento se puede identificar el origen, el tipo de tráfico, protocolo, puertos,

etc. y con esto saber si se trata de tráfico malicioso o alguna falla en la configuración de un equipo.

La clasificación de tráfico malicioso dependerá del mecanismo de detección, ya sea un equipo honeypot, un IDS u otro dispositivo, basándose en firmas o en el análisis del contenido del tráfico que dichas herramientas realicen. Mientras tanto, la detección de anomalías se puede identificar cuando se detectan patrones no necesariamente maliciosos y que corresponden a comportamientos como por ejemplo cuando un dispositivo de enrutamiento está mal configurado ocasionando que uno o varios equipos estén realizando conexiones recurrentes e innecesarias a otros equipos.

Ya sea que la Darknet esté compuesta por segmentos dedicados de direcciones IP o en equipos bien identificados de la red en producción, el tráfico no asignado es siempre detectable.

La figura 9 muestra un esquema básico de recopilación de información de cada sensor de la Darknet hacia un servidor dedicado.

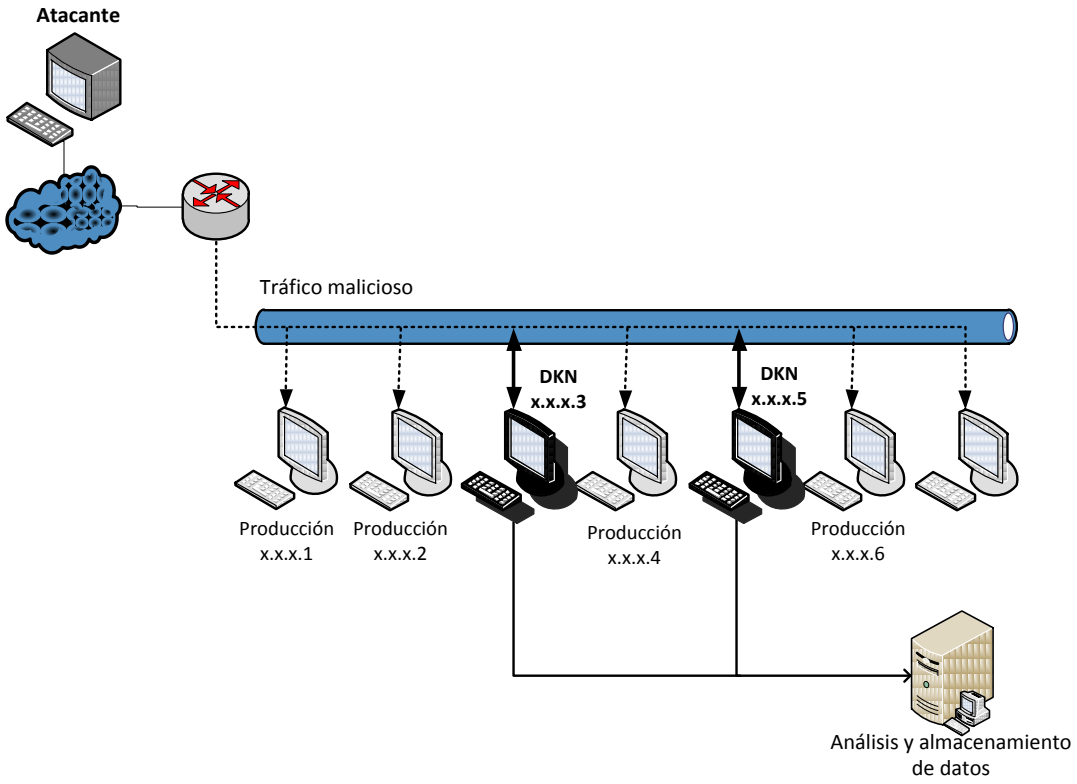


Figura 9. Esquema de funcionamiento de una Darknet.

Para que una Darknet pueda detectar algún ataque o cierto tipo de tráfico malicioso, debe forzosamente involucrar al menos a una de las direcciones IP de la misma. Esto quiere decir que si un ataque es dirigido a una IP fuera de la Darknet, entonces es muy probable que el ataque no pueda ser identificado. Afortunadamente, para los fines de detección, muchas amenazas en la red, principalmente las automatizadas como gusanos, bots, entre otros, presentan patrones en los cuales al intentar propagarse envían peticiones de conexión o algún tipo de escaneo al mayor número de equipos a su alcance. Esto revela las fuentes de los ataques puesto que en teoría el tráfico dirigido a equipos sin asignar no debería existir y en este ejemplo, el tráfico potencialmente sospechoso se convierte en tráfico malicioso identificado.

La gran ventaja de una Darknet es que al presentar bajos porcentajes de falsos positivos, los eventos identificados representan un estimado general de la actividad de tráfico malicioso que se recibe o circula en la red. Para obtener esta información pueden ser utilizadas diversas herramientas dependiendo del tipo de análisis, procesamiento y resultados esperados. La tabla 5 se muestra un ejemplo de tipos de análisis y herramientas utilizadas en la implementación de una Darknet.

Tabla 5. Tipos de análisis y herramientas utilizadas en una Darknet

TECNOLOGÍA	OBJETIVOS	EJEMPLOS
Honeypot	Simulación de servicios, captura de malware y control de tráfico	Dionaea, honeytrap, honeyd, kojoney, kippo, argos, honeybot, glastopf, google hack honeypot, honeywall, Hflow, etc.
IDS	Detección de tráfico malicioso mediante	Snort, Sguil, BASE, Suricata, Ossec HIDS, Prelude Hybrid IDS, Aide
Análisis de flujos	Análisis de flujos y generación de estadísticas de tráfico	Argus, Netflow, Hflow
Análisis de tráfico y protocolos	Análisis del tráfico de red: paquetes, protocolos, aplicaciones, etc.	Tcpdump, Wireshark, Tshark, Snort, Windump, ntop, etc.
Análisis de log	Análisis de logs de aplicaciones y sistema	Scripting perl, python, shell UNIX, Splunk, Logstash, etc.

Implementando alguna de estas tecnologías o combinaciones de ellas, se pueden detectar actividades como:

- Escaneos
- Propagación de gusanos, bots, virus
- Ataques de fuerza bruta
- Ataques específicos que utilicen técnicas de spoofing
- Fallas en la configuración de dispositivos
- Identificación de patrones de botnets o redes P2P
- Patrones anormales de tráfico
- Nuevas tendencias de ataques
- Entre otros

2.5 Análisis del modelo de detección en segmentos Darknet

Es muy importante hacer un análisis del modelo de detección para poder estimar de manera general la cantidad de tráfico esperado y los tiempos de captura, verificación, y almacenamiento de datos.

Actualmente, IPv4 (Internet Protocol versión 4) es la tecnología de direccionamiento de red desplegada en la mayoría de los dispositivos en el mundo. Es un protocolo de comunicación que permite un direccionamiento único de 32 bits para cada equipo dentro de la red. Esto quiere decir que el espacio de de asignación equivale a 2^{32}, lo cual resulta en 4.294.967.296 direcciones únicas.

Para poder identificar grupos de direcciones o segmentos, comúnmente se utilizan notaciones como /8, /16, /24, etc. refiriéndose al número de bits fuera de los 32. Por ejemplo, /8 hace referencia a un rango de 2^{24} direcciones que comparten los primeros 8 bits de la dirección. Así, /16 corresponde a un grupo de 2^{16} direcciones con los primero 16 bits comunes y /24 a 2^{8} direcciones compartiendo los primeros 24 bits. El caso de /32 hace referencia a una dirección IP única.

En el contexto de los telescopios de seguridad, estas notaciones son importantes ya que al referirse a bloques de red permiten hacer un cálculo estimado de la probabilidad de que un host dentro del espacio de monitoreo sea seleccionado como destino de alguna conexión por algún equipo en Internet. Si a esto se le agrega lo que la definición de Darknet marca sobre la naturaleza del tráfico en segmentos no utilizados, entonces dicha probabilidad se refiere a la recepción de tráfico de red potencialmente anómalo o malicioso.

La probabilidad *p* está dada por la cantidad de direcciones en el espacio de monitoreo de la Darknet entre la cantidad del espacio total de direcciones. Entonces, para IPv4 la probabilidad de detección en un segmento /x está dada por la ecuación:

$$P_x = \frac{2^{32-x}}{2^{32}} \qquad \text{o bien} \qquad P_x = \frac{1}{2^x}$$

Esto quiere decir que para un segmento /8, la probabilidad equivale a:

$$P_8 = \frac{1}{2^8} = \frac{1}{256} \qquad \text{o bien} \qquad P_8 = \frac{2^{24}}{2^{32}} = \frac{1}{2^8} = \frac{1}{256}$$

Y para /16 y /24 se tiene una probabilidad respectivamente de:

$$P_{16} = \frac{1}{2^{16}} = \frac{1}{65536}$$

$$P_{24} = \frac{1}{2^{24}} = \frac{1}{16777216}$$

De manera general, $P_x = \frac{1}{2^x}$ puede ser aplicada a cualquier otro espacio de direcciones, tal es el caso de IPv6 el cual maneja 128 bits.

2.5.1 TIEMPOS DE DETECCIÓN

Las Darknet son útiles para observar eventos aleatorios y espontáneos entre equipos. La medición del tiempo de detección es útil para poder establecer la duración mínima de espera para observar un evento según el tamaño de la misma. Algunas amenazas,

como los gusanos, intentan propagarse a todos los hosts posibles mientras que otros se propagan a rangos y ritmos específicos. Haciendo cálculos estimados de esta duración tomando como base el rango de objetivos en la Darknet (IP's alcanzadas), se puede hacer también un estimado de su efectividad.

Cuando un host toma como objetivo una dirección IP uniformemente aleatoria en todo un espacio de monitoreo, la probabilidad de detección en una dirección única está basada en una *distribución geométrica*. Una Darknet puede visualizar parte de este espacio, entonces, esta porción denominada p, se refiere a la probabilidad de que un paquete de red alcance una dirección dentro de ella. Si el host envía múltiples paquetes, entonces el número de paquetes vistos en el espacio de la Darknet está descrito por una *distribución binomial* con un parámetro p.

2.5.2 DETECCIÓN EN EQUIPOS ÚNICOS

Se asume que cada dirección asignada a la Darknet corresponde a un equipo independiente, entonces cuando un host genera múltiples paquetes, cada paquete tiene p posibilidades de alcanzar a la Darknet, y por lo tanto que la actividad sea detectada. Aplicando un análisis matemático, cada host objetivo seleccionado por el host fuente es una prueba de Bernoulli[42].

Considerando el número de paquetes generado por un host como el producto de la frecuencia con que los paquetes son enviados, r, y el tiempo T transcurrido., entonces la probabilidad de que al menos un paquete sea visto en la Darknet en un tiempo T está dada por la ecuación $P(t \leq T) = 1 - (1 - p)^{rT}$, la cual corresponde a una distribución geométrica. Su planteamiento se justifica en que corresponde a la distribución de probabilidad del número X del ensayo de Bernoulli necesaria para obtener un éxito contenido en un conjunto de pruebas, interpretando esto, la prueba corresponde al envío del paquete y el éxito a la recepción de la conexión en una IP dentro de la Darknet.

Desarrollando la expresión se tiene que:

$$T = \frac{-1}{rlog_{\frac{1}{1-P(t \leq T)}}(1-p)} = \frac{\log[1-P(t \leq T)]}{rlog(1-p)} = \frac{\log(Z)}{rlog(1-p)}$$

Así, la ecuación

$$T = \frac{-1}{rlog_{\frac{1}{Z}}(1-p)}$$

representa el tiempo T antes de observar al menos un paquete de un determinado evento con probabilidad Z (para Z=1-P(t≤T)). Entonces, la probabilidad corresponde a observar al menos un paquete con un objetivo determinado a una tasa de *r* pruebas por unidad de tiempo durante un lapso T, con lo cual podemos inferir un claro impacto según el tamaño del espacio de monitoreo dado el posible crecimiento exponencial entre los diferentes tipos de espacio de monitoreo (/8, /16, /24, etc.)

Con lo anterior, el número de paquetes esperados hasta que es visto el primero de ellos es:

$$\mu_N = \frac{1}{p}$$

con una varianza de:

$$\sigma^2{}_N = \frac{1-p}{p^2}$$

Ya que el interés es saber la tasa de transferencia de los paquetes y el intervalo de tiempo, se sustituye el número absoluto de paquetes enviados con rT obteniendo el tiempo transcurrido:

$$\mu_T = \frac{1}{rp}$$

Así, en este caso es más útil saber que el tiempo esperado de observar una amenaza proveniente de un equipo infectado con un gusano del estilo code-red en una red /8 es de 25.6 segundos, a diferencia de calcular que la Darknet tiene un 63.284% de

probabilidad de visualizar un ataque como ese en los mismos 25.6 segundos. Esto quiere decir que se pueden hacer ajustes según la probabilidad, por ejemplo, para observar un paquete del ejemplo anterior con una probabilidad del 99.999% se requieren aproximadamente 4.9 minutos.

La figura 10 muestra la relación tiempo-porcentaje de la probabilidad de observar al menos un paquete de un host que aleatoriamente selecciona un objetivo que esté dentro del espacio de monitoreo de Darknets de diferentes tamaños. En ella se puede observar la relación entre el tamaño del espacio de monitoreo y la efectividad de la Darknet para detectar determinados tipos de eventos. Esto a su vez denota la importancia que tiene la inversión de direcciones IP en este tipo de modelos de detección.

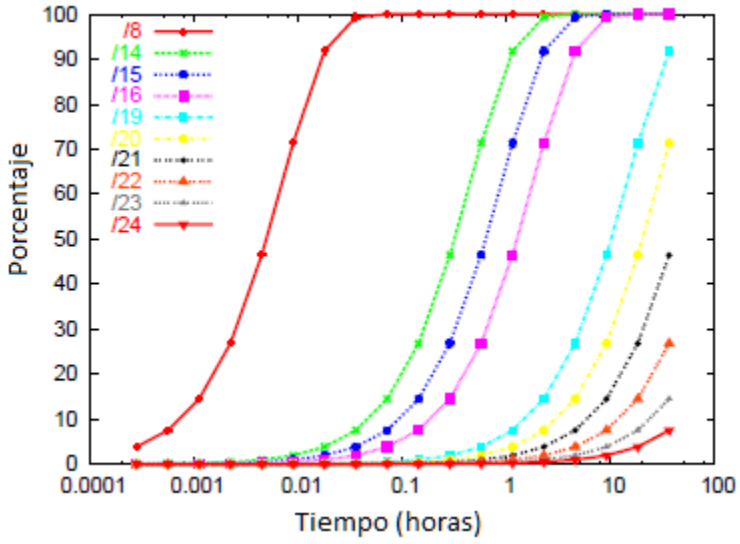


Figura 10. Relación tiempo-porcentaje para detección de un evento según el tamaño de una Darknet

Por su parte, la tabla 6 muestra una estadística de tiempos según el tamaño de una Darknet para un caso de envío de 10 paquetes por segundo provenientes de un equipo cualquiera. La primera columna indica la duración del evento para que la Darknet visualice el 95% de ellos, mientras que la última columna muestra la duración del evento para el cual la Darknet perdería el 95% de los mismos. Las columnas del centro demuestran la media y el promedio, es decir, los casos posiblemente más comunes. En este caso se debe tomar en cuenta que un evento en la práctica tiene una duración

limitada, por lo cual en la mayoría de las ocasiones será suficiente visualizar una mínima cantidad de paquetes para poder clasificarlo.

Tabla 6. Duración de eventos y porcentaje de detección según el tamaño de una Darknet para un caso específico.

Red	95%	Promedio	Media	5%
/8	1.3 min	25.6 seg	17.7 seg	1.31 seg
/14	1.4 hrs	27.3 min	18.9 min	1.4 min
/15	2.7 hrs	54.6 min	37.9 min	2.8 min
/16	5.5 hrs	1.82 hrs	1.26 hrs	5.6 min
/19	1.8 días	14.6 hrs	10.1 hrs	44.8 min
/20	3.6 días	29.1 hrs	20.8 hrs	1.49 hrs
/21	7.3 días	58.3 hrs	40.4 hrs	2.99 hrs
/22	14.5 días	4.85 días	3.36 hrs	5.98 hrs
/23	29.1 días	9.71 días	6.73 hrs	12.0 hrs
/24	58.2 días	19.4 días	13.5 días	23.9 hrs

2.5.3 DETECCIÓN DE MÚLTIPLES PAQUETES

En un esquema similar pero tomando en cuenta la detección de múltiples paquetes provenientes de una misma fuente, se pueden categorizar los tipos de eventos detectados y a su vez, disminuir la posibilidad de que se trate de un falso positivo y aumentando la posibilidad de identificar un potencial evento anómalo pudiendo ser malicioso como un ataque de negación de servicio. Esto es importante tomarlo en cuenta ya que la naturaleza de una Darknet también define la recepción de tráfico debido a fallas en la configuración (por ejemplo fallas en un router), o alguna anomalía que no se trate de una amenaza.

Dependiendo de las características de los eventos a ser monitoreados y propiamente de su diseño, el umbral k de paquetes que pueden ser seleccionados considerando la probabilidad de ver k o más paquetes de N transmitidos es:

$$P(vistos \ \geq k) = 1 - \sum_{y=0}^{k-1} \binom{N}{y} p^y (1-p)^{N-y}$$

La formula anterior se refiere a la definición de una distribución binomial, la cual mide el número de éxitos en una secuencia de n ensayos independientes de Bernoulli con una probabilidad fija p de ocurrencia del éxito entre los ensayos los cuales tienen dos posibles resultados, éxito(p) o fracaso(q=1-p). Interpretando esto, entonces el cálculo de la probabilidad de ver al menos 100 paquetes en un segmento /8 provenientes de un ataque DoS con una tasa de 500 paquetes por segundo y con una duración de 1 minuto está dada por:

N = 500pps * 60 sec = 30000 paquetes

K = 100 paquetes

p = 2^{-8}

$$P = 1 - \sum_{y=0}^{100-1} \binom{30000}{y} (2^{-8})^y (1-2^{-8})^{30000-y} = 95.2\%$$

En cambio, en una red /16:

N = 500pps * 60 sec = 30000 paquetes

K = 100 paquetes

p = 2^{-16}

$$P = 1 - \sum_{y=0}^{100-1} \binom{30000}{y} (2^{-8})^y (1-2^{-8})^{30000-y} = 6.7x10^{-195}$$

Se puede apreciar la variación considerable de la probabilidad según el tamaño del espacio de monitoreo dado el crecimiento exponencial de un segmento a otro de la red. En el Anexo 1 se muestra una comparativa de diferentes probabilidades para redes /8 y /16 haciendo variaciones en el número de paquetes enviados.

2.6 DARKNETS EN AMBIENTES ACADÉMICOS

Las redes académicas, principalmente de Universidades, representan un potencial campo de acción para la implementación de Darknets y honeynets. Esto se debe a que este tipo de ambientes posee características como:

- Altos anchos de banda desde y hacia Internet.
- Gran número de equipos utilizados.
- Muchos tipos de servicios, sistemas y arquitecturas utilizadas.
- Equipos con grandes capacidades de almacenamiento y procesamiento.
- Posible administración autónoma en cada área, dependencia, escuela, facultad, edificio, etc.
- Falta de restricciones en varias partes de la red.
- Existencia de tráfico de red inusual debido a proyectos de investigación.

Lo anterior también se complementa con que el principal objetivo de este tipo de implementación está enfocado a la investigación y desarrollo de nuevas tecnologías y herramientas en el campo.

Dependiendo de la infraestructura de la Darknet, siempre se debe tomar en cuenta que es necesario mantener una coordinación estrecha con los administradores de red o encargados de cada área. Esto es porque si se utilizan direcciones IP que ellos administran, es posible que al recibir y capturar tráfico malicioso se viole alguna política de seguridad y levante alertas de sistemas de monitoreo desplegados en la red.

En un documento [28] publicado por The Honeynet Project sobre Honeynets en Universidades, se mencionan algunas lecciones aprendidas, en este caso en The Georgia Institute of Technology (Georgia Tech). Tomando en cuenta que las características de una honeynet y una Darknet son similares, se puede hacer referencia a dichas lecciones:

1) Iniciar poco a poco: Si se desea implementar una honeynet en una organización grande, se debe iniciar con algo pequeño. Esto permitirá evaluar y entender cómo se analizarán los datos y desarrollar sistemas personalizados para ello.
2) Mantener buenas relaciones con los administradores de red: Esto es crítico puesto que son ellos quienes proporcionan la infraestructura lógica de IP's para realizar las tareas, además, en los casos en los que ellos no hayan detectado algún tipo de ataque, exploit o tráfico malicioso específico, quien implementa la honeynet (o Darknet en este caso), podría ser la primera persona que los notifique de ello.
3) Enfocarse en los ataques provenientes de la red de la organización: Este tipo de ataques son de los que más daño causan a las organizaciones. Debe informarse inmediatamente a los administradores sobre los equipos que hayan sido comprometidos dentro de la organización.
4) No publicar el rango de las direcciones IP de la honeynet (o Darknet).
5) No sobreestimar la cantidad de tiempo necesario para analizar los datos recopilados.
6) Para implementar una honeynet no se necesitan equipos con altas prestaciones de procesamiento o almacenamiento (aunque esto depende del esquema de implementación).

Definitivamente, el despliegue tanto de honeynets como Darknets en ambientes académicos traen grandes beneficios y sobre todo permiten obtener y analizar información muy útil sobre detección de tráfico malicioso.

2.7 DARKNETS A GRAN ESCALA

A nivel global existen varios proyectos de diferentes tamaños y características, algunos más complejos que otros, sin embargo persiguen un concepto común: el monitoreo de la actividad de tráfico de red. Estos proyectos ofrecen distintos tipos de información siendo una de sus principales diferencias el tamaño y la forma de recopilación y procesamiento de los datos. Algunos de ellos, debido a su tamaño y capacidad, se han convertido en referencias importantes para consultar la actividad del tráfico de red y las tendencias en Internet.

A continuación se menciona un panorama general sobre algunos de los proyectos de este tipo para poder hacer una comparativa en los modelos de detección y capacidades en general.

2.7.1 INTERNET MOTION SENSOR (IMS)

Es un proyecto desarrollado entre la firma de seguridad Arbor Networks y la Universidad de Michigan. Consiste en un sistema de monitoreo de las amenazas de Internet a nivel global cuyo objetivo es medir, clasificar y dar seguimiento a dichas amenazas. Posee una infraestructura distribuida de sensores ubicados en distintos lugares abarcando segmentos de red desde /25 hasta /8 en redes académicas, comerciales e ISP's y cuenta con una infraestructura de almacenamiento por jerarquías. Su funcionamiento se basa en diversas tecnologías de detección entre las que se encuentran honeypots, análisis de flujos, análisis de payloads, etc. La arquitectura general del proyecto IMS persigue tres objetivos:

- Mantener un nivel de interacción capaz de identificar tráfico del mismo servicio
- Clasificación de las amenazas emergentes
- Visibilidad de Internet más allá de las fronteras geográficas u operativas

Los sensores desplegados están clasificados en "dark IP" (segmentos de la Darknet) y "topology" (segmentos dentro de una topología asignada). La figura 11 muestra la infraestructura del IMS.

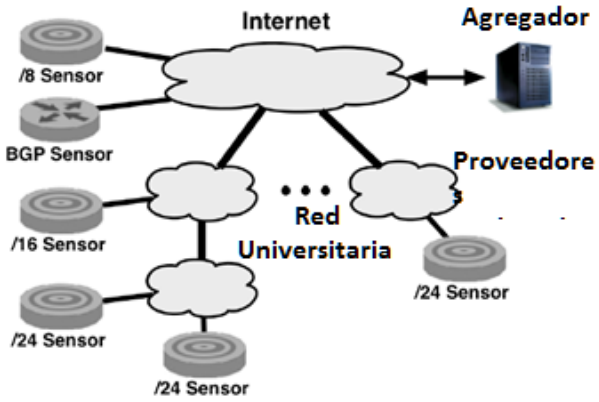


Figura 11. Infraestructura del Internet Motion Sensor

Con esta infraestructura, es capaz de detectar y analizar amenazas como gusanos, escaneos, ataques DoS, entre otros, con la característica de hacerlo en tiempo-real.

La figura 12 muestra el procedimiento de análisis y manejo de información del IMS.

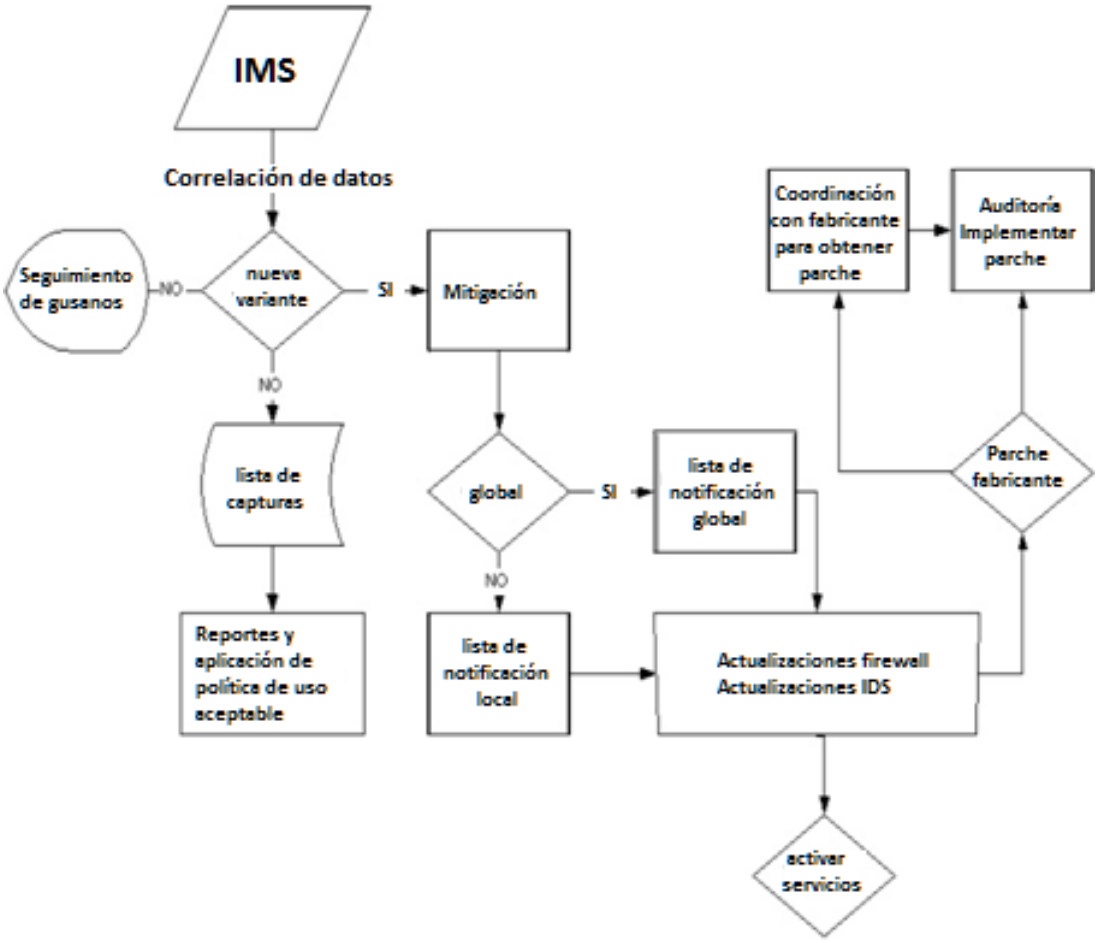


Figura 12. Esquema de funcionamiento del IMS

2.7.2 CAIDA

Cooperative Association for Internet Data Analysis. Es una asociación cuyos objetivos son:

- Medir y entender el tráfico en Internet
- Desarrollar herramientas para la medición y análisis
- Recopilar y proveer datos sobre Internet de diferentes rubros: seguridad, dns, routing, topologías, etc.
- Visualización de red

Posee una de las infraestructuras más grandes a nivel global para la medición y análisis del tráfico en Internet. Consta del llamado "Archipelago" o "Ark" el cual en su última actualización en Febrero de 2010 tenía desplegada 41 monitores en 25 países. La tabla 7 y la figura 13 muestran la distribución por continente y su ubicación geográfica respectivamente.

Tabla 7. Infraestructura de "Archipelago" de CAIDA

Por continente	Por organización
17 Norteamérica	21 Académicas
14 Europa	10 Redes de investigación
5 Asia	5 Infraestructuras de red
2 Oceanía	4 Redes comerciales
2 Sudamérica	1 Red comunitaria
1 Africa	

Figura 13. Monitores del Archipelago de CAIDA

Uno de los proyectos de CAIDA es el UCSD Network Telescope, el cual consiste en una Darknet con un potencial de una red /8, esto quiere decir aproximadamente 16 millones de direcciones IP. Este Telescopio tiene como objetivos la detección de ataques de negación de servicios, propagación de gusanos, y detección general de tráfico malicioso generado por agentes automatizados.

2.7.3 TEAM CYMRU, THE DARKNET PROJECT

Team Cymru es una organización especializada en la investigación sobre seguridad en Internet. Uno de sus proyectos es "The Darknet Project", el cual al igual que sus similares, es capaz de identificar actividad maliciosa en Internet y a su vez generar estadísticas de tráfico para saber qué es lo que pasa en la red. Utiliza tecnologías como el análisis de flujos y el análisis de tráfico de red.

La infraestructura de este proyecto consta de 8 Darknets desplegadas en diferentes zonas geográficas tal como se muestra en la tabla 8.

Tabla 8. Darknets de "The DArknet Project"

Darknet	Espacio de IP's
Darknet 1 (ARIN/US)	6 Redes /24 (1,536)
Darknet 2 (ARIN/US)	4 Redes /16 (262,144)
Darknet 3 (ARIN/US)	10 Redes /24 (2,560)
Darknet 4 (ARIN/CA)	1 Red /16 (65,536)
Darknet 5 (ARIN/US)	1 Red /17 (32,768)
Darknet 6 (ARIN/US)	1 Red /24 (256)
Darknet 7 (RIPENCC/NL)	2 Redes /16 (131,072)
Darknet 8 (ARIN/US)	2 Redes /16 (131,072)
	Total = 626,944 DARK IP

A partir de herramientas de monitoreo como RRDTool, es posible monitorear la actividad general de las Darknets, sin embargo, gracias a herramientas de análisis de flujos como Argus, se puede monitorear la actividad específica de algún puerto, IP, protocolo, etc. como el ejemplo de la figura 14, el cual corresponde al monitor de actividad de las darknets de Team Cymru.

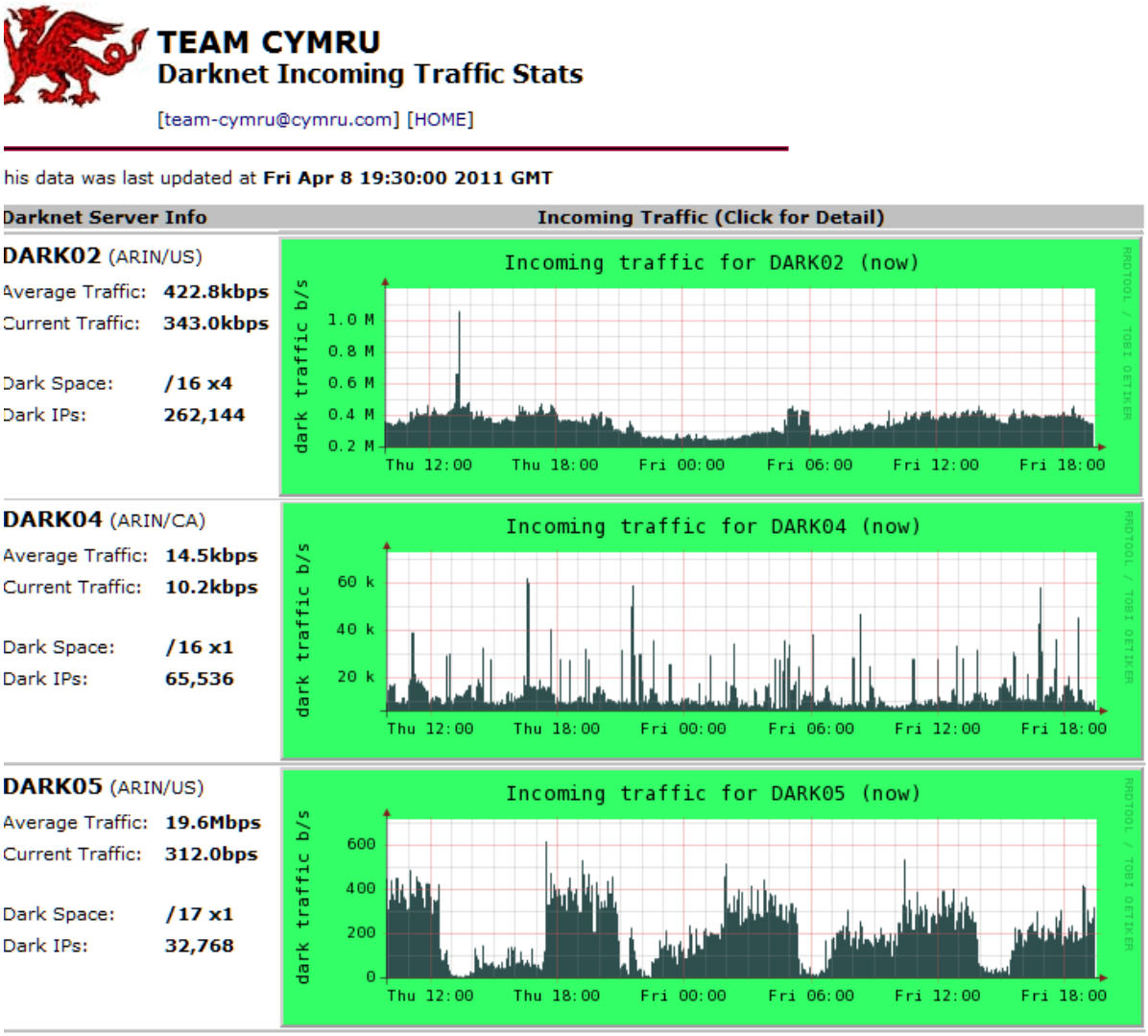


Figura 14. Actividad de las Darknet de The Darknet Project de Team-Cymru

En la página de Team-Cymru [53] sobre su proyecto The Darknet Project, se proporciona un manual práctico[9] de cómo implementar una Darknet.

2.7.4 iSINK (INTERNET SINK)

Es un proyecto desarrollado en la Universidad de Wisconsin el cual consiste en una arquitectura altamente escalable para la medición, monitoreo y análisis automatizado de trafico malicioso. iSink es un sistema basado en tecnologías honeypot

[9] http://www.team-cymru.org/Services/Darknets.html

implementado sobre una arquitectura Darknet, cuya capacidad de emulación de respuestas permite monitorear y clasificar ataques a lo largo de grandes subredes. De acuerdo a su equipo desarrollador en The Wisconsin Advanced Internet Laboratory (WAIL)[10] de la Universidad de Wisconsin-Madison, este sistema implementa un protocolo para la generación de firmas NIDS de manera automatizada las cuales presuponen una baja probabilidad de falsos positivos comparable con algunos sistemas populares NIDS. Los objetivos de iSink son:

- Abordar el problema del diseño e implementación de un sistema de monitoreo para grandes segmentos de red.
- Crear un sistema altamente escalable con un nivel suficiente de interacción para poder detectar gusanos, ataques, fallas en la configuración, etc.

Las características de iSink son:

- Usa componentes activos y pasivos.
- Utilizar técnicas de muestreo en sus componentes para incrementar la escalabilidad.
- Potencial de aproximadamente 100,000 direcciones IP en una Darknet.

La figura 15 muestra la infraestructura de esta darknet.

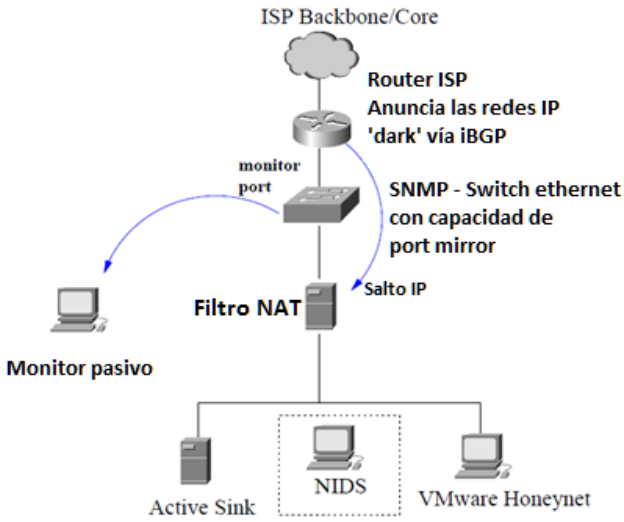


Figura 15. Esquema de conectividad de iSink

[10] http://wail.cs.wisc.edu/

La implementación de iSink consta de los siguientes módulos:

- Passive monitor
- Active Sink
- NAT filter
- Vmware honeynets
- NIDS

2.7.5 THE IUCC/IDC INTERNET TELESCOPE

Es un telescopio de red desarrollado por Israel InterUniversity Computation Center (IUCC). Básicamente permite el monitoreo de trafico sobre una Darknet la cual utiliza la técnica de backscatter[11] para la detección de ataques de IP spoofing. Su espacio de direcciones es un segmento de red /16. Los principales eventos detectados por este telescopio se muestran en la tabla 9.

Tabla 9. Estadísticas del telescopio de IUCC/IDC

Tipo de paquete	Porcentaje
Escaneo de puertos/equipos	92%
DDOS Backscatter	5%
Fallas de configuración	2%
Otros	1%

Es capaz de generar información estadística tomando como criterios el origen y el destino de los paquetes. Sin embargo, existen algunos ataques que no pueden ser vistos por este telescopio como los de bogon[12], ataques sin suplantación de IP y ataques de Botnets.

[11] Se refiere a las respuestas que un equipo devolvería a un equipo víctima de suplantación de IP. Si un atacante hace una suplantación de manera aleatoria, el ataque es visible debido a que el telescopio mandaría respuestas del tipo (SYN-ACK) a equipos aleatorios.

[12] Es un ataque que proviene de una dirección IP que no está en las tablas de ruteo de ningún dispositivo en Internet. Existe una lista definida por IANA de estas direcciones

2.7.6 *INTERNET BACKGROUND NOISE (IBN)*

Es un proyecto desarrollado por la organización SWITCH[13] que consiste en un sistema de monitoreo de tráfico de red basado en tecnologías Darknet. Todo el tráfico de las direcciones IP no utilizadas es redirigido a un servidor IBN el cual recopila y procesa la información. Tiene un potencial de aproximadamente tres segmentos /17 para monitoreo. Su principal objetivo es generar estadísticas según el tipo de protocolo, puertos destino, tipos de mensajes ICMP, etc. así como se aprecia en la figura 16.

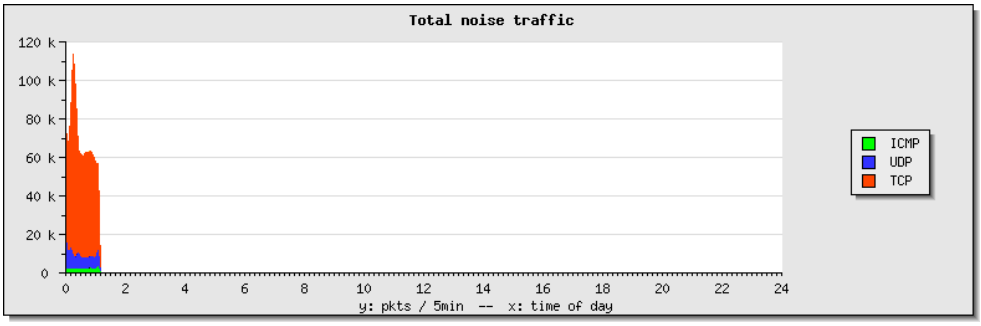


Figura 16. Estadísticas generadas por IBN

2.7.7 *SHADOWSERVER*

Shadowserver Foundation es un grupo de especialistas en seguridad que se encargan de recopilar, dar seguimiento y reportar malware, actividad de botnets y fraudes electrónicos. Su misión principal es mejorar la seguridad en Internet mediante la concientización de la presencia de servidores comprometidos, atacantes maliciosos y la propagación de malware. Sus tareas fundamentales se basan en:

- Captura y recepción de software malicioso o información relacionada de dispositivos comprometidos.
- Análisis de malware mediante sandbox, desensamblado y otras técnicas.
- Monitorear y reportar atacantes maliciosos.
- Seguimiento y reporte de actividad de botnets.
- Diseminación de información de *cyber-amenazas.*
- Coordinación en respuesta a incidentes.

---

http://www.team-cymru.org/Services/Bogons/

[13] http://www.switch.ch/about/

Su estructura de funcionamiento se representa en la figura 17.

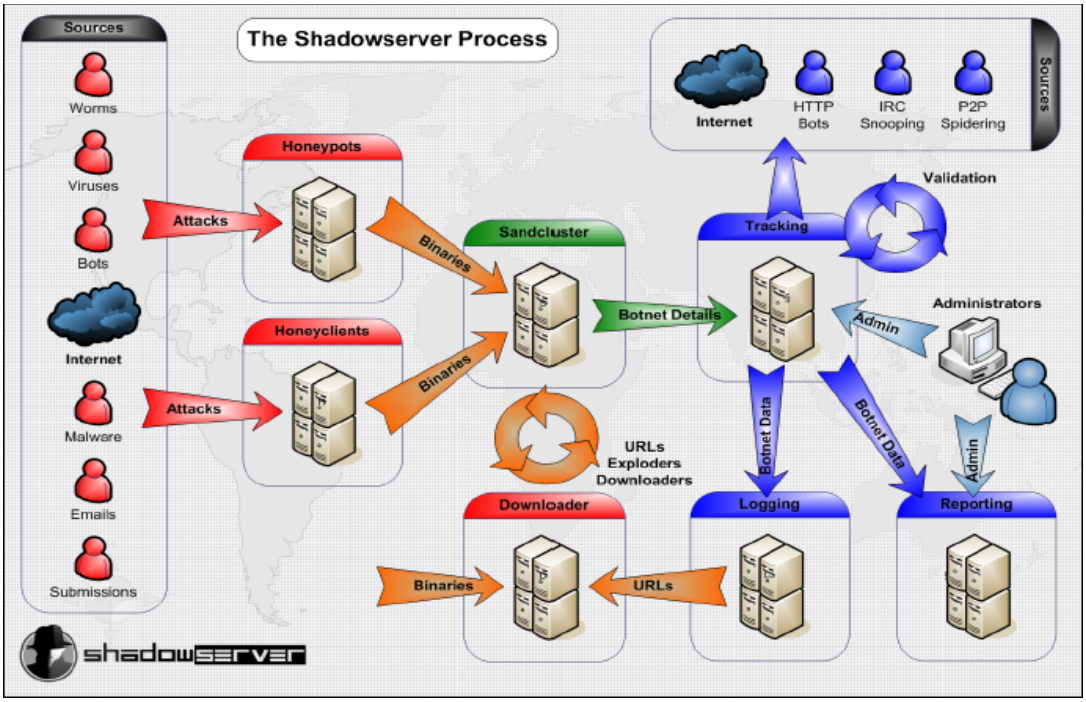


Figura 17. Proceso de manejo de información de Shadowserver Foundation

El UNAM-CERT tiene un convenio con Shadowserver para intercambio de información sobre actividad maliciosa detectada la cual se utiliza específicamente para complementar el proceso de atención a incidentes de equipos de RedUNAM.

CAPÍTULO 3

DISEÑO DE UN MECANISMO DE DETECCIÓN DE TRÁFICO MALICIOSO PARA REDUNAM

En este tercer capítulo se presenta el diseño de un mecanismo de detección de tráfico malicioso para RedUNAM. Abarca el análisis del problema y de todas las consideraciones necesarias para la implementación de una Darknet en la red académica más grande de México.

3.1 ANÁLISIS DEL PROBLEMA

RedUNAM es una red de datos de gran escala en términos de ambientes académicos. Su infraestructura se compone de miles de dispositivos entre equipos, servidores, switches, routers, etc. Debido a la grandeza de la Universidad tanto en cuestión geográfica como su población de más de 300,000 alumnos[14], se trata de un entorno complejo y una fuente extraordinaria de generación, distribución, transferencia y almacenamiento de información.

En realidad, el monitoreo del tráfico malicioso en RedUNAM es una cuestión demasiado compleja debido a las políticas de administración descentralizada. Esto quiere decir que cada dependencia tiene la responsabilidad de administrar su propia red, por lo que todos los eventos relacionados con tráfico malicioso no están monitoreados por un sistema centralizado, sin embargo, se cuenta con la capacidad técnica de implementar un sistema de detección generalizado a través del Telescopio de Seguridad de la UNAM.

Para el caso de RedUNAM, existe la Subdirección de Seguridad de la Información / UNAM-CERT dependiente de la Dirección General de Cómputo y Tecnologías de la Información y Comunicación (DGTIC). El UNAM-CERT se encarga de la atención a los incidentes de seguridad informática detectados y reportados por diversos mecanismos, sin embargo, la capacidad de detección por medio de métodos alternativos como una Darknet, y el aprovechamiento de ser un entorno académico, representan grandes ventajas para poder mejorar la información que se obtiene sobre

[14] Cifras del año 2010 del Portal de Estadística universitaria http://www.estadistica.unam.mx/numeralia/

tráfico malicioso para fines de detección y respuesta a incidentes y con enfoque también para el campo de la investigación.

3.2 DARKNET COMO MECANISMO DE DETECCIÓN DE TRÁFICO MALICIOSO PARA REDUNAM

La implantación de una Darknet dentro de la infraestructura de RedUNAM es totalmente factible y provechosa. El potencial de dos segmentos clase B[15] públicos permite obtener gran cantidad de información sobre tráfico malicioso dentro y fuera de la Universidad. Si bien no todas las direcciones están destinadas para la Darknet, se tiene un potencial de miles de direcciones IP disponibles para ser utilizadas. El mecanismo de detección de la Darknet está diseñado para procesar la información y generar una fuente de datos para el Telescopio de Seguridad de la UNAM. Como se mencionó en un apartado anterior, es un mecanismo que conjunta varias fuentes de información sobre tráfico de red malicioso entre las que se encuentran honeypots, sensores de spam, IDS, etc. La figura 18 muestra su estructura general.

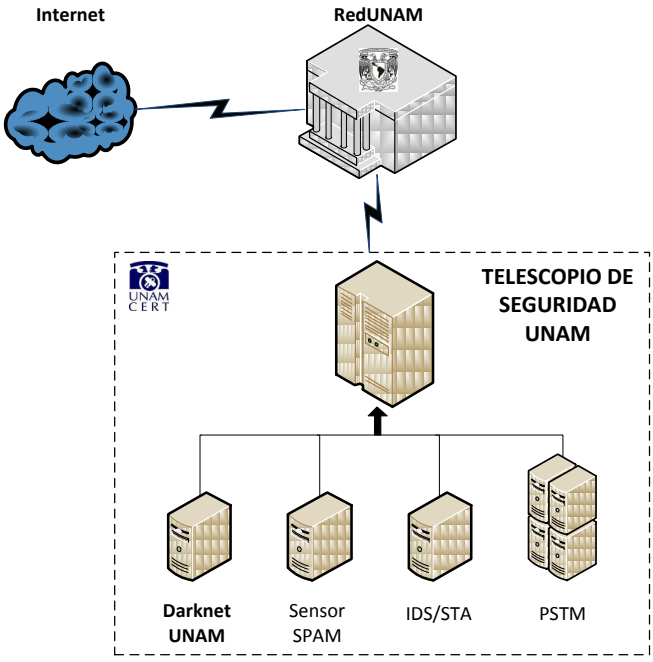


Figura 18. Esquema general del telescopio de seguridad de la UNAM

[15] Un espacio de 131,050 direcciones IPV4

La implantación de la Darknet en el TSU, lo convierte en su principal motor de generación de información debido al gran tamaño de la misma y la cantidad de información que genera.

3.3 Análisis de la infraestructura

RedUNAM consta de dos segmentos clase B, esto quiere decir que se tiene un espacio de direcciones IP públicas de aproximadamente 130,000 hosts. El número total de host en realidad es mayor si considera la gran cantidad de redes privadas con múltiples equipos dentro de ella. La tasa de transferencia de tráfico en los core[16] de RedUNAM, tiene una capacidad de 10Gbps, teniendo enlaces a 1Gbps y 100Mbps según la dependencia, instituto o facultad.

Tomando en cuenta que la Darknet utilizará miles de direcciones IP las cuales recibirán, procesarán y almacenarán el tráfico dirigido hacia ellas, se cuenta con un esquema de conectividad de gigabit ethernet[17]. Es importante señalar que dicho esquema está diseñado para funcionar de manera centralizada, es decir, se cuenta con un sistema principal que realiza las tareas de funcionamiento de la Darknet.

Una vez que la Darknet se enlace con el TSU, se logrará una correlación de eventos y un modelo de detección general sobre el tráfico entrante y saliente. Es muy importante tener presente que por el modelo de administración de la red de la Universidad, no se tiene como objetivo detectar todo evento malicioso o anómalo dentro de RedUNAM, de hecho no es posible detectar alguna amenaza específica si no se relaciona en algún momento con el tráfico de las direcciones IP de la Darknet.

3.4 Diseño de una Darknet para RedUNAM

Para poder definir las características de la Darknet-UNAM, de manera inicial se tomaron en cuenta los siguientes aspectos:

[16] Equipos principales de conectividad de RedUNAM que enrutan el tráfico entre RedUNAM y el exterior
[17] Conectividad con una tasa de transferencia aproximada de 1000 mbps, también conocida como GigaE

- Modelo de la infraestructura de RedUNAM.
- Cantidad de tráfico de red a recibir y procesar.
- Cantidad de información generada para almacenar.
- Recursos físicos de procesamiento y almacenamiento.
- Conectividad entre los servidores de la Darknet y los equipos en el UNAM-CERT.
- Tipos de tecnología a utilizar: honeypots, análisis de flujos, IDS, etc.
- Nivel de interacción y complejidad de la aplicación.
- Tecnologías utilizadas en proyectos similares.

De alguna manera cada punto influyó para definir las características técnicas, lógicas y físicas finales del diseño del motor de detección de tráfico malicioso, tomando siempre como punto fundamental el objetivo de la implementación, que es incorporar esta fuente de detección al Telescopio de Seguridad de la UNAM para mejorar la capacidad de obtención de información de incidentes de seguridad dentro y fuera de RedUNAM.

3.4.1 ALCANCE DE LA DARKNET

El alcance de la Darknet-UNAM tiene los siguientes puntos fundamentales:

- Detección de eventos de tráfico malicioso hacia y desde RedUNAM, involucrados en el espacio de monitoreo de la misma.
- Publicación de los detalles de la información a través del telescopio de seguridad de la UNAM.
- Interacción con el proceso de respuesta a incidentes dentro de RedUNAM por parte del UNAM-CERT.
- Generación de información para fines estadísticos sobre los eventos detectados.

Todo el procesamiento y almacenamiento de información se lleva a cabo en servidores dedicados y es una de las funciones fundamentales del sistema.

La importancia de la implantación de la Darknet como motor de detección de tráfico malicioso, representa la principal fuente de información para el telescopio de seguridad de la UNAM. Por este motivo, el impacto del proyecto es alto en relación a la cantidad de eventos detectados, procesados, almacenados y reportados.

3.4.2 REDIRECCIÓN DEL TRÁFICO "NO ASIGNADO"

Para el funcionamiento de la Darknet se cuenta con un esquema centralizado de recopilación de tráfico de red. Existe un servidor Darknet principal al cual es redirigido todo el tráfico que de manera global en RedUNAM no está asignado. Esto es posible gracias al apoyo por parte del departamento de operación de la red[18] de la DGTIC-UNAM, quienes han configurado todos los dispositivos de enrutamiento correspondientes para poder enviar dicho tráfico.

Debido a la cantidad de tráfico recibido, el enlace entre los dispositivos de enrutamiento y el servidor Darknet es a 1 Gbps en fibra óptica. Esto permite tener una conectividad adecuada para la cantidad de información recibida y transferida, sin embargo, como se verá más adelante, muchos de las principales consideraciones de rendimiento tienen que ver con la capacidad de procesamiento de conexiones de los equipos y con la tasa de transferencia entre el servidor Darknet y los dispositivos de almacenamiento.

La redirección configurada de todos los segmentos no asignados en los routers principales permite que el servidor Darknet reciba todas las conexiones hacia dichos segmentos en lugar de ser desechadas. Esto da la posibilidad de crear la interacción correspondiente con cada una de las conexiones para posteriormente procesar la información y almacenarla en una base de datos. La complejidad de dicha interacción es directamente proporcional con la carga que se demandará al sistema por cada conexión recibida.

[18] Departamento encargado de la configuración, instalación y mantenimiento de los enlaces y dispositivos de conectividad en RedUNAM, perteneciente a la Dirección General de Cómputo y Tecnologías de la Información y Comunicación (DGTIC).

De hecho, como parte de la redirección general, también se reciben todas las conexiones provenientes de segmentos privados que son desechados por los routers cuando se intenta rutear a cualquier segmento bogon. Esto permite detectar ciertos tipos de ataques, sin embargo dichos paquetes no son analizados por la darknet, sino por otros módulos del TSU con tecnologías IDS y de análisis de tráfico estructurado.

La figura 19 muestra la infraestructura general de la Darknet UNAM.

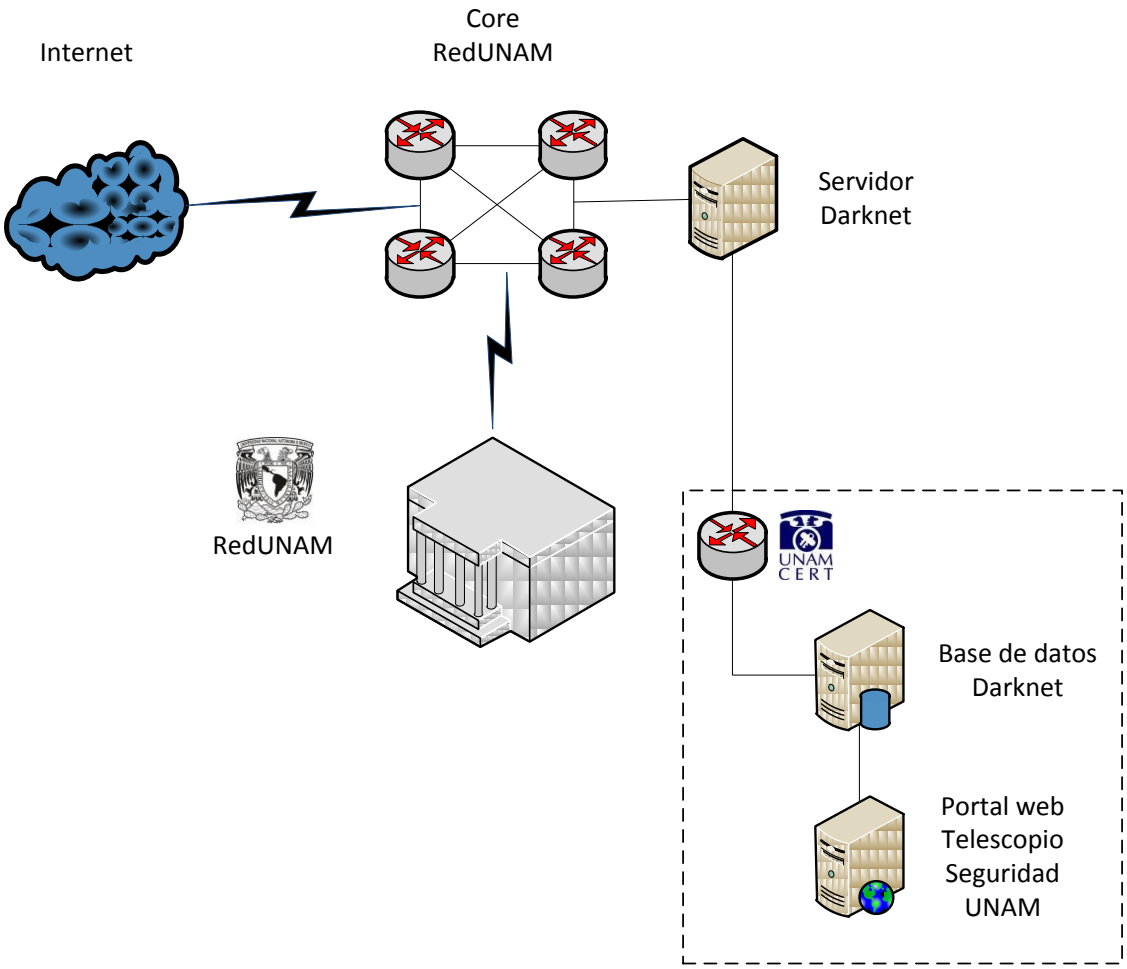


Figura 19. Esquema general de la Darknet UNAM

3.4.3 CONFIGURACIÓN DE LOS SERVIDORES

Los servidores de la Darknet tienen el rol principal de recibir el tráfico de red hacia las direcciones no utilizadas y crear una interacción. En ésta se incluyen aspectos como la

emulación de servicios, utilización de tecnologías de análisis de tráfico y detección de patrones. Para poder realizar dichas tareas, el servidor principal funciona con al menos dos interfaces de red: una para recepción de tráfico y otra para administración remota. Como ya se mencionó, una de ellas corresponde a una interfaz de fibra óptica 10/100/1000 que tiene una conexión con el router que envía el tráfico. La otra es una interfaz Ethernet UTP 10/100/1000, sin embargo como solo se utiliza para administración de sistema, la funcionalidad a 100Mbps es suficiente.

Los equipos cuentan con un sistema operativo GNU/Linux Debian mínimo y personalizado con ciertas características específicas de algunas herramientas de seguridad y propiamente del sistema desarrollado.

Es importante mencionar que en esta primera etapa del proyecto se tiene un esquema de un servidor Darknet principal. Esto se debe a motivos de pruebas, recursos y diseño general, sin embargo, para las siguientes fases de la Darknet se tienen proyectadas algunas características de cómputo distribuido para mejorar el rendimiento y poder procesar mayor cantidad de información y de manera más profunda. La tabla 10 muestra el esquema actual de los roles y características de los servidores actuales del proyecto:

Tabla 10. Esquema de roles de los servidores del proyecto

SERVIDOR PRINCIPAL	
Sistema operativo	GNU/Linux
Rol principal	Recepción y análisis del tráfico de red
Tecnologías implementadas	Honeypot, IDS, análisis de flujos
Carga aproximada del servidor	95% Recepción, análisis y procesamiento 5% Reservada para administración
Tipo	Servidor de rack multicore

SERVIDOR DE ALMACENAMIENTO	
Sistema operativo	GNU/Linux
Rol principal	Almacenamiento de información procesada
Tecnologías implementadas	Motor de base de datos, automatización de procesos de manejo de información
Carga aproximada del servidor	50% Procesos de almacenamiento 45% Consultas bajo demanda 5% Reservada para administración
Tipo	Servidor de rack multicore

SERVIDOR DE PUBLICACIÓN	
Sistema operativo	GNU/Linux
Rol principal	Sistema Web de administración y acceso a la información del TSU.
Tecnologías implementadas	Servidor Web y aplicaciones interactivas de visualización de la información
Carga aproximada del servidor	30% Procesos web 70% Disponible
Tipo	Servidor de rack multicore

3.4.4 PROCESAMIENTO DE LA INFORMACIÓN

El procesamiento de la información es una parte fundamental del proyecto. Esto implicó el análisis exhaustivo de procedimientos y algoritmos que mejor se adaptaran a al esquema del mismo.

Los aspectos fundamentales a tomar en cuenta fueron:

- Cantidad de información a analizar:
    - Cantidad aproximada de direcciones IP en el espacio de monitoreo
    - Cálculo aproximado de la cantidad de tráfico de red recibido
- Tecnologías utilizadas para detección:
    - Honeypots: tipos de herramientas y su carga de procesamiento
    - IDS: Cantidad de generación de datos de alertas
    - Flujos: Análisis estadístico y estructurado de tráfico de red
- Estructura de la información almacenada
    - Formato de la información
    - Nivel de detalle de información de incidentes detectados
    - Cantidad de evidencia almacenada
- Tipos de procesamiento
    - Tiempo real
    - Automatización programada

Debido al tamaño de la Darknet, en realidad el principal problema radica en diseñar y adaptar el mejor algoritmo para procesamiento de la información en cuestión de rendimiento y estabilidad de los sistemas. Esto quiere decir que se puede analizar una mayor cantidad de información disminuyendo la profundidad de las técnicas de análisis, sin embargo, en esta fase del proyecto se buscó encontrar un punto medio para poder realizar un análisis útil a la mayor cantidad de información posible. A grandes rasgos, en esta primera versión de la Darknet el sistema de procesamiento se adaptó a un modelo de análisis en tiempo real con procesos independientes, lo cual permitió reducir de manera drástica el tiempo que los equipos tardaban en obtener la información desde que se recibe una conexión hasta que se almacena en la base de datos.

La figura 20 muestra el esquema de procesamiento de la información por bloques principales.

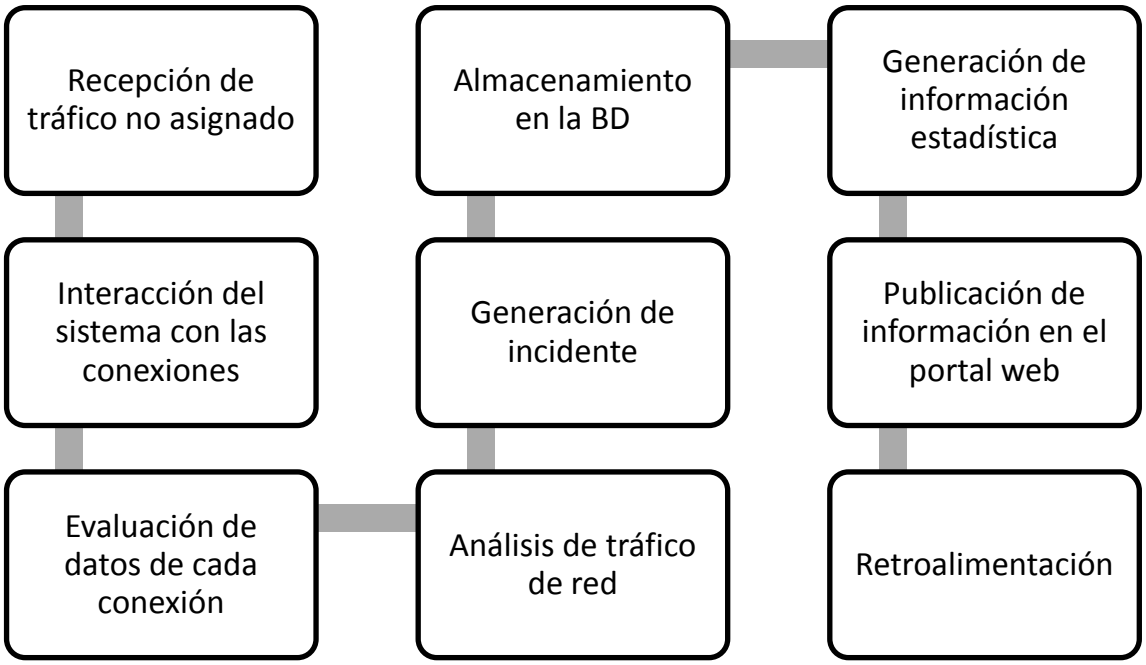


Figura 20. Procesamiento de información de la Darknet

En el siguiente capítulo se detalla este procedimiento abordando características específicas sobre el algoritmo y herramientas utilizadas.

3.4.5 OBTENCIÓN DE RESULTADOS

Para la obtención de resultados se tomó en cuenta que el sistema desarrollado cumpliera con los siguientes objetivos:

- Obtener información detallada de los incidentes generados.
  - Marcas de tiempo, protocolo, origen, destino, tipo de evento, etc.
- Capturar evidencia de tráfico malicioso.
  - Malware, binarios, flujos, etc.
- Comparar el tipo y formato de información obtenida con otros proyectos similares en el mundo.
- Reducir al máximo posible la cantidad de falsos positivos.

La correcta obtención de resultados implica que el telescopio de seguridad proporcione una fuente confiable de información sobre el tráfico malicioso detectado. Su aprovechamiento se dará cuando a partir de la interfaz final de acceso a la información (web) se obtenga de una manera estructurada todas las características antes mencionadas tanto de forma automática (sistema de atención a incidentes) y bajo demanda (consulta de información específica con fines informativos y/o estadísticos).

De esta manera, el diseño supone la capacidad de obtener información estadística y de referencia sobre datos procesados como:

- Ataques detectados
- Direcciones IP fuente de tráfico malicioso
- Tendencias de actividad anómala dentro y fuera de RedUNAM
- Interpretación de tráfico de red malicioso
- Muestras de malware
- Evidencia de tráfico malicioso (escaneos, gusanos, ataques, etc.)

Toda esta información será proporcionada con datos detallados, es decir, se podrá acceder a los eventos relacionados a un incidente proporcionando la información completa de tiempo y conectividad como protocolo, direcciones y puertos origen/destino, evidencia de posible actividad maliciosa e incluso a una muestra del malware relacionado con el evento según el tipo de actividad maliciosa. Para el caso de muestras de malware, el TSU implementará un proceso automatizado para combinarlo con una sandnet[19] (desarrollada en el proyecto UNAM-Malware[20]) y poder analizarlas.

En el quinto capítulo de esta tesis se aborda un análisis de resultados obtenidos en algunas pruebas del sistema en producción.

[19] Mecanismo de análisis automatizado de muestras de malware.
[20] Proyecto perteneciente a la SSI/UNAM-CERT http://www.malware.unam.mx encargado del análisis de malware.

3.4.6 UTILIDAD DE LA INFORMACIÓN

La información obtenida con la Darknet puede ser aprovechada en diferentes áreas:

1) Sistema de atención a incidentes (SAI) en el UNAM-CERT
    - Se obtiene un mayor detalle de información por incidente
    - Se proporciona un complemento a las fuentes de información actuales

2) Referencia en el Telescopio de Seguridad de la UNAM
    - Se le da al TSU características adicionales y le da soporte para hacer una comparativa con los pocos proyectos similares de su tipo.

3) Vinculación con organismos nacionales e internacionales para intercambio de datos sobre tráfico malicioso.
    - Fortalecer la relación con organismos enfocados a la seguridad en cómputo para compartir información, alertar y generar estadísticas.

CAPÍTULO 4

IMPLANTACIÓN DE UNA DARKNET A GRAN ESCALA EN REDUNAM

En este capítulo se aborda la estructura general de la Darknet UNAM explicando a detalle su esquema de funcionamiento ya implementado en RedUNAM, así como algunas otras características disponibles en el motor de detección. Se explica a un nivel técnico la implementación y adaptación de tecnologías honeypot, IDS y de análisis de flujos para su interacción con el sistema. Asimismo, se mencionan algunos aspectos relacionados con la efectividad y el rendimiento del sistema para la detección en una fase de producción bajo análisis.

4.1 ALCANCE DE LA IMPLEMENTACIÓN

Como se ha mencionado anteriormente, los factores principales que le dan a la Darknet UNAM su capacidad de detección son:

- Tecnologías honeypot
- Sistema de detección de intrusos (IDS)
- Análisis de flujos

El aprovechamiento de cada una de estas características permite obtener datos que al correlacionarlos en una base se genera información útil para la detección y atención a incidentes sobre tráfico de red malicioso.

El esquema actual de la Darknet y su interconexión con el Telescopio de Seguridad de la UNAM se encuentra en una fase de producción bajo análisis, lo cual significa que a pesar de que ya se obtienen datos útiles y a su vez éstos son almacenados, existen procesos por mejorar y nuevas características por implantar. En el despliegue planteado en esta tesis se cubren dos áreas fundamentales:

- Capacidad de análisis y detección de amenazas potenciales
    - Este factor corresponde fundamentalmente al desarrollo del sistema, es decir, a toda la información proporcionada por el diseño y las herramientas utilizadas.

- Análisis del potencial de una Darknet (investigación)

- Esto corresponde a todas las características que a pesar de estar disponibles, están implementadas de una manera general, lo cual quiere decir que de antemano se han tomado en cuenta posibles mejoras y métodos alternativos para su entero aprovechamiento. Las características referidas son el posible análisis de malware automatizado e interconexión con una sandnet, la detección y análisis de shellcodes, interacción con herramientas y/o mecanismos para detección de botnets y mejoras en la interacción de las herramientas honeypot.

De manera general las capacidades del motor de detección son:

- Detección de tráfico potencialmente sospechoso
  - Detección de escaneos (basado en patrones)
  - Propagación de gusanos, bots, virus (basado en firmas)
  - Ataques de fuerza bruta
  - Ataques específicos que utilicen técnicas de spoofing
  - Identificación de patrones de botnets o redes P2P
- Captura de malware
- Generación de un análisis estructurado de tráfico de red automatizado
  - Análisis de alertas de IDS
    - Información estadística
  - Análisis de flujos de red
    - Detección basada en firmas
    - Información estadística
- Fallas en la configuración de dispositivos
- Recopilación y almacenamiento de evidencias
- Análisis de payloads
- Nuevas tendencias de ataques

Como se verá posteriormente, a partir de un modelo de configuración centralizada es posible la interacción de las diferentes herramientas y tecnologías utilizadas.

4.2 ESQUEMA DE FUNCIONAMIENTO Y HERRAMIENTAS UTILIZADAS

El motor de detección consiste en la utilización de herramientas de diferentes tipos de las cuales se aprovecha su información y se hace un análisis de la misma. Asimismo, se tiene implementado un algoritmo de procesamiento el cual toma como factores fundamentales:

- Tipo de información a recopilar
- Eficiencia en el procesamiento de una gran cantidad de datos
- Establecimiento de un formato unificado de información

El punto esencial del éxito del sistema es su diseño por módulos. Esto implica que para cada una de las características existe un procedimiento específico para procesamiento de la información.

La figura 21 ilustra el diseño general del sistema con sus dos módulos principales:

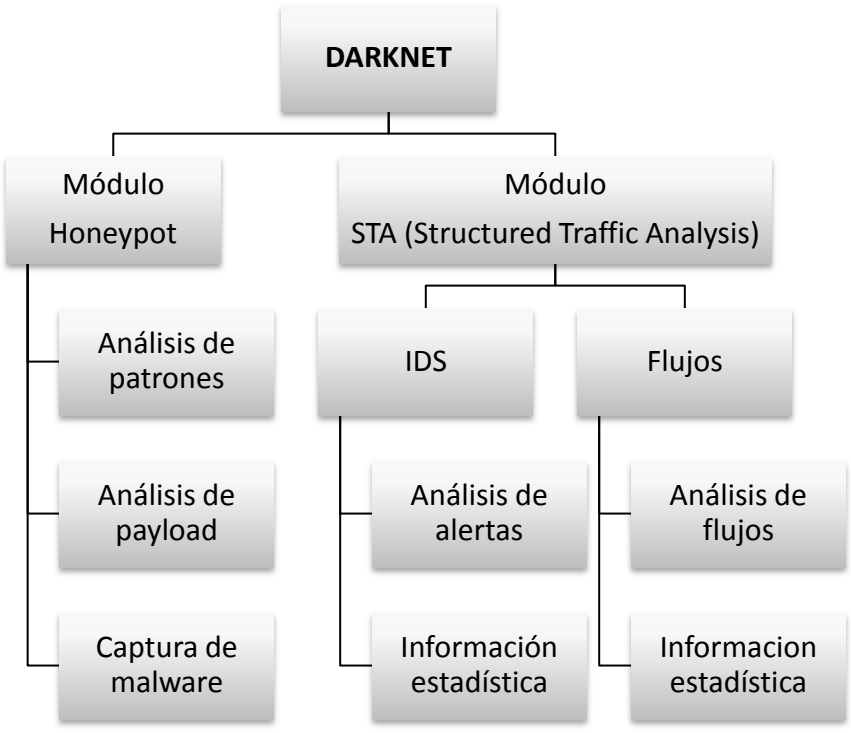


Figura 21. Esquema general del sistema de la Darknet UNAM

Dentro de cada módulo se utilizan diversas herramientas de software libre y algunas desarrolladas especialmente para el motor de detección. En los siguientes apartados se da un contexto técnico sobre las herramientas de software utilizadas.

4.2.1 HERRAMIENTAS HONEYPOT

Este módulo se encarga de procesar los datos capturados y generados por las herramientas honeypot. Básicamente se utilizan dos herramientas principales:

4.2.1.1 Honeytrap

Es un honeypot de baja interacción desarrollado por Tillmann Werner[21]. Tiene la característica principal de atender conexiones bajo demanda en cualquier puerto solicitado por un equipo origen. Una vez que la conexión es establecida, los datos transferidos (payload) son almacenados en el equipo y son posteriormente procesados por el módulo de la Darknet. Posee dos modos de funcionamiento: normal y mirror. En el primero simplemente se recibe la conexión y la emulación consiste en respuestas de saltos de líneas, es decir, se enviarán respuestas predeterminadas del tipo "\n" esperando que el equipo origen de la conexión responda a ellas. Esto implica una emulación básica pero poco práctica, ya que si el protocolo necesita de respuestas específicas no se logrará una correcta interacción, no obstante se puede dar una transferencia de datos útiles (incluso binarios) para detectar o identificar algún tipo de amenaza. Este mismo esquema tiene algunos módulos de emulación de servicios específicos que permiten la captura de malware. Por otro lado, el modo mirror consiste en responder a las conexiones entrantes con la misma respuesta que el equipo origen de la conexión devolvería en el mismo puerto. Esto tiene la desventaja de la necesidad de que el equipo origen se encuentre escuchando en el mismo puerto de lo contrario la respuesta será nula, pero la ventaja de que en caso afirmativo, la emulación es mucho más completa y con mayor probabilidad de llegar a un punto en la interacción en donde incluso se puedan identificar ciertos patrones o aspectos

[21] Miembro de The Honeynet Project en el Giraffe Chapter, especialista de seguridad en cómputo autor y coautor de varias herramientas de seguridad como honeytrap, nebula, multicap*, entre otras y enfocado al análisis de malware (reversing malware). Actualmente se desempeña como investigador en Kaspersky Labs. En 2010 asistió al Honeynet Annual Workshop organizado en México por el UNAM-Chapter y al Congreso de Seguridad en Cómputo organizado por el UNAM-CERT.

relacionados con tráfico malicioso dentro del payload. Este modo es muy práctico para la implementación de un modelo de detección de shellcodes. El esquema actual de la Darknet funciona con ambos modos, pero principalmente con el modo mirror ya que durante la etapa de pruebas mostró ser más eficiente.

La figura 22 ejemplifica los modos de funcionamiento de esta herramienta:

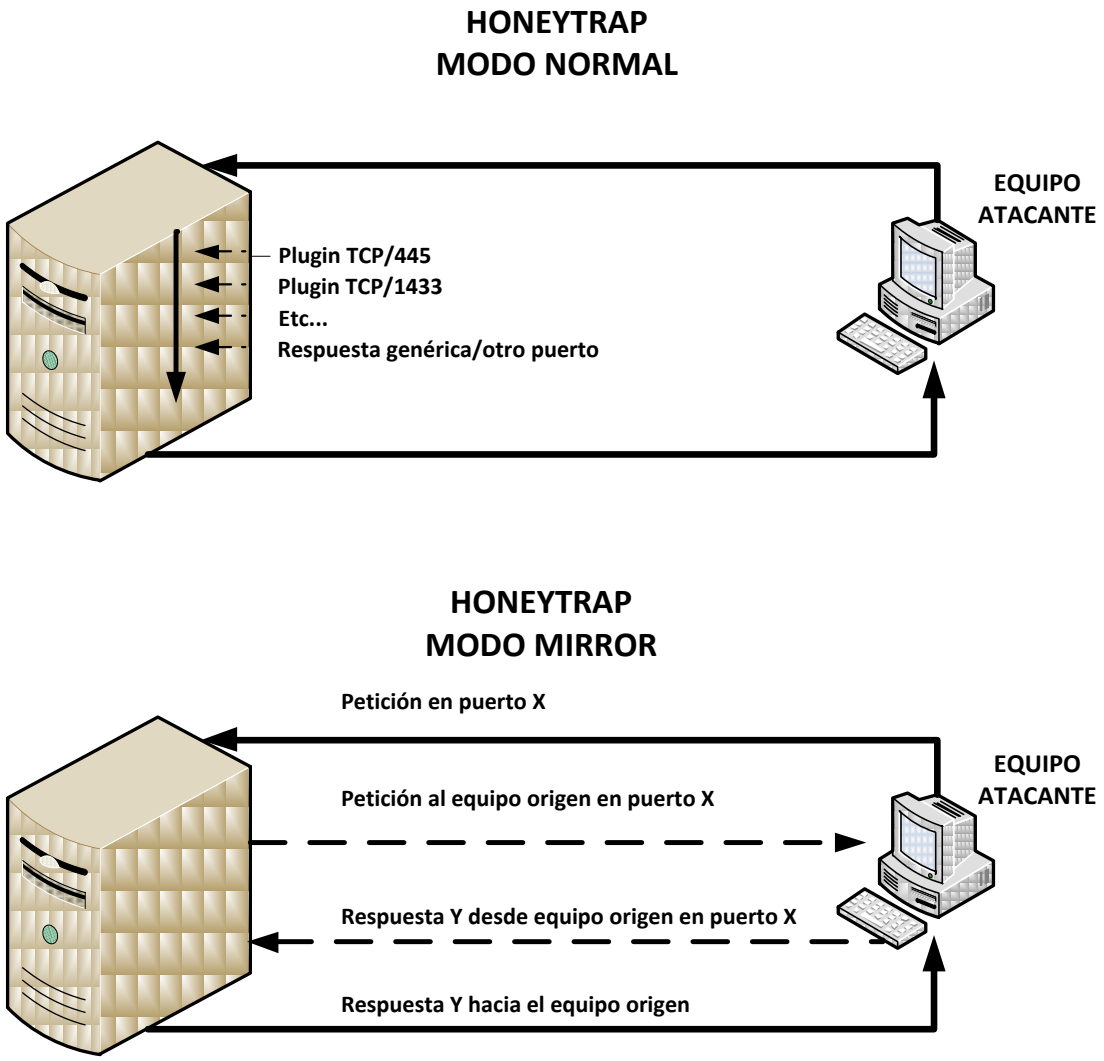


Figura 22. Modos de funcionamiento de honeytrap

4.2.1.2 Dionaea

Esta herramienta es un honeypot de baja interacción cuya principal característica es su capacidad para la captura de muestras de malware. Tiene sus antecedentes en la herramienta Nepenthes desarrollada por The Honeynet Project y cuya evolución resultó gracias a la inclusión del proyecto en Google Summer of Code (GSoC) 2009. Actualmente es soportada por algunos miembros del Giraffe Chapter como Markus Koetter y Mark Schloesser[22] entre otros.

Sus principales mejoras están en el nivel de interacción de los servicios emulados y en la capacidad de una captura de malware más confiable y en mayor cantidad. En Dionaea se reduce la cantidad de servicios emulados pero se profundiza en los que opera. En esta nueva versión se incluyeron características importantes como la capacidad de almacenar la información mediante sqlite e implementa un Shell interactivo de python. Otra de las ventajas, tomando en cuenta los objetivos del TSU, es que puede implementar el protocolo XMPP[23] para el envío de información capturada por el honeypot hacía repositorios externos de análisis como carnivore.it [43], creación de los mismos autores de esta herramienta, o hacia alguna sandbox específica.

4.2.1.3 Kippo

Esta es una herramienta honeypot para la emulación del servicio de Secure Shell (SSH). Está diseñada principalmente para interactuar con los ataques de fuerza bruta y almacenar las bitácoras. Está inspirada, pero no basada, en otra herramienta llamada Kojoney. Su principal potencial está en la emulación completa del servicio incluyendo un sistema de archivos falso y una instalación de un sistema operativo Debian 5.0. En caso de que un atacante tuviera acceso al honeypot, es capaz de almacenar todas las herramientas maliciosas descargadas para un análisis posterior.

[22] Miembro de The Honeynet Project quien también asistió al Honeynet Annual Workshop en 2010 organizada por el UNAM-Chapter y quién dio una retroalimentación para el desarrollo de algunas características del módulo de dionaea en la Darknet UNAM.

[23] Extensible Messaging and Presence Protocol, es una tecnología para la comunicación en tiempo real utilizada por varios tipos de aplicaciones.

En el portal del proyecto [26] se pueden apreciar algunos ejemplos de bitácoras generadas por kippo.

Las bitácoras de Kippo son procesadas y el payload de la conexión contiene toda la interacción que un intruso haya tenido con el Shell del sistema virtual creado por el honeypot, junto con toda la información (URL, malware, etc.)

4.2.2 HERRAMIENTAS STA (STRUCTURED TRAFFIC ANALYSIS)

Este módulo implementa una metodología de análisis de tráfico estructurado en el cual se obtiene información estadística, evidencia e interpretación basada en la organización de los datos recopilados por el IDS y las herramientas de análisis de flujos.

La correlación entre la información del IDS y el análisis de flujos es fundamental para realizar un análisis exhaustivo de determinados eventos detectados en la Darknet. Además, debido a la manera en que se organiza la información, es posible obtener pistas específicas y presentar una mejor evidencia sobre los eventos clasificados por el motor de detección.

4.2.2.1 Herramientas IDS

Este módulo se basa en la utilización del motor de detección y generación de alertas del IDS Snort. Dicha herramienta es considerada una de las mejores para la detección de tráfico malicioso basada en la definición de firmas y de un preprocesamiento del tráfico de red. En el TSU es utilizado en dos esquemas: en la Darknet como parte de un análisis estructurado de tráfico de red y como analizador de tráfico en puertos espejos (o port mirror) en el core de RedUNAM.

Para la generación de alertas, se utilizan varios conjuntos y tipos de reglas. Principalmente se utilizan las creadas por el VRT (Vulnerability Research Team de Sourcefire [50] el cual es un equipo especializado para la generación de firmas

oficiales de Snort. Este conjunto de reglas, al ser evaluadas minuciosamente antes de su liberación, supone una mayor confiabilidad y menor probabilidad de falsos positivos. El otro conjunto de firmas que se utilizan es el de Emerging Threats[24] el cual es un grupo externo a Sourcefire especializado en el desarrollo de firmas para IDS's y que también tiene un alto grado de confiabilidad. Para ambos casos, se utilizan solamente las reglas útiles para la detección de amenazas específicas tales como escaneos, malware, DoS, botnets, entre otros. La razón de esto es que por la gran cantidad de tráfico a analizar, se debe disminuir la cantidad de falsos positivos, alertas innecesarias y la carga de procesamiento del sistema.

Snort cuenta con diversos componentes como parte de su motor de detección. La figura 23 muestra de manera general el funcionamiento de este IDS.

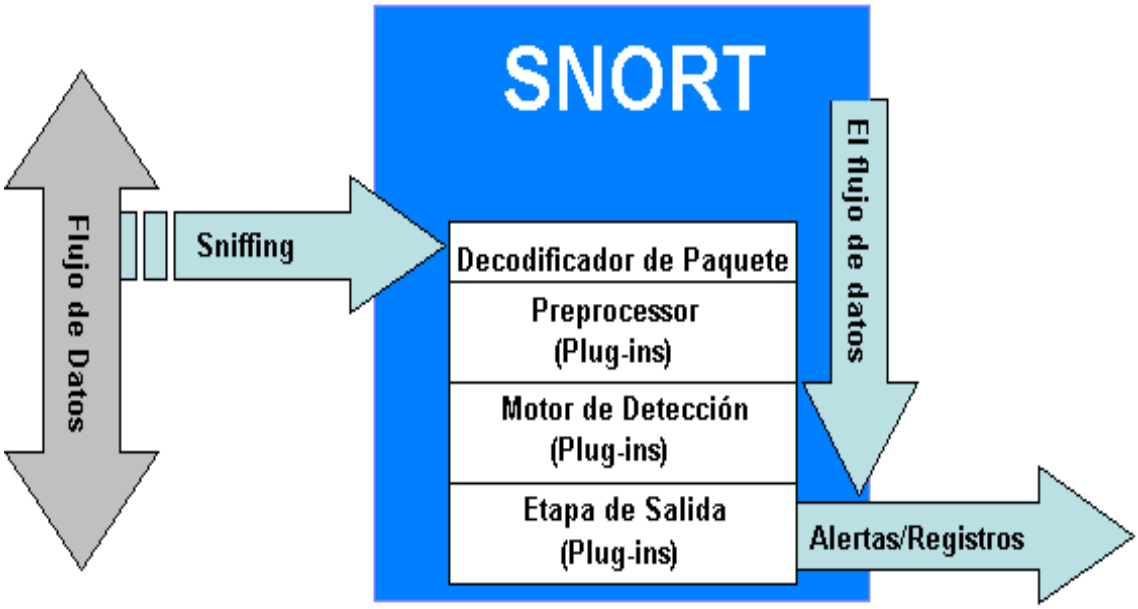


Figura 23. Diagrama de funcionamiento del IDS Snort

A pesar de que Snort tiene un propio formato de generación de alertas detectadas, toda la información obtenida es transformada a un formato único (el cual se abordará más adelante) para almacenamiento en la base de datos del TSU.

[24] http://www.emergingthreats.net/

4.2.2.2 Herramientas de análisis de flujos

Para la extracción y análisis de flujos, la Darknet utiliza una herramienta desarrollada por Qosient [2] llamada Argus. Esta herramienta es muy versátil ya que permite, a partir del tráfico de red, obtener los flujos y a su vez analizarlos en tiempo real o de manera estructurada. Como se habló anteriormente, la ventaja de obtener flujos y discriminar de manera general el contenido del tráfico, permite obtener un panorama general de toda la actividad entrante y saliente de comunicación entre equipos en la red monitoreada. Aplicando este concepto a la infraestructura de la Darknet, permite hacer una detección de patrones para poder ser interpretados y así determinar posibles anomalías o amenazas de seguridad, por ejemplo, visualizar demasiada actividad en determinados puertos utilizados por malware o por servicios comúnmente atacados.

Argus consta de dos partes fundamentales:

- Servidor Argus

  Permite obtener los flujos a partir del tráfico de red. Define varios modos de salida de datos, ya sea mediante un socket o hacia un archivo binario que puede ser leído por el cliente de Argus. El propósito de escuchar en un socket, es poder visualizar los flujos en tiempo real.

- Cliente Argus

  Consta de una serie de herramientas *'ra tools'* que permiten visualizar los flujos obtenidos por el servidor Argus. Estas herramientas especializadas permiten no solo visualizar la información sino procesarla, filtrarla, extraer contenido específico, ordenarla, entre otros.

Argus, al transformar a un formato binario la información recopilada en el tráfico de red, permite un manejo eficiente de los datos de flujos, y gracias a las características del cliente, se puede implementar fácilmente en procesos automatizados de manejo de información.

La figura 24 muestra el proceso de interacción entre el cliente y el servidor de Argus.

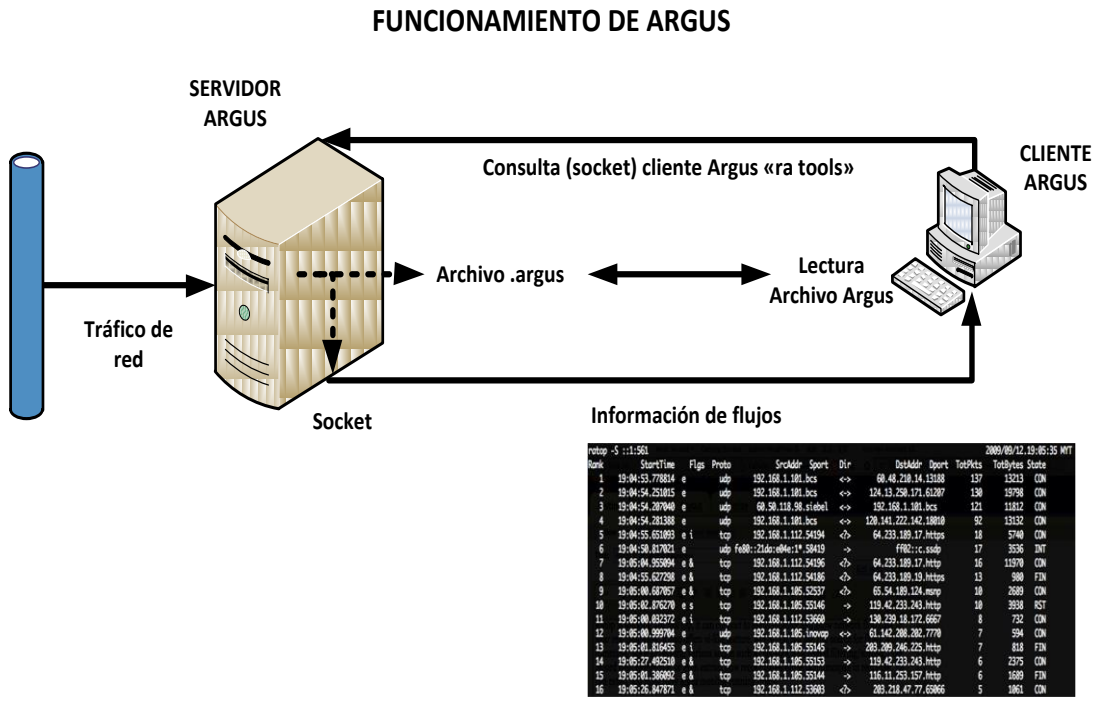


Figura 24. Funcionamiento de Argus

4.2.3 HERRAMIENTAS GENERALES

La Darknet realiza un análisis a cada uno de los payloads de las conexiones recibidas por las herramientas honeypot. Este proceso involucra la búsqueda de determinados patrones que comúnmente sugieren actividad maliciosa o que de alguna manera pueden dar una pista de ella. Los parámetros buscados y procesados son:

- URL's: La búsqueda de URL permite una identificación de sitios maliciosos desde los cuales posiblemente se descargue algún tipo de malware o den una pista del origen del tráfico malicioso.
- Dominios: Al igual que las URL's, los dominios permiten obtener pistas del origen de posible actividad maliciosa, así como estadísticas de sitios atacados.
- Correos electrónicos: La extracción de direcciones de correo electrónico permite la identificación de amenazas como spam. Esto a su vez facilita la clasificación de la naturaleza del tráfico, es decir, posiblemente spam, phishing, scam, etc.

- Direcciones IP: Al igual que las anteriores, permite identificar el origen de tráfico malicioso y obtener una base para el seguimiento del tráfico malicioso con tecnologías honeypot.

Para poder lograr lo anterior, se han creado diversos submódulos de procesamiento basados en el lenguaje de programación Perl para analizar los payloads recibidos específicamente por honeytrap. La finalidad fundamental es la detección de amenazas mediante búsqueda de patrones.

4.3 INTEGRACIÓN Y FUNCIONAMIENTO DE MÓDULOS

La parte fundamental de este sistema es el procesamiento de la información. Esto es debido a que independientemente de la capacidad de captura de malware, emulación de servicios, captura de payloads, etc. debe clasificarse y organizarse de manera que pueda ser interpretada de la mejor manera por una analista de red, además, algunas de las características de detección no podrían aprovecharse sin un previo procesamiento, tal es el caso de los patrones de escaneos, ataques DoS, etc.

Este módulo principal del sistema está desarrollado en el lenguaje de programación Perl, combinando algunas características del uso del Shell del sistema GNU/Linux.

En realidad, a pesar de que varios módulos en el sistema corresponden a herramientas honeypot, IDS, flujos, y de otro tipo como bases de datos, parámetros del sistema, redirección del tráfico, etc. todos están estrechamente relacionados, ya que la salida de unas es la entrada de otras. Esto parece trivial en un esquema de procesamiento de información, sin embargo, en la infraestructura de la Darknet se tiene un diseño de tal manera que independientemente del tipo de herramienta, la información pueda ser procesada con un formato de salida unificado. Esto quiere decir que en un desarrollo posterior se pueden agregar otros módulos de recopilación o captura de datos de manera sencilla y la información extraída será compatible con el formato de almacenamiento del TSU.

La integración de las herramientas se basa en el modelo de la figura 25.

Figura 25. Organización de herramientas en módulos de la Darknet

A continuación, la figura 26 presenta el diagrama principal correspondiente al proceso de captura, clasificación, procesamiento y almacenamiento de los datos que fluyen en el motor de detección:

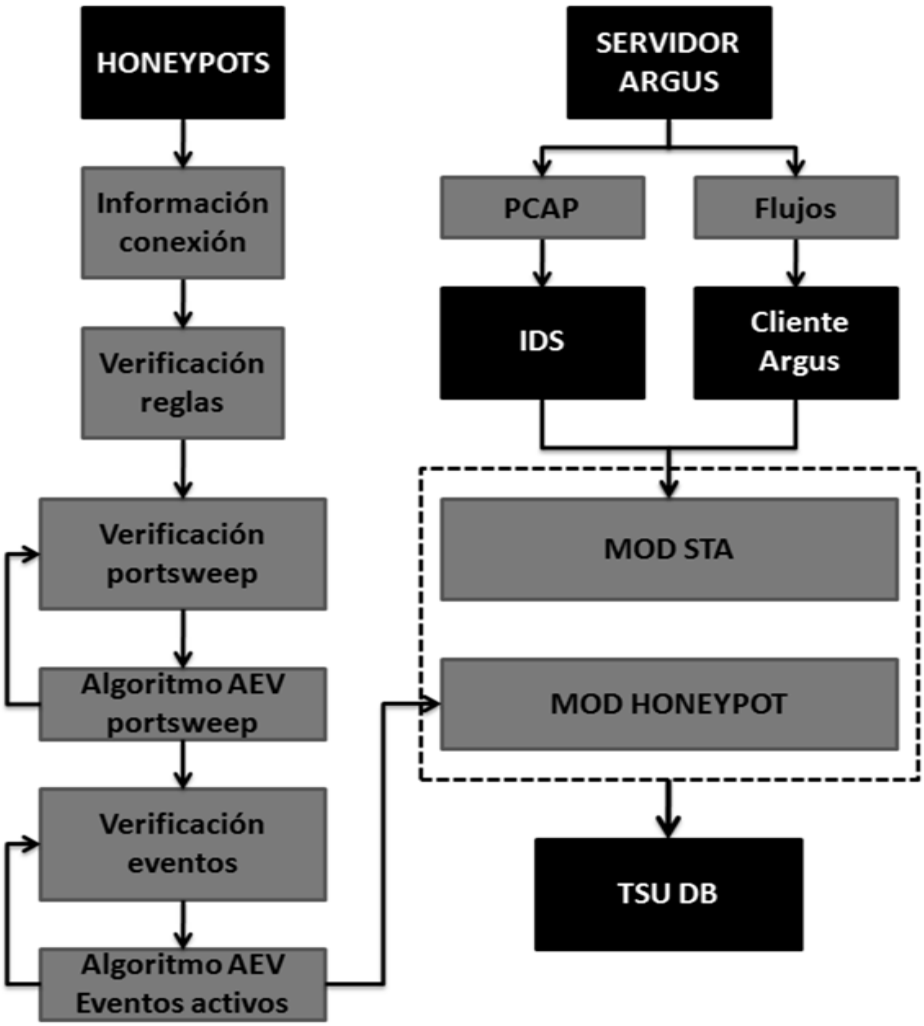


Figura 26. Proceso de manejo de información de la Darknet UNAM

4.3.1 CAPTURA Y RECOPILACIÓN DE DATOS

La captura de datos corresponde al primer submódulo del sistema. Para este procedimiento se utilizan diversas herramientas mostradas y explicadas anteriormente. En esta primera versión de la Darknet se tienen tres fuentes principales de captura de datos de entrada. La función específica de estas herramientas es:

- Honeytrap:

  Recepción de conexiones bajo demanda, captura mínima de malware y almacenamiento de payloads. Esta herramienta se comunica directamente con el módulo de la DKN enviándole la información de conectividad de cada petición y almacenando el payload con un nombre identificador.
- Dionaea

  Captura de muestras de malware y almacenamiento de bitácoras de conexiones. La información de conectividad es enviada también al módulo DKN para ser procesada y almacenada.
- Snort

  Captura de tráfico de red relacionado con alertas detectadas. Snort se ejecuta en modo demonio para poder analizar el tráfico. Los datos recopilados son tomados por el módulo DKN para realizar un procesamiento en tiempos determinados.

4.3.2 FUNCIONAMIENTO MÓDULO HONEYPOT

Este módulo procesa toda la información de las herramientas honeypot y hace un análisis automatizada de la misma. Consta de tres submódulos principales:

- DKN connection: A partir de las características del evento, lo clasifica y lo agrega a un incidente específico. Toda la información es enviada al DKN agent.
- DKN agent: Procesa los payloads de los incidentes y mantiene un orden de ejecución. Utiliza al módulo DKN store para almacenar la información procesada.

- DKN store: Almacena la información del incidente en la base de datos y en un formato de salida unificado, junto con su payload y evidencia relacionada.

A grandes rasgos, el funcionamiento de este módulo se basa en el modelo de la figura 27:

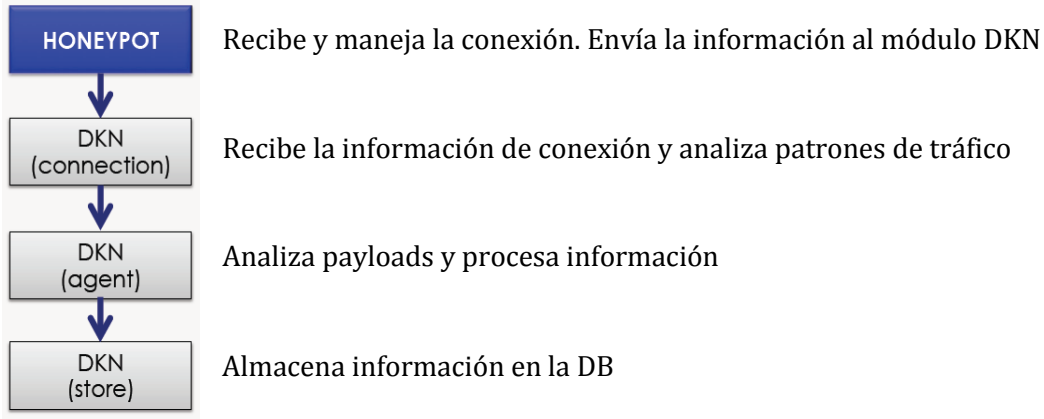


Figura 27. Diagrama del módulo honeypot

4.3.2.1 Desarrollo y adaptación de herramientas

Para poder acoplar de la mejor manera las diversas herramientas, fue necesario realizar algunos cambios en las mismas. Específicamente, fue Honeytrap la herramienta a la que se hicieron algunas modificaciones en su código (lenguaje C) para poder implementarlas de manera "ad-hoc" en el diseño de la Darknet; a las otras simplemente se les configuró de manera muy específica para que pudieran lograr una interacción con el módulo DKN.

Por sí mismo Honeytrap almacena la información capturada, sin embargo, implementarla de forma normal en la Darknet UNAM implica varios problemas relacionados con el rendimiento. Una vez que los datos de las conexiones y payloads originalmente son almacenados en un archivo de log, éstos tienen que ser extraídos por un *parser* el cual tomaría demasiado tiempo procesando los datos tomando en cuenta que son decenas de GB de información al día.

Analizando las diferentes opciones, se buscó hacer un algoritmo que permitiera procesar los datos en tiempo real y con un rendimiento que fuera lo suficientemente eficiente para poder manejar la gran cantidad de datos de la Darknet. Para esto, primeramente se estableció el mecanismo para que honeytrap enviara la información directamente al módulo DKN sin tener que pasar por un archivo de log. La figura 28 explica gráficamente este proceso.

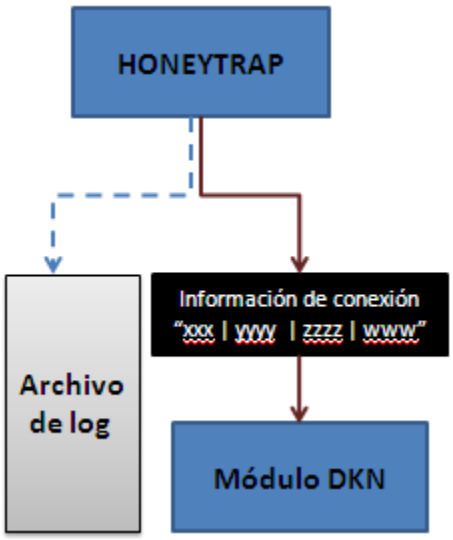


Figura 28. Adaptación de honeytrap para implementación en Darknet UNAM

Como se puede interpretar en la figura, la modificación a honeytrap incluye establecer un formato de entrada específico para que pueda ser procesado por el módulo DKN. Este formato es el que precisamente puede acoplar cualquier herramienta con la finalidad de que sea compatible con el módulo DKN y por consiguiente con el TSU.

En términos técnicos, honeytrap ejecuta una llamada al sistema que a su vez invoca al módulo DKN pasándole como argumento la información de conectividad del evento. Al hacer esto, cada que se recibe una conexión se ejecuta un nuevo proceso hijo de Honeytrap que invoca al módulo DKN, aprovechando todo el tiempo de procesamiento para todas las conexiones recibidas en lugar de ejecutar un parser sobre un archivo de varios GB de información. Esto incrementa el nivel de rendimiento de la Darknet y cumple con el objetivo inicial de poder procesar toda la información recibida. La parte de procesamiento y análisis de payloads también

maneja un concepto parecido, sin embargo es independiente ya que esta fase solo corresponde a la recepción de datos de entrada para poder ser procesados en profundidad.

4.3.2.2 Clasificación de la información

La clasificación de los eventos detectados en la Darknet se realiza tomando criterios predefinidos a partir de los cuales el módulo de procesamiento hace una identificación comparando las características del tráfico de red.

4.3.2.2.1 Detección por reglas

El modulo DKN tiene la capacidad de predefinir reglas de detección de manera dinámica. Estas reglas funcionan de manera parecida a las de cualquier IDS, sin embargo solamente necesitan de la información general del tráfico de red para poder catalogarlo.

De esta manera, cada que el módulo DKN recibe un evento desde las herramientas honeypot, verifica si las características del tráfico corresponden a las de un patrón definido en las reglas. En caso afirmativo el evento será catalogado, de lo contrario lo definirá como un evento general.

La sintaxis para las reglas del módulo DKN es:

DESCRIPCION | PROTOCOLO | PUERTO DESTINO | PATRON | TIEMPO EVENTO

Cada campo corresponde a:

- *DESCRIPCION: Nombre del evento*
- *PROTOCOLO: Protocolo del flujo recibido*
- *PUERTO DESTINO: Puerto del equipo destino.*
- *PATRON: Cadenas o patrón buscado en el payload de la conexión. Sobre este patrón hará un conteo por cada coincidencia encontrada.*
- *TIEMPO EVENTO: Umbral de tiempo en que un evento se considerará parte del mismo incidente.*

Así, un ejemplo de conjunto de reglas sería:

SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK|TCP|22|-|300
SQL WORM 1433|TCP|1433|-|300
IRC CHAT-POSIBLE BOT|TCP|6666|PING|300
IRC CHAT-POSIBLE BOT|TCP|6667|PONG|300
POSIBLE WORM MS-DS 445|TCP|445|-|300

Interpretando algunas de las reglas del ejemplo anterior, todos los paquetes que se transmitan bajo el protocolo TCP hacia al puerto 22, serán catalogados como un posible escaneo SSH o ataque de fuerza bruta. Tomando en cuenta la naturaleza de la Darknet, recibir paquetes hacia este puerto implicaría una muy baja probabilidad de que se tratara de un falso positivo. Asimismo, para el ejemplo de un evento de IRC-CHAT o Posible bot, cualquier paquete TCP hacia el puerto 6666 se catalogaría como tal, se buscaría en el payload la cadena "PING" (debido a que esta cadena es parte del protocolo IRC) y se haría un conteo del número de coincidencias encontradas.

Todo este conjunto de reglas es almacenado en un archivo de texto el cual es leído por el módulo DKN y es definido en un archivo de configuración principal del sistema.

La definición de reglas es un mecanismo eficiente de clasificación de eventos ya que al igual que las firmas de un IDS, mientras mejor se especifiquen las características buscadas en los paquetes de red, mejor organizados y más certeros serán los eventos detectados.

4.3.2.2.2 Detección por patrones

La segunda característica para clasificar la información de eventos recibidos es la detección por medio de patrones. Este modelo de detección se basa en un algoritmo propio que analiza no solo las características del tráfico de red, sino el comportamiento de los paquetes recibidos.

Su objetivo principal es identificar ataques comunes como escaneos y poder determinar de qué tipo son, por ejemplo barrido de puertos, escaneo de puertos específicos, etc. La implementación de este modelo es fundamental para poder organizar mejor los incidentes detectados y minimizar de manera estructurada y funcional la información procesada.

Para ejemplificar la utilidad de esta característica, imagínese un barrido de puertos: el equipo origen envía miles de paquetes a un mismo equipo recorriendo todos los puertos posibles (65535). Si solamente se recibe cada paquete y se contabiliza, tendríamos miles de incidentes reportados cuando en realidad a pesar de ser independientes en conectividad puesto que van a puertos distintos, todos pertenecen al mismo incidente. Un caso similar sería cuando un equipo origen envía un escaneo de puertos; si envía paquetes a todos los equipos de uno o varios segmentos buscando desde un mismo puerto origen el servicio de SSH (TCP/22), entonces no levantará un incidente por cada paquete, sino que lo catalogará dentro de uno mismo. El módulo es capaz de identificar este tipo de situaciones.

Los siguientes diagramas de la figura 29 ejemplifican los casos anteriores:

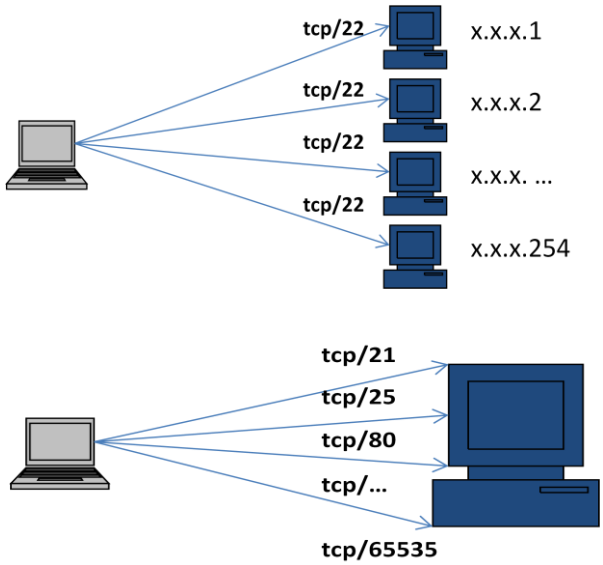


Figura 29. Patrones identificados por el módulo DKN

Los patrones identificables por el módulo están previamente definidos mediante un algoritmo que toma en cuenta el factor de rendimiento del sistema. De manera general todos los eventos se van almacenando en un archivo. Cuando este archivo es leído y vaciado para poder recopilar más eventos, el lapso que tarda en realizarlo implica que se deja de capturar eventos en ese instante. En el esquema de la Darknet se reciben miles de eventos por segundo, por lo que aunque sea de milisegundos el lapso en que el sistema tarda en hacer la lectura del archivo de eventos recibidos, se tiene una

omisión de decenas o centenas de ellos, lo cual implica una pérdida considerable si se extrapola al lapso de una hora o un día. Para resolverlo, el algoritmo se apoya de buffers temporales iterativos que permite la captura de paquetes en todo momento. El diagrama de la figura 30 explica el algoritmo implementado:

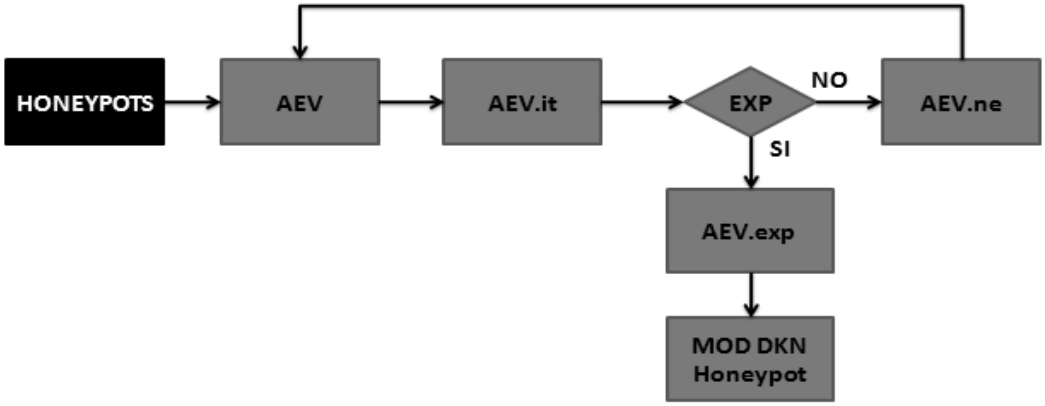


Figura 30. Algoritmo expiración y manejo de cola de eventos

1) *El honeypot alimenta constantemente un archivo de eventos.*
2) *El agente DKN renombra el archivo original tomándolo como el AEV (Active Events) de la iteración actual (AEV.it) y se genera un nuevo AEV que sigue siendo llenado por los honeypots mientras se verifican los eventos actuales.*
3) *Si los eventos han expirado se guardan en otro archivo (AEV.exp) y si no se llevan a un buffer que se volcará nuevamente en el archivo original AEV que continúa siendo llenado por el honeypot. La expiración es la clave del algoritmo ya que mientras no hayan expirado, según el umbral definido en las reglas, los eventos con las mismas características pertenecerán al mismo incidente.*
4) *Todos los eventos expirados son enviados al módulo honeypot para su análisis.*

En las pruebas realizadas el algoritmo funciona de manera eficiente. Hasta el momento no se han presentado casos de pérdidas u omisiones en el conteo de paquetes, pues uno de los mecanismos de control es verificar las bitácoras de cada una de las herramientas de tal manera que el número total de paquetes recibidos por las mismas concuerde con el número de paquetes procesados. Entonces, en cuanto a efectividad y rendimiento, este módulo se considera funcional.

4.3.2.3 Análisis de payloads

Una vez que la información ha sido clasificada según una regla o patrón identificado, el incidente abierto recibirá eventos relacionados hasta que expire. Este tiempo está definido en cada regla y como parámetro general en el archivo de configuración para todos los eventos que no hayan sido predefinidos específicamente.

El agente DKN se ejecuta de manera cíclica revisando de manera constante los incidentes que han expirado. Cuando en una iteración de revisión encuentra alguno de ellos, ejecuta un nuevo proceso hijo para analizar el payload relacionado a la conexión. Al igual que la interacción entre las herramientas honeypot, la ejecución de un proceso hijo resulta más eficiente y determinados casos necesario debido a que si el análisis a cada payload fuera secuencial, el sistema generaría colas demasiado largas para el caso de análisis de payloads o conjuntos de payloads grandes. En cambio, al ser procesos independientes, el sistema puede seguir trabajando aún cuando un proceso tarde demasiado tiempo o supere el umbral entre una iteración de revisión por el agente DKN y la siguiente.

Como se mencionó anteriormente, el análisis que se hace a los payloads incluye la búsqueda de URL's, dominios, direcciones IP, correos electrónicos y patrones específicamente definidos en las reglas de detección del módulo DKN. Durante este análisis se hace un conteo de cada concordancia y los datos son ordenados de manera que puedan ser interpretados posteriormente por el analista de una manera sencilla.

4.3.2.4 Almacenamiento de la información

Una vez que el módulo DKN ha procesado un incidente, la información está lista para ser almacenada. Existen dos formatos de salida en que el módulo DKN puede generar la información final procesada.

4.3.2.4.1 Formato unificado TSU

Este es un formato de almacenamiento en archivos de texto plano con una sintaxis predefinida para que dicho archivo pueda ser almacenado en el TSU.

Esta sintaxis es un formato de pipes (X|Y|Z|…|…) con campos en un orden específico que tiene tres aspectos fundamentales:

- *Información general de cada evento*

  En los primeros campos de datos se almacena información como marcas de tiempo (timestamp inicial y final), protocolo, IP origen, tipo de incidente, etc.
- *Ruta a un archivo de detalles*

  En el penúltimo campo se hace referencia a la ruta de un archivo que contiene los detalles de todos los eventos relacionados al incidente. Como se mencionó anteriormente, cada incidente puede ser una relación uno a muchos, es decir, un incidente puede tener uno o muchos eventos. Este archivo de detalles contiene la bitácora detallada de cada evento detectado proporcionando cada aspecto analizado en la conexión y en su payload con el siguiente formato:

  timestamp|srcip|sport|dstip|dstport|correos|url|ip's|patrón

  Los campos de los patrones de búsqueda contienen a su vez el número de concordancias encontradas de la forma:

  …|url1(X),url2(Y)|ip1(Z),ipN(W)|patronregla(N)

  De esta manera, es fácilmente identificable alguna anomalía o aspecto sospechoso de actividad maliciosa.

- *Ruta a un archivo de payloads*

  En el último campo de datos se hace referencia a un archivo empaquetado en el formato .tar.gz que contiene todos los payloads en formato binario o texto que se hayan obtenido durante la conexión. En casos específicos estos binarios corresponden a malware capturado por la herramienta honeypot.

Para cada uno de los aspectos mencionados se genera un archivo con la información referida, por lo que en el proceso final se generan 3 archivos. En el anexo C se muestra en ejemplo de la información obtenida para un incidente.

4.3.2.4.2 Almacenamiento directo en base de datos (postgresql)

El segundo modo de almacenamiento es mediante la inserción de información en una base de datos. Este modo es más eficiente y estructurado debido a que los datos se encuentran distribuidos en tablas y las consultas específicas de información pueden ser obtenidas mediante sintaxis SQL.

El esquema de la base de datos se diseñó específicamente para poder obtener cualquier tipo de información de mejor manera posible en cuanto rendimiento, estructura y aprovechamiento de la información.

El módulo DKN puede insertar directamente en la base de datos todos los incidentes procesados incluyendo los payloads binarios relacionados. Toda esta información puede ser aprovechada mediante un sistema web que haga consultas sobre los incidentes relacionados.

La inserción directa tiene el inconveniente de que en entornos de mediana o gran escala representa una carga adicional de procesamiento, y tomando en cuenta que pueden ser cientos de incidentes por segundo se vuelve un problema significativo para el rendimiento general del sistema. Por esta razón, en el TSU los servidores de la Darknet envían toda la información recopilada hacía el servidor de base de datos pero en formato unificado. Esto solo representa transferencia de datos mediante un canal seguro SSH, sin la necesidad de hacer cientos de conexiones, consultas y transferencias por cada incidente. Entonces, cuando el servidor de base de datos recibe la información, la inserta en sí misma adoptando la carga de procesamiento que tendría originalmente el servidor de la Darknet y a su vez, evitando tráfico de red de consultas e inserción de datos. En ambientes pequeños, por ejemplo sensores Darknet

con pocas direcciones IP (decenas o cientos), la inserción directa es adecuada y funcional.

4.3.3 FUNCIONAMIENTO MÓDULO STA

El segundo módulo principal del sistema realiza un análisis estructurado del tráfico de red. Se denomina análisis de tráfico estructurado porque la información que se obtiene parte de cuatro bases principales las cuales persiguen objetivos específicos que permiten un estudio detallado y facilitan el seguimiento de una investigación en el tráfico de red.

A continuación en la tabla 11 se detallan las características de cada factor:

Tabla 11. Factores del análisis de tráfico estructurado

Factor	Descripción
Datos de contenido completo	Corresponden a las capturas completas del tráfico de red. En este caso se utilizan herramientas como los analizadores de protocolos o sniffers (tcpdump y snort)
Datos de sesión	Obtienen solamente la información de las sesiones, que en el caso práctico corresponde a los flujos del tráfico de red. Se utiliza Argus para realizar el análisis.
Datos de alertas	Obtiene la información relacionada con las alertas del IDS. Esto complementa las anomalías interpretadas en el análisis de flujo y da una base para una investigación a fondo. Se utiliza Snort.
Datos estadísticos	Conjunta la información de ambos análisis para poder crear referencias y la interpretación final por parte del analista. La utilización de datos estadísticos es uno de los mecanismos de detección de patrones de tráfico malicioso y de identificación de falsos positivos.

A pesar de no seguir una metodología estándar, es una técnica utilizada comúnmente en el análisis de tráfico de red cuyo principal objetivo es reducir tiempos analizando de lo general a lo particular. En [21] se puede consultar un artículo técnico relacionado con el análisis estructurado de tráfico de red. El esquema de funcionamiento de este módulo se representa en la figura 31:

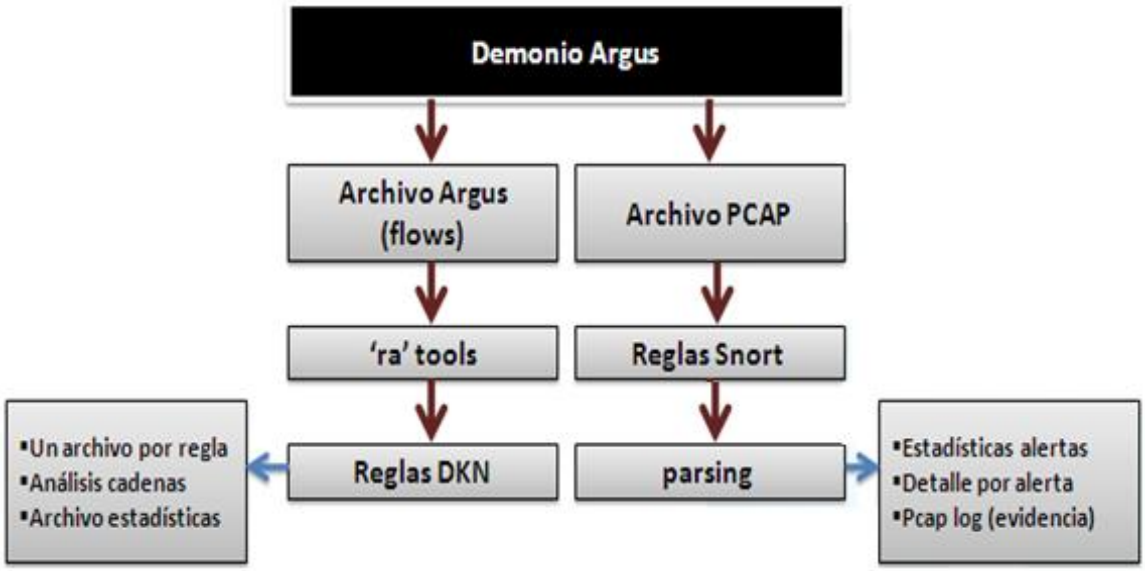


Figura 31. Esquema de funcionamiento módulo STA

4.3.3.1 Análisis de flujos

Argus permite obtener información sobre los flujos del tráfico de red, analizando sus características e incluso buscando patrones específicos tanto en los datos de conectividad como en el contenido del payload.

El análisis a los flujos que realiza el módulo DKN incluye dos partes fundamentales:

- Datos estadísticos

  Genera tablas involucrando parámetros de interés como hosts más comunes, puertos más atacados, uso por protocolo. Para cada incidente genera un archivo extrayendo todas las sesiones involucradas ordenadas de manera descendente lo cual da una pista al analista sobre la cantidad y tipo de actividad de tráfico por cada incidente. Adicionalmente, genera otros archivos con la información sobre

cantidad de bytes/paquetes transferidos, estadísticas de uso por protocolo, actividad entre hosts y conteo general de las sesiones.

- Datos de payload

  Tomando como referencia los patrones de búsqueda definidos en las reglas del módulo DKN, Argus hace una búsqueda de dichos patrones en el payload de los paquetes y lo pone en contexto con el flujo. Esta característica es fundamental para la identificación de amenazas específicas como ataques de fuerza bruta, conversaciones de bots, etc. y a su vez proporciona la evidencia para el análisis forense.

En el Anexo D se muestra un ejemplo del análisis de flujos de un incidente.

4.3.3.2 Detección por IDS

El submódulo de procesamiento de información de alertas obtenidas por Snort, aprovecha toda la información generada y la transforma en el formato unificado del TSU para que pueda ser ingresada en la base de datos.

Como parte del análisis estructurado, también obtiene información estadística general del tráfico de red tomando como parámetros:

- Top de alertas
- Top de direcciones IP origen
- Top de direcciones IP destino
- Top de puertos origen
- Top de puertos destino
- Top de clasificación de alertas

Asimismo, genera un archivo por cada alerta detectada en donde incluye la información de los equipos relacionados indicando los puertos de la conexión. Esto es útil para poder realizar una interpretación y búsqueda específica de equipos relacionados con posible actividad maliciosa, así como para detectar posibles falsos positivos.

Todas las firmas de detección del IDS son independientes a las reglas definidas para el módulo DKN. Las firmas de Snort permiten detección de amenazas, anomalías, patrones, a partir de la verificación directa del tráfico de red o como en este caso a partir de archivos de captura, mientras que las reglas DKN permiten una clasificación y detección a partir de la información proporcionada por herramientas y de sus payloads correspondientes.

En el Anexo E se muestra en ejemplo de la información generada por el submódulo.

4.4 RENDIMIENTO GENERAL DEL SISTEMA

Debido al tamaño de la Darknet, el rendimiento fue uno de los principales factores a tomar en cuenta en el diseño y desarrollo de la infraestructura y del sistema de captura, procesamiento y almacenamiento de información.

Actualmente, la Darknet no se encuentra trabajando al 100% de su capacidad ya que los servidores con los que se cuenta no son suficientes para poder procesar todas las tareas necesarias. El potencial aproximado de direcciones IP como espacio de monitoreo de la Darknet oscila alrededor de 50,000. En un principio, varios diseños del esquema de manejo de la información no sólo resultaban imprácticos, sino que era imposible procesar toda la información recibida. Solo por mencionar un ejemplo, el diseño original con simples scripts extrayendo información con grep, awk y consultas a servidores externos, tardaba más de 36 hrs en procesar menos del 10% de información de un día. El problema radicaba en que era un procesamiento secuencial y se hacía por lapsos, lo cual generaba una cola interminable y representaba desperdiciar el tiempo entre cada momento de la ejecución.

El diseño del módulo DKN está enfocado a procesar información a un nivel de búsqueda de patrones y revisión de las características del tráfico. El diseño modular resulta útil ya que en el momento de implementar nuevas características se puede repartir las tareas de manera más sencilla, es decir, tener servidores dedicados para determinadas tareas que a su vez puedan repartirse la carga de datos.

En general, el rendimiento tuvo considerables mejoras, sin embargo aún hay áreas en las que se debe trabajar.

CAPÍTULO 5

ANÁLISIS DE RESULTADOS

La etapa de pruebas de la Darknet se realizó con aproximadamente el 30% de la capacidad de la misma, esto quiere decir aproximadamente 15,000 direcciones IP.

Las características del motor de detección se fueron implementando y activando paulatinamente, midiendo la carga de procesamiento, estabilidad, tiempos y efectividad de detección. El siguiente esquema muestra las etapas de la fase de prueba:

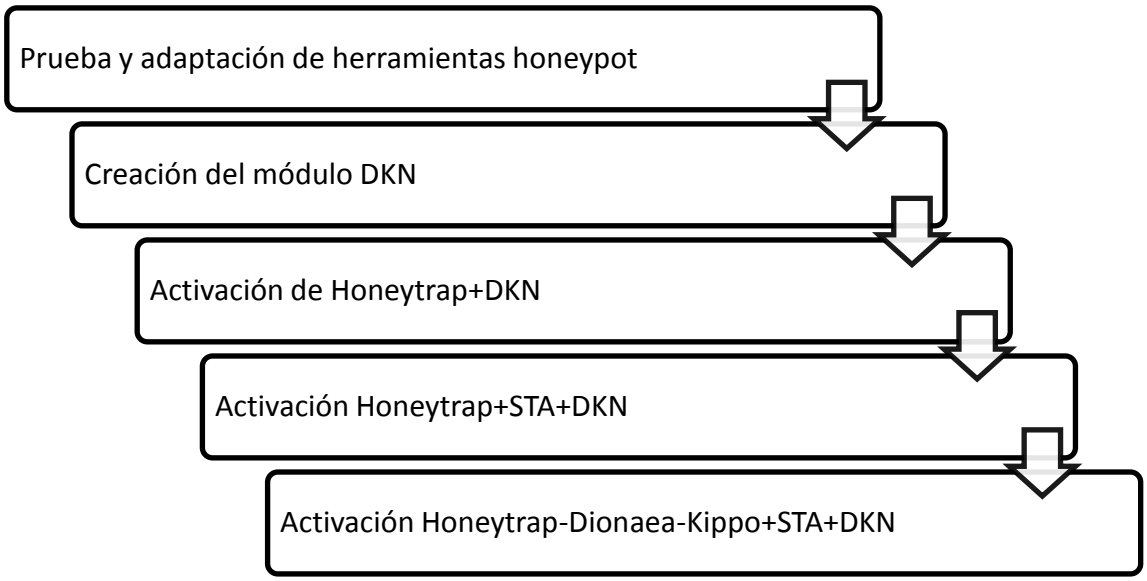


Figura 32. Etapas de la fase de prueba de la Darknet

El análisis de los resultados se basa en la medición de tres aspectos:

- Carga del sistema
- Efectividad de detección y clasificación
- Utilidad de la información

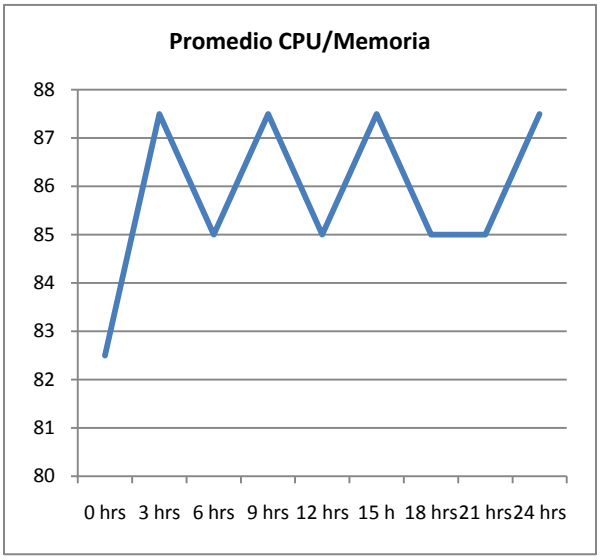
5.1 CARGA DEL SISTEMA

Una vez que se activaron todas las características, y tomando aproximadamente el 30% de las direcciones IP disponibles para la Darknet, el sistema llega a un máximo de 90% de utilización. Este parámetro es calculado a partir del monitoreo de los recursos del equipo: memoria y procesador.

La tabla 12 muestra la carga del sistema en un día promedio:

Tabla 12. Carga del sistema de la Darknet

Tiempo	Memoria	Procesador	Promedio CPU/Mem
0 hrs	75%	90%	82.5
3 hrs	80%	95%	87.5
6 hrs	80%	90%	85
9 hrs	85%	90%	87.5
12 hrs	80%	90%	85
15 h	85%	90%	87.5
18 hrs	85%	85%	85
21 hrs	80%	90%	85
24 hrs	85%	90%	87.5



Es importante reservar una determinada capacidad de carga del sistema debido a que ciertas tareas de administración lo requieren y se debe evitar que el sistema colapse. Esto en realidad es esencial ya que se observó que el sistema dejaba de responder después de estar varias horas activo al superar constantemente el 95% de la carga del sistema. En parte esto se debe a la inestabilidad de las herramientas honeypot en ciertas situaciones de conexiones masivas.

5.2 EFECTIVIDAD DE DETECCIÓN Y CLASIFICACIÓN

La característica del módulo DKN para poder detectar y clasificar eventos según reglas y patrones preestablecidos ha resultado muy efectiva. De millones de conexiones que se tienen diariamente, se logra una clasificación que reduce el número de eventos agrupando, según los criterios, en incidentes.

El saber que un barrido de puertos de 65535 conexiones únicas corresponde a un incidente, o que un equipo realiza conexiones en el puerto 22 en segmentos de red

completos buscando atacar el servicio de SSH, son ejemplos prácticos de la organización que se logra.

Se han hecho distintas mediciones de las cuales se puede extrapolar el potencial completo de la Darknet una vez que se active la detección para todo el espacio de monitoreo posible de aproximadamente entre 45,000 y 50,000 direcciones IP.

A continuación la tabla 13 muestra distintas pruebas evaluando la cantidad de incidentes detectados en un día, y tomando como base la constante de las distintas muestras, se hace una extrapolación del número aproximado con la capacidad completa.

Tabla 13. Estadísticas de detección de incidentes y captura de malware de la Darknet UNAM

Cantidad de direcciones IP activadas	**Número aproximado de incidentes detectados**	**Cantidad aproximada de payloads binarios analizados**	**Cantidad de muestras únicas de malware capturado**
100	2,000	3,000	40
255	4,500	8,000	80
2,550	55,000	80,000	1,100
15,000	320,000	450,000	6,500
30,000	*630,000*	*900,000*	*12,000*
50,000	*1,050,000*	*1,500,000*	*20,000*

Como se puede apreciar, la cantidad de payloads analizados es mayor a la cantidad de malware identificado. Esto se debe a dos razones fundamentales. En primer lugar, la captura de malware la realizan dos herramientas honeypots diferentes. Una de ellas, Dionaea, tiene una capacidad mayor para la captura de malware sin embargo no captura payloads de cualquier conexión por lo que el módulo DKN no los analiza, simplemente registra que la herramienta capturó un posible malware. Mientras tanto, Honeytrap captura todos los payloads de las conexiones, aún cuando no necesariamente se trate de malware y de hecho es mínima la captura de malware con esta herramienta que está más enfocada al análisis del payload de la conexión. La otra

razón se debe a que se trata de muestras únicas, es decir, quizá en algunos incidentes se captura el mismo malware pero solo se registra una vez.

El número de incidentes detectados es el otro factor de medición el cual puede ser un número considerable dependiendo de la capacidad que se active en la Darknet, sin embargo, como cualquier sistema de detección de tráfico malicioso, cierta cantidad de los incidentes representan falsos positivos. Actualmente no está implementado un mecanismo para hacer una evaluación y verificación completa de la cantidad de ellos, no obstante, mediante la definición de más reglas se ayuda a disminuir el número de eventos que correspondan a uno.

5.3 UTILIDAD DE LA INFORMACIÓN

Una vez que la información se tiene almacenada en la base de datos es fácilmente accesible. Ya que el diseño permite obtener un detalle muy preciso de cada incidente, el aprovechamiento que se le da a la misma se ha enfocado principalmente a la atención de incidentes y análisis de nuevas amenazas.

A partir de la implantación de la Darknet, se puede reportar un incidente proporcionando detalles que permitan a los dueños o administradores de los equipos detectados como origen del tráfico, mitigar de una manera más precisa o tener la evidencia suficiente para entender el problema. A pesar de que esto aún no está en producción con el Sistema de Atención a Incidentes del UNAM-CERT, el motor de detección ya almacena en la base de datos los eventos detectados, por lo que la extracción de información es de forma manual mediante consultas de base de datos y visualización de registros.

El otro enfoque de la información obtenida es la parte de investigación, la cual abre muchas expectativas debido al potencial del gran espacio de monitoreo y de ser un entorno académico. Desde su desarrollo se han hecho cambios al motor en cuanto a características y capacidades por lo que cada mejora implica un mejor aprovechamiento del TSU.

En el Anexo F se muestran ejemplos de información almacenada en la base de datos.

5.4 TAREAS PENDIENTES Y MEJORAS POSIBLES

El estado actual de la Darknet cumple con el objetivo fundamental de ser un motor de detección del TSU, sin embargo aún hay tareas por completar y aspectos a mejorar en desarrollos posteriores.

a) Interfaz WEB del TSU

   Para que la información pueda ser aprovechada de manera más sencilla y práctica, la interfaz del TSU es un desarrollo que se encuentra en proceso. El objetivo fundamental es extraer mediante Web toda la información obtenida por la Darknet.

b) Automatización con el SAI

   Se debe terminar la interacción con el Sistema de Atención a Incidentes para que el proceso de aprovechamiento y extracción de información sea de manera automatizada.

c) Integración con la sandnet del Proyecto Malware UNAM

   Se planea desarrollar un módulo que permita la integración de las muestras recopiladas por la Darknet con el laboratorio de análisis de malware perteneciente al UNAM-CERT.

d) Integración con modelos de datos compartidos

   Se planea integral al TSU un módulo para poder compartir la información con organizaciones externas. Algunos ejemplos son los formatos de Shadowserver y algunas herramientas desarrolladas por miembros de The Honeynet Project.

e) Detección de shellcodes

   La detección de shellcodes mediante distintas técnicas es una característica factible de implementar en el módulo DKN. Las mejoras implican cambios en el módulo de análisis de payloads.

f) Integración de más herramientas honeypots

CONCLUSIONES

El trabajo de tesis presentado corresponde al diseño e implantación de un motor de detección de tráfico malicioso para el Telescopio de Seguridad de la UNAM como una fuente de información, proporcionando un mecanismo complementario para la atención a incidentes de seguridad informática dentro y fuera de RedUNAM.

El diseño de un mecanismo de detección en redes de gran tamaño y complejidad como RedUNAM implica un análisis exhaustivo del proceso de manejo y almacenamiento de la información debido a la gran cantidad de datos involucrados.

La elección de diferentes tecnologías y técnicas utilizadas depende de la información que se desea obtener, del enfoque de uso de dicha información y del o los sistemas con los cuales deberá interactuar. En este caso, debido a la necesidad de interacción de un mecanismo de detección con el proceso de atención a incidentes de seguridad informática dentro de RedUNAM, y al enfoque de investigación en técnicas de detección de tráfico malicioso, las tecnologías y técnicas utilizadas (honeypots, IDS, análisis estructurado de tráfico de red) resultaron convenientes y proporcionan una base para un sistema de monitoreo complejo que permita identificar de manera acertada gran variedad de amenazas dentro la red de la Universidad.

El diseño de la Darknet como un motor de detección y del sistema de procesamiento de la información para su implantación en el Telescopio de Seguridad de la UNAM, ofrecen una gran versatilidad para la generación de nuevos mecanismos que apoyen en las tareas de detección, procesamiento y almacenamiento de la información. Una de las razones es el diseño modular que permite cierta independencia entre las tecnologías utilizadas y ofrece la capacidad de tener facilidad de interacción con posibles nuevas herramientas y técnicas de captura, detección y procesamiento de información.

Uno de los principales problemas encontrados durante el desarrollo de este proyecto fue la adaptación de un proceso adecuado para que la Darknet tuviera el mejor rendimiento posible. El rendimiento de la misma con los primeros diseños del sistema era totalmente inadecuado debido a la gran cantidad de tiempo que tomaba en capturar y procesar la información. Después de un análisis y de estudiar distintas

alternativas en diferentes módulos de la Darknet, se logró que la información se manejara de manera eficiente y se pudiera aprovechar en el TSU a partir de un procesamiento en "tiempo real".

Este tipo de sistemas de detección son escasos debido a sus características. Por un lado, se necesita de una inversión de direcciones IP la cual es proporcional a la efectividad del sistema, y por otro lado, tienen un cargado enfoque para el área de investigación por lo que comúnmente es implementado por organismos académicos, de investigación o cuya misión es la generación de tecnologías de seguridad tal como las firmas antivirus.

El tamaño y capacidad de la Darknet UNAM es considerable en términos académicos, incluso es más grande que algunos de los proyectos similares en otras Universidades internacionales. Algunos otros son mucho más grandes ya que conjuntan varias organizaciones como universidades, ISP's, compañías, e incluso departamentos de defensa. Dentro de un ambiente académico a nivel nacional e incluso en Latinoamérica, no se tiene conocimiento de un proyecto similar con las características de la Darknet UNAM, lo cual la convierte en una referencia importante en ese contexto.

Con la información obtenida ahora por el TSU, el UNAM-CERT puede ser también una referencia para la detección de incidentes de seguridad informática dentro y fuera de RedUNAM mediante la notificación de incidentes a las fuentes de eventos detectados. También, gracias a las características de la información obtenida se pueden lograr intercambios de información más detallados con otras organizaciones de seguridad internacionales.

Si bien las tecnologías honeypots no son nuevas, su correcta implementación y sobre todo el buen aprovechamiento de la información generada, representan un campo explotable en el área de la seguridad en cómputo ya que permiten obtener información alternativa o complementaria a la que generan otros tipos de sistemas de detección como los IDS, firewalls, etc.

El sistema completo de la Darknet dentro del TSU aún presenta algunas tareas y retos qué resolver. Se debe hacer énfasis en mejorar los aspectos de rendimiento y la adaptación de nuevas tecnologías de detección ya que mientras mayor y mejor es la información proporcionada, la demanda en recursos físicos de procesamiento y almacenamiento aumenta.

Esta primera versión de la Darknet UNAM cumple con los objetivos propuestos, y es importante mencionar que para optimizar su utilidad, es necesario contar con un proceso de actualización y mejora constante. Las siguientes versiones de la misma y las nuevas características implementadas llevarán poco a poco a una consolidación del Telescopio de Seguridad de la UNAM como efectivo sistema de detección de amenazas de seguridad en cómputo.

REFERENCIAS

[1] Anderson Neil, Della Maggiora Paul, Doherty Jim, "Cisco Networking Simplified", Segunda edición. Cisco Press. 2007.

[2] Argus – Auditing Network Activity http://www.qosient.com/argus/

[3] Argus practical botnet detection http://www.rawpacket.org/anonymous/papers/Argus-PracticalBotNetDetection.pdf

[4] Arnold Jon, Dwarshius Russell, Howell Paul, Jahanian Farnam, Malan Rob, Ogden Jeff, Poland Jon, Smart Matthew, University of Michigan, Merit-Research Paper, "Observations and Experiences Tracking Denial-Of-Service Attacks Across a Large Regional ISP", http://www.arbornetworks.com/index.php?option=com_docman&task=doc_download&gid=97

[5] Bailey Michael, Cooke Evan, Jahanian Farnam, Nazario Jose, Watson David, University of Michigan, Arbor Networks ,"The Internet Motion Sensor: A Distributed Blackhole Monitoring System", http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.75.4081&rep=rep1&type=pdf

[6] Bailey Michael, Cooke Evan, Jahanian Farnam, Nazario Jose, Watson David, University of Michigan, Arbor Networks , Research Paper, "Toward Understanding Distributed Blackhole Placement", Presented at The 2nd Workshop on Rapid Malcode (WORM), http://www.arbornetworks.com/index.php?option=com_docman&task=doc_download&gid=107

[7] Bailey Michael, Cooke Evan, Jahanian Farnam, Watson David, Electrical Engineering and Computer Science Department, University of Michigan, Nazario Jose, Arbor Networks , "The Internet Motion Sensor: A distributed global scoped Internet threat monitoring system ", http://eecs.umich.edu/techreports/cse/04/CSE-TR-491-04.pdf

[8] Barford Paul, University of Wisconsin at Madison, "Toward Self-Directed Network Intrusion Detection and Prevention", Conferencia, http://www.csail.mit.edu/events/eventcalendar/calendar.php?show=event&id=871

[9] Binkley Jim (Portland State University), "Anomaly-based BotServer (and more!) Detection", http://cert.org/flocon/2006/presentations/botserver2006_ppt.pdf

[10] Bjarte Malmedal, Gjøvik University College, "Using Netflows for slow portscan detection", 2005, http://www.malmedal.net/Malmedal_Master_Thesis.pdf

[11] Bullard Carter, QoSient LLC, "Network Flow Data Fusion. GeoSpatial and NetSpatial Data Enhancement", FloCon 2010,

https://tools.netsa.cert.org/wiki/download/attachments/10027010/Bulla rd_DataFusion.pdf

[12] Chuvakin Anton blog, http://chuvakin.blogspot.com/

[13] Chuvakin Anton homepage, http://www.chuvakin.org/

[14] Gadsden Richard, "Mass-Mailing Worms: Prevention, Detection and Response (A Case Study)", SANS Institute InfoSec Reading Room, http://www.sans.org/reading_room/whitepapers/malicious/mass-mailing-worms-prevention-detection-response-a-case-study_1148

[15] Gartner Inc. "Gartner Information Security Hype Cycle Declares Intrusion Detection Systems a Market Failure" , http://gartner.com/5_about/press_releases/pr11june2003c.jsp

[16] Haile Jed, "Using Argus Audit Trails to Enhance IDS Analysis", Nitro Data, Systems, http://cansecwest.com/core03/jhaile-cansec03.ppt

[17] Honeynets, conoce a tu enemigo. Presentación en V Foro de seguridad RedIRIS Detección de Intrusiones. Abril de 2007, http://www.rediris.es/cert/doc/reuniones/fs2007/archivo/Honeynets_RaulSiles_VForoSeguridadRedIRIS_Abril2007.pdf

[18] Honeypots Definitions and Value of Honeypots, http://www.tracking-hackers.com/papers/honeypots.html

[19] Huffaker Bradley, Cooperative Association for Internet Data Analysis CAIDA, San Diego Supercomputer Center, University of California, San Diego. "CAIDA, Report 2010", http://caida.org/publications/presentations/2010/caida_update/

[20] Ido Dubrawsky, Wes Noonan,. "Firewalls fundamentals", Cisco Press, 2006.

[21] Insecure magazine Issue 4. Octubre 2005, http://www.net-security.org/dl/insecure/INSECURE-Mag-4.pdf

[22] Internet Background Noise (IBN), http://www.switch.ch/security/IBN/

[23] Internet Security Monitoring, Monitoring and coordination of intrusion detection, http://www.authorstream.com/Presentation/Wanderer-17376 -intrusion-detection-monitoring-Internet-Security-Current-Practice-Protecting-against-Intrusions-One-Promising-Approa-as-Entertainment-ppt-powerpoint

[24] Introduction to Intrusion Detection, http://ciscosecurity.org.ua/1587051672/ch10lev1sec1.html

[25] Kent Karen, Northcutt Stephen, Winters Scott, W. Ritchey Ronald, Zeltser Lenny, "Inside Network Perimeter Security", SAMS, 2005.

[26] Kippo Project http://code.google.com/p/kippo/

[27] Know Your Enemy: GenII Honeynets, http://old.honeynet.org/papers/gen2/index.html

[28] Know Your Enemy: Honeynets in Universities, http://old.honeynet.org/papers/edu

[29] Know Your Enemy: Honeynets, http://old.honeynet.org/papers/honeynet/

[30] Lee Rob, "SANS Computer Forensics and e-Discovery, Interview with Michael Cloppert", 2009, http://blogs.sans.org/computer-forensics/2009/03/04/michael-cloppert-computer-forensic-hero/

[31] Mason Andrew, "Cisco firewall Technology", Cisco Press, 2007.

[32] McPherson Danny, Arbor Networks; Tim Battles, AT&T; Bailey Michael, Cooke Evan, University of Michigan – Presentation, "Tracking Global Threats with the Internet Motion Sensor", http://www.arbornetworks.com/index.php?option=com_docman&task=doc_download&gid=108

[33] McPherson Danny, Nazario Jose, Arbor Networks; Michael Bailey, University of Michigan - Presentation, "Measuring Global Worm Activity", Presentacion en NANOG 30, http://www.arbornetworks.com/index.php?option=com_docman&task=doc_download&gid=103

[34] McRee Russ, "Argus – Auditing network activity". 2007, http://holisticinfosec.org/toolsmith/docs/november2007.pdf

[35] McRee Russ, "Expanding Response: Deeper Analysis for Incident Handlers", SANS Institute InfoSec Reading Room. 2008.

[36] Moore David, Cooperative Association for Internet Data Analysis – CAIDA, San Diego Supercomputer Center, University of California, San Diego, "Detecting Internet Worms", http://www.caida.org/publications/presentations/2005/detecting_worms

[37] Moore David, Cooperative Association for Internet Data Analysis – CAIDA, San Diego Supercomputer Center, University of California, San Diego, "Network Telescopes", http://caida.org/publications/presentations/2003/dimacs0309/

[38] Moore David, Cooperative Association for Internet Data Analysis – CAIDA, San Diego Supercomputer Center, University of California, San Diego, "Network Telescopes: Observing Small or Distant Security Events", http://www.caida.org/publications/presentations/2002/usenix_sec

[39] Moore David, Cooperative Association for Internet Data Analysis – CAIDA, San Diego Supercomputer Center, Geoffrey M. Voelker, Stefan Savage, Department of Computer Science and Engineering, University of California, San Diego, "Network Telescopes: Technical Report", http://caida.org/publications/papers/2004/tr-2004-04/

[40] On the Design and Use of Internet Sinks for Network Abuse Monitoring, http://pages.cs.wisc.edu/~vinod/raid-paper.pdf

[41] Packet Capture library, http://www.tcpdump.org/pcap3_man.html

[42] Papoulis, A. "Bernoulli Trials." 3-2 in Probability, Random Variables, and Stochastic Processes, 2nd ed. New York: McGraw-Hill, pp. 57-63, 1984.

[43] Portal de herramientas de seguridad de Carnivore, http://carnivore.it

[44] Richard Bejtlick, "Implementing Network Security with Open Source Tools", http://sce.uhcl.edu/yang/teaching/csci5931webSecuritySpr04/8-21sSecurity-FINAL%20net%20security%20monitoring.pdf

[45] SANS – Internet Storm Center, "Survival time", http://isc.sans.edu/survivaltime.html

[46] SGUIL: The analyst console for network security monitoring, http://sguil.sourceforge.net/

[47] Shannon Colleen, Cooperative Association for Internet Data Analysis CAIDA, San Diego Supercomputer Center, University of California, San Diego, "CAIDA Activities", http://caida.org/publications/presentations/2007/terena_caida/

[48] Shannon Colleen, Moore David, Cooperative Association for Internet Data Analysis, CAIDA, San Diego Supercomputer Center, University of California, San Diego , "Security Data Collection at CAIDA",http://www.caida.org/publications/presentations/2004/security_collection_wide/

[49] Shannon Colleen, Moore David, Cooperative Association for Internet Data Analysis, CAIDA, San Diego Supercomputer Center, University of California, San Diego, "Network Telescopes: Remote Monitoring of Internet Worms and Denial-of-Service Attacks", http://www.caida.org/publications/presentations/2004/network_telescopes/

[50] Sourcefire Vulnerability Researh Team, http://www.snort.org/snort-rules/

[51] Spenneberg Ralf, “Keeping an eye on the network with Argus WATCHFUL EYE”, Linux Magazine. Febrero 2007, http://www.linux-magazine.com/w3/issue/75/Argus.pdf

[52] Splunk IT Search for Log Management, Operations, security compliance, http://www.splunk.com/

[53] Team Cymru, The Darknet Project http://www.team-cymru.org/Services/Darknets.html

[54] The Cooperative Association for Internet Data Analysis, http://www.caida.org/research/security/

[55] The IUCC/IDC Internet Telescope, http://noc.ilan.net.il/research/telescope/

[56] The Shadowserver Foundation, http://www.shadowserver.org/wiki/pmwiki.php/Shadowserver/Organizations

[57] Thomas Joshua, Conley Thomas, Ohio University, Internet Traffic Analysis for Threat Detection. Midwest Regional Conferences, 2005, http://www.educause.edu/Resources/InternetTrafficAnalysisforThre/159269

[58] Trost Ryan, “Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century”, Addison-Wesley Professional, 2009.

[59] UCSD Network telescope, http://www.caida.org/data/passive/network_telescope.xml

[60] Van Epp Peter (Simon Fraser University), “Using archived argus flow records to secure and troubleshoot your network”, http://www.internet2.edu/presentations/jtvancouver/20050720-Argus-VanEpp.pdf

[61] Vinod Yegneswaran, University of Wisconsin, “Empirical Foundations for Network Defense”, Conferencia, http://www.cs.pitt.edu/events/talks/06-2/vinod-yegneswaran.01mar2006.php

GLOSARIO

Argus	Herramienta para el análisis de flujos del tráfico de red.
CAIDA	Cooperative Association for Internet Data Analysis. Organización dedicada al monitoreo y análisis del tráfico en Internet.
CERT	Computer Emergency Response Team.
DKN	Término para hacer referencia a una Darknet en esta tesis.
DoS	Denial of service. Ataque de negación de servicio que atenta contra la disponibilidad de un sistema o red.
Firewall	Sistema de control y filtrado de paquetes de red.
Honeynet	Estrictamente se refiere a un honeypot de alta interacción (equipo real); comúnmente hace referencia también a un conjunto de honeypots en un ambiente controlado.
Honeypot	Sistema diseñado para ser atacado o alcanzado por tráfico o entidad maliciosa con el fin de detectar y analizar la actividad que ocurre en él.
Honeywall	Sistema de control y análisis del tráfico de red en una honeynet.
IBN	Internet Background Noise. Telescopio de seguridad de la organización SWITCH.
ICMP	Internet Control Message Protocol. Protocolo para control y notificación de errores en el protocolo de Internet (IP).
IDP	Intrusion Detection and Prevention, Sistema de detección y prevención de intrusiones en un sistema o red.
IDS	Intrusion Detection System, Sistema de detección de intrusos en un sistema o red.
IMS	Internet Motion Sensor. Telescopio de red desarrollado por la Universidad de Michigan y la firma Arbor Networks.
IP	Internet Protocol, protocolo de comunicación no orientado a conexión utilizado para establecer comunicación entre equipos de una red.

IPS	Intrusion Prevention System, sistema de prevención de intrusos en un sistema o red.
ISP	Internet Service Provider. Proveedor de servicios de Internet.
Malware	Término para hacer referencia a software malicioso (malicious software)
Payload	Contenido del campo de datos en un paquete de red.
P2P	Peer to Peer. Red de computadoras en las que las conexiones se hacen entre equipos sin necesidad de clientes o servidores. Cada equipo es un nodo.
Ra tolos	Conjunto de herramientas del cliente de Argus.
SAI	Sistema de Atención a Incidentes de la Subdirección de Seguridad de la Información / UNAM-CERT.
Snort	Sistema de detección de intrusos desarrollado por Sourcefire.
SSH	Secure Shell, servicio de red para estableces comunicaciones cifradas entre un cliente y un servidor UNIX.
TCP	Transmission Control Protocol, protocolo de la capa de transporte del modelo TCP/IP.
TSU	Telescopio de Seguridad de la UNAM.
UNAM	Universidad Nacional Autónoma de México.
URL	Uniform Resource Locator. Secuencia de caracteres en un formato estándar para nombrar recursos en Internet.
VRT	Vulnerability Research Team. Equipo de investigación y desarrollo de la firma de seguridad Sourcefire, creadora del IDS Snort.

ANEXOS

ANEXO A – COMPARATIVA DE TIEMPOS DE DETECCIÓN PARA DARKNETS DE DIFERENTES TAMAÑOS

Tabla A.1 Muestra las probabilidades de ver al menos k paquetes en una red /8 cuando un equipo envía 500 paquetes por segundo durante 60 segundos.

k	N	P	P	P
50	500pps * 60sec = 30000 paq	2^{-8}	$P = 1 - \sum_{y=0}^{50-1} \binom{30000}{y} (2^{-8})^{y} (1 - 2^{-8})^{30000-y}$	**≈100%**
75	500pps * 60sec = 30000 paq	2^{-8}	$P = 1 - \sum_{y=0}^{75-1} \binom{30000}{y} (2^{-8})^{y} (1 - 2^{-8})^{30000-y}$	**99.998%**
100	500pps * 60sec = 30000 paq	2^{-8}	$P = 1 - \sum_{y=0}^{100-1} \binom{30000}{y} (2^{-8})^{y} (1 - 2^{-8})^{30000-y}$	**95.2%**
125	500pps * 60sec = 30000 paq	2^{-8}	$P = 1 - \sum_{y=0}^{125-1} \binom{30000}{y} (2^{-8})^{y} (1 - 2^{-8})^{30000-y}$	**24.66%**
150	500pps * 60sec = 30000 paq	2^{-8}	$P = 1 - \sum_{y=0}^{150-1} \binom{30000}{y} (2^{-8})^{y} (1 - 2^{-8})^{30000-y}$	**0.19%**
200	500pps * 60sec = 30000 paq	2^{-8}	$P = 1 - \sum_{y=0}^{200-1} \binom{30000}{y} (2^{-8})^{y} (1 - 2^{-8})^{30000-y}$	**2.05x10-15%**

Tabla A.2 Muestra las probabilidades de ver al menos k paquetes en una red **/16** cuando un equipo envía 1000 paquetes por segundo durante 7 minutos.

K	N	P	P	P [%]
2	1000pps*420sec= 420000paq	2^{-16}	$P = 1 - \sum_{y=0}^{2-1} \binom{420000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**95.397**
4	1000pps*420sec= 420000paq	2^{-16}	$P = 1 - \sum_{y=0}^{4-1} \binom{420000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**76.594**
6	1000pps*420sec= 420000paq	2^{-16}	$P = 1 - \sum_{y=0}^{6-1} \binom{420000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**45.905**
8	1000pps*420sec= 420000paq	2^{-16}	$P = 1 - \sum_{y=0}^{8-1} \binom{420000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**19.769**
10	1000pps*420sec= 420000paq	2^{-16}	$P = 1 - \sum_{y=0}^{10-1} \binom{420000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**6.187**

Tabla A.3 Muestra las probabilidades de ver al menos k paquetes en una red **/16** cuando un equipo envía 1000 paquetes por segundo durante 8 minutos.

K	N	P	P	P [%]
2	1000pps*480sec= 480000paq	2^{-16}	$P = 1 - \sum_{y=0}^{2-1} \binom{480000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**97.683**
4	1000pps*480sec= 480000paq	2^{-16}	$P = 1 - \sum_{y=0}^{4-1} \binom{480000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**85.459**
6	1000pps*480sec= 480000paq	2^{-16}	$P = 1 - \sum_{y=0}^{6-1} \binom{480000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**59.74**
8	1000pps*480sec= 480000paq	2^{-16}	$P = 1 - \sum_{y=0}^{8-1} \binom{480000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**31.405**
10	1000pps*480sec= 480000paq	2^{-16}	$P = 1 - \sum_{y=0}^{10-1} \binom{480000}{y} (2^{-16})^{y} (1 - 2^{-16})^{420000-y}$	**12.312**

ANEXO B – ESQUEMA DE LA BASE DE DATOS PARA ALMACENAMIENTO DE LA INFORMACIÓN DE LOS INCIDENTES DE LA DARKNET

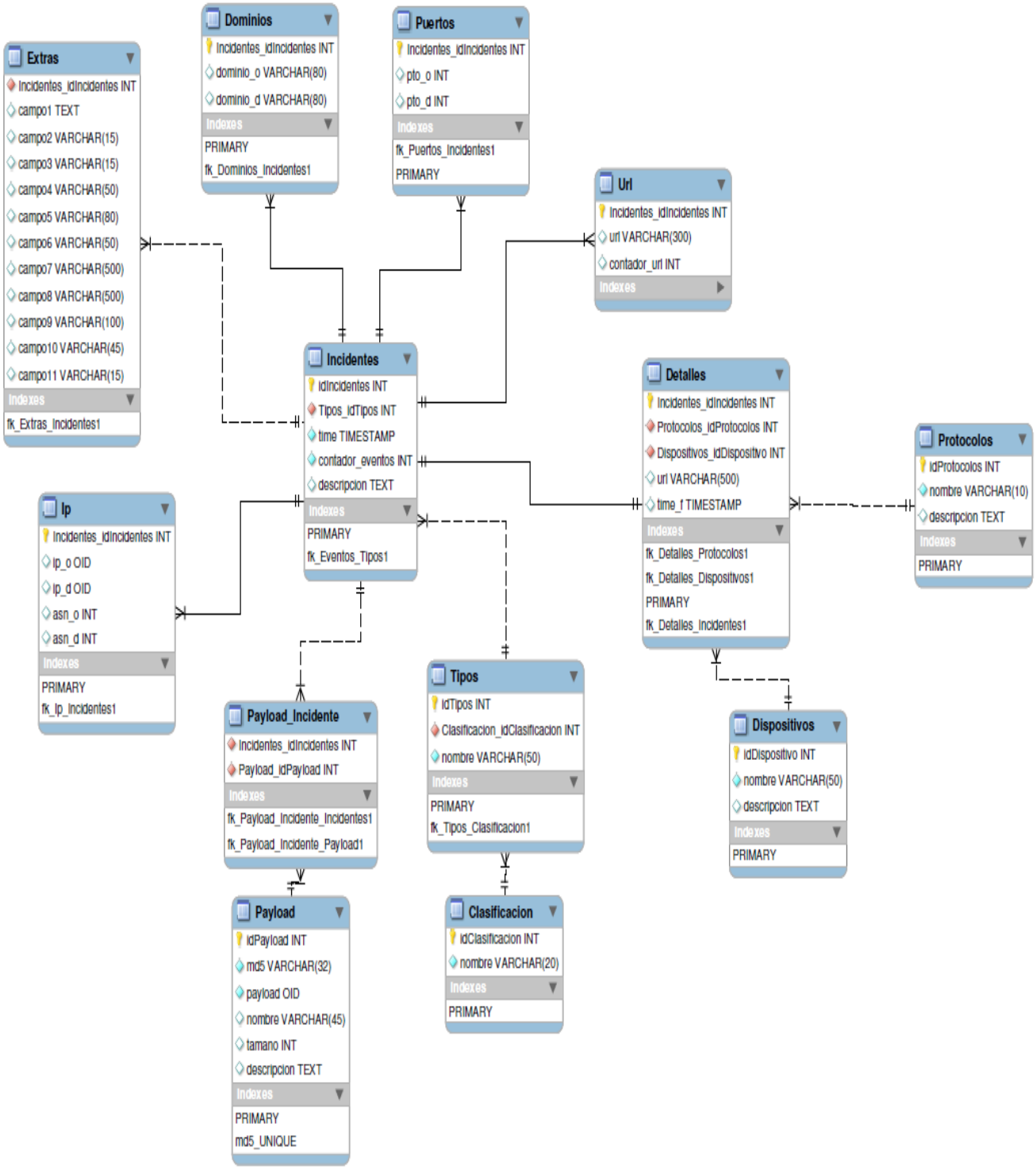


Diagrama B.1 Esquema de la base de datos del TSU

ANEXO C – EJEMPLO DEL FORMATO UNIFICADO PARA ALMACENAMIENTO DE INFORMACIÓN DE INCIDENTES

a) Archivo de incidentes

dkn|163|tcp|83.110.67.184|||||||1271362670|1271362370|SQL WORM 1433|/data/dkn/events_connections/tcp-83.110.67.184-1433-1271362370.det|/data/dkn/events_connections/tcp-83.110.67.184-1433-1271362370.tgz|

b) Archivo de detalles

/data/dkn/events_connections/tcp-83.110.67.184-1433-1271362370.det

TS SRCIP&SPORT SRCIP&DPORT MD5 PAYLOAD STRINGS(Rules)

1271362370|83.110.67.184|3518|132.247.0.232|1433|||||||
1271362378|83.110.67.184|4368|132.247.0.232|1433|285850d4aff8df0e2839ecd6bca68011|||||-(0)|
1271362378|83.110.67.184|4374|132.247.0.232|1433|36dc32801e14fbcdd23436759389f4d4|||||-(0)|
1271362378|83.110.67.184|4394|132.247.0.232|1433|cfd5cff90daae596afab961957826d3c|||||-(0)|
1271362378|83.110.67.184|4432|132.247.0.232|1433|b27e34b029eafa04e44fa4af416ed8cd|||||-(0)|
1271362378|83.110.67.184|4442|132.247.0.232|1433|910729ad1d2de99522b537b05ffd00a2|||||-(0)|
1271362378|83.110.67.184|4446|132.247.0.232|1433|ab1336d70e64574b411b6a133129c557|||||-(0)|

c) Archivo de payloads

```
tmp# tar -zxvf tcp-83.110.67.184-1433-1271362370.tgz
data/honeytrap/attacks/285850d4aff8df0e2839ecd6bca68011
data/honeytrap/attacks/36dc32801e14fbcdd23436759389f4d4
data/honeytrap/attacks/cfd5cff90daae596afab961957826d3c
data/honeytrap/attacks/b27e34b029eafa04e44fa4af416ed8cd
data/honeytrap/attacks/910729ad1d2de99522b537b05ffd00a2
data/honeytrap/attacks/ab1336d70e64574b411b6a133129c557
```

ANEXO D – EJEMPLO DEL ANÁLISIS DE FLUJOS (STA)

a) Archivo general de sesiones de la captura de tráfico en un lapso determinado.

Paquetes/sesión	*IP origen*	*IP destino&puerto*	*Protocolo*
50	113.138.135.169	132.247.0.6.445	tcp
45	113.134.23.32	132.247.0.92.445	tcp
43	113.138.135.169	132.247.0.8.445	tcp
33	180.183.218.8	132.247.0.20.445	tcp
28	180.183.218.8	132.247.0.16.445	tcp
22	113.138.135.169	132.247.0.5.445	tcp
20	113.134.23.32	132.247.0.83.445	tcp
16	113.134.23.32	132.247.0.97.445	tcp
13	67.46.8.42	132.247.0.5.445	tcp
11	67.46.8.42	132.247.0.5.139	tcp
10	66.82.9.22	132.247.0.5.80	tcp
10	180.183.218.8	132.247.0.19.445	tcp
7	180.183.218.8	132.247.0.6.445	tcp
5	180.183.218.8	132.247.0.14.445	tcp

b) Archivo con información estadística sobre la captura

racount	records	total_pkts	src_pkts	dst_pkts	total_bytes	src_bytes	dst_bytes
sum	650	5807	2939	2868	806674	407881	398793

c) Archivo con la relación de actividad entre equipos

```
2.89.94.160: (1) 132.247.0.18
46.166.86.35: (1) 132.247.0.125
50.22.42.62: (6) 132.247.0.66, 132.247.0.70, 132.247.0.78 - 132.247.0.79, 132.247.0.83,
132.247.0.93
58.170.110.136: (1) 132.247.0.80
59.182.11.129: (1) 132.247.0.85
67.46.8.42: (1) 132.247.0.5
83.211.35.77: (1) 132.247.0.15
88.119.145.173: (1) 132.247.0.33
91.220.176.249: (1) 132.247.0.125
92.240.68.153: (1) 132.247.0.186
95.211.81.35: (1) 132.247.0.122
113.134.23.32: (12) 132.247.0.83 - 132.247.0.86, 132.247.0.88 - 132.247.0.90,
132.247.0.92 - 132.247.0.94, 132.247.0.96 - 132.247.0.97
113.138.135.169: (4) 132.247.0.5 - 132.247.0.8
113.165.167.160: (1) 132.247.0.157
114.203.34.204: (1) 132.247.0.113
115.22.231.89: (1) 132.247.0.146
132.247.0.97: (1) 113.134.23.32
178.37.16.115: (1) 132.247.0.216
180.183.218.8: (15) 132.247.0.2 - 132.247.0.3, 132.247.0.5 - 132.247.0.9, 132.247.0.11 -
132.247.0.16, 132.247.0.19 - 132.247.0.20
201.48.211.161: (1) 132.247.0.96
201.225.119.155: (1) 132.247.0.201
202.116.160.171: (1) 132.247.0.159
208.64.126.120: (3) 132.247.0.165, 132.247.0.245, 132.247.0.248
```

d) Archivo con el análisis de payloads de los flujos según reglas

El módulo genera n archivos como este según el número de reglas especificadas.

more stamod_sessions_POSIBLE_WORM_MS-DS_445-tcp-445-string.dat

```
18    113.138.135.169        132.247.0.6.445    s[50]=.....SMBr.......PC NETWORK
15    113.138.135.169        132.247.0.8.445    s[50]=.....SMBr.......PC NETWORK
14      113.134.23.32       132.247.0.92.445    s[50]=.....SMBr.......PC NETWORK
11      180.183.218.8       132.247.0.20.445    s[50]=.....SMBr.......PC NETWORK
 7      113.134.23.32       132.247.0.83.445    s[50]=.....SMBr.......PC NETWORK
 3      180.183.218.8       132.247.0.20.445    s[50]=.....SMBs.........BSRSPYL.A2.
 3    113.138.135.169        132.247.0.6.445    s[50]=...<.SMBs.........BSRSPYL A2.
 2    113.138.135.169        132.247.0.6.445    s[50]=...<.SMBs.........BSRSPYL .A2
 2    113.138.135.169        132.247.0.5.445    s[50]=.....SMBs.........BSRSPYL A2.
 2      113.134.23.32       132.247.0.97.445    s[50]=.....SMBr.......x..PC NETWORK
```

more stamod_sessions_SSH_SCAN_O_POSIBLE_SSH_BRUTEFORCE_ATTACK-tcp-22-string.dat

```
 1      114.247.15.79       132.247.0.92.22     s[50]=SSH-2.0-libssh-0.2.......diff
 1      114.247.15.79       132.247.0.92.22     s[50]=SSH-2.0-libssh-0.2..@....diff
 1      114.247.15.79       132.247.0.92.22     s[50]=SSH-2.0-libssh-0.2..A....diff
 1      114.247.15.79       132.247.0.92.22     s[50]=SSH-2.0-libssh-0.2.......diff
 1      114.247.15.79       132.247.0.92.22     s[50]=SSH-2.0-libssh-0.2.......diff
 1      114.247.15.79       132.247.0.92.22     s[50]=SSH-2.0-libssh-0.2.......diff
 1      114.247.15.79       132.247.0.92.22     s[50]=SSH-2.0-libssh-0.2..h....diff
 1      114.247.15.79       132.247.0.92.22     s[50]=...q3....Y..!..T.N>...4R...y.
 1      114.247.15.79       132.247.0.92.22     s[50]=.n@..?..W....-.1k.Qk.V.t.>A.k
 1      114.247.15.79       132.247.0.92.22     s[50]=N.U+|]o~...8..%..6...[.m.oF..
 1      114.247.15.79       132.247.0.92.22     s[50]=..l. ..*r........k....5L1..\v
 1      114.247.15.79       132.247.0.92.22     s[50]=...].h.o......S.L....IOsQ.n.3
```

ANEXO E – EJEMPLO DE ANÁLISIS DE TRÁFICO DE RED CON IDS (STA)

a) Top de incidencias

```
::::::::::::::
top.alert
::::::::::::::
297     |       SQL SA brute force login attempt TDS v7/8
45      |       WEB-IIS view source via translate header
12      |       SQL Worm propagation attempt
12      |       SQL version overflow attempt
12      |       SQL Worm propagation attempt OUTBOUND
6       |       NETBIOS SMB-DS srvsvc NetrPathCanonicalize WriteAndX little endian overflow attempt
4       |       POLICY VNC server response
3       |       ET SCAN Behavioral Unusual Port 1433 traffic, Potential Scan or Infection
2       |       ET EXPLOIT Microsoft Windows NETAPI Stack Overflow Inbound - MS08-067 (25)
2       |       EXPLOIT RealVNC server authentication bypass attempt
2       |       ET EXPLOIT Microsoft Windows NETAPI Stack Overflow Inbound - MS08-067 (15)
1       |       ET SCAN Sipvicious Scan

::::::::::::::
top.class
::::::::::::::
297     |        [Classification: An attempted login using a suspicious username was detected] [Priority:
2]
45      |        [Classification: access to a potentially vulnerable web application] [Priority: 2]
24      |        [Classification: Misc Attack] [Priority: 2]
22      |        [Classification: Attempted Administrator Privilege Gain] [Priority: 1]
9       |        [Classification: Misc activity] [Priority: 3]
1       |        [Classification: Attempted Information Leak] [Priority: 2]

::::::::::::::
top.dstip
::::::::::::::
276     |       SQL SA brute force login attempt TDS v7/8||132.247.0.25
21      |       SQL SA brute force login attempt TDS v7/8||132.247.0.11
3       |       WEB-IIS view source via translate header||132.247.0.8
3       |       WEB-IIS view source via translate header||132.247.0.6
2       |       POLICY VNC server response||132.247.0.26
2       |       POLICY VNC server response||132.247.0.58
2       |       WEB-IIS view source via translate header||132.247.0.4
1       |       SQL version overflow attempt||132.247.0.200
1       |       EXPLOIT RealVNC server authentication bypass attempt||132.247.0.26
1       |       SQL Worm propagation attempt OUTBOUND||132.247.0.158
1       |       SQL Worm propagation attempt||132.247.0.167
1       |       SQL Worm propagation attempt||132.247.0.230
1       |       SQL Worm propagation attempt||132.247.0.117
1       |       SQL Worm propagation attempt||132.247.0.19
1       |       SQL Worm propagation attempt||132.247.0.87
1       |       SQL Worm propagation attempt||132.247.0.162
1       |       SQL Worm propagation attempt||132.247.0.178
1       |       SQL version overflow attempt||132.247.0.162
1       |       WEB-IIS view source via translate header||132.247.0.21
1       |       SQL Worm propagation attempt||132.247.0.57
1       |       SQL version overflow attempt||132.247.0.169

::::::::::::::
top.dstport
::::::::::::::
297     |       SQL SA brute force login attempt TDS v7/8||1433
45      |       WEB-IIS view source via translate header||80
12      |       SQL Worm propagation attempt OUTBOUND||1434
12      |       SQL version overflow attempt||1434
12      |       SQL Worm propagation attempt||1434
4       |       POLICY VNC server response||5900
3       |       ET SCAN Behavioral Unusual Port 1433 traffic, Potential Scan or Infection||1433
2       |       EXPLOIT RealVNC server authentication bypass attempt||5900
2       |       ET EXPLOIT Microsoft Windows NETAPI Stack Overflow Inbound - MS08-067 (25)||445
2       |       ET EXPLOIT Microsoft Windows NETAPI Stack Overflow Inbound - MS08-067 (15)||445
1       |       ET SCAN Sipvicious Scan||5060
```

```
::::::::::::::
top.srcip
::::::::::::::
276     |       SQL SA brute force login attempt TDS v7/8||216.14.118.202
21      |       SQL SA brute force login attempt TDS v7/8||62.149.163.103
19      |       WEB-IIS view source via translate header||116.10.255.17
13      |       WEB-IIS view source via translate header||221.172.58.255
6       |       SQL Worm propagation attempt OUTBOUND||211.138.238.198
6       |       SQL Worm propagation attempt||211.138.238.198
6       |       SQL version overflow attempt||211.138.238.198
5       |       WEB-IIS view source via translate header||117.206.96.68
2       |       POLICY VNC server response||190.209.54.203
```

b) Información por alerta

```
::::::::::::::
ETEXPLOITMicrosoft_Windows_NETAPI_Stack_Overflow_Inbound_-_MS08-067__25).top
::::::::::::::


[  TOP IP ADDRESS ]
[ ET EXPLOIT Microsoft Windows NETAPI Stack Overflow Inbound - MS08-067 (25)  ]
[  DST CONNECTIONS  ]
                  1    ->      132.247.0.57
                  1    ->      132.247.0.18


[  TOP IP ADDRESS ]
[ ET EXPLOIT Microsoft Windows NETAPI Stack Overflow Inbound - MS08-067 (25)  ]
[  SRC CONNECTIONS  ]
                  1    ->      132.247.17.12
                  1    ->      132.247.17.4


[  TOP PORTS ]
[ ET EXPLOIT Microsoft Windows NETAPI Stack Overflow Inbound - MS08-067 (25)  ]
[  DST CONNECTIONS  ]
                  2    ->      [ 445 ]


[  TOP PORTS ]
[ ET EXPLOIT Microsoft Windows NETAPI Stack Overflow Inbound - MS08-067 (25)  ]
[  SCR CONNECTIONS  ]
                  1    ->      [ 2714 ]
                  1    ->      [ 2677 ]



::::::::::::::
NETBIOSSMB-DSsrvsvc_NetrPathCanonicalize_WriteAndX_little_endian_overflow_attempt.top
::::::::::::::


[  TOP IP ADDRESS ]
[ NETBIOS SMB-DS srvsvc NetrPathCanonicalize WriteAndX little endian overflow attempt  ]
[  DST CONNECTIONS  ]
                  1    ->      132.247.0.76
                  1    ->      132.247.0.57
                  1    ->      132.247.0.15
                  1    ->      132.247.0.18
                  1    ->      132.247.0.8
                  1    ->      132.247.0.39


[  TOP IP ADDRESS ]
[ NETBIOS SMB-DS srvsvc NetrPathCanonicalize WriteAndX little endian overflow attempt  ]
[  SRC CONNECTIONS  ]
                  3    ->      132.247.17.4
                  3    ->      132.247.17.12


[  TOP PORTS ]
[ NETBIOS SMB-DS srvsvc NetrPathCanonicalize WriteAndX little endian overflow attempt  ]
[  DST CONNECTIONS  ]
                  6    ->      [ 445 ]
```

ANEXO F – EJEMPLO DE INFORMACIÓN ALMACENADA EN EL TSU

Con el diseño de la base de datos se pueden hacer consultas tan específicas como se deseen. A continuación se muestran dos ejemplos.

a) Incidentes reportados

```
1673745 | 2011-02-03 02:24:51 | 5b799e45c5f0dafa24cbe619231d2cc2 | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673745 | 2011-02-03 02:24:51 | f7f3d106107fbf546fc794241575da7c | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673745 | 2011-02-03 02:24:51 | 01ca38ad531f36358934cd3502b3334c | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673745 | 2011-02-03 02:24:51 | 93c9266d46ec8d198e2098dff76e924d | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673745 | 2011-02-03 02:24:51 | 173dc42d82b8601bed9eed90b24ec081 | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673745 | 2011-02-03 02:24:51 | 153d7cd4d182df7ad684d65d91875c91 | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673745 | 2011-02-03 02:24:51 | f04ede2677abfa415ebdf67c75fceb91 | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673745 | 2011-02-03 02:24:51 | 29d9545b825a128c214c603acc5e13ee | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673745 | 2011-02-03 02:24:51 | 7afc42de0745f7e3047ff02b4b0e57fc | SSH SCAN O POSIBLE SSH BRUTEFORCE ATTACK
1673748 | 2011-02-03 02:26:09 | a0aa4a74b70cbca5a03960df1a3dc878 | SQL WORM 1434
1673749 | 2011-02-03 02:27:08 | 99e30239984e4c331bcc2986643d066b | GENERAL SCAN PORT [17871]
1673751 | 2011-02-03 02:30:01 | d56e7f0145ed8e9b996c169d576bdbef | GENERAL SCAN PORT [19814]
1673752 | 2011-02-03 02:30:40 | 07d39626a6f0a2d4b0bcb98b5d30d1b6 | GENERAL SCAN PORT [33435]
1673754 | 2011-02-03 02:30:54 | 07d39626a6f0a2d4b0bcb98b5d30d1b6 | PORTSWEEP
1673756 | 2011-02-03 02:33:08 | 0262ca5d4446a40394f6f7e83804bdd4 | GENERAL SCAN PORT [18423]
1673757 | 2011-02-03 02:33:16 | 2afe80d2d26ad50b52f0157eacb5870c | GENERAL SCAN PORT [16617]
1673758 | 2011-02-03 02:34:09 | 99e30239984e4c331bcc2986643d066b | GENERAL SCAN PORT [17871]
1673773 | 2011-02-03 02:42:54 | 07d39626a6f0a2d4b0bcb98b5d30d1b6 | PORTSWEEP
1673774 | 2011-02-03 02:43:45 | d56e7f0145ed8e9b996c169d576bdbef | GENERAL SCAN PORT [19814]
1673775 | 2011-02-03 02:43:55 | 99e30239984e4c331bcc2986643d066b | GENERAL SCAN PORT [6305]
1673779 | 2011-02-03 02:45:20 | 40329bee034a98c02454a3479ef5e6a9 | GENERAL SCAN PORT [19838]
1673783 | 2011-02-03 02:49:13 | 4302dbc6caa9c9ec6d0e3f47282746c0 | GENERAL SCAN PORT [19812]
1673784 | 2011-02-03 02:50:28 | b79073b70f3c61c6af2f07997a247b33 | GENERAL SCAN PORT [17871]
1673785 | 2011-02-03 02:50:28 | 99e30239984e4c331bcc2986643d066b | GENERAL SCAN PORT [17871]
1673786 | 2011-02-03 02:50:34 | c90329e824064e777f8719643320923f | GENERAL SCAN PORT [19677]
1673788 | 2011-02-03 03:04:14 | ef401db3ed6bb1ab29c5cdc06cfeb636 | GENERAL SCAN PORT [6305]
1673789 | 2011-02-03 03:04:14 | 99e30239984e4c331bcc2986643d066b | GENERAL SCAN PORT [6305]
1673797 | 2011-02-03 03:06:20 | d56e7f0145ed8e9b996c169d576bdbef | GENERAL SCAN PORT [19814]
1673798 | 2011-02-03 03:06:21 | 16d5ef7e85234f256f5e4b6b70519f27 | GENERAL SCAN PORT [4899]
1673798 | 2011-02-03 03:06:21 | d5e36a27c2d953a56652820e7563fa49 | GENERAL SCAN PORT [4899]
1673799 | 2011-02-03 02:52:13 | c9edc6804196dc694eb346cf62f9067f | GENERAL SCAN PORT [6133]
1673800 | 2011-02-03 02:52:26 | d56e7f0145ed8e9b996c169d576bdbef | GENERAL SCAN PORT [19814]
1673801 | 2011-02-03 02:52:26 | 15fd6c77950d29548982fb44b6ddd4a3 | GENERAL SCAN PORT [19814]
1673802 | 2011-02-03 02:53:58 | a0aa4a74b70cbca5a03960df1a3dc878 | SQL WORM 1434
1673807 | 2011-02-03 02:57:44 | 99e30239984e4c331bcc2986643d066b | GENERAL SCAN PORT [17871]
1673809 | 2011-02-03 02:59:34 | d56e7f0145ed8e9b996c169d576bdbef | GENERAL SCAN PORT [19814]
1673811 | 2011-02-03 02:59:50 | 420bc97a9e2304d22d7382fec7a8571c | GENERAL SCAN PORT [19556]
1673812 | 2011-02-03 03:00:09 | 685d502a075912c4227e12e28c7885a8 | GENERAL SCAN PORT [23680]
1673816 | 2011-02-03 03:00:32 | a0aa4a74b70cbca5a03960df1a3dc878 | SQL WORM 1434
1673819 | 2011-02-03 03:02:57 | a0aa4a74b70cbca5a03960df1a3dc878 | SQL WORM 1434
1673821 | 2011-02-03 03:11:08 | 62abf75a6760ee1a4cdcb2a1fe89d0ac | GENERAL SCAN PORT [19814]
1673823 | 2011-02-03 03:11:32 | 99e30239984e4c331bcc2986643d066b | GENERAL SCAN PORT [6305]
1673826 | 2011-02-03 03:12:20 | a0aa4a74b70cbca5a03960df1a3dc878 | SQL WORM 1434
1673827 | 2011-02-03 03:12:33 | a0aa4a74b70cbca5a03960df1a3dc878 | SQL WORM 1434
1673828 | 2011-02-03 03:14:01 | 3c0a1a13a0469bb1803c5e4270ea91a4 | GENERAL SCAN PORT [51464]
1673833 | 2011-02-03 03:15:43 | d56e7f0145ed8e9b996c169d576bdbef | GENERAL SCAN PORT [19814]
```

b) Detalles de eventos específicos

```
1673734 | 2011-02-03 02:19:48 | 218.15.136.38 | 32911  | 132.248.194.73 | 22 | b83ba7ca61feeb830a04fe4ae889966c
1673734 | 2011-02-03 02:19:52 | 218.15.136.38 | 33158  | 132.248.194.73 | 22 | 7c6147f2417a5b74b1fe9585a43dd676
1673734 | 2011-02-03 02:19:55 | 218.15.136.38 | 33397  | 132.248.194.73 | 22 | 44eec06e38e7649b7d98412b18983087
1673734 | 2011-02-03 02:19:58 | 218.15.136.38 | 33660  | 132.248.194.73 | 22 | aae658a6b4b4ed274c6225c18e8a4346
1673734 | 2011-02-03 02:20:02 | 218.15.136.38 | 33910  | 132.248.194.73 | 22 | a8cbf2a3a2e442fc5059ce1ad9371742
1673734 | 2011-02-03 02:20:05 | 218.15.136.38 | 34155  | 132.248.194.73 | 22 | e5cbae0b5ca3b2a8b022a389339c7d2b
1673734 | 2011-02-03 02:20:09 | 218.15.136.38 | 34397  | 132.248.194.73 | 22 | 311a9b35473b14284d24d6c3d9eea2cb
1673734 | 2011-02-03 02:20:12 | 218.15.136.38 | 34631  | 132.248.194.73 | 22 | b50533d3352fcc9c3855783192ca8081
1673734 | 2011-02-03 02:20:16 | 218.15.136.38 | 34904  | 132.248.194.73 | 22 | e4c6cd45d1c296ff1ec9a87c03507a74
1673734 | 2011-02-03 02:20:19 | 218.15.136.38 | 35127  | 132.248.194.73 | 22 | 5ae5b60ec9eec88177817a728e956e2c
1673734 | 2011-02-03 02:20:23 | 218.15.136.38 | 35374  | 132.248.194.73 | 22 | 49bd374892df8ef3686a10803c4620bc
1673734 | 2011-02-03 02:20:26 | 218.15.136.38 | 35627  | 132.248.194.73 | 22 | d28eedc5a297e4ebffd93edd456bea87
```

ANEXO G – EJEMPLOS DE BITÁCORAS DEL MÓDULO

a) DKN AGENT

```
[2011-01-28 14:24:29] - [AGENT | check_active_events   ] STARTING SCP TRANSFER MODE (NO DB) (360)sec
[2011-01-28 14:24:29] - [AGENT | check_active_events   ] STARTING STAMOD (120)sec
[2011-01-28 14:24:29] - [AGENT | stamod                  ] ERROR Unable to exec STAMOD. Min time must be
(300) sec.
[2011-01-28 14:24:29] - [AGENT | stamod                ] FIXING The program will execute each (300) sec.
[2011-01-28 14:24:29] - [AGENT | stamod                ] Executing STA Module PID Handler(15841)
[2011-01-28 14:24:29] - [AGENT(0) | check_active_events] STARTING VERIFICATION PROCESS
[2011-01-28 14:24:29] - [AGENT(0) | expire_event_file  ]      Expiring event AEV: [X|tcp-109.72.207.69-
53|1296182653|1296182353|GENERAL SCAN PORT [53]|]
[2011-01-28  14:24:29]  -  [AGENT(0)  |  expire_event_file   ]         Adding  incident  to  agent  file  :
[/data/dkn/tmp/AGENT0.exp]
[2011-01-28 14:24:29] - [AGENT(0) | expire_event_file  ]      Expiring event AEV: [X|tcp-92.240.68.152-
80|1296183021|1296182721|GENERAL SCAN PORT [80]|]
[2011-01-28  14:24:29]  -  [AGENT(0)  |  expire_event_file   ]         Adding  incident  to  agent  file  :
[/data/dkn/tmp/AGENT0.exp]
[2011-01-28 14:24:29] - [AGENT(0) | expire_event_file  ]      Expiring event AEV: [X|udp-132.248.204.1-
46594|1296183027|1296182727|GENERAL SCAN PORT [46594]|]
[2011-01-28  14:24:29]  -  [AGENT(0)  |  expire_event_file   ]         Adding  incident  to  agent  file  :
[/data/dkn/tmp/AGENT0.exp]
[2011-01-28 14:24:29] - [AGENT(0) | expire_event_file  ]     Expiring event AEV: [X|udp-132.248.204.1-53-
132.248.194.73|1296183027|1296182730|PORTSWEEP|]
[2011-01-28  14:24:29]  -  [AGENT(0)  |  expire_event_file   ]         Adding  incident  to  agent  file  :
[/data/dkn/tmp/AGENT0.exp]
[2011-01-28 14:24:29] - [AGENT(0) | expire_event_file  ]      Expiring event AEV: [X|tcp-12.139.31.211-
135|1296183095|1296182795|GENERAL SCAN PORT [135]|]
[2011-01-28  14:24:29]  -  [AGENT(0)  |  expire_event_file   ]         Adding  incident  to  agent  file  :
[/data/dkn/tmp/AGENT0.exp]

 [2011-01-28  14:24:33]  -  [AGENT(0)  |  proc_events              ]         Logging  new  Incident  from
[/data/dkn/events_connections/udp-119.248.51.32-37927-1296186220.evt]
[2011-01-28  14:24:33]  -  [AGENT(0)  |  createtar                    ]          Creating  Tar  file
(/data/dkn/events_connections/udp-114.252.89.39-37927-1296186220.tgz)                                  with
(/data/dkn/tmp/attacks/445f18dc3bb3de26d491685b249a9dd1)
[2011-01-28  14:24:33]  -  [AGENT(0)  |  proc_events              ]         Logging  new  Incident  from
[/data/dkn/events_connections/udp-114.252.89.39-37927-1296186220.evt]
[2011-01-28  14:24:33]  -  [AGENT(0)  |  createtar                    ]          Creating  Tar  file
(/data/dkn/events_connections/udp-60.6.52.199-37927-1296186147.tgz)                                    with
(/data/dkn/tmp/attacks/0eb0c3ca996d3648705416c4a69132cf)
[2011-01-28  14:24:33]  -  [AGENT(0)  |  proc_events              ]         Logging  new  Incident  from
[/data/dkn/events_connections/udp-60.6.52.199-37927-1296186147.evt]
[2011-01-28  14:24:33]  -  [AGENT(0)  |  createtar                    ]          Creating  Tar  file
(/data/dkn/events_connections/udp-120.8.105.140-37927-1296186178.tgz)                                  with
(/data/dkn/tmp/attacks/473ddda95ba18f00c4123e7f3e895b76)
[2011-01-28  14:24:33]  -  [AGENT(0)  |  proc_events              ]         Logging  new  Incident  from
[/data/dkn/events_connections/udp-120.8.105.140-37927-1296186178.evt]
[2011-01-28  14:24:33]  -  [AGENT(0)  |  createtar                    ]          Creating  Tar  file
(/data/dkn/events_connections/udp-110.245.128.240-37927-1296186169.tgz)                                with
(/data/dkn/tmp/attacks/29204203152f28cbac02df22e4d86a8
0)
[2011-01-28  14:24:33]  -  [AGENT(0)  |  proc_events              ]         Logging  new  Incident  from
[/data/dkn/events_connections/udp-110.245.128.240-37927-1296186169.evt]
[2011-01-28  14:24:33]  -  [AGENT(0)  |  createtar                    ]          Creating  Tar  file
(/data/dkn/events_connections/udp-124.67.37.34-37927-1296186201.tgz) with

[2011-01-28  14:30:34]  -  [AGENT(33)  |  proc_events              ]         Logging  new  Incident  from
[/data/dkn/events_connections/udp-132.248.204.1-53-132.248.194.73-1296246342.evt]
[2011-01-28  14:30:35]  -  [AGENT  |  transfer_file                    ]             Transfering
(192.168.1.1:2290|dkn|****|/data/incidents/|/data/dkn/events_connections/alert-TCP-201.127.23.156-
COMMUNITYWEB-ATTACKSGFIMailSecurit
yManagementHostOverflowAttemptLongAcceptParameter.det) PID Handler (15840)->(21512)
[2011-01-28  14:30:35]  -  [AGENT  |  transfer_file                    ]             Transfering
(192.168.1.1:2290|dkn|****|/data/incidents/|/data/dkn/events_connections/alert-TCP-189.136.44.47-
COMMUNITYWEB-ATTACKSGFIMailSecurity
ManagementHostOverflowAttemptLongAcceptParameter.det) PID Handler (15840)->(21512)
[2011-01-28  14:30:35]  -  [AGENT  |  transfer_file                    ]             Transfering
(192.168.1.1:2290|dkn|****|/data/incidents/|/data/dkn/events_connections/tcp-221.172.58.255-445-
1296183060.det) PID Handler (15840)-
>(21511)
[2011-01-28  14:30:35]  -  [AGENT  |  transfer_file                    ]             Transfering
(192.168.1.1:2290|dkn|****|/data/incidents/|/data/dkn/events_connections/alert-TCP-187.137.97.229-
COMMUNITYWEB-ATTACKSGFIMailSecurit
yManagementHostOverflowAttemptLongAcceptParameter.det) PID Handler (15840)->(21512)
```

b) DKN STA

```
[2011-01-28 14:29:29] - ************* [STRUCTURED TRAFFIC ANALYSIS MODULE] **************
[2011-01-28 14:29:29] - ************************************************************

[2011-01-28 14:29:29] - [STAMOD | getdata       ] STARTING STA MODULE
[2011-01-28 14:29:29] - [STAMOD | getdata       ] |- Creating STA log directory        (/data/dkn/stamod/20110128-1429) ->
[2011-01-28 14:29:29] - [STAMOD | getdata       ] |- Creating STA Argus directory      (/data/dkn/stamod/20110128-1429/argus)
->
[2011-01-28 14:29:29] - [STAMOD | getdata       ] |- Creating STA Snort directory      (/data/dkn/stamod/20110128-1429/snort)
->
[2011-01-28 14:29:29] - [STAMOD | getdata       ] |- Creating STA procfiles directory   (/data/dkn/stamod/20110128-
1429/snort/procfiles/) ->
[2011-01-28 14:29:29] - [STAMOD | getdata       ] |- Creating STA alert top directory   (/data/dkn/stamod/20110128-
1429/snort/alert_top/) ->
[2011-01-28 14:29:29] - [STAMOD | getdata       ] |- Creating STA alert info directory  (/data/dkn/stamod/20110128-
1429/snort/alert_info/) ->
[2011-01-28 14:29:29] - [STAMOD | argus_exec    ] |- Stopping argus daemon ... OK
[2011-01-28 14:29:29] - [STAMOD | argus_exec    ] |- Starting argus daemon ... OK
[2011-01-28 14:29:29] - [STAMOD | argus_data    ] |- GENERATING ARGUS DATA
[2011-01-28 14:29:29] - [STAMOD | argus_data    ]      |- Creating racount file          ... OK
[2011-01-28 14:29:32] - [STAMOD | argus_data    ]      |- Creating hosts file            ... OK
[2011-01-28 14:29:37] - [STAMOD | argus_data    ]      |- Creating traffic sessions file  ... OK
[2011-01-28 14:29:44] - [STAMOD | argus_data    ]      |- Creating traffic sessions of defined rules ->
[2011-01-28 14:29:44] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-22-[-]>  ... OK
[2011-01-28 14:29:46] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-22>  ... OK
[2011-01-28 14:29:48] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-1433-[-]>  ... OK
[2011-01-28 14:29:51] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-1433>  ... OK
[2011-01-28 14:29:53] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-1434-[-]>  ... OK
[2011-01-28 14:29:55] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-1434>  ... OK
[2011-01-28 14:29:58] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <udp-1434-[-]>  ... OK
[2011-01-28 14:30:00] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <udp-1434>  ... OK
[2011-01-28 14:30:02] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-6666-[PING]>  ... OK
[2011-01-28 14:30:04] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-6666>  ... OK
[2011-01-28 14:30:07] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-6667-[PONG]>  ... OK
[2011-01-28 14:30:08] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-6667>  ... OK
[2011-01-28 14:30:11] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-6668-[-]>  ... OK
[2011-01-28 14:30:13] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-6668>  ... OK
[2011-01-28 14:30:15] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-6669-[-]>  ... OK
[2011-01-28 14:30:18] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-6669>  ... OK
[2011-01-28 14:30:20] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-25-[-]>  ... OK
[2011-01-28 14:30:22] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-25>  ... OK
[2011-01-28 14:30:25] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-5900-[-]>  ... OK
[2011-01-28 14:30:27] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-5900>  ... OK
[2011-01-28 14:30:29] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-139-[-]>  ... OK
[2011-01-28 14:30:32] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-139>  ... OK
[2011-01-28 14:30:36] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-445-[-]>  ... OK
[2011-01-28 14:30:46] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-445>  ... OK
[2011-01-28 14:30:53] - [STAMOD | argus_radata_r]         |- Creating stats REGEX PAYLOAD for rule <tcp-9000-[PING]>  ... OK
[2011-01-28 14:30:56] - [STAMOD | argus_radata_r]         |- Creating stats GENERAL for rule <tcp-9000>  ... OK
[2011-01-28 14:30:59] - [STAMOD | snort_data     ] |- STARTING SNORT ANALYSIS WITH (/data/dkn/stamod/20110128-1429/20110128-
1429.cap) and OUTDIR (/data/dkn/stamod/20110128-1429) SNORTCONF (/usr/local/snort/etc/s
nort.conf)    ...  [2011-01-28  14:31:33]  -  [STAMOD  |  snort_data        ]            |-  Setting  SNORT  alert  file  to
(/data/dkn/stamod/20110128-1429/snort/procfiles//sta_snortalert) ->
[2011-01-28 14:31:33] - [STAMOD | snort_data     ]        |- Setting SNORT log file (pcap) to (/data/dkn/stamod/20110128-
1429/snort/procfiles//sta_snortalert) ... OK
[2011-01-28  14:31:33]  -  [STAMOD  |  convalert        ]          |-  Converting  to  pipe  format  (/data/dkn/stamod/20110128-
1429/snort/procfiles//sta_snortalert)
[2011-01-28 14:31:33] - [STAMOD | snort_top     ]      |- Starting SNORT top processing
[2011-01-28 14:31:33] - [STAMOD | snort_alert   ]      |- Processing information of top alerts [dstip]
[2011-01-28 14:31:33] - [STAMOD | snort_alert   ]      |- Processing information of top alerts [srcip]
[2011-01-28 14:31:33] - [STAMOD | snort_alert   ]      |- Processing information of top alerts [dstport]
[2011-01-28 14:31:33] - [STAMOD | snort_alert   ]      |- Processing information of top alerts [srcport]
[2011-01-28 14:31:33] - [STAMOD | getdata       ] |- Verifying argus daemon or retrying start ... [2011-01-28 14:31:33] -
[STAMOD | argus_exec    ] |- Starting argus daemon ... [2011-01-28 14:31:33] -
Process terminated.
```