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:

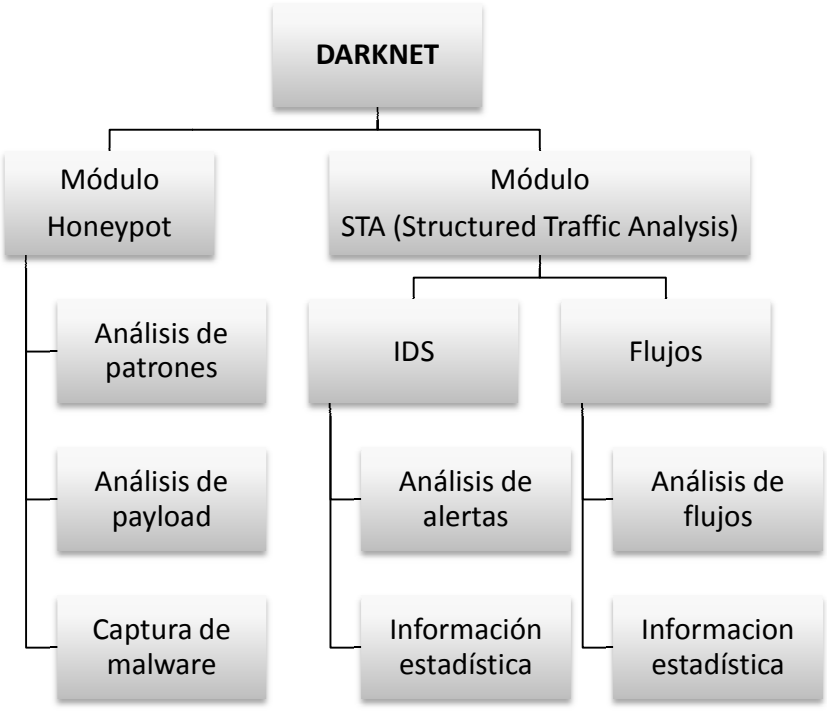


Figura 21. Esquema general del sistema de la Darknet UNAM

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:

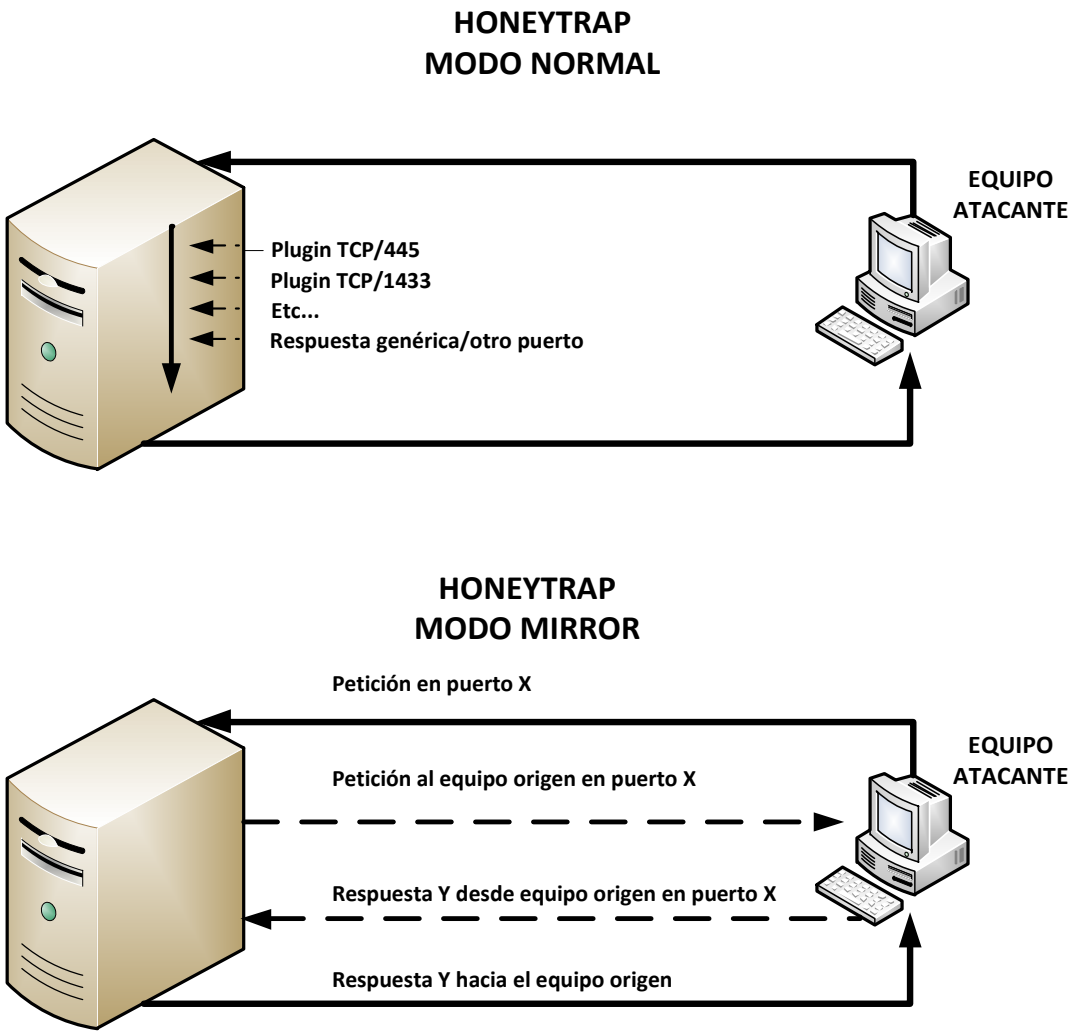


Figura 22. Modos de funcionamiento de honeytrap

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.

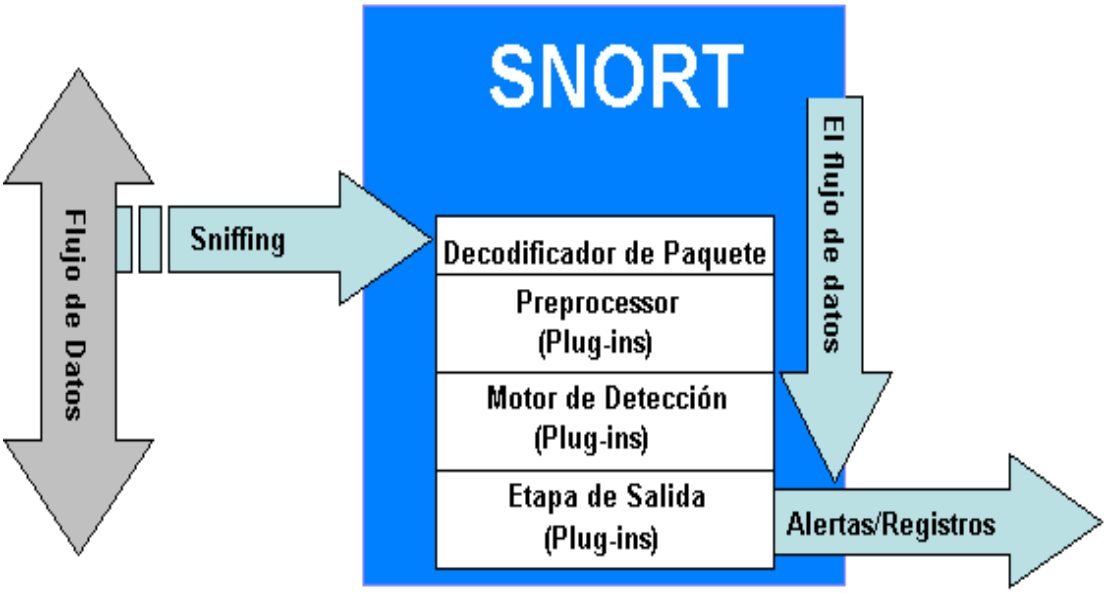


Figura 23. Diagrama de funcionamiento del IDS Snort

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.

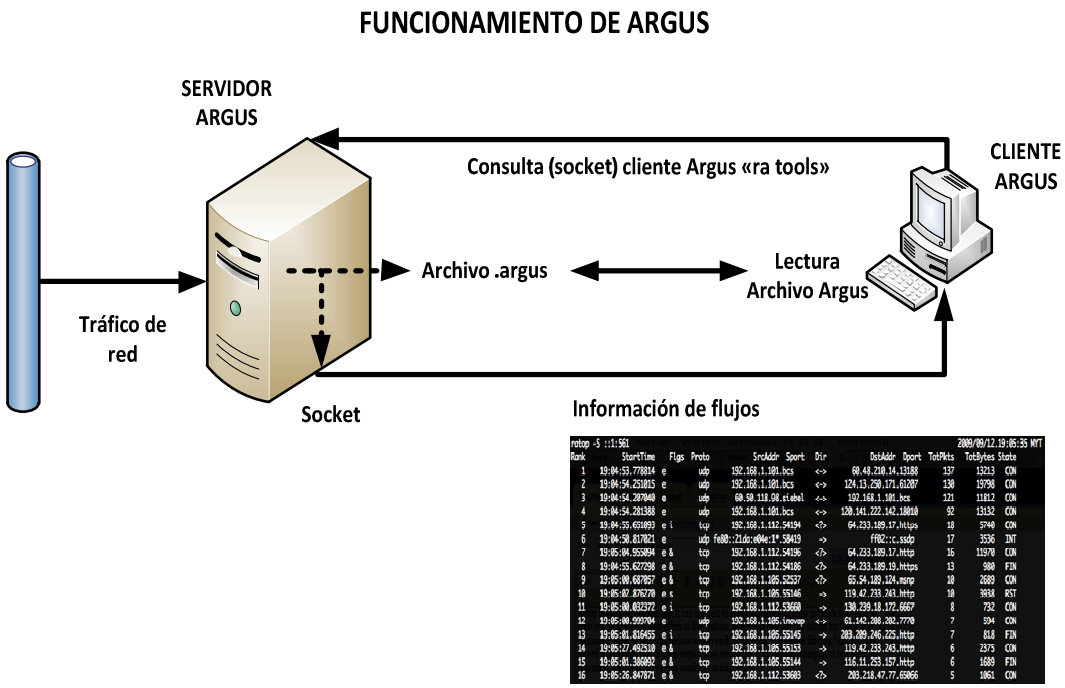


Figura 24. Funcionamiento de Argus

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:

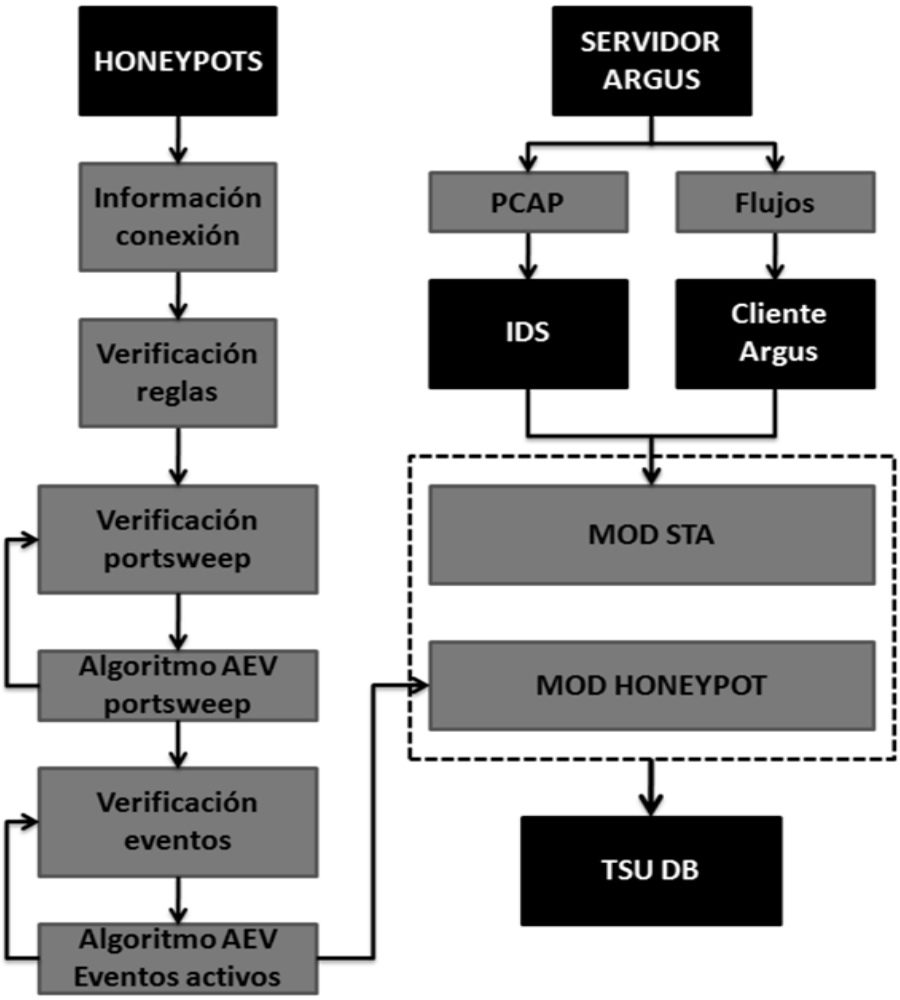


Figura 26. Proceso de manejo de información de la Darknet UNAM

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.

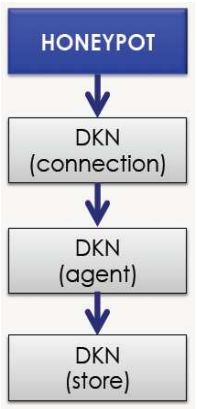
4.3.2 FUNCIONAMIENTO MÓDULO HONEYPOT

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:



Recibe y maneja la conexión. Envía la información al módulo DKN

Recibe la información de conexión y analiza patrones de tráfico

Analiza payloads y procesa información

Almacena información en la DB

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.

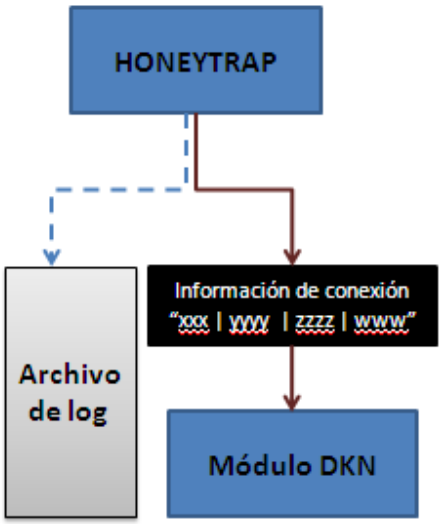


Figura 28. Adaptación de honeytrap para implementación en Darknet UNAM

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:

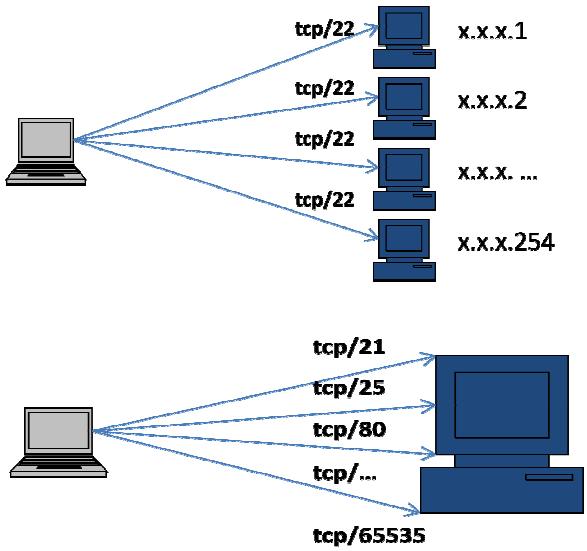


Figura 29. Patrones identificados por el módulo DKN

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:

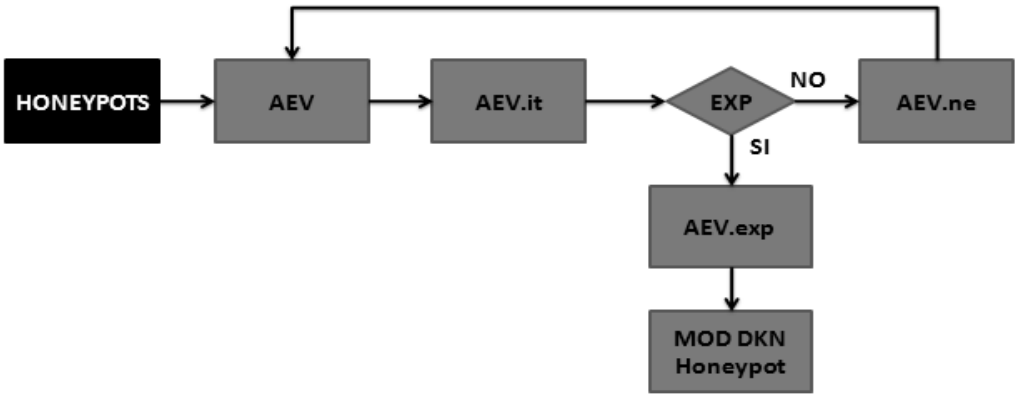


Figura 30. Algoritmo expiración y manejo de cola de eventos

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:

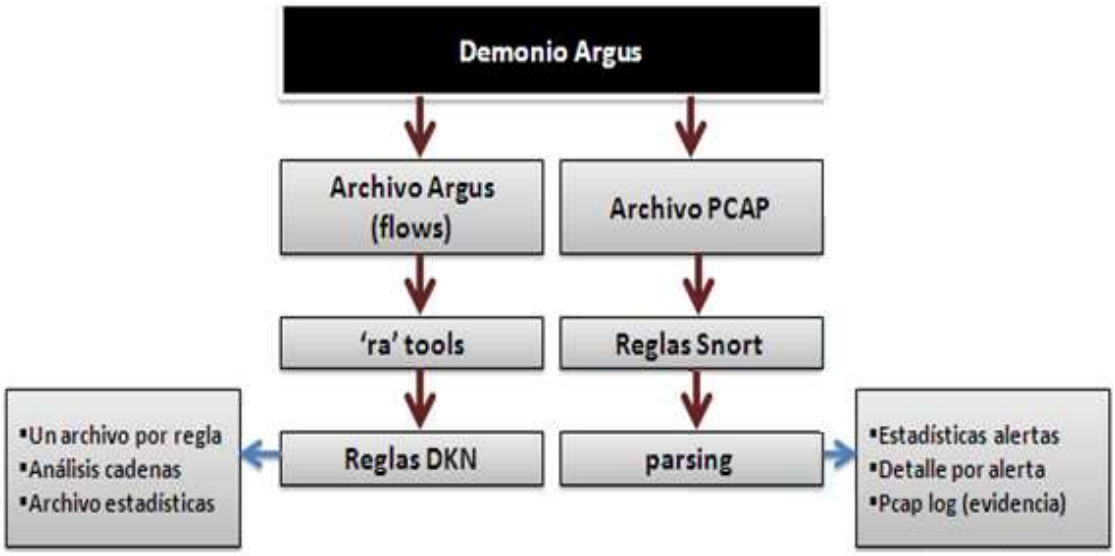


Figura 31. Esquema de funcionamiento módulo STA

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.