JNIC2016

II Jornadas Nacionales de Investigación en Ciberseguridad

Granada 15–17 de junio

GOBIERNO DE ESPAÑA
MINISTERIO DE INDUSTRIA, ENERGÍA Y TURISMO

10 incibe_
2006-2016
TRABAJANDO POR LA CONFIANZA DIGITAL

ugr Universidad de Granada

Detectando botnets con dos modelos de abstracción: Flujos de tráfico e inspección de paquetes de red

Manuel Gil Pérez
Dpto. Ingeniería de la Información y las Comunicaciones
Universidad de Murcia, 30071 Murcia, España
Email: mgilperez@um.es

Gregorio Martínez Pérez
Dpto. Ingeniería de la Información y las Comunicaciones
Universidad de Murcia, 30071 Murcia, España
Email: gregorio@um.es

***Abstract*—La detección de redes botnet se ha convertido en los últimos años en una prioridad de primer orden. Los ataques por Denegación Distribuida de Servicio (DDoS) se han incrementado en dos años en más de un 200% debido a estas redes, donde los atacantes tienen a su disposición miles o millones de dispositivos comprometidos para ejecutar remotamente sus ataques DDoS. En este artículo se propone la detección y mitigación de las redes botnet mediante un doble análisis de las fuentes de información. Primero, realizando la monitorización de los flujos del tráfico de red y, posteriormente, confirmando el supuesto ataque mediante una inspección profunda de los paquetes de red.**

***Index Terms*—Ataques por denegación de servicios, detección de botnets, gestión del riesgo, amenazas heterogéneas**

Tipo de contribución: *Investigación en desarrollo*

I. Introducción

Los ataques han evolucionado en los últimos años desde un modo de actuación de los atacantes centrado en un único punto de origen, desde su propia máquina, hacia un enfoque más avanzado donde el ataque se lanza desde múltiples puntos de la red. Los atacantes han optado por hacer uso de sistemas comprometidos previamente para lanzar un ataque orquestado hacia un objetivo común, creando para ello una red botnet con cientos o miles de *bots* que el atacante puede controlar de manera remota [1], [2]. Uno de los objetivos de estos atacantes es la ejecución de un ataque de Denegación Distribuida de Servicio (DDoS), donde el atacante intenta deshabilitar algún servicio del sistema objetivo inundando su red de tráfico y/o sobrecargando sus recursos computacionales [3].

Las redes botnet han sido reconocidas por Joseph Demarest, director de la ciberdivisión del FBI, como una de las mayores amenazas actuales de Internet. Ante un Comité del Senado de Estados Unidos, Demarest ha afirmado que "cada segundo, 18 ordenadores pasan a formar parte de una red botnet" [4]. Este hecho se traduce en que unos 567 millones de equipos se ven comprometidos por año, donde se llegan a estimar pérdidas económicas rondando los 110 000 millones de dólares a nivel mundial. Como solución, en [5] se estudian diversas técnicas para detectar y desmantelar redes botnet, donde caben destacar las basadas en la detección de solicitudes a servidores DNS, para localizar el *servidor C&C* (Command & Control), y las basadas en minería de datos, cuyo objetivo es el análisis del tráfico de red para detectar los canales C&C. Estos canales, proporcionados por un servidor C&C central, son utilizados por el atacante para dirigir de manera remota todos los bots ante la proyección de su ataque orquestado.

A pesar de los esfuerzos que se están haciendo, los Sistemas de Detección de Intrusiones (IDS) capaces de poder detectar la ejecución de redes botnet, utilizando alguna de las técnicas descritas más arriba, se han visto sobrepasadas por tener que gestionar enormes cantidades de información en tiempo real; normalmente, paquetes de red. Este problema se debe a que las redes actuales proporcionan tasas de transmisión cada vez más altas, pasando en muy pocos años de los 100 Mbps a los 10 Gbps actuales. Esto hace que la información que fluye por una red sea demasiado grande para que un IDS sea capaz de recuperar y analizar cada paquete de red. Por ejemplo, Snort funciona correctamente en redes de hasta 1 Gbps, pero a partir de 1.5 Gbps empieza a sufrir problemas descartando paquetes de red por sobrecarga [6]. Cualquier solución que se presente con el afán de minimizar esa tasa de pérdida de paquetes se prevee insuficiente en un futuro cercano, ya que se vaticinan redes con velocidades de transmisión todavía más elevadas. En concreto, existe un proyecto en marcha llamado Advanced Networking Initiative, financiado por el gobierno de Estados Unidos, con el que pretenden alcanzar velocidades de 1 Tbps, teniendo ya cuatro laboratorios conectados bajo una red de comunicaciones de 100 Gbps [7].

A raíz de la enorme cantidad de información que los IDSs deben de gestionar para sus propósitos de detección, muchos trabajos han propuesto que esa detección de ataques se lleve a cabo mediante el análisis de *flujos del tráfico de red* en lugar de hacerlo con el uso de paquetes de red. Lo que pretenden es una evolución de la Inspección Profunda de Paquetes (DPI) que hacen los IDSs actuales hacia un modelo donde el análisis de las fuentes de información (i.e., paquetes de red) se realice a un nivel de abstracción más alto [8], [9]. Es decir, moviendo mayoritariamente el proceso de análisis desde el *payload* de los paquetes hacia sus cabeceras [10]. Además de minimizar drásticamente la gran cantidad de información a analizar, este nuevo enfoque también permite examinar tráfico encriptado; uno de los grandes problemas que sufren los IDSs actuales al tener que examinar el payload de los paquetes de red.

A. Contribuciones del presente trabajo

El objetivo principal detrás del presente trabajo se centra en la detección y mitigación de redes botnet capaces de ejecutar ataques DDoS de manera orquestada. La solución propuesta posibilita la detección de estas redes botnet y su mitigación de manera reactiva, cuando el ataque DDoS se está ejecutando, pero también proactivamente antes de su ejecución.

De manera más específica, la detección se desacopla en dos fases bien diferenciadas, aunque complementarias entre sí. En una primera fase, el análisis del tráfico de red se realiza desde un punto de vista de alto nivel, monitorizando únicamente los flujos del tráfico de red, ya que un análisis en profundidad no

es todavía viable por la gran cantidad de paquetes circulando por la red. Esta fase de detección a alto nivel se ha postulado como uno de los escenarios a tratar en un proyecto nacional y en marcha llamado DHARMA (*Análisis y Gestión Dinámica de Riesgos con Amenazas Heterogéneas*) [11], [12].

En una segunda fase de detección, una vez que la primera haya detectado una lista de bots sospechosos de formar parte de una red botnet, se despliegan en el sistema un conjunto de IDSs capaces de confirmar la existencia real de dicha botnet. Es decir, la detección traslada su objetivo de análisis desde un punto de vista de alto nivel hacia uno más bajo con procesos que permitan una inspección profunda de paquetes (DPI). Para dar solución a esta fase de detección a bajo nivel se utilizan los resultados obtenidos en el proyecto nacional RECLAMO (*Red de Sistemas de Engaño Virtuales y Colaborativos basados en Sistemas Autónomos de Respuesta a Intrusiones y Modelos de Confianza*) [13], [14]. Entre los resultados de este proyecto, también se proporcionó un modelo que era capaz de inferir la respuesta más apropiada al tipo concreto de ataque, el cual se puede utilizar aquí como contramedida frente a la detección de una red botnet. Este modelo se basa en el concepto *deception responses*: desviar el ataque hacia una red trampa (*honeynet*) donde se confina para su posterior análisis.

Ambos proyectos nacionales de investigación (DHARMA y RECLAMO) se presentan con más detalle en la Sección III.

B. Organización del artículo

Este artículo de investigación se estructura como se indica a continuación. En la Sección II se revisan varios trabajos sobre la detección de redes botnet y ataques DDoS. La Sección III detalla la arquitectura de componentes que se propone en este trabajo para la detección de estas redes botnet, mientras que la Sección IV se adentra en detalles sobre cómo poder realizar dicha detección, así como un conjunto de contramedidas que se proponen para su mitigación. Por último, en la Sección V se hace un esbozo de las conclusiones extraídas de este trabajo y se postulan ciertas vías de investigación futuras.

II. Trabajo relacionado

Continuando con la discusión de la sección anterior, se ha comentado que las competencias que deben tener los distintos IDSs para sus propósitos de detección implica la adopción de "buenos" algoritmos con unas grandes y precisas capacidades de detección, así como un rápido procesamiento de los datos que obtiene de las fuentes de información; ante todo, paquetes de red. Sin estas capacidades, los IDSs no podrán realizar sus procesos de monitorización y análisis en tiempo real, haciendo casi imposible la detección de los ataques. Teniendo en mente esta problemática, existen trabajos que hacen modificaciones a Snort –estándar de facto en la detección de intrusiones– para alcanzar velocidades más altas sin pérdida de paquetes de red. Por ejemplo, Gnort es una solución basada en Snort donde las operaciones que tienen un alto coste computacional, como la comprobación de coincidencia con un determinado patrón de detección, son ejecutadas por la tarjeta gráfica en lugar de por la CPU [15]. Esta solución consigue que la adaptación de la herramienta Snort pueda funcionar correctamente sobre redes con velocidades de 2.3 Gbps. A pesar de ello, Gnort está lejos de gestionar gran cantidad de información en tiempo real en las nuevas redes con altas tasas de transmisión.

Para seguir mejorando ese rendimiento en el análisis de la información, otros trabajos se centraban en técnicas avanzadas de paralelización sobre tecnologías hardware. Concretamente, en [16] se presenta un estudio completo de las ventajas y los inconvenientes de varias técnicas de paralelización, como las basadas en FPGA (Field Programmable Gate Array), llegando a soportar velocidades de hasta 4 Gbps sin pérdida [17], o las basadas en ASIC (Application-Specific Integrated Circuit) que alcanzaban velocidades de 7.2 Gbps [18]. A pesar de que estas soluciones daban mejoras sustanciales, también son enfoques hardware cuyos costes de implantación son elevados.

Debido a la problemática que existe en no poder gestionar grandes cantidades de información en tiempo real, por parte de los IDSs actuales (como Snort, por ejemplo), ha conducido a que muchos trabajos de investigación trasladen los objetivos de dicha gestión (monitorización y análisis) hacia un nivel de abstracción más alto. A este respecto, en los últimos años han aparecido trabajos que abogan por llevar a cabo esa gestión monitorizando, exclusivamente, los flujos del tráfico de la red. Entre esa lista de trabajos, en [19] se presenta un conjunto de métricas de bajo nivel con las que poder detectar los ataques DDoS mediante el análisis de flujos del tráfico de red. En este trabajo se propone el uso de mapas auto-organizados (SOM) como una red neuronal artificial no supervisada, usando para ello una serie de las métricas anteriores para caracterizar los flujos del tráfico de red durante la fase de entrenamiento. En otro trabajo más actual, en [20], se presenta un prototipo para la detección de ataques DDoS en una red en producción con NetFlow –protocolo de Cisco Systems para analizar flujos de tráfico de la red. Los autores de este trabajo demuestran con experimentos en una red real que es posible detectar ataques DDoS analizando solamente sus flujos de red, aunque también afirman que su prototipo conlleva un 20% adicional en el uso de la CPU de los routers Cisco Catalyst 650.

Los trabajos anteriores se centran en la detección de ataques DDoS de manera reactiva, cuando el ataque se está llevando a cabo. Sin embargo, también existen otros trabajos centrados en la detección de manera proactiva de este tipo de ataques mediante la detección y el desmantelamiento de redes botnets, causantes principales de ataques DDoS. En la literatura actual se pueden encontrar estudios que analizan en profundidad no sólo trabajos que permiten realizar esa detección, sino también la topología y arquitectura detrás de las redes botnets y cómo poder llegar a detectarlas [2], [21]. De esos trabajos, destacar BotHunter [22] y BotMiner [23], dos de los mecanismos más populares en la detección de redes botnet, aunque focalizados en la inspección del payload de los paquetes de red.

A pesar de los grandes avances realizados por los trabajos anteriores, la detección de redes botnet analizando solamente los flujos del tráfico de la red puede conducir a una detección insuficiente. Confirmar la existencia de dicha red maliciosa es necesaria antes de lanzar las contramedidas oportunas con las que mitigar esa presunta red botnet. Debido a ello, el presente trabajo propone una solución para la detección y mitigación de redes botnet mediante su análisis a dos niveles de abstración: primero a alto nivel, analizando exclusivamente los flujos del tráfico de red; y, posteriormente, a bajo nivel para confirmar la existencia de dicha red, realizando una inspección profunda de paquetes (DPI) con los IDSs tradicionales.

III. Arquitectura para detectar redes botnet

La detección y mitigación de redes botnet se ha establecido como objetivo de uno de los escenarios planteados dentro del proyecto nacional DHARMA (Análisis y Gestión Dinámica de Riesgos con Amenazas Heterogéneas). Bajo sus componentes, DHARMA propone gestionar y evaluar de manera dinámica el nivel de riesgo de los activos de una organización cuando está bajo la amenaza de ser objetivo de ataque contra su seguridad. En particular, en DHARMA se propone el cálculo del riesgo existente en el posible establecimiento de una red botnet bajo sus dominios, analizando para ello los flujos del tráfico de red de todos sus activos. Es decir, detectando una presunta botnet a un nivel alto de abstracción a través del módulo Evaluación Dinámica del Riesgo (DRA) propuesto por DHARMA.

En la Fig. 1 se muestra una visión a alto nivel de las capas para el DRA del proyecto DHARMA.

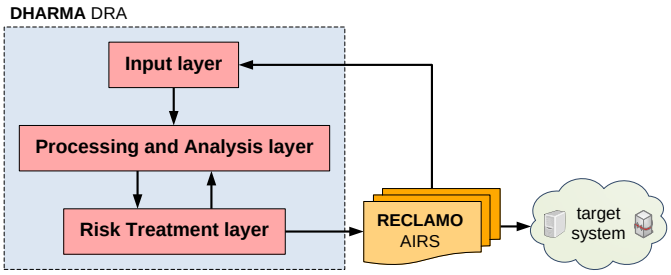

Fig. 1. Capas para la evaluación dinámica del riesgo en DHARMA.

Cuando el nivel de riesgo por la presencia de una presunta red botnet es alto, antes de lanzar las contramedidas oportunas para su desmantelamiento (i.e., la fase de mitigación) se tiene que confirmar la existencia real de esa red botnet; es decir, no es un *falso positivo*. La sospecha infundada de un alto riesgo en que exista una red botnet real puede inducir al DRA a que ponga en marcha unos mecanismos de respuesta que podrían interferir en el correcto funcionamiento de los activos que se están protegiendo. Una segunda fase de detección a más bajo nivel es por tanto necesaria, haciéndola mediante un análisis en profundidad de los paquetes que fluyen por la red.

A este respecto, y como se puede ver en la Fig. 1, esta nueva detección se realiza con el Sistema Autónomo de Respuesta a Intrusiones (AIRS) obtenido como uno de los resultados del proyecto RECLAMO (Red de Sistemas de Engaño Virtuales y Colaborativos basados en Sistemas Autónomos de Respuesta a Intrusiones y Modelos de Confianza). Cuando esta segunda fase de detección confirme la existencia de la red botnet, como respuesta para su mitigación final se pone en marcha una red trampa (honeynet) donde se desvía el tráfico de la red botnet para su aislamiento y posterior análisis. La construcción y el despliegue de la honeynet es resultado de RECLAMO.

A continuación se presentan los dos proyectos anteriores en detalle, ambos financiados por el MINECO, en los cuales se muestran sus componentes para la detección y mitigación de redes botnet, objetivo de este trabajo de investigación.

A. *DHARMA: Análisis y gestión dinámica del riesgo*

El objetivo principal de DHARMA es asistir en la gestión y evaluación dinámica del riesgo en tiempo real, a fin de ofrecer mecanismos innovadores de defensa para prevenir y/o mitigar amenazas potenciales sobre los activos de una organización.

La estructuración de la arquitectura multinivel propuesta en este proyecto se establece en tres capas de procesamiento (ver Fig. 1). Los objetivos de cada capa son las siguientes:

- *Input layer*: conjunto de gran variedad de sensores para la monitorización de los diversos activos de la organización, sus vulnerabilidades y potenciales amenazas ante las que podrían ser víctimas de un ataque, como ataques DDoS conducidos por una red botnet.
- *Processing and Analysis layer*: componente principal de DHARMA. Es decir, el controlador DRA diseñado para evaluar dinámicamente el riesgo de los activos y también lanzar las contramedidas correspondientes según el nivel de impacto y riesgo calculado previamente.
- *Risk Treatment layer*: conjunto de contramedidas que se pueden llevar a cabo para reducir el riesgo de la amenaza detectada, e incluso mitigar sus efectos.

Aunque todas las capas anteriores son necesarias, quizá la de *Processing and Analysis* sea la más importante por albergar el controlar DRA. Este controlador se encarga de normalizar, correlacionar, procesar y analizar los datos recibidos de la fase de monitorización de los activos. Además, también será capaz de volver a evaluar el nivel de impacto y riesgo de los activos en cada momento y, si fuera necesario, lanzar la respuesta o contramedida más apropiada para su tratamiento.

Con respecto a la detección a un nivel alto de abstracción de redes botnet, en la Fig. 2 se presenta la arquitectura de los componentes de DHARMA adaptándolos a este escenario en concreto para la detección de redes botnet mediante el análisis de los flujos del tráfico de red. En [12] se puede encontrar la arquitectura completa de DHARMA, mientras que en la Fig. 2 sólo se muestra el escenario para detectar redes botnet.

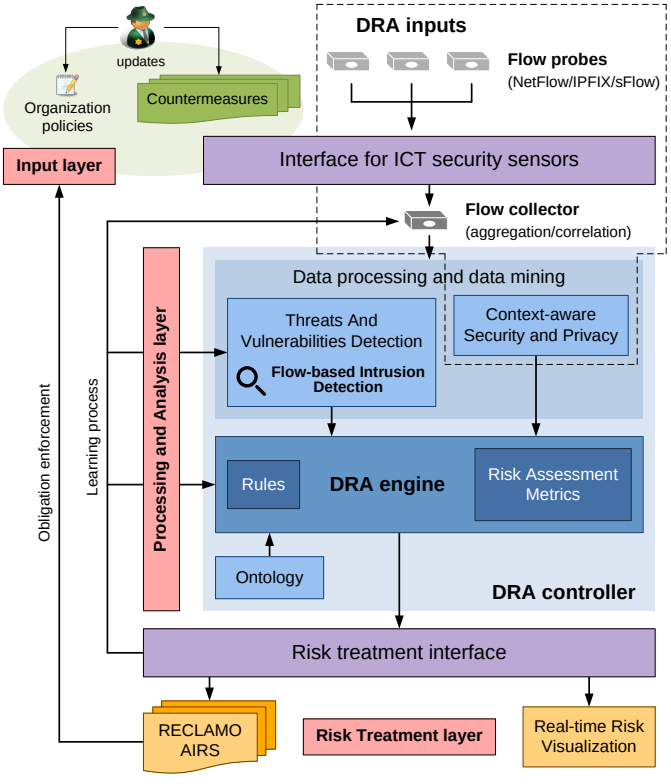

Fig. 2. Arquitectura de DHARMA adaptada para la detección de las redes botnet mediante flujos de red.

Como se puede ver, en la capa superior de la Fig. 2 (*Input layer*) se encuentran los sensores que se hayan constantemente recopilando los flujos del tráfico que fluye por la red, llamados *Flow probes*, relacionados con todos los componentes de red instalados en el sistema. La monitorización gracias al conjunto de sensores, estratégicamente desplegados en el sistema, van a permitir que se extraigan ciertas métricas con las que llegar a detectar ataques DDoS, notificando sobre un alto tráfico de la red (*congestión*), e identificar posibles conexiones periódicas con un presunto servidor C&C, las cuales siguen normalmente un patrón de comportamiento muy específico. Ambos tipos de detección habilitan la posibilidad de detectar redes botnet de manera reactiva, durante la ejecución del ataque DDoS debido a una alta congestión del tráfico de red, y de manera proactiva antes de que su propietario (*botmaster*) difunda los comandos necesarios para lanzar un próximo ataque DDoS.

Las estadísticas que generan los sensores sobre el tráfico de la red se envían a un colector, llamado *Flow collector*, para su almacenamiento y posterior análisis. Entre los formatos que se pueden utilizar, para el intercambio de flujos de red entre esos componentes, se propone el uso de estándares bien conocidos como, por ejemplo, NetFlow, IPFIX o sFlow. Este colector es el encargado de aunar todas estas estadísticas, representando las métricas necesarias para la detección de redes botnet y/o la ejecución de un ataque DDoS, y lanzar entonces los procesos de agregación y correlación de toda esa información.

Posteriormente, la capa de procesamiento (*Processing and Analysis*) recibirá estadísticas de las métricas anteriores y, a través del módulo *Threats And Vulnerabilities Detection*, será capaz de realizar un proceso de detección a alto nivel sobre la existencia de una red botnet o la ejecución de un ataque DDoS según la congestión de red que haya. Esta detección se lleva a cabo analizando solamente los flujos del tráfico de red de los activos que están siendo monitorizados. Dependiendo de este proceso de detección, el motor para evaluar dinámicamente el riesgo (*DRA engine*) será el encargado de calcular el nivel de impacto y riesgo asociados a esos activos.

La exactitud y eficiencia al evaluar el nivel de impacto y el riesgo tiene una estrecha relación con qué tecnología el DRA va a inferir dicho cálculo, pero también teniendo en cuenta el conjunto de métricas (distintas a las de detección) que utilice para cuantificar las distintas variables del sistema. Entre estas métricas de evaluación se pueden tener en mente, por ejemplo, la importancia del activo que se está protegiendo, el impacto global para la organización de la amenaza detectada o métricas específicas que permitan estimar el tiempo de exposición. De entre esas posibles tecnologías, destacar el uso de la teoría de la evidencia de Dempster-Shafer, Modelos Ocultos de Markov (HMM) o modelos estadísticos bayesianos, entre otros.

Una vez que el controlador DRA haya analizado y evaluado el nivel de impacto y riesgo, en DHARMA se concretan varias formas de tratamiento de esos resultados. Referente al trabajo de investigación que aquí se presenta, en respuesta a la posible detección de una red botnet o la ejecución de un ataque DDoS se propone utilizar el AIRS que se ha obtenido como resultado del proyecto RECLAMO. El objetivo es conocer si realmente existe dicha amenaza (red botnet o un ataque DDoS) antes de lanzar la respuesta definitiva para mitigarla, habilitando para ello la inspección profunda de paquetes de RECLAMO.

B. RECLAMO: Sistema autónomo de respuesta a intrusiones

El objetivo principal de RECLAMO era la investigación de nuevas soluciones de reacción frente a ataques de red, con la idea de ir un paso más allá a los enfoques tradicionales de los IDSs existentes con métodos clásicos de detección y reacción.

Por tanto, y como salida de la capa de tratamiento del riesgo (*Risk Treatment*) de las Figs. 1 y 2, en este trabajo se propone que la detección a un alto nivel de abstracción tanto de redes botnet como de la ejecución de ataques DDoS, analizando en ambos casos los flujos del tráfico de red que se ha presentado más arriba como objetivo del proyecto DHARMA, se llegue a confirmar poniendo en marcha un nuevo proceso de detección a más bajo nivel mediante RECLAMO. Es decir, analizando ahora en mayor detalle los paquetes de red de todos aquellos dispositivos o componentes que se han identificado a través de DHARMA como sospechosos de ser parte de una red botnet, o también potenciales sospechosos de ser causantes del ataque DDoS detectado con anterioridad.

En la Fig. 3 se presenta la arquitectura que fue diseñada en el proyecto RECLAMO para la detección y la correspondiente reacción de manera autónoma frente a ataques de red.

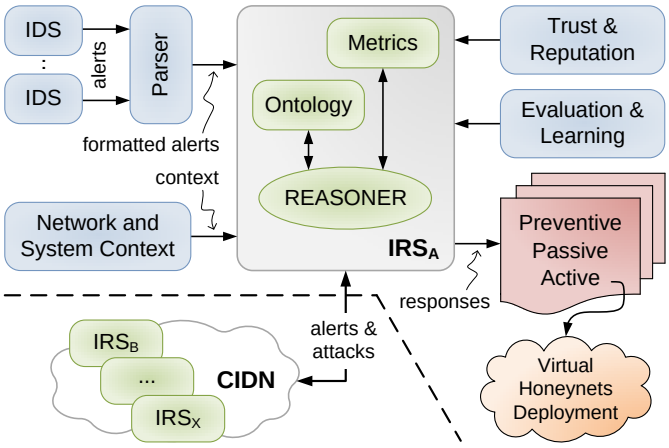

Fig. 3. Arquitectura de RECLAMO para la detección autónoma de ataques.

Los presuntos ataques que pueden ser detectados, a partir de las alertas que son generadas por los distintos IDSs de manera individual, son analizados en tiempo real mediante un modelo de intrusiones, respuestas y métricas de seguridad con las que lanzar un proceso de inferencia para analizar en profundidad la amenaza detectada. Conceptos como los sistemas autónomos, ontologías, gestión de la confianza mediante la reputación, así como las redes colaborativas para la detección colaborativa de amenazas se encuentran claramente identificados en la Fig. 3. Todos estos conceptos se consideran como parte fundamental en el AIRS propuesto por RECLAMO. En [14] se puede ver información más detallada sobre este proyecto.

Cuando la confirmación de la presencia de una red botnet, o ejecución de un ataque DDoS, haya tenido el éxito esperado, como respuesta se pone en marcha la fase de reacción definida y desarrollada en el proyecto RECLAMO. Específicamente, la respuesta está basada en el concepto *deception response* que implica la generación y despliegue de una red trampa (también llamada honeynet) donde el ataque va a ser redirigido para su aislamiento y posterior análisis para evaluar el resultado de la respuesta, y así aprender nuevas maneras de inferencia.

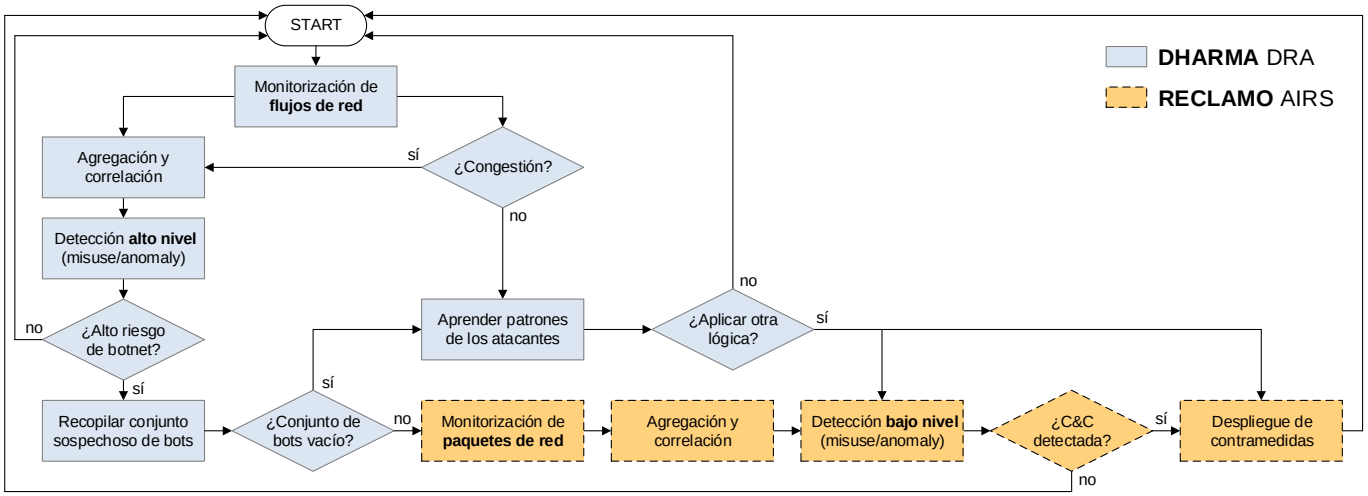

Fig. 4. Diagrama de flujos para la detección de redes botnet a dos niveles de abstracción y el despliegue de contramedidas para su mitigación.

IV. Procesos de detección y mitigación

En esta sección se presenta una descripción más detallada y completa de los dos procesos para la detección y mitigación de redes botnet, así como la ejecución de ataques DDoS, donde se van a ampliar aquellos conceptos que han sido introducidos en la Sección III durante la presentación de los dos proyectos relacionados con este trabajo de investigación.

Como guía para explicar las diferentes fases propuestas, en la Fig. 4 se muestra un diagrama de flujos donde se desgranan los principales pasos que se tienen que ir realizando.

A. Detección mediante flujos del tráfico de red

Como se puede ver en la Fig. 4, todos los pasos (bloques de actividad) marcados en azul están relacionados en cómo hacer la detección a un alto nivel de abstracción mediante el análisis de flujos del tráfico de la red. En concreto, estos pasos se han marcado como objetivo en el proyecto DHARMA, el cual ha sido analizado en la Sección III-A.

Durante esta fase de detección, la *Monitorización de flujos de red* se va a ejecutar constantemente con el afán de:

1) Detectar redes botnet. Esta detección depende según el tipo de red botnet, y según la "unión" que haya entre los bots y, habitualmente, el servidor C&C del botmaster. Más allá de su tipología, la gran mayoría de redes botnet se caracterizan por tener patrones de comportamiento relativamente similares. En el fondo, casi siempre se va a producir una comunicación de manera periódica entre los bots y el servidor C&C a modo de *keep-alive*. Además, en muchos casos incluso es el propio bot el que le pregunta al servidor C&C si hay alguna acción o comando que tendría que ejecutar en nombre del botmaster. Por ejemplo, el envío por inundación (o *flooding*) de una gran cantidad de solicitudes HTTP, paquetes ICMP o TCP/SYN, a una determinada dirección IP por ser la víctima objetivo.

Por tanto, la monitorización de los flujos del tráfico de red realizada por los *Flow probes* (ver Fig. 2) tiene como objetivo extraer una serie de métricas de bajo nivel con las que agrupar el tráfico por comportamientos similares entre pares de hosts. A continuación se enumeran algunos ejemplos de potenciales métricas de bajo nivel que se podrían utilizar para el análisis de flujos del tráfico de red. Algunas de esas métricas han sido identificadas y explicadas en detalle en [19], [24].

- Average number of packets per flow (ANPPF)
- Percentage of correlative flow (PCF)
- One direction generating speed (ODGS)
- Ports generating speed (PGS)
- Average of bytes per flow (ABf)
- Average of duration per flow (ADf)
- Percentage of pair-flows (PPf)
- Growth of single-flows (GSf)
- Growth of different ports (GDP)

Por ejemplo, tanto con ANPPF como con ABf se puede ver si entre pares de determinados hosts hay una cierta similitud entre su número de paquetes o bytes, respectivamente. Ambas métricas podrían dar una evidencia de si hay hosts en la red que está siendo protegida de comportamientos similares, y así poder marcar esos hosts como posibles bots de una red botnet para un posterior análisis más en profundidad.

2) Detectar ataques DDoS. De manera bastante similar al punto anterior, en este caso también se va a realizar un análisis constante de los flujos del tráfico de red utilizando un conjunto de métricas de bajo nivel parecidas a las listadas más arriba. El objetivo es comprobar si hay *congestión* en la red, indicando la posibilidad de un ataque DDoS. Este tipo de ataque puede ser a causa de una red botnet que, siendo así, es posible que no se haya detectado de manera proactiva a tráves del punto anterior. Este hecho se puede deber, principalmente, a que

- el punto anterior que se acaba de presentar no tiene las competencias apropiadas para hacerlo, ya que se trata de un nuevo tipo de red botnet, por ejemplo; o
- los componentes de la red botnet, sobre todos los bots, se encuentran más allá de las fronteras de la red que se está protegiendo, y por ende no se han podido encontrar patrones similares en el comportamiento que ayudarían en la detección de la nombrada red botnet.

A partir de este primer paso, e independientemente de si se ha detectado una red botnet o un ataque DDoS, se siguen los puntos introducidos en la Sección III-A donde se procede a la evaluación del impacto y el riesgo que dicha detección tiene sobre los activos de la organización. En caso de haberlo (ver Fig. 4), se recopila la lista de posibles bots para efectuar otro proceso de detección que confirme que realmente lo son.

B. Detección mediante la inspección profunda de paquetes

Este segundo proceso de detección se lleva a cabo después de que un primero, explicado en la subsección anterior, haya encontrado evidencias de albergar una posible red botnet en la red o que se está siendo víctima de un ataque DDoS a causa de una botnet, la cual o no se ha podido detectar anteriormente o el ataque proviene desde bots externos. En cualquiera de esos dos casos, lo que sí que hay que hacer es una comprobación minuciosa de que esa posible amenaza ha ocurrido realmente. Para ello, en este trabajo de investigación se propone ejecutar un nuevo proceso de detección a más bajo nivel que el anterior mediante una inspección profunda de paquetes (DPI).

Para llevar a cabo esta nueva fase de detección, y como ya se ha comentado en la Sección III-B, se ha hecho uso de la solución obtenida como resultado en el proyecto RECLAMO. En este caso en particular, para la detección de redes botnet o ataques DDoS, se ha adaptado la solución de RECLAMO para realizar la inspección profunda de paquetes con el objetivo de confirmar si la amenaza ha ocurrido o no en la realidad. En la Fig. 4, todos estos pasos se han marcado en amarillo.

Para la puesta en marcha de esta segunda fase de detección, un número de IDSs son instalados de manera estratégica en la red de monitorización. Estos IDSs son configurados para sólo analizar los paquetes de red de aquellos posibles bots que han sido identificados como tales durante la detección a alto nivel de la subsección anterior. En gran medida, esa configuración dependerá del tipo concreto de botnet que, presuntamente, se ha detectado con anterioridad. Por ejemplo, una posible regla en Snort podría ser como la siguiente, a fin de poder detectar solicitudes HTTP de comandos que el bot tendría que ejecutar. Esa solicitud estaría destinada al servidor C&C.

```
alert http $HOME_NET any -> $EXTERNAL_NET any \
(msg:“Solicitud de comando desde un bot”; nocase; \
content:“GET /*/commands.php?uid=”;)
```

Una vez confirmada la existencia de una red botnet, ya haya sido porque se ha identificado o no la ejecución de un ataque DDoS, el siguiente y último paso es la puesta en marcha de una serie de contramedidas para su mitigación. Este paso es el *Despliegue de contramedidas* de la Fig. 4.

Comentar que, aunque aquí se haya propuesto realizar este segundo proceso de detección después de que el primero haya encontrado indicios de una posible red botnet, una inspección profunda de paquetes siempre se puede ejecutar cuando se está produciendo un ataque DDoS, aun cuando el primer proceso de detección mediante flujos del tráfico de red no haya podido encontrar evidencias de una posible red botnet.

C. Fase de mitigación mediante el despliegue de una honeynet

Entre las posibles alternativas de respuesta frente a ataques, en este trabajo se propone la creación y despliegue de una red trampa personalizada (o *honeynet*) para el ataque detectado, y luego confirmado, por los puntos anteriores. Estas redes como respuesta han sido adoptadas por ser uno de los resultados del proyecto RECLAMO, presentado en la Sección III-B.

Los componentes a clonar para la creación y despliegue de redes trampa tienen una alta relación según el tipo de amenaza que se haya detectado. Las distintas alternativas para llevar a cabo la respuesta deseada se detallan a continuación.

1) Bots detectados en la misma red de monitorización. En este caso, los componentes a clonar son esos bots detectados, donde las comunicaciones entre cada bot con el servidor C&C tienen que redirigirse al nuevo host clonado actuando como si fuera el bot real. Para que el nuevo bot no revele modificación alguna frente al servidor C&C (i.e., no llegue a ser detectado como un host falso), esta réplica de host tiene que “simular” el mismo comportamiento que mantenía el bot original.

2) Servidor C&C en la propia red de monitorización. La clonación se haría de ese servidor C&C, redireccionando todo su tráfico malicioso hacia su réplica y bloqueando el tráfico de salida del original hacia sus bots detectados, los cuales pueden o no formar parte de la red interna de monitorización. En caso de serlo, esos bots también se clonarían como se ha explicado en el punto anterior, y la red botnet parecería que sigue siendo la misma desde el punto de vista del botmaster.

3) Red de bots externos a la red de monitorización. Esta detección se habrá realizado a consecuencia de la ejecución de un ataque DDoS. En este caso, ni el servidor C&C ni los bots se pueden clonar porque ninguno de ellos pertenece a la red que se está protegiendo. El único componente que se clona es el host víctima del ataque DDoS, y redireccionando el tráfico entrante hacia el nuevo host. De esta forma se llegaría a liberar de tráfico malicioso a la víctima, haciendo que el ataque DDoS no consiga el éxito esperado.

Como punto final aclarar que la clonación no erradicaría el problema, sino que dicho problema se llega a aislar en una red virtualizada de cara a que los bots gestionados remotamente por el botmaster no sean capaces de ejecutar su ataque final; es decir, la ejecución final del ataque DDoS. La erradicación definitiva de la red botnet solamente se puede llevar a cabo si se puede eliminar el software malicioso que los bots tienen instalados. Si están instalados en los dispositivos móviles de los usuarios finales, esa erradicación solo se podría llevar a cabo notificando (por ejemplo, por parte del ISP) a dichos usuarios para que desinstalen el software malicioso. Por tanto, la erradicación completa (es decir, la eliminación de software malicioso) no depende del sistema que se está protegiendo, ya que esa eliminación de malware solamente se podrá llevar a cabo en los hosts donde el sistema tiene acceso.

V. Conclusiones y vías futuras

En este presente trabajo de investigación se ha presentado cómo poder realizar la detección y mitigación de redes botnet mediante el análisis del tráfico de red a dos niveles distintos de abstracción: monitorizando, en primer lugar, sólo los flujos del tráfico de red; y, en segundo lugar, un proceso de detección a más bajo nivel para confirmar que esa red botnet existe en la realidad. Ambas fases han sido marcadas como objetivos para dos proyectos nacionales de investigación, donde la primera fase de monitorización a alto nivel se ha establecido como uno de los objetivos principales dentro de DHARMA, uno de esos dos proyectos que actualmente está en desarrollo.

Como posibles vías futuras de investigación, se propone el estudio y adaptación de nuevos mecanismos basados en SDN (Software-Defined Network) con los que proporcionar mayor dinamismo en la detección de nuevas redes botnet como, por

ejemplo, las móviles (*mobile botnets*). Los distintos procesos de monitorización tendrán que ajustar sus parámetros para la configuración de forma totalmente dinámica conforme los bots vayan desplazándose por la red. De esta manera, también será necesario la adopción de otros mecanismos como los basados en NFV (Network Function Virtualization) para que todos los sistemas que realicen dicha monitorización puedan reubicarse inmediatamente de un lugar a otro con técnicas más avanzadas de virtualización.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el MINECO (proyecto DHARMA, *Análisis y Gestión Dinámica de Riesgos con Amenazas Heterogéneas*, código TIN2014-59023-C2-1-R) y por la Comisión Europea (FEDER/ERDF).

REFERENCIAS

[1] S.S.C. Silva, R.M.P. Silva, R.C.G. Pinto, R.M. Salles: "Botnets: A survey", en *Computer Networks*, vol. 57, no. 2, pp. 378-403, 2013.

[2] R.A. Rodríguez-Gómez, G. Maciá-Fernández, P. García-Teodoro: "Survey and taxonomy of botnet research through life-cycle", en *ACM Computing Surveys*, vol. 45, no. 4, pp. 45:1-45:33, 2013.

[3] V.L. Thing, M. Sloman, N. Dulay: "A survey of bots used for distributed denial of service attacks", en *IFIP TC-11 22nd International Information Security Conference*, pp. 229-240, 2007.

[4] J. Demarest: "Taking down botnets: Public and private efforts to disrupt and dismantle cybercriminal networks", 2014. http://www.fbi.gov/news/testimony/taking-down-botnets

[5] M. Feily, A. Shahrestani, S. Ramadass: "A survey of botnet and botnet detection", en *3rd International Conference on Emerging Security Information, Systems and Technologies*, pp. 268-273, 2009.

[6] V. Richariya, U.P. Singh, R. Mishra: "Distributed approach of intrusion detection system: Survey", en *International Journal of Advanced Computer Research*, vol. 2, no. 6, pp. 358-363, 2012.

[7] U.S. Department of Energy: "100 Gbps science network". http://science.energy.gov/ascr/news-and-resources/100gbpsnetwork

[8] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, C. Kruegel: "DISCLOSURE: Detecting botnet command and control servers through large-scale NetFlow analysis", en *28th Annual Computer Security Applications Conference*, pp. 129-138, 2012.

[9] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, D. Garant: "Botnet detection based on traffic behavior analysis and flow intervals", en *Computers & Security*, vol. 39, pp. 2-16, 2013.

[10] R. Hofstede, P. Celeda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, A. Pras: "Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX", en *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037-2064, 2014.

[11] P. Holgado, M. Gil Pérez, G. Martínez Pérez, V.A. Villagrá: "Evolving from a static toward a proactive and dynamic risk-based defense strategy", en *I Jornadas Nacionales de Investigación en Ciberseguridad*, pp. 129-136, 2015.

[12] Proyecto DHARMA: "Dynamic heterogeneous threats risk management and assessment". http://dharma.inf.um.es

[13] M. Gil Pérez, V. Mateos Lanchas, D. Fernández Cambronero, G. Martínez Pérez, V.A. Villagrá: "RECLAMO: Virtual and collaborative honeynets based on trust management and autonomous systems applied to intrusion management", en *7th International Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 219-227, 2013.

[14] Proyecto RECLAMO: "Virtual and collaborative honeynets based on trust management and autonomous systems applied to intrusion management". http://reclamo.inf.um.es

[15] G. Vasiliadis, S. Antonatos, M. Polychronakis, E.P. Markatos, S. Ioannidis: "Gnort: High performance network intrusion detection using graphics processors", en *11th International Symposium on Recent Advances in Intrusion Detection*, pp. 116-134, 2008.

[16] W. Jiang, Y.-H.E. Yang, V.K. Prasanna: "Scalable multi-pipeline architecture for high performance multi-pattern string matching", en *2010 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1-12, 2010.

[17] J. Yu, B. Yang, R. Sun, Z. Chen: "FPGA-based parallel pattern matching algorithm for network intrusion detection system", en *2009 International Conference on Multimedia Information Networking and Security*, pp. 458-461, 2009.

[18] Y.-M. Hsiao, M.-J. Chen, Y.-S. Chu, C.-H. Huang: "High-throughput intrusion detection system with parallel pattern matching", en *IEICE Electronics Express*, vol. 9, no. 18, pp. 1467-1472, 2012.

[19] R. Braga, E. Mota, A. Passito: "Lightweight DDoS flooding attack detection using NOX/OpenFlow", en *2010 IEEE 35th Conference on Local Computer Networks*, pp. 408-415, 2010.

[20] D. van der Steeg, R. Hofstede, A. Sperotto, A. Pras: "Real-time DDoS attack detection for Cisco IOS using NetFlow", en *2015 IFIP/IEEE International Symposium on Integrated Network Management*, pp. 972-977, 2015.

[21] M. Mahmoud, M. Nir, A. Matrawy: "A survey on botnet architectures, detection and defences", en *International Journal of Network Security*, vol. 17, no. 3, pp. 272-289, 2015.

[22] G. Gu, P. Porras, V. Yegneswaran, M. Fong, W. Lee: "BotHunter: Detecting malware infection through IDS-driven dialog correlation", en *16th USENIX Security Symposium on USENIX Security Symposium*, pp. 12:1-12:16, 2007.

[23] G. Gu, R. Perdisci, J. Zhang, W. Lee: "BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection", en *17th Conference on Security Symposium*, pp. 139-154, 2008.

[24] Y. Feng, R. Guo, D. Wang, B. Zhang: "Research on the active DDoS filtering algorithm based on IP flow", en *5th International Conference on Natural Computation*, vol. 4, pp. 628-632, 2009.