UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E INFORMÁTICOS

Tema:

"Análisis y detección de botnets mediante minería de datos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato"

Proyecto de Trabajo de Graduación. Modalidad: Proyecto de Investigación. Trabajo Estructurado de Manera Independiente, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos.

SUBLINEA DE INVESTIGACIÓN: Seguridad Informática.

AUTOR: Robinson Adrián Zurita Amores.

TUTOR: PhD. Félix Oscar Fernández Peña.

Ambato – Ecuador

Noviembre - 2017

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Investigación sobre el Tema: "Análisis y detección de botnets mediante minería de datos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.", del señor Robinson Adrián Zurita Amores, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, noviembre de 2017

EL TUTOR

PhD. Félix Oscar Fernández Peña

AUTORÍA

El presente proyecto de investigación titulado: "Análisis y detección de botnets mediante minería de datos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato." es absolutamente original, auténtico y personal, en tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, noviembre de 2017

Robinson Adrián Zurita Amores

CC: 1803622529

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato, noviembre de 2017

Robinson Adrián Zurita Amores

CC: 1803622529

APROBACIÓN COMISIÓN CALIFICADORA

La comisión calificadora del presente proyecto conformada por los señores docentes Ing. David Guevara Mg. e Ing. Dennis Chicaiza Mg., revisó y aprobó el Informe Final del proyecto de graduación titulado "Análisis y detección de botnets mediante minería de datos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.", presentado por el señor Robinson Adrián Zurita Amores de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato, noviembre de 2017

Ing. Elsa Pilar Urrutia
PRESIDENTE DEL TRIBUNAL

Ing. David Guevara
DOCENTE CALIFICADOR

Ing. Dennis Chicaiza
DOCENTE CALIFICADOR

DEDICATORIA

Dedico este proyecto a mi familia, la ciencia informática y la música pues son estas las cosas que me han inspirado y le han dado el curso a mi vida.

Adrián Zurita Amores

AGRADECIMIENTO

Agradezco a mi familia y todas las personas que han aportado con su conocimiento, paciencia y experiencia para que pueda lograr mis objetivos. También quiero extender mi agradecimiento especial a la Facultad de ingeniería en Sistemas Electrónica e Industrial y a mi director de tesis PhD. Félix Fernández por toda la colaboración y apoyo para realizar este proyecto.

Adrián Zurita Amores.

RESUMEN

El notable crecimiento de dispositivos con acceso a Internet y las grandes cantidades de información y aplicaciones que existe en la red, han hecho que en la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato se crean vulnerabilidades, incluyendo la ejecución de malwares y el funcionamiento de botnets desde la infraestructura de red de la institución.

La detección distribuida y en tiempo real de botnets puede inducir grandes costos de infraestructura y equipos de comunicación en la red. Cada dispositivo conectado a la red realiza tareas y peticiones a diferentes servidores de manera independiente, por lo que es difícil detectar cada botnet de manera exhaustiva. En este proyecto se propone un método de detección de botnets mediante minería de datos, que utiliza los *logs* de los *queries* del servidor dns como fuente de datos para identificarlos. Para realizar dicha minería de datos se utilizó un software llamado Splunk el cual permite analizar los *logs* y determinar las conexiones a servidores C&C (*Command and control*) los cuales son encargados de controlar los bots infectados en los dispositivos. El procedimiento resultante fue complementado con técnicas de *Machine Learning*, específicamente con el uso del algoritmo de predicción llamado "*random forest".* De esta forma se obtienen resultados con un margen de error de +/- 5.44 en 7 min 45 segundos aproximadamente para 18,748,713 de eventos analizados desde el 01 al 06 de diciembre del 2017, lo cual corrobora la validez de la propuesta.

Palabras Clave: Internet, Botnets, minería de datos, servidores C&C, Machine learning.

SUMMARY

The notable growth of devices with Internet access and the large amount of information and software in the network have increased the vulnerabilities in the “Facultad de Ingeniería en Sistemas Electrónica e Industrial” of the “Universidad Técnica de Ambato”. Those vulnerabilities include malware and botnets infections.

Remote and real-time detection of botnets can lead to big infrastructure costs and communication equipment in the network. Each device connected to the network performs tasks and requests to different servers independently, making it difficult to detect every botnet in the network. This project proposes a method of detecting botnets with data mining, which uses the DNS server query records as a data source to identify botnets. In order to perform this data mining, Splunk was used to analyze the query logs and to determine the connections to the C&C (Command and Control) servers, which are responsible for controlling infected bots on devices. The resulting procedure was complemented with the machine learning algorithm of *Random Forest*. This way, results were obtained with an error margin of +/- 5.44 in approximately 7 min 45 seconds for 18,748,713 of events analyzed since 01 to 06 of December 2016, as validation of the proposal.

Key words: Internet, botnets, data mining, C& C servers, machine learning.

GLOSARIO DE TÉRMINOS

Botnet: Es un término que hace referencia a un conjunto o red de robots informáticos o bots, que se ejecutan de manera autónoma y automática. El artífice de la botnet puede controlar todos los ordenadores/servidores infectados de forma remota.

Servidores C&C: Servidores de control y mando (*command and control*) y otros elementos que son usados para controlar botnets.

Minería de Datos: Es un campo de la estadística y las ciencias de la computación referido al proceso que intenta descubrir patrones en grandes volúmenes de conjuntos de datos.

Machine Learning: Es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender.

DNS: El sistema de nombres de dominio (DNS, por sus siglas en inglés, Domain Name System) es un sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP como Internet o una red privada.

Random Forest: Es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos.

Logs: Se usa el término log, historial de log o registro a la grabación secuencial en un archivo o en una base de datos de todos los acontecimientos (eventos o acciones) que afectan a un proceso particular (aplicación, actividad de una red informática, etc.).

Query: Es un término informático que se utiliza para hacer referencia a una interacción con

una base de datos o un servidor web. Es la parte de una URL que contiene los datos que deben pasar a aplicaciones web.

Malware: código maligno, software malicioso, software dañino o software malintencionado; tiene como objetivo infiltrarse o dañar una computadora.

Splunk: Es un software para buscar, monitorizar y analizar datos generados por máquinas (Big Data) de aplicaciones, sistemas e infraestructura IT a través de una interfaz web.

Margen de Error: se refiere a la cantidad de error de muestreo aleatorio resultado de la elaboración de una encuesta.

ÍNDICE

APROBACIÓN DEL TUTOR II
AUTORÍA III
DERECHOS DE AUTOR IV
APROBACIÓN COMISIÓN CALIFICADORA V
DEDICATORIA VI
AGRADECIMIENTO VII
RESUMEN VIII
SUMMARY IX
GLOSARIO DE TÉRMINOS X
INTRODUCCIÓN XVI
CAPÍTULO I 1
1.1 Tema de investigación 1
1.2 Planteamiento del problema 1
1.2.1 Contextualización 1
1.3 Delimitación 2
1.4 Justificación 3
1.5 Objetivos 5
1.5.1 Objetivo General 5
1.5.2 Objetivos Específicos 5
CAPÍTULO II 6
2.1 Antecedentes Investigativos 6
2.2 Fundamentación Teórica 9
2.2.1 Nombres de Dominio 9
2. 2.2 Botnet 11
2.2.3 Servidores C&C. 13
2.2.4 Splunk. 15
2.2.5 Machine Learning: 18
2.2.6 Algoritmo Random Forest 19

CAPÍTULO III 22
METODOLOGÍA 22
3.1. Modalidad de la investigación 22
3.3 Recolección de la información 23
3.4 Procesamiento de análisis de datos 23
3.5 Desarrollo del proyecto 23
CAPÍTULO IV 25
4.1 Desarrollo para el análisis de detección de Botnets. 25
4.1.1. Análisis de la red de la FISEI. 25
4.1.2 Equipo para el Desarrollo de la propuesta. 30
4.1.3 Instalación de Splunk 30
4.1.3 Creación de los *datasources* en base a los archivos recolectados en la FISEI. 35
4.1.4 Minería de Datos. 40
4.1.5 Ejemplos de análisis de los dns-queries. 41
4.1.6 Detección de botnets. 43
4.1.7 Diagrama de flujo de los procesos realizados para la detección de botnets: 45
4.1.8 Query para detectar botnets: 46
4.1.9 Margen de error 46
4.1.10 Gráfico actual vs predicted: 47
4.2. Interpretación de Resultados 48
4.2.1 Resultados De la Predicción: 49
4.2.2 Gráfico Actual VS Predicciones: 51
4.3.3 Margen de Error: 52
4.2.4 Comprobación de botnets: 52
CAPÍTULO V 60
5.1 Conclusiones: 60
5.2 Recomendaciones: 61
Bibliografía o Referencias. 63
ANEXOS Y APENDICES 66
Manual de instalación y uso de detección de botnets. 66

INDICE DE FIGURAS

Fig. 1: Jerarquía Dns 10
Fig. 2: Estructura de la red de la FISEI 27
Fig. 3: Esquema General de Comunicaciones 28
Fig. 4: Informe de Peticiones DNS 29
Fig. 5: Página de Descarga de Splunk 31
Fig. 6: Instalación de splunk 32
Fig. 7: Plugins Adicionales 33
Fig. 8: Instalación de Splunk Machine Learning 33
Fig. 9: Instalación de DNS Analytics for Splunk 34
Fig. 10: Instalacion de URL Toolbox 35
Fig. 11: Creacion de Datasources en Splunk 36
Fig. 12: Data Inputs 36
Fig. 13: Directorio de Archivos de origen de datos 37
Fig. 14: Establecer el Tipo de Datos en Splunk 38
Fig. 15: Propiedades de DataSource 38
Fig. 16: Confirmacion de DataSource 39
Fig. 17: Busqueda de un DataSource 40
Fig. 18: Top 20 de queries más solicitados 41
Fig. 19: Grafico por horas 42
Fig. 20: Gráfico por “record-type” 42
Fig. 21: Diagrama de flujo del proceso de detección de botnets 45
Fig. 22: Grafico Actual vs. Predicted 48
Fig. 23: Gráfico actual Vs. Predicciones 51
Fig. 24: Margen de error 52
Fig. 25: Verificacion de la botnet sso.anbtr.com 53
Fig. 26: Verificacion de la botnet global.ymtracking.com 54
Fig. 27: Verificación de la botnet sur.ly 55
Fig. 28: Verificación de la botnet ant.trenz.pl 56
Fig. 29: Verificación de la botnet bdb.com.my 57
Ilustración 30: Resultado de dominio malicioso 58

ÍNDICE DE TABLAS.

Tabla 1: Resultados de la Predicción 49
Tabla 2: Botnets Positivos 50
Tabla 3: Falsos Positivos 50

INTRODUCCIÓN

El notable crecimiento de dispositivos conectados a Internet en la población de Ambato según el INEC [1], y en la Facultad de Ingeniería en Sistemas Electrónica e Industrial, es notable. La cantidad de información gestionada y la continua evolución de los medios de comunicación han hecho que el acceso a la red sea un requerimiento imprescindible para la formación académica de los estudiantes y la administración de la FISEI [21]. Consecuentemente, al manejar un tráfico de red considerable y varios dispositivos conectados, se producen vulnerabilidades de todo tipo, incluyendo puertas traseras [17], las que son aprovechadas por hackers y cibercriminales para enviar malware a los distintos dispositivos conectados.

En este contexto aparecen los denominados botnets. Algunos de los más conocidos incluyen malwares como Stuxnet [18] y Zeus [19]; estos sistemas utilizan la contaminación de botnets para generar una capacidad de ataque "ilimitada" que permita obtener información sensible y para realizar ataques DDOS. Según el informe de amenaza de McAfee [20], la intensidad de los ataques de botnets aumenta con el crecimiento de Internet. Por lo tanto, el problema de detección de botnets requiere de atención en toda institución con una red de tráfico considerable para garantizar la seguridad digital de los usuarios de la red.

La mayoría de las investigaciones existentes sobre la infección de botnets de defensa en las redes se centran en la construcción de infraestructuras con equipos de comunicación costosos, el desarrollo y adquisición de frameworks de código privativo y soluciones antivirus que dependen de una distribución completa para su ejecución [3, 4, 5]. Aunque estos sistemas distribuidos de detección y defensa son más fáciles de desplegar, sólo son eficaces para detectar algunos nodos comprometidos de toda la red o cierta clase de botnets específicos, pero no amenazas globales debido a la cantidad de registros que una red produce a lo largo del tiempo.

El estudio principal de este proyecto se centra en el análisis de botnets mediante la minería de datos, obteniendo como origen de datos los logs de *queries* del servidor DNS que captura en su totalidad todas las peticiones de los dispositivos conectados, por lo que se puede obtener un análisis y una visión global de las peticiones al servidor DNS.

Splunk Enterprise facilita la recopilación, el análisis y el uso de los *big data* generados por una infraestructura tecnológica [12], en este caso los archivos de query-logs del servidor DNS, con las facilidades y plugins de este software se puede proporcionar una manera prometedora de lidiar con problemas de seguridad como la detección de botnets.

La tarea de detección se completa con el desarrollo de un proceso que analiza los tiempos de conexión, números de conexiones muertas y consultas a bases de datos de botnets descubiertos, con la cooperación del machine learning y el algoritmo "*random forest*" se, puede realizar un análisis y una detección de botnets eficiente con un margen de error de +- 5.44 aproximadamente, lo que demuestra que es un método válido, los resultados experimentales fueron validados con botnets ya descubiertas.

El método desarrollado ayuda a generar ficheros que pueden ser incluidos en las listas negras de los servidores dns o proxies para ser bloqueados definitivamente, lo cual ayudará a que los dispositivos conectados a la red no respondan a las órdenes de los servidores C&C, las limitaciones de este método son que no se puede detectar un botnet en tiempo real sino cuando el ataque ya fue realizado así que funciona como una medida correctiva.

CAPÍTULO I
EL PROBLEMA

1.1 Tema de investigación

Análisis y detección de botnets mediante minería de datos en la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

1.2 Planteamiento del problema

1.2.1 Contextualización

En Ecuador, con el pasar de los años, ha existido un crecimiento notorio en el acceso a Internet. Para el año 2015 se ha presentado un crecimiento del 41.0% [1], en el área urbana y un 13.7% [1], en el área rural con acceso a Internet, por lo que la generación y manipulación de la información cada vez es más compleja lo que permite a los creadores de malware tener un mercado más amplio y rentable en el país [1]. De la misma manera, el acceso a Internet en las universidades del país es un requisito necesario para la educación de los estudiantes, lo cual conlleva a que las vulnerabilidades y la protección de datos de estudiantes y personal con acceso a la red [1], sean cada vez mayores, comprometiendo así la seguridad de sus datos personales.

En la Universidad Técnica de Ambato, al igual que en el resto de Ecuador, los sistemas y la información han ido creciendo a un paso acelerado, lo que provoca una gran demanda en cuanto a la capacidad de red, conectividad y procesamiento de información [21]. Al no contar con un método adecuado para garantizar la seguridad en los procesos, en muchas ocasiones se crean vulnerabilidades y puertas traseras por donde los creadores de malware aprovechan para enviar botnets y programas con código malicioso. Consecuentemente, estos equipos se utilizan para realizar ataque de denegación de servicios, *keyloggers*, etc.

Las botnets se han convertido en un problema real y de rápida evolución; constituyen una de las amenazas de seguridad más preocupantes debido a que causan pérdidas financieras, denegación a los servicios que prestan y graves daños a las organizaciones de todo el mundo a las cuales se las ataca [13].

La evolución de las botnets y sus métodos de infección hacen que cada vez sea más complicada y difícil de implementar su detección. Del mismo modo, es de señalar que el código de los *frameworks* actuales que sirven para captura de datos no es libre. Consecuentemente, la mano de obra, el tiempo y el factor económico también tienen un mayor impacto.

1.3 Delimitación

Área Académica: Hardware y Redes.

Línea de Investigación: Sistema Administrador de Recursos.

Sub línea de Investigación: Seguridad Informática.

Delimitación Espacial: Universidad Técnica de Ambato, Av. Los Chasquis (Huachi Chico).

Delimitación Temporal: La duración del proyecto es de 6 meses a partir de la fecha de aprobación del perfil por parte del Honorable Consejo Directivo de la Facultad.

1.4 Justificación

El constante crecimiento de la información y los procesos que maneja la Facultad de Ingeniería en Sistemas Electrónica e Industrial hace que las vulnerabilidades y los ataques de criminales informáticos vaya cada vez en mayor aumento. Algunas de las razones para que ocurra esto es que al usar la tecnología actual, mediante la red de la UTA se gestionan datos bancarios, identidades de usuarios, credenciales y hasta el propio ancho de banda. Todos estos elementos poseen un valor monetario significativo. Con un usuario y contraseña se pueden administrar transferencias bancarias, tarjetas de crédito, correos electrónicos con los cuales se pueden hacer compras, suplantación de identidades, el ancho de banda se puede ocupar para generar correos spam y generar tráfico a servidores específicos con el objetivo de saturar la red dejando sin servicios virtuales a los usuarios.

Tomar el control de una gran cantidad de máquinas en Internet se ha convertido en una tarea relativamente trivial, debido al elevado número de usuarios inexpertos que están conectados a la red y que no tienen en consideración los más mínimos requisitos de seguridad a cumplir.

Por otra parte, se han desarrollado, y son fácilmente localizables, muchas herramientas automáticas que ejecutan ataques de este tipo sin necesidad de conocimiento alguno sobre el funcionamiento de las mismas. Esto hace que cualquier usuario sin mucha experiencia, pueda realizar este tipo de ataques. Además, incluso si la máquina objetivo identificará a todas las máquinas que la están atacando, sería difícil decidir qué acción tomar en contra de un número tan elevado de máquinas. Estos ataques seguirán produciéndose ya que en el mercado negro tienen un valor significativo. Un ejemplo claro es que un ataque DDOS (denegación de servicios) por una semana a un sitio web cuesta alrededor de $150 usd. según *Trend Micro Incorporated* [2], lo cual lo hace un negocio rentable en la actualidad.

Para prevenir dichos ataques a la red de la Universidad Técnica de Ambato existen herramientas de análisis de tráfico de la red y de generación de traza de las operaciones del servidor DNS a partir de una análisis previo a dicha red, a partir de lo cual se pretende evaluar si es posible determinar, mediante un análisis, las conexiones recurrentes y potencialmente de ataque, para incrementar la seguridad del uso de la red por estudiantes y personal de la institución, en general.

Una manera de detectar botnets en una red que se ha estado estudiando en los últimos años es el uso de técnicas de minería de datos, para descubrir patrones en las características de un sistema que describen el comportamiento de los programas y los usuarios [11]. Con ello, se podría crear un clasificador que reconozca anomalías e intrusiones. Este método utiliza como información variables o parámetros medidos en el propio sistema y no en los paquetes de información que viajan por la red o que se encuentran en archivos de log. Sin embargo, ni la documentación de los métodos creados, ni las herramientas para llevar a cabo estos estudios, están disponibles de forma gratuita. El presente estudio pretendió determinar un procedimiento de detección de botnets que funcione para la FISEI haciendo uso de la información que proveen los logs del servidor DNS de la UTA.

1.5 Objetivos

1.5.1 Objetivo General

Analizar y detectar botnets mediante minería de datos en la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato.

1.5.2 Objetivos Específicos

- Analizar el tráfico de red de la Universidad Técnica de Ambato en donde se presentan cargas excesivas de datos y saturación en la red.

- Detectar botnets utilizando técnicas de minería de datos adecuadas.

- Obtener un reporte de alertas de máquinas infectadas en la subred.

CAPÍTULO II
MARCO TEÓRICO

2.1 Antecedentes Investigativos

Se ha realizado una búsqueda de proyectos en la base de datos de "scopus", en la cual se han encontrado 246 documentos de trabajos referentes a ataques de botnets. En ellos se han desarrollado investigaciones que tratan sobre los métodos de infección mediante el protocolo P2P [21], mecanismos de defensa e infección mediante contenidos multimedia [22], transmisión de código malicioso por redes LTE [23], tráfico de red y botnets en la nube [24]. Estos trabajos tienen como objetivo analizar y determinar la presencia de botnets y servidores C&C, además de implementar técnicas para su prevención y control.

El trabajo realizado por Ruidong Chen et al. propone un sistema eficiente de detección de botnets basado en el análisis del tráfico de la red, el cual puede manejar grandes anchos de banda [3]. Los autores antes mencionados desarrollaron un framework modular el cual utiliza PF_RING que es un procesador de paquetes de alta velocidad, resolviendo así la alta demanda de paquetes capturados en la herramienta Libcap, en la cual se desarrolla parte del proyecto. RF-RING tiene baja latencia y baja sobrecarga para extraer los campos de tráfico requeridos [3].

Las características de los módulos del framework combinan las ventajas de la detección existentes, métodos basados en comportamientos estadísticos de flujo y similitud de tráfico. De esta forma se logra seleccionar "conversaciones prometedoras" que son las que el framework desarrollado detecta como amenazas haciendo uso del Algoritmo Random-forest [3]. Para detectar botnets, el trabajo en mención utiliza técnicas de aprendizaje de máquina utilizando la herramienta de minería de datos WEKA [3].

Los experimentos llevados a cabo por estos autores permitieron obtener resultados experimentales que muestran que entre los algoritmos de clasificación evaluados por ellos, la tasa de detección del algoritmo Random-forest es la más alta (hasta un 93,6%), y la tasa de falsos positivos es sólo del 0,3%. El resultado que se obtuvo con este algoritmo fue diez veces superior al de detección basada en las características del flujo de tráfico [3].

La limitante del resultado obtenido por Ruidong et al. está en que el código no está disponible para su uso y que RF-RING es de licencia privativa. Esta propuesta requiere, además, de recursos de hardware de alto costo que permitan el análisis del flujo de datos en tiempo real.

Por otro lado, Chen et al. han propuesto un esquema ligero para detectar botnets utilizando teoría de grafos [4]. Ello les permite resistir los ataques de botnets por software en subredes específicas. Esta propuesta implementa una Software-Defined Network (SDN) como aplicación inteligente, llamada BotGuard. Esta Aplicación inteligente tiene un sistema de alertas junto con la vista global de la red y la capacidad de controlarse a través de Internet [4].

BotGuard es diferente de las investigaciones existentes en SDNs; puede reducir el retraso de la detección de botnets en la vigilancia temprana de intervalos de tiempo de su ciclo de vida. Teóricamente, se probó que el sistema de detección desarrollado puede detectar botnets en tiempo real. Los resultados experimentales muestran que BotGuard obtiene una alta precisión mayor al 90% de detección de botnets y servidores C&C [4]. La propuesta de

Chen et al. Tiene como limitante el que se requiere contar con una infraestructura que permita registrar el flujo de acceso a Internet y tenerlo disponible para su uso en tiempo real. El costo, desde el punto de vista del hardware, encarece la solución.

En el proyecto realizado por Sharma et al., se propone analizar los dns/ip's sospechosos detectados mediante el sistema BotMAD, desarrollado en esta investigación para detectar botnets basados en tráfico DNS. Este trabajo se compone de dos módulos [5]:

> **Módulo de análisis**: Inspecciona y decodifica los paquetes DNS desde una aplicación de programación para captura de paquetes (PCAP). También determina los dominios / IP's maliciosos por escáneres de reputación de terceros en lo que utiliza el módulo ET del motor SNORT IDS. Este módulo se utiliza, fundamentalmente, para la generación de mensajes emergentes de amenazas y alertas offline.
>
> **Módulo de Inspecciones:** El archivo de dominios se puede utilizar para los dominios de la lista negra y pueden ser utilizados, bloqueando y mejorando la seguridad de la red.

En la investigación se encontró que el conjunto de datos de dominios maliciosos no está actualizado y disponible públicamente en su totalidad. Existen pocos servicios de diferentes conjuntos de datos, pero el problema es que no son uniformes, no son fiables y no son precisos. El objetivo del trabajo por Sharma et al., fue preparar la lista negra de Dominios/IP's a través del procesamiento de dataset PCAP de Honeynet, así como las muestras de malware capturadas [5].

El sistema desarrollado es aplicable para detectar bots en una red basada en tráfico DNS [5]. La implementación de este proyecto y los módulos que ocupa requieren de software y hardware que no dispone la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

Después del análisis de las fuentes consultadas se decide que la solución a proponer debe ser desarrollada utilizando software libre, que no requiera costosos recursos de hardware y que quede bien documentada para su posterior uso.

2.2 Fundamentación Teórica

2.2.1 Nombres de Dominio

Cada computadora en la red pública de Internet tiene una dirección numérica única (similar a la exclusividad de los números de teléfono) que consiste en una cadena de números que a la mayoría de las personas les resulta difícil recordar. Esta cadena se denomina "dirección IP" [6].

Para facilitar la búsqueda de una ubicación determinada en Internet, se creó el Sistema de nombres de dominio o DNS. El DNS traduce la dirección IP en una dirección alfanumérica única llamada nombre de dominio, que es más fácil de recordar [6].

No se trabaja con direcciones IP sino que se utilizan nombres de dominio para el acceso a servidores remotos en Internet. Lo mismo ocurre con los botnets dinámicos, que cambian la ruta de acceso a su C&C pero requieren del servicio DNS para actualizar el ip con el que se estarán comunicando [6].

Al asociar una cadena de letras conocida (el nombre de dominio) con una dirección IP, el DNS hace que sea mucho más fácil para el usuario de Internet recordar sitios web y direcciones de correo electrónico. En el ejemplo anterior, la sección "icann.org" de la dirección es el nombre de dominio. La sección "www." le indica al navegador que está buscando ese nombre de dominio en la interfaz de Internet [6].

Los nombres de dominio también se pueden utilizar para enviar correo electrónico. Ya sea que envíe comunicaciones personales o de negocios, quiere estar seguro de que el mensaje

llegará al destinatario deseado. Si utilizamos una analogía con el sistema telefónico, al marcar un número, suena el teléfono en una ubicación en particular porque hay un plan de numeración central que garantiza que cada número de teléfono sea único [6].

Jerarquía distribuida DNS

El DNS es, simplificando, un sistema de bases de datos distribuido y jerárquico. Esto significa que no existe un único servidor que almacene todos los dominios IP del mundo, sino que tenemos una estructura en la que existe un nodo por cada nivel de jerarquía (TLD, SLD,...), al menos en los tres primeros niveles [7].

La jerarquía DNS está ejemplificada en la Fig. 1:

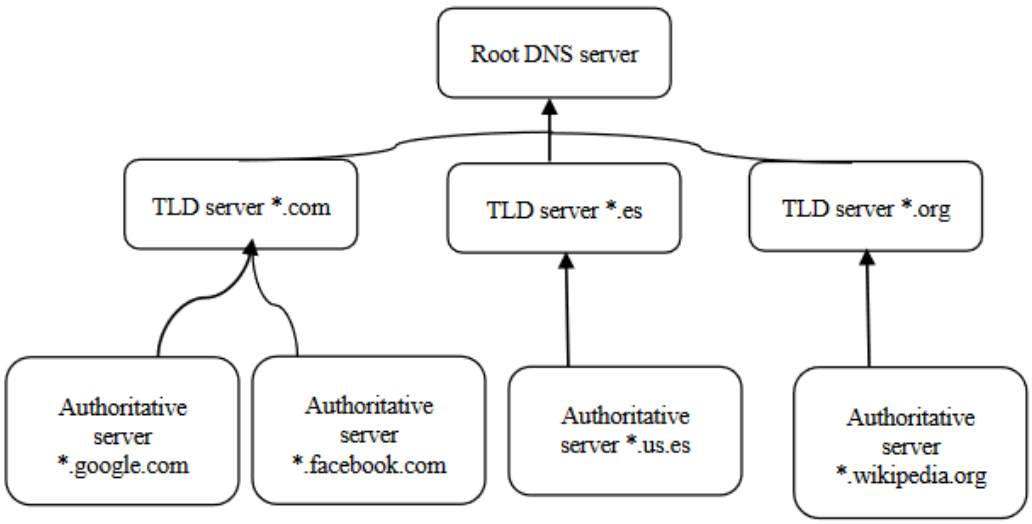


Fig. 1: Jerarquía Dns

En la Fig. 1 se pueden distinguir tres tipos de servidores DNS:

• **Root Servers**

Hoy en día existen 13 root DNS servers en todo el mundo (no servidores físicos; cada operador usa equipamiento de redundancia para asegurar disponibilidad). Estos servidores no guardan ninguna tupla dominio IP, sino que miran la etiqueta más a la derecha del dominio pedido, y apuntan al servidor TLD correspondiente. [7]

•**TLD servers**

Cada servidor TLD se usa para resolver dominios autoritativos bajo su jurisdicción (por ejemplo, el TLD *.com será cuestionado, y deberá proveer información sobre la localización de *.elpais.com o *.google.com) [7].

•**Authoritative servers**

Cada compañía u organización que tenga hosts públicamente accesibles desde Internet ejecuta un Authoritative DNS server. Por ejemplo, cada dominio de *.google.com como puede ser mail.google.com se resuelve mediante un servidor autoritativo bajo el control de Google. Esto garantiza que se puedan actualizar con flexibilidad y rapidez las tuplas dominio IP dentro de una empresa [7].

2. 2.2 Botnet

Es una colección de software robots (o bots) que funcionan de forma autónoma. Tienen la capacidad de recibir comandos desde un bot central controlado por el atacante [8].

También pueden infectar máquinas para implantarles otros bots, aumentando el número de bots en la botnet. Estos bots generan comportamientos variados sobre el tráfico IP, dependiendo de su forma de comunicación y expansión. Algunas botnet funcionan de forma centralizada. Para este caso, el tráfico de la botnet es representado por un conjunto de

máquinas que se comunican con una máquina especial, o viceversa, la máquina comandante se comunica con todas las máquinas en su botnet. De esta forma la cabeza de la botnet (herder) puede organizar un ataque [8].

Los bots traen su propio mecanismo de expansión. Este es similar al de los gusanos. Por ejemplo, utiliza técnicas de escaneo de redes en busca de máquinas y así transmitir el bot [8].

Las botnet son generalmente utilizadas para fines maliciosos. Pueden realizar ataques como los denominados ataques de denegación de servicios distribuidos (distributed denial of service, DDoS). Estos ataques buscan sobrecargar los recursos de la máquina atacada, en algunos casos mediante el envío masivo de paquetes IP [8].

Su característica principal es el uso de los canales command and control (C&C) para conectar los bots a sus botmasters.

El uso de los botmasters de los C&C sufre del llamado problema Single Point of Failure. Esto significa que si la dirección o el dominio del C&C es identificado y neutralizado, el botmaster pierde el control de la botnet completa, al menos temporalmente [9].

Lo que hace a las botnets una amenaza tan grave es que se usan predominantemente para actividades ilegales e ilícitas, como Spamming por email,robos de identidades o ataques de denegación de servicio distribuidos [9].

Muchos sistemas de detección de botnets emplean blacklists de C&C ya conocidas, y así bloquean su tráfico. Esta detección de botnets es estática, ya que esta blacklist se actualiza sólo después de ejecutar procesos (en muchas ocasiones manuales) de detección de dominio [9].

2.2.3 Servidores C&C.

El modelo Centralizado de Comando y Control es el predominantemente usado por las botnets actuales. Mediante este modelo, el botmaster selecciona un solo sistema infectado con ancho de banda suficiente para ser el punto de contacto (el servidor C&C) con todos sus bots. El servidor C&C es usualmente una computadora comprometida, corre servicios de red como IRC, http, y otros. Cuando una computadora es infectada por un bot, ésta se une a la botnet mediante una conexión al servidor C&C. El bot tiene que esperar los comandos del botmaster a través del servidor C&C. Las botnets suelen incluir mecanismos para proteger sus comunicaciones. Por ejemplo, los canales IRC pueden ser protegidos por contraseñas que sólo conocen los bots y sus botmasters para prevenir intrusiones [10].

Algunos autores de botnets han iniciado la construcción de sistemas de comunicación alternativos, los cuales son más resistentes a las fallas en la red. El modelo C&C basado en comunicaciones “peer to peer” (P2P) es mucho más difícil de detectar y destruir. En tanto que los sistemas de comunicación no dependen de unos cuantos servidores selectos, la destrucción de uno o incluso varios de sus bots, no necesariamente conduce a la destrucción del botnet. Sin embargo, los sistemas P2P tienen ciertas restricciones. Primero, solo soportan conversaciones de pequeños grupos de usuarios, normalmente en rangos de 10 a 50 (muy pocos, si se le compara con una red robot “pequeña” de 1000 computadoras, en una botnet con C&C centralizado). Segundo, no aseguran la entrega de mensajes y de latencia en la propagación, ya que este modelo de botnet es más difícil de coordinar que aquellos que usan la centralización C&C [10].

Los bots se comunican entre sí y con sus botmasters, a través de protocolos de red bien definidos. En lugar de crear nuevos protocolos de red utilizan, en la mayoría de los casos, protocolos de comunicación existentes, que son implementados por herramientas públicas de software disponibles [10].

El protocolo IRC predomina en las comunicaciones de botnet. Este protocolo, diseñado para comunicaciones grupales en foros de discusión a los que se llama “canales” (del inglés *channels*), también permite la comunicación uno a uno por medio de mensajes privados. En sí, el protocolo IRC puede ser utilizado por los botmasters para dirigir su ejército botnet usando las habilidades de la comunicación grupal), y controlar selectivamente algunos de los bots (comunicación uno a uno) para actividades específicas. Los firewalls pueden conFig.rse para bloquear el tráfico IRC, pero es mucho más complicado detectar los canales IRC integradas en las comunicaciones en protocolo HTTP [10].

Es por esta razón que el protocolo HTTP es hoy día un método popular de comunicación utilizado por los botnets. El uso del protocolo HTTP dificulta la detección del botnet ya que puede confundirse con el resto del tráfico de Internet. Adicionalmente, la mayoría de las políticas del firewall son implementadas en el servidor de acceso a la red (gateway), donde se puede bloquear el tráfico entrante y saliente utilizando el protocolo IRC. No obstante, el hecho de que los botnets usen el protocolo http, les permite evadir generalmente las políticas de seguridad del firewall [10].

Algunos botnets más avanzados utilizan protocolos IM (Mensajería Instantánea) y peer-to-peer (P2P). Aunque el número de botnets que utilizan protocolos diferentes al IRC y HTTP es relativamente pequeño, estos protocolos podrían alcanzar un uso más generalizado en un futuro cercano, lo cual implicará un reto más complejo para la detección de botnets [10].

Las redes robot se hacen cada día más sofisticados y, de esta forma, más hábiles para evadir la detección. No sólo son los mejores para evadir los motores antivirus y sistemas de detección de intrusos (IDS) basados en firmas; son también más evasivos a los sistemas de detección basados en identificación de anomalías [10].

Los botnets evaden los antivirus y los sistemas IDS basados en firmas, mediante métodos como los empaquetadores ejecutables, rootkits y otras técnicas de evasión de protocolos, los cuales perfeccionan la supervivencia de los botnets y el porcentaje de éxito de nuevos

anfitriones comprometidos. Asimismo, los botnets también han añadido- y continúan haciéndolo- nuevos mecanismos para disimular los rastros de su comunicación. Como se mencionó anteriormente, algunos botnets ya están abandonando el IRC, y se mudan hacia protocolos modificados IRC o HTTP, y más recientemente a protocolos VoIP. Algunas veces, los bots utilizan esquemas de encriptación para prevenir que su contenido sea revelado. Actualmente los botnets utilizan TCP (Transmisión Control Protocol), ICMP (Internet Control Message Protocol), e incluso IPv6 (el último nivel de creación de túneles en protocolo de Internet). La aparición masiva de estos nuevos botnets es sólo cuestión de tiempo [10].

2.2.4 Splunk.

Splunk Enterprise facilita la recopilación, el análisis y el uso del valor oculto de los *big data* generados por una infraestructura tecnológica, sistemas de seguridad y/o aplicaciones empresariales, lo que le proporciona la información necesaria para lograr un rendimiento operacional y resultados empresariales excelentes [12].

Algunas fuentes principales de datos de análisis de splunk, también ofrece la opción de construir también fuentes de datos personalizadas según sea el caso [12]:

- Proxy logs = estos registros son buenos para el análisis C2 de archivos, dominios, descargas de archivos DLL / EXE.

- Antivirus logs = estos registros son buenos para el análisis de malware, vulnerabilidades de hosts, portátiles, servidores, monitorean rutas de archivos sospechosas.

- Sistema operativo Logs = estos registros son buenos para el análisis de las actividades del servidor, tales como usuarios, servicios de desbordamiento, registros de seguridad.

- Firewall logs = registros de tráfico de red de direcciones IP de origen / destino, puertos, protocolos.

- Mail logs = registros de correo entrante / saliente para enlaces maliciosos, destinatarios seleccionados, archivos no autorizados enlazados, pérdida de datos, archivos adjuntos incorrectos.

- Custom apps logs = los registros podrían analizarse para posibles desbordamientos de búfer, inyección de código, análisis de inyección de SQL.

- Intrusion Prevention System logs = captura estos registros para alertar sobre firmas apagando, firmas de COTS, análisis de amenazas de paquetes de red defectuosos.

- Intrusion Detection System logs = registros de captura para alertar sobre firmas apagando, firmas personalizadas, malos paquetes de red.

- Database Logs = captura de estos registros para el acceso autorizado a tablas de datos críticos, registros autorizados, puertos op, cuentas de administrador

- (VPN) Logs = registros de captura para analizar usuarios que entran en la red para conocer la situación, monitorear subredes IP extranjeras, monitoreo del cumplimiento de los navegadores / aplicaciones de los hosts conectados.

- Vulnerability Scan Data = importar datos sobre activos, vulnerabilidades, datos de parches, etc.

- Web Apps logs = registros externos enfrentados para supervisar palabras clave SQL sospechosas, patrones de texto, REGEX para amenazas que llegan a través del navegador

- DNS logs = para correlacionar ip a ir a qué dominio en un nivel de cliente.

El amplio diapasón de fuentes de datos que utiliza SPLUNK da fe de la versatilidad de esta herramienta en el ámbito del análisis de los datos del funcionamiento de redes de datos y la relevancia de la asimilación de su uso en este ámbito [12].

Indexación de datos:

El valor central de Splunk para la mayoría de las organizaciones es su capacidad única de indexar orígenes de datos de modo que pueden llevarse a cabo búsquedas con un nivel alto de eficiencia para el análisis, la generación de reportes estadísticos y de alertas oportunas, según sea el caso [12].

Índices de Splunk:

Datos crudos mediante la creación de un mapa basado en el tiempo. Antes de que Splunk pueda buscar cantidades masivas de datos, Los índices de splunk son similares a los índices de un libro de texto, que apuntan a páginas con palabras clave específicas. En Splunk, las "páginas" son llamados sucesos [12].

Splunk divide un flujo de datos de máquina en eventos individuales. Un evento en datos de máquina puede ser tan simple como una línea en un archivo de registro o tan complicado como un rastreo de pila que contiene varios cientos de líneas [12].

2.2.5 Machine Learning:

Es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente. Aprender en este contexto quiere decir identificar patrones complejos en millones de datos. La máquina que realmente aprende es un algoritmo que revisa los datos y es capaz de predecir comportamientos futuros. *Automáticamente*, también en este contexto, implica que estos sistemas se mejoran de forma autónoma con el tiempo, sin intervención humana [14].

Ámbitos de aplicación del Machine Learning

Muchas actividades actualmente ya se están aprovechando del Machine Learning. Sectores como el de las compras online, el *online advertising* – dónde poner un anuncio para que tenga más visibilidad en función del usuario que visita la web o los filtros anti-spam llevan tiempo sacando partido a estas tecnologías [14].

El campo de aplicación práctica depende de la imaginación y de los datos que estén disponibles en la empresa. Estos son algunos ejemplos más [14]:

- Detectar fraude en transacciones.
- Predecir de fallos en equipos tecnológicos.
- Prever qué empleados serán más rentables el año que viene (el sector de los Recursos Humanos está apostando seriamente por el Machine Learning).
- Seleccionar clientes potenciales basándose en comportamientos en las redes sociales, interacciones en la web…
- Predecir el tráfico urbano.
- Saber cuál es el mejor momento para publicar tuits, actualizaciones de Facebook o enviar las newsletter.
- Hacer prediagnósticos médicos basados en síntomas del paciente.

- Cambiar el comportamiento de una app móvil para adaptarse a las costumbres y necesidades de cada usuario.
- Detectar intrusiones en una red de comunicaciones de datos.
- Decidir cuál es la mejor hora para llamar a un cliente.

2.2.6 Algoritmo Random Forest

Uno de los métodos más populares usados por los científicos de datos es el algoritmo Random Forest, es uno de los mejores algoritmos de clasificación, capaz de organizar grandes cantidades de datos con exactitud [15].

Random forest es una combinación de árboles predictores en la que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. El algoritmo para inducir un random forest fue desarrollado por Leo Breiman y Adele Cutler, siendo Random forests su marca de fábrica [15].

Este algoritmo mejora la precisión en la clasificación mediante la incorporación de aleatoriedad en la construcción de cada clasificador individual. Esta aleatorización puede introducirse en la partición del espacio (construcción del árbol), así como en la muestra de entrenamiento. El Random Forest comienza con una técnica de aprendizaje automático estándar llamada “árbol de decisiones”, que, en cuanto al conjunto, corresponde a un aprendizaje. En un árbol de decisión, una entrada se introduce en la parte superior y hacia abajo a medida que atraviesa el árbol de los datos, los cuales se acumulan en conjuntos más pequeños [15].

El Método Random Forest es fácil de aprender y de usar, tanto por profesionales como por personas con menos experiencia. Con poca investigación y tiempo de desarrollo, puede ser utilizado por personas con poca base estadística. En pocas palabras, este método nos

permite hacer de manera segura predicciones más precisas y sin la mayoría de los errores básicos comunes a otros métodos [15].

Principales beneficios:

- Precisión.
- Funciona de manera eficiente con bases de datos de gran tamaño.
- Maneja miles de variables de entrada sin necesidad de borrado.
- Da estimaciones de qué variables son importantes en la clasificación.
- Genera una estimación objetiva interna de la generalización de errores.
- Proporciona métodos eficaces para estimar datos que faltan.
- Los "bosques" generados se pueden guardar para uso futuro en otros datos.
- Los prototipos se calculan de manera que proporcionan información acerca de la relación entre las variables y la clasificación.
- Calcula proximidades entre pares de casos que se pueden utilizar en la agrupación, la localización de los valores extremos, o (en escala) dan interesantes vistas de los datos
- Ofrece un método experimental para la detección de interacciones de variables.

Tanto R como Python tienen potentes paquetes que implementan Random Forests [15].

SPL:

El Lenguaje de Procesamiento de Búsqueda Splunk (SPL) abarca todos los comandos de búsqueda y sus funciones, argumentos y cláusulas. Los comandos de búsqueda le dicen al software Splunk qué hacer con los eventos que recuperó de los índices. Por ejemplo, debe utilizar un comando para filtrar información no deseada, extraer más información, evaluar nuevos campos, calcular estadísticas, reordenar los resultados o crear un gráfico [16].

Algunos comandos de búsqueda tienen funciones y argumentos asociados con ellos. Estas funciones y sus argumentos sirven para especificar cómo actúan los comandos en sus resultados y en qué campos actúan. Por ejemplo, se puede utilizar funciones para formatear los datos en un gráfico, describir qué tipo de estadísticas calcular y especificar qué campos evaluar. Algunos comandos también utilizan cláusulas para especificar cómo agrupar los resultados de búsqueda [16].

CAPÍTULO III

METODOLOGÍA

3.1. Modalidad de la investigación

El presente proyecto considera las siguientes modalidades de investigación:

- **De campo:** Se acudirá al lugar de los hechos en los predios de la Facultad de Ingeniería en Sistemas Electrónica e Industrial - UTA (Huachi Chico).

- **Bibliográfica-Documental**, el proyecto se sustentará en artículos científicos, libros físicos y digitales e Internet como fuentes de información.

3.2 Población y muestra

Por la característica del proyecto no es necesario definir población.

3.3 Recolección de la información

Para la recolección, procesamiento y análisis de la observación se aplicará el método Observación con usuarios y personal que ocupan la red de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato.

3.4 Procesamiento de análisis de datos

El procesamiento y análisis de datos se realizará acorde a los siguientes ítems.

1. Obtener y recolectar información sobre el estado actual de la red.
2. Validar y Analizar la información recolectada.
3. Organizar la información obtenida por fechas.
4. Analizar los logs con una herramienta capaz de detectar botnets mediante minería de datos.
5. Creación de alertas para las posibles amenazas.
6. Diseñar un mecanismo de defensa y bloqueo para las conexiones a los servidores C&C.

3.5 Desarrollo del proyecto

1. Análisis
    1.1. Análisis de datos sobre la red de la FISEI.
        1.1.1. Definir el método y el tipo de archivos de los de los servidores.
        1.1.2. Obtener archivos de log y tráfico de red.
    1.2. Establecimiento del alcance del proyecto y aplicación.
    1.3. Definición de ámbitos del proyecto.

2. Desarrollo

2.1 Instalación y aplicación de un software de análisis de redes capaz de detectar botnets mediante minería de datos.

2.2 Creación de las instancias y orígenes de datos.

3. Recolección de Datos.

3.1 Recolectar la mayor cantidad de datos, mediante archivos y tráfico de red.

3.2 Almacenar y analizar los datos recolectados mediante el software.

4. Diseño.

4.1 Análisis de posibles y futuras amenazas de botnets y conexiones a servidores C&C.

4.2 Diseño de un mecanismo de defensa, bloqueo y auto detección de botnets.

4.3 Implementar el trabajo realizado.

CAPÍTULO IV
DESARROLLO DE LA PROPUESTA

4.1 Desarrollo para el análisis de detección de Botnets.

4.1.1. Análisis de la red de la FISEI.

Servidores de la Facultad:

En la unidad de laboratorios y redes de la FISEI se cuenta con 3 Servidores para la conexión de redes y acceso a internet los cuales son:

Servidor Proxy.- Este servidor se encarga de controlar los siguientes parámetros:

- Niveles de acceso: el acceso a la red de datos, es configurado por el protocolo AAA (*Authentication, Authorization and Accounting*), además por las políticas de seguridad establecidas en la institución.

- Reglas ACL: reglas de control de acceso las cuales determinan políticas centralizadas para la mejor efectividad en cuanto a la administración de la red en la FISEI.

Servidor de Enrutamiento.- Este servidor se encarga de controlar los siguientes parámetros:

- Se encarga de la seguridad y el acceso para usuarios característicos.

- Mejora la comunicación de las redes, determina reglas de conexión entre varias subredes y accesos a las mismas.

Servidor DNS.- Este servidor se encarga de controlar los siguientes parámetros:

- Traduce nombres entendibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente o dentro de la misma red.

- Utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet.

Es importante destacar en este proyecto que el servidor de DNS está centralizado para toda la institución como tal es decir la Universidad Técnica de Ambato, se toma en cuenta como parte de la red de la FISEI ya que los equipos y dispositivos conectados a esta red también ocupan el servicio mencionado y registran logs y tráfico.

Diagrama de la Estructura de red de la FISEI:

Como se puede observar en la figura 2. La red de la FISEI se compone de la siguiente manera [22]:

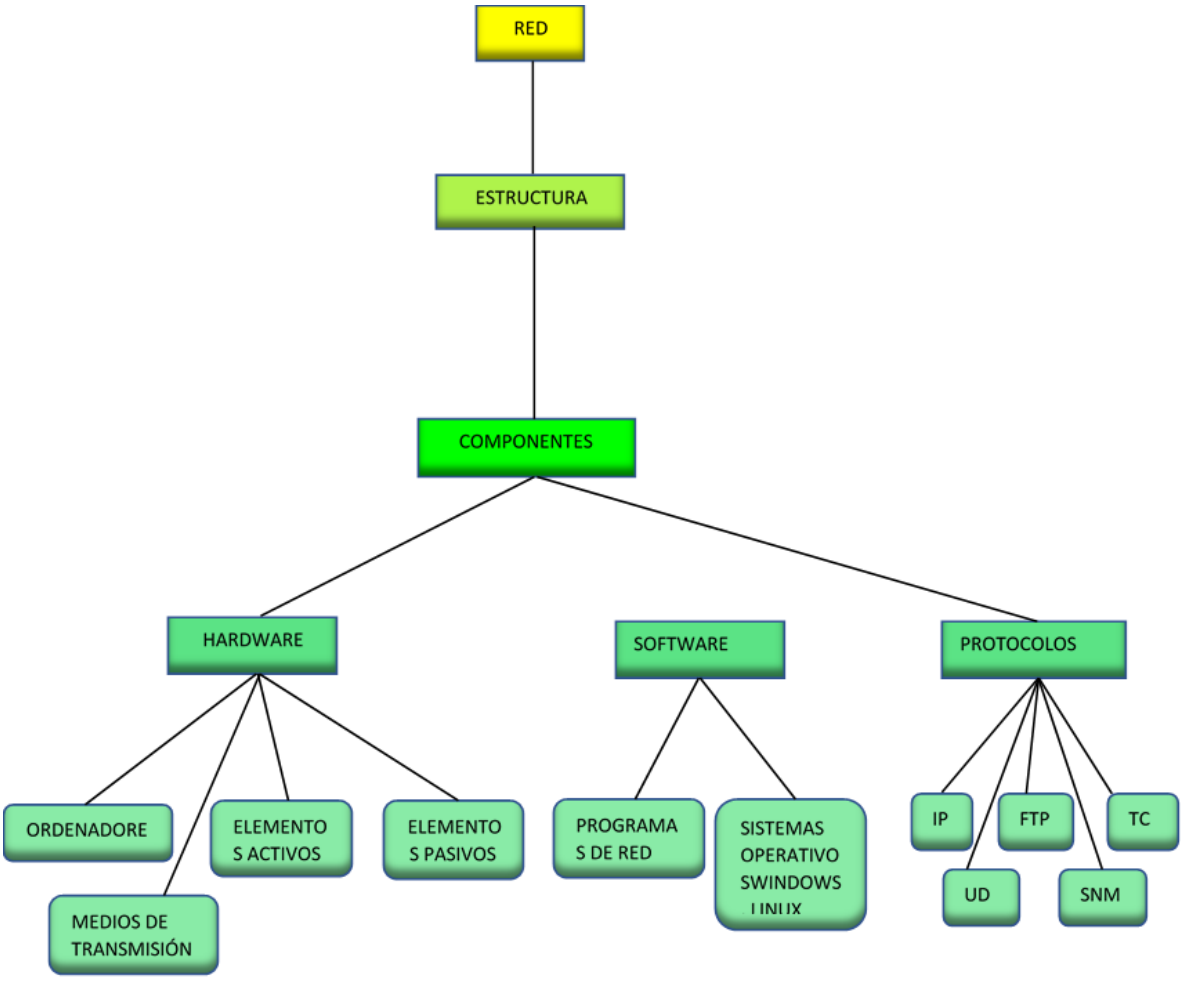


Fig. 2: Estructura de la red de la FISEI

La red de la FISEI se encuentra configurada en VLANS, para organizar a los usuarios de la red en grupos de trabajo lógico que sean independientes de la topología física del armario de instalación. Esto, a su vez, puede reducir el costo de movimientos, agregación, y cambios mientras se aumenta la flexibilidad de la red. Cada VLAN debe soporta el algoritmo Spanning Tree (IEEE 802.1d) para evitar bucles en la red. De la información obtenida mediante el monitoreo de la red de datos interna de la FISEI, se determinó el ancho de banda que utiliza, por otro lado se tiene en cuenta que el proveedor de Internet es Telconet con una comunicación de Fibra óptica.

Esquema general de comunicaciones de la FISEI ver figura 3:

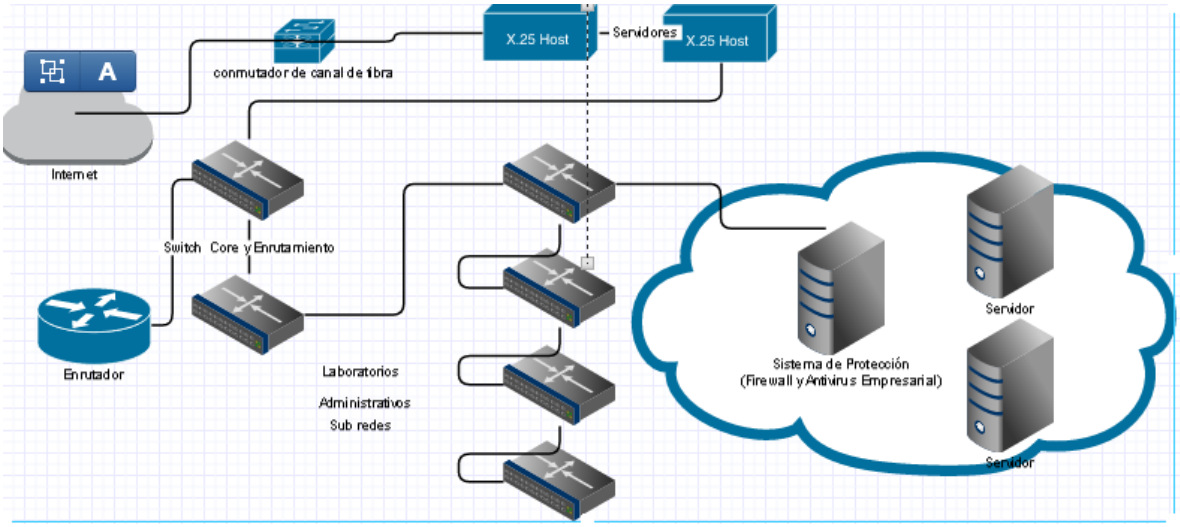


Fig. 3: Esquema General de Comunicaciones

Informe de peticiones DNS:

A.1. Información de registro
Tiempo de Inicio: 14:22:18
Tiempo de Finalización: 16:01:52

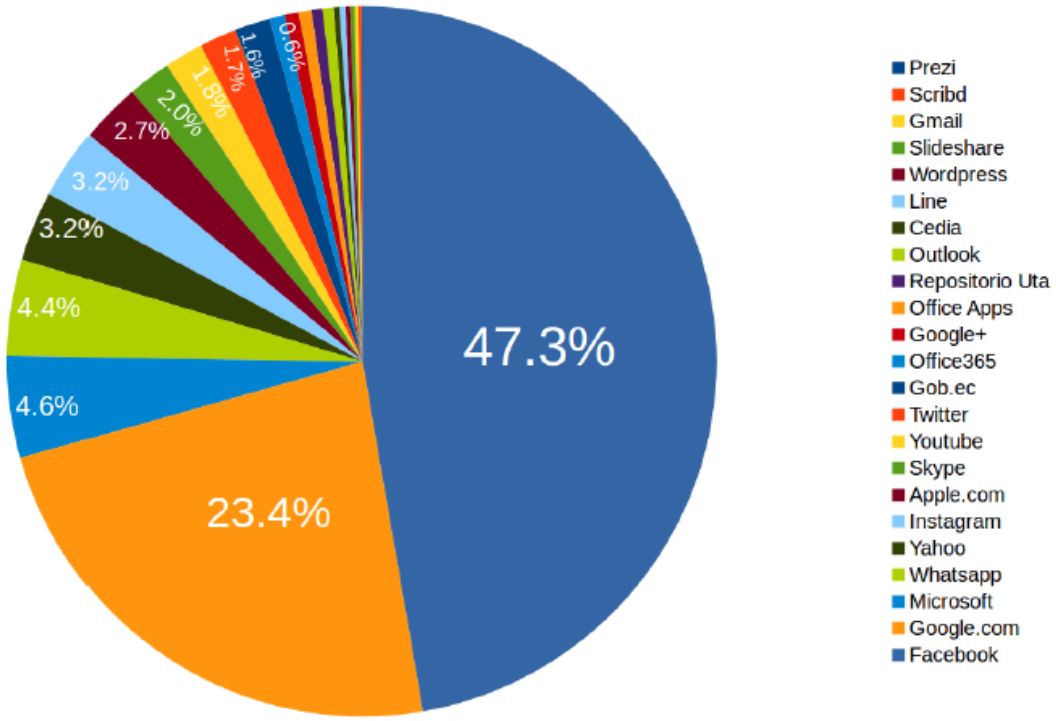


Fig. 4: Informe de Peticiones DNS

Según el análisis de la red en el presente proyecto, la bibliografía consultada y los antecedentes investigativos se decidió utilizar los archivos de logs del servidor DNS por los siguientes aspectos a continuación listados:

- En la actualidad los servidores C&C utilizan técnicas del *tipo Fast-Flux* y otras similares para dificultar el trazado de los servidores de control y detectar sus direcciones IP, los cuales pueden ser cambiados día a día. Los servidores C&C pueden también saltar de un ámbito DNS a otro, usando algoritmos de generación de dominio que suelen crear nuevos nombres DNS para su conexión con las botnets [10].

- Los servidores que utiliza la FISEI para el control y gestión de la red (proxy – Servidor de Enrutamiento) almacenan logs pero el objetivo principal de estos son controlar, gestionar y bloquear direcciones o redes por lo que no proporciona la información necesaria para determinar la conexión entre un servidor C&C y una botnet.

- La FISEI cuenta con equipos de interconexión y control de su red pero no con equipos especializados en la materia de botnets.

- Existen antecedentes investigativos y documentos técnicos [3][4][5], los cuales también utilizan registros y trafico DNS los cuales sustentan esta investigación.

4.1.2 Equipo para el Desarrollo de la propuesta.

Para el desarrollo del proyecto se utilizó la siguiente máquina portátil:

Marca: Alienware.
Modelo: M14x.
Procesador: Intel(R) Core(TM) i5-4200M CPU @ 2.50GHz .
Memoria RAM: 8GB DDR3.
Disco Duro: 600GB sata.
Sistema Operativo: Debian GNU/Linux 8.5 (jessie).

Las medidas de tiempo y cargas al sistema a continuación fueron tomadas en cuenta con el equipo mencionado.

4.1.3 Instalación de Splunk

Descarga: La descarga se realiza de la página web oficial del software; la versión que se utilizó para este proyecto será la enterprise para Linux como se puede ver en la figura 5.

https://www.splunk.com/es_es/products/splunk-enterprise.html

Fig. 5: Página de Descarga de Splunk

Seleccionar splunk-6.6.2-4b804538c686-linux-2.6-amd64.deb ya que se está utilizando la distribución debian.

Instalación:

En la consola como "root" ejecutar el siguiente comando:

```
dpkg -i splunk-6.6.2-4b804538c686-linux-2.6-amd64.deb
```

Es necesario que el sistema cumpla con las dependencias que el software requiere.

Iniciar splunk:

Una vez instalada la aplicación es necesario iniciar el servicio con el siguiente comando como administrador:

```
/opt/splunk/bin/splunk start
```

Al ejecutar el comando, el servicio se iniciará de la siguiente manera en la figura 6:

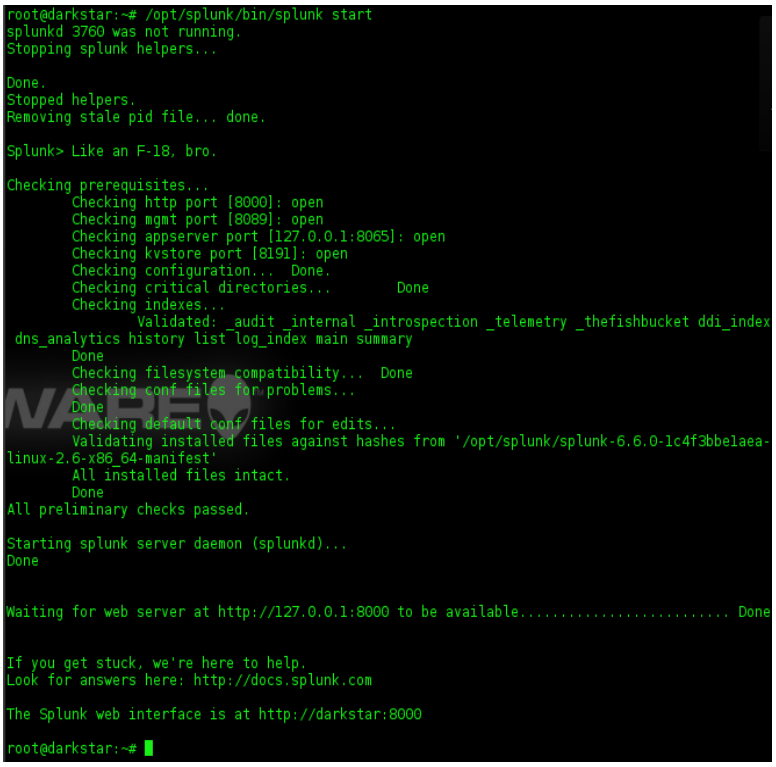


Fig. 6: Instalación de splunk

La información indica que la interfaz web de splunk es la siguiente: http://darkstar:8000

Instalación de Plugins adicionales necesarios para el proyecto:

Splunk permite la instalación de plugins adicionales desde su misma interfaz web como se observa en la figura 7.

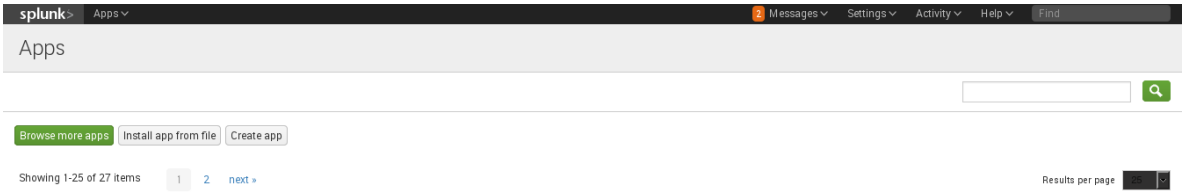


Fig. 7: Plugins Adicionales

Instalación de Splunk Machine Learning Toolkit:

Esta app permite:

- Predecir campos numéricos.
- Predecir campos categóricos (Regresión logística).
- Detectar valores atípicos numéricos (estadísticas de distribución).
- Detectar valores atípicos categóricos (medidas probabilísticas)
- Series de tiempo de predicción.
- Eventos numéricos del clúster.

Para el funcionamiento de esta app, es necesario instalar Python for Scientific Computing dentro de splunk asi como en la figura 8.

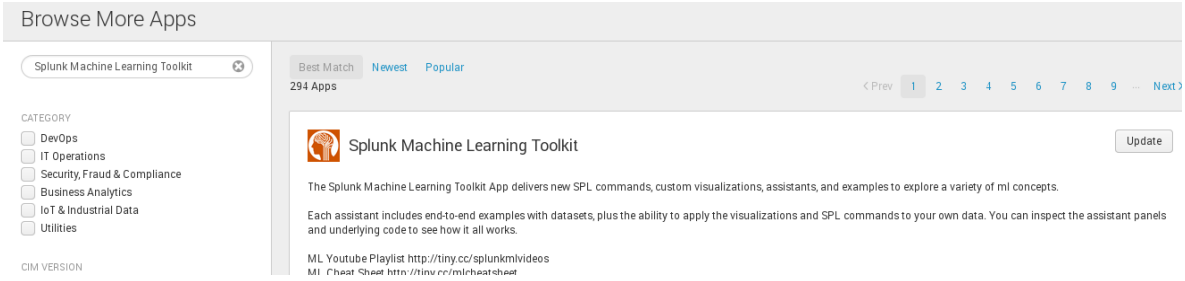


Fig. 8: Instalación de Splunk Machine Learning

Instalación de Splunk Add-on for ISC BIND:

Este plugin permite obtener el tipo de dato para los archivos de log DNS del servicio.

Instalación de DNS Analytics for Splunk:

Esta app permite analizar los eventos y ficheros dns en un ambiente amigable. Además incluye consultas útiles, tareas y gráficos a partir de los logs dns. Necesita una "API key" gratuita, la que se puede obtener de los desarrolladores figura 9.

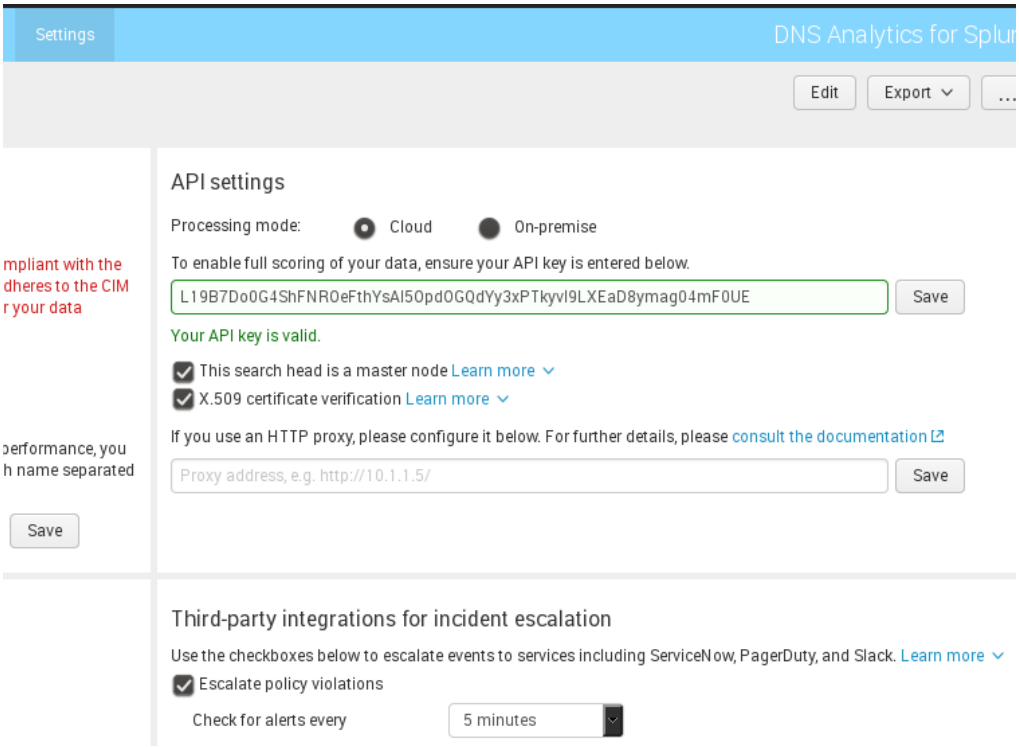


Fig. 9: Instalación de DNS Analytics for Splunk

Instalación de URL Toolbox:

Este conjunto de herramientas permite analizar y separar las url de los *queries* almacenados en los logs figura 10.

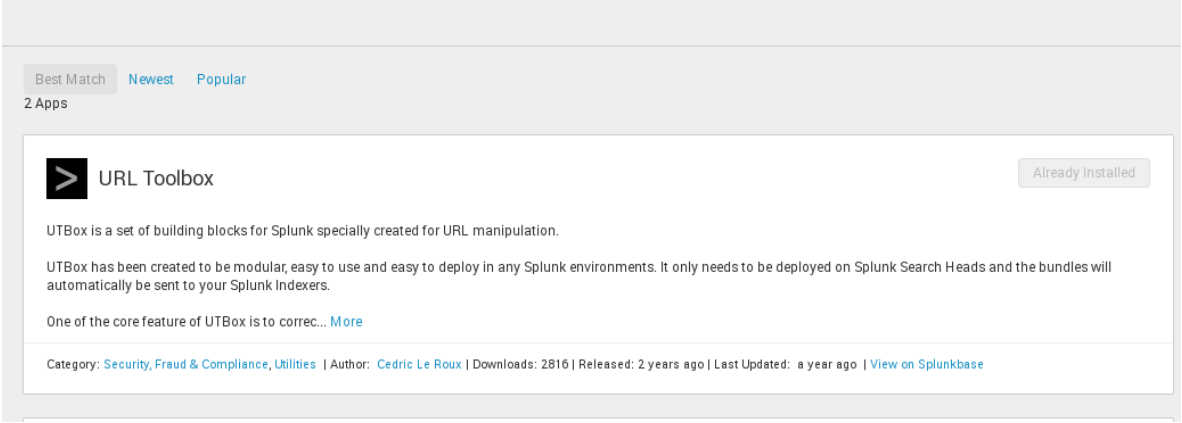


Fig. 10: Instalacion de URL Toolbox

4.1.3 Creación de los *datasources* en base a los archivos recolectados en la FISEI.

Con los complementos instalados, es necesario crear los data inputs como los denomina splunk de los archivos de logs de dns.

En el menú de splunk en “settings” seleccionar “data input” como se observa en la figura 11:

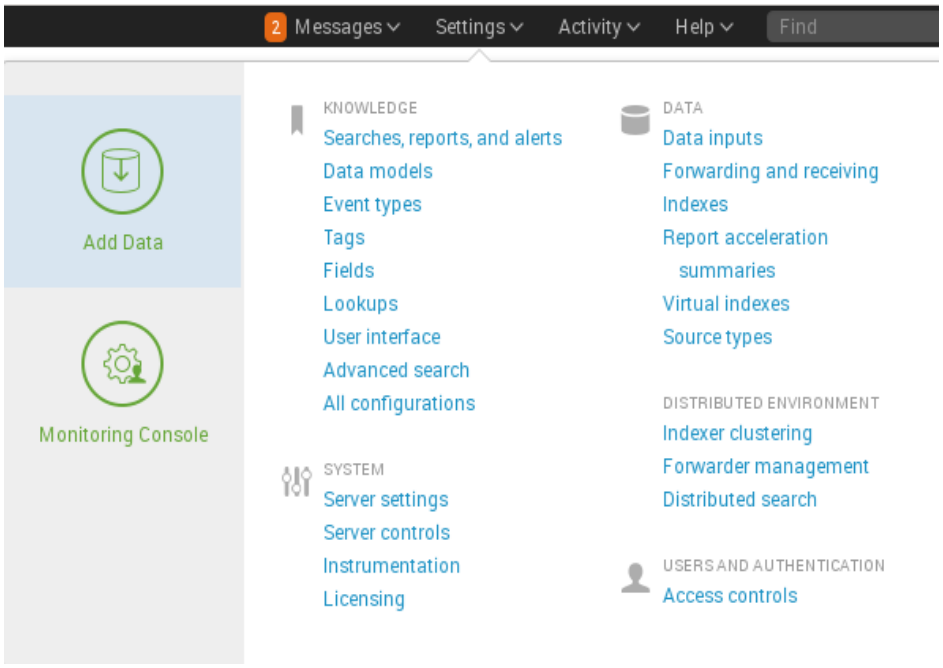


Fig. 11: Creacion de Datasources en Splunk

A continuación se debe seleccionar el tipo de data input “Files & Directories” como en la figura 12.

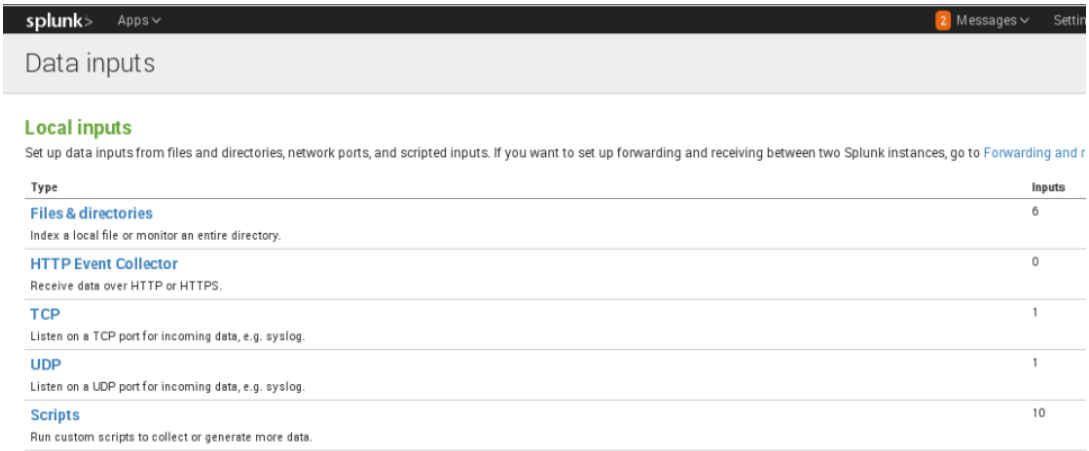


Fig. 12: Data Inputs

El siguiente paso es seleccionar uno de los archivos de logs que va a analizar la herramienta splunk así como se observa en la figura 13.

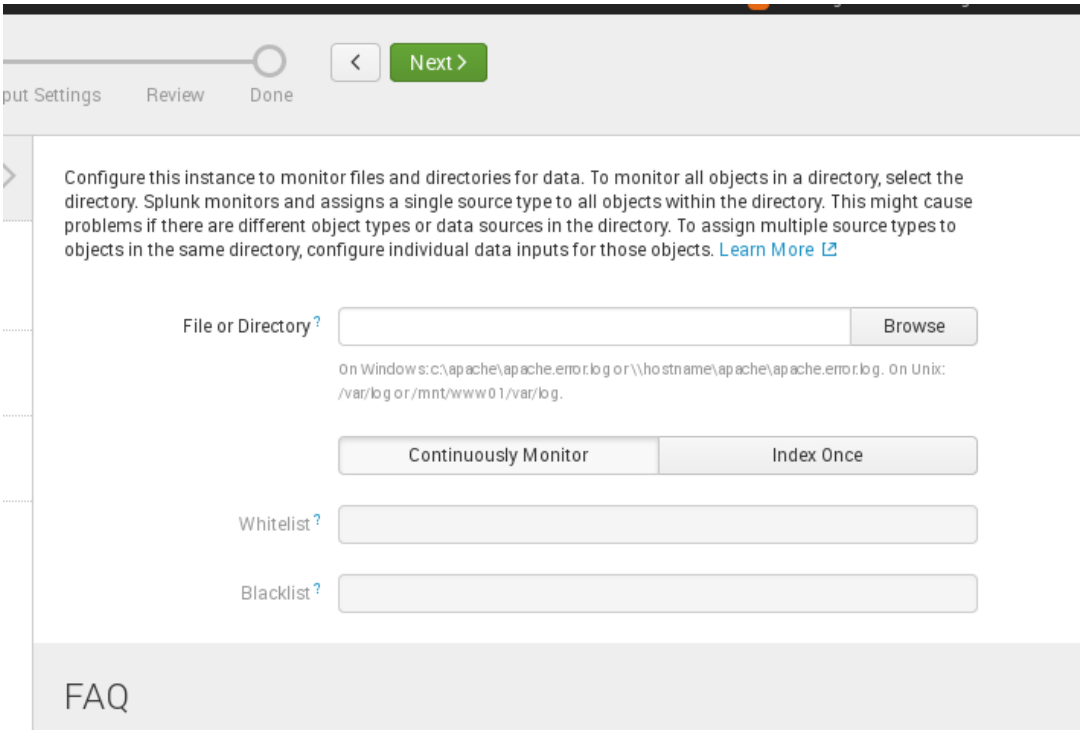


Fig. 13: Directorio de Archivos de origen de datos

Es necesario seleccionar el tipo de dato para poder analizar y realizar la minería de datos. En el caso de este trabajo es “isc:bind:query” ya que los archivos son *logs* de *queries* de *bind* como en la figura 14.

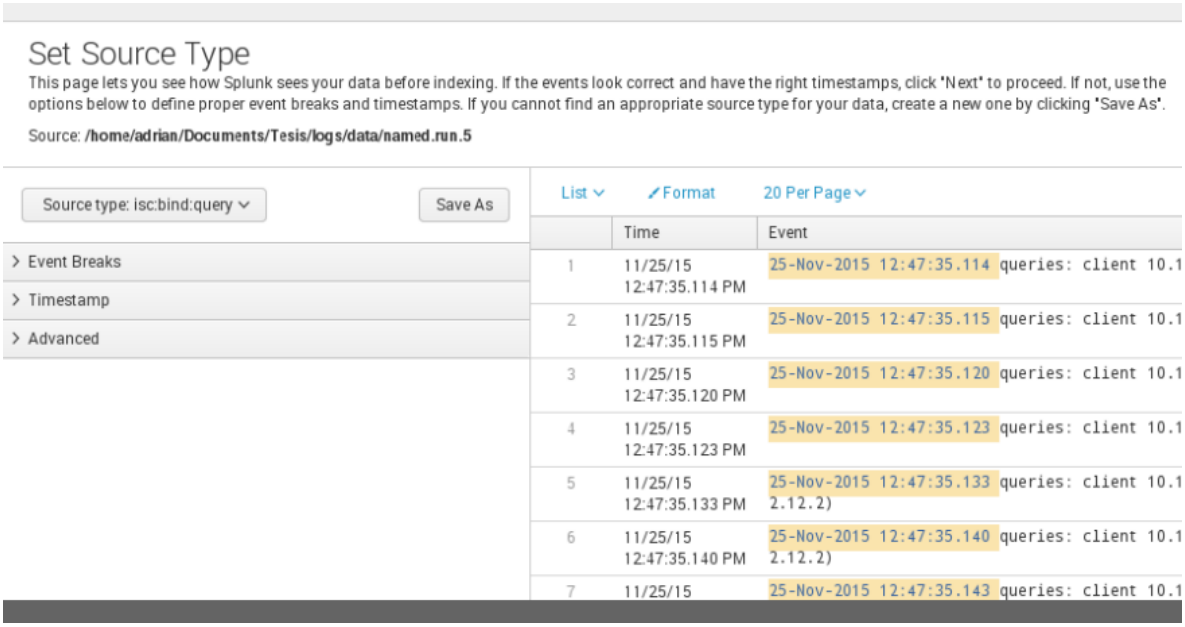


Fig. 14: Establecer el Tipo de Datos en Splunk

Se deben definir las siguientes propiedades y seleccionar un índice en común para todos los *data inputs*, con el fin de simplificar las búsquedas en una sola consulta como se puede observar en la figura 15.

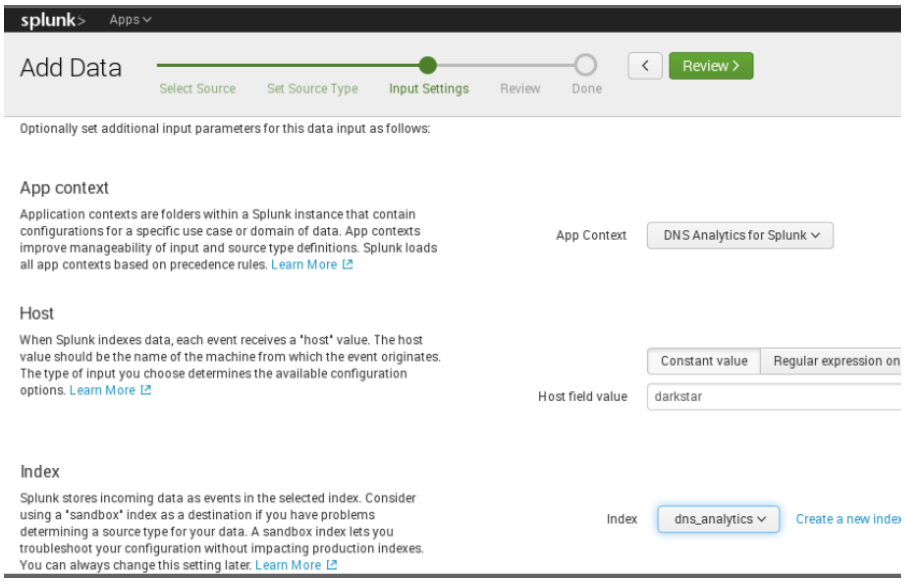


Fig. 15: Propiedades de DataSource

Si todos los pasos son satisfactorios splunk informa con una pantalla como la siguiente figura 16:

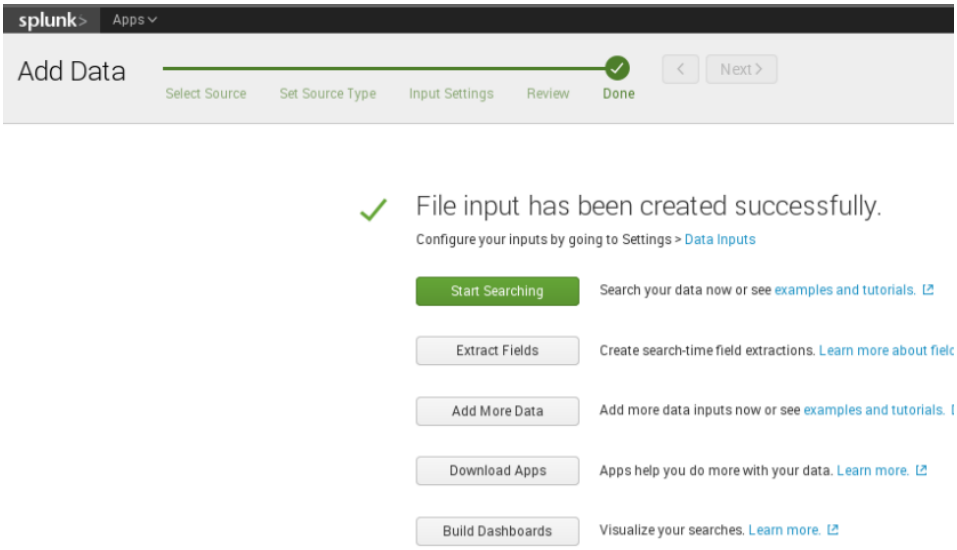


Fig. 16: Confirmacion de DataSource

Para que los índices se actualicen al instante y se puedan consultar los nuevos data inputs es necesario presionar el botón “Start Searching” este proceso toma alrededor de 45 segundos en la máquina descrita al inicio en finalizar la actualización como en la figura 17.

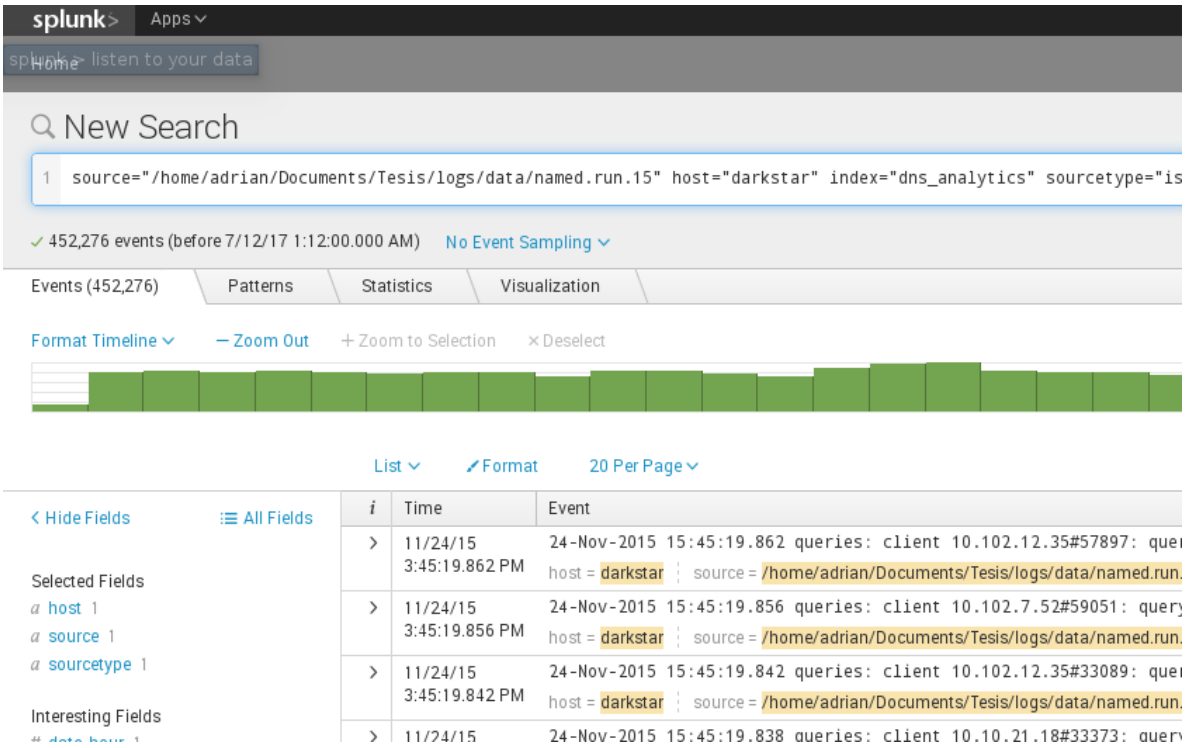


Fig. 17: Busqueda de un DataSource

Se han ingresado al sistema splunk un total de 36 archivos de logs. El sistema tardó 7 minutos con 34 segundos en indexar un total de 18748713 eventos en la máquina mencionada al inicio. Con estos eventos se puede hacer consultas, gráficos estadísticos de varios tipos.

4.1.4 Minería de Datos.

Gracias a los elementos incorporados de Splunk se pueden realizar tareas de minería de datos desde las más simples hasta las más complejas. Con los datos disponibles se pueden generar varios reportes y gráficos referentes al análisis de los log de *queries* del servidor de DNS, para el análisis de la red de la FISEI.

4.1.5 Ejemplos de análisis de los dns-queries.

Top 20 de los *queries* más solicitados figura 18:

```
sourcetype="isc:bind:query"| top limit=20 query
```

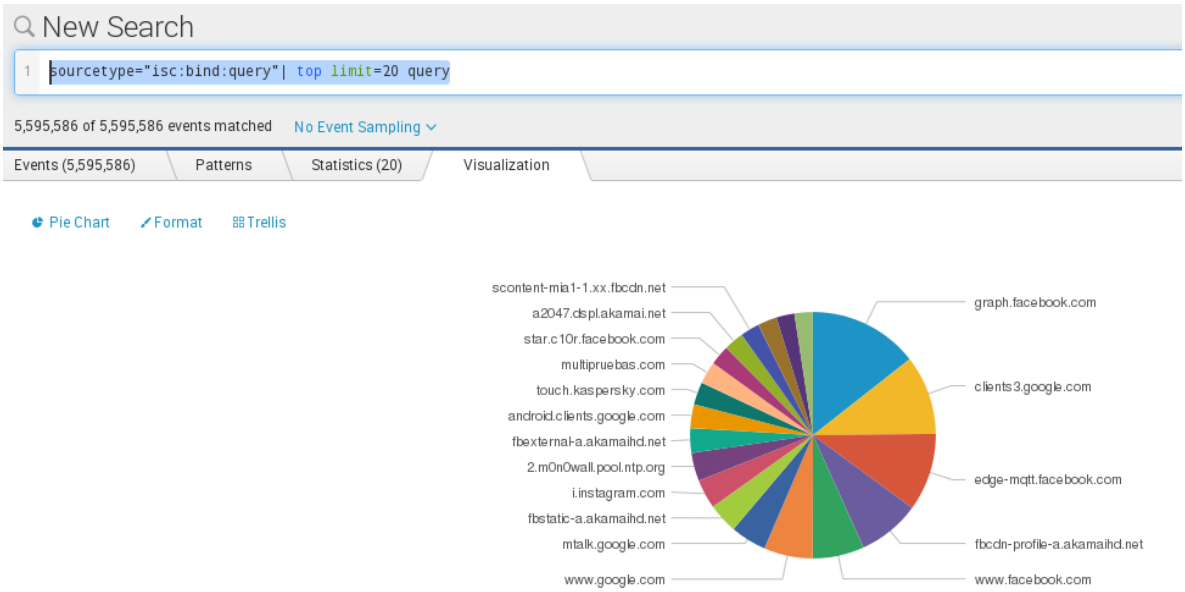


Fig. 18: Top 20 de queries más solicitados

Gráfico de horas donde más peticiones al servidor se realizan figura 19:

```
sourcetype="isc:bind:query"| top limit=20 date_hour
```

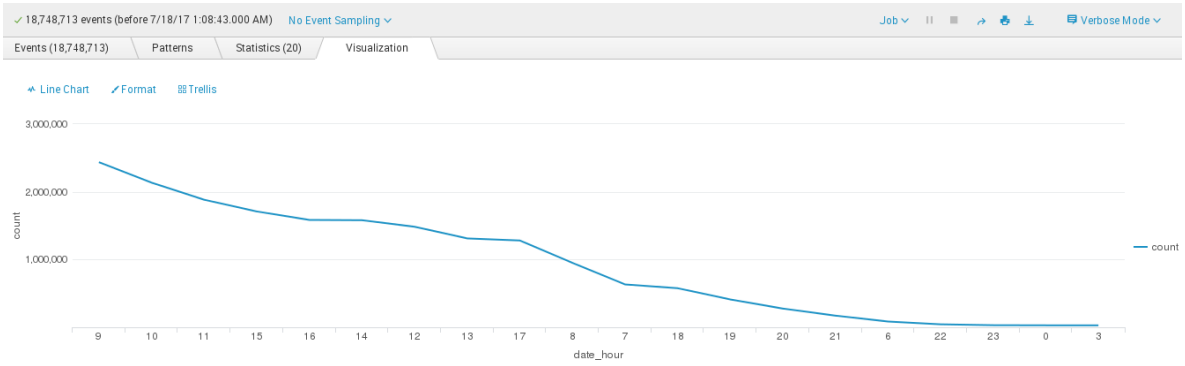


Fig. 19: Grafico por horas

Gráfico por “record-type” figura 20

```
sourcetype="isc:bind:query"| top record_type
```

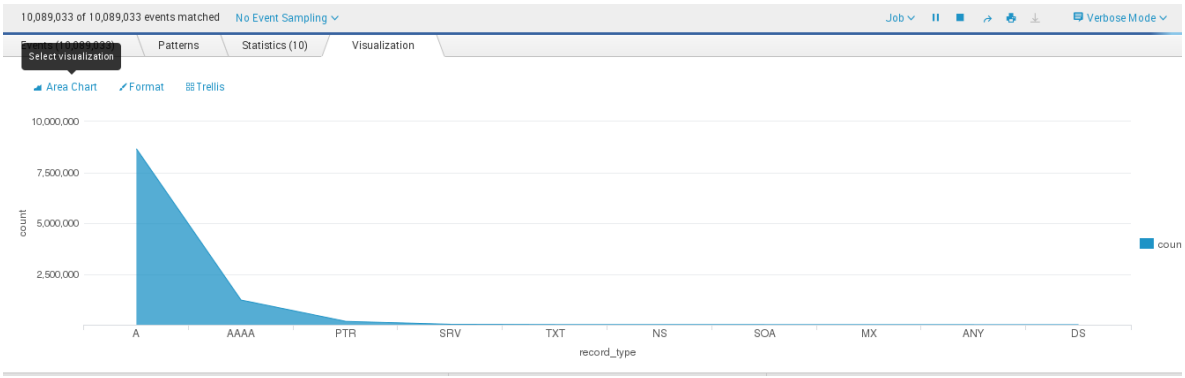


Fig. 20: Gráfico por “record-type”

Como se puede observar en las figuras 15, 16, 17. Splunk provee de un gran conjunto de consultas prediseñadas que obtienen información relevante para sacar conclusiones y tomar decisiones. En cuanto al análisis de una red, se pueden construir varias consultas personalizadas gracias a la cantidad de plugins que posee.

4.1.6 Detección de botnets.

Para la detección de botnets, en este proyecto se han considerado los siguientes criterios referentes a características propias de los servidores C&C.

Peticiones de DNS fallida (NXDOMAIN)

Una forma de detectar potenciales amenazas de malware pertenecientes a una botnet es el análisis estadístico de solicitudes de resolución de DNS fallidas ya que los dominios utilizados por las botnets no están siempre registrados.

Algunos tipos de botnets utilizan dominios de entropía baja para evitar su detección, en otras palabras, el uso de cada uno de los dominios no es muy probable, lo que significa que necesita una gran cantidad de dominios para funcionar y algunos de ellos pueden fallar en su resolución.

Seguimiento de dominios maliciosos

Consiste en supervisar todas las peticiones hechas al servidor de DNS y comprobar que el dominio que está resolviendo no está en ninguna lista negra, son una forma rápida y sencilla de detectar amenazas. El problema con esta técnica es que la botnet debe ser previamente conocida y sus dominios registrados, si la amenaza es nueva o si una botnet se actualiza con el uso de nuevos dominios y no se detecta.

Dominios con TTL bajos

Otro método utilizado por los creadores de botnets para dificultar su detección es la modificación de la IP asociada a un dominio, una técnica conocida como fast-flux. De esta manera, al cambiar la IP de destino, la detección de fallos es más difícil. Para llevar a cabo

este cambio estos dominios tienen un TTL o tiempo de vida muy bajo, esto obliga a los sistemas DNS a actualizar frecuentemente la memoria caché de resolución de la IP asociada al dominio o, en el caso de una TTL nula, ni siquiera almacenarla. Por lo tanto, las peticiones de DNS cuyo TTL es bajo son sospechosos.

Sin embargo, estas técnicas generan muchos falsos positivos, ya que hay sistemas legítimos conectados a Internet que utilizan este tipo de técnicas que cambian la IP asociada a un dominio, como balancear carga en sus sistemas.

Teniendo en consideración todos estos criterios, ha sido posible determinar un procedimiento integrado para sacar los queries ligados a servidores C&C en la base de datos de logs de la FISEI.

Para el análisis de botnets se ha construido la siguiente consulta la cual utiliza los plugins "Url-Toolbox" y "Splunk Machine Learning Toolkit" con los cuales se determina la entropía, dns-tunneling y conexiones a servidores C&C. Una vez que se obtiene la lista de *queries* sospechosos es necesario reducir el margen de error por lo que se empleó una técnica de aprendizaje de máquina con el algoritmo random - forest el cual nos permite establecer una predicción con un margen de error reducido.

4.1.7 Diagrama de flujo de los procesos realizados para la detección de botnets:

El siguiente proceso (figura 21) se planteó para la detección de botnets:

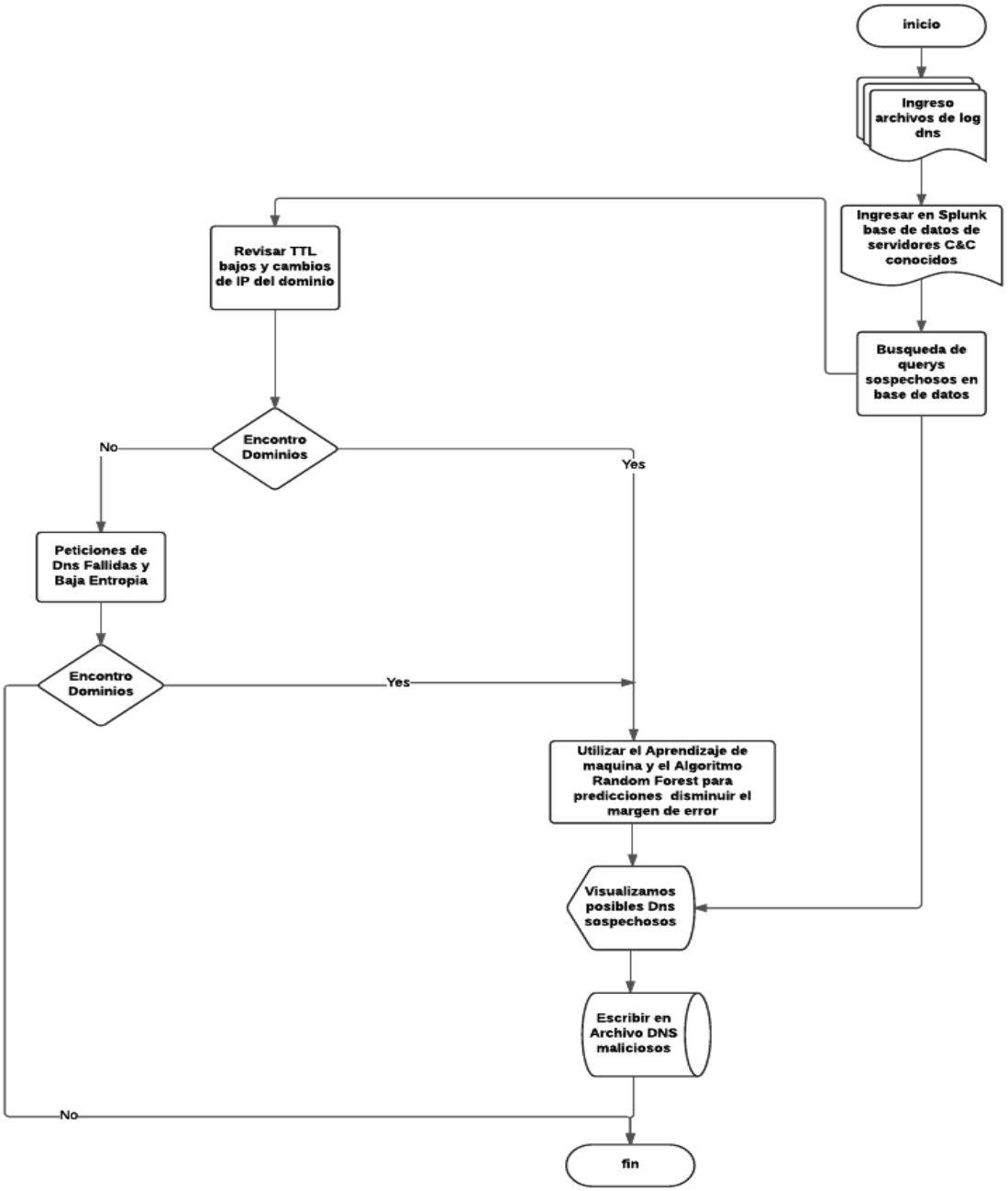


Fig. 21: Diagrama de flujo del proceso de detección de botnets

4.1.8 Query para detectar botnets:

El siguiente query se desarrolló para detectar botnets con el proceso explicado en el diagrama de flujo (figura 21), mediante el comando *lookup* analizamos la base de datos de botnets conocidos, el comando *eval* evaluna los dns y separa todo el DNS *query* en partes con lo que se puede evaluar el los TTLS y la entropía. Una vez realizado el análisis con los comandos `round('predicted(count)', 2) | eval residual = 'count' - 'predicted(count)'` se puede obtener la predicción relacionada con el machine learning .

```
index=dns_analytics   | eval ut_list = "iana"  | lookup
ut_parse_extended_lookup url as urllist list as ut_list  | spath
input=ut_subdomain_parts  | fields - ut_subdomain_parts  | lookup
boots domain AS query OUTPUTNEW domain as isFound  | where
isnotnull(isFound) | top  query | apply "default_model_name" |
eval predicted(count) = round('predicted(count)', 2) | eval
residual = 'count' - 'predicted(count)' | table "count",
"predicted(count)", residual, "percent" "query"
```

4.1.9 Margen de error

Para evaluar el margen de error se requiere realizar el mismo proceso de detección de botnets pero necesitamos agregar la función estadística "| `regressionstatistics("count", "predicted(count)")" la cual permite tener un margen de error acertado de la consulta realizada.

```
index=dns_analytics   | eval ut_list = "iana"  | lookup
ut_parse_extended_lookup url as urllist list as ut_list  | spath
input=ut_subdomain_parts  | fields - ut_subdomain_parts  | lookup
boots domain AS query OUTPUTNEW domain as isFound  | where
isnotnull(isFound) | top  query | apply "default_model_name" |
`regressionstatistics("count", "predicted(count)")`
```

4.1.10 Gráfico actual vs predicted:

Con el siguiente *query* podemos observar un cuadro comparativo del modelo actual y el que predice el sistema splunk para tener una referencia de la predicción realizada esto se logró gracias al comando: `| table _time, "count", "predicted(count)"`

```
index=dns_analytics   | eval ut_list = "iana"  | lookup
ut_parse_extended_lookup url as urllist list as ut_list  | spath
input=ut_subdomain_parts  | fields - ut_subdomain_parts  | lookup
boots domain AS query OUTPUTNEW domain as isFound  | where
isnotnull(isFound) | top  query | apply "default_model_name" |
table _time, "count", "predicted(count)"
```

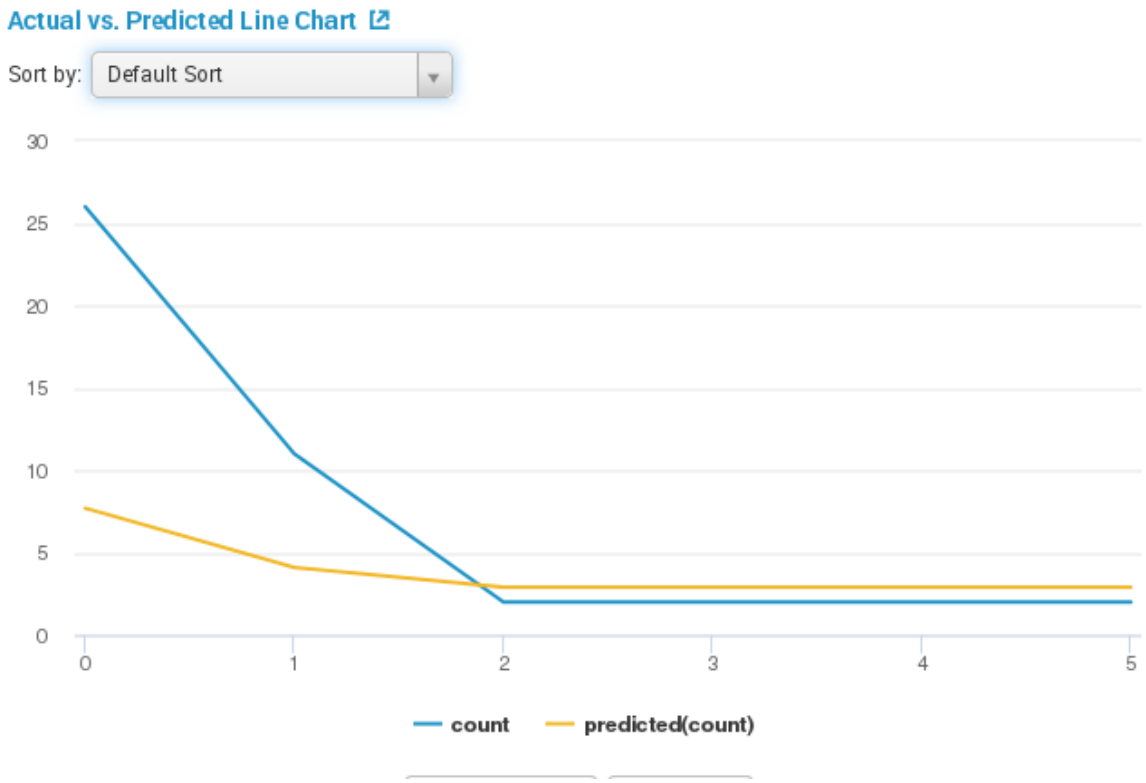


Fig. 22: Grafico Actual vs. Predicted

La figura 22 es el resultado de la consulta obtenida mediante el *query* en Splunk.

4.2. Interpretación de Resultados

Con los datos analizados se pueden obtener los siguientes resultados que pueden considerarse como servidores C&C mediante minería de datos y machine learning reduciendo así el margen de error.

4.2.1 Resultados De la Predicción:

Count	Predicted Count	Residual	Percent	Query
26	19.1	6.9	30.952381	sso.anbtr.com
22	15.2	6.8	26.190476	global.ymtracking.com
11	9.2	1.8	13.095238	sur.ly
6	2.4	3.6	7.142857	ant.trenz.pl
3	2.5	0.5	3.571429	bdb.com.my
2	2	0	2.380952	verdirectotv.com
2	2	0	2.380952	sta1.mueblesllluesma.com
2	2	0	2.380952	sipeliculas.com
2	2	0	2.380952	jump.aragontrack.com
2	2	0	2.380952	gruporeforma-blogs.com

Tabla 1: Resultados de la Predicción

La tabla 2. Nos muestra una predicción de *queries* considerados como botnets mediante el proceso realizado.

Positivos: Son considerados como positivos los resultados que tienen un "residual" mayor a 0 ya que son los que más tendencia poseen según el algoritmo de *random forest* de *machine learning*.

Falsos Positivos: Son considerados falsos positivos aquellos registros que tienen un "residual menor o igual a 0" por lo tanto se puede considerar como falsos positivos los queries que menor tendencia tienen en el modelo de predicción utilizado.

Mediante este análisis se puede concluir que las siguientes direcciones dns son botnets y falsos positivos:

Botnets
sso.anbtr.com
global.ymtracking.com
sur.ly
ant.trenz.pl
bdb.com.my

Tabla 2: Botnets Positivos

Falsos Positivos
verdirectotv.com
sta1.mueblesllluesma.com
sipeliculas.com
jump.aragontrack.com
gruporeforma-blogs.com

Tabla 3: Falsos Positivos

4.2.2 Gráfico Actual VS Predicciones:

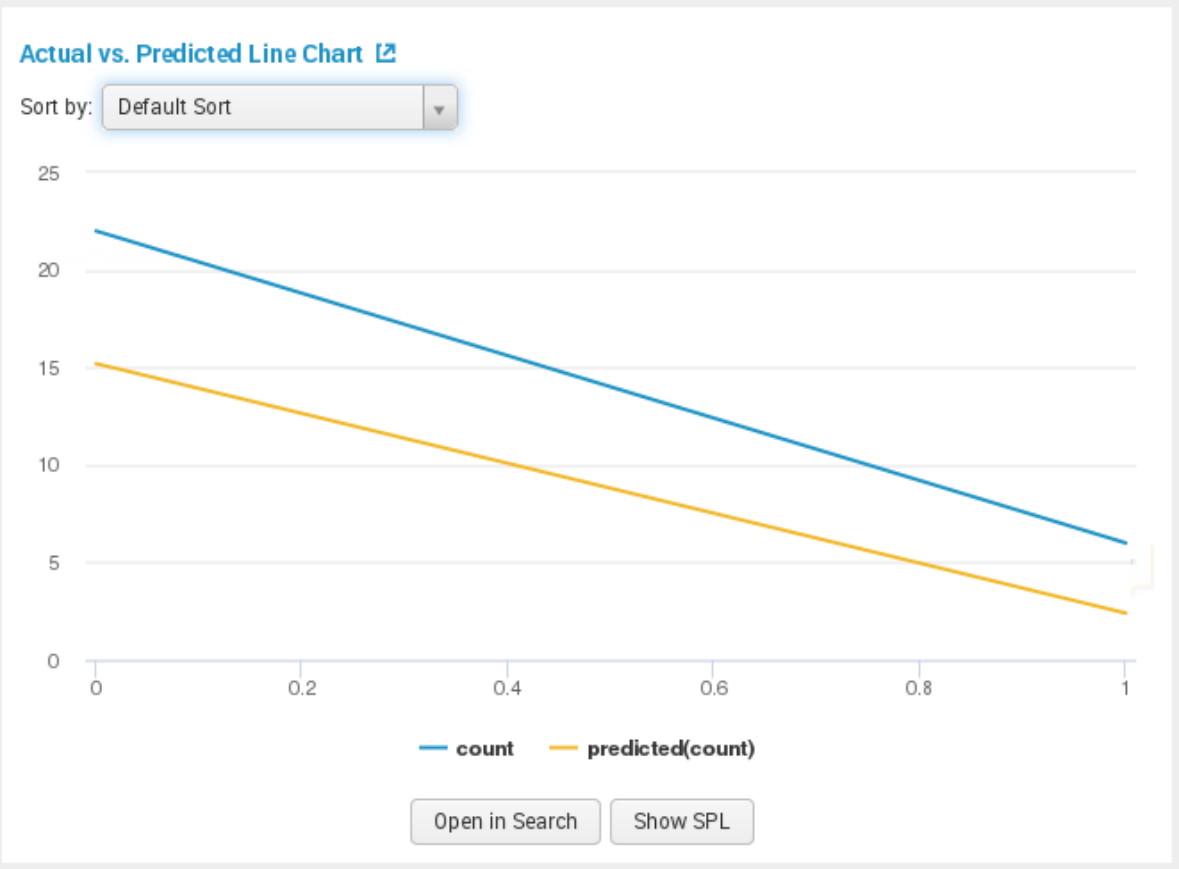


Fig. 23: Gráfico actual Vs. Predicciones

La figura 23 de predicciones del conteo de registros versus el conteo actual marcan una tendencia muy parecida y con la misma forma por lo que se puede interpretar como válida la predicción realizada por el modelo utilizado.

4.3.3 Margen de Error:

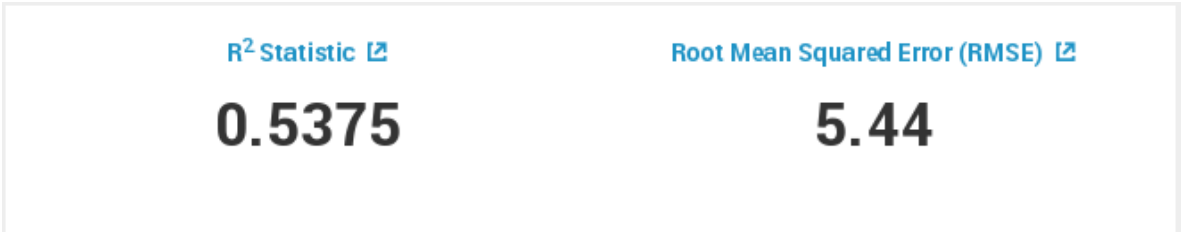


Fig. 24: Margen de error

Los resultados obtenidos en la figura 24 resulta un coeficiente de correlación de **0.54** que es un resultado aceptable ya que marca una tendencia a 1.

El margen de error es de **+-5.44** lo cual es aceptable ya que en otros trabajos similares se han obtenido cifras parecidas.

4.2.4 Comprobación de botnets:

Para comprobar si realmente los dominios obtenidos en el proceso de detección de botnets se comparó con el el "*blacklist*" de bots obtenidos de la página http://www.malware-domains.com/ la cual cuenta con dns registrados como maliciosos, se evaluaron los resultados de obtenidos considerados como positivos con resultados favorables:

sso.anbtr.com

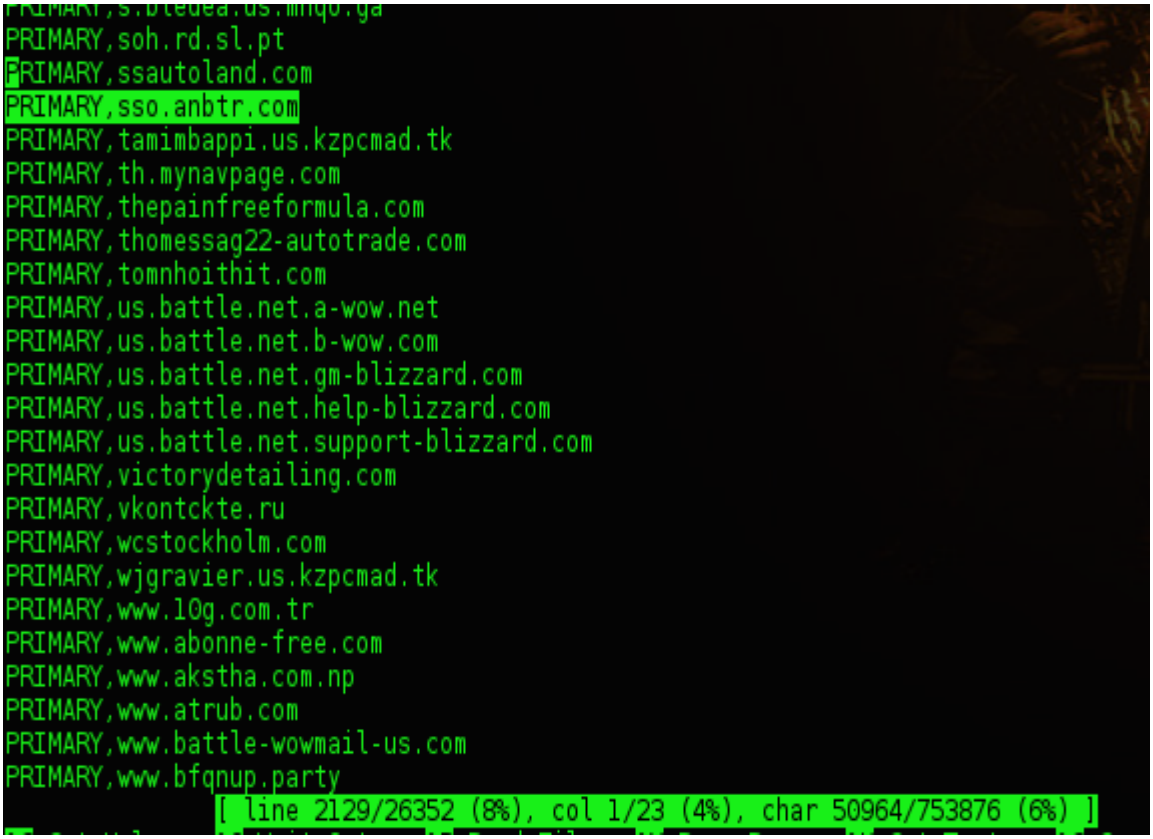


Fig. 25: Verificacion de la botnet sso.anbtr.com

El dominio mencionado se encuentra en la linea 2129 del archivo obtenido de malwaredomains (figura 25).

global.ymtracking.com

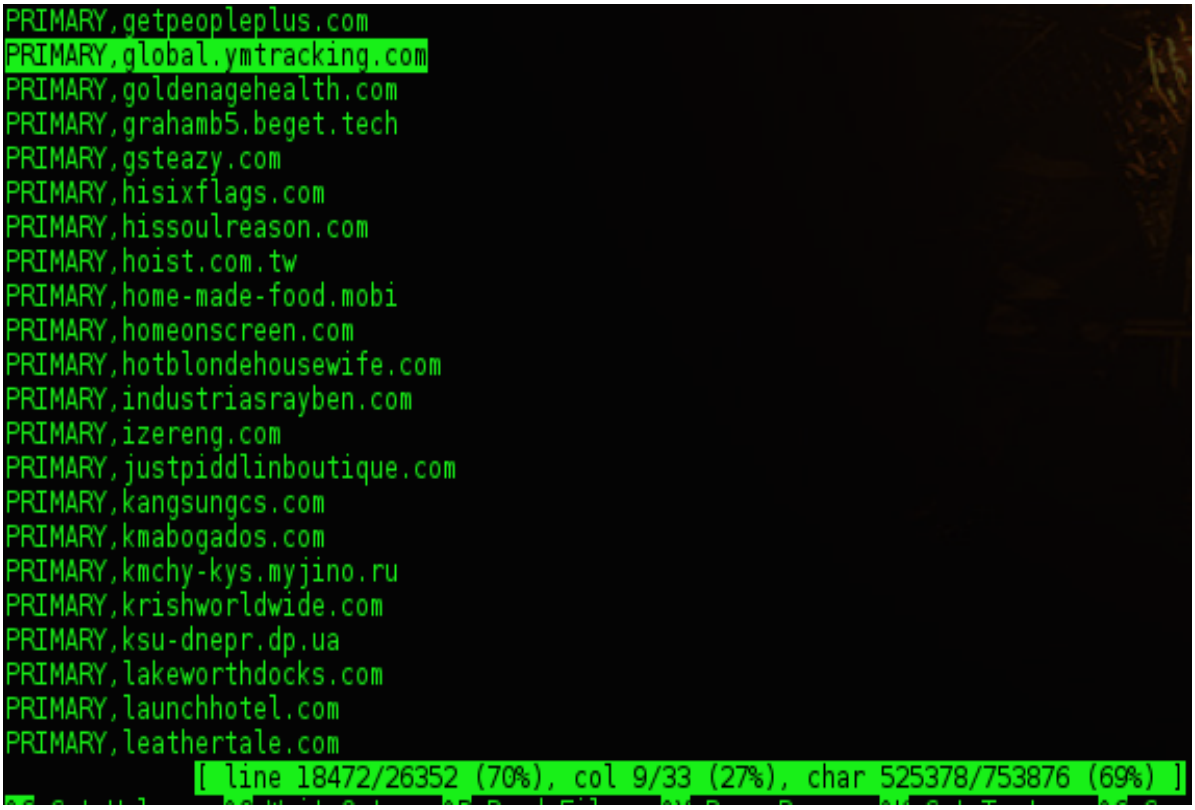


Fig. 26: Verificacion de la botnet global.ymtracking.com

El dominio mencionado se encuentra en la linea 18472 del archivo obtenido de malwaredomains (figura 26).

sur.ly

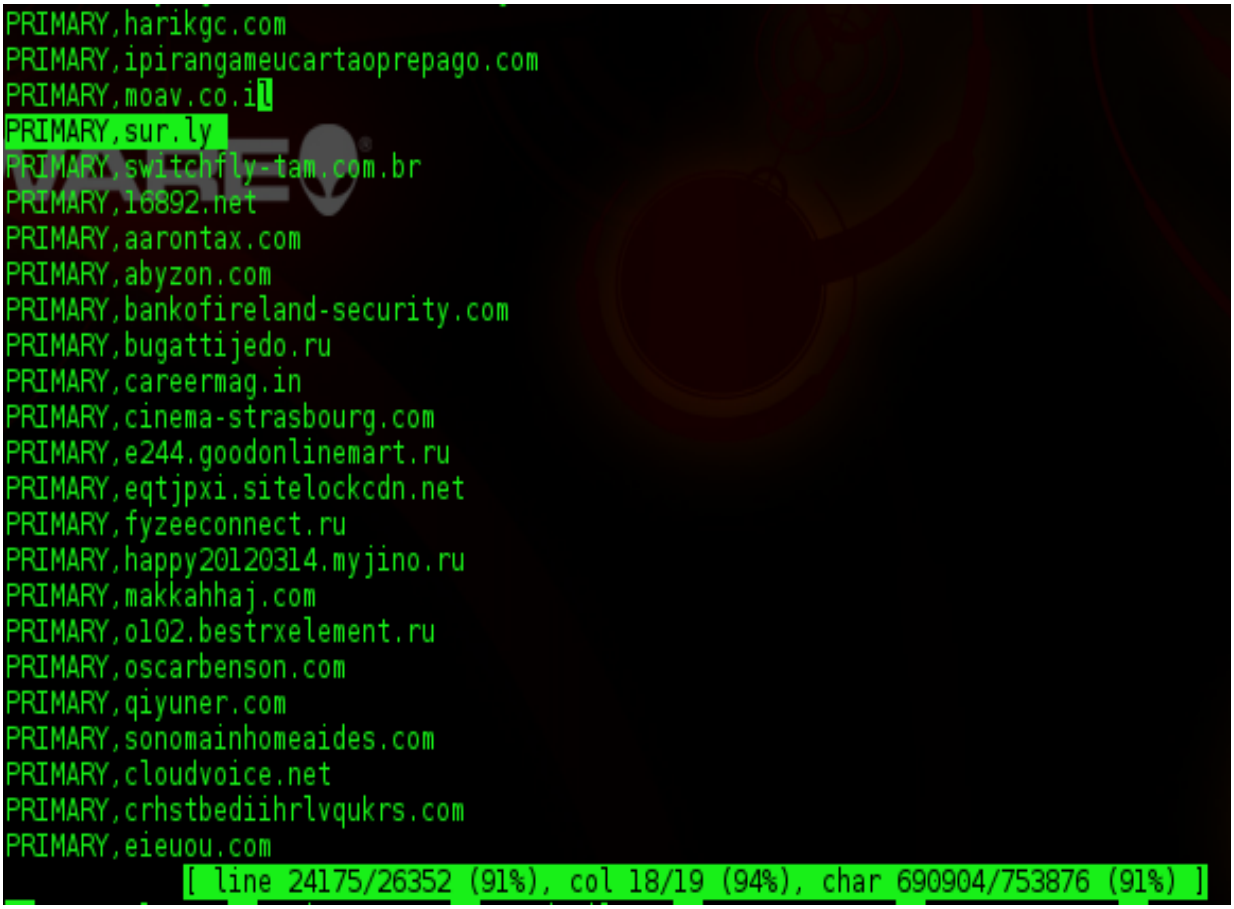


Fig. 27: Verificación de la botnet sur.ly

El dominio mencionado se encuentra en la linea 24175 del archivo obtenido de malwaredomains (figura 27).

ant.trenz.pl

Fig. 28: Verificación de la botnet ant.trenz.pl

El dominio mencionado se encuentra en la línea 803 del archivo obtenido de malwaredomains (figura 28).

bdb.com.my

Fig. 29: Verificación de la botnet bdb.com.my

El dominio mencionado se encuentra en la línea 25858 del archivo obtenido de malwaredomains (figura 29).

Analisis de las url's obtenidas como maliciosas:

Según los resultados obtenidos en la pagina web https://virustotal.com/ el dominio http://sso.anbtr.com/ es malicioso como se puede ver en la figura 30:

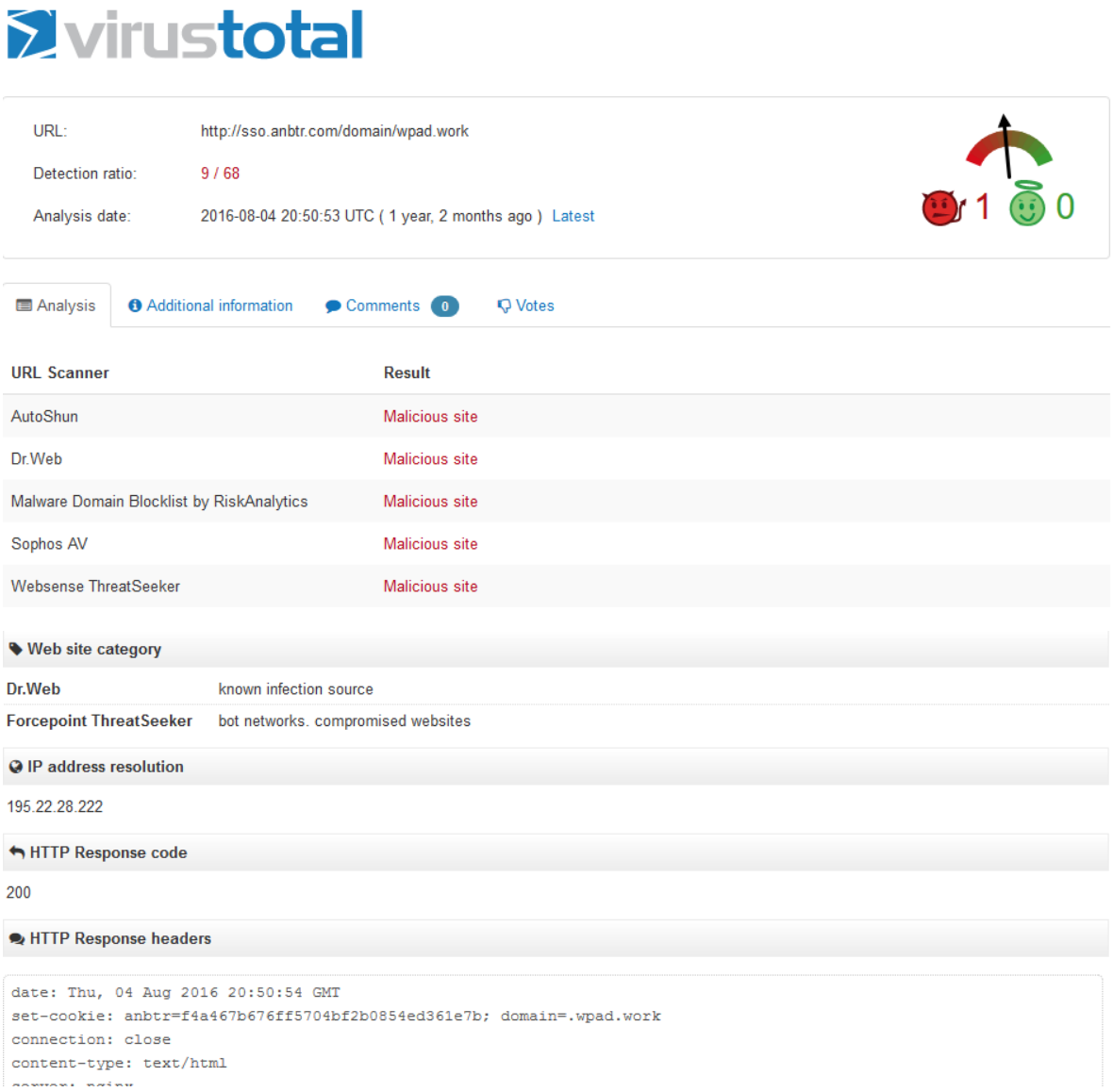
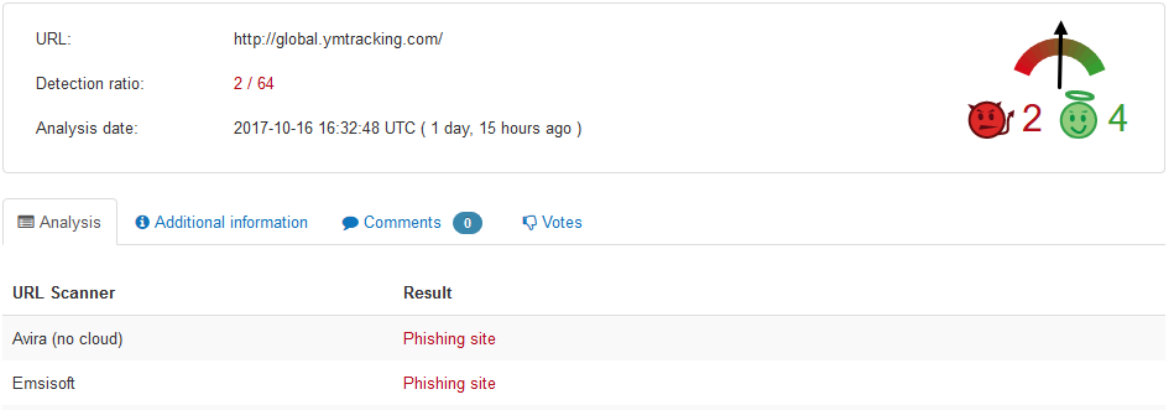


Ilustración 30: Resultado de dominio malicioso

Según los reportes esta botnet extrae información de usuarios conectados a internet y envía publicidad mediante ventanas emergentes del navegador y también es utilizada para saturar la red y equipos específicos.

De la misma manera se analizó el dominio http://global.ymtracking.com/, los reportes de este sitio maliciosos son de la técnica de hacking Phishing que también se pueden realizar mediante botnets.



Determinar el objetivo de una botnet, su tipo y su procedencia es complicado ya que con las técnicas actuales con las que se conectan en la red, los nombres de los dominios cambian constantemente dejando poca evidencia y documentación [10]. La información proporcionada por los logs del servidor DNS también es limitada y no se puede especificar con claridad el propósito de una botnet.

CAPÍTULO V
CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones:

- Los objetivos planteados en este proyecto se pudieron cumplir gracias a las herramientas utilizadas y la minería de datos que se pudo realizar de los registros de logs del servidor DNS.

- Gracias a la minería de datos y al machine learning se puede obtener un análisis global de todas las peticiones DNS con lo cual se pueden obtener resultados más exactos ya que se analizan todos los registros en su totalidad.

- Los resultados obtenidos considerados como positivos pueden ser exportados a listas negras en los servidores de dominio o proxies para prevenir ataques futuros en la red de la FISEI.

- Los resultados experimentales muestran que el proyecto de investigación obtiene una alta precisión de detección con un margen de error aceptable, considerando el

estudio de la red como base para extender el esquema en el futuro con mayores características y funcionalidades.

- Splunk facilita la extracción y el análisis de datos y logs generados por máquinas y gracias a sus plugins se pueden generar reportes y análisis complejos para facilitar la toma de decisiones.

- El método estudiado en este proyecto solamente es capaz de detectar botnets una vez que el ataque realizado por servidores C&C ya fue realizado ya que se basa en data generada en logs, por lo que que las medidas de protección son solamente correctivas y no preventivas.

- Con la información recolectada de los logs del servidor se puede detectar botnets, pero no se puede identificar con certeza el tipo de botnet, el origen ni los datos que se han enviado al servidor C&C lo cual significa una limitante en el proyecto realizado.

5.2 Recomendaciones:

- Se sugiere utilizar la investigación realizada con grandes cantidades de información ya que mientras más datos se obtengan los resultados tendrán mayor precisión.

- Para facilitar el ingreso de datos se recomienda mantener los logs en un solo archivo plano ya que el sistema autodetecta el crecimiento del mismo y se puede evitar el ingreso de nuevos conjuntos de datos.

- La herramienta splunk tienen varias características de análisis de datos generados por máquinas por lo que se recomienda utilizarla para la generación de informes y análisis de toda la red.

- Se recomienda realizar la detección de botnets en rangos de tiempo inferiores a un mes ya que cada día surgen nuevas amenazas y malware, se podría realizar un estudio para determinar el rango de tiempo.

Bibliografía o Referencias.

[1] Instituto nacional de estadística y censos "Encuesta Nacional de Ingresos y Gastos de hogares Urbanos y Rurales – ENIGHUR", http://www.ecuadorencifras.gob.ec/documentos/web-inec/Estadisticas_Sociales/TIC/2015/Presentacion_TIC_2015.pdf.

[2] M. Goncharov, «Russian Underground 101», Trend Micro Incorporated, https://www.trendmicro.de/cloud-content/us/pdfs/security-intelligence/white-papers/wp-russian-underground-101.pdf, 2012.

[3] Ruidong Chen, Weina Niu, Xiaosong Zhang, Zhongliu Zhuo y Fengmao Lv, "An Effective Conversation-Based Botnet Detection Method", Center for Cyber Security University of Electronic Science and Technology of China, Hindawi, China, ID 4934082, 2017.

[4] CHEN Jing, CHENG Xi, DU Ruiying, HU Li, WANG Chiheng, "BotGuard: Lightweight Real-Time Botnet Detection in Software Defined Networks", Wuhan University Journal of Natural Science, China, ID 1007-1202(2017)02-0103-11, Vol.22 No.2, 103-113, 2017.

[5] Pooja Sharma, Sanjeev Kumar, Neeraj Sharma, "BotMAD: Botnet Malicious Activity Detector Based on DNS Traffic Analysis", 2016 2nd International Conference on Next Generation Computing Technologies, (NGCT-2016), Dehradun, India 14-16, October 2016.

[6] ICANN, "Guía para principiantes nombres de dominio" https://www.icann.org/en/system/files/files/domain-names-beginners-guide-06dec10-es.pdf , 2015.

[7] José Pablo Hernández Valenciano, “Avances en la detección de botnets mediante el análisis de datos DNS pasivos”, http://bibing.us.es/proyectos/abreproy/90485/fichero/TFG_Jos%C3%A9PabloHern%C3%A1ndez.pdf , julio 2017.

[8] FRANCISCO DE BORJA ECHEVERRÍA SIERRALTA, “Implementación y Evaluación de Sistema de Monitoreo de Seguridad basado en flujos de paquetes IP” http://www.tesis.uchile.cl/tesis/uchile/2008/echeverria_fs/sources/echeverria_fs.pdf ,2008.

[9] Antonakakis y Perdisci ,“ From Throw Away Traffic to Bots: Detecting the Rise of DGA-Based Malware”, 2011.

[10] Evan Cooke, Farnam Jahanian, and Danny McPherson, “The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets, Proc. of Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '05)”, Boston, 2005.

[11] Bustos-Jimenez J - Saint-Pierre C. - Graves A, “Applying Process Mining Techniques to DNS Traces Analysis”, Chile 2014

[12] Splunk Enterprise, https://www.splunk.com/es_es/products/splunk-enterprise.html, julio 2017.

[13] “Botnets and cybercrimeIntroduction”- http://resources.infosecinstitute.com/botnets-and-cybercrime-introduction/ , Agosto 2017.

[14] Andrés González, “Qué es machine learning?”, http://cleverdata.io/que-es-machine-learning-big-data/ , agosto 2017.

[15] Maribel Tirados,"Algoritmo Random Forest", http://www.bigdatahispano.org/noticias/algoritmo-random-forest/, agosto 2017.

[16] Splunk Enterprise, "About the search language", https://docs.splunk.com/Documentation/SplunkCloud/6.6.1/Search/Aboutthesearchlanguage, Agosto 2017.

[17] Sistemas, C. I. D. E., Nelson, M., & Saltos, M. (2015). UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL.

[18] McCarty B. Botnets: Big and bigger [J]. IEEE Security & Privacy Magazine, 2003.

[19] Langner R. Stuxnet: Dissecting a cyberwarfare weapon [J]. IEEE Security & Privacy Magazine.

[20] Christiaan B, Carlos C, Cedric C, et al. Mcafee labs threat report [EB/OL]. http://www.mcafee.com/us/resources/reports/rp-threats-predictions-2016.pdf. Julio 2017

[21] Holguer Eduardo Chaso Salazar, "La Gestión de Seguridad Informática y su incidencia en la Información de la Universidad Técnica de Ambato", Universidad técnica de Ambato, Marzo 2017.

[22] David Fernando Sánchez Cunalata, "IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO Y PROTECCIÓN DE DATOS EN LA RED DE LA FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL.", Universidad Técnica de Ambato, Junio 2017.

ANEXOS Y APENDICES

Manual de instalación y uso de detección de botnets.

Instalacion de Splunk:

Descargar Splunk Enterprise desde la página oficial del producto:

https://www.splunk.com/es_es/products/splunk-enterprise.html

Seleccionar splunk-6.6.2-4b804538c686-linux-2.6-amd64.deb ya que se está utilizando la distribución debian o segun la distribucion donde se vaya .

Instalación de splunk en el sistema Operativo:

En la consola como “root” ejecutar el siguiente comando:

```
dpkg -i splunk-6.6.2-4b804538c686-linux-2.6-amd64.deb
```

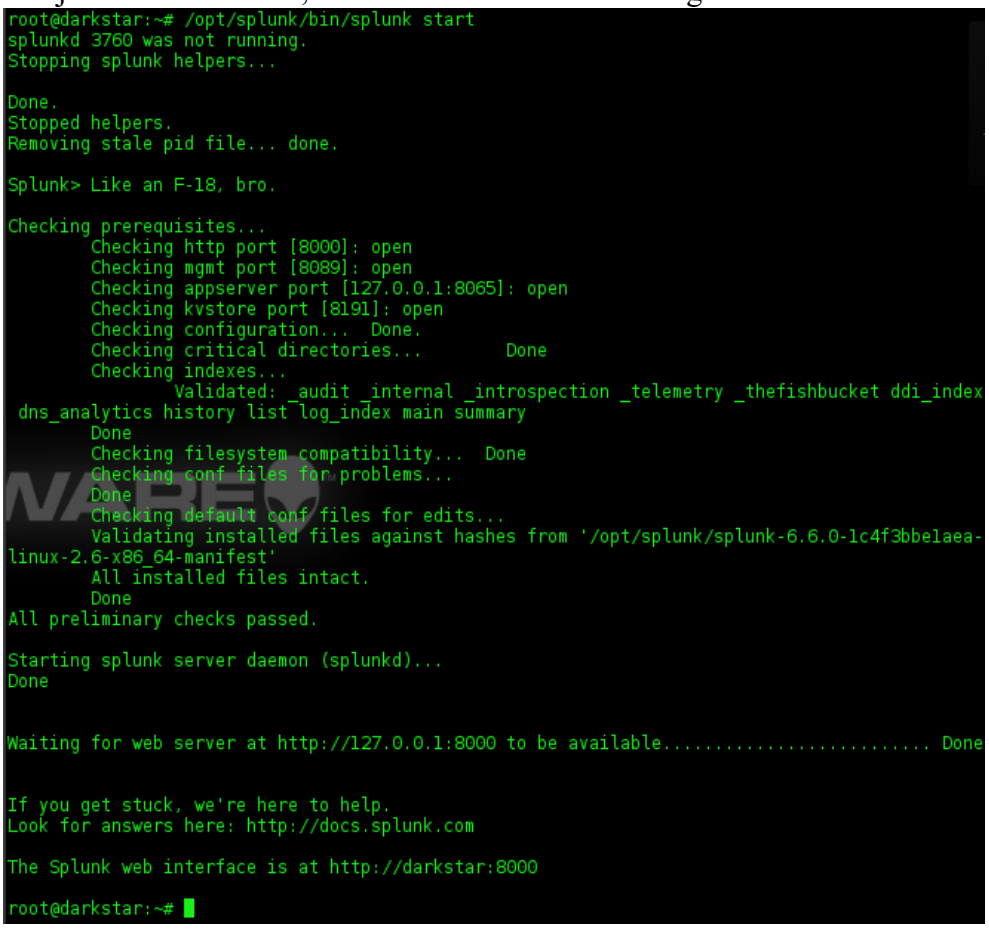
Es necesario que el sistema cumpla con las dependencias que el software requiere.

Iniciar splunk:

Ejecutar el siguiente comando:

```
/opt/splunk/bin/splunk start
```

Al ejecutar el comando, el servicio se iniciará de la siguiente manera:



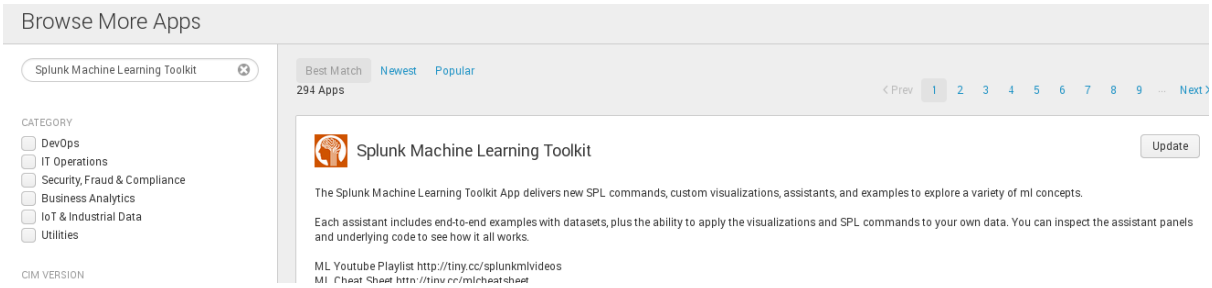
La información nos indica que la interfaz web de splunk es la siguiente: http://darkstar:8000

Instalación de Plugins adicionales necesarios para el proyecto:

Splunk permite la instalación de plugins adicionales desde su misma interfaz web.

Instalación de Splunk Machine Learning Toolkit:

Para el funcionamiento de esta app, es necesario instalar Python for Scientific Computing dentro de splunk.



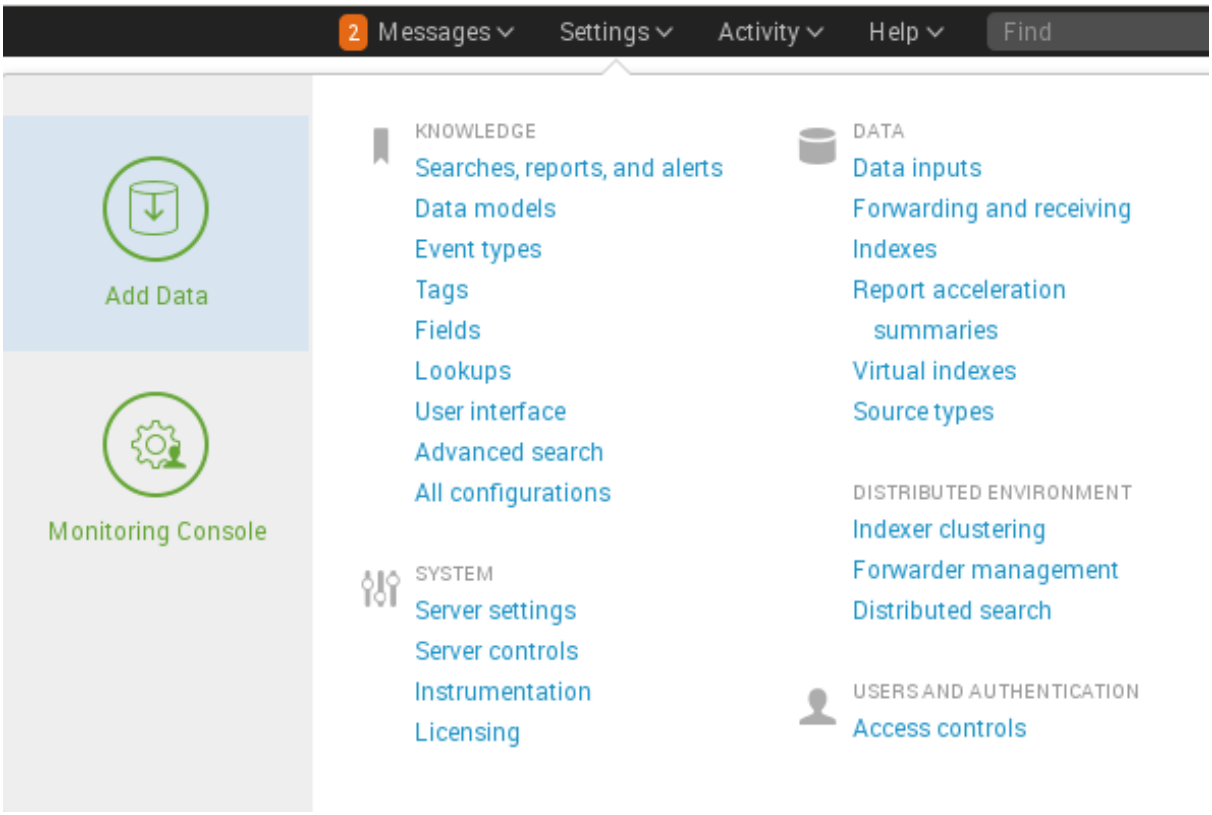
De la misma manera instalar los siguientes Plugins:

- **Instalación de Splunk Add-on for ISC BIND**
- **Instalación de DNS Analytics for Splunk**
- **Instalación de URL Toolbox**

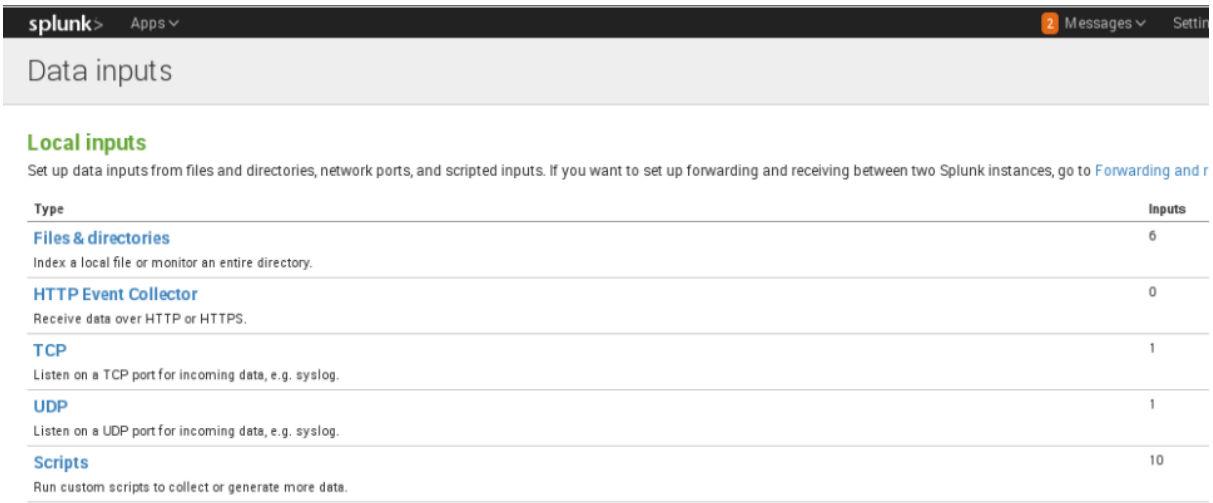
Manual de Usuario Para la detección de botnets:

Creación de los *datasources.*

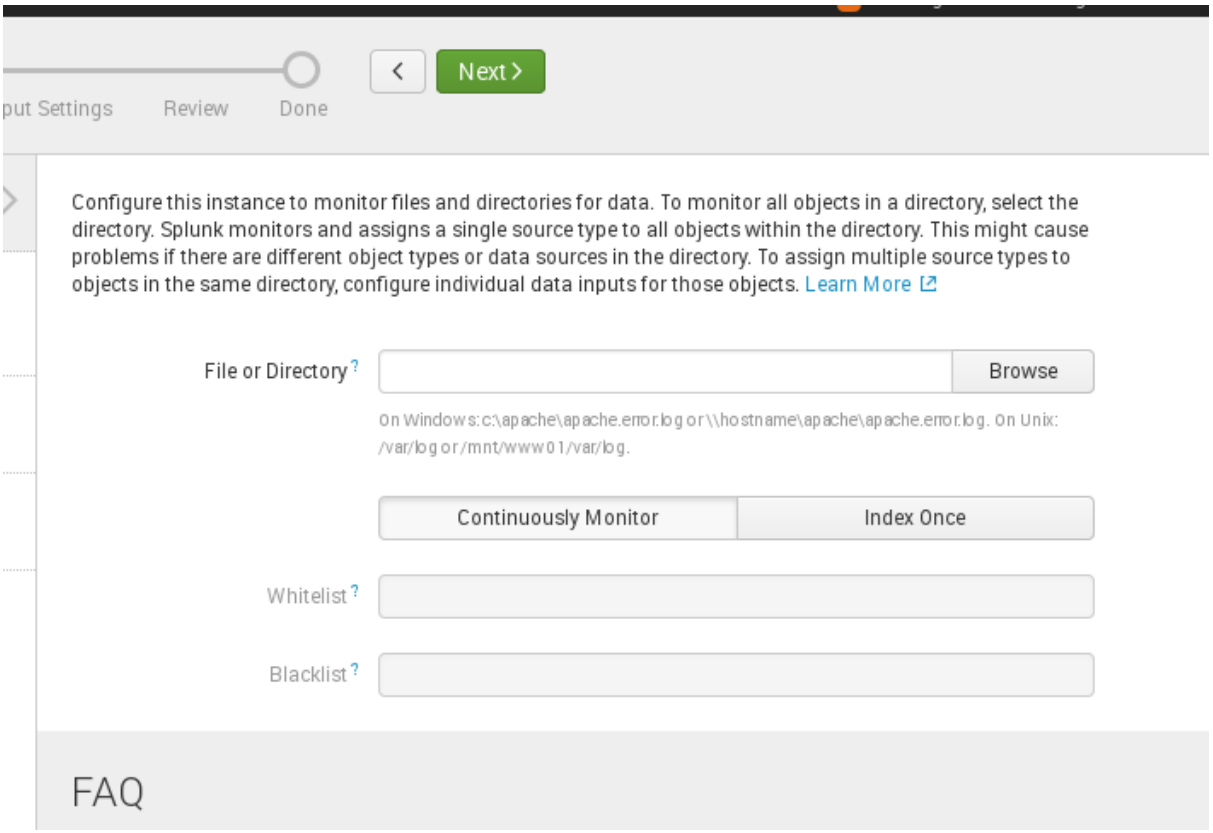
En el menú de splunk en “settings” seleccionar “data input”:



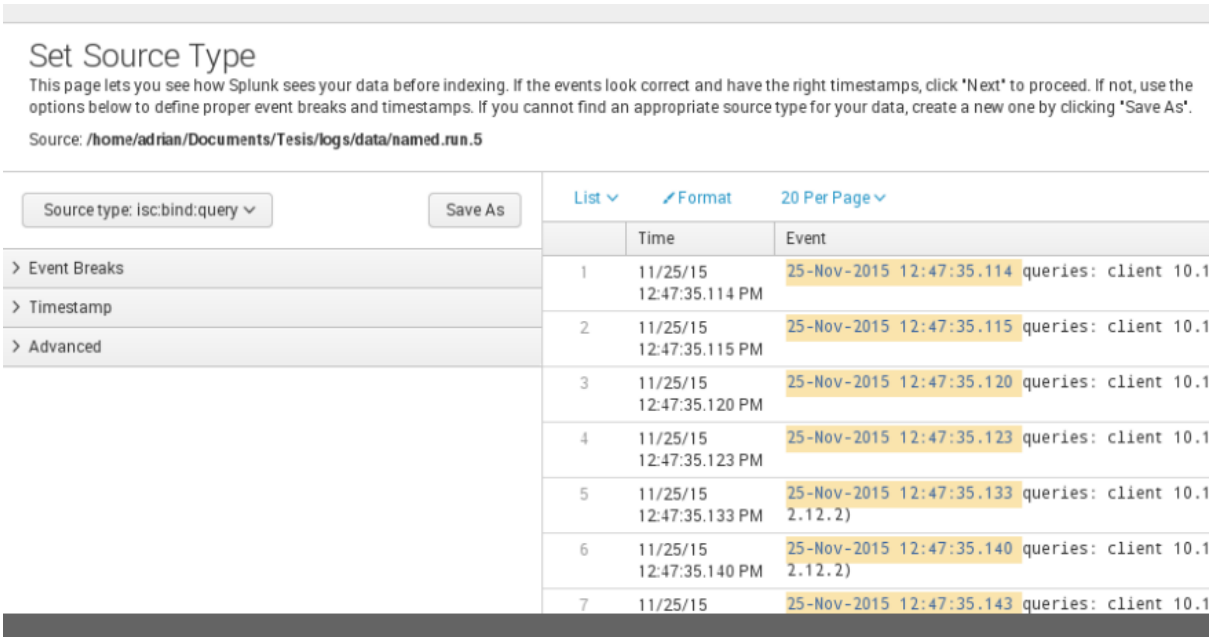
A continuación se debe seleccionar el tipo de data input “Files & Directories”



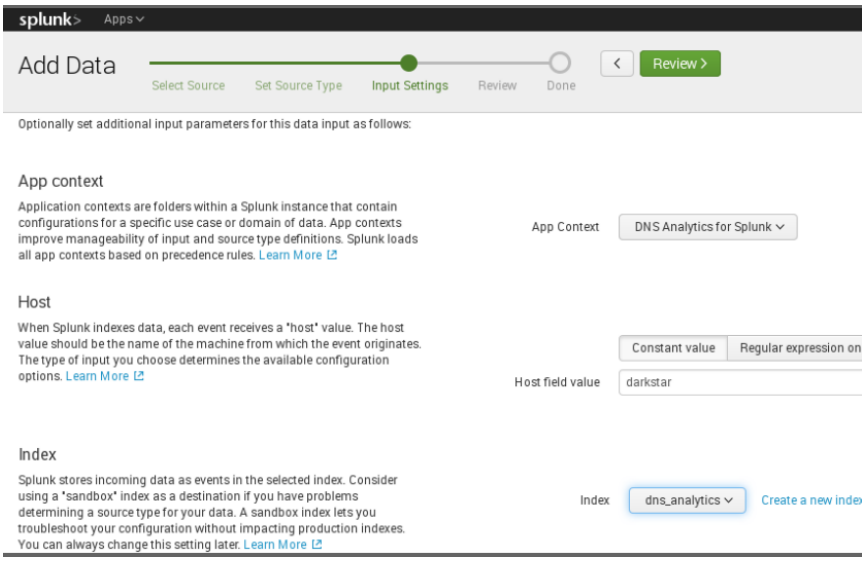
El siguiente paso es seleccionar uno de los archivos de logs que va a analizar la herramienta splunk



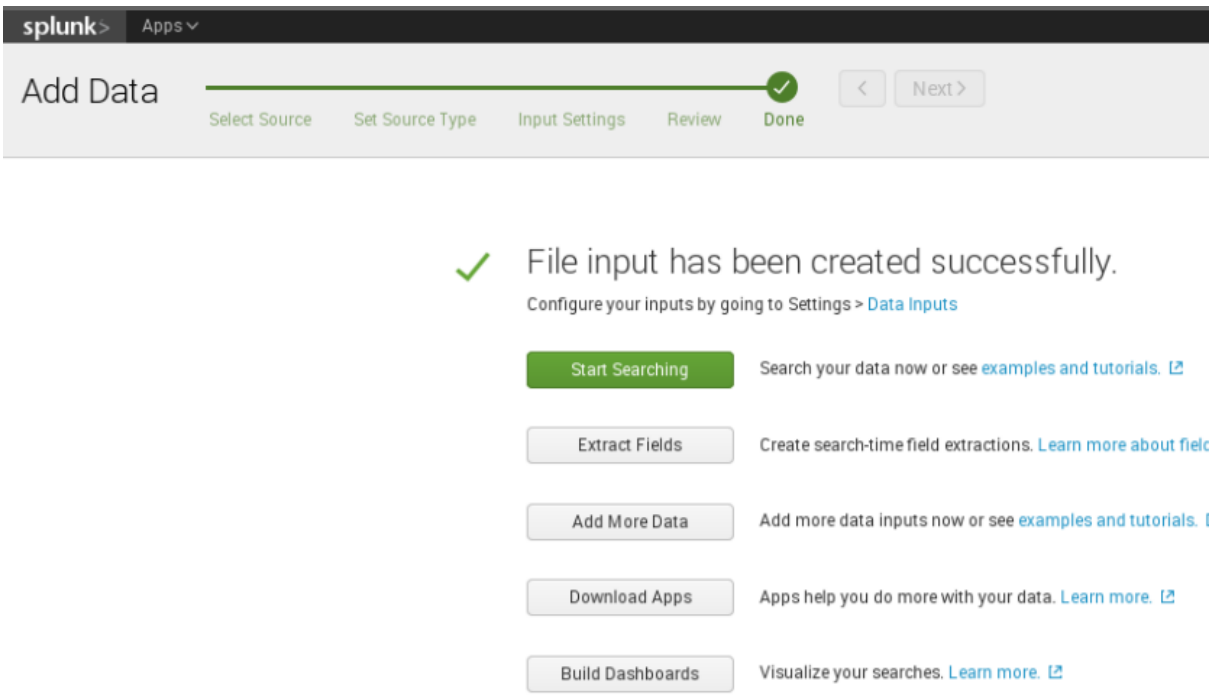
Es necesario seleccionar el tipo de dato para poder analizar y realizar la minería de datos. En el caso de este trabajo es “isc:bind:query” ya que los archivos son *logs* de *queries* de *bind*.



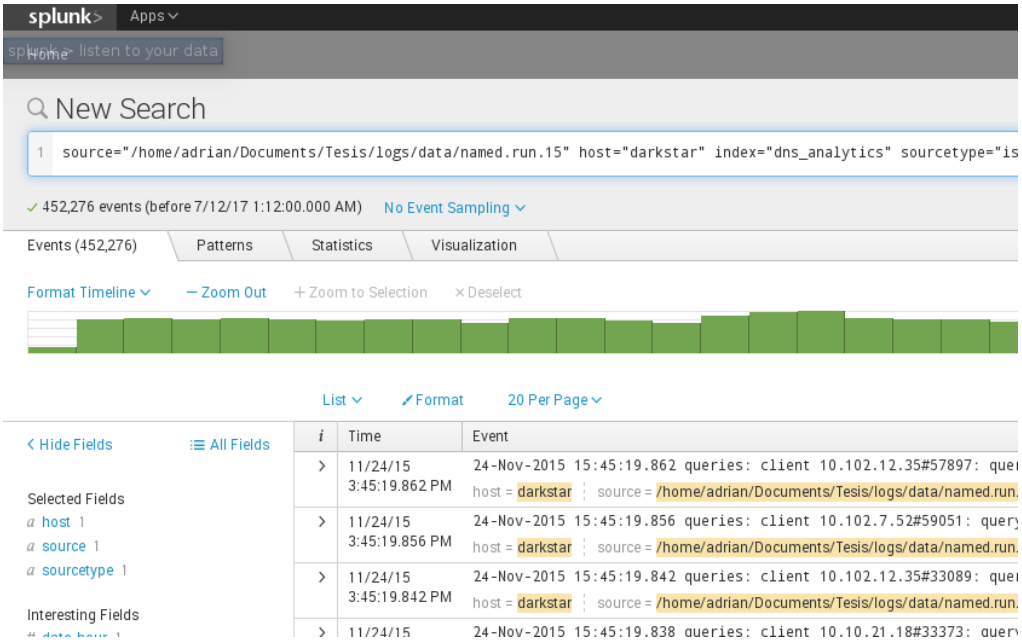
Se deben definir las siguientes propiedades y seleccionar un índice en común para todos los *data inputs*, con el fin de simplificar las búsquedas en una sola consulta.



Si todos los pasos son satisfactorios splunk informa con una pantalla como la siguiente:

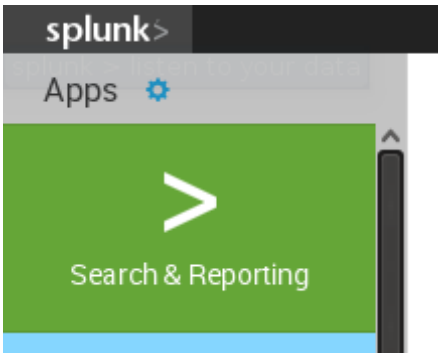


Para que los índices se actualicen al instante y se puedan consultar los nuevos data inputs es necesario presionar el botón “Start Searching” este proceso toma alrededor de 45 segundos en la máquina descrita al inicio en finalizar la actualización.

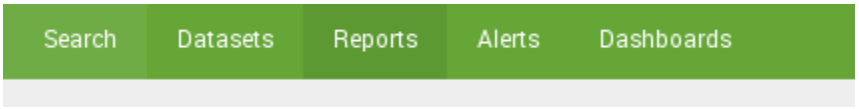


Realizar Consultas para obtener Botnets:

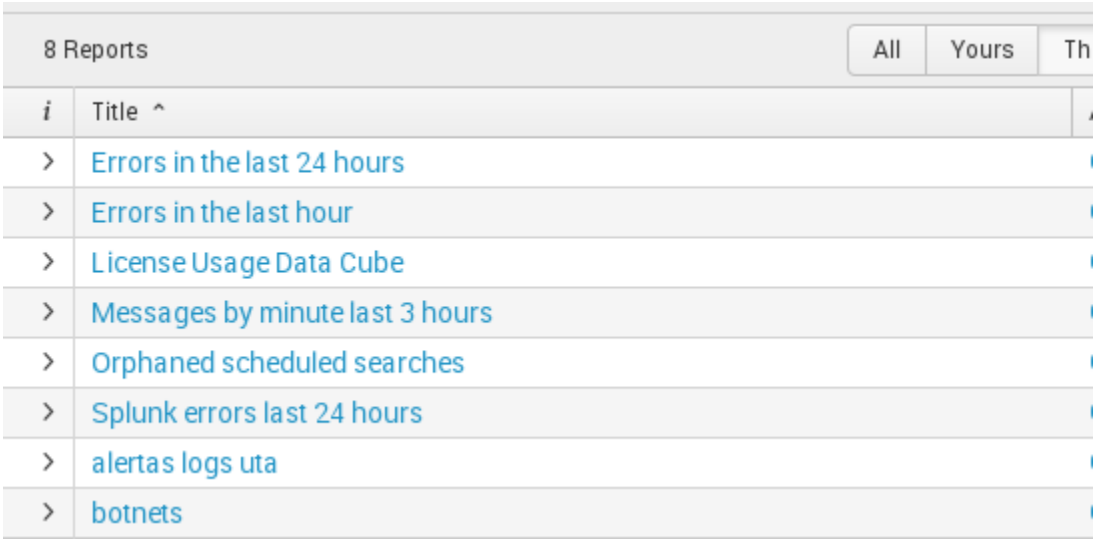
Ubicarse en la sección de “search & Reporting de splunk”



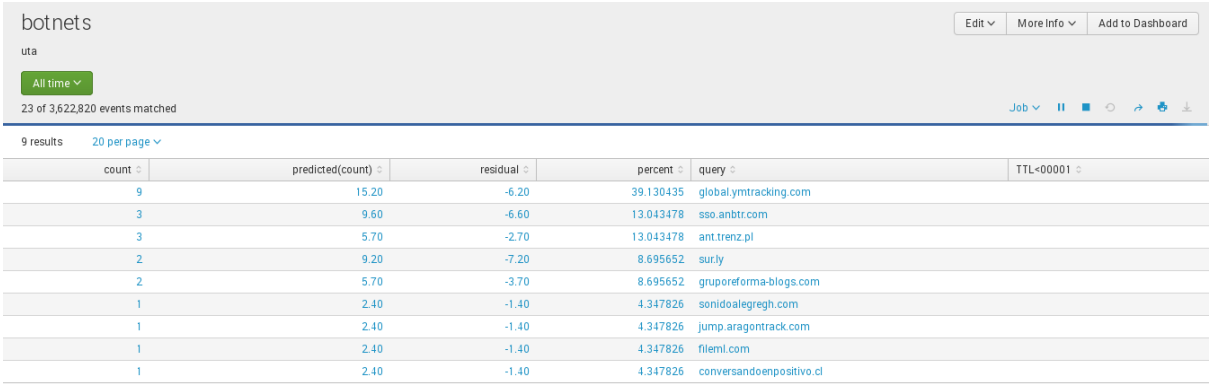
Dirigirse a la sección de “Reports”



Ejecutar el reporte llamado “botnets”



El Reporte devolverá la información requerida de la siguiente manera:


botnets

uta

All time

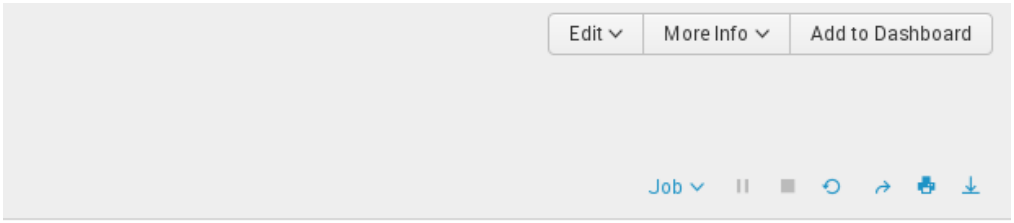
23 of 3,622,820 events matched

Edit | More Info | Add to Dashboard | Job

9 results | 20 per page

count	predicted(count)	residual	percent	query	TTL<00001
9	15.20	-6.20	39.130435	global.ymtracking.com	
3	9.60	-6.60	13.043478	sso.anbtr.com	
3	5.70	-2.70	13.043478	ant.trenz.pl	
2	9.20	-7.20	8.695652	sur.ly	
2	5.70	-3.70	8.695652	gruporeforma-blogs.com	
1	2.40	-1.40	4.347826	sonidoalegregh.com	
1	2.40	-1.40	4.347826	jump.aragontrack.com	
1	2.40	-1.40	4.347826	fileml.com	
1	2.40	-1.40	4.347826	conversandoenpositivo.cl	

Guardar en un archivo CVS o Exportar el reporte:




ent	query	TTL<00001
2381	sso.anbtr.com	
0476	global.ymtracking.com	

Con el Boton  se pueden exportar los resultados obtenidos a varios formatos: