CYBERINTELLIGENCE FOR IOT

DAVID ESTEBAN USECHE PELAEZ

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO
INGENIERÍA DE SISTEMAS
CIBERSEGURIDAD E INFRAESTRUCTURA
BOGOTÁ D.C.
2018

CYBERINTELLIGENCE FOR IOT

DAVID ESTEBAN USECHE PELAEZ

PROYECTO DE GRADO

DANIEL ORLANDO DÍAZ LÓPEZ, DOCTOR EN INFORMÁTICA

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO
INGENIERÍA DE SISTEMAS
CIBERSEGURIDAD E INFRAESTRUCTURA
BOGOTÁ D.C.
2018

Nota de aceptación

_____

_____

_____

_____

Director del Trabajo de Grado

_____

Documento de identidad N.°:

CONTENIDO

FIGURAS

Pág

TABLAS

Pág

RESUMEN

Mediante el presente proyecto se busca analizar, evaluar e integrar diversas herramientas y entornos de ciber inteligencia, enfocando este análisis a la defensa de activos y la identificación de adversarios, junto con la posterior compartición de información de inteligencia.

Un entorno de inteligencia evaluado es el formado por Logstash, Elasticsearch y Kibana (ELK), por otro lado, se analizaró la herramienta de gestión de información de amenazas MISP (Malware Information Sharing Platform) y la herramienta de análisis de redes de sistemas Malcom.

Las herramientas de inteligencia que se usaron para estudiar las distintas técnicas de anális de malware fueron: Radare2, Androguard, InmunityDBG, OllyDBG y Yara Rules. Entre estas herramientas se escogieron Radare2, Androguard y Yara rules por su potencial de automatización. Para integrar estas herramientas se usó r2Yara y Androguard para Yara.

También se estudiaron distintas arquitecturas y posibles integraciones con un centinela IoT de las herramientas usadas y desarrolladas.

Por último, se construyó un centinela para proteger a los dispositivos IoT frente a ataques de malware, este centinela integra las herramientas antes mencionadas junto con diversos APIs de análisis externo.

Este centinela funciona monitoreando la red y enviando a evaluación los archivos sospechosos. La evaluación se compone de 3 anillos de seguridad, uno de machine learning que categoriza aplicaciones como goodware o malware, uno que utiliza reglas de Yara y uno de análisis externo. Si algún anillo detecta una muestra como malware se crea un reporte en MISP y se comparte con los centinelas conectados a la misma comunidad MISP, para posteriormente generar nuevas reglas.

INTRODUCCIÓN

El auge de los dispositivos IoT, junto con la escasa atención que se les ha brindado en cuestión de seguridad ha hecho de este un campo con un alto potencial de ataque desde el punto de vista del malware, esto con recompensas para el atacante que pueden ir desde espionaje hasta el secuestro del equipo y toda su información.

De esto surge que la gestión de la seguridad de la información y la protección de activos de información (En este caso los dispositivos IoT) merecen una preocupación constante por parte de organizaciones que gestionan información de alta criticidad. Los adversarios cada vez encuentran formas más sofisticadas de lanzar ataques sobre objetivos de alto valor.

Sin la caracterización y perfilado adecuado del adversario y su modo de operación es casi imposible para las agencias de la ley actuar para la detección, procesamiento y prevención de ciber crímenes. Por medio del presente proyecto se busca implementar una solución de ciberseguridad que protege ecosistemas IoT utilizando técnicas forenses, correlación de eventos y aprendizaje de máquina.

ESTADO DEL ARTE

Para proteger a los dispositivos de Internet de las cosas (IoT) es necesario primero entender qué engloba este concepto y por qué se ha vuelto tan importante en los últimos años. Además de las amenazas que se han gestado contra esta nueva tecnología.

Un dispositivo IoT puede definirse como un elemento cotidiano que se conecta a internet y pertenece a una infraestructura de red dinámica que se autoconfigura y en la cual todos sus nodos se comunican entre sí[1].

El que estos dispositivos sean cotidianos ha causado que incluso desde 2015 hasta 2018 no solo sean una tendencia tecnológica, son una tendencia que avanza hacia dispositivos inteligentes capaces de realizar algunas tareas sin supervisión, interactuando y colaborando en la infraestructura montada alrededor de ellos[2]–[5].

Pero esta tendencia no se refleja solamente en la investigación sobre los dispositivos IoT, [6] sostiene que el mercado de IoT desde 2013 ha estado creciendo con un CAGR[1] del 17.5%, pasando de 9.1 billones de dispositivos IoT en 2013 a una proyección de 28.1 billones de dispositivos activos en 2020.

Visto desde el punto de vista de las ventas, en 2013 la recaudación fue de 1.9 trillones de dólares y se espera que en 2020 sea de 7.1 trillones de dólares[6]. Estos datos no solo dan a entender que el IoT es una tendencia a nivel de estudio e investigación, en el ámbito comercial están adquiriendo cada vez más relevancia en las empresas y en la vida diaria.

Un crecimiento tan grande de este mercado también implica un crecimiento en los ataques que los dispositivos IoT sufren. En caso de los IoT el número de ataques se ha incrementado en un 600% de 2016 a 2017, siendo Telnet[2] el servicio más atacado y los router el blanco más popular.[7]

En el caso del malware diseñado específicamente para los dispositivos móviles este crecimiento también es notable, puesto que en 2017 se detectaron 26 579 nuevas variantes de malware de todo tipo, en comparación con el 2016, donde se detectaron 17 214 nuevas variantes[7].

En la resolución de esta problemática varias arquitecturas de seguridad han surgido, una de estas arquitecturas es propuesta por Microsoft en [8]. Esta arquitectura se centra en el aislamiento y protección de tanto dispositivos como posibles usuarios finales, se divide en 4 zonas: Dispositivos, Gateway de zona, Gateway de la nube y servicios.

[1] https://support.office.com/es-es/article/calcular-la-tasa-anual-compuesta-de-crecimiento-cagr-3ccb7cd3-39b3-49ee-8b38-c19972607dfa

[2] https://es.ccm.net/contents/708-uso-de-telnet

En esta arquitectura los dispositivos IoT se encuentran en la zona de dispositivos, aislados de la internet pública, el Gateway de zona comunica a la zona de Dispositivos con el exterior, siendo bastante susceptible a ataques en caso de querer interferir con la zona interna.

La tercera zona es el Gateway de nube, esta zona comunica los distintos dispositivos y Gateways de zona por la red pública. Finalmente se tienen los servicios, estos son componentes que interactúan con las distintas zonas recolectando y analizando información.

Otra arquitectura para IoT se basan en certificados y confianza que se otorga al código que desea correr en el dispositivo, junto con su correspondiente manejo de autenticación en la nube, un punto interesante y a favor de esta arquitectura es que tambien provee autenticación y protección en las comunicaciones, todo basado en certificados y firmas digitales[9].

Finalmente, esta arquitectura propone construir una perspectiva holística de seguridad para un modelo ubicuo - unidad de IoT (U2IoT). Esto se propone desde tres perspectivas: información, física y de gestión. Esta propuesta se apoya en la inteligencia de la entidad, que permite a U2IoT adaptarse a los escenarios de amenaza y definir contramedidas específicas a nivel de información o capa física de acuerdo a lo que se ve en la red[10].

Para la arquitectura propuesta que pretende dar una solución a la problemática de los ataques a dispositivos IoT se utilizó ciberinteligencia. La ciberinteligencia puede definirse como la extracción, análisis y uso en soporte de distintas operaciones de la información que puede extraerse de espacios digitales[11].

En este caso se extraerá información de la red y los dispositivos, el análisis se realizará de forma automática y la operación a soportar es una de protección de dispositivos.

Se evaluaron distintas herramientas, cada una aportando información distinta de inteligencia. Las herramientas evaluadas fueron:

Logstash: Logstash es una herramienta open source que provee un framework para recolección, centralización, parseo, almacenamiento y búsqueda de logs, aunque no limitado a este tipo de información. Permite recibir información de distintas fuentes y definir el modo de salida de esta información una vez ha sido procesada. Además, es altamente escalable y extensible por medio de plug-ins[12].

Elasticsearch: Elasticsearch es un motor de analíticas y búsqueda distribuida open source, permite indexar (Organizar) la información. En Elasticsearch cada campo está indexado y puede buscarse sobre el en tiempo real, además se provee un API REST y un API de búsqueda que implementa su propio lenguaje[13].

Kibana: Kibana es una herramienta que provee una interfaz para usar servicios de búsqueda sobre información indexada, principalmente usando las distintas API que

provee Elasticsearch. Kibana permite explorar los datos, generar informes ejecutivos y gráficas a partir de ellos[14].

Suricata: Suricata es un motor de detección de amenazas a la red gratuito y de código abierto. El motor de Suricata es capaz de detectar intrusiones en tiempo real (IDS), prevenir intrusiones en la red (IPS), monitorear la seguridad de la red (NSM) y procesar pcaps fuera de línea[15].

Kismet: Kismet es un IDS que funciona sobre la capa 1 y capa 2 del estándar IEEE 802.11, funciona en tarjetas que permiten activar el modo monitor[15]. Tiene interfaz de consola e interfaz interactiva, en la cual pueden leerse los paquetes capturados.

OpenVas: OpenVas es un framework de gestión de dispositivos sobre una red, permite obtener información sobre los dispositivos examinados y un análisis de vulnerabilidades. Cuenta con una interfaz web para administración y la información generada puede ser exportada para análisis[15].

Las herramientas Suricata, Kismet, OpenVas y un OSSIM[3] fueron evaluadas e integradas en [15].

Malcom: Malcom es una herramienta diseñada para analizar la comunicación de red de un sistema usando representaciones gráficas del tráfico de red, y hacer referencias cruzadas con fuentes de malware conocidas. Esto resulta útil cuando se analiza cómo ciertas especies de malware intentan comunicarse con el mundo exterior.[4]

El objetivo de Malcom es agilizar el análisis de malware y la recopilación de información proporcionando una versión legible para el ser humano del tráfico de la red originado por un determinado host o red. Convierta la información de tráfico de red a inteligencia procesable más rápidamente.

MISP: MISP es una plataforma que permite la recolección y compartición de información referente a indicadores de compromiso (IoC)[5] de ataques o vulnerabilidades, los IoC se dividen en categorías y a su vez, las categorías se dividen en tipos[16].

Para almacenar la información MISP utiliza un objeto llamado **Evento**, un evento, aparte de la información de identificación, cuenta con **Atributos**, donde cada atributo contiene un elemento con mínimo una categoría, un tipo, un valor y un nivel de privacidad[16].

MISP ofrece dos formas para compartir o recibir información, la primera de ellas es realizar una sincronización entre instancias de MISP y la segunda es añadir un

[3] https://www.alienvault.com/products/ossim

[4] https://github.com/tomchop/malcom

[5] https://www.trendmicro.com/vinfo/us/security/definition/indicators-of-compromise

Feed, este feed es una fuente confiable de eventos e información de la cual la instancia local puede descargar nuevos eventos.[16]

Radare2 (r2): r2 es una reescritura desde cero de radare para proporcionar un conjunto de librerías y herramientas para trabajar con archivos binarios. El proyecto de Radare comenzó como una herramienta forense, un editor hexadecimal de línea de comandos capaz de abrir archivos de disco, pero con soporte posterior para analizar binarios, desensamblar código, depurar programas, adjuntar a servidores gdb remotos entre otros[17].

Ollydbg: Ollydbg es un debbugger a nivel de asembler en 32 bits hecho para Windows, se basa en simplicidad, permitiendo análisis de código - rastrea registros, reconoce procedimientos, bucles, llamadas a APIs, switches, tablas, constantes y cadenas. Además, al ser de código abierto permite la inclusión de muchas librerias externas para la personalización de la herramienta según las necesidades del analista[18].

Androguard: Androguard es un framework para hacer análisis estático de malware diseñado para dispositivos Android, está escrito en Python y tiene soporte para archivos DEX, ODEX, APK y binarios XML. Androguard usa una terminal de análisis desde la cual se puede invocar a los módulos aislados de esta herramienta para un análisis detallado[17].

Una de las funciones más interesantes de esta herramienta es Androrisk, que evalúa el archivo sospechoso en busca de elementos tales como ciertos permisos, librerías y factores de riesgo varios que puedan convertir a la aplicación en un peligro[17].

Yara rules: Yara es una herramienta diseñada (Pero no limitada) a hacer análisis de malware identificando, clasificando y evaluando por patrones las distintas muestras, Yara permite describir a muchas familias de malware por medio de la identificación y aplicación de estos patrones encontrados[19].

En la instalación de Yara rules se incluyeron 3 módulos que ofrecen distintos servicios y potencian el poder de detección de Yara, estos módulos son:

- Cuckoo[6]: Este módulo permite usar un reporte de Cuckoo sandbox[7] para construir reglas basadas en el comportamiento del archivo en un entorno simulado. Este módulo puede ser bastante útil en caso que la regla de Yara evalue un malware encriptado y los strings no sean visibles.
- R2Yara[8]: Este módulo permite usar las funciones de Radare2 para generar reglas acerca de lo que el archivo es, hashes, librerias usadas, exports, la información que Radare2 genera puede ser usada en la construcción de reglas.

[6] http://yara.readthedocs.io/en/v3.5.0/modules/cuckoo.html
[7] https://cuckoo.sh/docs/usage/index.html
[8] http://r2yara.readthedocs.io/en/latest/

- Androguard Yara[9]: Este módulo permite usar varias de las funcionalidades de Androguard para construir reglas de Yara. Especialmente util cuando se quieren generar reglas para detectar malware movil.

[9] https://docs.koodous.com/yara/androguard/

REVISIÓN DE SOLUCIONES PARA ANÁLISIS DE MALWARE

En esta sección se revisó un conjunto de herramientas de ciberseguridad que contienen características deseables en un centinela IoT al ayudar en su tarea de protección de dispositivos o hacer que este sea adaptable y colaborativo. Por sus funciones, las herramientas evaluadas pueden ser integradas en distintas soluciones de ciberinteligencia.

Entorno Logstash – Elasticsearch – Kibana (ELK)

Logstash, Elasticsearch y Kibana fueron desarrolladas por la misma empresa y tienen una afinidad e integración naturales, juntas forman el entorno ELK. Este entorno permite vigilar y analizar los datos desde que son recolectados hasta que son procesados en informes, todo esto de forma personalizable.

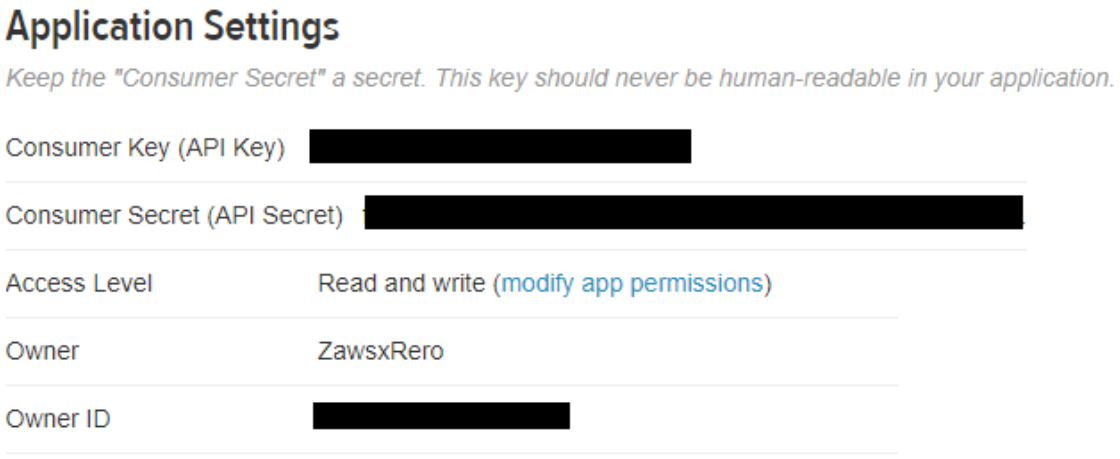
En la demostración de uso se recolectan datos de Twitter sobre tweets relacionados con malware como parte de un ejercicio de perfilamiento de adversarios. Esta recolección se hizo usando un plug in de Logstash, para indexarlos usando Elasticsearch y finalmente procesarlos con Kibana.

Instalación

En esta sección se instalarán las herramientas pertenecientes al entorno ELK en un entorno Ubuntu 16.04, usando la versión 6.2.4 en las 3 herramientas y con Java 1.8.

1. Obtener una clave de Twitter para API en https://apps.twitter.com/

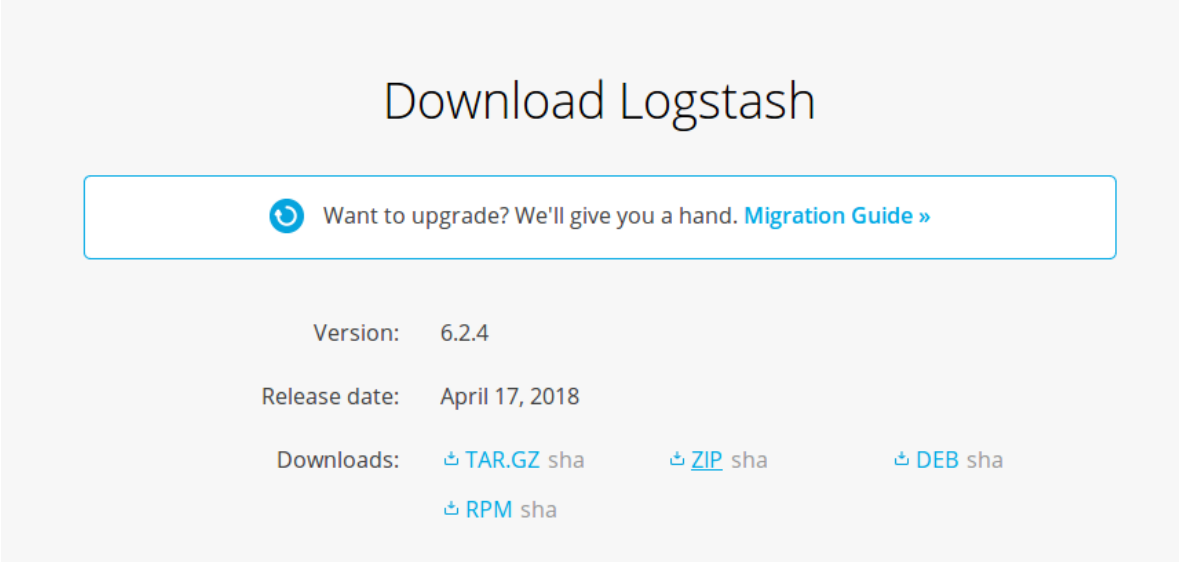
Figura 1. Aplicación en Twitter



Fuente: Propia

2. Descargar el .zip de Logstah de https://www.elastic.co/products/logstash

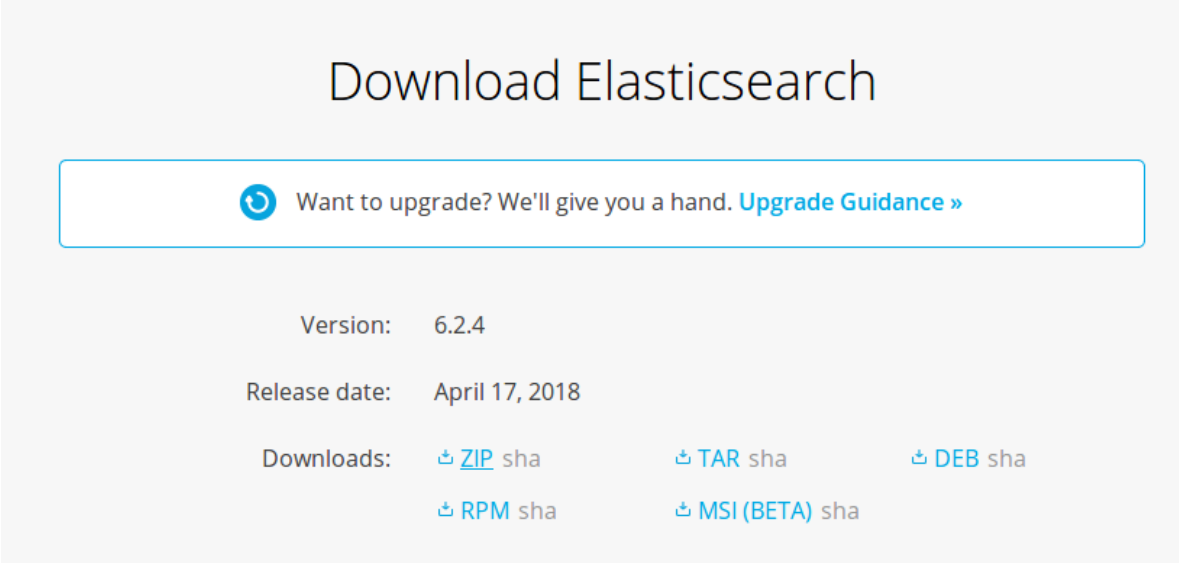
Figura 2. Descarga de Logstash



Fuente: Elasticsearch

3. Descargar él .zip de Elasticsearch de https://www.elastic.co/downloads/elasticsearch#ga-release

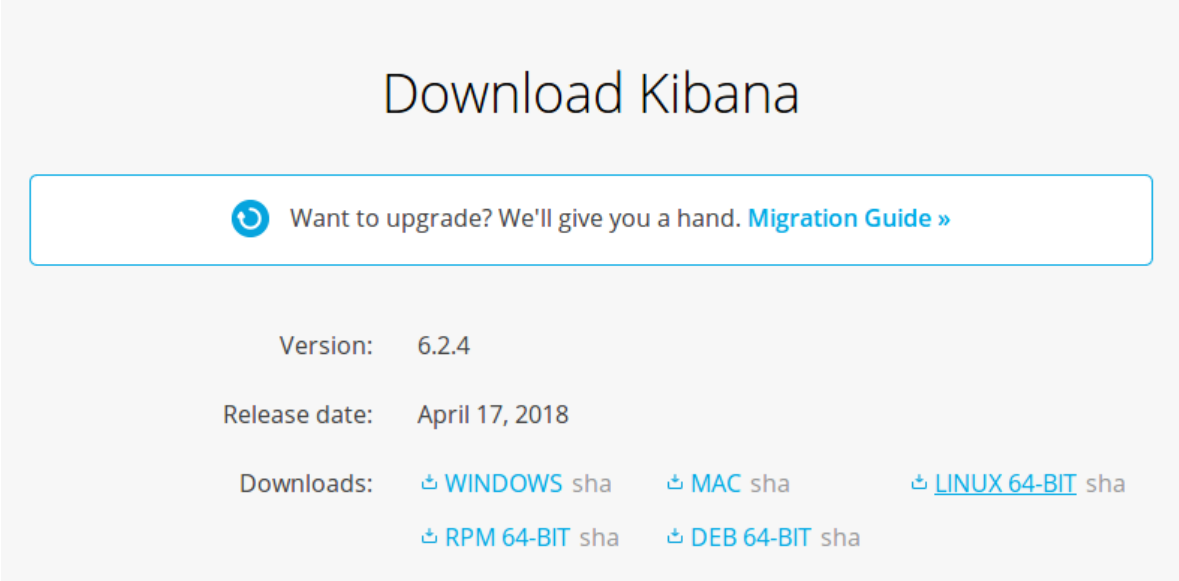
Figura 3. Descarga de Elasticsearch



Fuente: Elasticsearch

4. Descargar el .tar.gz de Kibana de https://www.elastic.co/products/kibana

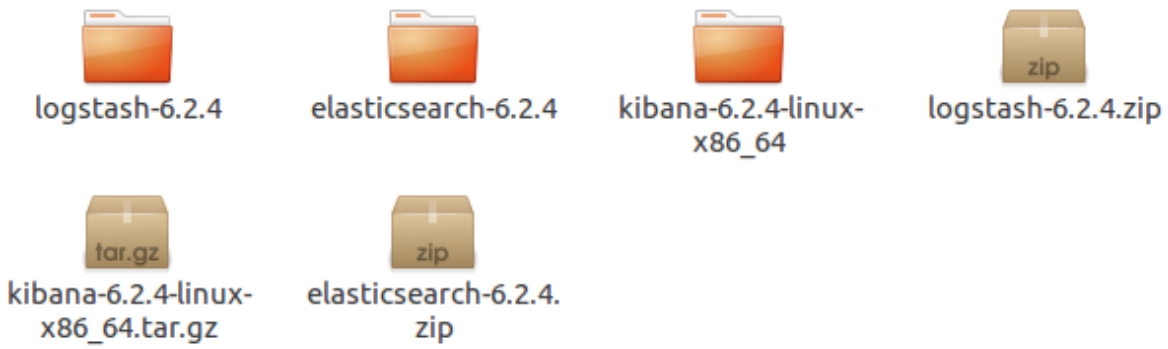
Figura 4. Descarga de Kibana



Fuente: Elasticsearch

5. Descomprimir los 3 archivos

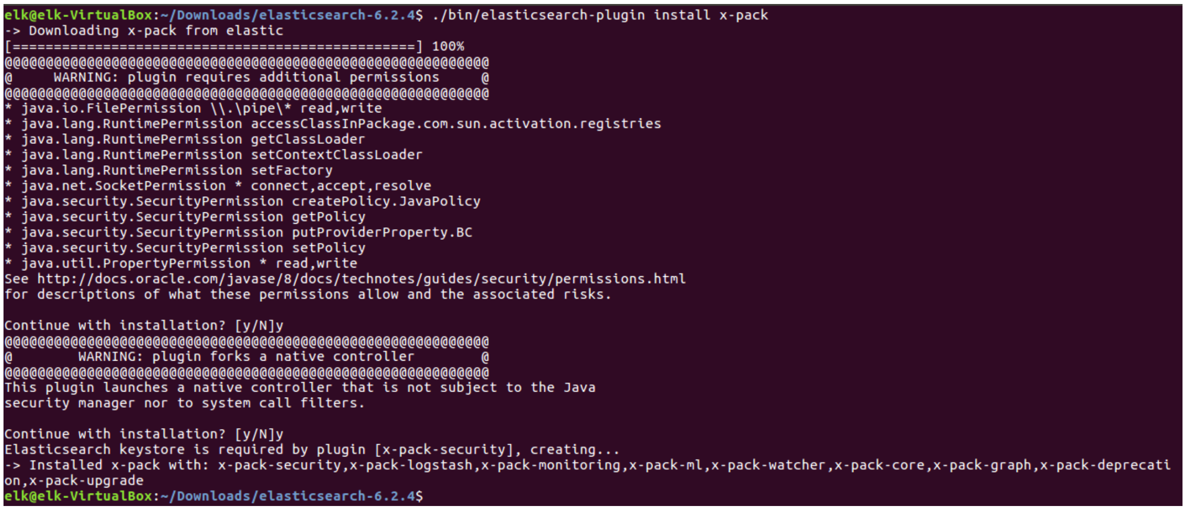
Figura 5. Herramientas descomprimidas



Fuente: Propia

6. Instalar x-pack en Elasticsearch

Figura 6. Instalación de x-pack - Terminal



```
elk@elk-VirtualBox:~/Downloads/elasticsearch-6.2.4$ ./bin/elasticsearch-plugin install x-pack
-> Downloading x-pack from elastic
[=================================================] 100%
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@     WARNING: plugin requires additional permissions     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
* java.io.FilePermission \\.\pipe\* read,write
* java.lang.RuntimePermission accessClassInPackage.com.sun.activation.registries
* java.lang.RuntimePermission getClassLoader
* java.lang.RuntimePermission setContextClassLoader
* java.lang.RuntimePermission setFactory
* java.net.SocketPermission * connect,accept,resolve
* java.security.SecurityPermission createPolicy.JavaPolicy
* java.security.SecurityPermission getPolicy
* java.security.SecurityPermission putProviderProperty.BC
* java.security.SecurityPermission setPolicy
* java.util.PropertyPermission * read,write
See http://docs.oracle.com/javase/8/docs/technotes/guides/security/permissions.html
for descriptions of what these permissions allow and the associated risks.

Continue with installation? [y/N]y
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@        WARNING: plugin forks a native controller        @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
This plugin launches a native controller that is not subject to the Java
security manager nor to system call filters.

Continue with installation? [y/N]y
Elasticsearch keystore is required by plugin [x-pack-security], creating...
-> Installed x-pack with: x-pack-security,x-pack-logstash,x-pack-monitoring,x-pack-ml,x-pack-watcher,x-pack-core,x-pack-graph,x-pack-deprecati
on,x-pack-upgrade
elk@elk-VirtualBox:~/Downloads/elasticsearch-6.2.4$
```

Fuente: Propia

7. Activar x-pack en Kibana

Figura 7. Activación de x-pack en Kibana - Terminal

```
elk@elk-VirtualBox:~/Downloads/kibana-6.2.4-linux-x86_64$ ./bin/kibana-plugin install x-pack
Attempting to transfer from x-pack
Attempting to transfer from https://artifacts.elastic.co/downloads/kibana-plugins/x-pack/x-pack-6.2.4.zip
Transferring 264988487 bytes....................
Transfer complete
Retrieving metadata from plugin archive
Extracting plugin archive
Extraction complete
Optimizing and caching browser bundles...
Plugin installation complete
elk@elk-VirtualBox:~/Downloads/kibana-6.2.4-linux-x86_64$
```

Fuente: Propia

8. Iniciar Elasticsearch

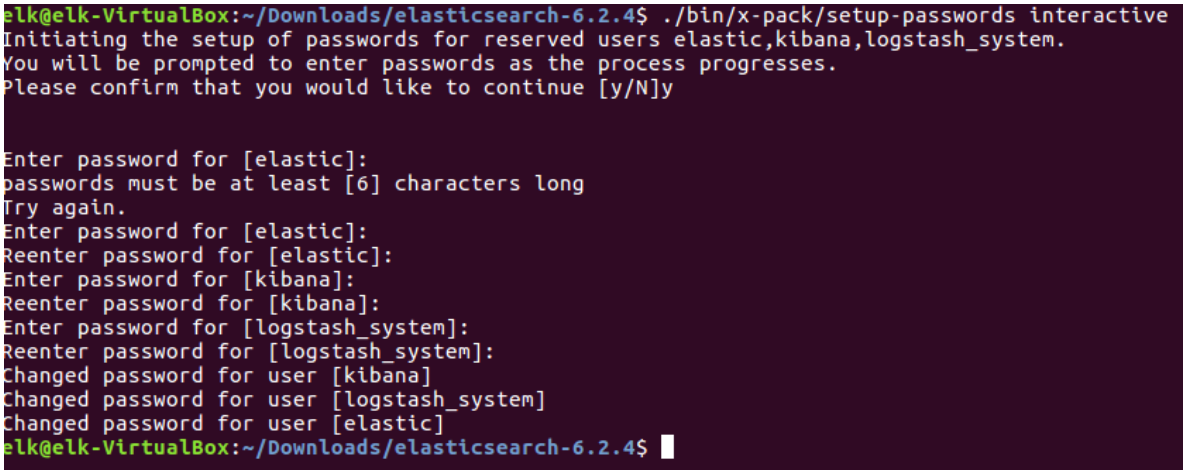
Figura 8. Inicio de Elasticsearch - Terminal

```
elk@elk-VirtualBox:~/Downloads/elasticsearch-6.2.4$ ./bin/elasticsearch
[2018-06-24T21:41:16,798][INFO ][o.e.n.Node               ] [] initializing ...
[2018-06-24T21:41:17,203][INFO ][o.e.e.NodeEnvironment    ] [y1Mng26] using [1] data paths, mounts [[/ (/dev/sda1)]], net usable_space [8.4gb]
, net total_space [16.5gb], types [ext4]
[2018-06-24T21:41:17,204][INFO ][o.e.e.NodeEnvironment    ] [y1Mng26] heap size [1015.6mb], compressed ordinary object pointers [true]
[2018-06-24T21:41:17,206][INFO ][o.e.n.Node               ] node name [y1Mng26] derived from node ID [y1Mng26oRgqqKEWanBof2w]; set [node.name]
 to override
[2018-06-24T21:41:17,206][INFO ][o.e.n.Node               ] version[6.2.4], pid[11104], build[ccec39f/2018-04-12T20:37:28.497551Z], OS[Linux/4
.10.0-28-generic/amd64], JVM[Oracle Corporation/OpenJDK 64-Bit Server VM/1.8.0_171/25.171-b11]
[2018-06-24T21:41:17,207][INFO ][o.e.n.Node               ] JVM arguments [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancy
Fraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=tru
e, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0,
 -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -Djava.io.tmpdir=/tmp/elasticsearch.ouOEb3LE, -XX:+HeapDumpOnOutOfMemoryError,
-XX:+PrintGCDetails, -XX:+PrintGCDateStamps, -XX:+PrintTenuringDistribution, -XX:+PrintGCApplicationStoppedTime, -Xloggc:logs/gc.log, -XX:+Use
GCLogFileRotation, -XX:NumberOfGCLogFiles=32, -XX:GCLogFileSize=64m, -Des.path.home=/home/elk/Downloads/elasticsearch-6.2.4, -Des.path.conf=/h
ome/elk/Downloads/elasticsearch-6.2.4/config]
```

Fuente: Propia

9. Configurar las contraseñas de x-pack en Elasticsearch

Figura 9. Configuración de x-pack - Terminal



Fuente: Propia

Configuración

En esta sección se realizó la configuración por medio de archivos de la recolección de datos en Logstash y la forma de envío a Elasticsearch, también la conexión de Kibana con Elasticsearch es configurada en esta sección.

1. Construir un archivo de configuración para Logstash. Como se puede ver en la Figura 10, este archivo describe la fuente de los datos (input), el cual es Twitter. Esta fuente requiere una aplicación creada. Para cada aplicación se generan unas credenciales de acceso, estas credenciales son: Consumer key, consumer secret, oauth_token y oauth_token_secret. Las keywords son las palabras clave que se buscarán en twitter, ya sea en tweets, hagstash o nombres de usuario, por último, full tweet indica si el tweet encontrado debe ser traido completo.

El output describe la forma en que los datos serán tratados luego de ser recolectados por Logstash, en este caso serán mandados a Elasticsearch con las credenciales en user, password.

Las restricciones para Logstash con esta configuración vienen siendo las que tenga el API de Twitter.

Un index es el nombre con el que se identificará al conjunto de datos recolectados por esa configuración específica.

Figura 10. Configuración de Logstash

Fuente: Propia

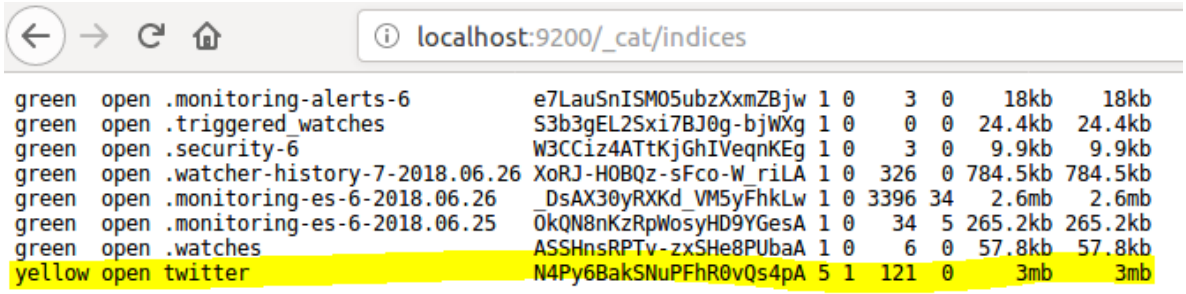
2. En caso de usar un template este debe ser definido. Un template puede usarse para definir qué campos son enviados, la estructura de estos o incluso el tipo con el que Elasticsearch debe identificarlos.

Figura 11. Template .json de Logstash

Fuente: Propia

3. Iniciar Logstash, para verificar que se creó el index y que recibe datos.

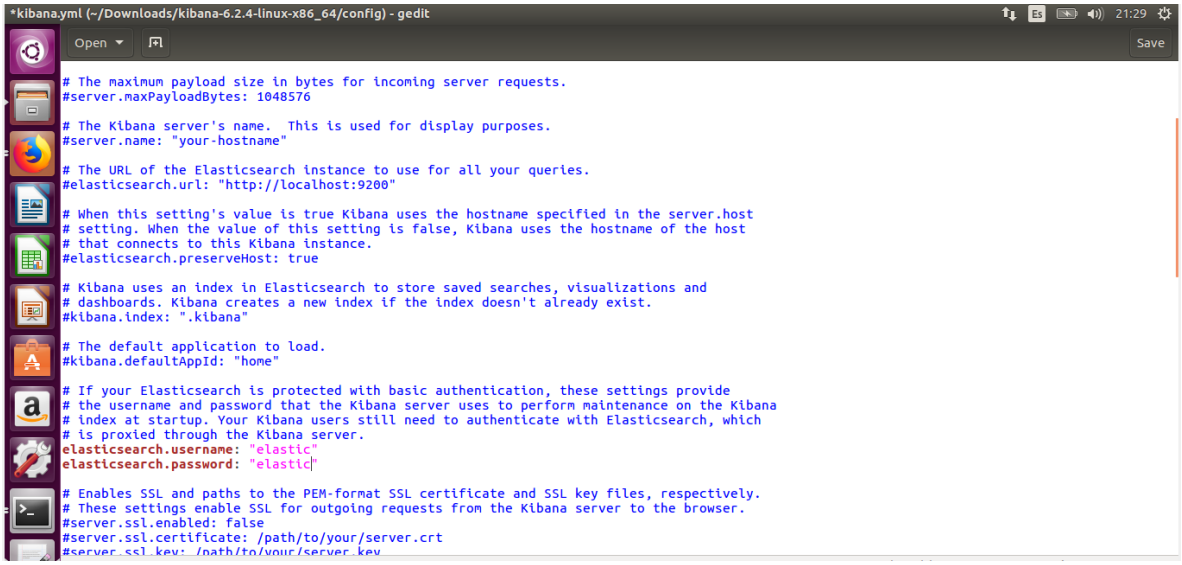
Figura 12. Índices presentes en Elasticsearch



Fuente: Propia

4. Configurar Kibana en kibana.yml, usar las credenciales definidas

Figura 13. Kibana.yml



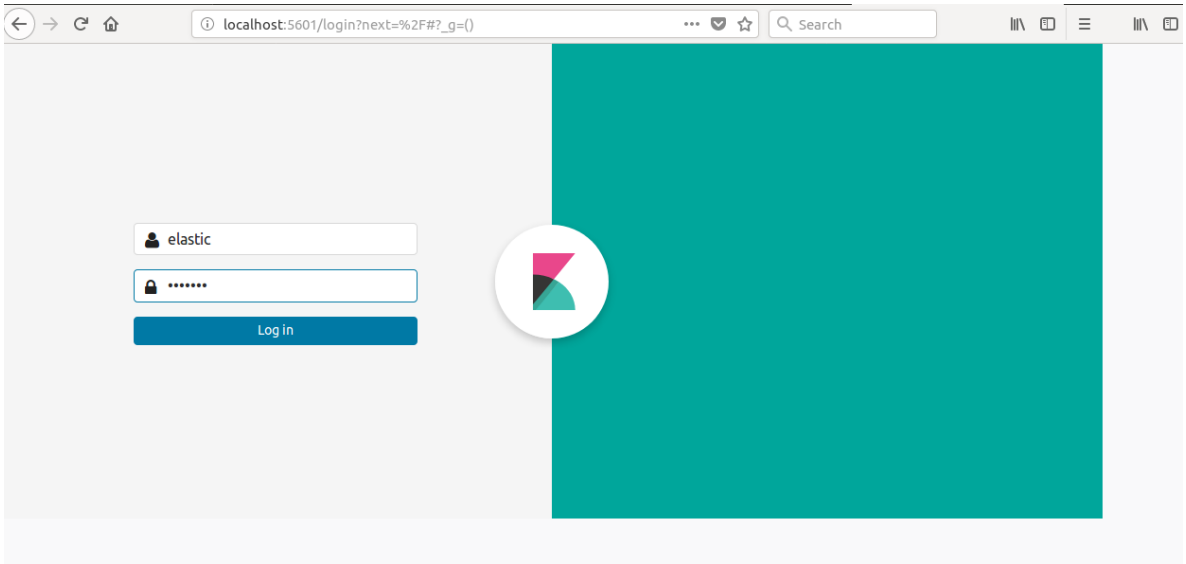
Fuente: Propia

Caso de uso con Kibana

En este caso de uso se validará la información obtenida mediante Logstash y se hará un ejemplo de uso de la consola de Elasticsearch, junto con una generación de gráficas de Kibana.

1. Ingresar a localhost:5601 y autenticarse con las credenciales creadas durante el proceso de configuración de x-pack en Elasticsearch.

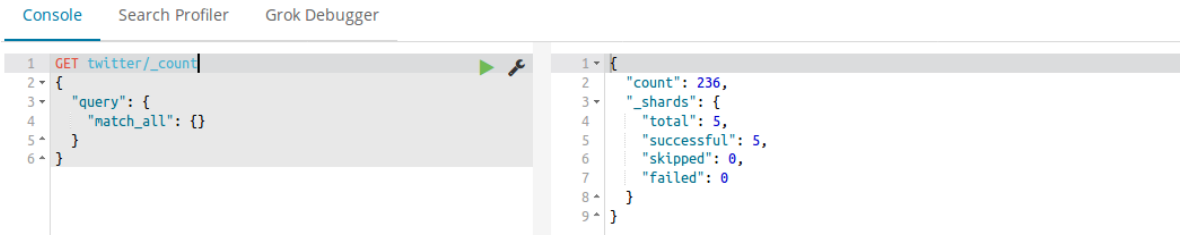
Figura 14. Pantalla de Log in de Kibana



Fuente: Propia

2. En la consola se pueden hacer consultas directamente a Elasticsearch. En la Figura 15 se puede ver una consulta del conteo total de registros para el índice elegido.

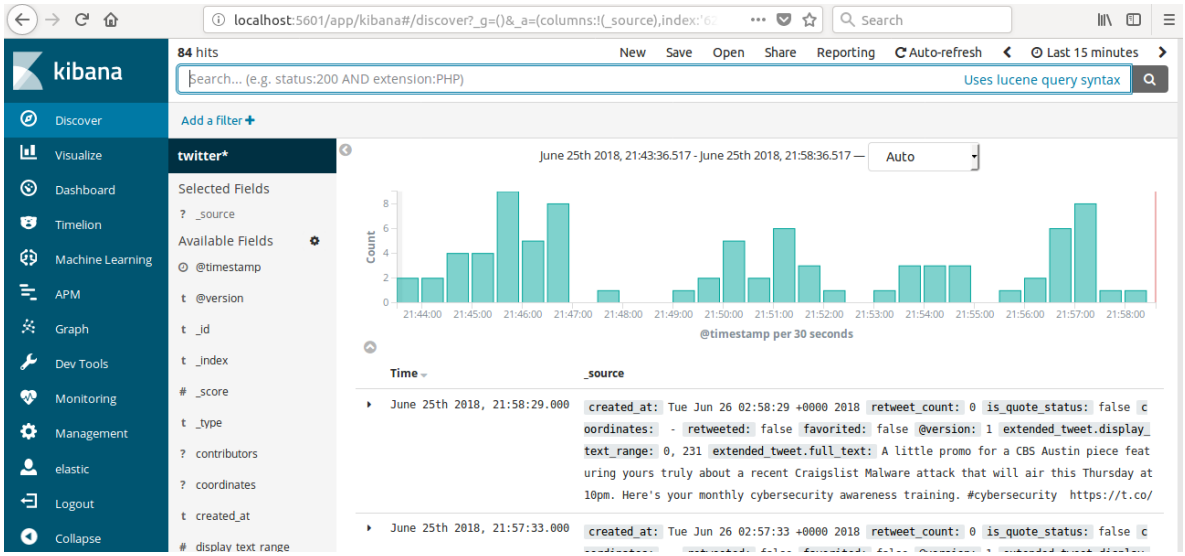
Figura 15. Consola de desarrollo de Elasticsearch



Fuente: Propia

3. Para ver los datos se usa la pestaña “discover”. La Figura 16 muestra los datos tal como fueron almacenados, aún no han sido procesados por Kibana.

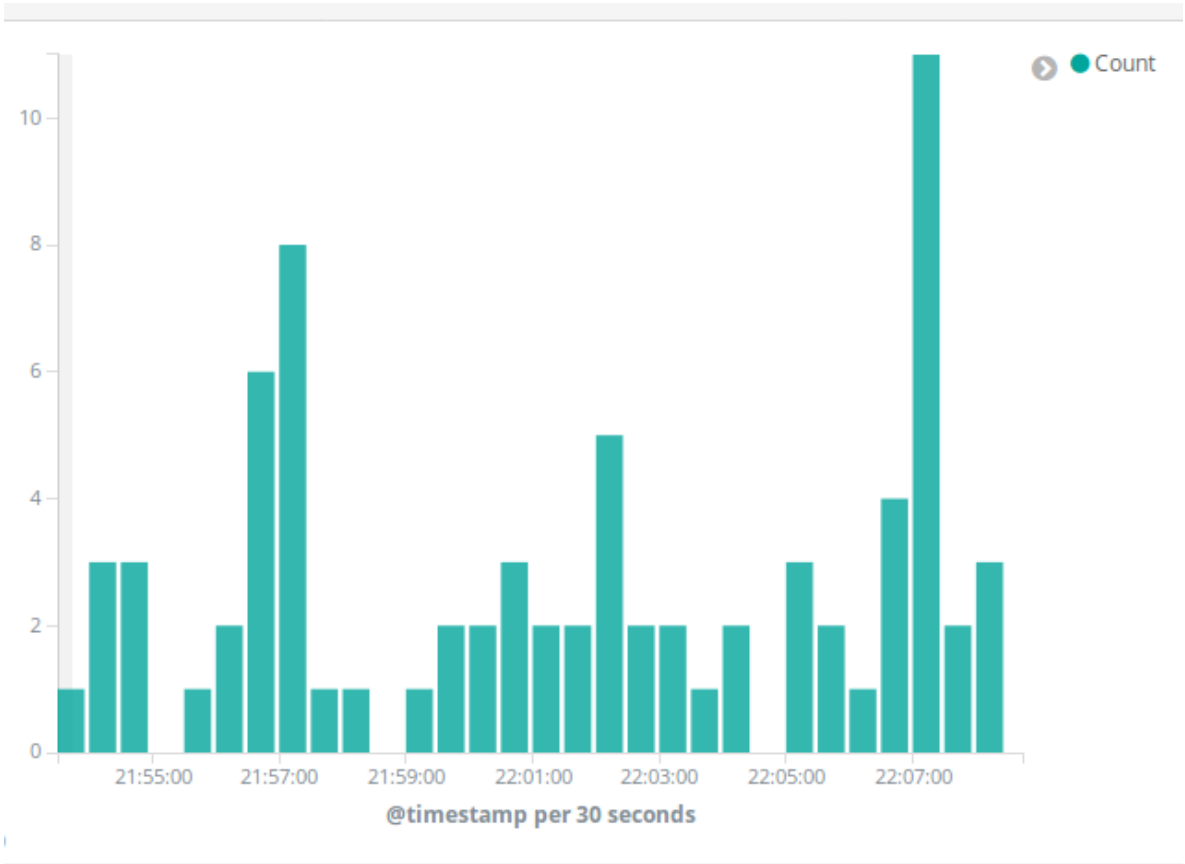
Figura 16. Discover de datos en Kibana



Fuente: Propia

4. Kibana permite hacer graficar los datos en informes de todo tipo. En la Figura 17 se puede observar un conteo de los tweets realizados cada 30 segundos que fueron recolectados.

Figura 17. Gráfica timestamp en Kibana



Fuente: Propia

Como se ha podido ver ELK permite reunir datos de distintas fuentes, tanto abiertas como aquellas que tenga la compañía, puesto que en caso de no existir un plug in puede construirse. Con estos datos se pueden hacer reportes ejecutivos y una analítica clara y concisa.

Pese a que fue un ejercicio interesante la recolección y analítica de datos no es la principal característica del centinela. El entorno ELK no está orientado a compartir la información obtenida ni los análisis realizados, de modo que este entorno fue descartado, sin embargo, se está evaluando la posibilidad de pasar información de eventos de MISP a este entorno y presentarla como reporte ejecutivo.

MISP

Compartir información de malware con comunidades y redes conectadas es bastante atractivo, esto junto con la correlación automática hacen de MISP una herramienta a estudiar y que posiblemente cumple con el criterio de volver al centinela colaborativo.

Instalación

Pese a que existen instrucciones de instalación bastante claras sobre MISP, se decidió usar una imagen generada automáticamente para ejecutar los casos de uso.

Descargar la máquina virtual de MISP[10] de e importarla como un servicio en VirtualBox. Para los casos de uso o bien la máquina debe ser importada dos veces o clonada.

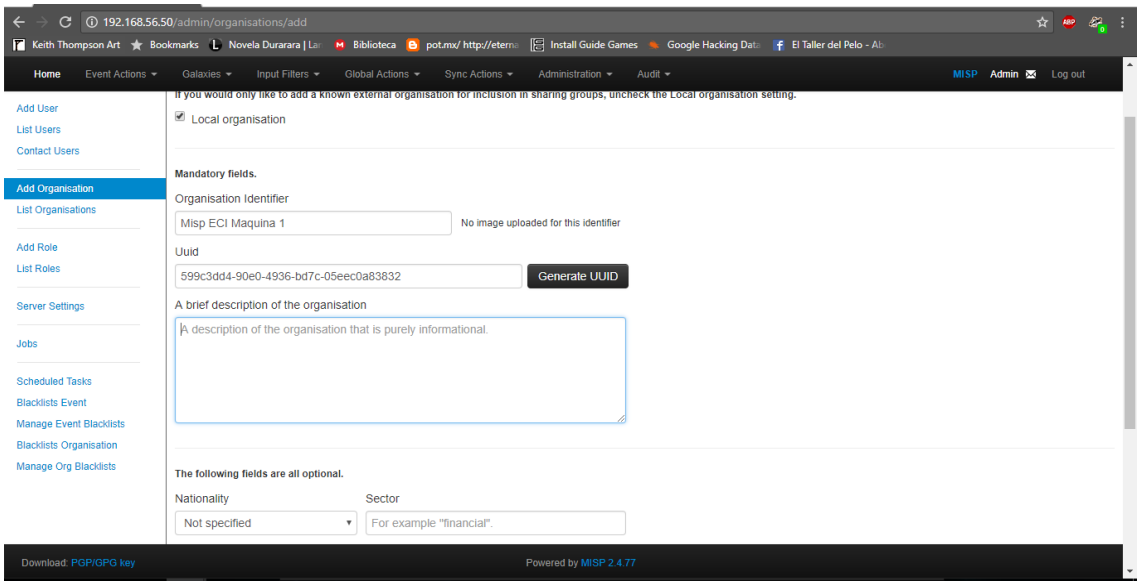
Caso de uso 1, gestión de organizaciones y sincronización de eventos.

En este caso de uso se realizó la creación de organizaciones en instancias distintas, creación de un evento en una de ellas para luego realizar una sincronización y compartirlo con la otra organización.

1. En ambas máquinas autenticarse como administrador y crear una organización

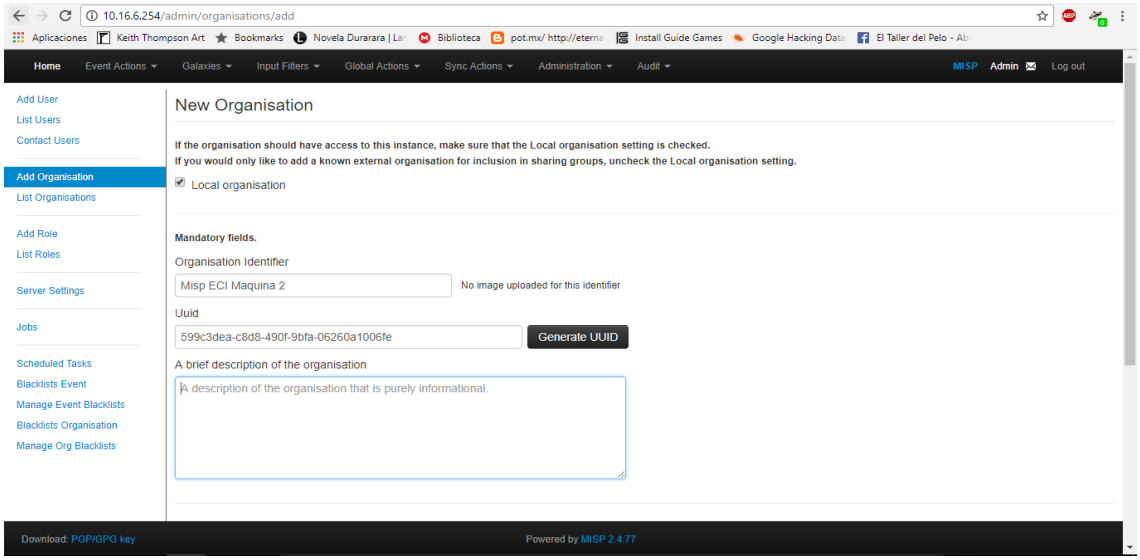
[10] https://www.circl.lu/misp-images/

Figura 18. Nueva organización de MISP – Máquina 1



Fuente: Propia

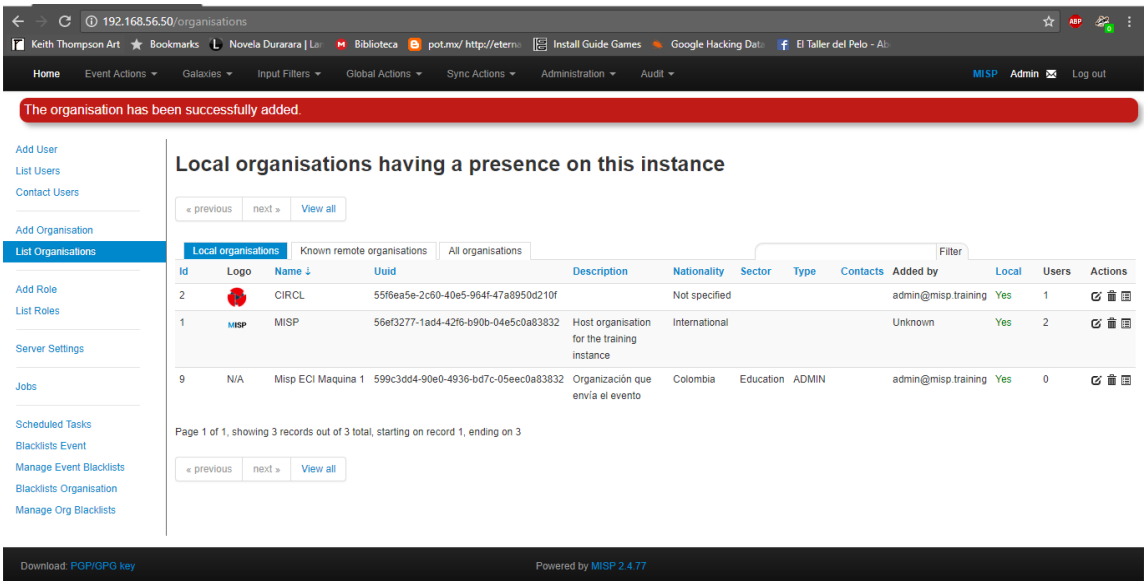
Figura 19. Nueva organización de MISP – Máquina 2



Fuente: Propia

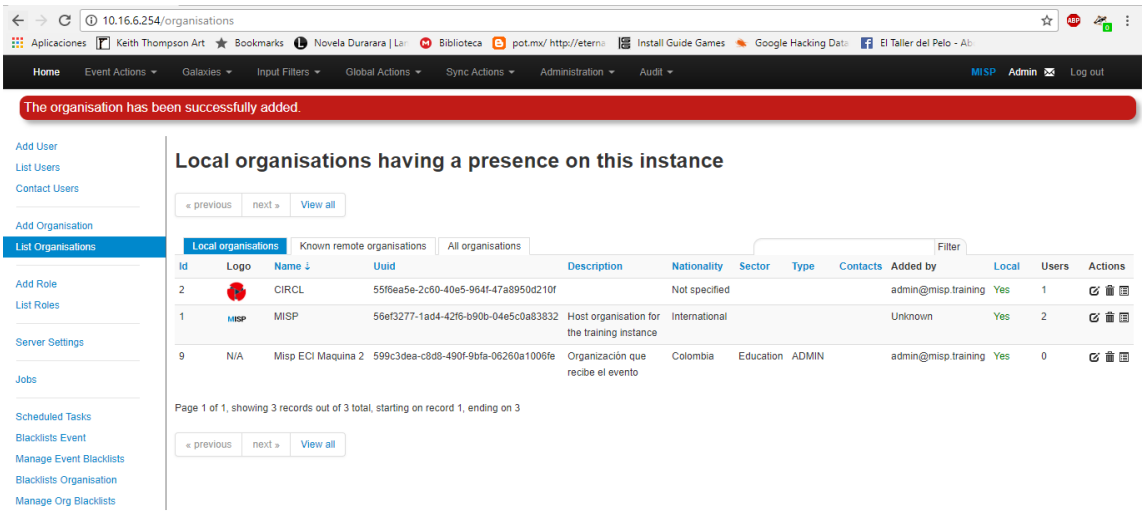
2. Se termina de crear las organizaciones como locales y se verifica que las instancias no interfieran entre ellas. Esta verificación se hace revisando las organizaciones en cada instancia y revisando que solo aparezca la creada en esa máquina.

Figura 20. Organizaciones MISP en la maquina 1



Fuente: Propia

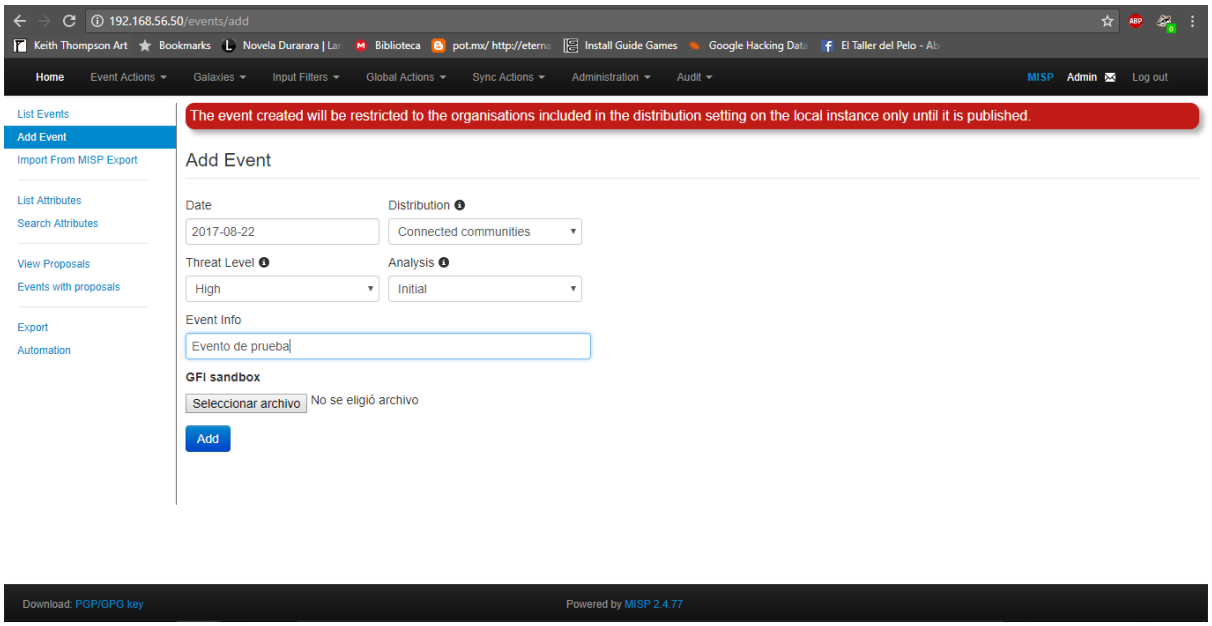
Figura 21. Organizaciones MISP en la maquina 2



Fuente: Propia

3. La organización de la máquina 1 creará un evento

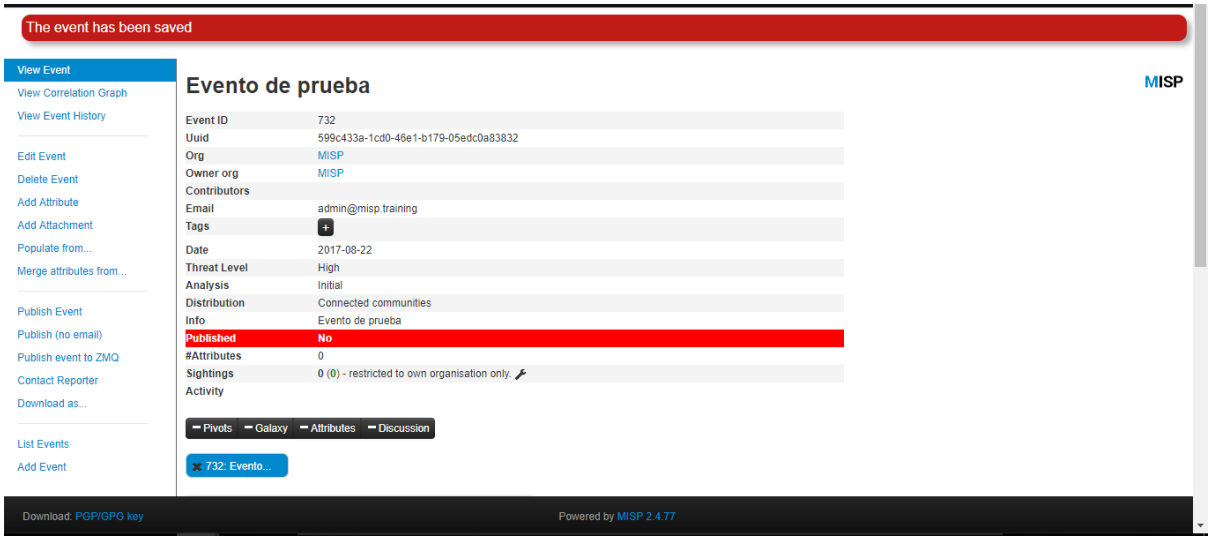
Figura 22. Creación de evento MISP



Fuente: Propia

Al escoger la opción de comunidades conectadas el evento se compartirá con las comunidades o instancias que estén sincronizadas.

Figura 23. Configuración del evento MISP



Fuente: Propia

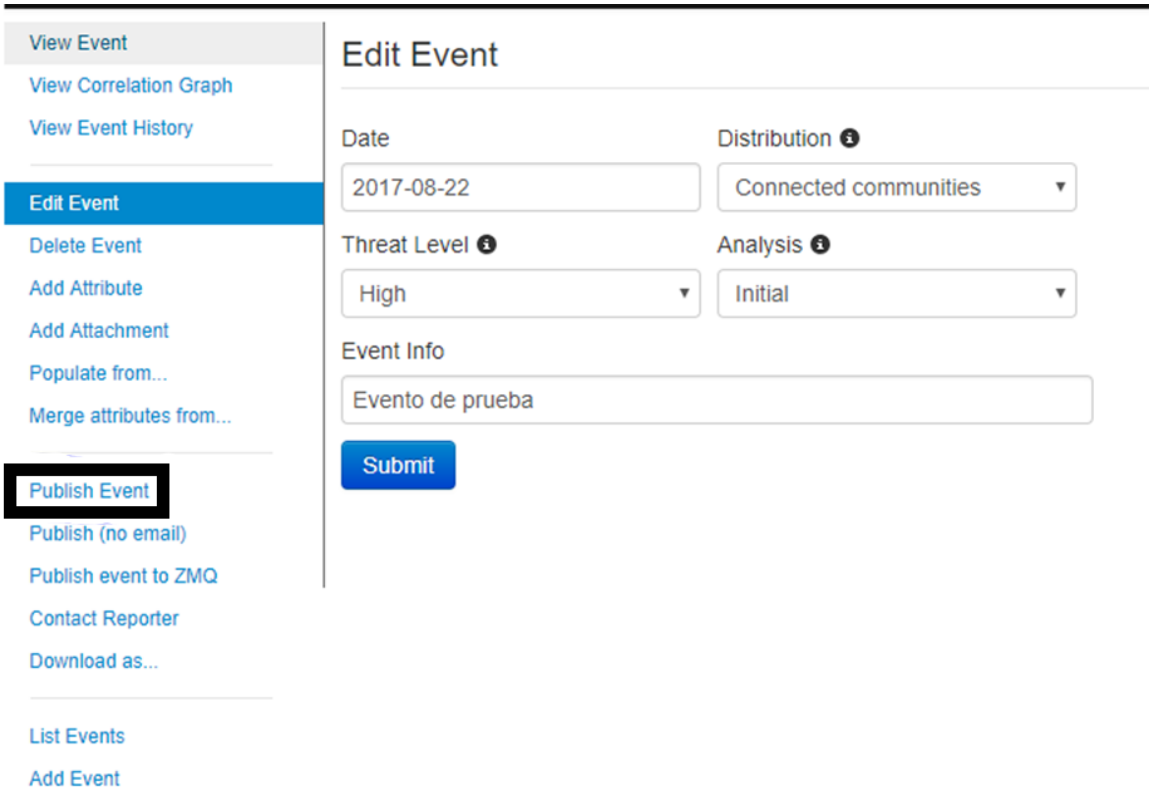
4. Se procede a darle atributos al evento, uno privado y uno público

Figura 24. Atributos asociados al evento MISP

Fuente: Propia

5. El evento es publicado.

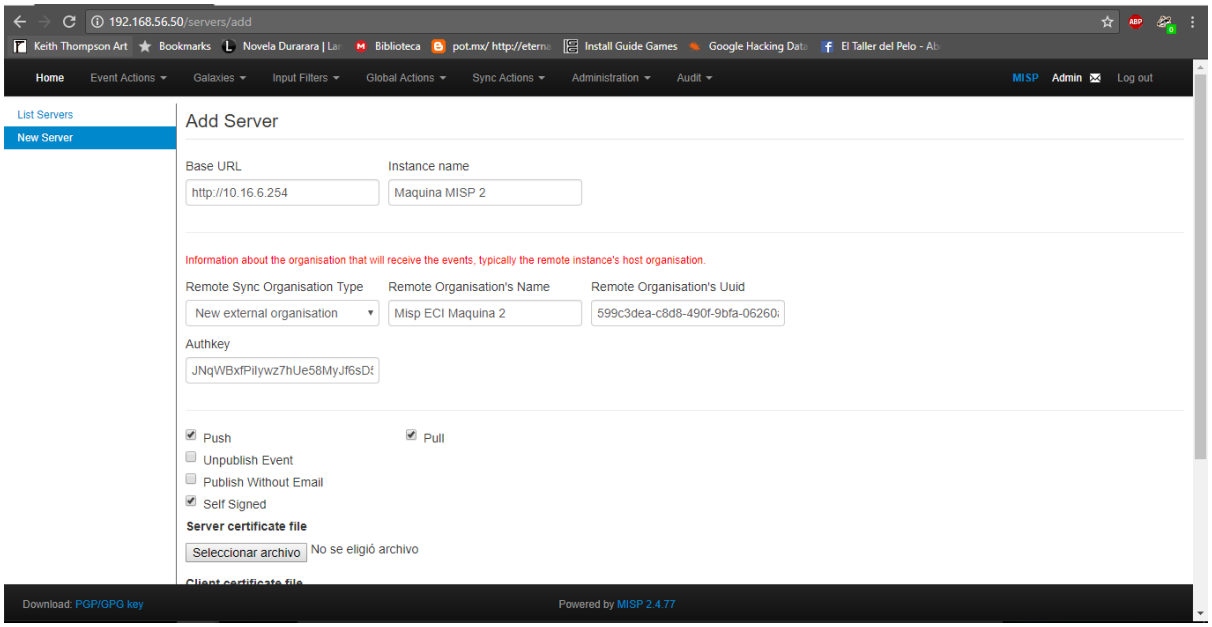
Figura 25. Atributos del evento antes de su publicación



Fuente: Propia

6. Ahora se realiza la sincronización entre máquinas. Para la sincronización se tienen las opciones de push, en la cual la máquina envía sus eventos a una instancia de MISP conectada y sincronizada. La otra operación es de pull, en esta operación la instancia solicita los nuevos eventos.

Figura 26. Sincronización entre máquinas



Fuente: Propia

El authkey se encuentra en el perfil de cada usuario con derechos de sincronización.

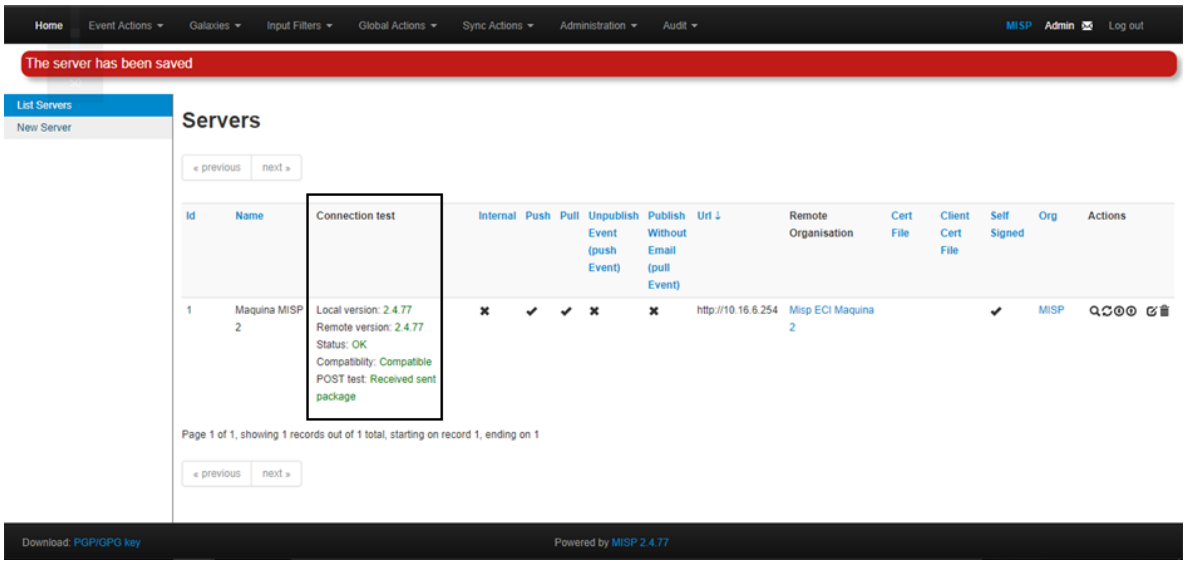
Figura 27. Clave de autenticación

Id	1
Email	admin@misp.training
Org	MISP
Role	admin
Autoalert	No
Contactalert	No
Authkey	JNqWBxfPilywz7hUe58MyJf6sD5PrTVaGm7hTn6c (reset)
NIDS Start SID	4000000
Termsaccepted	Yes
PGP key	N/A

Fuente: Propia

7. Como se puede ver en la Figura 28, desde la máquina 1 se puede hacer una prueba de conexión, en esta prueba se detecta la versión, status y compatibilidad de la máquina objetivo, para hacer la prueba se debe hacer click en el botón de “run”, al hacer la prueba las máquinas son capaces de comunicarse, es decir, pueden compartir información.

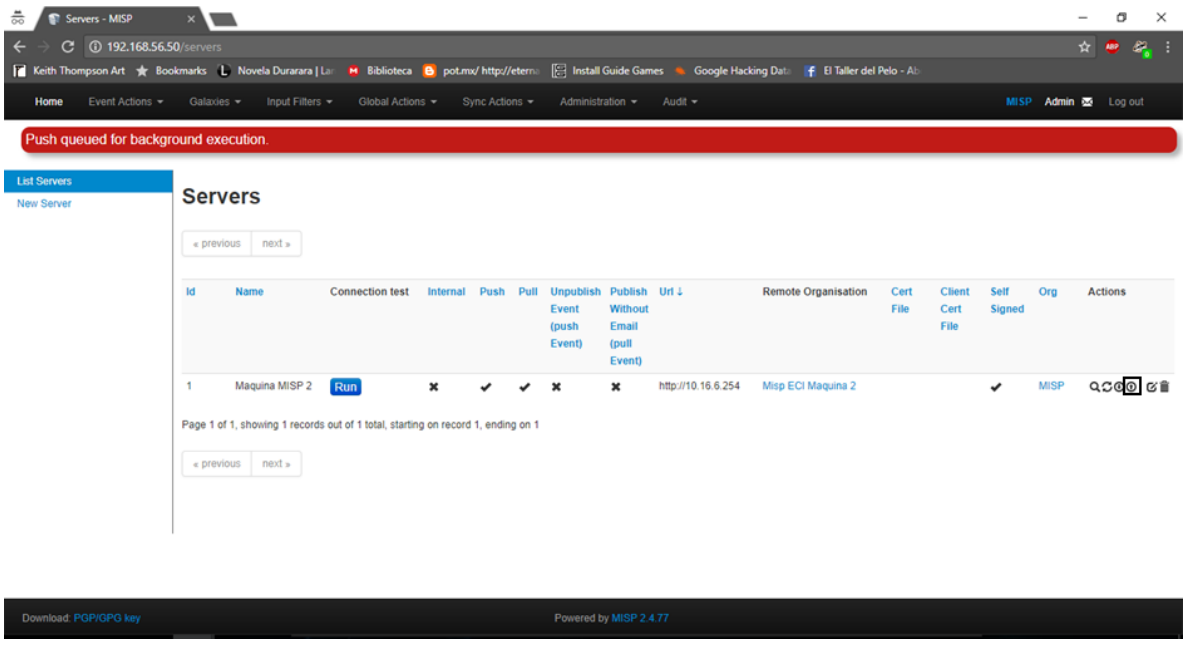
Figura 28. Comunicación entre instancias



Fuente: Propia

8. Desde el servidor 1 se hace push al servidor 2 para compartir los eventos. El botón de push está señalado en la Figura 29.

Figura 29. Compartir eventos

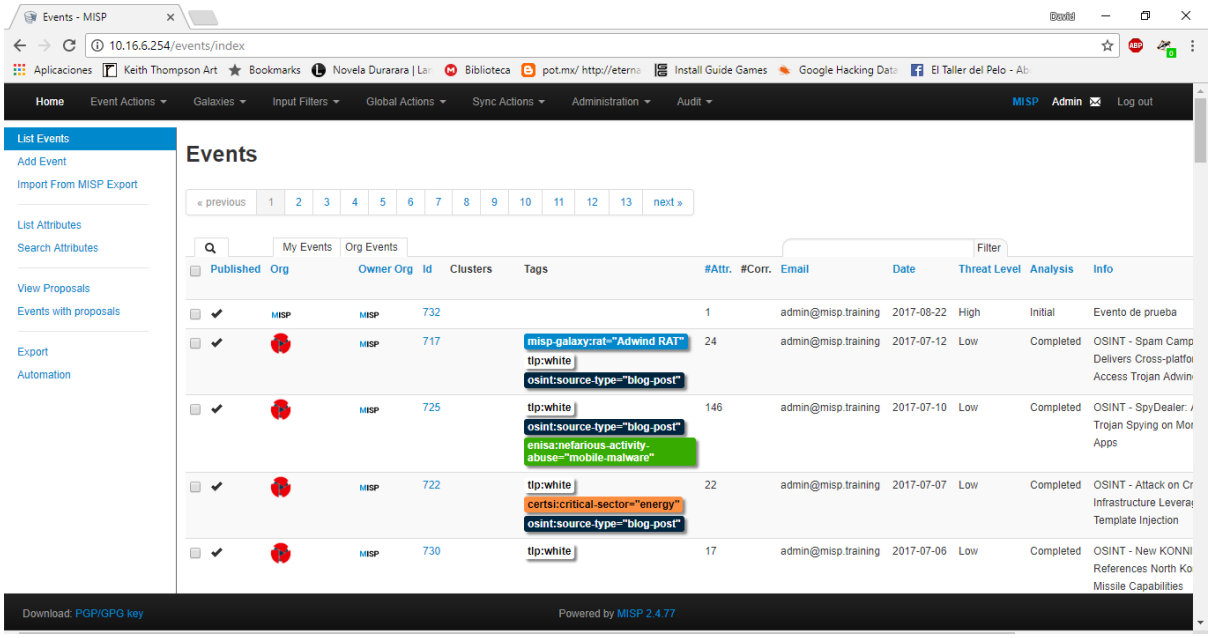


Fuente: Propia

Al actualizar la lista de eventos del servidor 2 se puede ver que el evento ha sido

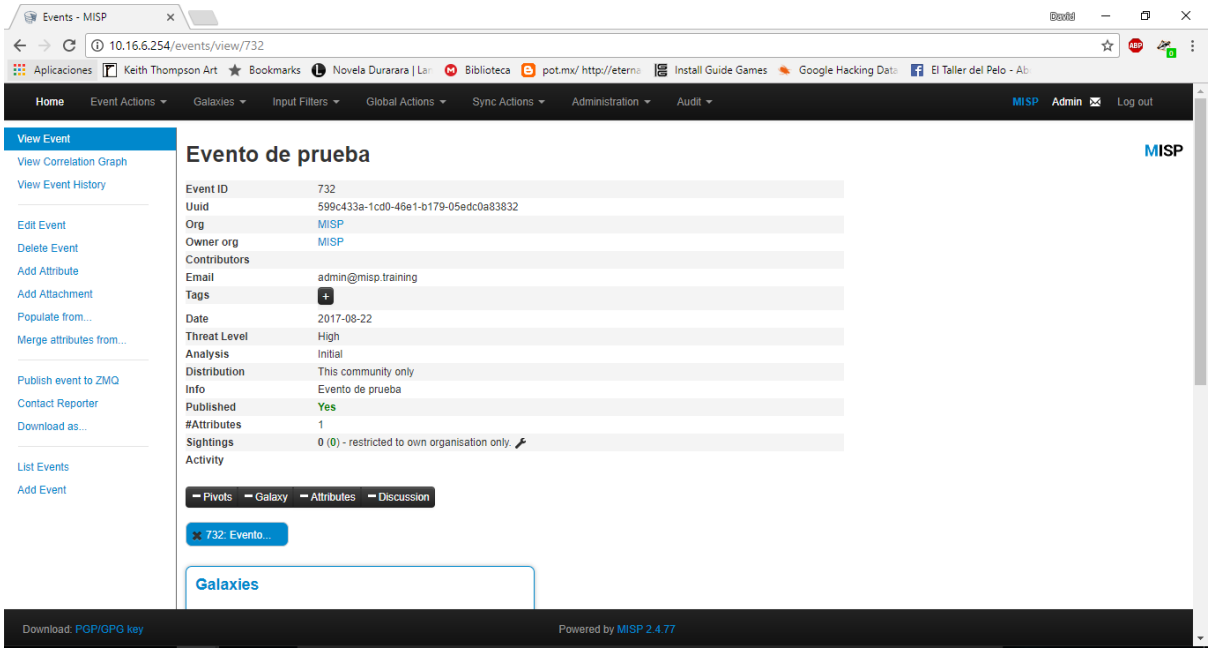
compartido. Como se puede notar, de los 2 atributos definidos en la máquina 1, solo 1 de ellos aparece compartido. El atributo que aparece es el hash md5, definido como un atributo global, el otro atributo es el nombre del sospechoso, pero al ser un evento de organización, solo la organización en la instancia original puede verlo.

Figura 30. Evento recibido por la maquina 2



Fuente: Propia

Figura 31. Detalles del evento recibido por la máquina 2

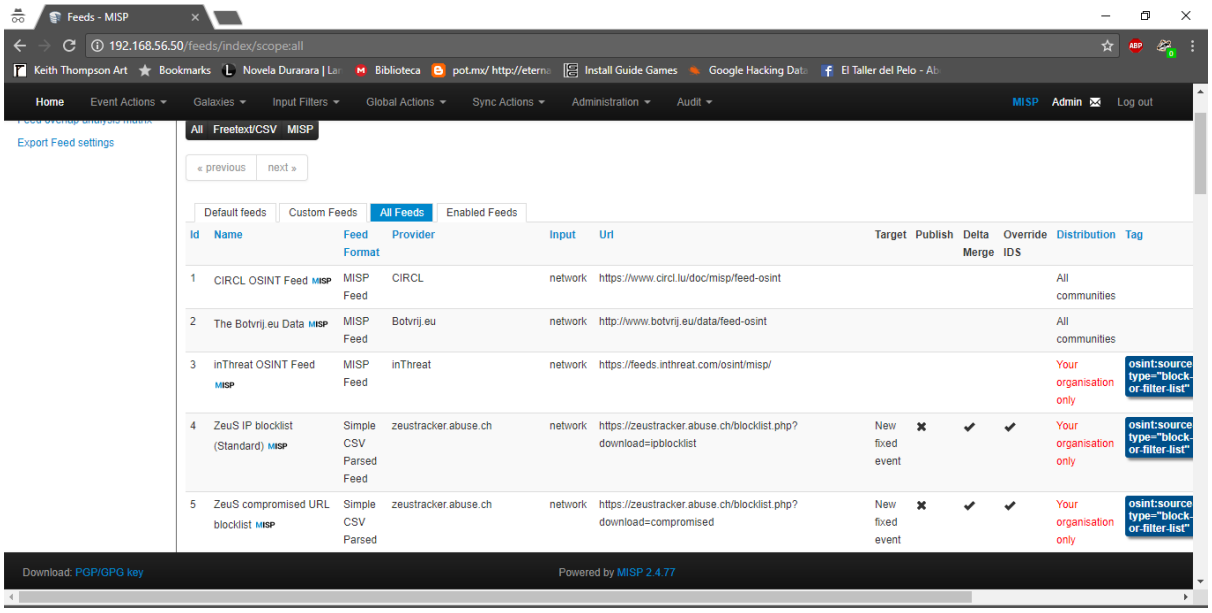


Fuente: Propia

Caso de uso 2, activación y añadidura de nuevos feeds a MISP

1. Para conectarse a un generador de información sobre amenazas, en la pestaña de Sync actions/list feeds, se muestran todos los posibles servidores.

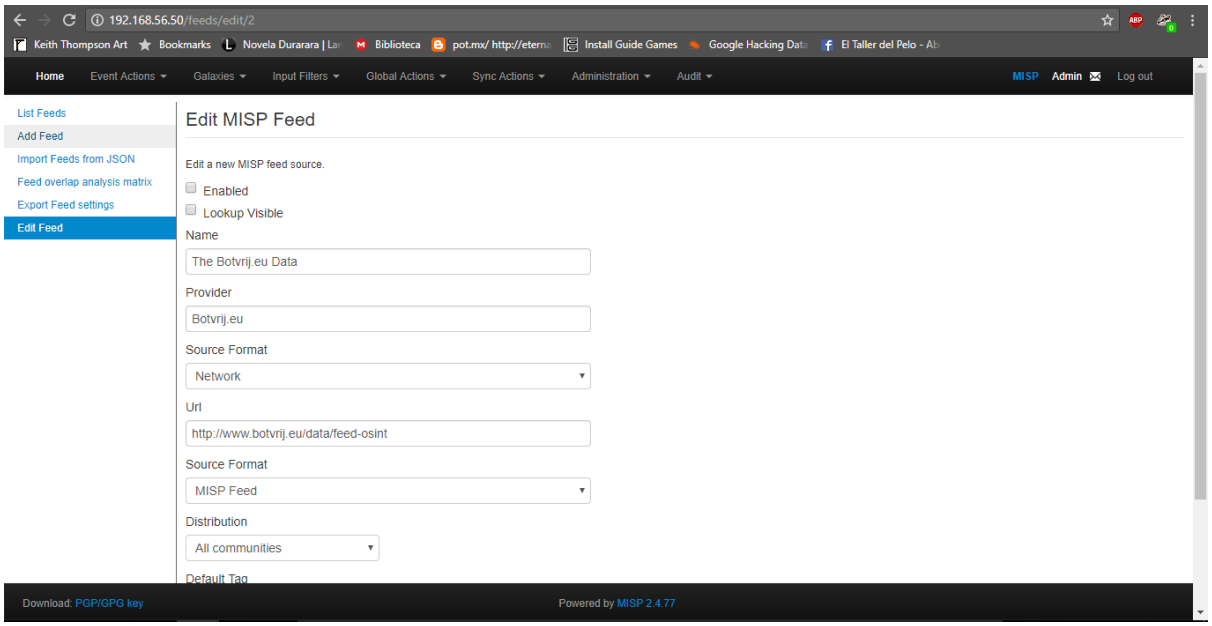
Figura 32. Información de feeds



Fuente: Propia

2. De ahí se toma uno que no esté prendido y se edita para activarlo.

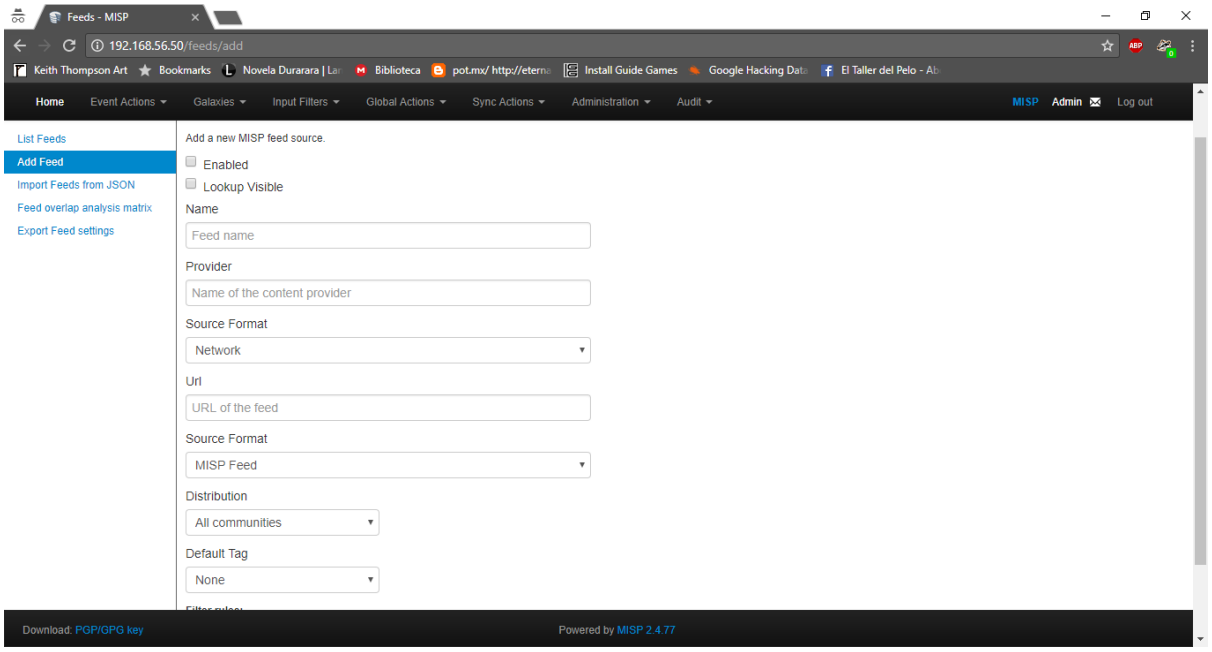
Figura 33. Feed en detalle



Fuente: Propia

3. Marcando el check box de “Enabled” los eventos provenientes del feed se empiezan a mostrar en la lista de eventos de la instancia.
4. Para añadir un feed inexistente en la lista se puede de dos formas La primera es añadiendo manualmente la información.

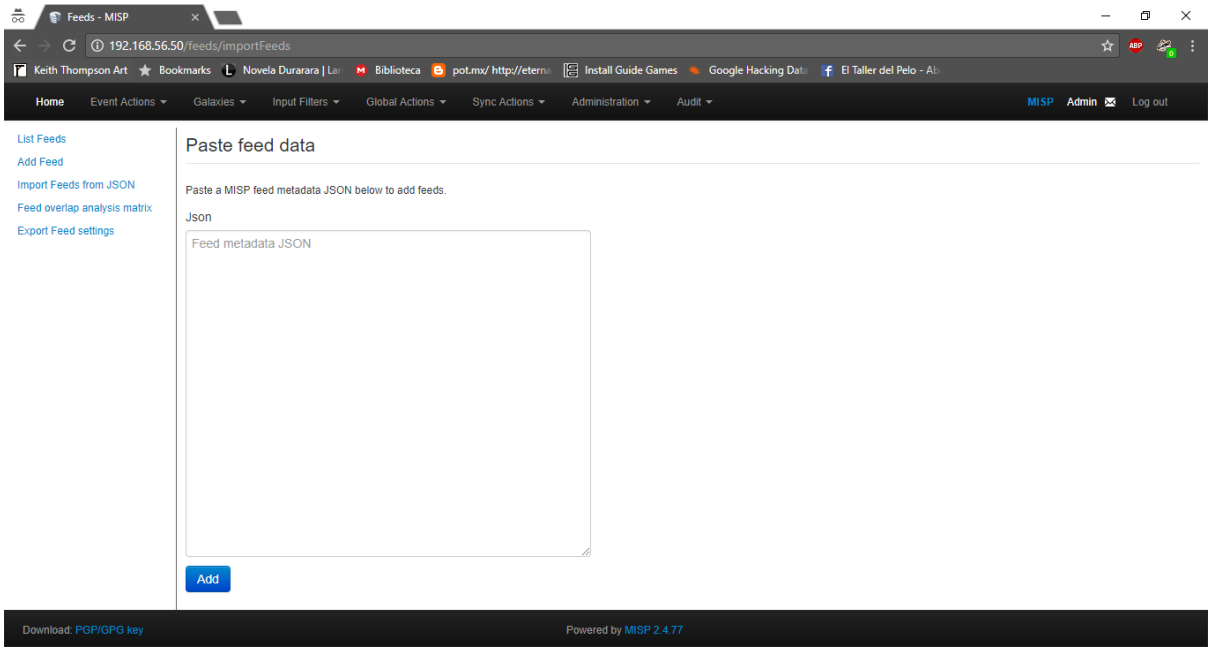
Figura 34. Añadir un feed



Fuente: Propia

La segunda es importar directamente los metadatos desde JSON.

Figura 35. Importar desde JSON

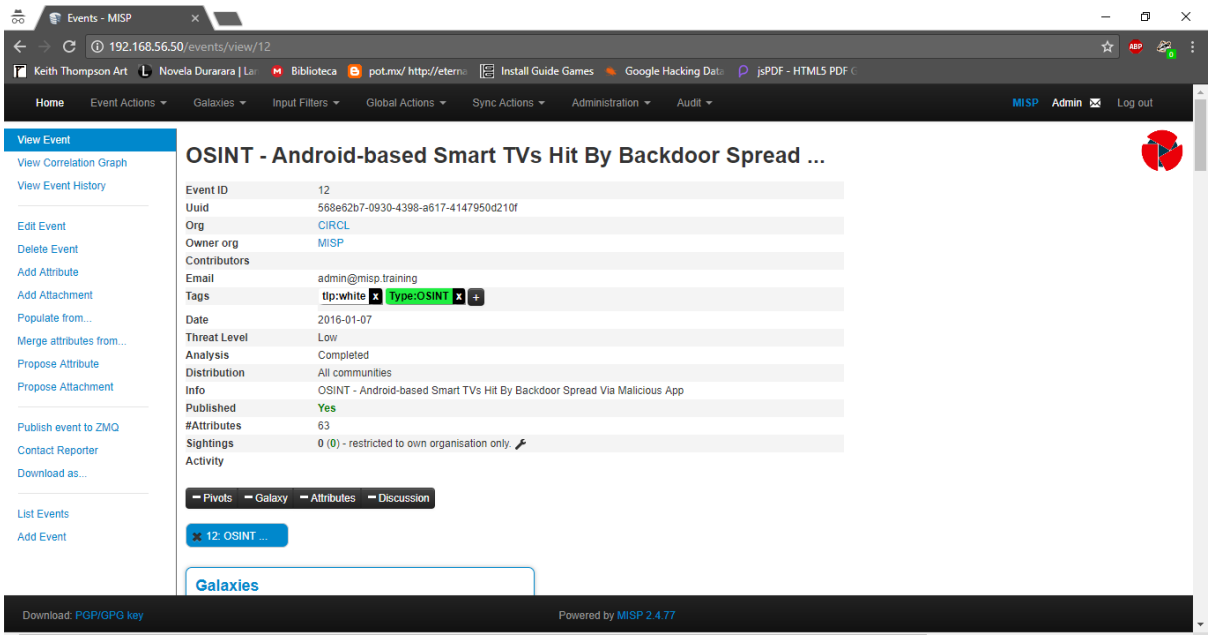


Fuente: Propia

Caso de uso 3, Correlación de eventos mediante atributos

1. Se escoge un evento, en este caso al buscar “IoT” entre los eventos disponibles se encuentra un evento (Número 12) en el cual un malware se expande por televisores inteligentes via Backdoors.

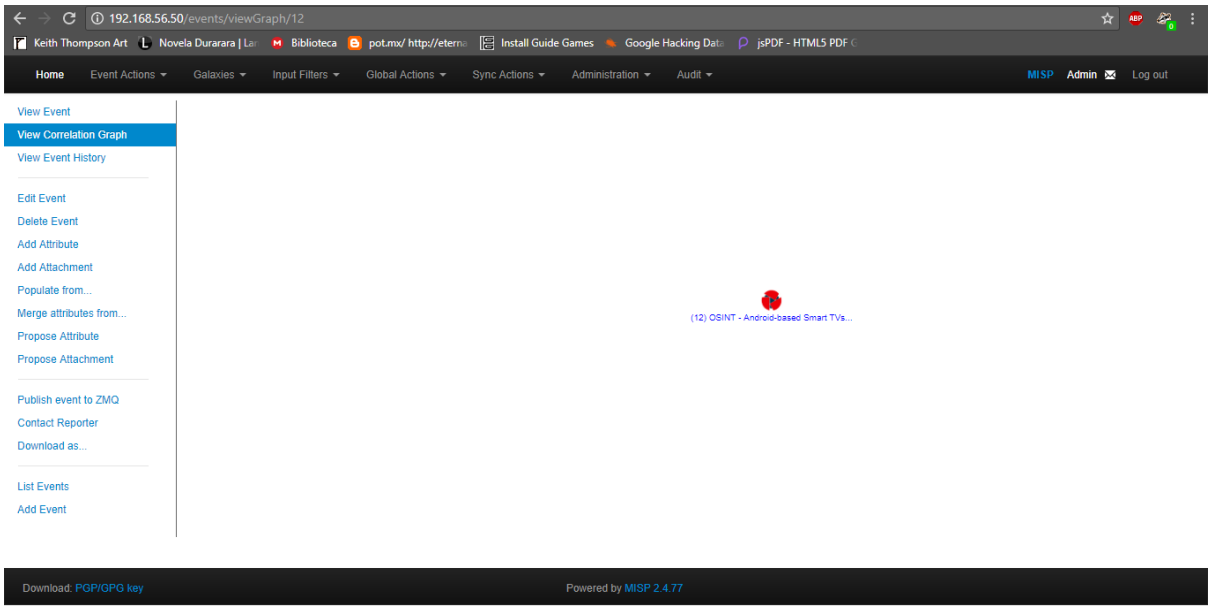
Figura 36. Evento por correlacionar



Fuente: Propia

2. Al ver el grafo de correlación se nota que es un evento solitario, es decir, este evento no está correlacionado, dando a entender que es la primera vez que los atributos descritos del malware han sido identificados en MISP.

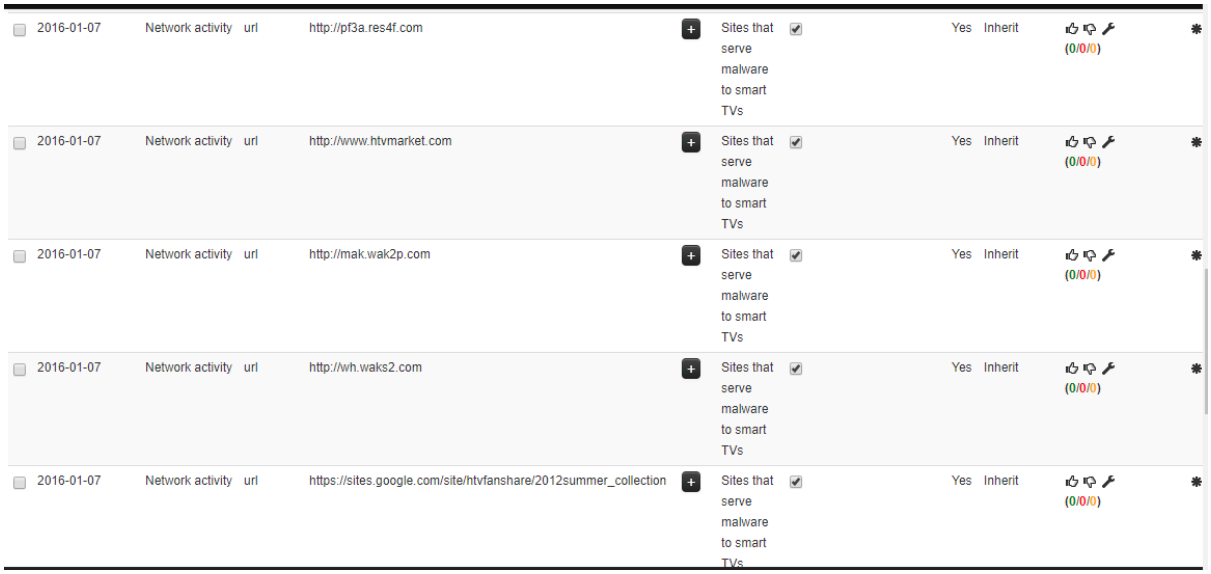
Figura 37. Primer grafo de correlación



Fuente: Propia

3. Se usarán los siguientes atributos del evento 12, que ya existe para crean un nuevo evento, que tendrá como id 733 y observar la correlación dada entre ambos eventos.

Figura 38. Atributos a correlacionar

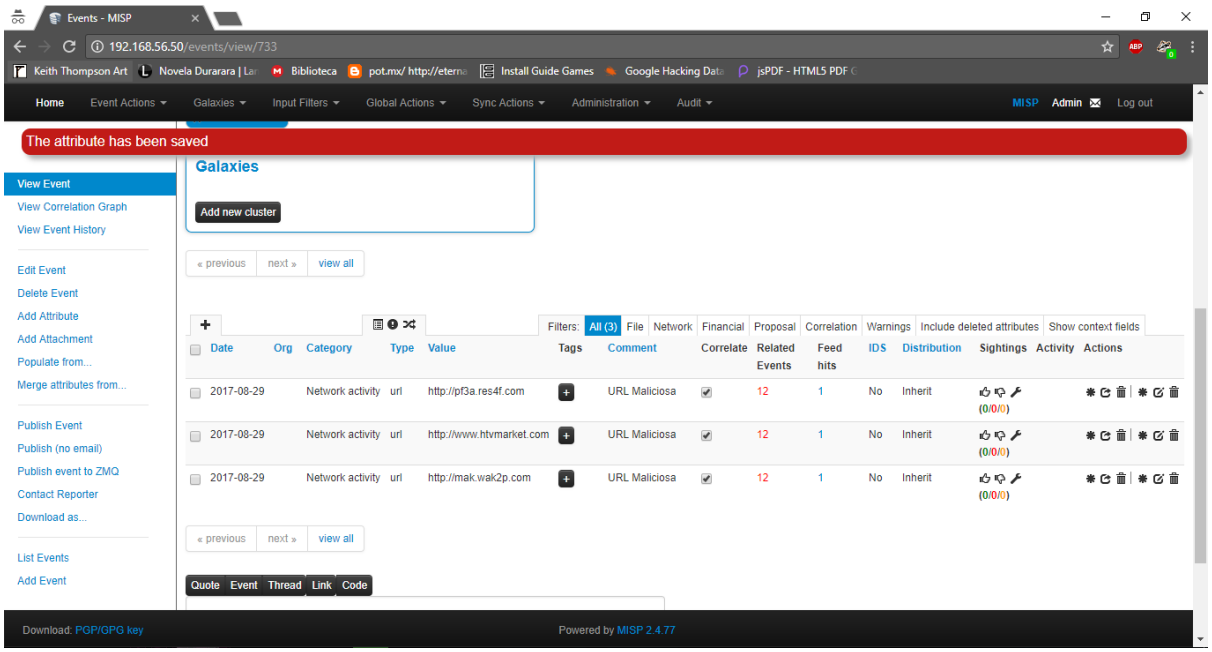


2016-01-07	Network activity	url	http://pf3a.res4f.com	+	Sites that serve malware to smart TVs	☑	Yes	Inherit	(0/0/0)
2016-01-07	Network activity	url	http://www.htvmarket.com	+	Sites that serve malware to smart TVs	☑	Yes	Inherit	(0/0/0)
2016-01-07	Network activity	url	http://mak.wak2p.com	+	Sites that serve malware to smart TVs	☑	Yes	Inherit	(0/0/0)
2016-01-07	Network activity	url	http://wh.waks2.com	+	Sites that serve malware to smart TVs	☑	Yes	Inherit	(0/0/0)
2016-01-07	Network activity	url	https://sites.google.com/site/htvfanshare/2012summer_collection	+	Sites that serve malware to smart TVs	☑	Yes	Inherit	(0/0/0)

Fuente: Propia

4. Se crea un evento (Número 733) y se añaden atributos comunes que los correlacionen, esta correlación servirá para establecer que los sitios realmente son maliciosos, puesto que han estado involucrados en varios incidentes y no son falsos positivos.
5. Como se puede ver, ya aparece un evento relacionado cuyo ID es 12, el evento original del cual fueron extraidas las URLs.

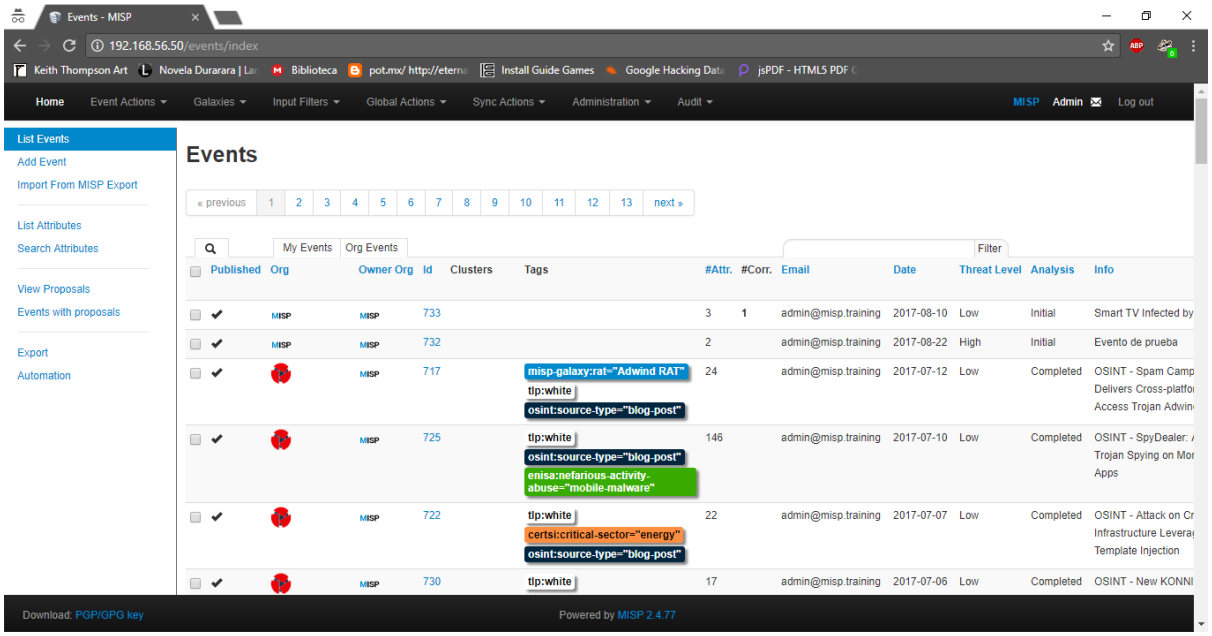
Figura 39. Evento correlacionado



Fuente: *Propia*

Al publicar el evento 733 se crea otro evento con un atributo, este atributo es una URL en común con el evento 733 y 3 URL's con el evento 12.

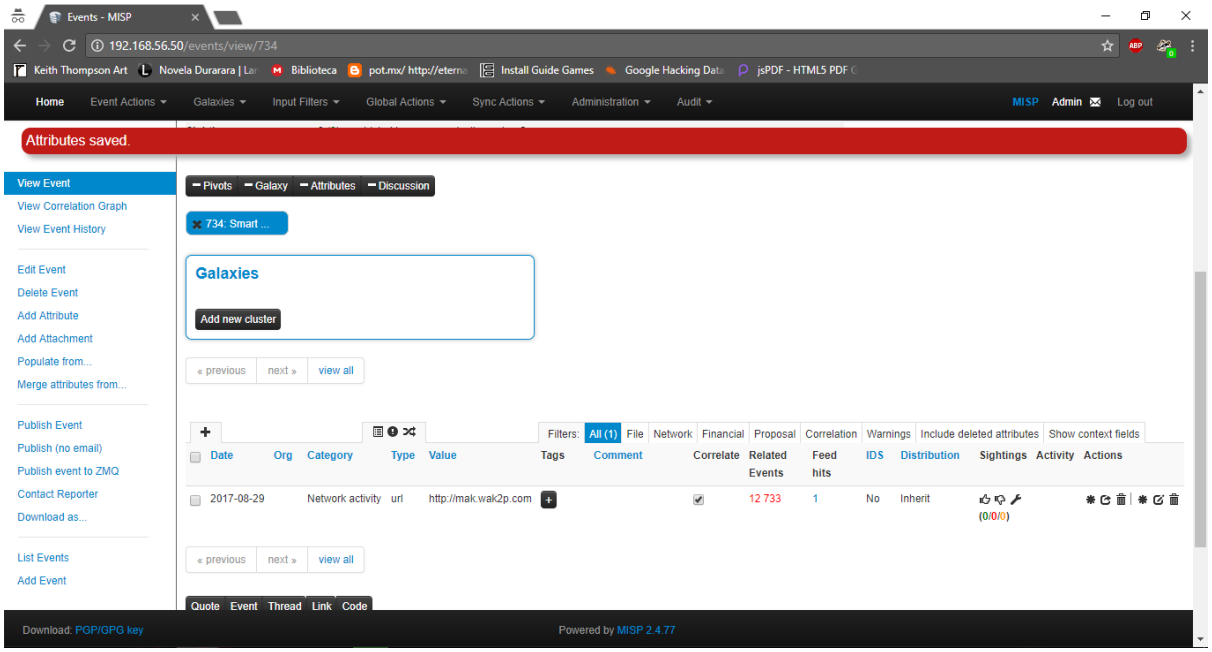
Figura 40. Nuevo evento a correlacionar



Fuente: Propia

En el nuevo evento se ve que la URL compartida con ambos aparece en la correlación

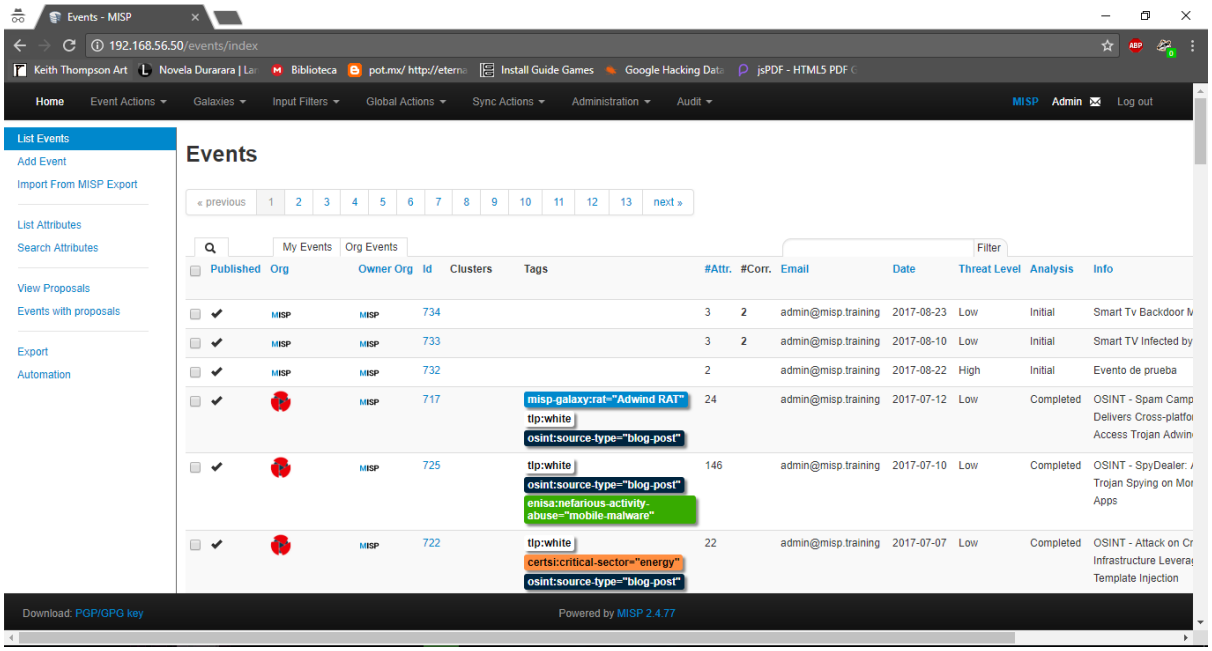
Figura 41. URL correlacionada en nuevo evento



Fuente: Propia

Se terminan de llenar los atributos y se publica el evento

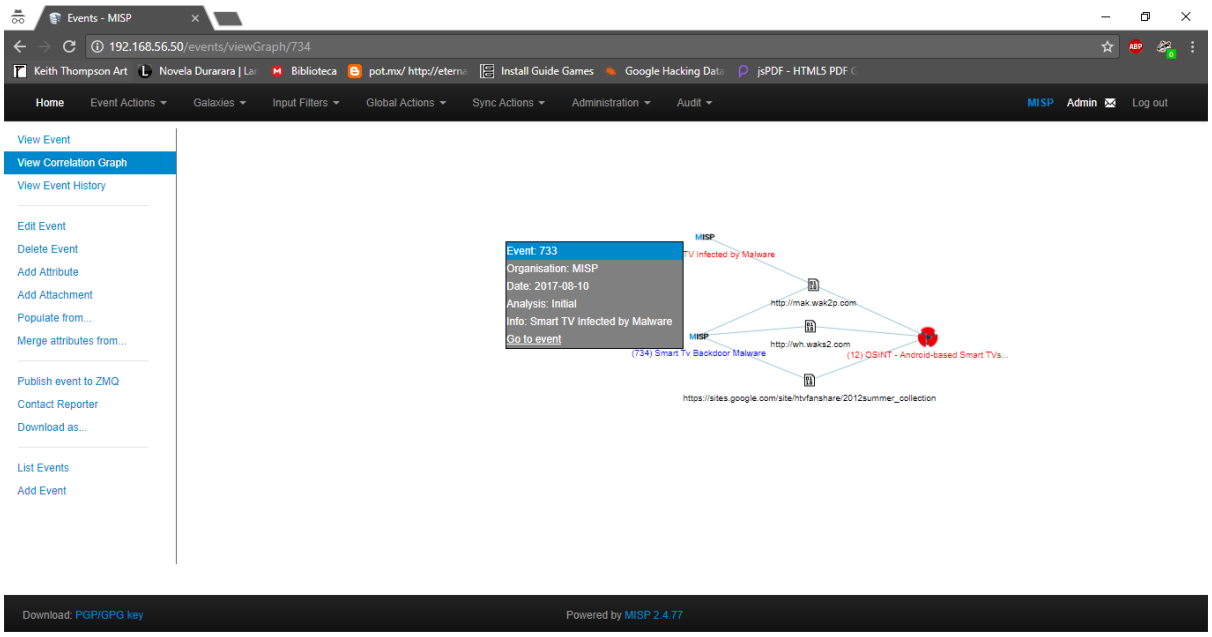
Figura 42. Eventos a correlacionar publicados



Fuente: Propia

6. Ahora se miran las correlaciones hechas seleccionando uno de los eventos

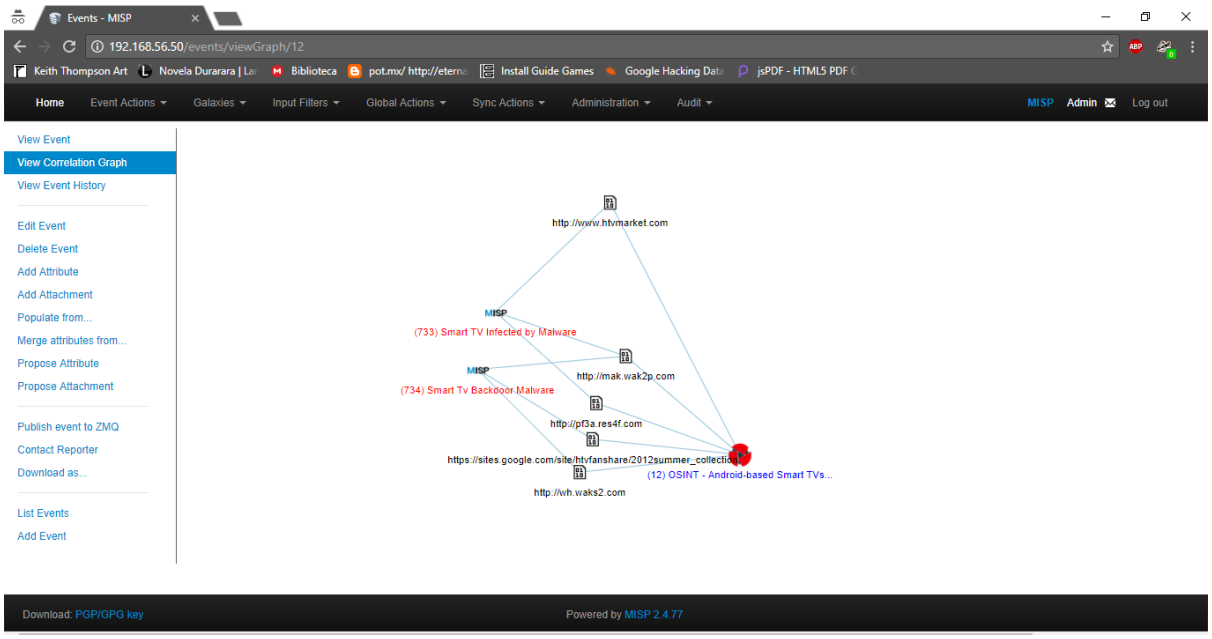
Figura 43. Grafo de correlación



Fuente: Propia

Como se puede ver, solo se muestran las correlaciones que se hacen con ese evento, se muestran los 3 atributos que el evento 734 comparte con el evento 12 y el atributo que comparte con el 733, para ver el grafo completo se selecciona el evento 12.

Figura 44. Grafo del evento 12



Fuente: Propia

Como se muestra en el grafo, todos los atributos que sean compartidos entre eventos se verán en la correlación, ya viendo en la herramienta RADARE los sha del malware o funciones propias de infección pueden convertirse en eventos de MISP y ser compartidos a lo largo de los centinelas.

Como se ha podido ver MISP permite no solo almacenar la información, también facilita su compartición y brinda herramientas para hacer un análisis de correlación que puede derivar en identificación de patrones, ataques comunes y adversarios.

Se decidió que MISP sería integrada al centinela por facilitar la comunicación, exportar datos y organizarlos, pero sobre todo la automatización, necesaria para que los centinelas puedan funcionar de forma autónoma, al tener un API y código de Python soportado oficialmente MISP se convirtió en el elegido para compartir y generar inteligencia.

Malcom

Debido a que en la arquitectura es necesario monitorear la red, una herramienta que permite analizar las comunicaciones de la red se vuelve bastante interesante de analizar.

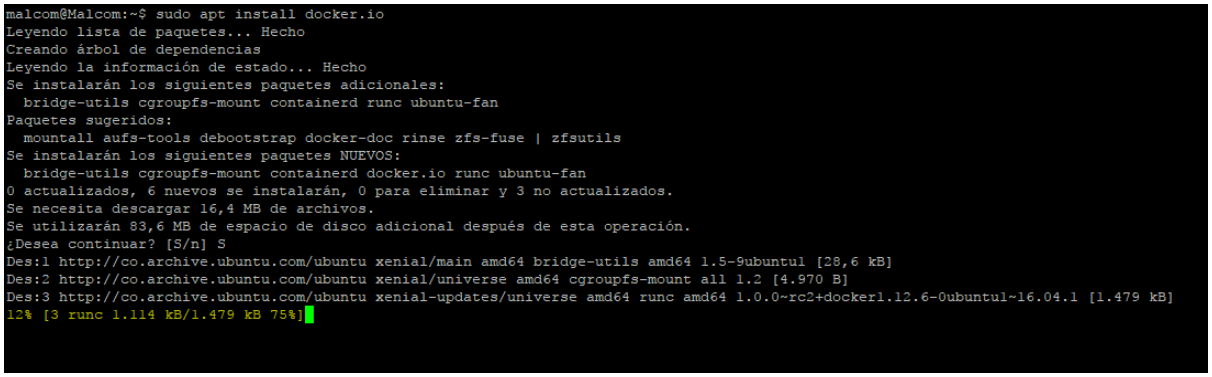
Instalación

Un primer intento de instalación se realizó usando Python3 e instalación por fuentes, pero problemas de versiones Malcom no pudo ser instalado por este

medio, es por esto que se instaló Docker[11] y se descargó una imagen estable de Malcom.

1. Instalar docker con sudo apt install docker.io

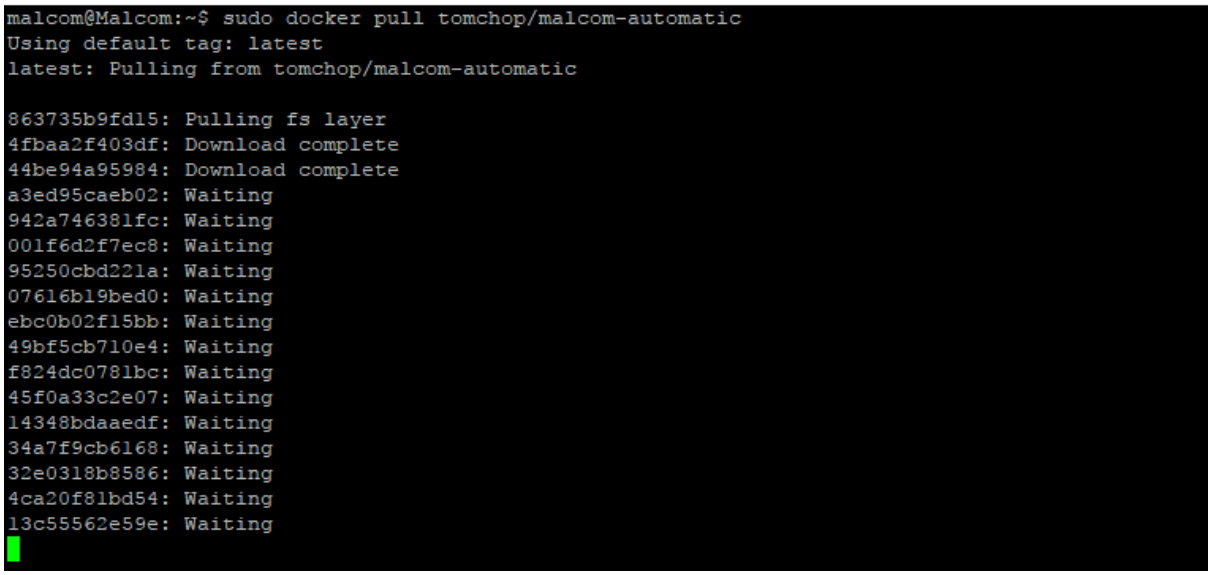
Figura 45. Instalación de Docker



Fuente: Propia

2.Descargar la instancia con sudo docker pull tomchop/malcom-automatic

Figura 46. Descarga de Malcom



Fuente: Propia

[11] https://www.docker.com/

Caso de uso

1. Ejecutar la instancia con sudo docker run -p 8080:8080 -d --name malcom tomchop/malcom-automatic

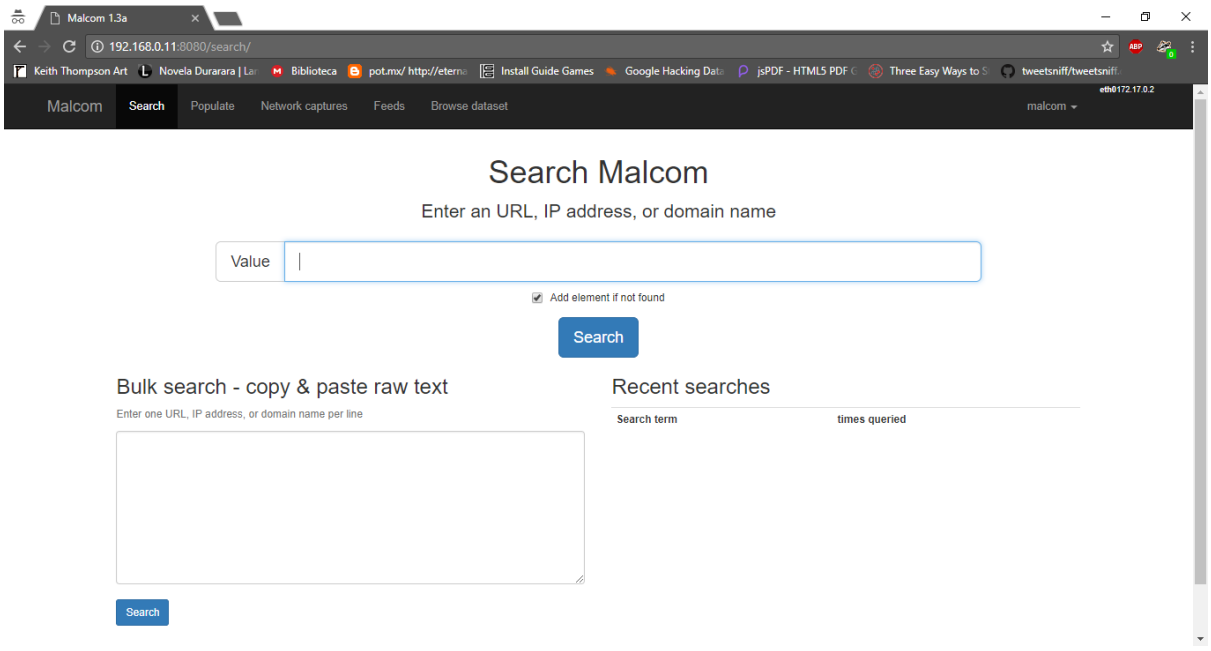
Figura 47. Ejecución de Malcom

```
malcom@Malcom:~$ sudo docker run -p 8080:8080 -d --name malcom tomchop/malcom-automatic
4c87279be5d6c164c81898fde55f144e849774dbf98d836798412c0b7f7c975c
malcom@Malcom:~$ _
```

Fuente: Propia

2. Ir a http://<host_malcom>:8080

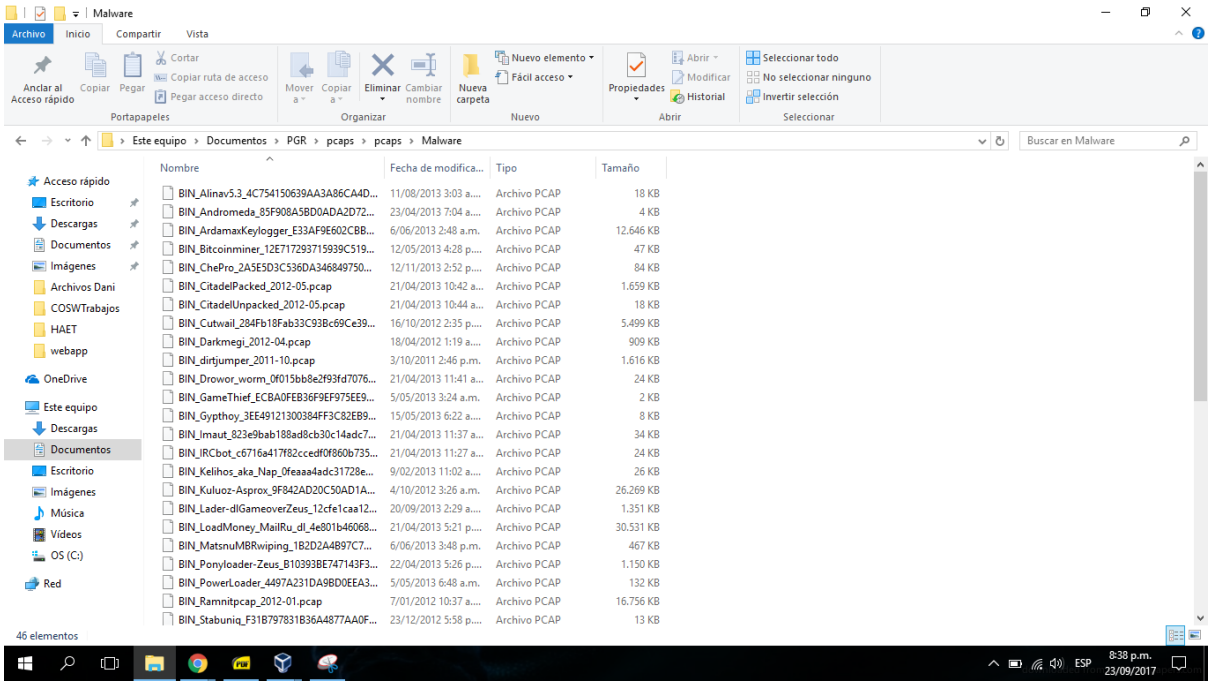
Figura 48. Página principal de Malcom



Fuente: Propia

3. Abrir los pcaps

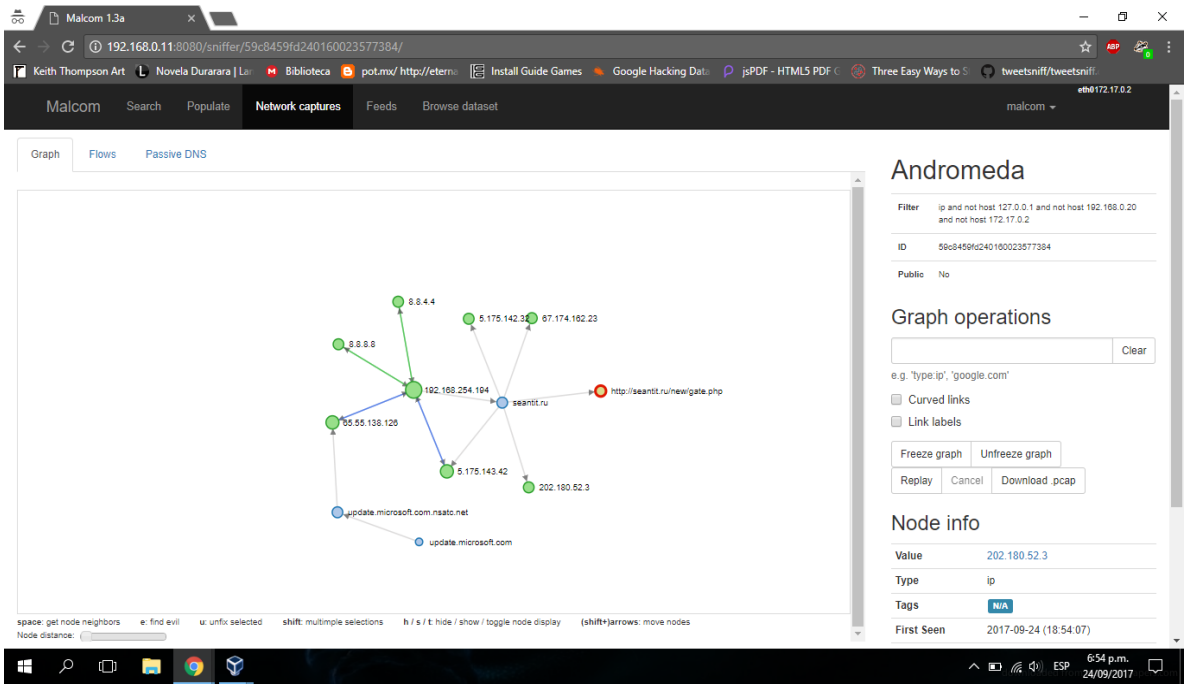
Figura 49. Pcaps de ejemplo



Fuente: Propia

4. Se hace la subida del pcap, se empieza con ejemplos de malware
5. El flujo de evento que se ha seguido y con el que se ha armado el grafo es el siguiente (Malware Andrómeda):
   Se mira el grafo de correlación

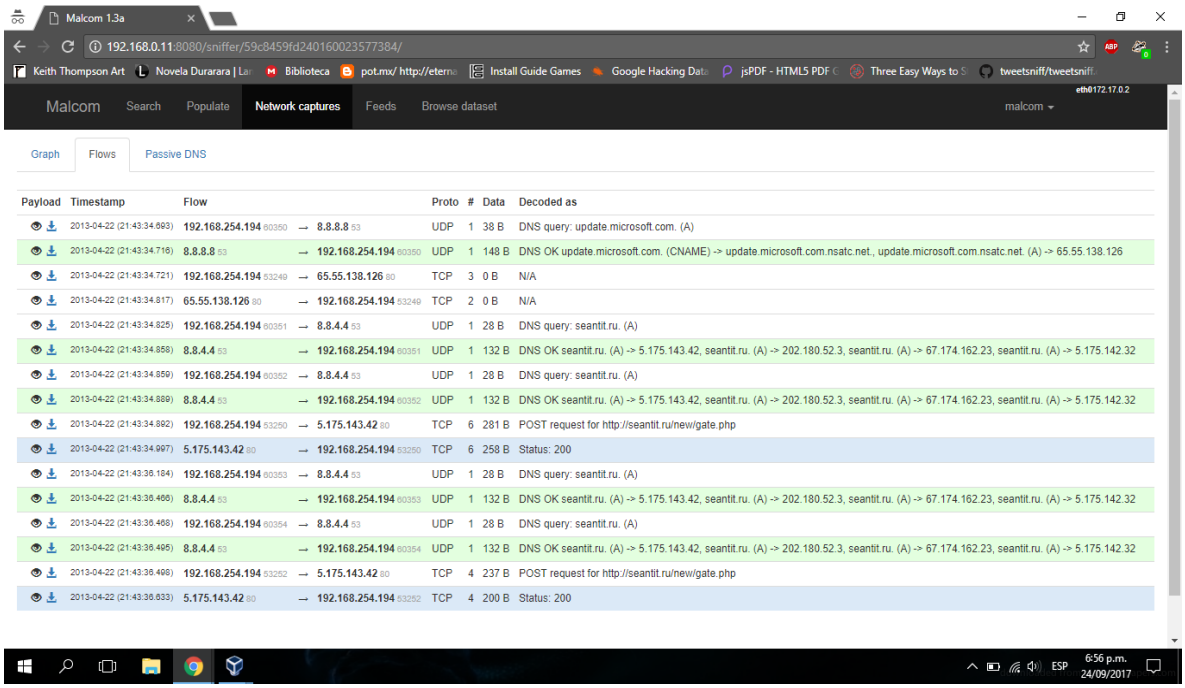
Figura 50. Grafo de Andrómeda



Fuente: Propia

Ahora se miran los flows de la figura 51 para entender de dónde salió el grafo.

Figura 51. Flow de información



Fuente: Propia

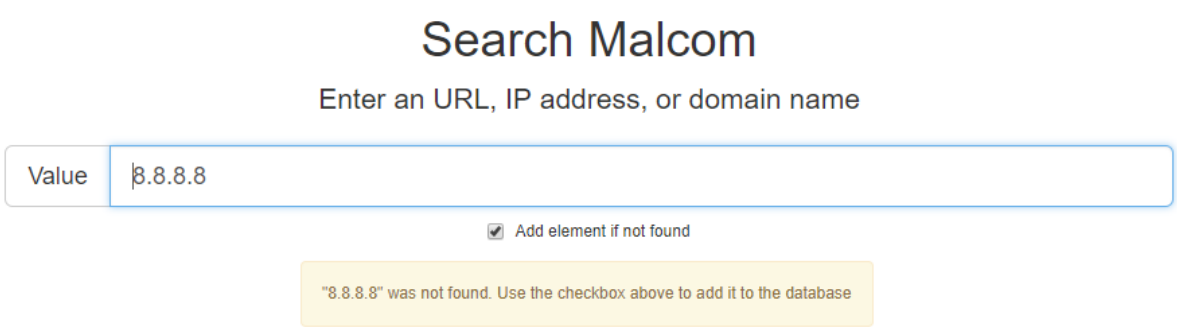
La máquina hace una consulta DNS a Google (8.8.8.8 y 8.8.4.4) sobre update.microsoft.com.La máquina establece una conexión con la ip retornada por el DNS. Luego de esto la máquina hace una solicitud DNS a http://seantit.ru y obtiene las direcciones ip seantit.ru. (A) -> 5.175.143.42, seantit.ru. (A) -> 202.180.52.3, seantit.ru. (A) -> 67.174.162.23, seantit.ru. (A) -> 5.175.142.32
Cuando hace la conexión se envía un post como se indica en la Figura 52. De esta forma se puede establecer que muy probablemente la máquina infectada está registrándose en el servidor de comando y control, las pistas son la URL del POST y el payload que este lleva, que puede ser una contraseña o identificación ante este servidor.

Figura 52. Información obtenida de Malcom

Fuente: Propia

6. Otra función de Malcom es la Búsqueda. Malcom busca en sus feeds el elemento, si no lo encuentra para Malcom no es un elemento maligno.

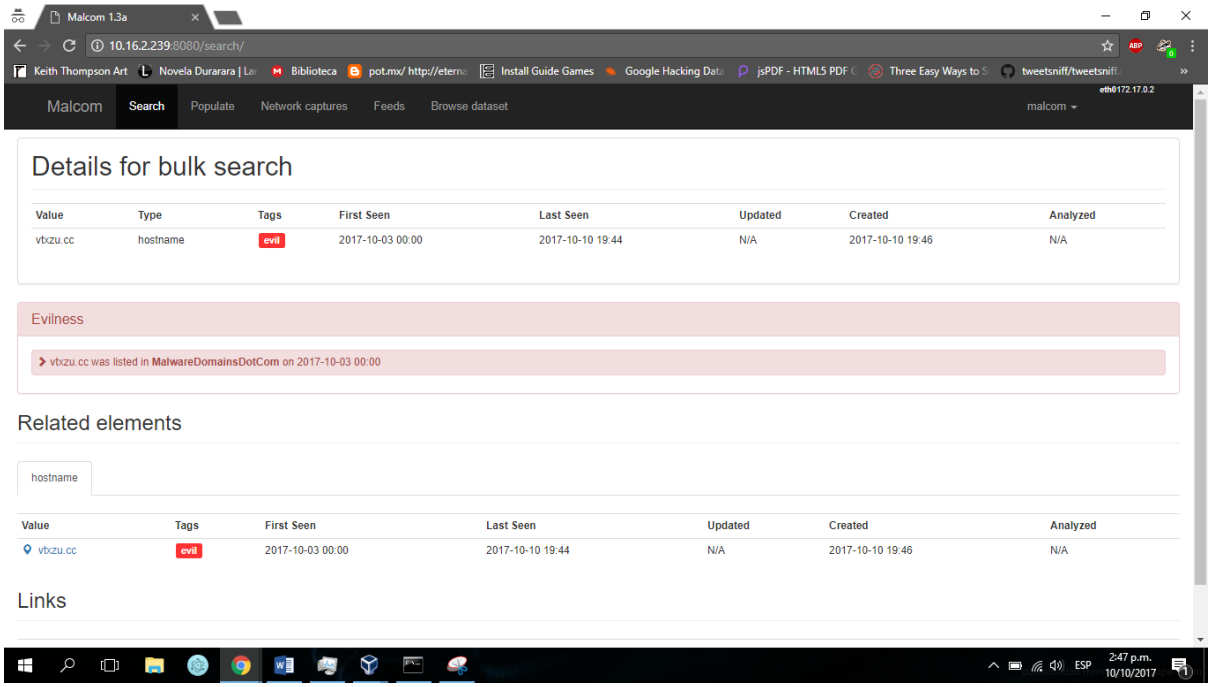
Figura 53. Elemento no existente



Fuente: Propia

En cambio, al buscar un elemento malicioso muestra la pantalla indicada en la Figura 54.

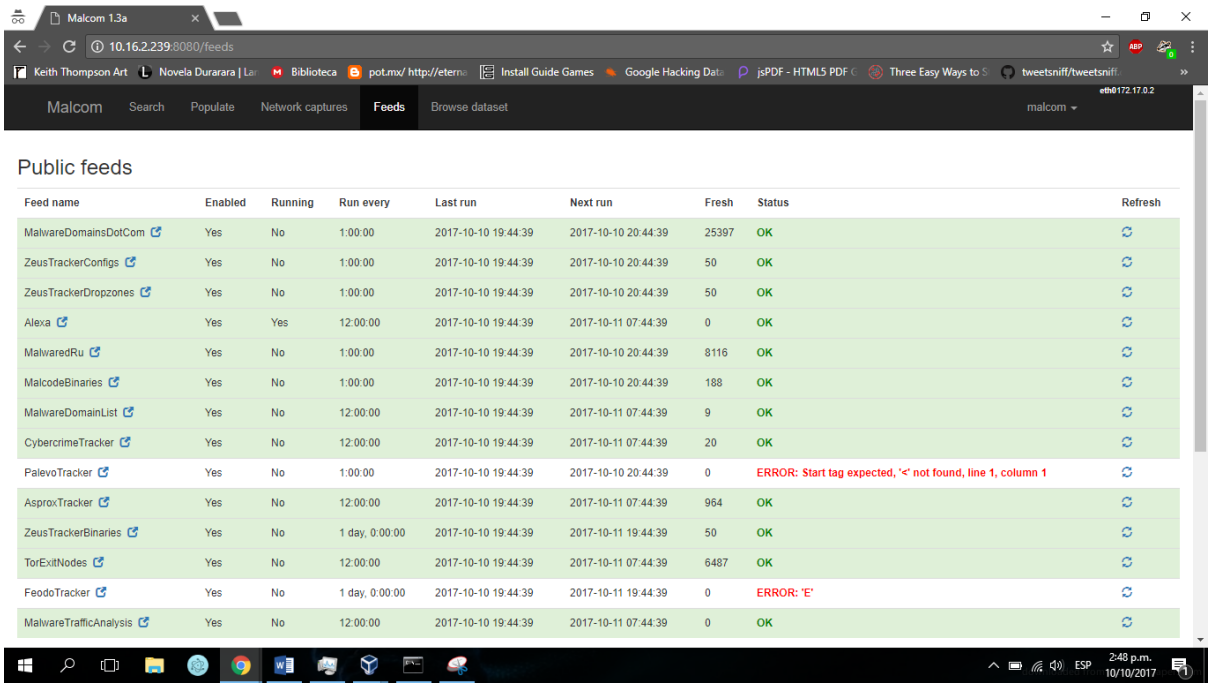
Figura 54. Elemento malicioso consultado



Fuente: Propia

En la sección “feeds” se muestran los sitios (Feeds) de confianza de los cuales Malcom está sacando información.

Figura 55. Feeds de Malcom



Fuente: Propia

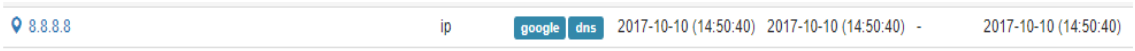
La función “populate data” sirve para añadir datos a Malcom, puede hacerse de uno a uno, o varios (uno por linea) o en un archivo que debe seguir este formato

Figura 56. Populate data en Malcom

```
<element>;<tag1>,<tag2>,<tag3>
google.com
http://evildrop.me/drop.php; citadel, evil
127.0.0.1; localhost, home
```

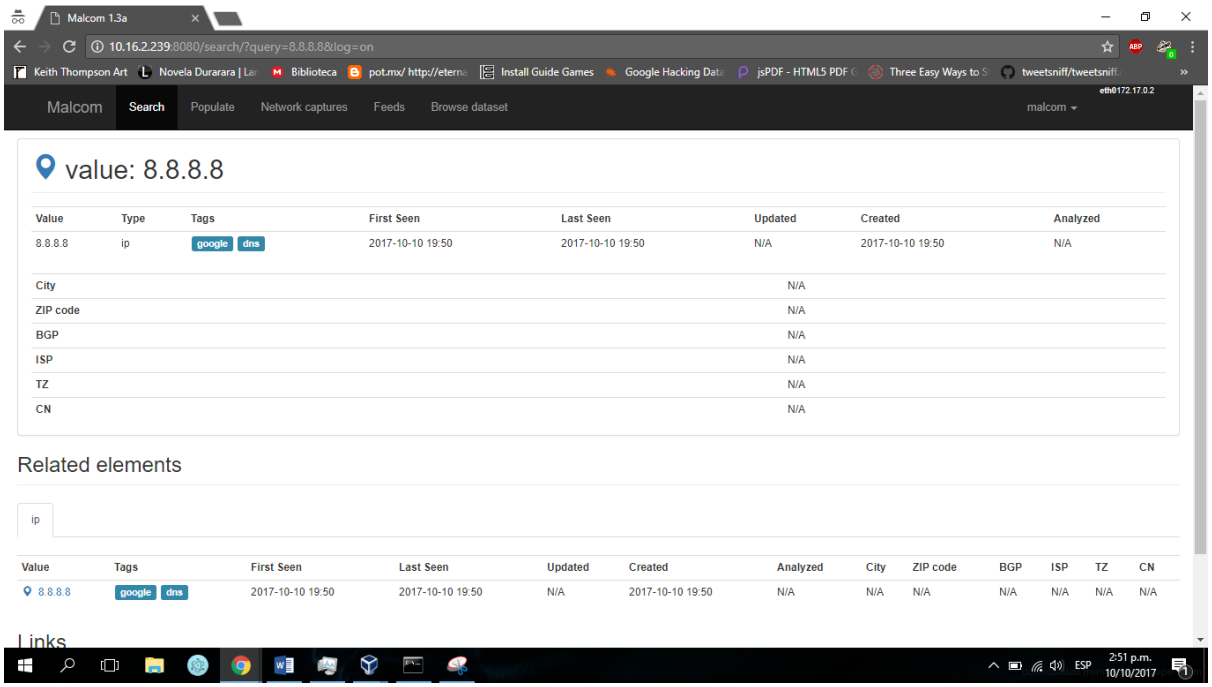
Fuente: Propia

Al añadir un nuevo elemento este se ve añadido en el browse datalist de Malcom



Y al consultarlo se muestra cómo se indica en la Figura 57.

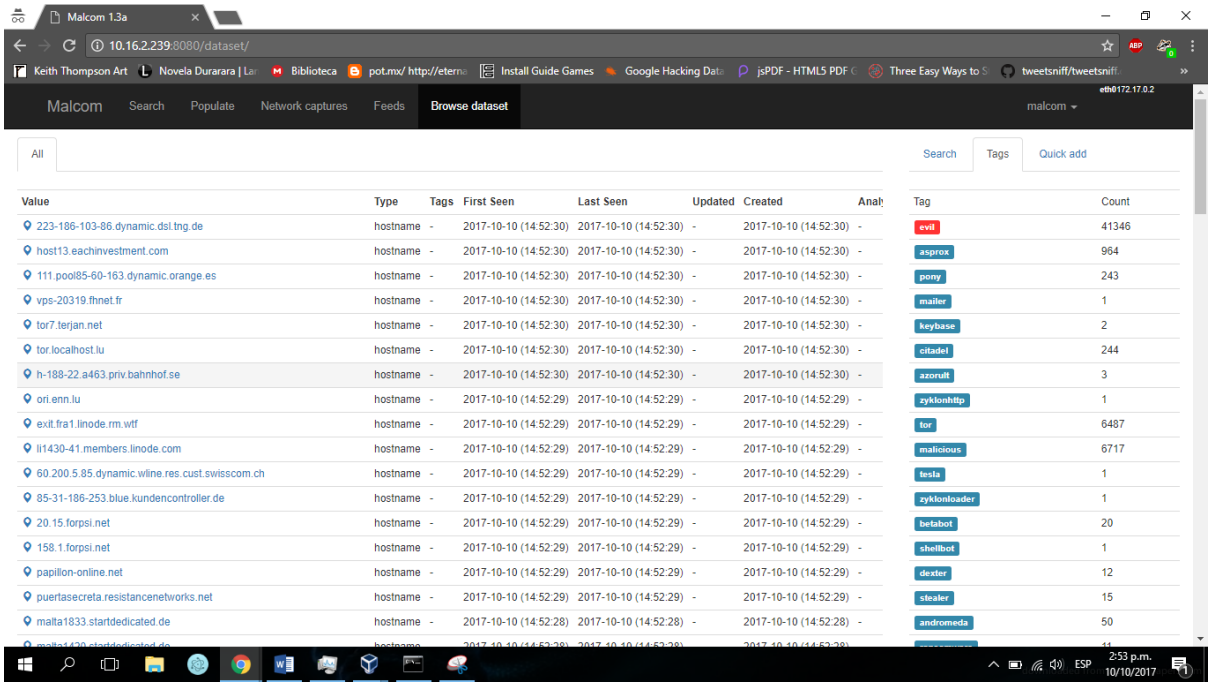
Figura 57. Elemento añadido



Fuente: Propia

La sección “datalist” de Malcom permite organizar por etiquetas, buscar directamente o añadir nuevos elementos que deben ser considerados maliciosos.

Figura 58. Datalist de Malcom



Fuente: Propia

Añadir un feed es un tanto engorroso, pues toca modificar el código fuente y añadirlo directamente, más información en https://github.com/tomchop/malcom/wiki/Adding-feeds-to-Malcom

Como se ha podido ver Malcom permite analizar a fondo la actividad en red de una muestra, interceptando peticiones de comando y control. Este monitoreo y análisis de la red puede resultar en

Malcom es una herramienta interesante y con gran potencial en la ciber inteligencia, en un futuro puede volverse a ella para analizar eventos de la red, más en este momento no permite compartir tanta información y correlacionarla como si lo hace MISP, de modo que se dejó de lado, siempre con posibilidad de volver a usarla, otro inconveniente encontrado es la poca documentación (Tanto oficial como externa) perteneciente a la herramienta que dificulta su dominio.

Radare2

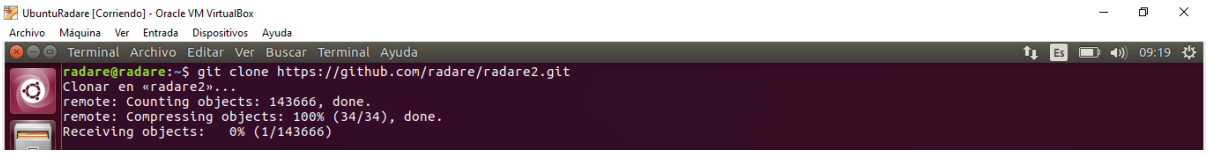
Su uso sencillo de consola, junto con la modularidad que ofrece esta herramienta, son bastante atractivos en el momento de buscar automatización en el proceso de análisis de malware.

Instalación

Al ser open source y una herramienta en constante evolución, es necesario instalar desde las fuentes, se requiere acceso a internet puesto que hay submódulos que se actualizan en la instalación.

1. Clonar el repositorio de radare con “git clone https://github.com/radare/radare2.git”

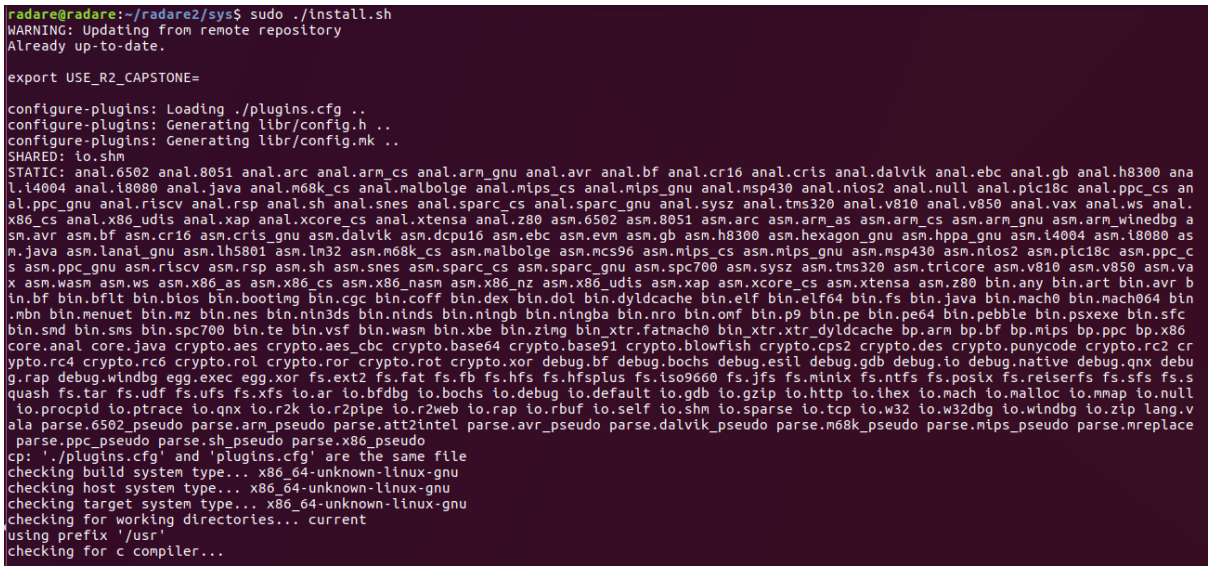
Figura 59. Descarga de Radare



Fuente: Propia

2. Entrar a radare2/sys y ejecutar el script de instalación con “sudo ./install.sh”

Figura 60. Instalación de Radare



Fuente: Propia

Radare2 ofrece distintas funcionalidades, dentro de las que se encuentran:

- rahash2 para obtener el hash de una muestra
- rabin2 para obtener los strings que están el código de una muestra
- rafind2 para hallar información de interés como datos cifrados
- r2 malware.exe y luego:
    - “il” para ver las librerias vinculadas (Linked libraries)
    - “ie” para ver los puntos de entrada (Entrypoint)
    - “iS” para ver las secciones del binario
    - “ii” para ver los imports
    - “p=e | more” para visualizar la entropía de la muestra. Esta visualización es sustancialmente interesante cuando la entropía es muy grande (hay secciones casi vacías mientras que otras tienen una cantidad de código muy grande).
    - “aaa” para hacer un análisis estático por default
    - “VF” para iniciar el modo visual
        - “_” para ejecutar una búsqueda en el binario

Como se ha podido ver Radare2 permite obtener información que identifica fácilmente a un archivo, junto con un análisis a bajo nivel de la memoria y recursos utilizados, por lo que se decidió utilizar esta herramienta por su capacidad de obtener información automáticamente, además de poder ser integrado fácilmente con Yara, pudiendo automatizar la aplicación de las reglas.

ImmunityDBG | OllyDBG

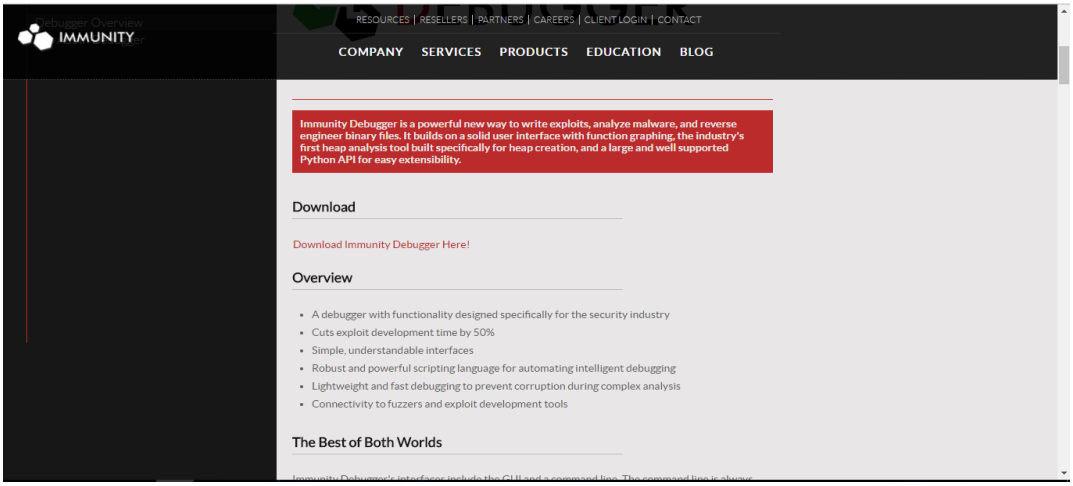
Estas herramientas permiten la ejecución de la muestra en un entorno de debbug y obtener información confiable acerca de su comportamiento y configuración a muy bajo nivel, resultando en un análisis más acertado de la misma.

Instalación

Al ser un programa diseñado para Windows, la instalación es bastante sencilla y solo requiere seguir los pasos en la página oficial.

1. Entrar al link https://www.immunityinc.com/products/debugger/

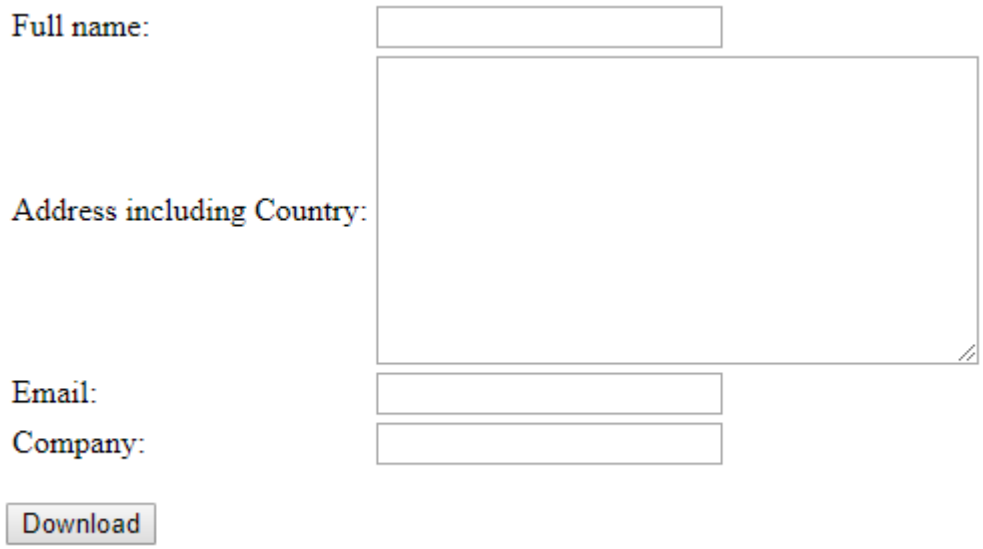
Figura 61. Descarga de InmunityDBG



Fuente: Immunity

2. Darle a download y llenar el formulario

Figura 62. Registro en InmunityDBG



Fuente: Propia

3. Una vez descargado ejecutarlo y seguir las instrucciones, de no tener python instalado Immunity lo instalará.

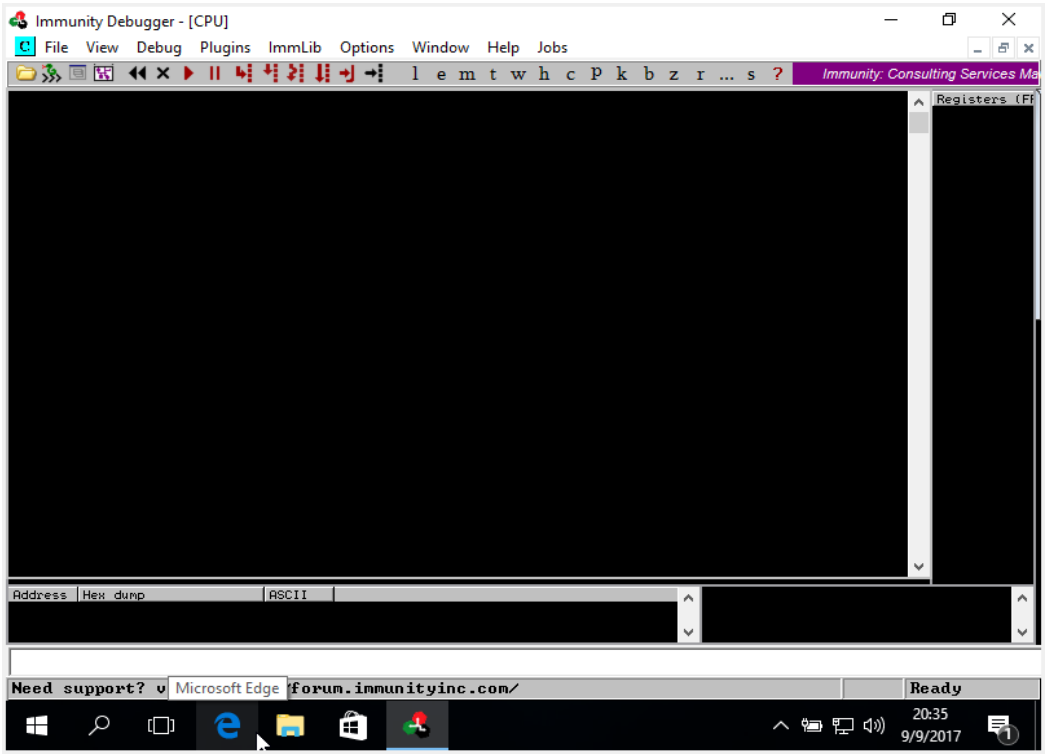
   Caso de uso

   Immunity está basado en OllyDBG y aparte de tener sus mismas funcionalidades incluye nuevas formas de analizar el código. Por esta razón ambos debugger serán tratados como el mismo

   Para hacer esto el Windows defender y antivirus deben estar desactivados

1. Abrir el Immunity debugger.

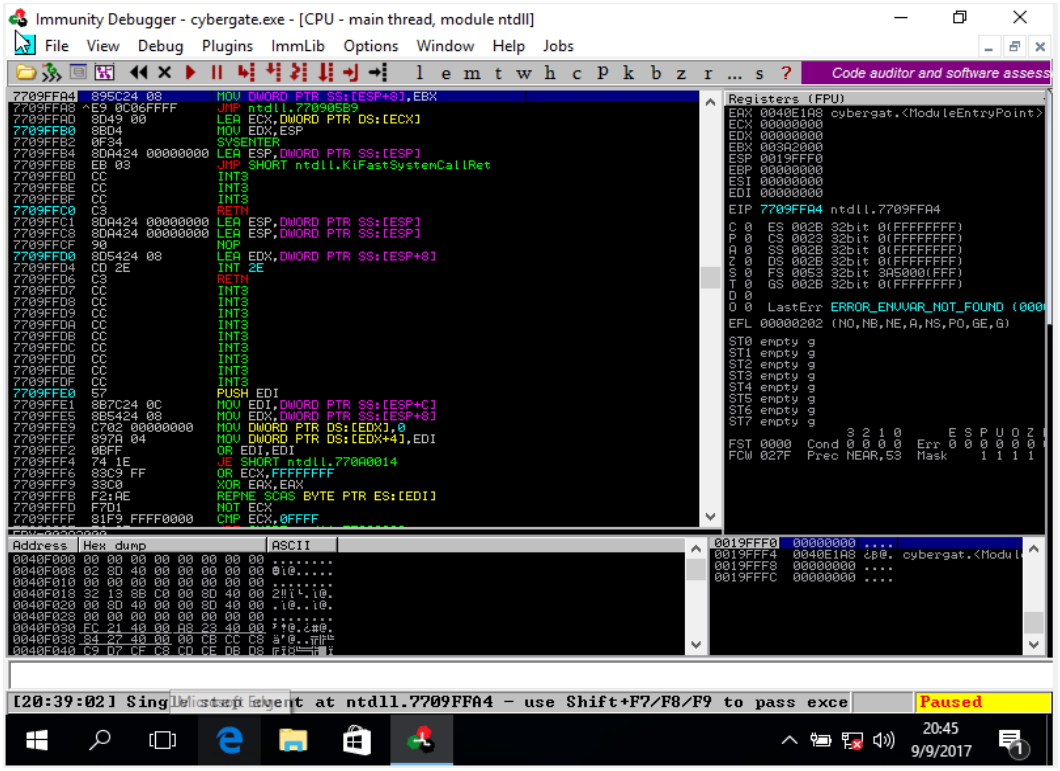
Figura 63. Vista de Inmunity Debugger



Fuente: Propia

2. Abrir la muestra del malware elegido.

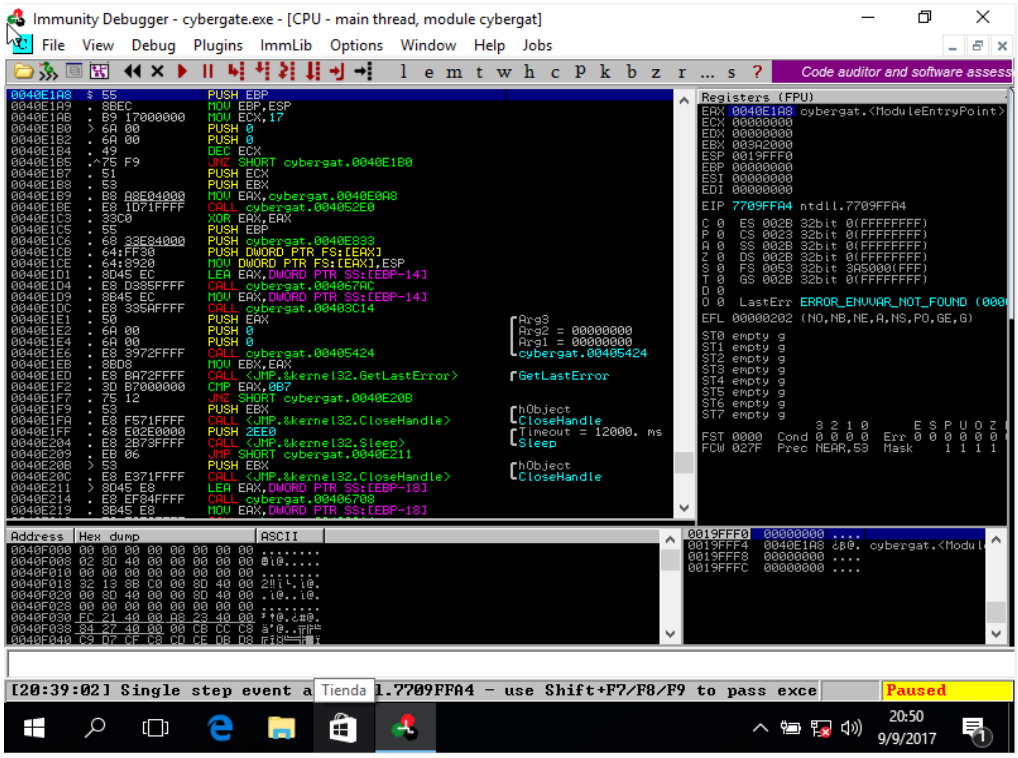
Figura 64. sample.exe



Fuente: Propia

3. Seleccionar el entry point. Este entry point puede obtenerse de herramientas como Radare2.

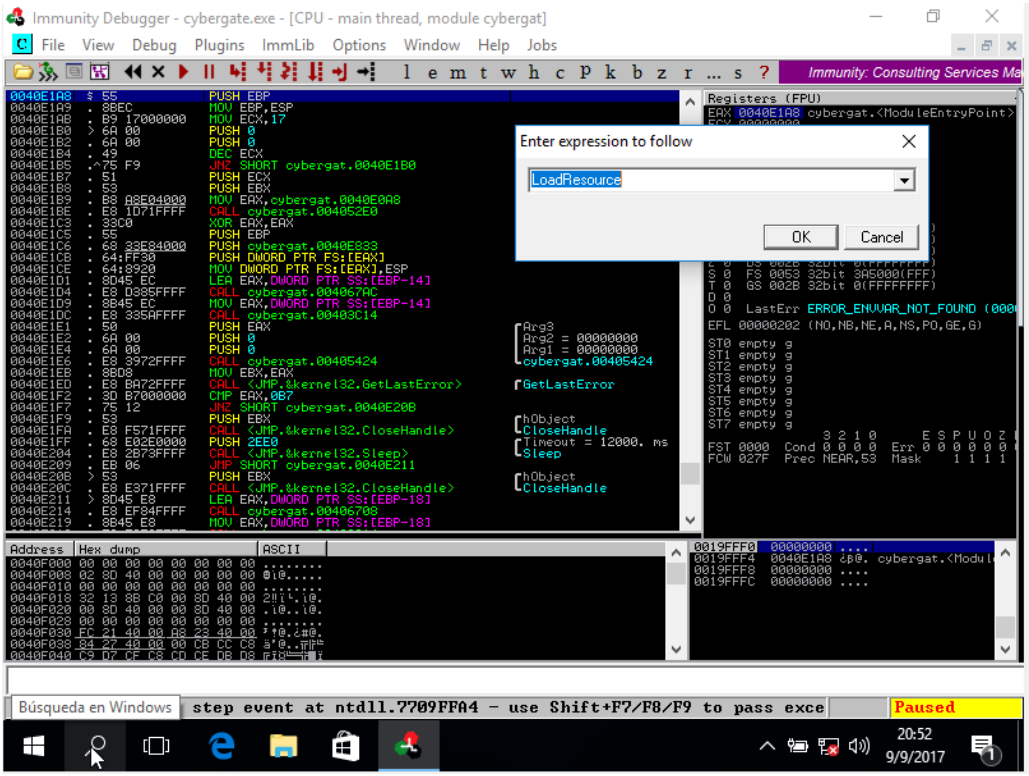
Figura 65. Entrypoint de la muestra



Fuente: Propia

4. Con ctrl+g aparece una ventana para buscar expresiones, buscar “LoadResource”.

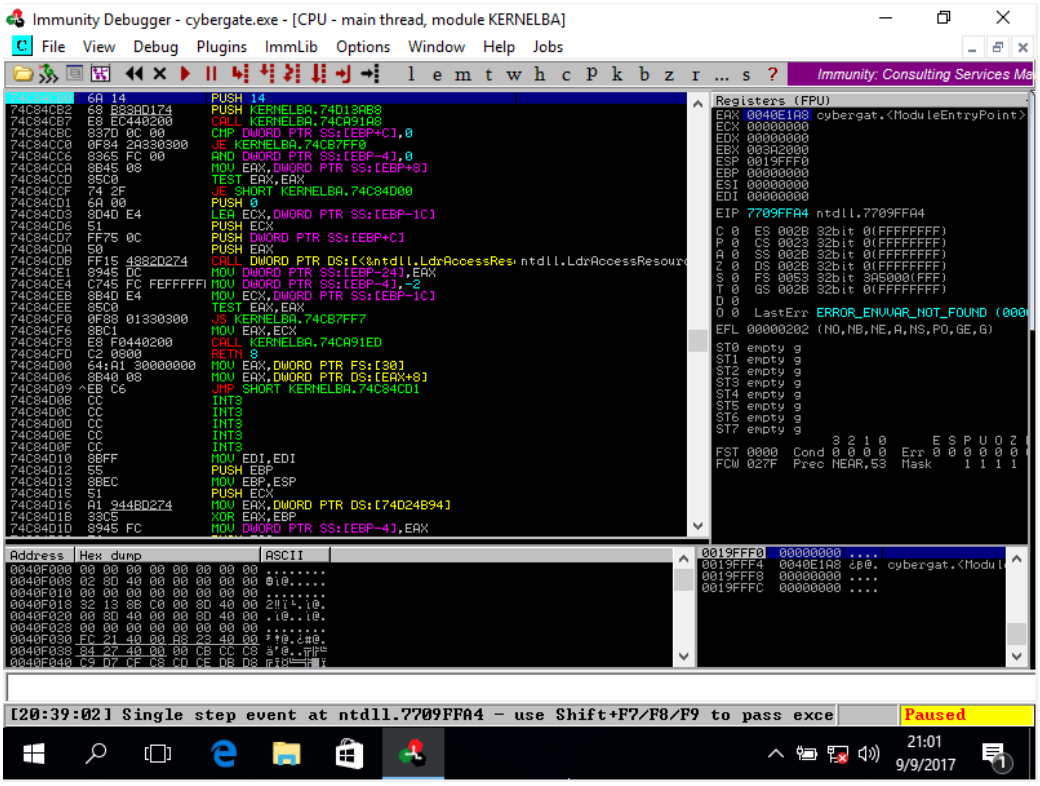
Figura 66. Búsqueda de carga de recursos



Fuente: Propia

5. Entrar a la expresión y al darle f2 se crea un breakpoint.

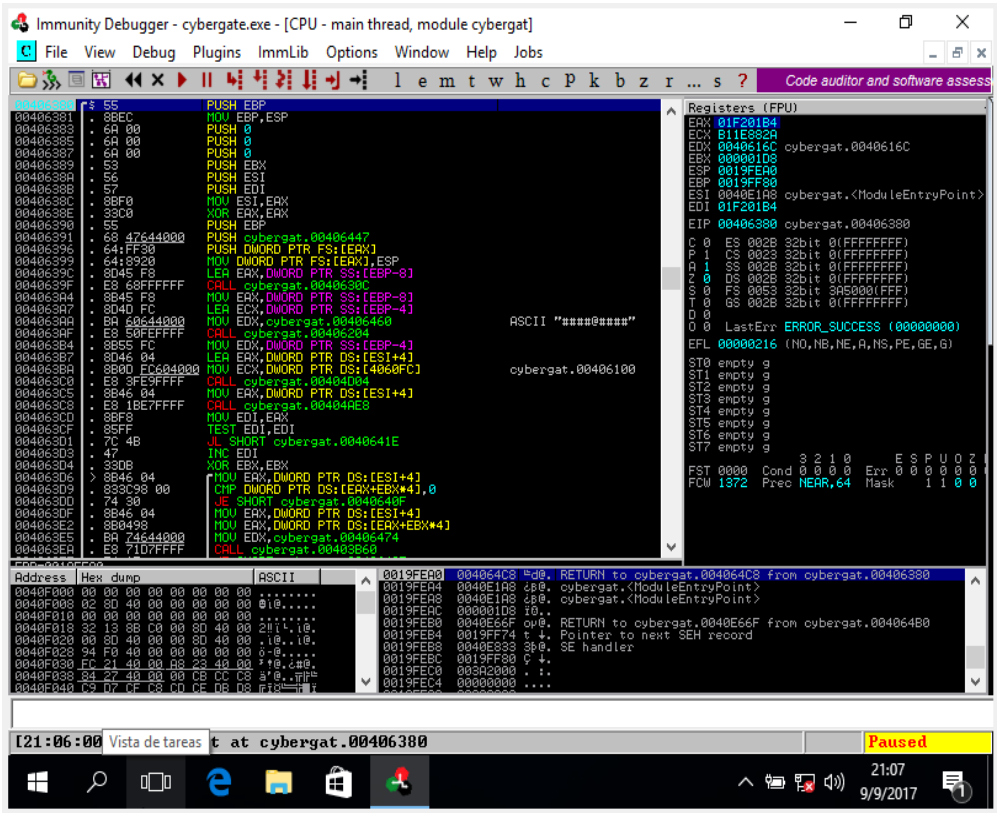
Figura 67. Breakpoint creado



Fuente: Propia

6. Con f9 se ejecuta hasta llegar al breakpoint definido.

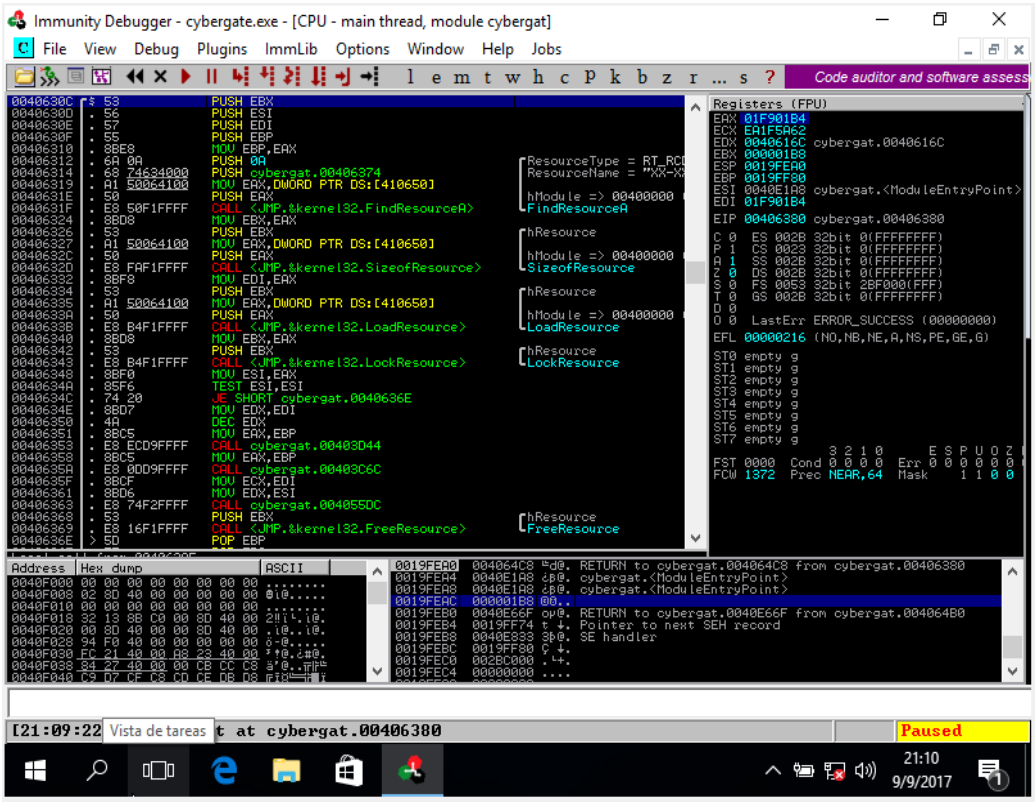
Figura 68. Ejecución del breakpoint



Fuente: Propia

7. Al analizar la primera función se ve la carga de recursos.

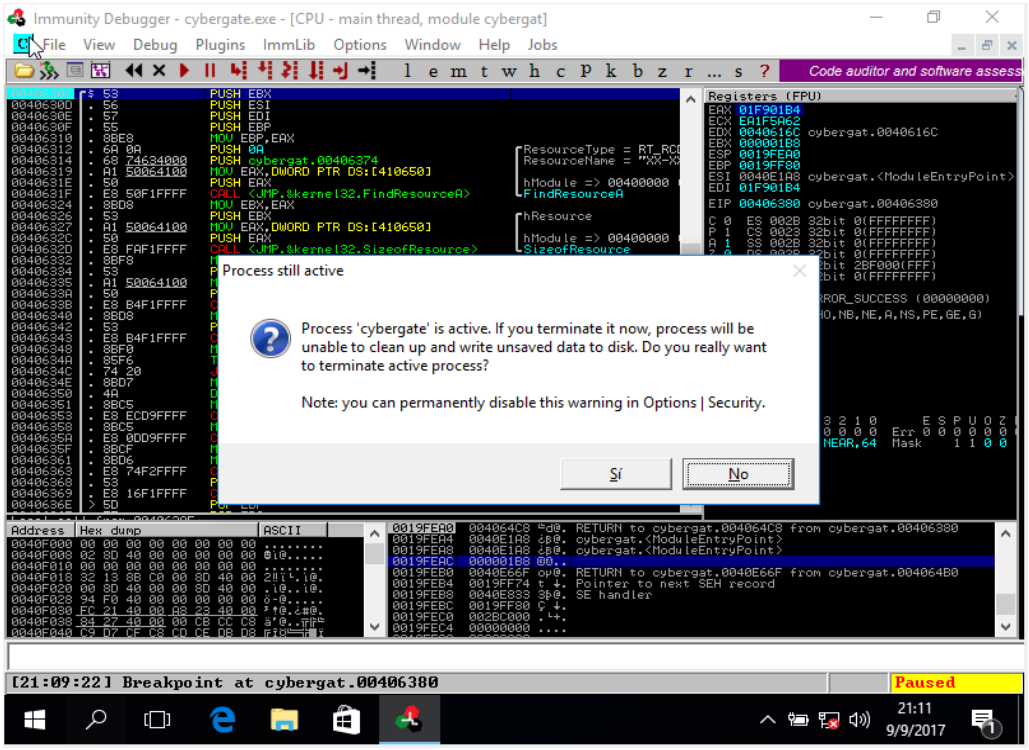
Figura 69. Carga de recursos en la ejecución



Fuente: Propia

8. Añadir un breakpoint en esta función y reiniciar el programa con ctrl+f2.

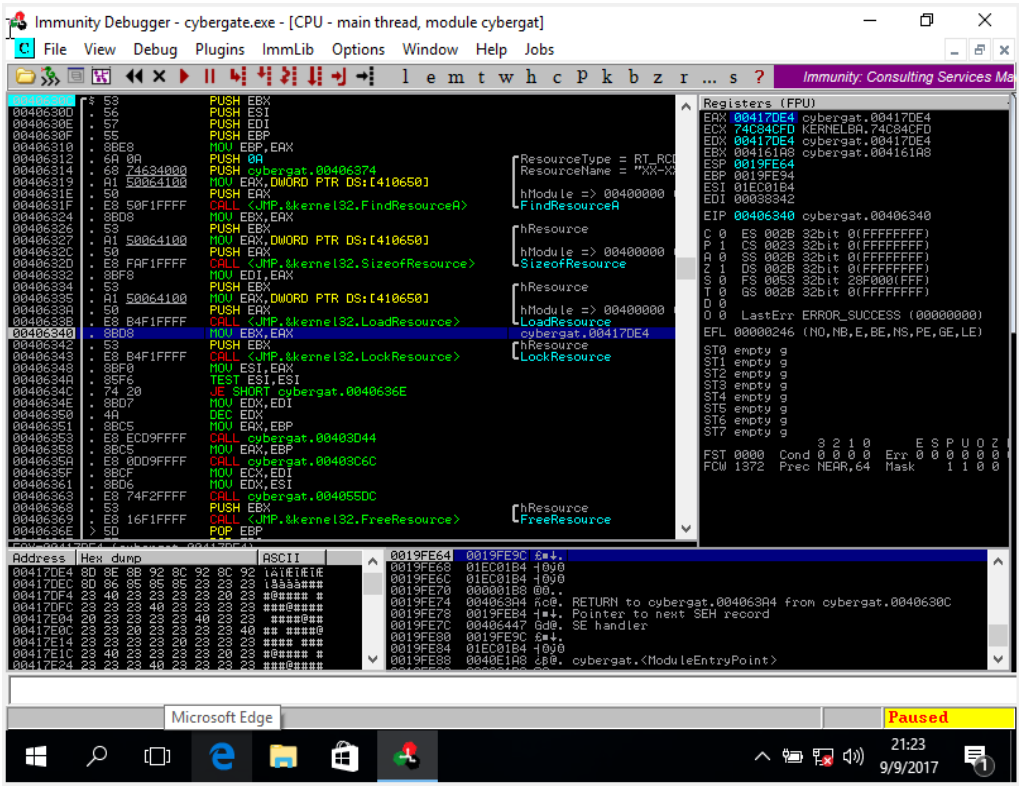
Figura 70. Nuevo breakpoint



Fuente: Propia

9. Ejecutarlo hasta llegar a ese breakpoint, luego ejecutar paso a paso con f8 hasta llegar a la funcion load resource. En EAX se verá la dirección de memoria del contenido cifrado.

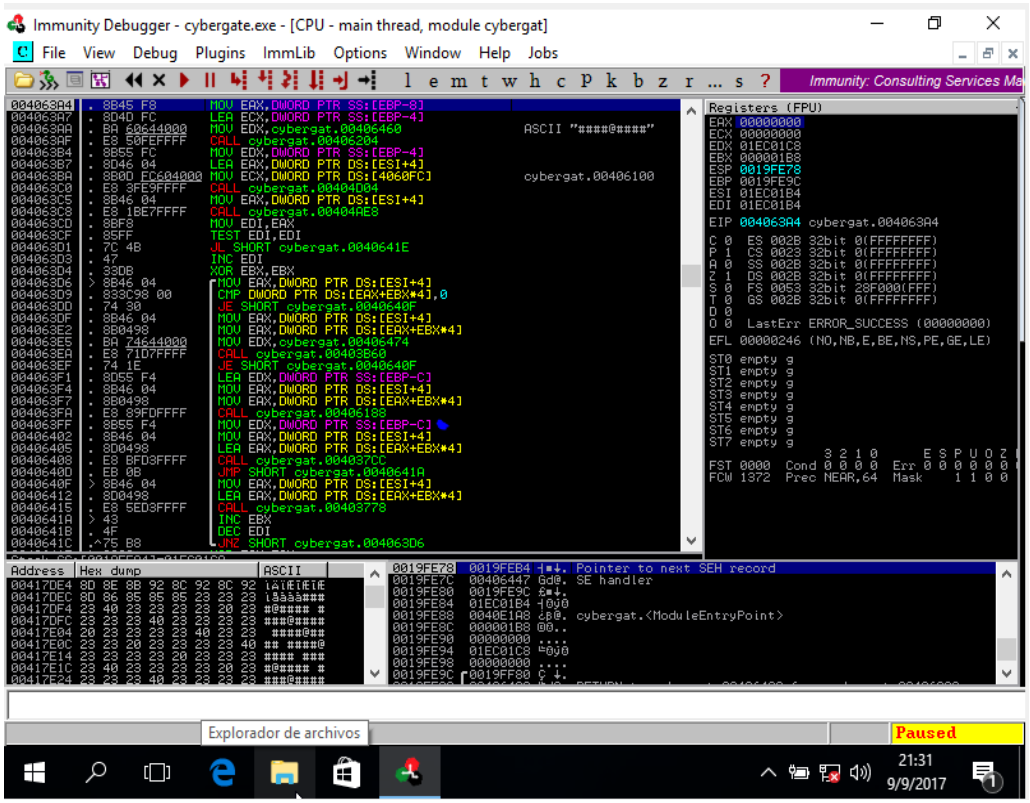
Figura 71. Dirección de memoria del cifrado



Fuente: Propia

10. Ya habiendo visto la configuración se continúa la ejecución en el bucle para descubrir la configuración.

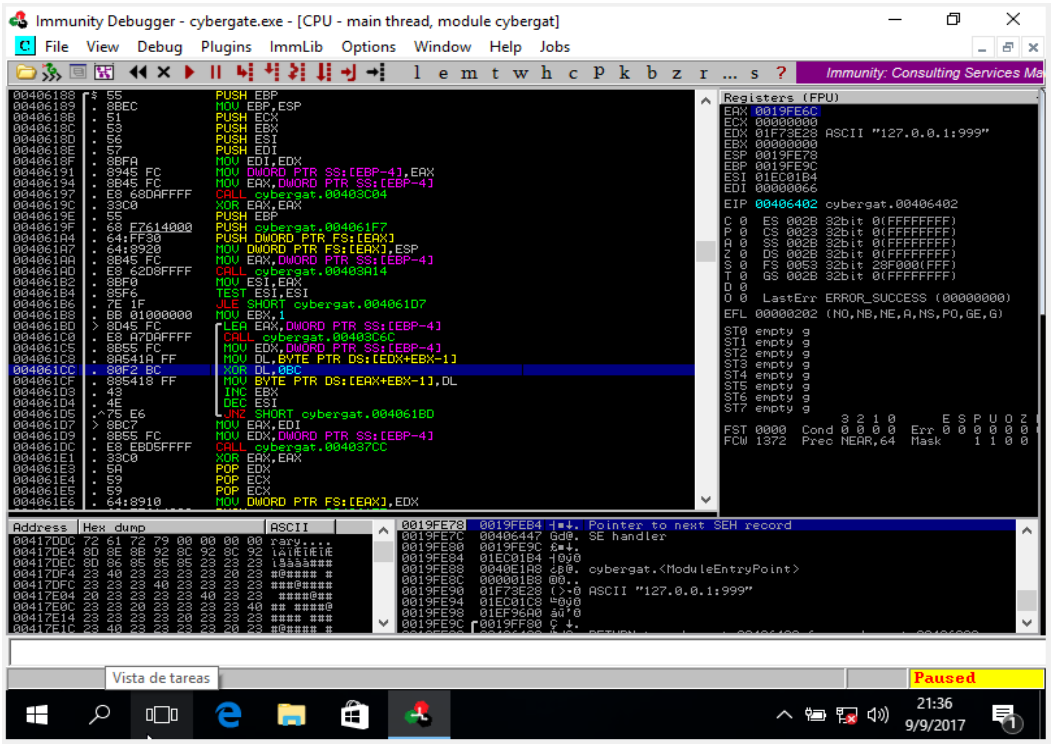
Figura 72. Rutina de configuración



Fuente: Propia

11. Eso significa que la función anterior es la que revela la configuración. El 0xBC es la clave del XOR. En esta función se está haciendo el descifrado de la configuración que utiliza el malware para funcionar.

Figura 73. Clave de la configuración

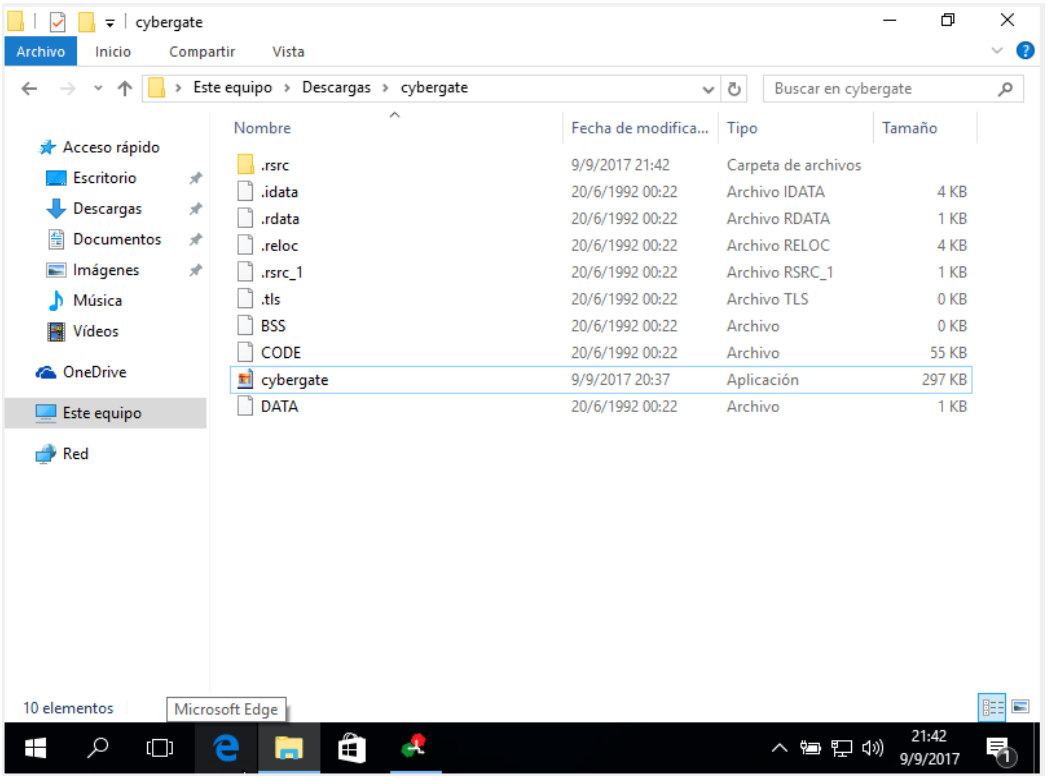


Fuente: Propia

12. En ese bucle se encuentra la clave para el resto de la configuración

13. Con 7zip extraer el .exe.

Figura 74. Extracción del ejecutable

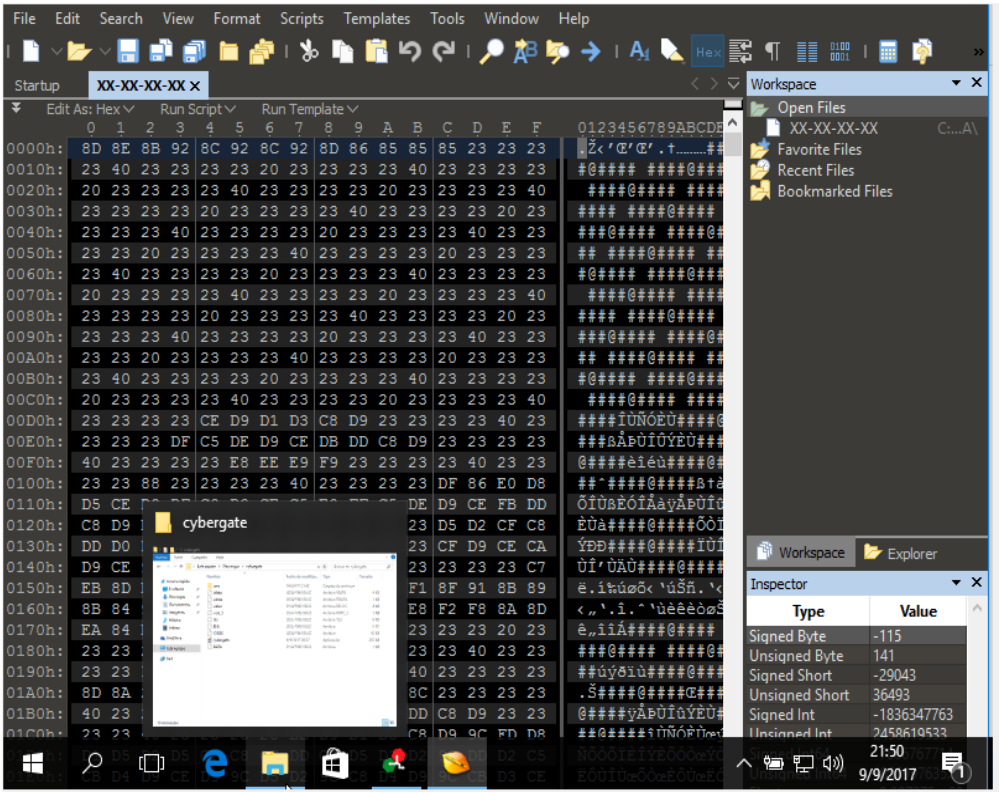


Fuente: Propia

14. Abrir ".rsrc/RCDATA/XX-XX-XX-XX" con 010 editor[12]. En la carpeta .rsrc se encuentran los recursos que utiliza el código, incluyendo los archivos de configuración, el archivo cifrado es XX-XX-XX-XX.

[12] https://www.sweetscape.com/010editor/

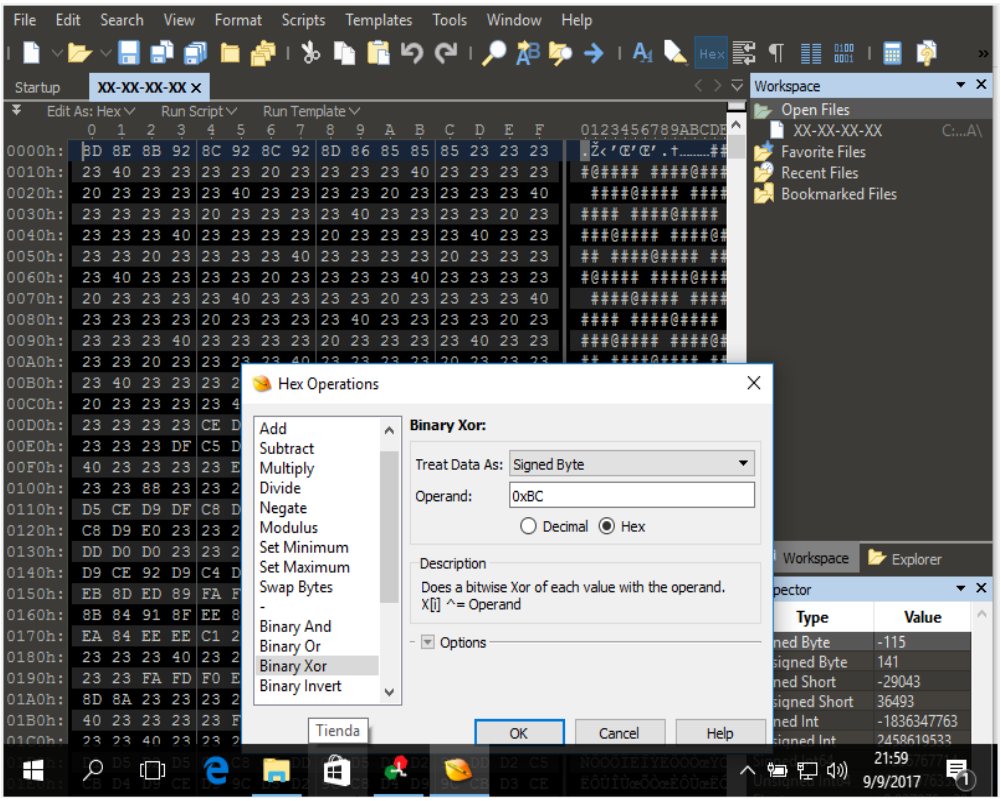
Figura 75. 010 editor



Fuente: Propia

15. Hacer la operación XOR con la clave encontrada anteriormente.

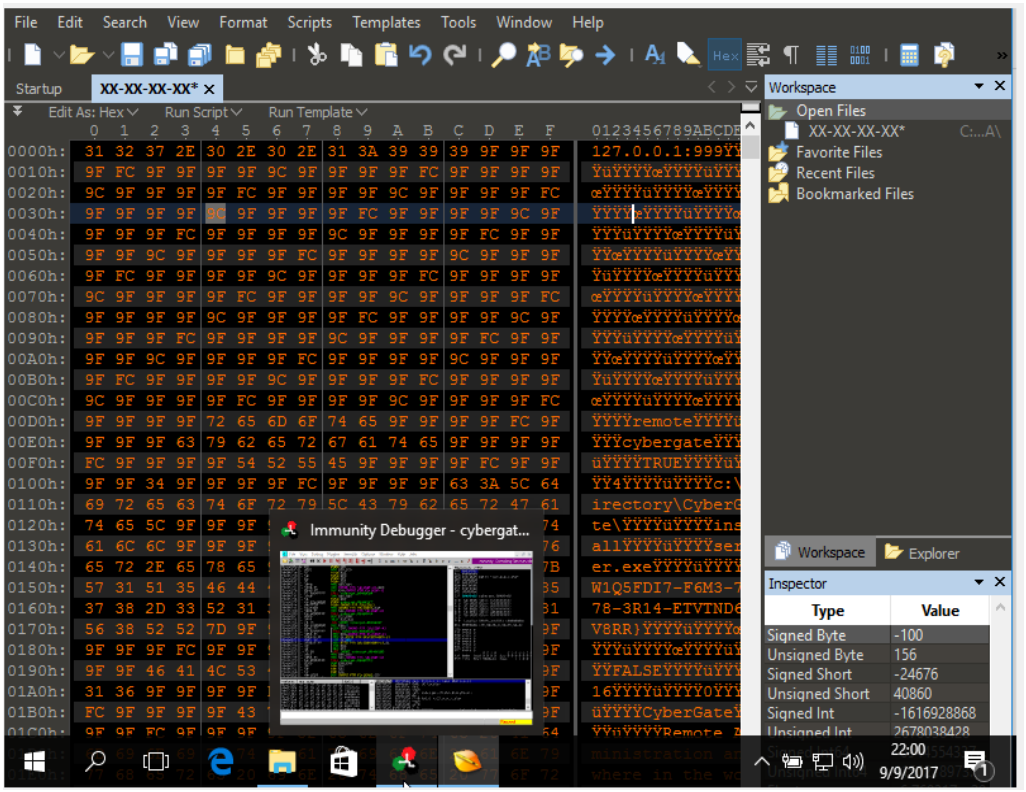
Figura 76. Descifrado de la configuración



Fuente: Propia

16. Como se puede ver, al hacer XOR se revela toda la información referente a la configuración del malware, incluyendo la dirección IP del servidor de comando y control.

Figura 77. Configuración en texto plano



Fuente: Propia

17. Con este script y conociendo la clave de XOR se puede automatizar el proceso

Como se ha podido ver OllyDBG e InmunityDBG permiten analizar a gran profundidad el código de una muestra, el poder extraer conclusiones usando los estados de la memoria en distintas partes de la ejecución del mismo convierten esta herramienta en algo con un gran potencial de análisis.

Pese a los beneficios que ofrecen estas herramientas la automatización no fue posible, de modo que para el centinela no eran apropiadas y fueron desechadas.

DISEÑO DE UN CENTINELA IOT

Para el diseño e implementación del centinela se tomaron en cuenta las siguientes características esperadas:

- Colaborativo: Los centinelas deben compartir información de amenazas entre ellos.
- No invasivo: Las funciones deben interactuar lo menos posible con el usuario.
- Adaptable: El centinela debe adaptar sus defensas a amenazas conocidas o amenazas que sean recibidas de forma colaborativa.
- Automático: Las acciones que tome el centinela deben estar lo más automatizadas posible.
- Fácilmente desplegable: Poca configuración es necesaria en el proceso de despliegue del centinela.

Los archivos deben ser capturados mediante los monitores en la red, y luego de ser capturados deben ser enviados a evaluación en el centinela.

Se propone entonces un centinela IoT que sigue una estructura de anillos. Cada anillo está formado por una o más herramientas de seguridad integradas. El orden en el cual una nueva muestra es evaluada, que corresponde al orden de los anillos, es:

1. Análisis interno (Reglas de Yara)
2. Modelo de machine learning basado en permisos.
3. Análisis externo (VirusTotal)

El diagrama de componentes en la Figura 96 ilustra las herramientas y funciones genéricas del centinela, junto con la estructura general de los anillos. Como se puede ver, el centinela cuenta con distintos módulos, siendo cada módulo una herramienta con una función específica. Los módulos están divididos según su función.

Monitoreo

Estos módulos se encargan de monitorear la red en busca de amenazas, ya sean archivos descargados o vulnerabilidades conocidas en dispositivos conectados, estos módulos son:

- Monitor Ethernet: Este módulo monitorea y detecta intrusiones en el estándar de comunicación ethernet. La implementación de este módulo debe ser capaz de capturar archivos en tiempo real y reensamblarlos, además de ser poder detectar comportamientos sospechosos y alertar para futura investigación.
  La implementación de este módulo se realizó usando Suricata, aparte de las reglas de detección de intrusos se añadió configuración para el reensamblado y redirección de archivos correspondientes a aplicaciones de IoT hacia la carpeta donde el centinela puede recolectarlos y analizarlos.

- Monitor Wifi: Este módulo monitorea y detecta intrusiones en el estándar 802.11. La implementación de este módulo debe ser capaz de capturar archivos en tiempo real y reensamblarlos, además de ser poder detectar comportamientos sospechosos y alertar para futura investigación. En la implementación se usa Kismet para detectar intrusiones a nivel de capa 1 y capa 2, el reensamblado de archivos y envío se realiza usando Suricata sobre la interfaz de red correspondiente a Wifi.
- Escaner de vulnerabilidades: Este módulo busca dispositivos vulnerables en la red y envía la información recolectada a un SIEM, que correlaciona esta información con la que le llega de los monitores de la red, de esta forma puede detectar ataques a la red. La implementación usada es OpenVas.

Análisis

Estos módulos se encargan de realizar los análisis en el centinela y devolver una evaluación, mediante la cual se inicia un proceso de reporte.

- Analizador externo: Este módulo recibe las muestras desde el centinela, las envía a un servidor externo a ser evaluadas y usando la respuesta del servidor determina si la muestra es un malware. La implementación usada es VirusTotal, en esta implementación la respuesta contiene el número de motores que detectaron la muestra. Esta muestra es considerada maliciosa si al menos un motor de antivirus la detectó.
- Analizador interno: Este módulo recibe la muestra y la analiza en el centinela usando análisis estático, retorna al evaluador si considera que la muestra es un malware. La implementación usada es Reglas de Yara, en esta implementación la respuesta contiene cuantas reglas detectaron la muestra evaluada. Esta muestra es considerada maliciosa si al menos una regla detectó la muestra.
  La necesidad de aplicar reglas de forma automática para la autonomía del centinela hizo que se examinara Yara como una posible solución a este problema, siempre viendo por la capacidad de usar la información del entorno para generar reglas de forma automática, o de descargar reglas generadas mediante ciberinteligencia por otra persona.
  La capacidad de usar Yara de forma automática, junto con la integración hecha con Radare2 y una posible integración con Cuckoo para vigilancia de comportamiento hizo que esta herramienta se añadiera al centinela y fuera parte esencial del proceso de evaluación en el mismo.

Herramientas

Estos módulos son herramientas varias usadas por el centinela.

- Reversor: Este módulo hace ingeniería inversa sobre una muestra y obtiene todo tipo de información que pueda obtenerse mediante el análisis estático del código. En este caso se tienen dos implementaciones, la primera es Radare2 y la segunda es Androguard.

En la evaluación se pudieron usar módulos individuales y esto resultó en un potencial de automatización para un modelo de machine learning. Siendo que tanto el modelo como los datos fueron construidos en Python el uso de módulos individuales de Androguard para extraer la información junto con la existente integración con las reglas de Yara, esta herramienta se añadió al centinela.

- Reportero: Este módulo genera un reporte de la muestra usando información del Reversor y envía el evento al servidor de compartición de información. Este módulo es implementado en Python 3.6 como código desarrollado.
- Evaluador: Este módulo obtiene los archivos de los monitores y los envía a lo largo de los anillos solicitando evaluación. Si algún anillo detecta el archivo como malicioso el evaluador inicia el proceso de reporte. Este módulo es implementado en Python 3.6 como código desarrollado.
- Generador de alertas: Este módulo genera una alerta al usuario acerca del malware detectado y el anillo que realizó la detección. Este módulo es el único que interactúa con el usuario, de la implementación depende lo invasivo que es.
  El módulo es implementado en Python 3.6 como código desarrollado y utiliza el API de Outlook para enviar un mensaje al usuario con la alerta.

Clientes

Estos módulos conectan el centinela con los servicios en la nube que consume, además de recibir e interpretar las respuestas recibidas.

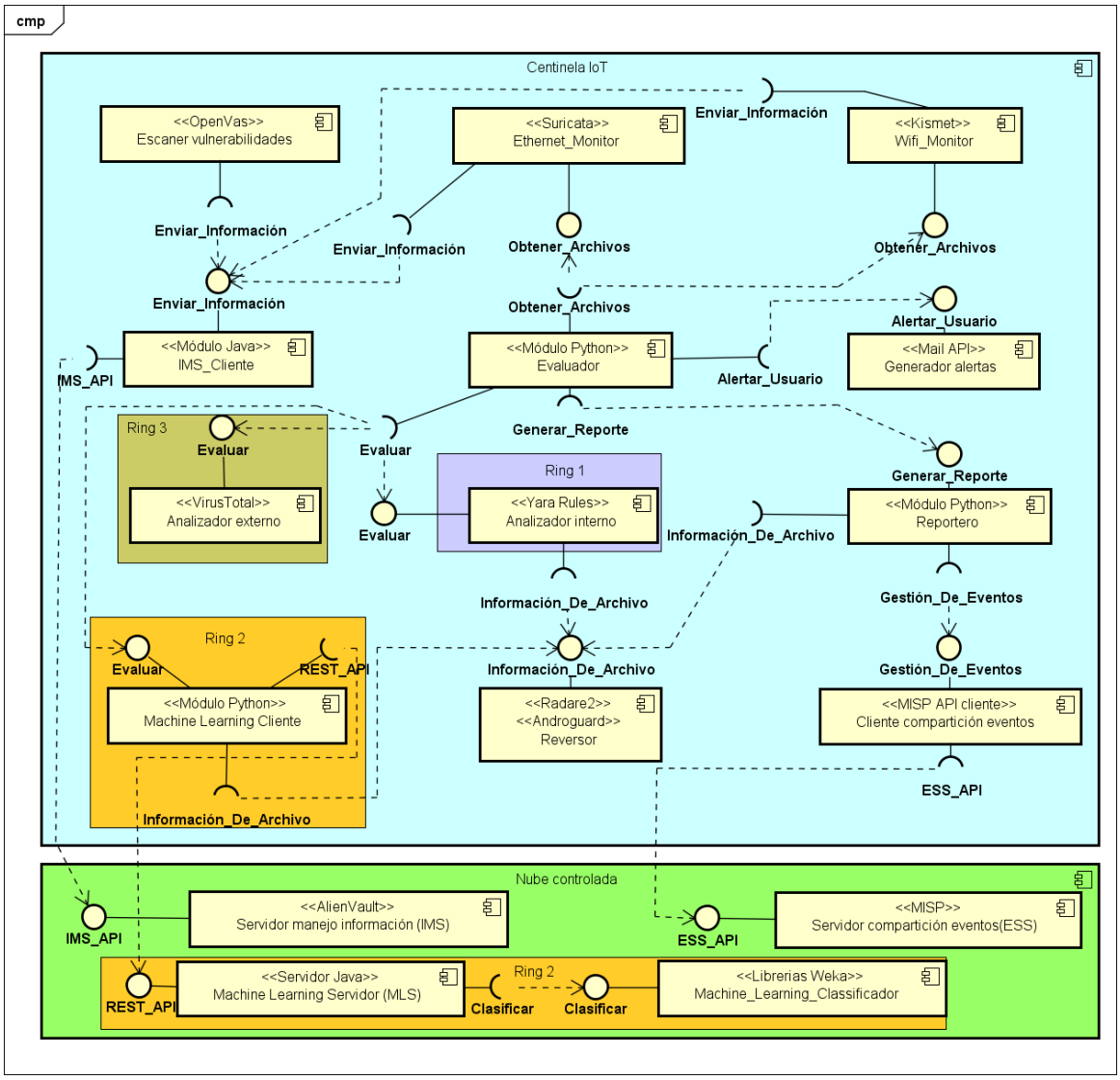
- IMS_Cliente: Este cliente conecta el centinela con el servidor de gestión de información (IMS por sus siglas en inglés), los módulos usando este cliente son aquellos de monitoreo. La implementación usada es un código en Java que conecta el centinela con estos servicios en la nube.
- ESS_Cliente: Este cliente, también llamado cliente de compartición de eventos, conecta el centinela con el servidor de compartición de eventos (ESS por sus siglas en inglés). La implementación usada es PyMISP, un módulo de Python 3.6 que utiliza el API REST de MISP para realizar operaciones de forma remota.
- Machine_Learning_Cliente: Este cliente conecta al centinela con el servidor que aloja las operaciones de clasificación con el modelo de machine learning. El cliente envía un reporte generado por el centinela al servidor y recibe la clasificación realizada por el mismo, a partir de esta el Evaluador puede determinar la naturaleza de la muestra. La implementación usada es un cliente REST que solicita clasificación sobre un String al servidor definido en el archivo de configuración, si el servidor retorna “MAL” el cliente retorna a su vez que la muestra es malware, si el servidor retorna “GOO” la muestra es considerada inofensiva.

Servidor en la nube

Estos son los módulos que contienen servicios demasiado pesados como para ser contenidos en el centinela, de modo que fueron separados en uno o más servidores en la nube dedicados, de este modo el centinela puede consumir libremente los servicios ofrecidos.

- IMS: Este servidor recibe información y la correlaciona. El IMS puede dar información al centinela para que actúe si detecta un ataque. La implementación usada es AlienVault.
- ESS: Este es el servidor que permite compartir información con una o más comunidades de confianza. La implementación usada es MISP.
- Servidor de machine learning: Este servidor contiene el modelo de machine learning y aplica el algoritmo a datos nuevos para generar una clasificación. La implementación es un servidor REST montado con Java y SpringBoot. Este servidor tiene el modelo final generado precargado y recibe un String con el reporte de permisos de una aplicación, una vez recibe el reporte le aplica el algoritmo de ML que haya sido definido y genera una clasificación, que es retornada como respuesta.
  Para la generación del modelo de machine learning se hicieron varios prototipos, en la elección del modelo se tomó en cuenta el porcentaje de acierto del algoritmo evaluado y los coeficientes de evaluación de este. Finalmente se escogió RandomForest, con una validación cruzada de 20 folds y sin límite de profundidad. Este modelo tiene un porcentaje de acierto del 91% y se usó en las pruebas del centinela.
- Clasificador de machine learning: Este módulo contiene los algoritmos de machine learning que pueden usarse en la clasificación. La implementación son los algoritmos preconstruidos de Weka, se importan las librerias y se aplican los algoritmos con los parámetros pertinentes.

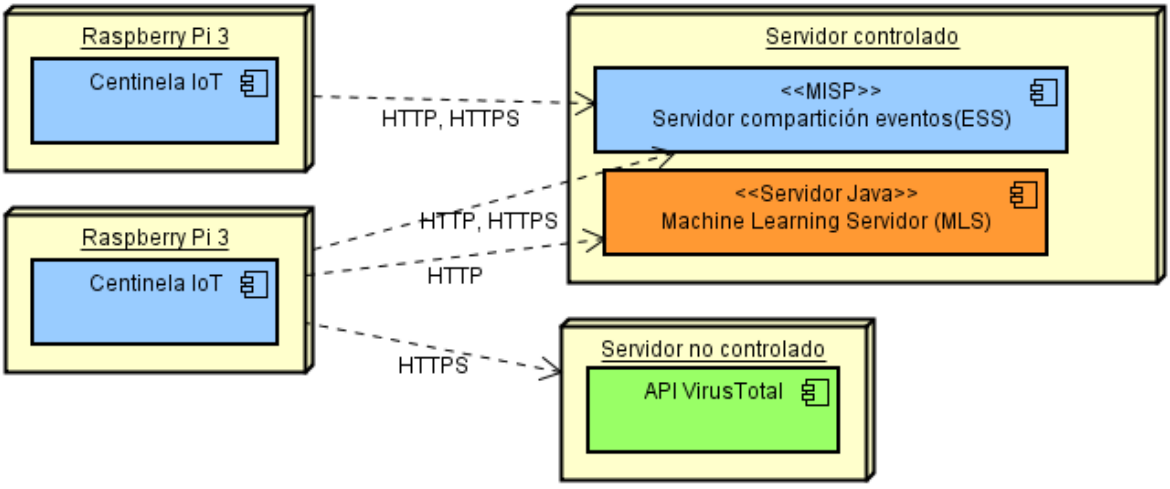
Figura 78. Diagrama de componentes del centinela



Fuente: Propia

En la Figura 97 se puede ver un diagrama de despliegue para la arquitectura del centinela, en este diagrama las instancias del centinela, cada una en una Raspberry Pi 3, se conectan a dos servidores, uno de ellos bajo control de la organización. Este servidor contiene la instancia de MISP y el servidor REST de machine learning. El servidor no controlado por la organización contiene el análisis externo de VirusTotal, el cual es accedido mediante una cuenta educativa con privilegios de consulta.

Figura 79. Diagrama de despliegue del centinela



Fuente: Propia

La propuesta de Centinela IoT presentada en este libro de proyecto de grado ha sido extendida y presentada por medio del artículo “TRIS: a Three-Rings IoT Sentinel to protect against cyber-threats” a “The Fifth International Conference on Internet of Things: Systems, Management and Security (IoTSMS 2018)” realizada del 15 al 18 de octubre del 2018 en la ciudad de Valencia, España [20].

VALIDACION DEL CENTINELA IOT

En la evaluación del desempeño del centinela se tuvieron en cuenta las siguientes variables:

- Peso del archivo: Se utilizaron archivos de menos de 1 MB, entre 1MB y 3MB, entre 3MB y 5MB y entre 5MB y 10MB.
- Velocidad de transmisión: En los experimentos se simuló un escenario de descarga de malware, de modo que se probaron velocidades de transmisión de: 10, 30, 50, 70 ,90, 110 y 120 Megabits por segundo (Mbps)

Se mantuvo constante el número de muestras en 25 malware por cada tamaño y se descargaron usando wget limitando la velocidad de descarga a la velocidad de evaluación del experimento.

Un set de reglas de Yara de prueba que detectara todas las muestras fue generado para probar la capacidad de Yara de resistir muestras mal formadas. Solo se evaluó la captura por Ethernet.

Los resultados fueron los siguientes donde se indica para cada caso el número de muestras que fueron detectadas por cada anillo de seguridad en función de la velocidad del canal y del tamaño de la muestra.

Tabla 1. Detecciones por anillo, distintas velocidades. Tamaño de las muestras: 1MB

Ring\Mbps	10	30	50	70	90	110	120
Yara	23	23	25	23	25	24	25
ML	1	0	0	1	0	0	0
VirusTotal	1	2	0	1	0	1	0

Fuente: Propia

Tabla 2. Detecciones por anillo, distintas velocidades. Tamaño de las muestras: 3MB

Ring\Mbps	10	30	50	70	90	110	120
Yara	25	20	17	22	21	24	23
ML	0	1	1	0	1	0	1
VirusTotal	0	3	7	2	2	0	0

Fuente: Propia

Tabla 3. Detecciones por anillo, distintas velocidades. Tamaño de las muestras: 5MB

Ring\Mbps	10	30	50	70	90	110	120
Yara	25	25	23	20	24	21	24
ML	0	0	0	1	0	2	0
VirusTotal	0	0	2	2	0	1	1

Fuente: Propia

Tabla 4. Detecciones por anillo, distintas velocidades. Tamaño de las muestras: 10MB

Ring\Mbps	10	30	50	70	90	110	120
Yara	25	25	25	24	24	23	18
ML	0	0	0	0	1	0	2
VirusTotal	0	0	0	1	0	0	3

Fuente: Propia

Los datos no muestran una correlación evidente salvo que el monitor reacciona bastante bien ante situaciones de estrés, pues, aunque el archivo quede lo suficientemente deformado como para no ser detectado por Yara, en la mayoría de los casos es detectado por el segundo o tercer anillo.

Es de destacar también que el segundo anillo no tiene un índice de detección tan alto como es de esperar, al analizar el output de los experimentos se descubrió que este anillo manda error cuando el archivo está mal formado y luego de ejecutarlo con los archivos originales el índice de detección alcanzó los valores esperados.

CONCLUSIONES Y TRABAJOS FUTUROS

De la experiencia de diseño e implementación del centinela, además de la experimentación realizada, se puede concluir que:

- La integración de herramientas diseñadas para escritorio en un ambiente móvil es posible, pese a las limitaciones técnicas derivadas de la capacidad de los dispositivos evaluados.
- En casos de estrés del centinela en los cuales el primer anillo no fue capaz de detectar el archivo malicioso, tanto el segundo como tercer anillo en la mayoría de los casos lograron contener la amenaza.
- La integración de estas herramientas de escritorio puede facilitarse al usar herramientas nativas del ambiente móvil, tal como es el caso Yara – Androguard.
- Las velocidades de transmisión por Ethernet manejadas en hogares (Entorno al cual el centinela va dirigido) no tienen un impacto significativo para muestras de menos de 10MB de tamaño.
- MISP permitió obtener un centinela colaborativo al crear una organización a la que pertenecen los centinelas y a la que estos contribuyen como usuarios automatizados con capacidades de publicación de eventos.
- La capacidad del centinela de generar nuevas reglas para el anillo 1 a partir de los resultados de los anillos 2 y 3 evidencia que es adaptable.
- Para ejecutar el centinela solo es necesario activar los módulos de monitoreo y el módulo de evaluación, además, la configuración está unificada en un solo archivo, haciendo de este fácilmente desplegable.
- Tanto la captura como la ejecución de todos los módulos y funciones del centinela se realizan sin supervisión humana, haciendo que este esté muy automatizado.
- Las alertas son modificables debido a la modularidad del código, por defecto, la única alerta se hace enviando un correo al usuario con el anillo que realizó la detección, el nombre del malware y las instrucciones a seguir, haciendo del centinela muy poco intrusivo.
- En los dos primeros anillos el performance en tiempo percibido por el usuario se degrada a medida que aumenta la velocidad, esto posiblemente debido a los errores de transmisión captados por Suricata e intentos de releer y reconstruir.

Como trabajo futuro se propone

- Probar el centinela en velocidades de transmisión de entornos corporativos. Esto con el fin de comprobar el grado de estrés y respuesta que tiene el centinela en un entorno con más requisitos de cómputo.
- Generar casos más realistas donde se evalúen las capacidades del centinela usando Wifi. Esto debido a que una gran parte de los dispositivos IoT se conectan a sus redes usando Wifi
- Integrar sandboxing al modelo de machine learning para tener más información con la que generar una clasificación más acertada. De esta forma puede

aumentarse el porcentaje de detección del modelo, puesto que se le añadiría información concerniente a la muestra analizada.

BIBLIOGRAFÍA

[1] Z. Hassan, Z. H. Ali, H. A. Ali, and M. M. Badawy, “Internet of Things View project Internet of Things (IoT): Definitions, Challenges and Recent Research Directions,” *Artic. Int. J. Comput. Appl. Int. J. Comput. Appl.*, vol. 128, no. 1, pp. 975–8887, 2015.

[2] Gartner, “Gartner Top 10 Strategic Technology Trends for 2018 - Smarter With Gartner.” .

[3] Gartner, “Gartner’s Top 10 Strategic Technology Trends for 2017 - Smarter With Gartner.” .

[4] Gartner, “Gartner Identifies the Top 10 Strategic Technology Trends for 2016.” .

[5] Gartner, “Gartner’s Top 10 Strategic Technology Trends for 2015 - Smarter With Gartner.” .

[6] D. Lund, C. Macgillivray, V. Turner, and M. Morales, “Worldwide and Regional Internet of Things (IoT) 2014–2020 Forecast: A Virtuous Circle of Proven Value and Demand IDC OPINION,” 2014.

[7] Symantec, “ISTR Internet Security Threat Report,” 2018.

[8] D. Betts and B. Lamos, “IoT Security Architecture | Microsoft Docs.” .

[9] Symantec, “An Internet of Things Reference Architecture,” 2016.

[10] H. Ning and H. Liu, “Cyber-Physical-Social Based Security Architecture for Future Internet of Things,” *Adv. Internet Things*, vol. 2, no. 1, pp. 1–7, Jan. 2012.

[11] M. M. Hurley, “For and from Cyberspace: Conceptualizing Cyber Intelligence, Surveillance, and Reconnaissance.” 2012.

[12] J. Turnbull, *Logstash Book*. Turnbull Press, 2013.

[13] C. Gormley, *Elasticsearch : the Definitive Guide.* O’Reilly Media, 2015.

[14] Y. Gupta, *Kibana Essentials.* Packt Publishing, 2015.

[15] N. Moreno Guataquira, S. Morón Castro, and A. F. Vega Torres, “Seguridad para IoT: solución para la gestión de eventos de seguridad en arquitecturas de Internet de las cosas,” 2017.

[16] C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody, “MISP,” in *Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security - WISCS’16*, 2016, pp. 49–56.

[17] K. Dunham, *Android malware and analysis*. CRC Press, Taylor & Francis Group, 2014.

[18] M. Sikorski and A. Honig, *Practical Malware Analysis : a Hands-On Guide to Dissecting Malicious Software.* No Starch Press, 2012.

[19] M. Mansoori, I. Welch, and Q. Fu, “YALIH, yet another low interaction honeyclient,” *Proc. Twelfth Australas. Inf. Secur. Conf. - Vol. 149*, pp. 7–15, 2014.

[20] D. Useche, D. Díaz López, P. Nespoli, and F. Gómez Mármol, “TRIS: a Three-Rings IoT Sentinel to protect against cyber-threats,” in *5th International Conference on Internet of Things, Systems, Management & Security (IOTSMS 2018), At Valencia, Spain*, 2018, p. 101.