Detección de *creepware* en computadoras personales mediante el uso de técnicas de ciberseguridad, análisis de datos y aprendizaje automático para clasificación de tráfico de red

Arancibia, Sofia Casandra
Universidad Nacional del Litoral – Facultad de Ingeniería y Ciencias Hídricas
sarancibia24@yahoo.com

Abstract

El presente trabajo propone el análisis del tráfico de red como método de detección de software malicioso (malware). En particular, el trabajo se enfoca en la detección de malware del tipo creepware en computadoras personales (notebook, netbook, desktop).
El creepware se caracteriza por el acceso ilegítimo a los dispositivos de audio y video, y la posterior exfiltración de los datos capturados. Este documento presenta el uso de métodos supervisados de aprendizaje automático (machine learning), para detección basada en la clasificación de tráfico de red. Como parte del trabajo se han generado datos de entrenamiento y pruebas mediante la detonación, en entornos controlados, de diferentes ejemplares del malware.
Como parte de los resultados se han identificado diferentes conjuntos de características que aplican al tráfico generado por el malware del tipo creepware, y que han permitido la clasificación y detección de los ataques con una precisión por encima del 90%.

Palabras Clave

machine learning, suppor vector machine, decision tree, k-nearest neighbours, creepware, tráfico de red

Introducción

La clasificación de tráfico de internet es una tarea esencial tanto para el manejo de grandes redes de datos, como para la prevención y detección de infecciones en un dispositivo. Los métodos tradicionales de clasificación de tráfico basados en protocolos y números de puerto, resultan incompletos al día de hoy, dado que muchas aplicaciones modernas utilizan puertos reservados dinámicamente, o puertos bien conocidos asociados con protocolos populares (como el puerto 80 utilizado por el protocolo http) con el fin de saltear firewalls (cortafuegos) y otros dispositivos de seguridad de red[1], técnica denominada por la base de conocimiento MITRE|ATT&CK como “Commonly Used Port (T1043)”. Por otro lado, los enfoques basados en inspección profunda de paquetes (Deep Packet Inspection - dpi), resultan ser de los más precisos, pero presentan la desventaja de poder aplicarse sólo sobre tráfico no encriptado, lo cual resulta ser lo menos usual con la ampliamente adoptada práctica de encriptación punto a punto.

En los últimos años, la clasificación de tráfico basada en características estadísticas se ha vuelto una temática de gran interés. Este trabajo se enfoca en la clasificación de tráfico de red a partir de características estadísticas extraídas por cada tipo de flujo. La propuesta implica la aplicación de métodos de aprendizaje supervisado por medio de los modelos: Máquina de Soporte Vectorial

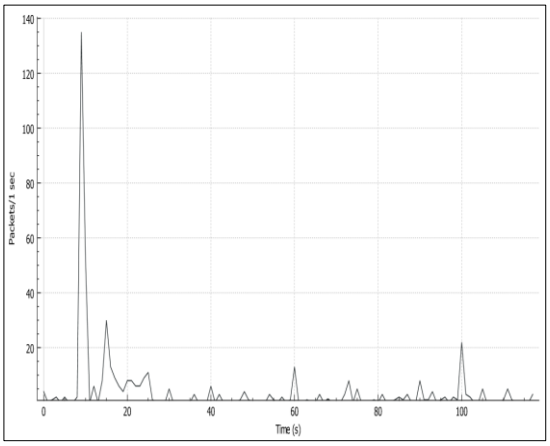

Fig. 1: Estado normal de la red

(Support Vector Machine - svm), k Vecinos más Cercanos (k-Nearest Neighbours - knn) y Árbol de Desición (Decision Tree – dt). Analizando el comportamiento de una red de datos doméstica frente a estímulos de diversa índole, puede apreciarse una significativa variabilidad en el número de paquetes entrantes y salientes. A priori, puede asociarse tal variabilidad a las aplicaciones en ejecución y las conexiones que éstas establecen con otros dispositivos en la misma red, o con otras redes (Internet). Observar mediante las fig. 1 y 2, que el número de paquetes aumenta significativamente bajo un escenario de ataque (fig. 2), con respecto a un estado normal del equipo (fig. 1).

Justificación y antecedentes

El malware es un tipo malicioso de software (de allí su denominación) destinado a acceder a un dispositivo de forma inadvertida con la intención de producir daños en el equipo afectado, esclavizarlo o secuestrarlo, para obtener información personal del usuario con el propósito de venderla o extorsionarlo, distribuir publicidad, y explotar de manera silenciosa las vulnerabilidades del sistema. Estos programas incluyen a troyanos, gusanos, spyware, adware, ransomware.
Este software se instala en el equipo frecuentemente a través de archivos descargados de internet, aparentemente inofensivos, pero además puede distribuirse mediante el correo electrónico, mensajería instantánea, dispositivos de almacenamiento extraíbles, etc. En muchos casos, los programas maliciosos

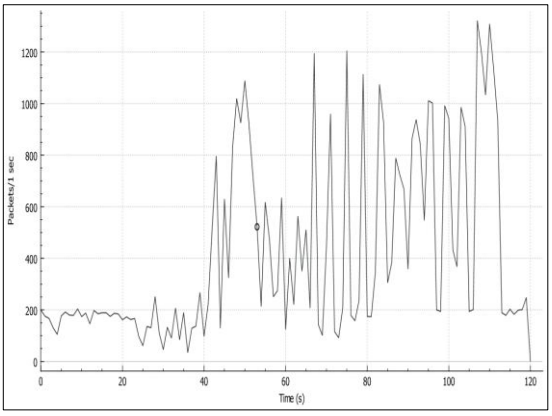

Fig. 2: Estado infectado de la red

están diseñados para burlar firewalls y antivirus, y algunos son alojados en aplicaciones que pueden obtenerse desde tiendas oficiales de descarga, lo cual otorga al usuario la confianza necesaria para descargar el archivo.
Frecuentemente, estos programas dejan huellas visibles al usuario que evidencian su presencia, como la lentitud repentina del equipo y ventanas emergentes, pero en los casos del software destinado a espionaje no es así, dado que requieren pasar desapercibidos, y detectarlos es una tarea desafiante.
Dentro de la categoría malware se encuentran los creepware, un caso particular de troyano que se instala en el equipo y permite al atacante, espiar a la víctima a través de su cámara web y micrófono, independientemente de que estos dispositivos estén integrados en el equipo o sean externos a él.
El caso por el cual este método de espionaje cobró mayor popularidad, fue el de la Miss Teen USA, Cassidy Wolf, quien fuera víctima de Blackshades, un programa creepware instalado en su equipo personal por un compañero de la universidad, quién tomó fotos y videos de ella durante un año, para luego extorsionarla con publicar dichas fotos en internet y arruinar su carrera de modelo a menos que ella accediera a grabar y enviar videos de contenido sexual. En el ámbito nacional, hace poco más de tres años, se conocía a ``Camus", un supuesto hacker que divulgó fotos y videos de alto contenido privado de artistas de la farándula argentina. Según se difundió, Camus habría logrado acceso a las cámaras web de las computadoras de las víctimas y grabó videos íntimos sin que ellos siquiera lo notaran. Tiempo después, pudo comprobarse que Camus sólo descargó dicho contenido de internet, sin vulnerar ningún equipo, pero la realidad es que este caso despertó muchas mentes inquietas, y develó en cierta manera, la posibilidad de acceder a imágenes o videos de terceros, en unos pocos y simples pasos.
En la República Argentina, según el artículo 183 del Código Penal, ampliado por el artículo 10 de la Ley 26.388: *Será reprimido con prisión de quince días a un año, el que destruyere, inutilizare, hiciere desaparecer o*

de cualquier modo dañare una cosa mueble o inmueble o un animal, total o parcialmente ajeno, siempre que el hecho no constituya otro delito más severamente penado.
[En la misma pena incurrirá el que alterare, destruyere o inutilizare datos, documentos, programas o sistemas informáticos; o vendiere distribuyere, hiciere circular o introdujere en un sistema informático, cualquier programa destinado a causar daños].
Según datos del FBI, se estima que sólo con el software BlackShades, 100 personas repartidas en 20 países espiaron a más de un millón de usuarios[3]. El peligro real de este malware radica en su facilidad de instalación y uso: las personas que descargan este software y comienzan a usarlo son conocidos como "script kiddies", es decir, hackers amateur que gustan de experimentar con sus máquinas, son amantes de las fallas de las mismas y con frecuencia quieren demostrar sus conocimientos, emplean herramientas y técnicas utilizadas por los hackers, para lograr penetrar sistemas o inutilizarlos[4]. Con lo anterior, se quiere decir que cualquier usuario con mínimos conocimientos sobre descarga e instalación de software, puede obtener y hacer uso de este tipo de herramientas de malware, en pos de perpetrar el espionaje. Esta facilidad implica que la cantidad de personas involucradas en ciberataques del tipo mencionado, tanto como víctima o como intruso, irá en constante aumento y es, al mismo tiempo, la causa que motiva la realización de este proyecto.

Por otro lado, se ha mencionado que este tipo de malware responde a las características de los troyanos, y grandes firmas de antivirus han desarrollado productos contra estos. El problema aquí, es que no existe una única versión del malware, por lo cual no es del todo seguro delegar al antivirus la tarea de detectarlo, dado que éste no siempre cuenta con la última "vacuna" (ni los usuarios con la última actualización del antivirus).

Contribuciones

En secciones previas se mencionó la posibilidad de que el atacante logre acceder al equipo objetivo aún estando presentes barreras como los antivirus y actualizados a su última versión. Contra esto, quienes disponen de equipos con cámara web integrada (cabe mencionar que el problema persiste aún utilizando una cámara web externa) suelen imaginar que "es fácil darse cuenta cuando se está siendo grabado, sólo basta con observar si el LED que está a un costado de la cámara está encendido"; lo que sucede es que éste LED puede estar roto, o incluso desactivado por los propios atacantes para evitar ser descubiertos (sin mencionar que no todos los equipos cuentan con tal LED). Actualmente, los mecanismos de protección a este tipo de ataque involucran deshabilitar ambos dispositivos ya sea mediante sistema operativo, desconectarlos del PC cuando no son utilizados o simplemente cubrir la lente de la cámara.
Al día de hoy, no se ha detectado en el mercado ninguna herramienta especializada en estos ataques; simplemente, se recurre a los antivirus, ignorando la rapidez y facilidad con la cual los atacantes logran re-adaptar sus códigos maliciosos para evadirlos.
En este proyecto se propone colaborar con el usuario común en la detección de esta clase de intrusión, mediante una herramienta destinada a la captura, procesamiento, análisis y clasificación del tráfico de red fluyendo a través el equipo del usuario, antes y durante un ataque como el descripto previamente.
La herramienta será independiente del troyano que está actuando, dado que se ocupará de analizar paquetes de datos enviados "por" el usuario. Esa característica permitirá al software subsistir en el tiempo y dar respuesta frente a las evoluciones o versiones del malware. Las actualizaciones de la misma no se basarán en la evolución del malware sino en la de los algoritmos para detectarlos que el autor considere más eficientes.
La herramienta dará aviso inmediato al usuario mediante un mensaje alusivo por pantalla, permitiendo actuar con rapidez y como lo considere apropiado, una vez alertado sobre

el ataque.

Dataset

Utilizar un modelo de aprendizaje automático implica mucho más que introducir datos en un algoritmo, y observar qué resultados se obtienen. La mayor parte del esfuerzo se destina a la construcción del conjunto de datos; ésta construcción implica la recolección de los datos en "crudo", selección de características, filtrado y transformación, para luego realizar un procedimiento final conocido como ingeniería de características (features engineering).
Para el caso de estudio de este proyecto, fue necesaria la elaboración de una base de datos específica a partir de la captura del tráfico de la red del autor. La decisión de sintetizar una base de datos radica fundamentalmente en que al momento, no existe ninguna base de datos pública dedicada a este tipo de estudios. Por otro lado, las bases de datos disponibles actualmente, fueron generadas con el propósito de clasificar diferentes ataques, lo cual no se alinea con los objetivos de este trabajo, el cual requiere el análisis de un tipo de tráfico en particular.
Realizar una recolección de datos con el fin de encontrar en ellos patrones que respondan a características de un ciberataque, implica que el equipo del cual se toman dichos datos debe estar bajo los efectos de tal ataque. Es por esto que en este trabajo se han utilizado dos máquinas virtuales : una tomaría el rol de víctima y la otra de atacante, en la que se detonaron diversos RATs para realizar el espionaje de la máquina víctima. Desde esta última, durante los ataques, se capturó todo el tráfico de red producido, utilizando la herramienta Wireshark.
Para cada malware ejcutado, se creó una carpeta independiente, que almacena exclusivamente el tráfico generado por cada uno (vale aclarar que los programas fueron ejecutados uno a la vez, conservando en su correspondiente fichero el tráfico generado).
Dicha captura se realizó permitiendo la generación de tráfico usual, ya sea la consulta de diversos sitios web, correo electrónico, etc., como así también tráfico producido por la infección.
La captura de tráfico produjo como resultado un conjunto de archivos `.pcapng`, un formato estándar para almacenar datos de red. La captura se desarrolló durante 3 días, de forma de recolectar la mayor cantidad de tráfico ambiente posible (redes sociales, edición online de texto, consulta de sitios web y correo electrónico, etc), a la vez que se realizaban ataques con diversas herramientas, durante intervalos de tiempo y horarios variables (la captura se realizó en horarios correspondientes a mañana, tarde y noche durante intervalos de 1 hora), de modo de asemejar un ataque real. Esta generación intencionada de tráfico hostil, se realiza con el fin de poder asociar esos flujos con el ataque realizado y poder comprobar posteriormente la fiabilidad de la clasificación producida.
De la captura realizada, se obtuvo una base de datos de 1381 archivos, cada uno con aproximadamente 1000 paquetes capturados. Los paquetes de cada captura serán luego agrupados en flujos, entendiendo un flujo como el conjunto de paquetes que cuentan con los siguientes parámetros en común: dirección IP de origen, dirección IP de destino, puerto de origen, puerto de destino y protocolos UDP o TCP[2]. De estos flujos se extrajeron luego parámetros que serán utilizados para calcular variables estadísticas.
Atendiendo a la importancia del sentido del flujo, es decir, flujo entrante o saliente, es que se decidió utilizar las direcciones de origen y destino de cada registro para construir una nueva característica, capaz de indicar en qué casos el equipo víctima está enviando datos a la red (siendo estos los casos de mayor interés), y en qué casos está recibiendo datos. Para dicha construcción, se utilizaron los criterios que se muestran en la siguiente tabla. Allí puede observarse que la nueva característica, denominada "dirección de flujo", toma el valor '1' en los casos en donde la dirección de origen pertenezca al *host* y '0' cuando esta dirección se utilice como destino. Cabe aclarar aquí, que debido a que los datos fueron generados y capturados en el

equipo del autor, todas las direcciones de origen y destino involucradas en las diversas comunicaciones generadas, son conocidas; el resto de las direcciones que participan del conjunto de datos, se consideran como direcciones de tráfico normal.

	Origen	**Destino**	**Dirección de flujo**
Enviando	ip víctima	ip atacante	1
Recibiendo	ip atacante	ip víctima	0

El lector habrá podido inferir el motivo por el cual resulta importante conocer la dirección de cada flujo: El propósito de este trabajo es identificar flujos que corresponden a "fuga" de información, en cuyo caso el flujo será saliente del equipo víctima. Es por este motivo, que se conservaron únicamente los flujos cuya dirección tome el valor 1. Finalmente, se ha agregado una columna adicional para el etiquetado de los flujos. Con las etiquetas se indicará durante el entrenamiento de los algoritmos, cuáles flujos son benignos y cuáles maliciosos con las etiquetas '0' y '1' respectivamente. A modo de resumen, en la siguiente tabla se presenta al lector la totalidad de las características extraídas y calculadas para este proyecto.

Caracteristica	**Descripción**
ip_src	Dirección IP de origen
ip_dst	Dirección IP de destino
src_port	Puerto de origen
dst_port	Puerto de destino
flow_dir	Dirección del flujo
mean_payload_size	Longitud de payload promedio
std_payload_size	Desviación estándar de la longitud de payload
var_payload_size	Varianza de la longitud de payload
max_payload_size	Longitud máxima de payload
min_payload_size	Longitud mínima de payload
quantity	Nuúmero de paquetes en el flujo
mean_frame_time_delta	Tiempo promedio entre la llegada de un paquete y el siguiente
sd_frame_time_delta	Deasviación estándar del tiempo entre la llegada de un paquete y el siguiente
var_frame_time_delta	Varianza del tiempo entre la llegada de un paquete y el siguiente
max_frame_time_delta	Tiempo máximo entre paquetes
min_frame_time_delta	Tiempo mínimo entre paquetes
label	Etiqueta del flujo

El proceso de construcción de la base de datos de este proyecto puede observarse en la fig. 3.

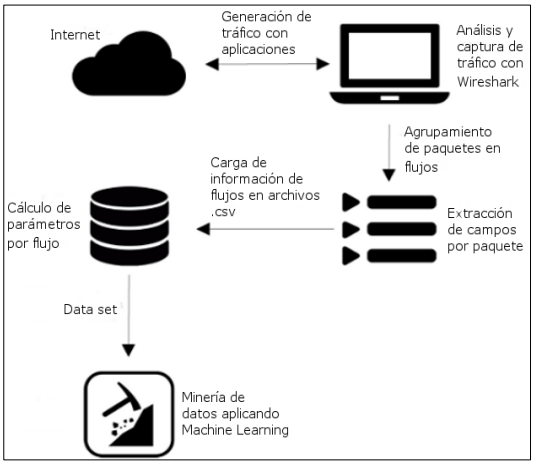

Fig. 3: Diagrama de flujo para el proceso de captura, procesamiento y organización del conjunto de datos

Resultados

Las pruebas realizadas para este proyecto se basan en la clasificación de tráfico generado por aplicaciones que a primera vista realizan acciones similares sobre el equipo que las ejecuta, como ser la transmisión de audio y/o video, extrayendo del mismo características que permitan individualizarlas con la mayor precisión posible. La diferencia a detectar responde al objetivo de discriminar aplicaciones de naturaleza inofensiva de aquellas

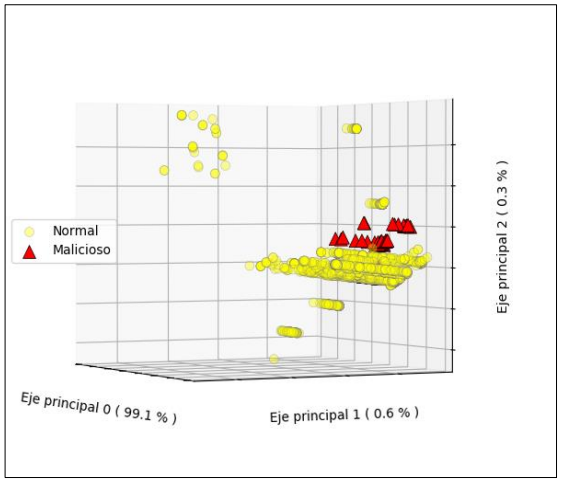

Fig. 4: Diagrama de dispersión del conjunto de datos multiclase proyectado sobre sus 3 componentes principales, las cuales conservan el 99.1%, 0.6% y 0.3% de la variabilidad total, respectivamente.

cuya intencionalidad es invadir la privacidad del usuario, vigilando su accionar diario o capturando videos/audios del mismo, por medio de su cámara web o micrófono, con la intención de obtener de ello entretenimiento personal o beneficios económicos mediante el chantaje o venta de los recursos multimedia obtenidos.

En este proyecto, para facilitar la visualización del conjunto de datos en su totalidad, se aplicó la reducción a 3 dimensiones utilizando el método de componentes principales, lo cual permite realizar un análisis del balance y distribución del conjunto. Esta representación puede observarse en la fig. 4. A partir de esta figura se podrán observar dos aspectos relevantes del conjunto. Primero, existe un gran desbalance entre las clases que representan tráfico de fondo o normal y aquellas que representan a los ataques, estos últimos indicados con triángulos rojos en el gráfico de dispersión. Esto se debe a que durante la captura de tráfico no se aplicó ningún tipo de filtro, con el objetivo de no desechar ningún paquete que pueda resultar importante para la identificación del tráfico buscado, por lo cual se han capturado paquetes de múltiples jerarquías; además, se debe considerar que bajo un escenario de ataque exitoso, el usuario ignora la presencia del atacante, por lo cual realiza en su equipo, todo tipo de actividades normales o rutinarias, como consulta y envío de correo electrónico, chat, eventualmente mirar películas o escuchar música, cargar/descargar archivos, etc., tareas que producen un volumen de tráfico significativo.

Segundo, puede observarse que si bien la gráfica presentada es una proyección a las componentes principales del conjunto, se puede lograr una representación aproximada de la distribución de las clases en un espacio alternativo. Observar además, que los patrones dentro del mismo pueden ser discriminados con cierta precisión por cada tipo de tráfico.

El paso siguiente fue entonces realizar la clasificación de los flujos por medio de modelos de machine learning cuyas funciones de decisión logren adaptarse a esta distribución para posteriormente efectuar una mejor toma de decisiones.

Para cada modelo se llevó a cabo su hiperparametrización, logrando el siguiente conjunto de parámetros:

- SVM:
  - kernel: lineal
  - c: 1000

- Decision tree:
  - min_sample_split: 10
  - max_depth: 5
  - min_samples_leaf: 5

- k-NN:
  - n_neighbours: 1
  - leaf_size: 10
  - algorithm: ball_tree

Cabe señalar que estos parámetros corresponden a los requeridos por los métodos incluídos en la librería scikit-learn. En las figuras 5 a 7 se presentan las matrices de confusión para cada modelo.

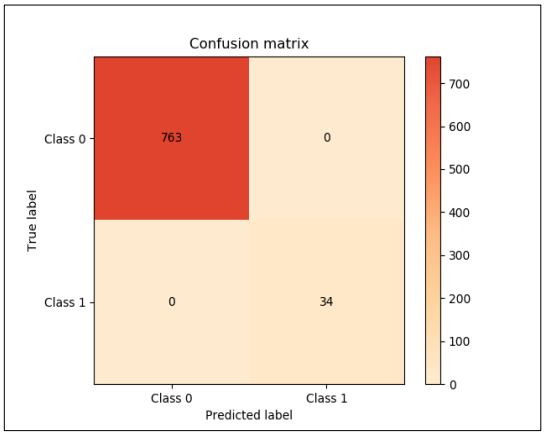

Fig. 5: Matriz de confusión obtenida durante la etapa de prueba por el algoritmo Decision Tree con profundidad máxima de 5 niveles.

Se puede concluir que los tres modelos empleados han logrado realizar la tarea de clasificación de manera correcta, observando un desempeño superior en el Árbol de decisión. Esto es debido a que el Árbol de Decisión tiende a realizar una selección de las características que destacan mejor las diferencias entre los patrones, otorgando menor peso a aquellas que menor influencia ejercen (consideradas ruidosas). Por el contrario, los restantes métodos no realizan esta selección, lo cual se refleja en un desempeño inferior.

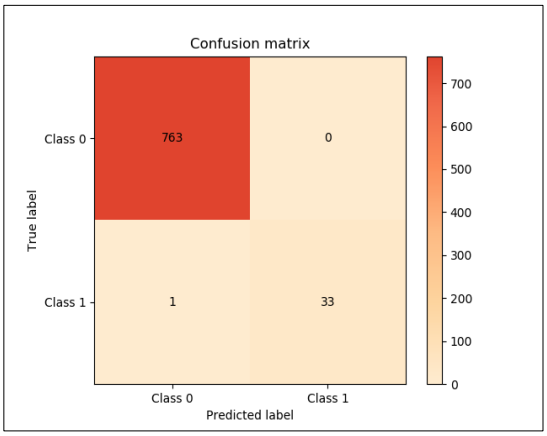

Fig. 6: Matriz de confusión obtenida durante la etapa de prueba por el algoritmo k-Nearest Neighbours con K=1 vecinos cercanos.

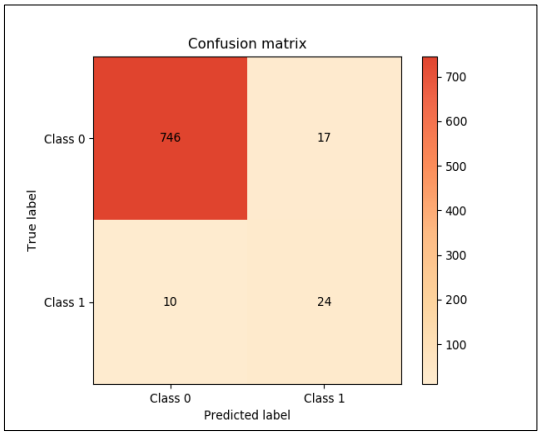

Fig. 7: Matriz de confusión obtenida durante la etapa de prueba por el algoritmo Support Vector Machine con kernel lineal y C=1000.

Conclusión

En el contexto de la seguridad en redes, tal como la detección de intrusiones y anomalías, la identificación y clasificación de tráfico es un tópico de relevancia creciente. La automatización del malware (es decir, la independencia proporcionada al software de fin malicioso para llevar a cabo sus tareas) ha resultado un factor decisivo, que fuerza al campo de la ciberseguridad al empleo de la estadística y el aprendizaje automático como mecanismos de detección y prevención de intrusiones. En este trabajo se realizó la tarea de recolección y preprocesado de datos para lograr entrenar diversos algoritmos que faciliten la tarea de la detección de espionajes tipo creepware en computadoras personales. Se han evaluado diferentes modelos de aprendizaje automático para realizar la clasificación de tráfico de red según responda a las características de tráfico inofensivo o malicioso. Además, se ha desarrollado una comparación del desempeño de los tres modelos a partir del entrenamiento de los mismos con un conjunto de datos obtenidos de orígenes conocidos, como así también a partir de utilizar distintas herramientas para penetrar sistemas víctima. Entrenar modelos que arrojen los resultados esperados resulta una tarea de suficiente dedicación para lograr extraer y presentar los datos correctos, y de forma adecuada para el

modelo, el cual asimismo debe ser seleccionado en concordancia con el tipo de dato a analizar; pero todo esto resulta en una amplia satisfacción al observar que los resultados obtenidos concuerdan con los esperados, y más aún si dichos resultados implican un avance en la resolución de problemas de la sociedad de hoy.

En base al trabajo realizado, pueden identificarse aspectos a considerar e implementar en versiones posteriores del sistema:

• Incorporar un interfaz gráfica que facilite la ejecución del sistema a usuarios no familiarizados con la línea de comandos.

• Proporcionar un mecanismo de rastreo de archivos y/o procesos relacionados a la intrusión, en caso de detectarse.

• Extrapolar el desarrollo a dispositivos móviles con sistema operativo Android.

El proyecto seguirá un proceso de mejora continua, en el cual se incorporarán cambios al sistema en base a la evolución de las tecnologías y métodos aplicados y no aplicados a este proyecto, con el fin de perfeccionar el desempeño del sistema.

Referencias

[1] Gabriel Gómez Sena and Pablo Belzarena. *Early traffic classification using Support Vector Machines*. LANC, 2009.

[2] Luis Gil Delgado. *Clasificación de tráfico de Internet mediante análisis de datos* (tesis de grado). PhD thesis, Departamento de Ingeniería Telemática y Electrónica. Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicaciones, Junio 2015.

[3] Gamen, Sebastián A. Éramos pocos y tu webcam te espía. https://bit.ly/ROTgIiB, 2015. Accedido el 3-04-2017.

[4] Jeimy J. Cano M. Computación Forense. Descubriendo los rastros informáticos. Alfaomega, México, 2009.

Datos de Contacto:

Sofia Arancibia. Universidad Nacional del Litoral, Facultad de Ingeniería y Ciencias Hídricas. Ruta Nacional N° 168 - Km 472,4. (3000) Santa Fe. sarancibia24@yahoo.com