

Sistema de Detección de Anomalías para protocolos propietarios de Control Industrial

Iñaki Garitano¹, Mikel Iturbe, Ignacio Arenaza-Nuño, Roberto Uribeetxeberria, Urko Zurutuza

Dpto. de Electrónica e Informática
Escuela Politécnica Superior
Mondragon Unibertsitatea

Email: {igaritano,miturbe,iarenaza,ruribeetxeberria,uzurutuza}@mondragon.edu

Resumen—Las Infraestructuras Críticas, ofrecen servicios esenciales para el funcionamiento de sociedades modernas y se controlan mediante Sistemas de Control Industrial. Garantizar su seguridad es primordial debido a las graves consecuencias que puede acarrear un ataque exitoso. Además, la reciente aparición de gusanos diseñados de manera exclusiva evidencia el creciente interés que sufren dichos sistemas. Las soluciones de seguridad existentes se centran en protocolos de red públicos de Sistemas de Control Industrial, dejando a un lado los propietarios, debido en gran medida a su desconocimiento. Con el propósito de ofrecer un mecanismo de seguridad integral, tanto para protocolos propietarios como públicos, a lo largo de este artículo se presenta un Sistema de Detección de Anomalías basado en el *payload* y el flujo de los paquetes, en conjunto con un método capaz de describir el comportamiento de red mediante un conjunto de reglas. La validación se ha realizado utilizando un Sistema de Control Industrial real. El bajo número de falsos positivos demuestra su validez.

Palabras clave—detección de anomalías (*anomaly detection*), protocolos propietarios (*proprietary protocols*), sistemas de control industrial (*Industrial Control Systems*)

I. INTRODUCCIÓN

Los Sistemas de Control Industrial (*Industrial Control Systems*, ICS) hacen referencia al conjunto de elementos especializados en la monitorización y control de procesos industriales, los cuales incluyen las Infraestructuras Críticas (*Critical Infrastructures*, CIs), necesarias para el correcto funcionamiento de las sociedades avanzadas.

Las Amenazas Persistentes Avanzadas (*Advanced Persistent Threats*, APTs) suponen una nueva generación de software malicioso y sofisticado. Con unas metas concretas y bien definidas, tienen un nivel de eficacia muy elevado y son sigilosas durante su ejecución, siendo capaces de ocultarse ante las posibles medidas de seguridad. Algunas de las APTs diseñadas para atacar a las CIs e ICSes son Stuxnet [1] y Duqu [2], cuyos objetivos son la interrupción de los servicios o el ciberespionaje y el robo de información.

Los ataques contra los ICSes, y en consecuencia la interrupción de sus servicios, podía acarrear serias consecuencias de diversa índole (económica, medioambiental...), potencialmente catastróficas, como muestran el impacto causado por algunos ataques anteriores [3]. Por ello, es de vital importancia

salvaguardar la seguridad y el correcto funcionamiento de las CIs e ICSes ante las APTs y otros softwares maliciosos, ya que de ello depende en gran medida el bienestar de las sociedades avanzadas. Además, la seguridad es un proceso continuo donde cada medida de seguridad aporta o establece una barrera más, en un entorno donde no existe la seguridad absoluta. De ahí que sea necesario encontrar y desarrollar nuevas técnicas de seguridad que sean capaces de detectar no solo ataques bien conocidos, sino también posibles amenazas que pudiesen alterar el funcionamiento de los ICSes.

Los Sistemas de Detección de Intrusiones (*Intrusion Detection Systems*, IDS) se clasifican en base a distintos parámetros como bien pueden ser: el origen de los datos auditados, el método de detección o el modo de repuesta. Si bien es cierto que los IDSes se pueden catalogar en base a distintos atributos, en el entorno de los ICSes se clasifican principalmente en base al método de detección. Aquí podemos diferenciar dos grandes familias: las que se basan en el conocimiento, también conocidos como los basados en firmas, y los que se basan en el comportamiento, conocidos como Sistemas de Detección de Anomalías (*Anomaly Detection Systems*, ADS).

El método tradicional de detección de intrusiones está basado en firmas, donde las firmas describen patrones de ataque y el tráfico de red es analizado para ver si corresponde con alguna de las firmas. Esta estrategia sólo detecta ataques conocidos, debido a que es necesario conocer los detalles de un ataque para crear las firmas que lo describan. Por ello, los IDSes basados en el conocimiento son ineficientes a la hora de detectar APTs [4]–[5] y ataques desconocidos (*zero-day attacks*).

Las diferencias existentes entre los ICSes y las tradicionales Tecnologías de la Información y de las Comunicaciones, tales como los protocolos de red correspondientes a los ICSes o la falta de recursos de la mayoría de los componentes industriales, ponen de manifiesto la necesidad de la creación de medidas de seguridad especialmente diseñados para los ICSes [6]–[7]. Si bien es cierto que la seguridad de red se puede aportar a través de mecanismos de seguridad, tales como cortafuegos o IDSes, estos últimos son especialmente adecuados debido a las peculiaridades del tráfico de red industrial. En la mayoría de los casos, aun teniendo un ICS que controla un proceso físico continuo y cambiante, la comunicación entre

¹Afiliación actual: University Graduate Center at Kjeller (UNIK), Noruega. Email: igaritano@unik.no

los *Master Terminal Unit* (MTU) o servidores de control y los *Remote Terminal Unit* (RTU) o *Programmable Logic Controller* (PLC), siguen patrones repetitivos y prácticamente estáticos [8]. Esta característica permite describir el tráfico de red a través de patrones de comportamiento para su posterior utilización por los IDSes.

Mientras que la mayoría de los ICSes utilizan protocolos públicos, también es cierto que un número importante de ICSes utilizan protocolos propietarios. Es decir, protocolos privados cuyas especificaciones se desconocen o sólo están disponibles bajo acuerdos de confidencialidad. Actualmente, la mayoría de las soluciones y tecnologías de seguridad sólo están disponibles para protocolos públicos, lo cual pone de manifiesto la necesidad de crear herramientas de seguridad capaces de trabajar tanto con protocolos públicos como propietarios.

I-A. Contribución y organización del artículo

A lo largo de este artículo se presenta un método para la detección de anomalías de protocolos industriales tanto propietarios como públicos, que se transportan por encima del protocolo TCP/IP. La detección se realiza mediante el análisis de la carga útil o *payload* de los paquetes de red y la información de los flujos de la red. La sección II recopila el trabajo realizado en el campo de los detectores de anomalías que agrupan la carga útil de los paquetes o trabajan con protocolos cifrados. La sección III describe la arquitectura del sistema de detección de anomalías presentado. La sección IV muestra los resultados experimentales que miden el rendimiento del sistema presentado. Por último, las secciones V y VI extraen las conclusiones e identifican posibles líneas futuras de trabajo, respectivamente.

II. TRABAJOS RELACIONADOS

Existe una gran variedad de ADSes relacionados con los ICSes ([9]–[10]). La mayoría de los ADSes responden a protocolos públicos, es decir, protocolos cuyas especificaciones son conocidas. Esto permite conocer el propósito de los campos que componen cada paquete de red, lo cual hace posible saber *qué* está transportando la carga útil del paquete. Así el ADS será capaz de detectar las anomalías cada vez que el contenido se queda fuera del criterio preestablecido. Esta metodología es conocida como *Deep Packet Inspection* (DPI). Sin embargo, en el caso de protocolos propietarios la carga útil de las tramas de red es ininteligible, lo cual dificulta en gran medida el uso de la metodología DPI.

Aunque no directamente relacionados con protocolos propietarios, varios ADSes agrupan las cargas útiles de los paquetes para detectar anomalías. Estos sistemas están generalmente basados en n-gramas, cuya viabilidad para detectar anomalías en cualquier tipo de tráfico lo demostraron Bigham et al. [11], incluso en algunos casos en los que el tráfico de red está cifrado.

Anagram [12] es un detector de anomalías basado en análisis de n-gramas capaz de detectar ataques miméticos. Para este fin utiliza la randomización junto con filtros Bloom,

reduciendo de esta manera la sobrecarga de cálculo. Aunque es una mejora de PAYL [13], sin embargo Anagram presenta algunas deficiencias y es susceptible a ataques como demostraron Pastrana et al. [14].

McPAD [15] utiliza una versión modificada de análisis basado en bigramas con el objetivo de detectar octetos correspondientes a *shellcodes*. Sin embargo, McPAD, en el caso de existir ligeras diferencias entre el conjunto de aprendizaje y un ataque, no es eficiente a la hora de detectar ataques.

Hadžiosmanović et al. [4] realizan una comparación de diferentes algoritmos basados en n-gramas para el análisis de anomalías en protocolos binarios entre los que se encuentra el protocolo de control industrial Modbus. Entre los algoritmos analizados, Anagram [12] es el que mejores resultados obtiene a la hora de detectar anomalías en las pruebas realizadas con el protocolo de control Modbus.

Otra aproximación a la detección de anomalías en tráfico desconocido es la realizada por Hoeve [16], el cual presenta una metodología para detectar intrusiones en tráfico de control cifrado. Para ello no inspecciona la carga útil de los paquetes, sino que se basa en separar las inserciones de tráfico producidas por comandos, y reconocer las inserciones conocidas para luego alertar de las que no lo son. Sin embargo, a la hora de detectar anomalías en tráfico de control, es más deseable una granularidad de inspección alta [17] ya que es capaz de identificar ataques de inyección de datos.

III. DESCRIPCIÓN DEL SISTEMA

Entre los ADS mencionados en la sección II no hay ninguno que combine la información granulada de la carga útil junto con la información de flujo que posibilite la detección de anomalías en protocolos de control industrial. En esta sección presentamos una solución aplicable a protocolos propietarios y públicos, que utiliza la agrupación de los octetos de la carga útil, sin intentar interpretar su contenido, junto con la información de flujo para la detección de anomalías. Cabe mencionar que la idea principal del ADS presentado es crear un modelo de comportamiento para cada segmento de red entre los posibles RTUs y MTUs del sistema. Posteriormente cada modelo es sintetizado en un conjunto de reglas los cuales describen el comportamiento esperado, no los patrones de ataque.

La figura 1 muestra la estructura del ADS propuesto. El proceso de generación del patrón de comportamiento del tráfico de red y su sintetización en reglas se hace en modo fuera de línea, para luego utilizarlo en tiempo real. Así se establece si el tráfico de red se rige bajo unos límites preestablecidos. Seis componentes forman el ADS presentado, descritos a continuación:

Analizador de paquetes. Captura el tráfico de red y filtra los protocolos comunes tales como DNS o ARP dejando sólo el tráfico ICS relevante. Una vez obtenido y filtrado, guarda el tráfico capturado en un fichero binario con formato *pcap*.

Extractor de características. Recibe un fichero de captura como entrada y extrae todas las características necesarias para

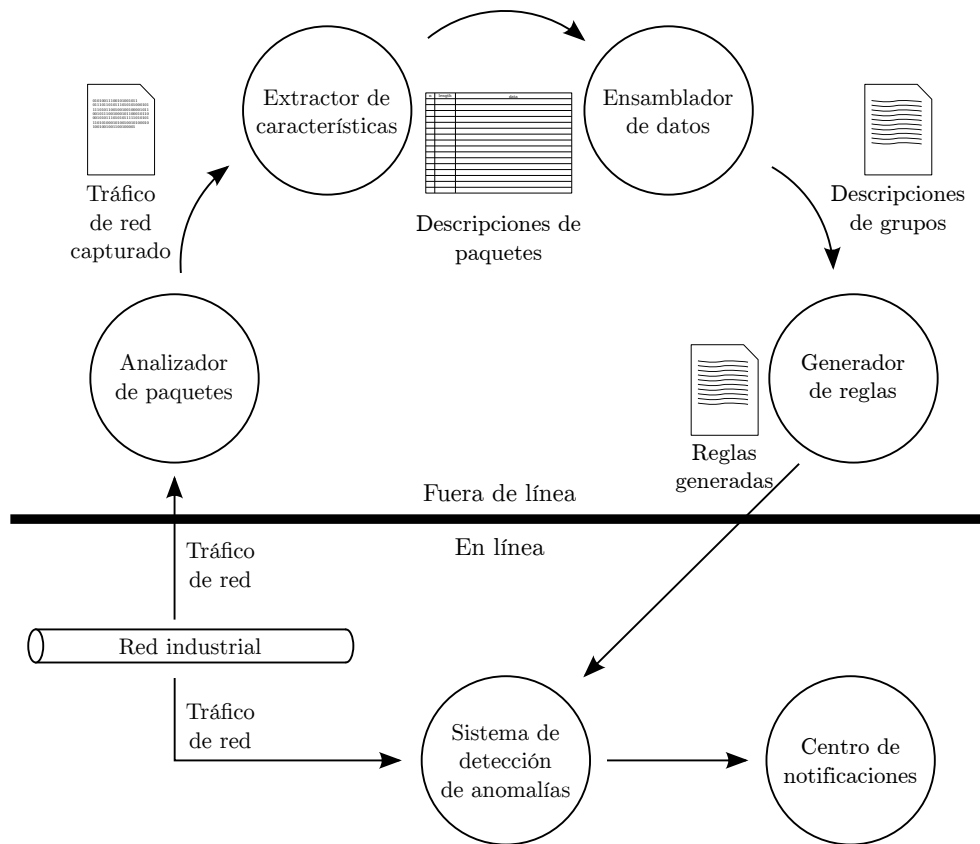


Figura 1. Componentes del Sistema de Detección de Anomalías.

crear el patrón de comportamiento que posteriormente será sintetizado en reglas. Estas son las características extraídas:

- Número de paquete
- *Timestamp*
- Dirección IP de origen y destino
- Número de puerto de origen y destino
- Longitud de la carga útil (número de octetos)
- Carga útil (cadena hexadecimal)

Ensamblador de datos. Es la pieza clave de todo el sistema y donde reside la lógica del método presentado. Su función se centra en analizar y comparar cada octeto de la carga útil de todos los paquetes que forman el mismo flujo de tráfico de red y en base a los criterios preestablecidos, formar grupos de octetos y extraer sus características. A continuación se detalla cada uno de los pasos que realiza el ensamblador de datos:

1. Ordena los paquetes según al flujo de red al que pertenecen, esto es, los clasifica en base a la dirección IP de origen y dirección IP de destino. Una vez ordenados, extrae el contenido por encima del protocolo TCP/IP, la carga útil, de cada uno de los paquetes.
2. Por cada flujo de red, organiza los octetos de la carga útil de los paquetes en filas y columnas. Cada fila contiene la carga útil de un único paquete, mientras que cada columna está formada por un único octeto, el octeto correspondiente a la posición de la columna.
3. Por cada columna compara todas las filas, esto es,

compara la misma posición de octeto de todos los paquetes y así identifica las columnas (los octetos) cuyo valor cambia entre diferentes filas (paquetes).




4. Agrupa las columnas adyacentes cuyo valor cambia entre las distintas filas. Así se crean grupos de octetos cuya longitud, n , es el número de octetos cambiantes consecutivos. La tabla I muestra cinco grupos distintos, siendo tanto el grupo dos como el cinco del flujo 2 bigramas, esto es, grupos formados por dos octetos.
5. Identifica la posición del octeto inicial de cada grupo y anota tanto la posición como el número de octetos que lo constituyen.
6. Identifica y anota todos los posibles valores de cada grupo, es decir, los distintos valores de los octetos que forman cada grupo.
7. Una vez identificados y definidos todos los grupos, la información se envía al siguiente componente del ADS, el generador de reglas.

Generador de reglas. Este componente es el encargado de sintetizar el patrón de comportamiento, cada grupo y sus características, en un conjunto de reglas. Dichas reglas serán utilizadas posteriormente por el ADS, con el fin de comparar la situación en tiempo real con el patrón de comportamiento de red y así detectar las anomalías. Nótese que cada conjunto de reglas describe el tráfico de un único proceso industrial y un único segmento de red, por ello es necesario crear un

Tabla I
EJEMPLO DE LOS GRUPOS DE OCTETOS.


Núm. paquete	Carga útil (Octetos)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	32	01	00	07	56	80	04	A0	00	00	40	61	20	...
2	32	01	00	07	66	80	05	60	00	00	40	71	20	...
7	32	01	00	07	76	80	04	A0	00	00	40	61	20	...

Flujo 1. Carga útil de los paquetes que forman el flujo dirección IP de origen 192.168.1.2 y dirección IP de destino 192.168.1.240.

 Grupo 1  Grupo 2  Grupo 3

Núm. paquete	Carga útil (Octetos)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	32	03	00	00	75	68	00	02	00	D0	00	00	04	...
4	32	03	00	00	76	68	00	02	00	90	00	00	04	...
5	32	07	00	00	00	00	00	0C	00	D6	00	01	12	...

Flujo 2. Carga útil de los paquetes que forman el flujo dirección IP origen 192.168.1.240 y dirección IP destino 192.168.1.2.

 Grupo 1  Grupo 2  Grupo 3  Grupo 4  Grupo 5

nuevo conjunto de reglas para cada segmento de red o proceso industrial, además de cuando el proceso en cuestión sufre alteraciones. Las reglas generadas siguen el formato de Snort [18], un IDS basado en el tráfico de red flexible y ligero. Las reglas permiten a una versión modificada de Snort verificar el contenido de cada paquete de red para comprobar si la carga útil contiene alguna anomalía.

Las reglas de Snort se dividen en dos secciones lógicas: la cabecera y las opciones. El propósito de la cabecera es establecer la acción que Snort deberá realizar en el caso de detectar alguna anomalía e identificar cada flujo de paquetes. Para ello, en la cabecera se definen las siguientes opciones:

- Acción de la regla. Define qué acción realizar cuando se cumplen las opciones definidas en la regla.
- Protocolo de red.
- Dirección IP de uno de los nodos del flujo.
- Máscara de red de uno de los nodos del flujo.
- Número de puerto de uno de los nodos del flujo.
- Dirección del tráfico.
- Dirección IP de otro de los nodos del flujo.
- Máscara de red de otro de los nodos del flujo.
- Número de puerto de otro de los nodos del flujo.

En el caso de las opciones de la regla, el número de palabras clave disponibles es muy elevada, a continuación se detallan aquellas que son necesarias para este caso en concreto:

- Mensaje. Define el mensaje a mostrar junto con la alerta.
- Tamaño de la carga. En este caso se indica el tamaño mínimo.
- Contenido de la carga. Permite buscar contenido específico en la carga útil del paquete. En este caso su cometido es detectar la ausencia de un valor en concreto.
- Posición del octeto inicial del patrón.
- Longitud del patrón (número de octetos).
- Flags del protocolo TCP.
- Número de identificación de la regla.
- Versión de la regla. Identifica inequívocamente revisiones

de reglas.

La tabla II muestra las dos reglas generadas para los primeros dos grupos del flujo 1 mostrados en la tabla I. La primera regla, la cual corresponde al primer grupo, verifica si los paquetes cumplen los siguientes requisitos:

- Utiliza el protocolo TCP.
- La dirección IP de origen es 192.168.1.2.
- El puerto de origen es 102.
- Está dirigido a cualquier puerto de la IP 192.168.1.240.
- El tamaño de la carga útil es mayor a ocho octetos.
- El contenido del octavo octeto no corresponde a los valores 0x56, 0x66 ó 0x76.
- Tiene los flags ACK y PUSH de TCP.

Sistema de detección de anomalías. El objetivo de este componente es comparar en tiempo real el tráfico de red con las reglas generadas en modo fuera de línea y en el caso de detectar alguna inconsistencia alertar e identificar los paquetes. Para ello, se utiliza una versión modificada de Snort [18]. La razón de modificar Snort es que, por defecto, Snort no puede valerse de las reglas creadas por el módulo generador de reglas. La mayor limitación reside en que Snort no acepta reglas con un número de opciones mayor a 256. Esto limita en gran medida el tamaño y la cantidad de miembros de los grupos generados. Debido a ello, se ha procedido a la modificación de Snort, evitando así que el número de opciones máximo sea un impedimento a la hora de la ejecución.

Centro de notificaciones. Este componente es una interfaz de usuario gráfica que posibilita el análisis de las anomalías reportadas por Snort. Su arquitectura se basa en *Basic Analysis and Security Engine* (BASE) [19].

IV. RESULTADOS EXPERIMENTALES

La validación del sistema presentado se ha realizado mediante el uso de tráfico de red real de un ICS. Cabe mencionar la dificultad de la obtención de tráfico ICS real y que esté libre de ataque. Además, hasta donde sabemos, no existe ningún

Tabla II
EJEMPLO CON LAS REGLAS CREADAS PARTIENDO DE DOS GRUPOS DE LA TABLA I.

```

alert tcp 192.168.1.2 102 -> 192.168.1.240 any (msg:"plc_group1"; dsize:8<; \
content:!"|56|"; offset:8; depth:1; \
content:!"|66|"; offset:8; depth:1; \
content:!"|76|"; offset:8; depth:1; \
flags:PA; sid:1000001; rev:1;)

alert tcp 192.168.1.2 102 -> 192.168.1.240 any (msg:"plc_group2"; dsize:10<; \
content:!"|04A0|"; offset:10; depth:2; \
content:!"|0560|"; offset:10; depth:2; \
flags:PA; sid:1000002; rev:1;)
    
```

conjunto de datos estándar para realizar pruebas de este tipo. El ICS del cual se ha obtenido el tráfico de red controla un proceso de laminación de metal. El tráfico de red ha sido capturado en dos segmentos de red distintos, lo cual permite la evaluación de la respuesta del sistema ante tráfico diferente. Las dos capturas se han realizado en condiciones libres de ataques, en un entorno sin anomalías antes, durante y después de la captura. Los Controladores Lógicos Programables (*Programmable Logic Controller*, PLC) y conmutadores de red que componen la red del ICS corresponden a las familias Siemens S7 y Siemens SCALANCE, respectivamente.

La tabla III muestra los detalles de cada una de las capturas de tráfico de red realizadas para el proceso de generación de reglas y validación del sistema presentado. Las capturas tienen una duración de entre dos horas y de dos horas y media, lo cual se traslada a un total de 309.830 y 329.742 paquetes de tráfico de red respectivamente. Una vez filtrados los paquetes, las capturas se componen 99.808 y 138.855 paquetes correspondientes al protocolo Siemens S7.

Tabla III
CAPTURAS DE RED PARA LA GENERACIÓN DE REGLAS Y VALIDACIÓN DEL SISTEMA.

Núm. captura	Duración (segundos)	Núm. total de paquetes	Núm. de paquetes útiles
1	9013	309.830	99.808
2	7198	329.742	138.855

Para el proceso de generación de reglas que describen el patrón de comportamiento del tráfico de red, se ha utilizado el 75 % de cada captura de red. La tabla IV muestra el número de paquetes utilizado para la creación del patrón de comportamiento, esto es, la creación de los grupos, así como el número de grupos generados para cada captura y su longitud máxima y mínima en número de octetos. Hay que tener en cuenta que cada grupo se sintetiza en una única regla. Una vez generadas las reglas fuera de línea, cada fichero de captura al completo ha sido comprobado con su conjunto de reglas correspondiente. El ADS ha verificado todo el tráfico, alertando al centro de notificaciones de cada una de las anomalías detectadas.

La tabla V muestra los resultados obtenidos. En ambos casos, se puede observar que el número de falsos positivos respecto al 25 % de los paquetes no utilizados en el proceso de generación de reglas es relativamente bajo, lo cual indica

Tabla IV
CARACTERÍSTICAS DE LOS PATRONES DE COMPORTAMIENTO GENERADOS.

Núm. captura	Núm. de paquetes	Núm. de grupos	Longitud mín. (octetos)	Longitud máx. (octetos)
1	74.856	52	1	71
2	104.141	6	1	227

que el conjunto de reglas creado tiene una exactitud muy alta. Si se analiza toda la captura de red, no solo el restante 25 % de los paquetes, sino el total de la captura, el número de falsos positivos se mantiene. Esto indica que el proceso de generación de reglas es capaz de sintetizar en reglas todas las posibles opciones y que el ADS no genera más falsos positivos que lo necesario. Obviamente, el porcentaje disminuye al comparar el número de falsos positivos con el total de los paquetes capturados (0,016 % y 0,052 % respectivamente). En cuanto al número de reglas activadas a causa de alguna anomalía detectada, en el caso de la primera captura sólo siete de ellas han generado una alerta. Eso significa que los 49 paquetes que han registrado un falso positivo, están concentrados en esas siete reglas. En cuanto a la segunda captura, la concentración de reglas activadas es todavía mayor, ya que pese a que hay un número mayor de paquetes falsos positivos, el número de reglas afectadas es menor. Sin embargo, porcentualmente, el número de reglas que generan falsos positivos es mayor en la segunda captura que en la primera.

Tabla V
RESULTADOS EXPERIMENTALES: ANOMALÍAS DETECTADAS.

	Núm. de captura	
	1	2
Paquetes falsos positivos (F.P.)	49	173
F.P. vs. restante 25 % de paquetes	0,2 %	0,5 %
F.P. vs. todos los paquetes	0,016 %	0,052 %
Reglas falsas positivas	7	2
F.P. vs. todas las reglas	13,46 %	33,33 %

El número de falsos positivos obtenido se debe en gran medida a la existencia de un identificador de paquete en el protocolo transportado por encima del protocolo TCP/IP. Generalmente, los protocolos de red disponen de un identificador de paquete, esto es, un número que identifica cada paquete de manera inequívoca. Este número se incrementa de manera gradual hasta llegar a un límite establecido por

el número de octetos destinados para su representación. Así, si las capturas de tráfico incluyeran todos los valores posibles del identificador de paquete, el número de falsos positivos se reduciría de manera considerable. Con lo cual, el hecho de aumentar el tiempo de captura o el porcentaje de la captura utilizada para la generación de las reglas, reduciría el número de alertas. Otra solución sería la de descartar la regla que verifica la sección del identificador de paquete, entendiendo que este campo puede contener todos los valores posibles. No aporta valor verificar un campo cuyo contenido puede ser cualquiera. Además, esta solución reduciría la carga de trabajo del ADS. Otro de los resultados a resaltar es el hecho de que el sistema ha sido capaz de detectar todos aquellos paquetes cuyo contenido se ha modificado de manera manual.

V. CONCLUSIONES

El creciente número de amenazas y el interés que despiertan los sistemas de control industrial hace necesaria su protección mediante la creación de nuevos sistemas de seguridad. En este artículo presentamos un sistema de detección de anomalías para protocolos propietarios de ICS basado en la agrupación de los octetos de la carga útil que además utiliza información de flujo de red. Los protocolos propietarios dificultan la creación de ADSes debido a la falta de las especificaciones del protocolo. Sin embargo, la carga útil del protocolo puede ser tratada como un conjunto de datos en crudo del cual se puede extraer un patrón de comportamiento habitual del sistema. La naturaleza repetitiva y estática del tráfico entre los MTUs y RTUs de los ICS hace posible que se puedan detectar anomalías, con un margen de falsos positivos relativamente bajo.

El sistema presentado consta de seis componentes, los cuales generan el patrón de comportamiento del tráfico de red, lo sintetizan en un conjunto de reglas y posteriormente, a través de un IDS modificado, detectan paquetes de red que se desvían del comportamiento habitual del sistema. La metodología descrita ha sido validada mediante dos capturas reales de tráfico de ICS, con un porcentaje de falsos positivos por debajo del 0,5%. El método es extensible a diferentes protocolos de red de control industrial, tanto propietarios como públicos, ya que el sistema se abstrae del significado de los diferentes paquetes, centrándose sólo en el contenido.

VI. TRABAJOS FUTUROS

A lo largo de este artículo, el tráfico se clasifica en base a las direcciones IP de origen y destino. La utilización de más características, como el tamaño del paquete, mejoraría el resultado del sistema de detección de anomalías. Generalmente los paquetes del mismo protocolo de un mismo tamaño tienen el mismo propósito (p. ej. los paquetes generados que actualizan el valor de una temperatura tendrán cargas útiles de la misma longitud).

Por otro lado, a lo largo de este trabajo solo se han contemplado aquellos octetos de la carga útil que cambian de valor a lo largo de distintos paquetes. La creación de grupos estáticos, es decir, secuencias de octetos idénticas a lo largo

de todos los paquetes pueden ayudar a una representación más exacta del tráfico de red. Esto haría más difícil enmascarar un ataque, ya que el atacante no sólo debería replicar el contenido dinámico de los paquetes, sino también el estático, dificultando así la inserción de código malicioso.

REFERENCIAS

- [1] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," *White paper, Symantec Corp., Security Response*, 2011.
- [2] B. Bencsáth, G. Pék, L. Buttyán, and M. Félégyházi, "Duqu: Analysis, detection, and lessons learned," in *ACM European Workshop on System Security (EuroSec)*, 2012.
- [3] B. Miller and D. Rowe, "A survey SCADA of and critical infrastructure incidents," in *Proceedings of the 1st Annual conference on Research in information technology*, pp. 51–56, ACM, 2012.
- [4] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zamboni, and S. Etalle, "N-gram against the machine: On the feasibility of the n-gram network analysis for binary protocols," in *Research in Attacks, Intrusions, and Defenses*, pp. 354–373, Springer, 2012.
- [5] C. Tankard, "Advanced Persistent threats and how to monitor and deter them," *Network security*, vol. 2011, pp. 16–19, August 2011.
- [6] Keith Stouffer, Joe Falco, and Karen Scarfone, "Guide to Industrial Control Systems (ICS) Security, Special publication 800-82," tech. rep., National Institute of Standards and Technology, June 2011.
- [7] B. Galloway and G. Hancke, "Introduction to Industrial Control Networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 860–880, 2012.
- [8] R. Chandia, J. Gonzalez, T. Kilpatrick, M. Papa, and S. Sheno, "Security Strategies for SCADA Networks," in *Critical Infrastructure Protection (E. Goetz and S. Sheno, eds.)*, vol. 253 of *IFIP International Federation for Information Processing*, pp. 117–131, Springer US, 2008.
- [9] I. Garitano, R. Uribeetxeberria, and U. Zurutuza, "A review of SCADA anomaly detection systems," in *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*, pp. 357–366, Springer, 2011.
- [10] B. Zhu and S. Sastry, "SCADA-specific intrusion detection/prevention systems: a survey and taxonomy," in *Proceedings of the 1st Workshop on Secure Control Systems (SCS)*, 2010.
- [11] J. Bigham, D. Gamez, and N. Lu, "Safeguarding SCADA Systems with Anomaly Detection," in *Computer Network Security (V. Gorodetsky, L. Popyack, and V. Skormin, eds.)*, vol. 2776 of *Lecture Notes in Computer Science*, pp. 171–182, Springer Berlin Heidelberg, 2003.
- [12] K. Wang, J. J. Parekh, and S. J. Stolfo, "Anagram: A content anomaly detector resistant to mimicry attack," in *Recent Advances in Intrusion Detection*, pp. 226–248, Springer, 2006.
- [13] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection*, pp. 203–222, Springer, 2004.
- [14] S. Pastrana, A. Orfila, J. E. Tapiador, and P. Peris Lopez, "Randomized Anagram revisited," *Journal of Network and Computer Applications*, 2014.
- [15] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, "McPAD: A multiple classifier system for accurate payload-based anomaly detection," *Computer Networks*, vol. 53, no. 6, pp. 864 – 881, 2009. Traffic Classification and Its Applications to Modern Networks.
- [16] M. Hoeve, "Detecting Intrusions in Encrypted Control Traffic," in *Proceedings of the First ACM Workshop on Smart Energy Grid Security, SEGS '13*, (New York, NY, USA), pp. 23–28, ACM, 2013.
- [17] D. Hadžiosmanović, D. Bolzoni, S. Etalle, and P. Hartel, "Challenges and opportunities in securing industrial control systems," in *Complexity in Engineering (COMPENG)*, 2012, pp. 1–6, June 2012.
- [18] M. Roesch, "Snort – Lightweight Intrusion Detection for Networks," in *Proceedings of LISA '99: 13th Systems Administration Conference (USENIX, ed.)*, vol. 99, (Seattle, WA, USA), pp. 229–238, November 1999.
- [19] A. Flores, J. Fields, A. Graham, J. Hart, K. Johnson, D. Mackie, S. Muller, T. Rupp, C. Svensson, and M. Valdez, "Basic Analysis and Security Engine," 2010. Online. Accedido el 14 de Marzo de 2014 <http://base.professionallyevil.com>.