



TecnoLógicas
ISSN: 0123-7799
ISSN: 2256-5337
tecnologicas@itm.edu.co
Instituto Tecnológico Metropolitano
Colombia

Predicción de ciberataques en sistemas industriales SCADA a través de la implementación del filtro Kalman

Quiroz Tascón, Stephen; Zapata Jiménez, Julián; Vargas Montoya, Héctor Fernando

Predicción de ciberataques en sistemas industriales SCADA a través de la implementación del filtro Kalman

TecnoLógicas, vol. 23, núm. 48, 2020

Instituto Tecnológico Metropolitano, Colombia

Disponible en: <https://www.redalyc.org/articulo.oa?id=344263272013>

DOI: <https://doi.org/10.22430/22565337.1586>
2020



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional.

Predicción de ciberataques en sistemas industriales SCADA a través de la implementación del filtro Kalman

Predicting Cyber-Attacks in Industrial SCADA Systems Through The Kalman Filter Implementation

Stephen Quiroz Tascón

Instituto Tecnológico Metropolitano, Colombia

stephenquiroz@gmail.com

 <http://orcid.org/0000-0002-5726-757X>

DOI: <https://doi.org/10.22430/22565337.1586>

Redalyc: <https://www.redalyc.org/articulo.oa?id=344263272013>

Julián Zapata Jiménez

Instituto Tecnológico Metropolitano, Colombia

Julianx16@gmail.com

 <http://orcid.org/0000-0002-2978-6858>

Héctor Fernando Vargas Montoya

Instituto Tecnológico Metropolitano, Colombia

hectorvargas@itm.edu.co

 <http://orcid.org/0000-0002-0861-2883>

Recepción: 28 Enero 2020

Aprobación: 21 Abril 2020

RESUMEN:

En los sistemas industriales SCADA (Supervisory Control And Data Acquisition), conocer el estado de cada dispositivo permite obtener información de su comportamiento. De esta forma se pueden deducir acciones y conformar estrategias diferentes que ayuden a reducir el riesgo cibernético. En este artículo de investigación aplicada, se presenta un modelo de predicción de posibles ciberataques en un sistema SCADA. Dicha predicción se hace con un filtro Kalman. Un filtro Kalman procesa los eventos de ciberseguridad capturados a través de un sistema de detección de intrusos (aplicado en un sistema de simulación de SCADA) y genera una proyección futura de la probabilidad de que se consolide un ataque. Con esta información, los administradores de sistemas podrán tomar alguna decisión sobre cómo actuar frente a inminentes ataques informáticos. Se realizó una instalación de diferentes componentes tecnológicos y se ejecutaron 3 ataques informáticos al SCADA: (i) posibles escaneos, (ii) robo de información y (iii) sobrescritura de comandos y datos generando Denial of Service o DoS. los eventos de seguridad fueron detectados por un sistema de detección de intrusos y enviados a un software configurado con las características del filtro Kalman para entregar como salida las posibles predicciones de ataques. Como resultado, se puede ver cómo a partir de las entradas es posible conocer la probabilidad de que un ataque informático sea exitoso con base en los eventos históricos y las fórmulas aplicadas del filtro.

PALABRAS CLAVE: Ataque informático, ciberseguridad, filtro Kalman, Control de supervisión y adquisición de datos, sistema de detección de intrusos.

ABSTRACT:

In industrial SCADA (Supervisory Control and Data Acquisition) systems, knowing the status of each device allows information to be collected on its behavior. In this way, actions can be deduced, and different strategies can be formed to help reduce cyber risk. In this article of applied research, a model of prediction of possible cyber-attacks in a SCADA system is presented. This prediction is made with a Kalman filter. A Kalman filter processes cyber security events captured through an intrusion detection system (applied in a SCADA simulation system) and generates a future projection of the probability of an attack being carried out. With this information, system administrators will be able to make some decisions about how to act against imminent cyber-attacks. An installation of different technological components was carried out and 3 cyberattacks to the SCADA were executed: (i) possible scans, (ii) theft of information and (iii) command and data overwriting generating Denial of Service or DoS. The security events were detected by an intrusion detection system and sent to a software, setup with Kalman filter features to deliver as output the

possible predictions of attacks. As a result, the probability of a successful computer attack can be seen from the entries based on the historical events and the applied filter formulas.

KEYWORDS: Cyber-attack, cyber-security, intrusion detection system, kalman filter, Supervisory Control and Data Acquisition.

1. INTRODUCCIÓN

Las tecnologías de información y comunicaciones, TIC, permiten a las organizaciones establecer procesos más efectivos, pero esto implica que los datos e información van en circulación por las redes y los sistemas de almacenamiento, y cualquier problema puede generar un riesgo para los procesos. Los datos y la información en las compañías son más que datos de ventas en valor, volúmenes en kilos, unidades vendidas, estadísticas que hacen referencia al comportamiento del mercado o hábitos de compra de sus consumidores. Incluyen también los que se gestan al interior del negocio, como los datos que se crean o generan a partir de su funcionamiento, por ejemplo, datos de clientes o la información de la maquinaria industrial, que es de suma importancia para la compañía, porque a partir de dicha información, se puede conocer el comportamiento de cada elemento de su estructura en tiempo real.

De acuerdo con la Asociación Colombiana de Ingenieros de Sistemas, ACIS [1] en su encuesta anual 2018-2019 se encontró que el 26 % de las organizaciones deben pensar en la ciberseguridad industrial, lo que lleva a establecer mecanismos que permitan la reducción de los ciberriesgos como un proceso sistémico.

Así mismo, acorde con el Instituto Nacional de Ciberseguridad de España, INCIBE [2], los ataques informáticos han ido en aumento en los últimos años, teniendo como objetivo infectar y obtener información de los sistemas industriales, en donde, para el año 2017, una amenaza persistente avanzada, APT, conocida como Triton, infectó varias compañías, cambiando el comportamiento de algunos equipos de protección.

Esta situación de riesgo puede aumentar en la medida en que las empresas que usan sistemas industriales para sus procesos crecen y se vuelven complejas. Por ello, ante un evento de seguridad (afectación a la disponibilidad, confidencialidad o integridad), es necesario que se activen los procedimientos y protocolos para la atención de dichos incidentes. Los procesos de atención a eventos de seguridad pueden ser clave en aquellas compañías cuyo objeto de negocio depende de los sistemas industriales, toda vez que estos, poseen una función específica (muy electrónica) que cada día se va acercando más a las redes de computadores y de telecomunicaciones para su gestión.

Para las organizaciones que cuentan con procesos industriales, la información de la maquinaria industrial debe partir del conocimiento y comportamiento de cada elemento de su estructura en tiempo real.

Estos datos son generados por PLC (Controlador Lógico Programable) [3], administrados de forma gráfica (pantalla) por los HMI [4] (Interfaz Hombre Máquina) y sistemas como SCADA [5].

Si estos componentes se conectan a las redes de computadores, existe la posibilidad de ataques informáticos, por lo cual, se hace necesario establecer mecanismos que ayuden a la identificación y reducción de los riesgos de ciberataques.

Este documento presenta una investigación aplicada de un modelo de predicción con base en diferentes ataques informáticos en sistemas industriales, identificados desde un sistema de detección de intrusos y aplicando un filtro Kalman.

En consecuencia, en los primeros capítulos se presenta un marco teórico sobre los sistemas SCADA, los sistemas de detección de intrusos y el filtro Kalman; luego tenemos la metodología que se ha seguido para llegar a los resultados y discusiones de las pruebas y, finalmente, las conclusiones del ejercicio.

1.1 Los sistemas SCADA

Siendo necesario el control de los PLC, Supervisory Control And Data Acquisition (SCADA) es una aplicación software de control de producción, que interactúa con los dispositivos externos de campo y ejerce control en los procesos automáticamente desde la pantalla de cualquier computador.

También entrega información del proceso a los usuarios dependiendo del rol, por ejemplo, operadores, supervisores de calidad, supervisión en general, mantenimiento, entre otros [6].

Los sistemas SCADA recolectan datos que se procesan con el fin de determinar si hacen parte de los niveles de tolerancia y si es necesario ejecutar medidas preventivas.

La arquitectura básica y genérica para un sistema SCADA está compuesta por los PLC, uno o varios servidores, las consolas desde donde se visualiza y opera el sistema y un servidor [7][8].

Hace algunos años, los sistemas SCADA eran más seguros contra las intrusiones y ataques que sufrían las redes de la organización, pero esto no quiere decir que estos sistemas fuesen mucho más resistentes, sino que estaban aislados y eran inaccesibles desde las redes administrativas o internet. En consecuencia, ataques de negación de servicio (DoS) afectan directamente la disponibilidad de los sistemas, siendo catastróficos para las empresas si se logran materializar (frenando la producción).

Actualmente, muchas empresas manejan sus sistemas industriales desde un segmento de red separado e implícitamente seguro, aun cuando en el mismo computador donde se administre el sistema haya instalado un programa de servicio de correo electrónico y con acceso a internet.

Diferentes ataques, vulnerabilidades y esfuerzos en ciberseguridad se han venido presentando en los últimos años[9] [10].

Algunos como las caídas de la planta de energía de Ucrania en el año 2015, considerado un ataque cibernético que afectó a 225 000 clientes, aproximadamente; o los esfuerzos de la Universidad Estatal de Washington (WSU) para crear un banco de pruebas con respecto a ciberataques para infraestructuras críticas.

Un método de detección de intrusión física para ICS [11] se propone como mecanismos de detección y control, pero solo cuando ya se ha ejecutado el ataque informático. Esto se hace mediante el análisis de la señal de comunicación en la capa física. Es evidente lo difícil que es evitar que los atacantes se introduzcan físicamente en el sistema de control industrial (ICS), y que además puedan conectar dispositivos externos al sistema para extraer información o inyectar datos falsos.

1.2 Los sistemas de detección de intrusos, IDS

Estos sistemas permiten la identificación de posibles ataques informáticos. Para ello se analiza la actividad de las redes, sistemas y servicios informáticos por accesos no autorizados o actividades maliciosas [12]. Los IDS pueden ser de hardware o de software, permitiendo cubrir diferentes necesidades empresariales a la hora de elegir un sistema de detección y monitoreo. También ayuda a los sistemas a identificar cualquier comportamiento anómalo que se encuentre en la red, monitoreando la actividad, validando las configuraciones de los servicios y eventualmente encontrando vulnerabilidades, así como detectando ataques tales como DoS/DDoS, Backdoors, Troyanos, entre otros. En particular para los SCADA, es necesario que se logren identificar escaneos, lectura de información y sobrescritura de datos y comandos dado el impacto que esto puede tener sobre dichos sistemas. Por esta razón, un IDS debe poder tener una tasa de detección cercana al 100 %, reduciendo los falsos positivos.

SNORT es un sistema de detección de intrusos de red, que usa código abierto, y que monitoriza el tráfico y luego lo verifica contra un conjunto de reglas que se configuran previamente. Existen 3 diferentes tipos de reglas, como reglas de la comunidad, reglas registradas y reglas comerciales, todas disponibles en su página web

[13]. SNORT permite la detección y alerta de eventos de seguridad para muchos protocolos y servicios, tales como bases de datos, servicio Web, servicios de correo, protocolos TCP y UDP, malformación de paquetes, negación de servicio, entre otros.

1.3 Filtro Kalman

Rudolf E. Kalman desarrolló un método matemático en 1960 que describe una solución basada en una función recursiva para el problema del filtrado lineal de datos discretos [14]. Dicho filtro hace parte de un contexto de modelos estado-espacio, donde el núcleo es la proyección o estimación que se hace con mínimos cuadrados recursivos. Dicha representación del sistema se define mediante un conjunto de variables que son denominadas estados. Cada estado tiene la información del sistema en cierto punto en el tiempo y esta información debe permitir la deducción de los diferentes comportamientos basándose en el pasado del sistema (así predice su comportamiento futuro).

El filtro Kalman es usado en muchos prototipos y mediciones, con el fin de realizar validaciones o ajustes de desviaciones [15]. A partir de las mediciones, dichas configuraciones permiten establecer salidas proyectadas que pueden ser interpretadas como base de los mismos prototipos o para ambientes reales. Así mismo, los filtros son usados para la optimización de procesos computacionales desde la predicción, medición y estimación de variables [16].

Esto se lleva a cabo desde unas mediciones iniciales combinadas con algunos datos estadísticos, permitiendo analizar el comportamiento de las mediciones en varios puntos del proceso.

Para el caso de este artículo, el filtro Kalman se usó para predecir comportamientos futuros con base en las entradas detectadas desde el IDS-SNORT, a partir de la ejecución de los ataques informáticos hacia el ambiente simulado de SCADA.

2. METODOLOGÍA

Obtener resultados de predicción supone establecer una serie de fases que permitan la configuración de las diferentes herramientas técnicas, así como el análisis de los resultados. Para eso, se consideraron las siguientes fases (Fig. 1).

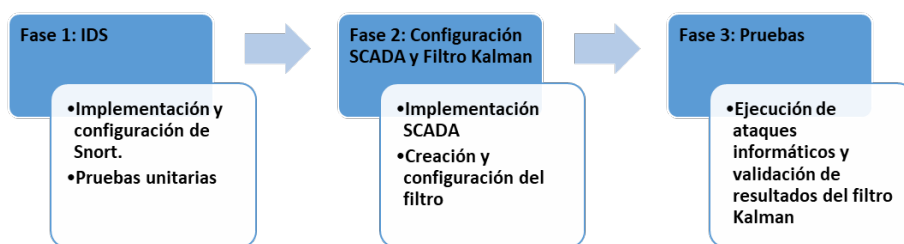


FIG. 1.
Fases de la metodología

Fuente: elaboración propia.

Es necesario contar con las herramientas que permitan la detección de diferentes ataques informáticos. En ese sentido, fue necesario definir una arquitectura y la implementación de los sistemas respectivos.

2.1 Sistema de detección de intrusos

Se configuró una red virtual y dentro de esta, un IDS-SNORT en Linux Ubuntu, así como las reglas para la detección de los 3 ataques informáticos generados: (i) posibles escaneos, (ii) robo de información y (iii) sobrescritura de comandos y datos generando negación de servicio, DoS.

Dichos ataques informáticos fueron seleccionados en consideración a los pilares de la seguridad y en representación del proceso base de una prueba de intrusión, que es iniciar con un escaneo de puertos (obtención de información), intentar robar información (pérdida de confidencialidad) y afectar el funcionamiento clave del SCADA (pérdida de disponibilidad) a través de un ataque de DoS.

En ese sentido, se hizo una prueba unitaria que validara que las reglas creadas sí se activan, esto es, ejecutando algunos de los ataques informáticos ya indicados desde la distribución Kali Linux.

En consideración a la problemática de los falsos positivos, el mecanismo para su reducción en el IDS fue la configuración exacta de la firma que corresponde al ataque generado y la prueba unitaria ejecutada para su validación. Por consiguiente, una vez se ejecutó la prueba, el IDS hizo una validación exacta con la firma y generó la respectiva alerta. Este mecanismo permitió la reducción de los falsos positivos y negativos, dado que solo se ejecutaron los ataques ya mencionados, en donde el IDS hace hit con la firma respectiva, obteniendo un grado de confiabilidad en alto grado.

2.2 Configuración del simulador SCADA y el filtro Kalman

Cuando se obtienen los resultados de la posible detección de los intrusos, es necesario establecer, acorde al modelo de predicción, cuál puede ser el comportamiento en el futuro de los riesgos asociados a dicha identificación.

En consecuencia, se implementó el filtro Kalman a través de un programa software construido en Python 3.7 [17], dada la versatilidad de este lenguaje de programación y la facilidad de integración con los sistemas Linux. El filtro Kalman se compone de 2 fases.

En la primera fase, por lo general llamada la fase de predicción, se genera un pronóstico del estado futuro en el tiempo, tomando en cuenta toda la información disponible en ese momento, que, para el caso, se trata de los ataques informáticos que puedan acontecer en un sistema y almacenando un vector de estados, representado en (1):

$$\hat{X}_k = Fx_{k-1} + Bu_{k-1} + W_k \quad (1)$$

En donde:

- \hat{X}_k Es un vector de estados
- K Es el momento o instante en que se toma la muestra (tiempo)
- Fx_{k-1} Matriz de transición en el instante
- Bu_{k-1} Matriz de estimación del estado anterior ($k-1$)
- W_k es el ruido con un valor promedio igual a cero y con varianza con valores aleatorios

Luego se determina la matriz de covarianza de error, la cual representa el aprendizaje para la corrección de errores a través de (2):

$$P_k = FP_{k-1} + Q_{k-1} \quad (2)$$

En donde:

- P_k Fase de corrección
- FP_{k-1} Estimación de la covarianza del error asociada a la estimación a priori
- Q_{k-1} Es la medición con el valor anterior de la predicción ($k-1$)

En la segunda fase, denominada la fase de corrección, se calcula un pronóstico mejorado del estado, de tal manera que el error es minimizado estadísticamente, para lo cual se corrige en la matriz de covarianza del error y la diferencia del filtro Kalman se calcula para minimizar el error en la estimación del nuevo estado a través de (3):

$$K_k = P_k \cdot H_k^T (H_k P_k \cdot H_k^T + R_k)^{-1} \quad (2)$$

En donde:

- K_k Disminución y corrección del resultado del filtro Kalman

- H_k / H_k^T Es la matriz que indica la relación entre las mediciones y el vector de estado al momento k hasta el t , en el supuesto que no hubiera ruido en la medición

- R_k La matriz de covarianza del ruido de las mediciones (depende de los sensores)

Finalmente, la variable de estado se complementa con la medición del ruido añadido al sistema con (4):

$$Z_k = H_k X_k + V_k \quad (4)$$

En donde:

- Z_k Es la medida para comparar el filtro de predicción en cada instante de tiempo t .

- V_k es el ruido con un valor promedio igual a cero y con varianza con valores aleatorios.

Predecir un evento futuro supone, entonces, lograr obtener datos presentes y aplicar las diferentes fórmulas Kalman para con ello tener un estimado de lo que puede suceder. El proceso general para aplicar los filtros se muestra en (Fig. 2).

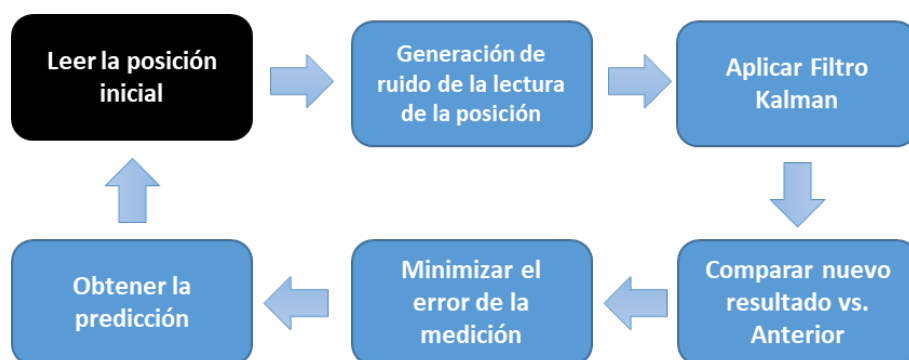


FIG. 2.

Proceso general en la aplicación de un filtro Kalman

Fuente: elaboración propia.

Así mismo, y considerando que la implementación real de un sistema SCADA es de suma complejidad técnica y económica por la naturaleza del hardware requerido, se usó Conpot [18], [19], [20] que es un honeypot simulador de un sistema SCADA, bajo un servidor interactivo, diseñado para que sea implementado y configurado de una forma virtualizada y se pueda modificar y extender de acuerdo con la necesidad de cada proyecto. Igualmente, desde el honeypot se pueden proporcionar diferentes protocolos de comunicación como Modbus TCP, SNMP y HTTP. Además, es capaz de simular una infraestructura crítica compleja, como es el sector eléctrico.

También puede brindar la posibilidad de conectarse a una interfaz máquina hombre o HMI (Human Machine Interfaz, por sus siglas en inglés) personalizada para emular un sistema real.

Dado que Conpot tiene las funciones básicas de un sistema industrial, se podría limitar a hacer múltiples y diversas pruebas de seguridad sobre este. En ese sentido, Conpot no contempla algunos componentes como temporizadores, entradas y salidas analógicas desde sensores, respaldo en memoria para recuperarse ante cortes de energía, entre otras, que sí están en los sistemas SCADA reales [21].

Así mismo, al ser una máquina virtual, dependerá de los recursos (CPU-memoria) del sistema anfitrión, diferente a un SCADA real, cuyo hardware está creado para su alto procesamiento; sin embargo, para las pruebas de seguridad acá estipuladas, las funciones base cumplen con los requerimientos.

2.3 Pruebas de validación

Los comportamientos de posibles intrusiones pueden ser anómalos o no, y por ello se integró el filtro Kalman al IDS (que es la fuente para el filtro) y se validó que el IDS estuviera en red con el sistema Conpot. Con ello, se realizaron las diferentes pruebas de seguridad (3 ataques informáticos: negación de servicio o DoS, sobreescritura de datos y escaneo del servicio) y de funcionalidad del filtro en el sistema integrado, obteniendo los respectivos resultados. Parte de los resultados implica identificar la probabilidad de que pueda ocurrir en el futuro y con ello establecer posibles rutas de mitigación de ataques (esto no se tratará en este documento).

Se creó un *dashboard* (panel de mando) en donde se puede visualizar el funcionamiento del sistema, el cual genera alertas basado en mensajes. En consecuencia, el mensaje 513 es de un posible ataque de negación de servicios o DoS, el 514 es un ataque de tipo escritura y el 515 es un ataque de tipo lectura.

3. RESULTADOS Y DISCUSIÓN

3.1 Configuración del IDS—SNORT

Para iniciar las implementaciones, se define la siguiente arquitectura (Fig. 3). La configuración de SNORT (versión snort-2.9.9.0) es la recomendada por el proveedor [22], la cual puede descargarse de manera gratuita desde el portal Web. Luego de su instalación se verifica su funcionamiento (Fig. 4).

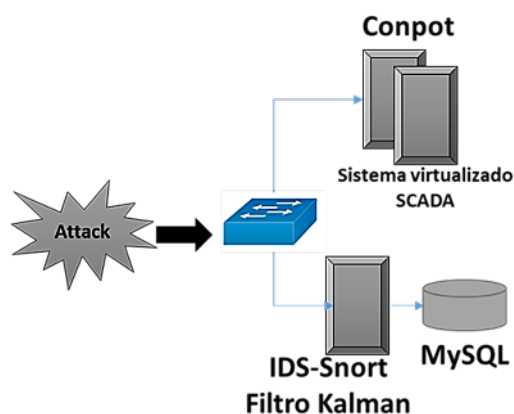


FIG. 3.
Arquitectura de implementación
Fuente: elaboración propia.

```

root@UMB:~# snort -v
Running in packet dump mode

--= Initializing Snort ==--
Initializing Output Plugins!
pcap BPF configured to passive.
Acquiring network traffic from "ens33".
Decoding Ethernet

--= Initialization Complete ==--

--> Snort! <--
Version 2.9.9.0 GRE (Build 56)
By Martin Roesch & The Snort Team: http://www.snort.org/contactteam
Copyright (C) 2014-2016 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.7.4
Using PCRE version: 8.38 2015-11-23
Using ZLIB version: 1.2.8

```

FIG. 4.

Verificación de SNORT funcionando

Fuente: elaboración propia.

Se instala la herramienta Barnyard2 [23] así como la base de datos MySQL [24] (ambas herramientas también son de uso libre), para que los datos generados a partir de la detección del IDS, sean almacenados en una base de datos estructurada y de esta manera poder realizar consultas de forma efectiva y ágil.

Para configurar el IDS, se crearon las siguientes reglas en el archivo de configuración default.rules de SNORT:

a. Regla 1: Escaneo

```
alert icmp $HOME_NET any -> any any (msg:"ICMP Detectado"; GID:1; sid:50000001; rev:001; classtype:icmp-event;)
```

b. Regla 2: Conexión Modbus

```
alert tcp $HOME_NET any -> 10.1.1.13 502 (content:"|02|"; offset:7; depth:1; flow:established, to_server; msg:"Conexión Modbus"; sid:1000001; rev:0; priority:5;)
```

c. Regla 3: Escritura de datos en PLC

```
alert tcp any any -> 10.1.1.13 502 (msg: "Escribiendo datos en PLC"; content:"|0f|"; offset:7; depth:1; sid:1111102; rev:2; priority:5;)
```

d. Regla 4: Lectura de datos del PLC

```
alert tcp any any -> 10.1.1.13 502 (msg:"Leyendo datos del PLC"; content:"|01|"; offset:7; depth:1; sid:1111103; rev:2; priority:3)
```

Para las pruebas unitarias de funcionamiento del sistema, se realizó un escaneo de puertos con la herramienta de uso libre nmap. Así se observa en (Fig. 5) cómo se va almacenando la información en la base de datos:

```

mysql> select count(*) from event;
+-----+
| count(*) |
+-----+
| 482542 |
+-----+
1 row in set (0.08 sec)

```

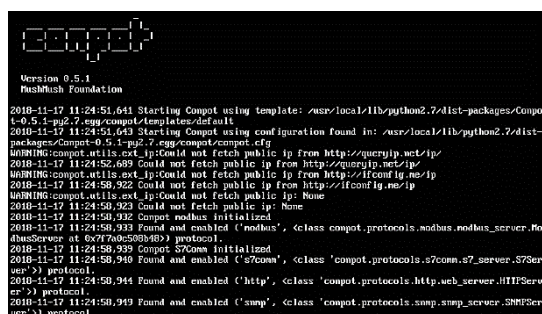
FIG. 5.

Número de eventos guardados en MySQL

Fuente: elaboración propia.

3.2 Configuración de Conpot y creación del filtro Kalman

Para la instalación y configuración de Conpot [25], primero se debe clonar el proyecto desde el repositorio github y luego verificar su correcto funcionamiento (Fig. 6) ejecutando el siguiente comando: `$ sudo Conpot -t default:`



```

Version 0.5.1
PushMush Foundation

2018-11-17 11:24:51.641 Starting Compot using template: /usr/local/lib/python2.7/dist-packages/Compo
t-0.5.1-py2.7.egg/compot/templates/default
2018-11-17 11:24:51.643 Starting Compot using configuration found in: /usr/local/lib/python2.7/dist-
packages/Compot-0.5.1-py2.7.egg/compot/compot.cfg
WARNING:compot.utils.ext_ip:Could not fetch public ip from http://queryip.net/ip/
2018-11-17 11:24:52.689 Could not fetch public ip from http://queryip.net/ip/
WARNING:compot.utils.ext_ip:Could not fetch public ip from http://ifconfig.me/ip
2018-11-17 11:24:58.922 Could not fetch public ip from http://ifconfig.me/ip
WARNING:compot.utils.ext_ip:Could not fetch public ip: None
2018-11-17 11:24:58.923 Could not fetch public ip: None
2018-11-17 11:24:58.932 Compot radius initialized
2018-11-17 11:24:58.933 Found and enabled 'modbus', <class 'compot.protocols.modbus.modbus_server.No
dusServer at 0x7f70a550b4b0'> protocol.
2018-11-17 11:24:58.939 Compot s7comm initialized
2018-11-17 11:24:58.940 Found and enabled 's7comm', <class 'compot.protocols.s7comm.s7_server.S7Ser
ver'> protocol.
2018-11-17 11:24:58.944 Found and enabled 'http', <class 'compot.protocols.http.web_server.HTTPSer
ver'> protocol.
2018-11-17 11:24:58.949 Found and enabled 'snmp', <class 'compot.protocols.snmp.snmp_server.SNMPSer
ver'> protocol.

```

FIG. 6.

Pantalla inicial del sistema virtualizado SCADA

Fuente: elaboración propia.

Para la creación del filtro Kalman, se debe crear inicialmente una tabla en la base de datos MySQL antes creada, la cual servirá como fuente de datos de extracción hacia la herramienta desarrollada en Python. Los datos de entrada (Fig. 7) hacia el filtro serán tomados de los eventos reportados por el IDS.

```

def inicio_kalman_xy():
    x = np.matrix('0. 0. 0. 0.').T
    P = np.matrix(np.eye(4))*1000 # Matriz de incertidumbre
    N = 20
    intAtaques=3
    v_sql = np.zeros((intAtaques,N))
    v_kalman = np.zeros((N,intAtaques))
    v_fecha = ['1','2','3','4','5','6','7','8','9','0', '1','2','3','4','5','6','7','8','9','0']
    true_x = np.arange(1, N + 1, 1)

```

FIG. 7.

Datos de entrada Filtro Kalman

Fuente: elaboración propia.

El programa que recibe los datos de entrada contiene las variables “X”, que es el vector de estado inicial; “P”, que es la matriz de incertidumbre; “N”, que es número de resultados que se desean evaluar simultáneamente, e “IntAtaques”, que es el número de incidentes o ataques que se están simulado.

Una vez capturados los valores, estos son entregados al filtro Kalman, (Fig. 8) el cual los procesa y entrega información con la predicción de los posibles ataques informáticos. Es importante aclarar que una de las ventajas que posee el filtro Kalman es la reducción del error en cada iteración, esto es, una reducción de la desviación estándar (σ) en la predicción.

```
def kalman(x, P, measurement, R, motion, Q, F, H):
    # Actualización de x, P basado en medición m
    # distancia entre la posición actual y la predicción
    y = np.matrix(measurement).T - H * x
    S = H * P * H.T + R # covarianza residual
    K = P * H.T * S.I # Aplicación de nuevo del filtro Kalman
    x = x + K*y
    I = np.matrix(np.eye(F.shape[0])) # matriz de identidad
    P = (I - K*H)*P

    # predicción x, P basado en la variación
    x = F*x + motion
    P = F*P*F.T + Q

    return x, P
```

FIG. 8.

Salida del Filtro Kalman.

Fuente: elaboración propia.

Luego de terminar la ejecución, la herramienta desarrollada en Python genera una salida (Fig. 9), de acuerdo con el ataque identificado.

```
def encabezado():
    #print("\r\n ")
    print("\r\n ")
    print(" F I L T R O   K A L M A N ")
    print("-----")
    print(" | 513 Es un ataque DoS")
    print(" | 514 Es un ataque Escritura")
    print(" | 515 Es un ataque Lectura")
    print(" | Pred Muestra la predicción del filtro kalman")
    print("----- \r\n")
```

FIG. 9.

Salida del Filtro Kalman.

Fuente: elaboración propia.

3.3 Pruebas funcionales de seguridad

Se ejecuta el primer ataque que corresponde a un evento de escritura, donde se realiza una modificación en los datos del PLC con datos aleatorios y este proceso se repite por 100 veces (Fig. 10).

```

root@kali:~/Escritorio/Ataques# python3.5 pEscrituraModBUS.py
¿Cuántas veces quiere relizar la escritura de datos? 100

Inicio de escritura de datos 11:32:29
Escritura número: 1 Hora: 11:32:29
Escritura número: 2 Hora: 11:32:29
Escritura número: 3 Hora: 11:32:29
Escritura número: 4 Hora: 11:32:29
Escritura número: 5 Hora: 11:32:30
Escritura número: 6 Hora: 11:32:30
Escritura número: 7 Hora: 11:32:30
Escritura número: 8 Hora: 11:32:30
Escritura número: 9 Hora: 11:32:30
Escritura número: 10 Hora: 11:32:30
Escritura número: 11 Hora: 11:32:31

```

FIG. 10.

Ataque 1. Sobrescribir datos al PLC Siemens 2700 con protocolo Modbus

Fuente: elaboración propia.

Como se observa, este evento tiene una tasa de 5 incidentes por segundo, que comprometen la integridad de la información.

Un segundo ataque es lanzado (Fig. 11), este consiste en leer toda la información que se aloja en el PLC comprometiendo la confidencialidad de los datos en él registrados. Dicho evento, lee la información del PLC de forma aleatoria, desde la posición % M100 hasta la % M114 y puede volcar los datos de los bloques de información, donde el atacante puede realizar una copia completa de los datos del dispositivo. Este evento también se ejecuta 100 veces con una velocidad de 4 lecturas por segundo.

```

root@kali:~/Escritorio/Ataques# python3.5 pLecturaModBUS.py
¿Cuántas veces quiere relizar la lectura de datos? 100

Inicio de lectura de datos 11:35:49
%M100      0
Lectura número: 1 Hora: 11:35:49
%M100      0
%M101      0
%M102      1
%M103      1
%M104      1
%M105      0
%M106      1
%M107      1
%M108      0
%M109      1
%M110      0
%M111      0
%M112      1
%M113      0
%M114      1
Lectura número: 2 Hora: 11:35:50
%M100      0
%M101      0
%M102      1
%M103      1
%M104      1
%M105      0
Lectura número: 3 Hora: 11:35:50
%M100      0
%M101      0
%M102      1
%M103      1
%M104      1
%M105      0

```

FIG. 11.

Ataque 2. Lectura de datos al PLC Siemens 2700 con protocolo Modbus

Fuente: elaboración propia.

Luego del segundo ataque (Fig. 11), desde el lado del sistema SCADA (Conpot), se evidencia cómo se captura cada intento del “atacante” por leer y sobrescribir la información que se encuentra alojada en el PLC Siemens 2700 a través del LOG, donde se destacan los siguientes elementos: la dirección IP y el ID de la máquina que se conecta al PLC del sistema SCADA, puerto, hora de conexión, hora de desconexión, entre otras opciones (Fig. 12).

```

2019-06-08 11:53:23,166 Modbus response sent to 10.1.1.12
2019-06-08 11:53:23,169 Modbus client disconnected. (74a637d8-1bd7-4686-9747-ed8499e2bf30)
2019-06-08 11:53:23,279 New Modbus connection from 10.1.1.12:58828. (74a637d8-1bd7-4686-9747-ed8499e2bf30)
2019-06-08 11:53:23,293 Modbus traffic from 10.1.1.12: {'function_code': 15, 'slave_id': 1, 'request': '0001000000009010f0064000a02d400', 'response': '0f0064000a'} (74a637d8-1bd7-4686-9747-ed8499e2bf30)
2019-06-08 11:53:23,298 Modbus response sent to 10.1.1.12
2019-06-08 11:53:23,300 Modbus client disconnected. (74a637d8-1bd7-4686-9747-ed8499e2bf30)
2019-06-08 11:53:23,364 New Modbus connection from 10.1.1.12:58830. (74a637d8-1bd7-4686-9747-ed8499e2bf30)
2019-06-08 11:53:23,376 Modbus traffic from 10.1.1.12: {'function_code': 1, 'slave_id': 1, 'request': '0001000000006010100640014', 'response': '0103d45003'} (74a637d8-1bd7-4686-9747-ed8499e2bf30)
2019-06-08 11:53:23,381 Modbus response sent to 10.1.1.12
2019-06-08 11:53:23,385 Modbus client disconnected. (74a637d8-1bd7-4686-9747-ed8499e2bf30)
2019-06-08 11:53:23,498 New Modbus connection from 10.1.1.12:58832. (74a637d8-1bd7-4686-9747-ed8499e2bf30)

```

FIG. 12.

Comportamiento del sistema SCADA Laboratorio CONPOT

Fuente: elaboración propia.

La generación de esta información se da uno a uno por cada evento que ocurra con el sistema SCADA (de 20 a 30 por segundo, aproximadamente) y no puede ser analizada por el personal técnico dada su velocidad de entrega. Por tal razón, los datos son capturados por el IDS SNORT para ser almacenados en la base de datos MySQL que se configuró anteriormente y es analizada por el visualizador.

Ahora que el sistema SCADA se encuentra bajo ataque, el IDS está capturando y almacenando toda la información generada por el atacante.

Estos datos son agrupados y seleccionados para que se inicie el proceso de predicción con el filtro Kalman, el cual se alimenta de toda la información que se encuentra en la base de datos y se muestra en pantalla en un lenguaje simple para que sea analizado por el personal técnico.

En este tablero se muestra la cantidad de eventos que se están materializando en tiempo real (Fig. 13). La información es actualizada segundo a segundo para que se obtenga toda la trazabilidad de los sucesos que ocurren dentro del sistema SCADA, mostrando las siguientes variables:

FILTRO KALMAN									

513	Es un ataque Dos								
514	Es un ataque Escritura								
515	Es un ataque Lectura								
Pred	Muestra la predicción del filtro kalman								

2019-06-08 12:01:57	0006	Real 513 = 0.0	Pred -3.35	Real 514= 0.0	Pred 0.00	Real 515= 1.0	Pred -2.07		
2019-06-08 12:01:56	0006	Real 513 = 0.0	Pred -1.28	Real 514= 4.0	Pred 6.69	Real 515= 3.0	Pred 3.07		
2019-06-08 12:01:55	0006	Real 513 = 0.0	Pred -0.49	Real 514= 3.0	Pred 3.46	Real 515= 3.0	Pred 3.25		
2019-06-08 12:01:54	0006	Real 513 = 0.0	Pred -0.19	Real 514= 3.0	Pred 3.14	Real 515= 4.0	Pred 4.86		
2019-06-08 12:01:53	0006	Real 513 = 0.0	Pred -0.07	Real 514= 4.0	Pred 4.72	Real 515= 3.0	Pred 2.67		
2019-06-08 12:01:52	0006	Real 513 = 0.0	Pred -0.03	Real 514= 4.0	Pred 4.34	Real 515= 5.0	Pred 6.36		
2019-06-08 12:01:51	0006	Real 513 = 0.0	Pred -0.01	Real 514= 5.0	Pred 6.00	Real 515= 5.0	Pred 5.47		
2019-06-08 12:01:50	0006	Real 513 = 0.0	Pred -0.00	Real 514= 5.0	Pred 5.35	Real 515= 5.0	Pred 5.19		
2019-06-08 12:01:49	0006	Real 513 = 0.0	Pred -0.00	Real 514= 4.0	Pred 3.62	Real 515= 4.0	Pred 3.42		
2019-06-08 12:01:48	0006	Real 513 = 0.0	Pred -0.00	Real 514= 5.0	Pred 5.74	Real 515= 5.0	Pred 5.62		
2019-06-08 12:01:47	0006	Real 513 = 0.0	Pred -0.00	Real 514= 5.0	Pred 5.14	Real 515= 5.0	Pred 5.14		
2019-06-08 12:01:46	0006	Real 513 = 0.0	Pred -0.00	Real 514= 6.0	Pred 6.64	Real 515= 6.0	Pred 6.82		
2019-06-08 12:01:45	0006	Real 513 = 0.0	Pred -0.00	Real 514= 6.0	Pred 6.52	Real 515= 5.0	Pred 4.98		
2019-06-08 12:01:44	0006	Real 513 = 0.0	Pred -0.00	Real 514= 6.0	Pred 6.20	Real 515= 6.0	Pred 6.43		
2019-06-08 12:01:43	0006	Real 513 = 0.0	Pred -0.00	Real 514= 6.0	Pred 6.21	Real 515= 6.0	Pred 6.33		
2019-06-08 12:01:42	0006	Real 513 = 0.0	Pred -0.00	Real 514= 6.0	Pred 6.26	Real 515= 6.0	Pred 6.33		
2019-06-08 12:01:41	0006	Real 513 = 0.0	Pred -0.00	Real 514= 6.0	Pred 6.04	Real 515= 6.0	Pred 6.33		
2019-06-08 12:01:40	0006	Real 513 = 0.0	Pred -0.00	Real 514= 7.0	Pred 7.99	Real 515= 5.0	Pred 4.59		
2019-06-08 12:01:39	0006	Real 513 = 0.0	Pred -0.00	Real 514= 5.0	Pred 4.17	Real 515= 6.0	Pred 6.35		
2019-06-08 12:01:38	0006	Real 513 = 0.0	Pred -0.00	Real 514= 6.0	Pred 6.31	Real 515= 6.0	Pred 6.09		

FIG. 13.

Ataque 2. Resultados de la lectura *Dashboard* de visualización

Fuente: elaboración propia.

-*Fecha*: en este campo se muestra el momento exacto en que ocurre el evento.

-*513 ataque DoS*: muestra la cantidad de eventos sucedidos por el escaneo de IP y puertos que se realizan dentro del sistema SCADA.

-*514 ataque de escritura*: es la cantidad de veces que un atacante sobrescribe los datos dentro del PLC, a través del puerto 502 de comunicación Modbus

-*515 ataque de lectura*: es la cantidad de veces que los datos del PLC han sido leídos y accedidos.

-*Predicción del filtro Kalman-Pred*: de acuerdo con la información histórica generada por los eventos de seguridad, el filtro actúa y muestra cuál sería el estado futuro por cada uno de los eventos que se encuentren configurados.

Por otro lado, para obtener los porcentajes de predicción y sus tendencias, se tiene:

-*Porcentaje de predicción-Pp*: es la relación de la predicción con respecto al valor real (Fig. 13), esto es (5):

$$Pp = 1 - (Pred - Real) * 100\% \quad (5)$$

Si el porcentaje está por encima del 100 % se debe descartar, dado que podría ser un falso positivo; así mismo se descarta si el valor es negativo, dado que no se tienen datos desde el IDS.

-*Desviación estándar-σ*: para obtener la tendencia del comportamiento (Fig. 13), se aplica la siguiente fórmula (6):

$$\sigma = ((Pred/Real) - 1) * 100\%K \quad (6)$$

-*Parámetro de medición*: en consideración a que la fuente son los diferentes ataques en tiempo real y que el objetivo es alertar de acuerdo con ese nivel de ataque, se espera que la tasa de predicción se aproxime al 100 % y que la desviación estándar sea cercana a cero (0), generando una reducción de falsos positivos y entregando una mayor certeza en la predicción de los posibles ataques.

Tras la ejecución del programa, el *dashboard* (Fig. 13) imprime los eventos que están sucediendo segundo a segundo dentro del sistema SCADA, registrando todos los posibles eventos de seguridad.

Para interpretar la información, esta debe ser analizada de forma vertical por cada evento que muestra el tablero, obteniendo el porcentaje de predicción acorde a la relación, de la siguiente manera:

En el caso del ataque 513 (que es un ataque de DoS), se identifica la palabra “Real”, esto hace referencia a la cantidad de eventos que se están registrando dentro del IDS SNORT; la columna “Pred” es la predicción que realiza el filtro Kalman, de acuerdo al historial que se va generando en la columna “Real”, cuyos eventos se encuentran almacenados en la base de datos del sistema SCADA, dado que en los 19 segundos de muestra, (Fig. 13) el IDS no detectó ningún evento relacionado con este tipo de ataque. El resultado o valor de predicción es negativo, porque el dato de entrada al filtro Kalman es cero.

Es importante recordar que dicho filtro recibe tres variables de entrada para realizar la predicción, los cuales son: dato real, predicción anterior y la desviación estándar (que va disminuyendo) y de esta manera mejora la predicción en cada iteración.

En los ataques 514 y 515, se observa cómo el IDS-SNORT va registrando los posibles eventos de seguridad, donde se identifica que estos van incrementando a medida que transcurre el tiempo y, a su vez, el filtro Kalman recibe estos datos para calcular la predicción, la tendencia y la intensidad de los ataques. Así mismo, podemos obtener el porcentaje de predicción para cada ataque, donde el ataque 514 (Fig. 14) logra resultados hasta un 98 % en la predicción con una tendencia progresiva ascendente.

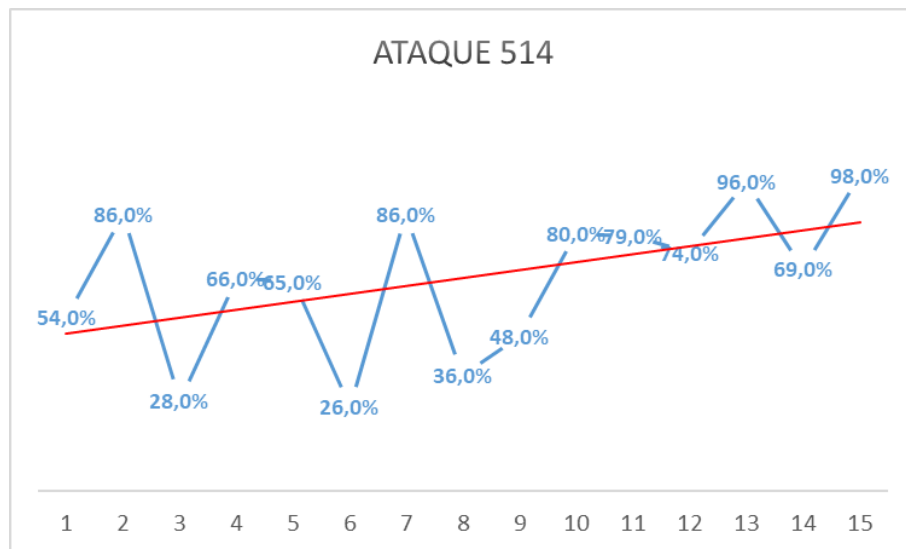


FIG. 14.
Porcentaje y tendencia de predicción para el ataque 514
Fuente: elaboración propia.

De igual forma, para el ataque 515 (Fig. 15) se tienen picos de 96 % de predicción con la misma tendencia a la aproximación buscada (100 %), esto puede ayudar al personal de seguridad a interpretar de forma más asertiva que una inspección física en los elementos tecnológicos.

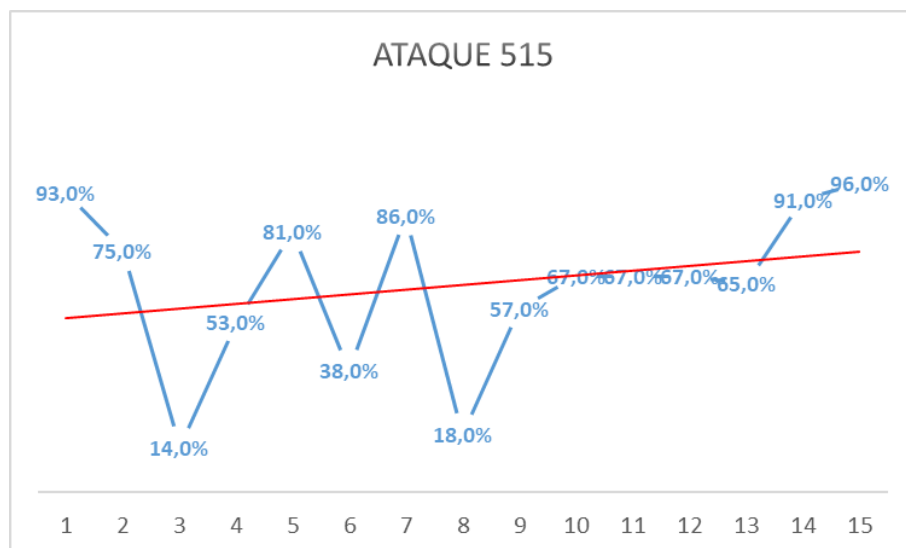


FIG. 15.
Porcentaje y tendencia de predicción para el ataque 515
Fuente: elaboración propia.

Ahora bien, con respecto a la reducción en la desviación estándar para el ataque 514 (Fig. 16) y el 515 (Fig. 17), es claro que dicha reducción del error tiene una tendencia a cero (0), siendo el ataque 514 el de mejor tendencia y más rápida convergencia a la reducción del error en la predicción.

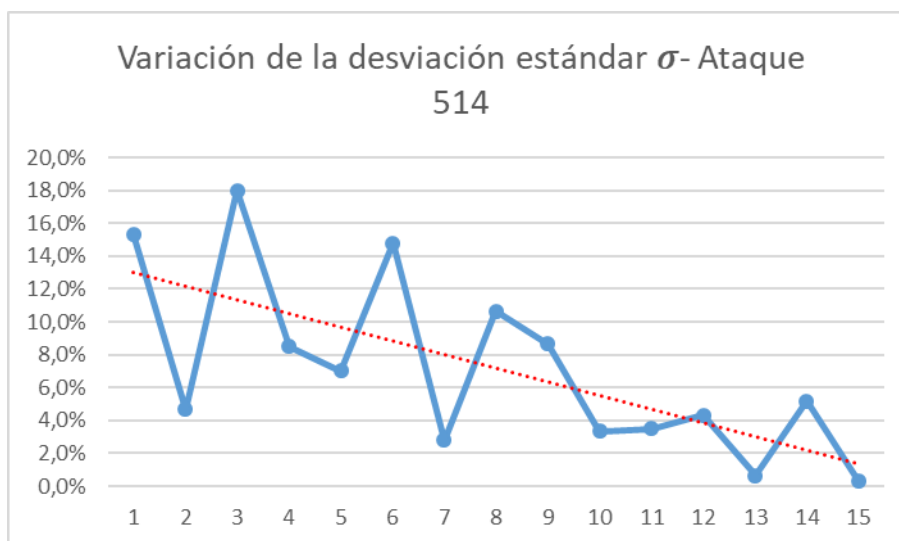


FIG. 16.

Tendencia de la desviación estándar para el ataque 514

Fuente: elaboración propia.



FIG. 17.

Tendencia de la desviación estándar para el ataque 515

Fuente: elaboración propia.

3.4 Discusión

Teniendo en cuenta que los riesgos son dinámicos en el tiempo, esta primera aproximación a la ejecución de solo tres ataques, permite visualizar un amplio rango de posibles eventos de seguridad que se pueden generar. Esto dependerá de la cantidad de reglas configuradas en el IDS y el potencial de aprendizaje del filtro Kalman, por lo cual, es posible ampliar el rango de ejecución de amenazas bajo una previa configuración de las reglas de identificación (en consideración a los falsos positivos y negativos, estos sí podrían variar acorde a la confiabilidad del IDS usado, dado el aumento y variabilidad de ataques).

Con respecto a otros procesos de seguridad sobre los sistemas SCADA, varias técnicas de detección y mitigación hacen uso de Machine Learning (ML) [26], mostrando una alta precisión en la detección de ataques, incluyendo DDoS.

Dichas pruebas se realizaron utilizando el conjunto de datos KDD'Cup99 para las técnicas de árbol de decisión, algoritmo de Random Forest (RF) y método Naive Bayes (NB). Como resultado, el clasificador RF fue el mejor con una ocurrencia del 99.99 %, mientras que NB obtuvo un 97.74 %.

Respecto a los resultados anteriores, y teniendo en cuenta que el filtro Kalman tuvo la mejor probabilidad de predicción (98 %) para ataques 514 (ataque de escritura) y mejor tendencia a la reducción del error, es claro que el filtro Kalman tiene mucho potencial y puede verse en igualdad de condiciones con respecto al uso de ML, con la diferencia de que el uso de Kalman se realizó sobre datos reales de ataques en línea (ejecutados y recolectados por un IDS en tiempo real).

Por otro lado, el uso de un árbol de ataque que potencializa la negación de servicio DDoS, permite identificar diferentes estrategias de control que ayudan a la mitigación de los riesgos.

Dicha mitigación se centra en la identificación de objetivos de ataque, así como conocer sus vectores y proponer la mitigación [27], que con respecto al uso de filtro Kalman, supone una visión estratégica diferente, dado que considera una realidad en el ataque y posible control, mientras que el uso del filtro Kalman toma la realidad del ataque y proyecta la probabilidad que este se repita en un futuro inmediato, acorde a su comportamiento.

Así mismo, en otros resultados utilizando algoritmos de aprendizaje supervisado [28] sobre un *dataset*, se validó a través de los clasificadores J48, IBK (Instance Based Learning), NB (Naive Bayes) y MPL (Multilayer Perceptron), la posible predicción de ataques en cinco categorías: Normal, Denial of Service (DOS), R2L (Unauthorized Access from Remote Machines), Probe, U2R (User to Root Attacks). Como resultado, el clasificador J48 obtuvo mejor predicción que los demás, con un 99.71 % de asertividad, seguido por NB con 99.43 % y MPL con un 98.57 %. En relación con la probabilidad obtenida en el filtro Kalman para los ataques en SCADA, podemos visualizar que esta primera aproximación tiene similares resultados de asertividad que el uso de clasificadores, pero con una tendencia de reducción de error importante sobre ataques reales (y no sobre datos estáticos).

Por último, los sistemas SCADA que apalancan los procesos de energía se vienen conectando cada vez más a las redes de computadores y, como resultado, se plantean algunas aproximaciones en términos de vulnerabilidades y como desde un diseño de arquitectura de red y sus componentes, las organizaciones pueden buscar la mitigación de diferentes ataques informáticos [29]. Esta estrategia puede apoyar de forma consistente las predicciones generadas desde el filtro Kalman, una vez sean identificado uno o varios ataques informáticos, generando acciones preventivas que permitan ajustar las diferentes arquitecturas de seguridad.

Todos los componentes instalados son de libre uso, por lo cual, es posible su montaje en cualquier momento, si se quieren recrear situaciones similares o aumentar el número de ataques para así poder tener resultados más amplios en la simulación de un SCADA.

4. CONCLUSIONES

Poder predecir posibles eventos de seguridad les permitirá a las organizaciones gestionar de manera más proactiva los riesgos en sus sistemas industriales. La prevención como elemento fundamental en los planes de tratamiento de riesgos permitirán establecer diferentes rutas de actuación para lograr mitigar posibles ciberataques.

A partir de la medición de los 3 ataques informáticos generados, se puede establecer una predicción temprana con una reducción del error tendiente a cero (0) para los ataques de escritura y lectura, permitiendo que el porcentaje de predicción tenga una tendencia hacia el logro del objetivo (llegar al 100 %), siendo el ataque 514 el de mejor convergencia.

Con ello, las personas puedan visualizar los posibles impactos generados en los sistemas industriales, por lo que el uso del filtro Kalman, en una primera aproximación, puede apoyar la identificación de posibles eventos de seguridad que puedan impactar negativamente la seguridad. Con ello, los administradores podrán

visualizar diferentes estrategias que le ayuden a actuar si el evento identificado se materializa o, mejor aún, establecer mecanismos de control para que no se materialicen.

En comparación con otros procesos investigativos y sus resultados, y considerando que el uso del filtro Kalman se aplicó en una red real, obtener un 98 % en la predicción con tendencia de error a cero (0) sobre ataques en línea, es un valor muy relevante (solo 1 punto porcentual por debajo de algunos resultados [26] [28] ejecutados con datos estáticos). Esto permite establecer un método funcional que puede ser afinado y utilizado para actuar frente a posibles eventos de seguridad.

Contar con sistemas de detección de intrusos que puedan identificar ataques a través de reglas o alertas en los sistemas industriales SCADA en tiempo real, permite tener las entradas necesarias hacia el filtro Kalman y así contar con una herramienta de predicción. La efectividad de la respuesta depende de la afinación misma de la entrada (el IDS) y de cómo las reglas configuradas permiten esa identificación de ataques informáticos.

Si bien el *honeypot* usado tiene diferentes limitaciones con respecto a los SCADA reales, las pruebas realizadas dentro del ambiente controlado permitieron establecer resultados acordes a las reglas configuradas en el IDS y a la programación del filtro Kalman. por ello, en este acercamiento los resultados podrían ser mejorados si se tiene un ambiente físico real.

En futuros trabajos es importante establecer un procedimiento para el manejo de incidentes de seguridad, que permita, a partir de las alertas tempranas, establecer una serie de pasos para la reducción de riesgos de exposición (manejo de futuros incidentes de seguridad). Así mismo, se propone el uso del filtro Kalman para otras plataformas y tecnologías, en donde se puedan tener una predicción de posibles ciberataques.

CONFLICTOS DE INTERÉS DE LOS AUTORES

Stephen Quiroz Tascón declara explícitamente que: “no tengo ningún tipo de conflicto de intereses que pueda influir de forma inapropiada en los resultados obtenidos o las interpretaciones propuestas”. Julián Zapata Jiménez declara explícitamente que: “no tengo ningún tipo de conflicto de intereses que pueda influir de forma inapropiada en los resultados obtenidos o las interpretaciones propuestas”. Héctor Fernando Vargas Montoya declara explícitamente que: “no tengo ningún tipo de conflicto de intereses que pueda influir de forma inapropiada en los resultados obtenidos o las interpretaciones propuestas”.

REFERENCIAS

- [1] A. R. Almanza J., “XIX Encuesta Nacional de Seguridad Informática Evolución del perfil del profesional de seguridad digital,” *Rev. sistemas*, no. 151, pp. 12–41, Jun.. 2019. <https://doi.org/10.29236/sistemas.n151a3>
- [2] Instituto Nacional de ciberseguridad (INCIBE), “Las claves de los últimos ataques en sistemas de control industrial,” 2018. Disponible en: <https://www.incibe-cert.es/blog/las-claves-los-ultimos-ataques-sistemas-control-industrial>
- [3] M. Ramirez, E. Miilán y V. Moreno “Herramienta para programar un controlador lógico programable basado en hardware reconfigurable”. *RIELAC*, Vol.22, Apr. 2011, pp.65 – 77. Disponible en: <http://rielac.cujae.edu.cu/index.php/riac/article/view/83>
- [4] A. Romero-Acero, A. Marín-Cano, y E. I. Arango-Zuluaga, “Plataformas de Laboratorio de Bajo Costo Basadas en el Protocolo ZigBee,” *TecnoLógicas*, pp. 411-423, Nov. 2013. <https://doi.org/10.22430/22565337.367>
- [5] M. Annor- y B. Pranggono, “Development of Smart Grid Testbed with Low-Cost Hardware and Software for Cybersecurity Research and Education,” *Wirel. Pers. Commun.*, vol. 101, no. 3, pp. 1357–1377, Apr. 2018. <https://doi.org/10.1007/s11277-018-5766-6>
- [6] E. Carozo Blumsztein y L. Vidal, “Sistemas SCADA, algunas recomendaciones de seguridad – Parte II,” *Revista. Seguridad* no. 19 Sep. 2013. Disponible en: <https://revista.seguridad.unam.mx/printpdf/2190>
- [7] D. J. Kalbfleisch, “SCADA Technologies and Vulnerabilities” Dec. 2013, pp. 1- 7. Disponible en: <http://www.cs.tufts.edu/comp/116/archive/fall2013/dkalbfleisch.pdf>

- [8] K. Coffey, R. Smith, L. Maglaras, y H. Janicke, "Vulnerability Analysis of Network Scanning on SCADA Systems," *Secur. Commun. Networks*, vol. 2018, pp. 1–21, Mar. 2018. <https://doi.org/10.1155/2018/3794603>
- [9] C.-C. Sun, A. Hahn y C.-C. Liu, "Cyber security of a power grid: State-of-the-art," *Int. J. Electr. Power Energy Syst.*, vol. 99, pp. 45–56, Jul. 2018. <https://doi.org/10.1016/j.ijepes.2017.12.020>
- [10] L. A. Maglaras et al., "Cyber security of critical infrastructures," *ICT Express*, vol. 4, no. 1, pp. 42–45, Mar-2018. <https://doi.org/10.1016/j.ict.2018.02.001>
- [11] P. Liu y T. Liu, "Physical Intrusion Detection for Industrial Control System," en *2018 IEEE Conference on Communications and Network Security (CNS)*, Beijing, 2018, pp. 1–2. <https://doi.org/10.1109/CNS.2018.8433194>
- [12] A. Warzynski y G. Kolaczek, "Intrusion detection systems vulnerability on adversarial examples," in *2018 Innovations in Intelligent Systems and Applications (INISTA)*, Thessaloniki, 2018, pp. 1–4. <https://doi.org/10.1109/INISTA.2018.8466271>
- [13] R. Teja Gaddam y M. Nandhini, "An analysis of various snort based techniques to detect and prevent intrusions in networks proposal with code refactoring snort tool in Kali Linux environment," en *2017 International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, 2017, pp. 10-15. <https://doi.org/10.1109/ICICCT.2017.7975177>
- [14] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *J. Basic Eng.*, vol. 82, no. 82, pp. 35-45, 1960. Disponible en: <http://www.unitedthc.com/DSP/Kalman1960.pdf>
- [15] C. D. Zuluaga-Ríos, M. A. Álvarez-López y A. A. Orozco-Gutiérrez, "A comparison of robust Kalman filtering methods for artifact correction in heart rate variability analysis," *TecnoLógicas*, vol. 18, no. 34, pp. 25-35, Jan. 2015. <https://doi.org/10.22430/22565337.213>
- [16] F. Baker y S. Thennadil, "Constrained Kalman Filtering: Improving Fused Information Retention During Constraining," en *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR)*, Międzyzdroje, Poland, 2019, pp. 434-437. <https://doi.org/10.1109/MMAR.2019.8864655>
- [17] Python Software Foundation "Python.org." Disponible en: <https://www.python.org/>
- [18] Honey.net.org, "CONPOT – Low interaction serverside ICS honeypot," 1990 - 2019 Accessed: 11-Nov-2019. Disponible en: <https://www.honeynet.org/projects/active/conpot/>
- [19] A. Jicha, M. Patton, H. Chen "SCADA honeypots: An in-depth analysis of Conpot." En *2016 IEEE Conference on Intelligence and Security Informatics (ISI) Tucson*. 2016 pp. 196-198. <https://doi.org/10.1109/ISI.2016.7745468>
- [20] MushMush Foundation Revision 1891107c "Welcome to Conpot's documentation!" — Conpot 0.6.0 documentation." Disponible en: <https://conpot.readthedocs.io/en/latest/index.html>
- [21] Siemens 2008, "SIMATIC - Manual del sistema de automatización S7-200". Número de referencia del manual: 6ES7298--8FA24--8DH0. Disponible en: http://www.west-l.com/uploads/tdpdf/s7-200_esp_man.pdf
- [22] Cisco, "SNORT Software", 2019.. Accessed: 11-Aug-2019. Disponible en: <https://www.snort.org/documents>
- [23] Barnyard2, "Bbarnyard2 Configuration." Disponible en: <https://github.com/firnsy/barnyard2>
- [24] Oracle Corporation, "MySQL Workbench versions 5.6", 2020. Disponible en: <https://www.mysql.com/>
- [25] S. A. Tovar Balderas Conpot: honeypot de sistemas de control industrial" *Revista .seguridad*, no 29. Jun. 2017. Disponible en: <https://revista.seguridad.unam.mx/numero29/conpot-honeypot-de-sistemas-de-control-industrial>
- [26] F. A. Alhaidari and E. M. AL-Dahasi, "New Approach to Determine DDoS Attack Patterns on SCADA System Using Machine Learning," en *2019 International Conference on Computer and Information Sciences (ICCIS)*, Sakaka, Saudi Arabia, 2019, pp. 1-6. <https://doi.org/10.1109/ICCISci.2019.8716432>
- [27] A. E. M. AL-Dahasi y B. N. Abbas Saqib, "Attack tree Model for Potential Attacks Against the SCADA System," en *2019 27th Telecommunications Forum (TELFOR)*, Belgrade, Serbia, 2019, pp. 1-4. <https://doi.org/10.1109/TELFOR48224.2019.8971181>

- [28] G. MeeraGandhi, "Machine Learning Approach for Attack Prediction and Classification using Supervised Learning Algorithms". *Int. J. Comput. Sci. Commun* Vol. 1, no. 2, Jul. 2010, pp. 247-250. Disponible en: <http://csjournals.com/IJCSC/PDF1-2/51..pdf>
- [29] T. Abdelghani, "Industrial control systems (ics) security in power transmission network," en *2019 Algerian Large Electrical Network Conference (CAGRE)*, Algiers, Algeria, 2019, pp. 1-4. <https://doi.org/10.1109/CAGRE.2019.8713289>

NOTAS

CONTRIBUCIÓN DE LOS AUTORES

¹ Conceptualización, validación, revisión y ajustes, supervisión, pruebas técnicas, adquisición de datos, ajustes y revisión final, correcciones de editorial y de evaluadores.

² Conceptualización, validación, recursos, revisión y ajustes, pruebas técnicas, adquisición de datos, ajustes y revisión final, correcciones de editorial y de evaluadores

³ Conceptualización, metodología, análisis formal del trabajo, escritura, revisión de resultados, formato y presentación, ajustes y revisión final, correcciones de editorial y de evaluadores.

INFORMACIÓN ADICIONAL

Cómo citar / How to cite: S. Quiroz Tascón, J. Zapata Jiménez, H. F. Vargas Motoya, "Predicción de ciberataques en sistemas industriales SCADA a través de la implementación del filtro Kalman", *TecnoLógicas*, vol. 23, no. 48, pp. 249-267, 2020. <https://doi.org/10.22430/22565337.1586>

ENLACE ALTERNATIVO

<https://revistas.itm.edu.co/index.php/tecnologicas/article/view/1586> (html)