PRUEBAS DE PENETRACIÓN A INVERSOR INTELIGENTE

INTELLIGENT INVERSOR PENETRATION TEST

Maximiliano Valencia Moctezuma
Transformadores Inteligentes de México
idi@trafos.mx

Sandra Verónica Moreno Tapia
Transformadores Inteligentes de México
idi@trafos.mx

Resumen

Los sistemas SCADA eran considerados como seguros debido a que se encontraban aislados del exterior. Actualmente, estos sistemas operan conectados al exterior y utilizan protocolos de comunicación abierta, estandarizada y con pocas o nulas medidas de seguridad. Para poder aumentar la integración de sistemas de energía renovables y tener un manejo eficiente de la energía se planea hacer uso de los sistemas SCADA. Las vulnerabilidades de los sistemas SCADA junto con las de los dispositivos conectados a la red local del usuario representan un riesgo a la red eléctrica. En este trabajo se presentan pruebas de penetración en un ambiente controlado a un inversor inteligente bajo los mismos ataques que se utilizan en los sistemas de Tecnologías de la Información.

Palabra(s) Clave: DER, Pruebas de penetración, SCADA.

Abstract

SCADA systems were considered safe because they were isolated from the outside. Currently, these systems operate connected to the exterior and use open, standardized communication protocols with little or no security measures. To increase the integration of renewable energy systems and have an efficient management of energy, it is planned to make use of SCADA systems. The vulnerabilities of the SCADA systems together with those of the devices connected to the user's local network represent a risk to the electricity network. In this paper

we present penetration tests in a controlled environment to an intelligent inverter under the same attacks that are used in conventional information systems.
Keywords**:** *DER, Penetration testing, SCADA.*

1. Introducción

Los sistemas SCADA son utilizados para el monitoreo y control remoto de procesos en diferentes industrias como es la eléctrica, manufacturera e hidráulica. La finalidad es controlar y monitorear diferentes procesos distribuidos en diferentes ubicaciones desde una estación central la cual puede leer datos y enviar señales de control en tiempo real a dispositivos remotos. Los comandos de control pueden ser automáticos o manuales dados por un operador.

Los protocolos de comunicación utilizados en los sistemas SCADA anteriormente eran principalmente protocolos propietarios lo que daba un falso sentido de seguridad debido a que no se tenía información sobre los protocolos utilizados y la planta se encontraba aislada del exterior (Cagalaban, 2009; GAO, 2004; Krutz, 2006; Ralston, Graham, y Hieb, 2007); pero en la actualidad los sistemas SCADA funcionan conectados con el exterior y utilizan protocolos de comunicación abiertos y estandarizados. Los protocolos de comunicación SCADA más utilizados son (Wiles, 2008): IEC 60870-5-101, DNP3, Modbus y IEC 61850. Los protocolos de comunicación del IEC (acrónimo en inglés, International Electrotechnical Commision) son más utilizados en Europa mientras que en América del Norte es más utilizado DNP3 (Makhija y Subramanyan, 2003). Los protocolos de comunicación mencionados son protocolos abiertos y estandarizados, además, son protocolos que comúnmente utilizan Ethernet y TCP/IP. En cuanto a seguridad, los protocolos de comunicación SCADA tienen pocas o nulas medidas de seguridad y la información es transmitida sin encriptación ni autenticación.

Los DER (acrónimo en inglés, Distributed Energy Resources) son una parte fundamental de las redes inteligentes (en inglés, Smart Grid) ya que permiten la integración de fuentes de energía renovables de manera distribuida. El inversor es el dispositivo con mayor desarrollo en normatividad debido a su papel clave en la integración y penetración de DER. Para poder incrementar la penetración de DER,

se deben agregar de características inteligentes al inversor los cuales son llamados inversores inteligentes y el concepto es planteado en el estándar IEC 61850-7-420; el uso de inversores inteligentes puede duplicar la integración de DER en la red eléctrica (Seal, 2013).

El EPRI (Electrical Power Research Institute), NIST (National Institute of Standards and Technology) y el estado de California, EU han definido que un inversor inteligente debe tener funciones autónomas básicas y avanzadas para el manejo de energía (CEC, 2014; EPRI, 2016; NIST, 2012) así como un sistema de comunicaciones interoperable que consta del modelo de información y funciones estándar definidas en los estándares IEC 61850-7-420 y IEC 61850-90-7 y el uso de los protocolos de comunicación SCADA DNP3, Modbus, SEP2 y IEC 61850 MMS. Esto permite estandarizar las operaciones e información utilizadas entre los DER y operadores de la red eléctrica entre las que se encuentran conexión/desconexión de la red indicado por el operador de la red, envío de información sobre el precio de la energía eléctrica, entre otras. Lo que permite ejecutar funciones de soporte a la red eléctrica y realizar operaciones complejas del mercado eléctrico en tiempo real. Esta conectividad vuelve vulnerables a los inversores que, de manera conjunta con otros dispositivos comprometidos, pueden representar un riesgo para la estabilidad de la red, o en menor grado puedan causar daños a los dispositivos eléctricos dentro de la red eléctrica de los usuarios.

Las prácticas de seguridad que son implementadas actualmente en los sistemas SCADA generan muchas vulnerabilidades en el sistema, algunas prácticas son contraseñas sencillas o nulas, falta de actualización de programas y sistemas operativos, soluciones de seguridad con configuración por default, entre otras (Zhu, Joseph, y Sastry, 2011). En (Federal Office for Information Security, 2016; GE, 2012; NERC, 2007) podemos encontrar las 10 mayores vulnerabilidades de los sistemas SCADA y en (ICS-CERT, 2018) podemos encontrar algunos ejemplos de amenazas a sistemas SCADA.

A continuación, se muestran algunos trabajos que utilizan las vulnerabilidades de los sistemas SCADA para hacer pruebas de penetración aplicando técnicas de los

sistemas de información convencionales. En (NI, 2015a) hacen ataques a redes de comunicación SCADA con el protocolo Modbus incluyendo ataques de inyección de paquetes y DoS (acrónimo en inglés Denial of Service). En (NI, 2015b), (NIST, 2012) hacen ataques MITM (acrónimo en inglés, Man In The Middle) por ARP (acrónimo en inglés, Address Resolution Protocol) spoofing en sistemas SCADA. En (Ralston, et al., 2007) hacen un ataque de inyección de paquetes en sistemas SCADA con el protocolo de comunicación IEC 60870-5-104 usando Ettercap. En (Seal, 2013) hacen un ataque DoS al protocolo Modbus utilizando TCP Modbus Hacker y presentan algunas metodologías para la detección del ataque.

En este trabajo se evalúa la seguridad de un inversor inteligente bajo los mismos ataques de los sistemas de información convencionales con la finalidad de evaluar vulnerabilidades y herramientas.

2. Métodos

El uso de dispositivos IoT (acrónimo en inglés, Internet of Things) es bastante amplio en diferentes áreas desde domótica hasta industrial. Su impacto es tan grande que inclusive se prevé que para el año 2020 se encuentren conectados entre 20 y 30 millones de dispositivos IoT. Su principal función es interactuar con información sobre el usuario con la finalidad de proveer estadísticas para la toma de decisiones y también permiten ejecutar acciones de forma remota. Sin embargo, la implementación de dispositivos IoT no se ha hecho de forma responsable ya que muchos de los dispositivos utilizados cuentan con severos problemas de seguridad a pesar de que en algunos casos procesan información sensible de los usuarios. Un ejemplo de esto es el estudio realizado por la empresa HP (HP, 2015), en donde evaluaron la seguridad en dispositivos IoT de diferentes fabricantes y mostraron que el 70 % de los dispositivos IoT más usados tienen serias vulnerabilidades.

Los dispositivos IoT pueden ser el medio de acceso a la red local del usuario, de esta manera se compromete la seguridad de la red local del usuario y de los dispositivos conectados a ésta.

Con el fin de evaluar la seguridad de un sistema DER se plantea el escenario en el cual un usuario doméstico cuenta con un inversor fotovoltaico conectado a su red local, así como otros dispositivos como laptops, celulares y dispositivos IoT. La plataforma de pruebas incluye una tarjeta sbRIO-9606 de National Instruments simulando un inversor inteligente, dicha tarjeta es un sistema embebido diseñado específicamente para el control de inversores electrónicos de potencia, un servidor SCADA en LabVIEW, un router y tarjetas Odroid XU4 simulando dispositivos IoT comprometidos por un atacante. En la figura 1 se muestra un diagrama de la plataforma de pruebas de penetración.

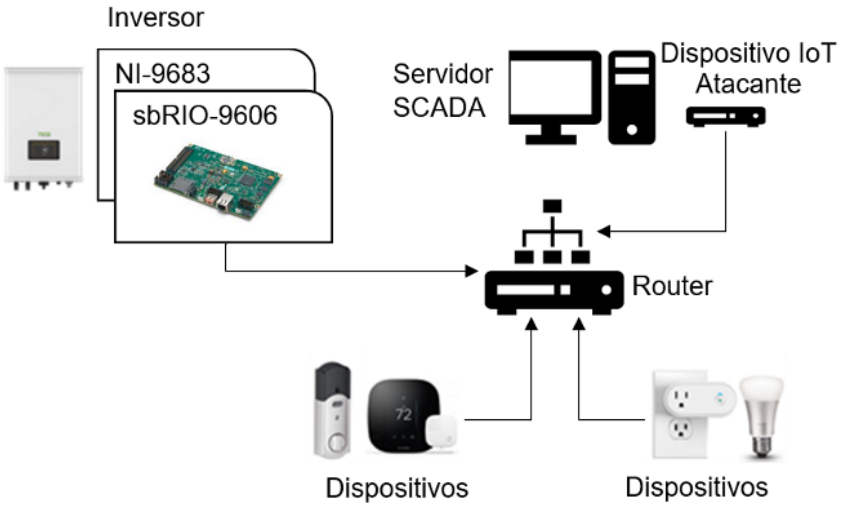

Figura 1 Plataforma de pruebas de penetración.

Implementación del protocolo de comunicación Modbus

Con el fin de evaluar el protocolo de comunicación Modbus, uno de los protocolos SCADA más utilizados y también para aplicaciones DER, se implementó en LabView utilizando una librería de National Instruments (NI, 2015a).

Modbus (Modbus-IDA, 2012) es un protocolo de comunicación abierto y público, podemos encontrar tres tipos de Modbus: ASCII y RTU que utilizan un medio serial como RS485 y Modbus TCP/IP que utiliza Ethernet. Modbus TCP utiliza una arquitectura maestro-esclavo en donde un maestro puede peticiones a un esclavo mediante códigos específicos que indican la acción requerida. El protocolo

Modbus/TCP es un protocolo en la capa de aplicación ISO/OSI, comúnmente utiliza el puerto TCP 502, como datos se utilizan bobinas que son Booleanos con una longitud de 1 bit y registros que son enteros sin signo con una longitud de 16 bits.

En cuanto a seguridad, el protocolo Modbus TCP no cuenta con autenticación ni encriptación lo que lo hace vulnerable a diferentes ataques. Algunos ejemplos de ataques son:

- Mensajes de difusión, donde un mensaje es enviado a todos los esclavos.
- Inyección de comandos.
- Escaneo de la red, donde el atacante analiza los paquetes entre esclavo y maestro con el fin de detectar direcciones y comandos.
- Repetición de mensajes, donde un atacante puede guardar el tráfico entre maestro y esclavo para posteriormente replicar los paquetes guardados.

Las funciones de comunicación Modbus que fueron implementadas al inversor inteligente son: envío de información sobre el voltaje generado y recepción de comandos de conexión/desconexión y apagado.

Escaneo de red

El primer paso para realizar en una prueba de penetración es el escaneo de la red. En esta etapa se hace un escaneo activo y/o pasivo de la víctima para identificar posibles superficies de ataque, vulnerabilidades, servicios, protocolos, direcciones y puertos que puedan ser utilizados por algún atacante con el fin de comprometer los sistemas.

En los sistemas SCADA se puede obtener información crítica sobre los procesos ya que, como se mencionó anteriormente, muchos de los protocolos de comunicación utilizados en SCADA no cuentan con encriptación y la información es transmitida en texto plano. La información que se puede obtener sobre la red es direcciones de origen y destino, puertos utilizados y comandos de control y configuración, pero también se puede obtener información sobre el dispositivo como fabricante, número de modelo, comandos permitidos y memorias.

Se deben tener en cuenta algunas consideraciones al usar herramientas de escaneo en sistemas SCADA por que pueden ocasionar un mal funcionamiento del sistema. Se recomienda utilizar una velocidad de escaneo baja, utilizar un escaneo TCP en lugar de un escaneo SYN y no utilizar el escaneo UDP ni funciones de identificación (en inglés, fingerprinting).

Algunas herramientas para realizar escaneos de red son: Wireshark, Tcpdump, Net2pcap, Tcptrace, Tcptrack, Nstreams, Argus, Karpski, Ipgrab, Nast, Aldebaran, Dsniff, Iptraf, Nmap, Amap, Hping3, Unicornscan, Paketto, Firewalk, Plcscan, Modscan, Nmap, Metasploit, entre otras.

MITM

En un ataque MITM un atacante se interpone entre dos dispositivos haciéndolos creer que se están comunicando directamente entre ellos (Francia, Thornton, y Brookshire, 2012; Skopik y Smith, 2015). Los métodos para llevar a cabo un ataque MITM son: ARP spoofing, DNS spoofing y ICMP spoofing.

El protocolo ARP tiene la función de asociar u obtener la dirección MAC de un dispositivo a partir de su dirección IP con el fin de poder enviar paquetes a la dirección correcta, también es utilizado para descubrir a otros dispositivos conectados en la misma LAN (acrónimo en inglés, Local Area Network).

Un conmutador de red asocia un puerto físico con una dirección MAC por lo que solo envía paquetes al puerto físico al cual está asociado la dirección MAC de destino. El conmutador de red guarda las relaciones de puerto-MAC en una tabla llamada CAM (acrónimo en inglés, Content Addressable Memory) también conocida como SAT (acrónimo en inglés, Source Address Table).

Para hacer un ataque de ARP spoofing (figura 2), se envían paquetes ARP para modificar las tablas ARP de los dispositivos de la red con el fin de asociar la dirección MAC del atacante con las direcciones IP de las víctimas. De esta manera, los paquetes que son enviados entre dos máquinas tienen la dirección IP de destino correcta pero la dirección MAC del atacante que, al ser recibidos por el conmutador de red, son enviados al puerto físico al que está conectado el atacante.

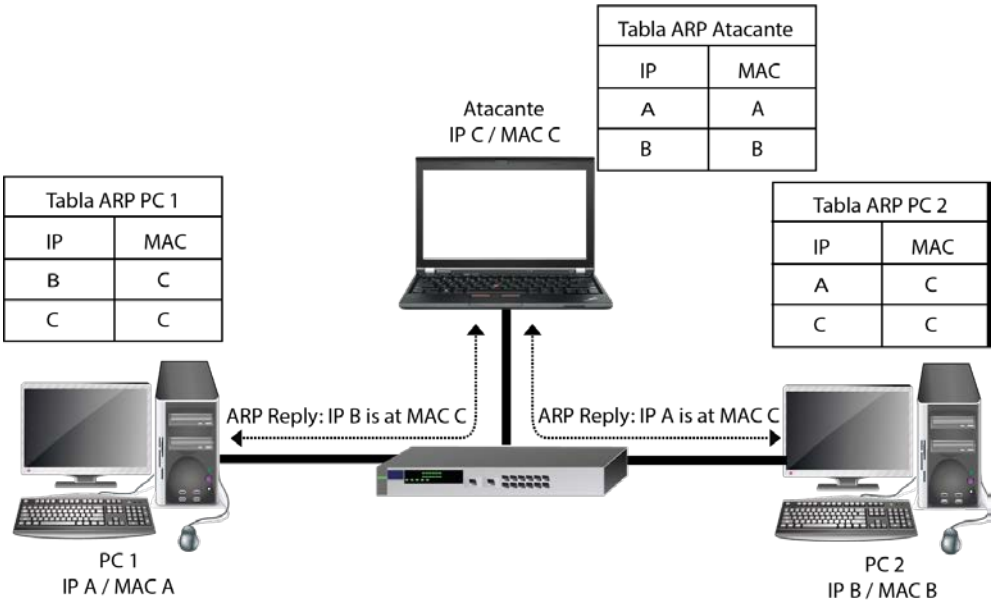

Figura 2 Ataque MITM.

Cuando los dispositivos se encuentran en un ataque MITM todos los paquetes que envían y reciben llegan primero al atacante y después son reenviados al dispositivo de destino. El atacante puede realizar diferentes ataques como:

- Sniffing, obtener todo el tráfico y buscar información específica como emails, contraseñas, cookies, etc.
- Proxy, modificar información de un protocolo, terminar conexiones, etc.
- Inyección de paquetes.
- Capturar y suprimir paquetes.
- Replicar y modificar paquetes capturados.

En un ataque de replicación de paquetes el atacante captura los paquetes que son enviados en la red para posteriormente replicarlos, algunos ejemplos de paquetes que son enviados son comandos de control y mediciones. Usar este ataque en sistemas SCADA sin modificar los paquetes significa que los dispositivos recibirían un duplicado de las mediciones, comandos de control, entre otros mensajes lo cual podría ocasionar un mal funcionamiento de los dispositivos. En un ataque de inyección de paquetes el atacante diseña o modifica paquetes guardados que serán enviados como comandos de control o mediciones con el fin de causar una

acción en los dispositivos. Los paquetes que son inyectados pueden ser capturados y modificados durante la transmisión, capturados y modificados después de la transmisión o diseñados completamente.

En sistemas SCADA, el ataque MITM es considerado un problema serio, debido a que el atacante podría enviar un comando de control causando así una manipulación maliciosa que podría apagar una planta eléctrica o dañar equipos.

Algunas herramientas para ataques MITM son: HexInject, Scapy, Mallory, Netsed, Ettercap, Bettercap, Mitmproxy, Dnsspoof, Arpspoof. Algunas herramientas para inyectar paquetes en sistemas SCADA son: Mbtget, Modbus Traffic Generator, Pymodbus, Modbus-cli, Metasploit, Colasoft Packet Builder, Tcpreplay.

3. Resultados

A continuación, se detallan las pruebas de penetración realizadas al inversor inteligente. En el escenario se hace la suposición de que el atacante tiene acceso a la red local o el ataque se origina dentro de la red local y se infectó un nodo IoT que tiene el sistema operativo Linux y permite ejecutar comandos de forma remota por ejemplo mediante SSH (acrónimo en inglés, Secure Shell). La intrusión a la red local se puede obtener a pesar de usar contraseñas WPA (Bruce, 2017; Ward, 2016). Los comandos utilizados se encuentran escritos como scripts y son ejecutados en el nodo IoT atacante con el sistema operativo Kali Linux. Se utilizó la distribución Kali Linux debido a que cuenta con diferentes herramientas para pruebas de penetración. Sin embargo, los programas utilizados pueden ser instalados, inclusive de manera remota, en diferentes distribuciones Linux que es el sistema operativo más utilizado para dispositivos IoT con una porcentaje arriba del 70 % (ARROW, 2016).

A continuación, se detallan los pasos utilizados para hacer un escaneo de la red:

- Identificación de dispositivos en la red local utilizando nmap.
- Identificación de puertos abiertos en el inversor inteligente utilizando nmap.
- Identificación de direcciones, puertos, comandos y memoria de Modbus utilizando Wireshark.

El primer paso es identificar los dispositivos que se encuentran en la red local y obtener direcciones IP y MAC, para esto utilizamos el comando de la ecuación 1.

$$nmap\ -\ sP\ 192.168.2.0/24 \quad (1)$$

Una vez obtenidas las direcciones IP, hacemos un escaneo SYN para identificar puertos abiertos y servicios del inversor inteligente dando la dirección IP de la ecuación 2.

$$nmap\ -\ sT\ \ 192.168.2.100 \quad (2)$$

El escaneo con nmap nos permitió identificar las direcciones IP y MAC de los dispositivos que se encuentran activos en la red local. También logramos identificar los servicios y puertos abiertos del inversor inteligente que podrían ser utilizados para hacer un ataque.

Ahora hacemos uso de la herramienta Wireshark para identificar las conexiones entre el maestro y esclavo, así como comandos de control y mapas de memoria.

Primero, se requirió de poder reenviar los paquetes recibidos a la máquina de destino utilizando el comando de la ecuación 3.

$$echo\ 1\ >/proc/sys/net/ipv4/ip\_forward \quad (3)$$

Para hacer el ataque ARP spoofing, utilizamos el programa arpspoof del paquete dsniff (ecuaciones 4 y 5).

$$arpspoof\ -i\ eth0\ -t\ 192.168.2.100\ 192.168.1.102 \quad (4)$$

$$arpspoof\ -i\ eth0\ -t\ 192.168.2.102\ 192.168.1.100 \quad (5)$$

Posteriormente, se utilizó el programa Wireshark para analizar el tráfico Modbus, figura 3.

El escaneo del protocolo Modbus nos permitió identificar lo siguiente:

- Dirección IP del maestro 192.168.2.102 con el puerto TCP 56652.
- Dirección IP del esclavo 192.168.2.100 con el puerto TCP 502.
- El maestro lee dos registros del esclavo iniciando con el registro en la dirección 0 para leer información sobre la operación del inversor.

- El esclavo lee dos bobinas a partir de la dirección 32 para leer los comandos de control del servidor.

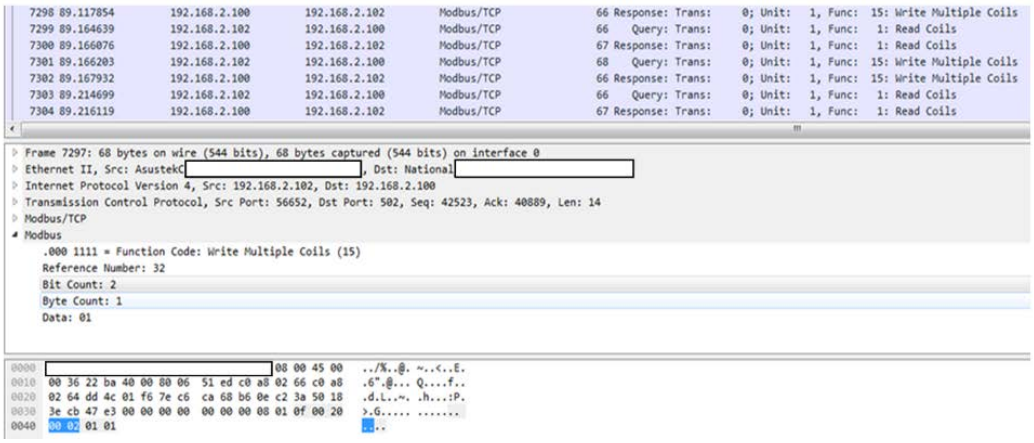

Figura 3 Escaneo en Wireshark del protocolo Modbus.

La información obtenida resulta bastante útil para ejecutar un ataque porque ahora se conocen las direcciones, puertos utilizados, comandos y mapas de memoria del maestro y esclavo. En este caso el nodo IoT contaba con un monitor, sin embargo, el uso de Wireshark en un nodo IoT comprometido no es posible debido a que Wireshark funciona con una interfaz de usuario y solamente se tiene acceso a una consola, considerando que solo se tiene acceso remoto al nodo IoT y/o el nodo IoT no cuenta con un monitor. Se puede utilizar tcpdump para observar mediante una consola los paquetes que están siendo transmitidos o guardar los paquetes en un archivo pcap para su posterior análisis en otro dispositivo utilizando Wireshark.
Los ataques MITM realizados fueron:

- Obtención de la contraseña del servicio ftp utilizando ettercap.
- Inyección de comandos Modbus utilizando mbtget.
- Replicación de paquetes utilizando tcpdump y tcpreplay.

A continuación, hacemos un ataque MITM mediante ARP spoofing utilizando ettercap, para esto utilizamos el comando de la ecuación 6.

$$ettercap - T - q - ieth0 \\ - Marp/192.168.2.100/192.168.2.102/\ /192.168.2.102/192.168.2.100 \\ / \qquad (6)$$

Cuando hacemos un ataque MITM con ettercap, el programa monitorea el tráfico en busca de información que puede ser útil como contraseñas transmitidas en texto plano, en este caso se logró obtener la contraseña del servicio ftp del dispositivo 192.168.2.100. Las credenciales de acceso fueron obtenidas cuando un usuario intentó acceder al servicio ftp del inversor inteligente. Si las credenciales de acceso obtenidas son correctas depende de si las que ingresó el usuario son correctas. Las contraseñas no pueden ser obtenidas con un ataque MITM cuando los dispositivos utilizan servicios encriptados como SSH, SFTP, o HTTPS.

El programa mbtget es un script en Perl que permite hacer operaciones en Modbus/TCP por lo que nos permite hacer un ataque de inyección de paquetes en una red SCADA que utiliza el protocolo Modbus. El comando de la ecuación 7 es utilizado para modificar lecturas sobre el voltaje generado por el inversor que son enviadas del esclavo al maestro (figura 4).

$$mbtget\ -\ w6\ 0\ -\ a\ 0\ 192.168.2.100 \tag{7}$$

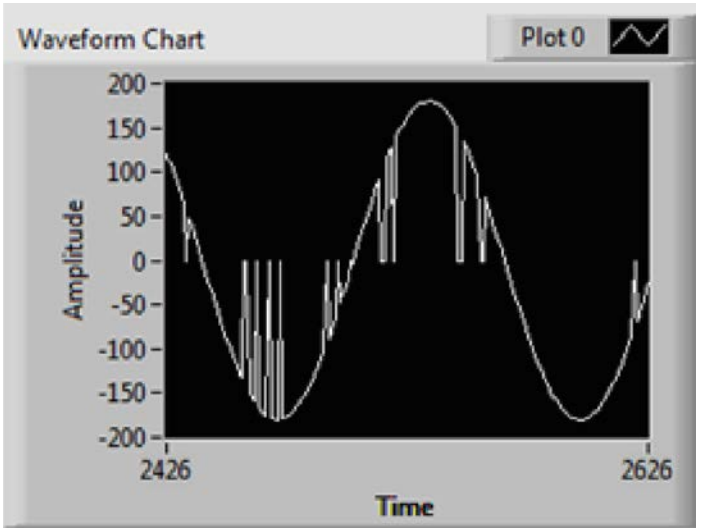

Figura 4 Inyección de lecturas en Modbus.

El comando de la ecuación 8 es utilizado para enviar el comando de control de apagado al esclavo.

$$mbtget - w5\ 1\ -\ a\ 31\ 192.168.2.102 \tag{8}$$

Con este comando se logró mandar un comando de control para indicar al inversor que se apagara Estos ataques también pueden ser realizados utilizando el módulo auxiliary/scanner/scada/modbusclient de Metasploit.

El programa tcpreplay nos permite hacer un ataque de replicación de paquetes, a continuación, se muestran los comandos para capturar los paquetes de la comunicación Modbus utilizando tcpdump y el uso de tcpreplay para replicar los paquetes, ecuaciones 9 y 10.

$$tcpdump - ieth0 - w/root/Desktop/Modbus.pcap\text{dstport502orsrcport502} \quad (9)$$

$$tcpreplay \ - i \ eth0 \ /root/Desktop/Modbus.pcap \quad (10)$$

Los paquetes replicados por tcpreplay no son aceptados en la capa de aplicación, son rechazados por el stack TCP/IP, esto se debe a que tcpreplay no modifica los campos SYN y ACK de la conexión TCP. Debido a que los paquetes no son aceptados en la capa de aplicación, este ataque tiene un bajo impacto en el funcionamiento de los sistemas SCADA. Este ataque podría afectar a los sistemas SCADA al pasar por firewalls como tráfico válido y por la cantidad de tráfico que es generada puede afectar la transmisión de paquetes. En este caso, la replicación de los paquetes afectó la transmisión de los paquetes al consumir los recursos de red del maestro y esclavo lo que ocasionó que se agotara el tiempo de espera en la comunicación TCP.

El programa bettercap permite hacer ataques MITM y tiene una opción para posteriormente hacer un ataque DoS (acrónimo en inglés, Denial of Service) al no redirigir los paquetes a la máquina de destino, para hacer este ataque utilizamos el comando de la ecuación 11.

$$bettercap \ - \ T \ 192.168.2.100{,}192.168.2.102 \ \ - -kill \quad (11)$$

Con el comando anterior se logró cortar completamente la comunicación entre esclavo y maestro. Funciona primero haciendo un ataque MITM mediante ARP spoofing y posteriormente no reenvía los paquetes al dispositivo de destino por lo que los dispositivos dejan de comunicarse.

4. Discusión

Los protocolos de comunicación SCADA fueron diseñados para ser utilizados en redes seguras y al no estar expuestos al exterior se tenía un falso sentimiento de seguridad de la red. Sin embargo, ya no se puede confiar en la supuesta seguridad inherente de los sistemas SCADA pues actualmente operan conectados al exterior y utilizan protocolos abiertos y estandarizados.

Se han publicado algunos estándares para la implementación de seguridad en los protocolos de comunicación SCADA. Algunos ejemplos son el estándar IEC 62351 para el intercambio de información en sistemas de potencia, el estándar IEEE 1686 para medidas de seguridad en Dispositivos Electrónicos Inteligentes y también se pueden encontrar recomendaciones de fabricantes como National Instruments (NI, 2015b). El estándar IEC 62351 incluye la autenticación de la transferencia de datos a través de firmas digitales, lo que garantiza únicamente el acceso autenticado. Sin embargo, estas mejoras de seguridad comúnmente no son implementadas o utilizadas en los dispositivos SCADA por cuestiones monetarias o facilidad de uso.

En este trabajo se presentaron las mayores superficies de ataque, vulnerabilidades y amenazas de dispositivos presentes en sistemas SCADA. Usando una plataforma de pruebas de penetración en un ambiente controlado se demostró que los sistemas y protocolos de comunicación SCADA son vulnerables a los mismos ataques que los sistemas de información convencionales. Además, se utilizaron herramientas disponibles para diferentes distribuciones Linux y de fácil acceso.

5. Bibliografía y Referencias

[1] Bruce, J. (2017). How easy is it to crack a Wi-Fi network?: https://www.makeuseof.com/tag/how-easy-is-it-to-crack-a-wifi-network-make useof-explains/.

[2] Cagalaban, G. (2009). SCADA Network Insecurity: Securing Critical Infrastructures through SCADA Security Exploitation. Journal of Security Engineering, 6 (6), 473–482.

[3] ARROW (2016). IoT Operating Systems: https://www.arrow.com/en /research-and-events/articles/iot-operating-systems

[4] CEC. (2014). Recommendations for updating the technical requirements for inverters in distributed energy resources.

[5] EPRI. (2016). Common Functions for Smart Inverters: 4th Edition.

[6] Federal Office for Information Security. (2016). Industrial Control System Security - Top 10 Threats and Countermeasures. BSI Publications on Cyber-Security, 1–20.

[7] Francia, G., Thornton, D., & Brookshire, T. (2012). Cyberattacks on SCADA Systems.

[8] GAO. (2004). Critical Infrastructure Protection Challenges and Efforts to Secure Control Systems.

[9] GE. (2012). Top 10 Cyber Vulnerabilities for Control Systems.

[10] HP. (2015). HP Study Reveals 70 Percent of Internet of Things Devices Vulnerable to Attack: http://www8.hp.com/in/en/hp-news/press-release.html?id=1744676#.WgTolltSxhE.

[11] ICS-CERT. (2018). US-CERT SCADA Vulnerabilities: https://ics-cert.us-cert.gov/content/overview-cyber-vulnerabilities.

[12] Krutz, R. L. (2006). Securing SCADA Systems. Wiley.

[13] Makhija, J., & Subramanyan, L. R. (2003). Comparison of protocols used in remote monitoring: DNP 3.0, IEC 870-5-101 & Modbus.

[14] Modbus-IDA. (2012). Modbus Application Protocol Specification: http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf.

[15] NERC. (2007). Top 10 Vulnerabilities of Control Systems and Their Associated Mitigations.

[16] NI. (2015a). LabVIEW MODBUS Library: http://www.ni.com/example /29756/en/.

[17] NI. (2015b). Overview of Best Practices for Security on RIO Systems: http://www.ni.com/white-paper/13069/en/.

[18] Ralston, P. A. S., Graham, J. H., & Hieb, J. L. (2007). Cyber security risk assessment for SCADA and DCS networks. ISA Trans.

[19] NIST. (2012). Advanced power system management functions and information exchanges for inverter-based DER devices, modelled in IEC 61850-90-7.

[20] Seal, B. K. (2013). Smart Inverters. EPRI: http://smartgrid.epri.com /doc/SmartInverters-SmartGridInformational Webcast.pdf.

[21] Skopik, F., & Smith, P. (2015). Smart Grid Security - Innovative Solutions for a Modernized Grid. Elsevier Science Publishing.

[22] Ward, M. (2016). How easy is it to hack a home network? BBC: www.bbc.com/news/technology-35629890

[23] Wiles, J. (2008). Techno Security's Guide to Securing SCADA: A Comprehensive Handbook on Protecting the Critical Infrastructure. Elsevier.

[24] Zhu, B., Joseph, A., & Sastry, S. (2011). A taxonomy of Cyber Attacks on SCADA Systems. IEEE International Conferences on Internet of Things and Cyber, Physical and Social Computing.