© de los textos: sus autores.
© de la edición: Fundación Tecnalia Research and Innovation.

I.S.B.N : 978-84-88734-13-6

Esta obra se encuentra bajo una licencia Creative Commons CC BY 4.0.

Cualquier forma de reproducción, distribución o trasformación de esta obra no incluida en la licencia Creative Commons CC BY 4.0 solo puede ser realizada con la autorización expresa de los titulares, salvo excepción prevista por la ley. Puede Vd. acceder al texto completo de la licencia en este enlace: https://creativecommons.org/licenses/by/4.0/deed.es

Aleatorización de direcciones IP para mitigar ataques de reconocimiento de forma proactiva en sistemas de control industrial

Xabier Etxezarreta, Iñaki Garitano, Mikel Iturbe y Urko Zurutuza

Departamento de Electrónica e Informática
Escuela Politécnica Superior
Mondragon Unibertsitatea
Goiru 2, E-20500 Arrasate-Mondragón
Email: {xetxezarreta,igaritano,miturbe,uzurutuza}@mondragon.edu

***Resumen*—Los sistemas de control industrial se utilizan en una gran variedad de procesos físicos, incluidas las infraestructuras críticas, convirtiéndose en el principal objetivo de múltiples ataques de seguridad. Un ataque malintencionado y exitoso contra estas infraestructuras podría causar graves consecuencias económicas y ambientales, incluyendo la pérdida de vidas humanas. Las redes estáticas, que caracterizan a los sistemas de control industrial, suponen una ventaja para los atacantes, permitiéndola explorar en busca de dispositivos o servicios vulnerables antes de realizar el ataque. Identificar dispositivos activos suele ser el primer paso para muchos ataques. Este trabajo presenta un sistema de defensa ante reconocimientos de red que se basa en la aleatorización temporal de las direcciones de red. La distorsión de la información obtenida inhabilita el conocimiento adquirido por parte de los atacantes dificultando así cualquier ataque que se apoya en el direccionamiento de la red. La aleatorización temporal de las direcciones de red se realiza de forma adaptativa minimizando así la sobrecarga introducida en la red y evitando cualquier error y latencia en las comunicaciones. La implementación así como las pruebas se han realizado en un laboratorio con equipamiento industrial real, demostrando así la efectividad de la solución presentada.**

***Index Terms*—Sistemas de control industrial, Moving Target Defense, Redes definidas por software, Seguridad en redes industriales**

Tipo de contribución: *Investigación original (límite 8 páginas)*

I. Introducción

Sistema de control industrial (ICS por sus siglas en inglés) es un término general que cubre varios elementos especializados utilizados para la monitorización y control de procesos industriales [1]. Están compuestos por diversos elementos como sensores, actuadores, controladores lógicos programables (PLC) o sistemas de control supervisor y adquisición de datos (SCADA). Se pueden encontrar en todo tipo de industrias, incluidas en las infraestructuras críticas, convirtiéndose en elementos imprescindibles para el bienestar y desarrollo económico de la sociedad. Ejemplos de infraestructura crítica incluyen centrales nucleares, sistemas de transporte, redes eléctricas, presas hidroeléctricas y plantas de fabricación críticas.

Tradicionalmente, los ICS han estado desplegados en entornos aislados, utilizando protocolos de comunicación y hardware propietarios. El aislamiento y la oscuridad han sido los pilares en los que se ha basado la seguridad en ICS, pero la integración de tecnología de la información (IT) ha expuesto los originalmente aislados ICS a las redes corporativas, incluyendo Internet. Este cambio de tendencia ha hecho que las técnicas tradicionales de aislamiento y seguridad por oscuridad dejen de ser efectivas en estos entornos. La naturaleza de los ICS hace difícil que las soluciones de seguridad IT cumplan con los requisitos de estos sistemas. Esto hace que sea necesario desarrollar soluciones de seguridad específicas para estos entornos. De acuerdo con la publicación NIST SP 800-82 Rev 2 [1], los ICS se diferencian de los sistemas IT en los siguientes aspectos:

- Los ICS se utilizan para controlar y monitorizar procesos y dispositivos físicos.
- Una interrupción no es aceptable. La disponibilidad es prioritaria frente a la integridad y la confidencialidad.
- El tiempo es crítico en los ICS, la latencia en las comunicaciones tiene que ser mínima.
- El periodo de sustitución/actualización de los dispositivos que componen los ICS es muy largo en comparación con los sistemas IT.
- La aplicación de parches de seguridad muchas veces es pospuesta debido a las necesidades de disponibilidad y fiabilidad.
- En muchos casos los ICS no tienen capacidad de integrar mecanismos de seguridad.

Si lo comparamos con las redes IT, las topologías de red industriales son generalmente estáticas y el tráfico de control es por naturaleza repetitiva y predecible, debido a que la mayor parte del tráfico es creado por procesos automatizados [2]. Esta característica estática de las redes industriales supone un escenario ventajoso para el atacante, permitiendo explorar vulnerabilidades antes de realizar el ataque. Debido a este problema, una tendencia de soluciones de seguridad proactivas empezaron a desarrollarse en respuesta a estos sistemas estáticos bajo el nombre de Moving Target Defense (MTD). MTD se puede definir como un sistema en constante cambio que desplaza o reduce la superficie de ataque, dificultando que un atacante pueda explorar y explotar vulnerabilidades fácilmente.

Las redes definidas por software (SDN) se han convertido en una tecnología prometedora para ICS, tanto para desarrollar

técnicas de MTD [3] como para el desarrollo de herramientas de detección y respuesta a intrusiones en general [4]. En el RFC 7140 [5] se define SDN como un conjunto de técnicas utilizadas para facilitar el diseño, la entrega y la operación de servicios de red de una forma determinista, dinámica y escalable. Para esto, el plano de control es separado y centralizado, mientras que el plano de datos se mantiene en los dispositivos de red, centrando su funcionalidad al reenvío de paquetes. En la figura 1 se representan los principales planos de la arquitectura SDN. En primer lugar, el plano de datos es donde se aplican las decisiones de reenvío y se procesa el tráfico. En segundo lugar se encuentra el plano de control, utilizado para proporcionar lógica al plano de datos mediante el uso de controladores SDN. La comunicación entre el plano de datos y de control se realiza mediante la *interfaz southbound* con protocolos específicos en los que OpenFlow es el principal exponente, pero existen otros (ej. NETCONF). Por último, el plano de aplicación es utilizado para desarrollar aplicaciones de negocio que interaccionan con la red mediante la *interfaz northbound*.

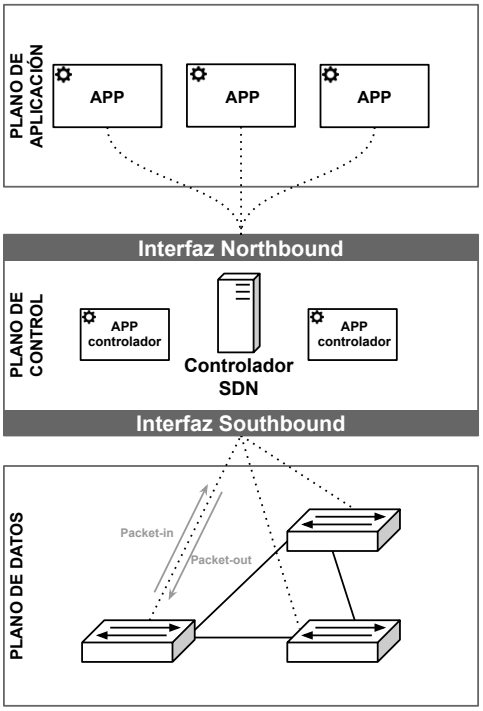


Figura 1: Arquitectura SDN.

Las redes tradicionales no están optimizadas para satisfacer las necesidades presentes y futuras de los ICS, que demandan más flexibilidad sin comprometer la calidad el servicio [6]. Desde el punto de vista del desarrollo de soluciones de detección y respuesta de intrusiones, las SDN ofrecen ventajas respecto a las redes tradicionales en los siguientes aspectos: (1) ofrecen una visión global sobre la red, (2) aumentan la programabilidad de la red integrando aplicaciones desarrolladas por el usuario y (3) permiten modificar el comportamiento de los flujos de red de forma dinámica.

En este artículo, se combinan los conceptos de SDN y MTD para desarrollar un mecanismo de defensa proactiva que sirve para mitigar ataques de reconocimiento en redes de control industrial. Las principales contribuciones de este trabajo se pueden resumir en los siguientes puntos:

- Desarrollo de un sistema MTD que aleatoriza las direcciones IP del tráfico de red en tiempo real, distorsionando la información obtenida por un atacante durante la fase de reconocimiento e impidiendo el acceso directo a los dispositivos mediante el uso de IPs reales. La aleatorización se realiza en los switches y es completamente transparente para los dispositivos finales de la red.
- La inicialización de las reglas de flujo que aleatorizan las direcciones IP y reenvían el tráfico a su destino se realiza mediante un *allowlist* predefinido. En este *allowlist* se define qué dispositivos están autorizados para comunicarse entre ellos. En base a esta información, se instalan las reglas de flujo pertinentes en los switches.
- La aleatorización de las direcciones IP se realiza de forma adaptativa, minimizando la latencia introducida y cumpliendo con los requisitos de tiempo real de las redes de control industrial. Para esto se hace uso de reglas de flujo de respaldo y el campo *priority* del protocolo OpenFlow.
- La solución se ha implementado y probado en un entorno con equipamiento industrial real, demostrando así la efectividad de la solución presentada.

II. Trabajos relacionados

Esta sección presenta una breve discusión acerca de los trabajos relacionados. Por un lado, se proporciona información acerca de las diferentes técnicas MTD de la literatura. Por otro lado, se discute la aplicabilidad de MTD en sistemas de control industrial.

II-A. Técnicas de Moving Target Defense

Las técnicas MTD tienen como objetivo cambiar la naturaleza estática de las redes modificando la superficie de ataque de forma dinámica. Estas técnicas pueden ser clasificadas en los siguientes cuatro grupos [7]:

MTD basado en *Shuffling*: Son técnicas que aleatorizan la configuración de la red para hacerla menos predecible y disminuir la superficie de ataque. En este grupo podemos encontrar técnicas que aleatorizan las direcciones IP [8], [9], [10], puertos [11], rutas por donde fluye el tráfico de red [12] o la cabecera de los paquetes [13]. También existen soluciones que combinan varias técnicas MTD a la vez basadas en *shuffling* [14].

MTD basado en Diversidad: Consiste en proporcionar servicios equivalentes pero con diferentes implementaciones. La diversidad de código tiene como objetivo dividir un programa en componentes que pueden ser implementados en diferentes entornos de ejecución [15]. Las técnicas de diversidad de software despliegan variantes equivalentes de servidores web, aplicaciones o servidores virtuales para mejorar la resiliencia de la red [16]. Por último, las técnicas de diversidad en lenguajes de programación permiten mitigar ataques de código o inyección SQL [17].

MTD basado en Redundancia: Consiste en desplegar réplicas que ofrecen la misma funcionalidad. Podemos encontrar técnicas que proporcionan redundancia en sesiones de red [18] o que despliegan réplicas de servidores con la misma funcionalidad [19].

MTD Híbrido: Estas técnicas combinan MTD basado en *Shuffling*, Diversidad y Redundancia [20], [21].

II-B. MTD en sistemas de control industrial

Las técnicas MTD han sido introducidas en sistemas de control industrial para revertir la naturaleza estática y predecible de estos sistemas. La investigación se ha centrado en desarrollar y adaptar técnicas MTD basadas en *shuffling*, especialmente técnicas que aleatorizan las direcciones IP y las rutas por donde fluye tráfico. Por un lado, las técnicas de aleatorización de direcciones IP para sistemas de control industrial de la literatura aprovechan la utilidad IPTables para realizar operaciones de traducción de direcciones. En los artículos [22] y [23] se puede observar que conforme el intervalo de aleatorización disminuye, el *Round Trip Time (RTT)* aumenta considerablemente, convirtiéndose en un problema para los sistemas que requieren unas comunicaciones con latencias bajas. Por otro lado, en relación con la aleatorización de rutas, los autores en [24] proponen una técnica para evitar que todo el tráfico vaya por el mismo camino, dispersando el tráfico por múltiples rutas para defenderse de intercepciones de tráfico no autorizadas. Como la transición a rutas alternativas no se realiza de forma adaptativa, los autores en [25] proponen utilizar el campo *hard-timeout* del protocolo OpenFlow para definir varias reglas de flujo a la vez y minimizar la latencia introducida en la red industrial basada en SDN.

A diferencia de las publicaciones existentes, este trabajo combina la tecnología SDN y la aleatorización de direcciones IP para desarrollar un mecanismo de defensa proactiva que puede ser implementado en entornos sensibles a la latencia o retardo en las comunicaciones como los sistemas de control industrial. Para esto, el proceso de aleatorización se implementa en los dispositivos de reenvío y la transición a diferentes direcciones IP aleatorias se realiza de forma adaptativa utilizando reglas de respaldo y el campo *priority* del protocolo OpenFlow.

III. Framework

En esta sección se presenta el mecanismo de defensa proactiva que proporciona una aleatorización de direcciones IP para una red de control industrial. Primero se detalla la arquitectura y las partes en la que está compuesta. En segundo lugar, se detalla la inicialización del sistema mediante el uso de un *allowlist*. Por último se define como se implementa de forma adaptativa la transición a nuevas direcciones IP aleatorias.

III-A. Arquitectura

La arquitectura está desarrollada para ser integrada y utilizada en un entorno industrial basado en SDN. En la figura 2 se representa la visión general de la arquitectura de aleatorización de direcciones IP. Los principales componentes de esta arquitectura son los siguientes:

- *Dispositivos finales:* Componen una gran variedad de dispositivos utilizados en entornos industriales como servidores SCADA, PLCs o estaciones de trabajo.
- *Switches OpenFlow:* Estos dispositivos de reenvío tienen como función enrutar el tráfico a su destino, en base a reglas de flujo existentes en sus tablas de enrutamiento. En el caso de la presente solución, estos dispositivos se utilizan para procesar los paquetes de red y aleatorizar las direcciones IP. Para esto, se instalan reglas de flujo que aplican acciones en paquetes en base a coincidencias.
- *Controlador SDN:* Es el responsable de la comunicación entre el plano de datos y plano de aplicación. El controlador obtiene las peticiones del módulo MTD y las transmite a los switches utilizando el protocolo OpenFlow.
- *Módulo MTD:* Es un programa desplegado en el plano de aplicación. Tiene como función inicializar y actualizar las reglas de flujo en los switches OpenFlow que realizan las traducciones entre direcciones IP reales y aleatorias. Esto se realiza utilizando la *interfaz northbound* del controlador SDN, en este caso una API REST.

III-B. Aleatorización de direcciones IP

En la fase de inicialización, las reglas de flujo iniciales que se instalan en los switches están basadas en un *allowlist* definido por el usuario. Aprovechando la naturaleza estática de las redes industriales y lo repetitivas y predecibles que son las comunicaciones del tráfico de control, en este *allowlist* se recogen las comunicaciones entre los dispositivos de la red que están autorizadas. Si una comunicación está autorizada, los dispositivos podrán comunicarse entre ellos utilizando IPs reales. Por el contrario, en comunicaciones no autorizadas, un dispositivo solo podrá comunicarse con otro dispositivo utilizando la dirección IP aleatoria asignada en ese momento al dispositivo de destino.

El primer paso consiste en generar y asignar una dirección IP aleatoria a cada dispositivo presente en la red. Estas direcciones IP asignadas solo serán válidas durante un intervalo MTD y serán reemplazadas por otras IPs aleatorias en el siguiente intervalo MTD. El intervalo MTD es definido por el usuario y tiene que ser adaptado para cada caso de uso. Al generar las IPs aleatorias se comprueba que no se hayan generado dos iguales, de esta manera se evita que haya colisiones y posibles errores en el funcionamiento del sistema. Hay que destacar que la configuración de red de los dispositivos finales no se cambia, las traducciones de direcciones IP se realizan en los switches y el proceso es completamente transparente.

Con las direcciones IP aleatorias generadas para cada dispositivo, se instalan reglas de flujo en los switches OpenFlow para que solo los dispositivos autorizados puedan comunicarse mediante el uso de direcciones IP reales. Existen dos opciones para cambiar o aleatorizar las direcciones IP de los paquetes: (1) enviar todos los paquetes al controlador SDN para que sean procesados de forma centralizada o (2) procesar los paquetes directamente en los switches. Para minimizar el retardo introducido en una comunicación, las reglas de flujo procesan el paquete directamente en el switch, evitando que cada paquete sea enviado al controlador SDN para su procesamiento. La tabla I representa una tabla de flujos de un switch OpenFlow. Para la comunicación entre dos dispositivos autorizados son necesarias ocho entradas en las tablas de flujo, cuatro en el switch de origen y cuatro en el switch de destino. Los paquetes IP y ARP son procesados con reglas de flujos independientes, ya que como se especifica en la especificación del protocolo OpenFlow [26], es necesario

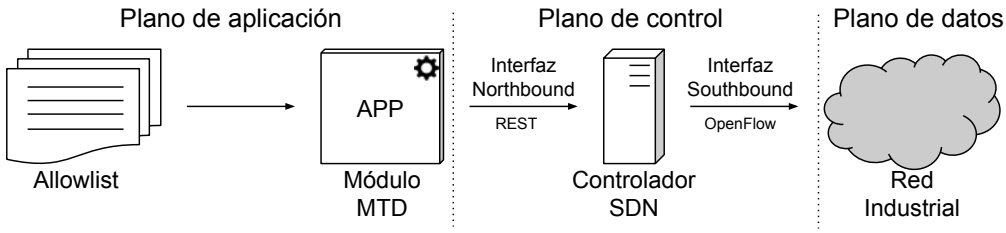


Figura 2: Visión general del sistema de aleatorización de direcciones IP.

definir el tipo de *frame ethernet* (*eth_type*) en la columna *match* para poder procesar paquetes con valores de la cabecera del protocolo IP. A cada tipo de paquete (IP o ARP) le pertenecen dos reglas de flujo en cada switch; uno para traducir las IPs reales a IPs aleatorias y otro para traducir las IPs aleatorias a IPs reales. Para todos los demás dispositivos no autorizados, solo son necesarios cuatro reglas de flujo que permiten la comunicación utilizando las direcciones IP aleatorias asignadas en ese momento.

Una vez que las reglas de flujo estén instaladas en los switches, el proceso que sigue cada paquete que va desde un origen a un destino está definido en el Algoritmo 1. Imaginemos una comunicación entre dos dispositivos finales h_1 y h_2. Cuando un paquete llega al switch OpenFlow de origen, es decir, al switch al que está conectado h_1, las direcciones IP reales (rIP) son cambiadas por IPs aleatorias (vIP) en caso de que la comunicación entre h_1 y h_2 esté autorizada. En caso contrario, solo se cambia la dirección IP de origen y se le da salida al paquete por un puerto del switch. Cuando el paquete llega al switch OpenFlow de destino, es decir, al switch al que está conectado h_2, las direcciones IP aleatorias vIP son traducidas a IPs reales rIP en caso de que la comunicación entre h_1 y h_2 esté autorizada. De lo contrario, si h_1 no tiene autorización para comunicarse con h_2 y la IP de destino utilizada por h_1 no coincide con la IP aleatoria asignada a h_2 en ese intervalo, el paquete es descartado evitando que llegue a su destino. Si la IP de destino utilizada por h_1 es la IP aleatoria asignada a h_2 en ese momento, el paquete es reenviado a su destino cambiando la IP aleatoria por la IP real del dispositivo de destino.

Algorithm 1 Proceso seguido por cada paquete para llegar a su destino.

for all packets pkt from h_1 to h_2 **do**
  if pkt is at src switch **then**
    set $pkt.src = vIP(h_1)$
    if h_1 is authorized to h_2 **then**
      set $pkt.dst = vIP(h_2)$
    end if
    $output \leftarrow out\_port$
  else if pkt is at dst switch **then**
    if h_1 is authorized to h_2 **then**
      set $pkt.src = rIP(h_1)$
      set $pkt.dst = rIP(h_2)$
      $output \leftarrow out\_port$
    else
      if $pkt.dst$ is $vIP(h_2)$ **then**
        set $pkt.dst = rIP(h_2)$
        $output \leftarrow out\_port$
      else
        $drop \leftarrow pkt$
      end if
    end if
  else
    $output \leftarrow out\_port$
  end if
end for

III-C. Transición adaptativa entre intervalos

Las direcciones IP aleatorias asignadas solo son válidas durante un intervalo de tiempo definido por el usuario. Al final de cada intervalo, las direcciones IP cambian por otras nuevas generadas aleatoriamente y son asignadas a cada dispositivo final, enrutando el tráfico utilizando las nuevas direcciones IP. Si las reglas de flujo que están siendo utilizadas son directamente borradas o actualizadas, las comunicaciones que aún estén utilizando las direcciones IPs aleatorias del intervalo anterior pueden ser enrutadas de forma incorrecta, generando una interrupción en la red. Para solventar este problema y evitar la introducción de retardos o interrupciones en la red, se ha diseñado un método de actualización de reglas de flujo que hace uso de reglas de respaldo y del campo *priority* del protocolo OpenFlow. De esta manera, conseguimos una transición adaptativa a unas direcciones IP aleatorias nuevas. La figura 3 muestra el proceso seguido al final de cada intervalo MTD y consta de cuatro fases.

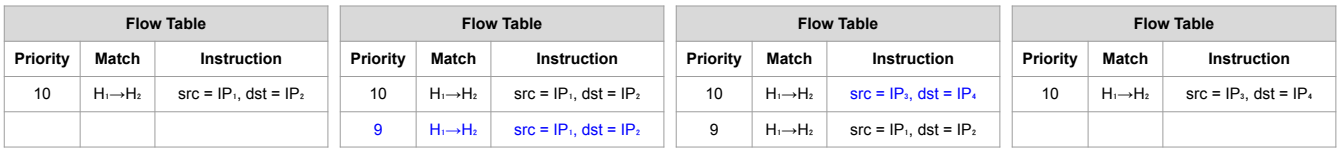
La primera fase, representada en la subfigura 3(a), hace referencia al estado de las reglas de flujo al final de cada intervalo MTD. En este estado, las reglas de flujo hacen traducciones entre IPs reales y aleatorias (y viceversa) asignadas en ese intervalo activo.

Cuando un intervalo MTD llega a su fin, se generan nuevas direcciones IP aleatorias que serán utilizadas en el siguiente intervalo. Si las reglas de flujo activas son actualizadas o eliminadas, el tráfico que aún esté utilizando las direcciones IP del intervalo anterior puede ser descartado o forzado a que pase por el controlador SDN. Esto se debe a que ese tráfico no dispone de ninguna regla de flujo para hacer *match*. En sistemas de control industrial donde el tiempo es crítico, interrumpir o introducir retardo en el tráfico no es aceptable. Para evitar estos problemas, al final de cada intervalo, se genera una regla de respaldo por cada regla de flujo activa en cada switch. Como se representa en la subfigura 3(b), la regla de respaldo es una copia de la regla de flujo activa pero con menor prioridad.

Con las reglas de respaldo, se procede a actualizar las reglas activas asignando las nuevas direcciones IP aleatorias

Tabla I: Reglas en una tabla de flujos de un switch OpenFlow.

Switch Flow Table		
Priority	**Match**	**Instruction**
10	src = $rIP(h_1)$, dst = $rIP(h_2)$, eth_type = IP	src = $vIP(h_1)$, dst = $vIP(h_2)$, ouput($port$)
10	src = $vIP(h_2)$, dst = $vIP(h_1)$, eth_type = IP	src = $rIP(h_2)$, dst = $rIP(h_1)$, ouput($port$)
10	src = $rIP(h_1)$, dst = $rIP(h_2)$, eth_type = ARP	src = $vIP(h_1)$, dst = $vIP(h_2)$, output($port$)
10	src = $vIP(h_2)$, dst = $vIP(h_1)$, eth_type = ARP	src = $rIP(h_2)$, dst = $rIP(h_1)$, output($port$)
5	src = $rIP(h_1)$, dst = $vIP(h_3)$, eth_type = IP	src = $vIP(h_1)$, output($port$)
5	src = $vIP(h_3)$, dst = $vIP(h_1)$, eth_type = IP	dst = $rIP(h_1)$, output($port$)
5	src = $rIP(h_1)$, dst = $vIP(h_3)$, eth_type = ARP	src = $vIP(h_1)$, output($port$)
5	src = $vIP(h_3)$, dst = $vIP(h_1)$, eth_type = ARP	dst = $rIP(h_1)$, output($port$)
0	any	drop



(a) Estado inicial.

(b) Se añade una regla de flujo respaldo con la misma instrucción, pero con menor prioridad.

(c) Se actualiza la regla con mayor prioridad asignando nuevas direcciones IP aleatorias.

(d) Se elimina la regla de flujo de respaldo.

Figura 3: Proceso de actualización de las reglas de flujo al final de un intervalo MTD.

generadas para el nuevo intervalo. Esta fase está representada en la subfigura 3(c). En este estado, el tráfico nuevo empezará a utilizar las direcciones IP aleatorias del nuevo intervalo, mientras que el tráfico generado en el intervalo anterior utilizará las reglas de flujo de respaldo. Con este método, se consigue una transición entre intervalos adaptativa, sin generar retardos ni pérdida de paquetes.

Por último, como se muestra en la subfigura 3(d), se eliminan las reglas de respaldo, volviendo al estado inicial. Este proceso se aplica de manera iterativa al final de cada intervalo, independientemente del tiempo de intervalo definido por el usuario.

IV. Resultados y discusión

Para probar la viabilidad del sistema de aleatorización de direcciones IP en un sistema de control industrial, se ha desplegado un entorno experimental industrial con equipamiento real: PLC AC800M y servidor SCADA, ambos de ABB. El banco de pruebas experimental está basado en una emulación de un sistema de entrada de un almacén. La entrada del almacén está formada por dos puertas correderas que se abren cuando un operario pasa por delante de un sensor de movimiento. Cuando el sensor de movimiento detecta el paso de un operario, las puertas se abren y se mantienen abiertas durante cinco segundos hasta que se vuelven a cerrar. Si el sensor detecta a un operario mientras la puerta está abierta, el tiempo de apertura de la puerta se prolonga cinco segundos más. Durante este proceso, se almacenan dos variables: (1) número de operarios que entran y (2) número de veces que se abre la puerta.

La figura 4 representa la topología del entorno de experimentación. En primer lugar, la emulación de la entrada del almacén se ejecuta en un PLC propietario de ABB AC800M ubicado en la red de control y está conectado directamente a un switch OpenFlow. En segundo lugar, en la red de supervisión, se ha desplegado un servidor SCADA ABB que está conectado a otro switch OpenFlow y solicita datos al PLC. La comunicación entre estos dispositivos ABB se realiza mediante el protocolo Manufacturing Message Specification (MMS). En tercer lugar, se ha desplegado un servidor para simular un potencial atacante. Como el resto de dispositivos, el servidor del atacante también está conectado a su propio switch. En último lugar, como controlador SDN se ha utilizado Ryu [27], la cual su *interfaz northbound* es utilizada por el módulo MTD para instalar y actualizar reglas de flujo en los switches OpenFlow.

IV-A. Rendimiento

Round Trip Time: Para ver la sobrecarga que introduce en la red el sistema de defensa MTD presentado en este artículo, se ha utilizado la medida *Round Trip Time (RTT)*. El RTT mide el tiempo necesario para que un paquete vaya y vuelva de un receptor. La solución se ha probado con la topología de experimentación en seis escenarios diferentes; por un lado, se han realizado las mediciones en una red estática sin aplicar MTD, simulando un estado normal, y por otro lado se ha utilizado la técnica de aleatorización de direcciones IP con intervalos de 60, 30, 10, 5 y 1 segundo. Para cada escenario, se han realizado 15 mediciones de 200 segundos de duración en cada una de ellas. En la tabla II se representan la media, desviación estándar y valores mínimo/máximo de RTT de los resultados de las mediciones.

Tabla II: RTT entre el servidor SCADA y el PLC con diferentes intervalos MTD.

		Intervalo MTD					
		Sin MTD	60s	30s	10s	5s	1s
RTT (ms)	avg	5.108	5.135	5.145	5.147	5.147	5.169
	stdv	0.091	0.099	0.107	0.122	0.148	0.217
	min	4.73	4.788	4.801	4.819	4.814	4.824
	max	5.666	5.799	6.076	6.023	6.41	6.762

Tamaño de las tablas de flujo: La cantidad de reglas de flujo que realizan las traducciones entre direcciones IP varía dependiendo de la cantidad de dispositivos autorizados que tenga definidas un dispositivo. Para la red MTD presentada en este artículo, son necesarias 4 reglas de flujo en el

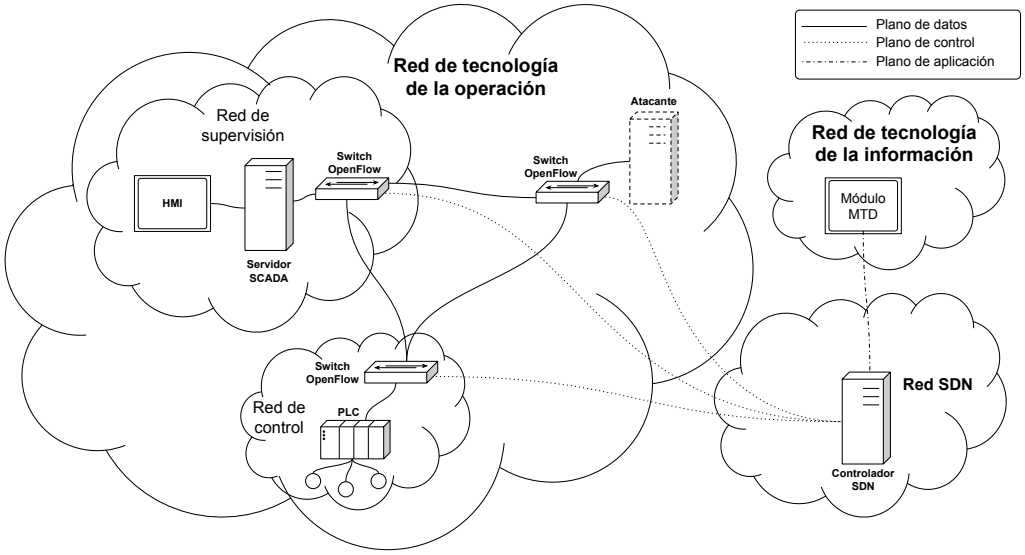


Figura 4: Topología utilizada para la evaluación.

switch de origen y el switch de destino que permitan realizar traducciones $rIP \rightarrow vIP$ y $vIP \rightarrow rIP$ para paquetes de tipo IP y ARP. En el caso de una red estática, si el reenvío de paquetes se realiza utilizando las direcciones IP/MAC de origen y destino de los dispositivos, serían suficientes 2 reglas de flujo por dispositivo autorizado.

Supongamos un conjunto de dispositivos de red $\delta \in \Delta$ conectados a un mismo switch S_w. N representa el numero de dispositivos disponibles en la red. El número necesario de reglas de flujo en una red estática y en una red MTD están definidos en las ecuaciones 1 y 2 respectivamente.

$$F_r(S_w) = 1 + \sum_{\delta \in \Delta} 2N \tag{1}$$

$$F_r^*(S_w) = 1 + \sum_{\delta \in \Delta} 4N \tag{2}$$

La figura 5 muestra la relación entre la cantidad de reglas de flujo necesarias en un switch por cada dispositivo conectado y el número de dispositivos presentes en la red.

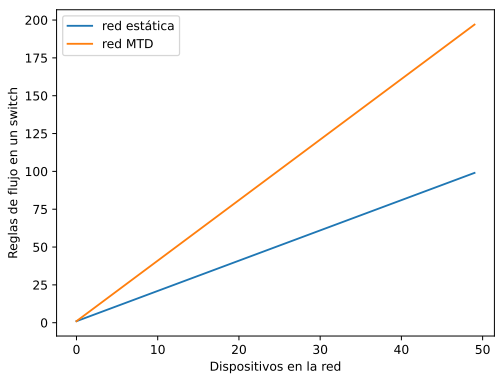


Figura 5: Relación entre número de reglas de flujo y número de dispositivos autorizados.

IV-B. Ataques de reconocimiento

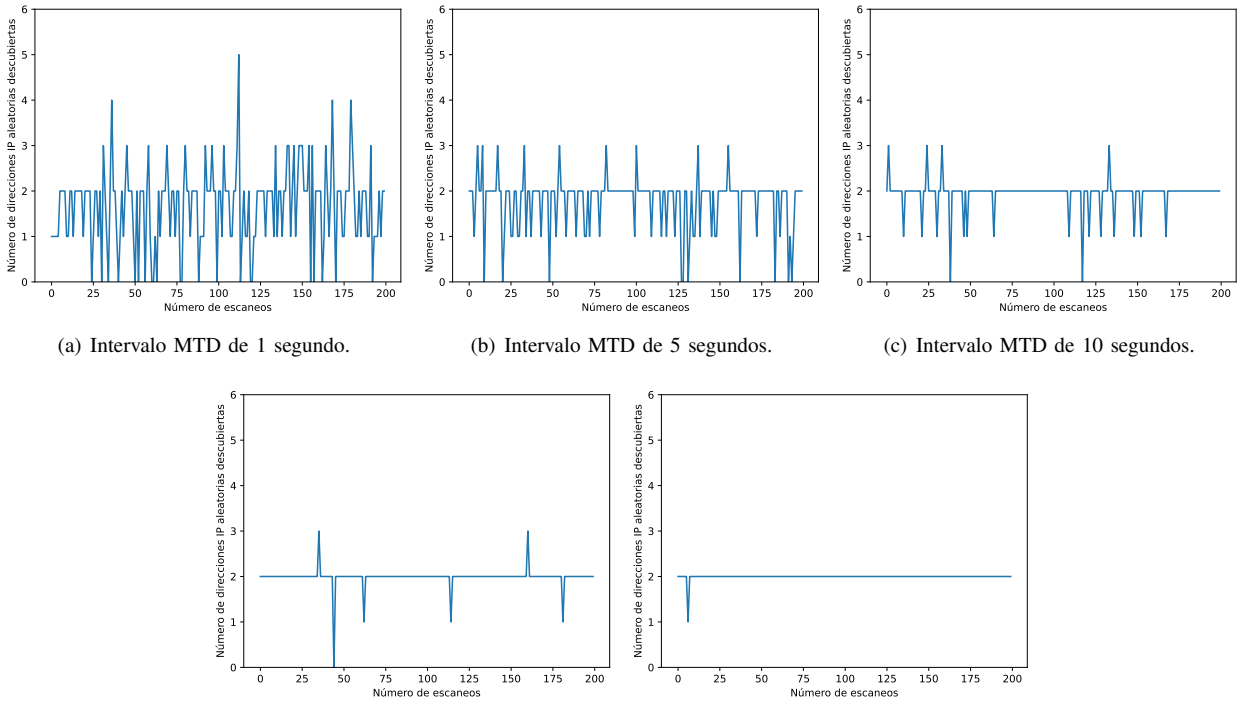
Para probar la eficacia de mitigar ataques de reconocimiento, se han realizado 200 escaneos consecutivos en nuestra red de tipo *clase A* con un espacio de direcciones IP de 2^8. Se ha utilizado la herramienta *nmap* [28] desde el servidor del atacante para realizar escaneos en busca de dispositivos activos en la red. La herramienta *nmap* dispone de múltiples métodos para descubrir dispositivos en una red. Entre estas técnicas se han utilizado y probado las siguientes: *TCP SYN Ping, TCP ACK Ping, ICMP Ping, IP Protocol Ping y ARP Ping*.

Por un lado, se han realizado escaneos en una red estática tradicional sin la defensa proactiva MTD activada. Escaneando todo el rango de direcciones IP, el atacante es capaz de identificar en cada escaneo todos los dispositivos activos de la red.

Por otro lado, con la defensa proactiva MTD activa, se han realizado 200 escaneos consecutivos con diferentes intervalos de aleatorización (1, 5, 10, 30 y 60 segundos). Las diferentes subfiguras representadas en la figura 6 reflejan el número de IPs aleatorias descubiertas por el atacante en 200 escaneos consecutivos y con diferentes intervalos MTD. Podemos observar que conforme el intervalo de aleatorización disminuye, la variación de direcciones IP aleatorias descubiertas entre diferentes escaneos es mayor. Esto se debe a que con intervalos de aleatorización mayores, aumenta la probabilidad de que un escaneo a todo el rango de direcciones IP de la red se realice dentro de los límites de ese intervalo. En intervalos de aleatorización más bajos, la probabilidad de que un escaneo completo se ejecute en más de un intervalo es mayor, aumentando la variabilidad de los resultados.

IV-C. Discusión

El rendimiento en términos de RTT es similar con diferentes intervalos MTD gracias a que la transición a nuevas IPs aleatorias se realiza de forma escalonada utilizando reglas de flujos de respaldo. El intervalo MTD a utilizar dependerá de la criticidad del sistema a defender y es un valor que tiene que ser adaptado para cada caso de uso. Con un intervalo de



(a) Intervalo MTD de 1 segundo. (b) Intervalo MTD de 5 segundos. (c) Intervalo MTD de 10 segundos.

(d) Intervalo MTD de 30 segundos. (e) Intervalo MTD de 60 segundos.

Figura 6: Número de direcciones IP aleatorias descubiertas con diferentes intervalos MTD.

aleatorización menor, la información obtenida por el atacante es más difusa y la información obtenida en reconocimientos anteriores se vuelve irrelevante de forma más frecuente. Al contrario que en intervalos más largos, la información que se obtiene es más constante, incluso se puede obtener la misma información en varios escaneos consecutivos si se realizan en un mismo intervalo MTD. Se podría cuantificar la criticidad de cada dispositivo y definir unos intervalos de aleatorización más frecuentes en las comunicaciones más críticas e intervalos menos frecuentes para comunicaciones menos críticas.

La cantidad de reglas de flujo necesarias en los switches encargadas de la traducción de las direcciones de red aumenta considerablemente en función del número de dispositivos con los que se puede comunicar un dispositivo en concreto. En estos casos, si una tabla de flujos de un switch tiene muchas entradas, el rendimiento de la red puede verse afectado negativamente y es algo a tener en cuenta especialmente en entornos sensibles al tiempo. Como alternativa y para optimizar el proceso en redes grandes donde son necesarias muchas reglas de flujo, se podría utilizar el procesamiento en *pipeline* que proporcionan los switches OpenFlow. Un switch OpenFlow puede contener varias tablas de flujo en cadena donde las reglas pueden ser instaladas en diferentes tablas. De esta forma, se podrían hacer grupos de reglas de flujo para evitar que los paquetes sean procesados por todas las entradas en una única tabla.

Por último, la seguridad por oscuridad es una técnica que utiliza el ocultamiento para proporcionar seguridad. MTD puede considerarse como una técnica de seguridad por oscuridad, especialmente las técnicas de *shuffling* que basan su seguridad en impedir que un atacante descubra posibles vectores de ataque ocultando la configuración, servicios o dispositivos disponibles en la red. Como se cita en el volumen 2 del SP 800-160 de NIST [29], la seguridad por oscuridad no puede ser el principal mecanismo de defensa. Este tipo de técnicas pueden ser utilizadas como una capa complementaria de seguridad en entornos seguros y resilientes.

V. Conclusiones y líneas futuras

Este artículo presenta un mecanismo que aleatoriza las direcciones IP en comunicaciones industriales utilizando el paradigma SDN. El objetivo principal de este sistema es mitigar ataques de reconocimiento y evitar que un equipo no autorizado pueda comunicarse con su objetivo de manera convencional al perder la información obtenida en ataques anteriores. Esto se consigue asignando una dirección IP aleatoria a cada dispositivo de la red que solo son válidas durante un periodo de tiempo limitado. Los resultados demuestran que este sistema ayuda a que la información obtenida a través de ataques de reconocimiento pierda relevancia debido a que solo es válida temporalmente. También, gracias a que la transición a nuevas direcciones IP aleatorias se realiza de forma adaptativa, el retardo introducido en comparación a una red estática tradicional es mínimo, permitiendo que la solución pueda ser implementada en sistemas donde el tiempo es crítico.

Como líneas futuras, tenemos como objetivo desarrollar un sistema que permita detectar comportamientos o comunicaciones sospechosas en sistemas de control industrial con la aleatorización de direcciones IP activa. También, queremos explorar la posibilidad de integración de *honeypots* industriales.

AGRADECIMIENTOS

Este trabajo ha sido desarrollado por el grupo de sistemas inteligentes para sistemas industriales apoyado por el Departamento de Educación, Política Lingüística y Cultura del Gobierno Vasco (IT1676-22). Ha sido parcialmente financiado por el proyecto REMEDY. Este proyecto ha recibido financiación del Departamento de Desarrollo Económico e Infraestructuras bajo el acuerdo de concesión KK-2021/00091.

REFERENCIAS

[1] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ics) security," *NIST special publication*, vol. 800, no. 82, pp. 16–16, 2015.

[2] M. Iturbe, I. Garitano, U. Zurutuza, and R. Uribeetxeberria, "Visualizing network flows and related anomalies in industrial networks using chord diagrams and whitelisting." in *VISIGRAPP (2: IVAPP)*, 2016, pp. 101–108.

[3] J. Zheng and A. S. Namin, "A survey on the moving target defense strategies: An architectural perspective," *Journal of Computer Science and Technology*, vol. 34, no. 1, pp. 207–233, 2019.

[4] M. Sainz, M. Iturbe, I. Garitano, and U. Zurutuza, "Software defined networking opportunities for intelligent security enhancement of industrial control systems," in *International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding*, H. Pérez García, J. Alfonso-Cendón, L. Sánchez González, H. Quintián, and E. Corchado, Eds. Cham: Springer International Publishing, 2018, pp. 577–586.

[5] M. Boucadair and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment," RFC 7149, Mar. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7149

[6] E. Molina and E. Jacob, "Software-defined networking in cyber-physical systems: A survey," *Computers & Electrical Engineering*, vol. 66, pp. 407–419, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790617313368

[7] J.-H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, D. S. Kim, H. Lim, and F. F. Nelson, "Toward proactive, adaptive defense: A survey on moving target defense," *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 709–745, 2020.

[8] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: Transparent moving target defense using software defined networking," in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks*, ser. HotSDN '12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 127–132. [Online]. Available: https://doi.org/10.1145/2342441.2342467

[9] D. P. Sharma, D. S. Kim, S. Yoon, H. Lim, J.-H. Cho, and T. J. Moore, "Frvm: Flexible random virtual ip multiplexing in software-defined networks," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 579–587.

[10] Y. Zhou, G. Cheng, and S. Yu, "An sdn-enabled proactive defense framework for ddos mitigation in iot networks," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5366–5380, 2021.

[11] A. Chowdhary, A. Alshamrani, D. Huang, and H. Liang, "Mtd analysis and evaluation framework in software defined network (mason)," in *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, ser. SDN-NFV Sec'18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 43–48. [Online]. Available: https://doi.org/10.1145/3180465.3180473

[12] A. Aydeger, M. H. Manshaei, M. A. Rahman, and K. Akkaya, "Strategic defense against stealthy link flooding attacks: A signaling game approach," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 751–764, 2021.

[13] Y. Wang, Q. Chen, J. Yi, and J. Guo, "U-tri: Unlinkability through random identifier for sdn network," in *Proceedings of the 2017 Workshop on Moving Target Defense*, ser. MTD '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 3–15. [Online]. Available: https://doi.org/10.1145/3140549.3140554

[14] A. R. Chavez, W. M. Stout, and S. Peisert, "Techniques for the dynamic randomization of network attributes," in *2015 International Carnahan Conference on Security Technology (ICCST)*. IEEE, 2015, pp. 1–6.

[15] H. Koo, Y. Chen, L. Lu, V. P. Kemerlis, and M. Polychronakis, "Compiler-assisted code randomization," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 461–477.

[16] Y. Huang and A. K. Ghosh, "Introducing diversity and uncertainty to create moving attack surfaces for web services," in *Moving target defense*. Springer, 2011, pp. 131–151.

[17] M. Taguinod, A. Doupé, Z. Zhao, and G.-J. Ahn, "Toward a moving target defense for web applications," in *2015 IEEE International Conference on Information Reuse and Integration*, 2015, pp. 510–517.

[18] Y. Li, R. Dai, and J. Zhang, "Morphing communications of cyber-physical systems towards moving-target defense," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 592–598.

[19] A. Kanellopoulos and K. G. Vamvoudakis, "A moving target defense control framework for cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 65, no. 3, pp. 1029–1043, 2020.

[20] H. Alavizadeh, J. B. Hong, J. Jang-Jaccard, and D. S. Kim, "Comprehensive security assessment of combined mtd techniques for the cloud," in *Proceedings of the 5th ACM Workshop on Moving Target Defense*, ser. MTD '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 11–20. [Online]. Available: https://doi.org/10.1145/3268966.3268967

[21] H. Alavizadeh, J. Jang-Jaccard, and D. S. Kim, "Evaluation for combination of shuffle and diversity on moving target defense strategy for cloud computing," in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2018, pp. 573–578.

[22] J. Ulrich, J. Drahos, and M. Govindarasu, "A symmetric address translation approach for a network layer moving target defense to secure power grid networks," in *2017 Resilience Week (RWS)*, 2017, pp. 163–169.

[23] A. C. Pappa, A. Ashok, and M. Govindarasu, "Moving target defense for securing smart grid communications: Architecture, implementation amp; evaluation," in *2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2017, pp. 1–5.

[24] E. Germano da Silva, L. A. Dias Knob, J. A. Wickboldt, L. P. Gaspary, L. Z. Granville, and A. Schaeffer-Filho, "Capitalizing on sdn-based scada systems: An anti-eavesdropping case-study," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, 2015, pp. 165–173.

[25] G. K. Ndonda and R. Sadre, "A low-delay sdn-based countermeasure to eavesdropping attacks in industrial control systems," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017, pp. 1–7.

[26] O. N. Foundation. (2015) Openflow switch specification, version 1.3.5. [Online]. Available: https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.3.5.pdf

[27] Ryu sdn framework. [Online]. Available: https://ryu-sdn.org/

[28] Nmap: the network mapper - free security scanner. [Online]. Available: https://nmap.org/

[29] R. Ross, V. Pillitteri, R. Graubart, D. Bodeau, and R. McQuaid, "Developing cyber resilient systems: a systems security engineering approach," National Institute of Standards and Technology, Tech. Rep., 2019.