Universidad de Buenos Aires
Facultad de Ciencias Económicas, Ciencias Exactas y Naturales e Ingeniería

Carrera de Especialización en Seguridad Informática.

Trabajo Final De Especialización

Seguridad en redes conmutadas y enrutadas Cisco

Autor: Juan Felipe Torres Santamaría
Tutor del Trabajo Final: Javier Vallejos

Año de Presentación: 2021
Cohorte del Cursante: 2020

Declaración Jurada de origen de los contenidos

Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Trabajos Finales vigente y se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual".

Firmado
Juan Felipe Torres Santamaría
C.C: 1018448959, Colombia.

Resumen

El presente Trabajo Final de Especialización se enfoca en identificar y analizar los aspectos de seguridad de los protocolos que se configuran en los conmutadores y enrutadores del fabricante Cisco, el cual ha aparecido en numerosas ocasiones como líder en el cuadrante mágico de Gartner para infraestructura de acceso LAN inalámbrica y por cable [1], infraestructura WAN de borde [2], redes de centros de datos [3], entre otros. En primer lugar, efectuaré una evaluación de los algoritmos criptográficos empleados por el *Internetwork Operating System* (IOS) de Cisco, enfocándome en verificar si actualmente son considerados como seguros o inseguros. En segundo lugar, realizaré una descripción de los protocolos más relevantes que se emplean para configurar los planos de datos, control y gestión de estos dispositivos de red. En tercer lugar, mostraré los comandos que se emplean en el IOS de Cisco para la configuración de los diferentes protocolos. Finalmente, a partir del análisis del nivel de seguridad de los algoritmos criptográficos, la descripción de los protocolos y los comandos que se pueden ejecutar para su configuración, estableceré recomendaciones y cuestiones que se deben tener en cuenta para garantizar su protección.

Palabras clave:

- Conmutador.
- Enrutador.
- IOS.

Fundamentación del tema elegido

Los conmutadores y enrutadores generalmente incluyen una seguridad mínima o nula de forma predeterminada, lo que permite a un atacante acceder fácilmente al sistema operativo del dispositivo y a la red interna del usuario. Posteriormente, el atacante toma control y captura los datos que se transmiten y almacenan. Las consecuencias para la víctima pueden incluir la pérdida de información personal, datos financieros, contraseñas de cuentas y su identidad.

Así mismo, en las redes de conmutadores y enrutadores se utilizan protocolos como Telnet y FTP que son intrínsecamente inseguros, puesto que no cifran las conexiones, permitiendo que datos sensibles como nombres de usuario y contraseñas sean fácilmente interceptados [4]. Este tipo de protocolos no debería emplearse, sin embargo, es común encontrarlos configurados en redes en producción.

Del mismo modo, se emplean versiones inseguras de protocolos, que deberían dejar de configurarse en favor de versiones seguras. Por ejemplo, SNMPv1 y SNMPv2c usan cadenas de comunidad, similar a contraseñas sin cifrar, como estrategia de autenticación para obtener acceso a un dispositivo administrado. Por el contrario, SNMPv3 mediante algoritmos hash garantiza que un mensaje SNMP no se modifique en tránsito y valida su origen [4].

Finalmente, la mayoría de los protocolos deben ser puestos a punto ya que su configuración básica deja brechas de seguridad. Protocolos de enrutamiento de puerta de enlace interior como EIGRP y OSPF envían de forma predeterminada mensajes de saludo a través de las interfaces cuyas redes han sido referenciadas para ser enrutadas. El envío de estos mensajes sin cifrar a través de interfaces no seguras permite a un usuario malintencionado crear vecinos no deseados, a partir de los cuales puede aprender, inyectar y modificar las rutas existentes [4].

Este trabajo de especialización se enfoca en identificar las problemáticas de seguridad previamente mencionadas, enfocándose en los planos de datos, control y gestión, de los dispositivos conmutadores y enrutadores de Cisco. Además, a través de un proceso de identificación y análisis, busca determinar las prácticas que deben seguirse para mitigar los riesgos que representan estas vulnerabilidades y malas prácticas de configuración.

Objetivos

- Exponer la evolución a nivel de seguridad que han tenido los dispositivos conmutadores y enrutadores.
- Evaluar el nivel de seguridad de los algoritmos criptográficos empleados por el IOS de Cisco.
- Analizar las vulnerabilidades de seguridad que se presentan en los protocolos del plano de datos de conmutadores y enrutadores Cisco, así como las prácticas que deben ser empleadas para su mitigación.
- Examinar los problemas de seguridad que se presentan en los protocolos del plano de control de conmutadores y enrutadores Cisco, al igual que las recomendaciones para tratarlos.
- Evaluar las debilidades de seguridad que se presentan en los protocolos del plano de gestión de conmutadores y enrutadores Cisco, junto con las sugerencias para reforzarlas.

Alcance

En el presente Trabajo Final de Especialización incluye una breve reseña de la evolución en seguridad y funcionamiento de los conmutadores y enrutadores. Así mismo, contiene una evaluación del nivel de seguridad de los algoritmos criptográficos empleados por el IOS de Cisco. Del mismo modo, incorpora una explicación de los protocolos más importantes que se utilizan para configurar los planos de datos, control y gestión en estos dispositivos de red. Igualmente, abarca los comandos de configuración que se emplean en el IOS de Cisco para la configuración de los protocolos previamente descritos. Finalmente, contiene recomendaciones que se derivan del análisis de seguridad de los algoritmos criptográficos, la descripción de los protocolos y los comandos que pueden ser ejecutados para su configuración.

El presente Trabajo Final de Especialización no contiene un diseño topológico de red físico ni lógico, ni tampoco la configuración de conmutadores o enrutadores utilizando el IOS de Cisco. Por consiguiente, las recomendaciones y conclusiones se fundamentan en un análisis teórico. Sin embargo, como Trabajo Final de Maestría lo aquí expuesto será materia de trabajo, de modo que se pueda realizar un análisis más profundo del funcionamiento real y configuración segura de estos dispositivos de red.

Relevancia.

Partiendo del análisis desarrollado dentro de este Trabajo Final de Especialización, se espera poder identificar y determinar la configuración que asegura los protocolos más relevantes utilizados en los planos de datos, control y gestión, de los conmutadores y enrutadores de Cisco. Lo anterior, permitirá a los ingenieros de redes y personas interesadas en el campo de la seguridad informática tener una base para la implementación segura de los protocolos en redes en producción.

Tabla de contenido

1. Evolución de los dispositivos conmutadores y enrutadores

1.1. Evolución de las redes conmutadas

Las redes de comunicación surgieron de la necesidad de compartir un conjunto de recursos distribuidos e intercambiar información. En el caso de las redes locales (LAN) prevalecieron las llamadas redes de difusión, en las cuales todas las estaciones comparten un mismo canal de comunicación. Dos topologías de red se definieron para permitir la interconexión y el acceso al canal de comunicaciones. Por una parte, las redes con topología en bus cuyo acceso al canal se resolvía por contienda, dando lugar a frecuentes colisiones entre las estaciones que deseaban transmitir. Por otra parte, las redes con topología en anillo en las que un testigo definía la estación que podía enviar información a través del canal. Razones de orden económico hicieron que las redes con topología en bus y con protocolo de acceso al canal basado en contienda, definidas como Ethernet, prevalecieran [5].

Las redes Ethernet utilizaban generalmente cable coaxial y conectores BNC. La topología física de la red era en bus y se instalaba en cada estación una tarjeta de red que tenía una interfaz física que la conectaba al canal, generando tantos puntos de falla como estaciones. Además, la red era completamente plana, ya que la topología física (la forma como se interconectaban las estaciones) coincidía con la topología lógica (la forma como las estaciones pugnaban por el uso del canal) [5].

Las estaciones de red interconectadas con topología en bus dieron lugar al concepto de dominio de colisión. Un dominio de colisión es el conjunto de todas aquellas estaciones que en un momento dado compiten por el uso del canal. En la medida que el dominio de colisión es más grande, así también lo es el número potencial de colisiones, ya que resulta imposible predecir con exactitud el momento en el cual una estación va a transmitir, disminuyendo el desempeño de la red [5].

El siguiente paso en la evolución tuvo lugar con la aparición de los sistemas de cableado estructurado. El cable coaxial fue cambiado por cable *Unshielded Twisted Pair* (UTP), lo que significó un retroceso en términos del canal, dado que las características de transmisión del cable coaxial son superiores a las ofrecidas por el cable UTP, pero permitió centralizar la red y reducir los puntos de falla al integrar un concentrador (*hub*) [5].

Los concentradores transmiten los datos a través de todos sus puertos excepto aquel por el que los reciben, motivo por el cual todos los dispositivos conectados al concentrador reciben los datos, pero únicamente el dispositivo destino las procesa [5]. Lo anteriormente descrito, implica que los concentradores mantienen la baja eficiencia de la red y presentan serios problemas de seguridad, debido a que cualquier usuario malintencionado que tiene acceso al concentrador recibe todos datos que son enviados. En consecuencia, a pesar de reducir los puntos de falla, se puede establecer que en este etapa de la evolución, se evaluaban únicamente las mejoras en la transmisión de datos, sin tener en cuenta aspectos de seguridad.

El crecimiento desmesurado en el volumen de tráfico generado en el dominio de colisión produjo un colapso en las redes, que fue denominado como la “crisis del ancho de banda”. Para enfrentar este desafío se presentaron dos posibles soluciones: la utilización de tecnologías de alta velocidad y la segmentación. La utilización de tecnologías de alta velocidad se concentraba en contar con un canal de mayor capacidad, pero no se hacía nada respecto al tamaño del dominio de colisión, la fuente del problema. Por el contrario, segmentar la red implicaba descomponer un dominio de colisión en dos o más, de tal manera que se disminuyera el número de estaciones de red que pugnan por el uso del canal. Este último enfoque, enfrentó las causas del problema y provocó un replanteamiento en el diseño y en la estructura de red dando origen al conmutador (*switch*) [5].

Un conmutador trabaja en la capa de enlace de datos del modelo OSI (capa 2) y crea un dominio de colisión independiente para cada puerto, permitiendo la transferencia de datos sin interferencia. De esta forma, los dispositivos que se conectan a los conmutadores en conexiones punto a punto disfrutan de un ancho de banda full-duplex dedicado [6]. Se debe tener en cuenta, que la creación de dominios de colisión por puerto permite a un dispositivo recibir y transmitir sin pugnar por el canal, pero la información sigue siendo transmitida a todos los equipos de la red.

Para entender el próximo paso en la evolución de la red es necesario entender el concepto de dominio de difusión. El tráfico de difusión viaja desde una única fuente a todos los destinos de una subred, esta subred define el dominio de difusión. La difusión excesiva tiene un efecto indeseado sobre el desempeño de la red, porque la congestiona e implica un consumo de recursos de cómputo en cada estación, ya que todas deben procesarlo [5].

Para contener el tráfico de difusión se propusieron dos opciones: la utilización de redes virtuales (VLAN) y el desarrollo de conmutadores que extendieron el concepto de conmutación hacia la capa de red del modelo OSI (capa 3). Una VLAN es un agrupamiento virtual por software de nodos en un dominio de difusión que permite contener y separar los flujos de tráfico. Por otra parte, un conmutador de capa tres es capaz de segmentar la red para separar el tráfico de difusión [5]. Al segmentar el tráfico, ambas soluciones permiten mejorar la seguridad de la red, ya que se pueden establecer un grupo de dispositivos que deben comunicarse y asignar aquellos con los que no se debe interactuar a otra VLAN o red. Lo anteriormente expuesto implica un primer avance en la seguridad de las redes conmutadas.

Como paso final en el desarrollo, ha surgido el concepto de conmutación multicapa (*multilayer switching*). Los dispositivos multicapa combinan características de capa dos y capa tres en conmutadores híbridos que pueden enrutar paquetes a velocidades de hardware. Además, a medida que la tecnología ha evolucionado se ha adicionado la posibilidad de analizar aquella información contenida en los paquetes que pudiera colaborar en el proceso de reenvío. De esta forma, los conmutadores multicapa examinan la información relacionada con la capa 2 hasta la capa 7 del modelo OSI, aumentando los niveles de seguridad de la red [5]. Con las funcionalidades descritas previamente, los conmutadores multicapa integran en un único dispositivo las capacidades de conmutación, enrutamiento y seguridad, lo que los hace adaptables y les permite satisfacer las diferentes necesidades de comunicación y seguridad que tienen las redes en la actualidad.

Cisco propone una estructura para la red constituida por tres niveles: el nivel de acceso, el nivel de distribución y el nivel núcleo. Las responsabilidades de cada nivel son complementarias, jerárquicas y especializadas. El nivel de acceso es el punto donde se conectan las estaciones de usuario. Por su parte, el nivel de distribución constituye el punto de acceso a los diferentes grupos de trabajo, por la naturaleza de su ubicación, es el sitio en donde se imponen las restricciones de acceso y de seguridad. Finalmente, el nivel de núcleo se encarga de transportar grandes volúmenes de datos a gran velocidad [5].

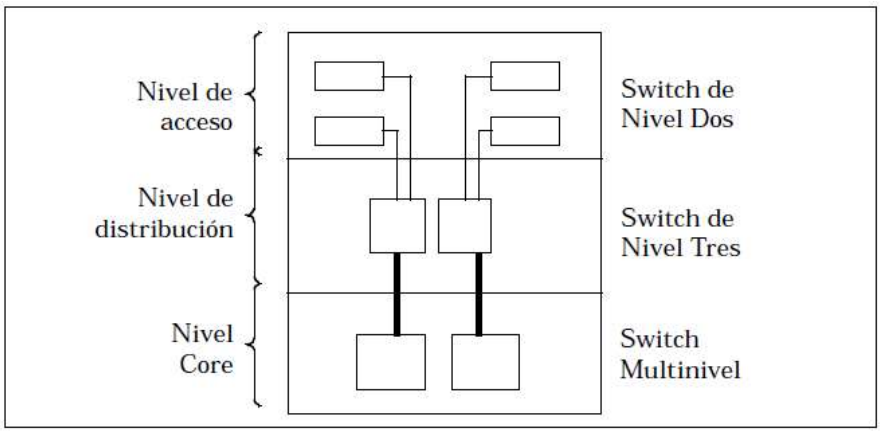

Figura 1. Estructura jerárquica de red [5].

1.2. Funcionamiento de las redes conmutadas

Un conmutador de capa 2 se encarga del reenvío de tramas. Para conocer dinámicamente la ubicación de las estaciones, un conmutador escucha las tramas entrantes, inspecciona su dirección MAC de origen y mantiene una tabla de direcciones MAC de reenvío. Si una trama posee una MAC de origen que aún no está en la tabla, la dirección MAC, el puerto del conmutador y la VLAN son registradas [7].

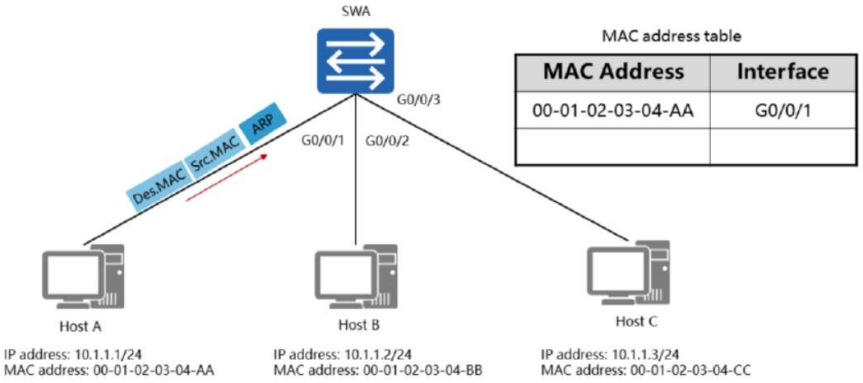

Figura 2 Envío de solicitud ARP por parte de un dispositivo de usuario final [7].

Las tramas entrantes también incluyen la dirección MAC de destino. Nuevamente, el conmutador busca esta dirección en la tabla de direcciones. Si se encuentra, la trama se puede reenviar al puerto del conmutador correspondiente. Si por el contrario la dirección no se ha registrado, el conmutador reenvía la trama desbordando todos los puertos asignados a la VLAN de origen, exceptuando el puerto de entrada. Esto se conoce como inundación de unidifusión desconocida, porque se desconoce la ubicación del destino de unidifusión [7].

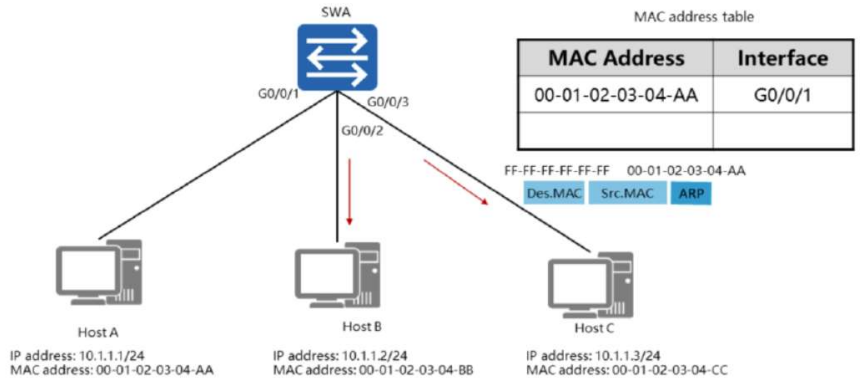

Figura 3. Reenvió de la solicitud ARP por parte del conmutador [7].

El dispositivo con la dirección solicitada responde a la inundación de unidifusión desconocida, de este modo, el conmutador puede registrar su MAC, puerto y VLAN. El próximo envío hacia esta dirección MAC se realiza de manera directa empleando una trama de unidifusión [7]

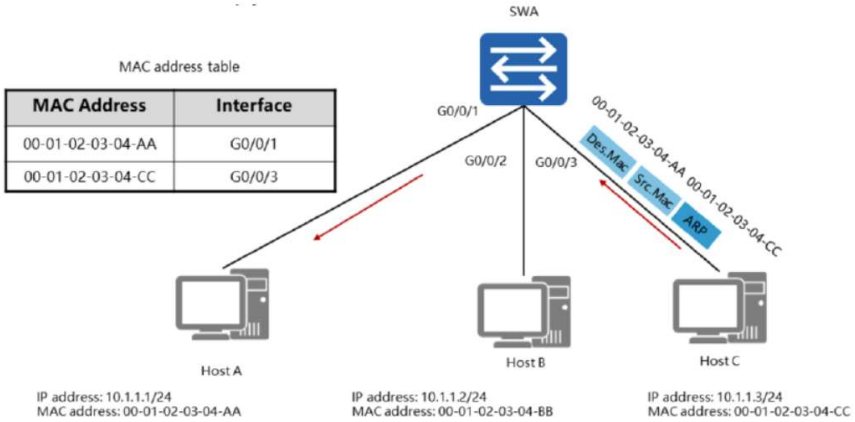

Figura 4. Respuesta de la solicitud ARP [7].

Como se expone en el capítulo 3.1.2.1, este funcionamiento de las redes conmutadas puede ser explotado a través de un ataque que genera que el conmutador se comporte como un concentrador, reenviando las tramas a todos los dispositivos conectados. Del mismo modo, como se muestra en el capítulo 3.1.2.4.3, un usuario malintencionado puede falsificar su dirección MAC para responder a solicitudes ARP y realizar un ataque de hombre en el medio. Sin embargo, existen diferentes funcionalidades expuestas en este Trabajo Final de Especialización, que se han desarrollado y que permiten asegurar las redes conmutadas.

En su nivel más básico, un conmutador permite aislar los medios de Ethernet entre dispositivos conectados de las siguientes formas [6]:

- Cada puerto brinda un dominio de colisión, evitando que el medio sea compartido y que las estaciones tengan tomar turnos para transmitir.
- Las conexiones pueden funcionar en modo full-dúplex, debido a que no existe control de acceso al medio.
- Cada puerto de conmutador ofrece ancho de banda dedicado.
- Cada trama recibida en un puerto de conmutador se comprueba en busca de errores. Solamente las tramas sin errores se transmiten.
- Puede limitar el tráfico de difusión a un umbral.

1.3. Evolución de las redes enrutadas

Los fundamentos del enrutamiento de red fueron desarrollados a finales de la década de 1960 por un grupo académico de investigadores conocido como *Advanced Research Projects Agency* (ARPA). El objetivo de la agencia era construir una red de cuatro procesadores de mensajes de interfaz (IMP) para enviar datos a través de líneas telefónicas. ARPA lanzó una licitación para el desarrollo, que fue ganada por una pequeña empresa conocida como Bolt, Beranek y Newman (BBN) [8].

En agosto de 1968, BBN usó computadoras Honeywell 516 como IMP para transmitir, enrutar, almacenar y verificar datos entre nodos de la red, creando el antecesor del enrutador de red moderno. El 1 de octubre de 1969, el primer conjunto de caracteres se transmitió con éxito a través de la nueva red y se fundó la *Advanced Research Projects Agency Network* (ARPANET) [8].

En 1980, la Universidad de Stanford desarrolló un multiprocesador conocido como "*Blue Box*", que proporcionaba la capacidad para que las escuelas y departamentos de Stanford se comunicaran entre sí. William Yeager, un ingeniero investigador de la facultad de medicina, escribió un programa de enrutamiento que permitía conectar las computadoras del departamento médico con las del departamento de informática. Yeager pasó a escribir un programa más sofisticado, a través del cual pudo enrutar varios protocolos, incluido el relativamente nuevo Protocolo de Internet (IP) [8].

En este punto de la evolución, los desarrollos se fundamentaban en poder integrar en los enrutadores diferentes tipos de interfaces que permitieran la interconexión entre dispositivos de red como unidades centrales, estaciones de trabajo, miniordenadores e impresoras. Para lograr este objetivo, se desarrollaron varios protocolos que, sin embargo, no contemplaban la transmisión de datos, el control de su flujo ni la administración de los enrutadores de manera segura.

En 1985, Len Bosack y Sandy Lerner, quienes habían estado involucrados en el proyecto *Blue Box*, le pidieron a Yeager su código de software para modificarlo de modo que pudieran enrutar solamente protocolos de Internet. De manera controvertida, el año anterior, Bosack y Lerner habían fundado una empresa llamada Cisco y utilizaron el trabajo de Yeager para impulsar el desarrollo de sus productos. En 1986, Cisco introdujo su primer enrutador comercial multiprotocolo, llamado *Advanced Gateway Server* (AGS), que admitía TCP/IP [8].

La década de 1990 se inició con una serie de eventos que simbolizaron la evolución continua y la comercialización que pronto llegaría de Internet. ARPAnet dejó de existir y en su lugar los proveedores comerciales de servicios de Internet empezaron a transportar el tráfico de la red troncal. El evento principal de la década de 1990 fue la aparición de la aplicación World Wide Web, que llevó Internet a los hogares y empresas de millones de personas en todo el mundo [9].

La acelerada evolución del Internet ha venido de la mano de nuevos requerimientos y necesidades para las redes de enrutadores, encargadas de realizar todo el proceso de conectividad. Además, los fabricantes incorporan nuevas funcionalidades cada día con la finalidad de que sus dispositivos sean más atractivos para los consumidores finales, por ejemplo, puertos que funcionan como conmutadores y conectividad inalámbrica a través de Wifi o LTE. Sin embargo, a pesar de su importancia, la seguridad no ha sido un aspecto determinante en su desarrollo y es frecuente encontrarse con configuraciones por defecto, protocolos inseguros y versiones heredadas en ambientes productivos. En consecuencia, fabricantes como Cisco brindan diferentes opciones que permiten bridar protección a estos dispositivos de red y que serán analizadas a lo largo del presente Trabajo Final de Especialización.

1.4. Funcionamiento de las redes enrutadas

Los enrutadores se encargan del envío de paquetes entre dispositivos que están en subredes diferentes. Un enrutador conoce una red si se configura manualmente o se emplea un protocolo de enrutamiento dinámico para intercambiar y actualizar información de esa red. Una vez el enrutador conoce la red, emplea algoritmos que le permiten determinar la ruta que debe tomar un paquete y lo reenvía hacia la interfaz de salida correspondiente [4].

Cuando un dispositivo final envía un paquete, primero analiza si el destino se encuentra dentro de su mismo segmento de red. Si el destino se encuentra en su misma red, realiza en procedimiento mencionado en el capítulo 1.2. Sin embargo, si el destino se encuentra en una red diferente, realiza una solicitud ARP al enrutador que cumple la función de puerta de enlace predeterminada. La dirección IP de la puerta de enlace se puede configurar manualmente o aprenderse dinámicamente a través del protocolo de configuración dinámica de host (DHCP).

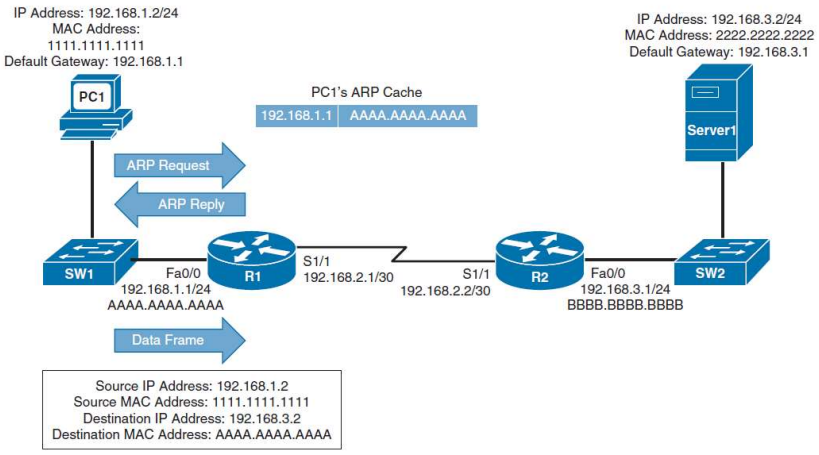

Figura 5. Enrutamiento básico, envío del paquete a la puerta de enlace [4].

El enrutador recibe la trama y revisa el encabezado IP. Posteriormente, verifica su tabla de enrutamiento (creada a través del intercambio y análisis de información de enrutamiento), determina la mejor ruta y reenvía el paquete hacia la red de destino.

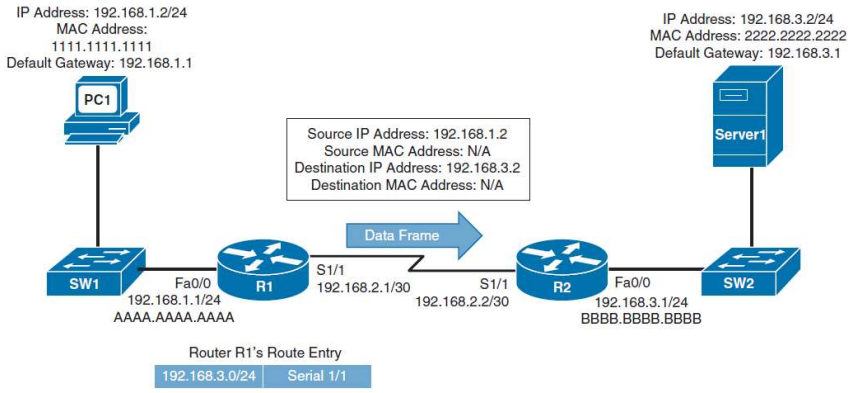

Figura 6. Enrutamiento básico, reenvío del paquete por la puerta de enlace [4].

Como se muestra en el capítulo 3.1.2.4.1, este proceso de enrutamiento puede ser vulnerado al introducir un servidor DHCP que falsifica la dirección IP de la puerta de enlace predeterminada. De este modo, todos los paquetes que se envían a una subred diferente pasan primero por el dispositivo del atacante, donde se efectúa un ataque de hombre en el medio. Del mismo modo, como se expone en los capítulos 3.2.2.1.2 y 3.2.2.1.2, los protocolos de enrutamiento también son vulnerables ante ataques que alteran la tabla de enrutamiento y que permiten controlar el flujo de los paquetes que se trasmiten entre las diferentes redes de una compañía y hacia Internet. No obstante, se han desarrollado funcionalidades para proteger las redes enrutadas que se exponen en el presente Trabajo Final de Especialización.

2. Algoritmos Criptográficos

Los conmutadores y enrutadores de Cisco emplean para asegurar sus planos de datos, control y administración diferentes algoritmos criptográficos. El análisis de estos algoritmos es necesario para conocer si han sido quebrados o siguen siendo criptográficamente seguros. En este capítulo se analizan los criptosistemas simétricos, asimétricos de clave pública, de reducción y de autenticación de mensajes empleados por el IOS de Cisco.

2.1. Algoritmos de Cifrado

Según el padre de la teoría de la información Claude Shannon, hay dos operaciones primitivas con las que se pueden construir algoritmos de cifrado fuertes [10]:

- ✓ Confusión: es una operación de cifrado en la que la relación entre la clave y el texto cifrado se oscurece. Hoy en día, un elemento común para lograr confusión es la sustitución.
- ✓ Difusión: es una operación de cifrado en la que la influencia de un símbolo de texto sin formato se extiende sobre muchos símbolos de texto cifrado con el objetivo de ocultar las propiedades estadísticas del texto sin formato. Un elemento de difusión simple es la permutación.

Los cifrados que solo generan confusión o difusión no son seguros. Sin embargo, a través de la concatenación de tales operaciones, se puede construir un cifrado fuerte [10].

2.2. Criptografía simétrica

Considerando un criptosistema (**P, C, K, E, D**) tal que:

- ✓ **P** es el conjunto de los posibles textos planos.
- ✓ **C** es el conjunto de los posibles textos cifrados.
- ✓ **K**, es el conjunto de las posibles claves.
- ✓ $\forall$ **k** $\in$ **K** $\exists$ una regla de cifrado $\mathbf{e_k} \in \mathbf{E}$ y de descifrado $\mathbf{d_k} \in \mathbf{D}$ tales que $\mathbf{e_k: P \rightarrow C}$ y $\mathbf{d_k: C \rightarrow P}$ Son funciones biunívocas **/** $\mathbf{d_k(e_k(x)) = x}$ $\forall$ **x** $\in$ **X**

Es simétrico si para cada par asociado (**e**, **d**) es posible, dado uno de ellos, computar el otro. Usualmente **e = d**, pero podría usarse cualquier función conocida biunívoca tal que **d = f(e)**.

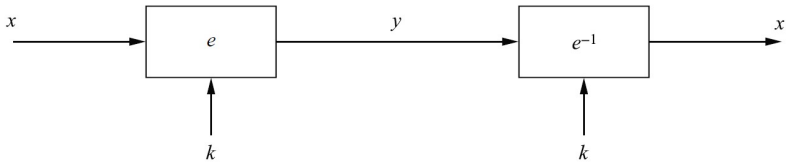

Figura 7. Principio de cifrado de clave simétrica [10].

Dado que las claves de cifrado y descifrado en los criptosistemas simétricos son computables una a partir de la otra (generalmente iguales), se requiere de un canal de comunicación seguro para su distribución. Adicionalmente, una red con **n** usuarios posee **n(n-1)/2** claves y cada usuario almacena **n-1** claves, lo cual implica un exigencia muy alta de capacidad de almacenamiento [10].

A continuación, se describen los criptosistemas simétricos empleados por el IOS de Cisco para asegurar los planos de datos, control y gestión de sus dispositivos conmutadores y enrutadores.

2.2.1. El Cifrado de Vigenère

El cifrado de Vigenère es un método de sustitución polialfabética debido a que transforma los elementos del alfabeto en diferentes caracteres [10]. Por ejemplo, para la clave **K** = CIPHER y el texto plano **P** = INSECURE CRYPTOSYSTEM, el texto cifrado **C** es:

	I	N	S	E	C	U	R	E	C	R	Y	P	T	O	S	Y	S	T	E	M
P	8	13	18	4	2	20	17	4	2	17	24	15	19	14	18	24	18	19	4	12
K	2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7	4	17	2	8
C	10	21	7	11	6	11	19	12	17	24	2	6	21	22	7	5	22	10	6	20
	K	V	H	L	G	L	T	M	R	Y	C	G	V	W	H	F	W	K	G	U

Figura 8. Ejemplo de cifrado de Vigenère.

El método de Vigenère es fácilmente cripto analizable empleando el test de Kasiski junto con el índice mutuo de coincidencia. El test de Kasiski permite descubrir la longitud de clave, por su parte, el índice mutuo de coincidencias presenta máximos cuando hay correlación significativa debida a la redundancia del lenguaje [11]. En consecuencia, este método de cifrado es inseguro y se debe evitar su uso en redes productivas.

2.2.2. Estándar de cifrado de datos (DES)

DES fue escogido como *Federal Information Processing Standard* (FIPS) por el gobierno de los Estados Unidos en 1976. Este algoritmo generó gran controversia, las críticas más relevantes contra su fuerza criptográfica se centraron en dos argumentos. Primero, el espacio de claves de 56 bits es demasiado pequeño, lo que lo hace vulnerable a ataques de fuerza bruta. Segundo, los criterios de diseño se mantuvieron en secreto, generando sospechas sobre la posible existencia de una puerta trasera que podía ser empleada por sus diseñadores [10].

Para el proceso de cifrado, un texto plano **P** se divide en bloques $\mathbf{P_i}$ con una longitud de 64 bits. Posteriormente, se realiza una permutación bit a bit inicial **IP** de cada bloque. Luego, **IP** se divide en dos mitades $\mathbf{L_0}$ y $\mathbf{R_0}$ de 32 bits que son la entrada a una red denominada como Feistel, que consta de 16 rondas. La estructura de Feistel solo cifra la mitad izquierda de los bits de entrada por cada ronda y su salida se puede expresar como [10]:

$$L_i = R_{i-1}, i = 1, \dots, 16$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i), i = 1, \dots, 16$$

Después de la ronda 16, las mitades de 32 bits $\mathbf{L_{16}}$ y $\mathbf{R_{16}}$ se intercambian nuevamente, y la permutación final $\mathbf{IP^{-1}}$ es la última operación de DES que corresponde a la inversa de la permutación **IP** inicial. En cada ronda, se deriva una clave $\mathbf{k_i}$ a partir de la clave **K** principal [10].

Las dos propiedades básicas del cifrado descritas por Shannon, se realizan dentro de la función **f**. Una vez que la función **f** se ha diseñado de forma segura, la seguridad de un cifrado Feistel aumenta con el número de bits de clave utilizados y el número de rondas [10].

A pesar de un criptoanálisis muy intensivo durante la vida útil de DES, los ataques analíticos actuales no son muy eficientes. Sin embargo, DES se puede romper con relativa facilidad con un ataque exhaustivo de búsqueda de claves debido al tamaño empleado de 56 bits. Adicionalmente, DES es vulnerable al criptoanálisis diferencial que permite disminuir la complejidad que representa un ataque de fuerza bruta [10]. Por lo tanto, el uso de DES no es recomendado.

2.2.3. 3DES

3DES consta de tres cifrados DES consecutivos con claves diferentes. Sean **P** el texto plano y **C** el texto cifrado, 3DES utiliza el siguiente modo de cifrado-descifrado-cifrado [10].

$$\mathrm{C} = \mathrm{E}_{\mathrm{k3}}(\mathrm{D}_{\mathrm{k2}}(\mathrm{E}_{\mathrm{k1}}(\mathrm{M})))$$
$$\mathrm{P} = \mathrm{D}_{\mathrm{k1}}(\mathrm{E}_{\mathrm{k2}}(\mathrm{D}_{\mathrm{k3}}(\mathrm{C})))$$

En 1992 Keith W. Campbell y Michael J. Wiener lograron determinar en su trabajo *DES is not a Group*, que el conjunto de permutaciones DES no está cerrado bajo la composición funcional y por lo tando no es grupo. En consecuencia, si la clave usual de DES emplea 56 bits, la clave de 3DES usa 112 bits para dos claves y 168 bits en caso de utilizar 3 claves, permitiendo aumentar la complejidad del algoritmo [10].

$$\mathrm{E}_{\mathrm{k3}}\left(\mathrm{D}_{\mathrm{k2}}\left(\mathrm{E}_{\mathrm{k1}}(\mathrm{M})\right)\right) \neq \mathrm{E}_{\mathrm{k4}}(\mathrm{M})$$

3DES sigue siendo considero como un algoritmo simétrico de cifrado seguro, capaz de resistir los ataques de fuerza bruta. Sin embargo, no es muy eficiente cuando se implementa en software y es 3 veces más lento que DES. Además, su tamaño de bloque relativamente corto de 64 bits es un inconveniente en ciertas aplicaciones. Finalmente, la computación cuántica requiere de longitudes de clave del orden de 256 bits. Todas estas consideraciones llevaron al NIST a la conclusión de que se necesitaba un cifrado de bloques completamente nuevo como reemplazo de DES [10].

2.2.4. Estándar de cifrado avanzado (AES)

DES fue oficialmente reemplazado en 2001 por AES (Rijndael). AES es el cifrado simétrico más utilizado en la actualidad y no se conocen ataques analíticos con una probabilidad razonable de éxito. AES cifra bloques de 128 bits empleando claves de 128, 192 o 256 bits. A diferencia de DES, AES no tiene una estructura Feistel y cifra los 128 bits en una iteración [10].

Se necesitan cálculos de campo de Galois para todas las operaciones dentro de AES. En términos generales, un campo de Galois es un conjunto finito de elementos en el que podemos sumar, restar, multiplicar e invertir. El número de elementos en el campo se llama orden. Un campo de orden **m** solo existe si **m** es una potencia prima, es decir, $\mathbf{m = p^n}$, para algún entero positivo **n** y un entero primo **p**. AES posee un orden de 256 y se denota como $\mathbf{GF(2^8)}$. Se eligió este campo porque cada uno de los elementos del campo se puede representar como un byte [10].

Dado que el orden del campo finito $\mathbf{GF(2^8)}$ no es primo, las operaciones de suma y multiplicación se representan como polinomios con coeficientes en **GF(2)**. Los polinomios tienen un grado máximo de **m**−1, por lo que hay **m** coeficientes en total para cada elemento. En el campo $\mathbf{GF(2^8)}$, que se utiliza en AES, cada elemento $\mathbf{A} \in \mathbf{GF(2^8)}$ se representa como [10]:

$$A(x) = a_7x^7 + \cdots + a_1x + a_0, a_i \in GF(2) = \{0, 1\}$$

AES consta de capas (*layers*) que manipulan los 128 bits de la ruta de datos, también conocida como estado del algoritmo (*state*). Cada ronda, con la excepción de la primera, consta de tres capas [10]. A continuación, se describe cada una de estas.

2.2.4.1. Capa de Sustitución

Esta capa se puede visualizar como una fila de 16 S-Box paralelas, cada una con 8 bits de entrada y salida. Las 16 S-Box son idénticas y tienen una estructura algebraica fuerte. Una S-Box puede verse como una transformación matemática de dos pasos [10].

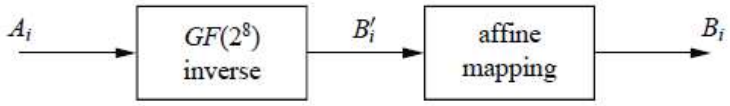

Figura 9 Las dos operaciones dentro del AES S-Box [10].

La inversión en **GF(2^8)** proporciona un alto grado de no linealidad, garantizando una óptima protección contra algunos de los ataques analíticos más fuertes conocidos. El paso afín "destruye" la estructura algebraica del campo de Galois, previniendo ataques que explotan la inversión del campo finito [10].

2.2.4.2. Capa de difusión

En AES, la capa de difusión consta de dos subcapas, la transformación ShiftRows y la transformación MixColumn. La transformación ShiftRows desplaza cíclicamente las filas de la matriz de estado. Por su parte, la transformación MixColumn mezcla cada columna de la matriz de estado. Dado que cada byte de entrada influye en cuatro bytes de salida, la operación MixColumn es el principal elemento de difusión en AES [10].

2.2.4.3. Capa de adición de claves

El número de rondas internas del cifrado es una función de la longitud de la clave. Así mismo, el número de subclaves es igual al número de rondas más uno, debido a una adición XOR de una subclave que se utiliza previo a la primera ronda [10].

key lengths	# rounds = n_r
128 bit	10
192 bit	12
256 bit	14

Tabla 1. Longitudes de clave y número de rondas para AES [10].

Las dos entradas a la capa de adición de claves son la matriz de estado actual de 16 bytes y una subclave que también consta de 16 bytes. Las dos entradas se combinan mediante una operación XOR bit a bit. A través de un programa de claves se toma la clave de entrada original y se derivan las subclaves utilizadas en cada ronda de AES. Existen diferentes programas de claves para los tres tamaños de clave AES, pero son bastante similares. Las subclaves AES se calculan de forma recursiva, es decir, para derivar la subclave $\mathbf{k_i}$, se debe conocer la subclave $\mathbf{k_{i-1}}$ [10].

2.2.4.4. Seguridad de AES

AES utiliza aritmética de campo de Galois y proporciona una fuerte difusión y confusión, a través de las capas de sustitución, difusión y adición de claves. Lo anteriormente descrito, permite a AES proporcionar una excelente seguridad a largo plazo y evitar el criptoanálisis. Hasta el momento, no se conoce mejor sistema de ataque que el de fuerza bruta. Del mismo modo, AES es eficiente en software y hardware. Por lo tanto, AES es el algoritmo criptográfico simétrico que debe utilizarse en los protocolos empleados por enrutadores y conmutadores de Cisco.

2.3. Criptografía asimétrica

En los criptosistemas de clave pública el receptor del mensaje posee una clave **k** que consta de dos partes, una pública $\mathbf{k_{pub}}$ que es conocida por todos y se utiliza para cifrar el mensaje, y una privada $\mathbf{k_{pr}}$ que solo es conocida por el receptor y que se emplea para descifrar el mensaje [10].

Las principales funcionalidades que proporciona la criptografía de clave pública incluyen el establecimiento de claves secretas a través de un canal inseguro, el no repudio e integridad del mensaje empleando algoritmos de firma digital, la identificación de entidades utilizando protocolos de desafío-respuesta y firmas digitales, y el cifrado de mensajes para garantizar su confidencialidad. Además, la capacidad de almacenamiento de claves se reduce drásticamente en comparación con los criptosistemas simétricos, ya que se pasa de un sistema de **n(n-1)/2** claves a uno de **2n** claves. Sin embargo, el uso de criptosistemas simétricos es necesario debido a que tienen menor carga y emplean claves mucho más cortas que los asimétricos, lo que los hace más veloces. Por lo tanto, se cifra con criptosistemas híbridos [10].

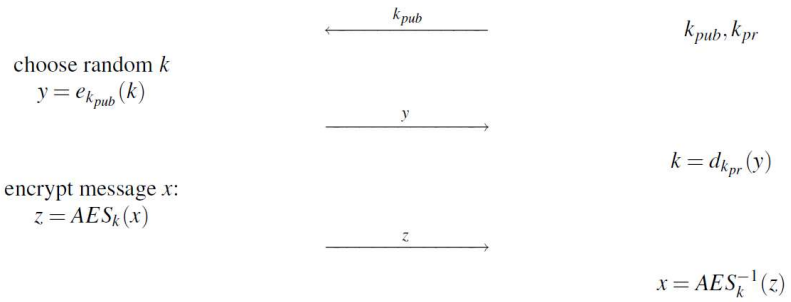

Figura 10. Cifrado con criptosistemas híbridos [10].

A continuación, se describen los criptosistemas asimétricos empleados por el IOS de Cisco para asegurar los planos de datos, control y gestión de sus dispositivos conmutadores y enrutadores.

2.3.1. Rivest, Shamir y Adleman (RSA)

RSA es el algoritmo criptográfico de clave pública más utilizado y se emplea tanto para cifrar como para firmar digitalmente. Su funcionamiento se basa en el problema de factorización de enteros. Dados dos números primos grandes, es fácil calcular el producto. Sin embargo, es muy difícil factorizar el producto resultante. El algoritmo es el siguiente [10]:

- Elegir dos primos distintos **p** y **q**
- Calcular **n=p·q**, **ø(n)=(p-1)·(q-1)**
- Elegir c tal que **MCD(c, ø(n))=1**
- Calcular **d**, solución de la ecuación modular **c·dmod(ø(n))=1**
- La clave privada es **(c,n)** y la pública es **(d,n)**
- El menaje cifrado corresponde a $\mathbf{M_1 = M^c mod(n)}$
- El mensaje descifrado corresponde a $\mathbf{M = M_1^d mod(n)}$

Empleando este sistema cada usuario tiene un clave pública **(d,n)** y una privada **(c,n)**, donde todos los números involucrados son enteros. La seguridad de este sistema criptográfico depende directamente de salvaguardar los valores de **p**, **q**, **ø(n)** y **(d, n)** [10].

Se han propuesto numerosos ataques contra RSA. Sin embargo, suelen centrarse en aprovechar las debilidades de implementación en lugar del algoritmo en sí. Los ataques de protocolo aprovechan las debilidades en la forma en que se utiliza RSA, los más conocidos explotan la maleabilidad de RSA y pueden evitarse mediante el uso de relleno (*padding*). Por otra parte, los métodos criptoanalíticos se basan en la factorización del módulo, descomponiéndolo en sus números primos **p** y **q**. Para evitar este ataque, se recomienda elegir parámetros RSA en el rango de 2048 a 4096 bits los cuales garantizan la seguridad a largo plazo. Finalmente, los ataques de canal lateral explotan información sobre la clave privada que se filtra a través de canales físicos, como el consumo de energía o el comportamiento del tiempo. Hoy en día, se conocen numerosas contramedidas para evitar estos ataques [10].

En síntesis, es necesario tener en cuenta las consideraciones de seguridad previamente expuestas al momento de utilizar e implementar RSA en los dispositivos conmutadores y enrutadores de Cisco. Desde la parte de configuración que es determinada por el ingeniero de red y que no depende del fabricante, siempre es necesario al momento de generar el par de claves RSA emplear un módulo de por lo menos 2048 bits o superior.

2.4. Algoritmos de reducción

Para un mensaje en particular, el resumen del mensaje o valor hash, puede verse como la huella digital, es decir, una representación única. El uso de funciones hash en criptografía es múltiple: las funciones hash son una parte esencial de los esquemas de firma digital, códigos de autenticación de mensajes y almacenamiento de contraseñas [10].

A diferencia de todos los demás algoritmos criptográficos, las funciones hash no tienen claves. Hay tres propiedades centrales que las funciones hash deben poseer para ser seguras [10].

- Resistencia a la preimagen: las funciones hash deben ser unidireccionales, es decir, dada una salida hash **z**, debe ser computacionalmente imposible encontrar un mensaje de entrada **x** tal que $\mathbf{z = h(x)}$.
- Resistencia a la segunda preimagen: es esencial que dos mensajes diferentes no tengan el mismo valor hash. Esto significa que debería ser computacionalmente inviable crear dos mensajes diferentes $\mathbf{x_1 \neq x_2}$ con valores hash iguales $\mathbf{z_1 = h(x_1) = h(x_2) = z_2}$. En este caso, se fija $\mathbf{x_1}$ y se varía el valor de $\mathbf{x_2}$.
- Resistencia a colisiones: no es factible computacionalmente encontrar dos entradas diferentes $\mathbf{x_1 \neq x_2}$ con $\mathbf{h(x_1) = h(x_2)}$. A diferencia de la resistencia a segunda preimagen, el atacante es libre de elegir tanto $\mathbf{x_1}$ como $\mathbf{x_2}$, por lo que esta propiedad es más difícil de lograr.

Para frustrar los ataques de colisión basados en la paradoja del cumpleaños, la longitud de salida de una función hash debe ser aproximadamente el doble de la longitud de salida que protege contra un segundo ataque de preimagen. Por esta razón, todas las funciones hash tienen una longitud de salida de al menos 128 bits [10].

Las funciones hash MD4, MD5 y SHA1 han sido comprometidas a través de ataques de colisiones, que fueron publicados en los artículos *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD* y *How to Break MD5 and Other Hash Functions* y *Finding collisions in the full SHA-1.*

Por otra parte, ataques a SHA-256 y SHA-512 publicados en el artículo *Preimage Attacks on 41-Step SHA-256 and 46-Step SHA-512,* consiguieron romper 41 de las 64 iteraciones del SHA-256 y 46 de las 80 iteraciones del SHA-512. Además, SHA-2 ha sido debilitado por el ataque de colisiones diferenciales (Chabaud-Joux), el cual genera ataques de preimagen y segunda preimagen que crean colisiones empleando los mismo principios del criptoanálisis diferencial [12].

Como consecuencia de las debilidades previamente expuestas, la NIST desarrolló la *SHA-3 Cryptographic Hash Algorithm Competition*, cuyo vencedor fue el algoritmo KECCAK. Según los requisitos de la NIST, KECCAK satisface los siguientes requisitos de seguridad: (i) al menos una variante de la función hash soporta de forma segura HMAC y hash aleatorio. (ii) para todos los valores de resumen de **n** bits, proporciona una resistencia de preimagen de aproximadamente **n** bits, (iii) una resistencia de segunda preimagen de aproximadamente **n - L** bits, donde la primera preimagen tiene una longitud máxima de **2L** bloques, (iv) una resistencia a colisiones de aproximadamente **n/2** bits, y (v) es resistente al ataque de alargamiento-extensión. Finalmente, (vi) para cualquier **m ≤ n**, la función KECCAK especificada, tomando un subconjunto fijo de **m** bits de la salida de la función, satisface las propiedades (ii) - (v) con **n** reemplazado por **m** [13].

Dado que MD4, MD5 y SHA1 han sido quebrados empleando ataques de colisiones, SHA-256 y SHA-512 han sido debilitados por ataques de colisiones diferenciales y KECCAK (SHA-3) cumple con los requerimientos anteriormente expuesto, debe ser este último algoritmo de reducción el empleado siempre que se requiera implementar una función hash.

2.5. HMAC

Un código de autenticación de mensaje (MAC) agrega una etiqueta de autenticación a un mensaje **x** empleando una clave simétrica **k**, que se emplea tanto para generar una etiqueta de autenticación como para verificarla [10].

Al momento de enviar el mensaje **x**, el emisor calcula el MAC **m** en función de **x** y **k**. Posteriormente, envía tanto **x** como **m** al receptor. Al recibir el **x** y **m**, el receptor vuelve a calcular el código de autenticación de mensaje con el mensaje recibido **x** y la clave simétrica **k** [10].

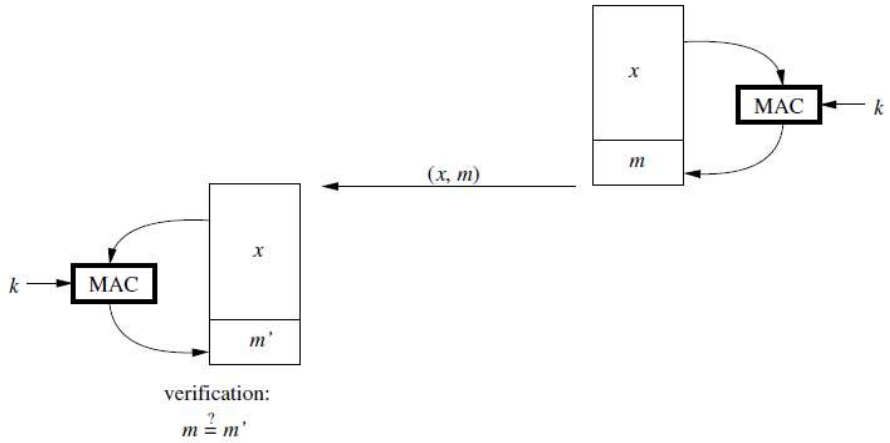

Figura 11. Principio de los códigos de autenticación de mensajes (MAC) [10].

El cálculo del MAC produce un resultado incorrecto si el mensaje **x** fue alterado en tránsito y permite al receptor verificar si el emisor fue el creador del mensaje, ya que solo las dos partes comparten la clave secreta **k**. En consecuencia, un MAC brinda integridad y autenticación de mensajes. Sin embargo, debido a que una clave secreta simétrica no está ligada a una determinada persona sino a dos partes, un juez no puede distinguir entre el emisor y el receptor en caso de disputa. Por consiguiente, un MAC no brinda no repudio [10].

En la práctica, los MAC se construyen a partir de cifrados en bloque o de funciones hash que crean los denominados códigos de autenticación de mensajes basados en hash (HMAC) [10], el IOS de Cisco emplea esta última opción que se expone a continuación.

Un esquema de HMAC seguro consiste en un hash interno y un hash externo, que se define con la siguiente ecuación [10].

$$\mathrm{HMAC}_k(m) = h\{(k^+ \oplus \mathrm{opad}) || h[(k^+ \oplus \mathrm{ipad}) || x]\}$$

- || denota concatenación.
- **k^+** es la expansión de la clave simétrica **k** con ceros a la izquierda hasta la longitud bloque de entrada de la función hash.
- opad es el relleno exterior (01011100,01011100, . . . ,01011100)
- ipad es el relleno interior (00110110,00110110, . . . ,00110110).

En términos de eficiencia computacional, debe tenerse en cuenta que el mensaje **x** solo tiene hash en la función interna. Por lo tanto, la carga computacional introducida a través de la construcción del HMAC es muy baja. Además, se puede demostrar que, si un atacante puede romper el HMAC, también puede romper la función hash utilizada para su construcción, lo que requiere encontrar colisiones o calcular la salida de la función sin conocer su valor inicial. Por lo tanto, la fuerza criptográfica de HMAC depende de la potencia criptográfica de la función de hash subyacente, el tamaño de su salida y el tamaño y calidad de la clave [10]. Tomando como base lo descrito en el capítulo 2.4, para brindar seguridad al HMAC, es necesario emplear SHA-3 con algoritmo criptográfico de reducción.

3. Aseguramiento de los planos de datos, control y gestión en los conmutadores y enrutadores Cisco

Existen tres planos principales dentro de los dispositivos conmutadores y enrutadores de Cisco que deben protegerse: el plano de datos, el plano de control y el plano de gestión. Esta protección se realiza mediante el proceso de configuración, motivo por el cual, es necesario explicar las convenciones que el software Cisco IOS utiliza para describir los comandos que son ejecutados [14].

- Barras verticales (|): separan los argumentos alternativos y mutuamente excluyentes.
- Corchetes ([]): indican elementos opcionales.
- Llaves ({}): indican una opción requerida.
- Llaves entre corchetes ([{}]): indican opciones requeridas dentro de elementos opcionales.
- Corchetes angulares (<>): indican argumentos en contextos que no permiten la cursiva, y en los ejemplos indican cadenas de caracteres que ingresa un usuario que no aparecen en la pantalla.
- **Negrita**: indica comandos y palabras clave.
- *Cursivas*: indican variables de usuario.

Además, la interfaz de línea de comando (CLI) de CISCO, usa una estructura jerárquica que requiere el ingreso a distintos modos. Cada modo determina los comandos que pueden ser ejecutados. El IOS de Cisco posee tres modos principales que, por razones de seguridad, establecen diferentes niveles de acceso [15].

Modo de comando	**Funcionalidad**
EXEC de usuario >	Realizar pruebas básicas, enumerar información del sistema y verificar el estado del dispositivo. No permite ningún comando de configuración.
EXEC privilegiado #	Configurar los parámetros operativos y de administración del dispositivo, motivo por el cual debe estar protegido con contraseña.
Configuración global (config)#	Ejecutar comandos que se aplican a las funciones que afectan al dispositivo en su conjunto.

Tabla 2. Modos principales de comando del IOS de Cisco [15].

Desde el modo de configuración global se puede acceder a otros submodos de operación más específicos. Al ingresar a estos modos específicos, el prompt de entrada al dispositivo cambia para señalar el modo de configuración en uso. Todo cambio de configuración afecta únicamente los procesos relativos a ese modo particular [15].

- Submodo de configuración de interfaz (config-if)#
- Submodo de configuración de línea (config-line)#

A continuación, se definen cada uno de los planos previamente mencionados, se hace una descripción sobre los protocolos más importantes que los conforman, se muestran los comandos que emplea el IOS de Cisco para su configuración y se exponen aquellos que deben ser ejecutados, teniendo en cuenta lo expuesto en el capítulo 2 y la funcionalidad de cada comando, para garantizar la seguridad.

3.1. Aseguramiento del plano de datos

El plano de datos se ocupa de todas las tareas de reenvío. En los conmutadores, estas tareas corresponden al envío de tramas de acuerdo con la tabla de direcciones MAC y el control de acceso entre redes de área local virtual (VLAN). La falta de protección de este plano en los conmutadores permite a usuarios fraudulentos acceder a la red falsificando su dirección MAC, ejecutar ataques de desbordamiento de la tabla CAM, ingresar de forma ilegítima a diferentes VLAN y reenviar paquetes DHCP deshonestos [6]. Por su parte, en un enrutador estas tareas corresponden al encapsulamiento de una trama de datos en un paquete, el envío de paquetes de acuerdo con la tabla de enrutamiento y el filtrado de mensajes empleando listas de control de acceso (ACL). La desprotección de este plano en los enrutadores permite a los atacantes insertar contenido malicioso como virus o troyanos, falsificar sus direcciones IP y controlar el tránsito de paquetes en la red [16].

3.1.1. Seguridad del plano de datos en enrutadores

3.1.1.1. Listas de control de acceso (ACL)

Una ACL es una lista secuencial de instrucciones que permiten o deniegan las direcciones IP y los protocolos de la capa de transporte. Existen diferentes tipos de ACL que brinda gran flexibilidad para asegurar la red. Los tipos de ACL definidos por Cisco son los siguientes [4].

- ACL estándar: Hacen coincidir el tráfico según la dirección IP de origen.
  Router(config)# **access-list** *number* {**permit** | **deny**} {*source-wildcard* | **any**}
- ACL extendidas: Hacen coincidir el tráfico según la dirección IP de origen, la dirección IP de destino y una variedad de otros criterios, como los números de puerto.
  Router(config)# **access-list** *number* {**permit** | **deny**} *protocol source-address source-wildcard destination-address destination-wildcard*
- ACL basadas en el tiempo: Permiten que protocolos específicos sean admitidos o filtrados, dependiendo de rangos de tiempo periódicos o absolutos.
  Router(config)# **access-list** *number* <*matching-parameters*> **time-range** *name*
- ACL de infraestructura: Se trata de una ACL extendida con varias instrucciones, que se aplica a los enrutadores ubicados en el borde de una red empresarial, con el propósito de evitar que el tráfico malicioso ingrese.

Las ACL expuestas previamente, permiten asegurar el plano de datos de los dispositivos. Además, existen ACL que ayudan a proteger el plano de control, por ejemplo, se puede configurar una ACL que restrinja la entrega de actualizaciones de enrutamiento de un protocolo en particular. Del mismo modo, existen ACL que aseguran el plano de gestión, por ejemplo, es posible configurar una ACL que permita únicamente el tráfico *Secure Shell* (SSH), bloquee el tráfico Telnet y controle la dirección IP de origen desde la cual se realiza la conexión [4].

Debido a que las ACL estándar únicamente contienen la IP de origen, se deben aplicar en el enrutador donde se encuentra la red destino, con el fin de evitar que el tráfico de ese origen sea bloqueado para otros destinos. Por otra parte, las ACL extendidas al poseer tanto la IP de origen como la IP destino y número de puerto, deben configurarse cerca al destino ya que contienen la información necesaria para efectuar un bloqueo granular y su aplicación en otra parte de la red genera un consumo innecesario de ancho de banda.

Por su parte, la aplicación efectiva de las ACL basadas en el tiempo, depende directamente de la configuración de la hora local. Como se explica en el capítulo 3.3.5, esta configuración puede aplicarse de forma manual o empleando el protocolo de tiempo de red (NTP), siendo esta última opción la más recomendada ya que permite la sincronización precisa y automática de todos los dispositivos de red.

Finalmente, las ACL de infraestructura deben bloquear los paquetes fragmentados, permitir los protocolos de enrutamiento necesarios y el tráfico de gestión de red, bloquear todo el demás tráfico destinado a la red interna y permitir el tráfico cuyo origen y destino es la red externa [4].

En resumen, dada la gran flexibilidad, posibilidades de aplicación y beneficios en materia de seguridad para los planos de datos, control y gestión de enrutadores y conmutadores Cisco, las ACL son una herramienta fundamental que debe analizarse y configurarse en los puntos críticos de la red.

3.1.1.2. Reenvío de ruta inversa de unidifusión (uRPF)

El protocolo uRPF evita que el tráfico malintencionado ingrese a la red al bloquear los paquetes que contienen una dirección IP falsificada. Cuando una paquete llega a una interfaz del enrutador, uRPF verifica que la dirección IP sea accesible según la base de información de reenvío (FIB) del enrutador. Adicionalmente, el enrutador puede verificar si el paquete llega a la misma interfaz que usaría para enviar el tráfico de regreso a esa dirección IP [4].

Existen 2 modos de operación de uRPF [4]:

- Modo estricto (*strict*): verifica que la dirección IP de origen de un paquete entrante sea accesible según la FIB del enrutador y revisa que el paquete llegue por la misma interfaz que el enrutador usaría para enviar el tráfico de regreso a esa dirección IP.
- Modo suelto (*loose*): únicamente verifica que la dirección IP de origen de un paquete sea accesible según la FIB del enrutador.

El comando que debe ser ejecutado para configurar la funcionalidad uRPF, se muestra en seguida.

Router(config-if)# **ip verify unicast source reachable-via** {**rx** | **any**} [**allow-default**] [**allow-self-ping**] [*acl*]

- **rx**: habilita uRPF en modo estricto.
- **any**: habilita uRPF en modo suelto.
- **allow-default**: acepta una ruta predeterminada como forma válida de devolver un paquete a una dirección IP de origen.
- **allow-self-ping**: permite que un enrutador se haga ping a sí mismo al verificar la accesibilidad de una dirección IP.
- *acl*: Identifica una lista de control de acceso que permite o deniega el tráfico que no supera la verificación de uRPF.

Después de una falla en la verificación de uRPF, si un paquete coincide y es permitido por la ACL asociada, se transmite. Sin embargo, si un paquete no pasa la verificación de uRPF y la ACL asociada lo niega, el paquete se descarta. A continuación, se puede observar un ejemplo de topología en el que se aplica el protocolo uRPF realizado por Cisco [4].

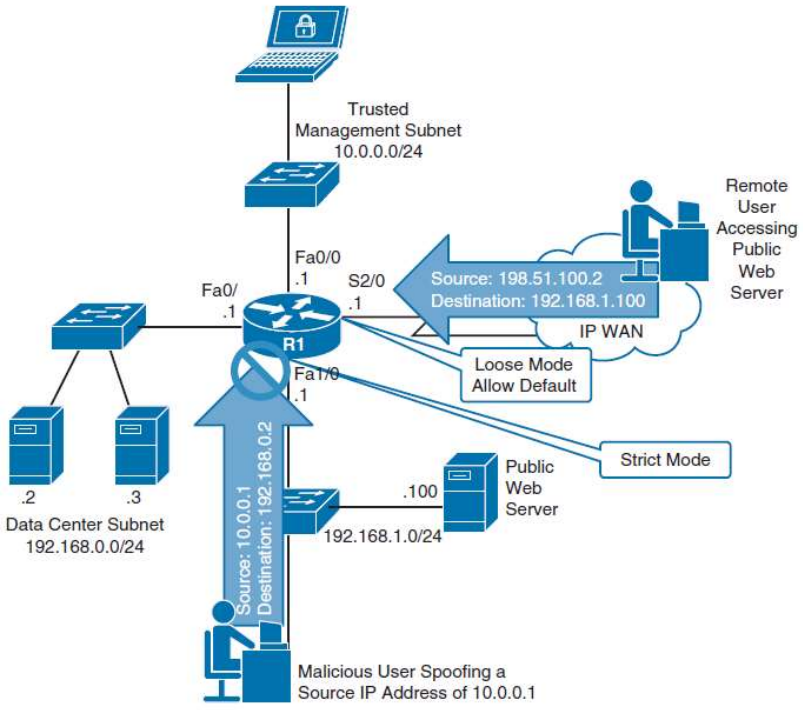

Figura 12.Ejemplo de topología para uRPF [4].

Como se puede observar, dentro de la red LAN se debe utilizar el modo uRPF estricto, ya que permite descartar el tráfico en caso de que se reciba en una interfaz que no coincide con la verificación de uRPF. Por otra parte, se requiere el uso del modo uRPF suelto para la interfaz WAN, agregando la opción **allow-default**, puesto que es necesario admitir la ruta predeterminada como una forma válida de volver a una dirección IP de origen. Finalmente, se recomienda evitar la opción **allow-self-ping**, puesto que la verificación de uRPF siempre será válida, lo que representa un riesgo de seguridad. Aplicando las consideraciones previamente mencionadas, uRPF permite bloquear los paquetes que poseen una dirección IP falsificada.

3.1.2. Seguridad del plano de datos en conmutadores

3.1.2.1. Seguridad del puerto (*Port Security*)

Los conmutadores Cisco permiten controlar el acceso a la red a través de la funcionalidad de seguridad de puertos, que controla las direcciones MAC de los dispositivos que se pueden conectar. Cuando las estaciones de trabajo son estacionarias, se espera que las direcciones MAC sean siempre las mismas. Por otra parte, si las estaciones son móviles, sus direcciones MAC se aprenden de forma dinámica o agregando listas de forma manual [6].

La seguridad del puerto se configura para que un puerto aprenda un número específico de direcciones MAC, lo que limita el número de asignaciones MAC/puerto que puede aprender, evitando los ataques de desbordamiento de tabla CAM. Este ataque consiste en llenar la tabla CAM con asignaciones falsas de direcciones MAC/puerto. Una vez que la tabla CAM se llena, el conmutador entra en modo de apertura por falla e inunda todos los puertos, permitiendo a un atacante capturar todos los datos de la red y crear una denegación de servicios. Los pasos de configuración son los siguientes [6].

- Habilitar la seguridad del puerto.
  Switch(config-if)# **switchport port-security**
- Especificar el número máximo de direcciones MAC permitidas. Por defecto, la seguridad de puertos admite una única dirección MAC.

Switch(config-if)# **switchport port-security maximum** *max_addr*

- Configurar el puerto para que las direcciones MAC aprendidas sean persistentes ante reinicios del conmutador.
  Switch(config-if)# **switchport port-security mac-address sticky**
- (Opcional) Definir estáticamente una o más direcciones MAC.
  Switch(config-if)# **switchport port-security mac-address** *mac-addr*
- Definir la acción que ejecuta el puerto si se supera el número máximo de direcciones MAC permitidas.
  Switch(confi-if)# **switchport port-security violation** {**shutdown** | **restrict** | **protect}**
  - ➔ **shutdown**: es el modo por defecto, el puerto se coloca inmediatamente en el estado desactivado por error y se apaga. El puerto debe volver a habilitarse manualmente o mediante una recuperación automática.
  - ➔ **restrict**: el puerto permanece activo, pero se descartan los paquetes. El conmutador mantiene un recuento continuo del número de paquetes infractores, y puede enviar una captura SNMP y un mensaje syslog como alerta de la infracción.
  - ➔ **protect**: el puerto permanece activo, descarta los paquetes de direcciones infractoras, pero no mantiene ningún registro ni envía alertas.

En base al análisis anteriormente descrito, se concluye que es necesario realizar un análisis de la red para determinar el número máximo de estaciones de trabajo que se pueden conectar a un puerto en particular. Así mismo, se recomienda que las direcciones aprendidas dinámicamente sean persistentes ante reinicios, para lo cual es necesario habilitar el aprendizaje de direcciones MAC pegajoso (**sticky**). Del mismo modo, se sugiere el uso de la opción de infracción **restrict**, ya que restringe efectivamente los paquetes, no requiere una intervención adicional para habilitar el puerto, mantiene un registro e informa sobre la infracción. Empleando las prácticas previamente mencionadas, se puede realizar un control efectivo sobre las estaciones de trabajo que se conectan a la red.

3.1.2.2. Autenticación basada en puertos (IEEE 802.1X)

Los conmutadores de Cisco admiten la autenticación basada en puertos, una combinación de autenticación AAA y seguridad de puertos. Tanto el conmutador como el dispositivo final deben admitir el estándar 802.1X, a través del protocolo de autenticación extensible sobre LAN (EAPOL) [6].

Si una PC está configurada para utilizar 802.1X, pero el conmutador no lo admite, la PC abandona el protocolo y se comunica normalmente. Sin embargo, si el conmutador está configurado para 802.1X, pero la PC no lo admite, el puerto del conmutador permanece en el estado no autorizado y no reenvíe el tráfico a la PC. El estado autorizado del puerto finaliza cuando el usuario cierra la sesión [6].

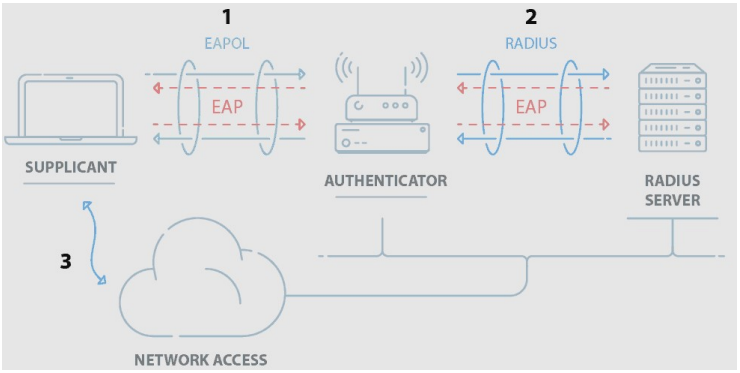

Figura 13. Componentes de 802.1X [17].

Aunque muchas plataformas de conmutadores Cisco permiten configurar otros métodos de autenticación, solo RADIUS es compatible con 802.1X. El proceso de configuración es el siguiente [6].

- Habilitar AAA en el conmutador.
  Switch(config)# **aaa new-model**
- Definir el servidores RADIUS, junto con una contraseña precompartida que permite cifrar la sesión de autenticación.
  Switch(config)# **radius-server host** {*hostname* | *ip-address*} [**key** *string*]
- Definir RADIUS como el método de autenticación para 802.1X.
  Switch(config)# **aaa authentication dot1x default group radius**

- Habilitar 802.1X en el conmutador.
  Switch(config)# **dot1x system-auth-control**
- Configurar cada puerto de conmutador que utilizará 802.1X.
  Switch(config-if)# **dot1x port-control** {**force-authorized** | **force-unauthorized** | **auto**}
  - ➔ **force-authorized**: estado predeterminado, el puerto autoriza a todos los clientes sin autenticación.
  - ➔ **force-unauthorized**: el puerto no autorizar a ningún cliente conectado.
  - ➔ **auto**: si la solicitud 802.1X tiene éxito, el puerto pasa del estado no autorizado al estado autorizado.

En síntesis, la autenticación basada en puertos se considera más flexible y segura que la seguridad del puerto, ya que existen métodos que permiten quebrar esta última funcionalidad al modificar por software la dirección MAC de los dispositivos. Por el contrario, la autenticación 802.1X asegura la conexión al emplear un servidor RADIUS que solicita credenciales de usuario y contraseña para habilitar el puerto. Dentro de las opciones de control de puerto se recomienda el uso de funcionalidad **auto** que habilita la autenticación 802.1X. Por el contrario, se debe prevenir el uso de la opción **force-authorized**, puesto que cualquier dispositivo que se conecta tiene acceso a la red, evadiendo la validación contra el servidor RADIUS.

3.1.2.3. Protección contra acceso ilegítimo a una VLAN

Una red de área local virtual (VLAN) permite crear redes lógicas independientes dentro de una misma red física. El protocolo IEEE 802.1Q se utiliza para transportar diferentes VLAN a través de enlaces troncales, empleando información de etiquetado e introduciendo el concepto de VLAN nativa. Las tramas que pertenecen a la VLAN nativa no se encapsulan con ninguna información de etiquetado, lo que proporciona encapsulación troncal para dispositivos no compatibles con 802.1Q [6].

3.1.2.3.1. Suplantación de identidad del conmutador

De modo predeterminado, los puertos de un conmutador pueden negociar dinámicamente su uso y su modo de encapsulación mediante el intercambio de mensajes del protocolo de enlace troncal dinámico (DTP). DTP puede ser aprovechado por un usuario malintencionado que simula conectar un conmutador y negocia un enlace troncal. De esta forma, el atacante tiene acceso a cualquier VLAN que tenga permiso para pasar por la troncal que, de forma predeterminada, son todas las VLAN configuradas en el conmutador [6].

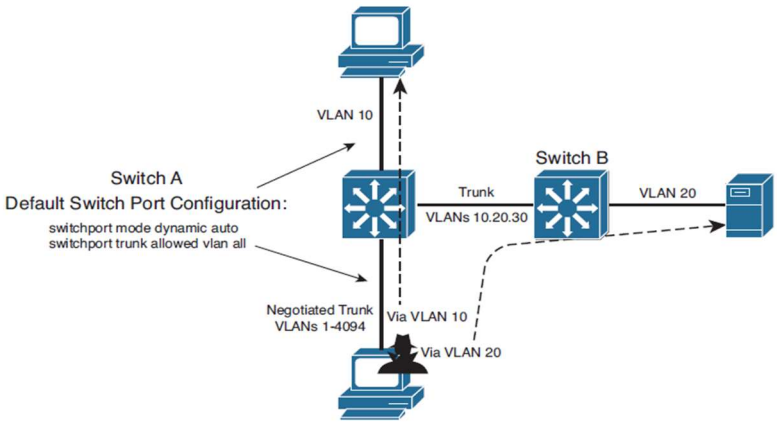

Figura 14. Suplantación de conmutador para obtener acceso a un enlace troncal [6].

Como se observa, el atacante se conecta a un puerto de conmutador y simula la negociación de un puerto troncal. Una vez finaliza el proceso de negociación se establece el enlace troncal y el atacante recibe los paquetes destinados a todas las VLAN. En consecuencia, es necesario fijar el modo de enlace como troncal fijo o acceso y posteriormente deshabilitar la funcionalidad DTP empleando los comandos que se muestran enseguida.

- Configuración de un puerto como enlace troncal fijo.
  Switch(config-if)# **switchport mode trunk**
- Configuración de un puerto como enlace de acceso.
  Switch(config-if)# **switchport mode access**
- Deshabilitar el protocolo DTP.
  Switch(config-if)# **switchport nonegotiate**

A partir del análisis previo, se concluye que es necesario configurar el enlace como troncal fijo o acceso y posteriormente deshabilitar la funcionalidad DTP para eliminar cualquier duda sobre su funcionamiento. Además, es conveniente adicionar en los puertos de acceso la configuración de autenticación basada en puertos IEEE 802.1X expuesta en el capítulo 3.1.2.2 para aumentar el nivel de seguridad. Finalmente, es una buena práctica de seguridad deshabilitar lógicamente los puertos que no están siendo utilizados, de esta forma se evita el uso no deseado de puertos que no han sido configurados con opciones de seguridad o que no se han dispuesto para ser usados en la red.

3.1.2.3.2. Salto de VLAN (VLAN Hopping)

Al proteger los enlaces troncales, también se debe considerar la amenaza denominada salto de VLAN. Aquí, un atacante ubicado en una VLAN de acceso puede crear y enviar tramas con etiquetas 802.1Q falsificadas para que las cargas útiles del paquete aparezcan finalmente en una VLAN totalmente diferente. Esta amenaza requiere que el atacante esté conectado a un puerto de conmutador en modo de acceso y que el enlace troncal tenga la VLAN de acceso del atacante como su VLAN nativa [6].

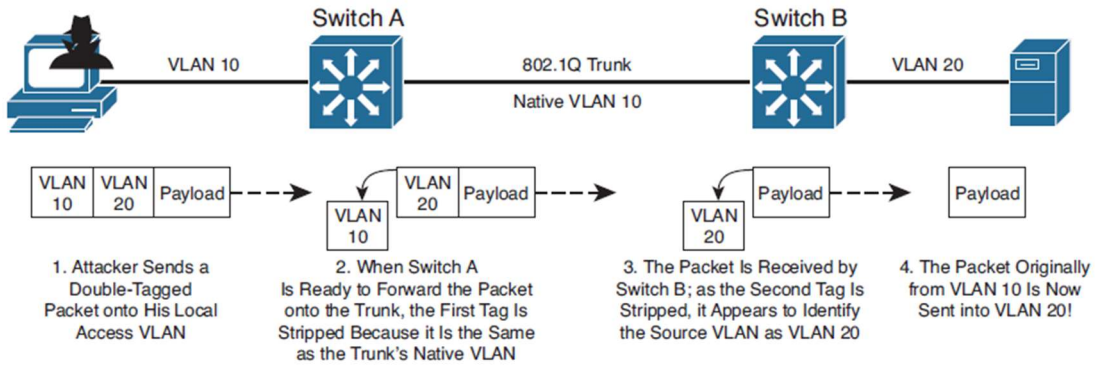

Figura 15. Proceso de ataque de salto de VLAN [6].

Como se observa, el atacante envía una trama con dos etiquetas de VLAN concatenadas, al llegar al puerto del conmutador, este identifica que se trata de la VLAN nativa y reenvía la trama sin encapsularla con información de etiquetado. De esta forma, el conmutador en el otro extremo del enlace troncal recibe la trama con la VLAN objetivo del atacante y la reenvía a las estaciones de usuario de ese segmento de red.

Dado que la clave de este tipo de ataque gira en torno al uso de VLAN nativas sin etiquetar, es necesario asociar cada puerto de acceso no utilizado con una VLAN falsa o aislada. Además, se debe configurar la VLAN nativa del enlace troncal con un ID falso o no utilizado, eliminando su uso en ambos extremos. Los comandos de configuración se muestran a continuación [6].

- Para enlaces en modo de acceso:
  Switch(config-if)# **switchport access vlan** *black-hole-vlan*
- Para enlaces troncales:
  Switch(config-if)# **switchport trunk native vlan** *black-hole-vlan*
  Switch(config-if)# **switchport trunk allowed vlan remove** *black-hole-vlan*

Una alternativa al procedimiento anterior es obligar a todos los enlaces troncales 802.1Q a que agreguen etiquetas a las tramas para la VLAN nativa. El ataque de salto de VLAN de doble etiqueta no funciona porque el conmutador no elimina la primera etiqueta con la ID de VLAN nativa [6].
Switch(config)# **vlan dot1q tag native**

Analizando las dos opciones de configuración previamente descritas que se ejecutan para prevenir los ataques de salto de VLAN, se concluye que la primera opción, en la cual la VLAN nativa no se transmite a través del enlace troncal, solamente es funcional si todos los dispositivos son compatibles con IEEE 802.1Q. Por otra parte, la segunda opción permite que los dispositivos que no soportan IEEE 802.1Q pueda acceder y comunicarse en la red, al mismo tiempo que se evitan los ataques de salto de VLAN. Hoy en día la mayoría de los dispositivos soportan IEEE 802.1Q, motivo por el cual generalmente se configura la primera opción.

3.1.2.4. Prevención de ataques de suplantación de identidad

3.1.2.4.1. Indagación DHCP (DHCP Snooping)

Un atacante puede introducir un servidor DHCP deshonesto con el objetivo de enviar respuestas que sustituyan la puerta de enlace predeterminada con su propia dirección IP. Cuando una dispositivo final de usuario envía paquetes destinados a direcciones fuera de la red local, van primero a la máquina del atacante donde son examinados y luego enviados al destino correcto para evitar sospechas [6].

Los conmutadores de Cisco usan la función de indagación DHCP para ayudar a mitigar este tipo de ataque. Cuando está habilitada, los puertos del conmutador se clasifican como confiables o no confiables. Los servidores DHCP legítimos se encuentran en puertos confiables, mientras que todos los demás dispositivos se encuentran detrás de puertos no confiables [6].

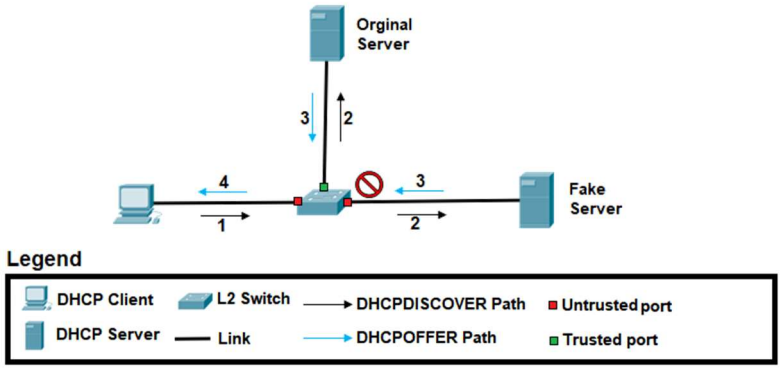

Figura 16. Indagación DHCP [18].

Cualquier respuesta DHCP que provenga de un puerto no confiable se descarta y el puerto se apaga cambiando su estado a deshabilitado por error. Además, la indagación de DHCP crea una base de datos que contiene la dirección MAC del cliente, la dirección IP ofrecida, el tiempo de concesión, etc. El proceso de configuración de la indagación DHCP es el siguiente [6].

- Habilitar la indagación DHCP.
  Switch(config)# **ip dhcp snooping**
- Identificar las VLAN en las que se implementa la indagación DHCP.
  Switch(config)# **ip dhcp snooping vlan** *vlan-id [vlan-id]*
- Identificar los puertos donde se encuentran los servidores DHCP confiables. De forma predeterminada, se desconfía de todos los puertos del conmutador
  Switch(config-if)# **ip dhcp snooping trust**
- Limitar la tasa de solicitudes DHCP de los puertos en los que se desconfía. Por defecto, aceptan una tasa ilimitada.
  Switch(config-if)# **ip dhcp snooping limit rate** *rate*

Teniendo en cuenta lo anteriormente expuesto, se concluye que para evitar el ataque de suplantación del servidor DHCP, es necesario realizar la configuración para todas las VLAN en uso, de modo que se proteja la red en su totalidad. Además, es necesario clasificar los puertos confiables como aquellos detrás de los cuales se encuentran los servidores DHCP en los que se confía. Finalmente, se recomienda limitar el número de solicitudes DHCP por segundo que un puerto no confiable admite, realizando un análisis previo para determinar el valor correcto del parámetro *rate*, ya que, si se configura un número muy bajo, es probable que algunas partes de la red no reciban direccionamiento IP.

3.1.2.4.2. Protección de IP de origen (IP Source Guard)

Una PC deshonesta puede suplantar su dirección IP al falsificarla o tomarla de otro dispositivo. Las direcciones falsas se utilizan a menudo para disfrazar el origen de los ataques de denegación de servicio. Si la dirección de origen no existe, ningún tráfico de retorno encontrará su camino de regreso al origen [6].

Los conmutadores de Cisco usan la función de protección de IP de origen para detectar y suprimir los ataques de suplantación de direcciones, incluso si ocurren dentro de la misma VLAN. Esta funcionalidad usa la base de datos de indagación DHCP en conjunto con entradas originadas estáticamente. Los paquetes que llegan a un puerto de conmutador habilitado con la protección IP de origen se someten a las siguientes condiciones de verificación [6].

- La dirección IP de origen es idéntica a la dirección IP aprendida por la indagación DHCP o una entrada estática: el conmutador crea automáticamente una ACL dinámica para filtrar el tráfico en el puerto, agrega la dirección IP de origen aprendida a la ACL y aplica la ACL a la interfaz donde se aprende la dirección.
- La dirección MAC de origen es idéntica a la dirección MAC aprendida en el puerto del conmutador y mediante la indagación de DHCP: la seguridad del puerto se utiliza para filtrar el tráfico.
- Si la dirección MAC o IP son diferentes a la aprendida o configurada estáticamente, el conmutador descarta el paquete.

Para configurar la protección IP de origen, primero se debe configurar la inspección DHCP, como se presentó en el capítulo 3.1.2.4.1. Además, para detectar las direcciones MAC falsificadas, es necesario habilitar la seguridad del puerto como se mostró en el capítulo 3.1.2.1. La protección de IP de origen del conmutador se habilita con el siguiente comando de configuración [6].
Switch(config-if)# **ip verify source** [**port-security**]

En síntesis, con el objetivo de evitar efectivamente la suplantación de direcciones IP, es necesario configurar la inspección DHCP y la seguridad del puerto. Además, se requiere emplear el comando **ip verify source** en conjunto con la opción **port-security**, puesto que la primera inspecciona la dirección IP de origen mientras que la segunda habilita la inspección de la dirección MAC de origen, sometiendo la trama a una verificación de seguridad más fuerte. Finalmente, se debe configurar una unión estática de dirección MAC, VLAN, dirección IP e interfaz del conmutador para los dispositivos que no emplean el protocolo DHCP.

3.1.2.4.3. Inspección dinámica de ARP (DAI)

Como se expuso en el capítulo 1.2, los conmutadores usan el protocolo de resolución de direcciones (ARP) para resolver una dirección MAC desconocida a partir de una dirección IP conocida. El dispositivo origen transmite una solicitud ARP y el dispositivo que posee la dirección IP de destino replica con una respuesta ARP que contiene su dirección MAC [6].

Si un atacante suplanta respuestas ARP con su propia dirección MAC, el dispositivo origen agrega la dirección MAC falsificada a su caché ARP y reenvía los paquetes al adversario. De este modo se realiza un ataque de hombre en el medio, en el que se interceptan y reenvían las tramas, después de examinar su contenido [6].

Los conmutadores Cisco utilizan la función de inspección dinámica de ARP (DAI) para mitigar este ataque. Al usar DAI los puertos del conmutador se clasifican como confiables o no confiables. Cuando se recibe una respuesta ARP en un puerto no confiable, el conmutador verifica la dirección MAC e IP contra valores configurados estáticamente o aprendidos dinámicamente empleando la indagación DHCP. Si la respuesta ARP contiene valores no válidos, se descarta y se genera un mensaje de registro. El proceso de configuración se muestra enseguida [6].

- Habilitar la DAI en una o más VLAN.
  Switch(config)# **ip arp inspection vlan** *vlan-range*
- Identificar los puertos confiables. De forma predeterminada, todos los puertos de conmutador se consideran como no confiables.
  Switch(config-if)# **ip arp inspection trust**
- Configurar una lista de acceso ARP, que defina las uniones estáticas de direcciones MAC y direcciones IP permitidas.
  Switch(config)# **arp access-list** *acl-name*
  Switch(config-acl)# **permit ip host** *sender-ip* **mac host** *sender-mac*
- Aplicar la lista de acceso ARP a la DAI.
  Switch(config)# **ip arp inspection filter** *acl-name* **vlan** *vlan-range*

Cuando se interceptan las respuestas ARP, su contenido se compara primero con las entradas de la lista de acceso y luego contra la base de datos de indagación DHCP. Si no se encuentra ninguna coincidencia, la respuesta ARP se considera inválida [6].

De forma predeterminada, solo se validan las direcciones MAC e IP contenidas dentro de la respuesta ARP. En consecuencia, no se evalúan las direcciones MAC e IP del encabezado Ethernet de la respuesta ARP. Esta validación permite determinar si lo contenido en la respuesta ARP realmente corresponde a los valores de la trama Ethernet y se habilita con el siguiente comando de configuración [6].

Switch(config)# **ip arp inspection validate** {[**src-mac**] [**dst-mac**] [**ip**]}

- ➔ **src-mac**: comprueba la dirección MAC de origen del encabezado Ethernet, contra la dirección MAC del remitente contenida en la respuesta ARP.
- ➔ **dst-mac**: verifica la dirección MAC de destino del encabezado Ethernet, con la dirección MAC objetivo en la respuesta ARP.
- ➔ **ip**: valida la dirección IP del remitente contra la dirección IP de destino en la respuesta ARP.

En conclusión, para evitar eficazmente la suplantación de respuestas ARP, se requiere habilitar la funcionalidad DAI en todas las VLAN que se están empleando. Así mismo, se deben identificar los puertos confiables, por ejemplo, aquellos que se emplean para conectarse a otros conmutadores. Además, es necesario configurar la lista de acceso que define las uniones estáticas de direcciones MAC y direcciones IP de los dispositivos que no utilizan el protocolo DHCP. Del mismo modo, se requiere habilitar la funcionalidad de inspección DHCP según lo expuesto en el capítulo 3.1.2.4.1 para permitir que las entradas dinámicas sean analizadas. Finalmente, es necesario habilitar la validación de las direcciones MAC e IP contenidas dentro de la respuesta ARP contra las direcciones MAC e IP del encabezado Ethernet para determinar su validez y robustecer el nivel de seguridad.

3.2. Aseguramiento del plano de control.

El plano de control se ocupa de cualquier acción que controla el plano de datos. La mayoría de estas acciones tienen que ver con la creación de las tablas utilizadas por el plano de datos, tales como la tabla de enrutamiento IP y tabla de direcciones MAC. Las funciones principales de un conmutador en el plano de control son crear una tabla de reenvío de capa 2, reenviar el tráfico en función de esa tabla y prevenir la formación de bucles empleando el protocolo de árbol de expansión (STP). La carencia de salvaguardas en este plano permite a un atacante hacerse con el control del flujo de las tramas [16]. Por su parte, en los enrutadores el plano de control se centra en la creación de la tabla de enrutamiento IP empleando rutas estáticas o a través de protocolos de enrutamiento dinámico. La falta de protección de este plano permite a un a un usuario malintencionado hacerse con el control de las decisiones de reenvío de los paquetes al aprender, inyectar y modificar las rutas de una organización [16].

3.2.1. Seguridad del plano de control en conmutadores

3.2.1.1. Protección del árbol de expansión (STP)

En una red, la implementación de rutas alternativas es necesaria para proporcionar redundancia y ofrecer mayor fiabilidad. Sin embargo, a nivel de capa 2 del modelo OSI, sin la presencia de un protocolo que evite la formación de bucles, las tramas se reenvían indefinidamente, ya que no cuentan con un campo de tiempo de vida (TTL). En consecuencia, se consume tanto el ancho de banda de la red como la CPU de los dispositivos, degradando su funcionamiento hasta hacerla inutilizable. STP permite la convergencia de una red de conmutadores hacia un árbol de expansión sin bucles, al calcular una ruta libre de bucles, en la que se activan los enlaces redundantes en caso de falla. De forma predeterminada, STP está habilitado para todas las VLAN activas y en todos los puertos de un conmutador [19].

Existen múltiples variantes del protocolo STP original estandarizado en IEEE 802.1D, que se han desarrollado para minimizar su tiempo de convergencia. El protocolo de árbol de expansión rápido (RSTP) definido en IEEE 802.1w converge y se recupere de fallas en un tiempo muy inferior al ofrecido por STP. Por su parte, el protocolo de árbol de expansión múltiple (MSTP), especificado en el estándar IEEE 802.1s, permite configurar diferentes instancias de STP, que habilitan el balanceo de carga entre grupos de VLAN [6].

Una red que ejecuta STP utiliza unidades de datos de protocolo de puente (BPDU) para la comunicación entre conmutadores. Cada BPDU contiene la dirección MAC y prioridad de un conmutador, que en conjunto definen un ID de puente. Empleando el ID de puente, los conmutadores eligen un conmutador que sirve como raíz o referencia del árbol de expansión. La ubicación del puente raíz debe ser predecible y se debe proteger. Para evitar que un conmutador no autorizado se conecte a la red y sea capaz de convertirse en el puente raíz, Cisco agregó las funciones protección de puente raíz y protección de BPDU [6].

3.2.1.1.1. Protección del puente raíz (Root Guard).

Si se introduce un conmutador en la red, con un ID de puente más deseable que el del puente raíz actual, este toma el rol de puente raíz y el árbol de expansión STP converge hacia una nueva topología que puede ser totalmente inaceptable. Además, mientras la topología converge, es posible que la red deje de estar disponible durante algunos segundos [6].

La protección de puente raíz permite que mientras se reciben BPDU con un ID de puente más deseable en un puerto, este se mantenga en estado inconsistente, previniendo el envío y la recepción de datos. De esta forma, se evita la elección de un conmutador no previsto como puente raíz. Para habilitar esta funcionalidad se debe ejecutar el siguiente comando [6].
Switch(config-if)# **spanning-tree guard root**

En resumen, empleando la protección del puente raíz se mantiene la consistencia y eficiencia en la topología sin bucles del árbol de expansión STP. Además, se previene la convergencia hacia una topología con bajo rendimiento y una posible indisponibilidad en la red. Es necesario tener en cuenta que la protección del puente raíz se debe emplear en los puertos de conmutador donde nunca se espera recibir BDPU del puente raíz para cualquier VLAN, ya que el puerto en su totalidad entra en estado inconsistente.

3.2.1.1.2. Protección contra BPDU inesperadas (BPDU Guard)

Los puertos de conmutador con conexiones hacia dispositivos de usuario final, en los que nunca se espera la formación de bucles de capa 2, se configuran generalmente con la funcionalidad de puerto rápido (*Port fast)*. Esta funcionalidad permite un acceso rápido a la red y disminuye el tiempo de convergencia de STP. Sin embargo, si se conecta un conmutador en lugar de un dispositivo de usuario final al puerto, se puede generar un bucle y existe la posibilidad de que se anuncie como el nuevo puente raíz [6].

La protección contra BPDU inesperadas, evita que los puertos de usuario final acepten BPDU, al colocarlos inmediatamente en una condición de error en la que se apagan lógicamente. Esto impide cualquier posibilidad de que se conecte un conmutador al puerto, ya sea intencionalmente o por error. Por defecto, la protección contra BPDUs inesperadas se encuentra deshabilitada en todos los puertos del conmutador, el proceso de configuración se muestra enseguida [6].

- Para habilitar la protección BPDU en todos los puertos de usuario final.
  Switch(config)# **spanning-tree portfast bpduguard default**
- Para habilitar la protección BPDU en un puerto en particular.
  Switch(config-if)# **spanning-tree bpduguard enable**

En síntesis, la protección contra BPDU inesperadas permite asegurar la integridad de los puertos de conmutador que tienen la funcionalidad de puerto rápido habilitada. Se recomienda la configuración de esta protección en todos los puertos de usuario final empleando el comando de configuración **spanning-tree portfast bpduguard default**. Además, es necesario tener en cuenta que al detectar una BPDU, el puerto se apaga lógicamente en una condición de error y debe reactivarse manualmente o recuperarse automáticamente. Finalmente, nunca se debe habilitar la protección en puertos ascendentes del conmutador, donde es posible recibir BPDU legítimas, ya que esto impediría que el puerto pueda ser utilizado.

3.2.2. Seguridad del plano de control en enrutadores

3.2.2.1. Seguridad en protocolos de enrutamiento

El plano de control de los enrutadores se ocupa de tomar decisiones de reenvío de paquetes empleando rutas estáticas, protocolos de puerta de enlace interior (IGP) y protocolos de puerta de enlace exterior (EGP). Estos protocolos deben ser puestos a punto para mitigar el riesgo que implica realizar configuraciones básicas, las cuales permiten la formación de vecinos ilegítimos que pueden aprender, inyectar y manipular las rutas en una organización [4].

3.2.2.1.1. Categorías de los protocolos de enrutamiento

Los protocolos de enrutamiento se pueden categorizar en vector distancia, estado de enlace y vector camino. Los protocolos vector distancia y estado de enlace son utilizados como protocolo de puerta de enlace interior (IGP), es decir son utilizaos dentro del mismo sistema autónomo. Por otra parte, los protocolos vector camino son utilizados como protocolo de puerta de enlace exterior (EGP), siendo usados entre sistemas autónomos. Por definición, un sistema autónomo (AS) es un grupo de redes IP que poseen una política de rutas propia e independiente, realizando su propia gestión del tráfico [4].

A continuación, se realiza el análisis de seguridad para los protocolos OSPF y BGP, los cuales son los más utilizados actualmente en las redes en producción.

3.2.2.1.2. Seguridad en OSPF

El protocolo OSPF sigue tres pasos generales para agregar rutas a la tabla de enrutamiento IP [4].

1. **Descubrimiento de vecinos:** los enrutadores OSPF envían mensajes de saludo para descubrir vecinos. La información sobre las vecindades es almacenada en la tabla de vecinos.
2. **Intercambio de la base de datos de topología**: cada enrutador OSPF envía mensajes de anuncio de estado de enlace (LSAs) para dar a conocer la información de la topología de red. Los enrutadores almacenan esta información en su base de datos de estado de enlace (LSDB).
3. **Métrica OSPF y elección de rutas**: cada enrutador OSPF analiza de forma independiente los datos de la LSDB y elige las mejores rutas. En particular, OSPF utiliza el algoritmo de camino más corto primero (SPF) para analizar los datos, elegir la mejor ruta a cada red de destino y agregar la información de interfaz de siguiente salto en la tabla de enrutamiento IP.

3.2.2.1.2.1. Descubrimiento de vecinos

OSPF envía mensajes de saludo a través de las de interfaces del enrutador con el objetivo de descubrir vecinos cuando se cumplen dos requisitos [4].

- Se ha habilitado OSPF en la interfaz.
  Router(config-router)# **network** *ip-address wild-card* **area** *area-id*
- La interfaz no ha sido referenciada como pasiva.
  Router(config-router)# **passive-interface** *type number*

Para comprobar que dos enrutadores pueden ser vecinos, OSPF a través de sus paquetes de saludo verifica los siguientes parámetros [4].

- Los enrutadores pueden enviar/recibir paquetes IP entre sí.
- Las direcciones IP de las interfaces están en la misma red.
- Las interfaces conectadas no son pasivas.
- Las interfaces están en la misma área OSPF.
- El intervalo de saludo y el temporizador de tiempo muerto coinciden.
- Los ID entre enrutadores son únicos.
- Si está configurada, la autenticación debe ser exitosa.

3.2.2.1.2.2. Intercambio de la base de datos de topología

Los enrutadores OSPF pasan por diferentes estados, en cada uno de los cuales se transmiten mensajes que permiten a cada enrutador conocer la topología de la red y crear una LSDB. Al final de este proceso, ambos enrutadores tienen una LSDB idéntica y se consideran vecinos adyacentes. En ese punto, los dos enrutadores pueden ejecutar el algoritmo SPF para elegir la mejor ruta hacia cada red de destino [4].

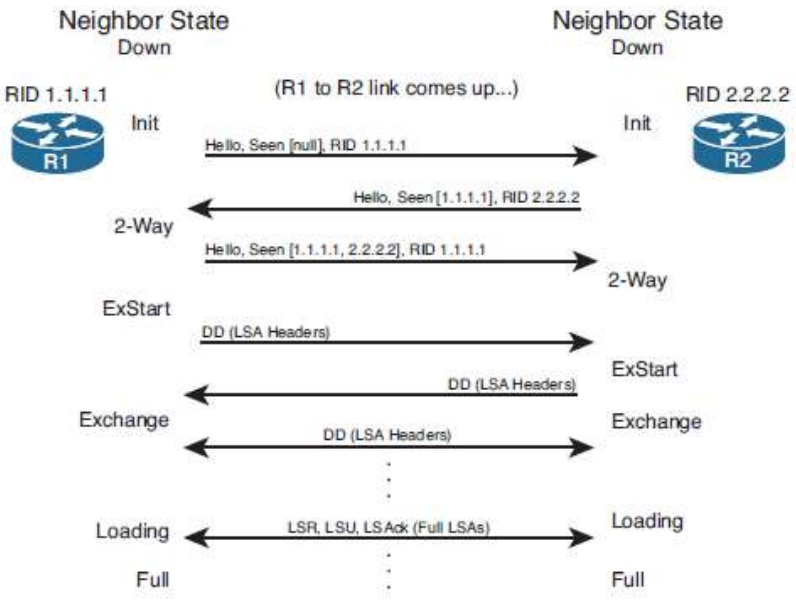

Figura 17. Proceso de intercambio de la LSDB entre dos vecinos adyacentes [4].

3.2.2.1.2.3. Métrica OSPF y elección de rutas

El proceso de elección de la mejor ruta hacia una red de destino por parte de un enrutador OSPF pasa por dos etapas. Primero, cada enrutador OSPF analiza la LSDB y determina todas las rutas que permiten alcanzar la red. Segundo, agrega el costo de las interfaces a lo largo de esa ruta. OSPF calcula el costo de una interfaz empleando la siguiente fórmula [4].

$$Costo = \frac{BW_{Reference}}{BW_{Interface}}$$

El ancho de banda de referencia posee por defecto un valor de 100 Mbps. Por otra parte, los valores de ancho de banda de algunas interfaces se muestran a continuación [4].

Tipo de interfaz	Ancho de banda (kbps)
Serial	1.544
Giga Ethernet	1.000.000
Fast Ethernet	100.000
Ethernet	10.000

Tabla 3.Relación entre tipo de interfaz y ancho de banda [4].

OSPF suma los valores de los costos de cada interfaz y elige la que posee el menor costo total. Esta ruta se agregue a la tabla de enrutamiento IP [4].

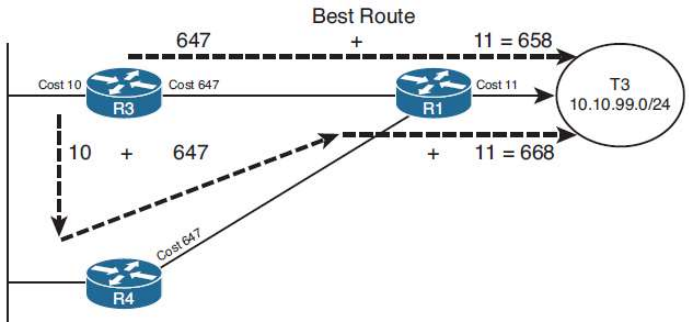

Figura 18. Cálculo del costo OSPF [4].

3.2.2.1.2.4. Creación de vecinos no deseados

De manera predeterminada, un enrutador OSPF envía mensajes de saludo sin cifrar a través de las interfaces referencias con el comando **network**. Un atacante puede realizar un análisis de tráfico y detectar los paquetes OSPF que transitan en la red. Teniendo en cuenta los requerimientos para la formación de vecinos descritos en el capítulo 3.2.2.1.2.1, el atacante está en la capacidad de formar una relación de vecino. Una vez establecida la relación de vecino, el atacante puede modificar y anunciar nuevas rutas, con el objetivo de hacerse con el control del envío de paquetes [4].

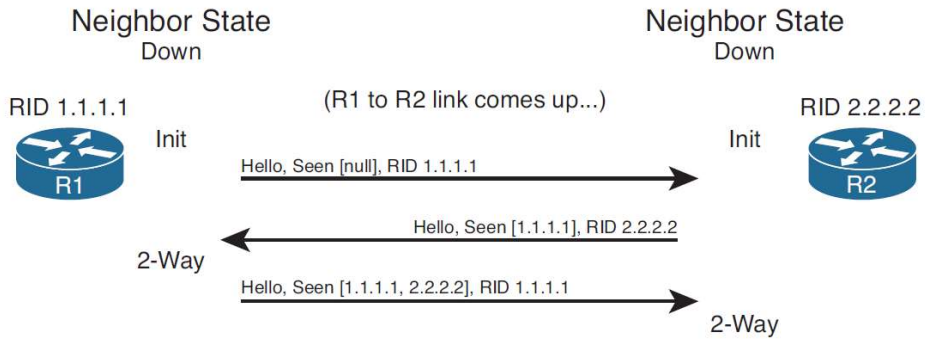

Figura 19.Envío de mensajes de saludo entre enrutadores OSPF [4] .

3.2.2.1.2.5. Prevención de vecinos no deseados mediante interfaces pasivas

Debido a la formación de vecinos no deseados, se puede elegir limitar OSPF a aquellas interfaces que tienen vecinos legítimos. Sin embargo, las subredes que no están conectadas a vecinos legítimos frecuentemente requieren ser anunciadas para ser accedidas. La funcionalidad de interfaz pasiva permite anunciar la red y deshabilitar el envío de mensajes de saludo OSPF, de este manera se evita la formación de vecinos no deseados. El comando de configuración que debe ser ejecutado se muestra a continuación [4].

Router(config-router)# **passive-interface** *type number*

3.2.2.1.2.6. Prevención de vecinos no deseados mediante autenticación

La autenticación de vecinos es una forma adicional de prevenir y mejorar la seguridad de OSPF, ya que permite intercambiar información de actualización de enrutamiento de manera segura. El proceso de configuración se describe enseguida [20].

- Crear un llavero de claves.
  R1(config)#**key chain** *name*
- Establecer una identificación de la clave. La identificación debe ser la misma entre dos enrutados con vecindad legítima.
  R1(config-keychain)# **key** *key-id*
- Seleccionar el algoritmo criptográfico de autenticación.
  R1(config-keychain-key)# **cryptographic-algorithm** [**md5** | **hmac-sha-1** | **hmac-sha-256** | **hmac-sha-384** | **hmac-sha-512**]
- Determinar los periodos de envío y aceptación de la clave, que determina su validez.
  R1(config-keychain-key)# **accept-lifetime** *hh:mm:ss month-start day-start year-start hh:mm:ss month-stop day-stop year-stop*
  R1(config-keychain-key)# **send-lifetime** *hh:mm:ss month-start day-start year-start hh:mm:ss month-stop day-stop year-stop*
- Configurar la contraseña que se va a utilizar y que corresponde a la clave precompartida.
  R1(config-keychain-key)# **key-string** *password*
- Habilitar la autenticación en la interfaz que conecta los enrutadores vecinos.
  R1(config-if)# **ip ospf authentication key-chain** *name*

Como práctica recomendada de seguridad se deben crear varias claves dentro del llavero de claves, cada una con un tiempo de aceptación y envío diferente. Los periodos de tiempo durante los cuales cambian las claves deben ser acordes con las políticas de seguridad de la organización.

Adicionalmente, como se expone en los capítulos 2.4 y 2.5, es necesario evitar la configuración del algoritmo HMAC con hashes MD5 y SHA-1, ya que estos han sido comprometidos a través de ataques de colisiones, seleccionando en su lugar como algoritmo de reducción SHA-3. Sin embargo, dado que el IOS de Cisco solamente soporta algoritmos SHA-2 se requiere configurar la opción **hmac-sha-512** para brindar el mayor grado de seguridad.

Por otra parte, se debe tener en cuenta que el IOS de Cisco posee una vulnerabilidad en la seguridad del almacenamiento de la clave de autenticación precompartida, ya que emplea el algoritmo de Vigenère. Como se describe en el capítulo 2.2.1 este algoritmo es fácilmente cripto analizable, permitiendo a un atacante que accede a la configuración del dispositivo, descubrir la clave fácilmente. Así pues, se requiere emplear servidores de autenticación, autorización y contabilización expuestos en el capítulo 3.3.2 en conjunto con protocolos seguros de conexión remota detallados en el capítulo 3.3.3, con el fin de proteger el acceso a la configuración del dispositivo.

En conclusión, para asegurar el protocolo de enrutamiento OSPF en todos los enrutadores, se debe emplear la funcionalidad de interfaces pasivas, que permite anunciar una red y deshabilitar el envío de mensajes de saludo OSPF a través de la interfaz referenciada si esta no posee vecinos confiables, en conjunto con la autenticación de vecinos OSPF, que habilita el intercambio de información de actualización de enrutamiento de manera segura.

3.2.2.1.3. Seguridad en BGP

BGP es el protocolo mediante el cual se intercambia información de enrutamiento dentro de Internet global. BGP forma una relación entre vecinos antes de enviar información de enrutamiento empleando una conexión TCP (puerto 179) y mensajes de unidifusión que requieren configurar explícitamente la dirección IP del vecino. Lo anterior se debe a que los vecinos pueden no estar en una red en común. Además, BGP emplea una variedad de atributos de ruta (PA) que, en conjunto con el algoritmo de vector camino, permiten influir con gran flexibilidad en la elección de las mejores rutas [4].

3.2.2.1.3.1. Creación de vecinos BGP

BGP define dos clases de vecinos, internos (iBGP) que se encuentran dentro del mismo AS y externos (eBGP) que pertenecen a diferentes AS. Los requisitos para formar vecindades BGP son los siguientes [4].

- Configurar explícita los números de sistema autónomo (ASN) y las direcciones IP de cada uno de los vecinos empleando el comando: Router(config-router)# **neighbor** *neighbor-ip* **remote-as** *asn*
- Los ID de enrutador BGP entre vecinos deben ser diferentes.
- Si está configurada, la autenticación debe ser exitosa.
- Cada enrutador debe ser parte de una conexión TCP con el otro enrutador.

3.2.2.1.3.2. Intercambio de la tabla BGP

Similar a OSPF, la relación de vecindad BGP pasa por una serie de estados a lo largo del tiempo. El proceso general funciona de la siguiente manera [4].

1. Un enrutador intenta establecer una conexión TCP con la dirección IP que figura en el comando **neighbor**, utilizando el puerto de destino 179.
2. Luego de establecer la conexión TCP, el enrutador envía su primer mensaje BGP (*Open*) que contiene varios parámetros BGP, incluidos aquellos que deben verificarse antes de permitir que los enrutadores se conviertan en vecinos.
3. Después de que se haya enviado y recibido el mensaje *Open* y de verificar que los parámetros entre vecinos coincidan, se forma la relación de vecindad y se alcanza el estado establecido. En este estado se envían los mensajes de actualización, que enumeran los atributos de ruta (PA) y los prefijos.

A continuación, se ilustran los diferentes estados y mensajes que se transmiten para permitir el intercambio de la tabla BGP [21].

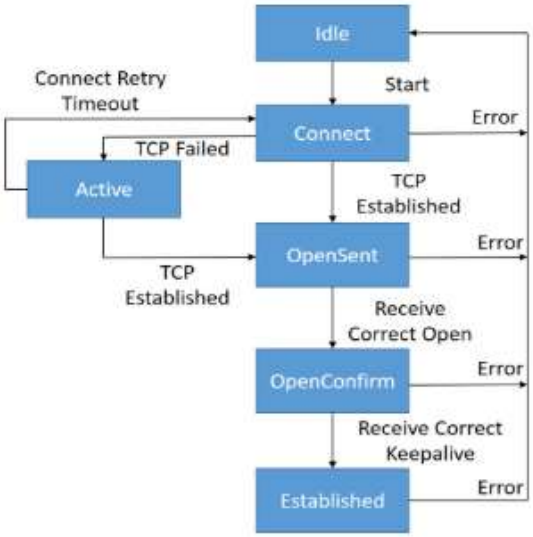

Figura 20. Máquina de estados BGP [21].

3.2.2.1.3.3. Elección de rutas en BGP

Cuando todos los PA se encuentran en sus valores por defecto, la elección de la mejor ruta recae en la longitud del camino del sistema autónomo (*AS_Path length*). Este atributo enumera los AS en la ruta de extremo a extremo y realiza dos funciones clave [4].

- Prevención de bucles: Cuando un enrutador eBGP recibe una actualización, si su propio ASN ya está en el *AS_Path* recibido, esa ruta ya se ha anunciado en el ASN local y debe ignorarse.
- Cálculo de la longitud de *AS_Path*. La mejor ruta para un prefijo dado es elegida basándose en el AS_PATH más corto.

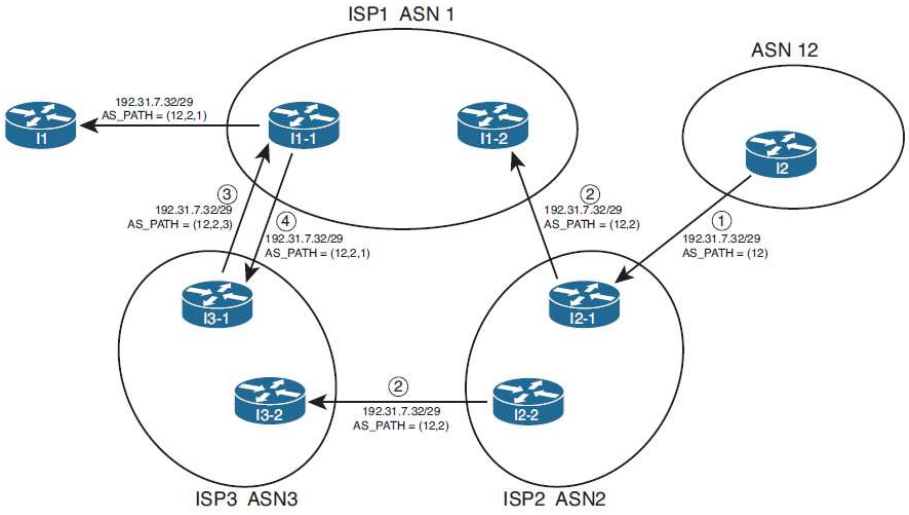

Figura 21. Funcionamiento atributo BGP AS_Path [4].

3.2.2.1.3.4. Autenticación de vecinos

Dado que BGP requiere para establecer una vecindad que cada uno de los pares sea referenciado por su vecino, indicando tanto su sistema autónomo como su dirección IP para crear la conexión TCP, no es posible establecer una relación de vecinos no deseados. Sin embargo, en el escenario en el cual dos enrutadores han establecido una sesión TCP, un atacante puede secuestrar esa sesión existente y corromper la tabla BGP [4].

Figura 22. Secuestro de la sesión TCP [22].

En el plano de control, BGP utiliza el algoritmo MD5 como mecanismo de autenticación. Cuando la autenticación está habilitada, cualquier segmento TCP que pertenece a BGP se intercambia entre pares, se verifica y luego se acepta solo si la autenticación es exitosa. La autenticación a través de MD5 se realiza empleando el comando **neighbor** *neighbor-ip* **password** *key* [4]. Sin embargo, como ya se ha mencionado previamente, este algoritmo de reducción es inseguro.

La aplicación de BGP usa la API de TCP para configurar un llavero (*key chain*) en una conexión TCP que puede ser asociado con una opción de autenticación TCP (TCP-AO) con un vecino. Si BGP está configurado con la clave TCP MD5, no permite configurar TCP-AO y viceversa. Además, la sesión BGP no se establece cuando un dispositivo está configurado con la opción TCP MD5 y su par con la opción TCP-AO. Existen dos opciones para configurar BGP [23].

- **include-tcp-options**: especifica que los encabezados de la opción TCP (diferentes a la opción TCP-AO) se incluirán al calcular el resumen MAC de los paquetes.
- **accept-ao-mismatch-connections**: determina que se acepta una conexión sin TCP-AO cuando el par no contiene esta opción.

La opción de autenticación TCP (TCP-AO) permite emplear para la autenticación el algoritmo criptográfico HMAC-SHA-256 que aún se considera seguro. El proceso de configuración de TCP-AO junto con BGP es el siguiente [23].

- Crear un llavero TCP-AO con un nombre determinado.
  Router(config)# **key chain** *key-chain-name* **tcp**
- Crear una clave con un ID específico.
  Router(config-keychain-tcp)# **key** *key-id*
- Especificar el identificador de envío de la clave.
  Router(config-keychain-tcp-key)# **send-id** *send-id*
- Especificar el identificador de recepción de la clave.
  Router(config-keychain-tcp-key)# **recv-id** *recv-id*
- Seleccionar el algoritmo a utilizar para calcular el MAC de los segmentos TCP.
  Router(config-keychain-tcp-key)# **cryptographic-algorithm** {**aes-128-cmac** | **hmac-sha-1** | **hmac-sha-256**}
- (Opcional) Indicar si se deben utilizar opciones de TCP distintas de TCP-AO para calcular el MAC.
  Router(config-keychain-tcp-key)# **include-tcp-options**
- Especificar el tiempo durante el cual el envío de la clave es válido.
  Router(config-keychain-tcp-key)# **send-lifetime** [**local**] *start-time {***infinite** | *end-time* | **duration** *seconds}*
- Especificar el tiempo durante el cual la clave es aceptada.
  Router(config-keychain-tcp-key)# **accept-lifetime** [**local**] *start-time {***infinite** | *end-time* | **duration** *seconds}*
- Especificar la clave maestra para derivar claves de tráfico. Las claves maestras deben ser idénticas en ambos pares.
  Router(config-keychain-tcp-key)# **key-string** *master-key*

- (Opcional) Indicar si el receptor debe aceptar segmentos para los cuales el MAC en el TCP-AO entrante, no coincide con el MAC generado en el receptor.
  Router(config-keychain-tcp-key)# **accept-ao-mismatch**
- Configurar un vecino BGP usando TCP-AO:
  Router(config-router)# **neighbor** *neighbor-ip* **ao** *{key-chain-name}* **[include-tcp-options] [accept-ao-mismatch-connections]**

Se debe evitar la configuración de la autenticación empleando el comando **neighbor** *neighbor-ip* **password** *key*, que utiliza MD5 como algoritmo de reducción criptográfico. Por el contrario, es preciso realizar la configuración expuesta en este capítulo para TCP-AO, empleando el algoritmo criptográfico **hmac-sha-256**, el cual es más seguro que **hmac-sha-1** y **aes-128-cmac**.

Además, el parámetro **accept-ao-mismatch,** determina que el receptor debe aceptar segmentos para los cuales el MAC en el TCP-AO entrante, no coincide con el MAC generado. Por lo tanto, se debe prevenir el uso de este parámetro, ya que deshabilita la funcionalidad TCP-AO y la transferencia de claves en las conexiones asociadas. En su lugar, es necesario utilizar el parámetro **include-tcp-options**.

Finalmente, es necesario configurar diferentes claves dentro del llavero de claves, cada una con un tiempo de aceptación y envío diferente. Los periodos de tiempo durante los cuales cambian las claves deben ser acordes con las políticas de seguridad de la organización. TCP-AO elige la clave que tiene la vida útil de envío más larga. Si hay dos claves con la misma duración de envío, se selecciona la primera clave.

En resumen, a pesar de que la formación de vecinos no deseados en BGP es improbable, dado que se requiere que cada uno de los pares sea referenciado por su vecino, es imprescindible autenticarlos y asegurar su conexión TCP, empleando TCP-AO.

3.3. Protección del Plano de Gestión

El plano de gestión es utilizado para acceder y controlar los dispositivos de red, por este motivo, es el candidato principal para los ataques que buscan vulnerar la seguridad [4]. El uso de algoritmos criptográficos poco seguros para el almacenamiento de claves, la implementación de sistemas de autenticación, autorización y registro descentralizados, el acceso remoto a través de protocolos que han sido quebrados y el empleo de protocolos inseguros de administración de red, permiten que los dispositivos sean fácilmente accedidos por parte de un usuario malintencionado.

3.3.1. Protección de contraseñas

Las contraseñas deben ser almacenadas empleando algoritmos seguros, de modo que se pueda evitar el acceso no autorizado. Es una mala idea almacenar las contraseñas en texto sin cifrar, porque si un atacante logra penetrar hasta donde están almacenadas, obtiene acceso a todos los sistemas de la red. Por otra parte, es necesario prevenir el uso de una clave maestra de administración, porque si logra ser obtenida y descifrada por un usuario malintencionado, este obtiene acceso a todos los dispositivos de la red. Finalmente, dado que un hash no es reversible, no hay forma de saber con certeza la contraseña que lo produjo, de modo que, si un atacante obtiene acceso al almacén de contraseñas, no puede comprometer la seguridad de la red [24]. En consecuencia, se debe almacenar el hash de las contraseñas en lugar de la contraseña en sí.

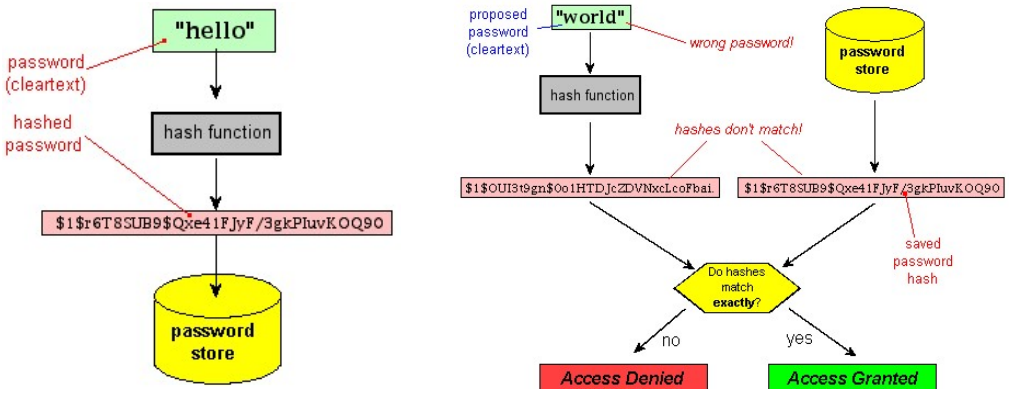

Figura 23. Hashing de contraseñas [24].

Como podemos observar, para verificar que la contraseña ingresada por un usuario es legítima, se le aplica la función de hash y se verifica su coincidencia con el valor almacenado en el almacén de contraseñas. Si los valores coinciden se permite el acceso, de lo contrario el acceso es denegado.

El IOS de Cisco emplea la contraseña de habilitación (*enable password*), la contraseña de línea (*line password*) y la contraseña de nombre de usuario (*username password*) [4]. A continuación, se explica la funcionalidad de cada una de estas contraseñas y su método de almacenamiento.

3.3.1.1. Contraseña de habilitación

Como se mencionó en el capítulo 3, el acceso al modo EXEC privilegiado debe estar protegido con contraseña, ya que un usuario que accede a este modo puede configurar los parámetros operativos del dispositivo. La contraseña que el administrador emplea para prevenir el acceso no autorizado se conoce como contraseña de habilitación (*enable password*). Esta contraseña, se puede cifrar usando los siguientes comandos [4].

- Router(config-router)# **enable password** *password*
- Router(config-router)# **enable secret [4 | 5]** *password*

El comando **enable password** *password*, guarda la contraseña como texto plano en la configuración, pero se puede cifrar empleando el servicio de cifrado de contraseñas del IOS de Cisco con el siguiente comando [4].
Router(config-router)# **service password-encryption**

Este servicio de cifrado utiliza el algoritmo de Vigenère, representado como cifrado tipo 7, el cual como se presentó en el capítulo 2.2.1, es fácilmente vulnerado empleando el test de Kasiski junto con el índice mutuo de coincidencia.

Por otra parte, el comando **enable secret [4 | 5]** *password* guarda la contraseña como un valor hash, empleando los algoritmos criptográficos de reducción MD5 o SHA-256, representados como cifrado tipo 5 y cifrado tipo 4 respectivamente [4]. Debido a que MD5 es inseguro, es preciso utilizar SHA-256, ya que aún es considerado como fiable.

Teniendo en cuenta lo anteriormente expuesto, la configuración de la contraseña de habilitación debe realizarse empleando el comando **enable secret 4** *password*. De esta forma, se garantiza que la contraseña es almacenada eficazmente y que solo los usuarios autorizados pueden acceder al modo EXEC privilegiado donde se configuran los parámetros operativos del dispositivo.

3.3.1.2. Contraseña de línea

La contraseña de línea se utiliza para autenticar a un usuario que intenta iniciar sesión en una de las líneas (VTY, consola o auxiliar) del dispositivo de red. La sintaxis del comando de configuración de la contraseña de línea se muestra enseguida [4].
Router(config-line)# **password** *password*

Al emplear este comando la contraseña aparece en la configuración en ejecución del enrutador en texto sin cifrar. Para cifrar la contraseña, se puede utilizar nuevamente el servicio de cifrado de contraseñas, pero esto solo evita que un observador casual pueda visualizar la contraseña [4].

En consecuencia, para garantizar el acceso al dispositivo, en lugar de emplear contraseñas de línea se requiere configurar combinaciones de nombre de usuario/contraseña y solicitar el ingreso de dichas credenciales antes de acceder a una de las líneas del enrutador, como se describe en el capítulo 3.3.1.3.

3.3.1.3. Contraseña de nombre de usuario

Este método emplea una combinación de nombre de usuario/contraseña, que se almacenan localmente. La base de datos de usuarios se puede configurar empleando los siguientes dos comandos [4].

- Router(config)# **username** *username* **password** [**0** | **7**] *password*
- Router(config)# **username** *username* **privilege** *level* **secret** *password*

En el primer comando, la opción 0 especifica una contraseña almacenada en texto plano, mientras que la opción 7 especifica una contraseña cifrada con el algoritmo de Vigenère. Ambas opciones como se ha expuesto anteriormente son inseguras. Por otra parte, el segundo comando genera un hash SHA-256 de la contraseña [4]. SHA-256 sigue siendo considerado como seguro, por consiguiente, para asegurar la contraseña de nombre de usuario se debe emplear el segundo comando.

El comando **username** *username* **privilege** *level* **secret** *password*, además permite la configuración de diferentes niveles de privilegio. Los enrutadores y conmutadores Cisco poseen 16 niveles de privilegios (0-15). Los tres niveles empleados por defecto se muestran a continuación [25].

- Nivel 0: solo hay unos pocos comandos disponibles, el comando más utilizado es el de habilitación **enable**.
- Nivel 1: este es el nivel de usuario EXEC predeterminado. Puede usar algunos de los comandos **show**, pero no puede ejecutar comandos de configuración.
- Nivel 15: el nivel de privilegio más alto. Puede emplear todos los comandos disponibles en el dispositivo.

Los niveles de privilegios superiores admiten todos los comandos de los niveles de privilegios inferiores. Esto implica que se tienen algunas opciones para asignar comandos a un cierto nivel de privilegio [25].

- Asignar algunos comandos de nivel de privilegio 15 al nivel 1 para que todos los usuarios que pueden iniciar sesión en el dispositivo puedan usarlos.

- Mover algunos comandos del nivel 1 a un nivel superior para evitar el uso de algunos comandos por parte de los usuarios del nivel 1.
- Crear un nuevo nivel de privilegio y asignarle algunos comandos de nivel 15. Esto es una buena opción cuando se trabaja con diferentes grupos de usuarios.

Cuando se asignan los comandos a los diferentes niveles de privilegio, es necesario tener en cuenta los modos soportados por el IOS de Cisco y las convenciones que el software Cisco IOS utiliza para describir los comandos que son ejecutados, amos temas expuestos en el capítulo 3. En seguida, se muestra el proceso de configuración [25].

- Crear un nuevo usuario y asignarle un nivel de privilegios.
  username *username* **privilege** *level* **secret** *password*
- Configurar los comandos que pueden ejecutar los usuarios que perteneces a un nivel de privilegios específico.
- Router(config)# **privilege** *mode* **level** [**0-15**] **command sub-command**

En conclusión, tomando en consideración lo descrito con anterioridad, es necesario utilizar el comando que incluye la funcionalidad **secret** para que la contraseña de nombre de usuario sea almacenada empleando el algoritmo criptográfico de reducción SHA-256, el cual hoy en día sigue siendo considerado como seguro. Además, se recomienda utilizar la funcionalidad **privilege**, ya que permite realizar un proceso de autorización que controla los comandos que pueden utilizar los usuarios. Para esta clasificación de usuarios, niveles de privilegio y comandos, es necesario tener en cuenta el principio de menor privilegio y asignar los permisos de acuerdo con los roles a los que pertenecen los usuarios. De esta forma, se configura de manera efectiva una capa adicional de seguridad.

3.3.2. Autenticación, autorización y contabilización (AAA)

Los servicios AAA permiten tener un repositorio centralizado para las credenciales de usuario. En la base de datos AAA los usuarios pueden agregarse y eliminarse rápidamente sin la necesidad de volver a configurar cada dispositivo, lo que hace que la solución sea más escalable. Un servidor AAA ofrece los siguientes tres servicios [4].

- Autenticación: El acceso a un dispositivo se otorga solo después de que se haya validado la identidad del usuario. En este caso, cuando alguien intenta iniciar sesión de forma remota, debe proporcionar su nombre de usuario y contraseña.
- Autorización: El dispositivo permite el acceso a determinados servicios o comandos según el nivel de privilegios del usuario autenticado. Si el servidor de autorización tiene una entrada para un usuario con un servicio o comando asociado, el dispositivo permite realizar esa tarea.
- Contabilización: El servicio de contabilización recopila y almacena información sobre la sesión de un usuario. Esta información se puede utilizar, por ejemplo, para mantener un seguimiento de auditoría de lo que hizo un usuario en la red.

Los dispositivos de Cisco pueden utilizar los protocolos TACACS+ y RADIUS para comunicarse con los servidores AAA. Tanto TACACS+ como RADIUS utilizan un modelo cliente/servidor. Cuando un usuario intenta conectarse a un dispositivo, este último desafía al usuario para obtener sus credenciales y luego se las pasa servidor AAA. Si el usuario pasa la autenticación, el servidor AAA devuelve un mensaje de aceptación. De lo contrario, se devuelve un mensaje de rechazo. La siguiente tabla describe las características principales de estos dos protocolos [4].

Característica	TACACS+	RADIUS
Protocolo de la capa de transporte.	TCP 49	UDP 1812 y 1813
Modularidad.	Proporciona servicios separados para la autenticación, autorización y contabilización.	Combina las funciones de autenticación y autorización.
Cifrado.	Cifra el paquete entero.	Solamente cifra las contraseñas.
Funcionalidad de contabilización.	Ofrece características básicas de contabilización	Ofrece características robustas de contabilización
Basado en estándar.	No, propietario de Cisco.	Sí.

Tabla 4. Comparación de los protocolos TACAS+ y RADIUS [4].

RADIUS usa UDP mientras que TACACS+ usa TCP. Debido a que TCP ofrece un transporte orientado a la conexión, mientras que UDP ofrece una entrega con el mejor esfuerzo, RADIUS carece del nivel de soporte integrado que ofrece un transporte TACACS+ TCP [26].

Adicionalmente, RADIUS cifra solo la contraseña en el paquete de solicitud de acceso del cliente al servidor. Lo anterior permite que información como nombres de usuario, servicios autorizados y contabilidad, pueda ser capturada por un tercero. TACACS+ en cambio, cifra todo el cuerpo del paquete lo que brinda una comunicación más seguras [26].

Además, los paquetes de autenticación de acceso enviados por el servidor RADIUS al cliente contienen información de autorización, lo que dificulta su desvinculación. Por el contrario, TACACS+ permite soluciones de autenticación, autorización y contabilidad separadas. De esta forma, TACACS+ proporciona un mayor control sobre los comandos que se pueden ejecutar al tiempo que desacopla el mecanismo de autenticación [26].

Así mismo, RADIUS no admite protocolos como el protocolo de acceso remoto AppleTalk (ARA), el protocolo de control de protocolo de tramas NetBIOS y la interfaz de servicios asíncronos de Novell (NASI). En contraste, TACACS+ ofrece soporte multiprotocolo [26].

Igualmente, RADIUS no permite a los usuarios controlar los comandos que pueden ejecutar en un dispositivo. En consecuencia, no es tan útil para la administración de dispositivos de red, ni tan flexible para los servicios de terminal. En contraposición, TACACS+ proporciona métodos para controlar la autorización de los comandos, tanto por usuario como por grupo [26].

Finalmente, aunque RADIUS está basado en un estándar, debido a varias interpretaciones de su RFC, el cumplimiento no garantiza la interoperabilidad. Si los clientes usan solo los atributos RADIUS estándar en sus servidores, pueden interoperar entre varios proveedores siempre que estos proveedores implementen los mismos atributos. Sin embargo, si un cliente utiliza atributos ampliados específicos del proveedor, la interoperabilidad no es posible [26].

Debido a las consideraciones anteriormente expuestas, es necesario configurar y utilizar el protocolos TACACS+ para comunicarse con los servidores AAA siempre que sea posible. Sin embargo, existe la posibilidad de incompatibilidad con el protocolo TACACS+, solo en estos casos debe validarse la opción de utilizar el servicio AAA empleando el protocolo RADIUS, el se basa en un estándar de la industria.

3.3.2.1. Configuración de la Autenticación

El proceso de configuración de un servidor AAA para la autenticación es el siguiente [6]:

1. Habilitar globalmente AAA en el dispositivo.
   Switch(config)# **aaa new-model**
2. Configurar el servidor destino junto con su contraseña secreta.
   Switch(config)# **{radius-server | tacacs-server} host** {*hostname* | *ip-address*} **key** *secret*

3. Definir un grupo que contendrá la lista de servidores.
   Switch(config)# **aaa group server {radius | tacacs+}** *group-name*
   Switch(config-sg)# **server** *ip-address*
4. Activar la autenticación para el ingreso.
   Switch(config)# **aaa authentication login** {**default** | *list-name*} **group** *group-name method1* [*method2* …]
   Cada método se debe enumerar en el orden en que va a probar. Si ninguno de los servidores del primer método responde, el dispositivo prueba los servidores del siguiente método enumerado. Los métodos pueden ser **tacacs+, radius** o **local.**

En conclusión, se recomienda utilizar el método **default** al momento de activar la autenticación para el ingreso. De esta forma, la autenticación se aplica a todas las líneas del dispositivo (VTY, consola o auxiliar). Del mismo modo, aunque idealmente la autenticación para el ingreso a un dispositivo de red remoto se realiza empleando un servidor AAA externo, también es necesario almacenar las contraseñas localmente como se expuso en el capítulo 3.3.1.3 y configurarlas como último recurso. Lo anterior, permite que en caso de falla del servidor AAA se pueda acceder remotamente a los dispositivos. Por lo tanto, es necesario configurar como *method1* RADIUS y como *method2* local.

3.3.2.2. Configuración de la Autorización

Para configurar un servidor AAA con autorización, es necesario definir una lista de métodos de autorización que se probarán en secuencia utilizando el siguiente comando de configuración global [6].
Switch(config)# **aaa authorization** {**commands** | **config-commands** | **configuration** | **exec** | **network** | **reverse-access**} {**default** | *list-name*} *method1* [*method2* …]

Los servicios o comandos que dependen de la autorización se configuran empleando las siguientes palabras clave [6].

- **commands**: el servidor debe devolver el permiso para usar cualquier comando en cualquier nivel de privilegio.
- **config-commands**: el servidor debe devolver el permiso para utilizar cualquier comando de configuración.
- **configuration**: el servidor debe devolver el permiso para ingresar al modo de configuración del conmutador.
- **exec:** el servidor debe devolver el permiso para iniciar un exec (shell).
- **network**: el servidor debe devolver el permiso para utilizar servicios relacionados con la red.
- **reverse-access**: el servidor debe devolver permiso al usuario para acceder a una sesión Telnet inversa.

Por otra parte, los métodos se muestran en seguida [6].

- **group** *group-name*: Las solicitudes se envían a los servidores de un grupo específico.
- **group** {**radius** | **tacacs+**}: Las solicitudes se envían a todos los servidores de este tipo.
- **if-authenticated**: Las solicitudes se otorgan si el usuario ya está autenticado.
- **none**: No se utiliza ninguna autorización externa; todos los usuarios son autorizados con éxito.

A partir de la configuración descrita, se debe tener en cuenta que la autorización AAA separa los comandos del conmutador y los comandos de configuración. Por lo tanto, la configuración que autoriza a los usuarios a ejecutar cualquier comando del conmutador y a realizar cambios de configuración, implica la utilización en conjunto de los servicios **commands** y **config-commands**. Así mismo, evaluando los servicios que dependen de la autorización, se recomienda evitar el uso de **reverse-access**, ya que las sesiones Telnet son inseguras como se presenta en el capítulo 3.3.3.

Así mismo, es necesario prevenir el uso de los métodos **if-auhenticated** y **none**. Aunque la primera opción en más segura que la segunda, ya que requiere una verificación previa de autenticación, ninguna de las dos realiza un control efectivo de autorización. Por lo tanto, se recomienda utilizar el comando **group tacacs+** para asegura la autorización AAA, ya que además de brindar los beneficios expuestos en el capítulo 3.3.2, TACACS+ también autoriza el uso de comandos específicos mientras que RADIUS ofrece un enfoque de todo o nada.

En conclusión, para realizar una autorización efectiva, se deben emplear los servicios **commands** y **config-commands**, en conjunto con el método TACACS+.

3.3.2.3. Configuración de la Contabilización

La configuración de un servidor AAA para la contabilización requiere definir una lista de métodos de contabilización que se probarán en secuencia utilizando el siguiente comando de configuración [6].
Switch(config)# **aaa accounting** {**system** | **exec** | **commands** *level*} {**default** | *list-name*} {**start-stop** | **stop-only** | **none**} *method1 [method2 …]*

La función que activa la contabilización se determina con una de las siguientes palabras clave [6].

- **system**: se registran los principales eventos, como una recarga.
- **exec**: se registra la autenticación del usuario en una sesión EXEC, junto con información sobre la dirección del usuario y la hora y duración de la sesión.
- **commands** *level*: se registra información sobre cualquier comando que se ejecute en un nivel de privilegio específico, junto con el usuario que emitió el comando.

Se puede especificar los tipos de registros de contabilización que se envían al servidor de contabilización utilizando las siguientes palabras clave [6].

- **start-stop**: los eventos se registran cuando comienzan y se detienen.
- **stop-only**: los eventos se registran solo cuando se detienen.
- **none**: no se registra ningún evento.

Tomando en consideración la configuración descrita, se concluye que para mantener un registro de los comandos que ingresan los usuarios que beneficie tanto a la seguridad como a los procesos de auditoría, es necesario utilizar la palabra clave **commands** y registrar los eventos tanto cuando comienzan y se detienen empleando la palabra clave **start-stop**.

3.3.3. Protocolos de conexión remota

Muchos ingenieros de redes suelen utilizar el protocolo Telnet para conectarse de forma remota a los dispositivos de red. Telnet envía los datos en texto sin cifrar, de modo que un usuario malintencionado que ejecuta un ataque de hombre en el medio puede visualizar información confidencial como nombres de usuario y contraseñas. Por otra parte, el protocolo SSH cifra el tráfico a través de un par de claves privada y pública, que son generadas empleando el algoritmo criptográfico asimétrico RSA [4]. La fortaleza criptográfica de RSA se expuso en el capítulo 2.3.1.

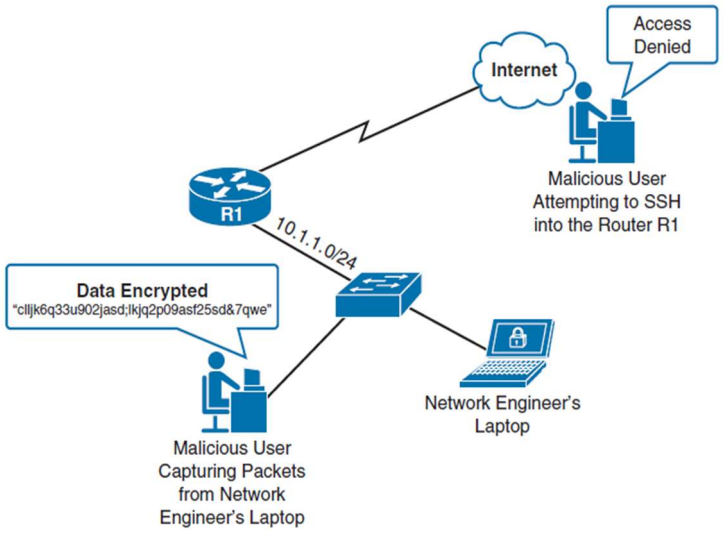

Figura 24. Defensa contra el acceso VTY no autorizado [4].

Los pasos para configurar SSH en un enrutador o conmutador Cisco son los siguientes [27].

1. Activar globalmente AAA en el dispositivo.
   Router(config)# **aaa new-model**
2. Configurar la autenticación AAA.
   Router(config)# **aaa authentication login default group radius local**
3. Crear un nombre de usuario y una contraseña local como respaldo.
   Router(config)# **username** *username* **privilege** *level* **secret** *password*
4. Configurar el servidor RADIUS.
   Router(config)# **radius-server host** *ip-adddress* **key** *password*
5. Especificar un nombre para el dispositivo. El nombre es uno de los elementos utilizados para crear el par de claves RSA.
   Router(config)# **hostname** *host-name*
6. Especificar un nombre de dominio para el dispositivo. El nombre de dominio también se utiliza para crear el par de claves RSA.
   Router(config)# **ip domain-name** *domain-name*
7. Generar el par de claves RSA.
   Router(config)# **crypto key generate rsa modulus** *size_of_modulus*
8. Configurar la versión de SHH.
   Router(config)# **ssh version** [**1** | **2**]
9. Habilitar SSH como el único protocolo de transporte de línea.
   Router(config-vty)# **transport input ssh**
10. Establecer la configuración de AAA para iniciar la sesión de la línea.
    Router(config-vty)# **login authentication default**

El nivel de privilegio debe ser asignado de acuerdo con el principio de menor privilegio y teniendo en cuenta el rol que desempeña el usuario en la organización. Además, según el documento "*SSL and TLS Deployment Best Practices"* del 2021 de GitHub, la seguridad proporcionada por claves RSA de 2048 bits es suficiente. Del mismo modo, es necesario utilizar la versión 2 de SSH, puesto que la versión 1 es vulnerable a un agujero de seguridad que permite a un intruso insertar datos en el flujo normal de la comunicación.

En conclusión, para evitar comprometer la conexión remota, se debe asignar un nivel de usuario adecuado, emplear SSHv2, utilizar un tamaño de módulo de al menos 2048 bits y habilitar SSH como el único protocolo de transporte de línea.

3.3.4. Protocolo simple de administración de red (SNMP)

SNMP permite a un dispositivo de red compartir información sobre sí mismo y sus actividades. Un sistema SNMP completo consta de los siguientes tres componentes [4].

- Administrador SNMP: Ejecuta una aplicación de administración de red para sondear y recibir datos de cualquier dispositivo.
- Agente SNMP: Es una pieza de software que se ejecuta en un dispositivo de red administrado. El propio dispositivo recopila todo tipo de datos y los almacena en una base de datos local, denominada MIB que se actualiza en tiempo real.
- Base de información de administración (MIB): Se define mediante una serie de objetos que poseen una estructura en forma de árbol jerárquico. Cada variable en la MIB es referenciada por un identificador de objeto (OID), que es una larga cadena de índices concatenados que siguen la ruta desde la raíz del árbol hasta la ubicación de la variable.

Un administrador SNMP puede enviar información, solicitar información o recibir información no solicitada de un dispositivo administrado a través de la OID de la variable específica, empleando los siguientes mensajes [4].

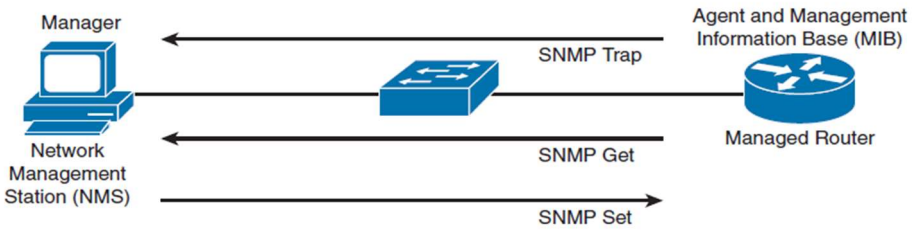

Figura 25. Componentes y mensajes de administración de red SNMP [4].

- GET: Recupera información de un dispositivo administrado.
- SET: Establece una variable o desencadena una acción en un dispositivo administrado.
- Trap: Mensaje no solicitado enviado desde un dispositivo administrado a un administrador, que notifica al administrador sobre un evento.

3.3.4.1. Seguridad en SNMPv1 y SNMPv2c

La seguridad integrada con SNMPv1 y SNMPv2c se considera débil debido a que usan cadenas de comunidad (muy parecidas a contraseñas) que son enviadas como texto en claro, para obtener acceso de solo lectura o lectura y escritura en un dispositivo administrado. Por ende, a través de un ataque de hombre en el medio, se puede obtener la cadena de comunidad y con esto leer o escribir variables en la base de datos MIB de un agente [6].

3.3.4.2. Seguridad en SNMPv3

SNMPv3 corrige algunas debilidades de seguridad de SNMPv1 y SNMPv2c, ya que busca garantizar la confidencialidad, integridad y autenticación de los mensaje SNMP. Los modelos y niveles de seguridad soportados por el IOS de Cisco, para las diferentes versiones de SNMP son los siguientes [28].

Modelo	Nivel de seguridad	Autenticación	Cifrado
SNMPv1	NoAuthNoPriv	Cadena de comunidad	Ninguno
SNMPv2c	NoAuthNoPriv	Cadena de comunidad	Ninguno
SNMPv3	NoAuthNoPriv	Nombre de usuario	Ninguno
SNMPv3	AuthNoPri	HMAC-MD5-96 HMAC-SHA-96	Ninguno
SNMPv3	AuthPri	HMAC-MD5-96 HMAC-SHA-96	DES, 3DES o AES

Tabla 5. Modelos de seguridad y niveles de seguridad admitidos por el IOS de Cisco [28].

Teniendo en cuenta la descripción realizada en el capítulo 2.4 para las funciones criptográficas de reducción MD5 y SHA, se puede concluir que SNMPv3 no garantiza ni la integridad ni la autenticación de los mensajes SNMP, ya que estos dos algoritmos han sido comprometidos por ataques de colisiones. Por otra parte, según lo descrito en el capítulo 2.2.4, se debe configurar AES en lugar de DES o 3DES debido a que el primero ya ha sido quebrado y el segundo es muy lento. La configuración de SNMPv3 se describe a continuación [6].

1. Definir una vista específica para los usuarios.
   Switch(config)# **snmp-server view** *view-name oid-tree*
2. Configurar un nombre de grupo para establecer las políticas de nivel de seguridad de los usuarios de SNMPv3.
   Switch(config)# **snmp-server group** [*group-name* {**v3** [**auth** | **noauth** | **priv**]}] [**read** *read-view*] [**write** *write-view*] [**notify** *notify-view*]
3. Crear el nombre de usuario que el administrador SNMP utiliza para comunicarse con el dispositivo.
   Switch(config)# **snmp-server user** *user-name group-name* {**v3** [**encrypted**] [**auth** {**md5** |**sha**} *auth-password*]} **priv** {**des** | **3des** | **aes** {**128** | **192** | **256**} *priv-password*
4. Identificar el administrador SNMP que recibe los mensajes.
   Switch(config)# **snmp-server host** *host-address* **[informs] version 3 priv** *username* [*trap-type*]

Si no se define una vista específica para los usuarios, todas las variables MIB son visibles. Al utilizar el comando **snmp-server view** se define una vista específica para los usuarios, en la que solamente las variables MIB configuradas con el parámetro *oid-tree* son visibles. Esto crea una capa adicional de seguridad y por ende se recomienda su configuración.

Además, al configurar el nombre de grupo y establecer las políticas de seguridad asociadas, es necesario seleccionar el nivel de seguridad **priv**, puesto que permite que los paquetes sean autenticados y encriptados. La vista configurada se emplea acá para limitar el acceso a las opciones de lectura, escritura o notificación.

Así mismo, al momento de crear el nombre de usuario, se requiere emplear la opción de cifrado **aes 256** puesto que es la más segura. También, se debe configurar la opción **encrypted** dado que, si no se utiliza, las contraseñas son almacenadas como texto sin cifrar.

Finalmente, cuando se configura el administrador SNMP que recibe los mensajes, se recomienda utilizar la palabra clave **informs** para que el dispositivo envíe mensajes SNMP no solicitados y configurar el nivel de seguridad **priv** de modo estos paquetes también sean autenticados y encriptados.

3.3.5. NTP

La correlación de eventos requiere de marcas de tiempo precisas, que permitan realizar el seguimiento de los registros y evitar cualquier inconsistencia que se pueda presentar. Aunque se puede configurar individualmente el reloj en cada uno de los dispositivos de red, estos relojes pueden variar con el tiempo y no coincidir. Tener un punto de referencia en común permite que la solución sea escalable y que los eventos sean correlacionados adecuadamente. El protocolo de tiempo de red (NTP), sincronizar los relojes de los sistemas informáticos [4].

NTP usa un valor denominado estrato (*stratum*), para indicar el nivel de credibilidad de una fuente de tiempo. Los valores de estrato válidos están en el rango de 0 a 15, con un valor de 16 que se usa para indicar que un dispositivo no tiene su hora sincronizada. Los valores de estrato inferior se consideran más autoritarios que los valores de estrato superior, siendo el valor de estrato 0 el más autoritario [4].

Desde una perspectiva de seguridad, un atacante puede introducir su propio dispositivo NTP en una red y anunciar una hora falsa a los dispositivos, dando como resultado información de marca de tiempo imprecisa en los registros y afectando las listas de acceso basadas en rangos de tiempo. Para mitigar este riesgo, se debe configurar la autenticación NTP. Los pasos de configuración se muestran enseguida [4].

- Configurar tanto la clave secreta como un ID de clave.
  Router(config)# **ntp authentication-key** *key-id* {**md5** | **cmac-aes-128** | **hmac-sha1** | **hmac-sha2-256**} *key*
- Indicar al dispositivo que autentique las fuentes de tiempo.
  Router(config)# **ntp authenticate**
- Especificar el ID de la clave que es confiable para la autenticación NTP.
  Router(config)# **ntp trust-key** *key-id*
- Configure la dirección IP y el ID de clave específica del servidor NTP.
  Router(config)# **ntp server** *ip-address* **key** *key-id*

En este caso, se debe configurar el HMAC con SHA-256 para asegurar la integridad y autenticación del servidor NTP, ya que este algoritmo de reducción criptográfico no ha sido vulnerado con ataques de colisiones y se considera hoy en día como seguro. Adicionalmente, aunque es posible configurar la hora local de un dispositivo como fuente NTP, se recomienda en su lugar emplear un servidor NTP externo fiable.

4. Conclusiones

A lo largo del presente Trabajo Final de Especialización se han expuesto los protocolos más relevantes que se emplean para configurar los planos de datos, control y gestión, de los conmutadores y enrutadores de Cisco. Así mismo, teniendo en cuenta un análisis de los algoritmos criptográficos soportados y las funcionalidades de cada comando, se ha concluido sobre la forma como los protocolos deberían ser configurados para garantizar un buen nivel de seguridad.

En este sección, se identifican de forma reducida el conjunto de prácticas que permitan asegurar los planos de datos, control y gestión, en los dispositivos conmutadores y enrutadores de Cisco.

El plano de datos de enrutadores y conmutadores Cisco requiere de la configuración de las siguientes capas de seguridad.

- Control de acceso a la red: Las listas de control de acceso (ACL) brindan una gran variedad de posibilidades para controlar los paquetes que se aceptan o rechazan en una red. Además, permiten controlar los periodos de tiempo dentro de los cuales se aplican estas reglas, lo que brinda la posibilidad de establecer en las organizaciones políticas de seguridad asociadas a los periodos de trabajo y descanso de forma aislada. Así mismo, son eficientes para evitar que tráfico malicioso como virus o troyanos ingrese a la red en forma de paquetes fragmentados, la cual ha sido una técnica muy utilizada por los atacantes. Del mismo modo, permiten controlar el tráfico que proviene de redes externas, y cuyo origen no deberían ser servicios de la red interna. Finalmente, son un complemento de protocolos y funcionalidades que se emplean en los planos de control y gestión, al asignar las direcciones IP de origen o destino en las que se confía. En síntesis, las ACL son una herramienta fundamental que debe configurarse en los puntos de la red donde la seguridad es crítica.
- Protección contra el acceso ilegítimo a un puerto: La seguridad del puerto limita las direcciones MAC que puede aprender un conmutador en un puerto específico, evitando que usuarios malintencionados se conecten a la red o que efectúen un ataque de desbordamiento de la tabla MAC. Sin embargo, debido a que existen varios programas que permiten modificar por software la dirección MAC, se recomienda utilizar el estándar IEEE 802.1X para proporcionar una capa más segura de autenticación basada en puertos, ya que al emplear un servidor RADIUS, solicita credenciales de usuario y contraseña antes de habilitar el acceso a la red LAN.

Así mismo, para proteger el acceso a los puertos de red de conmutadores y enrutadores, se deben deshabilitar lógicamente los puertos que no están siendo utilizados. De esta forma, se evita el uso de puertos que no han sido configurados con opciones de seguridad o que no se han dispuesto para ser utilizados en la red.

- Protección contra el acceso ilegítimo a una red de área local virtual (VLAN) de conmutador: El protocolo IEEE 802.1Q se utiliza para transportar diferentes VLAN a través de enlaces troncales, empleando información de etiquetado. El protocolo DTP permite que los puertos de un conmutador negocien dinámicamente su uso y su modo de encapsulación. Sin embargo, DTP puede ser aprovechado por un usuario malintencionado que simula conectar un conmutador y negocia un enlace troncal, con el fin de obtener acceso a todas las VLAN configuradas.

  Así mismo, un atacante ubicado en la VLAN nativa puede crear y enviar tramas con etiquetas 802.1Q falsificadas para que las cargas útiles del paquete aparezcan en una VLAN totalmente diferente, generando un ataque de salto de VLAN. Para prevenir este ataque, en redes donde todos los dispositivos son compatibles con 802.1Q, se debe configurar la VLAN nativa del enlace troncal con un ID falso y eliminar su uso en ambos extremos. Por el contrario, en redes donde existen dispositivos que no son soportan 802.1, se recomienda que todos los enlaces troncales 802.1Q agreguen etiquetas a las tramas para la VLAN nativa. Aplicando estos métodos, se elimina el riesgo de sufrir un ataque de salto de VLAN.

  En consecuencia, para prevenir el acceso ilegítimo a una VLAN, es necesario fijar el modo de enlace como troncal fijo o acceso, deshabilitar la funcionalidad DTP y utilizar algunos de los métodos previamente mencionados para mitigar los ataques de salto de VLAN.

- Prevención de ataques de suplantación de identidad en conmutadores: En los conmutadores Cisco se debe configurar la funcionalidad de indagación DHCP para evitar que un atacante intercepte y examine el contenido de los paquetes que se envían fuera de la red, al introducir un servidor DHCP deshonesto. La indagación DHCP clasifica como confiables únicamente los puertos conectados a servidores DHCP legítimos y descartar cualquier respuesta DHCP que provenga de un puerto no confiable.

  Además, se recomienda configurar en conjunto con la indagación DHCP las funcionalidades de protección de IP de origen y seguridad del puerto, ya que en conjunto permiten detectar y suprimir los ataques de suplantación de direcciones IP y MAC, incluso si ocurren dentro de la misma VLAN.

  Por último, como se expuso en el capítulo 1.2, los conmutadores usan el protocolo de resolución de direcciones (ARP) para resolver una dirección MAC desconocida a partir de una dirección IP conocida. Un usuario malintencionado puede suplantar respuestas ARP con su propia dirección MAC, para realizar un ataque de hombre en el medio. Para mitigar este ataque, los conmutadores Cisco utilizan la función de inspección dinámica de ARP (DAI), que clasifica los puertos como confiables o no confiables. Cuando se recibe una respuesta ARP en un puerto no confiable, el conmutador verifica sus valores de dirección MAC e IP contra valores configurados estáticamente o aprendidos dinámicamente empleando la indagación DHCP. Si la respuesta ARP contiene valores no válidos, se descarta y se genera un mensaje de registro.

- Prevención de ataques de suplantación de identidad en enrutadores: El protocolo uRPF evita que el tráfico con direcciones IP falsificadas sea transmito en la red, al verificar que la dirección IP de origen de un paquete recibido en una interfaz, sea accesible según la base de información de reenvío (FIB) del enrutador y que dicha interfaz se la misma por la cual enviaría el tráfico de regreso. Como se expuso en el capítulo 3.1.1.2, Cisco permite un control que diferencia entre interfaces que se conectan a la red interna e interfaces que se conectan a la red externa, al momento de realizar la configuración de uRPF esta diferenciación debe ser tomada en cuenta para prevenir que tráfico legítimo sea descartado. En conclusión, uRPF es un protocolo que debe configurarse en todas las interfaces del enrutador para prevenir el flujo de tráfico proveniente de direcciones IP falsificadas.

Por su parte, el plano de control de enrutadores y conmutadores Cisco de debe configurar de las siguientes capas de seguridad.

- Protección del árbol de expansión en conmutadores: El protocolo STP permite la convergencia de una red de conmutadores hacia un árbol de expansión sin bucles, brindando a nivel de capa 2 del modelo OSI redundancia y fiabilidad. Sin embargo, un usuario malintencionado puede introducir un conmutador en la red para suplantar el rol de puente raíz y generar tanto un cambio en la topología que puede ser inaceptable, como un periodo de indisponibilidad mientras dicha topología converge.

  Para prevenir este comportamiento indeseable, se puede implementar en los puertos donde nunca se espera recibir BPDU de puente raíz, la funcionalidad de protección de puente raíz. Esta funcionalidad se encarga de prevenir el envío y la recepción de datos en un puerto en el que se reciben BPDU con un ID de puente raíz más deseable, evitando la elección de un conmutador no previsto como puente raíz.

Del mismo modo, se puede configurar en los puertos de usuario final la protección contra BPDU inesperadas. Esta funcional por su parte, evita que el puerto acepte cualquier BPDU al colocarlo inmediatamente en una condición de error en la que se apaga lógicamente.

En síntesis, al emplear la funcionalidad de protección de puente raíz en los puertos donde nunca se espera recibir BPDU de puente raíz y la funcionalidad de protección contra BPDU inesperadas en los puertos de usuario final, se protege adecuadamente el árbol de expansión de los conmutadores y se garantiza una red a nivel de capa 2 redundante y fiable.

- Seguridad en protocolos de enrutamiento: Los enrutadores que utilizan el protocolo de puerta de enlace interior OSPF, envían paquetes de saludo a través de las interfaces cuyas subredes son anunciadas, con el objetivo de formar relaciones de vecindad que permitan el intercambio de información de estado de enlace. No obstante, cuando los paquetes son transmitidos a través de interfaces no confiables, se pueden formar relaciones de vecindad con dispositivos ilegítimos que buscan aprender, inyectar y modificar las rutas de la organización.

  Para prevenir la formación de vecinos no deseados, se recomienda utilizar la funcionalidad de interfaz pasiva. Esta funcionalidad permite anunciar la subred de una interfaz, al tiempo que evita que envíe paquetes de saludo. Del mismo modo, es necesario emplear la autenticación de vecinos, la cual mediante una clave precompartida, genera un código de autenticación de mensaje hash (HMAC), que permite determinar los vecinos legítimos. Dicho HMAC debe ser generado empleando la opción hmac-sha-512, ya que esta brinda el mayor grado de seguridad. Adicionalmente, se recomienda generar un llavero con varias claves con el fin de determinar diferentes periodos de aceptación y envío acordes con las políticas de seguridad de la organización.

Por otra parte, la formación de vecinos no deseados en el protocolo de enrutamiento de puerta de enlace exterior BGP es poco probable, dado que se requiere que cada uno de los pares sea referenciado por su vecino para crear la conexión TCP. Sin embargo, en el escenario en el cual dos enrutadores han establecido una sesión TCP, un atacante puede secuestrarla para corromper la tabla BGP.

Para mitigar esta amenaza, se debe configurar La opción de autenticación TCP (TCP-AO), la cual permite la autenticación de vecinos y el aseguramiento de la conexión TCP. Para configurar TCP-AO se recomienda emplear el algoritmo criptográfico hmac-sha-256 ya que ofrece el mayor nivel de seguridad de las opciones posibles. Además, es necesario configurar diferentes claves dentro del llavero de claves, cada una con un tiempo de aceptación y envío acordes con las políticas de seguridad de la organización

Finalmente, en el plano de gestión de enrutadores y conmutadores Cisco es necesario configurar las siguientes capas de seguridad.

- Protección de contraseñas: Para almacenar eficazmente la contraseña de habilitación y prevenir que usuarios malintencionados accedan al modo EXEC privilegiado, esta debe configurarse empleando el algoritmo criptográfico de reducción SHA-256, ya que sigue siendo considerado como seguro en la actualidad y de las opciones de configuración posibles es el que brinda el mayor grado de seguridad.

  Por otra parte, para autenticar a un usuario que intenta iniciar sesión en una de las líneas (VTY, consola o auxiliar) del dispositivo de red, es necesario prevenir el uso de contraseñas de línea ya que se almacenan en la configuración como texto plano y solamente puede cifrarse empleando el servicio de cifrado de contraseñas que utiliza el algoritmo criptográfico inseguro de Vigenère. Por el contrario, se recomienda configurar combinaciones de nombres de usuario y contraseña, utilizando la opción que genera un hash SHA-256 de la contraseña y que permite realizar un proceso de autorización a través de diferentes niveles de privilegio.

- Autenticación, autorización y contabilización (AAA): los servicios AAA permiten tener un repositorio centralizado y escalable para las credenciales de usuario. La autenticación valida la identidad del usuario. Por su parte, la autorización determina el nivel de privilegios del usuario autenticado. Finalmente, la contabilización recopila y almacena información sobre la sesión del usuario.

  Los dispositivos Cisco pueden utilizar los protocolos TACACS+ y RADIUS para comunicarse con los servidores AAA. RADIUS usa UDP, cifra únicamente la contraseña en el paquete de solicitud de acceso del cliente al servidor, combina las funciones de autenticación y autorización, no admite ciertos protocolos y ofrece un enfoque de autorización de todo o nada. Por otra parte, TACAS+ usa TCP, cifra todo el cuerpo del paquete, proporciona servicios separados para la autenticación, autorización y contabilización, ofrece soporte multiprotocolo y proporciona métodos para controlar de forma granular la autorización de los comandos. En consecuencia, se recomienda configurar y utilizar el protocolos TACACS+ para comunicarse con los servidores AAA.

  La autenticación AAA debe activarse para verificar la identidad de los usuarios en todas las líneas (VTY, consola o auxiliar) de dispositivo. Del mismo modo, se recomienda configurar localmente combinaciones de nombre de usuario y contraseñas, para ser utilizadas como último recurso en caso de falla del servidor AAA. Por otra parte, la autorización AAA requiere de la habilitación de los servicios que permiten la autorización granular de los comandos del conmutador y de los comando de configuración que un usuario puede ejecutar. Finalmente, para mantener un registro de los comandos que ingresan los usuarios que beneficie tanto a la seguridad como a los procesos de auditoría, es necesario registrar información sobre cualquier comando que se ejecute en un nivel de privilegio específico, junto con el usuario que emitió el comando y registrar los eventos cuando comienzan y terminan. Siguiendo estas recomendaciones, se puede garantizar una configuración segura de los servicios AAA.

- Protocolos de conexión remota: para la conexión de forma remota a los dispositivos de red, se recomienda utilizar el protocolo SSH en su versión 2, ya que la versión 1 es vulnerable a un agujero de seguridad que permite a un intruso insertar datos en el flujo normal de la comunicación. Así mismo, es necesario emplear un módulo de al menos 2048 bits al momento de generar el par de claves RSA. Por último, se debe emplear un servidor AAA centralizado en conjunto con la configuración de nombres de usuario y contraseña locales a los que se les asignen diferentes niveles de privilegios. De esta forma, se asegura la conexión remota y se garantiza la confidencialidad al cifrar el tráfico, la integridad al utilizar la versión 2 de SSH y la disponibilidad al tener como respaldo nombres y contraseñas de usuario locales en caso de que se pierda la conexión con el servidor AAA remoto.
- Administración de red: el protocolo simple de administración de red (SNMP) permite administrar y monitorear los dispositivos de red. Se recomienda prevenir el uso de las versiones inseguras SNMPv1 y SNMPv2, que utilizan cadenas de comunidad enviadas como texto en claro y no proveen ningún nivel de integridad ni autenticación, en favor de la versión SNMPv3 que provee un alto nivel de cifrado al emplear AES y un bajo nivel de integridad y autenticación a través de HMAC-MD5-96 y HMAC-SHA-96. Así mismo, es necesario configurar vistas específicas para los usuarios, en las que se segmente la información que pueden leer o escribir dependiendo de su rol en la organización. Finalmente, al momento de crear las claves utilizadas en la computación del hash, es necesario almacenarlas como texto cifrado, puesto que de lo contrario son almacenas como texto en claro. Tomando en cuenta estas consideraciones se provee una administración y monitoreo seguro de los dispositivos conmutadores y enrutadores de Cisco.

- Sincronización de relojes: el protocolo de tiempo de red (NTP) se encarga de sincronizar los relojes de los sistemas informáticos. Esta sincronización, permite un análisis consistente y un monitoreo coherente de los registros que generan los dispositivos, favoreciendo las labores de auditoría y correlación de eventos. Del mismo modo, la sincronización de relojes permite que las ACL que referencian periodos de tiempo, se ejecuten correctamente. Dada su relevancia, es necesario configurar la autenticación NTP para evitar que un usuario malintencionado anuncie una hora falsa, que genere marcas de tiempo imprecisas y afecte las ACL previamente mencionadas.

  Para cumplir con este objetivo, se debe configurar el código de autenticación de mensaje hash (HMAC) con la opción SHA-256, que permite asegurar tanto la integridad como la autenticación del servidor NTP. Además, es necesario configurar la hora empleando un servidor NTP externo fiable en lugar de un dispositivo local como fuente NTP. Tomando en consideración estas recomendaciones, se sincronizan los relojes de los sistemas de forma segura, favoreciendo la auditoría y la correlación de eventos.

Se puede concluir que existe una gran cantidad de protocolos que deben ser utilizados para configurar los planos de datos, control y gestión de los conmutadores y enrutadores Cisco. Esta configuración debe realizarse de forma correcta para asegurar los tres pilares de la seguridad informática que son la confidencialidad, integridad y disponibilidad. Como pudimos ver a lo largo de este Trabajo Final de Especialización, Cisco ha implementado varios algoritmos criptográficos y funcionalidades para cumplir con este objetivo. Sin embargo, aún queda trabajo por hacer, ya que cada día los atacantes intentan descubrir nuevas vulnerabilidades tanto en los métodos de implementación como en las primitivas criptográficas. Además, el desarrollo de la computación cuántica plantea nuevos retos a la criptografía moderna.

5. Bibliografía

[1] P. Shenoy, «Gartner names Cisco a Leader in LAN Access for 5th Year Straight,» 12 Noviembre 2019. [En línea]. Available: https://blogs.cisco.com/networking/gartner-names-cisco-a-leader-in-lan-access-for-5th-year-straight. [Último acceso: 3 Marzo 2021].

[2] P. Shenoy, «Gartner Names Cisco a Leader in 2020 Magic Quadrant for WAN Edge Infrastructure,» 30 Septiembre 2020. [En línea]. Available: https://blogs.cisco.com/networking/gartner-names-cisco-a-leader-in-2020-magic-quadrant-for-wan-edge-infrastructure. [Último acceso: 3 Marzo 2021].

[3] D. McGinniss, «From Data Center to Cloud, Guidance for Managing Data Everywhere,» Cisco Systems, 15 Julio 2020. [En línea]. Available: https://blogs.cisco.com/datacenter/from-data-center-to-cloud-guidance-for-managing-data-everywhere. [Último acceso: 3 Marzo 2021].

[4] K. Wallace, CCNP Routing and Switching ROUTE 300-101, Indianapolis, Indiana: Pearson Education, 2015.

[5] A. Pachon, «La evolución en la arquitectura de las redes,» *Sistemas & Telematica,* vol. 1, nº 1, pp. 89-100, 28 Julio 2006.

[6] D. Hucaby, CCNP Routing and Switching SWITCH 300-115, Indianapolis, Indiana: Pearson Education, 2015.

[7] Jorge, «Conoce el proceso de aprendizaje de direcciones MAC para un switch,» Huawei Technologies Co., Ltd, 6 Septiembre 2019. [En línea]. Available: https://forum.huawei.com/enterprise/es/conoce-el-proceso-de-aprendizaje-de-direcciones-mac-para-un-switch/thread/565583-100237. [Último acceso: 5 Abril 2021].

[8] P. Szewczyk y R. Macdonald, «Broadband Router Security: History, Challenges and Future Implications,» *Journal of Digital Forensics, Security and Law,* vol. 12, nº 4, pp. 55-74, Diciembre 2017.

[9] J. F. Kurose y K. W. Ross, Computer Networking A Top-Down Approach, Sexta ed., New Jersey: Pearson Education, 2013.

[10] C. Paar y J. Pelzl, Understanding Cryptography A Textbook for Students and Practitioners, Springer, 2010.

[11] J. Ramió Aguirre, «Class4crypt c4c6.6 Criptoanálisis a la cifra de Vigenère por el método Kasiski [Video],» 23 Marzo 2020. [En línea]. Available: https://www.youtube.com/watch?v=K3tpKeDQs6s. [Último acceso: 20 Mayo 2021].

[12] P. Hawkes, M. Paddon y R. Gregory, «On Corrective Patterns for the SHA-2 Family,» 22 Agosto 2004. [En línea]. Available: https://eprint.iacr.org/2004/207. [Último acceso: 10 Septiembre 2021].

[13] E. Andreeva, B. Mennink, B. Preneel y M. Skrobot, Progress in Cryptology - AFRICACRYPT 2012, Berlín: Springer Nature, 2012.

[14] Cisco Systems, «Cisco Technical Tips Conventions,» 20 Septiembre 2004. [En línea]. Available: https://www.cisco.com/c/en/us/support/docs/dial-access/asynchronous-connections/17016-techtip-conventions.html. [Último acceso: 16 Junio 2021].

[15] Cisco Systems, «Catalyst 3560 Switch Command Reference,» 2005. [En línea]. Available: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3560/software/release/12-2_25_sed/command/reference/3560cr.pdf. [Último acceso: 22 Junio 2021].

[16] Global Knowledge, «How to Secure Cisco Routers and Switches,» 17 Junio 2018. [En línea]. Available: https://www.globalknowledge.com/us-en/resources/resource-library/articles/how-to-secure-cisco-routers-and-switches/. [Último acceso: 30 Octubre 2020].

[17] securew2, «What is 802.1X? How Does it Work?,» [En línea]. Available: https://www.securew2.com/solutions/802-1x. [Último acceso: 10 Junio 2021].

[18] DHCP snooping, 10 Abril 2021. [En línea]. Available: https://en.wikipedia.org/wiki/DHCP_snooping. [Último acceso: 7 Agosto 2021].

[19] R. Molenaar, «Introduction to Spanning-Tree,» [En línea]. Available: https://networklessons.com/tag/stp/introduction-to-spanning-tree. [Último acceso: 10 Julio 2021].

[20] R. Molenaar, «OSPF HMAC-SHA Extended Authentication,» [En línea]. Available: https://networklessons.com/cisco/ccie-routing-switching-written/ospf-hmac-sha-extended-authentication. [Último acceso: 5 Septiembre 2020].

[21] Huawei Technologies, «BGP Implementation,» 8 Diciembre 2020. [En línea]. Available: https://support.huawei.com/enterprise/en/doc/EDOC1000178110/b38b2b8e/bgp-implementation. [Último acceso: 20 Agosto 2021].

[22] MDN Web Docs, «Session Hijacking,» 3 Febrero 2021. [En línea]. Available: https://developer.mozilla.org/en-US/docs/Glossary/Session_Hijacking. [Último acceso: 20 Agosto 2021].

[23] Cisco Systems, «IP Routing: BGP Configuration Guide, Cisco IOS XE Gibraltar 16.12.x,» 7 Mayo 2021. [En línea]. Available: https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_bgp/configuration/xe-16-12/irg-xe-16-12-book/bgp-support-for-TCP-AO.html. [Último acceso: 20 Julio 2021].

[24] S. Friedl's, «An Illustrated Guide to Cryptographic Hashes,» 15 Febrero 2005. [En línea]. Available: http://www.unixwiz.net/techtips/iguide-crypto-hashes.html. [Último acceso: 2 Agosto 2021].

[25] R. Molenaar, «AAA Local Command Authorization,» [En línea]. Available: https://networklessons.com/cisco/ccie-routing-switching/aaa-local-command-authorization. [Último acceso: 12 Septiembre 2021].

[26] Cisco Sytems, «TACACS+ and RADIUS Comparison,» 14 Enero 2008. [En línea]. Available: https://www.cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/13838-10.html#comp_udp_tcp. [Último acceso: 20 Julio 2021].

[27] C. A. Romero Goyzueta, «Securing SSH Access Using AAA and Radius Server on Cisco Router [Video],» 12 Septiembre 2018. [En línea]. Available: https://www.youtube.com/watch?v=wzkMvPpA97w. [Último acceso: 7 Agosto 2021].

[28] Cisco Systems, «Cisco UCS Manager System Monitoring Guide Using the CLI, Release 4.2,» 27 Agosto 2021. [En línea]. Available: https://www.cisco.com/c/en/us/td/docs/unified_computing/ucs/ucs-manager/CLI-User-Guides/System-Monitoring/4-2/b_UCSM_CLI_System_Monitoring_Guide_4-2/b_UCSM_CLI_System_Monitoring_Guide_chapter_01001.html. [Último acceso: 5 Mayo 2021].

[29] National Institute of Standards and Technology, «SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions,» 4 Agosto 2015. [En línea]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf. [Último acceso: 7 Agosto 2021].