



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

**GRADO EN INGENIERÍA EN TECNOLOGÍAS Y SERVICIOS
DE TELECOMUNICACIÓN**

ÁREA DE INGENIERÍA TELEMÁTICA

TRABAJO FIN DE GRADO N° 18010063

**Herramienta de detección de vulnerabilidades en capas 2 y 3 en el ámbito de
una red corporativa**

**Dña. del Fueyo Mendoza, Tamara
TUTOR: D. Nuño Huergo, Pelayo
COTUTOR: D. González Bulnes, Francisco**

FECHA: Julio 2018

Índice general

1. Introducción	1
2. Motivaciones y objetivos	3
3. Estado actual: Alternativas	4
4. Análisis de protocolos y vulnerabilidades	6
4.1. Capa de enlace (Capa 2 según modelo OSI)	6
4.1.1. ARP	6
4.1.2. STP	10
4.2. Cisco	13
4.2.1. CDP	13
4.2.2. DTP	15
4.2.3. VTP	18
4.2.4. Problema <i>Double Tagging</i>	21
4.3. Capa de red (Capa 3 según modelo OSI)	23
4.3.1. RIP	23
4.3.2. OSPF	28
4.4. Protocolos restantes	33
4.4.1. DHCP	33

4.4.2. HSRP	37
5. Soluciones	42
5.1. Capa de enlace (Capa 2 según modelo OSI)	42
5.1.1. ARP	42
5.1.2. STP	43
5.2. Cisco	44
5.2.1. CDP	44
5.2.2. DTP	45
5.2.3. VTP	46
5.2.4. Problema <i>Double Tagging</i>	47
5.3. Capa de red (Capa 3 según modelo OSI)	48
5.3.1. RIP	48
5.3.2. OSPF	49
5.4. Protocolos restantes	50
5.4.1. DHCP	50
5.4.2. HSRP	51
6. Trabajo realizado	53
6.1. Visión general	54
6.2. Detección	55
6.2.1. Capa de enlace (Capa 2 según modelo OSI)	55

6.2.1.1.	ARP	55
6.2.1.2.	STP	57
6.2.2.	Cisco	57
6.2.3.	Capa de red (Capa 3 según modelo OSI)	59
6.2.3.1.	RIP	59
6.2.3.2.	OSPF	60
6.2.4.	Protocolos restantes	61
6.2.4.1.	DHCP	61
6.2.4.2.	HSRP	64
6.3.	Notificación	65
6.3.1.	<i>snObjs</i> MIB	66
6.3.2.	<i>snEvnts</i> MIB	68
6.3.2.1.	<i>arpEv</i>	68
6.3.2.2.	<i>ripv1Ev</i>	69
6.3.2.3.	<i>ripv2Ev</i>	69
6.3.2.4.	<i>ospfEv</i>	69
6.3.2.5.	<i>cdpEv</i>	69
6.3.2.6.	<i>vtpEv</i>	70
6.3.2.7.	<i>dtpEv</i>	70
6.3.2.8.	<i>dhcpDiscEv</i>	70
6.3.2.9.	<i>dhcpOffEv</i>	70

6.3.2.10.	<i>dhcpAckEv</i>	71
6.3.2.11.	<i>dhcpOffAckEv</i>	71
6.3.2.12.	<i>stpEv</i>	71
6.3.2.13.	<i>hsrpEv</i>	71
7.	Pruebas y Resultados	72
7.1.	Pruebas individuales	72
7.1.1.	Capa de enlace (Capa 2 según modelo OSI)	72
7.1.1.1.	ARP	73
7.1.1.2.	STP	76
7.1.2.	Cisco	76
7.1.2.1.	CDP	77
7.1.2.2.	DTP	78
7.1.2.3.	VTP	78
7.1.3.	Capa de red (Capa 3 según modelo OSI)	79
7.1.3.1.	RIP	79
7.1.3.2.	OSPF	81
7.1.4.	Protocolos restantes	82
7.1.4.1.	DHCP	82
7.1.4.2.	HSRP	89
7.2.	Prueba genérica	90

8. Manual de usuario	91
8.1. Pasos previos	91
8.2. Ejecución de la herramienta	91
9. Planificación	98
9.1. Organización temporal	98
9.2. Tareas realizadas	98
10.Conclusiones	102
11.Líneas futuras	104
12.Glosario	106
Bibliografía	111



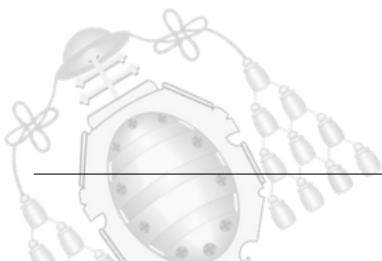
1. Introducción

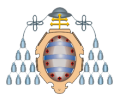
En los últimos años el uso de nuevas tecnologías en el ámbito corporativo se ha visto disparado debido al constante avance de las mismas. La ciberseguridad ha adquirido un papel muy importante, llegando a convertirse en una necesidad para las empresas de hoy en día, ya que los datos almacenados por estas tienen un valor incalculable en el mercado y los delincuentes lo saben. Por esta razón, los tests de penetración (*pentesting*), al igual que los detectores de vulnerabilidades, tienen una alta relevancia en el mundo empresarial, dado que pueden evitar que una compañía se vea sumida en el caos, manchando su reputación y provocando grandes pérdidas económicas por la explotación de alguna vulnerabilidad.

La inversión en herramientas de seguridad por parte de las empresas de cualquier sector se ve recompensada a largo plazo, ya que las pérdidas que podría ocasionar un hipotético ataque son bastante mayores que los gastos de prevención del mismo.

El *pentesting* trata de detectar brechas de seguridad en una red mediante la realización de ataques controlados, simulando las acciones que realizaría un atacante. Una vez finalizado el análisis se realiza un documento que es entregado a la empresa para que conozca los problemas de seguridad y tome medidas respecto a los mismos, reduciendo así el riesgo de ataques. Se persigue una mejora continua en la seguridad de la empresa, por lo que es recomendable la realización de forma periódica.

Con el fin de incrementar la seguridad de la empresa, existen múltiples herramientas que simulan diversos tipos de ataques, permitiendo comprobar de forma controlada y legítima cuales son las debilidades de la red corporativa. Además, también existen herramientas de detección de vulnerabilidades que informan en caso de hallar puntos débiles en la red [1], por ejemplo: una mala configuración de los protocolos usados, descuidos de fabricantes, existencia de puertas traseras, etc. [2]



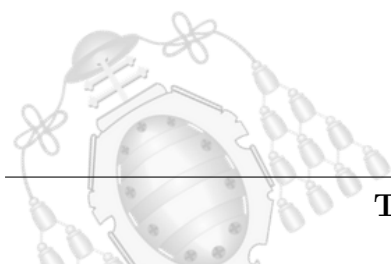


En la actualidad, el número de incidentes provocados por una mala configuración o algún descuido son cada vez más frecuentes, haciendo así que sea cada vez más necesario la realización de análisis de vulnerabilidades de forma periódica.

Con el presente proyecto se pretende crear una herramienta que ayude al administrador, de forma clara y sencilla, a conocer si algunos de los protocolos comúnmente usados en una LAN están correctamente configurados, facilitando así el análisis de vulnerabilidades. Se ha tenido en cuenta que la persona responsable de la red tiene más tareas que realizar, por lo que el sistema ha de ser autónomo, liberando al usuario de preocupaciones. En caso de que la herramienta detecte algún tipo de incongruencia relacionada con los protocolos se avisa al administrador inmediatamente mediante el envío de una notificación al servidor. Para aumentar la seguridad en las redes corporativas, se avisa al administrador de forma instantánea de las vulnerabilidades detectadas.

A lo largo de este documento se pueden encontrar los siguientes puntos:

- Capítulo 2: Motivaciones y objetivos
- Capítulo 3: Estado actual: Alternativas
- Capítulo 4: Análisis de los protocolos y vulnerabilidades
- Capítulo 5: Soluciones para las vulnerabilidades
- Capítulo 6: Trabajo realizado
- Capítulo 7: Pruebas y resultados obtenidos
- Capítulo 8: Manual de usuario
- Capítulo 9: Planificación del proyecto
- Capítulo 10: Conclusiones
- Capítulo 11: Líneas futuras
- Capítulo 12: Glosario

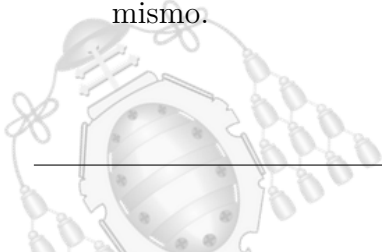


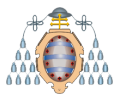


2. Motivaciones y objetivos

Con este proyecto se pretende crear una herramienta que facilite al administrador el *pentesting* de una red corporativa gracias a la utilización de una interfaz gráfica intuitiva. Además, haciendo uso de esta herramienta, el análisis de la red y la detección de vulnerabilidades o ataques se hace de forma autónoma, liberando al usuario de una elevada carga de trabajo y obteniendo el mayor rendimiento posible. No obstante, se ha buscado la forma de que el administrador no sea ajeno a todo lo que ocurre en la red, sino que ante la detección de una mala configuración o ataque se le avise de forma inmediata enviando una notificación SNMP. A continuación, se plantean los objetivos a cumplir con este proyecto:

- Analizar los protocolos de uso más frecuente en redes corporativas, identificando aquellos que puedan ser potencialmente peligrosos: ARP, STP, CDP, DTP, VTP, RIP, OSPF, DHCP y HSRP.
- Describir las razones por las cuales estos protocolos pueden resultar una amenaza.
- Determinar las posibles soluciones para que la red no sea vulnerable.
- La herramienta implementada debe detectar los paquetes de los protocolos mencionados anteriormente que puedan suponer una amenaza para la LAN, ya sea por una mala configuración o debido a un ataque. Para ello se utiliza la herramienta Scapy.
- Para agilizar la respuesta en caso de detectar alguna anomalía, se avisa al administrador de la red de forma inmediata, mediante el envío de una notificación SNMP a la dirección IP del servidor indicada en la interfaz gráfica, ante cualquier situación anómala.
- Por último, se pretende incluir una interfaz gráfica lo suficientemente intuitiva para que cualquier usuario con conocimientos básicos acerca de protocolos de red sea capaz de arrancar el programa y comprender la información devuelta por el mismo.





3. Estado actual: Alternativas

A lo largo de esta sección se detallan algunas de las herramientas que existen actualmente y que cumplen funciones similares a la creada:

- **NetworkRecon.ps1** [3]: Herramienta que avisa ante la detección de paquetes pertenecientes a protocolos considerados potencialmente peligrosos. Algunos de estos son: CDP, LLDP, DTP, VTP, STP, OSPF, DHCP, HSRP, etc. Se muestra un mensaje por consola ante la detección de un posible ataque o de un paquete que no se debería recibir.

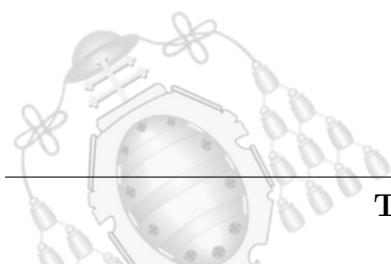
A diferencia de la herramienta desarrollada, esta usa el lenguaje PowerShell. Asimismo, no cuenta con interfaz gráfica ni con el envío de traps a un servidor SNMP, sino que como se comentaba anteriormente, se muestran los mensajes localmente a través de la consola.

- **Netcrunch** [4]: Software comercial para la monitorización de red. Puede supervisar hasta 65 servicios de red diferentes (por ejemplo: ping, DNS, DHCP, FTP, etc).

Esta herramienta, al igual que la creada, envía eventos que avisan al usuario de lo que ocurre en la red. Entre las diferentes opciones que presenta para el envío de estos eventos, se encuentra SNMP.

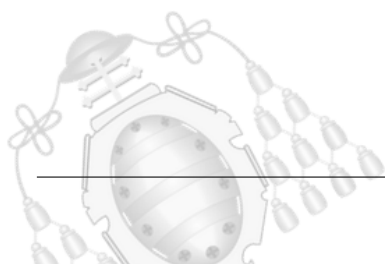
- **Yersinia** [5]: Herramienta con múltiples funcionalidades, entre las cuales se encuentra el análisis de paquetes.

Los protocolos que permite examinar son: STP, CDP, DTP, DHCP, HSRP, 802.1Q y VTP. Toda la herramienta está escrita en C, a diferencia de la propia que se ha escrito en Python. Al igual que la alternativa anterior, tampoco se cuenta con el envío de traps a un servidor SNMP.





- **ROCIO** [6]: Herramienta que permite realizar auditorías de seguridad a los diferentes dispositivos, verificando el nivel de seguridad de estos mediante la comprobación de unas reglas predefinidas. La principal diferencia encontrada con respecto a la herramienta creada es que ROCIO no trabaja en tiempo real, es decir, es estática, ya que analiza los ficheros de configuración de los dispositivos de capas 2 y 3, que deben ser subidos a la plataforma de análisis.





4. Análisis de protocolos y vulnerabilidades

En el siguiente capítulo se detallan algunos de los protocolos que se consideran potencialmente peligrosos en redes corporativas, y por consiguiente, los analizados a lo largo del proyecto. Para cada uno de los mismos se realiza una breve introducción y se explican sus vulnerabilidades.

4.1.- Capa de enlace (Capa 2 según modelo OSI)

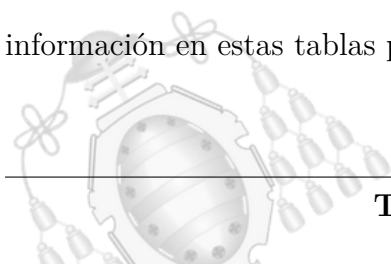
Esta capa, situada entre la física y la de red, es responsable del intercambio de información entre un dispositivo y la red, controlando el acceso al medio y los posibles errores de transmisión. En este apartado se han tenido en cuenta dos protocolos comúnmente usados: ARP y STP.

4.1.1.- ARP

El protocolo ARP es un protocolo muy extendido y usado actualmente debido a que permite encontrar la dirección MAC asociada a una determinada dirección IP. Sin embargo, a pesar de su popularidad cuenta con algunas vulnerabilidades que podrían provocar serios problemas en una red corporativa si no se perciben a tiempo.

ARP hace uso de dos tipos de mensajes: *Request* y *Reply*. Inicialmente se envía un mensaje de tipo *Request* a la dirección broadcast “FF:FF:FF:FF:FF:FF” preguntando por una dirección IP concreta, entonces se espera por un mensaje de tipo *Reply* que indica la dirección MAC correspondiente a la IP enviada.

Para el almacenamiento de estas asociaciones se hace uso de tablas ARP, donde se almacena de forma limitada o ilimitada cada par de direcciones. El almacenamiento de información en estas tablas puede ser de dos tipos: estático o dinámico.



En ARP estático las direcciones son añadidas directamente por el administrador, de esta forma las entradas de las tablas son únicamente *read-only* y se almacenan de forma permanente, solo desaparecen cuando se apaga el dispositivo. Haciendo uso de este tipo de configuración se reduce el uso de los recursos de la red. Asimismo, se goza de una configuración más segura ya que no es posible sufrir ataques de tipo *ARP Spoofing*, como se comenta más adelante. Aun así, hay que tener en cuenta que la configuración y actualización puede resultar tediosa y conducir a un mayor número de fallos debidos a despistes del administrador, dificultando así la escalabilidad y flexibilidad de la red.

Por otro lado, ARP dinámico almacena los pares de direcciones en las diferentes entradas de la tabla al igual que ARP estático, pero con una fecha de vencimiento ya que las direcciones MAC pueden cambiar. Estas entradas se aprenden a través del protocolo, por lo que las tablas están dotadas de menor seguridad que en el caso anterior y son potencialmente manipulables, actualizándose cada vez que reciben un mensaje de tipo *Reply* (Figura 4.1), ya sea añadiendo una nueva entrada a la tabla o modificando una existente [7] [8]. Para este tipo de configuración surge uno de los principales problemas que tiene el protocolo: *ARP Spoofing*.

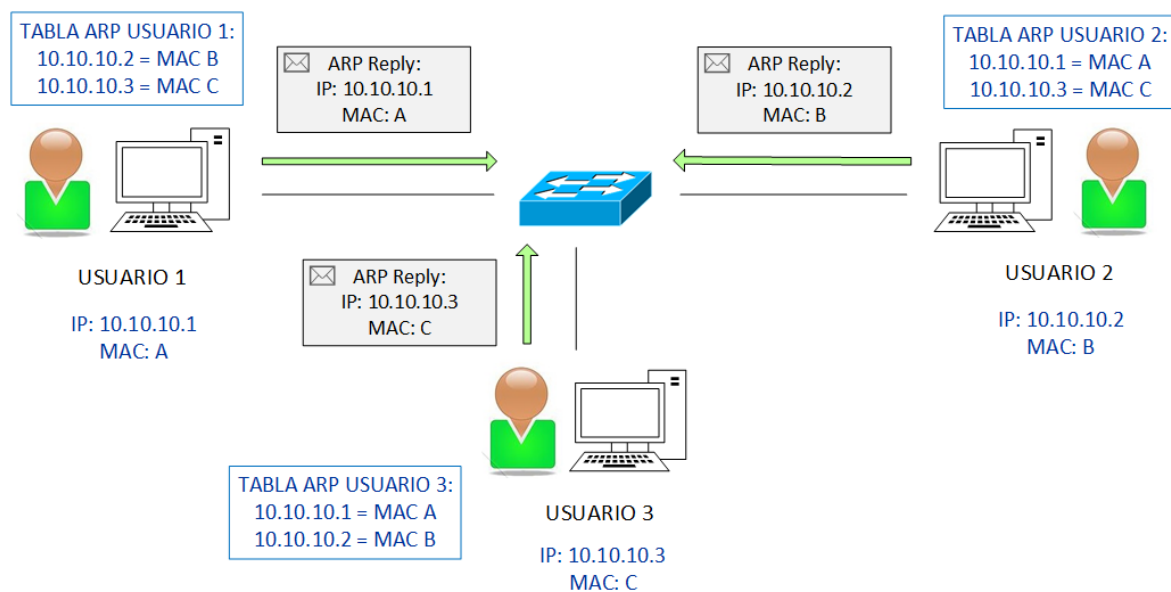
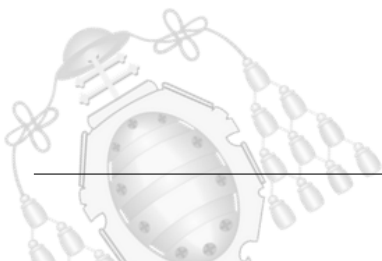


Figura 4.1.- Funcionamiento ARP dinámico





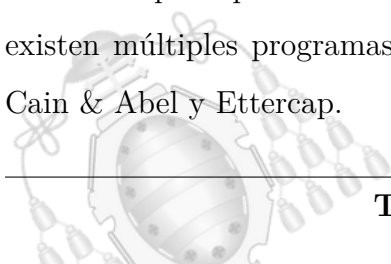
ARP *Spoofing* es una técnica que redirige el tráfico hacia el ordenador de un usuario fraudulento en vez de hacia el de la persona legítima. Esto le permite leer, modificar e incluso detener los paquetes que se envían a través de la red hacia esa dirección, ya que la dirección MAC que se tiene asociada es errónea.

Normalmente este ataque se usa como un puente para realizar otros, como puede ser denegación de servicio (*DoS*) o *man in the middle*.

- Denegación de servicio (*DoS*): Se asocian múltiples IPs a una sola dirección MAC, provocando así que el elemento con dicha dirección se sature por el exceso de paquetes recibidos. Además, podría darse la situación opuesta, es decir, se capturan todos los paquetes y se eliminan parcial o totalmente de la red.
- *Man in the middle*: Todos los paquetes enviados entre dos extremos serán interceptados por el atacante, este podrá realizar con ellos lo que desee, ya sea alterar su contenido o simplemente observarlo y posteriormente reenviarlo a su destinatario real [10].

Lo más habitual suele ser la visualización de la información que se transmite entre dos elementos de la red. Un ejemplo de este tipo de ataque es *session hijacking*, también conocido como “secuestro de cookie”. Consiste en el robo de identificadores de sesión, permitiendo al atacante hacerse pasar por el usuario correspondiente.

Su realización consiste en el envío de mensajes ARP falsos que asocien la dirección MAC del atacante a una dirección IP determinada (por ejemplo: un ordenador o una puerta de enlace), en vez de asociarla a la dirección MAC real. El envenenamiento de las tablas es posible debido a que se inundan las mismas con muchos paquetes ARP adulterados; por lo tanto, la tabla de la víctima se rellena o actualiza con información maliciosa. Esto es posible porque los mensajes ARP no disponen de ningún tipo de autenticación. Asimismo, se ha de realizar desde una máquina conectada directamente a la LAN o mediante intrusión (acceso ilícito a una máquina de dicha red), lo cual es uno de los principales inconvenientes de este ataque [9]. Para la realización del mismo existen múltiples programas, algunos de los más populares son: arpspoof, Arposion, Cain & Abel y Ettercap.





En las figuras 4.2 y 4.3 se puede ver un ejemplo de ARP *Spoofing*, ya que hay una alteración en la tabla ARP del usuario 1. La dirección MAC de la IP que tiene el usuario 1 (MAC: C) no se corresponde con el que debería (MAC: A). Esto provoca que todos los paquetes que se envían desde el usuario 2 hacia el usuario 1 pasen necesariamente por el usuario 3, el cual podrá operar con ellos como guste, observando las tramas que recibe y reenviando, alterándolas o incluso eliminándolas.

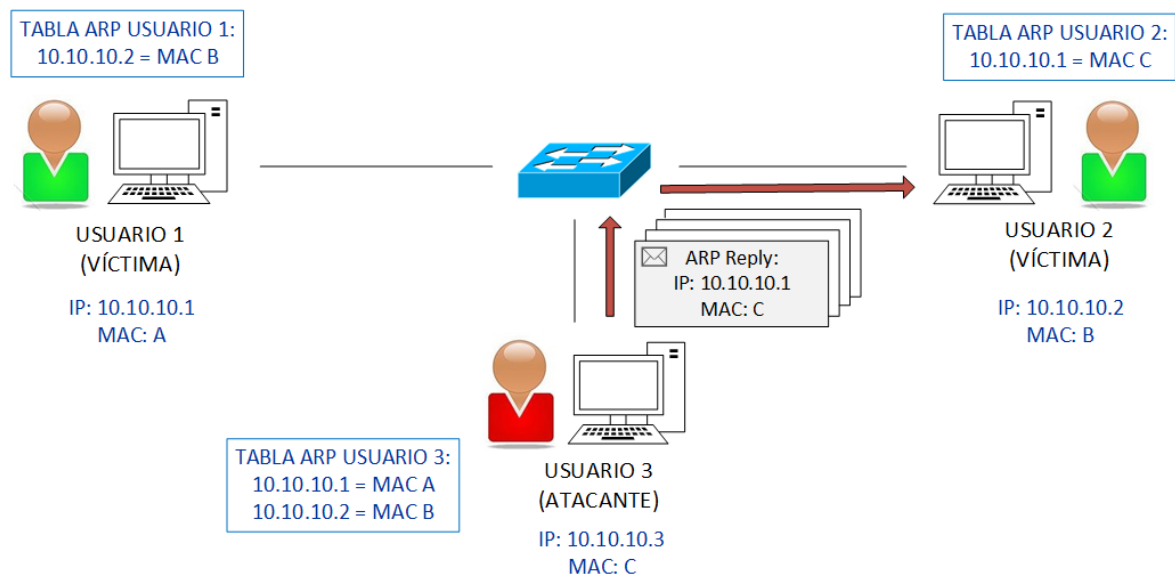


Figura 4.2.- Envío de mensajes ARP *Reply* fraudulentos

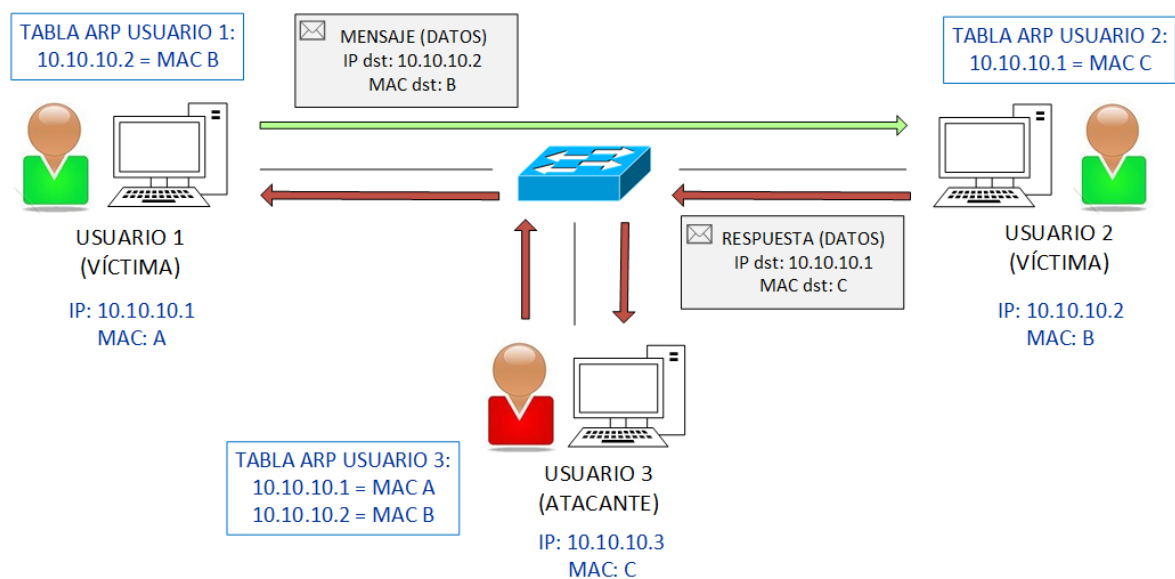
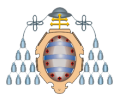


Figura 4.3.- ARP *Spoofing*



4.1.2.- STP

Protocolo orientado a solucionar el problema de los bucles en capa 2. Actualmente viene habilitado en todos los switches de Cisco por defecto, pero esto puede acarrear algunos problemas si no se realiza una correcta configuración.

Gracias a la designación de un switch raíz (*root*) se asegura que solo existe una ruta lógica entre todos los destinos de la LAN, para ello se bloquean aquellos puertos susceptibles de generar bucles (puertos no denominados como raíz). STP bloquea el mínimo número de enlaces imprescindibles para obtener una topología lógica sin bucles. Para la determinación del *root* se tienen en cuenta dos factores: la prioridad del switch (por defecto es 32768) y la dirección MAC del mismo. A la hora de determinar el *root* hay que asegurarse que este sea el más conveniente, teniendo en cuenta para ello su posición en la red, su fiabilidad y su rapidez. Asimismo, es recomendable tener un switch de respaldo por si el *root* falla, como se puede observar en la figura 4.4. Si en esta figura se activase cualquiera de los enlaces que actualmente están bloqueados, se generaría un bucle. Por este motivo, y únicamente por este, esos enlaces deben quedar bloqueados.

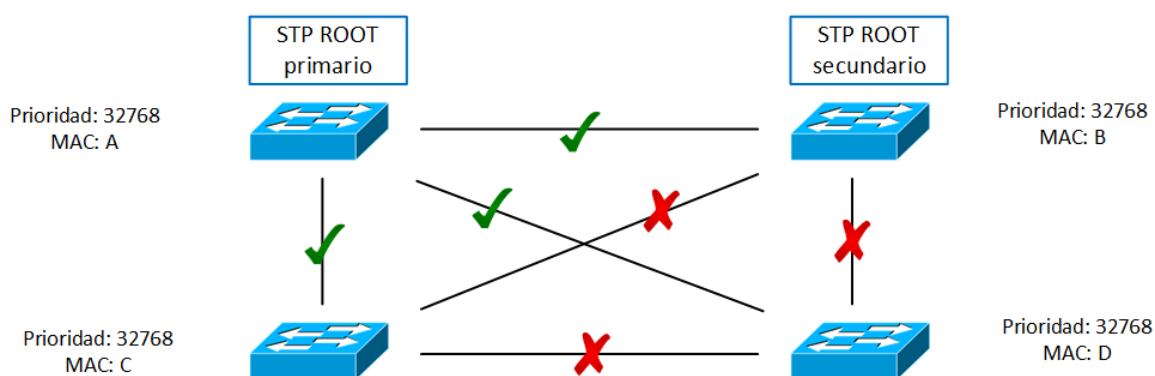


Figura 4.4.- Configuración inicial STP

Las unidades de datos del protocolo se denominan BPDU. Estos paquetes contienen información de los switches presentes en la red, es decir, el *Bridge Identifier* (BID: prioridad + MAC del elemento) de cada uno, esto permite la elección del *root*. Aquel que posea el BID de menor valor será el nuevo *root* de la red. Una vez definido todos

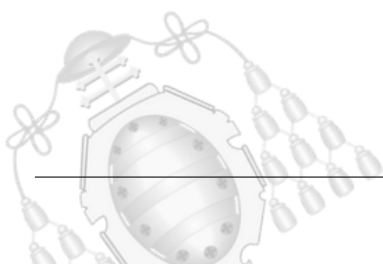


los switches de la red, excepto él mismo, determinan un puerto como puerto raíz siendo este el de menor coste hacia el *root* [13] [14].

Una mala configuración de este protocolo podría acarrear una denegación de servicio si el atacante tiene acceso a la red [16]. Esto es posible ya que se obliga a los diversos dispositivos que participan en STP a recalcular todas las rutas de las que disponen, provocando así una situación de inestabilidad y malgasto de recursos. Si la situación se llevase al extremo podrían producirse bucles en la red e incluso que está se venga abajo. Es un ataque muy simple ya que solo requiere el envío de paquetes BPDU generados de forma aleatoria, de esta forma se tendría una situación análoga a conectar múltiples de dispositivos a la red casi simultáneamente y que estos participasen en el protocolo STP.

Otra posibilidad, como se muestra en las figuras 4.5 y 4.6, es aquella en la que el atacante tiene acceso directo a la red y se hace con el rol de *root* [15]. Para ello envía un paquete BPDU haciendo creer al resto de elementos de la red que es uno más y quiere ser partícipe de STP; es necesario que el valor del BID indicado en dicho paquete sea inferior al del resto. Para el ejemplo mostrado en este documento se establece el caso más extremo, aquel en el que el valor de prioridad es 0. De esta forma si el resto de switches están configurados por defecto se asegura que consigue el rol deseado ya que tendrá el menor BID posible. Sin embargo, este ataque podría hacerse de forma más sutil si se conocen los BID del resto de elementos de la red, el atacante determina que su identificador sea ligeramente inferior al del *root* actual consiguiendo así que el cambio sea prácticamente inapreciable por el administrador. Si el atacante se hiciese con el papel de *root* controlaría todo el tráfico de la red, por lo que podría ver, alterar o incluso eliminar la información que circula por la misma.

Para la realización de estos ataques, explotando así la mala configuración de la LAN, es común hacer uso de Yersinia. Sin embargo, también es posible generar los paquetes con Scapy y enviarlos por la red.



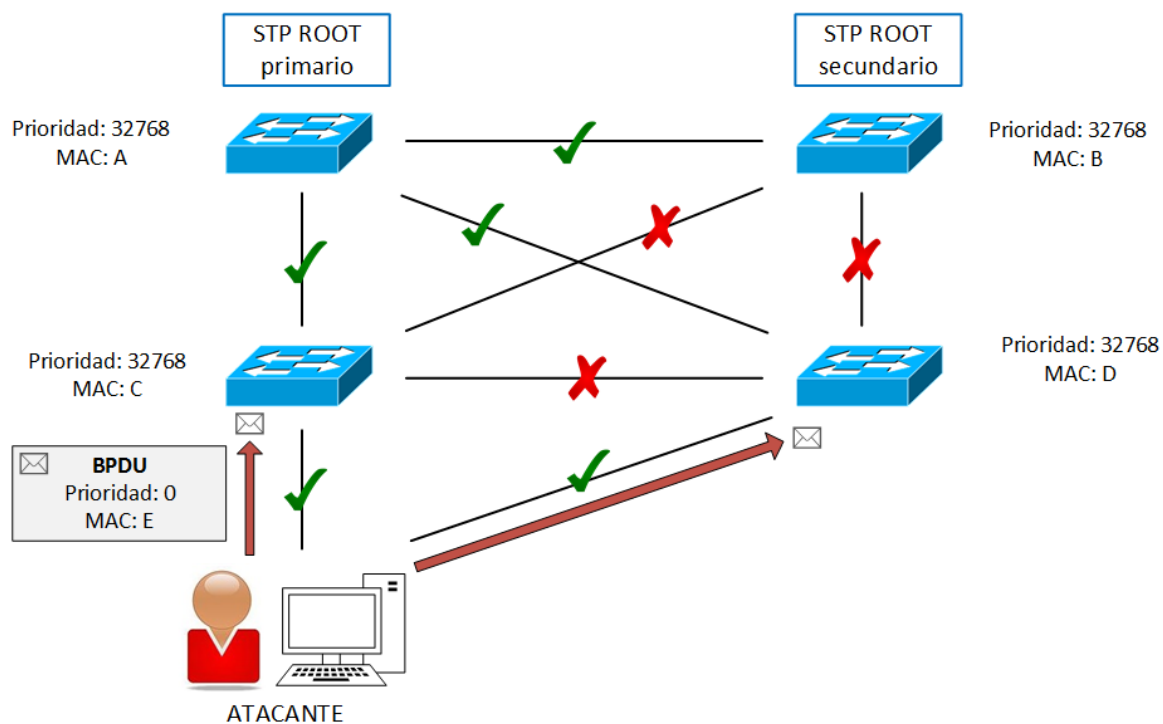
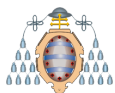


Figura 4.5.- Envío BPDU fraudulento

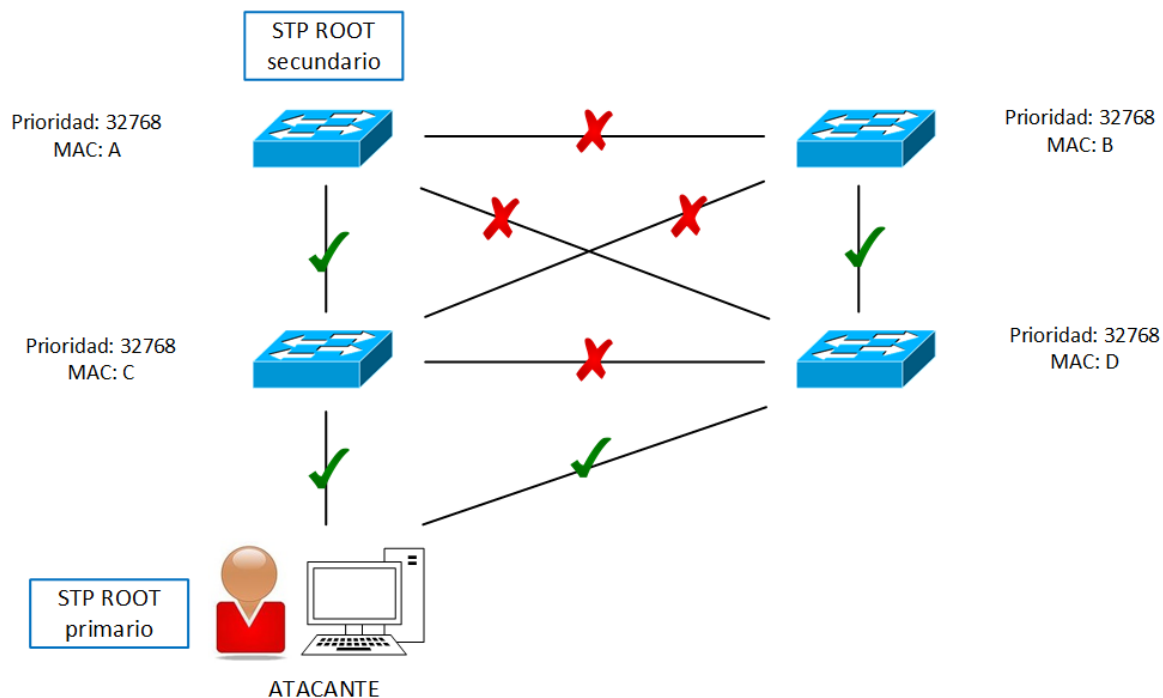
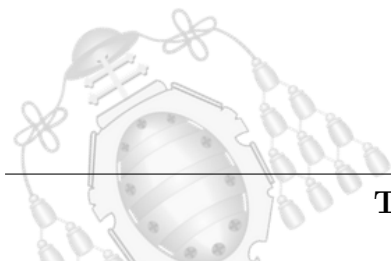


Figura 4.6.- Cambio de *root*





4.2.- Cisco

En este apartado se explican las principales vulnerabilidades de algunos protocolos específicos de Cisco, empresa dedicada al diseño y venta de equipos de telecomunicación, como son CDP, DTP y VTP. Los dos primeros vienen activos por defecto, por lo que se incrementa la posibilidad de sufrir ataques si no se configuran correctamente.

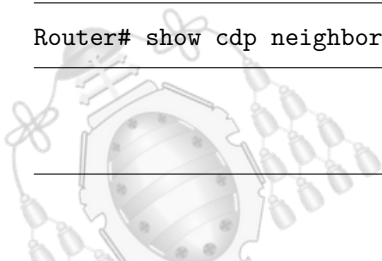
4.2.1.- CDP

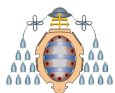
CDP permite la conexión entre los elementos que componen la red, esto es posible porque descubre los dispositivos Cisco que se tiene conectados directamente. Dicho de otra forma, es un protocolo extremo a extremo. Proporciona simplicidad en la configuración de las conexiones ya que se posibilita hacerlo de forma automática. Este protocolo, que opera en la capa 2 del modelo OSI, suele venir activo por defecto en dispositivos Cisco, como los switches y bridges. Si bien no fuese así se podría configurar en todos ellos. Tiene como finalidad compartir información periódicamente cada 60 segundos, sin cifrar, del hardware y software de los equipos adyacentes dentro de una red. Entre la información que contienen estos paquetes se encuentra:

- Nombre del dispositivo (Hostname)
- Imagen del OS
- Tipo y modelo del dispositivo
- Dirección IP
- Interfaz que genera los mensajes CDP
- VLAN nativa

La información contenida por los mensajes es fácilmente extensible debido al uso del esquema de codificación TLV (*Type-Length-Value*). Todo lo que se recibe de dispositivos adyacentes se almacena o actualiza en una tabla que puede ser consultada posteriormente con el comando [20]:

```
Router# show cdp neighbors
```





A continuación, en la figura 4.7 se muestra el resultado obtenido tras ejecutar este comando en un ordenador conectado directamente a un switch.

```
Switch#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge
                  S - Switch, H - Host, I - IGMP, r - Repeater, P - Phone
Device ID      Local Intrfce    Holdtme    Capability    Platform    Port ID
Switch         Fas 0/24        170        S             2950        Fas 0/24
Switch#show cdp neighbors detail

Device ID: Switch
Entry address(es):
Platform: cisco 2950, Capabilities: Switch
Interface: FastEthernet0/24, Port ID (outgoing port): FastEthernet0/24
Holdtime: 166

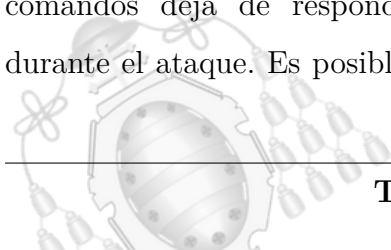
Version :
Cisco Internetwork Operating System Software
IOS (tm) C2950 Software (C2950-I6Q4L2-M), Version 12.1(22)EA4, RELEASE
SOFTWARE(fc1)
Copyright (c) 1986-2005 by cisco Systems, Inc.
Compiled Wed 18-May-05 22:31 by jharirba

advertisement version: 2
Duplex: full
```

Figura 4.7.- Resultado comando *show cdp neighbors*

No obstante, esta información es tan útil como peligrosa, debido a que se almacena información valiosa de la red al alcance de cualquier persona, pudiendo ser esta un atacante. Asimismo, como bien se indicaba antes, los mensajes CDP no utilizan ningún tipo de autenticación, sino que se envían en texto plano y con un formato de paquete definido y explicado en la web de Cisco, lo que hace que sea muy vulnerable a posibles ataques.

El principal tipo de ataque que podría sufrir la red es denegación de servicio (*DoS*) [21] como se puede ver en la figura 4.8. Una vez se conoce el formato de los paquetes que se envían a través de la red, es tan sencillo como enviar múltiples mensajes CDP con diferentes identificadores y direcciones MAC de origen, provocando así un consumo de CPU muy elevado y colapsando el dispositivo. Si el elemento de red que sufre el ataque recibe una cantidad excesiva de paquetes podría quedarse sin memoria y dejar de funcionar correctamente, obligando al administrador a resetearlo para recuperar la actividad normal del mismo, ya que la línea de comandos deja de responder haciendo imposible la desactivación del protocolo durante el ataque. Es posible realizarlo haciendo uso de la herramienta ya nombrada



con anterioridad, Yersinia.

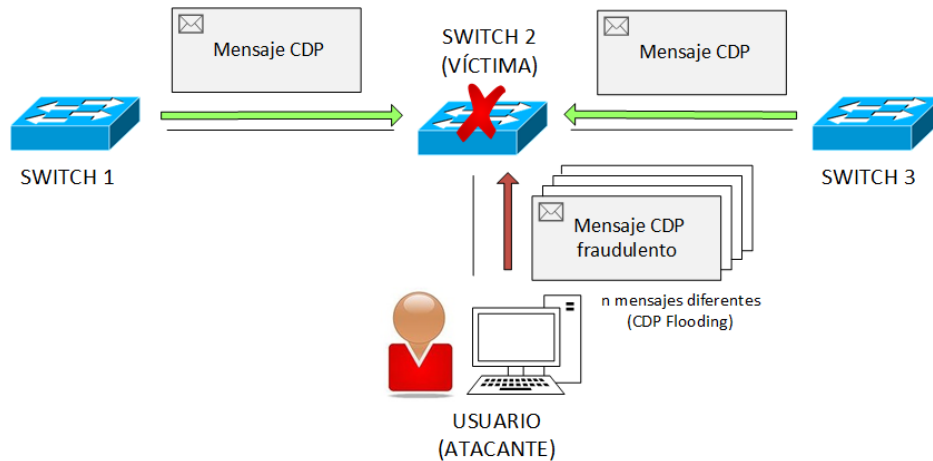


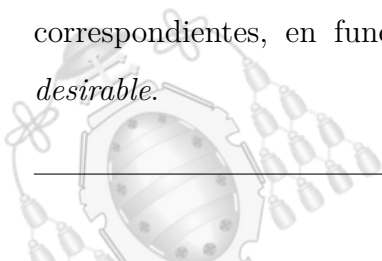
Figura 4.8.- Ataque CDP

Por otro lado, el atacante podría capturar los mensajes CDP generados por la red, haciendo uso de un sniffer, para conocer así cuales son los diferentes sistemas operativos que usan los dispositivos y poder explotar las vulnerabilidades de los mismos.

Si el atacante cuenta con acceso al router vía Telnet o SNMP se puede explotar con mayor eficacia la información proporcionada por el protocolo CDP, ya que se conseguiría conocer la topología de la red en gran detalle (modelos de los dispositivos, direcciones IP, sistemas operativos). Gracias a la información extraída de la red, y a una lista de vulnerabilidades de los diversos sistemas operativos instalados en los dispositivos podría llegar a hacerse un ataque muy efectivo a dicha red.

4.2.2.- DTP

DTP permite la configuración automática de los puertos troncales. Tiene como cometido la gestión automática del estado de un enlace entre dispositivos, resultando en una configuración del puerto en modo troncal o acceso. Existen dos configuraciones por defecto, ambas necesitan que los dos dispositivos tengan activo DTP en los puertos correspondientes, en función del dispositivo que se use: *dynamic auto* y *dynamic desirable*.





- *Dynamic auto*: Modo por defecto en los switches Catalyst 2960 de Cisco. Envía mensajes DTP de forma periódica indicando que soporta la opción de enlace troncal, el puerto espera pasivamente por mensajes hasta que se conecta un dispositivo al otro extremo. Si se conecta un puerto en modo *on* o *dynamic desirable*, el enlace se establece como troncal (Figura 4.9); si por el contrario el puerto remoto se encuentra en modo *dynamic auto*, el enlace se establece como de acceso (Figura 4.10), precisando de configuración adicional.

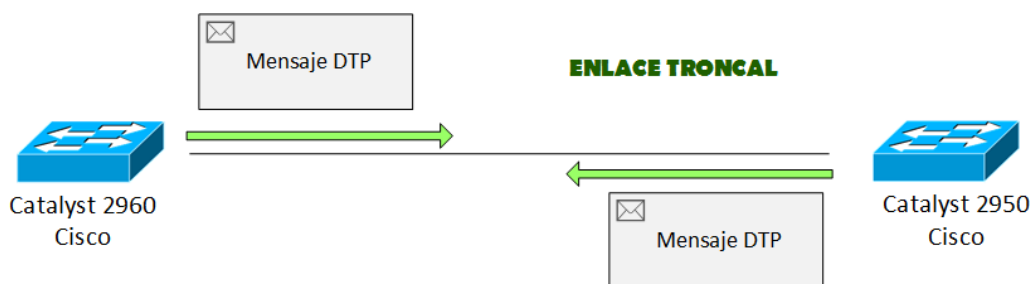


Figura 4.9.- Configuración de un switch Catalyst 2960 y un Catalyst 2950

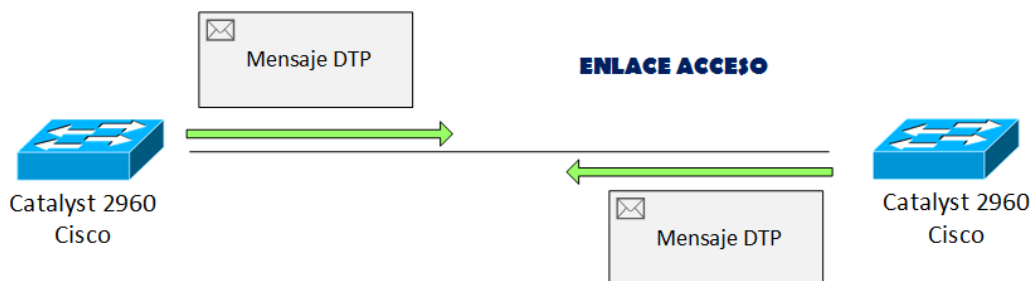
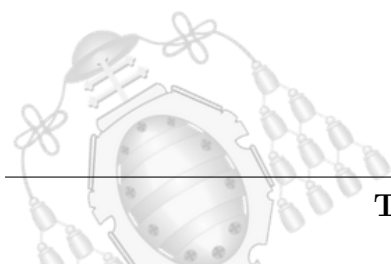


Figura 4.10.- Configuración de 2 switches Catalyst 2960

- *Dynamic desirable*: Modo por defecto en los switches Catalyst 2950 de Cisco. Se intenta convertir el enlace en troncal. De esta forma, sea cual sea el estado del puerto remoto (*dynamic auto*, *dynamic desirable* u *on*) se establecerá el enlace como troncal (Ejemplo en Figura 4.11).



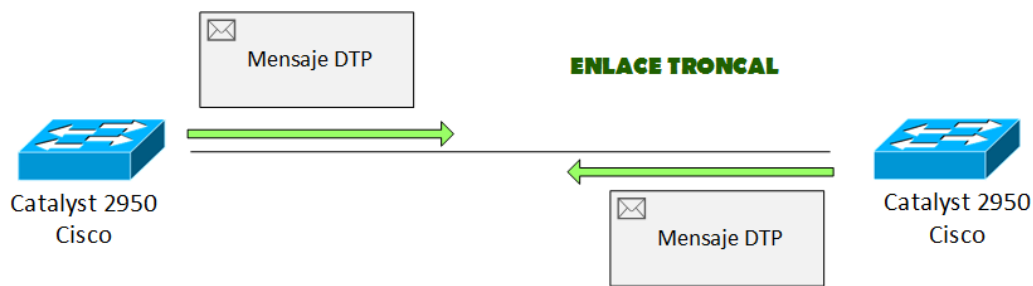


Figura 4.11.- Configuración de 2 switches Catalyst 2950

DTP viene activo por defecto en los elementos de la marca Cisco, proporcionando al administrador una mayor facilidad a la hora de configurar la red [24] [25].

No obstante, el uso de este protocolo, sin autenticación y presente en todos los puertos del switch, hace que la red sea vulnerable. Como se puede ver en la figura 4.12, si el administrador de la red dejase la configuración por defecto de este protocolo se podría sufrir un ataque del tipo *VLAN Hopping - Switch Spoofing*, es decir, que una persona no legítima forzase un enlace a convertirse en troncal [26].

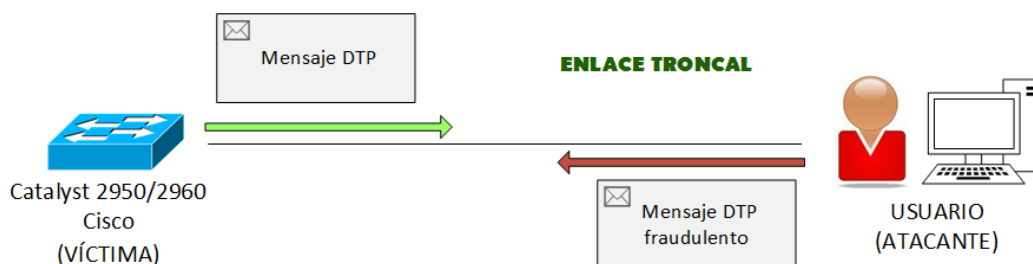
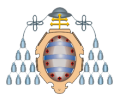


Figura 4.12.- Envío mensaje DTP fraudulento

El procedimiento para ello consiste en el envío de mensajes DTP al switch (comprobando previamente que el protocolo está activo en el mismo) haciéndole creer que el usuario ilegítimo es un dispositivo fiable de la red. Para llevar a cabo esta negociación, primero es necesario el envío de tres paquetes, uno por segundo, indicando: estado del enlace troncal y tipo de encapsulamiento; a continuación, se envía un paquete DTP cada 30 segundos. Si el puerto al que se conecta estuviese en alguno de los siguientes modos: *dynamic auto*, *dynamic desirable* u *on*, el enlace se establecería como troncal, logrando así el objetivo deseado.



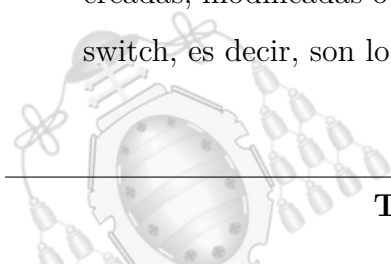
En caso de que se lograra persuadir al switch, el atacante conseguiría acceso a todas las VLANs de la red, es decir, a todo el tráfico de difusión y parte del de unicast que circula por la misma. Haciendo uso de este privilegio se pueden llevar a cabo ataques como *man in the middle* o acceder a los dispositivos de la red mediante fuerza bruta.

El ataque mencionado puede realizarse mediante la conexión de un ordenador a dicho puerto, o conectando un switch ilegítimo que fuerce el establecimiento del enlace como troncal. En caso de hacerse desde un ordenador se puede llevar a cabo de forma sencilla haciendo uso del software Yersinia.

4.2.3.- VTP

VTP es el protocolo usado para administrar y configurar las VLANs en los dispositivos de Cisco, de forma simplificada y centralizada. Permite borrar, crear y renombrar las VLANs de múltiples switches, todos ellos pertenecientes al mismo dominio. Sin la ayuda de este protocolo, el administrador debe configurar cada VLAN de forma manual en todos los switches que componen el dominio, lo cual puede ser complejo y conducir a fallos. Existen 3 modos de operación VTP: servidor, cliente y transparente.

- Servidor: Puede crear, borrar y modificar cualquiera de las VLANs que conforman el dominio. Todas las VLANs configuradas que tiene el servidor son anunciadas y sincronizadas con el resto de switches del dominio a través de los enlaces troncales. Es el modo por defecto que utilizan los switches.
- Cliente: No puede crear, borrar ni modificar información relacionada con las VLANs, pero sí que sincroniza su información con aquella que recibe de los anuncios de los servidores de su dominio. Esta información es eliminada en el momento que se reinicia el switch, pero mientras esté activo se conserva.
- Transparente: No procesa los anuncios VTP que recibe, simplemente reenvía los mensajes al resto de switches que componen el dominio. Todas las VLANs creadas, modificadas o borradas en un switch transparente solo las percibe dicho switch, es decir, son locales ya que no pertenecen al dominio.





Si se quiere que dos dispositivos que usan VTP puedan compartir información, es necesario que pertenezcan al mismo dominio. De no ser así se descartan los mensajes que se reciben.

Hay dos tipos de mensajes VTP que permiten que todo funcione correctamente:

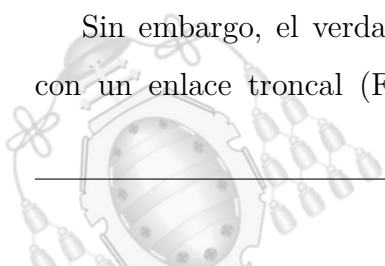
- Anuncios de resumen: Contiene nombre de dominio, número de revisión y otros detalles. Se envían periódicamente cada 5 minutos o cuando hay un cambio de configuración, haciendo uso para ello de la VLAN 1 y una dirección *multicast*. Tiene como finalidad la sincronización de la información de las VLANs dentro del dominio.
- Anuncios de subconjunto: Contiene información de las VLANs. Se envía junto con un anuncio de resumen cuando hay algún cambio en la configuración. Tiene como finalidad la actualización de la información relacionada con las VLANs.

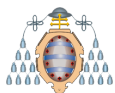
Para controlar los anuncios que se reciben y evitar confusiones en la configuración se hace uso del número de revisión. Si el número de revisión del anuncio que se recibe es mayor, entonces se aplica la configuración. De lo contrario se ignora [29] [30].

A pesar de las múltiples ventajas presentadas por este protocolo, entre ellas la sencillez de configuración del mismo, también se encuentran algunos problemas de seguridad.

En caso de estar usando la versión 1 o 2 del protocolo se podría lograr que un switch configurado como cliente, alterase la configuración del resto. Esto se debe a que el cliente, a pesar de escuchar anuncios del servidor, también envía anuncios VTP. Basta con enviar un mensaje resumen con un número de revisión superior al último recibido. Este iría seguido de un mensaje de subconjunto con la configuración actualizada de las VLANs. De esta forma, teniendo acceso a un troncal, bastaría con esperar por un mensaje VTP para generar a partir del mismo un anuncio de resumen fraudulento con su correspondiente anuncio de subconjunto.

Sin embargo, el verdadero problema sería que un atacante se conectase a la red con un enlace troncal (Figuras 4.13 y 4.14). Mediante el envío de mensajes VTP





fraudulentos podría provocar una denegación de servicio borrando una o varias de las VLANs configuradas en la red [31] [32]. Los puertos permanecerían configurados para las VLANs borradas, por lo que quedarían inoperativos provocando una denegación de servicio (Figura 4.15).

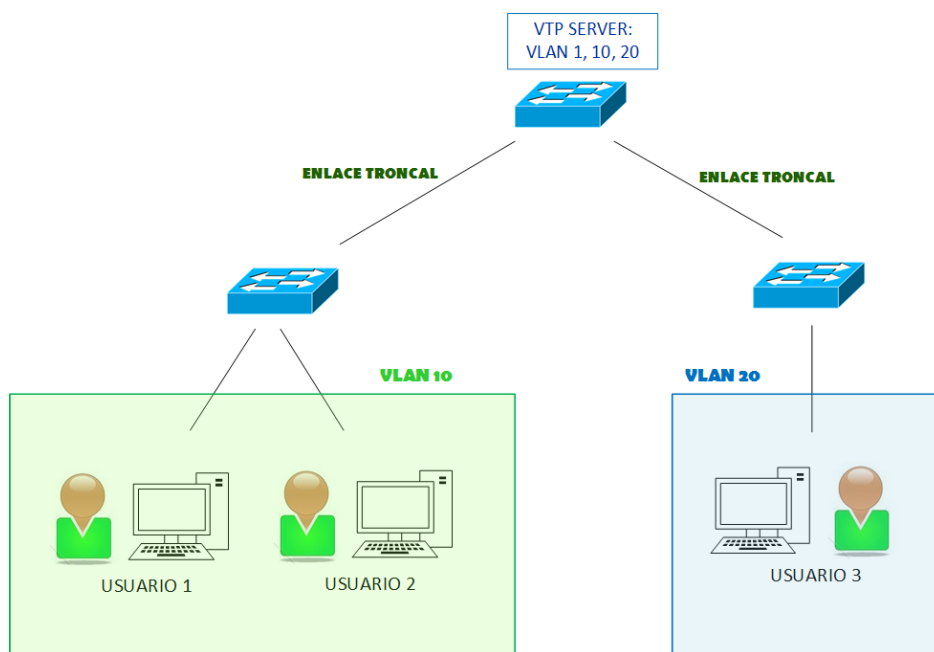


Figura 4.13.- Configuración inicial VTP

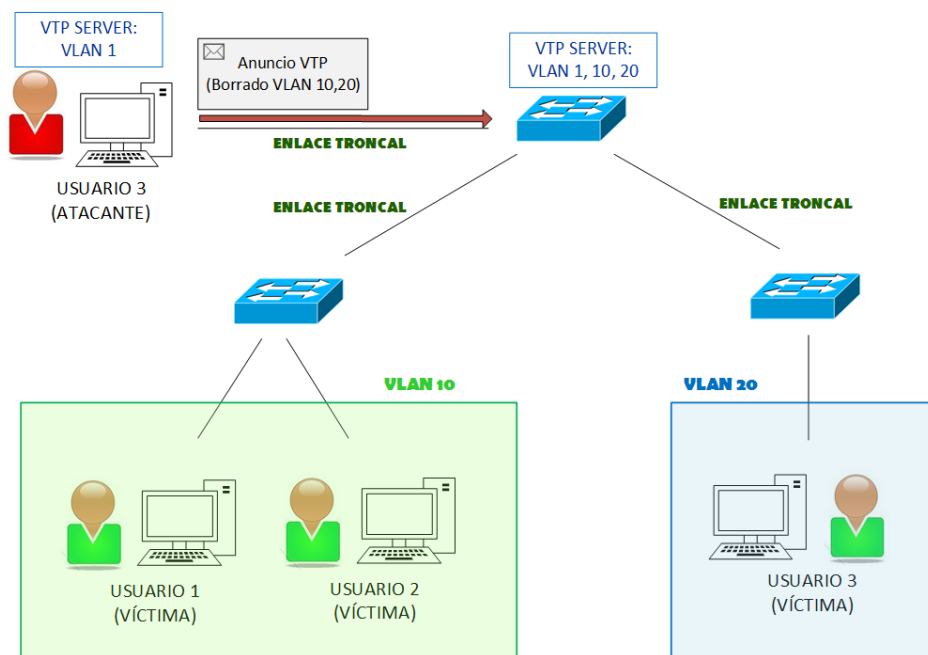
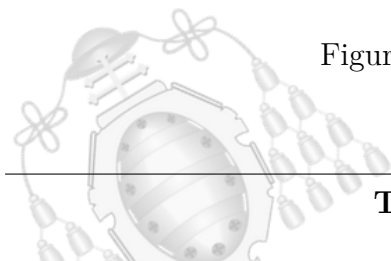


Figura 4.14.- Anuncio VTP fraudulento



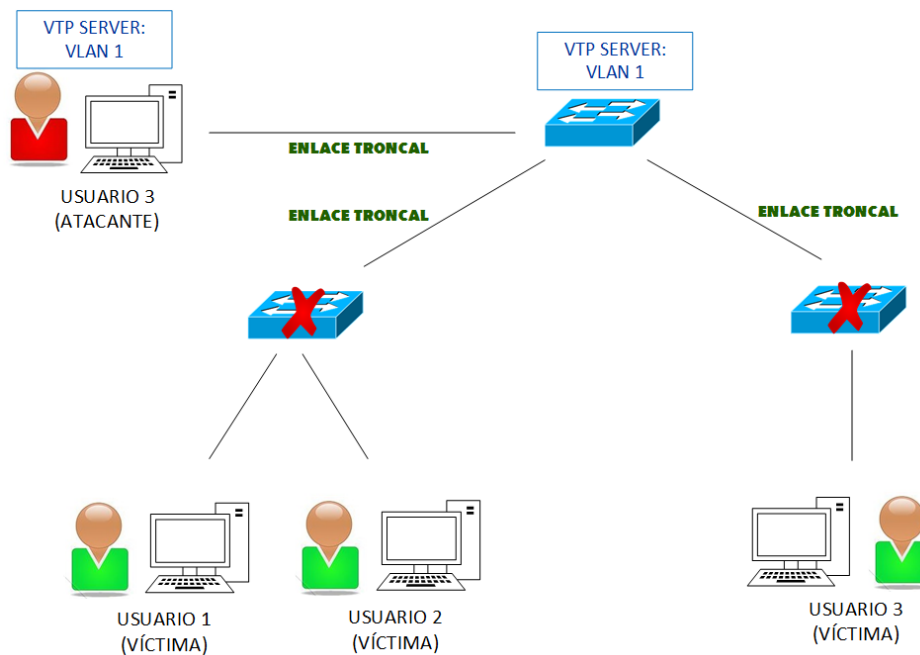
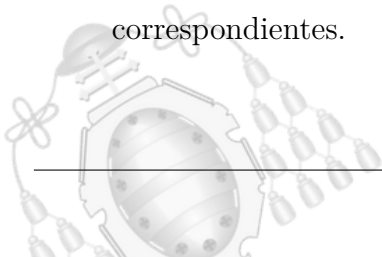


Figura 4.15.- Resultado ataque VTP

4.2.4.- Problema *Double Tagging*

Todos estos protocolos tienen en común el uso por defecto de la VLAN 1 como VLAN nativa. Esto hace que el uso de los mismos provoque que la red sea vulnerable a ataques del tipo *VLAN Hopping - Double Tagging*, por lo que el atacante podría obtener acceso a una VLAN no autorizada explotando dicha configuración. Para ello es necesario conectarse a una interfaz de tipo troncal y enviar un mensaje que contenga dos etiquetas [34] [35] como se muestra en la figura 4.16:

- La primera etiqueta indica la VLAN nativa de dicho troncal (es decir, la 1), esta es leída por el primer switch y posteriormente eliminada por el puerto troncal del switch por donde se recibe la trama.
- Una vez eliminada la primera etiqueta se reenvía por el resto de sus troncales sin dicha etiqueta, ya que por defecto la VLAN nativa no se etiqueta. La segunda etiqueta contiene la VLAN de destino, es decir, la de la víctima. Esta es leída por el segundo switch, el cual entregará el paquete a los destinatarios correspondientes.



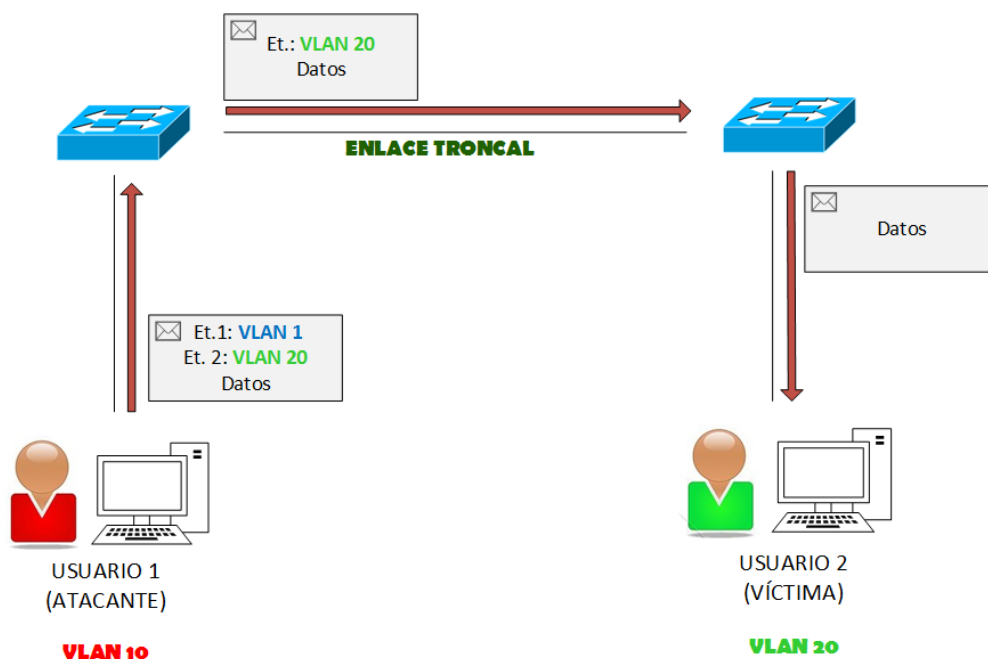
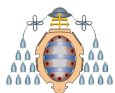
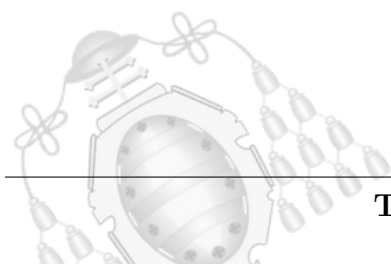


Figura 4.16.- Ataque *Double Tagging*

De forma tan sencilla se puede llegar a acceder a una VLAN no autorizada, simplemente haciendo uso de la configuración predeterminada de los switches. Su realización es posible tanto desde un puerto troncal como uno que no lo es, en este último caso es necesario que esté asignado a la VLAN nativa, lo cual requiere una situación muy particular como puede ser aquella en la que voz y datos llegan al mismo puerto. En ese caso, la voz usaría una VLAN concreta y los datos la nativa. Si en cambio se tiene un puerto de acceso configurado para la VLAN nativa, el ataque no funcionaría ya que los switches descartan el tráfico etiquetado que reciben en los puertos de acceso.

En caso de que se cambiase la VLAN nativa, quitando la que viene por defecto y estableciendo una nueva, y el atacante conociese, descubriese o adivinase cual es el identificador que se ha puesto, se podría sufrir este ataque igualmente. Esto se debe a que el atacante explota dicha VLAN, independientemente del número que la identifique. La principal diferencia es que si se deja el valor predeterminado las probabilidades de sufrir el ataque aumentan de forma notable.





4.3.- Capa de red (Capa 3 según modelo OSI)

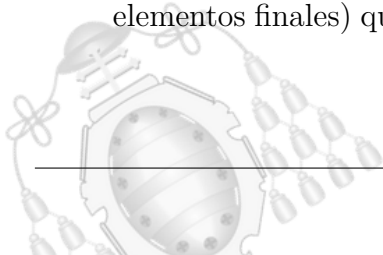
Esta capa, situada entre la de enlace y la de transporte, proporciona conectividad y determina la ruta entre dos elementos por la cual ha de enviarse la información. Su principal función es conseguir que los mensajes lleguen desde el origen hasta el destino correctamente, sin necesidad de que estos tengan una conexión directa. En este apartado se ha decidido añadir dos protocolos de enrutamiento muy populares, a la par que vulnerables: RIP y OSPF.

4.3.1.- RIP

RIP es un protocolo de encaminamiento interno, utilizado por los routers, que permite intercambiar información acerca de las redes a las que se encuentran conectados. Usa el número de saltos como métrica, por lo que no se tiene en cuenta las condiciones del enlace (por ejemplo: ancho de banda, retardo...), haciendo que sea simple a la par que ineficiente en muchas situaciones. Para determinar el camino más corto para alcanzar el destino hace uso del algoritmo de encaminamiento *Bellman Ford*, basado en el vector distancia que calcula la ruta óptima a partir del número de saltos. Si el número de saltos para alcanzar la red deseada supera los 15, se considera inalcanzable. Actualmente existen tres versiones del protocolo, no obstante, la más extendida es la versión 2 debido a las ventajas presentadas respecto a la primera (subredes, CIDR, VLSM, autenticación, etc) [39] [40]. La versión 3 del protocolo no se usa tanto porque es para IPv6, no demasiado extendido aún.

Existen dos modos de funcionamiento dentro del protocolo:

- Activo: Envía toda o parte de su tabla de rutas a sus vecinos en forma de respuesta, ya sea por una actualización periódica, una respuesta a una pregunta o por un cambio en la topología. Lo habitual es que solamente los routers operen en este modo.
- Pasivo: Se establece en aquellas interfaces (normalmente, las que conectan con elementos finales) que se desea que no sean partícipes del protocolo RIP, es decir,





no se difunden actualizaciones por las mismas. En cambio, la red a la que están conectadas sí que se anuncia.

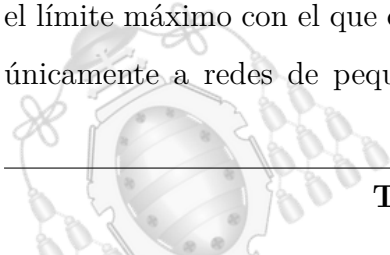
RIP cuenta con dos tipos de paquetes:

- *Request*: Son enviados para solicitar toda o parte de una tabla de rutas. Normalmente se utilizan cuando se conecta un nuevo router y se quiere completar la tabla de rutas de forma rápida sin necesidad de esperar por los mensajes que se envían de forma periódica.
- *Response*: Puede ser generado por alguna de las siguientes razones:
 - Como respuesta a un mensaje de tipo *Request*
 - Actualización periódica
 - Actualización por un cambio en la topología de la red

Existen tres tipos de temporizadores para controlar los mensajes de RIP:

- Temporizador periódico: Controla los mensajes RIP que se envían de forma periódica, que son enviados cada 30 segundos. En caso de producirse algún cambio se informa antes de que expire el temporizador.
- Temporizador de caducidad: Determina el tiempo que ha de pasar antes de considerar una red como inalcanzable, es decir, poner como distancia a la misma 16 saltos. En este caso, el margen de tiempo establecido para recibir actualizaciones es de 180 segundos.
- Temporizador de colección de basura: Controla el tiempo que pasa desde que una ruta se considera inválida hasta que se elimina de la tabla de rutas. El valor predeterminado es 240 segundos, es decir, 60 segundos después de la expiración del temporizador de caducidad.

RIP se caracteriza por su fácil configuración, además de que es soportado por múltiples fabricantes, lo que hace que sea más habitual su uso. Sin embargo, cuenta con diversas desventajas, entre ellas se encuentra la forma de determinar la mejor métrica, o el límite máximo con el que cuenta la misma, restringiendo así el uso de este protocolo únicamente a redes de pequeño o mediano tamaño [38]. Asimismo, es vulnerable a



múltiples ataques de seguridad, los cuales varían en función de la versión del protocolo que se utilice.

Uno de los ataques más peligrosos que se pueden sufrir cuando se hace uso de este protocolo es *RIP Poisoning* [42]. Para su realización el atacante intercepta mensajes legítimos del protocolo que posteriormente utiliza para reenviar una actualización fraudulenta que altere la tabla de rutas, haciéndose pasar por un elemento de la red. Como se puede ver en la figura 4.17, partiendo de una situación de funcionamiento normal, solamente es necesario enviar un mensaje que informe de que la distancia al destino es menor a través del atacante (Figuras 4.18 y 4.19), consiguiendo así que la ruta establecida inicialmente se actualice y se comience a usar la del atacante. Una vez alterada la ruta hacia una red (Figura 4.20) se puede llevar a cabo uno de los ataques más comunes en el ámbito de la ciberseguridad: *man in the middle*. De esta forma, el atacante podría ver o incluso alterar la información que se envía a una determinada red, pudiendo realizar incluso un ataque del tipo *session hijacking*. Uno de los programas que facilitan su realización es *Loki*.

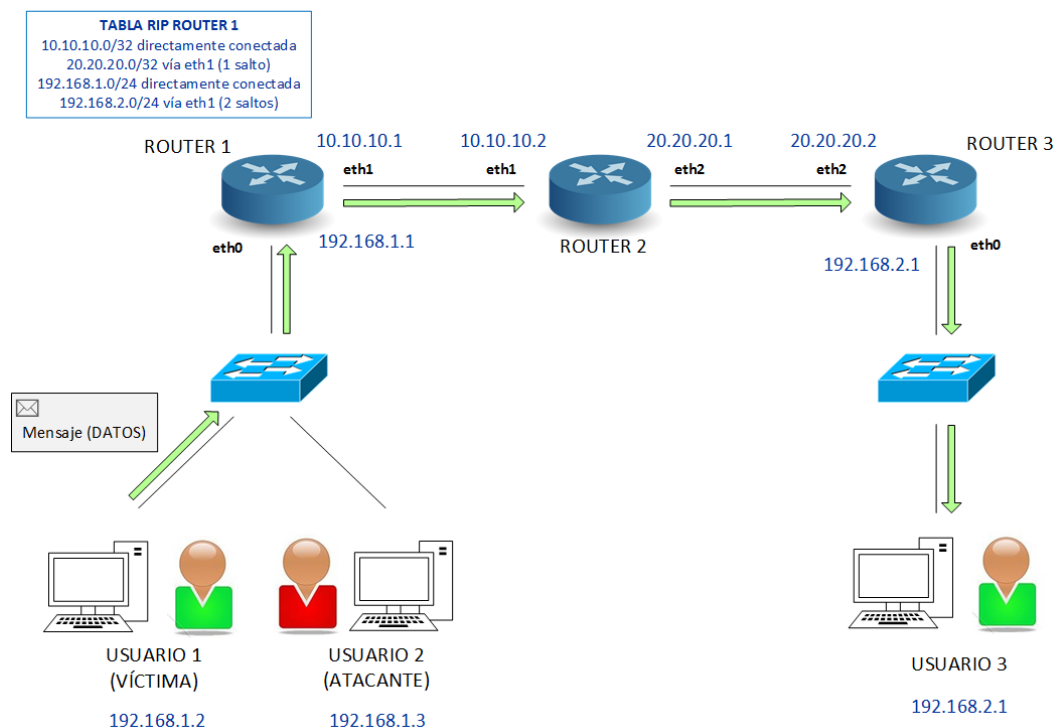
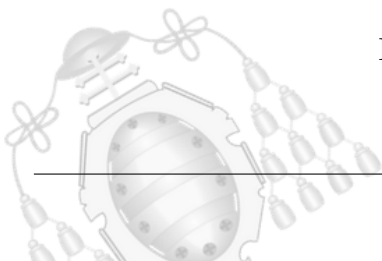


Figura 4.17.- Situación normal RIP



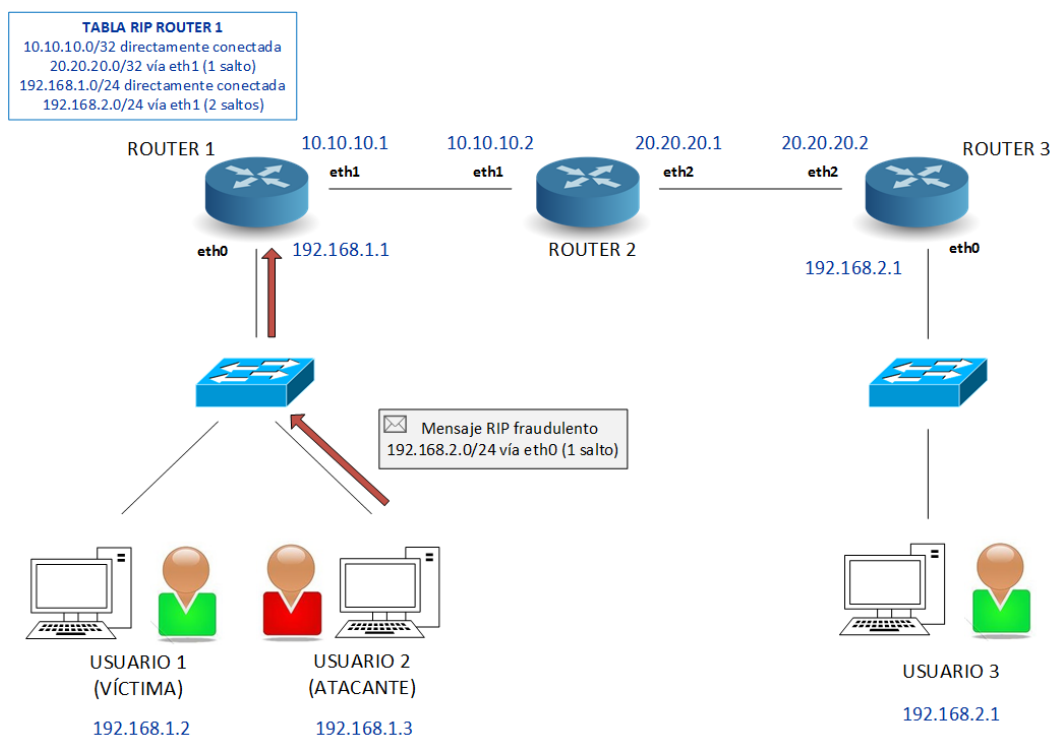
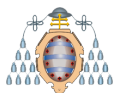


Figura 4.18.- Envío mensaje RIP fraudulento

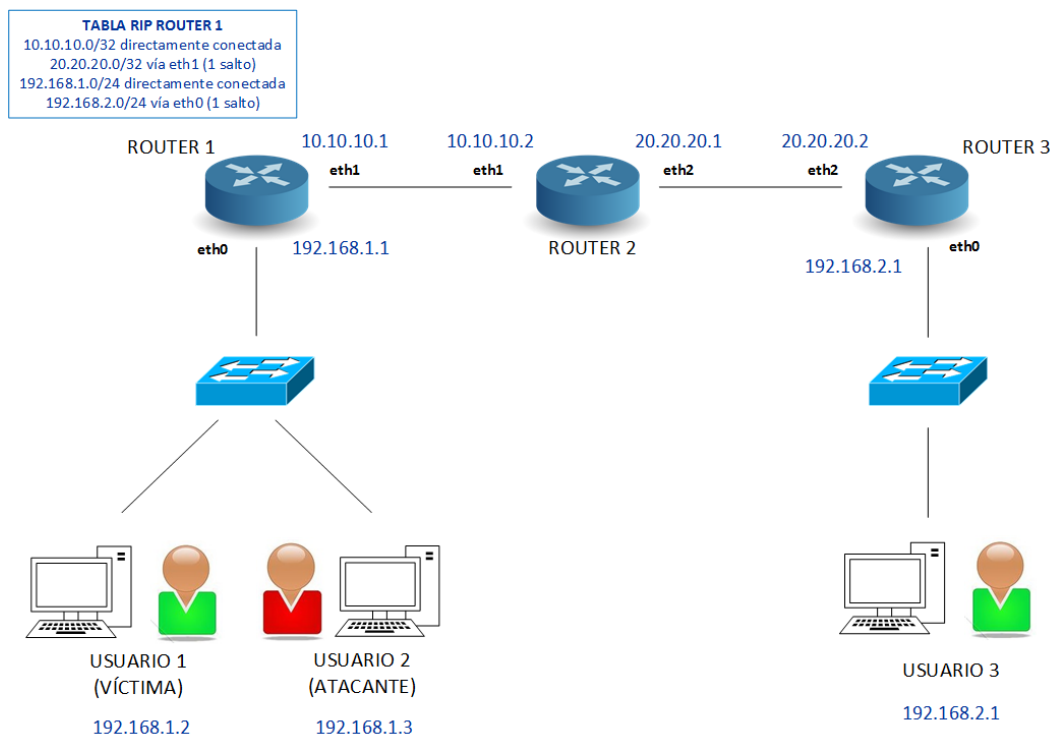
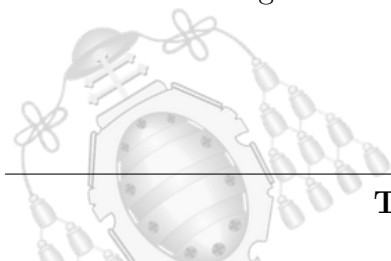


Figura 4.19.- Alteración tabla rutas (RIP *Poisoning*)



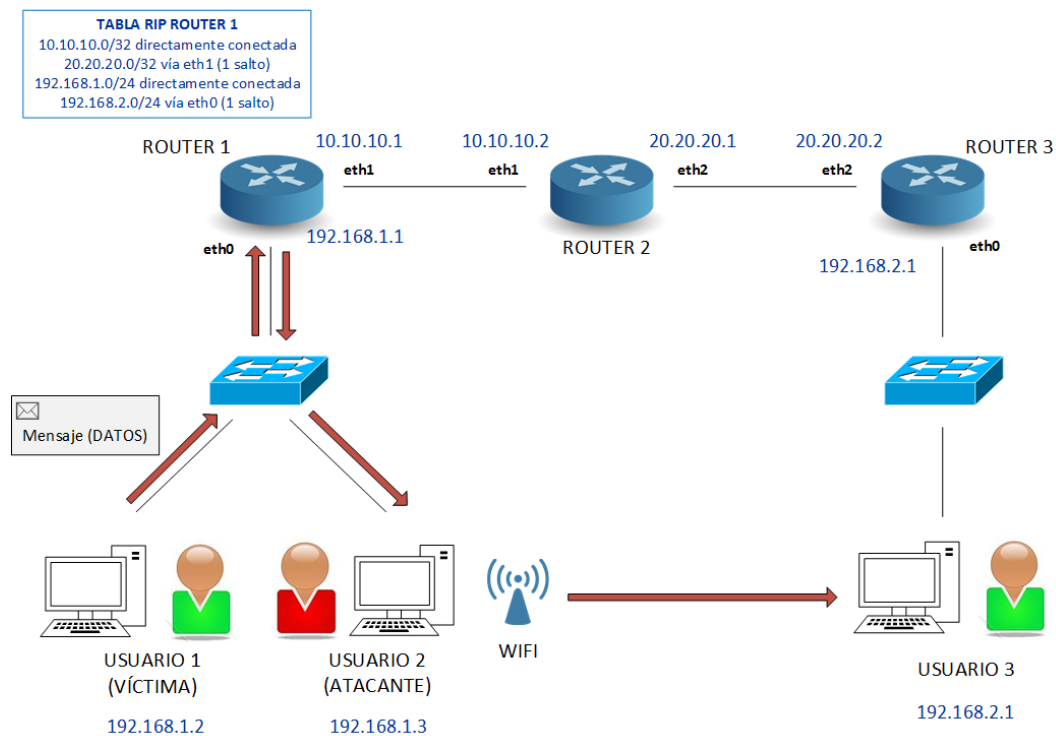
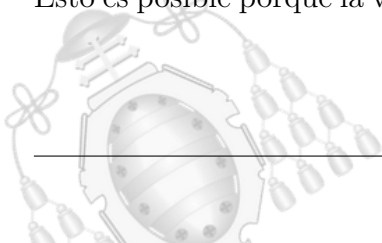


Figura 4.20.- Ataque *man in the middle*

Aunque este ataque se puede sufrir con las dos primeras versiones del protocolo, en la versión 2 se podría llegar a mitigar el alcance del mismo ya que los mensajes cuentan con la ventaja de que se pueden autenticar (para ello se usa MD5). Así, se consigue que el usuario necesite conocer la contraseña para llevar a cabo dicho ataque. Si la contraseña no coincide con la establecida, se ignoran los mensajes recibidos. El ataque para la versión 2 del protocolo también podría realizarse con *Loki*, ya que este software cuenta con la posibilidad de realizar un ataque *MD5 auth cracking*, facilitando así el descifrado de la contraseña.

Por otro lado, también se podría sufrir una denegación de servicio distribuida (*DDoS*) cuando se hace uso de la versión 1 del protocolo. *DDoS Reflection* provoca que el dispositivo se sature debido a que recibe más peticiones de las que puede gestionar. Estas pueden venir todas de un mismo elemento o de varios. Para este último caso se hace uso de terceros encargados de realizar las solicitudes indicadas por el atacante [41]. Esto es posible porque la versión 1 del protocolo no utiliza ningún tipo de autenticación,





lo que hace que sea vulnerable a la recepción de una cantidad de paquetes de tipo *Request* mayor a la que es capaz de manejar.

4.3.2.- OSPF

OSPF es un protocolo de encaminamiento interior al igual que RIP. El algoritmo de encaminamiento utilizado permite calcular ruta óptima entre dos nodos de un sistema autónomo (conjunto de redes y routers que se encuentran bajo el mismo control administrativo), haciendo uso del algoritmo de *Dijkstra*. Para el cálculo de la métrica se tiene en cuenta el ancho de banda. A diferencia de RIP la métrica no es tan limitante, por lo que es el protocolo IGP (*Interior Gateway Protocol*) más utilizado en redes de gran tamaño. Todos los routers dentro de una misma área comparten una base de datos estado-enlace que describe la topología de la red y que se actualiza con los anuncios de estado-enlace (LSA) recibidos de los diferentes routers, lo que hace que se produzcan pocos errores debido a la visión independiente de la red proporcionada por cada router.

Existen varios tipos de routers:

- *Designated Router* (DR): La existencia de este tipo de router surge de la necesidad de reducir el tráfico que circula por la red. Mantiene la información del área centralizada, almacenando en una tabla toda la topología de la red y enviando actualizaciones al resto de routers.
- *Backup Designated Router* (BDR): El DR supone un punto único de fallo (SPOF), por lo que es necesario tener un dispositivo que actúe, sustituyéndole, en caso de que este se caiga. Para que esto sea posible ha de estar sincronizado con el DR.
- *Internal Router* (IR): Tiene todas sus interfaces en la misma área. Solo tienen una base de datos estado-enlace, la del área en la que se encuentran.
- *Area Border Router* (ABR): Conecta con más de un área. Tiene tantas bases de datos estado-enlace como áreas a las que está conectado.
- *Backbone Router* (BR): Router conectado al área 0¹. Tienen al menos una interfaz conectada al área 0, por lo que dentro de este tipo se pueden englobar

¹Área encargada de conectar el resto de áreas y distribuir la información de encaminamiento entre las mismas



tanto aquellos que componen el área 0, como los que están conectados al mismo y a otras áreas diferentes a la vez (ABR).

- *ASBoundary Router* (ASBR): Independiente de las tres definiciones anteriores. Un router puede ser ASBR y ABR, BR o IR. Este tipo de router es el encargado de conectar con otros sistemas autónomos. Estos no tienen por qué usar OSPF.

Existen diferentes estructuras de datos:

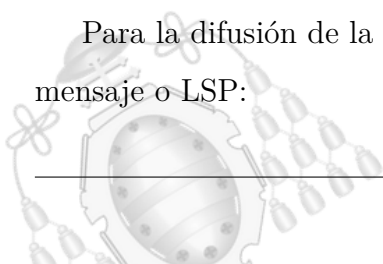
- Base de datos estado-enlace: Almacena todos los LSAs recibidos, por lo que permite conocer la topología completa de la red.
- Tabla de vecinos: Lista de routers con los que se intercambia información OSPF.
- Tabla de rutas: Indica el camino para alcanzar un destino concreto.

Anuncio de estado-enlace (LSA): Describe el estado de las interfaces de un router.

Existen cinco tipos:

- *Router LSA* (Tipo 1): Contiene el identificador del router y los enlaces conectados al mismo. En caso de haber elegido un DR, indica la IP del mismo.
- *Network LSA* (Tipo 2): Enviado por el DR, contiene información de las redes conectadas al mismo, ayudando así a conocer la topología de la red.
- *Summary LSA* (Tipo 3 y 4):
 - Tipo 3: Resumen de las redes de un área realizado por un ABR para enviar a otras áreas.
 - Tipo 4: Identifica al ASBR
- *External LSA* (Tipo 5): Describen las rutas a destinos fuera del sistema autónomo. Son generados en los ASBR.
- *NSSA External* (Tipo 7): Solo existe en redes Cisco. En un NSSA los LSAs de tipo 5 son filtrados, por lo que se deben usar LSAs de tipo 7 para que al llegar al ABR no se filtren y sean convertidos a LSAs de tipo 5 para posteriormente enviarse.

Para la difusión de la topología de la red, OSPF cuenta con los siguientes tipos de mensaje o LSP:





- Paquetes *Hello* (Tipo 1): Se envía de forma periódica. Establece y mantiene las relaciones con los routers vecinos. Además, permiten identificar al DR y al BDR.
- Paquetes de descripción de la base de datos estado-enlace (Tipo 2): Intercambia la información necesaria para que un router en fase de iniciación pueda completar la base de datos estado-enlace o en aquellas situaciones en las que dos nodos conectados quieren sincronizar su información. Se envía un resumen de los datos contenidos en la base de datos estado-enlace.
- Solicitud del estado de los enlaces o LSR (Tipo 3): Solicita información actualizada de alguna entrada de la base de datos, es decir, LSAs específicas.
- Actualización del estado de los enlaces o LSU (Tipo 4): Responde a las solicitudes de estado enviando las LSAs solicitadas.
- Confirmación del estado de los enlaces o LSAck (Tipo 5): Enviado cuando se recibe una actualización de estado para confirmar la recepción.

Durante la configuración del protocolo, los routers descubren quienes son sus vecinos (enviando para ello mensajes *hello*) además de elegir el DR y el BDR. Se intercambian LSAs, y sincronizar así la base de datos estado-enlace. Por último, se calcula la tabla de rutas y se anuncia el estado de los enlaces [43] [44].

A pesar de ser uno de los protocolos de enrutamiento interior más extendidos cuenta con algunas brechas de seguridad que podrían provocar que la red dejase de funcionar correctamente.

Si se hace uso de una contraseña, en vez de utilizar MD5, y además no se utilizan interfaces pasivas, esta podría ser capturada por un atacante (*Eavesdropping*). Una vez se obtiene la contraseña (Figura 4.21), el objetivo del ataque es conseguir que se reconozca al atacante como un router legítimo gracias a que la autenticación es correcta (Figura 4.22), consiguiendo así formar parte de la red. Esto podría provocar que el atacante realizase alteraciones indebidas en las tablas de rutas o incluso que se le eligiese DR (Figura 4.23), haciéndose con el control del área ya que toda la información se centraliza en dicho punto [45]. No obstante, se puede llegar a evitar que el atacante se haga pasar por un elemento legítimo de la red haciendo uso de las interfaces pasivas como se indica más adelante. Por el contrario, los datos de

enrutamiento no van cifrados por lo que se podría obtener información sensible de la topología de la red fácilmente.

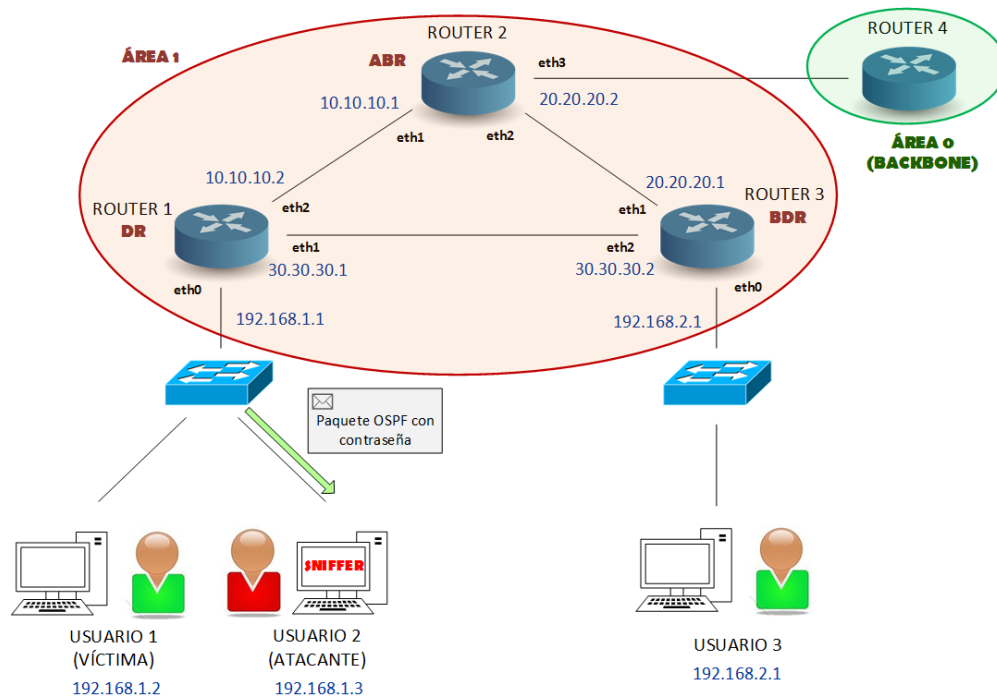


Figura 4.21.- Situación normal OSPF

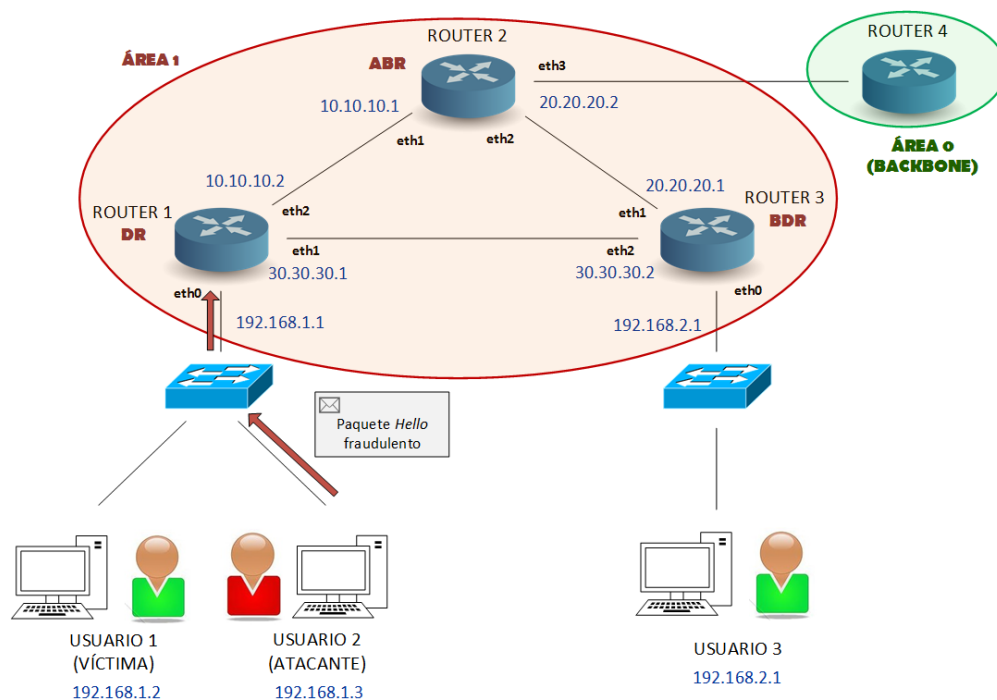


Figura 4.22.- Envío paquete *hello* fraudulento

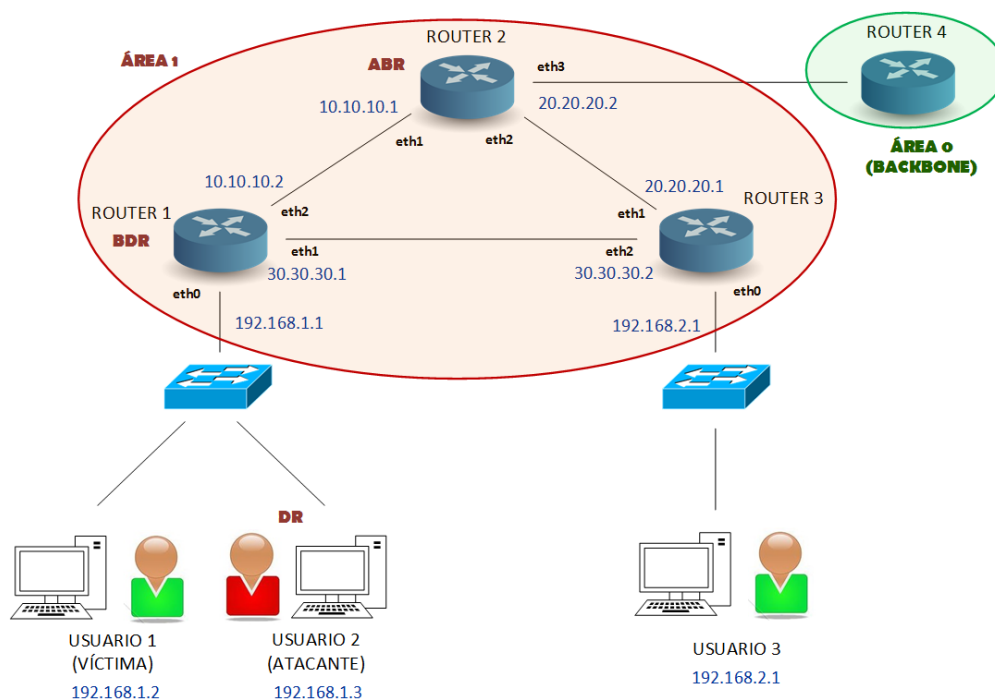
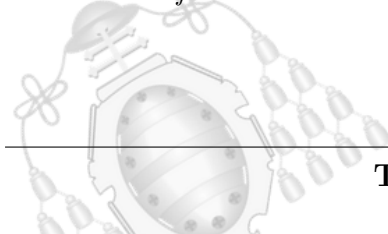


Figura 4.23.- Cambio de DR

Por otro lado, se podría sufrir una denegación de servicio. Para ello solo es necesario el envío de una cantidad elevada de LSA fraudulentos, consiguiendo que la base de datos estado-enlace alcance su límite, de forma que no acepte más actualizaciones [46]. También podría llevarse a cabo mediante la desviación de todo o gran parte del tráfico por un enlace de capacidad limitada, consiguiendo así la congestión del mismo. Otro modo sería modificar la topología de la red para enviar dicho tráfico por rutas más largas, malgastando así los recursos e incrementando el retardo.

Otros ataques que podrían sufrirse son *Persistent OSPF Attacks* [47]:

- *Remote False Adjacency*: Mediante este ataque se hace creer al router víctima que en uno de sus enlaces tiene conectado un router *fantasma* con una red determinada a la que se envían los paquetes. Esta red puede establecerse en cualquier lugar del sistema autónomo por lo que podría vincular dos redes del mismo, consiguiendo así que todo el tráfico que se envía entre ellas pasé por el router *fantasma*.





- *Disguised LSA*: Consiste en el envío de un LSA fraudulento en nombre de un router víctima y con las mismas características que uno legítimo. El problema reside en que este LSA puede considerarse un duplicado del anterior e ignorarse. La solución es inundar la red con el LSA fraudulento, provocando que se envíe el LSA legítimo, entonces el fraudulento se considera una entrada válida, dejando al legítimo como un duplicado.

4.4.- Protocolos restantes

Por último, en este apartado se han añadido aquellos protocolos que por alguna razón no se considera oportuno incluirlos en secciones previas, ya sea porque no se adapta a ninguna o porque se adapta a varias, pero que se considera que son lo suficientemente importantes como para formar parte del proyecto presente. Estos protocolos son: DHCP y HSRP.

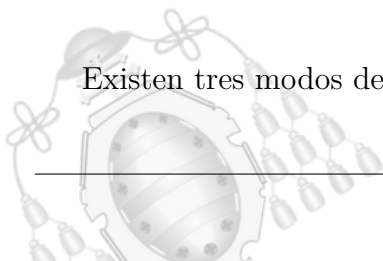
4.4.1.- DHCP

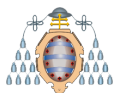
DHCP es un protocolo de la capa de aplicación que permite que los equipos conectados a una red puedan obtener diferentes parámetros de configuración sin necesidad de introducirlos manualmente. Algunos de estos parámetros son:

- Dirección IP única
- Máscara de subred
- Dirección IP de la puerta de enlace
- Servidor DNS

La información se distribuye desde un servidor DHCP, dispositivo encargado de gestionar todas las solicitudes, por lo que su dirección IP ha de ser fija. Este protocolo permite al administrador de la red la asignación de direcciones IP de forma centralizada y sin necesidad de hacerlo máquina a máquina, reduciendo así el trabajo a realizar por el mismo.

Existen tres modos de funcionamiento en DHCP:





- Manual: El administrador asocia en el servidor cada dirección IP a un cliente determinado. De esta forma, cuando este cliente solicita una IP, el servidor busca la dirección MAC en la tabla y asigna la IP establecida por el administrador.
- Automático: Se asigna una dirección IP cuando el dispositivo se conecta por primera vez, y se mantiene de forma indefinida hasta que este la libera. A diferencia del caso anterior, la IP es aleatoria, es decir, no está predefinida por el administrador. Suele hacerse uso de este tipo de configuración en redes con poca variación en el número de usuarios.
- Dinámico: Al igual que en el modo automático, la asignación de direcciones IP es aleatoria. Sin embargo, con este tipo de configuración se promueve la reutilización de direcciones IP, ya que se asignan de forma temporal. Una vez expira el tiempo establecido, el dispositivo pierde la IP y necesita solicitar otra para seguir operando en la red.

Para que un cliente obtenga una dirección IP utilizando un servidor DHCP (funcionamiento básico) son necesarios los siguientes pasos [50]:

1. El cliente envía un mensaje *DHCP Discovery* a la dirección *broadcast* llegando así a todos los integrantes de la red, lo cual le permite localizar el servidor.
2. Ante la recepción del *Discovery*, aquellos elementos de la red que tengan la función de servidor (podría haber más de uno) contestan con un mensaje *DHCP Offer*. En este paquete se ofrece al cliente los parámetros de configuración (generalmente dirección IP, máscara de subred y puerta de enlace predeterminada, aunque puede haber más datos). Puede enviarse a una dirección *broadcast* (a toda la red) o *unicast* (solo al cliente).
3. El cliente transmite su elección, es decir, la IP elegida entre las múltiples ofertadas en caso de haber varios servidores. Para ello envía un paquete *DHCP Request* a la dirección *broadcast*, de forma que todos los servidores (al igual que el resto de elementos de la red) conocen con qué IP se queda el cliente. A su vez esto permite confirmar los parámetros ofrecidos por el servidor con anterioridad.





4. Este proceso finaliza con el envío de un paquete *DHCP Ack* por parte del servidor elegido, el cual confirma los parámetros acordados con el cliente. Al igual que el *DHCP Offer*, puede enviarse una dirección *broadcast* o *unicast*.

Una vez finalizado este proceso (Figura 4.24), el servidor DHCP almacena, de forma temporal o permanente, en su base de datos la relación entre la MAC del cliente y la IP asignada al mismo [51] [52]. A partir de ese momento, el dispositivo correspondiente se conecta siempre a la red con esa dirección, restringiendo así el uso de la misma para otro cliente.

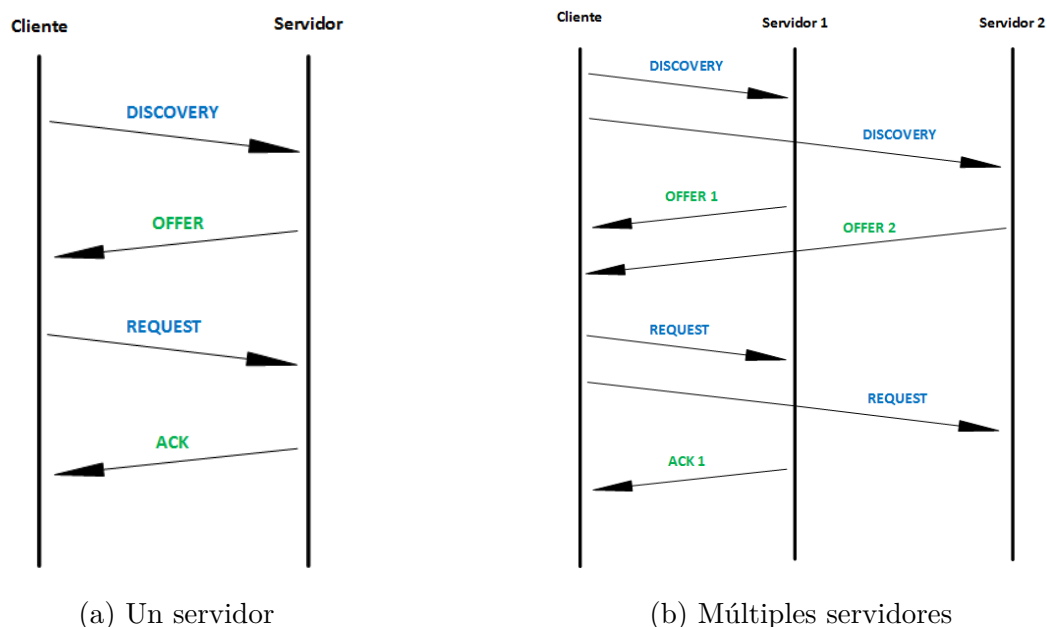
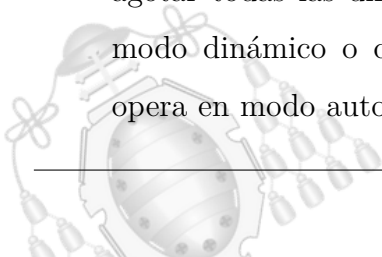


Figura 4.24.- Mensajes DHCP

Este protocolo cuenta con un inconveniente, su facilidad de ser manipulado. Esto se debe a que las solicitudes *DHCP Discovery* se hacen a una dirección *broadcast*, lo cual implica que todos los dispositivos de la red reciben este paquete, haciendo así más sencilla la realización de ataques. Algunos de estos son [53]:

- *DHCP Starvation* [54]: Ataque contra el servidor que tiene como finalidad agotar todas las direcciones de las que dispone de forma temporal si opera en modo dinámico o de forma permanente (hasta que se reinicie el servidor) si opera en modo automático. De esta forma, el servidor no es capaz de responder



a las peticiones legítimas por culpa de la congestión sufrida (denegación de servicio). Para ello se inunda con paquetes *DHCP Discovery*, haciendo uso de direcciones MAC fraudulentas. Su realización es posible con el software Yersinia.

- *DHCP Spoofing* [55]: En este caso es necesario hacer uso de un servidor fraudulento (*rogue server*). Una vez se ha introducido en la red, el atacante se queda esperando a recibir paquetes *DHCP Discovery* para contestar con un *DHCP Offer* más rápido que el servidor legítimo de dicha red, como se puede observar en la figura 4.25. En ese paquete se añaden parámetros de configuración adulterados, entre ellos la dirección de la puerta de enlace, la cual se va a corresponder con la dirección del atacante. De esta forma todo el tráfico pasa por el atacante antes de llegar a su destino (*man in the middle*), pudiendo ver o incluso alterar el contenido. Su realización es posible con el software Metasploit.

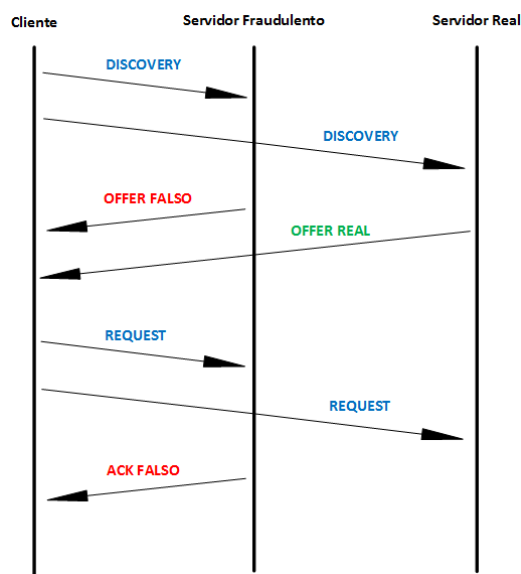


Figura 4.25.- Ataque *DHCP Spoofing*

- *DHCP Ack Injection Attack* [56]: Mejora de *DHCP Spoofing* ya que asegura que la víctima reciba el paquete ilegítimo, además no es necesario conocer el rango de direcciones IP que son válidas. Se escucha todo el proceso hasta que el cliente envía un *DHCP Request* solicitando los parámetros de configuración previamente ofertados, en ese momento el atacante envía un paquete *DHCP Ack* adulterado con los parámetros de configuración que considere oportunos (Figura

4.26), consiguiendo así que el cliente crea que la información recibida es válida. Su realización es posible con la herramienta DHCP Ack Injector.

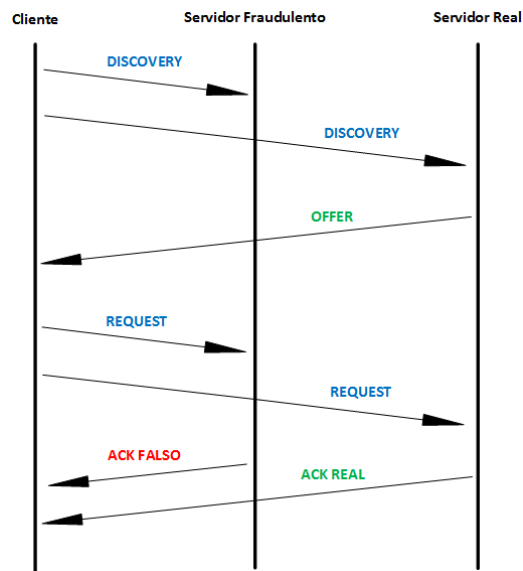
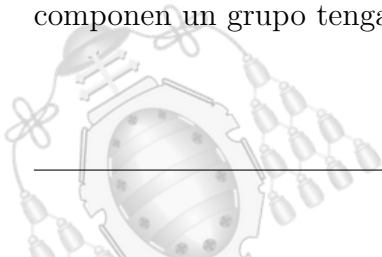


Figura 4.26.- Ataque *DHCP Ack Injection*

4.4.2.- HSRP

HSRP es un protocolo de capa 3 propiedad de Cisco que permite aumentar la disponibilidad de la red gracias al uso de puertas de enlace redundantes [58], todo de forma transparente al usuario. Este protocolo de redundancia de primer salto (FHRP) evita que haya puntos únicos de fallo (SPOF) en la red y que en caso de haberlos la recuperación sea inmediata.

Para aumentar la disponibilidad de la red haciendo uso de HSRP, se crea un grupo formado por varios routers (pueden coexistir varios grupos en una red), identificados por una dirección MAC y una dirección IP virtuales, entre los cuales se localiza un router activo (*active router*) que es el encargado de enrutar el tráfico recibido por la puerta de enlace. En caso de que el maestro falle se cuenta con un router de respaldo (*standby router*), el cual asume el control ejerciendo las mismas funciones que el router activo. Para que esto funcione correctamente es necesario que todos los routers que componen un grupo tengan la misma configuración.



La elección del router activo se realiza haciendo uso de prioridades, por defecto su valor es 100. Aquel router que esté configurado inicialmente con una prioridad superior al resto es el que ocupa el rol de activo, por lo que todo el tráfico enviado a la puerta de enlace se enruta por el mismo (Figura 4.27). El router de respaldo asume que el router activo se ha caído si después de un determinado tiempo deja de recibir mensajes *hello* del mismo. Por otro lado, el router activo también podría avisar de su caída decrementando la prioridad (ha de estar configurado previamente con el comando *preempt*), lo que facilita que el router de respaldo sepa que ha de asumir el control [59] [60].

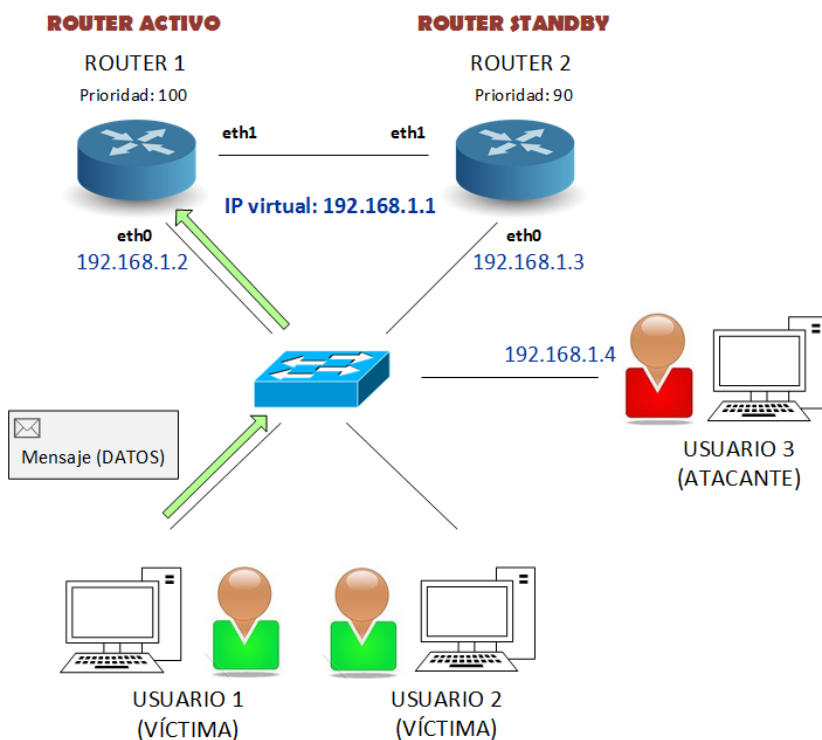
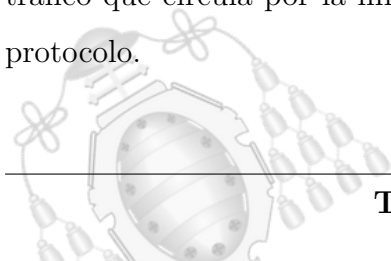


Figura 4.27.- Situación normal HSRP

HSRP proporciona una alta disponibilidad, no obstante, en caso de no realizarse una correcta configuración, puede terminar siendo una brecha de seguridad para la red en la que se usa. Si el usuario consiguiese acceso directo a la red, podría esnifar el tráfico que circula por la misma (Figura 4.28), obteniendo así información acerca del protocolo.



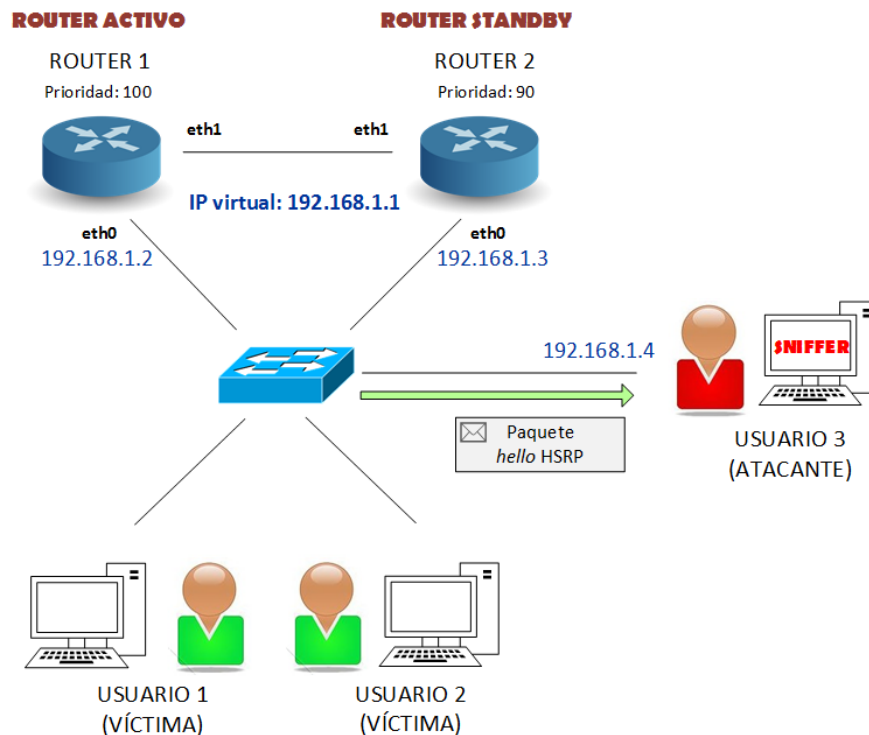
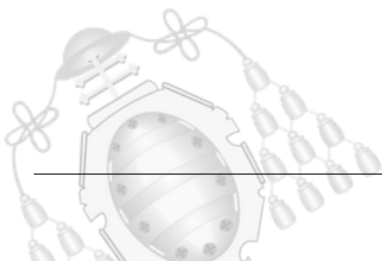


Figura 4.28.- Atacante esnifa tráfico HSRP

Una vez son capturados los paquetes, se puede usar su contenido para atacar la red ya que este indica el estado de los routers y sus prioridades, por lo que el atacante llega a conocer la configuración del protocolo en dicha red. Después de conseguir toda la información necesaria, el atacante puede hacerse pasar por un router legítimo de la red y ser partícipe de HSRP [61] [62]. Para que el ataque tenga éxito, se ha de enviar un paquete que contenga una prioridad superior a la que tiene el router activo en ese momento (Figura 4.29). De esta forma, el atacante se convierte en el nuevo router activo (Figura 4.30), por lo que toda la información enviada a la puerta de enlace pasa por él.



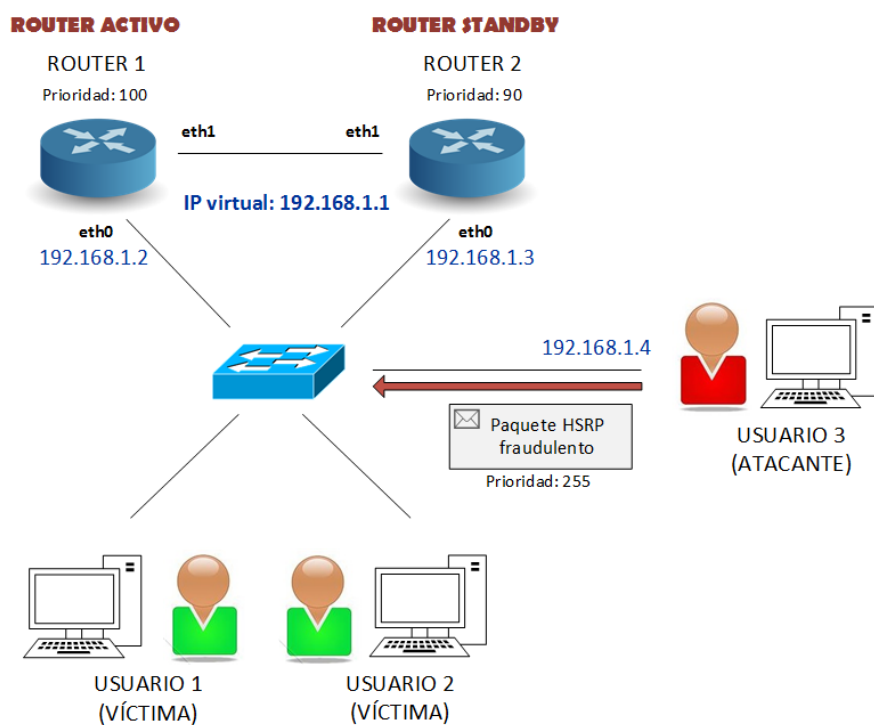
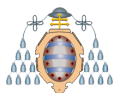


Figura 4.29.- Envío mensaje HSRP fraudulento

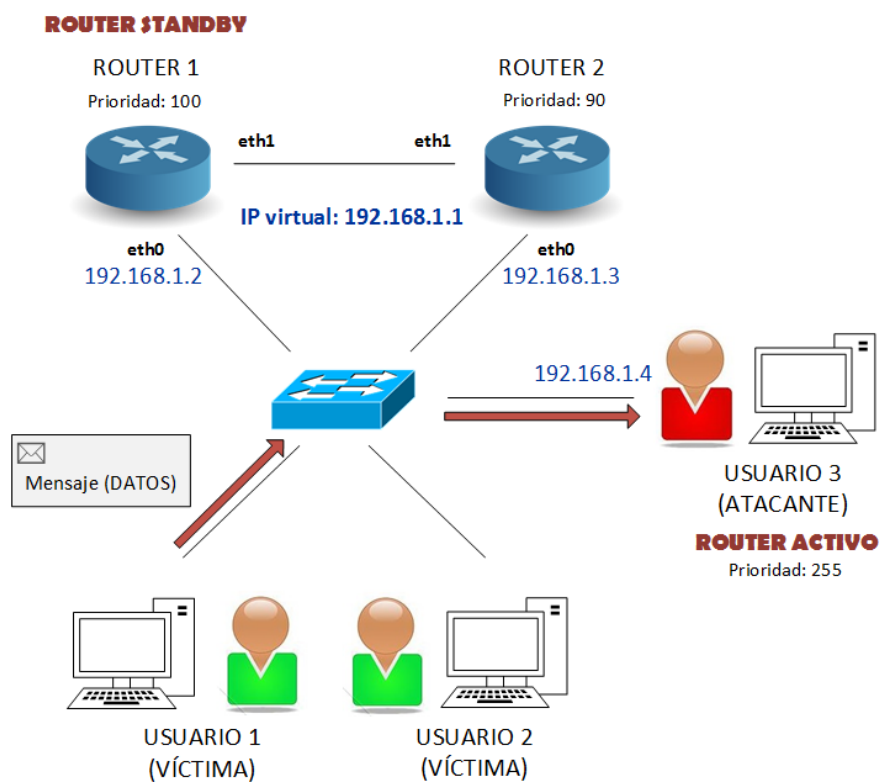
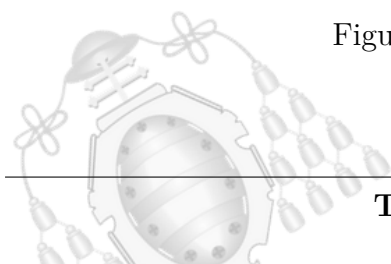


Figura 4.30.- Cambio de router activo

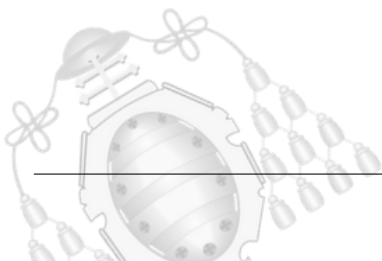


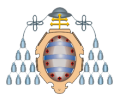


En función de lo que se haga con los paquetes recibidos pueden darse tres casos posibles [63]:

- *Man in the middle*: Los usuarios creen que la conexión se establece entre ellos directamente, sin embargo, los datos transmitidos por ambos extremos son interceptados por el atacante. Por lo general, este ataque tiene como principal finalidad leer la información enviada por los usuarios, para aprovechar con fines malignos lo que se descubre. También podría ser modificada por el atacante o incluso se podría llegar a borrar los paquetes enviados a través de la red.
- Denegación de servicio (*DoS*): Provocar que el acceso a un recurso o dispositivo no sea posible para los usuarios legítimos. Esto se lleva a cabo mediante el desvío de los paquetes a través de un dispositivo que no sea capaz de manejar tal carga de mensajes, provocando que este se sature y deje de responder.
- Degradación del servicio: Envío intermitente de paquetes fraudulentos, generando así períodos de corte en la red.

Para la realización de estos ataques se puede hacer uso de diferentes programas: Scapy, Yersinia o Loki.





5. Soluciones

Tras exponer las vulnerabilidades presentadas por cada protocolo se ha decidido detallar a lo largo del presente capítulo las diferentes soluciones para las mismas. De esta forma, el administrador podría evadir las brechas de seguridad correspondientes haciendo uso de las siguientes recomendaciones.

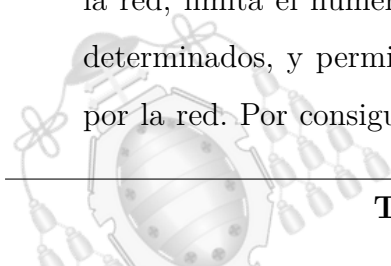
5.1.- Capa de enlace (Capa 2 según modelo OSI)

En esta sección se detallan las múltiples soluciones a los problemas de seguridad presentados por los protocolos de capa 2: ARP y STP.

5.1.1.- ARP

ARP Spoofing es uno de los principales ataques que se pueden sufrir cuando se hace uso de ARP dinámico. No obstante, a continuación, se enumeran algunas buenas prácticas que pueden ayudar a prevenir de dicho ataque [9] [11] [12]:

- **ARP estático:** En este caso es el administrador de la red el que rellena y actualiza la tabla ARP de forma manual; por lo tanto, no se puede añadir información ilegítima en ningún momento. Sin embargo, no es la solución definitiva, sobre todo en redes de gran tamaño, ya que supone un gran esfuerzo y una alta probabilidad de fallo durante su configuración o actualización.
- **Configurar DAI:** DAI aporta seguridad mediante la comprobación de los paquetes ARP que circulan por la red, encargándose de la validación de los mismos. Es posible configurarlo en redes que usen DHCP *Snooping* o no. Esta funcionalidad del protocolo DHCP valida las direcciones IP que se permiten en la red, limita el número de dispositivos que acceden a la LAN en unos puertos determinados, y permite un control estricto de los paquetes ARP transmitidos por la red. Por consiguiente, cualquier tipo de mensaje que no cumpla con esto





y que sea susceptible de ser considerado un ataque del tipo ARP *Spoofing*, se bloquea. Sin embargo, si se desea utilizar DAI en redes que no usen DHCP *Snooping* es necesario configurar manualmente las relaciones entre IP y MAC. De esta forma, se pueden comprobar los paquetes ARP que circulan por la red.

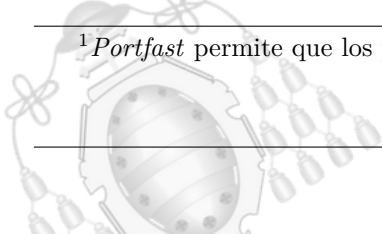
- **Software de detección y prevención de *ARP spoofing*:** Hay muchos que permiten la detección de un ataque de suplantación ARP, normalmente se limitan a escuchar los diferentes mensajes de respuesta y considerar según unas pautas si son peligrosos o no. Arpwatch es un software de Unix que está continuamente escuchando las respuestas ARP, y si detecta un cambio en una entrada de la tabla avisa de forma inmediata al administrador de red mediante un correo electrónico. Además de este, existen más programas, por ejemplo: ArpDefender, AntiARP, ArpOn...

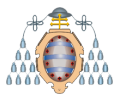
5.1.2.- STP

Una mala configuración en STP puede provocar muchos problemas graves en la red si el atacante conociese estas vulnerabilidades y decidiese explotarlas para realizar acciones inadecuadas. A continuación, se muestran algunas pautas de configuración que es conveniente tener en cuenta para hacer un buen uso del protocolo como administrador [17] [18] [19]:

- ***BPDU Guard*:** Desactiva el puerto si recibe un paquete BPDU. *BPDU Guard* asume que no se pueden recibir paquetes BPDU por puertos *Portfast*¹ activos, ya que son dispositivos finales. Cuando esto sucede el puerto pasa al estado *errDisable*, y solo se puede volver a habilitar de forma manual o estableciendo un *timeout*. Asimismo, dependiendo de la configuración que se tenga puede aparecer un mensaje *syslog* por consola. Es conveniente añadir un *timeout* para volver a levantar el puerto, si se dejase el puerto caído de forma indefinida y el atacante fuese cambiando de interfaz se favorecería la denegación de servicio.

¹*Portfast* permite que los puertos pasen al estado de reenvío de forma inmediata





Su configuración se puede hacer de dos formas: global o por interfaz. Si se hace de forma global, *BPDU Guard* se habilita en todos los puertos *Portfast* activos, para ello es necesario la siguiente sentencia:

```
Switch(config)# spanning-tree portfast bpduguard default
```

No obstante, si se quiere hacer puerto por puerto es necesario añadir lo siguiente:

```
Switch(config-if)# spanning-tree bpduguard enable
```

- **Root Guard:** Permite que el switch que hemos elegido como *root* siga siéndolo sin necesidad de establecer una prioridad muy baja para ello. Gracias a esta configuración, todos los paquetes BPDU con un BID inferior al del *root* que se reciben por una interfaz que lo tenga habilitado se descartan y se coloca el puerto en estado *root-inconsistent*, provocando que no pueda enviar ni recibir datos, solo paquetes BPDU. Es necesario configurarlo a nivel de puerto de la siguiente forma:

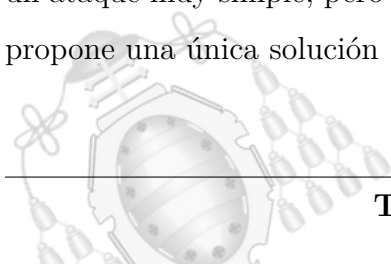
```
Switch(config)# spanning-tree guard root
```

5.2.- Cisco

En esta sección se detallan las múltiples soluciones a los problemas de seguridad presentados por los protocolos de Cisco: CDP, DTP y VTP.

5.2.1.- CDP

Tener CDP activo incrementa las posibilidades de sufrir una denegación de servicio, un ataque muy simple, pero muy dañino y difícil de parar. Ante esta vulnerabilidad se propone una única solución [22] [23]:





- **Desactivar CDP:** En caso de no ser necesario su uso la mejor opción es deshabilitarlo. Si se opta por esta opción es necesario tener en cuenta que con deshabilitarlo de algunas interfaces no es suficiente ya que la tabla se seguirá generando, lo único que se consigue es evitar el envío de anuncios del protocolo. Por lo tanto, para conseguir mayor seguridad en la red es preciso desactivarlo del dispositivo, eliminando así las tablas y los mensajes CDP.

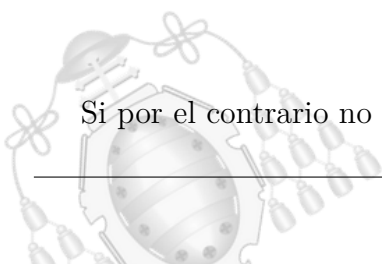
5.2.2.- DTP

Si se decide utilizar DTP en la red es necesario tener cuenta algunas de las recomendaciones citadas a continuación, ya que mantenerlo activo por defecto puede ocasionar problemas graves si el atacante lo descubre y explota las vulnerabilidades con las que cuenta, como se indicaba anteriormente. Se proponen diversas soluciones en función de si es necesario o no el puerto. Para aquellas situaciones en las que se necesite el puerto [26] [27]:

- **Configurar manualmente los puertos:** Por defecto los puertos vienen en modo *dynamic auto* o *dynamic desirable*, lo cual facilita la configuración del enlace troncal. Sin embargo, también facilita los ataques, ya que supone un agujero de seguridad en la red. Por esta razón se recomienda a los administradores de la red la configuración manual de los puertos, a pesar de lo tedioso que resulta, ya que de esta forma se pueden evitar la explotación de esta vulnerabilidad.
- **Deshabilitar los anuncios DTP:** Incluso haciendo uso de una configuración estática, los anuncios DTP siguen enviándose de forma periódica. No obstante, se cuenta con la ventaja de que estos anuncios pueden deshabilitarse, evitando así las posibles negociaciones. Para ello es necesario el siguiente comando [28]:

```
Switch(config-if)# switchport nonegotiate
```

Si por el contrario no es necesario mantener el puerto activo, se recomienda:



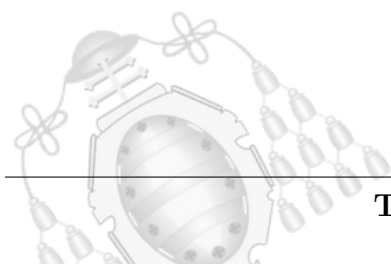


- **Apagar los puertos no utilizados:** Todos los puertos del switch vienen activos por defecto, por lo que se recomienda apagar aquellos que no se vayan a usar.
- **Configurar manualmente los puertos como puertos de acceso:** Es recomendable, siempre que sea posible, poner los puertos que no se usan como puertos de acceso de una VLAN no utilizada por más usuarios. De esta forma se tiene una mejor configuración desde el punto de vista de la seguridad de la red y en caso de necesidad se podría modificar el puerto, configurándolo de nuevo como troncal.
- **Asignarles VLANs no existentes:** Aquellos puertos que no se estén usando se les asigna una VLAN inexistente, de esta forma si alguno se activase de forma accidental no habría filtración de la información que circula por la red.

5.2.3.- VTP

Ante la posibilidad de que un atacante alterase la configuración de las VLANs determinadas por el administrador, haciendo uso para ello de VTP, se proponen algunas soluciones para mitigar este ataque [31] [32]:

- **No usar VTP:** En caso de no ser necesario el uso de este protocolo para configurar la red se recomienda deshabilitarlo, reduciendo así la vulnerabilidad a ataques.
- **Autenticación:** Se hace uso de la función de contraseña proporcionada por VTP. Usa el algoritmo MD5, codifica las contraseñas en 16 bytes. Estas contraseñas se envían dentro de los anuncios del protocolo. Es importante que todos los switches tengan el mismo nombre de dominio y la misma contraseña, de lo contrario la comunicación no sería posible.



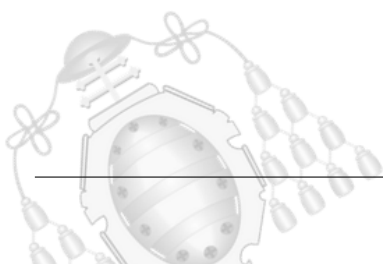


- **Apagar los puertos no utilizados:** Todos los puertos del switch vienen activos por defecto, por lo que se recomienda apagar aquellos que no se vayan a usar.
- **Usar versión 3:** Añade el concepto de *Primary Server*, haciendo así que solo ese switch pueda crear, borrar o modificar las VLANs. Además, cuenta con mejoras de seguridad en la autenticación [33].

5.2.4.- Problema *Double Tagging*

Como bien se comentaba con anterioridad, estos tres protocolos cuentan con una misma brecha de seguridad, por consiguiente, todos son susceptibles de sufrir un ataque del tipo *Double Tagging*. A continuación, se detallan algunas soluciones posibles a este problema [36] [37]:

- **No usar la VLAN predeterminada:** Evitar el uso de la VLAN 1 como nativa, ya que viene por defecto en todos los dispositivos, por lo que su uso supone una brecha de seguridad en la red. Se recomienda el uso de otro identificador de VLAN nativa para los enlaces troncales, preferiblemente uno que no se esté usando en ningún puerto de usuario.
- **Apagar los puertos no utilizados:** En caso de que no sea necesario el uso de algún puerto, la mejor opción es su desactivación. No obstante, si fuese necesario mantener activo dicho puerto, se recomienda el uso de una VLAN en los puertos troncales que no se esté usando ya en los puertos que conectan con los usuarios, como bien se indicaba en la primera solución.
- **Desactivar DTP en los puertos que no se necesite:** De esta forma se puede evitar la creación de enlaces troncales no deseados.





5.3.- Capa de red (Capa 3 según modelo OSI)

En esta sección se detallan las múltiples soluciones a los problemas de seguridad presentados por los protocolos de capa 3: RIP y OSPF.

5.3.1.- RIP

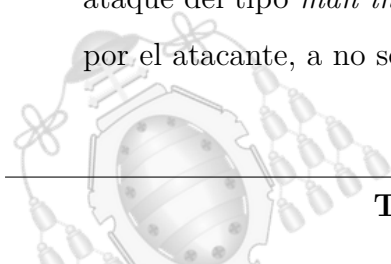
RIP es un protocolo muy extendido susceptible de sufrir múltiples ataques que acaben con la integridad de la red. Con el fin de mitigar esto y las posibles consecuencias ocasionadas se proponen algunas soluciones a continuación [41] [42]:

- **Versión 2 del protocolo:** Debido a las múltiples mejoras de esta versión respecto a la primera, es recomendable el uso de la misma. Entre algunas de las mejoras encontramos: el uso de *multicast* en lugar de *broadcast*, admite subredes, máscaras de longitud variable y CIDR. Pero lo más importante, desde el punto de vista de la seguridad, es que los intercambios están autenticados.
- **Usar *redistribute connected*:** Aquellas redes en las que hay usuarios finales no se deben anunciar en el protocolo RIP haciendo uso del comando *network*, sino que se deben distribuir con el siguiente comando:

```
Router(config-router)# redistribute connected
```

De esta forma, se anuncian todas las redes conectadas directamente al router. No obstante, aquellas que no se han anunciado previamente con *network* no son partícipes del protocolo, por lo que no pueden recibir ni enviar mensajes RIP.

- **Usar autenticación:** Siempre que sea posible es aconsejable el uso de contraseña para los intercambios. De esta forma se evita que, si se sufre un ataque del tipo *man in the middle*, los mensajes puedan ser leídos o modificados por el atacante, a no ser que este conozca la contraseña.





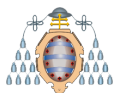
- **Definir las rutas de forma estática:** El uso de rutas estáticas permite que las tablas de rutas no sean alteradas por mensajes RIP fraudulentos, ya que en caso de recibir uno de estos se ignora, no llevando a cabo los cambios anunciados. Esto se debe a que las rutas estáticas tienen preferencia sobre las dinámicas.
- **Usar ACLs:** Permiten controlar el flujo del tráfico dentro de la red. Aceptan o deniegan el tráfico en función del origen (es posible el uso de otros parámetros a la hora de decidir), consiguiendo así que, si un usuario no legítimo intenta enviar tráfico, este se descarte. Es una de las principales soluciones existentes para los ataques del tipo *Reflection DDoS* en aquellas redes que hagan uso de la versión 1 del protocolo y no quieran cambiar a la 2.

5.3.2.- OSPF

A pesar de ser uno de los protocolos más extendidos para el enrutamiento en redes de gran tamaño, cuenta con algunas brechas de seguridad. No obstante, a continuación, se indican algunas buenas prácticas que permiten al administrador hacer uso de este protocolo sin miedo a sufrir ataques de seguridad [47] [48]:

- **Usar autenticación MD5:** Para incrementar la seguridad se recomienda usar MD5 en vez de texto plano o una simple contraseña. La contraseña no se intercambia entre los extremos, sino que se utiliza una función hash que permite la autenticación. Este hash podría llegar a descifrarse, con ayuda de programas *auth cracking*, pero a diferencia de las otras dos posibilidades, se cuenta con menor vulnerabilidad frente a ataques.
- **Determinar las interfaces pasivas:** Se establecen todas las interfaces conectadas a usuarios finales como pasivas, de esta forma si se conectase un sniffer al otro lado, este no recibiría ningún tipo de mensaje del protocolo OSPF, evitando así el uso de los mismos con malas intenciones. Por el contrario, la red seguirá incluyéndose en el enrutamiento [49].





Para establecer las interfaces como pasivas es necesario usar el siguiente comando:

```
Router(config-router)# passive-interface X
```

donde X determina la interfaz.

- **Números de secuencia aleatorios:** De esta forma se puede conseguir que los ataques del tipo *LSA Disguised* no se produzcan, ya que uno de los campos necesarios es el número de secuencia y si este no tiene un comportamiento previsible no puede generarse el LSA fraudulento.
- **Habilitar *TTL Security*:** Para prevenir ataques del tipo *Remote false adjacency* se añade a los paquetes salientes un valor TTL (*Time To Live*), descartando todos los paquetes entrantes con un valor TTL inferior al umbral que se defina.

5.4.- Protocolos restantes

En esta sección se detallan las múltiples soluciones a los problemas de seguridad presentados por los protocolos DHCP y HSRP.

5.4.1.- DHCP

Son múltiples las opciones que existen para realizar ataques contra DHCP, por lo que a continuación se indican algunas soluciones para evitar que un usuario ilegítimo se conecte a la red y se convierta en servidor DHCP, pudiendo alterar el correcto funcionamiento de la misma [53] [54]:

- **VACL:** Determina qué direcciones pueden responder a solicitudes DHCP. De esta forma, se limita que las respuestas solo procedan de servidores reales, y no de servidores fraudulentos. Su uso consigue evitar que un elemento conectado a





la red pueda convertirse en servidor DHCP sin consentimiento del administrador. No obstante, el atacante podría falsificar sus direcciones MAC e IP, evadiendo así la seguridad impuesta por las VACLs.

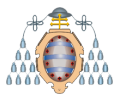
- **DHCP Snooping** [57]: Proporciona una mayor seguridad. Se configuran dos tipos de puertos: *trusted* y *untrusted*. Los primeros permiten el envío de cualquier tipo de mensaje DHCP. Por norma general se suelen conectar servidores DHCP a estos puertos. Los segundos, por el contrario, no permiten enviar mensajes DHCP que no sean comunes en un cliente, privando así a dichos puertos de realizar las funciones de un servidor. Estos suelen configurarse en los puertos que conectan con los usuarios finales, evitando que un atacante pueda hacerse pasar por un servidor (*rogue server*).
- **Port-security**: Limitando el número máximo de direcciones MAC en un puerto se puede llegar a evitar ataques del tipo *DHCP Starvation*, ya que para su realización se utilizan muchas direcciones MAC aleatorias generadas por un mismo dispositivo.
- **Apagar los puertos no utilizados**: En caso de que no sea necesario el uso de algún puerto, la mejor opción es su desactivación, evitando así posibles servidores fraudulentos en los puertos inutilizados de la red.

5.4.2.- HSRP

HSRP es un protocolo que permite que la red esté disponible la mayor parte del tiempo gracias al uso de routers redundantes. No obstante, podría ocurrir que un usuario ilegítimo se hiciese pasar por uno de esos routers, provocando que la red deje de funcionar correctamente. A continuación, se detallan algunas soluciones para evitar que esto pase [61] [63]:

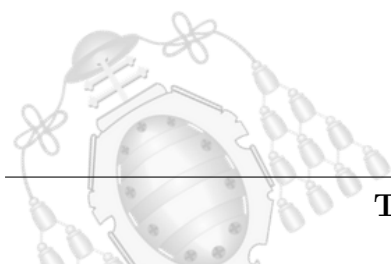
- **Usar autenticación MD5**: Para incrementar la seguridad se recomienda usar MD5 en vez de texto plano. Esto permite que aquellos mensajes que vengan sin





autenticar sean descartados, por lo que los posibles ataques se ignoran y por consiguiente, no se llevan a cabo los cambios ilegítimos. Este hash podría llegar a descifrarse, con ayuda de programas *auth cracking*, pero a diferencia del texto plano, se cuenta con menor vulnerabilidad frente a ataques ya que se requiere un mayor esfuerzo para alterar la configuración establecida por el administrador.

- **Poner la máxima prioridad al router activo:** No es la opción más recomendable, pero puede evitar que un usuario ilegítimo se haga con el rol de router activo al enviar un mensaje con una prioridad mayor al actual [62]. El problema presentado por esta solución es que se haría inviable el uso de los *interface tracking*.
- **ACL:** Permite filtrar el tráfico, permitiendo o rechazando los paquetes conforme a unas reglas preestablecidas. De esta forma, se controla el envío de paquetes fraudulentos desde usuarios ilegítimos.
- **Implementar HSRP con IPSec:** IPSec es un protocolo encargado de que las conexiones sobre IP se cifren y autenticuen proporcionando así una mayor seguridad. A diferencia de TLS o SSH, este protocolo opera sobre la capa 3 del modelo OSI, lo que permite que pueda ser usado en la capa de transporte (UDP y TCP). Gracias a su uso se garantiza una conexión segura, ya que se cumplen tres de los criterios principales de la seguridad: confidencialidad, integridad y autenticación.





6. Trabajo realizado

Gran parte del desarrollo de la herramienta se ha llevado a cabo en la distribución de Kali Linux, haciendo uso de la herramienta Scapy [67], desarrollada en Python, que permite la captura, generación y manipulación de paquetes. Estas son algunas de las ventajas que presenta la herramienta. No obstante, cuenta con muchas más opciones para el manejo de paquetes, permitiendo detectar, escanear o atacar redes. Asimismo, ofrece la posibilidad de que el usuario pueda crear funciones de alto nivel e implementar protocolos que no están disponibles por defecto en la herramienta, como OSPF.

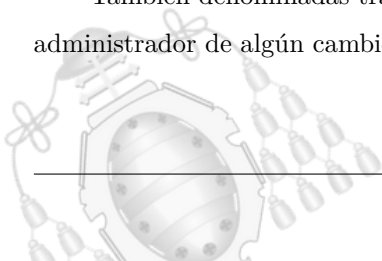
Por otro lado, también se ha hecho uso de la herramienta MIB Browser [68] para la recepción de notificaciones¹, siendo posible el uso de otra diferente en un futuro. Esta parte del proyecto se ha llevado a cabo usando como servidor SNMP un ordenador con el sistema operativo de Windows.

MIB Browser es una herramienta que, mediante SNMP, permite obtener información de un dispositivo o aplicación, para lo que se pueden cargar MIBs específicos. En el presente proyecto se usa para recibir notificaciones de la herramienta ante la detección de alguna anomalía en la red, sin embargo, también permite la realización de diferentes acciones para obtener datos concretos de un dispositivo o aplicación de la red.

Por último, se han utilizado los entornos de desarrollo: PyQt, para la realización de la interfaz gráfica, y PyCharm, para añadir funcionalidad a la misma.

A lo largo de este capítulo se expone el tipo de detección (vulnerabilidad o ataque) que se realiza para cada protocolo. Además, se explica de forma detallada las características de cada protocolo tenidas en cuenta para realizar dicha detección y el envío de las notificaciones correspondientes.

¹También denominadas traps. Son mensajes SNMP enviados por un agente con el fin de avisar al administrador de algún cambio o problema en la red.





6.1.- Visión general

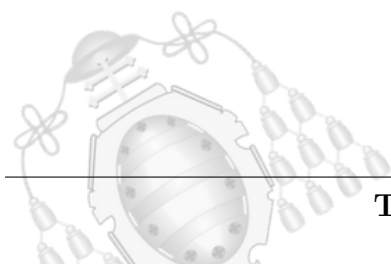
Esta sección muestra una visión global del trabajo realizado en el presente proyecto. Se ha procurado que la descripción sea lo más visual posible, haciendo uso para ello de la tabla mostrada a continuación.

La herramienta diseñada no realiza un único tipo de detección para todos los protocolos, sino que detecta vulnerabilidades, ataques y en algunos casos concretos ambas opciones, según se ha considerado apropiado durante la realización del proyecto. Como se puede observar en la tabla 6.1, las combinaciones a la hora de detectar una mala configuración o un posible ataque son muy diversas, no obstante, en todos los casos se envía una notificación personalizada al servidor SNMP que avisa del problema detectado.

	Mala configuración	Posible ataque	Envío de notificación
ARP	X	X	X
STP	X		X
CDP	X		X
DTP	X		X
VTP	X		X
RIP	X	X	X
OSPF	X	X	X
DHCP		X	(X) ²
HSRP		X	X

Cuadro 6.1.- Visión global

²El envío y recepción de estas notificaciones no es siempre posible como bien se explica más adelante en el capítulo de Pruebas y Resultados 7





6.2.- Detección

A continuación, se detallan las particularidades tenidas en cuenta para detectar los diferentes protocolos, y dentro de los mismos percibir posibles vulnerabilidades o ataques. Asimismo, se ha añadido al final de cada subapartado un fragmento de código donde se muestra el filtro del sniffer utilizado en cada caso.

6.2.1.- Capa de enlace (Capa 2 según modelo OSI)

En esta sección se indica el proceso de detección seguido para los protocolos ARP y STP.

6.2.1.1.- ARP

En caso de que el usuario marque la casilla de ARP, se le da la posibilidad de elegir entre dos opciones, dependientes de la configuración de la red:

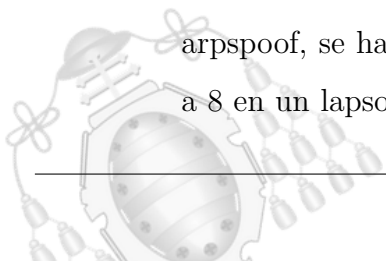
- **ARP estático:** Las entradas de la tabla ARP son fijas (generadas por el administrador). Si el usuario indica que su red hace uso de esta configuración, se considera que no debería haber mensajes ARP *Reply* en la red. Toda la información de la tabla ARP es indicada por el administrador de forma manual, sin aprender nada a través del protocolo, por lo que se informa ante la recepción de cualquier mensaje ARP *Reply*.

Podría tratarse de una mala configuración o de un intento de envenenamiento de la tabla ARP.

- **ARP dinámico:** Las entradas de la tabla ARP se van aprendiendo a través de dicho protocolo. En este caso se envía una notificación al usuario si:

- Se reciben múltiples mensajes ARP *Reply* con la misma dirección IP y MAC de origen en un periodo de tiempo.

Tras la realización de múltiples pruebas (Figura 6.1), con ayuda del comando *arp spoof*, se ha determinado que el número de paquetes ha de ser superior a 8 en un lapso de 20 segundos.



- Se reciben dos mensajes ARP *Reply* con diferentes direcciones MAC de origen, pero una misma dirección IP de origen. Se establece un tiempo de detección idéntico al del caso anterior.

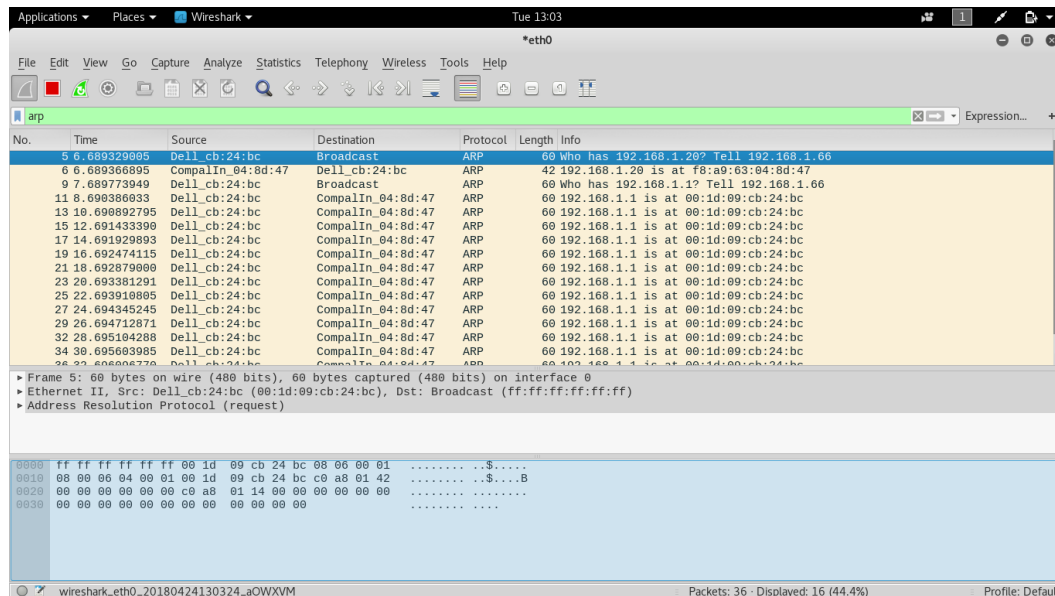


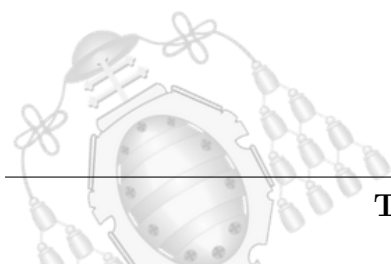
Figura 6.1.- Tiempo entre mensajes ARP *Spoof*

En ambas situaciones se trata de detectar en tiempo real un caso de ARP *Spoofing*, es decir, intento de envenenar la tabla ARP mediante el envío periódico de mensajes de dicho protocolo. Estos mensajes contienen información modificada, en la cual se asocia la dirección MAC del atacante con la IP elegida.

Para su detección, se comprueban los mensajes *ARP Reply*, previamente mencionados, ya que contienen información susceptible de ser alterada, al contrario que los *ARP Request* que únicamente preguntan por dicha información.

La comprobación del tipo de mensaje ARP que se recibe es posible gracias al segundo byte de *opcode* que indica si es: *Request* (01) o *Reply* (02).

```
fltA="(arp)"
if(globV.form1.arpCheck.isChecked()):
    filtros.append(fltA)
```





6.2.1.2.- STP

Como ya se indicaba en la primera sección de este capítulo 6.1, se da la opción de controlar este protocolo para detectar una mala configuración del mismo en vez de un posible ataque.

El funcionamiento de este protocolo ha de ser ajeno a los usuarios de la red, solo debe afectar a los elementos de interconexión. En caso de que el ordenador en el que se aloja la herramienta de detección de vulnerabilidades recibiese un paquete STP (BPDU) se avisaría al administrador mediante el envío de una notificación, ya que estos mensajes no deberían llegar a los dispositivos finales.

Para capturar paquetes STP se usa la dirección *multicast* “01:80:C2:00:00:00” [66], utilizada por los switches como dirección de destino para enviar los paquetes del protocolo (BPDU).

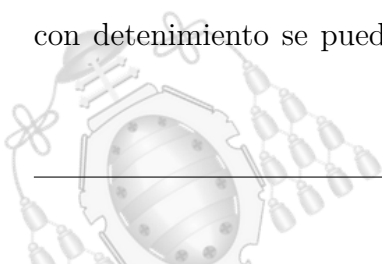
```
fltG="(ether dst 01:80:C2:00:00:00)"  
if(globV.form1.stpCheck.isChecked()):  
    filtros.append(fltG)
```

6.2.2.- Cisco

En esta sección se indica el proceso de detección seguido para los protocolos CDP, DTP y VTP.

Para los tres protocolos se detecta una mala configuración de los mismos, ya que algunos (CDP y DTP) vienen activos por defecto en los switches Cisco con el fin de facilitar la configuración de la red. No obstante, esto muchas veces provoca más problemas de los que teóricamente soluciona.

Su detección es posible gracias a que todos ellos envían la información a la misma dirección de destino *multicast* “01:00:0C:CC:CC:CC” [66]. Sin embargo, si se analizan con detenimiento se puede detectar que el segundo byte del campo PID varía de un





protocolo a otro (Figuras 6.2, 6.3 y 6.4), permitiendo distinguirlos.

```
▶ Frame 16: 452 bytes on wire (3616 bits), 452 bytes captured (3616 bits) on interface 0
▼ IEEE 802.3 Ethernet
  ▶ Destination: CDP/VTP/DTP/PAgP/UDLD (01:00:0c:cc:cc:cc)
  ▶ Source: CiscoInc_30:af:81 (cc:d5:39:30:af:81)
  Length: 438
▼ Logical-Link Control
  ▶ DSAP: SNAP (0xaa)
  ▶ SSAP: SNAP (0xaa)
  ▶ Control field: U, func=UI (0x03)
  Organization Code: Cisco (0x00000c)
  PID: CDP (0x2000)
▶ Cisco Discovery Protocol
```

Figura 6.2.- PID CDP

```
▶ Frame 171: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface 0
▶ ISL
▼ IEEE 802.3 Ethernet
  ▶ Destination: CDP/VTP/DTP/PAgP/UDLD (01:00:0c:cc:cc:cc)
  ▶ Source: CiscoInc_30:af:81 (cc:d5:39:30:af:81)
  Length: 34
  Trailer: 000e000000000000000000000000
  Frame check sequence: 0xb92c5403 [correct]
  [FCS Status: Good]
▼ Logical-Link Control
  ▶ DSAP: SNAP (0xaa)
  ▶ SSAP: SNAP (0xaa)
  ▶ Control field: U, func=UI (0x03)
  Organization Code: Cisco (0x00000c)
  PID: DTP (0x2004)
▶ Dynamic Trunk Protocol: (Operating/Administrative): Access/Auto (0x04) (Operating/Administrative): ISL/802.1Q (0x45): cc:d5:39:30:af:81
```

Figura 6.3.- PID DTP

```
▶ Frame 338: 99 bytes on wire (792 bits), 99 bytes captured (792 bits) on interface 0
▼ IEEE 802.3 Ethernet
  ▶ Destination: CDP/VTP/DTP/PAgP/UDLD (01:00:0c:cc:cc:cc)
  ▶ Source: CiscoInc_30:af:83 (cc:d5:39:30:af:83)
  Length: 85
▼ Logical-Link Control
  ▶ DSAP: SNAP (0xaa)
  ▶ SSAP: SNAP (0xaa)
  ▶ Control field: U, func=UI (0x03)
  Organization Code: Cisco (0x00000c)
  PID: VTP (0x2003)
▶ VLAN Trunking Protocol
```

Figura 6.4.- PID VTP

La comprobación del tipo de protocolo detectado ha sido posible gracias a que se conoce el valor de dicho byte como se puede ver en las figuras previamente mostradas: CDP (00), VTP (03) y DTP (04).

```
fltD="(ether dst 01:00:0C:CC:CC:CC)"
```



```
if(globV.form1.cdpCheck.isChecked() or globV.form1.dtpCheck.isChecked() or  
    globV.form1.vtpCheck.isChecked()):  
    filtros.append(fltD)
```

6.2.3.- Capa de red (Capa 3 según modelo OSI)

En esta sección se indica el proceso de detección seguido para los protocolos RIP y OSPF.

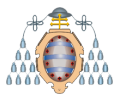
6.2.3.1.- RIP

La principal razón por la que se ha decidido controlar los mensajes RIP es el descubrimiento de una mala configuración. Para la detección de dichos mensajes se ha tenido en cuenta que RIP utiliza UDP como protocolo de transporte y envía sus mensajes a través del puerto 520 [64]. Se pueden distinguir dos tipos según la versión:

- **RIP versión 1:** Cualquier mensaje que se detecte procedente de esta versión del protocolo se informa ya que no debería estar activo debido a las limitaciones que presenta con respecto a la segunda versión. Entre ellas se encuentra que las actualizaciones se envían mediante *broadcast*, en vez de *multicast*; diferencia, junto con el campo que identifica la versión del protocolo, que se usa para la identificación de mensajes correspondientes a esta versión del protocolo.
- **RIP versión 2:** Informa al usuario ante la detección de una mala configuración, en la cual el router no tiene declaradas como interfaces pasivas aquellas que conectan con los dispositivos finales.

Uno de los campos que permite la identificación de esta versión del protocolo es el envío de los mensajes a la dirección *multicast* 224.0.0.9 [66], por lo tanto, los paquetes que tienen como destino dicha IP, y además un 2 en el campo de la versión, son capturados.

```
fltC="(udp and port 520 and (dst host 224.0.0.9 or dst host 255.255.255.255))"  
if(globV.form1.ripCheck.isChecked()):
```



```
filtros.append(fltC)
```

Ante la detección de dichos mensajes, ya sea de una versión u otra, la herramienta tiene como cometido avisar al administrador de la red, de forma inmediata, mediante el envío de la notificación correspondiente.

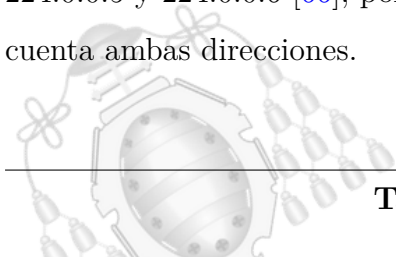
Por otro lado, se contempla además la posibilidad de detección de un ataque y su notificación de forma inmediata al administrador de la red; teniendo en cuenta en todo momento las limitaciones que se tienen ya que la herramienta nunca se arranca en el router sino en un ordenador de la red corporativa.

Si se detecta que un elemento de la red con una IP diferente a la de la puerta de enlace del router está enviando mensajes a la dirección *multicast* o *broadcast* haciéndose pasar por un dispositivo partícipe de RIP, se avisa al administrador ya que probablemente se trate de un ataque. Sin embargo, si el ataque se efectuase de forma directa al router no sería posible su detección ya que el ordenador en el que se sitúa la herramienta de detección desconoce los paquetes que llegan a la dirección *unicast* del router.

6.2.3.2.- OSPF

Al igual que el apartado anterior, primordialmente se persigue detectar una mala configuración del protocolo, en la cual no se hace uso de interfaces pasivas en los enlaces que conectan con los dispositivos finales de la red. Sin embargo, también se tiene en consideración, al igual que en RIP y siguiendo el mismo patrón, la detección de posibles ataques haciendo uso del protocolo OSPF, además del aviso inmediato al administrador.

Respecto a RIP existe una diferencia importante, OSPF no usa UDP como protocolo de transporte, sino que se encapsula directamente sobre IP y se pone un 89 [65] en el campo del protocolo. Asimismo, los mensajes se envían a dirección de destino *multicast* 224.0.0.5 y 224.0.0.6 [66], por lo que para la detección de los mismos se han tenido en cuenta ambas direcciones.





```
fltB="(ip proto 89 and (dst host 224.0.0.5 or dst host 224.0.0.6))"  
if(globV.form1.ospfCheck.isChecked()):  
    filtros.append(fltB)
```

6.2.4.- Protocolos restantes

En esta sección se indica el proceso de detección seguido para los protocolos DHCP y HSRP.

6.2.4.1.- DHCP

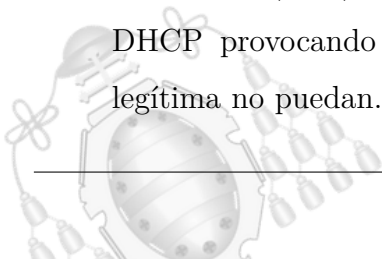
Para este protocolo se ha decidido implementar la detección de 3 tipos de ataques en tiempo real. Determinando para ello que el filtro de captura contenga el protocolo UDP, usado por DHCP para el transporte de los paquetes, además de los puertos 67 y 68 [64] sobre los que irán los mensajes.

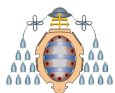
```
fltE="(udp and (port 68 or port 67))"  
if(globV.form1.dhcpCheck.isChecked()):  
    filtros.append(fltE)
```

Se han analizado varios tipos de paquetes:

- **DHCP *Discovery***: Es enviado desde el cliente hacia el servidor para que este le asigne una dirección IP y otros parámetros DHCP. Para poder distinguirlo del resto se ha tenido en cuenta que el puerto de origen es el del cliente (68) y el de destino el del servidor (67). Además, se hace uso del campo de opciones de DHCP que indica el tipo de mensaje que es, en este caso ese campo tiene valor 1.

Haciendo uso de este tipo de paquetes se podría llevar a cabo una denegación de servicio (*DoS*), es decir, agotar todas las IPs de las que dispone el servidor DHCP provocando que otros usuarios que intentan conseguir una de forma legítima no puedan.





Para la detección de dicho ataque se da la opción al usuario de introducir diversos parámetros durante la configuración de la herramienta: máscara de la red que quiere controlar, porcentaje de dirección solicitadas y tiempo en el que solicitan las mismas. En caso de que la herramienta detecte en dicho tiempo una solicitud de IPs mayor al porcentaje establecido, se envía una notificación que avisa al administrador de un posible ataque.

Se ha decidido realizar una interfaz dinámica que facilite la configuración al usuario, para ello se han realizado múltiples pruebas con la herramienta Yersinia, pudiendo establecer así cuantos mensajes son recibidos por segundo en caso de sufrir una denegación de servicio. En la tabla 6.2 se pueden observar las conclusiones extraídas. Todos los tiempos estimados en la misma han sido determinados para detectar de forma holgada la solicitud de más del 50 % de las IPs del servidor.

Máscara	Tiempo estimado (segundos)
16	64
17	32
18	16
19	8
20	4
21	4
22	4
23	4
24	2
25	2
26	2
27	2
28	2
29	2
30	2

Cuadro 6.2.- Estimación tiempos DHCP *Discovery*





- **DHCP Offer:** Paquete que se envía desde el servidor hacia el cliente como respuesta a la solicitud de los parámetros DHCP correspondientes. En este caso, a diferencia del anterior el puerto de origen es el del servidor (67) y el de destino el del cliente (68). Además, el campo de opciones de DHCP tiene valor 2.

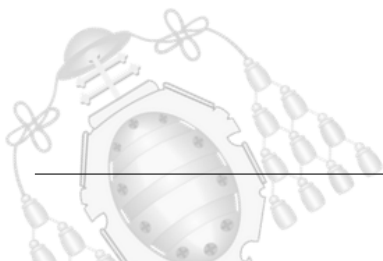
El control de estos mensajes es necesario ya que podría darse una situación en la cual un cliente recibiese dos ofertas procedentes de diferentes servidores. La IP que se asigna al cliente depende de cual de ambas respuestas llegue antes, ya que este se queda con aquella que recibe primero. Por ello, se ha decidido controlar los mensajes DHCP Offer que se reciben a lo largo de 3 segundos desde que se recibe uno de estos, dejando así tiempo suficiente para la detección, ya que dichos mensajes llegan de forma simultánea. En caso de que contengan direcciones IP con diferentes orígenes se enviará una notificación al administrador de la red, delegando en este la responsabilidad de determinar si ambos servidores son legítimos o no.

- **DHCP ACK:** Acuse de recibo del servidor al cliente. La forma de distinguirlo del resto es bastante similar a los mensajes Offer, ya que los puertos de origen y destino usados son los mismos. Sin embargo, el campo que indica el tipo de mensaje tiene valor 5, permitiendo así la diferenciación entre un tipo de paquete y otro.

La mecánica de este caso es similar al anterior, en ambos casos se trata de detectar un servidor fraudulento.

Para el caso de *Offer* y *ACK* además de comprobar si se reciben dos mensajes iguales, pero con diferentes direcciones IP de origen, se ha querido detectar aquellas situaciones en las que la IP de origen de la oferta y la del asentimiento no coinciden.

La detección de los ataques relacionados con los mensajes *Offer* y *ACK* es posible siempre que estos se envíen de forma directa al ordenador en el que se arranca el sniffer o en aquellas situaciones en las que todos los dispositivos de la red reciban dichos mensajes, es decir, que el envío sea *broadcast*.





6.2.4.2.- HSRP

En este último caso, al igual que en el anterior, se avisa al administrador de la red de forma inmediata si se detecta un posible ataque. Es decir, si la herramienta detecta algún paquete HSRP cuyo origen no coincide con las puertas de enlace introducidas por el usuario durante la configuración de la misma, se envía una notificación.

Para la detección de estos mensajes se ha tenido en cuenta que HSRP utiliza UDP como protocolo de transporte y envía sus mensajes a través del puerto 1985 [64]. Asimismo, se ha decidido controlar ambas versiones del protocolo. Se sabe que los routers envían los paquetes a la dirección *multicast* 224.0.0.2 [66] (versión 1) o 224.0.0.102 [66] (versión 2), lo cual permite una detección más precisa de los mismos.

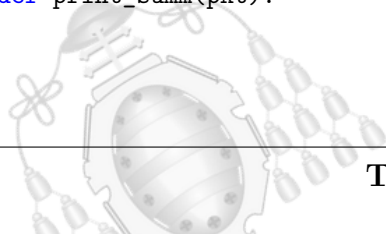
```
fltF="(udp and port 1985 and (dst host 224.0.0.2 or dst host 224.0.0.102))"
if(globV.form1.hsrpCheck.isChecked()):
    filtros.append(fltF)
```

Una vez seleccionados los protocolos que se quieren controlar, se crea el filtro general, compuesto de todos los añadidos anteriormente a la lista de filtros. Posteriormente, se arranca el sniffer como se puede ver a continuación:

```
sniff(filter=fltTotal,prn=print_summ,iface=typeConec,stop_filter=stop_alert)
```

En caso de detectar algún paquete que coincida con el filtro establecido, se llama a la función *print_summ* que determina como tratar cada paquete según el tipo. Para ello, se usa un diccionario que asocia a cada tipo de mensaje una función determinada. Estas se encargan de analizar en mayor detalle el paquete recibido y en caso de coincidir con la información deseada, envían la notificación correspondiente.

```
def print_summ(pkt):
```





```
fndict={"STP": print_sum_STP, "UDP": print_sum_udp, "ARP": print_sum_ARP, "IP":  
        print_sum_IP, "Dot3": print_sum_CISCO}  
keynames=["STP", "UDP", "ARP", "IP", "Dot3"]  
for proto in keynames:  
    if (proto in pkt):  
        return fndict[proto](pkt)
```

Para el envío de la notificación se crea un paquete SNMP como el mostrado a continuación, que se particulariza posteriormente para cada caso.

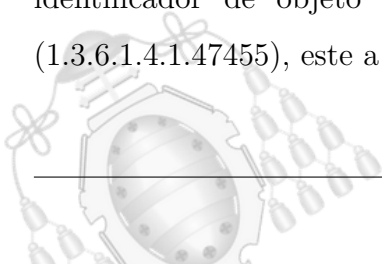
```
pkt_snmp=IP(dst=snmpServer)/UDP(sport=udpPort,  
                                dport=udpPort)/SNMP(community=comm,  
                                PDU=SNMPtrapv2(varbindlist=[SNMPvarbind(oid=evTypeOID,  
                                value=evTypeValue),SNMPvarbind(oid=trapOID, value=EvOID),  
                                SNMPvarbind(oid=macOID, value=macValue), SNMPvarbind(oid=ipOID,  
                                value=ipValue)]))  
send(pkt_snmp,verbose=False,iface=interf)
```

6.3.- Notificación

Una de las partes esenciales del proyecto es la notificación al administrador de los diferentes eventos cuando se detecta un paquete que puede ser notablemente peligroso, o indicativo de una mala configuración de la red. Se ha creado para ello dos MIBs que se deben de cargar en la herramienta SNMP de la persona que gestiona la red, en este caso MIB Browser (Figura 6.5). Estos son:

- *snObjs* MIB: Contiene los objetos incluidos en los diferentes eventos.
- *snEvnts* MIB: Engloba todos los eventos/notificaciones susceptibles de ser recibidos por parte de la herramienta.

Ambos están compuestos por diferentes elementos, no obstante, tienen algo en común, ya que todos estos elementos es posible identificarlos mediante un identificador de objeto (OID). En este caso los dos cuelgan del objeto sniffer (1.3.6.1.4.1.47455), este a su vez depende del nodo *enterprises* (1.3.6.1.4.1) [69].



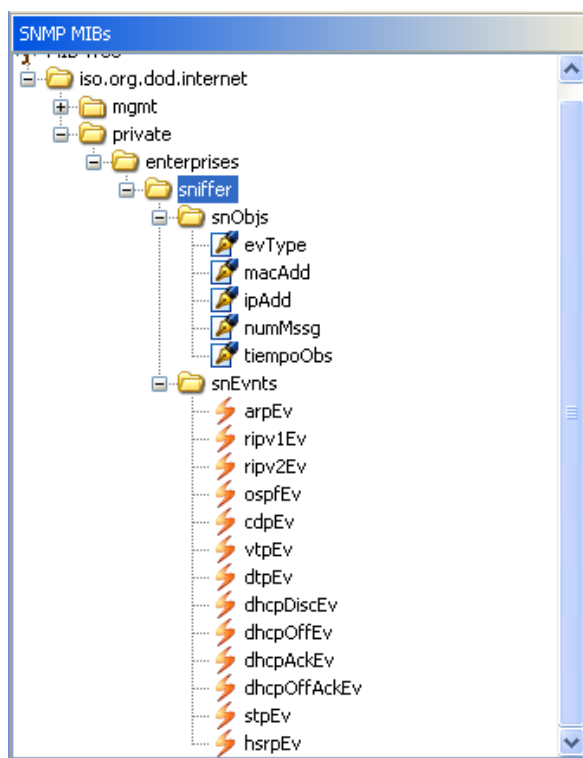
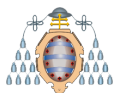


Figura 6.5.- MIBs cargados en MIB Browser

6.3.1.- *snObjs* MIB

Identificado por el OID 1.3.6.1.4.1.47455.1. Debajo del mismo se definen 5 tipos de objetos que ayudan a entender mejor el evento recibido:

- **evType** (1.3.6.1.4.1.47455.1.1.0): Identifica el tipo de evento que se recibe. Hay tres tipos, identificados todos por un número entero:
 - (1) El evento recibido avisa de una mala configuración del protocolo
 - (2) El evento recibido informa de un posible ataque
 - (3) El evento recibido avisa de una anomalía, puede ser una mala configuración o un posible ataque. Aplicable en aquellas circunstancias en las que no hay datos suficientes para distinguir si es una cosa u otra.

En su descripción se ha determinado que la sintaxis del objeto sea del tipo *Integer*, al cual se le han aplicado restricciones de rango, de forma que los únicos números aceptados son los indicados anteriormente.





- **macAdd** (1.3.6.1.4.1.47455.1.2.0): Indica una dirección MAC, el significado de la misma puede variar dependiendo del evento recibido. Esta información tiene como finalidad facilitar al administrador el arreglo de un problema de mala configuración o incluso interrupción de un ataque.

En su descripción se ha determinado que la sintaxis del objeto sea del tipo *MacAddress* (*octet string* de tamaño 6)³, importado desde SNMPv2-TC.

- **ipAdd** (1.3.6.1.4.1.47455.1.3.0): Indica una dirección IP, el significado de la misma puede variar dependiendo del evento recibido. Al igual que la dirección MAC tiene como propósito facilitar la resolución del problema, ya sea una mala configuración o un posible ataque.

En su descripción se ha determinado que la sintaxis del objeto sea del tipo *IpAddress* (*octet string* de tamaño 4)³, importado desde SNMPv1-SMI.

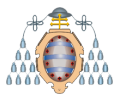
- **numMssg** (1.3.6.1.4.1.47455.1.4.0): Indica el número de mensajes recibidos en un período de tiempo concreto. Utilizado en aquellas situaciones en las que se detecta un posible ataque, proporcionando así al administrador más información de porqué se ha considerado como tal.

En su descripción se ha determinado que la sintaxis del objeto sea del tipo *Integer*, sin limitar dicho número ya que se desconoce el máximo de mensajes que se pueden recibir.

- **tiempoObs** (1.3.6.1.4.1.47455.1.5.0): Indica el período de tiempo establecido para la detección de múltiples mensajes del mismo tipo. Este valor es enviado en las mismas situaciones que *numMssg* para ayudar al administrador a comprender la gravedad del problema detectado. Ambos están estrechamente relacionados e identifican el mismo tipo de anomalía, es decir, un posible ataque.

En su descripción se ha determinado que la sintaxis del objeto sea del tipo *Integer*, y al igual que *numMssg* no se limita este número ya que se desconoce el tiempo máximo a establecer.

³Un octeto es una unidad digital de información, compuesta por 8 bits, o lo que es lo mismo, un byte [70]. Un 'octet string' es una secuencia de longitud variable de octetos.



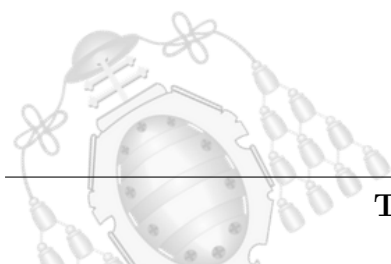
6.3.2.- *snEvents* MIB

Se identifica por el OID 1.3.6.1.4.1.47455.2. Debajo del mismo se definen 13 tipos de notificaciones que permiten identificar el protocolo e incluyen objetos de los nombrados anteriormente para describir mejor el evento recibido. Estas notificaciones son:

6.3.2.1.- *arpEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.1. Se envía ante la detección de alguna anomalía en el protocolo ARP, independientemente del tipo de configuración utilizada, ya sea estática o dinámica.

- **ARP estático:** La notificación enviada contiene los objetos *evType* y *macAdd*. El problema es desconocido ya que no se puede saber con exactitud si el mensaje ARP *Reply* detectado es una mala configuración en un dispositivo de la red o un posible ataque (*EvType*=3). Por ello, se añade la dirección MAC del dispositivo que emite dicho mensaje, delegando así en el administrador la responsabilidad de determinar de qué tipo de problema se trata.
- **ARP dinámico:** Es posible recibir dos tipos de notificaciones diferentes.
 - Caso 1: La notificación enviada contiene los objetos *evType*, *numMssg*, *tiempoObs*, *macAdd* e *ipAdd*. Informa de un posible ataque (*EvType*=2) del tipo ARP *Spoofing*, detallando el número de mensajes ARP *Reply* iguales que se reciben en un periodo de tiempo. Asimismo, indica la dirección MAC que envía estos mensajes y la dirección IP que anuncia.
 - Caso 2: La notificación enviada contiene los objetos *evType*, *macAdd* e *ipAdd*. Informa de un posible ataque (*EvType*=2) ante la detección de dos mensajes ARP *Reply* que anuncian la misma dirección IP para dos MACs diferentes.





6.3.2.2.- *ripv1Ev*

Se identifica por el OID 1.3.6.1.4.1.47455.2.2. Se distinguen dos tipos de notificaciones diferentes dependiendo de si se detecta una mala configuración o un posible ataque.

- Caso 1: La notificación enviada contiene los objetos *evType*, *macAdd* e *ipAdd*. Informa de una mala configuración (*evType*=1) del protocolo en algún elemento de la red, para facilitar la identificación del mismo se indica la dirección MAC y la IP de origen de dicho paquete RIPv1.
- Caso 2: La notificación enviada contiene los objetos *evType*, *macAdd* e *ipAdd*. Informa de un posible ataque (*evType*=2) ante la detección de un paquete RIPv1 con una dirección IP de origen diferente a la puerta de enlace. Con el fin de facilitar la detección del elemento fraudulento se incluye la dirección MAC de origen de dicho paquete, además de las direcciones IP: puerta de enlace y origen del paquete.

6.3.2.3.- *ripv2Ev*

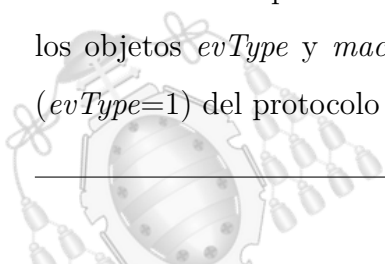
Se identifica por el OID 1.3.6.1.4.1.47455.2.3. Funcionamiento análogo al anterior, pero con la versión 2 del protocolo RIP.

6.3.2.4.- *ospfEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.4. Funcionamiento análogo al protocolo RIP, pero con paquetes OSPF.

6.3.2.5.- *cdpEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.5. La notificación enviada contiene los objetos *evType* y *macAdd*. Se envía ante la detección de una mala configuración (*evType*=1) del protocolo CDP. Con el fin de facilitarle al administrador la labor de





identificación del dispositivo que emitió este mensaje, se incluye la dirección MAC de origen del mismo.

6.3.2.6.- *vtpEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.6. Funcionamiento análogo al protocolo CDP, pero con paquetes VTP.

6.3.2.7.- *dtpEv*

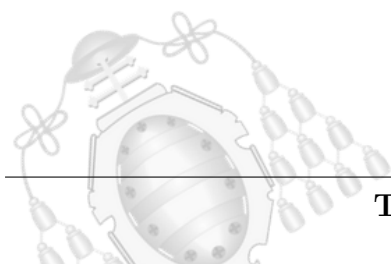
Se identifica por el OID 1.3.6.1.4.1.47455.2.7. Funcionamiento análogo al protocolo CDP, pero con paquetes DTP.

6.3.2.8.- *dhcpDiscEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.8. La notificación enviada contiene los objetos *evType*, *numMssg* y *tiempoObs*. Informa de un posible ataque (*evType=2*) del tipo DHCP *Starvation*, detallando el número de mensajes DHCP *Discovery* recibidos en un periodo de tiempo. De esta forma, el administrador puede determinar si un número elevado de paquetes en dicho intervalo de tiempo es sospechoso o no.

6.3.2.9.- *dhcpOffEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.9. La notificación enviada contiene los objetos *evType*, *macAdd* e *ipAdd*. Informa de un posible ataque (*evType=2*) ante la recepción de dos mensajes DHCP *Offer* diferentes. Debido a que no se conoce la dirección IP del servidor DHCP legítimo, la notificación incluye ambas direcciones MAC e IP, delegando en el administrador la responsabilidad de identificar al servidor DHCP fraudulento.





6.3.2.10.- *dhcpAckEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.10. Funcionamiento análogo al anterior, pero con mensajes DHCP *Ack*.

6.3.2.11.- *dhcpOffAckEv*

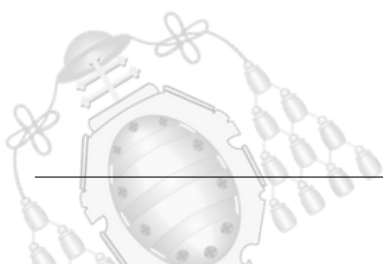
Se identifica por el OID 1.3.6.1.4.1.47455.2.11. Funcionamiento análogo a los dos anteriores, con una ligera diferencia. En este caso, se notifica si el servidor que envía el mensaje DHCP *Offer* y el DHCP *Ack* no es el mismo.

6.3.2.12.- *stpEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.12. La notificación enviada contiene los objetos *EvType* y *macAdd*. Se envía ante la detección de una mala configuración (*evType*=1) del protocolo STP. Junto con ello se añade la dirección MAC del *root*, permitiendo así que el administrador conozca quien desempeña este rol y pueda verificar que lo hace un switch determinado.

6.3.2.13.- *hsrpEv*

Se identifica por el OID 1.3.6.1.4.1.47455.2.13. La notificación enviada contiene los objetos *EvType*, *macAdd* e *ipAdd*. Informa de un posible ataque (*evType*=2) ante la detección de un paquete HSRP con una dirección IP de origen diferente a las puertas de enlace determinadas por el usuario. Con el fin de facilitar la detección del elemento fraudulento se incluye la dirección MAC y la IP de origen de dicho paquete.



7. Pruebas y Resultados

Con el fin de probar de forma adecuada la herramienta implementada se ha desplegado una red como la mostrada en la figura 7.1. Se ha hecho uso de esta topología para la realización de todas las pruebas mostradas a lo largo de este capítulo. Asimismo, la recepción de las notificaciones se hace mediante la herramienta MIB Browser.

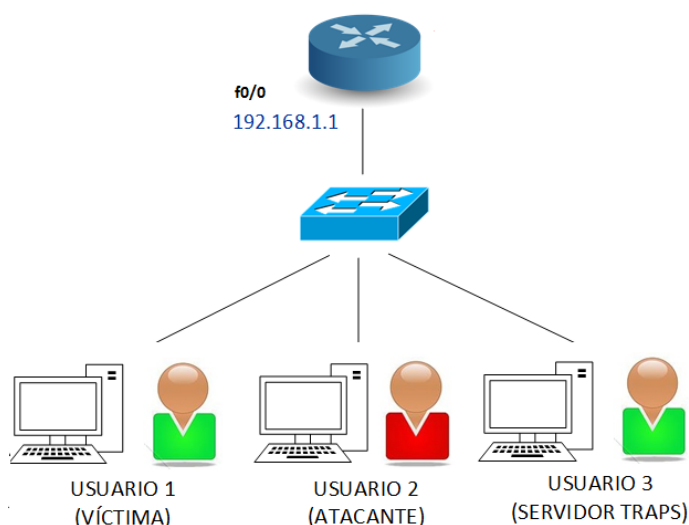


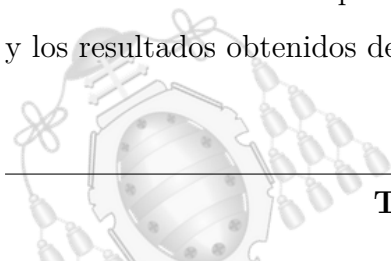
Figura 7.1.- Escenario de pruebas

7.1.- Pruebas individuales

Se ha decidido comenzar con la realización de pruebas individuales, es decir, seleccionando cada uno de los protocolos de forma independiente. De esta forma, la herramienta solo detecta paquetes de un protocolo concreto, y en caso de que estos coincidan con los patrones definidos en el código, se envía la notificación al servidor.

7.1.1.- Capa de enlace (Capa 2 según modelo OSI)

En esta sección se explican las pruebas realizadas para los protocolos ARP y STP, y los resultados obtenidos de las mismas.





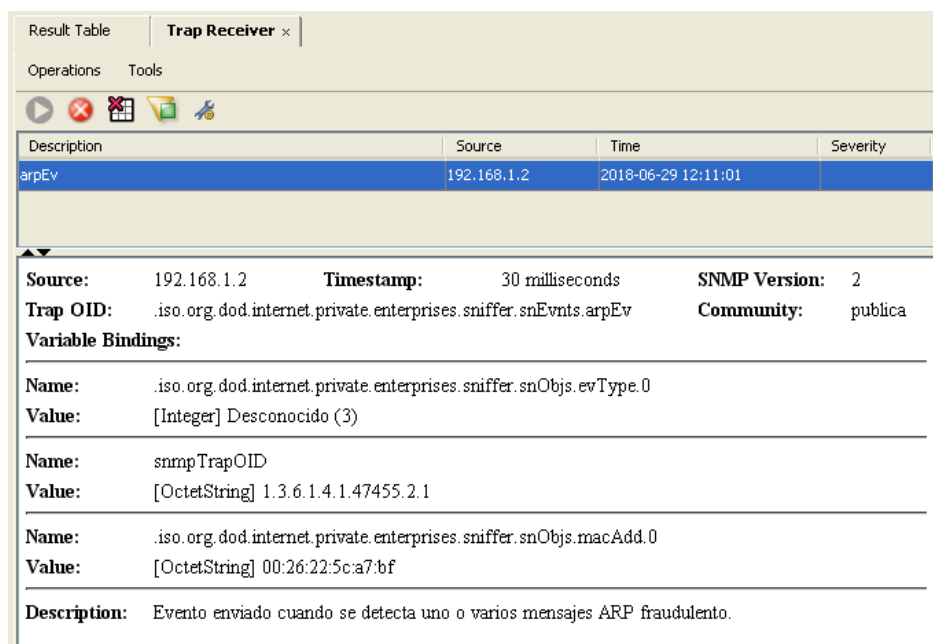
7.1.1.1.- ARP

Para la realización de las pruebas de este protocolo se distinguen dos casos:

- **Estático:** Esta configuración de la herramienta tiene como finalidad avisar al administrador ante la detección de cualquier mensaje ARP *Reply* en un escenario donde las tablas ARP de los equipos de la red se configuran de manera estática.

Como bien se muestra en la figura 7.2, una vez se detecta un mensaje de este tipo, la herramienta envía una trap con la dirección MAC del dispositivo emisor del mensaje, todo ello con el fin de facilitarle el administrador la identificación del problema.

Para la realización de esta prueba no ha sido necesario generar ningún paquete, se han aprovechado los generados de forma periódica por los dispositivos de la red.



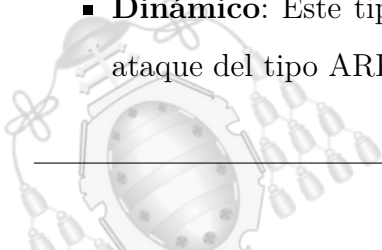
The screenshot shows a 'Trap Receiver' window with a table of received traps. The first trap is highlighted, showing details for an ARP event. Below the table, the trap details are expanded, showing source, timestamp, SNMP version, trap OID, community, variable bindings, and a description.

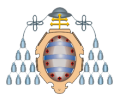
Description	Source	Time	Severity
arpEv	192.168.1.2	2018-06-29 12:11:01	

Source:	192.168.1.2	Timestamp:	30 milliseconds	SNMP Version:	2
Trap OID:	.iso.org.dod.internet.private.enterprises.sniffer.snEvnts.arpEv			Community:	publica
Variable Bindings:					
Name:	.iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0				
Value:	[Integer] Desconocido (3)				
Name:	snmpTrapOID				
Value:	[OctetString] 1.3.6.1.4.1.47455.2.1				
Name:	.iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0				
Value:	[OctetString] 00:26:22:5c:a7:bf				
Description:	Evento enviado cuando se detecta uno o varios mensajes ARP fraudulento.				

Figura 7.2.- Notificación ARP Estático

- **Dinámico:** Este tipo de configuración tiene como finalidad la detección de un ataque del tipo ARP *Spoofing*, para ello se han distinguido dos casos:





1. Detección de múltiples mensajes ARP *Reply* en un período de tiempo determinado, todos ellos provenientes del mismo dispositivo y anunciando la misma dirección IP.

Con el fin de facilitarle al administrador la resolución del problema, se le avisa de un posible ataque enviando la trap correspondiente. Además, en este se incluye la dirección MAC de dicho dispositivo y la IP anunciada (Figura 7.3a).

Este ataque se ha realizado haciendo uso de la herramienta arpspoof instalada en la distribución de Kali.

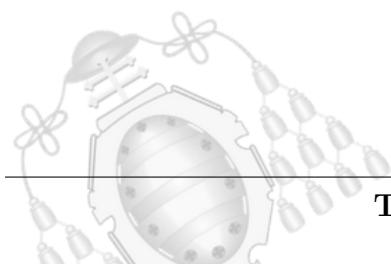
2. Detección de dos mensajes ARP *Reply* provenientes de diferentes dispositivos y anunciando la misma IP.

Al igual que en el caso anterior, se avisa al administrador de un posible ataque mediante el envío de la trap correspondiente. Además, en este se incluyen ambas direcciones MAC y la IP anunciada (Figura 7.3b).

Este ataque se ha realizado haciendo uso de Scapy, para ello se genera y se envía el paquete mostrado a continuación:

```
pkt=Ether(src=mac_router, dst=mac_usuario_1)/ARP(op=2,  
          hwsrc=mac_aleatoria, psrc=IP_puerta_enlace_router,  
          hwdst=mac_usuario_1, pdst=IP_usuario_1)  
sendp(pkt)
```

Asimismo, se han utilizado los mensajes ARP legítimos que se envían por la red desde la MAC del usuario 1.





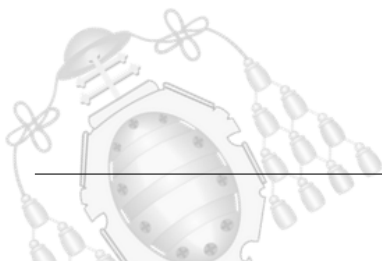
Result Table			
Trap Receiver x			
Operations Tools			
Description	Source	Time	Severity
arpEv	192.168.1.4	2018-06-30 18:41:34	
arpEv	192.168.1.4	2018-06-30 18:41:34	
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.1			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.numMsg.0			
Value: [Integer] 10			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.tiempoObs.0			
Value: [Integer] 20			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:1d:09:c2:e5:87			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.1.1			
Description: Evento enviado cuando se detecta uno o varios mensajes ARP fraudulento.			

(a) Posible ataque ARP - Caso 1

Result Table			
Trap Receiver x			
Operations Tools			
Description	Source	Time	Severity
arpEv	192.168.1.4	2018-06-30 18:41:34	
arpEv	192.168.1.4	2018-06-30 18:41:34	
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0			
Value: [Integer] Ataque (2)			
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.1			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:18:73:c3:37:16			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:1d:09:c2:e5:87			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.1.1			
Description: Evento enviado cuando se detecta uno o varios mensajes ARP fraudulento.			

(b) Posible ataque ARP - Caso 2

Figura 7.3.- Notificaciones ARP Dinámico



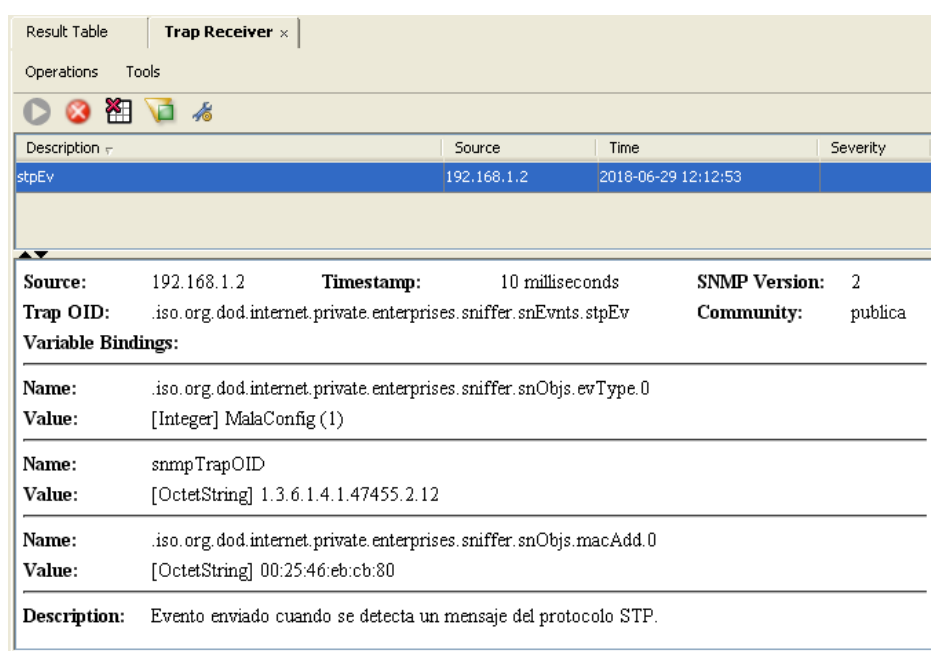


7.1.1.2.- STP

En este caso se avisa al administrador ante la detección de cualquier paquete STP, ya que se asume que es un indicio de una posible mala configuración del protocolo.

El aviso al administrador se realiza mediante el envío de la trap correspondiente, indicando el tipo de problema que se trata, es decir, una mala configuración. Además, se incluye la dirección MAC del *root bridge* (Figura 7.4), permitiendo así que el administrador pueda comprobar si coincide con el esperado o ha sufrido alguna alteración indeseada.

Para la realización de esta prueba no ha sido necesario generar ningún paquete, se han aprovechado los generados de forma periódica por el dispositivo de la red de capa 2.



Description	Source	Time	Severity
stpEv	192.168.1.2	2018-06-29 12:12:53	

Source:	192.168.1.2	Timestamp:	10 milliseconds	SNMP Version:	2
Trap OID:	.iso.org.dod.internet.private.enterprises.sniffer.snEvnts.stpEv			Community:	publica
Variable Bindings:					
Name:	.iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0				
Value:	[Integer] MalaConfig (1)				
Name:	snmpTrapOID				
Value:	[OctetString] 1.3.6.1.4.1.47455.2.12				
Name:	.iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0				
Value:	[OctetString] 00:25:46:eb:cb:80				
Description:	Evento enviado cuando se detecta un mensaje del protocolo STP.				

Figura 7.4.- Notificación STP

7.1.2.- Cisco

En esta sección se explican las pruebas realizadas para los protocolos CDP, DTP y VTP, y los resultados obtenidos de las mismas. La detección de cada uno de estos



protocolos se ha hecho de forma independiente. A pesar de ello, se incluyen todos juntos dentro de un mismo apartado con el fin de facilitar la lectura del documento.

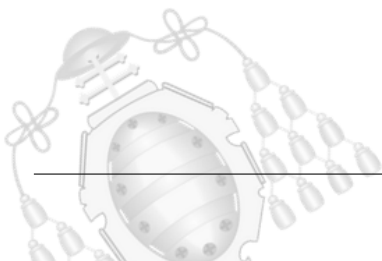
Se avisa al administrador ante la detección de un paquete del protocolo seleccionado, ya que se asume que es un indicio de mala configuración del mismo. El aviso al administrador se realiza mediante el envío de la trap correspondiente, el cual contiene la dirección MAC de origen de dicho paquete. Todo ello, con el fin de facilitar al administrador la identificación del problema. En las figuras 7.5, 7.6 y 7.7 se puede observar la recepción de dichas traps según se seleccione un protocolo u otro durante la configuración de la herramienta.

Para la realización de esta prueba no ha sido necesario generar ningún paquete, se han aprovechado los generados de forma periódica por el dispositivo de la red de capa 2.

7.1.2.1.- CDP

Result Table			
Trap Receiver x			
Operations Tools			
Description	Source	Time	Severity
cdpEv	192.168.1.2	2018-06-29 12:15:03	
Source: 192.168.1.2 Timestamp: 10 milliseconds SNMP Version: 2			
Trap OID: .iso.org.dod.internet.private.enterprises.sniffer.snEvnts.cdpEv Community: publica			
Variable Bindings:			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0			
Value: [Integer] MalaConfig (1)			
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.5			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:25:46:eb:cb:86			
Description: Evento enviado cuando se detecta un mensaje del protocolo CDP (Cisco).			

Figura 7.5.- Notificación CDP





7.1.2.2.- DTP

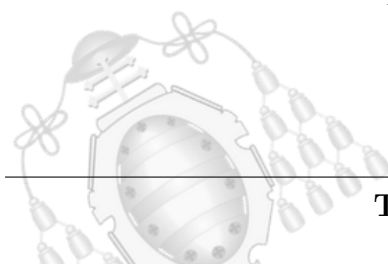
Result Table			
Trap Receiver x			
Operations Tools			
Description Source Time Severity			
dtpEv 192.168.1.2 2018-06-29 12:16:46			
Source: 192.168.1.2 Timestamp: 10 milliseconds SNMP Version: 2			
Trap OID: .iso.org.dod.internet.private.enterprises.sniffer.snEvnts.dtpEv Community: publica			
Variable Bindings:			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0			
Value: [Integer] MalaConfig (1)			
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.7			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:25:46:eb:cb:86			
Description: Evento enviado cuando se detecta un mensaje del protocolo DTP (Cisco).			

Figura 7.6.- Notificación DTP

7.1.2.3.- VTP

Result Table			
Trap Receiver x			
Operations Tools			
Description Source Time Severity			
vtpEv 192.168.1.2 2018-06-29 12:31:29			
Source: 192.168.1.2 Timestamp: 10 milliseconds SNMP Version: 2			
Trap OID: .iso.org.dod.internet.private.enterprises.sniffer.snEvnts.vtpEv Community: publica			
Variable Bindings:			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0			
Value: [Integer] MalaConfig (1)			
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.6			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:25:46:eb:cb:86			
Description: Evento enviado cuando se detecta un mensaje del protocolo VTP (Cisco).			

Figura 7.7.- Notificación VTP





7.1.3.- Capa de red (Capa 3 según modelo OSI)

En esta sección se explican las pruebas realizadas para los protocolos RIP y OSPF, y los resultados obtenidos de las mismas.

7.1.3.1.- RIP

Para la realización de las pruebas de este protocolo se distinguen dos casos:

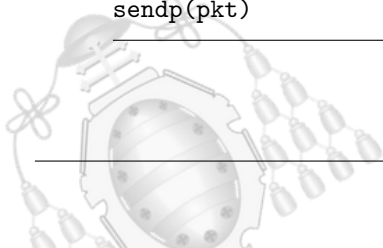
1. **Configuración errónea:** Se avisa al administrador ante la detección de un paquete RIP, ya que se asume que la recepción de estos mensajes por parte de los usuarios finales es fruto de una mala configuración del protocolo. Esta notificación contiene la dirección MAC e IP de origen de dicho paquete con el fin de facilitarle al administrador la resolución del problema. En la figura 7.8 se puede observar la recepción de dicha trap.

Para la realización de esta prueba no ha sido necesario generar ningún paquete, se han aprovechado los generados de forma periódica por el dispositivo de red de capa 3, estando este mal configurado de forma intencionada.

2. **Posible ataque:** Se avisa al administrador ante la detección de un paquete RIP procedente de una dirección IP diferente a la de la puerta de enlace legítima. Esta notificación contiene la dirección MAC fraudulenta y ambas direcciones IP, tanto la de la puerta de enlace real como la de origen del paquete (Figura 7.9). Al igual que en casos anteriores, la finalidad de esta información es facilitarle al administrador la resolución del problema.

Este ataque se ha realizado haciendo uso de Scapy, para ello se genera y se envía el paquete mostrado a continuación:

```
pkt=Ether(dst='01:00:5E:00:00:09')/IP(src='192.168.3.1',  
dst='224.0.0.9')/UDP(dport=520, sport=520)/RIP(version=2)  
sendp(pkt)
```





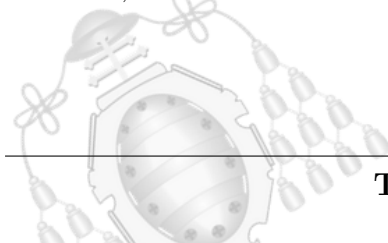
Result Table			
Trap Receiver x			
Operations Tools			
Description	Source	Time	Severity
ripv2Ev	192.168.1.2	2018-06-29 12:39:15	
Source: 192.168.1.2 Timestamp: 10 milliseconds SNMP Version: 2			
Trap OID: .iso.org.dod.internet.private.enterprises.sniffer.snEvnts.ripv2Ev Community: publica			
Variable Bindings:			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0			
Value: [Integer] MalaConfig (1)			
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.3			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:18:73:c3:37:16			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.1.1			
Description: Evento enviado cuando se detecta un mensaje del protocolo RIP v2.			

Figura 7.8.- Notificación configuración errónea RIPv2

Result Table			
Trap Receiver x			
Operations Tools			
Description	Source	Time	Severity
ripv2Ev	192.168.1.4	2018-06-30 18:50:16	
Variable Bindings:			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0			
Value: [Integer] Ataque (2)			
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.3			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:1d:09:c2:e5:87			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.1.1			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.3.1			
Description: Evento enviado cuando se detecta un mensaje del protocolo RIP v2.			

Figura 7.9.- Notificación posible ataque RIPv2

Las imágenes y el código mostrado pertenecen a la versión 2 del protocolo, no obstante, la herramienta contempla las dos primeras versiones.

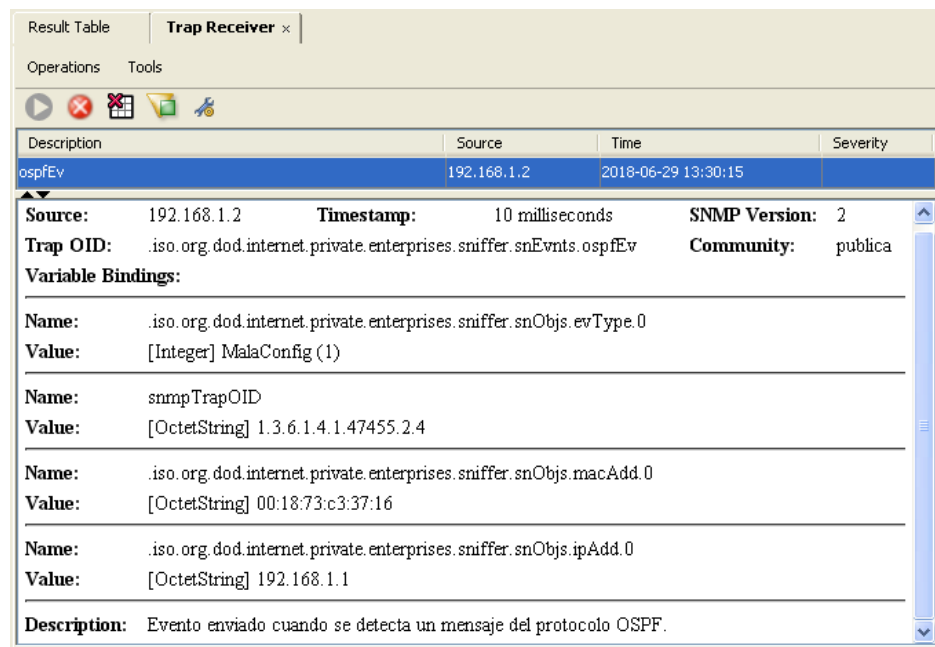




7.1.3.2.- OSPF

Para la realización de las pruebas de este protocolo se distinguen dos casos al igual que en RIP. Se siguen por ello los mismos criterios para la detección y el envío de traps, contando con una única diferencia entre ambos apartados: el cambio de protocolo. Los dos casos que se pueden distinguir son los siguientes:

1. **Configuración errónea** (Figura 7.10): Para la realización de las pruebas, al igual que en RIP, se han aprovechado los paquetes generados de forma periódica por el dispositivo de red de capa 3, estando este mal configurado de forma intencionada.
2. **Posible ataque** (Figura 7.11): En este caso la realización de las pruebas difiere un poco del anterior. El paquete generado con Scapy es ligeramente diferente, ya que este se adapta a OSPF. Por lo demás, el envío es idéntico a RIP.



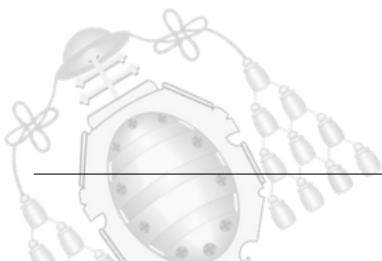
The screenshot shows a 'Trap Receiver' window with a table of trap notifications. The first notification is for 'ospfEv' from source '192.168.1.2' at time '2018-06-29 13:30:15'. Below the table, the details of the trap are shown, including the source, timestamp, SNMP version, trap OID, community, and variable bindings.

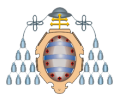
Description	Source	Time	Severity
ospfEv	192.168.1.2	2018-06-29 13:30:15	

Source: 192.168.1.2 **Timestamp:** 10 milliseconds **SNMP Version:** 2
Trap OID: .iso.org.dod.internet.private.enterprises.sniffer.snEvnts.ospfEv **Community:** publica
Variable Bindings:

Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0	Value: [Integer] MalaConfig (1)
Name: snmpTrapOID	Value: [OctetString] 1.3.6.1.4.1.47455.2.4
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0	Value: [OctetString] 00:18:73:c3:37:16
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0	Value: [OctetString] 192.168.1.1
Description: Evento enviado cuando se detecta un mensaje del protocolo OSPF.	

Figura 7.10.- Notificación configuración errónea OSPF










Result Table

Trap Receiver x

Operations

Tools



Description	Source	Time	Severity
ospfEv	192.168.1.4	2018-06-30 19:06:15	

Variable bindings:

Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0

Value: [Integer] Ataque (2)

Name: snmpTrapOID

Value: [OctetString] 1.3.6.1.4.1.47455.2.4

Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0

Value: [OctetString] 00:1d:09:c2:e5:87

Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0

Value: [OctetString] 192.168.1.1

Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0

Value: [OctetString] 192.168.3.1

Description: Evento enviado cuando se detecta un mensaje del protocolo OSPF.

Figura 7.11.- Notificación posible ataque OSPF

7.1.4.- Protocolos restantes

En esta sección se explican las pruebas realizadas para los protocolos DHCP y HSRP, y los resultados obtenidos de las mismas.

7.1.4.1.- DHCP

En este apartado se pueden distinguir múltiples pruebas realizadas, no obstante, se agrupan en dos casos diferentes:

1. **Posible ataque DHCP *Starvation*:** Si se pide más de un porcentaje determinado de direcciones al servidor DHCP en un período de tiempo establecido, se avisa al administrador ya que podría tratarse de un ataque.

Con el fin de que el administrador conozca lo que pasa en la red y pueda determinar si se trata de un ataque o no, se incluye el número de paquetes detectados y el período de tiempo en el que se detectaron.

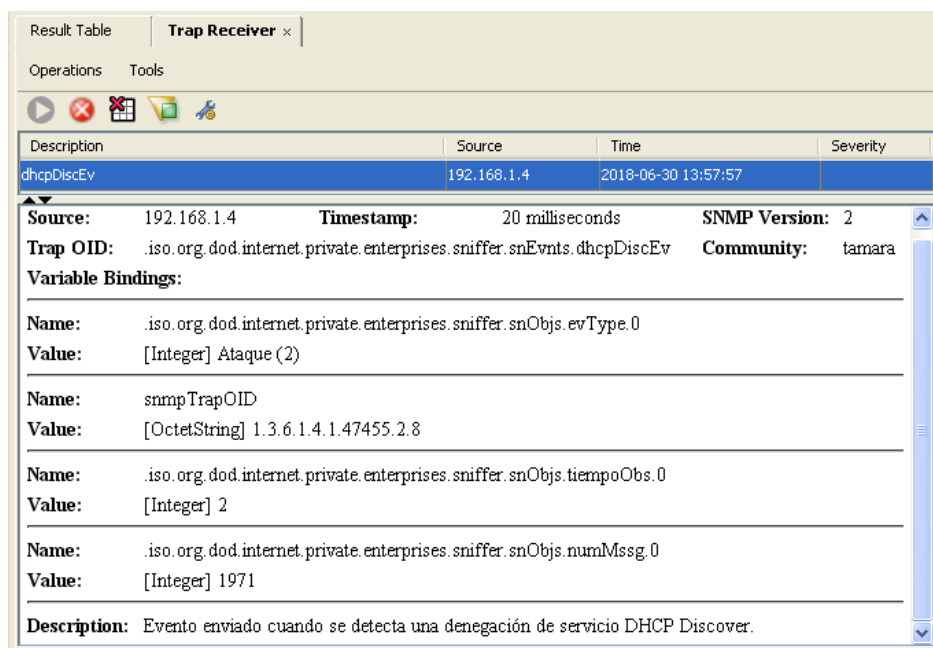
La simulación del ataque se ha realizado haciendo uso de la herramienta Yersinia,





instalada por defecto en la distribución de Kali. Las pruebas realizadas para este caso han sido las siguientes:

- **Red de clase C:** El número máximo de direcciones es 254. En la figura 7.12 se observa la trap enviada ya que se detecta la solicitud de más del 50 % de las direcciones en 2 segundos (intervalo por defecto configurado en la herramienta. Para más información consultar el manual de usuario).
- **Red de clase B:** El número máximo de direcciones es 65534. En la figura 7.13 se observa la trap enviada ya que se detecta la solicitud de más del 50 % de las direcciones en 64 segundos (intervalo por defecto configurado en la herramienta. Para más información consultar el manual de usuario).

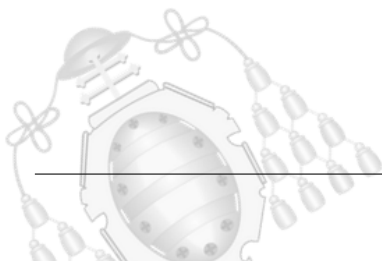


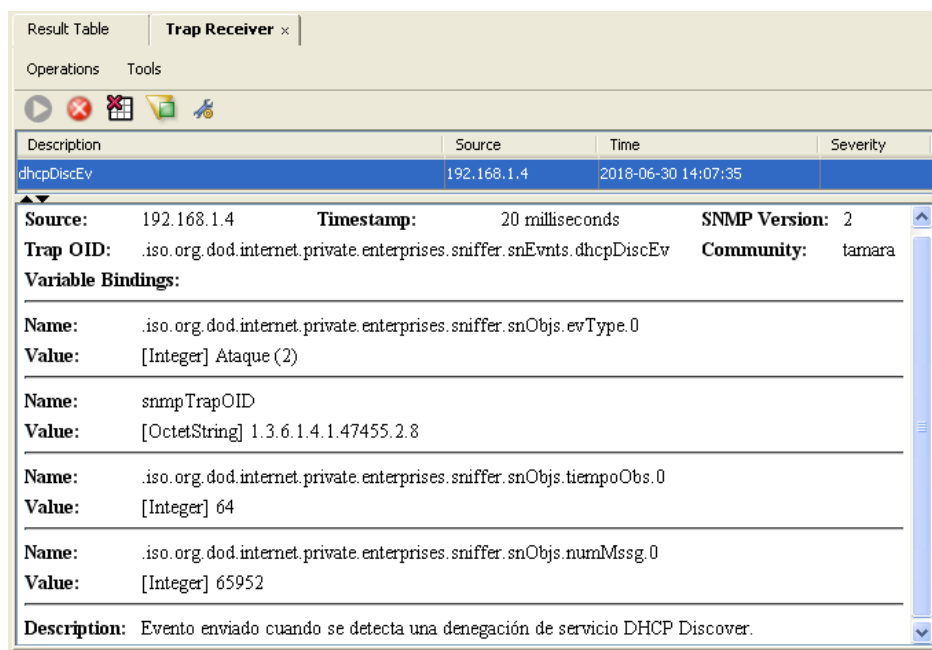
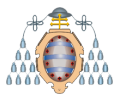
The screenshot shows a window titled "Trap Receiver" with a "Result Table" tab. It displays an SNMP trap event. The table has columns for Description, Source, Time, and Severity. Below the table, detailed information is provided for the selected event.

Description	Source	Time	Severity
dhcpDiscEv	192.168.1.4	2018-06-30 13:57:57	

Source:	192.168.1.4	Timestamp:	20 milliseconds	SNMP Version:	2
Trap OID:	.iso.org.dod.internet.private.enterprises.sniffer.snEvnts.dhcpDiscEv			Community:	tamara
Variable Bindings:					
Name:	.iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0				
Value:	[Integer] Ataque (2)				
Name:	snmpTrapOID				
Value:	[OctetString] 1.3.6.1.4.1.47455.2.8				
Name:	.iso.org.dod.internet.private.enterprises.sniffer.snObjs.tiempoObs.0				
Value:	[Integer] 2				
Name:	.iso.org.dod.internet.private.enterprises.sniffer.snObjs.numMssg.0				
Value:	[Integer] 1971				
Description: Evento enviado cuando se detecta una denegación de servicio DHCP Discover.					

Figura 7.12.- Notificación posible ataque DHCP *Discovery* - Clase C





The screenshot shows a 'Trap Receiver' window with a 'Result Table' tab. The table has columns for Description, Source, Time, and Severity. A single row is visible with the following data:

Description	Source	Time	Severity
dhcpDiscEv	192.168.1.4	2018-06-30 14:07:35	

Below the table, the details of the trap are displayed:

Source: 192.168.1.4 **Timestamp:** 20 milliseconds **SNMP Version:** 2
Trap OID: .iso.org.dod.internet.private.enterprises.sniffer.snEvnts.dhcpDiscEv **Community:** tamara

Variable Bindings:

Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.evType.0	Value: [Integer] Ataque (2)
Name: snmpTrapOID	Value: [OctetString] 1.3.6.1.4.1.47455.2.8
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.tiempoObs.0	Value: [Integer] 64
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.numMssg.0	Value: [Integer] 65952

Description: Evento enviado cuando se detecta una denegación de servicio DHCP Discover.

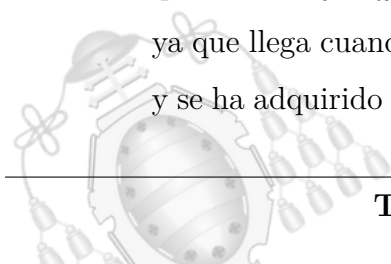
Figura 7.13.- Notificación posible ataque DHCP *Discovery* - Clase B

2. **Suplantación del servidor DHCP:** En caso de detectar mensajes Offer y ACK con procedencias distintas, se avisa al administrador ya que podría tratarse de un ataque.

La simulación del ataque se ha realizado haciendo uso de la herramienta Metasploit, instalada por defecto en la distribución de Kali. Se han tenido en cuenta dos posibles situaciones:

- **Anunciar una red distinta:** La red anunciada por el atacante es una red diferente a la que anuncia el servidor DHCP legítimo. Por ello, si el ataque tiene éxito, la víctima adquiere una IP que no está en la misma red que el servidor que recibe las traps. Si esto ocurre, el administrador no recibe notificaciones en dicho servidor, lo cual provoca que este desconozca lo que pasa en la red. Para evitarlo, la herramienta muestra por pantalla un mensaje con la información necesaria para saber qué protocolo y tipo de paquetes son los que provocan esta situación (Figura 7.14b).

Durante la realización de esta prueba se puede observar en la figura 7.14a que el mensaje *Offer* legítimo llega con bastante retraso respecto al ilegítimo, ya que llega cuando ha terminado la negociación con el servidor fraudulento y se ha adquirido una dirección IP falsa. No obstante, como bien se indicaba



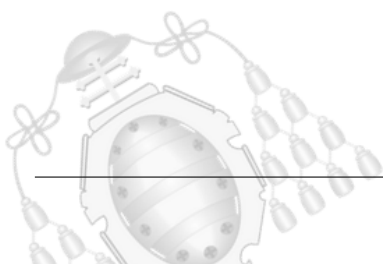


antes, la herramienta muestra un mensaje de advertencia con el fin de informar al administrador de que algo extraño ocurre en la red.

- **Anunciar la misma red:** En este caso se tiene una situación ligeramente diferente al anterior ya que se anuncia la misma red que el servidor DHCP legítimo.

En la figura 7.15a se puede observar que ocurre lo mismo que en el caso anterior, es decir, el mensaje *Offer* del servidor DHCP legítimo llega cuando ya ha terminado la negociación fraudulenta, aun así, la herramienta muestra un mensaje que avisa de un posible ataque (Figura 7.15b).

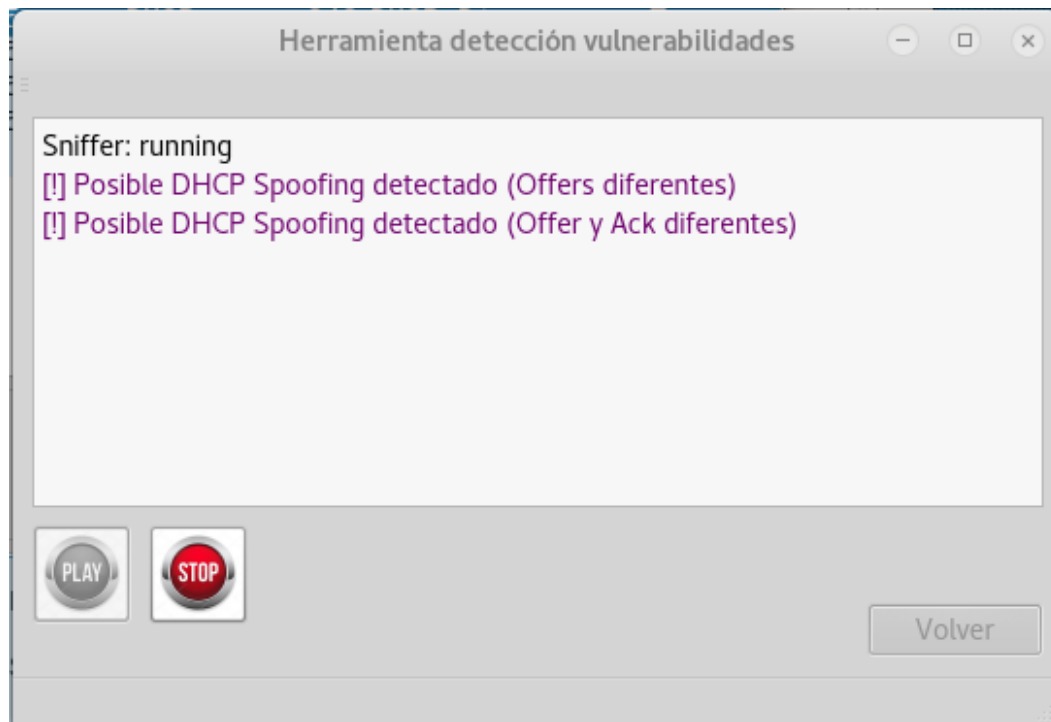
Además, bajo estas circunstancias sí que se envían las traps correspondientes al servidor, ya que este pertenece a la misma red. En la figura 7.16 se puede observar las traps que avisan de un posible ataque, estas contienen la dirección MAC e IP de los dos mensajes detectados. Toda esta información, al igual que en casos anteriores, tiene la finalidad de facilitarle al administrador la resolución del problema, ya que conociendo las direcciones MAC e IP legítimas se podría determinar que dispositivo está realizando el ataque.





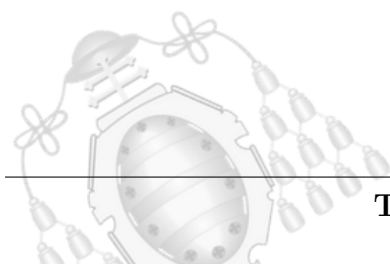
No.	Time	Source	Destination	Protocol	Length	Info
1044	741.161217532	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa415b224
1045	741.162200322	192.168.3.2	255.255.255.255	DHCP	372	DHCP Offer - Transaction ID 0xa415b224[Malformed Packet]
1046	741.162303754	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xa415b224
1047	741.163060089	192.168.3.2	255.255.255.255	DHCP	372	DHCP ACK - Transaction ID 0xa415b224[Malformed Packet]
1058	743.159973071	192.168.1.1	192.168.1.22	DHCP	342	DHCP Offer - Transaction ID 0xa415b224

(a) Captura *Wireshark*



(b) *Feedback* herramienta

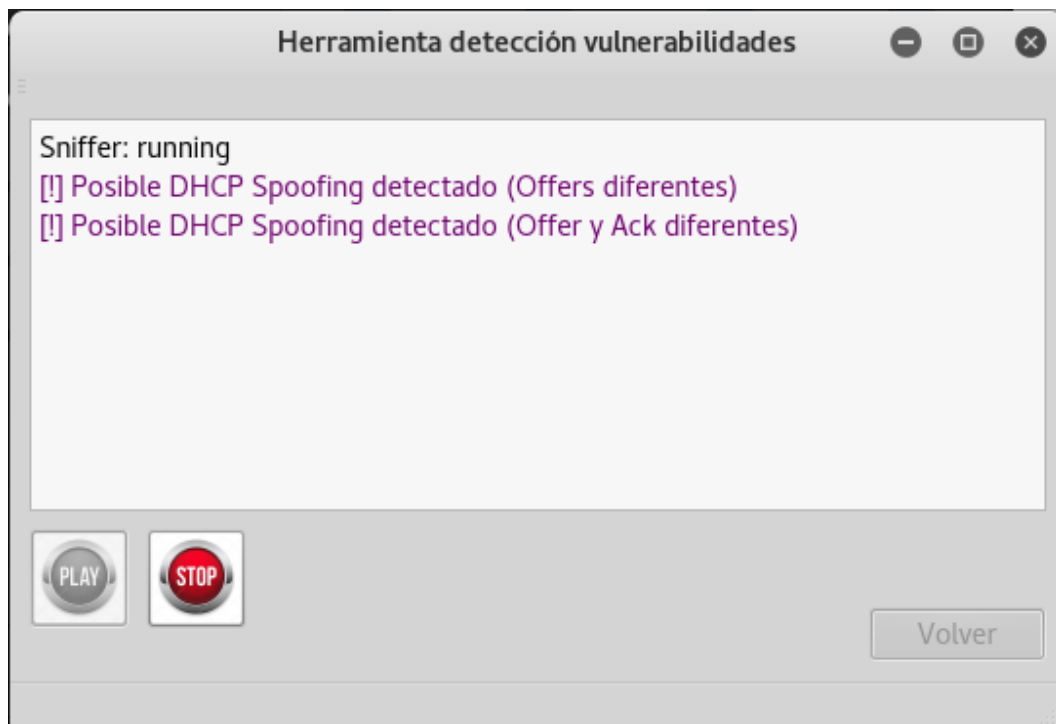
Figura 7.14.- Resultados DHCP anunciando dos redes diferentes





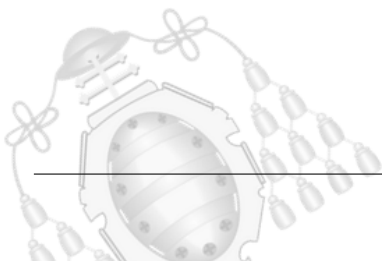
No.	Time	Source	Destination	Protocol	Length	Info
29	31.821089967	192.168.1.11	192.168.1.2	DHCP	342	DHCP Release - Transaction ID 0xe8d9a028
30	32.064254515	192.168.1.4	255.255.255.255	DHCP	342	DHCP Inform - Transaction ID 0x7d2924a4
43	45.909397081	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa67b4808
44	45.910385562	192.168.1.2	255.255.255.255	DHCP	372	DHCP Offer - Transaction ID 0xa67b4808[Malformed Packet]
45	45.910584642	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xa67b4808
46	45.911354242	192.168.1.2	255.255.255.255	DHCP	372	DHCP ACK - Transaction ID 0xa67b4808[Malformed Packet]
49	47.911361110	192.168.1.1	192.168.1.8	DHCP	342	DHCP Offer - Transaction ID 0xa67b4808

(a) Captura *Wireshark*



(b) *Feedback* herramienta

Figura 7.15.- Resultados DHCP anunciando la misma red





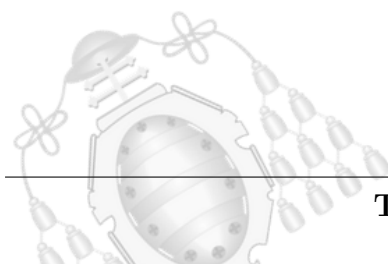
Result Table			
Trap Receiver x			
Operations Tools			
Description	Source	Time	Severity
dhcpOffEv	192.168.1.11	2018-07-03 19:37:20	
Value: [Integer] Ataque (2)			
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.9			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:1d:09:c2:e5:87			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.1.2			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:18:73:c3:37:16			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.1.1			
Description: Evento enviado cuando se detectan dos mensajes Offer de diferentes servidores.			

(a) Notificación Offers diferentes

Result Table			
Trap Receiver x			
Operations Tools			
Description	Source	Time	Severity
dhcpOffAckEv	192.168.1.11	2018-07-03 19:37:20	
Value: [Integer] Ataque (2)			
Name: snmpTrapOID			
Value: [OctetString] 1.3.6.1.4.1.47455.2.11			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:1d:09:c2:e5:87			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.1.2			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.macAdd.0			
Value: [OctetString] 00:18:73:c3:37:16			
Name: .iso.org.dod.internet.private.enterprises.sniffer.snObjs.ipAdd.0			
Value: [OctetString] 192.168.1.1			
Description: Evento enviado cuando se detectan un mensaje Offer y un mensaje ACK de diferentes servidores.			

(b) Notificación Offer - Ack diferentes

Figura 7.16.- Notificaciones DHCP anunciando la misma red





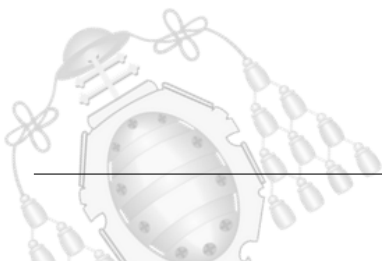
7.1.4.2.- HSRP

Para este caso se ha realizado la prueba para la versión 1 del protocolo, la cual consiste en el envío de un paquete HSRP fraudulento haciendo uso para ello de Scapy como se muestra en el código a continuación:

```
pkt=Ether(dst='01:00:5E:00:00:02')/IP(src='192.168.3.1',  
    dst='224.0.0.2')/UDP(dport=1985, sport=1985)  
sendp(pkt)
```

Ante la detección de este mensaje fraudulento se avisa al administrador mediante el envío de la trap correspondiente. Se consideran ilegítimos aquellos paquetes cuyo origen no coincide con ninguna de las puertas de enlace introducidas por el usuario durante la configuración de la herramienta. La notificación enviada contiene la dirección MAC e IP de origen del paquete fraudulento (Figura 7.17), facilitando así la localización del atacante.

Figura 7.17.- Notificación HSRP





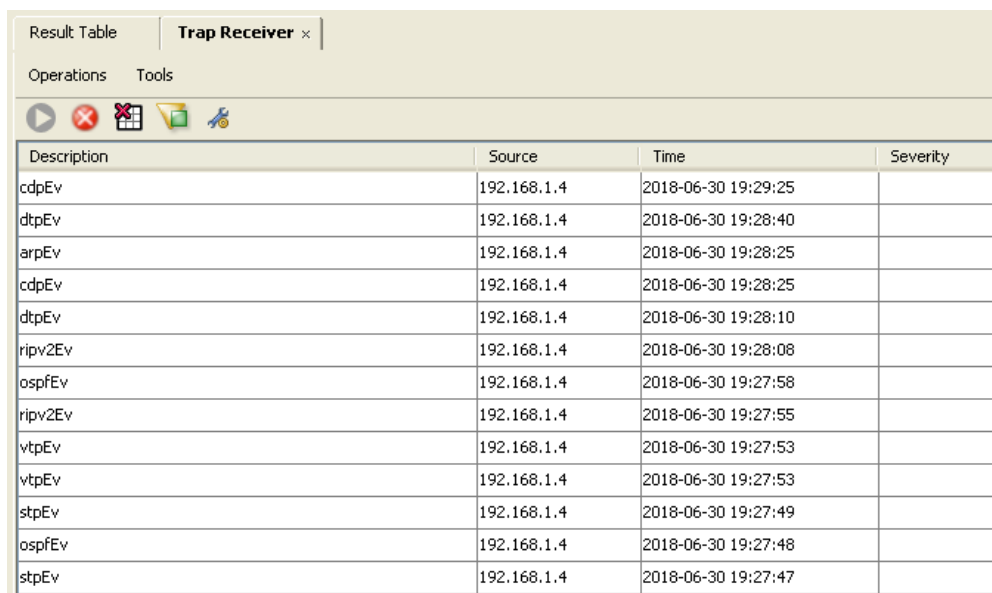
Las imágenes y el código mostrado pertenecen a la versión 1 del protocolo, no obstante, la herramienta contempla las dos versiones.

7.2.- Prueba genérica

Por último, se ha realizado una prueba global en la cual se muestra la recepción de paquetes de múltiples protocolos.

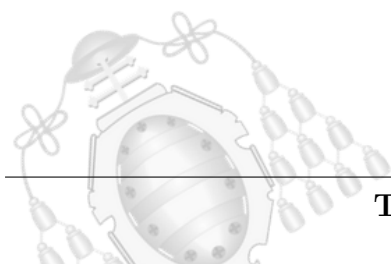
En la figura 7.18 se muestran mensajes de diversos protocolos, omitiendo la parte de ataques y centrándose únicamente en la detección de malas configuraciones. Para evitar que solo se mostrasen paquetes STP (frecuencia de envío muy alta) en la captura de pantalla se ha limitado el número de traps de cada protocolo a 2.

Algunos de los protocolos detectados durante la realización de esta prueba son: CDP (fila 1), DTP(fila 2), ARP (fila 3), RIPv2 (fila 6), OSPF (fila 7), VTP (fila 9) y STP (fila 11).



Description	Source	Time	Severity
cdpEv	192.168.1.4	2018-06-30 19:29:25	
dtpEv	192.168.1.4	2018-06-30 19:28:40	
arpEv	192.168.1.4	2018-06-30 19:28:25	
cdpEv	192.168.1.4	2018-06-30 19:28:25	
dtpEv	192.168.1.4	2018-06-30 19:28:10	
ripv2Ev	192.168.1.4	2018-06-30 19:28:08	
ospfEv	192.168.1.4	2018-06-30 19:27:58	
ripv2Ev	192.168.1.4	2018-06-30 19:27:55	
vtpEv	192.168.1.4	2018-06-30 19:27:53	
vtpEv	192.168.1.4	2018-06-30 19:27:53	
stpEv	192.168.1.4	2018-06-30 19:27:49	
ospfEv	192.168.1.4	2018-06-30 19:27:48	
stpEv	192.168.1.4	2018-06-30 19:27:47	

Figura 7.18.- Múltiples notificaciones





8. Manual de usuario

A continuación, se muestra de forma breve y sencilla como hacer uso de esta herramienta.

8.1.- Pasos previos

Si se hace uso de la herramienta por primera vez, o en un ordenador diferente, es necesario realizar los siguientes pasos antes de ejecutar el programa:

- **Instalar paquete *netifaces*:** Facilita el trabajo con direcciones de red, lo cual es necesario para la detección de algunos ataques. Para ello se ha de añadir la siguiente sentencia en la línea de comandos:

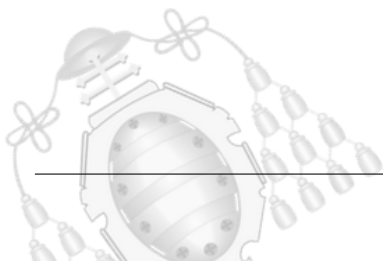
```
sudo easy_install netifaces
```

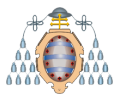
- **Actualizar la versión de Scapy:** Depende de la versión que se use podrían surgir fallos con el envío de los mensajes SNMP por un *bug* presente en la función *send*. Al igual que en el caso anterior se ha de añadir la sentencia mostrada a continuación en la línea de comandos:

```
pip install scapy --upgrade
```

8.2.- Ejecución de la herramienta

Una vez se han realizado estos pasos se puede proceder con el inicio de la herramienta. Esta cuenta con múltiples ventanas. A continuación, se muestra un diagrama de navegación (Figura 8.1) donde se indican las posibles rutas a seguir.





Más adelante en este capítulo se explica de forma detallada el mecanismo que se sigue durante la configuración de la herramienta.

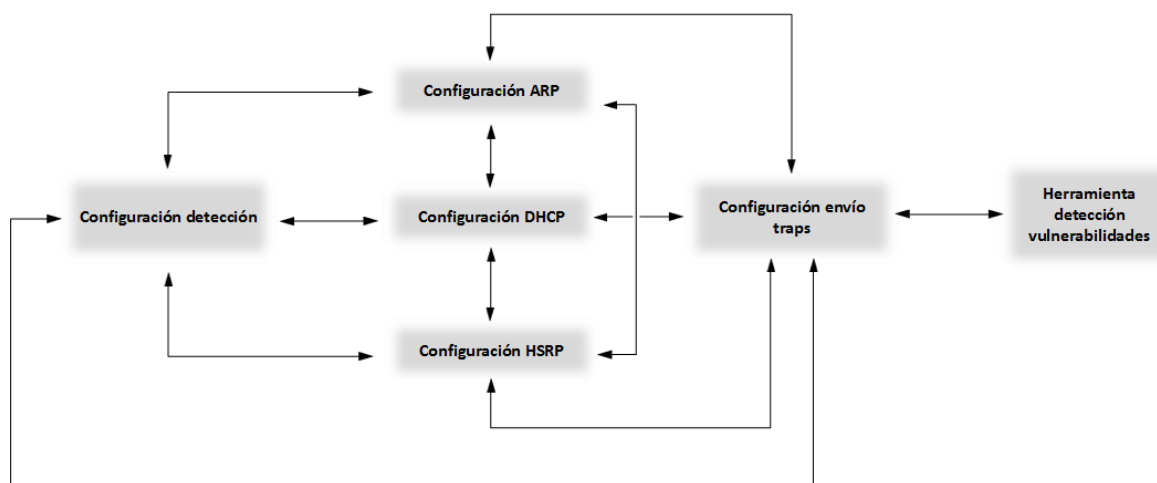


Figura 8.1.- Diagrama de navegación

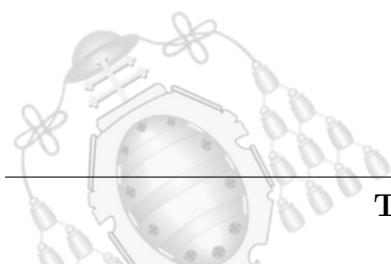
Para iniciar la herramienta es preciso añadir la siguiente sentencia en la línea de comandos:

```
python ubic/ConfInicial.py
```

donde *ubic* corresponde a la ubicación de la carpeta que contiene el script en Python que lanza la herramienta.

Una vez introducido este comando, se muestra la pantalla inicial de configuración de la herramienta (Figura 8.2). Esta pantalla ofrece la opción de elegir qué protocolos se quiere controlar, siendo necesaria la selección de al menos uno de estos para poder continuar con la configuración de la herramienta.

Por otro lado, también se ofrece la opción de elegir el tipo de conexión que se desea controlar: cableada o inalámbrica. Por defecto se presenta seleccionada la opción de conexión cableada.



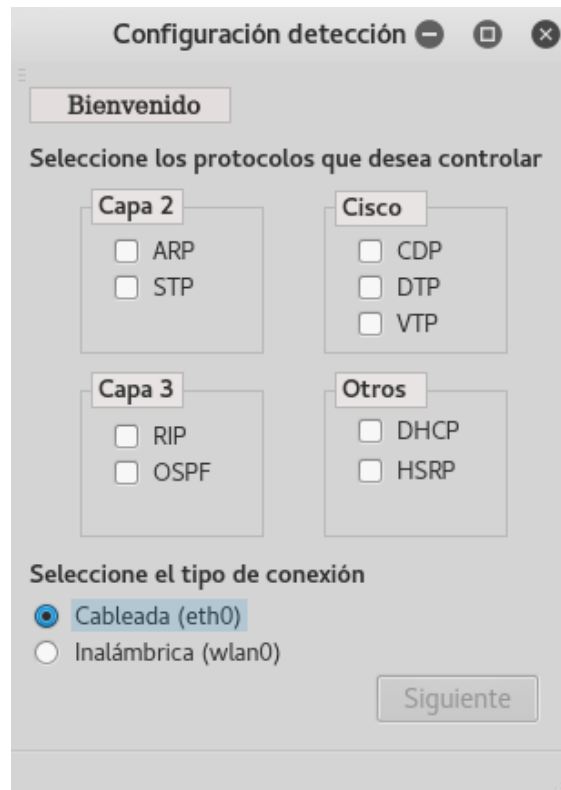


Figura 8.2.- Configuración detección

Cuando se pasa a la siguiente pantalla existen múltiples opciones según los protocolos seleccionados:

- **ARP seleccionado:** Se muestra una pantalla que permite elegir entre ARP estático o dinámico (Figura 8.3), por defecto está marcada la opción de ARP estático. Esta elección permite que el usuario de la aplicación indique el tipo de configuración de las tablas ARP que tiene.

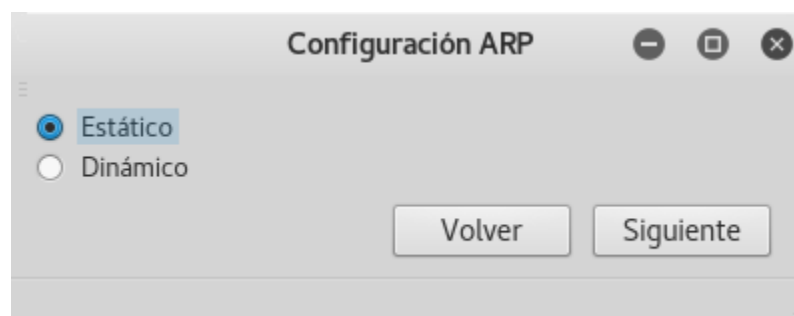
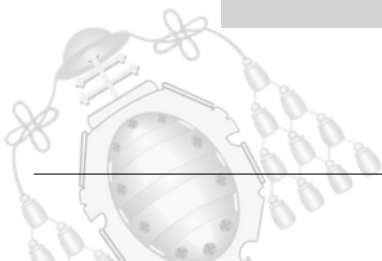


Figura 8.3.- Configuración ARP





- **DHCP seleccionado:** Se muestra una pantalla que permite configurar los parámetros necesarios para detectar un posible ataque de DHCP *Starvation* (Figura 8.4). Entre ellos se encuentra:
 - **Máscara de red:** Indica el tamaño de la red, es decir, el número de dispositivos que soporta la misma. Solo se permiten máscaras comprendidas entre 16 y 30, ambas inclusive. Por defecto el valor de la máscara es 24, es decir, una red de clase C.
 - **Porcentaje:** Indica el porcentaje mínimo de direcciones de la red que han de ser solicitadas en un período de tiempo para considerar un posible ataque. El valor introducido ha de estar comprendido entre 0 y 100. Por defecto el valor indicado es 50.
 - **Tiempo:** Indica el período de tiempo (en segundos) en el cual se ha de recibir un número de peticiones superior al porcentaje indicado para considerar un posible ataque. No hay límite superior de tiempo, no obstante, el valor introducido ha de ser siempre superior a 0. El valor por defecto de este campo depende de la máscara de red introducida. Los tiempos que recomienda la herramienta permiten detectar una solicitud superior al 50 % de las direcciones totales de la red. Todo ello tiene como finalidad facilitar la determinación de los parámetros durante la configuración de la herramienta.

Configuración DHCP

Máscara de red: 24

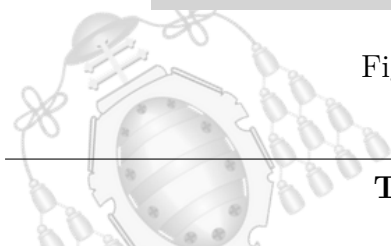
Rango máscara de red: 16-30.

Porcentaje de IPs de la red: 50

Período de tiempo: 2

Volver Siguiente

Figura 8.4.- Configuración DHCP



- **HSRP seleccionado:** Se muestra una pantalla que permite la configuración del protocolo HSRP (Figura 8.5), para ello es necesario introducir las direcciones IP de todas las puertas de enlace redundantes que tiene la red. Los valores correspondientes a dichas direcciones han de estar comprendidos entre 0 y 255. No se indica ningún valor por defecto. Sin embargo, es necesario introducir al menos una dirección IP para poder continuar con la configuración de la herramienta.

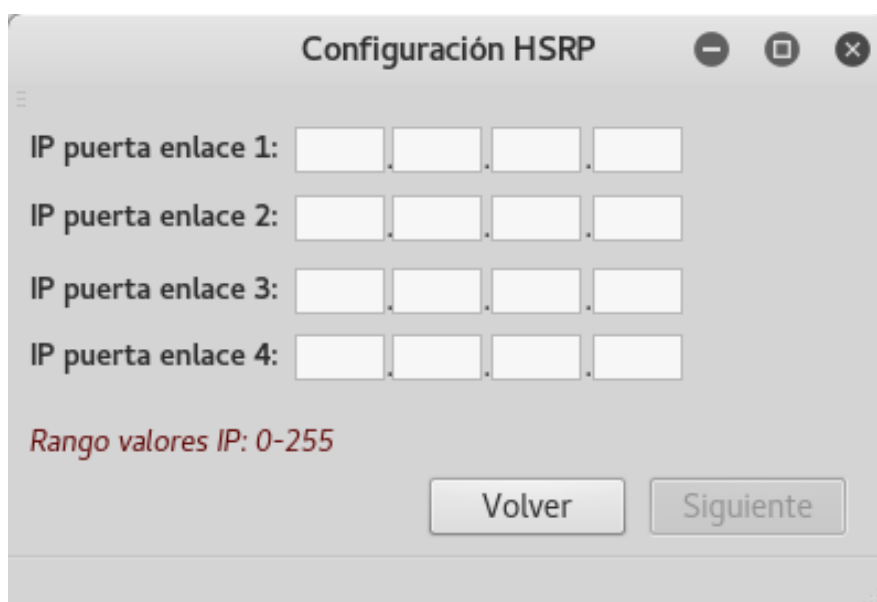
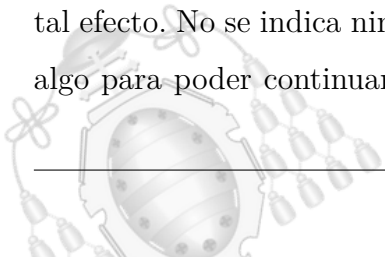


Figura 8.5.- Configuración HSRP

Independientemente de los protocolos seleccionados siempre se muestra la pantalla de configuración del envío de traps (Figura 8.6). En esta se ha de introducir la dirección IP del dispositivo donde está alojado el servidor SNMP. Los valores correspondientes a esta dirección han de estar comprendidos entre 0 y 255. No se indica ningún valor por defecto, sin embargo, es obligatorio introducir la dirección IP para poder continuar con la configuración de la herramienta.

La comunidad del servidor es otro campo obligatorio a parte de la dirección IP, esta tiene como finalidad añadir seguridad al envío de traps, sirviendo como contraseña a tal efecto. No se indica ningún valor por defecto, sin embargo, es obligatorio introducir algo para poder continuar con la configuración de la herramienta. No existe ninguna





limitación para este campo, ya que el valor introducido en el mismo podría ser cualquiera.

Además, esta pantalla permite limitar el número de traps de cada protocolo enviadas al servidor. Por defecto el envío no se encuentra limitado. En caso de ser marcada, se ha de rellenar el campo correspondiente al número de traps que se desea recibir, ya que se convierte en obligatorio para poder continuar con la configuración de la herramienta. No existe un límite superior para el número de traps, no obstante, el valor introducido ha de ser superior a 0.

Configuración envío traps

IP servidor Traps: [] [] [] []

Rango valores IP: 0-255

Comunidad servidor []

¿Desea limitar el número de avisos iguales? ☐ Sí ☒ No

Número de traps que desea recibir: []

Volver Siguiente

Figura 8.6.- Configuración envío traps

Por último, se tiene la pantalla que permite arrancar y parar la herramienta, además de proporcionar una retroalimentación para que el usuario pueda saber de forma breve lo que ocurre en la red (Figura 8.7). Los mensajes mostrados en esta pantalla son muy breves, pero permiten determinar si algún evento interesante desde el punto de vista de la seguridad de la red está ocurriendo, para poder comprobarlo posteriormente en el servidor SNMP de forma más precisa.

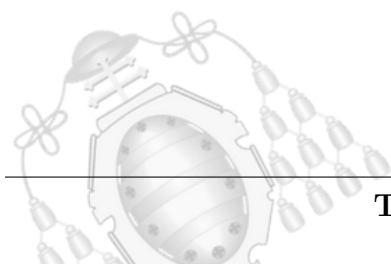
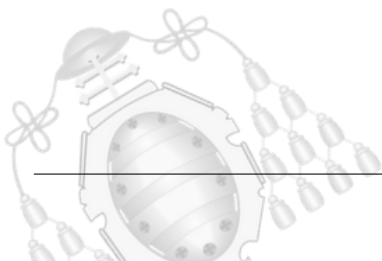
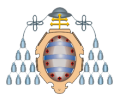




Figura 8.7.- Herramienta detección vulnerabilidades

Como se ha podido observar en las diversas capturas de pantalla de esta sección es posible volver hacia atrás en caso de querer cambiar alguno de los campos configurados previamente, proporcionando así una mayor flexibilidad en la configuración de la herramienta.





9. Planificación

En este capítulo se detalla la planificación temporal del proyecto, incluyendo una breve descripción de las principales tareas realizadas, así como la organización temporal de las mismas.

9.1.- Organización temporal

Se considera como fecha formal de inicio de este proyecto el 29 de enero de 2018 y como fecha de finalización el 6 de julio de 2018 (Figura 9.1).

9.2.- Tareas realizadas

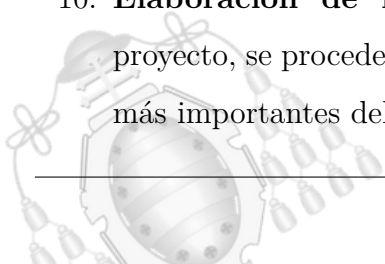
Las principales tareas realizadas en este proyecto son:

1. **Enfoque del proyecto:** Se marcan los objetivos que se pretenden lograr con el presente proyecto y los pasos necesarios para una correcta realización del mismo.
2. **Búsqueda de información:** Recopilación de información necesaria para realizar el trabajo de forma adecuada. Para ello ha sido preciso buscar información acerca de protocolos peligrosos en una red corporativa, así como de la distribución de Kali Linux y las herramientas necesarias (Python, Scapy y MibBrowser).
3. **Fijar protocolos a controlar:** Una vez recopilada la información relacionada con protocolos potencialmente peligrosos, se determina cuáles de estos han de ser detectados por la herramienta.
4. **Instalación necesaria:** Para comenzar con la realización del proyecto ha sido necesaria la instalación de la distribución de Kali Linux y los diferentes



entornos de trabajo necesarios (PyCharm, PyQt y MIB Browser).

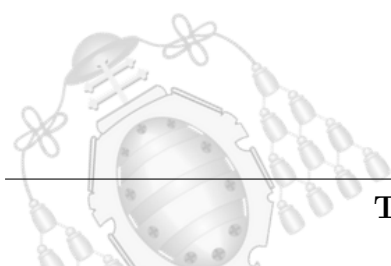
5. **Creación del script que detecta los protocolos:** Después de recopilar toda la información necesaria, y tener todas las herramientas listas para empezar a trabajar, se programa la detección de los protocolos previamente fijados. Para ello se hace uso de Scapy, teniendo siempre en cuenta los detalles específicos de cada protocolo para evitar falsos positivos.
6. **Diseño de Mibs:** Antes de proceder con el envío traps que avisen de un problema en la red, es preciso definir MIBs que ayuden a comprender estos mensajes recibidos. Para ello se definen dos MIBs diferentes: uno de objetos y otro de eventos.
7. **Programación de envío de notificaciones:** Haciendo uso nuevamente de Scapy, se generan las traps que se envían al servidor SNMP ante la detección de alguna anomalía.
8. **Creación de interfaz de usuario:** Uno de los propósitos principales perseguidos es la elaboración de una herramienta intuitiva y sencilla de usar para un administrador de red no muy experimentado. Por ello, una vez hecha la parte de detección y notificación se procede con la realización de una interfaz gráfica que haga más simple el uso de esta herramienta.
9. **Realización de pruebas en una LAN:** Con el fin de comprobar la herramienta implementada esta es sometida múltiples pruebas, que posteriormente son documentadas, con el objetivo de demostrar el correcto funcionamiento de la misma.
10. **Elaboración de la documentación:** Tras terminar la parte técnica del proyecto, se procede con la elaboración de una memoria explicativa de los puntos más importantes del trabajo.





EDT	Tareas
1	Enfoque del proyecto
2	Búsqueda de información
2.1	Búsqueda de información de protocolos peligrosos en una LAN
2.2	Búsqueda de información de Python y Scapy
2.3	Búsqueda de información de Kali
2.4	Búsqueda de información de MIB Browser
3	Fijar protocolos a controlar
4	Instalación necesaria
4.1	Instalación de Kali
4.2	Instalación de PyCharm
4.3	Instalación PyQt
4.4	Instalación de MIB Browser
5	Creación del script que detecta los protocolos
6	Diseño de Mibs
6.1	Diseño de Mibs de objetos
6.2	Diseño de Mibs de eventos
7	Programación de envío de notificaciones
8	Creación interfaz de usuario
9	Realización de pruebas en una LAN
10	Elaboración de documentación

Cuadro 9.1.- Tareas realizadas



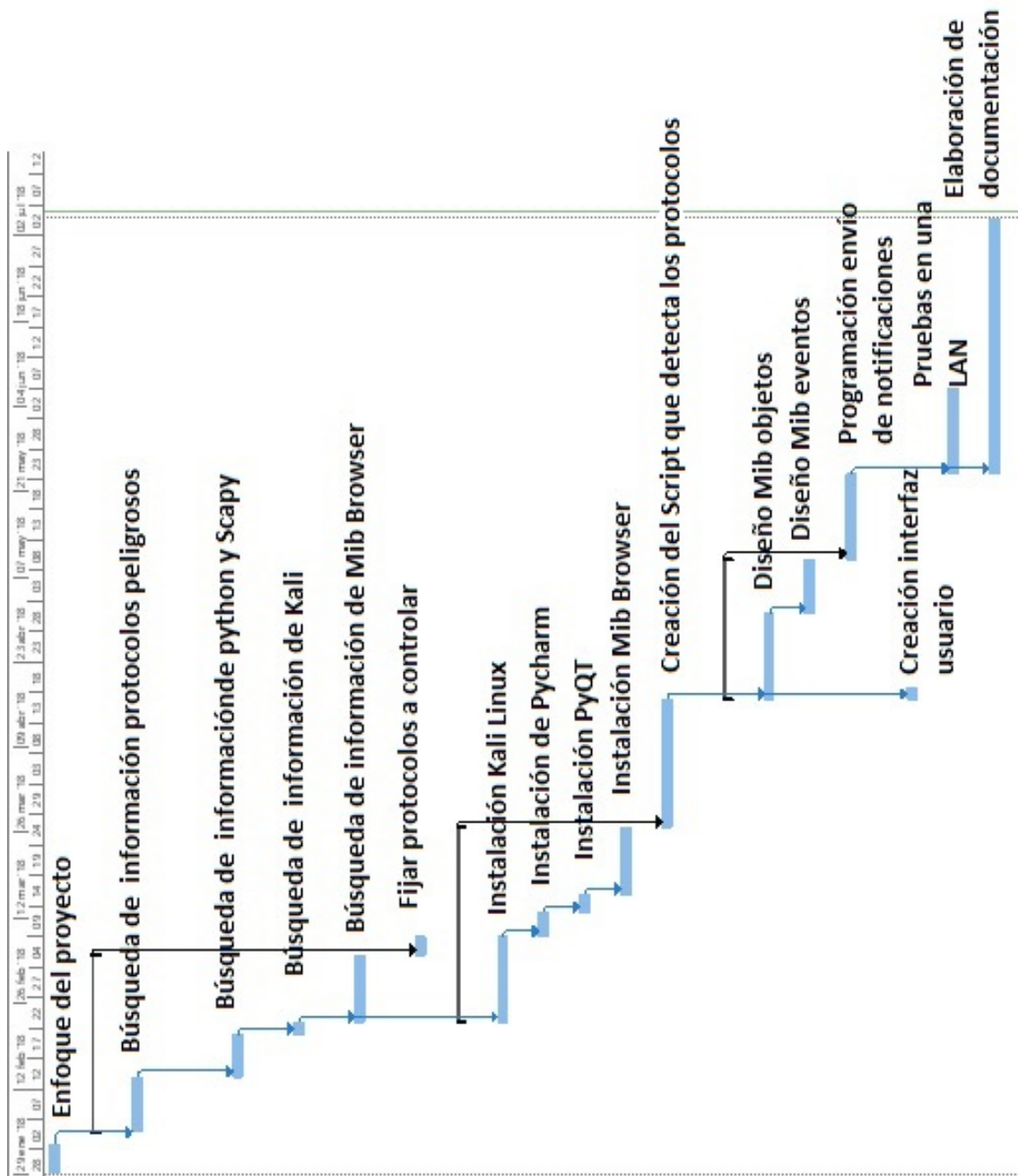
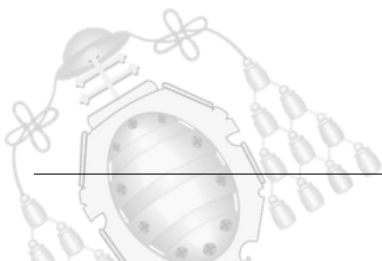


Figura 9.1.- Diagrama de Gantt





10. Conclusiones

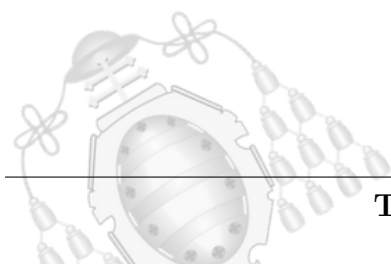
Tras finalizar la realización de la herramienta se puede concluir que se han cumplido todos los objetivos indicados al comienzo del documento. Todo ello, en un período de tiempo adecuado.

Se ha conseguido elaborar una herramienta capaz de detectar configuraciones erróneas y posibles ataques con una alta fiabilidad. Asimismo, se ha creado una interfaz intuitiva y simple que facilite el uso de esta herramienta a un usuario con poca experiencia.

Para la realización de este proyecto ha sido necesaria una labor de investigación exhaustiva que permitiese conocer los principales problemas que pueden acarrear algunos de los protocolos más comunes en redes corporativas. Además, se han aumentado ampliamente los conocimientos de Python adquiridos el primer año de carrera, donde únicamente se introducen conceptos básicos. También ha sido necesario aprender a trabajar con los paquetes de los diferentes protocolos a bajo nivel, haciendo uso para ello de la herramienta Scapy. Por último, en la parte de desarrollo se han reforzado los conocimientos de SNMP y Mib Browser aprendidos en la asignatura de Gestión de Redes de Telecomunicación.

A la hora de realizar las pruebas ha sido necesario aprender a usar herramientas del ámbito de la ciberseguridad, instaladas por defecto en la distribución de Kali Linux: Yersinia, arpspoof y metasploit.

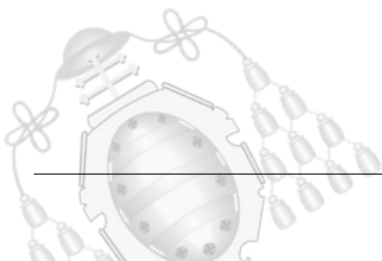
Por otro lado, gracias a este trabajo se han afianzado los conocimientos adquiridos estos últimos años durante la especialización en el área de telemática. Se han podido conocer las implicaciones que tendría una configuración errónea de los protocolos desde el punto de vista de la ciberseguridad.





Todos los factores nombrados con anterioridad han contribuido a desarrollar la capacidad de defenderse ante cualquier situación de forma autónoma, incrementando así la habilidad de auto aprendizaje.

Desde un punto de vista más personal, este proyecto me ha facilitado el acercamiento al mundo de la seguridad, ya que a lo largo de la carrera no se imparten apenas asignaturas relacionadas con ello. Por ello, este trabajo podría considerarse el primer paso de una carrera profesional dedicada a la ciberseguridad.

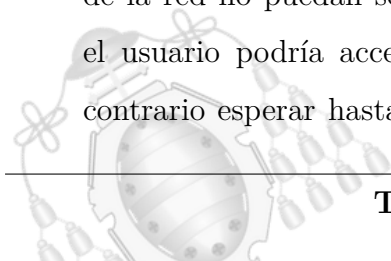




11. Líneas futuras

A continuación, se muestran algunas opciones de mejora para la herramienta, las cuales podrían llevarse a cabo en un futuro con el fin de perfeccionar la misma:

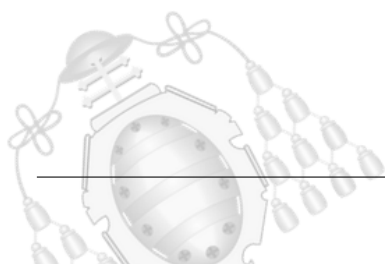
- Algunos protocolos no se han tenido en cuenta debido a que en el laboratorio donde se realizaron las pruebas no se disponía del equipamiento necesario para la realización de pruebas que acrediten el correcto funcionamiento de la herramienta. Algunos de los protocolos que se podrían añadir son los siguientes:
 - Routing: EIGRP, IGRP.
 - Anunciamiento de VLANs: MVRP, GVRP.
 - Redundancia de Routing: GLBP.
- Estudiar las diferentes formas de despliegue de la herramienta en una red corporativa real, algunas opciones serían:
 - Desplegarla únicamente en un ordenador de la red.
 - Desplegarla en múltiples ordenadores (por ejemplo: cada switch debe tener conectado, como mínimo, un ordenador con esta herramienta).
 - Desplegarla en todos los ordenadores de la red.
 - Usar *port mirroring*: Permite enviar por un puerto determinado (donde se colocaría el ordenador con esta herramienta) una copia de la información que llega a otros puertos del switch. El único problema de esta opción es que ese puerto solo acepta tráfico de control proveniente del dispositivo, lo cual complica el envío de traps. Para solucionarlo sería necesario hacer uso de una conexión inalámbrica por la que enviar dichos mensajes al servidor.
- Incluir un fichero en el que almacenar aquellas traps que por alguna condición de la red no puedan ser enviadas en un determinado momento. De esta forma, el usuario podría acceder a este fichero para comprobar que ocurre o por el contrario esperar hasta que se recupere la conexión con el servidor y se envíen

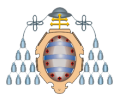




las traps correspondientes.

- Con el fin de añadir una mayor autonomía a la herramienta, podría estudiarse la viabilidad de reconfigurar los dispositivos de red de forma inmediata sin necesidad de que intervenga el administrador mediante el uso del protocolo SNMP.

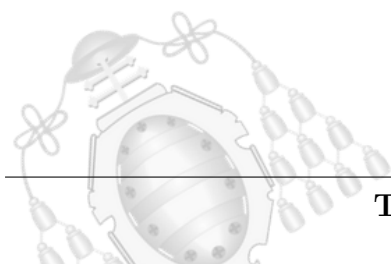




12. Glosario

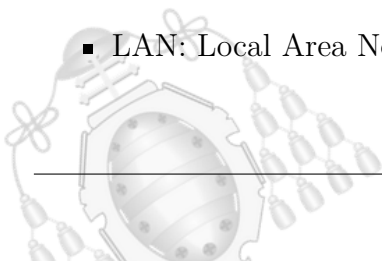
En esta sección se indica el significado de los diferentes acrónimos encontrados a lo largo de este documento.

- ABR: Area Border Router
- ACL: Access Control List
- ARP: Address Resolution Protocol
- ASBR: Autonomous System Boundary Router
- BDR: Backup Designated Router
- BID: Bridge Identifier
- BPDU: Bridge Protocol Data Units
- BR: Backbone Router
- CDP: Cisco Discovery Protocol
- CIDR: Classless Inter-Domain Routing
- DAI: Dynamic ARP Inspection
- DDoS: Distributed Denial of Service



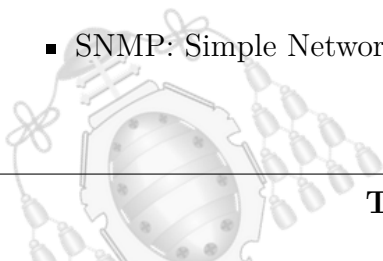


- DHCP: Dynamic Host Configuration Protocol
- DNS: Domain Name System
- DoS: Denial of Service
- DR: Designated Router
- DTP: Dynamic Trunk Protocol
- EIGRP: Enhanced Interior Gateway Routing Protocol
- FHRP: First Hop Redundancy Protocol
- GLBP: Gateway Load Balancing Protocol
- GVRP: Generic VLAN Registration Protocol
- HSRP: Hot Standby Router Protocol
- HTTPS: Hypertext Transport Protocol Secure
- IGP: Interior Gateway Protocol
- IGRP: Interior Gateway Routing Protocol
- IP: Internet Protocol
- IR: Internal Router
- LAN: Local Area Network



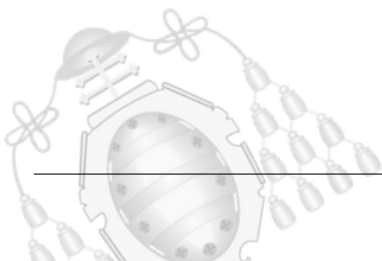


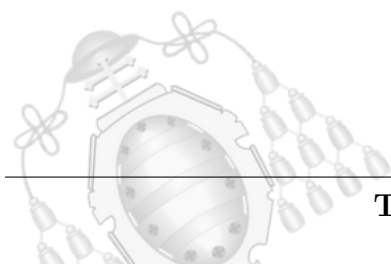
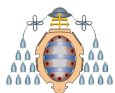
- LSA: Link-State Advertisement
- LSAck: Link-State Acknowledgment
- LSP: Link-State Packet
- LSR: Link-State Request
- LSU: Link-State Update
- MAC: Media Access Control
- MD5: Message-Digest Algorithm 5
- MIB: Management Information Base
- MVRP: Multiple VLAN Registration Protocol
- NSSA: Not-So-Stubby Area
- OID: Object Identifier
- OS: Operating System
- OSI: Open System Interconnection
- OSPF: Open Shortest Path First
- RIP: Routing Information Protocol
- SNMP: Simple Network Management Protocol





- SPoF: Single Point of Failure
- SSH: Secure Shell
- STP: Spanning Tree Protocol
- TCP: Transmission Control Protocol
- TLS: Transport Layer Security
- TLV: Type-Length-Value
- TTL: Time To Live
- UDP: User Datagram Protocol
- VACL: VLAN Access Control List
- VLAN: Virtual LAN
- VLSM: Variable Length Subnet Mask
- VTP: VLAN Trunking Protocol







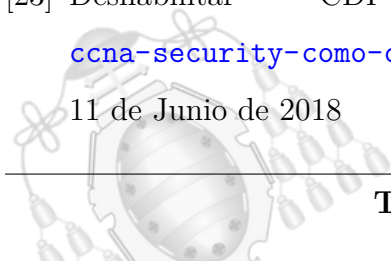
Bibliografía

- [1] Comparación pentesting vs análisis de vulnerabilidades <http://purplesec.com/pentest-vs-nva/> 24 de Junio de 2018
- [2] Explicación del análisis de vulnerabilidades <http://blog.cerounosoftware.com.mx/que-es-un-analisis-de-vulnerabilidades-informaticas> 24 de Junio de 2018
- [3] *NetworkRecon.ps1* tool <https://www.sans.org/reading-room/whitepapers/access/identifying-vulnerable-network-protocols-powershell-37722> 5 de Julio de 2018
- [4] *Netcrunch* tool <http://www.centria.es/netcrunch/> 5 de Julio de 2018
- [5] *Yersinia* tool http://www.blackhat.com/presentations/bh-europe-05/BH_EU_05-Berrueta_Andres/BH_EU_05_Berrueta_Andres.pdf 5 de Julio de 2018
- [6] Herramienta ROCIO <https://www.ccn-cert.cni.es/herramientas-ciberseguridad/rocio.html> 5 de Julio de 2018
- [7] Información general protocolo ARP http://cesar.blog.unq.edu.ar/modules/docmanager/get_file.php?curent_file=215&curent_dir=17 9 de Junio de 2018
- [8] RFC ARP <https://tools.ietf.org/html/rfc826> 9 de Junio de 2018
- [9] Ataque *ARP Spoofing* <https://www.veracode.com/security/arp-spoofing> 9 de Junio de 2018
- [10] Ataque *man in the middle* ARP <http://techgenix.com/Understanding-Man-in-the-Middle-Attacks-ARP-Part1/> 9 de Junio de 2018
- [11] Defensas contra *ARP Spoofing* https://en.wikipedia.org/wiki/ARP_spoofing#Defenses 9 de Junio de 2018



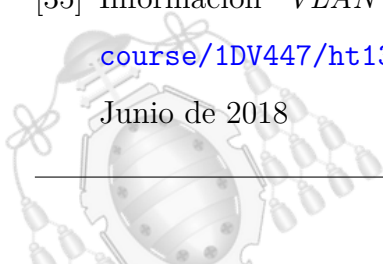


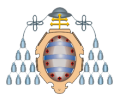
- [12] Defensas contra *ARP Spoofing* <http://www.jaringankita.com/blog/defense-arp-spoofing> 9 de Junio de 2018
- [13] Información general protocolo STP https://www.cisco.com/c/es_mx/support/docs/lan-switching/spanning-tree-protocol/5234-5.html 10 de Junio de 2018
- [14] RFC STP <https://tools.ietf.org/html/rfc7727> 10 de Junio de 2018
- [15] Ataques STP <https://howdoesinternetwork.com/2012/stp-attack> 10 de Junio de 2018
- [16] Ataques STP <https://www.s21sec.com/es/blog/2009/06/ataques-sobre-el-nivel-2-del-modelo-osi-iii-spanning-tree-protocol/> 10 de Junio de 2018
- [17] Defensas contra *BPDU Attack* <http://www.ciscozine.com/how-to-protect-against-bpdu-attack/> 10 de Junio de 2018
- [18] Defensas contra *BPDU Attack* <https://supportforums.cisco.com/t5/routing-y-switching-documentos/proteccion-a-la-operacion-de-stp/ta-p/3122368> 10 de Junio de 2018
- [19] Defensas contra *BPDU Attack* <https://www.securityartwork.es/2015/05/25/defensas-frente-a-ataques-stp/> 10 de Junio de 2018
- [20] Información general protocolo CDP <https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/cdp/configuration/15-mt/cdp-15-mt-book.pdf> 11 de Junio de 2018
- [21] Ataques CDP <https://howdoesinternetwork.com/2011/cdp-attack> 11 de Junio de 2018
- [22] Defensas contra ataques CDP <http://www.blacklabssecurity.info/cdp-exploitation.html> 11 de Junio de 2018
- [23] Deshabilitar CDP <http://blog.capacityacademy.com/2014/08/04/ccna-security-como-desactivar-el-protocolo-cdp-en-cisco-router/> 11 de Junio de 2018





- [24] Información general protocolo DTP <http://www.ciscopress.com/articles/article.asp?p=2181837&seqNum=8> 11 de Junio de 2018
- [25] Información general protocolo DTP https://es.wikipedia.org/wiki/Dynamic_Trunking_Protocol 11 de Junio de 2018
- [26] Ataque DTP <http://www.omnisecu.com/ccna-security/what-is-switch-spoofing-attack-how-to-prevent-switch-spoofing-attack.php#> 11 de Junio de 2018
- [27] Defensas contra ataques DTP <http://www.blacklabssecurity.info/dtp-exploitation.html> 11 de Junio de 2018
- [28] Deshabilitar DTP <http://packetlife.net/blog/2008/sep/30/disabling-dynamic-trunking-protocol-dtp/> 11 de Junio de 2018
- [29] Información general protocolo VTP <http://www.ciscopress.com/articles/article.asp?p=2348266&seqNum=2> 12 de Junio de 2018
- [30] Información general protocolo VTP <https://ticrt.wordpress.com/ccna/7-implementacion-vlans-troncales-y-vtp/> 12 de Junio de 2018
- [31] Ataque VTP y mitigación <http://slideplayer.com/slide/4251638/> (Transparencia 50) 13 de Junio de 2018
- [32] Ataque VTP y mitigación <https://es.slideshare.net/rishabhd/introduction-to-layer-2-attacks-mitigation> (Transparencia 14) 13 de Junio de 2018
- [33] Información sobre VTP version 3 https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-6500-series-switches/solution_guide_c78_508010.html 13 de Junio de 2018
- [34] Información *VLAN Hopping Attack* <https://howdoesinternetwork.com/2012/vlan-hopping-attack> 14 de Junio de 2018
- [35] Información *VLAN Hopping Attack* <http://orion.lnu.se/pub/education/course/1DV447/ht13/essays/Sebastian%20Lundberg%20Essay.pdf> 14 de Junio de 2018

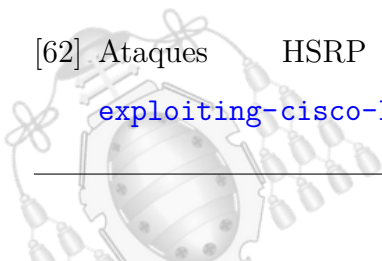




- [36] Defensas contra *VLAN Hopping Attack* https://en.wikipedia.org/wiki/VLAN_hopping 14 de Junio de 2018
- [37] Defensas contra *Double Tagging Attack* http://www.ronnybull.com/assets/docs/bullr1_nexus.pdf (Transparencia 36) 14 de Junio de 2018
- [38] Información protocolo RIP https://es.wikipedia.org/wiki/Routing_Information_Protocol 15 de Junio de 2018
- [39] RFC RIP versión 1 <https://tools.ietf.org/html/rfc1058> 15 de Junio de 2018
- [40] RFC RIP versión 2 <https://tools.ietf.org/html/rfc2453> 15 de Junio de 2018
- [41] Ataque *Reflection DDoS* y mitigación <https://blogs.akamai.com/2015/07/ripv1-reflection-ddos-making-a-comeback.html> 16 de Junio de 2018
- [42] Ataque RIP *Poisoning* y mitigación <https://www.giac.org/paper/gcih/239/security-ip-routing-protocols/102313> (Página 13) 16 de Junio de 2018
- [43] Información protocolo OSPF <https://www.mikroways.net/2009/07/20/introduccion-a-ospf/> 17 de Junio de 2018
- [44] RFC OSPF <https://tools.ietf.org/html/rfc2328> 17 de Junio de 2018
- [45] Ataques OSPF <http://www.hackplayers.com/2011/05/tipos-de-ataques-contra-ospf.html> 18 de Junio de 2018
- [46] Ataques OSPF <https://webcourse.cs.technion.ac.il/236349/Spring2013/ho/WCFiles/2009-2-ospf-report.pdf> 18 de Junio de 2018
- [47] *Persistent OSPF Attacks* y mitigación <http://theory.stanford.edu/~dabo/papers/ospf.pdf> 18 de Junio de 2018
- [48] Defensas contra ataques OSPF https://www.sanog.org/resources/sanog28/SANOG28-Tutorial_OSPF-Security-Attacks-and-Defences-Manjul.pdf 18 de Junio de 2018
- [49] Interfaces pasivas OSPF https://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Security/Baseline_Security/securebasebook/sec_chap3.html 18 de Junio de 2018



- [50] Información protocolo DHCP <https://www.1and1.es/digitalguide/servidores/configuracion/que-es-el-dhcp-y-como-funciona/> 19 de Junio de 2018
- [51] Información protocolo DHCP https://es.wikipedia.org/wiki/Protocolo_de_configuracion_dinamica_de_host 19 de Junio de 2018
- [52] RFC DHCP <https://tools.ietf.org/html/rfc2131> 19 de Junio de 2018
- [53] Ataques DHCP y mitigación https://smr.iesharia.org/wiki/doku.php/sgf:ut1:ataques_dhcp 20 de Junio de 2018
- [54] Ataque *DHCP Starvation* y mitigación <http://www.blacklabssecurity.info/dhcp-starvation.html> 20 de Junio de 2018
- [55] Ataque *DHCP Spoofing* <https://learningnetwork.cisco.com/docs/DOC-24355> 20 de Junio de 2018
- [56] Ataque *DHCP Ack Injector* <http://www.elladodelmal.com/2011/10/ataque-man-in-middle-con-dhcp-ack.html> 20 de Junio de 2018
- [57] *DHCP Snooping* <https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst6500/ios/12-2SX/configuration/guide/book/snoodhcp.pdf> 20 de Junio de 2018
- [58] Información protocolo HSRP https://www.cisco.com/c/es_mx/support/docs/ip/hot-standby-router-protocol-hsrp/9234-hsrpguidetoc.pdf 21 de Junio de 2018
- [59] Información protocolo HSRP <https://es.wikipedia.org/wiki/HSRP> 21 de Junio de 2018
- [60] RFC HSRP <https://tools.ietf.org/html/rfc2281> 21 de Junio de 2018
- [61] Ataques HSRP y mitigación <https://isc.sans.edu/forums/diary/Network+Reliability+Part+2+HSRP+Attacks+and+Defenses/10120/> 21 de Junio de 2018
- [62] Ataques HSRP y mitigación <https://portunreachable.com/exploiting-cisco-hsrp-63bd45f9af58> 21 de Junio de 2018





- [63] Ataques HSRP y mitigación <http://securityshiba.io/Network-Attack-HSRP-More-Like-Easy-Peasy/> 21 de Junio de 2018
- [64] Número de puertos UDP y TCP https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers 25 de Junio de 2018
- [65] Número de protocolo IP <https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml> 25 de Junio de 2018
- [66] Direcciones multicast https://en.wikipedia.org/wiki/Multicast_address 25 de Junio de 2018
- [67] Documentación Scapy <https://scapy.readthedocs.io/en/latest/> 1 de Julio de 2018
- [68] Documentación MIB Browser <http://www.ireasoning.com/browser/help.shtml> 1 de Julio de 2018
- [69] Información OID https://es.wikipedia.org/wiki/Identificador_de_objeto 2 de Julio de 2018
- [70] Información octeto [https://en.wikipedia.org/wiki/Octet_\(computing\)](https://en.wikipedia.org/wiki/Octet_(computing)) 2 de Julio de 2018

