Attack surface of VMWare/Virtualbox/ESXi virtual switches

## Virtual Switch

Uno switch virtuale (o vSwitch) è un programma software che abilita una macchina virtuale a comunicare con un'altra con gli stessi protocolli utilizzati negli switch fisici, senza la necessità di hardware di rete aggiuntivo. In generale, gli switch virtuali possono essere interfacciati alla rete fisica semplicemente associandoli a una o più interfacce fisiche disponibili negli host.

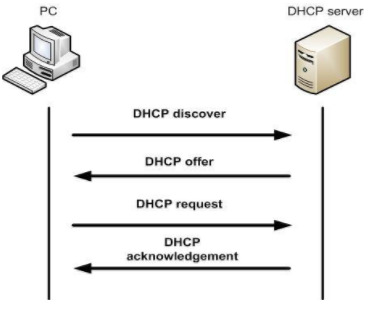
## vSwitch vs Physical Switch

Le principali differenze sono:

* Tipologia: vSwitch è un componente software (implementata su una macchina host) mentre physical switch è un componente hardware (formata da porte, circuiti e cavi);
* Connettività: gli switch fisici collegano dispositivi fisici in una rete utilizzando cavi fisici mentre i vSwitch collegano i dispositivi su interfacce di rete virtuali e connessioni di rete virtuali;
* Network Control: a differenza degli switch fisici, non necessitano di alcuna fase di apprendimento degli indirizzi MAC per la compilazione della MAC-Address table, perchè conoscono in modo autoritativo quali sono i dispositivi collegati su ogni porta.

## DHCP (stesso delle lezioni)

Quando entriamo in una rete c’è una configurazione automatica che ci consente di ottenere dei parametri per entrare nella rete tramite il DHCP. I parametri per entrare nella rete sono: un valido IP address, un valido DNS address di un DNS resolver, ed un valid gateway address e dobbiamo sapere in quale collision domain siamo (ovvero sapere chi sono i nostri vicini e chi sono fuori dal nostro collision domain).



Generalmente un device accetta la prima offerta che gli arriva da un DHCP server ed il DHCP non implementa nessuna macchina di sicurezza, questo significa che qualsiasi stazione che condivide il collision domain con la stazione attaccante, può ricevere le fake offerte di DHCP. Se l’attacco va a buon fine, l’attaccante può scegliere il DNS, l’IP e il gateway della vittima.

Generalmente la DHCP offer è unicast, am può essere anche broadcast.

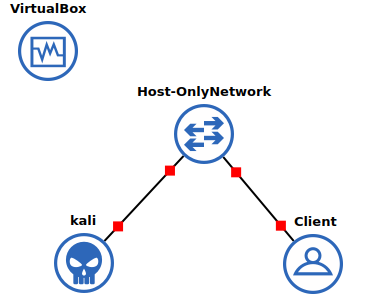
Durante il lease time, il client ha la possibilità di accettare l'offerta e utilizzare l'indirizzo IP assegnato. Se il client accetta l'offerta, il server DHCP prende nota dell'assegnazione e il client inizia a utilizzare l'indirizzo IP assegnato per la durata del lease time. Al termine del lease time, il client deve rinnovare la concessione dell'indirizzo IP attraverso un processo noto come DHCP Renewal o richiedere un nuovo indirizzo IP tramite una nuova richiesta DHCP.

## DHCP starvation e spoofing attacks

Nella DHCP starvation, il client attaccante invia così tanti messaggi DHCP Discover con falsi indirizzi MAC che il pool DHCP si riempie e il server non è in grado di servire client validi. Dopo aver eseguito l'attacco di starvation, l'attaccante può impostare un server DHCP non autorizzato e iniziare a servire la macchina vittima con indirizzi IP falsi. In questo modo l'attaccante può eseguire un attacco man in middle che acquisisce la richiesta del client inoltrando al server e riceve la risposta dal server e la invia al client.

## Configuration

Per performare questo attacco abbiamo utilizzato l’ambiente virtuale VirtualBox, versione 6.1.38 hostata su una macchina con Ubuntu 22.04.2 LTS. Per simulare il vSwitch abbiamo utilizzato l’adattatore Host Only Network. Per rappresentare il client abbiamo utilizzato “lubuntu-22.04.2-desktop-amd64” (versione light di Ubuntu) ma non è così rilevante quanto l’attaccante dove abbiamo montato l’OS “kali-linux-2023.2-virtualbox-amd64”.



## Perché Host Only Network

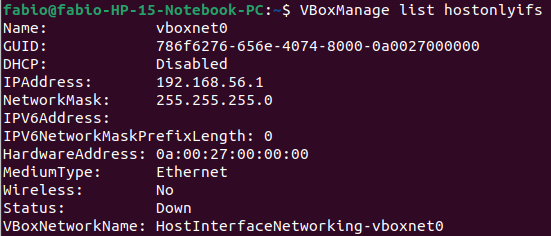
L'Host Network Manager di VirtualBox può essere considerato uno switch virtuale perché consente di creare e gestire reti virtuali all'interno dell'ambiente VirtualBox. Nel contesto di VirtualBox, Host Network Manager funge da interfaccia di controllo per la creazione e la configurazione di reti virtuali. Fornisce funzionalità simili a quelle di uno switch fisico, come la capacità di definire parametri di rete, assegnare indirizzi IP, configurare servizi DHCP e stabilire regole per l'instradamento del traffico. Creando reti virtuali utilizzando Host Network Manager, è possibile connettere più macchine virtuali alla stessa rete virtuale, consentendo loro di comunicare tra loro e con la macchina host. Questa funzionalità di switch virtuale consente configurazioni di rete flessibili e facilita test, sviluppo e altri scenari in cui sono necessari ambienti di rete virtualizzati. È importante notare che mentre Host Network Manager fornisce funzionalità simili a switch virtuali all'interno dell'ambiente VirtualBox, opera a livello di host e non si estende oltre il livello di virtualizzazione di VirtualBox per interagire con switch di rete fisici.

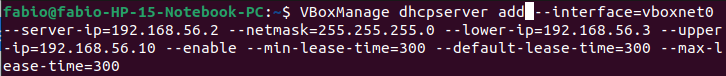
## Creazione e configurazione dell’Host Only Network

Una volta creata una scheda virtuale in Host Only Interface:



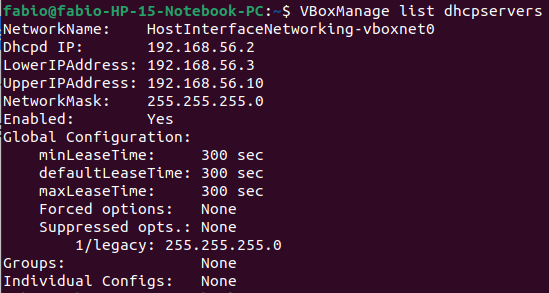
controlliamo che sia stata effettivamente creata:



Ora passiamo alla configurazione del DHCP server (non impostare il time lease nella demo):  


Tra i parametri che possiamo impostare c’è quello del time lease dove per simulare il nostro attacco con una tempistica bassa. Ogni device richiede di rinnovare il proprio ip ogni 300 secondi. Non è molto rilevante il tempo dei 300 secondi, più che altro ci viene utile il concetto che quando un utente chiude il computer, il suo IP verrà assegnato ad un altro device, in quanto non lo rinnoverà.

Verifichiamo che il nostro server DHCP è configurato correttamente:



Fatto ciò colleghiamo ogni device all’Host Only Network dalle impostazioni.

## Run dell’attacco

Da kali eseguiamo yersinia per performare l’attacco DHCP starvation: sudo yersinia -G.

Eseguiamo l’attacco finché non riempiamo le richieste dello vSwitch (pochi secondi) e poi non chiudiamo yersinia.

Proviamo a connettere una macchina per ottenere un nuovo IP e vediamo che essa non riesce. Per forzare un nuovo calcolatore a chiedere un nuovo IP: sudo dhcp -r <scheda rete> (per eliminare il suo IP) e poi sudo dhcp -v <scheda rete> (per ottenere un nuovo IP).

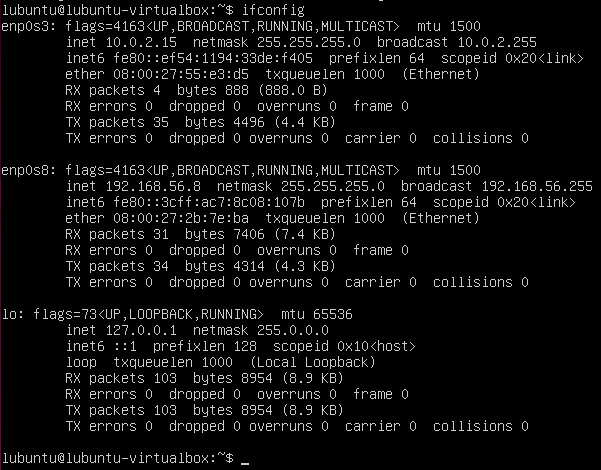
Tramite wireshark ma anche tramite Ettercap possiamo vedere le DHCP request. Una volta chiuso il nostro attacco yersinia possiamo vedere il MAC del device che vuole un nuovo IP. Quando chiudiamo l’attacco yersiania abbiamo qualche minuto.

In questo tempo, quando fermiamo il nostro attacco DHCP starvation, restiamo in ascolto per vedere se qualcuno sta avanzando DHCP request. Se le sta avanzando facciamo un DHCP server rogue che gli fornirà un indirizzo IP, DNS e gateway fake.

Ettercap (oppure tramite yersinia) viene settato in questo modo:

| Naturalmente i parametri possono essere cambiati, ciò che è importante è la maschera e il DNS server Ip che dev’essere l’IP dell’attaccante nel collision domain. |  |
| --- | --- |

Ora avviamo un client normale. Runniamo il comando ifconfig per vedere un po’ le sue schede di rete.



Come possiamo notare, la scheda “enp0s3” è quella collegata al NAT che consente di andare su internet, mentre “enp0s8” è quella collegata al nostro vSwitch. Attualmente “enp0s8” ha un indirizzo IP valido, assegnato dal reale DHCP server, ovvero la nostra vSwitch. Vediamo più nel dettaglio la configurazione di questa scheda:



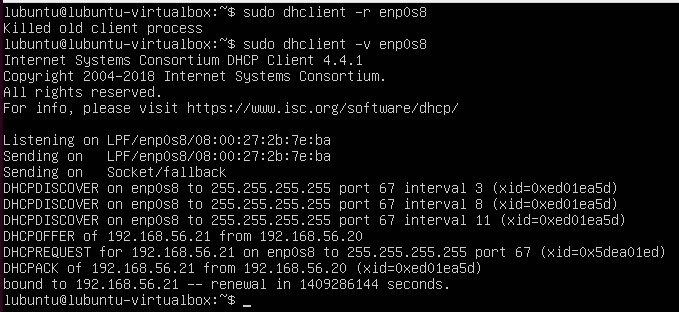
Come possiamo notare, la validità dell’IP assegnato dal nostro DHCP scadrà tra 51 secondi. Quando l’utente chiude il suo PC, noi avviamo il nostro attacco DHCP starvation, poiché gli scade il certificato all’utente e noi ce lo prendiamo con il nostro attacco DHCP. Quando lui poi riaccende il PC noi vediamo dei DHCP request ed allora avviamo il nostro DHCP rogue server per assegnargli un IP nostro.

Velocizziamo questa pratica eseguendo il comando sudo dhclient -r enp0s8 (che serve per eliminare l’IP assegnato a quella scheda di rete) seguito dal comando sudo dhclient -v enp0s8 (per richiedere un nuovo IP).

Quando eseguiamo l’attacco DHCP starvation, l’utente non riesce a trovare un IP valido:



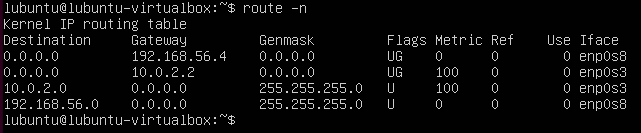
Ora avviamo il nostro DHCP rogue server:



Il device dell’utente invia una serie di DHCPdiscover senza successo le prime perché il legittimo DHCP server è sotto attacco. Poi avviando il nostro DHCP server rogue gli viene assegnato un IP.

Facciamo un test con traceroute per verificare il routing da quel dispositivo ad un altro host collegato al vSwitch. Quello che ci aspettiamo è che lui utilizza tranquillamente il vSwitch per comunicare con i suoi vicini, ovvero i devices del collision domain, invece per andare su internet i pacchetti vengano instradati su kali, perché il gateway del DNS inserito tramite il DHCP server rogue impone questa rotta e quindi siamo in una posizione man in the middle passiva attualmente. Da qui in avanti si può percorrere un attacco attivo.

Prima di vedere la traceroute bisogna configurare la nostra macchina kali. Dalla route table del client attaccato notiamo che ogni pacchetto con una destinazione diversa da quella del collision domain delle schede di rete, viene inviata o al 192.168.56.4 (il nostro fake DNS) oppure alla scheda di rete connessa ad internet (10.0.0.2). Tra le due viene preferita qulla con metric più bassa, ovvero la nostra.

Ora dobbiamo forwardare tutti i pacchetti sulla scheda eth0, ovvero quella collegata ad internet sulla macchina kali, ad eccezione dei pacchetti che hanno l’IP appartenente al nostro collision domain che devono rimanere sull’interfaccia eth1 altrimenti non utilizziamo lo vSwitch e non possiamo raggiungere i nostri vicini.

Innanzitutto dobbiamo abilitare l'inoltro del traffico nel kernel. Apriamo il file /etc/sysctl.conf utilizzando un editor di testo e assicurati che la riga seguente sia presente e non sia commentata: net.ipv4.ip\_forward=1. Salviamo con sudo sysctl -p.

Ora dobbiamo configurare la regola iptable come descritto sopra.

sudo iptables -A FORWARD -i eth1 -o eth0 -m iprange ! --src-range 192.168.56.0-192.168.56.255 -j ACCEPT

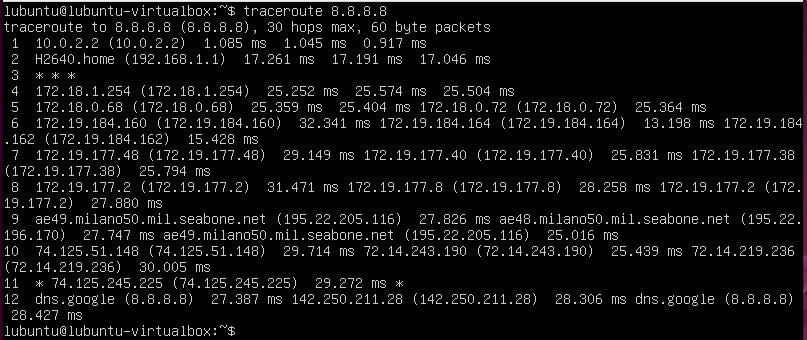
seguito dal MASUADE per consentire il traffico di ritorno:

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

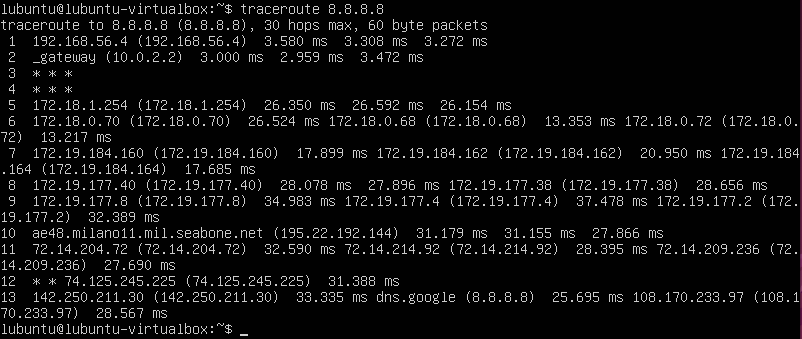
Salviamo con questo comando, rendendo la modifica persistente:

sudo iptables-save -f /etc/iptables/rules.v4

Ora finalmente possiamo vedere il confronto dei traceroutes prima e dopo l’attacco



(Prima: notare il primo nodo attraversato che è l’IP della scheda di rete montata sul client che consente di andare su internet)



(Dopo: notare il primo nodo attraversato che è l’IP di kali)

## Mitigazione

Ci sono diversi modi per mitigare questo tipo di attacco:

* Impostare IP statici dalle proprie macchine essendo a conoscenza del range di IP assegnabili dal vSwitch;
* Visualizzazione della cattura dhcp per vedere se siamo sotto questo attacco;
* Controllo del nostro indirizzo IP;
* Impostazione di un tempo altissimo per la scadenza degli IP, questo comporta che soltanto appena attacchiamo il device potremmo essere sotto attacco.

## Come lo attuo

Runna kali, fai l’attacco DHCP starvation; runni lubuntu; performi l’attacco DHCP rouge server; Assegni l’IP a lubuntu; spegni kali e lo riapri e facciamo le iptables e poi va.

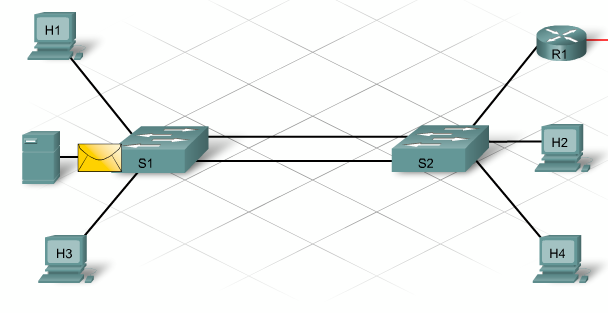
## Bibliografia

[**VBoxManage manual**](https://www.virtualbox.org/manual/ch08.html#vboxmanage-hostonlynet)

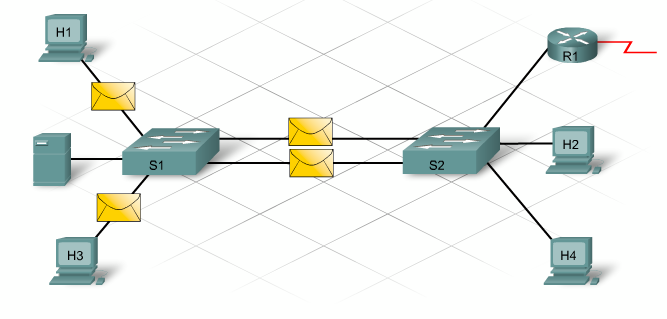
## Spanning Tree Protocol (SPT)

In molte situazioni è utile collegare tra loro più switch, per garantire l’espansione della rete; è utile avere anche dei collegamenti ridondanti, poiché se un cavo che collega tra loro due switch si rompe oppure le porte dove è connesso hanno dei problemi, la comunicazione tra host sui differenti switch viene a mancare.

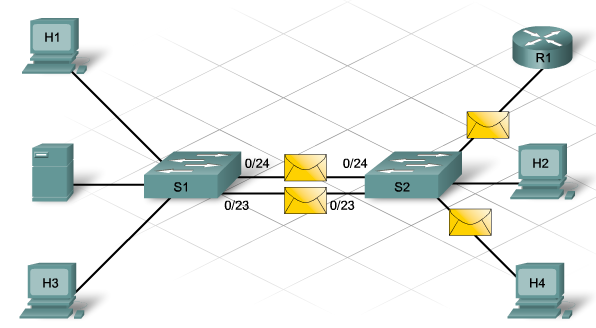
l server (sul lato sinistro) invia un messaggio in broadcast:



Lo switch S1 inoltrerà il frame su tutte le porte, tranne quella di arrivo:



Lo switch S2 riceve 2 frame e li elabora. Quello che arriva sull’interfaccia 0/24, essendo di tipo broadcast, verrà replicato su tutte le porte tranne la 0/24 (quella di arrivo). Lo stesso avviene per il frame arrivato sull’interfaccia 0/23, replicato su tutte le porte tranne la 0/23 (quella di arrivo). A questo punto 2 frame identici ripartono verso le interfacce dello switch S1:

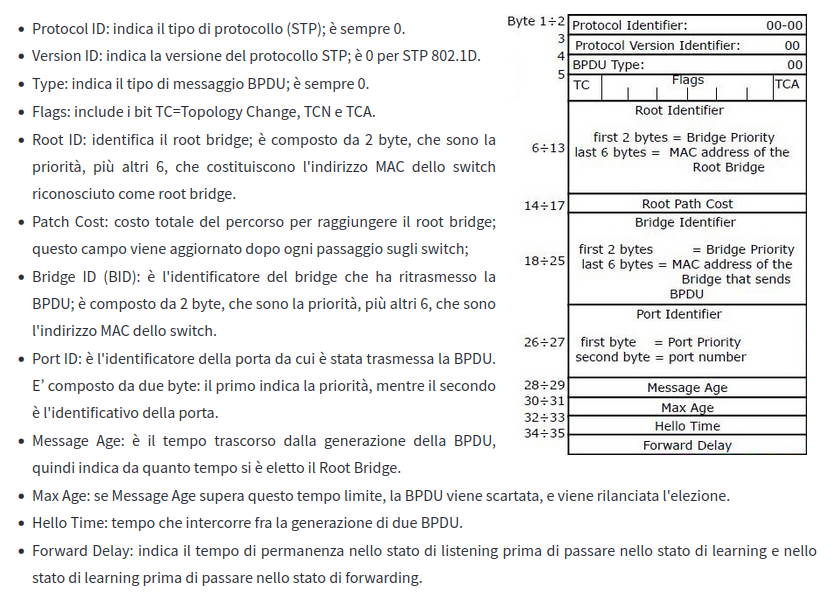


gli switch con STP abilitato controllano la rete per verificare l'esistenza di percorsi ciclici (loop): se vengono rilevati, alcune porte di collegamento vengono bloccate, spezzando in questo modo l’anello generato dai percorsi ridondati. Le altre porte vengono lasciate attive per la trasmissione delle trame.

STP definisce un albero che attraversa tutti gli switch della rete. Per evitare i loop nello switching, lo Spanning Tree Protocol:

* forza alcune interfacce ad uno stato di blocco o standby
* lascia le altre interfacce nello stato di inoltro
* configura la rete attivando il percorso lasciato in standby, nel caso in cui il percorso principale non fosse più disponibile

STP prevede la presenza di uno switch designato come root, detto **root bridge**, ovvero radice dello spanning tree. Il root bridge comunica con altri switch utilizzando particolari frame dette **BPDU, Bridge Protocol Data Units**, inviate in multicast ogni 2 secondi a tutti gli altri switch.



Quando STP è abilitato su un bridge di rete, ogni porta è impostata su uno dei cinque stati per controllare il frame forwarding:

1. **Disabilitato.** La porta non partecipa al frame forwarding o alle operazioni STP.
2. **Blocco.** La porta non partecipa al frame forwarding e scarta i frame ricevuti dal segmento di rete collegato. Tuttavia, la porta continua ad ascoltare ed elaborare BPDU.
3. **Ascoltando.** Dallo stato di blocco, la porta passa allo stato di ascolto. La porta elimina i frame dal segmento di rete collegato o inoltrati da un'altra porta. Tuttavia, riceve i BPDU e li reindirizza al modulo switch per l'elaborazione.
4. **Apprendimento.** La porta passa dallo stato di ascolto allo stato di apprendimento. Ascolta ed elabora i BPDU ma scarta i frame dal segmento di rete collegato o inoltrati da un'altra porta. Inizia anche ad aggiornare la tabella degli indirizzi con le informazioni apprese. Inoltre, elabora i frame utente ma non li inoltra.
5. **Inoltro.** La porta passa dallo stato di apprendimento allo stato di inoltro e inizia a inoltrare i frame attraverso i segmenti di rete. Ciò include i frame dal segmento di rete collegato e quelli inoltrati da un'altra porta. La porta continua inoltre a ricevere ed elaborare BPDU e la tabella degli indirizzi continua ad essere aggiornata.

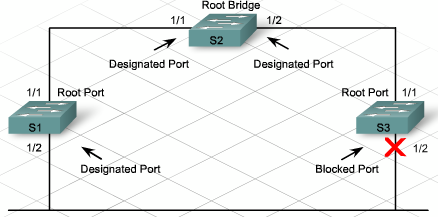
### **I ruoli delle porte**

Per la definizione dei ruoli delle porte sono previsti 3 passaggi fondamentali:

1. elezione del **root bridge**
2. elezione delle **root port**
3. elezione delle **designated port**

**1.** Il root bridge viene eletto in base al valore presente nel campo **Bridge Identifier (BID)** della BPDU: **“vince” lo switch con il valore minore**. **All’inizio, ogni switch ritiene di essere il root bridge**: genera quindi una BPDU, impostando nel campo Root ID il proprio identificatore. In questa fase si ha che Root ID = Bridge ID. Il frame viene quindi inoltrato a tutte le porte. Quando un altro switch riceve una BPDU, confronta il Root ID ricevuto con il proprio Bridge ID; se il proprio Bridge ID è inferiore, lo switch continuerà a ritenersi root, proseguendo nell’invio delle BPDU e propagando il proprio identificativo come root; se invece il valore ricevuto è minore, lo switch riconoscerà il nuovo root e invierà le BPDU inserendo il Root ID corretto. In base a questo meccanismo, dopo un po' di tempo tutti gli switch sapranno chi è il root bridge.

**2.** Tutti gli apparati che non sono diventati root bridge, avranno una porta configurata come **root port**, ossia una porta che connette lo switch al root bridge tramite il link a minor costo. Ogni switch calcola il percorso più breve verso il root bridge stesso. Le BPDU inviate dal root avranno un **costo pari a 0**. Uno switch che riceve una BPDU dal root vi aggiunge il costo della porta tramite cui ha ricevuto la BPDU stessa (vedere tabella sul valore del Patch Cost nei paragrafi precedenti). A questo punto verificherà su quale delle sue interfacce riceve la BPDU con costo minore: l'interfaccia con questa caratteristica diventerà **root port**. Di seguito tutti gli switch, tramite il meccanismo appena descritto, identificheranno la propria root port.



*Nell'esempio proposto, dopo l'elezione del root bridge vinta dallo switch S2, prendendo in considerazione lo switch S1 vediamo che esso può ricevere BPDU su tutte e due le interfacce 1/1 e 1/2. Le BPDU ricevute sulla 1/1 avranno un costo pari a 0, a cui va sommato il costo della porta stessa; quelle ricevute sulla 1/2 avranno un costo pari a quello della porta 1/1 di S3, a cui va sommato il costo della stessa 1/2 di S1. La root port di S1 sarà pertanto la 1/1.*

**3.** Secondo il protocollo STP, ogni segmento di rete può avere una sola porta designata, tecnicamente detta **designated port.** Questa porta permette l'inoltro del traffico verso il root bridge. La designated port è quella con root path cost inferiore e, in caso di parità, quella appartenente allo switch con BID inferiore. Le altre porte che si affacciano sullo stesso segmento di rete divengono delle **blocked ports**, in grado di ricevere BPDU ma non di inoltrare traffico.

## Come sfruttare tale attacco?

[Carenza ESXI 5.0](https://kb.vmware.com/s/article/2047822)