Ryan Stearns, Jack Holland
Lab 1
Due: 9/15/2014

VLAN Hopping Using Double Encapsulation

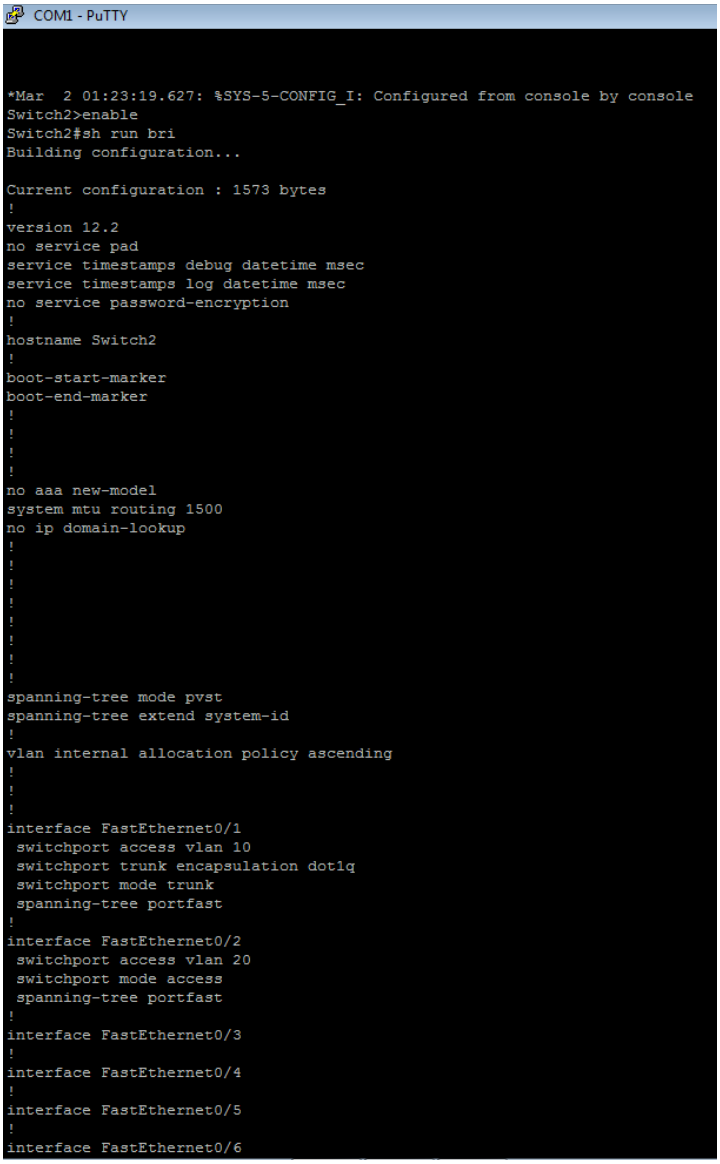
For our lab we configured two Cisco Catalyst 3560 Series Switches using the application Putty. We connected a Windows 7 Computer to each of the switches via a console cable for the configuration. Accordingly, we created a Virtual Local Area Network on each switch. For the first switch, which we named Switch1, we created VLAN 10 and configured port Fa0/1 as a trunk port and Fa0/2 as an access port for our Backtrack 5 Linux machine. For the second switch, labeled Switch2, we created VLAN 20 and configured the ports in the same way as was performed on the first switch, except for the access port was for a Windows machine this time. Below is the result of the **sh int trunk** command performed on Switch 2. As you can see, ports Fa0/1 was assigned as the trunk port and Fa0/2 was added to VLAN0020.

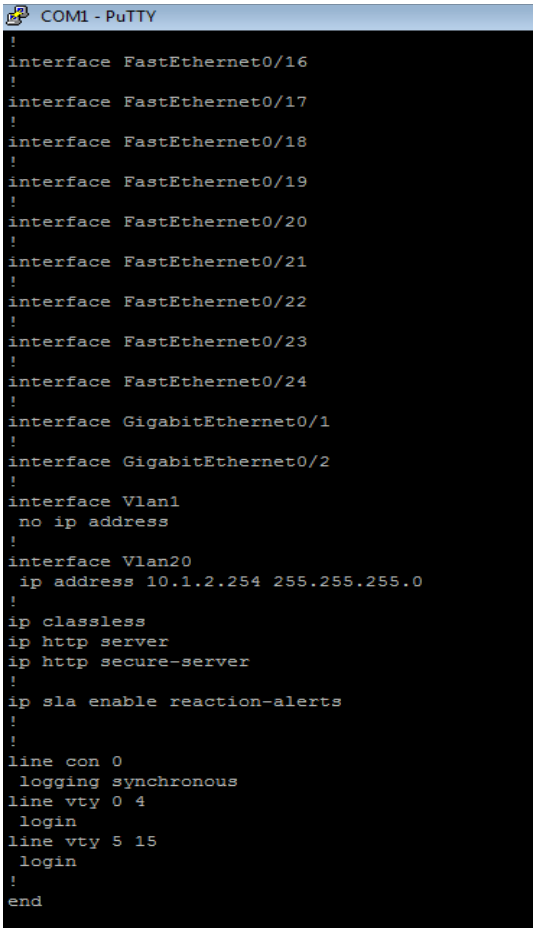
```
Switch2#
Switch2#sh int trunk

Port        Mode             Encapsulation  Status        Native vlan
Fa0/1       on               802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       1-4094

Port        Vlans allowed and active in management domain
Fa0/1       1,10,20

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/1       1,10,20
Switch2#show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/3, Fa0/4, Fa0/5, Fa0/6
                                                Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                                Fa0/11, Fa0/12, Fa0/13, Fa0/14
                                                Fa0/15, Fa0/16, Fa0/17, Fa0/18
                                                Fa0/19, Fa0/20, Fa0/21, Fa0/22
                                                Fa0/23, Fa0/24, Gi0/1, Gi0/2
10   VLAN0010                         active
20   VLAN0020                         active    Fa0/2
1002 fddi-default                     act/unsup
1003 token-ring-default               act/unsup
1004 fddinet-default                  act/unsup
1005 trnet-default                    act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
20   enet  100020     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
 --More--
```

Once the VLANs were created, we had to assign IP addresses to each VLAN. For VLAN 10 residing on Switch1, we assigned an IP address of 10.1.1.254 with a subnet mask of 255.255.255.0. For VLAN 20 residing on Switch2, we assigned an IP address of 10.1.2.254 with a subnet mask of 255.255.255.0.

From the VLAN connected machines we were able to set Static IP Addresses for the computers connected to VLAN 10 and VLAN 20. For the Backtrack Linux Machine connected to port Fa0/2 of Switch1, we set the IP Address to be assigned the IP 10.1.1.10 from the Backtrack machine using the GUI. For the Windows machine connected to port Fa0/2 of Switch2, we set the IP Address of 10.1.2.10 using the windows network configuration tools.

To ensure forwarding of the frame, we turned off Cisco Inter-Switch Link, the default method of encapsulation, and enabled IEEE 802.1Q Encapsulation for both trunk ports. We theorized at this point that if a frame was crafted with double encapsulation, that we should be able to successfully attack the Windows machine residing on VLAN 20 of Switch2, from the Backtrack machine which resides on VLAN 10 of Switch1. To test our small network, we pinged the switch opposite the one our host computer was connected to. The pinging was successful so we knew there were no connectivity issues between our two switches. Below is the running configuration of Switch 2. Switch 1 is basically identical.





We then crafted a double encapsulated frame in a packet manipulation program called Scapy on the Backtrack machine. We wrote a few different versions of a Python Script which invoked the program Scapy in an attempt to attack the Windows machine on VLAN 20 of Switch2. The script varies from directly setting the destination IP address and MAC address of the Windows machine, to sending the frame to a broadcast address. Below in bold is the the python script we wrote.

#! /usr/bin/env/ python

from scapy.all import*

a=Ether(dst='2C:44:FD:33:4C:E7',
src='00:1C:23:45:F0:AL')/Dot1q(vlan=10)/Dot1Q(vlan=20)/IP(dst='10.1.2.10',
src='10.1.1.10')/ICMP()

send(a)

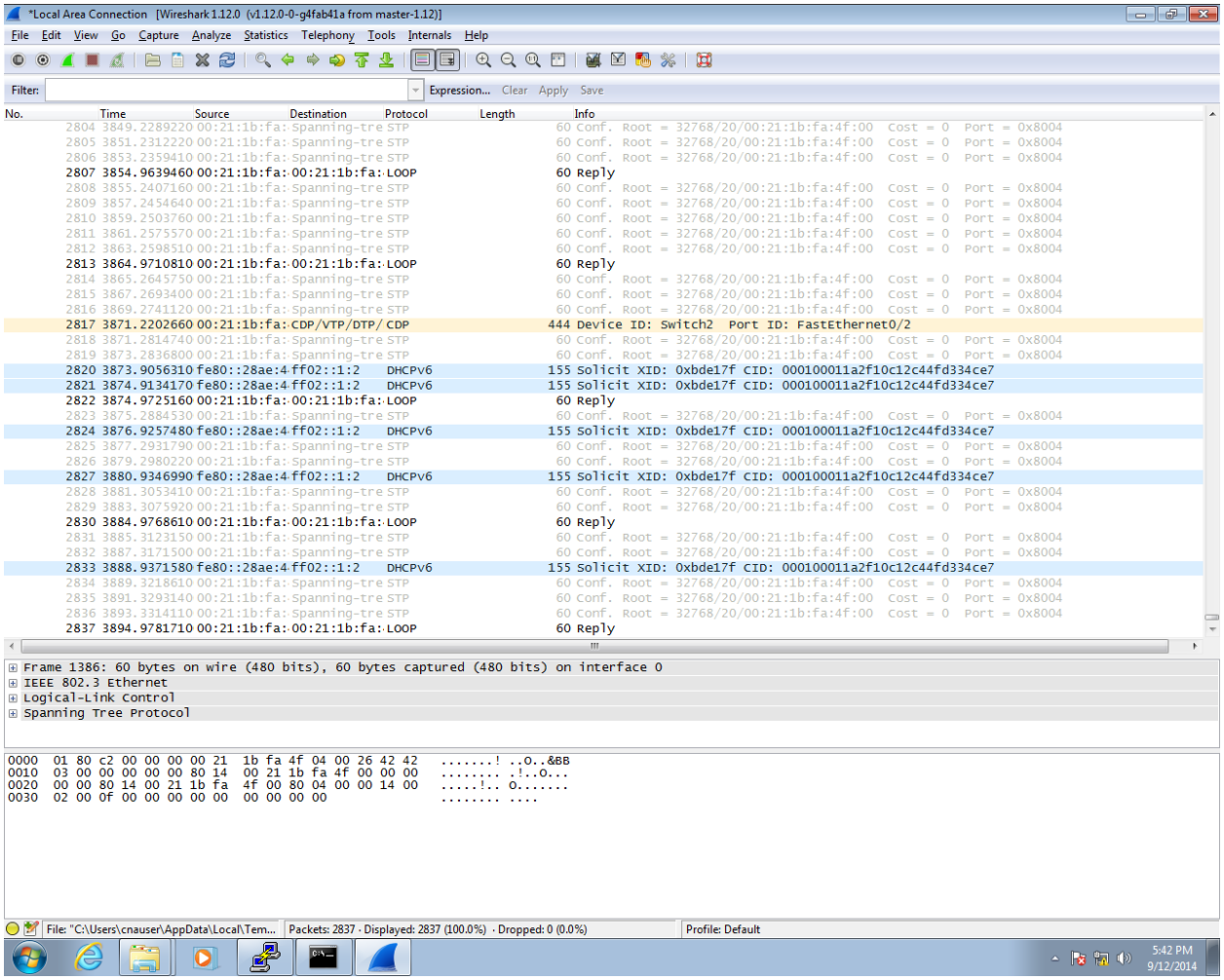
We were able to monitor the traffic on the Windows Machine from the network protocol analyzer program called Wireshark. Despite our best efforts, we were unable to capture the ICMP Packet on the Windows machine. A second Windows machine was hooked up to VLAN 10 alongside Backtrack, so we could monitor the packet being sent out.

After a ton of testing and rewriting code, we concluded that it was not possible to attack VLAN 20 from VLAN 10 using the double encapsulation VLAN hopping technique. Further research supported our conclusion, “Its possible to hop from VLAN 1 to other VLANs, but it’s not possible to hop from VLAN 2 or 3 to other VLANs. As VLAN 1 is the native VLAN (default configuration), only VLAN 1 is two times decapsulated” (Rouiller, 16) “In order to attack from one VLAN to another, attackers need to meet some specific conditions, but this is the set up by default. In order to avoid the possibility of VLAN hopping and double tagged 802.1q attacks, the administrator should dedicate VLAN other than VLAN 1 for trunking and assign specific VLAN.” (Rouiller, 24) Since we were dealing with VLAN 10 and VLAN 20, and not VLAN 1, this eliminated the vulnerability that is evident when using VLAN 1 as a trunk port.

Resources:

Rouiller, Steven A. *Virtual LAN Security: Weaknesses and Countermeasures.Http://www.sans.org*. SANS Institute, n.d. Web. 9 Sept. 2014.

Stretch, Jeremy. "Experimenting with VLAN Hopping." *PacketLife.net*. Packet Life, 22 Feb. 2010. Web. 9 Sept. 2014.



Above is the Wireshark program not picking up any of our ICMP traffic.