



Security+ Lab Series

Investigating ARP Poisoning

Copyright © 2018 Network Development Group, Inc.
www.netdevgroup.com

NETLAB Academy Edition, NETLAB Professional Edition, NETLAB+ Virtual Edition, and NETLAB+ are registered trademarks of Network Development Group, Inc.

Contents

Introduction	3
Objectives.....	3
Lab Topology	4
Lab Settings	5
1 Configure an Ubuntu Workstation as a MITM.....	6
2 Use Wireshark to View Traffic Moving Through the Ubuntu Workstation	9
3 Test the Current Network from Win16.....	11

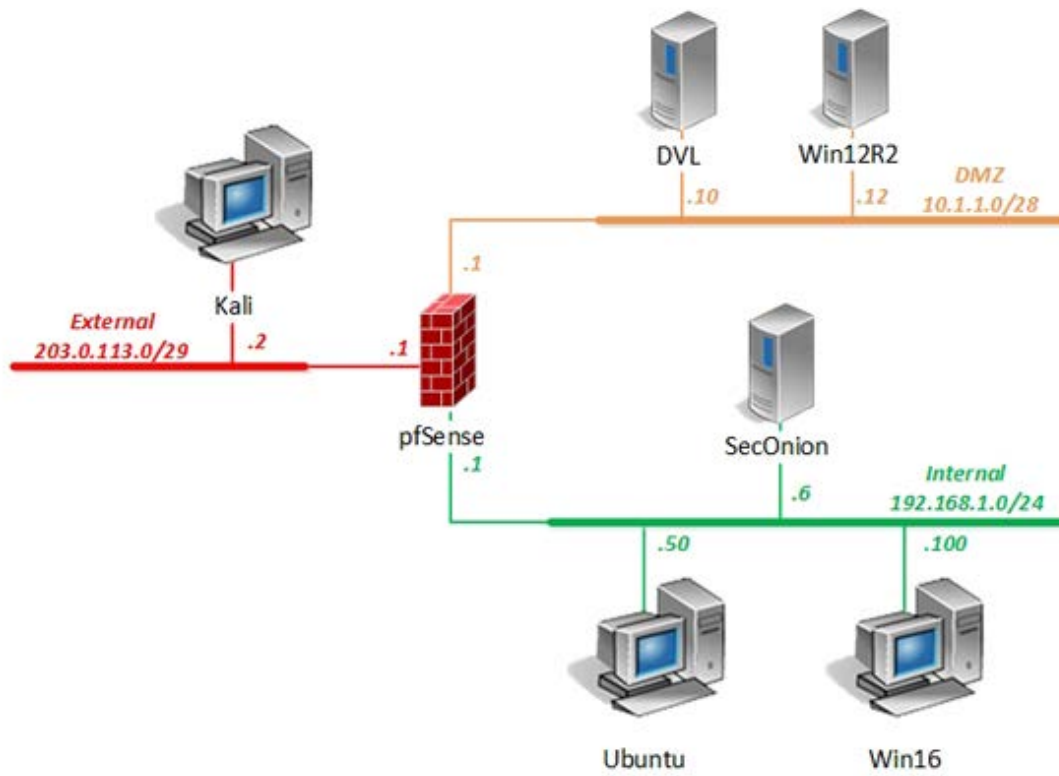
Introduction

In this lab, you will configure an *Ubuntu* workstation to spoof a *Windows* system as it attempts to communicate directly to the *pfSense* gateway. You will also configure the *Ubuntu* workstation to spoof the *pfSense* gateway that tries to communicate with the *Windows* system directly. You will then view the traffic being redirected across the *Ubuntu* workstation, which is now acting as a “*Man in the Middle*” (MITM).

Objectives

- Configure an *Ubuntu* workstation as a *MITM*
- Use *Wireshark* to view traffic moving through the *Ubuntu* workstation
- Test the current network from the *Win16* workstation

Lab Topology



Lab Settings

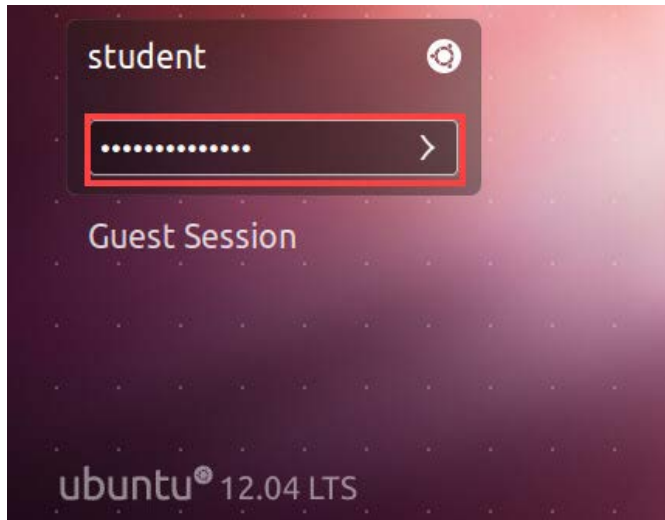
The information in the table below will be needed to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account	Password
DVL	10. 1. 1. 10 /28	root	toor
Kali	203. 0. 113. 2 /29	root	toor
pfSense	eth0: 192. 168. 1. 1 /24 eth1: 10. 1. 1. 1 /28 eth2: 203. 0. 113. 1 /29	admin	pfsense
Sec0nion	192. 168. 1. 6 /24	soadmin	mypassword
		root	mypassword
Ubuntu	192. 168. 1. 50 /24	student	securepassword
		root	securepassword
Win12R2	10. 1. 1. 12 /28	administrator	Train1ng\$
Win16	192. 168. 1. 100 /24	lab-user	Train1ng\$
		Administrator	Train1ng\$

1 Configure an Ubuntu Workstation as a MITM

In this task, you will configure the *Ubuntu* workstation to spoof the *MAC* address of the router acting as the default gateway.

1. Launch the **Ubuntu** virtual machine to access the graphical login screen.
2. Log in as **student** using the password **securepassword**.



3. Open a command terminal by clicking on the **terminal** icon located in the left taskbar.

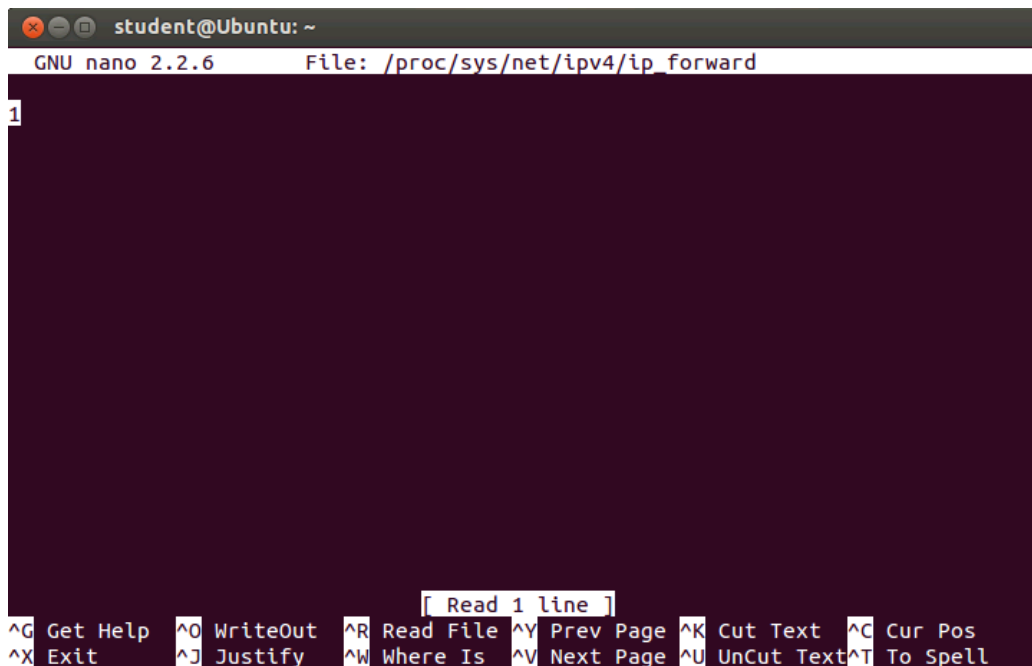


4. In the terminal, we will need to forward IP packets through the *Ubuntu* system. Enter the command below followed by pressing the **Enter** key to open the configuration file. If prompted for a password, enter **securepassword**.

```
student@Ubuntu: ~$ sudo nano /proc/sys/net/ipv4/ip_forward
```

```
student@Ubuntu:~$ sudo nano /proc/sys/net/ipv4/ip_forward
```

5. Change the value from **0** to **1** and then press **CTRL+X**. Notice a message appears asking whether you would like to save the file. Press the **Y** key for Yes. Then, when asked for the file name, leave the default file name and press the **Enter** key.



The screenshot shows a terminal window with the nano text editor open. The title bar indicates 'student@Ubuntu: ~' and 'GNU nano 2.2.6'. The file being edited is '/proc/sys/net/ipv4/ip_forward'. The content of the file is '1'. The bottom status bar shows various keyboard shortcuts: ^G Get Help, ^O WriteOut, ^R Read File, ^Y Prev Page, ^K Cut Text, ^C Cur Pos, ^X Exit, ^J Justify, ^W Where Is, ^V Next Page, ^U UnCut Text, ^T To Spell. A small box above the status bar says '[Read 1 line]'.

6. Enter the command below to deceive the gateway device by telling it that the *Ubuntu*'s IP address is **192.168.1.100**. If prompted for a password, enter **securepassword**.

```
student@Ubuntu: ~$ sudo arpspoof -i eth0 -t 192.168.1.1 192.168.1.100
```

```
student@Ubuntu:~$ sudo arpspoof -i eth0 -t 192.168.1.1 192.168.1.100
[sudo] password for student:
0:50:56:9c:59:78 0:50:56:9c:3f:57 0806 42: arp reply 192.168.1.100 is-at 0:50:56:9c:59:78
0:50:56:9c:59:78 0:50:56:9c:3f:57 0806 42: arp reply 192.168.1.100 is-at 0:50:56:9c:59:78
```

7. We are now going to tell the *Win16* system that we are the *192.168.1.1* gateway device. Open another terminal by right-clicking on the **Terminal** icon and selecting **New Terminal** and enter the command below. If prompted for a password, enter **securepassword**. You should now have two terminals opened with *ARP* spoofing.

```
[student@Ubuntu: ~]$ sudo arpspoof -i eth0 -t 192.168.1.100 192.168.1.1
```

```
student@Ubuntu:~$ sudo arpspoof -i eth0 -t 192.168.1.100 192.168.1.1
[sudo] password for student:
0:50:56:9c:59:78 0:50:56:82:56:8f 0806 42: arp reply 192.168.1.1 is-at 0:50:56:9c:59:78
```

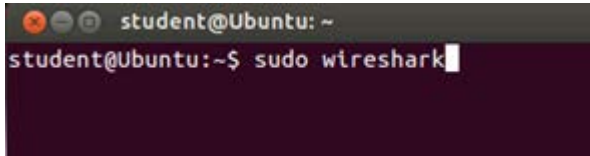
8. We have now logically placed the *Ubuntu* device between the *Win16* system and the *pfSense* gateway. Leave the *Ubuntu* screen opened to continue with the next task.

[illegible]

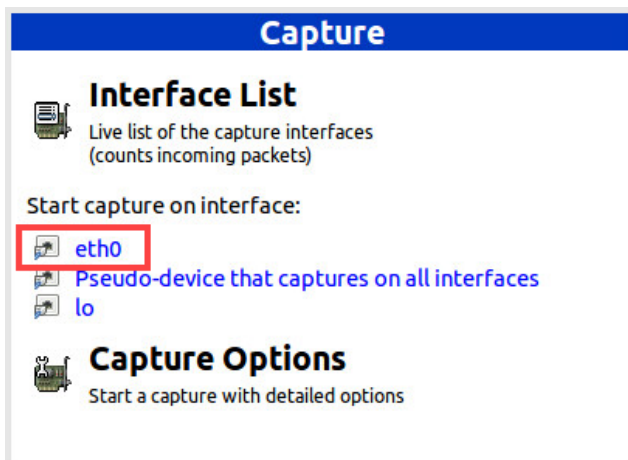
2 Use Wireshark to View Traffic Moving Through the Ubuntu Workstation

1. While on the *Ubuntu* system, open a third terminal and enter the command below to launch the **Wireshark** application. If prompted for a password, enter **securepassword**.

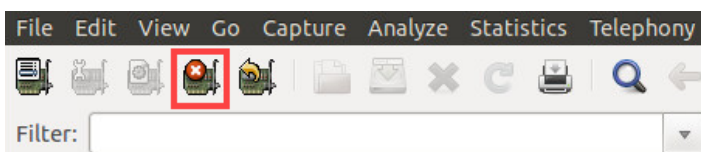
```
[student@Ubuntu: ~]$ sudo wireshark
```



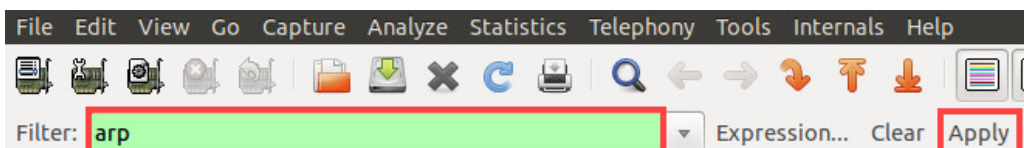
2. If prompted with a security warning, click **OK** to continue.
3. If an error appears regarding *init.lua*, click **OK** to continue.
4. Once *Wireshark* loads, click on **eth0** in the *Interface List* pane to start capturing on that interface.



5. After some packets have been captured and at least *30 seconds* has passed, stop the live capture by clicking on the **Stop the running live capture** button.



6. Filter the packets in the *Wireshark* output to only show *ARP* related packets by typing **arp** in the *Filter* text field, followed by clicking the **Apply** button.

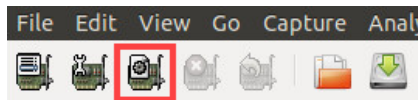


- Analyze the first couple of packets and notice that the `00:50:56:9c:59:78` MAC address has two IP addresses assigned to it.

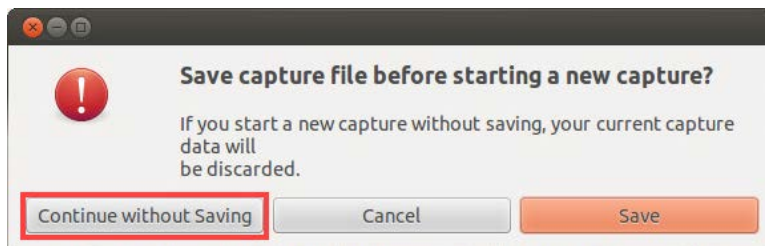
Filter: **arp** Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
5	1.386069	Vmware_9c:59:78	Vmware_82:56:8f	ARP	42	192.168.1.1 is at 00:50:56:9c:59:78
6	1.386143	Vmware_9c:59:78	Vmware_9c:3f:57	ARP	42	192.168.1.100 is at 00:50:56:9c:59:78
11	3.386442	Vmware_9c:59:78	Vmware_82:56:8f	ARP	42	192.168.1.1 is at 00:50:56:9c:59:78
12	3.386554	Vmware_9c:59:78	Vmware_9c:3f:57	ARP	42	192.168.1.100 is at 00:50:56:9c:59:78
23	5.386744	Vmware_9c:59:78	Vmware_82:56:8f	ARP	42	192.168.1.1 is at 00:50:56:9c:59:78
24	5.386822	Vmware_9c:59:78	Vmware_9c:3f:57	ARP	42	192.168.1.100 is at 00:50:56:9c:59:78

- Start another live capture by clicking on the **Start a new live capture** icon.

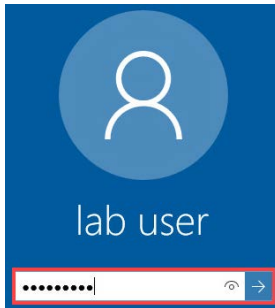


- When prompted, click **Continue without Saving**.



3 Test the Current Network from Win16

1. Launch the **Win16** virtual machine to access the graphical login screen.
2. While on the splash screen, focus on the *NETLAB+* tabs. Click the drop-down menu for the **Win16** tab and click on **Send CTRL+ALT+DEL**.
3. Log in as **lab-user** using the password **Training\$**.



4. Click on the **Windows Search** icon located in the taskbar and type **cmd** in the search field, followed by pressing the **Enter** key to launch the command prompt.



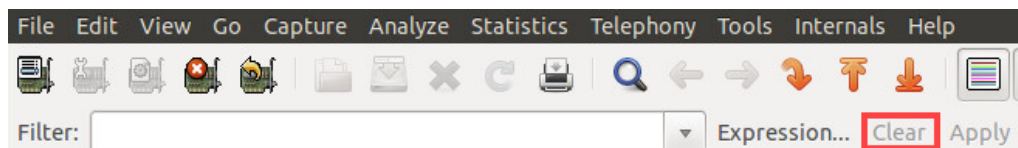
5. Create a persistent ping to the **192.168.1.1** IP address by entering the command below.

```
C:\Users\lab-user> ping -t 192.168.1.1
```

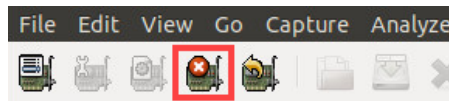
```
C:\Users\lab-user>ping -t 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
Reply from 192.168.1.1: bytes=32 time<1ms TTL=63
```

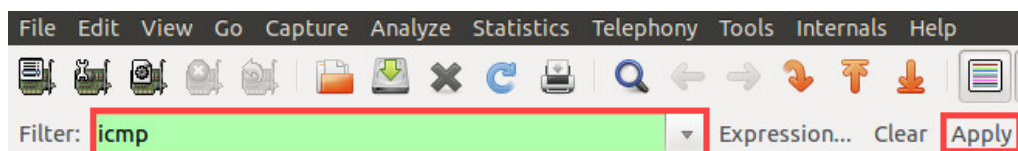
6. Change focus to the **Ubuntu** virtual machine.
7. Focus on the **Wireshark** application and make sure to clear the *Filter* by clicking on **Clear**.



8. Stop the capture.



9. Type **icmp** in the *Filter* field and click **Apply**.



10. Analyze the output. You should see the *ICMP* traffic of the *Win16* device traversing the *Ubuntu* device instead of going directly to the *pfSense* gateway. Notice that the destination *MAC* address for a ping request is actually *Ubuntu's* *MAC* address.

Filter:	icmp	▼	Expression...	Clear	Apply	
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.100	192.168.1.1	ICMP	74	Echo (ping) request
2	0.000024	192.168.1.100	192.168.1.1	ICMP	74	Echo (ping) request
3	0.000248	192.168.1.1	192.168.1.100	ICMP	74	Echo (ping) reply
4	0.000259	192.168.1.1	192.168.1.100	ICMP	74	Echo (ping) reply
9	1.015582	192.168.1.100	192.168.1.1	ICMP	74	Echo (ping) request
10	1.015606	192.168.1.100	192.168.1.1	ICMP	74	Echo (ping) request
11	1.015817	192.168.1.1	192.168.1.100	ICMP	74	Echo (ping) reply
12	1.015826	192.168.1.1	192.168.1.100	ICMP	74	Echo (ping) reply
21	2.031245	192.168.1.100	192.168.1.1	ICMP	74	Echo (ping) request
22	2.031271	192.168.1.100	192.168.1.1	ICMP	74	Echo (ping) request
23	2.031391	192.168.1.1	192.168.1.100	ICMP	74	Echo (ping) reply
24	2.031404	192.168.1.1	192.168.1.100	ICMP	74	Echo (ping) reply
31	3.046995	192.168.1.100	192.168.1.1	ICMP	74	Echo (ping) request

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

▶ Ethernet II, Src: Vmware_82:56:8f (00:50:56:82:56:8f), Dst: Vmware_9c:59:78 (00:50:56:9c:59:78)

▶ Destination: Vmware_9c:59:78 (00:50:56:9c:59:78)
▶ Source: Vmware_82:56:8f (00:50:56:82:56:8f)
Type: IP (0x0800)

▶ Internet Protocol Version 4, Src: 192.168.1.100 (192.168.1.100), Dst: 192.168.1.1 (192.168.1.1)

▶ Internet Control Message Protocol

11. Open another new terminal and enter the `ifconfig -a` command to view the *MAC* address for the *Ubuntu* system. You should see that the *MAC* address is `00:50:56:9c:59:78`.

```
student@Ubuntu:~$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:50:56:9c:59:78
          inet addr:192.168.1.100 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe9c:5978/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1787590 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1015211 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2152327773 (2.1 GB)  TX bytes:2109121489 (2.1 GB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:4108 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4108 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:286454 (286.4 KB)  TX bytes:286454 (286.4 KB)
```

12. The lab is now complete; you may end the reservation.