Security Insider Lab I - Report

by
Subbulakshmi Thillairajan, Fabian Göttl

Exercise 1:

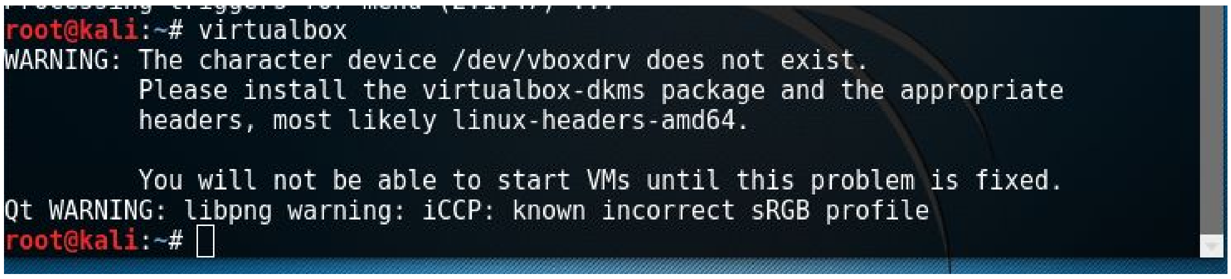
In order to get a Unix-based test environment, we copied an Ubuntu 16.04 VM disk image from the Lab's external hard drive and opened it in Virtualbox.
The virtual machine was configured with the following network settings:

- NAT Configuration
- Host IP: 10.0.0.33
- VM IP: 10.0.2.15

Next, we performed a package list and system update and installed all required applications by running:

- apt-get update
- sudo apt-get upgrade
- apt-get install apache2 php openssl openvpn iptables wireshark zenmap hping3

We faced a problem while running virtualbox at Kali. A warning message as given in Image 1 showed that the virtualbox-dkms package is missing, although it was installed.



```
root@kali:~# virtualbox
WARNING: The character device /dev/vboxdrv does not exist.
         Please install the virtualbox-dkms package and the appropriate
         headers, most likely linux-headers-amd64.

         You will not be able to start VMs until this problem is fixed.
Qt WARNING: libpng warning: iCCP: known incorrect sRGB profile
root@kali:~# 
```

Image 1: Warning message while running virtualbox.

We fixed the error by executing:

1. Sudo apt-get update
2. Sudo apt-get upgrade -y
3. Sudo apt-get dist -upgrade
4. Sudo apt-get autoclean
5. Sudo apt-get autoremove
6. Reboot

Finally, we have the two virtual machines *ClientVM* and *ServerVM*. We checked, if the apache is configured properly and successfully accessed the web server locally running at http://127.1.

Exercise 2:

First, we have to shut down VM. Then we are able to clone the VM by right-clicking at the VM machine's list entry.

Exercise 2.1:

Network Address Translation (NAT) is used to manage and direct the traffic between VM and an external network. All VM's share the host's internet connection and are able to connect to the internet. While using NAT, a direct communication between VM's is not possible, because the VMs are configured in a way like each VM is using a different virtual router. A similar network configuration is possible with *Bridged networking*. In this case, the VM is connected to the same network as the host. Therefore, the VM uses a IP within the same address block 10.0.0.* as the host. If both VMs share a identical MAC address, the network's DHCP server will allocate the same IP address to both machines.

Exercise 2.2:

We are able to save a VMs state by following methods:

- Taking a snapshot: Enables to revert the machine to this state at a later time.
- Suspend and resume: When we want to save the current state and continue with the same state later.
- Manual hard copy of files: The system is stored as image in one file. Hence, it can be cloned by copying the file.

Exercise 3:

We checked, if the apache web server is running by visiting http://127.1. The Apache2 test page was displayed.

Both Vms have *Bridged Networking* enabled and have the following IPs:

- VM Client IP: 10.0.0.48
- VM Server IP: 10.0.0.49

We visited http://10.49 at the ClientVM, which displayed the Apache2 test page.

Exercise 3.1:

In the used mode *Bridged networking* the host operating system is able to connect to the VMs web server. Also it is possible to connect to another PC in the lab as it is also connected to the same router as host.

Exercise 3.2:

For potential harmful attacks a more restrictive network mode would suitable. For example *Internal networking* only enables access between guests. Alternatively, *Host-only networking* only provides access between guests and from guests to host. In these modes, no external networks are in risk.

Exercise 3.3:

Type	Access Guest → Other guests	Access Host → Guest	Access Guest → External Network
Not attached	-	-	-
NAT	-	-	✔
NAT network	✔	-	✔
Bridged adapter	✔	✔	✔
Internal network	✔	-	-
Host-only adapter	✔	✔	-

Generic driver enables UDP tunnels and VDE networking betweens guests.

Exercise 4.1:
Promiscuous mode enables to monitor all traffic that the network card is able to see. If it is disabled, traffic that is addressed to the NIC is recorded only. The devices supporting promiscuous mode are able to sniff the whole traffic that is directed to their NIC.

Exercise 4.2:
Packets may have been filtered in a previous step of transmission. For example, a switch only directs packets to a port, if they are addressed to the according machine.

Exercise 4.3:
A non-root user can monitor interfaces by doing following steps:
Set run permissions of /usr/bin/dumpcap for wireshark group by executing:

- sudo dpkg-reconfigure wireshark-common
- Add current user to the wireshark group
- sudo gpasswd -a $USER wireshark

Exercise 4.4:
We installed the telnet server by executing:

- sudo apt-get install telnetd
- Edit /etc/inetd.conf and add line:
    - telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
- Restart inetd service:
    - /etc/init.d/openbsd-inetd restart
- Creating new unix account in Server VM by:

    - sudo useradd svsuser
    - sudo passwd svsuser
    - Typing in password *UniPassau* 2x
- Changing the network adapter of both Vms to *Host-only adapter*
- Successfully connected to ServerVM and tested a login over a telnet session

Exercise 4.5:
- Capture Filters: Filtering packets while capturing. These packet will not be recorded or stored.
- Display filters: They filters packets in the GUI's packet list.It possible to set color or to make them invisible.

Exercise 4.6:
Capture filter with source or destination IP address equal to 10.10.10.12 is as follows:

ip.addr == 10.10.10.12 ,

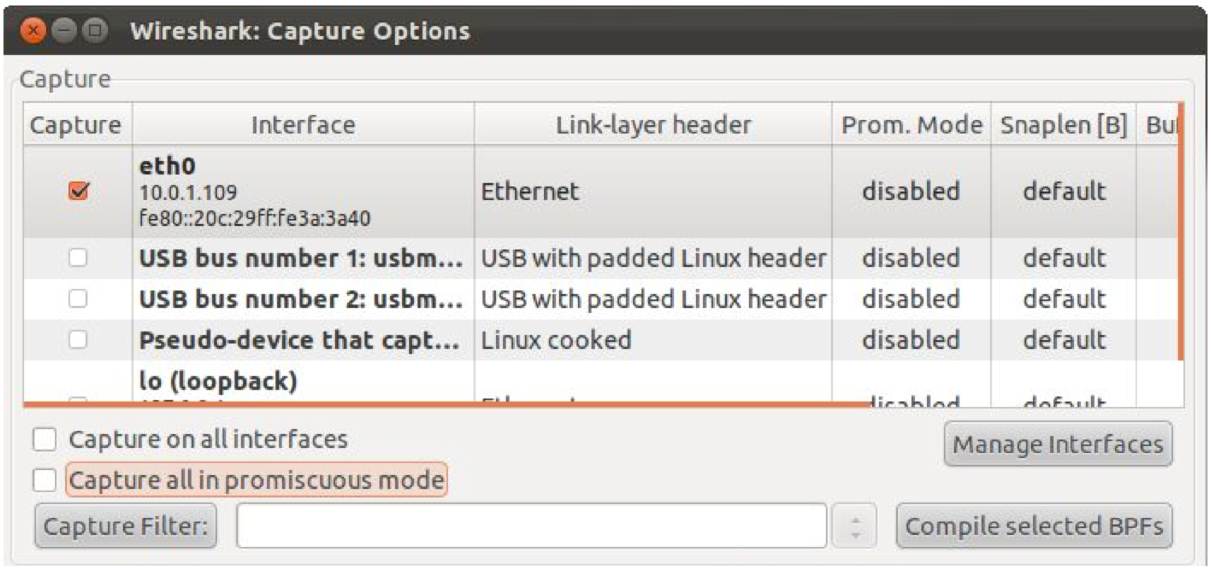
which is equivalent to

ip.src == 10.10.10.12

or

ip.dst == 10.10.10.12 .

Exercise 4.7:
The availability of *promiscuous mode* for a certain network interface is shown in the *Capture List* window. Additionally, packets recorded in *promiscuous mode* are labeled with a P flag. Image 2 shows the checkbox to enable promiscuous mode.


Wireshark: Capture Options

Capture

Capture	Interface	Link-layer header	Prom. Mode	Snaplen [B]	Bu
✓	eth0 10.0.1.109 fe80::20c:29ff:fe3a:3a40	Ethernet	disabled	default	
☐	USB bus number 1: usbm...	USB with padded Linux header	disabled	default	
☐	USB bus number 2: usbm...	USB with padded Linux header	disabled	default	
☐	Pseudo-device that capt...	Linux cooked	disabled	default	
☐	lo (loopback)				

☐ Capture on all interfaces

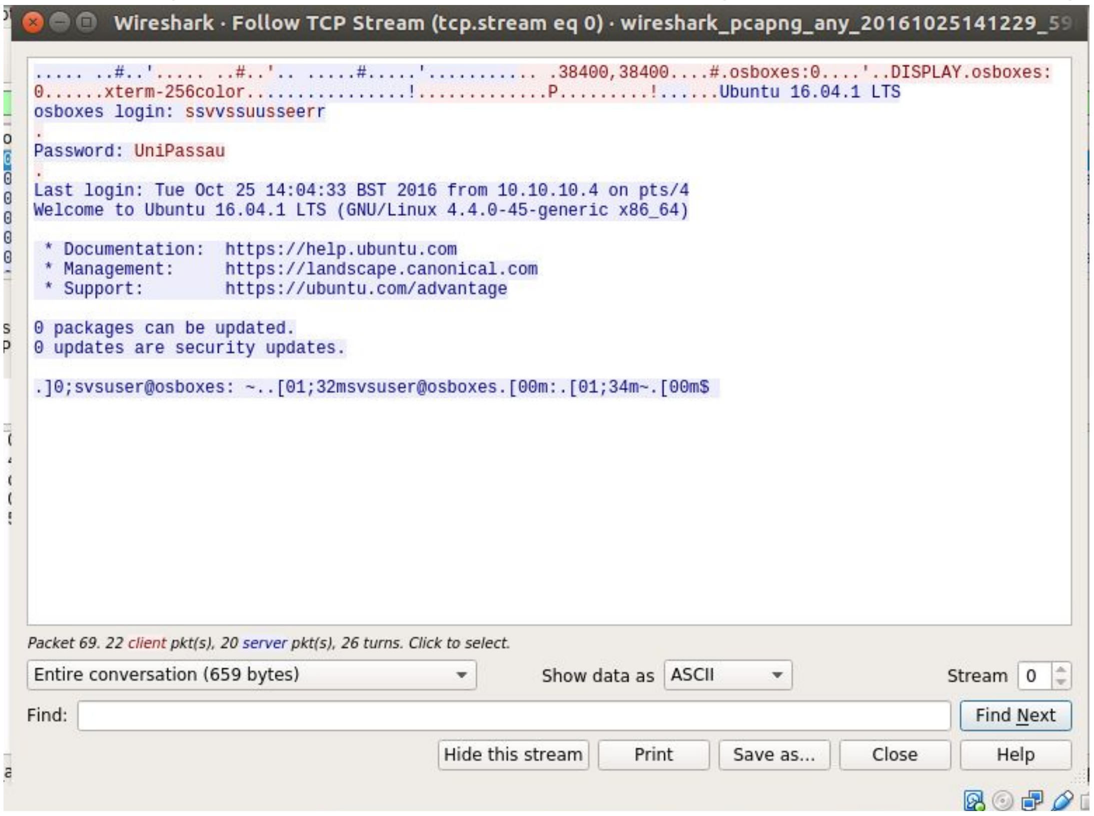
☐ Capture all in promiscuous mode

Image 2: Capture Options window listing all loggable interfaces. Additionally, it shows, if promiscuous mode is turned on for a interface.

Exercise 4.8:
1. The password gets transmitted from client to server. So we set a ip.src display filter to the ip address of the Client VM:

ip.src == 10.0.0.48

2. Each character that is typed in the telnet session gets transmitted in a own packet. The payload data can be read in the Data section of each packet's overview. The password can be read, because the packets are not encrypted. In order to see the content of the whole telnet session, we follow the TCP stream by right clicking at a packet and go to *Follow->Follow TCP Stream*. Image 3 shows the according window.


Wireshark · Follow TCP Stream (tcp.stream eq 0) · wireshark_pcapng_any_20161025141229_59

```
..... ..#..'..... ..#..'.. .....#.....'........... .38400,38400....#.osboxes:0....'..DISPLAY.osboxes:
0......xterm-256color...............!............P.........!......Ubuntu 16.04.1 LTS
osboxes login: ssvvssuusseerr
.
Password: UniPassau
.
Last login: Tue Oct 25 14:04:33 BST 2016 from 10.10.10.4 on pts/4
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-45-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

.]0;svsuser@osboxes: ~..[01;32msvsuser@osboxes.[00m:.[01;34m~.[00m$
```

Packet 69. 22 client pkt(s), 20 server pkt(s), 26 turns. Click to select.

Entire conversation (659 bytes)   Show data as ASCII   Stream 0

Image 3: Follow TCP Stream window shows the content of the telnet session.

Exercise 4.9:
Setting display filter to:
    http && ip.dst == 10.0.0.49

No foreign http traffic was recorded. Nothing suspicious was found.

Exercise 5:
Obtained information of zenmap are as follows: Open ports, offered protocol on a certain port, service name, service version number, operating system, kernel version, guessed uptime, hosted websites, NIC mac address, network distance

Exercise 5.1:
No, the zenmap values are not the same for all different machines. While comparing the values between a Debian and a Ubuntu VM we get different uptimes, kernel version, operating system names.

Exercise 5.2:
The TCP protocol allows certain parameters to be left up to the implementation. Each operating system and their subversions may have a different default values

set. These values can be used as signature, from which an OS may can be identified.

The signature may consist of:

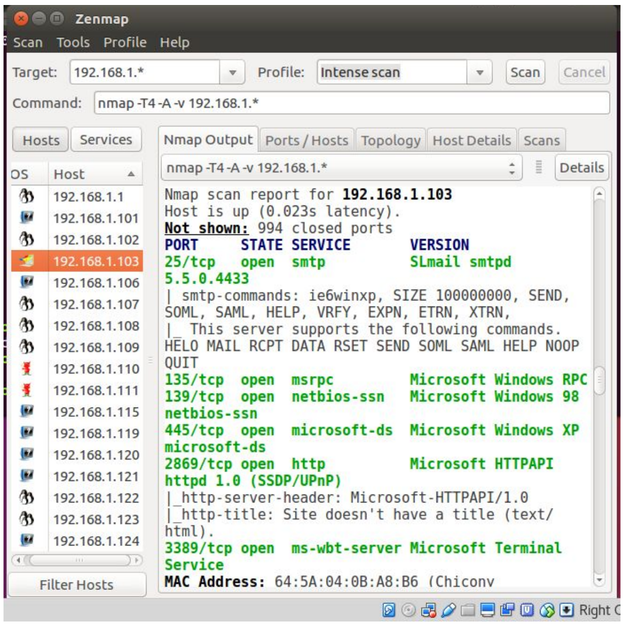
- Initial packet size (16 bits)
- Initial TTL (8 bits)
- Window size (16 bits)
- Max segment size (16 bits)
- Window scaling value (8 bits)
- "don't fragment" flag (1 bit)
- "sackOK" flag (1 bit)
- "nop" flag (1 bit)

Exercise 5.3:
Zenmap uses the unix tool nmap to scan for open ports. Nmap tries to connect to ports by opening a TCP connection, checks if a connection can be established, and finally closes the connection.

Exercise 5.4:
Zenmap uses nmap‘s OS fingerprinting technique, which sends up to 16 TCP, UDP and ICMP probes to known open/closed ports of the target. According to the responds, nmap uses the received data as signature and searches in its internal database for a relevant OS.

Exercise 5.5:
A service is a program that is listening on a port for connections and provides functionality.
It may listen to a default port, but can also be configured to listen to an arbitrary port.
A port enables to establish connections between a client and server application. A server is listening on certain port. A client connects to a given port.It is the end point of communication in a OS.

Exercise 5.6:
17 machines are running in the lab, among 7 UNIX-based machines, four unidentifiable operating systems and one Windows XP SP2/SP3 machine.



```
Nmap scan report for 192.168.1.103
Host is up (0.023s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE      VERSION
25/tcp   open  smtp         SLmail smtpd
5.5.0.4433
| smtp-commands: ie6winxp, SIZE 100000000, SEND,
SOML, SAML, HELP, VRFY, EXPN, ETRN, XTRN,
|_ This server supports the following commands.
HELO MAIL RCPT DATA RSET SEND SOML SAML HELP NOOP
QUIT
135/tcp  open  msrpc        Microsoft Windows RPC
139/tcp  open  netbios-ssn  Microsoft Windows 98
netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows XP
microsoft-ds
2869/tcp open  http         Microsoft HTTPAPI
httpd 1.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/1.0
|_http-title: Site doesn't have a title (text/
html).
3389/tcp open  ms-wbt-server Microsoft Terminal
Service
MAC Address: 64:5A:04:0B:A8:B6 (Chiconv
```

Image 4: Left: List of hosts with OS icons. **Right:** Port/Service scan overview of one host

Interesting machines:

IP	Service, Protocol, Port, Additional information
192.168.1.1	Linksys WRT54GL Router HTTP TCP 80: Password protected router configuration UpnP TCP 5431
192.168.1.102	VNC TCP 3389: May offer VNC (protocol 3.8) functionality, a connection via telnet was refused
192.168.1.103	SMTP TCP 25: Provides a smtpd 5.5.0.4433 mail server, may gets used to send mails MSRPC TCP 135 Netbios-ssn TCP 139: Windows 98 netbios service Microsoft-ds TCP 445: Windows XP-ds service Ldp TCP 646: filtered port HTTP TCP 2869: HTTPAPI httpd 1.0 service Ms-wbt-server TCP 3389: Terminal service

Exercise 6.1:

The ping utility is used for measure the response time between local and remote machine. This program sends an ICMP packet to the target machine, which may respond according to the protocol with a response packet. The packet includes a TTL number, which is decremented for each hop through which the packet is traveling.

For safety reasons, we changed the network settings to *internal networking*. This prevents access to foreign networks.

- Impersonating VM IP: 192.168.1.107
- NaiveVM IP: 192.168.1.109
- Host IP: 192.168.1.108
- Pinging the naïve VM by executing:
  - sudo hping3 192.168.1.109 -1
  - Set Wireshark display filter: ip.addr == 192.168.1.109
  - Wireshark captures ICMP packets: Echo (ping) request, Echo (ping) reply
- Spoof source IP address and set it to host address by executing:
  - sudo hping3 192.168.1.109 -1 -a 192.168.1.108
  - Wireshark captures ICMP packets *Echo (ping) request* from 192.168.1.108. The NaiveVM is responding to 192.168.1.108 with *Echo (ping) reply*.

Exercise 6.2:

- The command line for the spoofing PING request is:
  - sudo hping3 192.168.1.109 -1 -a 192.168.1.108
  - Root privileges are required to send ICMP packets
  - 192.168.1.109 is the target ip address
  - -1 command line argument: Send ping request by ICMP packets.
  - -a: Spoof the source address to the given IP address 192.168.1.108. Here the server will impersonate as the host, thus making it look like the host is trying to spoof the client.

Exercise 6.3:

The packets from the NaiveVM to the host gets dropped.

Exercise 7.1:

Without written confirmation we might break a system and risking a potential law suit. If we are scanning a harmful group, we might get attacked. Furthermore, our internet service provider might receive an abuse mail.

Exercise 7.2:

We were trying to use the SMTP Server at 192.168.1.103 to send mails:

Executing and sending a test e-mail: telnet 192.168.1.103 25

```
220 ie6winxp SMTP Server SLmail 5.5.0.4433 Ready ESMTP spoken here
MAIL FROM:<test@test.de>
250 OK
RCPT TO:<fabian@goettl.de>
250 OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
testmessage
.
250 OK, submitted and queued. (9E4466BF93DF4EEC9D5F1A24B8607C8E.SKM)
QUIT
```

The e-mail message was queued to send, but no mail was received.