Comparing Performance of Physical and Virtual Environment Penetration Testing Using Kali Linux

Yeraldina Estrella, BTech
The New York City College of Technology - CUNY, USA, YEstrella@CityTech.Cuny.edu

Mentor: Aparicio Carranza, PhD and Casimer DeCusatis, PhD
The New York City College of Technology - CUNY, USA, acarranza@CityTech.Cuny.edu

Marist College, USA, casimer.decusatis@marist.edu

Abstract– Penetration testing has become an important topic for modern cybersecurity environments. The widespread availability of virtual servers and appliances has made it possible to conduct some types of penetration testing in a virtual environment. This paper investigates the creation of a virtual penetration testing environment using Kali Linux, and compares its performance with a traditional physical testing environment. We also investigate the creation of a virtual network with five different operating systems to perform penetration testing, and compare which operating system is more secure. Tools such as Armitage are used to perform all exploits, Zenmap to access attacking information and Johnny to crack user passwords.

Keywords-- Virtualization, Kali Linux, Metasploit, Armitage, Zenmap.

I. Introduction

Server virtualization has become an essential element of many enterprise and cloud computing data centers [1-3]. Guest virtual machines (VMs) replicate the physical aspects of a host server, and are managed from the VM hypervisors. Hypervisors are commonly used to test the efficiency of new programs and operating systems without harming the host server. Well known hypervisors include VMWare, KVM, Red Hat Enterprise Virtualization, Citrix XenServer, and Microsoft Windows Server [4]. These virtual environments represent a new way to perform penetration testing and evaluate the security of a given hypervisor or operating system. After creating different virtual machines, tools already available in Kali Linux can simultaneously attack the different virtual machines. The results can show the level of difficulty in penetrating a given operating system. In this paper, we compare the behavior of several different operating systems when subjected to penetration test tools available in Kali Linux. We also compare the behavior of physical and virtual machines used for penetration testing applications

Our testing uses VMware Workstation, which is among the most popular type 2 hypervisors. A type 1 hypervisor has direct access to the physical hardware while a type 2 hypervisor loads inside an operating system like an application (see Fig. 1). VMware runs a similar hypervisor which is free for non-commercial use, called VMware Player, which lacks some of the capabilities of VMware Workstation [5-7]. VMware Workstation requires a 64 bit CPU to work, although it allows the creation of both 32 and 64 bit guest machines. With VMware Workstation, virtual machines can be created with 16 virtual processors. It has an 8TByte virtual disk, allowing each virtual machine to have 64 GBytes of memory. The virtual machines support USB, Bluetooth, SCSI, SATA and IDE disk controllers.

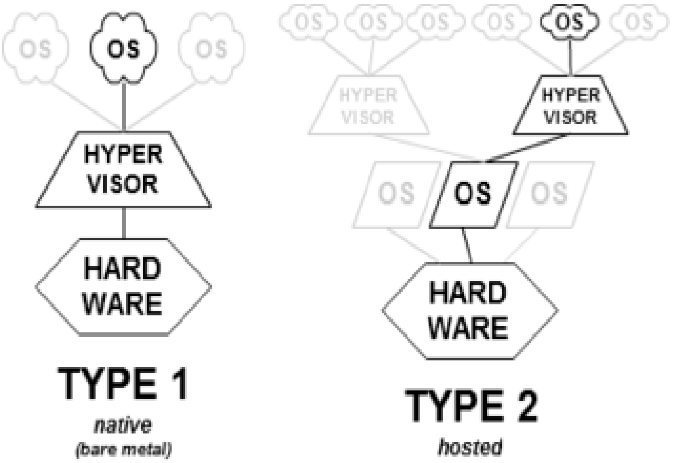


Fig. 1 - Type 1 and Type 2 Hypervisor

The physical environment we created consists of two Windows 7 OS machines (with and without Antivirus software), Ubuntu, Windows 8 and Kali Linux. Ubuntu was tested with and without firewall security. The following table shows the IPv4 addresses for each of the physical machines used:

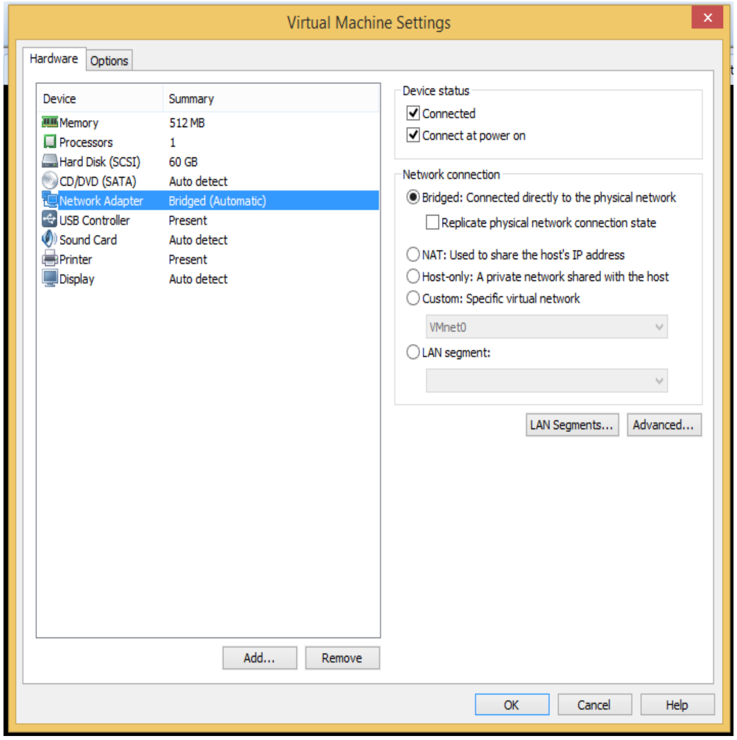
Operating System	IP Address
Kali Linux	192.168.1.5
Windows 7 (*no antivirus*)	192.168.1.8
Windows 7 (*with Antivirus*)	192.168.1.2
Windows 8	192.168.1.9
Ubuntu	192.168.1.6
Mac OS	192.168.1.4

Table 1 – Physical environment machines' IP address

II. CREATING A VIRTUAL ENVIRONMENT

The first step in creating a virtual environment is the creation of guest VMs. An ISO image of the operating system is required. From there, selecting File > New Virtual Machine leads the user to a virtual machine wizard. Within the virtual machine wizard the ISO file‘s location is selected. Workstation recognizes the type of operating system once the ISO image is mounted. From there the user name and password of the guest machine can be set as well as the amount of RAM and hard disk space [8]. Another benefit of hypervisors is that they indicate how much RAM and hard disk space is recommended for the virtual machine to run efficiently. For Linux based machines the recommended value is 20 GBytes. For Windows based machines, the recommended value is 60 GBytes, although Workstation will run reasonably well with as little as 30 GBytes if there are a small number of programs to be installed.

After the virtual machine creation is complete, an operating system must be installed. The five operating systems used for this work were Windows 7, Windows 8, Ubuntu 14.04, CentOS and Kali Linux 1.0.6. All operating systems we created were 64 bit. A virtual network must be created between the VMs, even if only a simple point-to-point connection is required. The VMs must be provisioned as if they were bare metal systems; IP addresses and default gateways as well as other settings either have to be configured or retrieved from a configuration file. The simplest way to connect the virtual machines in a network is to use configuration information from the host machine to create its own network. In order to do so, the user has to right-click on the virtual machine and go to settings. The settings menu will open hardware settings as shown in the screenshot of Fig. 2. The network adapter has to be selected. Within Network Adapter there are five types of connections that can be made: Bridges, NAT, Host-only, Custom and LAN segment[8]. With a Bridged connection, the virtual machine uses the settings of the actual physical network. NAT shares the hosts IP address. Host-only connection only creates one link between the virtual machine and the host. A custom connection can be used to create a virtual network, however, there wouldn’t be any settings configured. LAN segment is similar to the custom connection where a local area network is created and all of the virtual machines are connected to that segment. The preferred setting is the automatic “Bridged” connection and each virtual machine’s Network Adapter has to be set to Bridged. When this setting is selected for each virtual machine, they are individually assigned IP addresses in the same network. They are also assigned proper default gateways. Creating the Virtual Network relies on this setting within the Network Adapter [8].



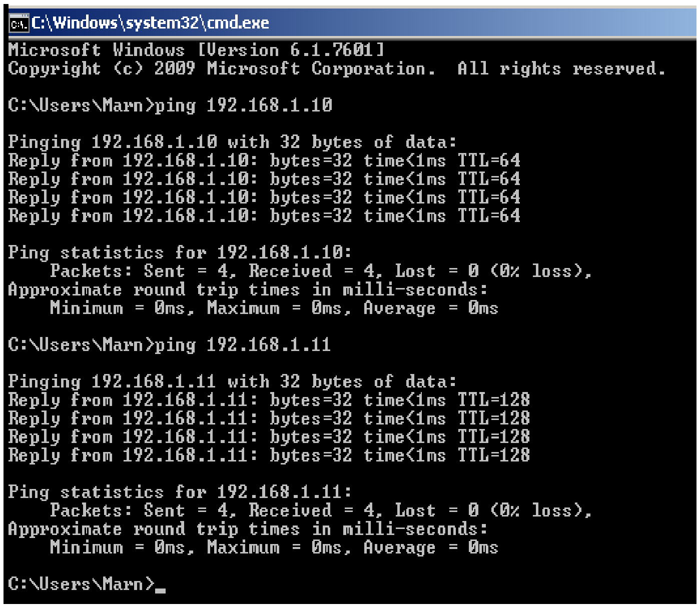
Device	Summary
Memory	512 MB
Processors	1
Hard Disk (SCSI)	60 GB
CD/DVD (SATA)	Auto detect
Network Adapter	Bridged (Automatic)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Device status
☑ Connected
☑ Connect at power on

Network connection
● Bridged: Connected directly to the physical network
□ Replicate physical network connection state
○ NAT: Used to share the host's IP address
○ Host-only: A private network shared with the host
○ Custom: Specific virtual network
VMnet0
○ LAN segment:

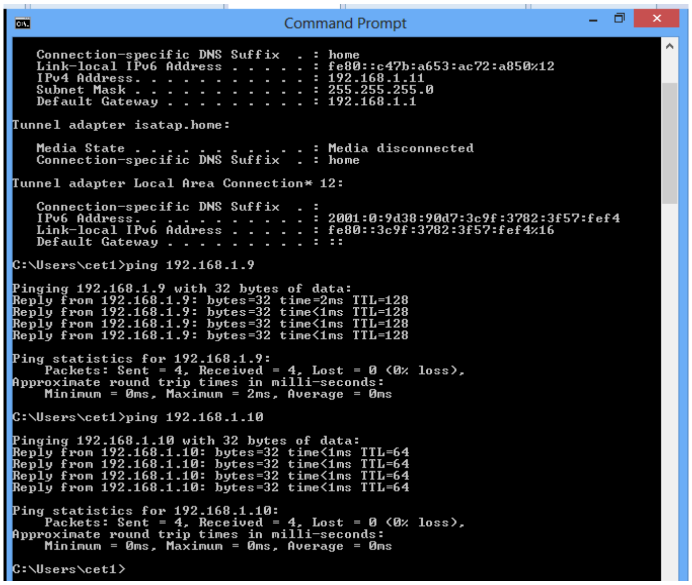
Fig. 2 – Hardware settings menu

In order to test the connection between the different virtual machines, “*ipconfig*” needs to be used for Windows 7 and 8 to determine the IP addresses. The IP addresses for CentOS and Ubuntu are obtained with the command “*ifconfig*”. After the IP addresses are found, pinging tests if the machines are connected. Screenshots showing the successful pinging between the four virtual machines are shown in Figs. 3 -6



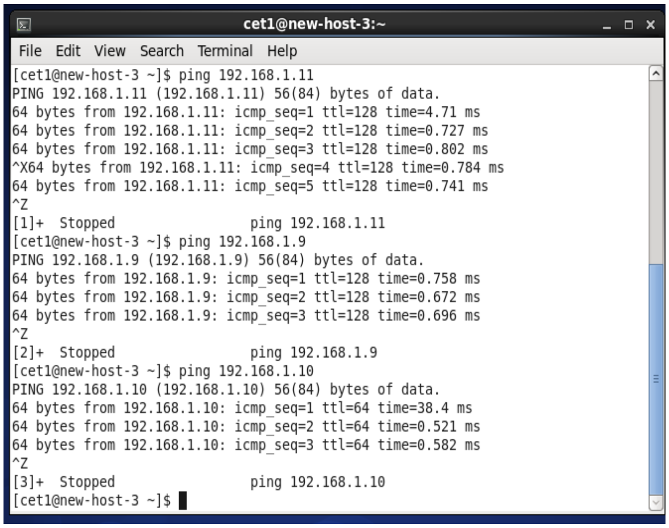
```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Marn>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: bytes=32 time<1ms TTL=64
Reply from 192.168.1.10: bytes=32 time<1ms TTL=64
Reply from 192.168.1.10: bytes=32 time<1ms TTL=64
Reply from 192.168.1.10: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Marn>ping 192.168.1.11

Pinging 192.168.1.11 with 32 bytes of data:
Reply from 192.168.1.11: bytes=32 time<1ms TTL=128
Reply from 192.168.1.11: bytes=32 time<1ms TTL=128
Reply from 192.168.1.11: bytes=32 time<1ms TTL=128
Reply from 192.168.1.11: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.11:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Marn>_
```

Fig. 3 - Windows 7 Pinging Ubuntu and Windows 8



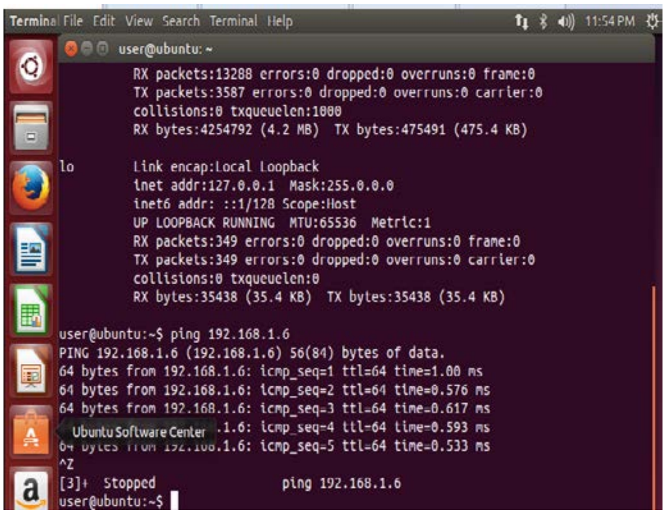
```
   Connection-specific DNS Suffix  . : home
   Link-local IPv6 Address . . . . . : fe80::c47b:a653:ac72:a850%12
   IPv4 Address. . . . . . . . . . . : 192.168.1.11
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.1.1

Tunnel adapter isatap.home:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : home

Tunnel adapter Local Area Connection* 12:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2001:0:9d38:90d7:3c9f:3782:3f57:fef4
   Link-local IPv6 Address . . . . . : fe80::3c9f:3782:3f57:fef4%16
   Default Gateway . . . . . . . . . : ::

C:\Users\cet1>ping 192.168.1.9

Pinging 192.168.1.9 with 32 bytes of data:
Reply from 192.168.1.9: bytes=32 time=2ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128
Reply from 192.168.1.9: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.9:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 0ms

C:\Users\cet1>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:
Reply from 192.168.1.10: bytes=32 time<1ms TTL=64
Reply from 192.168.1.10: bytes=32 time<1ms TTL=64
Reply from 192.168.1.10: bytes=32 time<1ms TTL=64
Reply from 192.168.1.10: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\cet1>
```

Fig. 4 - Windows 8 Pinging Windows 7 & Ubuntu



```
[cet1@new-host-3 ~]$ ping 192.168.1.11
PING 192.168.1.11 (192.168.1.11) 56(84) bytes of data.
64 bytes from 192.168.1.11: icmp_seq=1 ttl=128 time=4.71 ms
64 bytes from 192.168.1.11: icmp_seq=2 ttl=128 time=0.727 ms
64 bytes from 192.168.1.11: icmp_seq=3 ttl=128 time=0.802 ms
^X64 bytes from 192.168.1.11: icmp_seq=4 ttl=128 time=0.784 ms
64 bytes from 192.168.1.11: icmp_seq=5 ttl=128 time=0.741 ms
^Z
[1]+  Stopped                 ping 192.168.1.11
[cet1@new-host-3 ~]$ ping 192.168.1.9
PING 192.168.1.9 (192.168.1.9) 56(84) bytes of data.
64 bytes from 192.168.1.9: icmp_seq=1 ttl=128 time=0.758 ms
64 bytes from 192.168.1.9: icmp_seq=2 ttl=128 time=0.672 ms
64 bytes from 192.168.1.9: icmp_seq=3 ttl=128 time=0.696 ms
^Z
[2]+  Stopped                 ping 192.168.1.9
[cet1@new-host-3 ~]$ ping 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=64 time=38.4 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=64 time=0.521 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=64 time=0.582 ms
^Z
[3]+  Stopped                 ping 192.168.1.10
[cet1@new-host-3 ~]$ 
```

Fig. 5 - Centos Pinging Windows 8, Windows 7 & Ubuntu



```
          RX packets:13288 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3587 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4254792 (4.2 MB)  TX bytes:475491 (475.4 KB)

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:349 errors:0 dropped:0 overruns:0 frame:0
          TX packets:349 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:35438 (35.4 KB)  TX bytes:35438 (35.4 KB)

user@ubuntu:~$ ping 192.168.1.6
PING 192.168.1.6 (192.168.1.6) 56(84) bytes of data.
64 bytes from 192.168.1.6: icmp_seq=1 ttl=64 time=1.00 ms
64 bytes from 192.168.1.6: icmp_seq=2 ttl=64 time=0.576 ms
64 bytes from 192.168.1.6: icmp_seq=3 ttl=64 time=0.617 ms
                         .1.6: icmp_seq=4 ttl=64 time=0.593 ms
64 bytes from 192.168.1.6: icmp_seq=5 ttl=64 time=0.533 ms
^Z
[3]+  Stopped                 ping 192.168.1.6
user@ubuntu:~$ 
```

Fig. 6 - Ubuntu pinging Windows 8, Windows 7 & CentOS

III. PENETRATION TESTING WITH ARMITAGE

Kali Linux is an open source Linux based operating system which is particularly well known for its penetration testing applications. Various tools for penetration testingare found under the menu option Applications > Kali Linux. The available tests include Information Gathering Tests, Vulnerability Analysis Tests, Web Application Tests, Password Attacks, Wireless Attacks, Exploitation tools, Sniffing / Spoofing and much more. Some tests run under command line while others run in GUI mode [4].

Once a penetration tester gathers enough information about the targets, they can spend more time understanding the target's vulnerabilities. Using these vulnerabilities for different types of attacks isthe goal of exploitation testing.   A key part of Kali Linux used for exploitation testing is the Metasploit framework. Metasploit is one of the top ten server-side attacks in Kali Linux. It is used to penetrate various targets simultaneously and explore their vulnerabilities for analysis. The GUI equivalent of Metasploit is Armitage [3]. Armitage helps penetration testers see where they are attacking. Any action performed on Armitage is translated into a command for the Metasploit Framework making it user friendly for beginners.

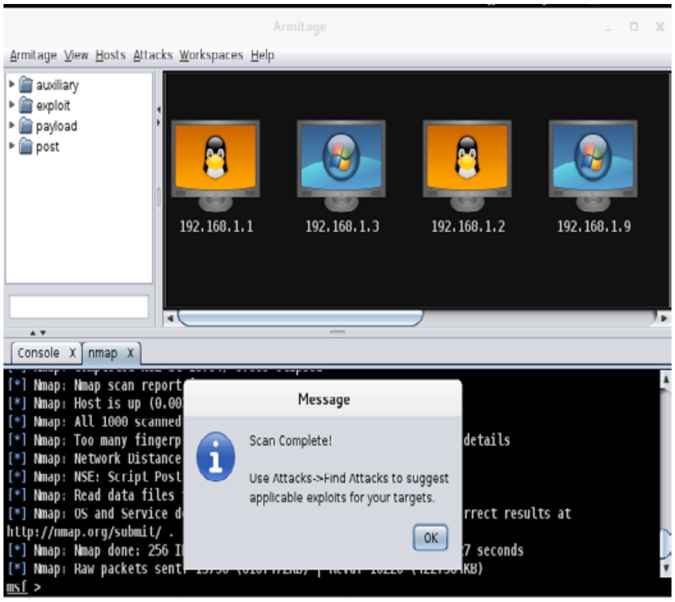
Armitage is not readily available within Kali Linux. In order to download it, the command **apt-get install update** has to be entered within the root terminal.This will ensure that Armitage installs and functions properly. After running the update enter the command apt-get install Armitage within the root terminal [2]. If the user tries to open Armitage after installation, they must first start postgresql by entering the command **service postgresqlstart**. Then use the command **service Metasploit start** to begin the Metasploit service.

After starting the proper services, Armitage can be started by entering the command **Armitage** in root terminal. Armitage will try to connect with an assigned host IP, password, and port. The password can be changed as desired. To open up the GUI, click **connect** as shown in figure 8 (additional related results are shown in figures 9-11). After connecting there will be a loading notification; this is used to indicate that Armitage is trying to connect, and may have a misleading message that the connection was refused when in fact the connection attempt is still in progress; do not cancel this loading notification.

Once the connection is made the GUI will be loaded up but showing no targets or any exploits. The targets have to be scanned. This is where the network is vital because if Kali Linux was not connected within a network with the other virtual machines, Armitage would not be able to scan them as targets.

To scan the targets, within the menu bar there is an option of Hosts > Nmap Scan. Select Intense Scan as it is more thorough than a Quick Scan. A Quick Scan will yield the same results. Intense Scans are more accurate, however, and include all the possible exploits per target. Once this option is selected, the user will be prompted to enter an IP address range for the targets. The IP address range for our virtual machines are from 192.168.1.6 – 192.168.1.11. The IP address range for our physical machines is from 192.168.1.1 – 192.168.1.10. However, all targets can be scanned by entering the 192.168.1.0/24.

When all the targets have been scanned, a completion notification will pop up. It also tells the user where to go for suggested attacks. Each target is vulnerable to a specific attack as shown in the Armitage workspace along with the type of OS for each target. The OS is determined by the symbols on the computer screens. The windows symbol means it is a Windows flavor. Linux is represented by a penguin mascot with an orange background.



```
[*] Nmap: Nmap scan report
[*] Nmap: Host is up (0.00
[*] Nmap: All 1000 scanned
[*] Nmap: Too many fingerp                                    details
[*] Nmap: Network Distance
[*] Nmap: NSE: Script Post
[*] Nmap: Read data files
[*] Nmap: OS and Service d                                    rrect results at
http://nmap.org/submit/ .
[*] Nmap: Nmap done: 256 I                                    7 seconds
[*] Nmap: Raw packets sent                                    KB)
msf >
```

Message

Scan Complete!

Use Attacks->Find Attacks to suggest applicable exploits for your targets.

OK

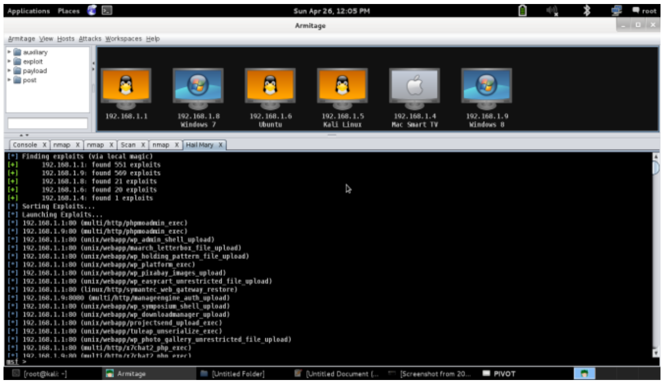
Fig. 7 – Armitage threat analysis results (*Virtual Machines*)

After the targets have been shown, the user can either right click on the desired target and select an attack or perform the so-called Hail Mary Attack. The Hail Mary attack performs all possible exploits on the desired targets. The Hail Mary attack is performed by going to Attacks > Hail Mary. Since it is a thorough scan it will take a few minutes to complete [2].

Although there were five virtual machines created, Armitage detects all possible targets. It is important to know which IP address is assigned to which operating system so that the results can be properly understood. In the physical environment, there were six different machines connected to the network. The Hail Mary attack will list the number of exploits for each virtual machine and physical machine.

The results for the VM show that CentOS had 9 exploits, Windows 7 and Windows 8 had 22 exploits. Armitage doesn't allow the attack on Ubuntu which is why it was not detected as a target or attacked.

Armitage did not detect the Ubuntu (*Firewall enabled*) machine and the windows 7 machine with Avast antivirus in the physical environment. Armitage doesn't allow the attacks on Ubuntu when the firewall is enabled neither on windows 7 with Antivirus software. Nevertheless, Armitage found exploits for the Ubuntu machine when we disabled the firewall. The results for the Physical machines show that Ubuntu had 20 exploits, Windows 7 had 21exploits, and Windows 8 had 569 exploits. The Mac machine that was connected to the network via wireless was detected by Armitage. Many times, a specific flavor of Windows, Mac or Linux is safer due to the fact that there are less penetration tools made for that operating system.



```
[*] Finding exploits (via local magic)
[+]      192.168.1.1: found 551 exploits
[+]      192.168.1.9: found 569 exploits
[+]      192.168.1.8: found 21 exploits
[+]      192.168.1.6: found 20 exploits
[+]      192.168.1.4: found 1 exploits
[*] Sorting Exploits...
[*] Launching Exploits...
[*] 192.168.1.1:80 (multi/http/phpmoadmin_exec)
[*] 192.168.1.9:80 (multi/http/phpmoadmin_exec)
[*] 192.168.1.1:80 (unix/webapp/wp_admin_shell_upload)
[*] 192.168.1.1:80 (unix/webapp/maarch_letterbox_file_upload)
[*] 192.168.1.1:80 (unix/webapp/wp_holding_pattern_file_upload)
[*] 192.168.1.1:80 (unix/webapp/wp_platform_exec)
[*] 192.168.1.1:80 (unix/webapp/wp_pixabay_images_upload)
[*] 192.168.1.1:80 (unix/webapp/wp_easycart_unrestricted_file_upload)
[*] 192.168.1.1:80 (linux/http/symantec_web_gateway_restore)
[*] 192.168.1.9:8080 (multi/http/manageengine_auth_upload)
[*] 192.168.1.1:80 (unix/webapp/wp_symposium_shell_upload)
[*] 192.168.1.1:80 (unix/webapp/wp_downloadmanager_upload)
[*] 192.168.1.1:80 (unix/webapp/projectsend_upload_exec)
[*] 192.168.1.1:80 (unix/webapp/tuleap_unserialize_exec)
[*] 192.168.1.1:80 (unix/webapp/wp_photo_gallery_unrestricted_file_upload)
[*] 192.168.1.1:80 (multi/http/x7chat2_php_exec)
[*] 192.168.1.9:80 (multi/http/x7chat2_php_exec)
msf >
```

Fig. 8 – Armitage Hail Mary Attack results (*Physical Machine*)

IV. PENETRATION TESTING WITH ZENMAP

Zenmap is another open source penetration test that helps gather information about different targets. It is the GUI version of the widely used Nmap (or Network Mapper). Like most tools in Kali Linux, Zenmap is accessed by going to Applications > Kali Linux > Information Gathering > Network Scanners > Zenmap. After selecting the tool, the GUI should open on its own as shown in Fig. 9.

A custom profile attack can be made with Zenmap as well as a regular manual attack. If a profile is created, the targets can be indicated as well as the attack and the results would be saved within that profile. In order to do that, the new profile has to be created by going to Profile > New Profile or Command. Then indicate the range of targets. Select to enable all advanced/aggressive options. Select operating system detection so the result shows the operating system of the target [4].

If the whole range of IP addresses is chosen, it can be difficult for the testing to locate a specific target since the list may be quite long. Another way to scan for information from the desired targets is by inputting the desired IP addresses in the command box. The target will automatically adjust. Once the targets have been added, select the type of Scan (*we used Intense Scan*). Then click Scan to begin running and retrieving information.

Zenmap contains a feature that helps the tester display network connections for different machines, as shown in figure 10.This is helpful in learning what other targets they can have access to through the target. The network is shown on a graph with labeled IP addresses.

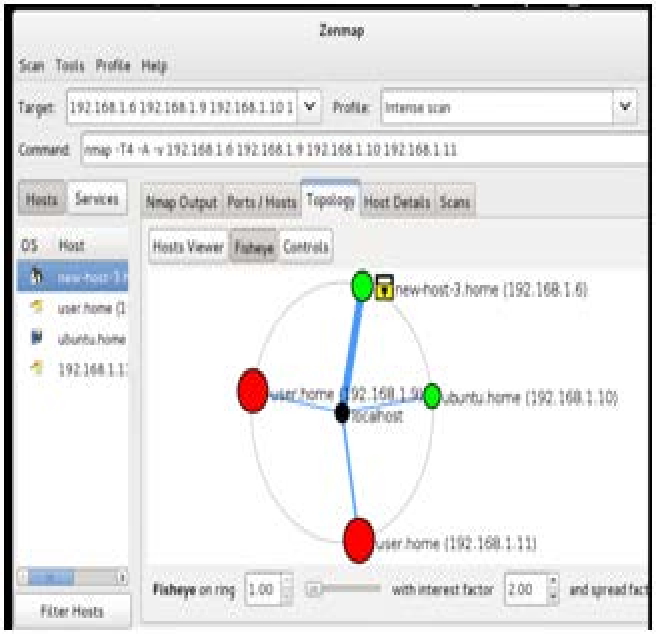

Target: 192.168.1.6 192.168.1.9 192.168.1.10 1
Profile: Intense scan
Command: nmap -T4 -A -v 192.168.1.6 192.168.1.9 192.168.1.10 192.168.1.11
Fisheye on ring 1.00 with interest factor 2.00 and spread fact

Fig. 9– Zenmap network connection results (*Virtual Machines*)

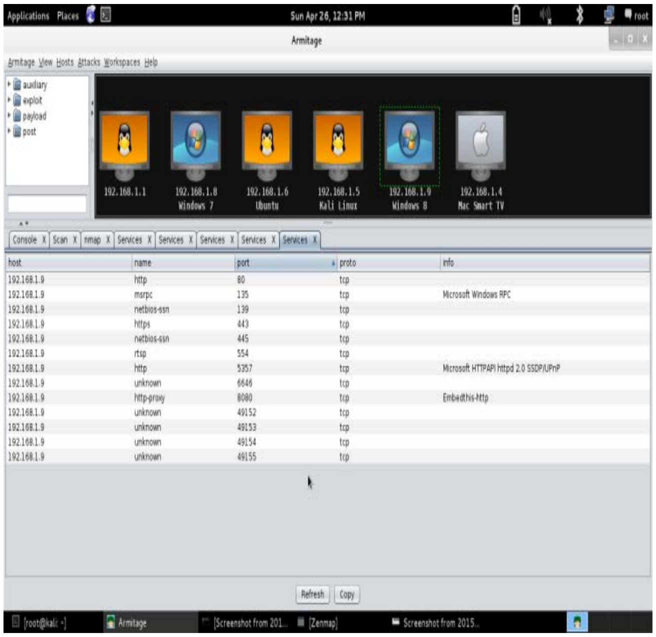
In our case, Figure 10 and figure 11 shows that all of the systems in the VMand the physical machines are connected totheir respective local host as well as each machine. The green dot represents that the system is a Linux based OS. The red dot represents a Windows based OS. The yellow dot represents other systems such as Mac based OS and the router.

As mentioned previouslyin the virtual environment, Zenmap recognized the OS of Ubuntu, but does not recognize any ports or information for Windows 8. Due to the image next to the IP address, Zenmap recognizes it's a Windows target. From the Nmap output, we can see the Windows 8 was scanned. However, Zenmap can't retrieve port information or host details. This means Windows 8 is more secure than Windows 7. Windows 7 shows 8 ports that are open while CentOS only shows a single open port. Ubuntu and Windows 8 on the other hand show no open ports. In other words, Windows 7 had more ports open for attack as compared to CentOS. Ubuntu was recognized, however, it did not have any open ports for attack.

As shown in figure 10 and 11, Zenmap recognized the OS of Ubuntu, Windows 7 (*without antivirus software*), Windows 8 and Mac OS. Nevertheless, Zenmap cannot retrieve port information or host details for Ubuntu when its firewall was enabled. From the Nmap output we can see the Windows 8 was scanned and shows 13 open ports. Nevertheless, in the Zenmap scan results we can see that Windows 8 had 15 open ports.

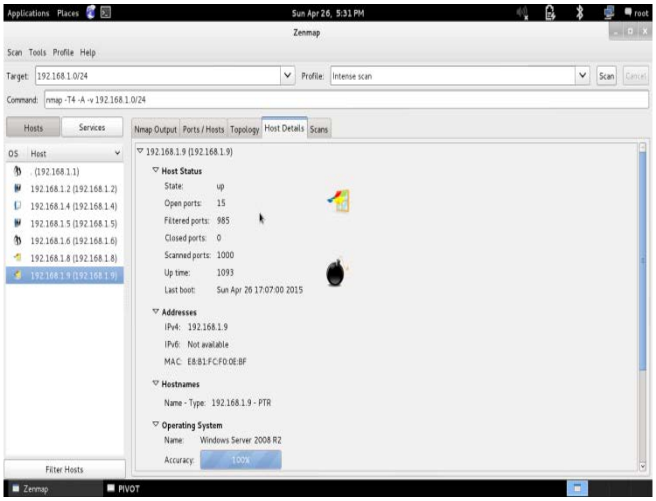
Both, Nmap and Zenmap show that Windows 7(*without antivirus software*) had 7 open ports. However, Zenmap can't retrieve port information or host details for the windows 7 machine with Antivirus software. The Nmap scan of Ubuntu shows 2 open ports. Similarly, the Zenmap scan report shows that Ubuntu (disabled firewall) had 2 open ports. Nmap recognized the Mac OS machine and found only one open port. Nevertheless, the Zenmap result shows that the Mac OS machine has 5 open ports.

The results for the physical environment means Windows 7 is more secure than Windows 8. Windows 7 shows 7 ports that are open while Windows 8 shows 13 open ports. Ubuntu, Mac OS and Windows 7OS showless open ports compared to Windows 8. In other words, Windows 8 had more ports open for attack as compared to the other operating systems.



host	name	port	proto	info
192.168.1.9	http	80	tcp	
192.168.1.9	msrpc	135	tcp	Microsoft Windows RPC
192.168.1.9	netbios-ssn	139	tcp	
192.168.1.9	https	443	tcp	
192.168.1.9	netbios-ssn	445	tcp	
192.168.1.9	rtsp	554	tcp	
192.168.1.9	http	5357	tcp	Microsoft HTTPAPI httpd 2.0 SSDP/UPnP
192.168.1.9	unknown	6646	tcp	
192.168.1.9	http-proxy	8080	tcp	Embedthis-http
192.168.1.9	unknown	49152	tcp	
192.168.1.9	unknown	49153	tcp	
192.168.1.9	unknown	49154	tcp	
192.168.1.9	unknown	49155	tcp	

Fig. 10– Armitage service results for Windows 8 *(Physical Machines)*



```
Target: 192.168.1.0/24          Profile: Intense scan
Command: nmap -T4 -A -v 192.168.1.0/24

. (192.168.1.1)
192.168.1.2 (192.168.1.2)
192.168.1.4 (192.168.1.4)
192.168.1.5 (192.168.1.5)
192.168.1.6 (192.168.1.6)
192.168.1.8 (192.168.1.8)
192.168.1.9 (192.168.1.9)

▽ 192.168.1.9 (192.168.1.9)
  ▽ Host Status
    State:          up
    Open ports:     15
    Filtered ports: 985
    Closed ports:   0
    Scanned ports:  1000
    Up time:        1093
    Last boot:      Sun Apr 26 17:07:00 2015
  ▽ Addresses
    IPv4: 192.168.1.9
    IPv6: Not available
    MAC:  E8:B1:FC:F0:0E:BF
  ▽ Hostnames
    Name - Type: 192.168.1.9 - PTR
  ▽ Operating System
    Name:     Windows Server 2008 R2
    Accuracy: 100%
```

Fig. 11– Zenmap results for Windows 8 *(Physical Machines)*

V. CONCLUSIONS

The tools from Kali Linux were used to help determine which operating system was tested to be most secure. By using Armitage and Zenmap the conclusions that were drawn were that in general Linux based systems are more secure than Windows. In the virtual environment, Armitage was unable to scan Ubuntu and only found 9 exploits for CentOS. Windows 7 and 8 on the other hand found 22 exploits. Windows 8 proves to be slightly more secure than Windows 7 due to the Zenmap results.

On the contrary, in the physical environment, Armitage was unable to scan Windows 7 with antivirus software and Ubuntu when its firewall was enabled. The Nmap scan on Windows 7 (*without antivirus*) found 21 exploits and 20 exploits for Ubuntu (disabled firewall). Windows 7and Ubuntu prove to be slightly more secure than Windows 8 due to the Zenmap results.

As mentioned before, many times operating systems that aren't commonly used are more secure. This is due to testers creating tools that are directed to attack the commonly used operating systems. It was assumed that Mac OS, Windows 7 and Ubuntu would be more secure than Windows 8 based on Armitage results alone. However, with Zenmap Mac OS machine has 5 ports open and Windows 7 (*with antivirus*) showed 7 ports open for attack. Ubuntu (*disabled firewall*) showed only 2 open ports for attack. Windows 7 (*with antivirus*) and Ubuntu *(enabled firewall*) showed no open ports shown in the Zenmap scan. The Nmap results and the Zenmap reults clearly demonstrate that Windows 8 is extremely more vulnerable than the other operating system machines. This is why testing more than one tool is important because from Armitage it would not be clear enough which OS is more secure. Combining analysis with Zenmap created a clearer picture. The results show that penetration testing using virtual and physical machines yields comparable results, provided that equivalent firewalls are implemented in both cases. Thus, virtual machines allow you to study penetration testing at much lower cost.

REFERENCES

[1] K. Jamsa, *Cloud Computing*, Jones & Bartlett Press, NY (2014)

[2] *Armitage Tutorial*. N.p., n.d. Web. 26 May 2014. http://www.fastandeasyhacking.com/manual

[3] Muniz, Joseph, and Aamir Aakhani. *Web Penetration Testing with Kali Linux a practical guide to implementing penetration testing strategies on websites, web applications, and standard web protocols with Kali Linux*. Birmingham: Packt Publishing, 2013

[4] "The Top 5 Enterprise Type 1 Hypervisors You Must Know." *Virtualization Software*.N.p., n.d. Web. 26 May 2014. http://www.virtualizationsoftware.com/top-5-enterprise-type-1-hypervisors/

[5] Troy, Ryan, and Matthew Helmke.*VMware cookbook*. Sebastopol, CA: O'Reilly Media, 2010. Print.

[6] "VMware Workstation." *: Run Multiple OS, Linux, Windows 8 & More;*. N.p., n.d. Web. 26 May 2014. http://www.vmware.com/products/workstation/

[7] "VMware Fusion Documentation Center." *VMware Fusion Documentation Center*.N.p., n.d. Web. 26 May 2014. http://pubs.vmware.com/fusion-4/index.jsp?topic=/com.vmware.fusion.help.doc/GUID-E498672E-19DD-40DF-92D3-FC0078947958.html

[8] "Zenmap - Official cross-platform Nmap Security Scanner GUI." *Zenmap - Official cross-platform Nmap Security Scanner GUI*.N.p., n.d. Web. 26 May 2014. <http://nmap.org/zenmap/>