CS 4235/6036 Lab One
Spring, 2016
Joel Odom, Course Instructor
Evan Downing, Teaching Assistant Responsible for this Lab

There are 20 points available for this lab. This lab is due Friday, 11 March. ***Please start early!*** You are encouraged to work in small groups of at most three students. We require only one submission per group. Remember to list all of your group members on your submission. *Include enough writing and screen captures in your submission so that we can see clearly that you completed the lab and so that we can award partial credit where possible.*

This lab was written and tested using VirtualBox. We recommend that you use it for consistency. You may use a different setup (VMware, QEMU, physical, etc.), but your results may differ from our results. It will be easiest for us to grade (and thusly in your best interest) to use the same setup and we did to write and test the lab.

Set up Kali Linux (2 Points)

Our course will use the Kali Linux penetration testing tool for our labs. Set up a Kali system any way you wish (virtual or otherwise), but we recommend you set it up as a virtual system (*i.e.*, using VirtualBox). You can get Kali Linux from https://www.kali.org/.

Disclaimer: *Kali is a powerful offensive security suite. Use it responsibly. Remember that Georgia Tech has no tolerance for unethical behavior on our computing systems. Minor pranks have landed students in disproportionately hot water, so please use common sense.*

When you get Kali Linux installed, make sure to install the whois command.

```
`$ sudo apt-get install whois`
```

What to submit:

- A screenshot of one security tool included in Kali Linux embedded into your solutions document.
- Explain the utility of the tool you've selected above.

DNS Architecture (4 Points)

This exercise will explore the architecture of the Domain Name System and how it can be exploited by an attacker. Follow the instructions in this exercise and answer the questions in your submission.

- In a DNS packet, what is the Transaction ID and how is it used?
- What OSI layer four protocol does DNS use? Why?

- Draw a diagram showing the full trip for a recursive DNS query for www.cc.gatech.edu. Label each entity involved (i.e., requesting host, local DNS server, root server, TLD server, and authoritative server). Assume that nothing is cached. Note that you do not have to label each entity with its IP address or hostname. Just the labels mentioned in the previous sentence will do.
- Using the terminal of your Kali server, type "whois gatech.edu" Include the output in your submission. Now type "traceroute www.cc.gatech.edu" How could this information be useful to an attacker?

DNS servers act on a first-come, first-serve basis regarding responses to queries. If multiple answers are sent to a computer with an open query, it will accept the first valid response it receives and simply drop any subsequent answers for that query for as long as the answer remains in the cache. Because of this behavior, coupled with your answers to (a) and (b) above, the DNS protocol is vulnerable to an attack called cache poisoning, where a malicious IP is inserted in the [domain : IP] lookup table. For example, a request for "example.com" to a poisoned server would return a malicious IP (perhaps for phishing or for hosting malware) rather than the actual IP for example.com.

- What is the impact to end users if they query a DNS server with a poisoned cache?
- Explain the process for performing a poisoning attack. Be sure to note why it works.
- Now that you understand the weaknesses of the basic Domain Name System, what is one way it could be improved? (Hint: look up DNSSEC, 0x20-bit Encoding, or other techniques.)

What to submit:

- Diagram of a recursive DNS query embedded in the document
- Output file of whois command
- Your answers to the above questions

SSL/TLS and Traffic Analysis (6 Points)

The Internet was designed without explicit protection for confidentiality nor for strong integrity. SSL/TLS was developed later (by Netscape Communications) to provide a reliable end-to-end secure communication service over TCP. In this section you will use the network analysis tool Wireshark to watch SSL/TLS at work when connecting to a secure web site (using HTTPS).

Launch Wireshark and instruct it to listen on the desired network interface (eth0 on your Kali VM). After starting the capture, you may begin to see packets start to appear immediately if there are other processes sending/receiving traffic on your network.

With a running capture, open a web browser and visit an unencrypted (HTTP only) web page of your choice. Do the same thing for an encrypted web page (one that uses HTTPS). Stop the capture and Inspect the contents of the captured packets to answer the following questions.

- What is different when visiting the site using HTTP versus HTTPS? (Protip: To see HTTP traffic, use Wireshark to filter on “http”. To see HTTPS traffic, use Wireshark to filter on “ssl” traffic.)
- Draw a diagram showing the SSL/TLS handshake procedure you observed. Attach a screenshot of the Wireshark capture that shows the first few packets of the handshake.
- Which organization(s) issued the certificate of the web server?
- What is the public RSA key of the web server?
- If the client (your browser) cannot verify the validity of a remote server's certificate and you choose to trust it anyway, what security properties do you lose?

What to submit:

- Diagram of SSL/TLS handshake embedded in the document
- Screenshot of the Wireshark capture embedded in the document
- Your answers to the above questions

nmap Security Scanner (4 points)

Nmap is a powerful tool used for probing network servers. It sends specially crafted packets to target hosts, analyzes the responses, and reports its findings to the user along with inferences about the systems on the network. In this section, you are given two pcap files that contain packets of nmap port scanning traffic. Use Wireshark (or other tools that can view .pcap files) to analyze the network traffic and answer the following questions.

*Note: Answer these questions for **both** of the provided .pcap files on T-Square*

- What is the IP address of the scanner?
- What is the IP address of the target?
- Which ports were scanned? (Hint: how were they chosen?)
- Which port scanning technique was used? Please justify your answer and describe how the technique works.
- What is the command and arguments for this scan? Your answer should be what you would type into the shell, *e.g.* nmap -sT -p443 127.0.0.1.

You will now launch an nmap scan against a virtual machine image. Metasploitable is an Ubuntu system intentionally configured to be vulnerable. We will use this VM in future homeworks, so hang on to it.

Download the Metasploitable image (~800MB in total) on T-Square. It comes in three parts. Use the ‘cat’ command to combine all of the zip files into one zip file. (The files are split up into three parts because T-Square has a file size limit.) The SHA-256 hash of the recombined (final) image is a26ea2d80de1884080913c94eb12cd62c82ee0c8aea889a7841131cef55823ae

Here is some nice documentation on how the rest of this setup will proceed:
http://www.howtohackin.com/blog/kali-linux-virtualbox-pentest-lab/

We won’t do exactly what they’re doing, but it will be similar.

Set up the Metasploitable image:

1. Start VirtualBox
2. Click “New”
3. Enter type as “Linux” and version as “Ubuntu (32-bit)”
4. Memory size: 512MB
5. Hard drive: “use existing hard drive” and select the .vmdk file in the metasploit folder you’ve unzipped

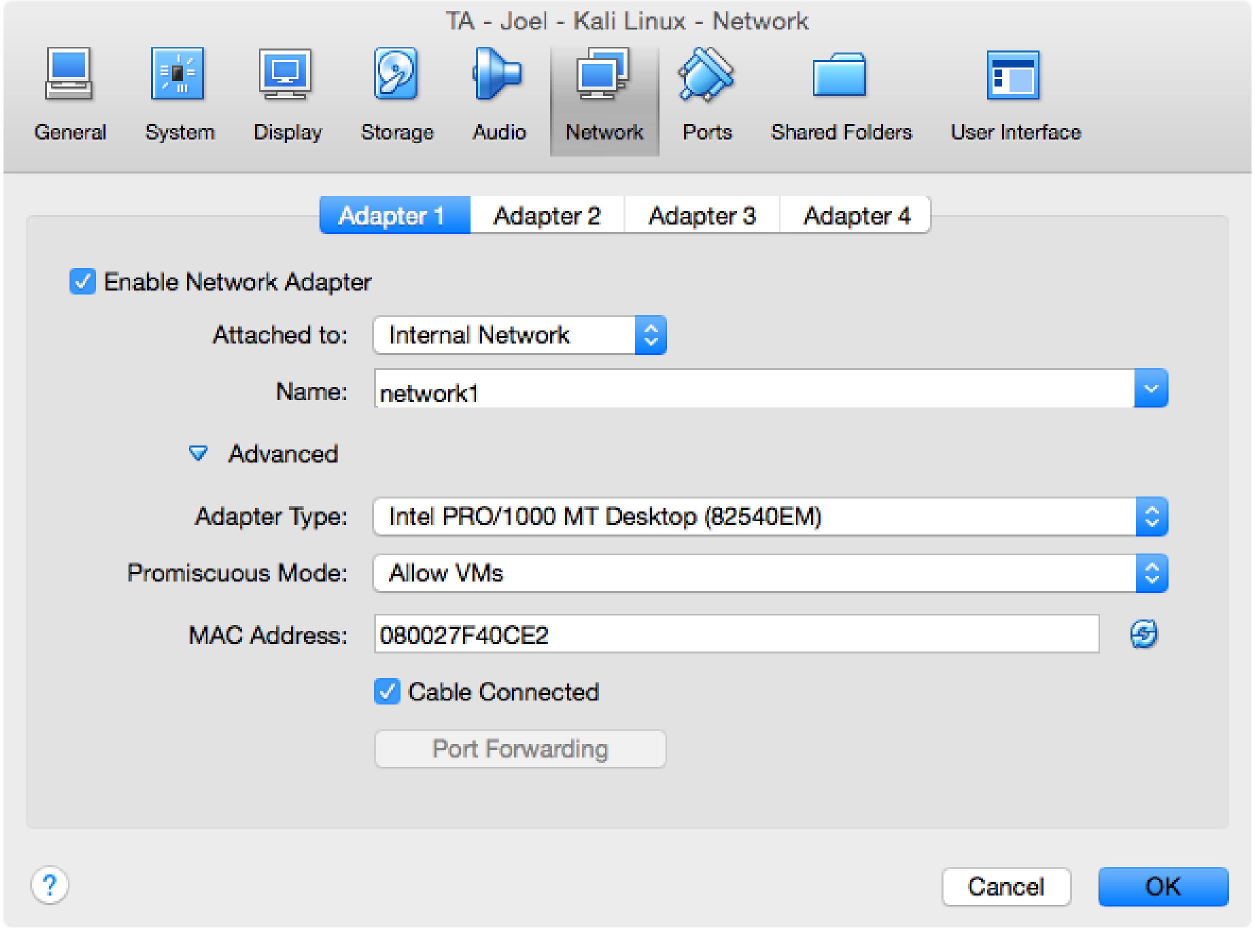
Set up your internal network as such (on your host computer that is running VirtualBox):

```
        $ vboxmanage dhcpserver add --netname network1 --ip 192.168.1.1 --netmask
        255.255.255.0 --lowerip 192.168.1.2 --upperip 192.168.1.254 --enable
```

```
        $ vboxmanage dhcpserver add --netname network2 --ip 192.168.2.1 --netmask
255.255.255.0 --lowerip 192.168.2.2 --upperip 192.168.2.254 --enable
```

Attach Kali Linux and Metasploit to both networks (listed as Internal Networks).

An example for attaching one interface:



☑ Enable Network Adapter

Attached to: Internal Network

Name: network1

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow VMs

MAC Address: 080027F40CE2

☑ Cable Connected

When you start both VMs, you can run `$ ip a` and see that both network interfaces exist in Kali Linux. However, only one exists in Metasploit (it should be 192.168.1.3).
Add the other one in Metasploit by executing (inside of the Metasploit VM):

`$ sudo ifconfig eth1 192.168.2.3 netmask 255.255.255.0 up`

- Submit the output of a full port TCP SYN scan on across the first network (network1) by running the following command on Kali (scanning Metasploit): nmap -p- 192.168.1.3
  - To do this, start each VM (Kali and Metasploit) using the internal network configurations described above.
  - Then use nmap in Kali to target the Metasploit VM's network1 IP address (192.168.1.3).
- Research nmap's different scan techniques. It performs the SYN scan (-sS) by default. Why?
- Choose one other scan technique (i.e. -sF, -sA, etc.) and discuss its advantages and disadvantages.

What to submit:

- Your answers to the above questions

- File containing output of full port scan (separate file from your solutions document)

iptables Firewall (4 Points)

A common firewall utility used on Linux systems is iptables. In this section, you will configure the iptables firewall on your Metasploitable VM and test your settings with Kali Linux. From the previous section, you have already set up two internal network interfaces. The first interface will be associated with 192.168.1.0/24, and the second with 192.168.2.0/24. Ensure that both networks are still active with the ping command, e.g. ping [IP_ADDR] . Both IPs should respond if the interfaces are configured correctly.

At this point, you should be able to SSH from Kali into Metaspoitable via both networks (the commands `$ ssh msfadmin@192.168.1.3` and `ssh msfadmin@192.168.2.3` should both work). This suggests that by default SSH connections are accepted from all sources. Your task is to configure iptables inside Metasploit with the following settings:

- Only allow SSH connections from the second (192.168.2.0/24) network
- Drop all other connections
- Change the default policy of the FORWARD chain to DROP.

Hint: Here is some good iptables material to read, as iptables can be difficult to read/understand and even more difficult to write properly.

- https://en.wikipedia.org/wiki/Iptables
- https://www.centos.org/docs/5/html/Deployment_Guide-en-US/ch-iptables.html
- https://wiki.debian.org/iptables
- http://www.howtogeek.com/177621/the-beginners-guide-to-iptables-the-linux-firewall/

Hint: Some more helpful links

- http://gr8idea.info/os/tutorials/security/iptables4.html
- http://unix.stackexchange.com/questions/136190/iptables-rule-to-allow-incoming-ssh-connections
- http://serverfault.com/questions/161401/how-to-allow-a-range-of-ips-with-iptables
- http://askubuntu.com/questions/115940/how-can-i-setup-ssh-so-that-it-is-restricted-to-my-local-network
- http://www.cyberciti.biz/tips/linux-iptables-4-block-all-incoming-traffic-but-allow-ssh.html

As some food-for-thought, you can think of Metasploitable as being a private server inside of a company. Even though other computers (i.e., Kali Linux) exist on the same network (192.168.1.0/24), that does not mean they have permission to talk to other computers on the network (i.e., Metasploitable).

- Provide screenshots of the command `$ ip a` inside of both Kali Linux and Metasploitable to show that the network interfaces are configured correctly.
- Provide a screenshot of your final iptables configuration in Metasploitable with the following command: `$ sudo iptables -L`

Run a basic nmap scan against Metasploitable using both networks (i.e., scan both 192.168.1.3 and 192.168.2.3). The first should return no open ports (they should be all filtered), and the second should return port 22 (SSH) as being open.

- Is it possible to configure a firewall to block nmap scans, but allow legitimate traffic? Why or why not?
- iptables can be configured to either REJECT or DROP unwanted packets.   DROP simply ignores the packet. What does REJECT do? Can you think of any circumstances where it might be better to use REJECT?
- Is a firewall alone sufficient to protect a network? Why or why not?

What to submit:

- Three screenshots as described above embedded into your solutions document.
- Answers to above questions.