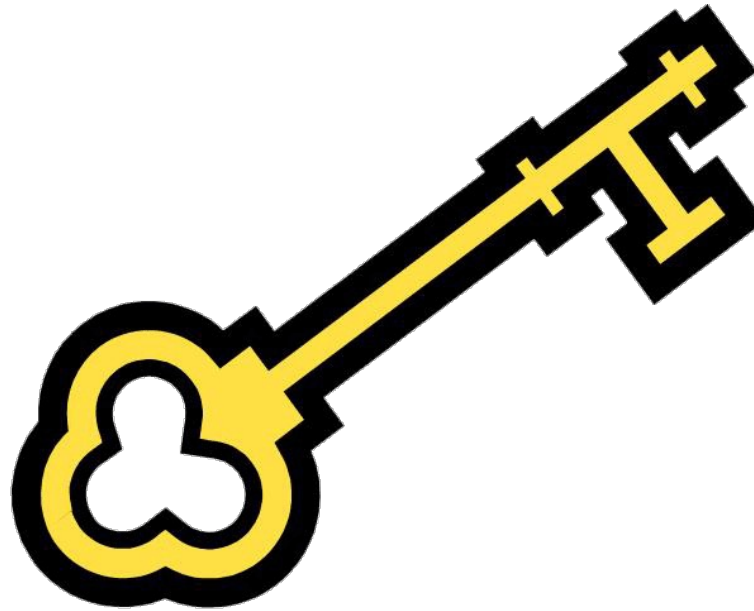


Packet Sniffing Lab



Purpose



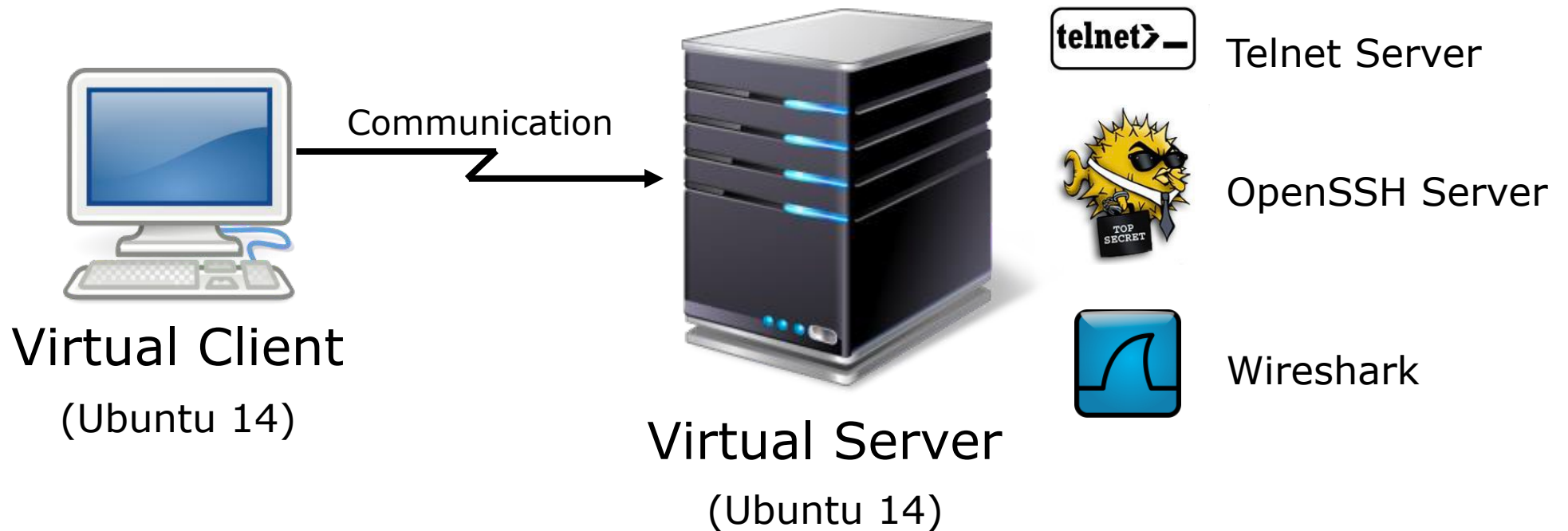
- Understanding network packets and protocols
- Using Wireshark to capture and analyze network packets



Virtual Lab Setup



- Two VM instances within the same subnet will be used



Lab Structure



- The *Packet Sniffing* lab consists of 5 experiments
 1. Telnet
 2. SSH
 3. TCP Traffic Analysis
 4. UDP Traffic Analysis



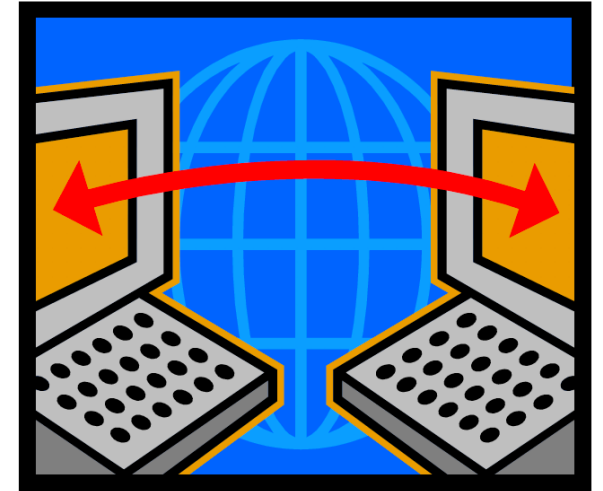
Lab Software Tools



AskCyPert

For this lab we will be using the following tools:

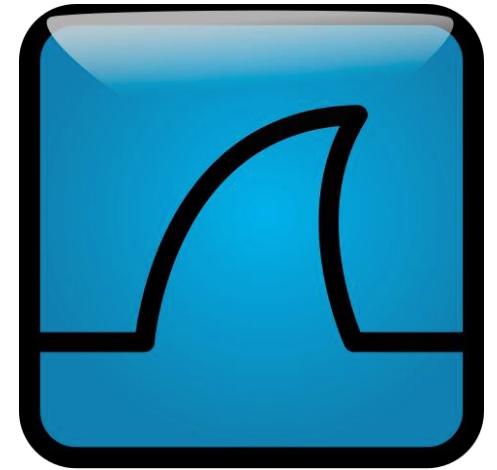
- **Telnet Client/Server:** The client establishes a remote connection to the server using Telnet protocol.
- **SSH Client and Server:** The SSH client establishes a remote connection to the SSH server using the *Secure Shell (SSH)* protocol.



Lab Software Tools



- **Wireshark:** Analyzes all packets of a network interface
 - We will use Wireshark to analyze all communications between the client and the server VMs
 - Wireshark will obtain all transferred information, including client's username and password
 - Confidentiality property of cybersecurity will be violated



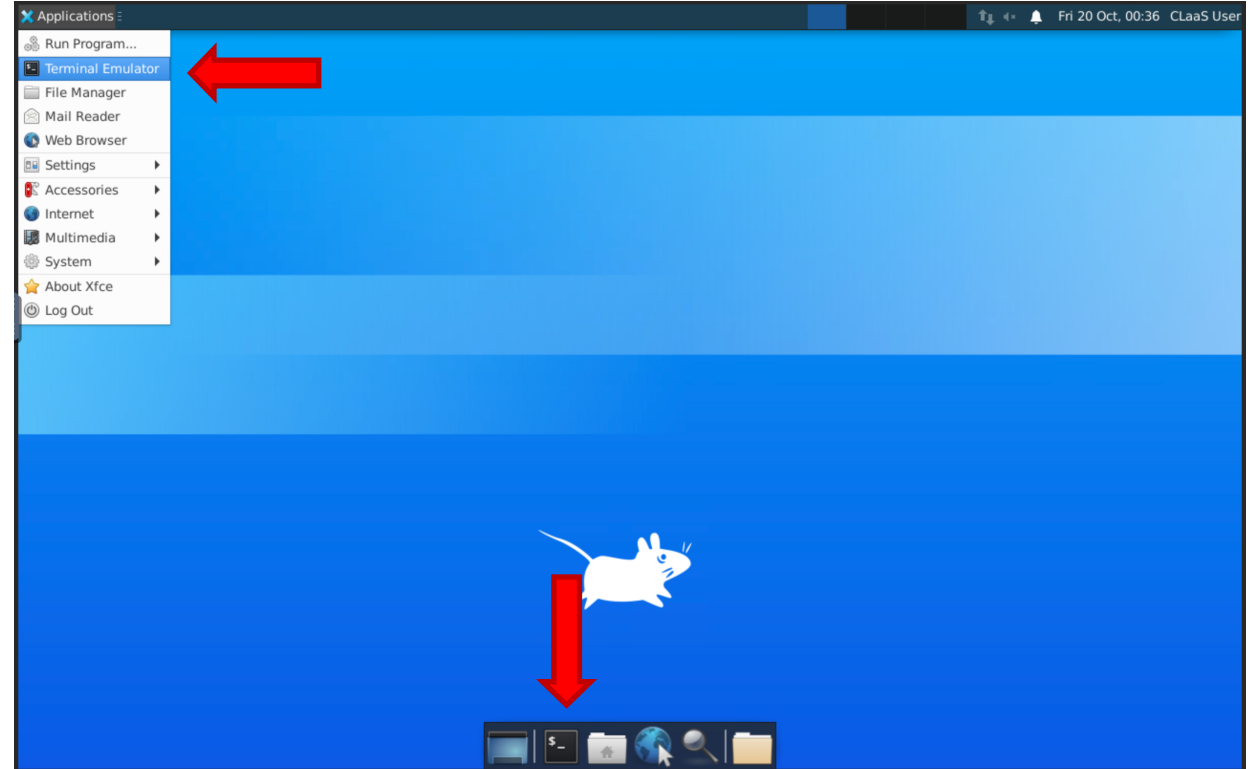


Experiment 0: Getting ready to start

Experiment Instructions



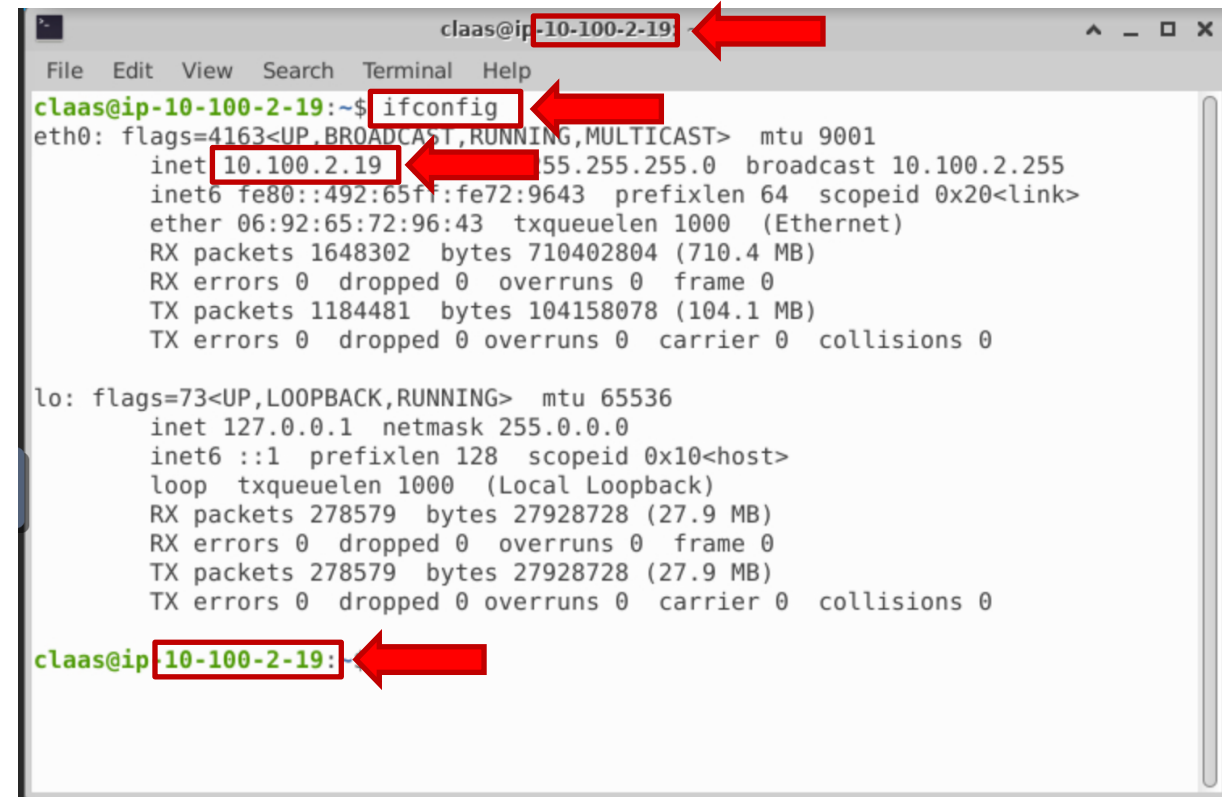
- Step 1: Open a terminal
 - For this lab we will be using the terminal.
 - There are two ways to open it.
 - From the menu
 - Or by clicking the icon on the dock:



Experiment Instructions



- Step 2: Get the IP addresses
 - We can see the IP address as soon as we open the terminal
 - Or by typing
`ifconfig`
 - *You will need the IP of the two VMs, for the rest of the lab.*



```
claas@ip-10-100-2-19
File Edit View Search Terminal Help
claas@ip-10-100-2-19:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 10.100.2.19 netmask 255.255.255.0 broadcast 10.100.2.255
    inet6 fe80::492:65ff:fe72:9643 prefixlen 64 scopeid 0x20<link>
    ether 06:92:65:72:96:43 txqueuelen 1000 (Ethernet)
    RX packets 1648302 bytes 710402804 (710.4 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 1184481 bytes 104158078 (104.1 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 278579 bytes 27928728 (27.9 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 278579 bytes 27928728 (27.9 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

claas@ip-10-100-2-19:
```




Experiment 1: Telnet

Experiment Instructions



- Step 1: Open Wireshark
 - On the Server VM
 - Open a terminal and type

wireshark



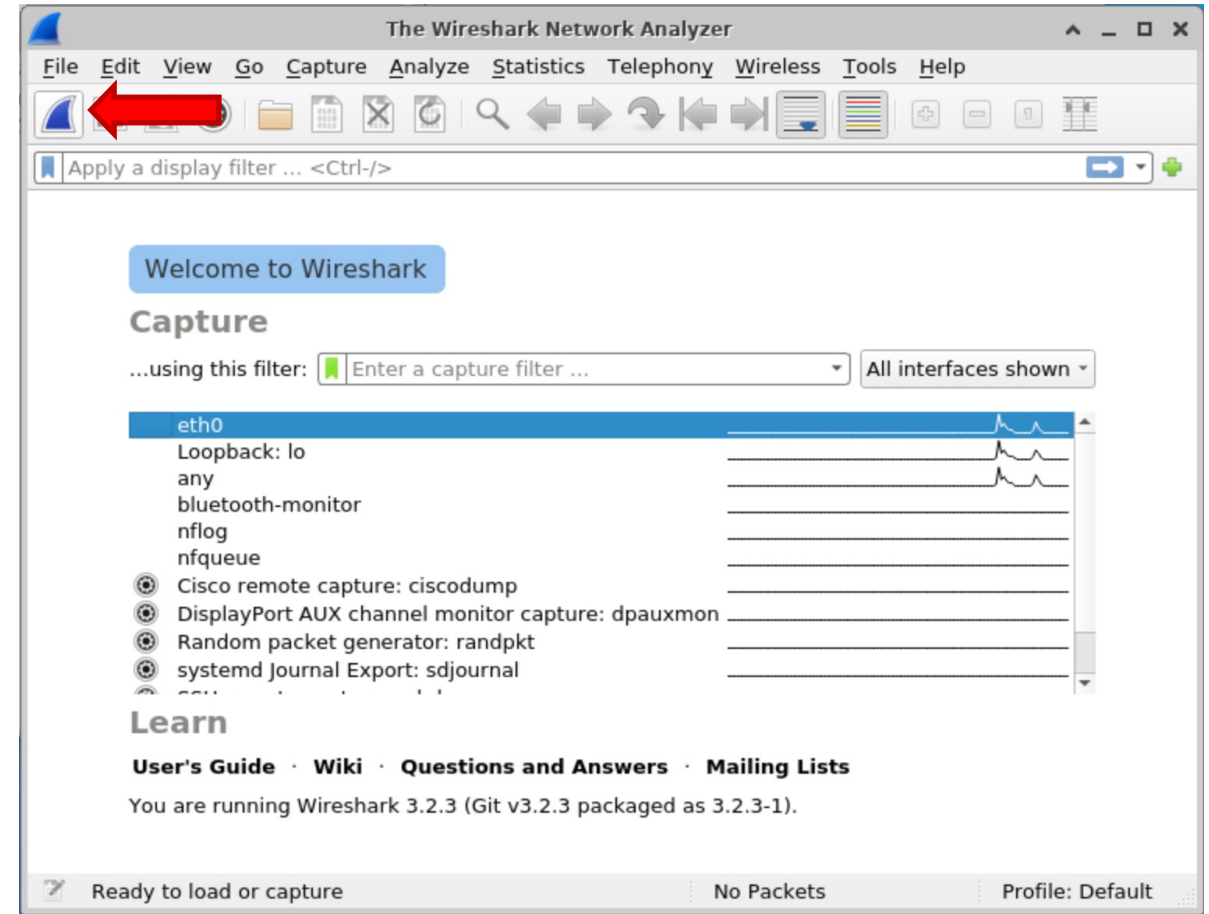
```
claas@ip-10-100-2-19: ~  
File Edit View Search Terminal Help  
claas@ip-10-100-2-19:~$ wireshark
```



Experiment Instructions



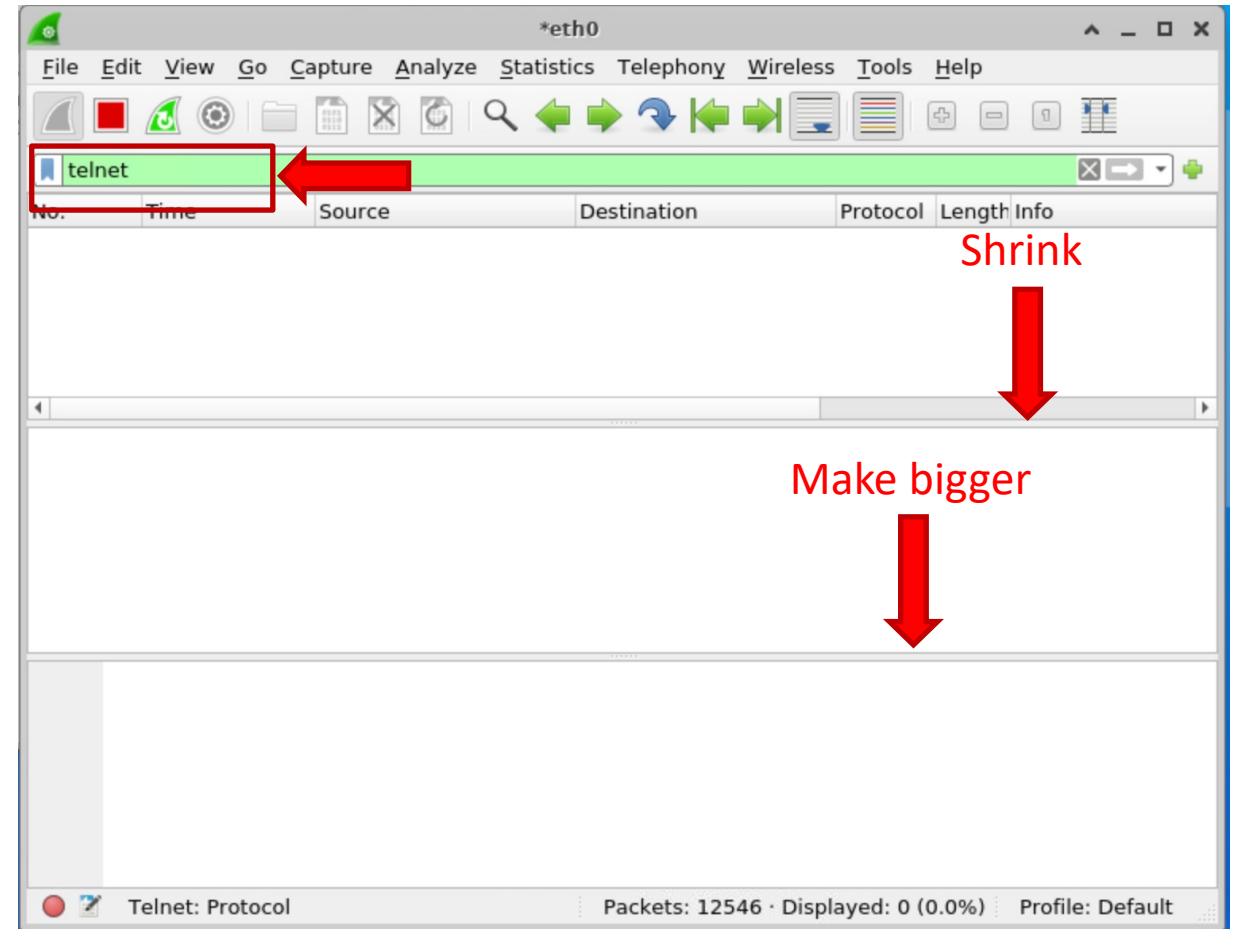
- Step 2: Prepare Wireshark
 - Once Wireshark is open select the *eth0* interface
 - Click the blue button to *Start capturing packets*



Experiment Instructions



- Step 3: Filter Telnet packets
 - In the filter field, type *telnet* to filter only telnet packets
 - Adjust the windows, this way we'll be able to see the packets better



Experiment Instructions

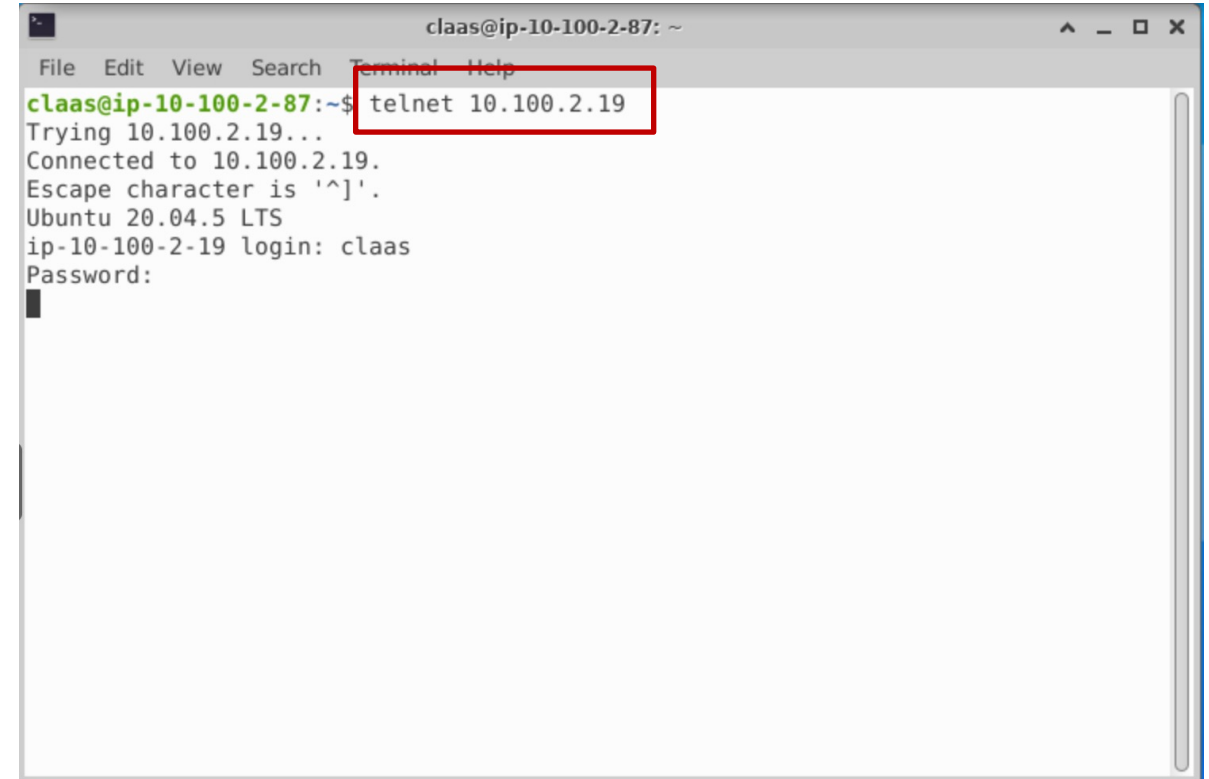


- Step 4: Generate Telnet packets

- Go back to the Client VM
- Open a terminal, and type

```
telnet <serverIPAddress>
```

- Once connected type:
 - Username: *claas*
 - Password: *Claas2022*



```
claas@ip-10-100-2-87: ~  
File Edit View Search Terminal Help  
claas@ip-10-100-2-87:~$ telnet 10.100.2.19  
Trying 10.100.2.19...  
Connected to 10.100.2.19.  
Escape character is '^]'.  
Ubuntu 20.04.5 LTS  
ip-10-100-2-19 login: claas  
Password:  
█
```



Command Explained

Start a Telnet communication

telnet <serverIPAddress>

With this computer
IP address of the server

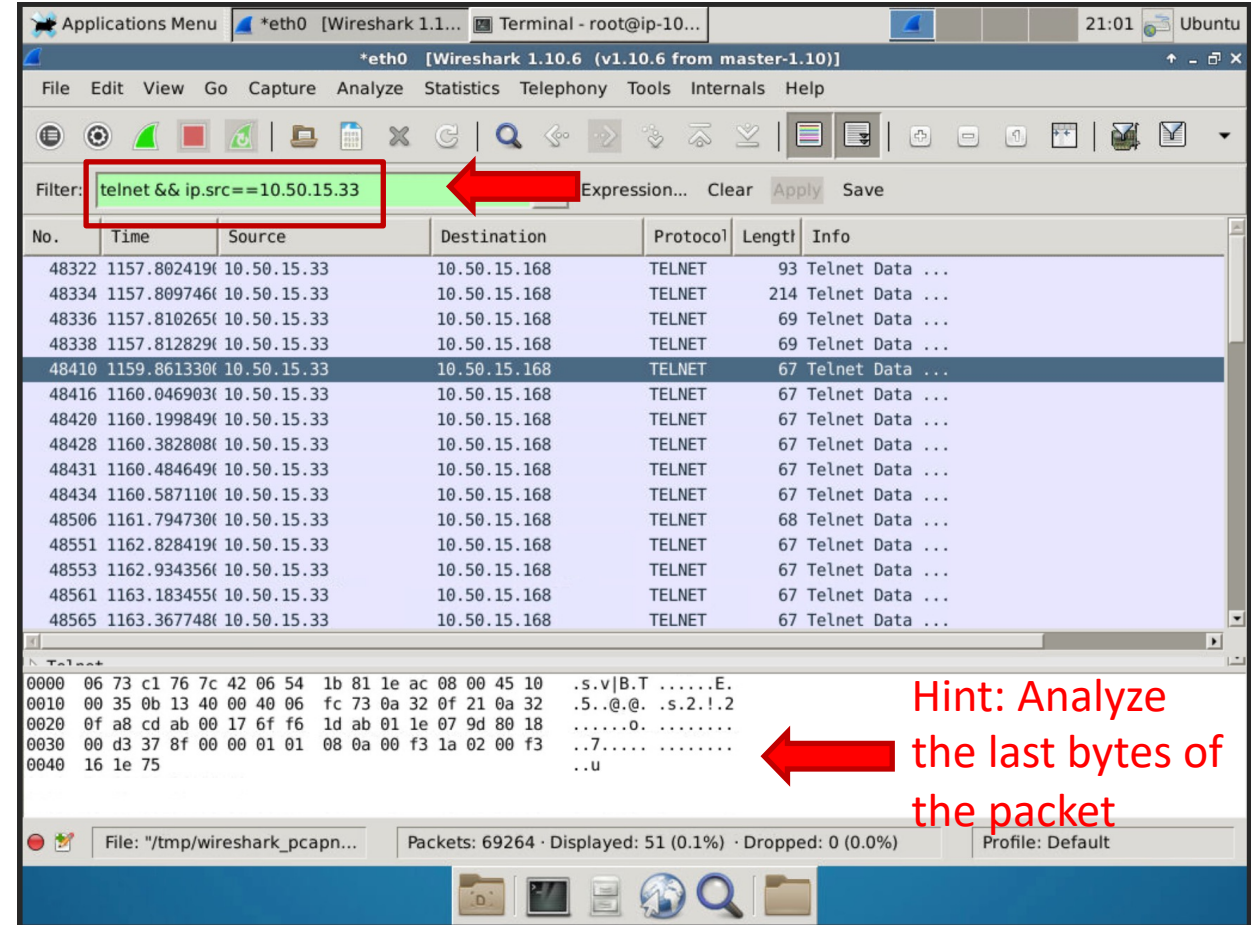
- Exit your Telnet session by pressing **Ctrl+C**
- Or type **exit**

Experiment Instructions



- Step 5: Analyze Telnet packets
 - Go back to Wireshark in the Server VM
 - Analyze the packets captured from the Telnet session
 - Hint: You might want to filter also by IP, to do this type on the filter field:

`telnet && ip.src==<clientIPaddress>`



Applications Menu *eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)] Terminal - root@ip-10... 21:01 Ubuntu

*eth0 [Wireshark 1.10.6 (v1.10.6 from master-1.10)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: `telnet && ip.src==10.50.15.33` Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
48322	1157.802419	10.50.15.33	10.50.15.168	TELNET	93	Telnet Data ...
48334	1157.809746	10.50.15.33	10.50.15.168	TELNET	214	Telnet Data ...
48336	1157.810265	10.50.15.33	10.50.15.168	TELNET	69	Telnet Data ...
48338	1157.812829	10.50.15.33	10.50.15.168	TELNET	69	Telnet Data ...
48410	1159.861330	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48416	1160.046903	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48420	1160.199849	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48428	1160.382808	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48431	1160.484649	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48434	1160.587110	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48506	1161.794730	10.50.15.33	10.50.15.168	TELNET	68	Telnet Data ...
48551	1162.828419	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48553	1162.934356	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48561	1163.183455	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...
48565	1163.367748	10.50.15.33	10.50.15.168	TELNET	67	Telnet Data ...

Telnet

```

0000  06 73 c1 76 7c 42 06 54 1b 81 1e ac 08 00 45 10  .s.v|B.T .....E.
0010  00 35 0b 13 40 00 40 06 fc 73 0a 32 0f 21 0a 32  .5..@.@. .s.2.!2
0020  0f a8 cd ab 00 17 6f f6 1d ab 01 1e 07 9d 80 18  .....0. ....
0030  00 d3 37 8f 00 00 01 01 08 0a 00 f3 1a 02 00 f3  ..7.....
0040  16 1e 75
    
```

File: "/tmp/wireshark_pcapn... Packets: 69264 · Displayed: 51 (0.1%) · Dropped: 0 (0.0%) Profile: Default

Hint: Analyze the last bytes of the packet

Experiment Summary



- **Please note**

- Telnet sends all packets (including username and password) in plaintext
- An attacker would have no difficulty reading it using Wireshark
- After retrieving the credentials, the attacker can connect to the server and have the same access privileges as the client



Lab Report



- Analyze the Telnet packets captured by Wireshark and describe them
- Describe the steps and their results in the report
- Obtain the username and password and some commands sent from the client, include them in your lab write-up, and explain how you have obtained them from Wireshark



How do you prevent such attacks?



- This security issue can be remedied if a communication protocol with encryption is used to transfer information between client and server machines.
- Unlike Telnet, the SSH protocol does not transmit data in plaintext. It encrypts transmitted data so that, even if attackers intercept the data, it is incomprehensible and thus still secure.





Experiment 2: SSH

Experiment Instructions



- For this experiment, you will follow the same steps as in the Telnet experiment, with these differences:
 - In step 3 we will filter SSH instead of Telnet packets
 - For step 4, on the terminal you'll type:

```
ssh claas@<serverIPAddress>
```

- After this, it will ask you if you want to establish a connection, type *yes*, and type the password *Claas2022*
- Proceed with the next steps



Command Explained

Start an SSH
communication

At

Separates the username
from the IP address

```
ssh claas@<serverIPAddress>
```

Username to
login with

With this computer
IP address of the server

- Exit your SSH session by pressing **Ctrl+C**
- Or type **exit**

Experiment Summary



- **Please note**
 - SSH uses encrypted packets
 - An attacker will fail to understand the contents without decrypting it



Lab Report



- Analyze the SSH packets captured by Wireshark and describe them

- Describe the steps and their results in the report



- Can you obtain a username and password, or any useful information sent via SSH? Justify your answer



Other features of Wireshark



- With Wireshark, you can also analyze the packet structure
- For the next experiments, you won't only view the contents of the packet, but also the multiple layers of encapsulation that go into creating a packet





Experiment 3: TCP Traffic Analysis



Experiment Instructions



- Step 1: Connect to a webpage
 - On the Server VM, you will use Elinks (a web-based browser)
 - Connect to a webpage typing

```
elinks www.google.com
```
 - Exit the terminal-based website by typing **Ctrl+C**



Experiment Instructions



- Step 2: Observe the TCP packets on Wireshark.
 - Go back to Wireshark and type on the filter field:

tcp

- Go through the packets in the list and observe the following:
 - Source and destination IPs
 - Source and destination port numbers on both the client and server
 - The packet parameters
 - Maximum and minimum values of these parameters



Experiment Instructions



- Step 3: Change the display filter to *HTTP*
 - Go through the packets in the list and add the following information to your lab report:
 - Describe the HTTP commands and their parameters
 - What are the maximum and minimum values of these parameters?
 - What are the listed HTTP headers and what are their values? (Some HTTP packets may not have headers)



Experiment Summary



- **Please note**

- The TCP protocol includes error checking and packet ordering information
- HTTP is a TCP protocol



Lab Report



- List source and destination IPs and port numbers from the captured TCP packets
- List TCP packet parameters and maximum/minimum values
- Describe HTTP commands and their parameters (include max/min values)
- List HTTP headers and their values





Experiment 4: UDP Traffic Analysis

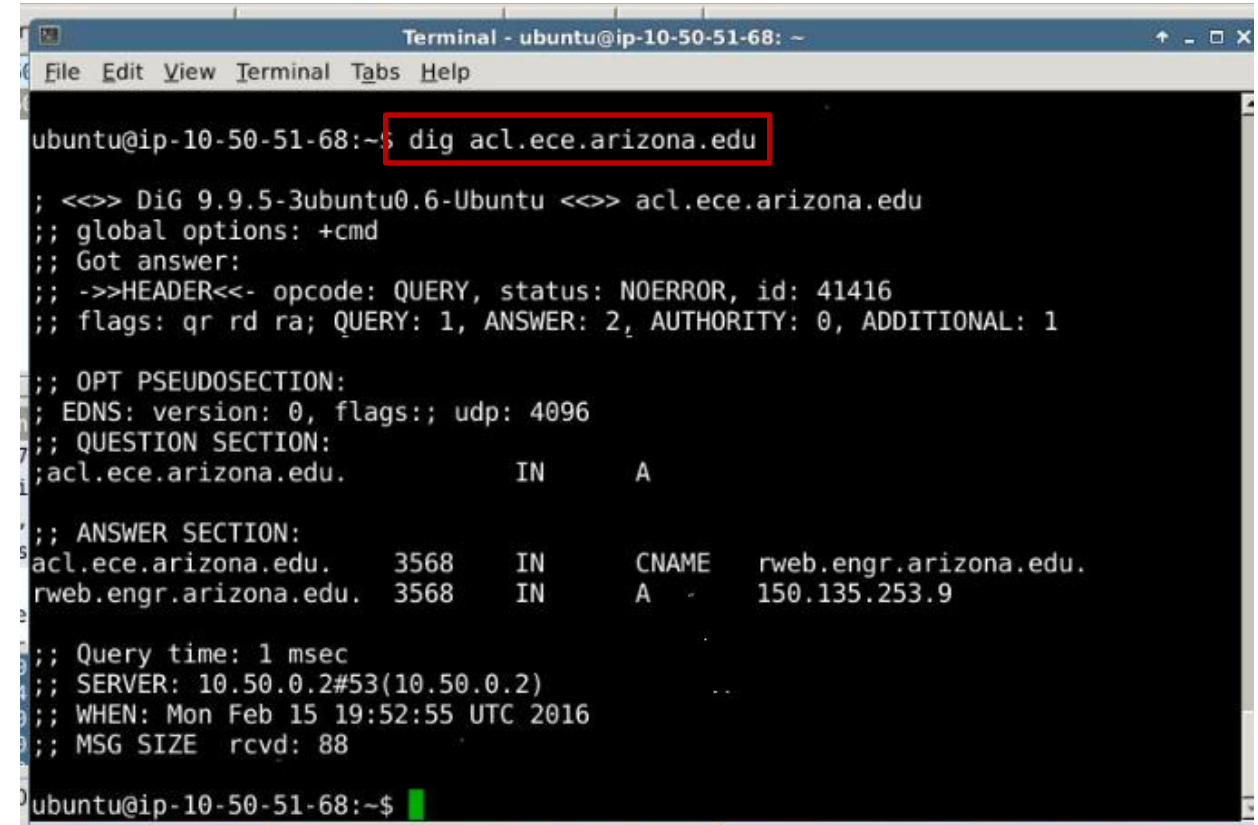


Experiment Instructions



- Step 1: Generate DNS traffic
 - From the terminal on the Server VM, type the following to generate DNS traffic:

```
dig acl.ece.arizona.edu
```



```
Terminal - ubuntu@ip-10-50-51-68: ~
File Edit View Terminal Tabs Help

ubuntu@ip-10-50-51-68:~$ dig acl.ece.arizona.edu

; <<>> DiG 9.9.5-3ubuntu0.6-Ubuntu <<>> acl.ece.arizona.edu
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 41416
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;; udp: 4096
;; QUESTION SECTION:
;acl.ece.arizona.edu.      IN      A
;
;; ANSWER SECTION:
acl.ece.arizona.edu.      3568    IN      CNAME   rweb.engr.arizona.edu.
rweb.engr.arizona.edu.    3568    IN      A       150.135.253.9

;; Query time: 1 msec
;; SERVER: 10.50.0.2#53(10.50.0.2)
;; WHEN: Mon Feb 15 19:52:55 UTC 2016
;; MSG SIZE rcvd: 88

ubuntu@ip-10-50-51-68:~$
```

Experiment Instructions



- Step 2: Observe the UDP packets on Wireshark.
 - Go back to Wireshark and type on the filter field:

udp

- Go through the packets in the list and observe the following:
 - Source and destination IPs
 - Source and destination port numbers on both the client and server
 - The packet parameters
 - Maximum and minimum values of these parameters



Experiment Instructions



- Step 3: Observe the DNS packets on Wireshark.
 - Go back to Wireshark and type on the filter field:

dns

- Examine the listed packets and find the following information:
 - DNS headers
 - Maximum and minimum values of these headers
 - The listed queries and answers



Experiment Instructions



- Step 4: Generate more DNS packets
 - Look up at least four domain names with the dig command on a terminal in the Server VM
 - Go back to Wireshark and examine the DNS packets you created. Add the following to your lab report:
 - What information does a DNS request packet contain?
 - What information does a DNS reply packet contain?
 - Which resource is handling each DNS request for each domain name?
 - How many steps are required to obtain an IP address for each request?



Experiment Summary



- **Please note**

- The UDP protocol is connectionless
- DNS is a UDP protocol
- The DNS protocol can have a variety of data sources, including authoritative name servers and resolvers



Lab Report



- List source and destination IPs and port numbers from the captured UDP packets
- List UDP packet parameters and maximum/minimum values
- List DNS headers and their values (include max/min values)
- List DNS queries and answers from captured packets
- What information does a DNS request packet contain?
- What information does a DNS reply packet contain?
- Which resource is handling each DNS request for each domain name?
- How many steps are required to obtain an IP address for each request?



Conclusion



- Wireshark is a very powerful tool that can have a variety of uses in computer and network security.
- In this lab, we barely scratched the surface of Wireshark's capabilities as a packet analyzer.
- We also experimented with different protocols and their structures.





End of Lab

