

Computer Security Capstone

Project II: MITM and Pharming Attacks in Wi-Fi Networks

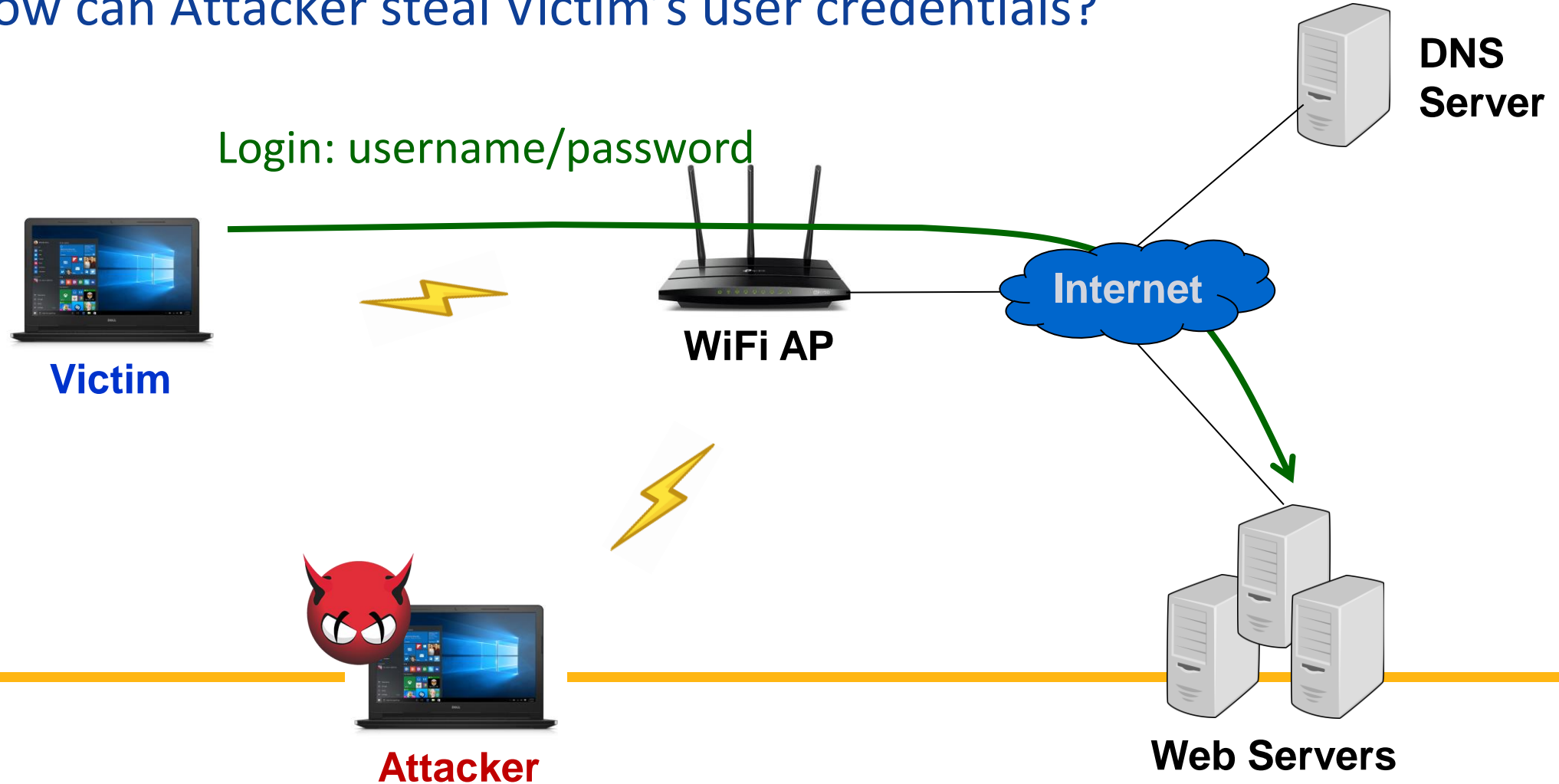
Chi-Yu Li (2022 Spring)
Computer Science Department
National Yang Ming Chiao Tung University

Goal

- Understand how user credentials can be leaked by a man-in-the-middle (MITM) attack over Wi-Fi networks
- You will learn how to
 - ❑ scan IP/MAC addresses of the devices in a Wi-Fi network
 - ❑ launch an ARP spoofing attack
 - ❑ launch a man-in-the-middle attack
 - ❑ launch a pharming attack

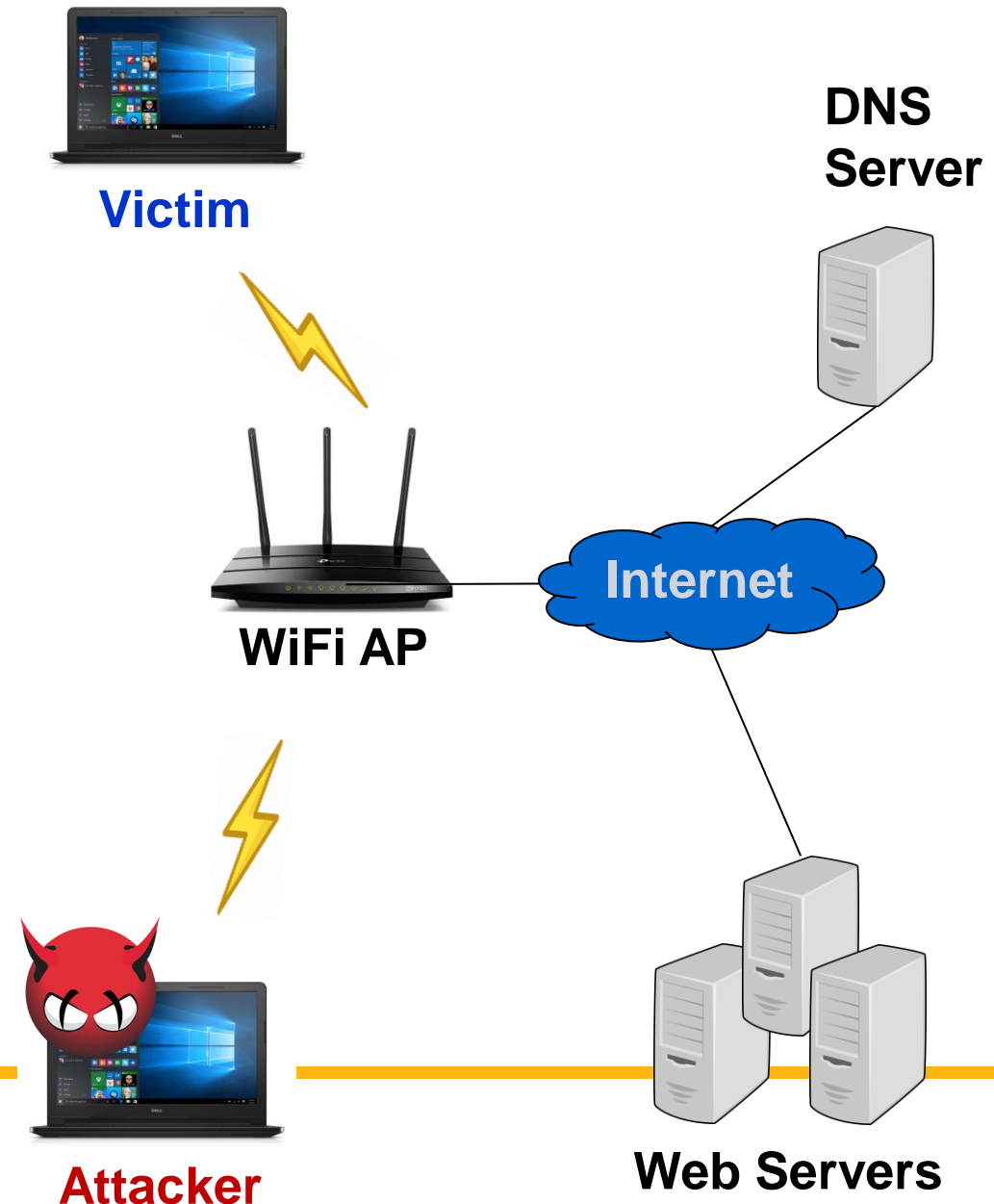
Attack Scenario

- How can Attacker steal Victim's user credentials?



Major Ideas

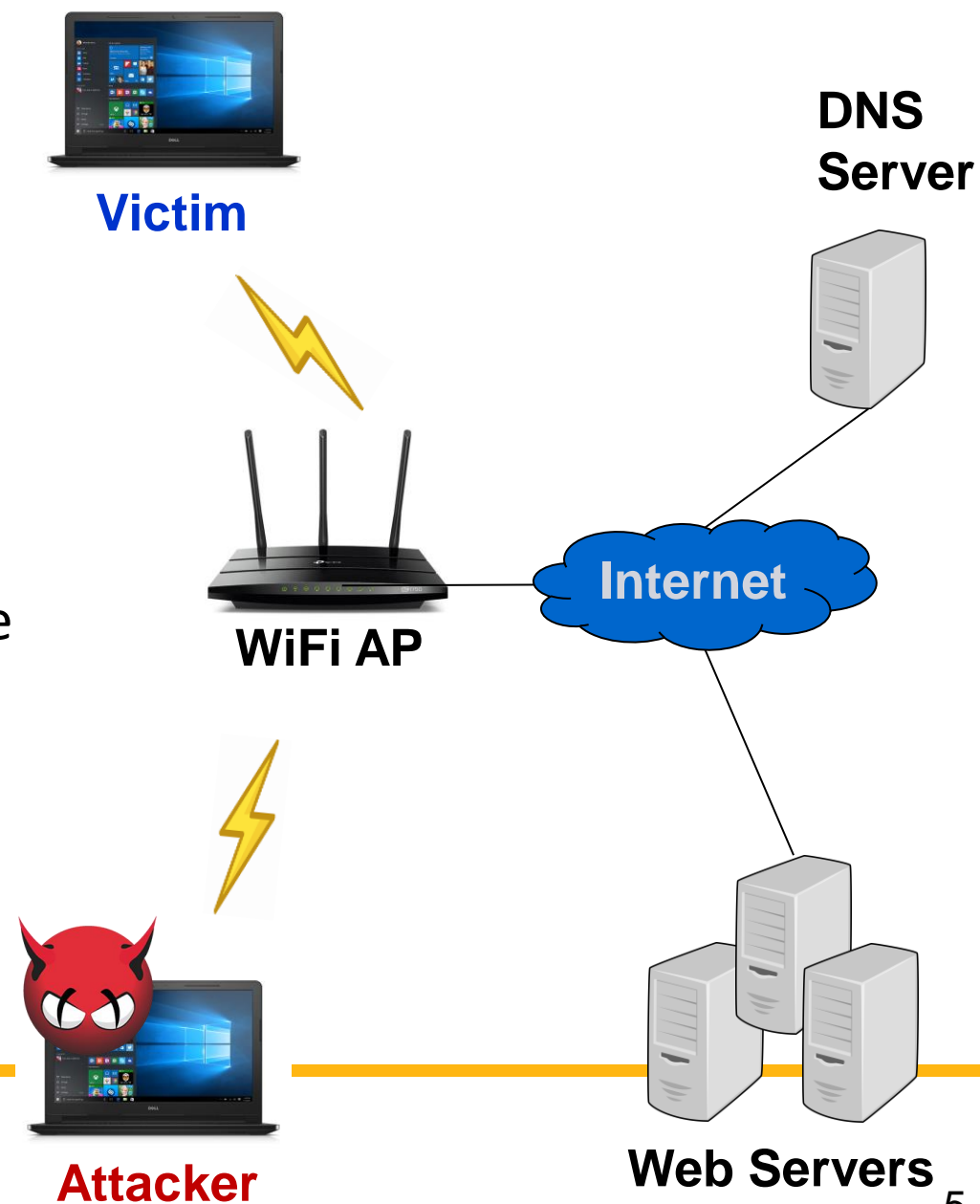
- Redirect Victim's traffic to Attacker
 - ❑ Man-in-the-middle based on ARP spoofing
 - ❑ How to know Victim's IP/MAC address?
- How about encrypted sessions?
 - ❑ MITM attack: split the encrypted sessions
 - ❑ Pharming attack: redirect HTTP requests to a phishing web page



Tasks: MITM and Pharming

● MITM Attack (50%)

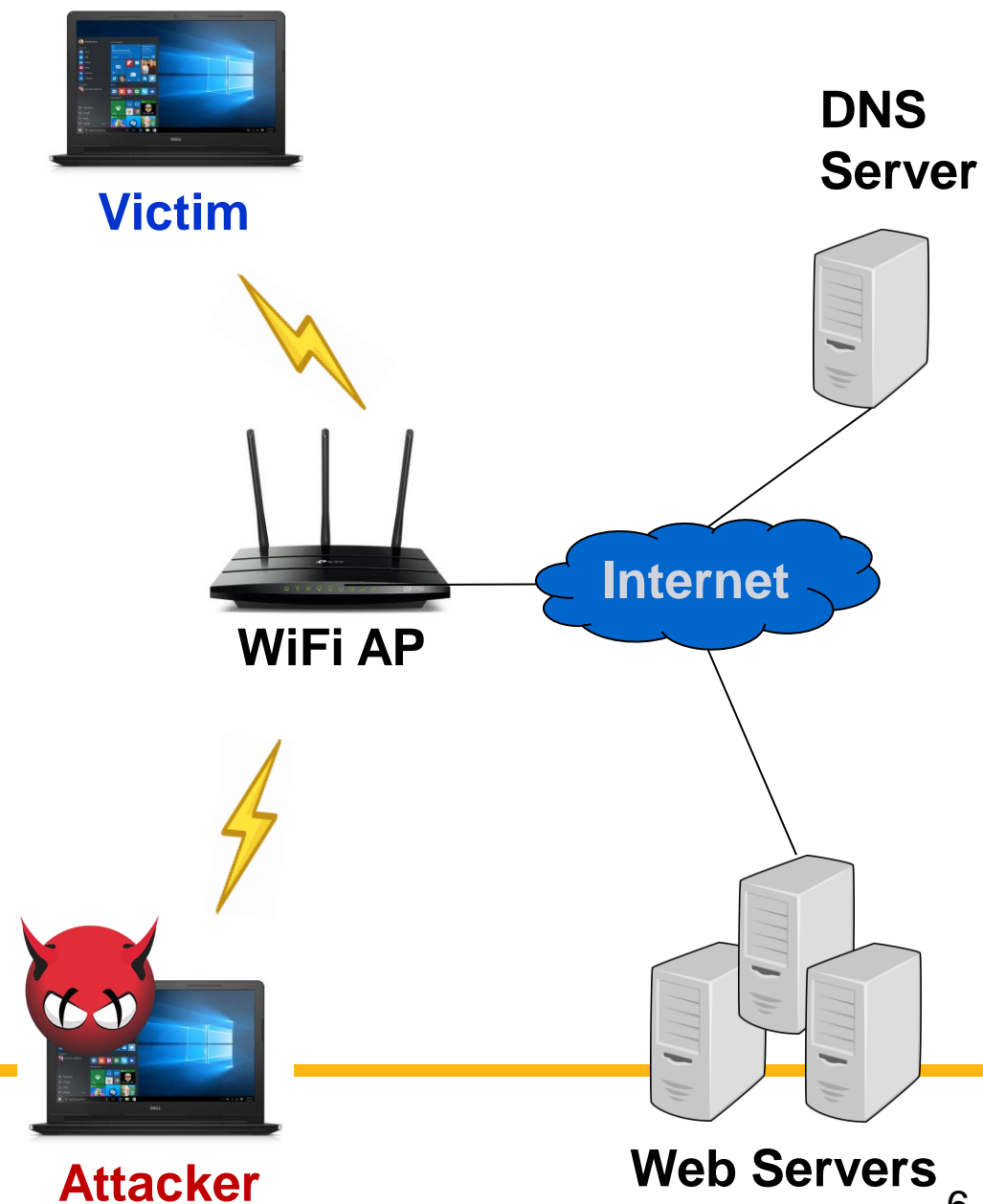
- ❑ Obtain all other client devices' IP/MAC addresses in a connected Wi-Fi network (Task I: 20%)
- ❑ ARP spoofing for all other client devices in the Wi-Fi network (Task II: 15%)
- ❑ Split SSL/TLS encrypted sessions and get the inputted username/password strings from HTTPS sessions (Task III: 15%)



Tasks: MITM and Pharming

● Pharming Attack (30%)

- ❑ Obtain all other client devices' IP/MAC addresses in a connected Wi-Fi network
- ❑ DNS spoofing attack for web services (Task IV: 30%)



Task I: Device Address Information Collection

- Scan all the devices' IP/MAC addresses in the Wi-Fi network
 - ▣ You can use 'scapy' and 'netifaces' library in Python or commands 'nmap', 'arp', and 'route'

```
cs2021@ubuntu:~/Desktop/project2$ sudo ./mitm_attack
Available devices
-----
IP                MAC
-----
172.16.186.1      00:50:56:c0:00:08
172.16.186.141    00:0c:29:f2:d2:ab
172.16.186.254    00:50:56:ed:bd:5e
```

- Fetch the IP/MAC addresses of all the other client devices

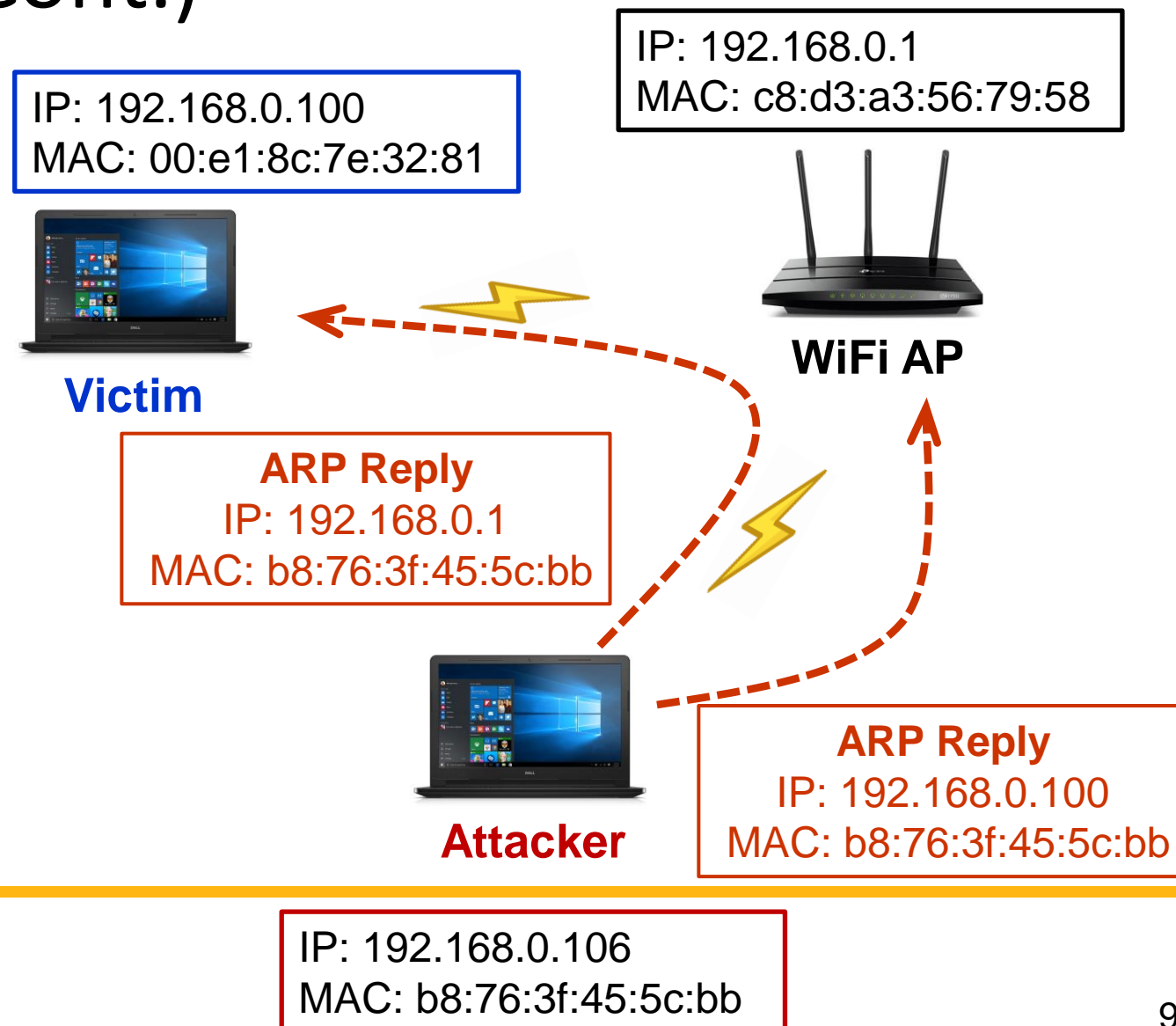
Task II: ARP Spoofing

- What is ARP (Address Translation Protocol)?

- ❑ A communication protocol: discovering the link layer (or MAC) address associated with a given IP
- ❑ A request-response protocol: messages are encapsulated by a link-layer protocol
 - ARP request: broadcast
 - ARP response: unicast
- ❑ Never routed across internetworking nodes

Task II: ARP Spoofing (Cont.)

- Generate spoofed ARP replies for all other client devices
 - ❑ You can use 'scapy' library in Python
- Both uplink and downlink should be considered
 - ❑ Other client devices' network services can work normally



Task II: ARP Spoofing (Cont.)

- An example trace of the successful ARP spoofing at Attacker

No.	Time	Source	Destination	Protocol	Length	Info
2743	152.022717290	192.168.0.100	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=327/18177, ttl=128
2744	152.025483284	192.168.0.100	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=327/18177, ttl=128
2745	152.049717459	8.8.8.8	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=327/18177, ttl=121
2746	152.053411633	8.8.8.8	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=327/18177, ttl=121

▶ Frame 2743: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 ▶ Ethernet II, Src: IntelCor_7e:32:81 (00:e1:8c:7e:32:81), Dst: HonHaiPr_45:5c:bb (b8:76:3f:45:5c:bb)
 ▶ Internet Protocol version 4, Src: 192.168.0.100, Dst: 8.8.8.8
 ▶ Internet Control Message Protocol

Victim → Attacker

No.	Time	Source	Destination	Protocol	Length	Info
2743	152.022717290	192.168.0.100	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=327/18177, ttl=128
→ 2744	152.025483284	192.168.0.100	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=327/18177, ttl=128
← 2745	152.049717459	8.8.8.8	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=327/18177, ttl=121
2746	152.053411633	8.8.8.8	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=327/18177, ttl=121

▶ Frame 2744: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 ▶ Ethernet II, Src: HonHaiPr_45:5c:bb (b8:76:3f:45:5c:bb), Dst: D-LinkIn_56:79:58 (c8:d3:a3:56:79:58)
 ▶ Internet Protocol version 4, Src: 192.168.0.100, Dst: 8.8.8.8

Attacker → AP

No.	Time	Source	Destination	Protocol	Length	Info
2743	152.022717290	192.168.0.100	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=327/18177, ttl=128
→ 2744	152.025483284	192.168.0.100	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=327/18177, ttl=128
← 2745	152.049717459	8.8.8.8	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=327/18177, ttl=121
2746	152.053411633	8.8.8.8	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=327/18177, ttl=121

▶ Frame 2745: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 ▶ Ethernet II, Src: D-LinkIn_56:79:58 (c8:d3:a3:56:79:58), Dst: HonHaiPr_45:5c:bb (b8:76:3f:45:5c:bb)
 ▶ Internet Protocol version 4, Src: 8.8.8.8, Dst: 192.168.0.100
 ▶ Internet Control Message Protocol

AP → Attacker

No.	Time	Source	Destination	Protocol	Length	Info
2743	152.022717290	192.168.0.100	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=327/18177, ttl=128
2744	152.025483284	192.168.0.100	8.8.8.8	ICMP	74	Echo (ping) request id=0x0001, seq=327/18177, ttl=128
2745	152.049717459	8.8.8.8	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=327/18177, ttl=121
2746	152.053411633	8.8.8.8	192.168.0.100	ICMP	74	Echo (ping) reply id=0x0001, seq=327/18177, ttl=121

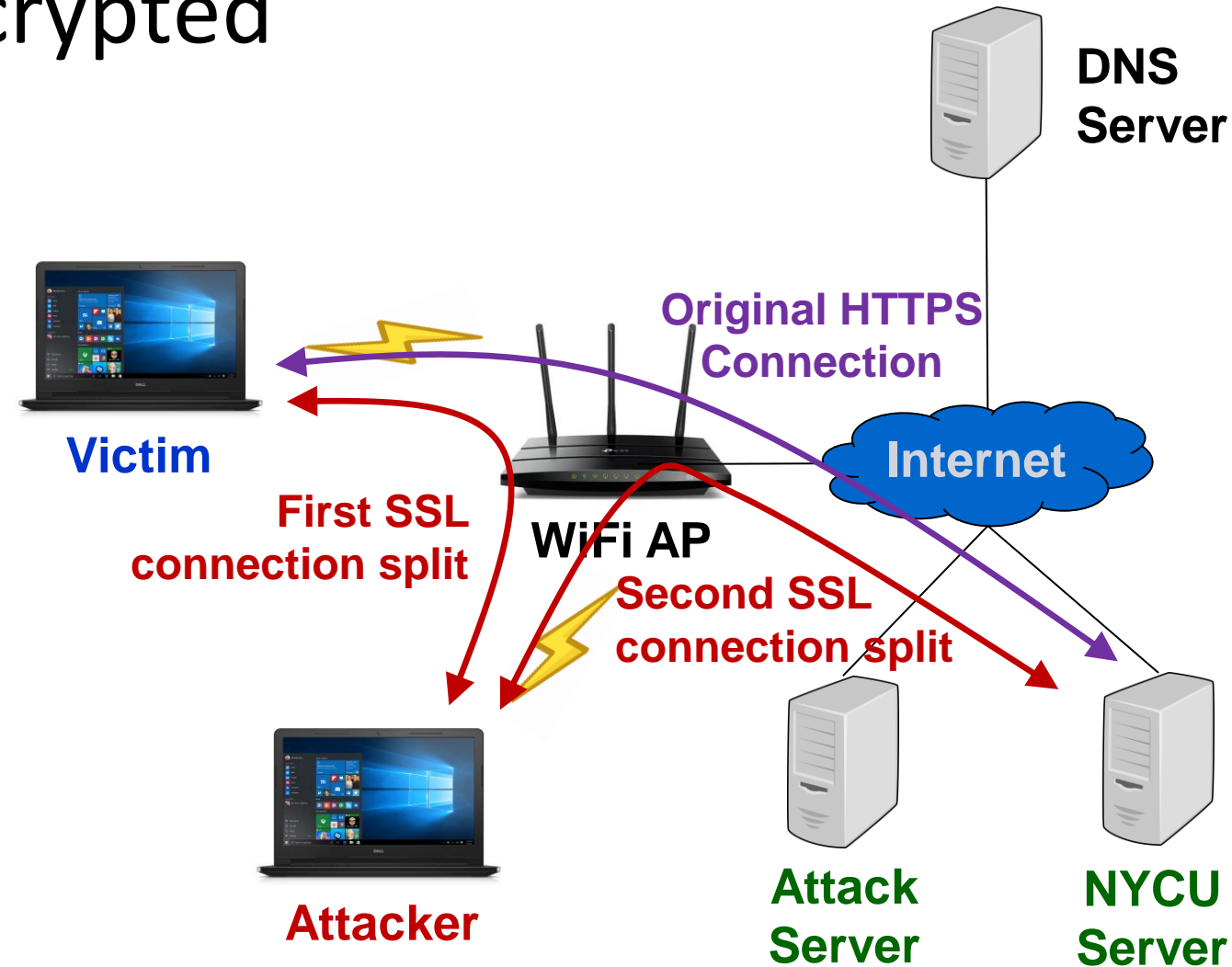
▶ Frame 2746: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
 ▶ Ethernet II, Src: HonHaiPr_45:5c:bb (b8:76:3f:45:5c:bb), Dst: IntelCor_7e:32:81 (00:e1:8c:7e:32:81)
 ▶ Internet Protocol version 4, Src: 8.8.8.8, Dst: 192.168.0.100
 ▶ Internet Control Message Protocol

Attacker → Victim

Task III: SSL Split on Encrypted SSL/TLS Connections

- Split SSL/TLS connections

- ❑ You can use 'sslsplit' command as a tool for this attack against encrypted network connection
- ❑ You are allowed to install certificates on the victim



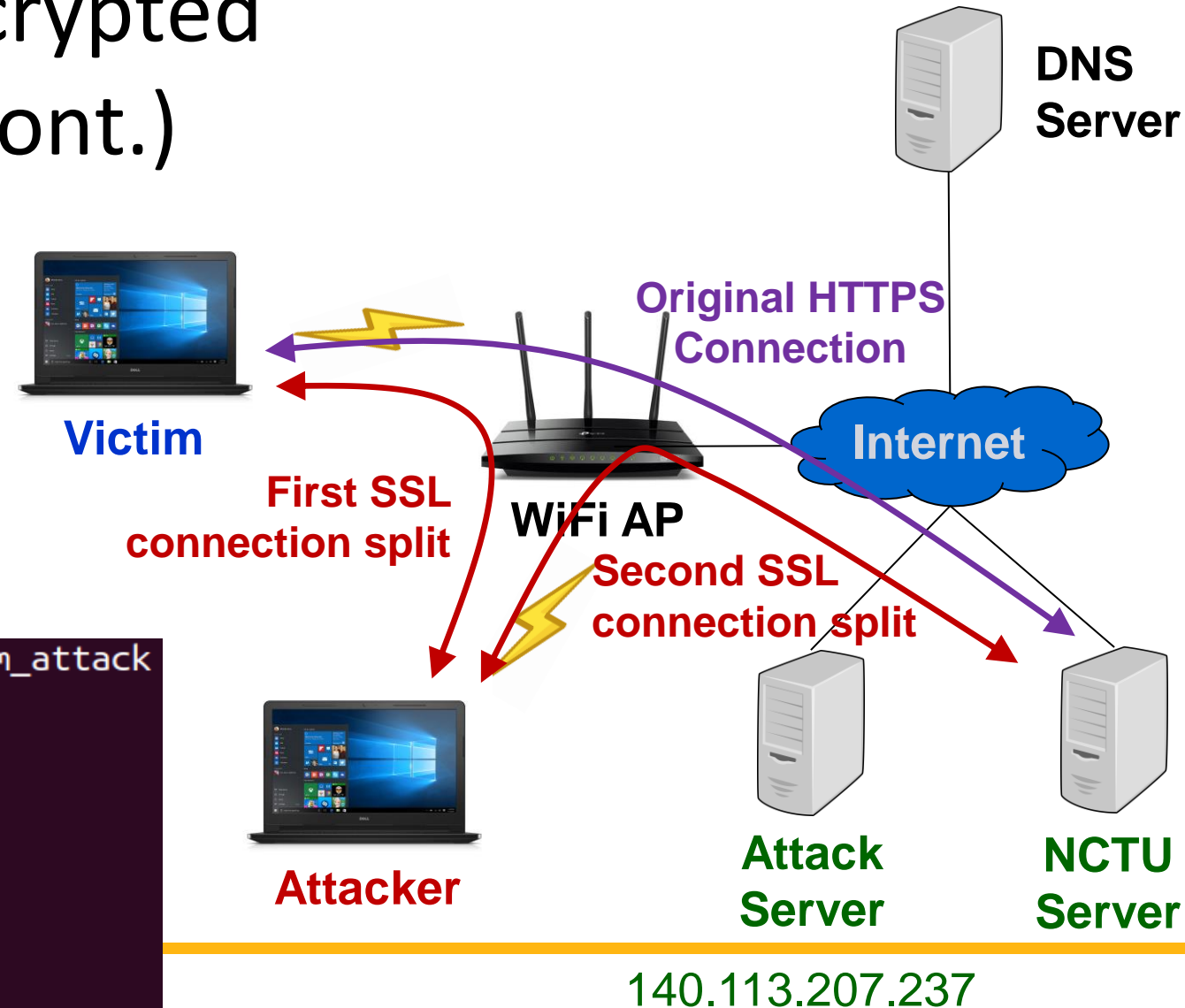
140.113.207.237

Task III: SSL Split on Encrypted SSL/TLS Connections (Cont.)

- Fetch all the inputted usernames/passwords on a specific web page
 - ❑ Parse HTTP content to print out usernames/passwords

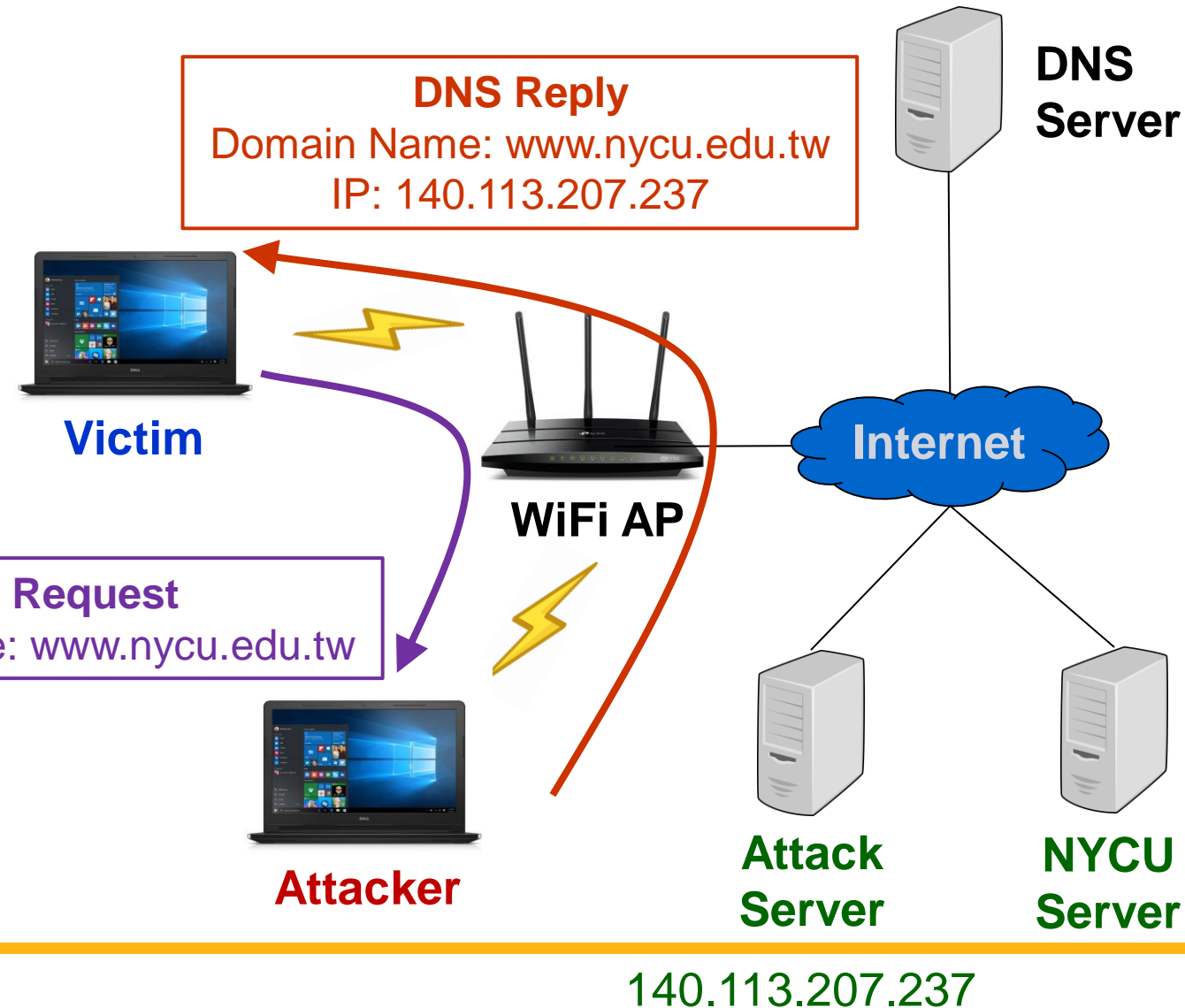
```
cs2021@ubuntu:~/Desktop/project2$ sudo ./mitm_attack
Available devices
-----
IP                MAC
-----
172.16.186.1      00:50:56:c0:00:08
172.16.186.141    00:0c:29:f2:d2:ab
172.16.186.254    00:50:56:ed:bd:5e

Username:  this_is_demo_username
Password:  this_is_demo_password
```



Task IV: DNS Spoofing

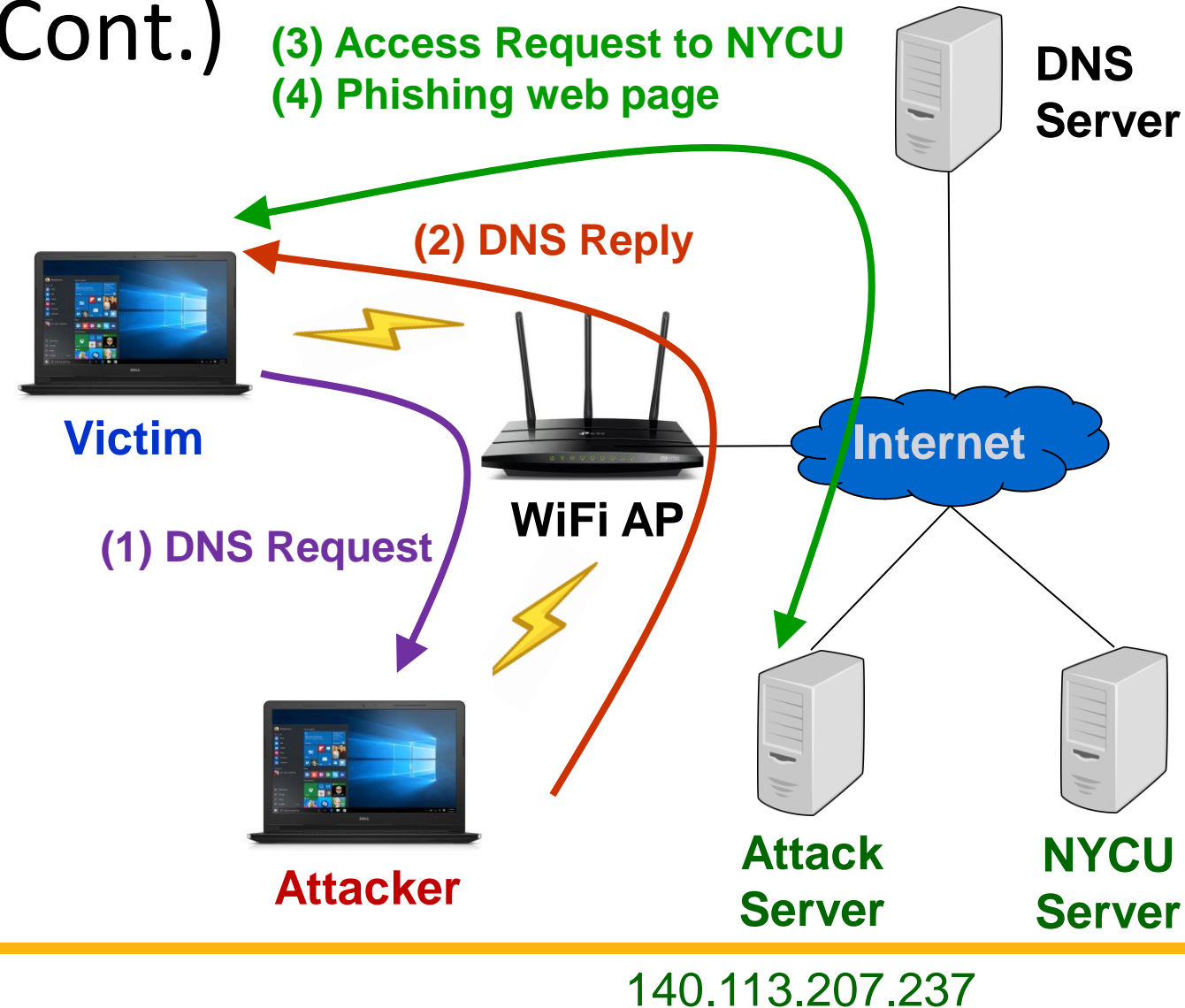
- Intercept DNS requests for a specific web page and generate spoofed DNS replies with the attack server's IP
 - You can use 'scapy' and 'NetfilterQueue' library in Python



Task IV: DNS Spoofing (Cont.)

● Successful attack

- ❑ An access request to NYCU home page will be redirected to the attack server (140.113.207.237)
- ❑ A phishing web page will be shown to Victim

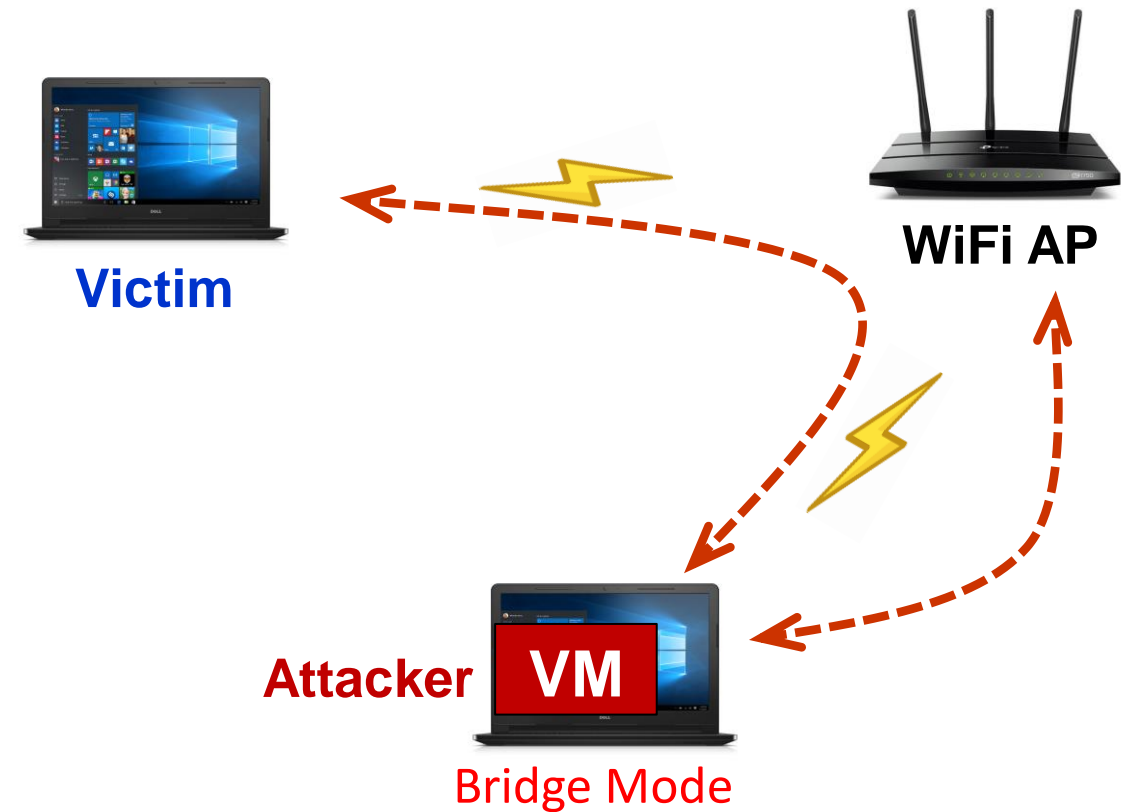


Requirements

- You need to develop/run your program in a given virtual machine
 - VM image: Please download it from [Link](#)
 - Username/password: cs2022/cs2022
- You are allowed to use C/C++ and Python
- You are allowed to team up. Each team has at most 2 students
 - Teams: discussions are allowed, but no collaboration
- Please submit your source codes and report to New E3

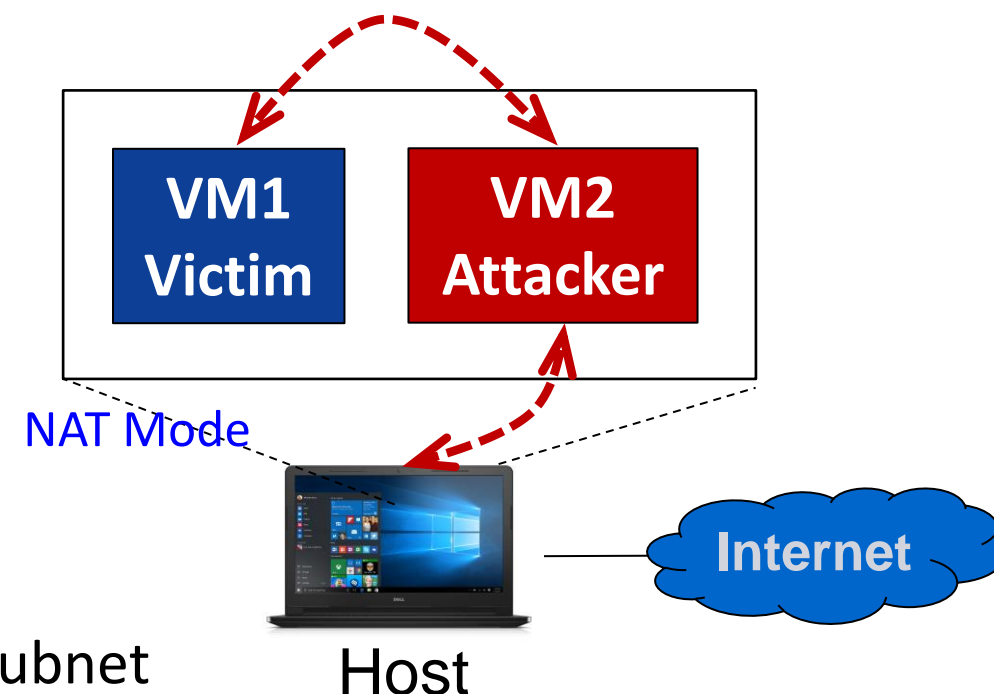
Test Scenario I: Target Scenario

- However, this scenario does not work for all the combinations of OS and VM software
 - ❑ Working: Linux + VirtualBox/VMware
 - ❑ Not working properly
 - Windows + VirtualBox/Vmware
 - MacOS + VirtualBox
- You can choose Test Scenario II



Test Scenario II: Alternative Scenario

- VM2 (Attacker) launches attacks on VM1 (Victim)
 - NAT mode shall be used for VMs
- Host is similar to the role of the AP in Test Scenario I
 - Scenario I: The Wi-Fi devices are in the same subnet
 - Scenario II: The VMs are in the same subnet



- Host can be connected to the Internet via Wi-Fi or wired Ethernet

Important: How to Prepare Your Attack Programs?

- Must provide a **Makefile** which compiles your source codes into two executable files, named **mitm_attack** and **pharm_attack** (Missing: -20%)
- Test requirements for the programs
 - ❑ Must be run in the given VM without any additional tools or libraries
 - ❑ Must use the following parameters
 - Test web page in the man-in-the-middle attack: <https://e3.nycu.edu.tw/login/index.php>
 - DNS spoofing for the NYCU home page: <http://www.nycu.edu.tw>
 - Attacker server IP in the DNS spoofing: [140.113.207.237](https://www.nycu.edu.tw)
 - ❑ Must work for the test commands: `./mitm_attack` and `./pharm_attack`

Important: How to Prepare Your Attack Programs?

- Results from the MITM attack (./mitm_attack)
 - ❑ Print out the IP/MAC addresses of **all the Wi-Fi devices or VMs** except for **Attacker and AP/Host**
 - ❑ Print out the username and password which a user submits to the website <https://e3.nycu.edu.tw/login/index.php> using any of **the Wi-Fi devices or VMs**
- Results from the pharming attack (./pharm_attack)
 - ❑ Print out the IP/MAC addresses of **all the Wi-Fi devices or VMs** except for **Attacker and AP/Host**
 - ❑ Redirect the NYCU home page (www.nycu.edu.tw) to the phishing page (140.113.207.237)
- Demo
 - ❑ Verify the MITM attack by giving inputs on the website using **one Wi-Fi device or VM**
 - Only allowed to run ./mitm_attack (No manual configuration is allowed)
 - ❑ Verify the pharming attack by accessing the NYCU page on **one Wi-Fi device or VM**
 - Only allowed to run ./pharm_attack (No manual configuration is allowed)

Important: How to Prepare Your Report?

- Item 1 (5%): please give evidence that you have finished the MITM attack
 - Specify your scenario (I or II) and Illustrate your results based on some snapshots
- Item 2 (5%): please give evidence that you have finished the pharming attack
 - Specify your scenario (I or II) and Illustrate your results based on some snapshots
- Item 3 (10%): please propose a solution that can defend against the ARP spoofing attack
 - No more than 200 English words
- Note: the report must be written in English with font size 11 or 12 in Times New Roman. It must be submitted in one PDF file with a name “report.pdf.”

Project Submission

- Due date: 4/19 11:55pm
- Makeup submission (75 points at most): TBA (After the final)
- Submission rules
 - ❑ Put all your files into a directory and name it using your student ID(s)
 - If your team has two members, please concatenate your IDs separated by “-”
 - ❑ Zip the directory and upload the zip file to New E3
 - ❑ A sample of the zip file: 01212112-02121221.zip
 - Makefile
 - mitm_attack.cpp
 - report.pdf
 - mitm_attack.h
 -

Project Demo

- Date: TBA
- Makeup submission (75 points at most): TBA (After the final)
- TA will prepare your zip file for you to demo
- You will
 - ❑ be asked to reproduce your MITM and pharming attacks
 - ❑ be only allowed to “make” to compile all your files, and run your attack binary programs or scripts
 - ❑ be not allowed to modify your codes or scripts
 - ❑ be asked some questions
 - E.g., How did you implement the SSL split? How did you resolve the certificate issue?
 - ❑ be responsible to show the outcome to TA and explain why you have successfully achieved the goals

Questions?