

# Computer Security

## Project 1: DNS Reflection and Amplification Attacks

Chi-Yu Li (2023 Fall)

Computer Science Department

National Yang Ming Chiao Tung University

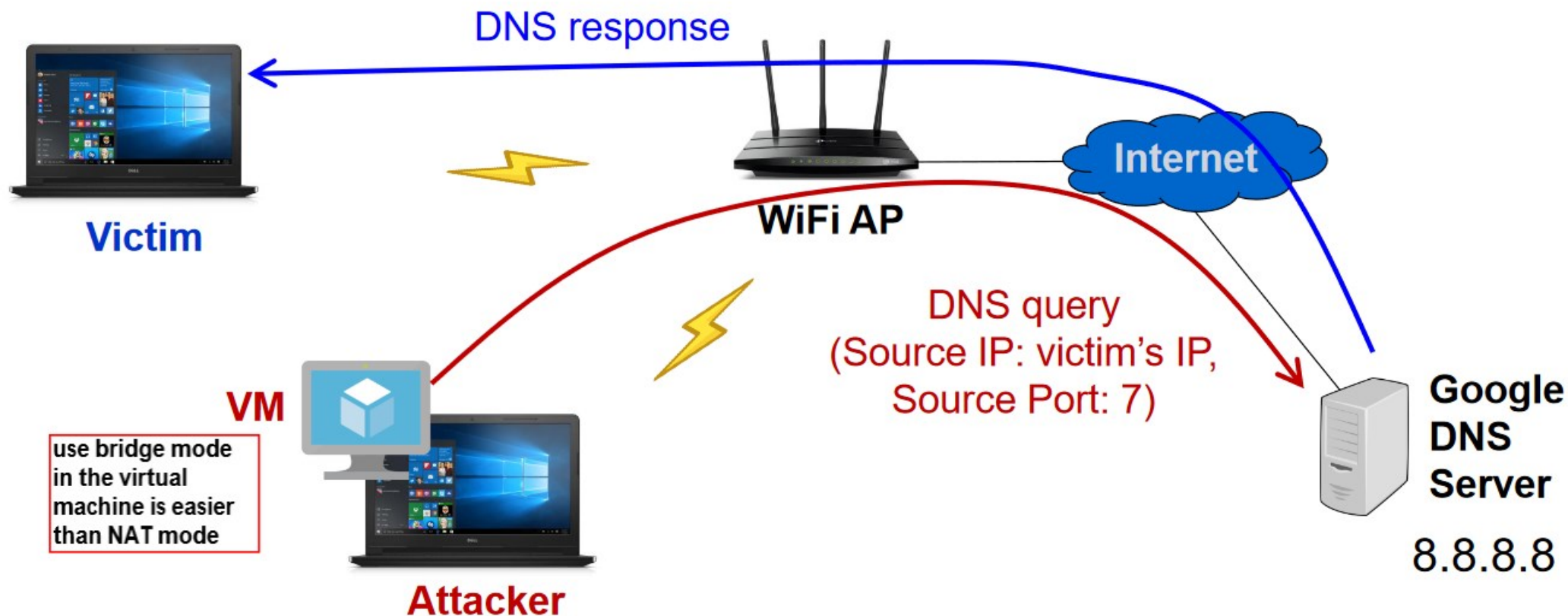
# Goal

- Understand how to launch DNS reflection and amplification attacks and then defend against them
- You will learn about
  - Program with raw sockets
  - Generate IP packets with spoofed IP addresses
  - Trace packets using Wireshark
  - Fabricate DNS query message
  - Launch DNS reflection and amplification attacks

# Requirements

- You need to develop/run your program in a given virtual machine
  - VM image: Please download it from here
    - Username: nems
    - Password: nems
    - Code: ~/dns
- The language you use must be C/C++
- You are allowed to team up. Each team has at most 2 students
  - Teams: discussions are allowed, but no collaboration
- Please submit your source code to New E3

# Your DNS Reflection Attack

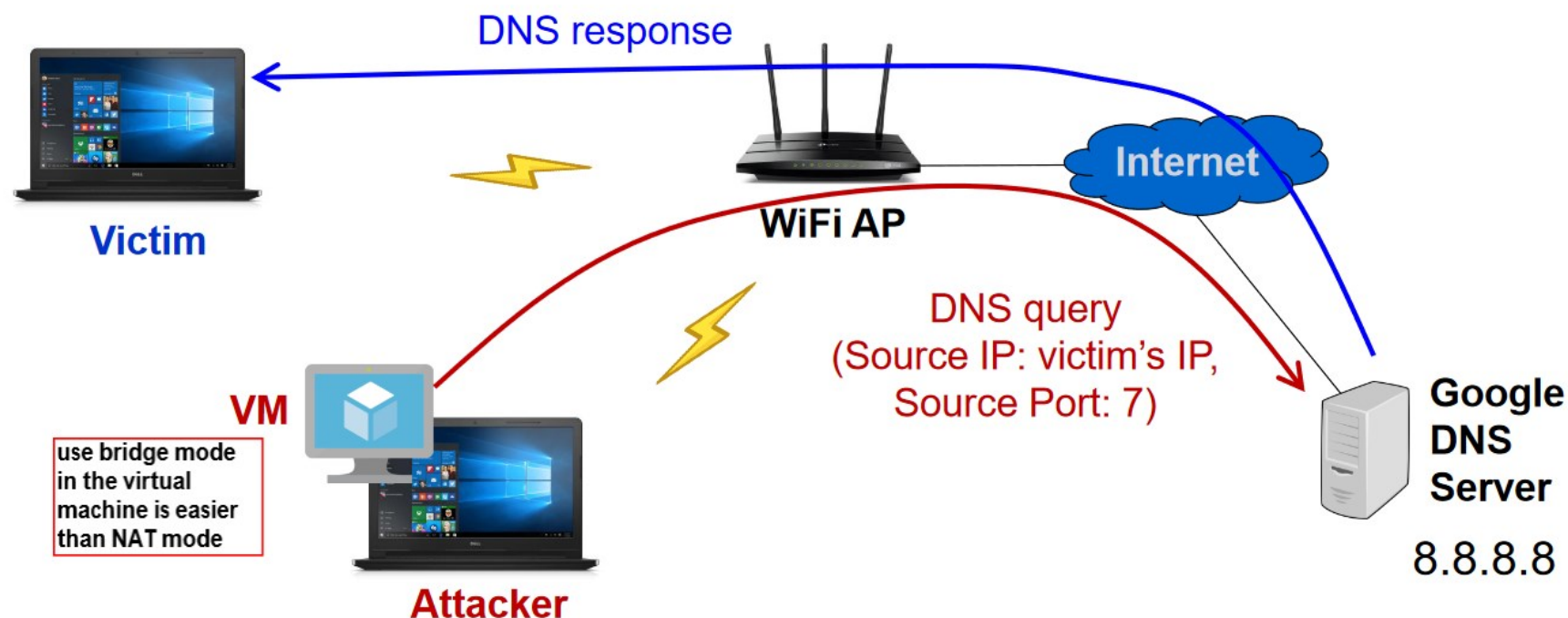


# Two Tasks

- Task I: DNS reflection attack (70 %)
- Task II: DNS amplification attack (30 %)
  - Amplification ratio:  $R = Sr/Sq$ 
    - $Sq$ : the packet size of the DNS query
    - $Sr$ : the packet size of the DNS response
  - $3 \leq R < 6$ : 10 %,  $6 \leq R < 10$ : 20 %,  $10 \leq R$ : 30 %

# Task I: DNS Reflection Attack

- (Given a DNS server's IP and the victim's IP)
- (Attacker) Fabricate a DNS query message in a UDP packet
- (Victim) Use Wireshark to check whether a corresponding DNS response is received

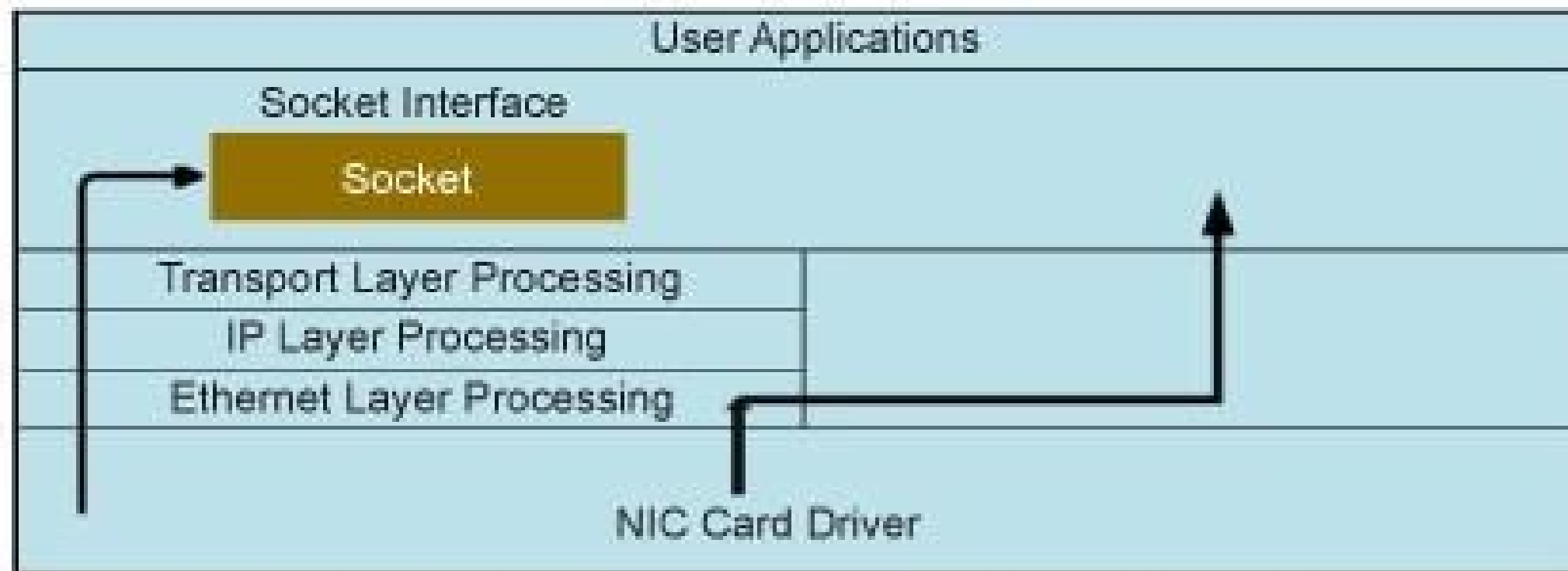


## Task II: DNS Amplification Attack

- (Given a DNS server's IP and the victim's IP)
- (Attacker) Fabricate a DNS query message that can trigger a large DNS response
  - ▣ Check the size of the UDP packet:  $Sq$
- (Victim) Check the size of the corresponding DNS response:  $Sr$ 
  - ▣ Obtain the amplification ratio:  $R = Sr/S$

# Hint I: How to Create IP Spoofing Packets?

- Using Raw Socket



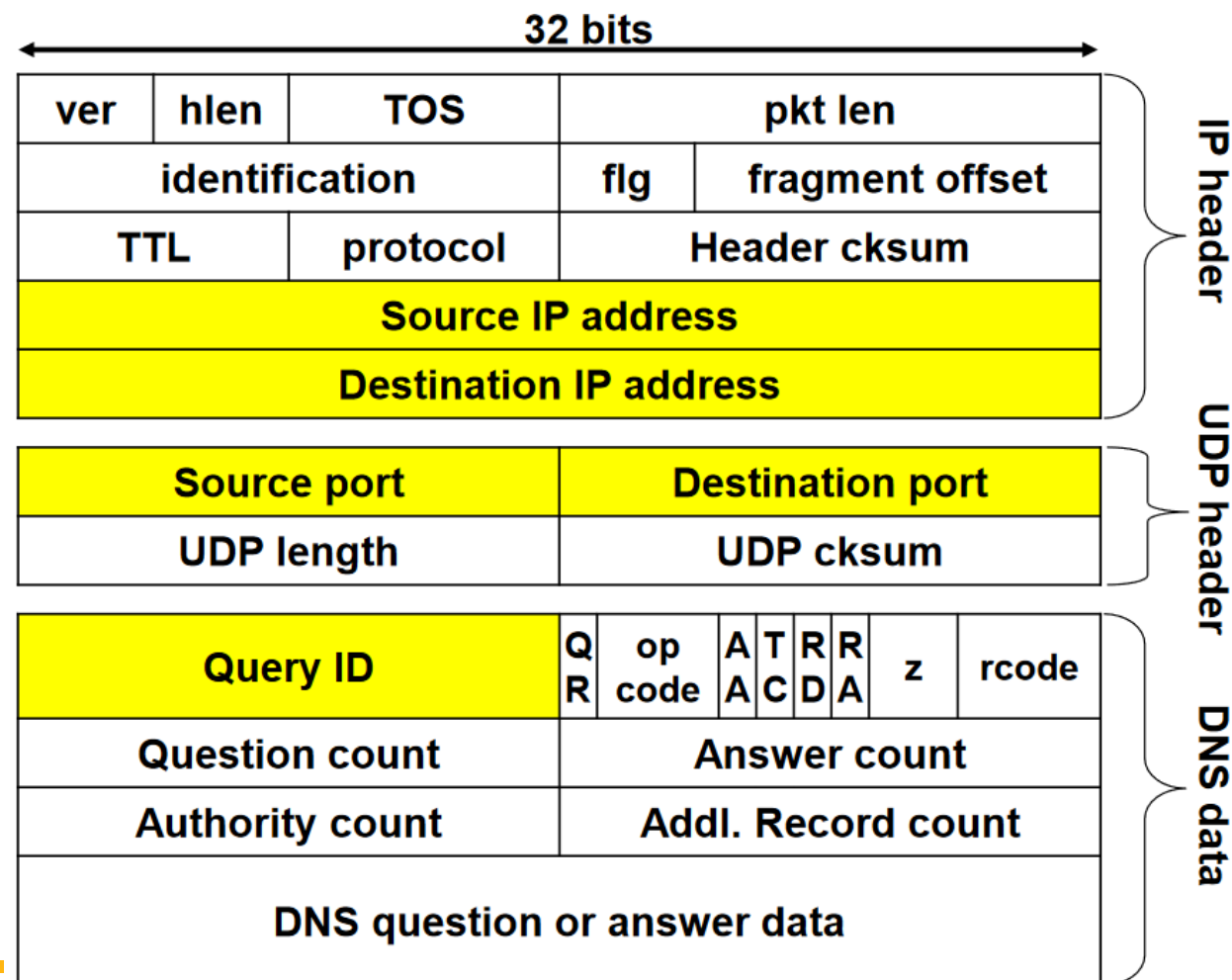


# Hint I: How to Create IP Spoofing Packets? (Cont.)

- Implementation based on raw socket
  - ❑ Create a raw socket with the UDP protocol
  - ❑ Fabricate the IP header
  - ❑ Fabricate the UDP header
  - ❑ Calculate the checksum over IP and UDP headers
  - ❑ Create DNS query in the UDP payload
- Tutorial

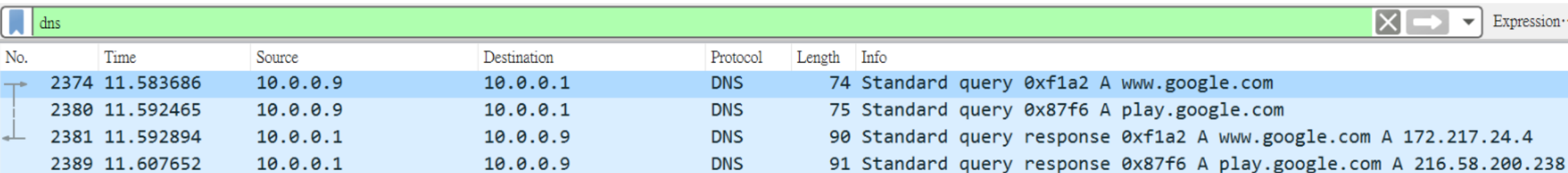
# Hint I: How to Create IP Spoofing Packets? (Cont.)

- DNS/UDP/IP packet format



# Hint II: How to Create a DNS Query Message?

- Generate a DNS query (e.g., using ping) and then capture it using Wireshark



No.	Time	Source	Destination	Protocol	Length	Info
2374	11.583686	10.0.0.9	10.0.0.1	DNS	74	Standard query 0xf1a2 A www.google.com
2380	11.592465	10.0.0.9	10.0.0.1	DNS	75	Standard query 0x87f6 A play.google.com
2381	11.592894	10.0.0.1	10.0.0.9	DNS	90	Standard query response 0xf1a2 A www.google.com A 172.217.24.4
2389	11.607652	10.0.0.1	10.0.0.9	DNS	91	Standard query response 0x87f6 A play.google.com A 216.58.200.238

# Hint II: How to Create a DNS Query Message? (Cont.)

- Fill in the content of the query based on the observation from Wireshark

2374	11.583686	10.0.0.9	10.0.0.1	DNS
2380	11.592465	10.0.0.9	10.0.0.1	DNS
2381	11.592894	10.0.0.1	10.0.0.9	DNS
2389	11.607652	10.0.0.1	10.0.0.9	DNS

>	Frame 2374: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on
>	Ethernet II, Src: IntelCor_96:0a:8c (fc:77:74:96:0a:8c), Dst: Netgear_a4:
>	Internet Protocol Version 4, Src: 10.0.0.9, Dst: 10.0.0.1
>	User Datagram Protocol, Src Port: 61039, Dst Port: 53
▼	Domain Name System (query)
	Transaction ID: 0xf1a2
>	Flags: 0x0100 Standard query
	Questions: 1
	Answer RRs: 0
	Authority RRs: 0
	Additional RRs: 0
▼	Queries
▼	www.google.com: type A, class IN
	Name: www.google.com
	[Name Length: 14]
	[Label Count: 3]
	Type: A (Host Address) (1)
	Class: IN (0x0001)

0000	dc ef 09 a4 33 f0 fc 77 74 96 0a 8c 08 00 45 00	....3..w t.....E.
0010	00 3c f0 3c 00 00 80 11 36 6b 0a 00 00 09 0a 00	..<.<.... 6k.....
0020	00 01 ee 6f 00 35 00 28 7c a5 f1 a2 01 00 00 01	...o.5.(  .....
0030	00 00 00 00 00 00 03 77 77 77 06 67 6f 6f 67 6c	.....w ww-googl
0040	65 03 63 6f 6d 00 00 01 00 01	e.com... ..

# Important: How to Prepare Your Attack Program?

- Must provide a Makefile which compiles your source codes into one executable file, named `dns_attack` (Missing: -20%)
- Test requirements for the program (Missing: -10% each)
  - Must be run in the given VM without any additional tools or libraries
  - Must work for the test command: `./dns_attack <Victim IP> <Victim Port> <DNS Server IP>`
    - E.g., `./dns_attack 10.0.0.2 7 8.8.8.8`
  - After being executed, the program shall send DNS query and then terminate
  - Use the last 16 bits of your student ID in the Query ID of the DNS queries
    - Use the ID of only one member in your team
    - E.g., Student ID: 0756842 → Query ID in hex: 0x8C6A

# Project Submission

- Due date: 11/1 23:59
- Demo: 11/2 and 11/3 evening
- 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 “-”
    - Please put the student ID used for the Query ID at the beginning of the name
  - ❑ Zip the directory and upload the zip file to New E3
  - ❑ A sample of the zip file: 01212112-02121221.zip
    - makefile
    - main.cpp
    - dns\_attack.cpp
    - dns\_attack.h
    - ...

# Questions?