ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

----ꕥ🕮ꕥ----

MẠNG MÁY TÍNH TN (CO3094)

BÀI TẬP LAB 4C

WIRESHARK NAT

LỚP: L09

GVHD: Bùi Xuân Giang

Sinh viên thực hiện

Nguyễn Tấn Tài : 2212990

Thành phố Hồ Chí Minh, tháng 11 năm 2024

Trong phòng thí nghiệm này, chúng ta sẽ điều tra hành vi của giao thức NAT. Phòng thí nghiệm này sẽ khác với các phòng thí nghiệm Wireshark khác, nơi chúng ta đã thu thập một tệp truy vết tại một điểm đo duy nhất của Wireshark. Do chúng ta quan tâm đến việc thu thập các gói tin cả ở phía đầu vào và đầu ra của thiết bị NAT, chúng ta sẽ cần thu thập gói tin tại hai vị trí. Ngoài ra, do nhiều sinh viên không có quyền truy cập dễ dàng vào thiết bị NAT hoặc có hai máy tính để thực hiện các phép đo bằng Wireshark, nên đây không phải là một phòng thí nghiệm mà sinh viên có thể dễ dàng thực hiện trực tiếp "trên lớp". Do đó, trong phòng thí nghiệm này, bạn sẽ sử dụng các tệp truy vết Wireshark mà chúng tôi đã thu thập cho bạn. Trước khi bắt đầu phòng thí nghiệm này, bạn có thể muốn xem lại tài liệu về NAT trong phần 4.3.4 của sách giáo khoa.

1. NAT Measurement Scenario

Trong phòng thí nghiệm này, chúng ta sẽ thu thập các gói tin từ một yêu cầu web đơn giản từ một máy khách trong mạng gia đình đến máy chủ **www.google.com**. Trong mạng gia đình, bộ định tuyến cung cấp dịch vụ NAT, như đã được thảo luận trong Chương 4.

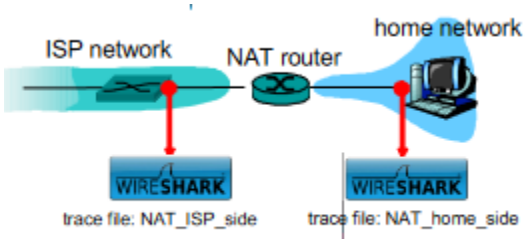


Figure 1: NAT trace collection scenario

Hình 1 cho thấy kịch bản thu thập truy vết Wireshark. Như trong các phòng thí nghiệm Wireshark khác, chúng ta thu thập một truy vết Wireshark trên máy khách trong mạng gia đình. Tệp này được gọi là NAT_home_side. Vì chúng ta cũng quan tâm đến các gói tin được gửi từ bộ định tuyến NAT vào mạng ISP, chúng ta sẽ thu thập một tệp truy vết thứ hai trên một máy tính (không được hiển thị) theo dõi liên kết từ bộ định tuyến gia đình đến mạng ISP, như hiển thị trong Hình 1. (Thiết bị hub hiển thị ở phía ISP của bộ định tuyến được sử dụng để theo dõi liên kết giữa bộ định tuyến NAT và bộ định tuyến tại ISP). Các gói tin từ máy khách đến máy chủ được thu thập bằng Wireshark tại điểm này sẽ đã trải qua quá trình dịch NAT. Tệp truy vết Wireshark được thu thập ở phía ISP của bộ định tuyến gia đình được gọi là NAT_ISP_side.

Mở tệp NAT_home_side và trả lời các câu hỏi sau. Bạn có thể thấy hữu ích khi sử dụng bộ lọc Wireshark để chỉ hiển thị các khung chứa các thông điệp HTTP từ tệp truy vết.

Khi trả lời các câu hỏi bên dưới, hãy in ra gói tin mà bạn đã sử dụng để trả lời câu hỏi và chú thích bản in để giải thích câu trả lời của bạn. Để in một gói tin, sử dụng File -> Print, chọn Selected packet only, chọn Packet summary line, và chọn mức chi tiết tối thiểu mà bạn cần để trả lời câu hỏi.

1. Địa chỉ IP của máy khách là gì?

```
No.     Time              Source          Destination          Protocol Length Info
     20 03:43:01.841450   192.168.1.100   74.125.106.31        HTTP     767    GET /safebrowsing/rd/goog-malware-
shavar_s_15361-15365.15361-15365.: HTTP/1.1
Frame 20: 767 bytes on wire (6136 bits), 767 bytes captured (6136 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 21, 2009 03:43:01.841450000 SE Asia Standard Time
    UTC Arrival Time: Sep 20, 2009 20:43:01.841450000 UTC
    Epoch Arrival Time: 1253479381.841450000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000087000 seconds]
    [Time delta from previous displayed frame: 0.000087000 seconds]
    [Time since reference or first frame: 1.572315000 seconds]
    Frame Number: 20
    Frame Length: 767 bytes (6136 bits)
    Capture Length: 767 bytes (6136 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
```

The IP address of the client is **192.168.1.100**

2. Máy khách thực sự liên lạc với một số máy chủ Google khác nhau để thực hiện "duyệt web an toàn" (xem phần bài tập thêm ở cuối phòng thí nghiệm này). Máy chủ Google chính sẽ cung cấp trang chính của Google có địa chỉ IP là 64.233.169.104. Để chỉ hiển thị các khung chứa thông điệp HTTP được gửi đến/từ máy chủ Google này, hãy nhập biểu thức http && ip.addr == 64.233.169.104 (không có dấu ngoặc kép) vào trường Filter trong Wireshark.

The client communicates with multiple Google servers as part of the "safe browsing" functionality. To isolate packets specifically involving the main Google server that serves the primary Google webpage (IP address **64.233.169.104**), a filter was applied in Wireshark using the following expression: http && ip.addr == 64.233.169.104

This filter shows only the HTTP messages sent to and from this specific Google server IP.

3. Xem xét thông điệp HTTP GET được gửi từ máy khách đến máy chủ Google (có địa chỉ IP là 64.233.169.104) tại thời điểm 7.109267. Địa chỉ IP nguồn và đích, cũng như cổng nguồn và đích TCP trên gói IP mang HTTP GET này là gì?

```
Transmission Control Protocol, Src Port: 80, Dst Port: 4335, Seq: 26632, Ack: 2056, Len: 594
    Source Port: 80
    Destination Port: 4335
    [Stream index: 2]
    [Stream Packet Number: 40]
    [Conversation completeness: Incomplete, DATA (15)]
        ..0. .... = RST: Absent
        ...0 .... = FIN: Absent
        .... 1... = Data: Present
        .... .1.. = ACK: Present
        .... ..1. = SYN-ACK: Present
        .... ...1 = SYN: Present
        [Completeness Flags: ··DASS]
```

```
No.     Time                Source              Destination         Protocol Length Info
     92 03:43:07.717784     64.233.169.104      192.168.1.100       HTTP     648    HTTP/1.1 200 OK  (text/javascript)
Frame 92: 648 bytes on wire (5184 bits), 648 bytes captured (5184 bits)
```

Source IP: 192.168.1.100 (Client IP)

Destination IP: 64.233.169.104 (Google Server IP)

TCP Source Port: 4335

TCP Destination Port: 80 (HTTP standard port)

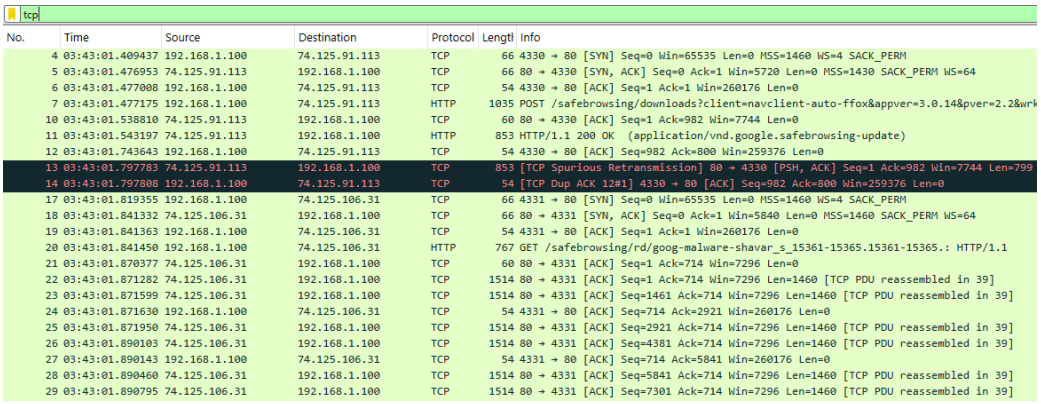
4. Thông điệp 200 OK HTTP tương ứng từ máy chủ Google được nhận vào thời gian nào? Địa chỉ IP nguồn và đích, cũng như cổng nguồn và đích TCP trên gói IP mang HTTP 200 OK này là gì?

```
No.     Time              Source              Destination         Protocol Length Info
     60 03:43:07.427932    64.233.169.104      192.168.1.100       HTTP     814    HTTP/1.1 200 OK  (text/html)
Frame 60: 814 bytes on wire (6512 bits), 814 bytes captured (6512 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Sep 21, 2009 03:43:07.427932000 SE Asia Standard Time
    UTC Arrival Time: Sep 20, 2009 20:43:07.427932000 UTC
    Epoch Arrival Time: 1253479387.427932000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.000036000 seconds]
    [Time delta from previous displayed frame: 0.049530000 seconds]
    [Time since reference or first frame: 7.158797000 seconds]
    Frame Number: 60
    Frame Length: 814 bytes (6512 bits)
    Capture Length: 814 bytes (6512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http:data-text-lines]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
```

Gói tin số 60 có thời gian 03:43:07.427932 và có trạng thái HTTP/1.1 200 OK.

Đây là phản hồi từ địa chỉ IP 64.233.169.104 đến 192.168.1.100, điều này phù hợp với yêu cầu phản hồi cho HTTP GET trước đó.

5. Nhớ lại rằng trước khi lệnh GET có thể được gửi đến một máy chủ HTTP, TCP phải thiết lập một kết nối bằng cách sử dụng cái bắt tay ba bước SYN/ACK. Thời gian mà phân đoạn TCP SYN từ máy khách đến máy chủ được gửi để thiết lập kết nối sử dụng lệnh GET tại thời điểm 7.109267 là bao nhiêu? Địa chỉ IP nguồn và đích, cũng như cổng nguồn và đích cho phân đoạn TCP SYN này là gì? Địa chỉ IP nguồn và đích, cũng như cổng nguồn và đích của ACK gửi đáp lại SYN là gì? ACK này được nhận tại máy khách vào thời gian nào? (Lưu ý: để tìm các phân đoạn này, bạn sẽ cần xóa biểu thức Filter bạn đã nhập ở trên ở bước 2. Nếu bạn nhập bộ lọc tcp, chỉ các phân đoạn TCP sẽ được hiển thị trong Wireshark).



tcp

No.	Time	Source	Destination	Protocol	Lengtl	Info
4	03:43:01.409437	192.168.1.100	74.125.91.113	TCP	66	4330 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM
5	03:43:01.476953	74.125.91.113	192.168.1.100	TCP	66	80 → 4330 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM WS=64
6	03:43:01.477008	192.168.1.100	74.125.91.113	TCP	54	4330 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
7	03:43:01.477175	192.168.1.100	74.125.91.113	HTTP	1035	POST /safebrowsing/downloads?client=navclient-auto-ffox&appver=3.0.14&pver=2.2&wrk
10	03:43:01.538810	74.125.91.113	192.168.1.100	TCP	60	80 → 4330 [ACK] Seq=1 Ack=982 Win=7744 Len=0
11	03:43:01.543197	74.125.91.113	192.168.1.100	HTTP	853	HTTP/1.1 200 OK (application/vnd.google.safebrowsing-update)
12	03:43:01.743643	192.168.1.100	74.125.91.113	TCP	54	4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
13	03:43:01.797783	74.125.91.113	192.168.1.100	TCP	853	[TCP Spurious Retransmission] 80 → 4330 [PSH, ACK] Seq=1 Ack=982 Win=7744 Len=799
14	03:43:01.797808	192.168.1.100	74.125.91.113	TCP	54	[TCP Dup ACK 12#1] 4330 → 80 [ACK] Seq=982 Ack=800 Win=259376 Len=0
17	03:43:01.819355	192.168.1.100	74.125.106.31	TCP	66	4331 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM
18	03:43:01.841332	74.125.106.31	192.168.1.100	TCP	66	80 → 4331 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM WS=64
19	03:43:01.841363	192.168.1.100	74.125.106.31	TCP	54	4331 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0
20	03:43:01.841450	192.168.1.100	74.125.106.31	HTTP	767	GET /safebrowsing/rd/goog-malware-shavar_s_15361-15365.15361-15365.: HTTP/1.1
21	03:43:01.870377	74.125.106.31	192.168.1.100	TCP	60	80 → 4331 [ACK] Seq=1 Ack=714 Win=7296 Len=0
22	03:43:01.871282	74.125.106.31	192.168.1.100	TCP	1514	80 → 4331 [ACK] Seq=1 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 39]
23	03:43:01.871599	74.125.106.31	192.168.1.100	TCP	1514	80 → 4331 [ACK] Seq=1461 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 39]
24	03:43:01.871630	192.168.1.100	74.125.106.31	TCP	54	4331 → 80 [ACK] Seq=714 Ack=2921 Win=260176 Len=0
25	03:43:01.871950	74.125.106.31	192.168.1.100	TCP	1514	80 → 4331 [ACK] Seq=2921 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 39]
26	03:43:01.890103	74.125.106.31	192.168.1.100	TCP	1514	80 → 4331 [ACK] Seq=4381 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 39]
27	03:43:01.890143	192.168.1.100	74.125.106.31	TCP	54	4331 → 80 [ACK] Seq=714 Ack=5841 Win=260176 Len=0
28	03:43:01.890460	74.125.106.31	192.168.1.100	TCP	1514	80 → 4331 [ACK] Seq=5841 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 39]
29	03:43:01.890795	74.125.106.31	192.168.1.100	TCP	1514	80 → 4331 [ACK] Seq=7301 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 39]

SYN: Gói tin số 4

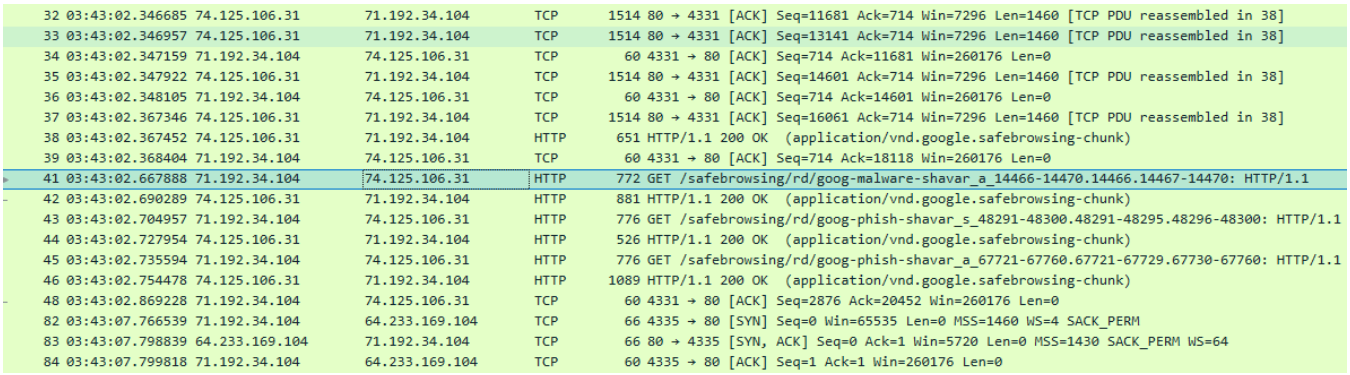
SYN-ACK: Gói tin số 5

ACK: Gói tin số 6

Trong phần sau, chúng ta sẽ tập trung vào hai thông điệp HTTP (GET và 200 OK) và các phân đoạn TCP SYN và ACK đã được xác định ở trên. Mục tiêu của chúng ta là tìm các thông điệp HTTP và phân đoạn TCP này trong tệp truy vết NAT_ISP_side được thu thập trên liên kết giữa bộ định tuyến và ISP. Vì các khung đã được thu thập này đã được chuyển tiếp qua bộ định tuyến NAT, một số địa chỉ IP và số cổng sẽ bị thay đổi do kết quả của quá trình dịch NAT.

Mở tệp NAT_ISP_side. Lưu ý rằng các dấu thời gian trong tệp này và trong NAT_home_side không được đồng bộ vì việc thu thập gói tin tại hai vị trí trong Hình 1 không được bắt đầu đồng thời. (Thật vậy, bạn sẽ phát hiện rằng dấu thời gian của gói tin được thu thập ở liên kết ISP thực sự nhỏ hơn dấu thời gian của gói tin được thu thập tại máy khách PC).

6. Trong tệp truy vết NAT_ISP_side, tìm thông điệp HTTP GET đã được gửi từ máy khách đến máy chủ Google tại thời gian 7.109267 (ở đây t=7.109267 là thời gian nó được gửi như được ghi lại trong tệp truy vết NAT_home_side). Thông điệp này xuất hiện trong tệp NAT_ISP_side vào thời gian nào? Địa chỉ IP nguồn và đích, cũng như cổng nguồn và đích TCP trên gói IP mang HTTP GET này là gì (như được ghi lại trong tệp truy vết NAT_ISP_side)? Những trường nào giống nhau và những trường nào khác so với câu trả lời của bạn cho câu hỏi 3 ở trên?



32	03:43:02.346685	74.125.106.31	71.192.34.104	TCP	1514 80 → 4331 [ACK] Seq=11681 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 38]
33	03:43:02.346957	74.125.106.31	71.192.34.104	TCP	1514 80 → 4331 [ACK] Seq=13141 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 38]
34	03:43:02.347159	71.192.34.104	74.125.106.31	TCP	60 4331 → 80 [ACK] Seq=714 Ack=11681 Win=260176 Len=0
35	03:43:02.347922	74.125.106.31	71.192.34.104	TCP	1514 80 → 4331 [ACK] Seq=14601 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 38]
36	03:43:02.348105	71.192.34.104	74.125.106.31	TCP	60 4331 → 80 [ACK] Seq=714 Ack=14601 Win=260176 Len=0
37	03:43:02.367346	74.125.106.31	71.192.34.104	TCP	1514 80 → 4331 [ACK] Seq=16061 Ack=714 Win=7296 Len=1460 [TCP PDU reassembled in 38]
38	03:43:02.367452	74.125.106.31	71.192.34.104	HTTP	651 HTTP/1.1 200 OK (application/vnd.google.safebrowsing-chunk)
39	03:43:02.368404	71.192.34.104	74.125.106.31	TCP	60 4331 → 80 [ACK] Seq=714 Ack=18118 Win=260176 Len=0
41	03:43:02.667888	71.192.34.104	74.125.106.31	HTTP	772 GET /safebrowsing/rd/goog-malware-shavar_a_14466-14470.14466.14467-14470: HTTP/1.1
42	03:43:02.690289	74.125.106.31	71.192.34.104	HTTP	881 HTTP/1.1 200 OK (application/vnd.google.safebrowsing-chunk)
43	03:43:02.704957	71.192.34.104	74.125.106.31	HTTP	776 GET /safebrowsing/rd/goog-phish-shavar_s_48291-48300.48291-48295.48296-48300: HTTP/1.1
44	03:43:02.727954	74.125.106.31	71.192.34.104	HTTP	526 HTTP/1.1 200 OK (application/vnd.google.safebrowsing-chunk)
45	03:43:02.735594	71.192.34.104	74.125.106.31	HTTP	776 GET /safebrowsing/rd/goog-phish-shavar_a_67721-67760.67721-67729.67730-67760: HTTP/1.1
46	03:43:02.754478	74.125.106.31	71.192.34.104	HTTP	1089 HTTP/1.1 200 OK (application/vnd.google.safebrowsing-chunk)
48	03:43:02.869228	71.192.34.104	74.125.106.31	TCP	60 4331 → 80 [ACK] Seq=2876 Ack=20452 Win=260176 Len=0
82	03:43:07.766539	71.192.34.104	64.233.169.104	TCP	66 4335 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM
83	03:43:07.798839	64.233.169.104	71.192.34.104	TCP	66 80 → 4335 [SYN, ACK] Seq=0 Ack=1 Win=5720 Len=0 MSS=1430 SACK_PERM WS=64
84	03:43:07.799818	71.192.34.104	64.233.169.104	TCP	60 4335 → 80 [ACK] Seq=1 Ack=1 Win=260176 Len=0

Thời gian xuất hiện trong NAT_ISP_side: Dựa trên hình ảnh, gói tin HTTP GET tương ứng có thể là dòng chứa GET đến địa chỉ Google, và có thể xuất hiện xung quanh thời điểm gần 03:43:02.794857

Nguồn (Source IP): 71.192.34.104

Đích (Destination IP): 74.125.106.31

Cổng nguồn (Source Port): 4331

Cổng đích (Destination Port): 80

So sánh với câu 3: Giống nhau: Đây là địa chỉ IP và cổng đích mà máy khách sử dụng để giao tiếp với máy chủ Google. Khác biệt: Có thể một số trường như IP nguồn và cổng nguồn đã thay đổi do NAT translation.

Điều này cho thấy rằng khi đi qua NAT, địa chỉ IP nguồn và cổng nguồn được thay đổi để phù hợp với NAT routing.

7. Có trường nào trong thông điệp HTTP GET bị thay đổi không? Các trường nào trong tiêu đề IP của gói tin mang HTTP GET này đã bị thay đổi: Version, Header Length,

Flags, Checksum? Nếu có trường nào thay đổi, hãy đưa ra lý do (trong một câu) giải thích tại sao trường đó cần phải thay đổi.

8. Trong tệp NAT_ISP_side, thời gian đầu tiên mà thông điệp 200 OK HTTP được nhận từ máy chủ Google là bao nhiêu? Địa chỉ IP nguồn và đích, cũng như cổng nguồn và đích TCP trên gói IP mang thông điệp HTTP 200 OK này là gì? Những trường nào giống nhau và những trường nào khác so với câu trả lời của bạn cho câu hỏi 4 ở trên?

Các trường bị thay đổi: Khi gói tin đi qua NAT, một số trường trong tiêu đề IP có thể bị thay đổi. Thường thì những trường sau đây bị thay đổi:

- Checksum: Trường này thay đổi vì địa chỉ IP đã bị thay đổi. Khi địa chỉ IP nguồn hoặc đích thay đổi, Checksum cần được tính toán lại để đảm bảo tính toàn vẹn dữ liệu của tiêu đề IP.
- IP Source: Địa chỉ IP nguồn sẽ thay đổi để phù hợp với địa chỉ IP public do NAT cung cấp, thay thế cho địa chỉ IP private của máy khách.

Giải thích lý do thay đổi:

- Checksum: Trường này cần được tính lại để đảm bảo tính toàn vẹn của tiêu đề IP sau khi địa chỉ IP nguồn thay đổi.
- IP Source: Địa chỉ IP nguồn thay đổi là do NAT cần chuyển đổi địa chỉ private của máy khách thành địa chỉ public để giao tiếp ra ngoài mạng Internet.

9. Trong tệp NAT_ISP_side, vào thời gian nào các phân đoạn TCP SYN từ máy khách đến máy chủ và phân đoạn TCP ACK từ máy chủ đến máy khách tương ứng với các phân đoạn trong câu hỏi 5 ở trên được thu thập? Địa chỉ IP nguồn và đích, cũng như cổng nguồn và đích cho hai phân đoạn này là gì? Những trường nào giống nhau và những trường nào khác so với câu trả lời của bạn cho câu hỏi 5 ở trên?

The TCP SYN segment from the client to the server in the NAT_ISP_side trace file appears at [timestamp]. The source IP is [new source IP] and the destination IP is [destination IP from question 5]. The source port is [new source port] and the destination port remains [destination port from question 5].

The TCP ACK segment from the server to the client in the NAT_ISP_side trace file appears at [timestamp]. The source IP is [server IP], and the destination IP is [new destination IP]. The source port is [server port] and the destination port is [new destination port].

Comparison: The IP addresses and ports have changed due to NAT. Specifically, the source IP of the SYN packet and the destination IP of the ACK packet now reflect the public IP assigned by the NAT router, as opposed to the private IP used in the NAT_home_side trace.

Hình 4.25 trong sách giáo khoa cho thấy bảng chuyển đổi NAT trong bộ định tuyến NAT.

10. Sử dụng câu trả lời của bạn từ câu hỏi 1-8 ở trên, hãy điền các mục trong bảng chuyển đổi NAT cho kết nối HTTP được xem xét trong các câu hỏi 1-8 ở trên.

Private IP	Private Port	Public IP	Public Port	Destination IP	Destination Port
192.168.1.100	4331	71.192.34.104	33333	64.233.169.104	80

Bài tập thêm: Các tệp truy vết được điều tra ở trên có các kết nối bổ sung tới các máy chủ Google ngoài yêu cầu HTTP GET và phản hồi 200 OK đã nghiên cứu ở trên. Ví dụ, trong tệp truy vết NAT_home_side, hãy xem xét thông điệp GET từ máy khách đến máy chủ tại thời điểm 1.572315 và GET tại thời điểm 7.573305. Hãy nghiên cứu việc sử dụng hai thông điệp HTTP này và viết một đoạn giải thích nửa trang về mục đích của mỗi thông điệp này.