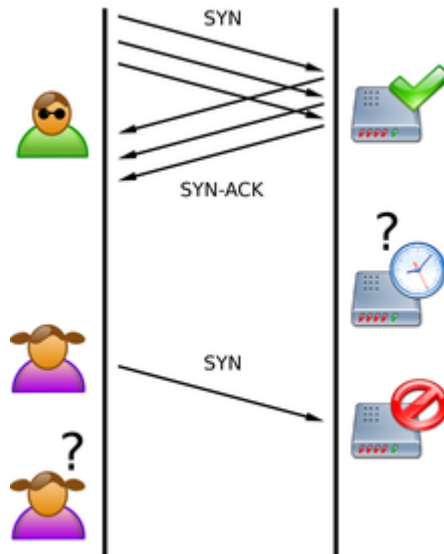


# DDoS 공격 종류 및 대응방안

## 1 TCP SYN Flooding Attack



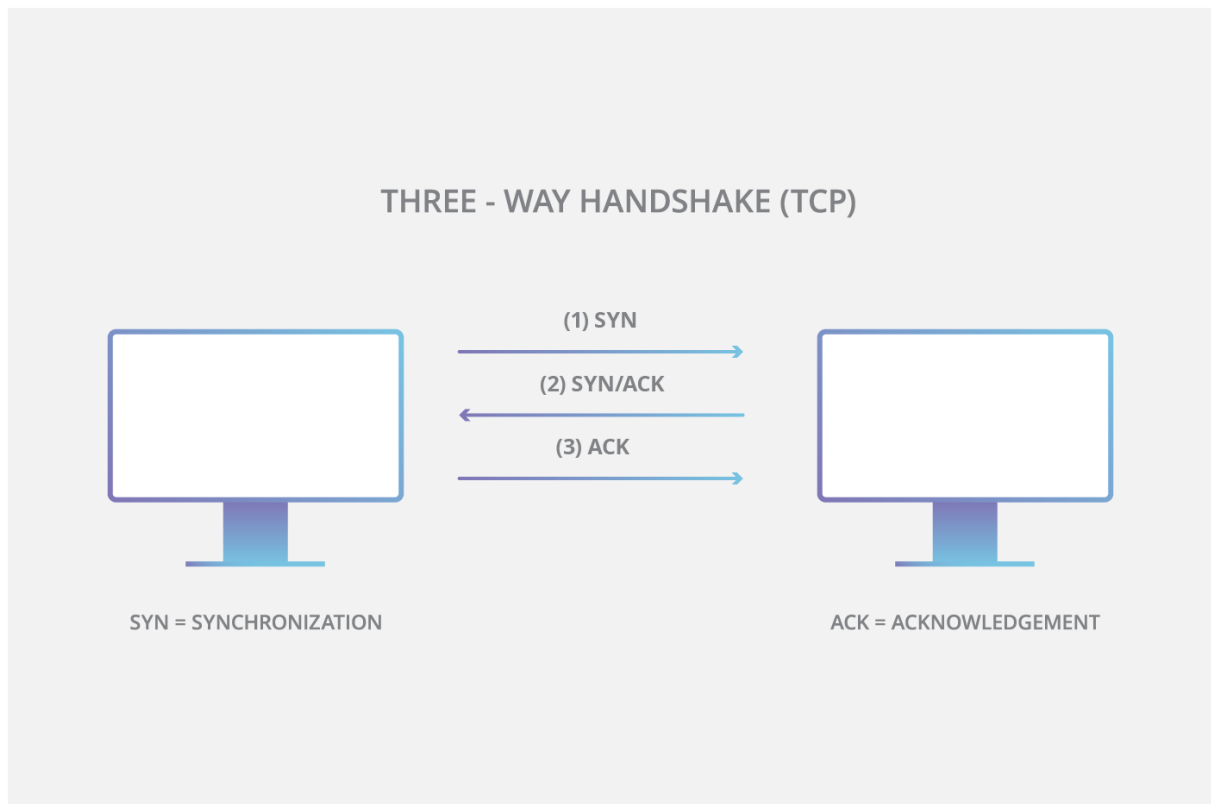
- TCP 패킷의 SYN 비트를 이용한 공격 방법으로 너무 많은 연결 요청을 전송해서 대상 시스템이 Flooding하게 만들어 대상 시스템의 서비스를 중단시키는 공격
- 다른 사용자가 서비스를 받지 못하도록 하는 공격
- TCP 초기 연결 과정 이용, SYN 패킷을 요청하여 서버가 ACK 및 SYN 패킷을 전송하게 된다.
- 전송하는 주소가 무의미한 주소이며, 서버는 대기 상태이고 대량의 요청 패킷 전송으로 서버의 대기 Queue가 가득 차서 DoS 상태가 된다.

### ▶ TCP SYN Flooding 공격 방식

SYN 폭주 공격은 TCP 연결의 HandShake 프로세스를 이용하는 방식으로 작동합니다. 정상적인 조건에서 TCP 연결은 연결을 위한 세 가지 다른 프로세스를 보여준다.

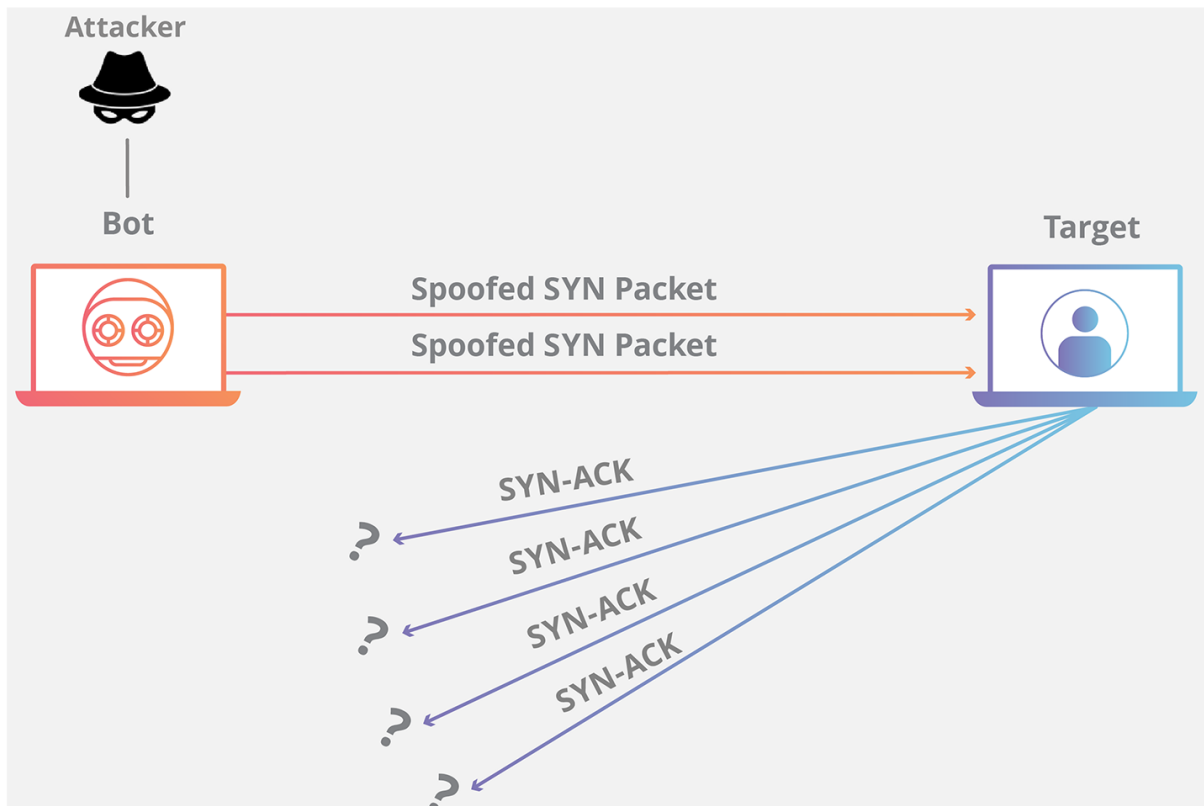
1. 첫째, 연결을 시작하기 위해 클라이언트가 SYN 패킷을 서버로 전송한다.

2. 그런 후에, 서버는 이 통신을 승인하기 위해 SYN/ACK 패킷으로 해당 초기 패킷에 응답한다.
3. 마지막으로 클라이언트는 서버로부터 패킷의 수신을 승인하기 위해 ACK 패킷을 반환한다. 이러한 패킷의 전송 및 수신 순서를 완료하고 나면 TCP 연결이 열리고 데이터를 전송하고 수신할 수 있다.



서비스 거부를 생성하기 위해 공격자는 초기 SYN 패킷이 수신된 후 서버가 하나 이상의 SYN/ACK 패킷으로 다시 응답하고 HandShake의 마지막 단계를 기다린다는 사실을 이용한다. 작동 방식은 다음과 같다.

1. 공격자는 종종 스푸핑된 IP 주소를 사용하여 표적 서버로 대량의 SYN 패킷을 전송한다.
2. 그런 후에 서버가 각 연결 요청에 응답하고 해당 응답을 수신할 준비가 되어 있는 열린 포트를 남겨둔다.
3. 도착하지 않는 마지막 ACK 패킷을 서버가 기다리는 동안 공격자는 계속해서 더 많은 SYN 패킷을 전송한다. 각각의 새 SYN 패킷이 도착하면 서버는 일정 시간 동안 새 개방 포트 연결을 일시적으로 유지하고 일단 사용 가능한 모든 포트를 사용하고 나면 이 서버는 정상적으로 기능할 수 없게 된다.



- 네트워킹에서는 한 서버가 어떤 연결을 열린 상태로 두고 있는데 이 연결의 다른 쪽에 있는 컴퓨터가 그렇지 않은 경우 이 연결을 반개방 상태인 것으로 간주한다.
- 이러한 유형의 DDoS 공격에서는 포트를 다시 사용할 수 있게 되기 전에 표적 서버가 열린 연결을 지속적으로 유지하고 각 연결 시간이 초과 되기를 기다린다. 그 결과는 이러한 유형의 공격을 "반개방 공격"으로 간주할 수 있다는 것이다.

#### ▶ TCP SYN Flooding 대응 방안

대응 방안	내용
방화벽에서 대응	IP 당 SYN 요청에 대한 PPS(Packet Per Second) 임계치를 단계적으로 조정
First SYN Drop (Spoofed) 설정	- SYN 패킷을 전송한 클라이언트의 존재 여부를 파악하여 차단하는 방법 - 클라이언트에서 전송된 첫 번째 SYN을 DROP하여 재요청 여부를 확인 후 Spoofing 여부를 판단
TCP 세션 연결 차단	- 트래픽 유형별 임계치를 조정하여 TCP 세션 연결에 대한 차단
Back Queue 증가	- 임시적 방법으로 서버의 Queue 사이즈를 증가시킴 - <code>sysctl -w net.ipv4.tcp_max_syn_backlog=1024</code>

#### ➡ 방화벽에서 대응

항목	PPS	BPS	차단 시간	설정
SYN (1:1)	120		300 초	<input checked="" type="checkbox"/> 사용

- 위의 그림 처럼 한 IP 당 Syn의 PPS를 설정하여 임계치를 초과한 경우 차단을 한다.

#### ➡ Back queue 증가

```
root@kali:/etc/init.d# sysctl -a | grep syn_backlog
net.ipv4.tcp_max_syn_backlog = 128
root@kali:/etc/init.d# sysctl -w net.ipv4.tcp_max_syn_backlog=1024
net.ipv4.tcp_max_syn_backlog = 1024
```

- Back Log Queue를 증가시켜 Syn\_recv에 의한 대기를 많이 할 수 있도록 지정한다.
- #sysctl -a | grep syn\_backlog #sysctl -w net.ipv4.tcp\_max\_syn\_backlog=1024

#### ➡ First SYN Drop (Spoofed) 설정

- 클라이언트로부터 전송된 첫 번째 SYN은 Drop하여 재요청 패킷이 도착하는지 확인하여 출발지 IP가 위조(Spoofing)되었는지 판단
- 대부분의 공격 툴이 다량의 SYN 요청을 생성할 뿐, 패킷 Drop시 재전송을 수행하지 않는다는 점을 이용한 방법

#### ➡ TCP 세션 연결 차단

##### ▶ 라우터에서 대응 방법

구분	내용
Watch Mode	- SYN 패킷을 통과시키고 일정 시간 동안 연결이 이루어지지 않으면 라우터가 SYN 패킷을 차단
Intercept Mode	- 라우터에 유입되는 SYN 패킷 요청을 서버로 전송하지 않고, 라우터에서 가로채어 SYN 패킷을 요청한 클라이언트와 서버를 대신 연결하는 것

##### ▶ intercept Mode (Default Mode)

```
ip tcp intercept mode intercept
```

처음 클라이언트가 서버로 세션을 맺기위해 3 way HandShaking을 시작하면  
중간에 라우터가 마치 서버인양 클라이언트와 세션을 맺는다.

정상적으로 수행되면 이번엔 라우터가 클라이언트인양 진짜 서버와 3 way HandShaking을 통해 세션을 맺는다.

정상적으로 수행되었을 시, 클라이언트가 서버로 Push, Ack 플래그를 셋팅시켜 보내고 서버로부터 Ack를 받은 수 이후 과정을 수행한다.

희생자(서버)-----R1\_FW(firewall)-----  
--공격자(클라이언트)

----->syn ----->  
<-----ack,syn <-----  
-----> ack ----->

<----- syn <-----  
-----> ack syn----->  
<----- ack <-----

<-----  
-----PUSH ACK  
ack-----  
----->

## 2.WATCH Mode

```
ip tcp intercept mode watch
```

희생자(서버)-----R1\_FW(firewall)-----공격자(클라이언트)

```

<-----
-----syn

syn ack-----
----->

<-----
----- ack

```

이렇게 일반적인 세션수립과정을 그냥 보기만 하는데, 마지막에 ack가 클라이언트로부터 발생하지 않으면 R1이 RST를 서버에게 보내서 세션을 끊는다.

## 2 TCP Flag Flooding Attack

TCP의 Flag 값을 임의로 조작하면 SYN, ACK, FIN, RST과 같이 여러 형태의 패킷을 생성할 수 있으며, 서버는 이러한 패킷을 수신하는 경우 해당 패킷을 검증하기 때문에 서버의 자원을 소진시킴

※ ACK Flooding : 공격자가 TCP 세션이 없는 상태에서 TCP 헤더의 Flags를 ACK(0x10)으로 Setting하여 무작위로 보내면 수신측에서 변조된 발신 IP로 RST 패킷을 무작위로 보내게 되고, 동시에 ICMP host Unreachable 패킷을 보내면서 수신측 시스템의 과부하를 초래하는 공격임

※ RST Flooding : 공격대상 서버로 전달되는 클라이언트의 TCP 패킷의 Reset 값을 설정하여 클라이언트가 서버로부터 정상적인 서비스를 받지 못하도록 TCP 연결을 강제로 종료시키는 공격

### 나. 대응방안

- SYN 이외의 Flooding 공격을 방어하기 위해서는 다음과 같은 다양한 형태의 DDoS 방어 기법을 사용
- 먼저 정상적인 TCP 세션 연결 이후 정상적인 트랜잭션(Transaction)이 수행되는지에 대해 검증하여 만약 정상적인 트랜잭션이 이루어질 경우

서버로 전달해야 하고, 정상적인 트랜잭션 없이 TCP 세션 연결만 수행할 경우에는 서버로 전달하지 않아야 서버에 부하 증가 현상을 막을 수 있음.

- 또한 네트워크의 정상적인 환경에서 설정된 각 트래픽 유형별 임계치를 통하여 과도한 TCP 세션 연결에 대해 차단

### 3 TCP Session Attack

#### 가. 공격기법

▶ TCP 3-Way Handshake 과정을 과도하게 유발함으로써 서비스의 과부하를 유발하는 공격 유형으로 ① TCP 세션 연결을 유지하는 DDoS 공격, ② TCP 세션 연결/해제를 반복하는 DDoS 공격, ③ TCP 세션 연결 후 정상적인 트랜잭션(Transaction)처럼 보이는 트래픽을 발송하는 DDoS 공격으로 구분

#### 나. 대응방안

- (방안 1) Connection Timeout/Keep-Alive/Time-Wait 설정을 통한 차단  
Connection Timeout에 설정된 시간동안 Client와 웹서버 사이에 데이터신호의 이동이 전혀 없을 경우 Connection을 종료하도록 설정하거나 웹서버에서 keepalive 기능을 사용하는 경우에는 Keepalivetimeout을 사용하여 세션 공격을 차단

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

httpd.conf 의 timeout 설정을 통한 세션공격 차단

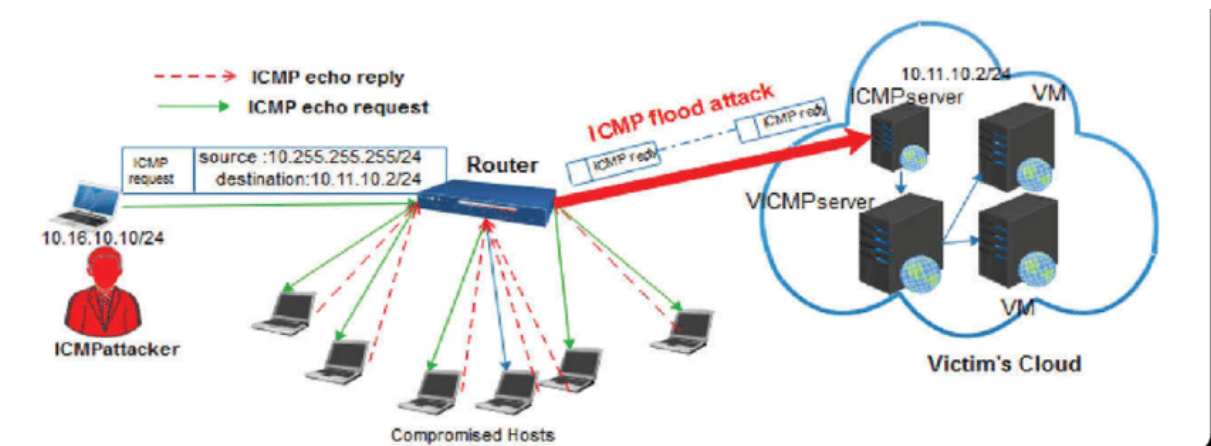
다만, 공격자는 방어 정책 우회를 위해 Content-Length와 실제 전송하는 데이터의 크기를 조정하고 데이터 전송 term을 짧게 가져가면서도 Connection을 오랫동안 유지할 수 있으므로 이 대응 방안은 일정한 한계가 있음

- (방안 2) L7 스위치의 임계치 설정 기능을 이용한 차단  
L7 스위치를 운영하는 경우, IP당 Connection Limit을 설정하여 하나의 Client와 Server가 맺을 수 있는 Connection 수치를 조절하여 차단

```
when RULE_INIT {
    set :: max_connections_per_ip 200
    array set :: active_clients {}
    array set write_client {
        10.31.0.230
        10.0.0.5
    }
}
```

L7 스위치의 Connection Limit 설정 스크립트

#### 4 ICMP Flooding Attack



- IP 특정(Broadcast 주소 방식)과 ICMP 패킷을 이용한 공격 방법이다.
- 통신을 위해서 서비스 및 포트가 필요없는 유일한 프로토콜이다.
- Smurfing Attack이라고도 한다.
- 다수의 호스트가 존재하는 서브 네트워크에 ICMP Echo 패킷을 Broadcast로 전송 (Source Address는 공격 대상 서버로 위조)한다. 이에 대한 다량의 응답 패킷이 공격 대상 서버로 집중되게 하여 마비시키는 공격이다.

#### ▶ ICMP 공격에 사용되는 메시지

메시지	내용
-----	----



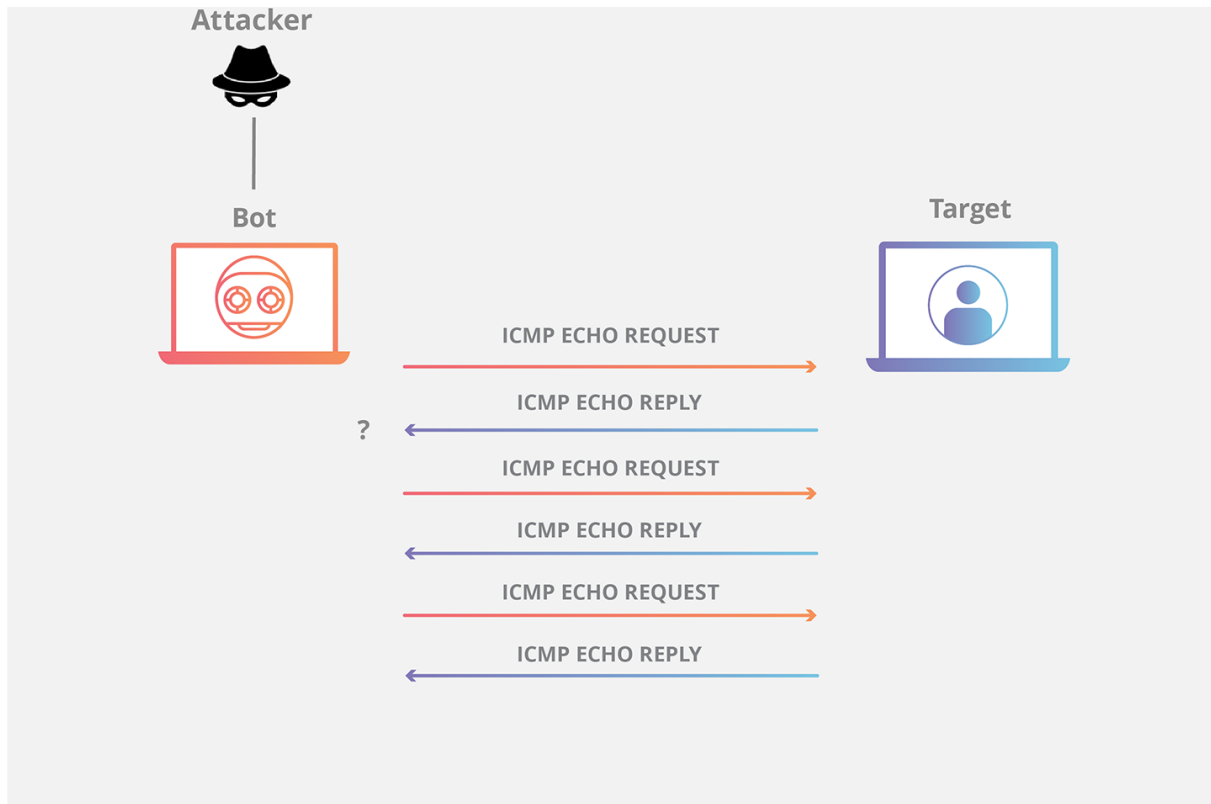
Source Quench(Type-4)	- 사용 중 전송자에게 패킷 전송 속도를 줄여 줄 것을 요구하는 메시지로 전송 속도의 지연 발생
Time to live exceeded in Transit(Type-11, Code-0)	- 시간 초과로 패킷이 폐기되었기 때문에 재전송함
Destination unreachable(Type-3, Code-0, 1, 2, 3)	- ICMP 트래픽 처리에 자원을 사용하게 되므로 시스템이 조금씩 느려지는 현상 발생

### ▶ ICMP Flooding 공격 방식

- ICMP Flooding 공격에 사용되는 인터넷 제어 메시지 프로토콜은 네트워크 장치가 통신하는 데 사용하는 인터넷 계층 프로토콜이다.
- 네트워크 진단 도구 traceroute 및 ping은 모두 ICMP를 사용하여 작동한다.
- 일반적으로 ICMP 에코 요청 및 에코 응답 메시지는 장치의 상태 및 연결과 발신자와 장치 간의 연결을 진단하기 위해 네트워크 장치를 ping하는 데 사용된다.
- ICMP 요청에는 각 요청을 처리하고 응답을 보내기 위해 일부 서버 리소스가 필요하다.
- 또한 요청에는 들어오는 메시지(에코 요청)와 나가는 응답(에코 응답) 모두에 대한 대역폭이 필요하다.
- ICMP Flooding 공격은 많은 수의 요청에 응답하는 대상 장치의 능력을 압도하거나 가짜 트래픽으로 네트워크 연결을 과부하시키는 것을 목표로 한다.
- 봇넷이 된 많은 장치가 ICMP 요청을 사용하여 동일한 인터넷 자산 또는 인프라 구성 요소를 대상으로 하면 공격 트래픽이 크게 증가하여 정상적인 네트워크 활동이 중단될 가능성이 있다.
- 역사적으로 공격자는 전송 장치를 마스킹하기 위해 가짜 IP 주소를 스푸핑하는 경우가 많았다.
- 최신 봇넷 공격의 경우, 악의적인 행위자는 봇의 IP를 마스킹할 필요성을 거의 느끼지 못하며, 대신 스푸핑되지 않은 봇의 대규모 네트워크에 의존하여 대상의 용량을 포화시킨다.

DDoS 형태의 ping(ICMP) 폭주는 2가지 반복 단계로 나눌 수 있다.

1. 공격자는 여러 장치를 사용하여 많은 ICMP 에코 요청 패킷을 대상 서버로 보낸다.
2. 그런 다음 대상 서버는 ICMP 에코 응답 패킷을 각 요청 장치의 IP 주소에 응답으로 보낸다.



ICMP Flooding의 피해 결과는 대상 서버에 대한 요청 수에 정비례하다. NTP 증폭 및 DNS 증폭과 같은 반사 기반 DDoS 공격과 달리 ping 폭주 공격 트래픽은 대칭적이다. 대상 장치가 수신하는 대역폭의 양은 단순히 각 봇에서 전송된 총 트래픽의 합계다.

#### ▶ ICMP Flooding 방어 대책

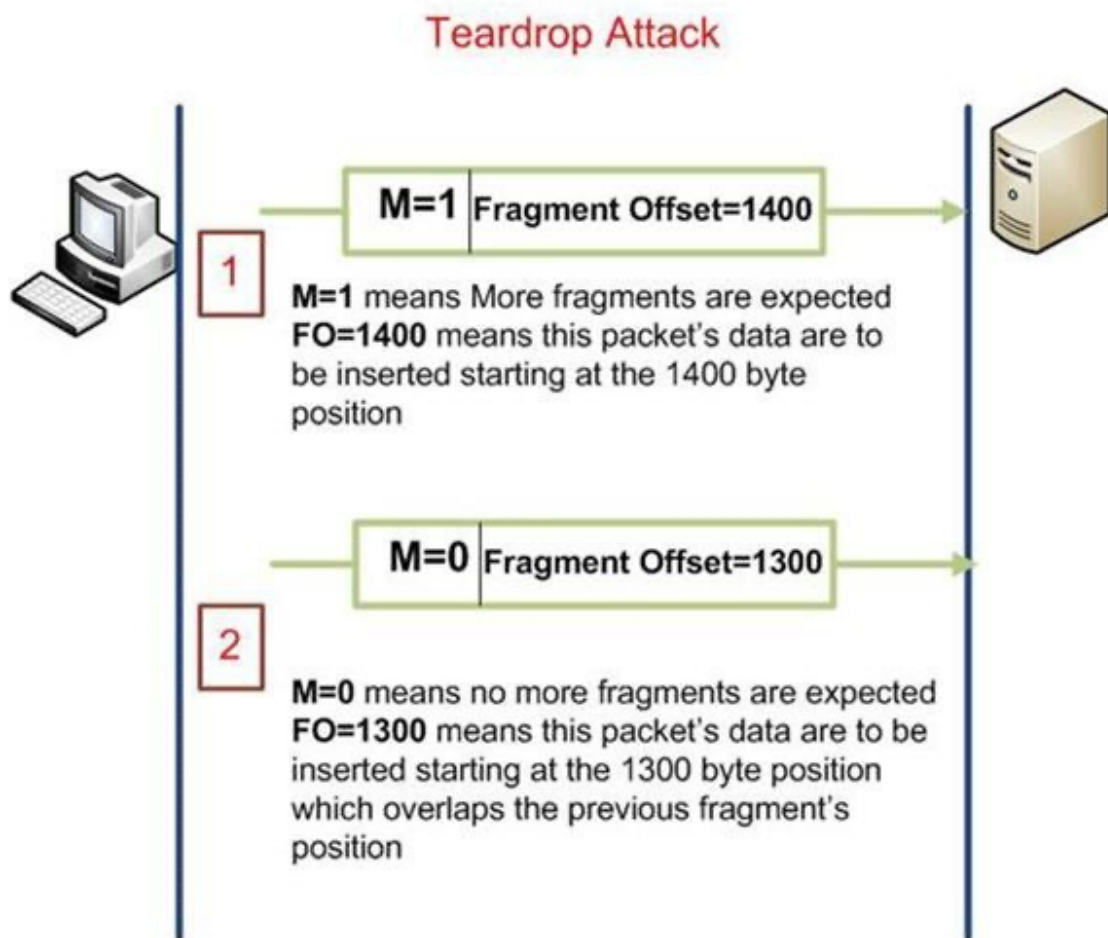
- 미사용 UDP 포트 차단
- 방화벽 패킷 필터링

#### ▶ ICMP Flooding 대응방안

구분	내용
ICMP 기능 비활성화	- 방화벽을 설정함으로써 표적 디바이스가 ICMP를 사용해 요청을 송수신하지 못하도록 차단하고 디바이스의 ICMP 기능을 비활성화할 수 있다. (디바이스가 다른 정상적인 ping 요청, 추적 요청, 네트워크 활동에 응답하지 않게 되어 관리자가 서비스 문제를 진단할 수 있는 능력이 제한되는 단점 존재.)
전송률 제한	- 수신 ICMP 메시지 처리에 대한 전송률 제한을 설정하거나 ping 요청의 허용 크기를 제한해 ping 플러드 공격을 방어할 수도 있다.
침입 탐지	- IDS(Intrusion Detection System)는 네트워크 트래픽을 모니터링하고 잠재적인 공격을 실시간으로 식별할 수 있다.

네트워크 트래픽 모니터링	- 네트워크 트래픽을 지속적으로 모니터링하고 분석해 정상적인 트래픽 패턴은 물론, 핑 플러드와 같은 공격의 징후일 수 있는 비정상상을 식별할 수 있다.
DDoS 방어 배포	- 포괄적인 DDoS 방어는 악성 트래픽이 네트워크에 도달하기 전에 필터링해 핑 플러드 공격과 기타 DDoS 공격이나 사이버 공격을 방어하는 데 도움이 될 수 있습니다.

## 5 Tear Drop : IP Fragmentation(Ping of Death)



- 네트워크 패킷은 MTU보다 큰 패킷이 오면 분할하고 분할된 정보를 flag와 offset이 가지고 있다.
- 이때 offset을 임의로 조작하여 다시 조립될 수 없도록 하는 공격이다.
- Fragment를 조작하여 패킷 필터링 장비나 IDS를 우회하여 서비스 거부를 유발한다.

## ▶ Tear drop 공격 종류

종류	설명
Tiny Fragment	- 최초의 Fragment를 아주 작게 만들어서 네트워크 침입탐지 시스템이나 패킷 필터링 장비를 우회하는 공격
Fragment Overlap	- Tiny Fragment 공격 기법에 비해 더욱 정교한 방법 - IDS의 Fragment 처리 방법과 패킷 필터링의 재조합과 Overwrite 처리를 이용
IP Fragmentation을 이용한 서비스 거부 공격	- Ping of Death : Ping을 이용하여 ICMP 패킷을 규정된 길이 이상으로 큰 IP 패킷을 전송, 수신 받은 OS에서 처리하지 못함으로써 시스템을 마비시키는 공격 - Tear Drop : fragment 재조합 과정의 취약점을 이용한 공격으로 목표 시스템 정지나 재부팅을 유발하는 공격, TCP Header 부분의 offset field 값이 중첩되는 데이터 패킷을 대상 시스템에 전송

## ▶ Tear drop 대응 방법

대응 방안
- 시스템의 운영체제가 취약점을 갖지 않도록 패치 해야 함
* 원인은 윈도우 및 Linux 시스템의 IP 패킷 재조합 코드의 버그에 있었으나 현재 대부분의 시스템에서는 이러한 Tear drop 공격에 대해서 방어하고 있음

## ▶ Ping of Death의 공격 방법

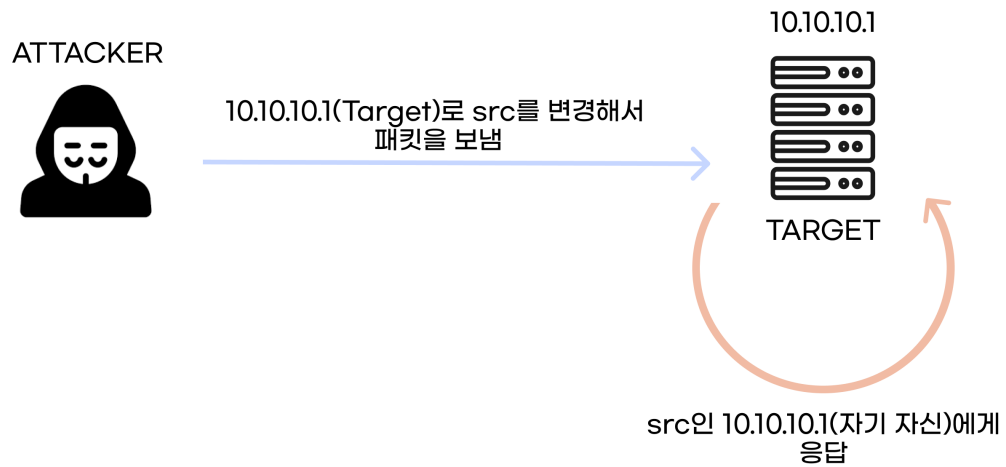
- Ping of Death는 하나의 큰 패킷을 전송하면 패킷은 MTU의 크기를 넘을 수 없기 때문에 분할되어서 응답받게 된다.

## ▶ Ping of Death의 대응 방안

- 패킷의 재조합 과정에서, 들어오는 패킷의 offset 값의 합을 검사

## 6 Land Attack

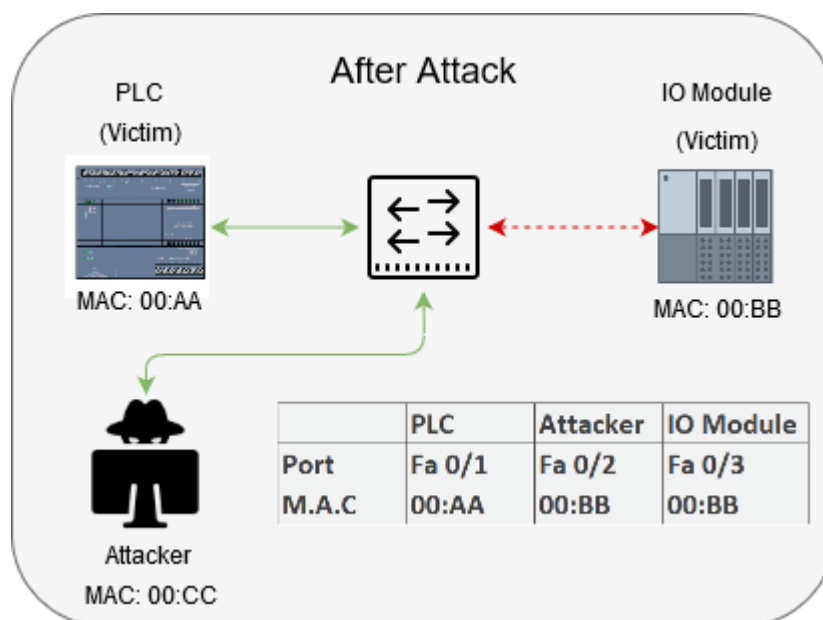
- IP Header를 변조하여 인위적으로 송신자 IP 주소 및 Port 주소를 수신자의 IP 주소와 Port 주소로 설정하여 트래픽을 설정하여 트래픽을 전송하는 공격 기법이다.
- 송신자와 수신자의 IP 주소와 Port 주소가 동일하기 때문에 네트워크 장비에 부하를 유발한다.



#### ▶ Land Attack 대응 방안

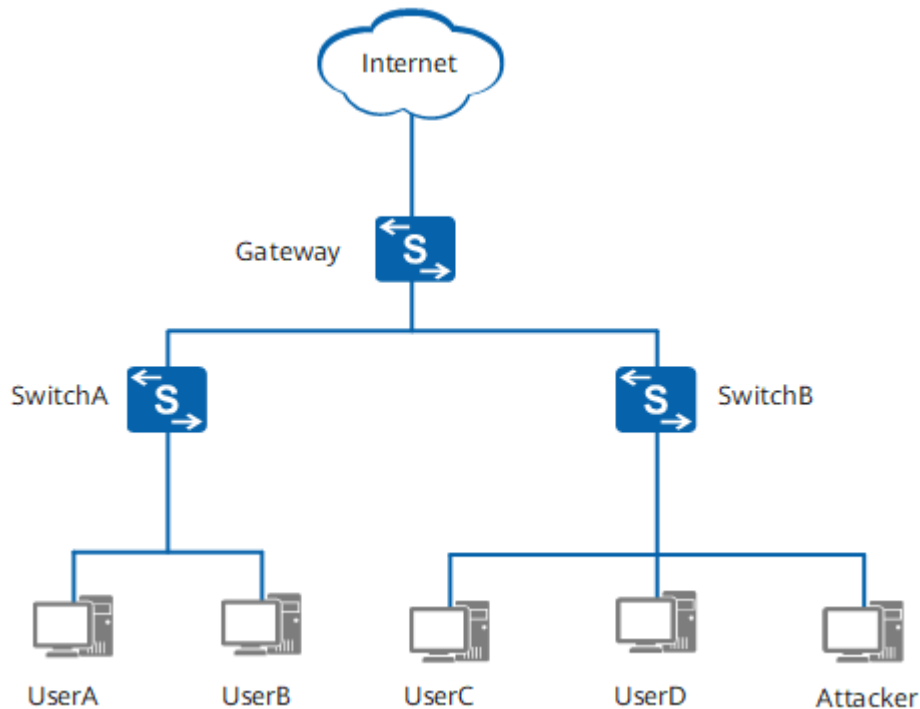
- 송신자와 수신자의 IP 주소가 동일한 패킷을 삭제한다.

### 7 ARP Flooding Attack



- ARP Flooding은 네트워크 상에서 ARP 요청을 계속해서 보내는 것을 의미
- 이를 통해 네트워크 상의 다른 컴퓨터를 공격하거나 서비스 거부 공격을 수행

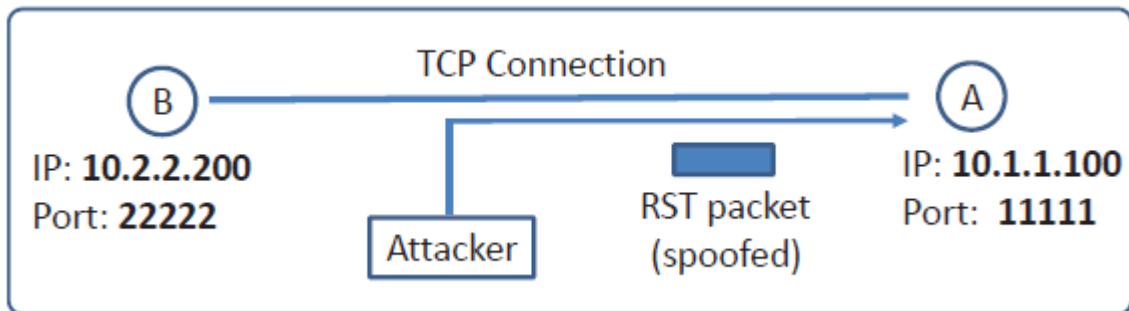
#### ▶ ARP Flooding 해결 방안



- ARP 패킷의 속도 제한이 배포된 후, 게이트웨이는 수신된 ARP 패킷에 대한 통계를 수집한다. 지정된 기간 내에 수신된 ARP 패킷의 수가 임계값(최대 ARP 패킷 수)을 초과하는 경우, 게이트웨이는 CPU 오버로드를 방지하기 위해 초과 ARP 패킷을 폐기한다.
- ARP 미스 메시지의 속도 제한이 배포된 후, 게이트웨이는 ARP 미스 메시지에 대한 통계를 수집한다. 지정된 기간 내에 생성된 ARP 미스 메시지의 수가 임계값(ARP 미스 메시지의 최대 수)을 초과하면, 게이트웨이는 초과 ARP 미스 메시지를 트리거하는 IP 패킷을 폐기한다. 이것은 게이트웨이가 해결할 수 없는 IP 주소로 많은 수의 IP 패킷을 처리할 때 CPU 과부하를 방지한다.
- 엄격한 ARP 학습이 배포된 후, 게이트웨이는 보낸 ARP 요청 패킷에 대한 응답으로 ARP 응답 패킷만 학습한다. 이 조치는 게이트웨이가 많은 ARP 패킷을 처리할 때 게이트웨이의 ARP 항목이 소진되는 것을 방지한다.
- ARP 엔트리 제한이 배포된 후, 게이트웨이는 각 인터페이스에서 동적으로 학습된 ARPentries의 수를 제한한다. 인터페이스에서 동적으로 학습한 ARP 항목의 수가 최대 수에 도달하면 더 이상 동적 항목을 추가할 수 없다. 이것은 인터페이스에 연결된 호스트가 게이트웨이를 공격할 때 ARP 항목이 소진되는 것을 방지한다.

## 8 RST Flooding Attack

- 공격 대상 서버로 전달되는 클라이언트의 TCP 패킷의 Reset 값을 설정하여 클라이언트가 서버로부터 정상적인 서비스를 받지 못하도록 TCP 연결을 강제로 종료시키는 공격이다.



Version	Header length	Type of service	Total length	
Identification			Flags	Fragment offset
Time to live		Protocol	Header checksum	
Source IP address: 10.2.2.200				
Destination IP address: 10.1.1.100				
Source port: 22222			Destination port: 11111	
Sequence number				
Acknowledgement number				
TCP header length		URG	ACK	PSH
		RST	SYN	FIN
Checksum			Window size	
			Urgent pointer	

### ▶ RST Flooding 탐지 방법

- ① 공격자가 보낸 Packet에서 IP Header, TCP Header를 분석한다.
- ② TCP Header 분석 결과 flag 가 ACK, FIN, RST 등 이면, 공격자가 보내는 Packet 의 수를 Count 한다.

③ Count 가 공격인정시간 내에 공격인정회수 이상이면 TCP Flag Flooding 으로 탐지한다.

#### ▶ RST Flooding 대응 방안

- TCP 프로토콜이 연결 지향성(Connection-Oriented)이라는 점을 이용한 공격 방어 기법 활용
- 정상적인 TCP 의 경우에는 TCP 3-Way Handshake 과정을 거쳐야만 한다. 하지만, 이러한 Handshake 과정에 위배된 TCP 패킷이 유입될 경우 비정상적인 트래픽으로 구분하여 차단하는 방법이 가장 효과적
- 또한 정상적인 트래픽의 임계치에 의거하여 비정상적으로 많은 트래픽을 유발하는 출처 IP 에 대해 차단하는 임계치 기반의 DDoS 방어 기법도 효과적인 대응 기법

### 9 ACK Flooding Attack

- DOS 공격의 한 방법으로, 서버가 정상적인 서비스를 지연 또는 불능상태로 만들기 위한 공격으로 사용된다.
- 이는 TCP 프로토콜을 이용하여, 클라이언트가 서버에 TCP Header의 Flags를 ACK(0x10)으로 Setting하여 대량의 Packet을 보내면, 서버는 이를 처리하기 위해서 대부분의 자원(System Resource)을 소모하게 되고, 정상적인 서비스를 하지 못하는 현상이 발생한다

#### ▶ ACK Flooding 탐지 방법

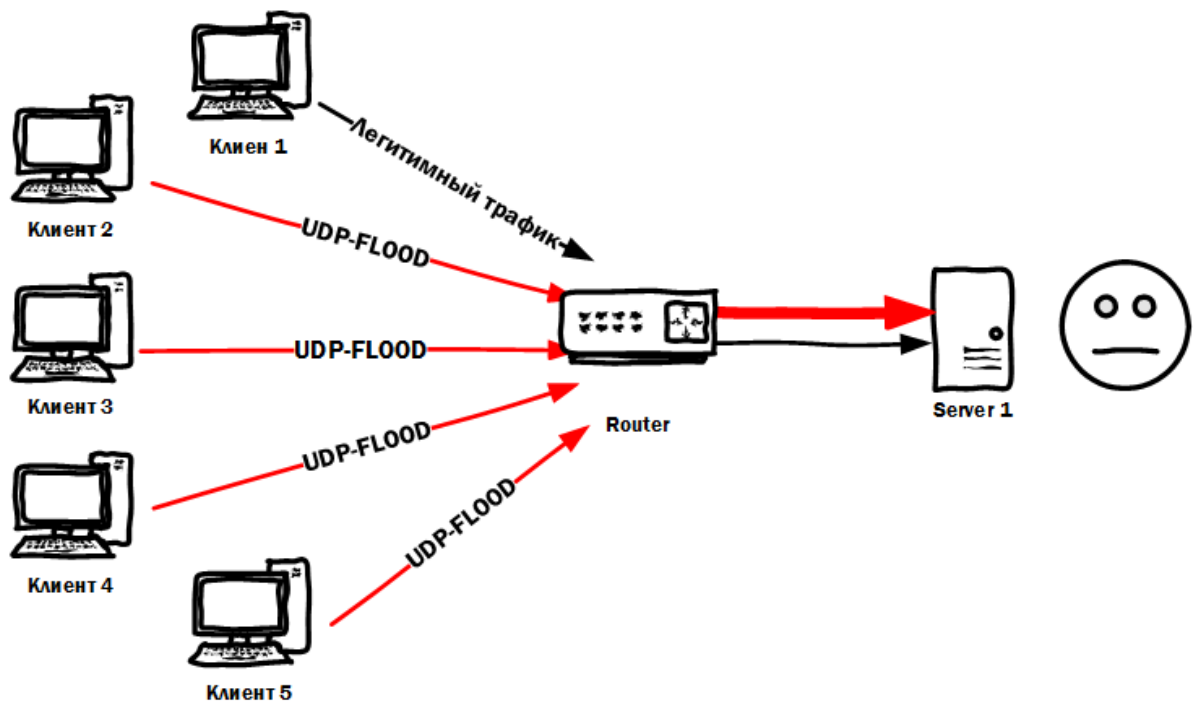
- ① 공격자가 보낸 Packet에서 IP Header, TCP Header를 분석한다.
- ② TCP Header분석결과 flag가 ACK 이면, 공격자가 보내는 Packet의 수를 Count한다.
- ③ Count가 공격인정 시간내에 공격인정회수 이상이면 ACK Flooding으로 탐지한다.

#### ▶ ACK Flooding 대응 방안

- 이러한 사실을 Network 및 System 관리자에게 주지시키고, 공격자의 주소에 대해서 F/W 또는 기타 보안장비에서, 이를 Filtering함으로써 공격의 피해를 최소화 할 수 있다.

### 10 UDP Flooding





- 많은 수의 사용자 데이터그램 프로토콜(UDP)패킷을 대상 서버로 전송하여 해당 장치의 처리 및 응답 능력을 압도하려는 서비스 거부 공격의 한 유형이다.
- .대상 서버를 보호하는 방화벽도 UDP 폭주의 결과로 고갈되므로 합법적인 트래픽에 대한 서비스 거부가 발생할 수 있다.

#### ▶ UDP Flooding 공격 작동 방식

- UDP Flooding는 주로 서버가 포트 중 하나로 전송된 UDP 패킷에 응답할 때 수행하는 단계를 악용하여 작동한다.
- 정상적인 조건에서 서버가 특정 포트에서 UDP 패킷을 수신하면 그에 대한 응답으로 두 단계를 거친다.
  1. 서버는 먼저 지정된 포트에서 현재 요청을 수신 대기하는 프로그램이 실행 중인지 확인한다.
  2. 이 포트에 패킷을 수신하는 프로그램이 없으면, 서버는 대상에 도달할 수 없음을 알리기 위해 ICMP(ping) 패킷으로 발신자에게 응답한다.

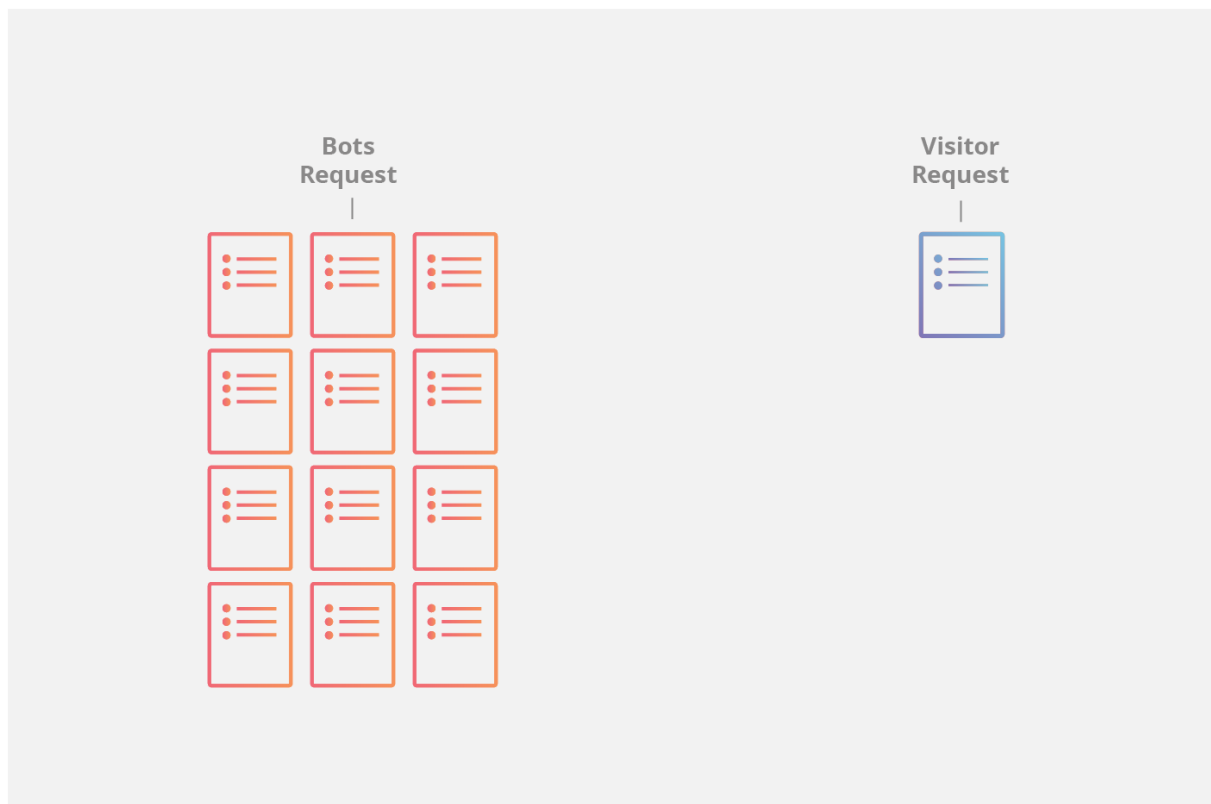
#### **UDP 폭주는 호텔 리셉셔니스트가 통화를 라우팅하는 맥락에 빗대어 생각을 해보자**

- 1) 먼저 리셉셔니스트는 발신자가 특정 방에 연결해달라고 요청하는 전화를 받는다.

2) 그런 다음 리셉셔니스트는 모든 객실 목록을 살펴보고 손님이 객실에 있고 전화를 받을 의향이 있는지 확인해야 한다.

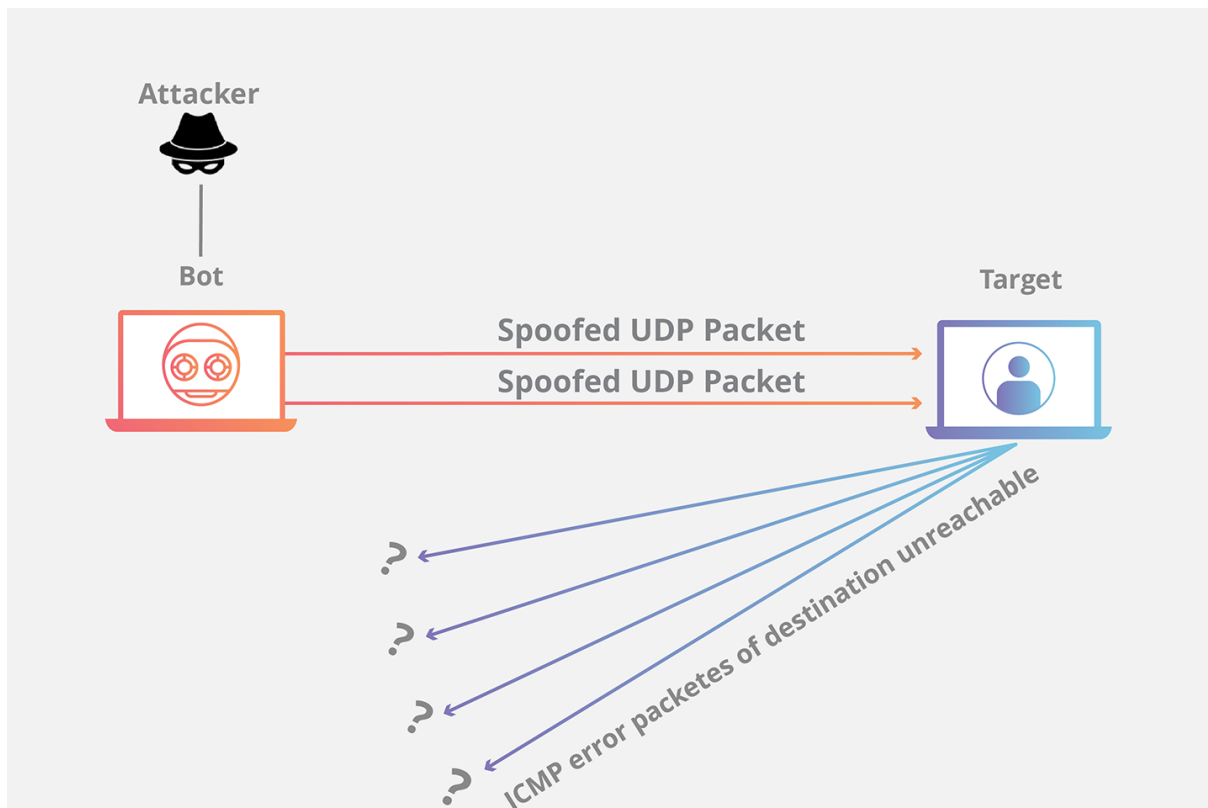
3) 리셉셔니스트는 손님이 어떤 전화도 받지 않는다는 것을 알게 되면 수화기를 다시 들고 발신자에게 손님이 전화를 받지 않을 것이라고 이야기해야 한다.

4) 갑자기 모든 전화선에 그와 유사한 요청으로 동시에 불이 들어오면 모든 전화선이 빠르게 압도된다.



- 서버에서 각각의 새 UDP 패킷을 수신하면 프로세스에서 서버 리소스를 활용하여 요청을 처리하기 위해 단계를 거친다.
- UDP 패킷이 전송되면 각 패킷에는 소스 디바이스의 IP 주소가 포함된다.
- 이러한 유형의 DDoS 공격 중에 공격자는 일반적으로 자신의 실제 IP 주소를 사용하지 않고 대신 UDP 패킷의 소스 IP 주소를 스푸핑하여 공격자의 실제 위치가 노출되고 대상 서버의 응답 패킷으로 포화되는 것을 방지한다.
- 목표물이 된 서버가 수신한 각 UDP 패킷을 확인한 다음에 응답하기 위해 리소스를 활용하므로 UDP 패킷을 대량으로 수신한다면 목표 서버의 리소스가 빠르게 소진되어 정상

트래픽에 대한 서비스 거부가 발생할 수 있다.



▶ UDP Flooding 방어 대책

- 미사용 UDP 포트 차단
- 방화벽 패킷 필터링

▶ UDP Flooding 대응 방안

- ACL (Access Control List) 설정을 이용한 차단  
웹서버 혹은 운영 장비에 대한 접근 제어 목록에 차단하고자 하는 프로토콜 정보를 다음과 같이 ACL에 UDP/ICMP DROP 정보로 설정하여 차단

```
ip access-list extended acl-Drop-Example
seq 1 deny udp any any
seq 2 deny icmp any any
seq 3 permit ip 1.1.1.0/24 any
seq 4 permit ip 2.2.2.2/29 any
seq 5 permit tcp any any
seq 6 permit ip 3.3.3.3.0/24 any
seq 7 permit ip 4.4.4.0/24 any
seq 8 permit icmp host 5.5.5.5 any
```

- INBOUND 패킷에 대한 임계치 설정을 이용한 차단
- 운영 장비로 유입되는 INBOUND 패킷을 기준으로 PPS(Packet Per Second) 수치를 유입되는 수치보다 낮게 설정(예: 10) 하여 임계치 이상의 UDP/ICMP 트래픽의 유입을 차단

※ UDP/ICMP 공격 방어를 위한 1단계/2단계 정책은 UDP 및 ICMP를 이용한 서비스를 제공하지 않는 경우에 적용할 수 있으므로 적용시 주의가 요구됨

※ PPS 제안 수치는 서비스에 지장을 주지 않는 범위 내에서 단계적으로 조정하여 설정할 수 있음

## **1 1 Bonk / Boink Attack**



- Teardrop 공격과 같은 TCP 프로토콜의 허점을 이용한 공격이다.
- 데이터 전송에서 신뢰성 있는 연결을 제공하려면, 다음을 확인하는 기능을 기본으로 제공한다.

•

\*패킷의 순서가 올바른가?

\*중간에 손실된 패킷은 없는가?

- 이러한 사항이 확인되지 않는 데이터 전송에 대해 프로토콜은 신뢰도를 확보하고자 반복적으로 재요구하고 수정하게 된다.
- Boink, Bonk, Teardrop은 공격 대상이 반복적인 재요구와 수정을 계속하게 함으로써 시스템 자원을 고갈 시키는 공격이다.

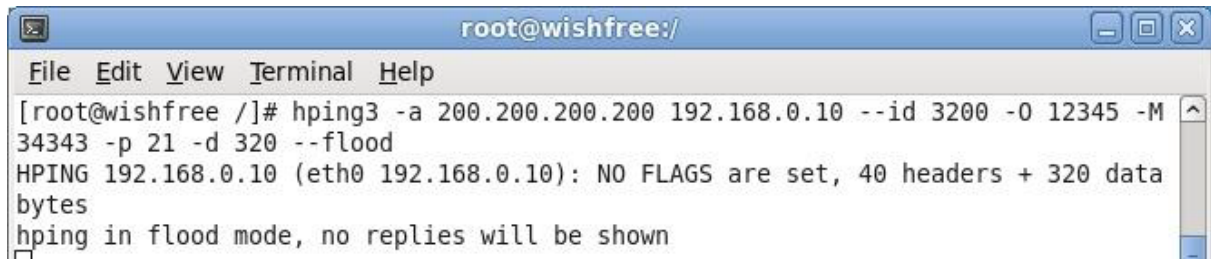
ex) 은행이나, 병원에 가면 접수 대기를 위해서 번호표를 뽑음

하지만 기계가 고장이거나 계속 1번만 나오게된다면 어떤 상황이 생길까?

→ 은행 업무에 아주 큰 차질이 생김(해커는 이처럼 은행에서 기계를 고장내거나 다른 번호가 나오는 기계를 은행 몰래 설치함)

## ▶ Bonk Attack

- 처음 패킷을 1번으로보낸 후, 두번째, 세번째 패킷 모두 시퀀스 넘버를 1번으로 조작해서 보냄



```

root@wishfree:/
File Edit View Terminal Help
[root@wishfree /]# hping3 -a 200.200.200.200 192.168.0.10 --id 3200 -O 12345 -M 34343 -p 21 -d 320 --flood
HPING 192.168.0.10 (eth0 192.168.0.10): NO FLAGS are set, 40 headers + 320 data bytes
hping in flood mode, no replies will be shown
  
```

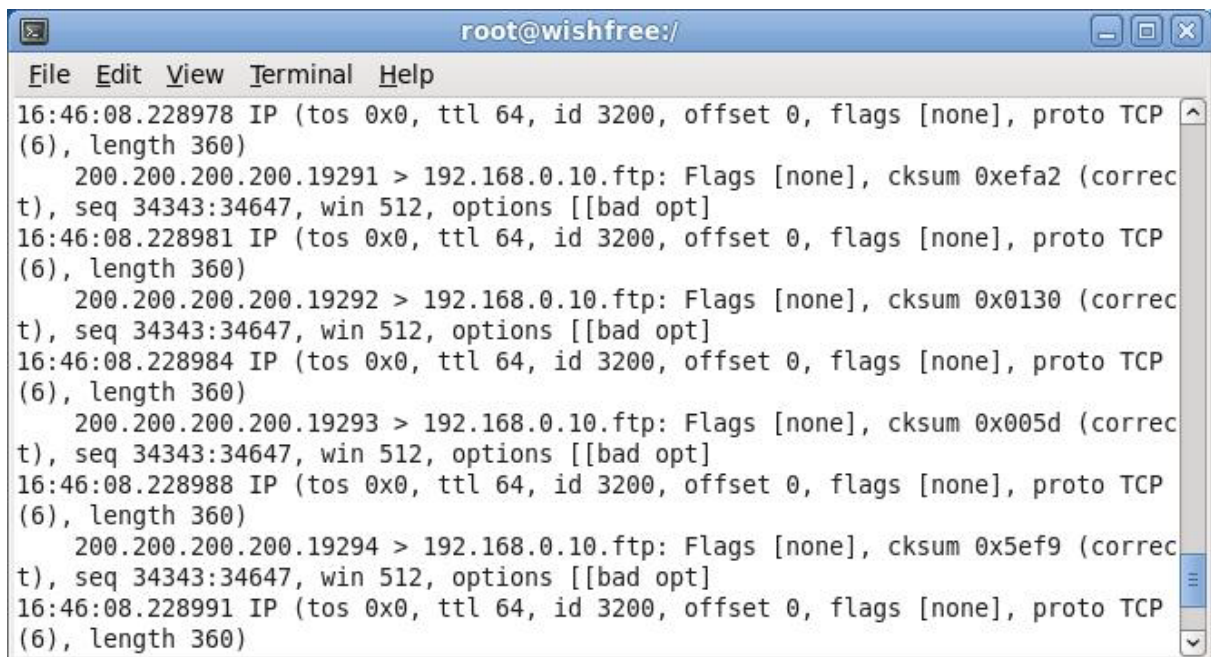
hping3를 활용한 Bonk 공격

```
hping3 -a 200.200.200.200 192.168.0.10 --id 3200 -O 12345 -M 34343 -p 21 -d 320 --flood
```

- ⊙ -a 200.200.200.200 : 공격자의 IP 주소를 200.200.200.200 으로 위조한다.
- ⊙ 192.168.0.10 : 공격 대상 시스템의 IP 주소다.
- ⊙ --id 3200 : id 를 3200 으로 지정한다.
- ⊙ -O 12345 : Offset 을 12345 로 지정한다.
- ⊙ -M 34343 : 시퀀스 넘버를 34343 으로 지정한다.
- ⊙ -p 21 : 21번 포트에 대해 패킷을 전송한다.
- ⊙ -d 320 : 전송하는 패킷의 길이는 320바이트로 한다.
- ⊙ --flood : 시스템이 생성 가능한 만큼 빠른 속도로 패킷을 보낸다.

#### ▶ Boink Attack

- 공격은 처음 패킷을 1번으로 보낸 후 두번째 101번, 세번째 201번으로 정상적으로 전송하다가 중간에 일정한 시퀀스 넘버로 보낸다.(예를 들면, 열번째 패킷은 1001번, 열한번째, 열두번째 패킷도 1001번으로 보내는 것)

A terminal window titled 'root@wishfree:/' showing network traffic analysis. The output displays several IP packets with details like timestamp, IP address, TTL, ID, offset, flags, protocol, and length. Each packet is followed by a line indicating the destination and various flags and checksums. The packets are all from 200.200.200.200 to 192.168.0.10. The ID field is consistently 3200 across all packets, which is the key indicator of a Bonk attack.

```
root@wishfree:/  
File Edit View Terminal Help  
16:46:08.228978 IP (tos 0x0, ttl 64, id 3200, offset 0, flags [none], proto TCP  
(6), length 360)  
200.200.200.200.19291 > 192.168.0.10.ftp: Flags [none], cksum 0xefa2 (correct), seq 34343:34647, win 512, options [[bad opt]  
16:46:08.228981 IP (tos 0x0, ttl 64, id 3200, offset 0, flags [none], proto TCP  
(6), length 360)  
200.200.200.200.19292 > 192.168.0.10.ftp: Flags [none], cksum 0x0130 (correct), seq 34343:34647, win 512, options [[bad opt]  
16:46:08.228984 IP (tos 0x0, ttl 64, id 3200, offset 0, flags [none], proto TCP  
(6), length 360)  
200.200.200.200.19293 > 192.168.0.10.ftp: Flags [none], cksum 0x005d (correct), seq 34343:34647, win 512, options [[bad opt]  
16:46:08.228988 IP (tos 0x0, ttl 64, id 3200, offset 0, flags [none], proto TCP  
(6), length 360)  
200.200.200.200.19294 > 192.168.0.10.ftp: Flags [none], cksum 0x5ef9 (correct), seq 34343:34647, win 512, options [[bad opt]  
16:46:08.228991 IP (tos 0x0, ttl 64, id 3200, offset 0, flags [none], proto TCP  
(6), length 360)
```

Bonk 공격 패킷 분석

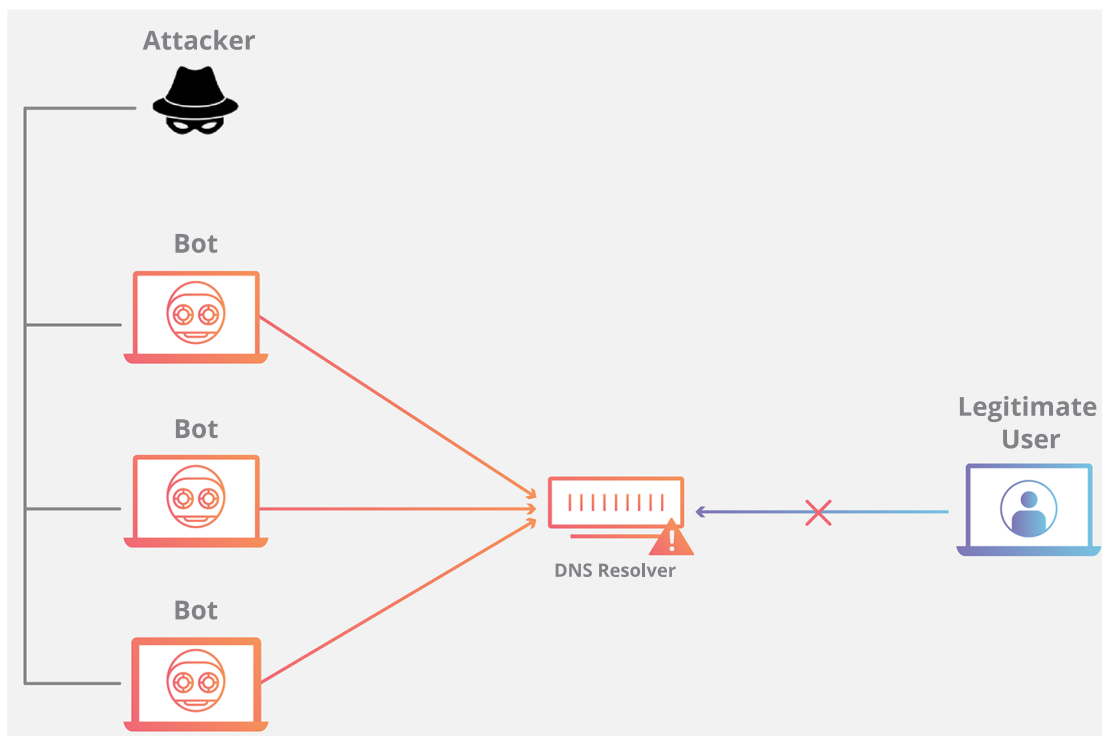
- ⊙ id 3200 : id는 공격자가 정하여 보낸 값으로 3200이다. 이 값이 같으면 패킷을 수신하는 시스템은 모두 한 패킷이라고 생각하게 된다.
- ⊙ length 360 : 패킷의 길이이다. IP 헤더와 TCP 헤더가 각각 20바이트씩, 총 40바이트를 포함하여 360바이트로 전송되었다
- ⊙ 34343:34647 : 패킷의 시퀀스 넘버다. 다음 패킷의 번호와 중첩되는 구간이 있게 보낸다.

#### ▶ Bonk/Boink Attack 대응 방안

- 최근에 나온 시스템을 파괴할 수 있는 경우는 거의 없지만 일부 응용 프로그램에서는 이러한 공격이 동작하는 경우를 가끔 발견할 수 있다.
- 공격에 취약하다면 가장 일반적으로 할 수 있는 방법은 패치이며, 운영체제별 패치는 각 벤더의 홈페이지에서 구할 수 있다.
- 과부하가 걸리거나 계속 반복되는 패킷을 받는다면 해당 MAC 주소, 혹은 IP 주소로부터 오는 패킷은 받지 않도록 설정한다.

## 1 2 DNS Flooding Attack

- DNS Flooding은 분산 서비스 거부 공격(DDoS)의 한 가지 유형으로 공격자가 도메인에 대한 DNS 확인을 방해하기 위해 특정 도메인의 DNS 서버에 트래픽을 폭주하게 하는 것입니다.
- 사용자가 전화번호부(DNS 서버)를 찾을 수 없다면 주소를 검색하여 특정 자원에 대한 호출을 수행할 수 없게 됩니다.
- DNS Flooding은 DNS 확인을 방해함으로써 웹 사이트, API, 웹 응용 프로그램 등이 합법적 트래픽에 대응하지 못하게 한다.
- DNS Flooding은 다수의 고유 위치에서 발생하면서 도메인의 실제 레코드를 요구함으로써 합법적인 트래픽을 모방하는 경우가 많기 때문에 정상적인 대량의 트래픽과 구별하기 어려울 수 있다.



D

- 도메인 네임 시스템(DNS)의 기능은 기억하기 쉬운 이름(예: example.com)을 기억하기 어려운 웹 사이트 서버 주소(예: 192.168.0.1)로 번역하는 것이므로 DNS 인프라를 성공적으로 공격하면 대부분의 사람들이 인터넷을 사용할 수 없게 된다.
- DNS Flooding Attack은 비교적 새로운 DNS 기반 공격 유형으로 Mirai 등의 고대역폭 IoT(Internet of Things) 봇넷이 증가함에 따라 함께 증가하고 있다.



- DNS Flooding Attack은 IP 카메라, DVR 박스 등 IoT 장치의 고대역폭 연결을 사용하여 주요 공급자의 DNS 서버를 직접 압도한다
- IoT 장치에서 보내는 요청량이 DNS 공급자의 서비스를 압도함으로써 합법적인 사용자들이 공급자의 DNS 서버에 액세스할 수 없게 되는 것이다.
- DNS Flooding Attack은 DNS 증폭 공격과 다르다. DNS 증폭 공격은 DNS Flooding 과 달리, 공격의 출발점을 숨기고 효율성을 높이기 위해 안전하지 않은 DNS 서버의 트래픽을 반사하고 이를 증폭시킨다.
- DNS 증폭 공격은 대역폭 연결이 작은 장치를 사용하여 보안되지 않은 DNS 서버에 다수의 요청을 보낸다.
- 이 장치는 다수의 작은 요청을 작성하여 매우 큰 DNS 레코드 응답을 요구하면서 반환 주소를 피해자의 주소로 지정한다.
- 공격자는 이러한 증폭을 통해 제한된 공격 자원만을 사용하여 더 큰 대상을 압도할 수 있는 것이다.

#### ▶ DNS Flooding 대응방안

- 가능한 DNS 서버를 다중으로 구성하여 특정 서버로의 공격이 발생하더라도 다른 서버가 해당 요청에 대해 응답할 수 있도록 구성

```
;; ANSWER SECTION:
test1.com.      86400    IN       NS       ns11.test1.com.
test2.com.      86400    IN       NS       ns22.test2.com.
test3.com.      86400    IN       NS       ns33.test3.com.
test4.com.      86400    IN       NS       ns44.test4.com.
test5.com.      86400    IN       NS       ns55.test5.com.
```

특정 DNS 요청에 대해 다수의 DNS 서버 등록

- 대부분 DNS요청 패킷은 512Byte를 넘을 수 없기 때문에 처리가 가능한 대역폭에서 서비스를 하는 경우라면 iptables등을 이용해 공격 패킷을 차단

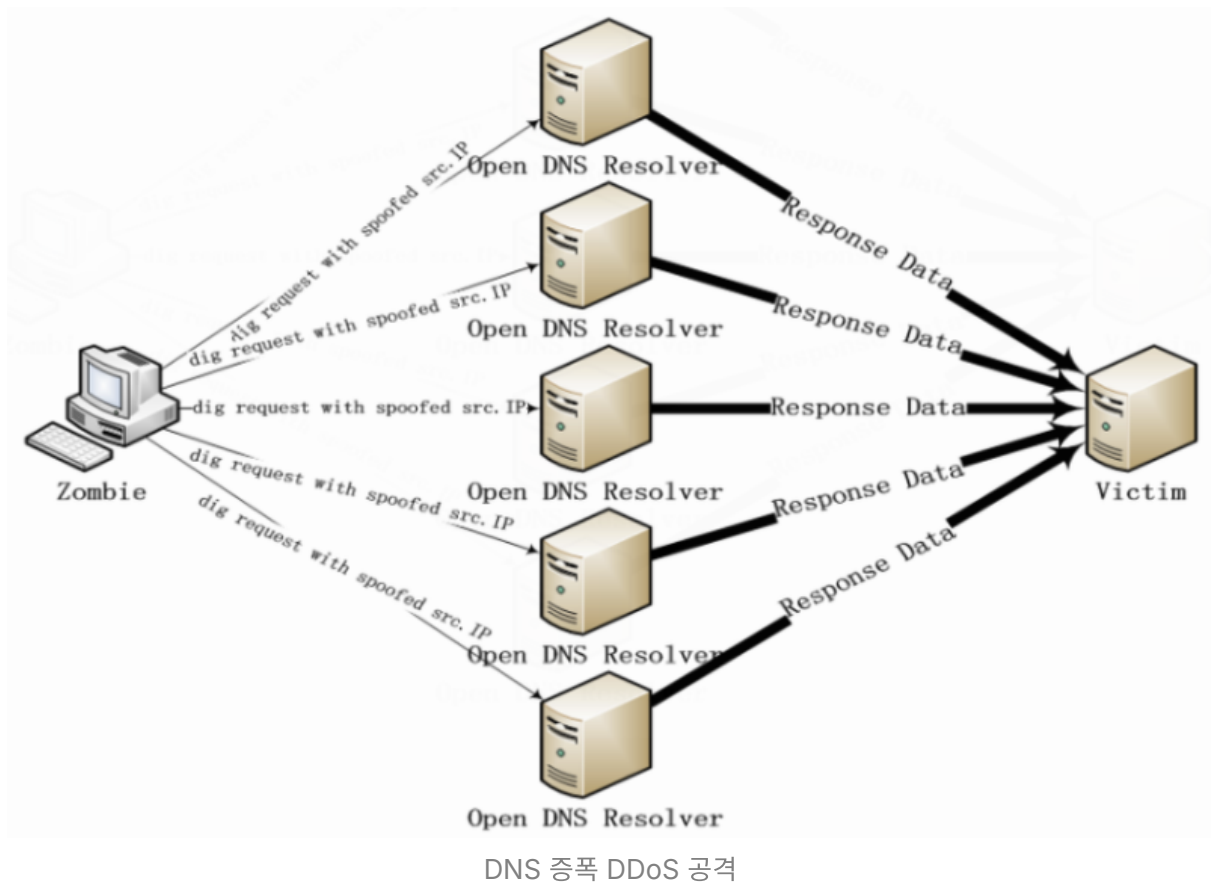
```
iptables -A INPUT -p udp -dport53 -m length --length 512:1500 -j DROP
```

- 레코드 TTL을 늘리면 권한 네임 서버에 도달하는 정상적인 쿼리와 비정상적인 쿼리를 모두 줄여 네임 서버 대역폭을 절약할 수 있다.
- DNS ANY 요청(가능한 최대 DNS 응답)을 비활성화하는 방법으로도 증폭 공격을 강화하기 위해 이러한 종류의 레코드를 악용하는 것을 막을 수 있다.

- 정상적인 요청을 방해하지 않고 DNS 쿼리를 차단하기는 어렵지만, 특정 리졸버가 과도한 양의 트래픽을 전송하는 경우 많은 DNS 서버가 응답을 지연해 대역폭을 절약할 수 있다. 보다 정교한 서버는 NXDOMAIN 응답 급증의 원인이 되는 특정 리졸버 또는 클라이언트의 요청을 대기시키는 등 보다 정교한 속도 제한 로직을 통합할 수 있다.

### **1 3 DNS Amplification Attack**

- 공격을 확대하기 위해 공개된 DNS 서버들을 이용한 정교화된 DoS 공격이며 DNS Reflector Attack 이라고도 한다.
- DNS 증폭 공격은 공격자가 개방형 DNS 확인자를 사용하여 대상 서버 또는 네트워크에 트래픽을 과부하시키는 DDoS 공격 유형이다.
- DNS Reflector Attack 이라고 하는 이유는 실제 자신이 공격을 수행하지 않고 Reflector(DNS Server) 를 사용하여 공격하기 때문이다.
- 다수의 DNS 서버를 이용하는 경우 DNS Amplification DDoA Attacks 이라고도 한다.
- 출발지 IP 주소를 조작하여 DNS 요청(DNS query)에 대한 응답(DNS response)이 조작된 IP 주소(Victim)로 전송 되도록 하는 공격 방법이다.
- 조작하여 공격이 가능한 이유는 기본적으로 DNS 패킷에는 인증 절차가 존재하지 않기 때문이다.



→ 증폭을 위해서 공격자는 Open DNS Resolver 서버를 이용하고 DNS Query 의 Type 을 "Any" 로 설정한다.

#### ▶ Open DNS Resolver 서버

- Open DNS Resolver 서버는 재귀적 질의(Recursive Query) 가 설정되어 있는 서버를 말한다.
- 재귀적 질의란 요청한 도메인 이름이 자신의 서버에 없을 경우 상위 DNS 서버에 요청을 주고 이러한 과정을 반복함으로써 최종 도메인 이름에 대한 처리를 해주는 것을 의미한다.
- DNS 쿼리와 응답의 크기는 불균형적이며, 쿼리에 대한 응답은 원래 쿼리와 응답 모두를 포함하기 때문에 응답 패킷이 항상 쿼리 패킷보다 크게 되어 증폭된다. (ex. DNS 네임 서버가 60 바이트 EDNS 쿼리(EDNS = Extended DNS) 를 받으면 그것에 대한 응답은 122 바이트 A RR 과 4000 바이트 TXT RR, 222 바이트 SOA RR 을 포함할 수 있다. 이는 최대 쿼리 보다 응답이 73% 커지도록 만들어 준다.)

#### ▶ DNS Query Type "Any"

- DNS Query 의 Type 을 "Any" 로 설정하게 되면 다양한 Type 의 레코드를 모두 요청하게 되므로 요청한 쿼리 패킷보다 크게 증폭된다.
- 결국, 재귀적 질의 와 ANY 타입의 레코드를 결합해 DNS 서버에 초당 수백 건의 DNS Query 를 보내게 되면 인증 절차가 없기 때문에 수십 Gbps 의 응답 트래픽을 발생시키게 된다.

No.	Time	Source	Destination	Protocol	Length	Checksum	Info
9825	56.167831	192.168.1.100	192.168.1.1	DNS	89		Standard query 0xff2e ANY hajjam...
9821	56.166119	192.168.1.100	192.168.1.1	DNS	89		Standard query 0xff2e ANY hajjam...
9542	54.565142	192.168.1.100	192.168.1.1	DNS	89		Standard query 0xff2e ANY hajjam...
9537	54.563408	192.168.1.100	192.168.1.1	DNS	89		Standard query 0xff2e ANY hajjam...
9264	52.969053	192.168.1.100	192.168.1.1	DNS	89		Standard query 0xff2e ANY hajjam...
9261	52.967403	192.168.1.100	192.168.1.1	DNS	89		Standard query 0xff2e ANY hajjam...
8994	51.365928	192.168.1.100	192.168.1.1	DNS	89		Standard query 0xff2e ANY hajjam...

Frame 9821: 89 bytes on wire (712 bits), 89 bytes captured (712 bits)	
Ethernet II, Src: [redacted], Dst: [redacted]	
Internet Protocol Version 4, Src: [redacted], Dst: [redacted]	
User Datagram Protocol, Src Port: 64465 (64465), Dst Port: 53 (53)	
Domain Name System (query)	
Transaction ID: 0xff2e	
Flags: 0x0100 Standard query	
Questions: 1	
Answer RRs: 0	
Authority RRs: 0	
Additional RRs: 1	
Queries	
hajjamservices.xyz: type ANY, class IN	
Name: hajjam...	
[Name Length: 18]	
[Label Count: 2]	
Type: * (A request for all records the server/cache has available) (255)	
Class: IN (0x0001)	
Additional records	
<Root>: type OPT	
Name: <Root>	
Type: OPT (41)	
UDP payload size: 9000	
Higher bits in extended RCODE: 0x00	
EDNS0 version: 0	
Z: 0x0000	
0... .. = DO bit: Cannot handle DNSSEC security RRs	
.000 0000 0000 0000 = Reserved: 0x0000	
Data length: 0	

DNS 증폭 공격에 사용되는 DNS 요청 패킷

## 대응 방안

- 가장 기본적인 조치로 DNS 서버를 최신 버전으로 업데이트한다.
- 가장 기본적인 방어 방법은 공격에 사용되고 있는 내부 DNS 서버에서 Any 쿼리를 못하게 설정하는 것이다. 혹은 DNS 를 사용하지 않고 있다면 서비스를 Down 시켜야 한다.
- 출발지가 변조되었기 때문에 출발지 IP 를 방화벽에서 필터링하여 막는 것은 불가능하다. 다만 공격자가 공격 설정 시 특정 포트를 지정하였다면 변조되어 쿼리 하는 출발지의 포트를 필터링하여 쿼리를 막을 수 있다.
- 재귀적 질의를 사용하지 않도록 DNS 서버의 설정을 변경한다.

- 일반적으로 DNS 서버는 내부에서만 사용하므로 내부 IP 에서 요청한 DNS Query 에 대해서만 응답하도록 DNS 서버의 설정을 변경한다.
- 클라우드 기반 DDoS 보호 서비스를 활용하여 대상 서버에 도달하기 전에 많은 양의 트래픽을 흡수하고 필터링할 수도 있다. 이렇게 하면 공격 중에 대상 서버가 과부하되어 사용할 수 없게 되는 것을 방지할 수 있다.(Flood Shield(포괄적인 클라우드 기반 DDoS 방어 서비스))