

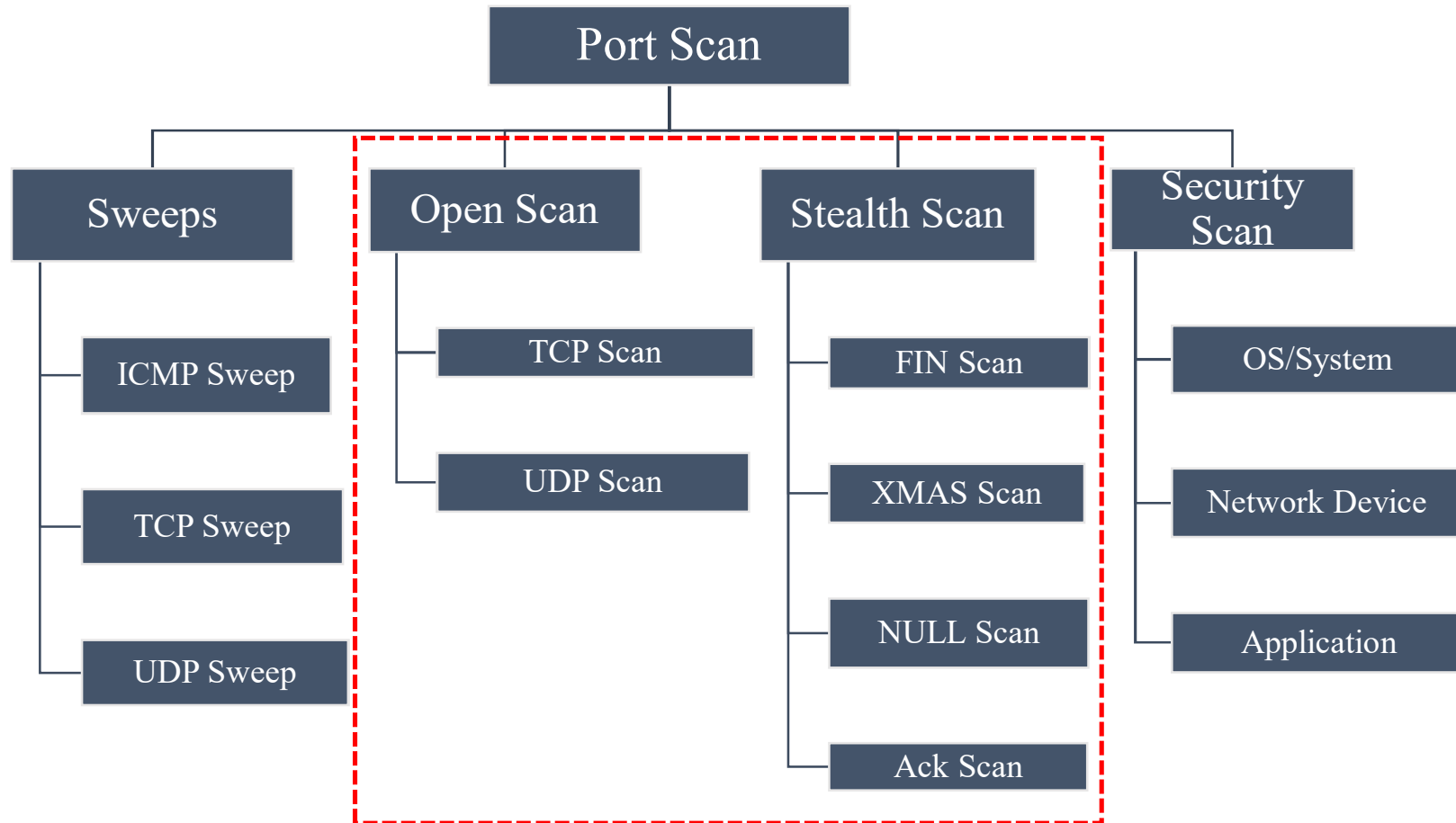
# 네트워크 공격과 패킷 분석 실습

1. Port Scan 공격과 패킷 분석
2. Pharming 공격과 패킷 분석
3. DDoS 공격과 패킷 분석

# 1. Port scan

- 실제 공격방법을 결정하거나 공격에 이용될 수 있는 네트워크 구조, 시스템이 제공하는 서비스 등의 정보를 얻기 위해 수행되는 방법
  - 공격 대상 보안 장비 사용현황
  - 우회 가능 네트워크 구조
  - 시스템 플랫폼 형태
  - 시스템 운영체제의 커널 버전의 종류
  - 제공 서비스 종류

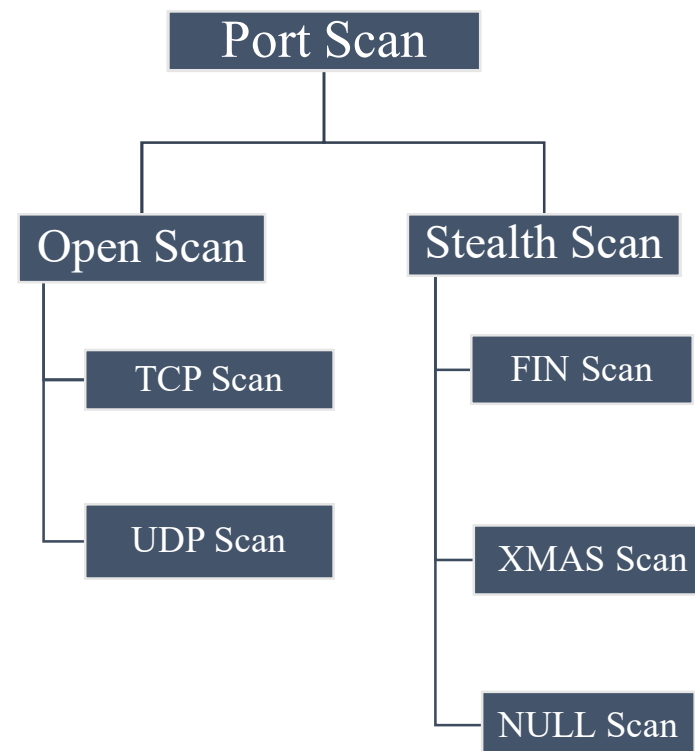
# 1) Port Scan 종류



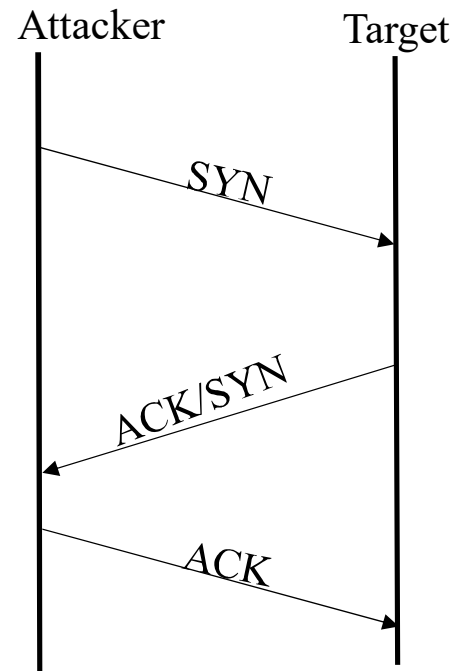
## 2) nmap(network mapper)

- 스캔 도구
- 운영체제 종류 및 사용 서비스에 대한 정보 스캔도구

스캔 옵션	내 용
-sT	connect( ) 함수를 이용한 Open 스캔
-sS	세션을 성립시키지 않는 TCP syn 스캔
-sF	Fin 패킷을 이용한 스캔
-sN	Null 패킷을 이용한 스캔
-sX	XMas 패킷을 이용한 스캔
-sU	UDP 포트 스캔
-sA	Ack 패킷에 대한 TTL 값의 분석



# TCP Full Open Scan

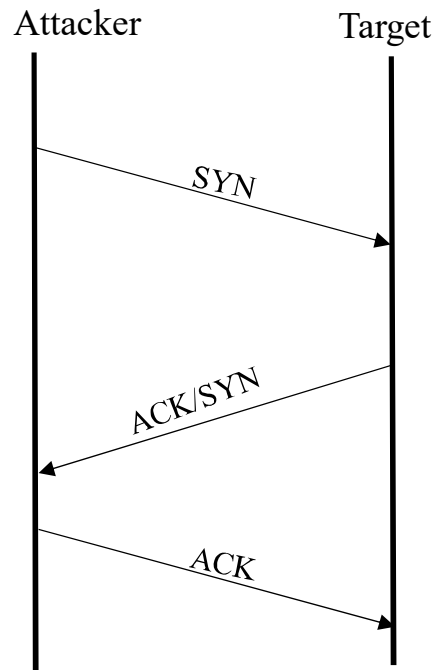


<<열린 포트인 경우 >>

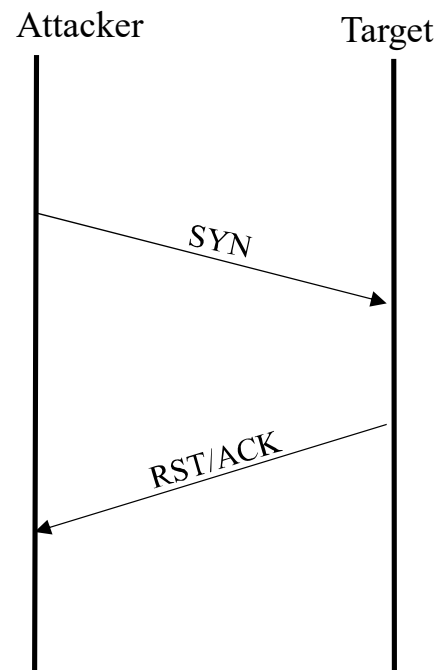
\$nmap -sT [대상IP]

- 포트가 열려 있는 경우 SYN/ACK 패킷 수신
- SYN/ACK에 ACK 패킷을 전송함으로써 연결을 완료
- 스캔하고자 하는 포트에 접속을 시도해 완전한 TCP 연결을 맺어 신뢰성 있는 결과 얻음
- 속도가 느리고 로그를 남기므로 탐지가 가능하다는 단점을 가짐

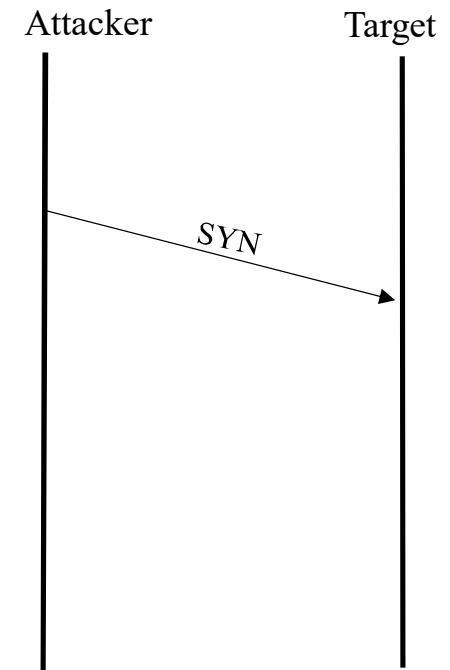
# TCP Full Open Scan



<<열린 포트인 경우 >>



<<닫힌 포트인 경우 >>



<<차단정책으로 차단되어 있는 경우>>

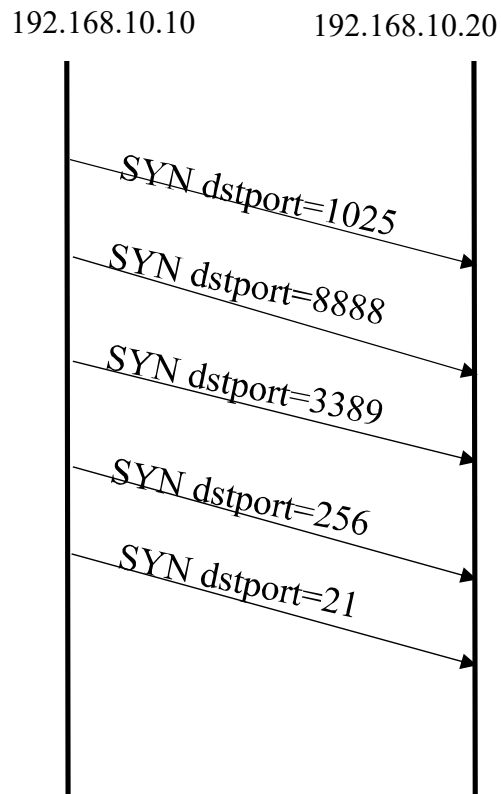
```
(root@kali)-[/home/kali/Downloads]
# nmap -sT 192.168.10.20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-12 21:03 EDT
Nmap scan report for 192.168.10.20
Host is up (0.0021s latency).
Not shown: 977 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
111/tcp   open  rpcbind
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
512/tcp   open  exec
513/tcp   open  login
514/tcp   open  shell
1099/tcp  open  rmiregistry
1524/tcp  open  ingreslock
2049/tcp  open  nfs
2121/tcp  open  ccproxy-ftp
3306/tcp  open  mysql
5432/tcp  open  postgresql
5900/tcp  open  vnc
6000/tcp  open  X11
6667/tcp  open  irc
8009/tcp  open  ajp13
8180/tcp  open  unknown
MAC Address: 00:0C:29:67:D2:B9 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds

(root@kali)-[/home/kali/Downloads]
#
```

`nmap -sT 192.168.10.20`

## • Packet List



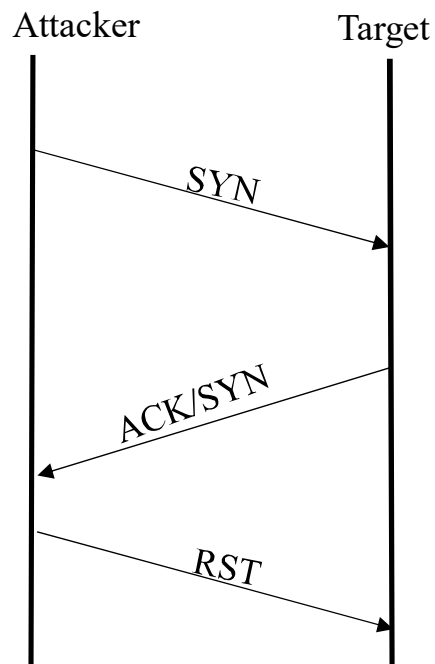
41	0.097772605	192.168.10.10	192.168.10.20	TCP	54 42298 → 111 [RST] Seq=1 Win=0 Len=0
42	0.097809593	192.168.10.10	192.168.10.20	TCP	58 42298 → 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
43	0.097812620	192.168.10.10	192.168.10.20	TCP	58 42298 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
44	0.097813590	192.168.10.10	192.168.10.20	TCP	58 42298 → 3389 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
45	0.097842748	192.168.10.10	192.168.10.20	TCP	58 42298 → 256 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
46	0.097845456	192.168.10.10	192.168.10.20	TCP	58 42298 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
47	0.097846796	192.168.10.10	192.168.10.20	TCP	58 42298 → 587 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
48	0.097848448	192.168.10.10	192.168.10.20	TCP	58 42298 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
49	0.097906852	192.168.10.10	192.168.10.20	TCP	58 42298 → 23 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
50	0.097944151	192.168.10.10	192.168.10.20	TCP	58 42298 → 554 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
51	0.097945166	192.168.10.10	192.168.10.20	TCP	58 42298 → 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
52	0.097946098	192.168.10.10	192.168.10.20	TCP	58 42298 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
53	0.097947010	192.168.10.10	192.168.10.20	TCP	58 42298 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
54	0.097947894	192.168.10.10	192.168.10.20	TCP	58 42298 → 3878 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
55	0.097948778	192.168.10.10	192.168.10.20	TCP	58 42298 → 720 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
56	0.098006524	192.168.10.20	192.168.10.10	TCP	60 25 → 42298 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS
57	0.098006609	192.168.10.20	192.168.10.10	TCP	60 143 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
58	0.098036006	192.168.10.10	192.168.10.20	TCP	54 42298 → 25 [RST] Seq=1 Win=0 Len=0
59	0.098084810	192.168.10.20	192.168.10.10	TCP	60 1025 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
60	0.098084868	192.168.10.20	192.168.10.10	TCP	60 8888 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
61	0.098233239	192.168.10.20	192.168.10.10	TCP	60 3389 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
62	0.098233397	192.168.10.20	192.168.10.10	TCP	60 256 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
63	0.098233436	192.168.10.20	192.168.10.10	TCP	60 21 → 42298 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS
64	0.098233478	192.168.10.20	192.168.10.10	TCP	60 587 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
65	0.098272993	192.168.10.10	192.168.10.20	TCP	54 42298 → 21 [RST] Seq=1 Win=0 Len=0
66	0.098342848	192.168.10.20	192.168.10.10	TCP	60 22 → 42298 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS
67	0.098342892	192.168.10.20	192.168.10.10	TCP	60 23 → 42298 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS
68	0.098342947	192.168.10.20	192.168.10.10	TCP	60 554 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
69	0.098342988	192.168.10.20	192.168.10.10	TCP	60 1723 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
70	0.098356802	192.168.10.10	192.168.10.20	TCP	54 42298 → 22 [RST] Seq=1 Win=0 Len=0
71	0.098391880	192.168.10.10	192.168.10.20	TCP	54 42298 → 23 [RST] Seq=1 Win=0 Len=0
72	0.098435526	192.168.10.20	192.168.10.10	TCP	60 995 → 42298 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

• Statistic > Conversations > TCP > **Port Number**

➔ 어떤 포트를 대상으로 스캔이 시도되었는지 쉽게 확인 가능



# TCP Half Open Scan

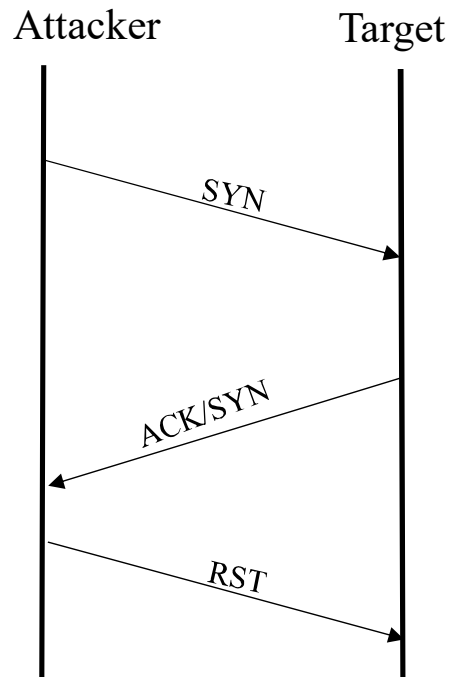


<<열린 포트인 경우 >>

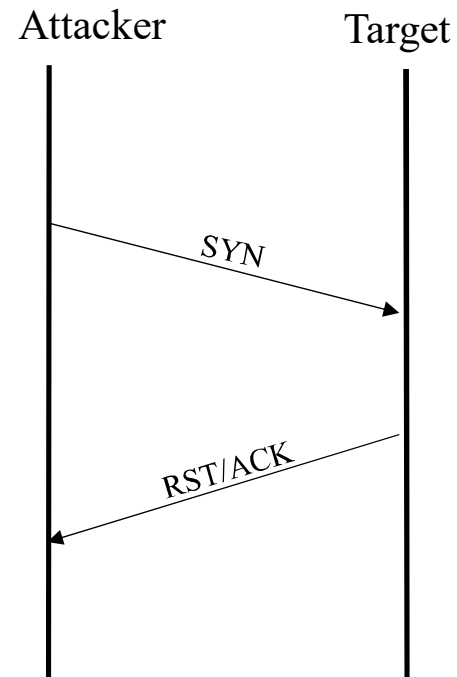
`$nmap -sS [대상IP]`

- 세션에 대한 로그가 남는 TCP Full Openscan을 보안하기 위한 기법
- 공격대상으로부터 SYN/ACK 패킷을 받으면 공격자는 RST 패킷을 보내 연결을 끊음
- 세션을 완전히 연결하지 않음
- 로그를 남기지 않아 추적이 불가능하도록 하는 기법

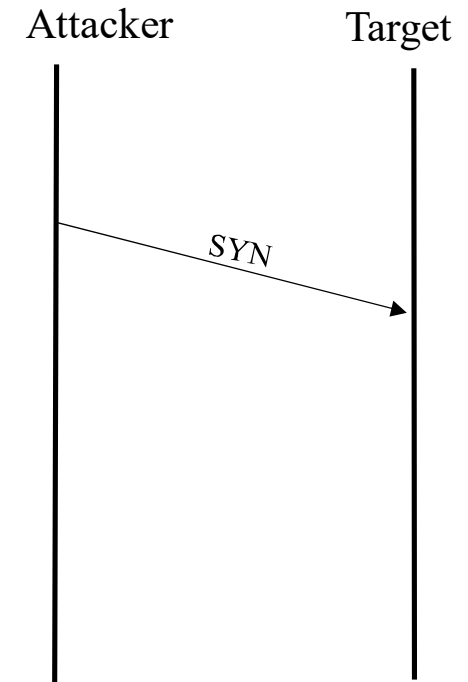
# TCP Half Open Scan



<<열린 포트인 경우 >>



<<닫힌 포트인 경우 >>

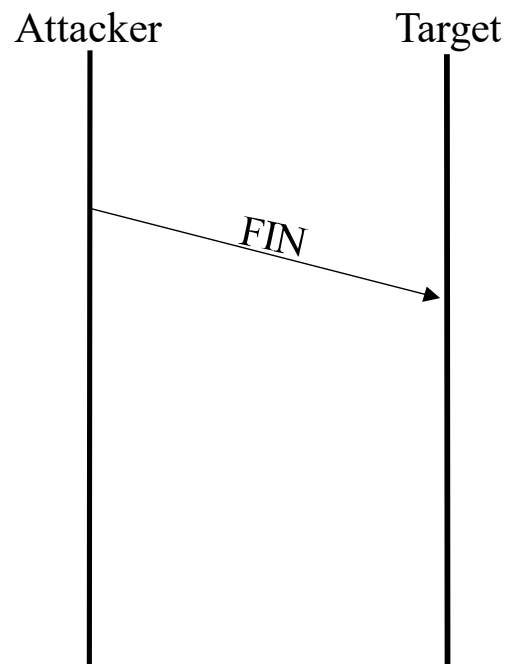


<<차단정책으로 차단되어 있는 경우>>

# Stealth Scan (스텔스 스캔)

- 3Way Handshaking 연결 기법을 이용한 것이 아님
- TCP 헤더를 조작하여 특수한 패킷을 만들어 스캔 대상의 시스템에 보내어 그 응답으로 포트 활성화 여부를 알아내는 기법
- 세션을 성립하지 않고 공격 대상 시스템 포트 활성화 여부를 알아내기 때문에 공격 대상 시스템에 로그를 남기지 않음
- 공격 대상의 시스템 관리자는 어떤 IP를 가진 공격자가 시스템을 스캔 했는지 확인 할 수 없음

# FIN scan

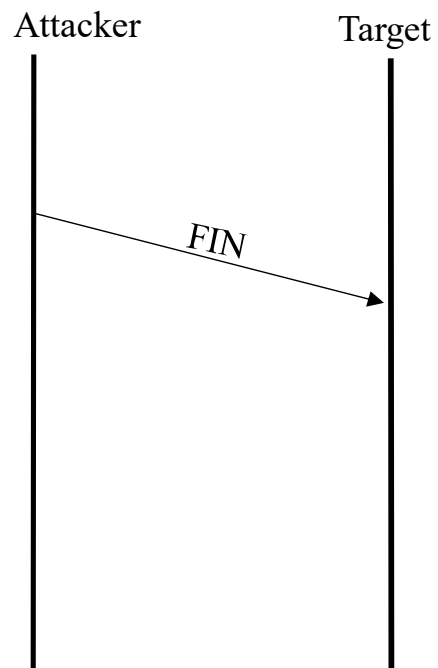


<< 열린 포트의 경우 >>

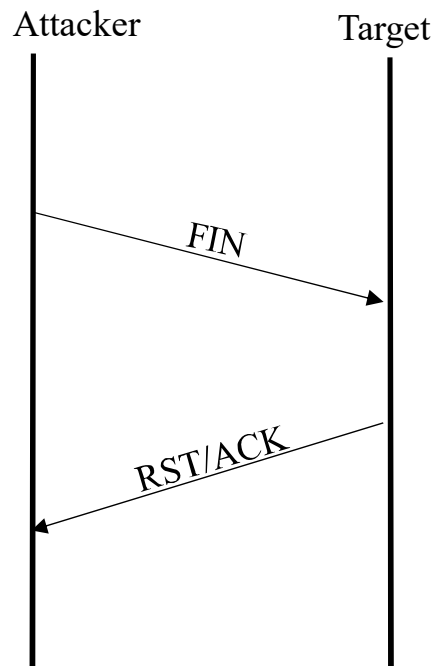
`$nmap -sF [대상IP]`

- TCP 헤더 내에서 FIN 플래그를 설정하여 공격 대상으로 메시지를 전송
- 포트가 열려 있는 경우 응답이 없음

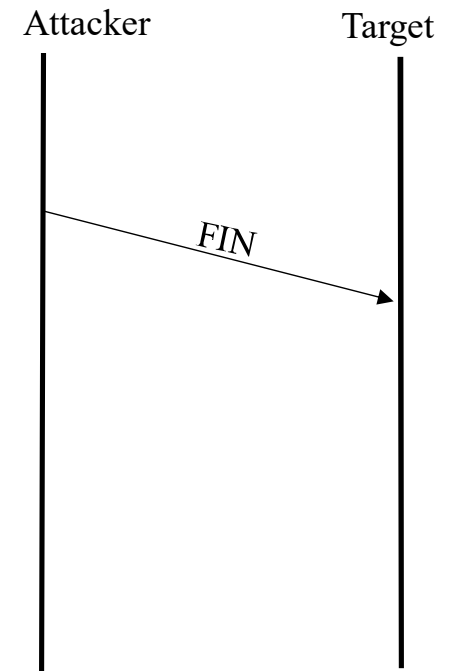
# FIN scan



<< 열린 포트의 경우 >>

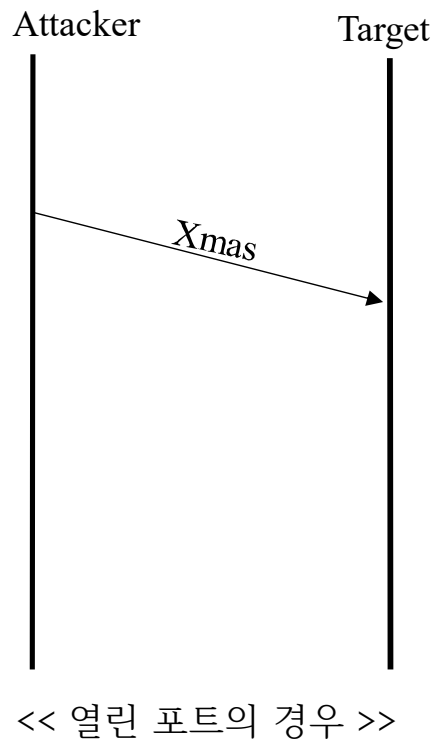


<< 닫힌 포트의 경우 >>



<< 차단정책으로 차단되어 있는 경우 >>

# Xmas scan



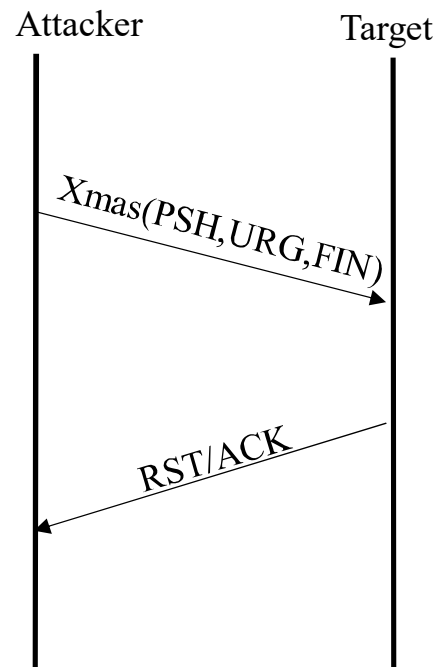
`$nmap -sX [대상IP]`

- TCP 헤더 내에서 URG, PSH, FIN을 동시에 설정해서 전송
- 포트가 열려 있는 경우 응답이 없음

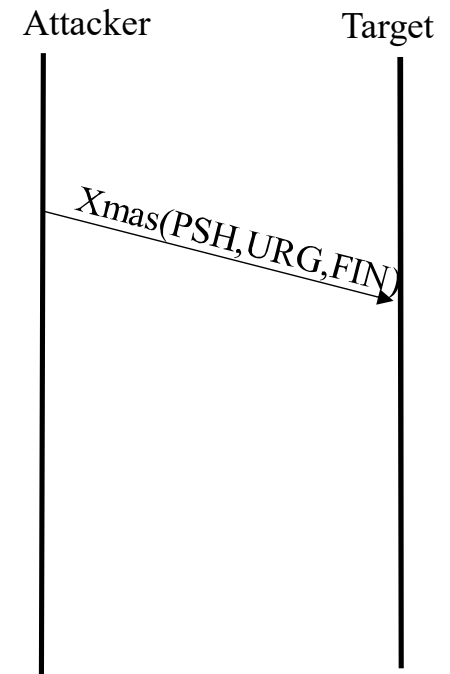
# Xmas scan



<< 열린 포트의 경우 >>



<< 닫힌 포트의 경우 >>



<< 차단정책으로 차단되어 있는 경우 >>

# Xmas scan

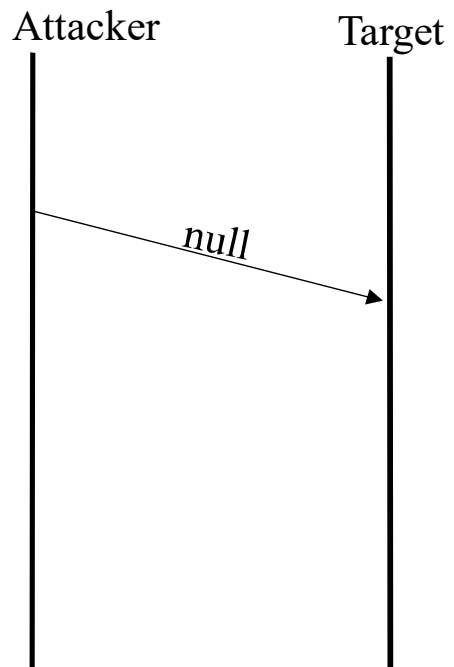
```
(root@kali)-[/home/kali/Downloads]
# nmap -sX 192.168.10.20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-12 21:38 EDT
Nmap scan report for 192.168.10.20
Host is up (0.0011s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE      SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
25/tcp    open|filtered smtp
53/tcp    open|filtered domain
80/tcp    open|filtered http
111/tcp   open|filtered rpcbind
139/tcp   open|filtered netbios-ssn
445/tcp   open|filtered microsoft-ds
```

No.	Time	Source	Destination	Protocol	Length	Info
40	0.116144157	192.168.10.10	192.168.10.20	TCP	54	36515 → 993 [FIN, PSH, URG] Seq=1 W
41	0.116318743	192.168.10.10	192.168.10.20	TCP	54	36515 → 554 [FIN, PSH, URG] Seq=1 W
42	0.116414998	192.168.10.10	192.168.10.20	TCP	54	36515 → 23 [FIN, PSH, URG] Seq=1 Wi
43	0.116504219	192.168.10.10	192.168.10.20	TCP	54	36515 → 8080 [FIN, PSH, URG] Seq=1
44	0.116724691	192.168.10.10	192.168.10.20	TCP	54	36515 → 256 [FIN, PSH, URG] Seq=1 W
45	0.116947632	192.168.10.10	192.168.10.20	TCP	54	36515 → 5900 [FIN, PSH, URG] Seq=1

[Conversation completeness: Incomplete (36)]  
[TCP Segment Len: 0]  
Sequence Number: 1 (relative sequence number)  
Sequence Number (raw): 1670894971  
[Next Sequence Number: 2 (relative sequence number)]  
Acknowledgment Number: 0  
Acknowledgment number (raw): 0  
0101 .... = Header Length: 20 bytes (5)  
▼ Flags: 0x029 (FIN, PSH, URG)  
000. .... = Reserved: Not set  
...0 .... = Nonce: Not set  
.... 0... = Congestion Window Reduced (CWR): Not set  
.... .0.. = ECN-Echo: Not set  
.... ..1. = Urgent: Set  
.... ...0 = Acknowledgment: Not set  
.... ....1.. = Push: Set  
.... .....0.. = Reset: Not set  
.... .... ..0. = Syn: Not set  
▶ .... .... ...1 = Fin: Set



# Null scan

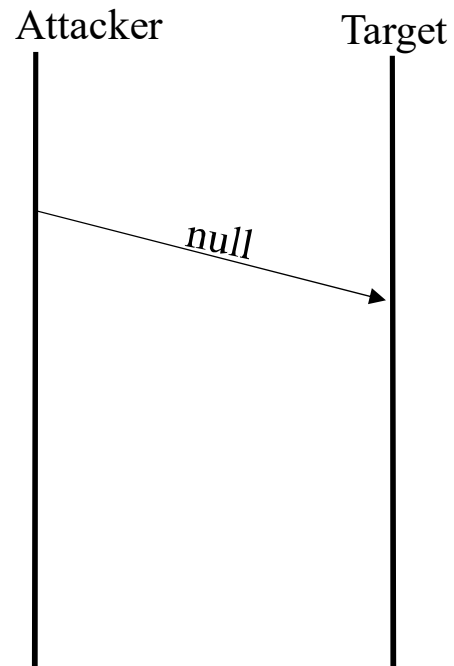


`$nmap -sN [대상IP]`

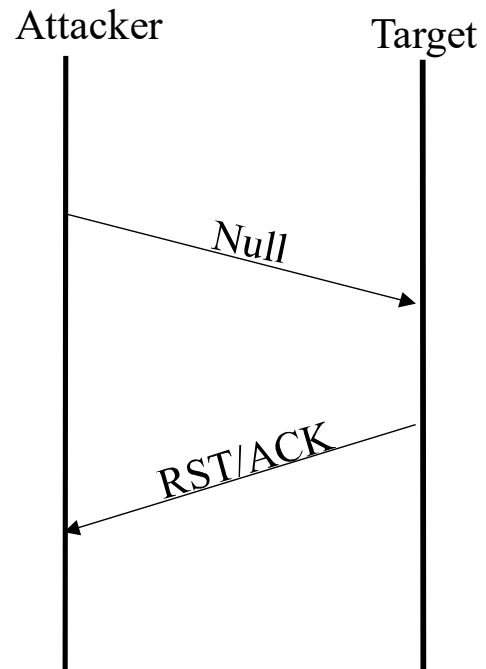
- TCP 헤더 내에 플래그 값을 설정하지 않고 패킷을 전송

<< 열린 포트의 경우 >>

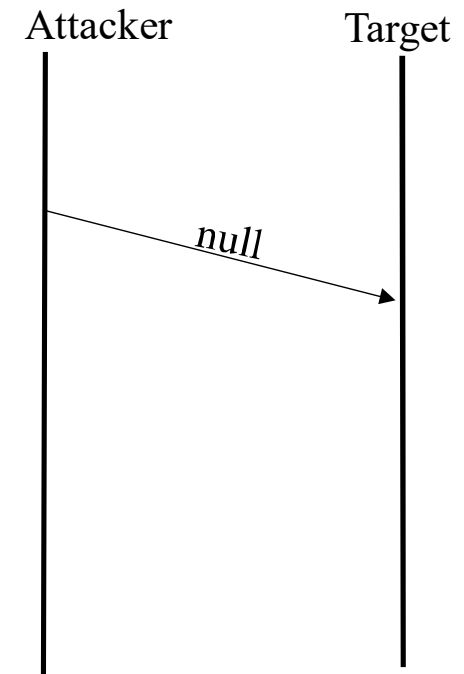
# Null scan



<< 열린 포트의 경우 >>



<< 닫힌 포트의 경우 >>



<< 차단정책으로 차단되어 있는 경우 >>

# Null scan

```
(root@kali)-[/home/kali/Downloads]
# nmap -sN 192.168.10.20
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-12 21:41 EDT
Nmap scan report for 192.168.10.20
Host is up (0.00016s latency).
Not shown: 977 closed tcp ports (reset)
PORT      STATE      SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
25/tcp    open|filtered smtp
```

19	0.099773438	192.168.10.10	192.168.10.20	TCP	54 34902 → 1025 [<None>]
20	0.099915964	192.168.10.10	192.168.10.20	TCP	54 34902 → 110 [<None>] S
21	0.099995867	192.168.10.10	192.168.10.20	TCP	54 34902 → 993 [<None>] S
22	0.100069133	192.168.10.10	192.168.10.20	TCP	54 34902 → 995 [<None>] S
23	0.100171487	192.168.10.10	192.168.10.20	TCP	54 34902 → 3389 [<None>]
24	0.100407173	192.168.10.10	192.168.10.20	TCP	54 34902 → 3306 [<None>]
25	0.100559211	192.168.10.10	192.168.10.20	TCP	54 34902 → 113 [<None>] S

```
[TCP Segment Len: 0]
Sequence Number: 1      (relative sequence number)
Sequence Number (raw): 4272708618
[Next Sequence Number: 1      (relative sequence number)]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
0101 .... = Header Length: 20 bytes (5)
```

▼ Flags: 0x000 (<None>)

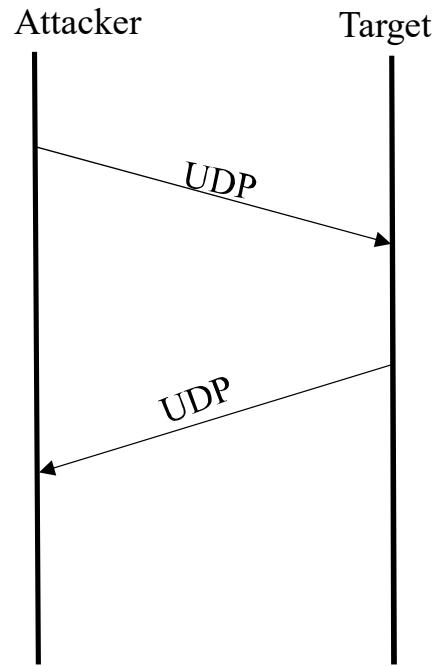
```
000. .... = Reserved: Not set
...0 .... = Nonce: Not set
.... 0... = Congestion Window Reduced (CWR): Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...0 = Acknowledgment: Not set
.... .... 0... = Push: Not set
.... ..... 0.. = Reset: Not set
.... ..... ..0. = Syn: Not set
.... ..... ...0 = Fin: Not set
```

# UDP Scan

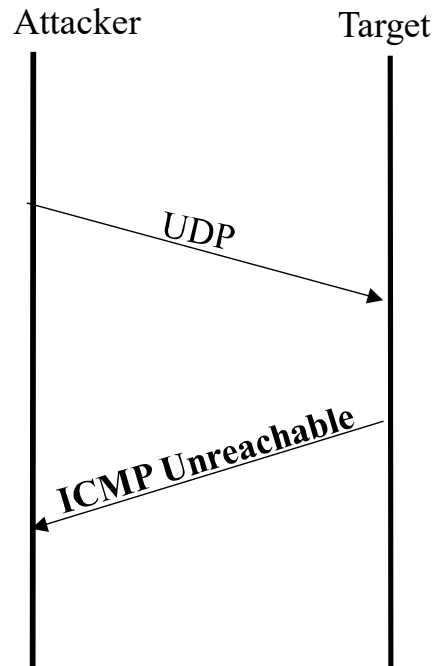
- UDP는 3-way handshake 와 같은 절차가 없음
- UDP 패킷을 전송 시 열려 있는 포트로부터 특정 UDP 응답값으로 수신
- 수신측의 포트가 닫혀 있는 경우 ICMP Port Unreachable 에러 메시지를 통해 포트 활성화 유무 확인

`$nmap -sU [대상서버IP]`

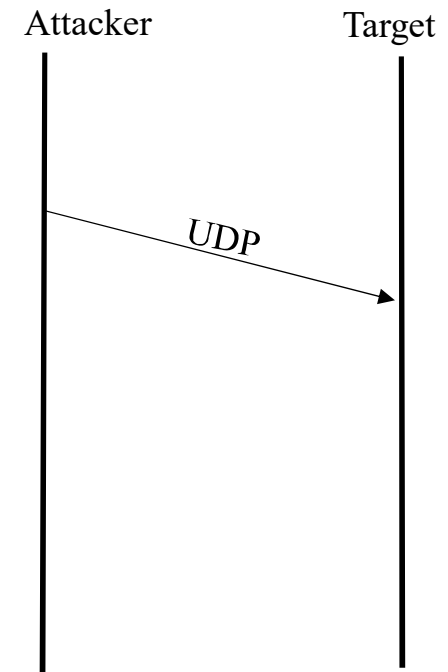
# UDP Scan



<<열린 포트인 경우 >>



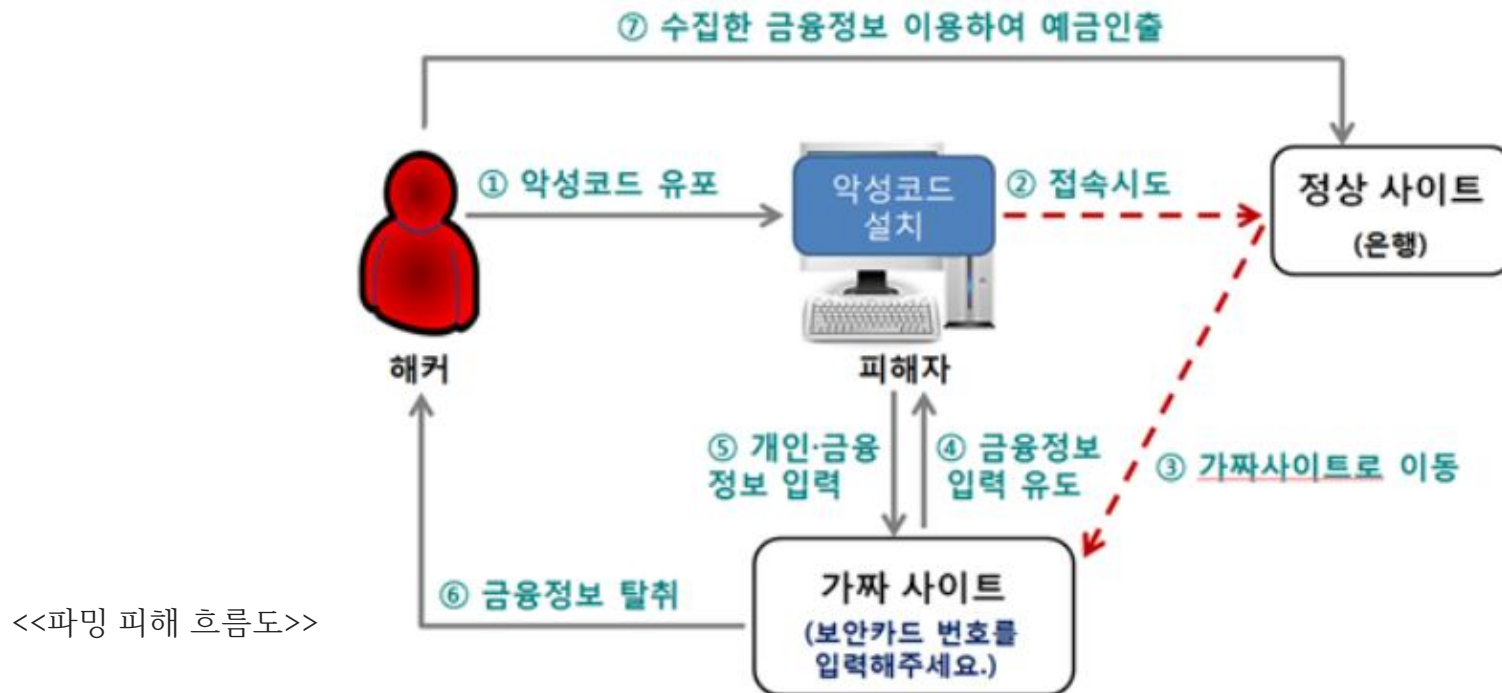
<<닫힌 포트인 경우 >>



<<차단정책으로 차단되어 있는 경우>>

## 2. Pharming Attack

- 피싱(Phishing)+ 조작(Farming)의 합성어
- 정상 사이트에 접속하더라도 가짜 사이트로 접속을 유도하여 금융거래 정보를 빼낸 후 금전적인 피해를 입히는 사기 수법

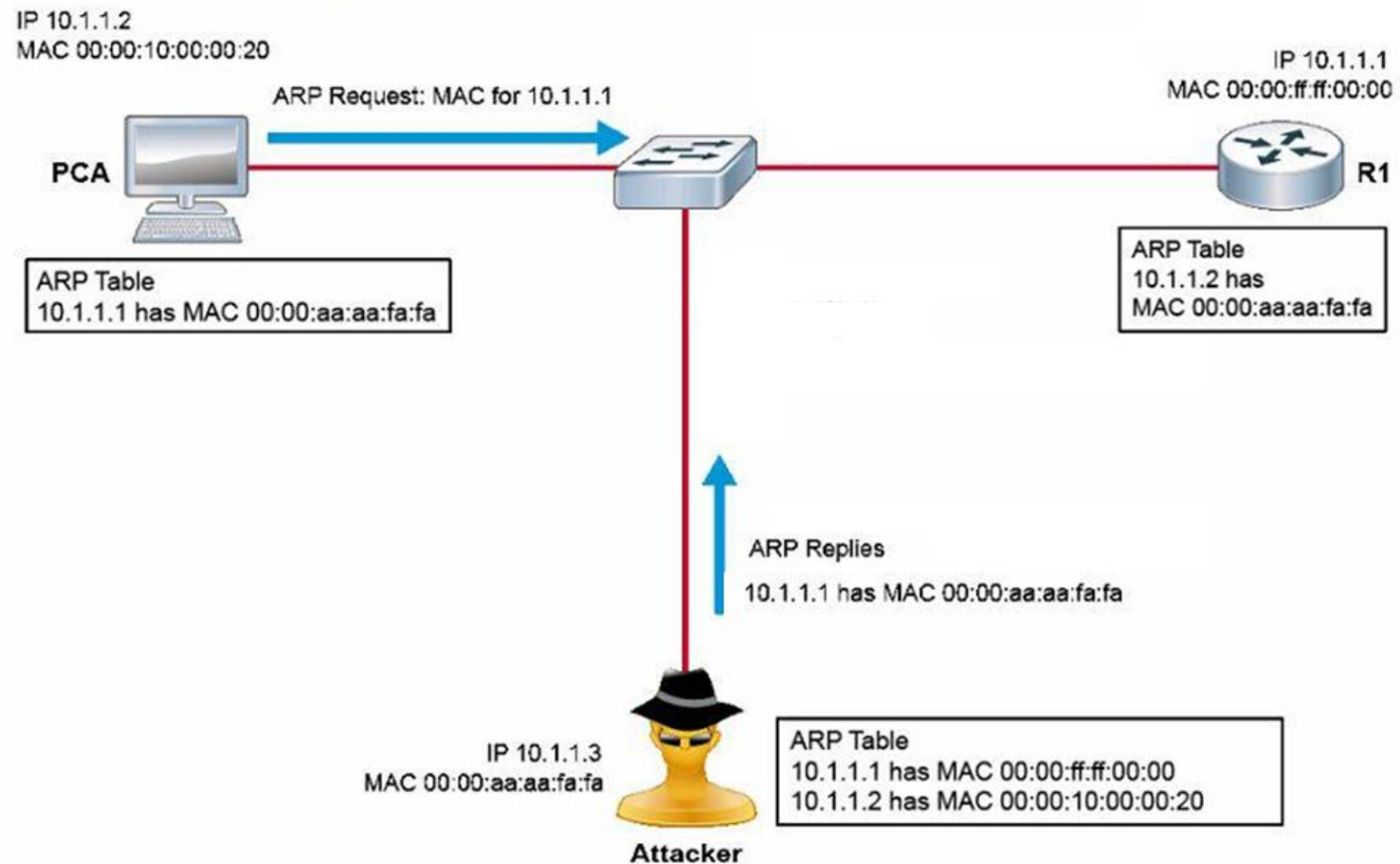


# Spoofing

- ‘속이다’는 의미
- 인터넷이나 로컬에서 존재하는 모든 연결에 spoofing 가능
- 정보를 얻어내기 위한 중간 단계의 기술로 사용하는 것 외에 시스템을 마비시키는 데 사용할 수도 있음
- 종류
  - ARP Spoofing
  - DNS Spoofing

# ARP Spoofing

- MAC 주소를 속이는 것
- 2계층에서 작동해 공격 대상이 같은 랜에 있어야 함





# ARP Spoofing

```
#arp spoof -i eth0 -t 192.168.10.40 192.168.10.2
```

```
(root@kali)-[/]
# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.10.10 netmask 255.255.255.0 broadcast 192.168.10.255
    inet6 fe80::32a8:b96:c197:1e6e prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:bc:ad:00 txqueuelen 1000 (Ethernet)
    RX packets 2675 bytes 252128 (246.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8961 bytes 608042 (593.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

<<공격 전>>

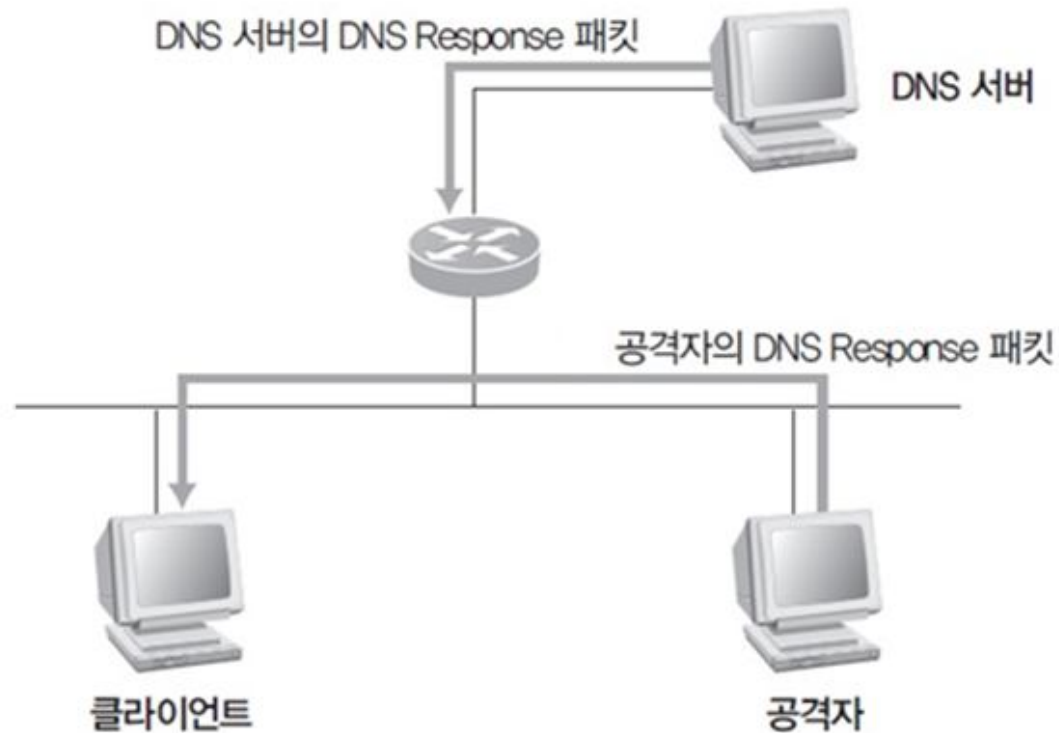
인터페이스: 192.168.10.40 --- 0xe	인터넷 주소	물리적 주소	유형
192.168.10.2	00-50-56-e5-76-78		정상적
<u>192.168.10.10</u>	<u>00-0c-29-bc-ad-00</u>		정상적
<u>192.168.10.255</u>	<u>ff-ff-ff-ff-ff-ff</u>		정상적
224.0.0.22	01-00-5e-00-00-16		정상적
224.0.0.251	01-00-5e-00-00-fb		정상적
224.0.0.252	01-00-5e-00-00-fc		정상적
239.255.255.250	01-00-5e-7f-ff-fa		정상적

<<공격 후>>

인터페이스: 192.168.10.40 --- 0xe	인터넷 주소	물리적 주소	유형
192.168.10.2	00-0c-29-bc-ad-00		정상적
192.168.10.10	<u>00-0c-29-bc-ad-00</u>		정상적
192.168.10.255	<u>ff-ff-ff-ff-ff-ff</u>		정상적
224.0.0.22	01-00-5e-00-00-16		정상적
224.0.0.251	01-00-5e-00-00-fb		정상적
224.0.0.252	01-00-5e-00-00-fc		정상적
239.255.255.250	01-00-5e-7f-ff-fa		정상적

# DNS Spoofing

- 실제 DNS 서버보다 빠르게 위조된 DNS response 패킷을 보내 공격 대상이 잘못된 IP 주소로 웹 접속을 하도록 만드는 공격 방법
- 클라이언트는 이미 DNS response를 받았으므로 정상 DNS response는 drop



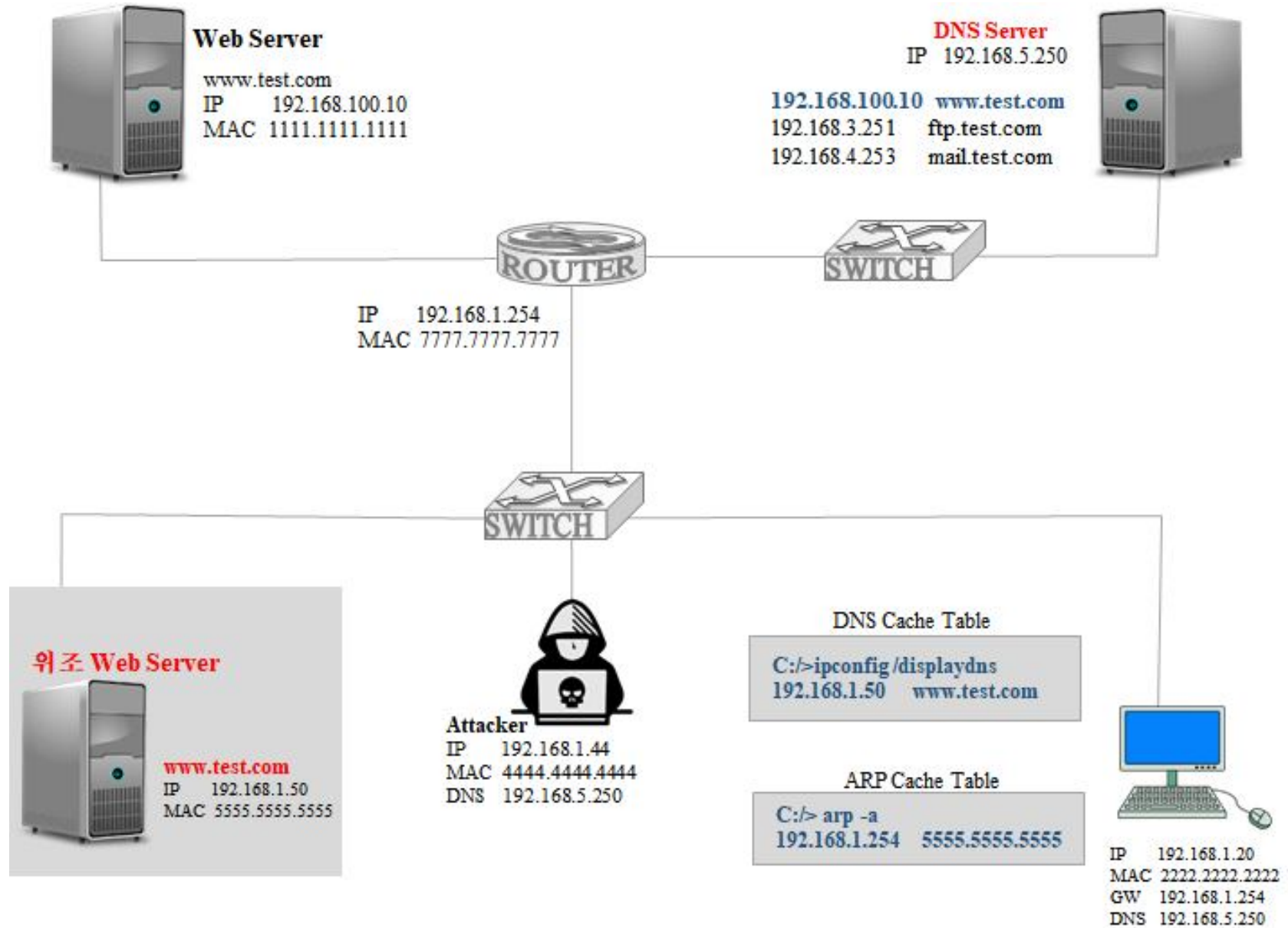
# DNS Spoofing

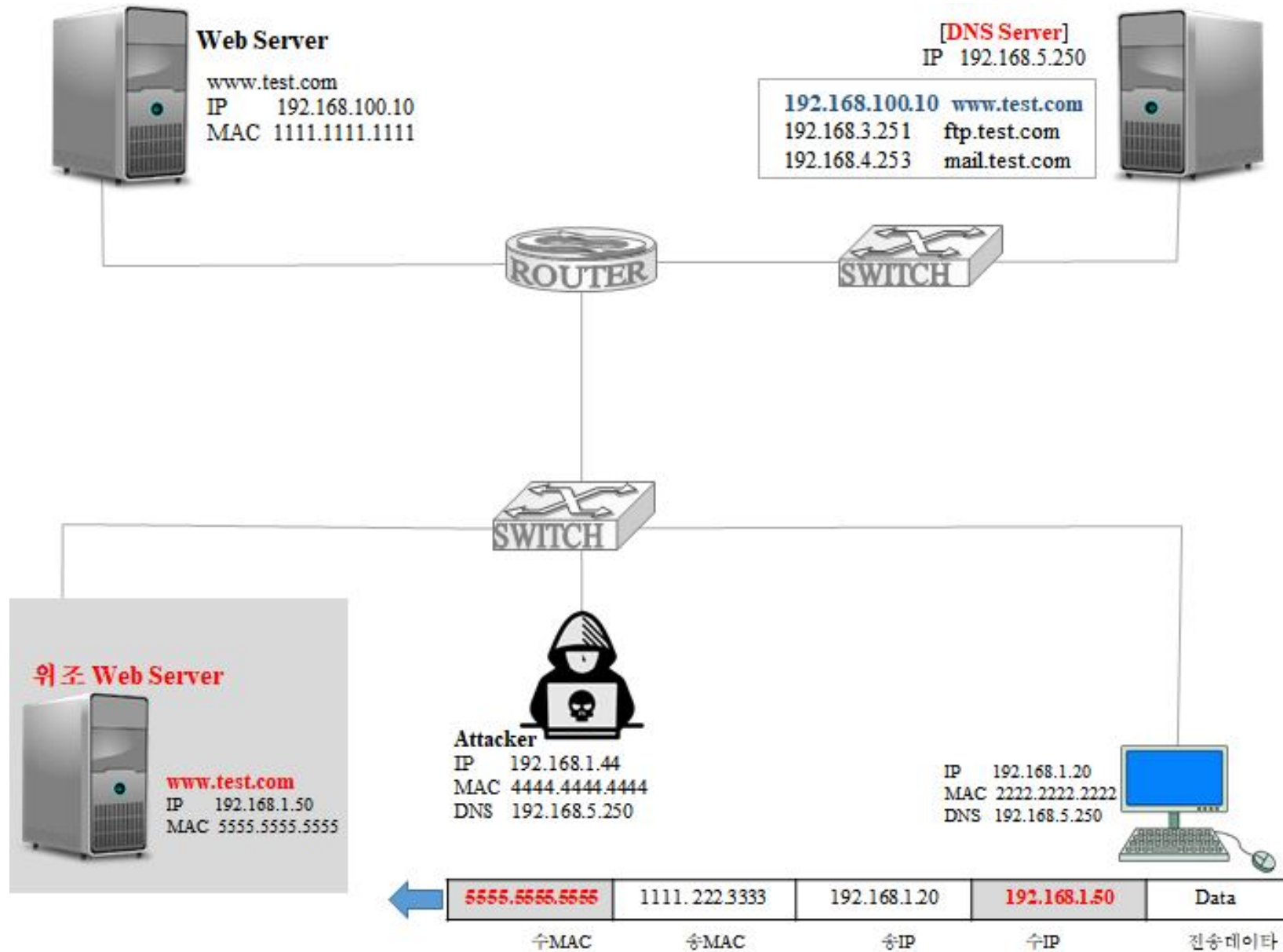
## <<DNS Table 생성 >>

```
#cd /  
#vi dns  
192.168.10.10 www.sks.com  
:wq!
```

## <<DSN Spoofing 수행 >>

```
#dnsspoof -f /dns  
DNS table 파일이름
```





### **3. DDoS(Distributed Deny of Service) Attack**

- 과도한 트래픽을 공격대상에게 전송하여 서비스를 불가능하게 하는 공격 기법
- 과도한 트래픽 또는 부하를 발생시켜 정상적인 통신이 불가능하게 만드는 통신 유형

# 1) 통신 기본 3요소

## ① 전송매체(회선)

- End-to-End 연결통로
- 각 전송 매체 별로 수용 가능한 대역폭을 보유



## ② 정보원(송수신자)

- End-to-End
- End-to-End 연결 중계장비
- 각각 처리할 수 있는 최대 선능 존재
- 최대 성능은 CPU/메모리 등 장착되는 부품에 따라 달라짐



## ③ 프로토콜

- 통신규약
- 정상적인 통신을 위해 미리 정의된 규약에 맞춰 데이터 송수신



## 2) DDoS 공격 원리

- 전송 매체 별 **자신이 수용 가능한 대역폭 이상의 트래픽이 전송될 경우**, 전송된 트래픽을 수용하지 못하여 정상적인 통신이 불가능해짐



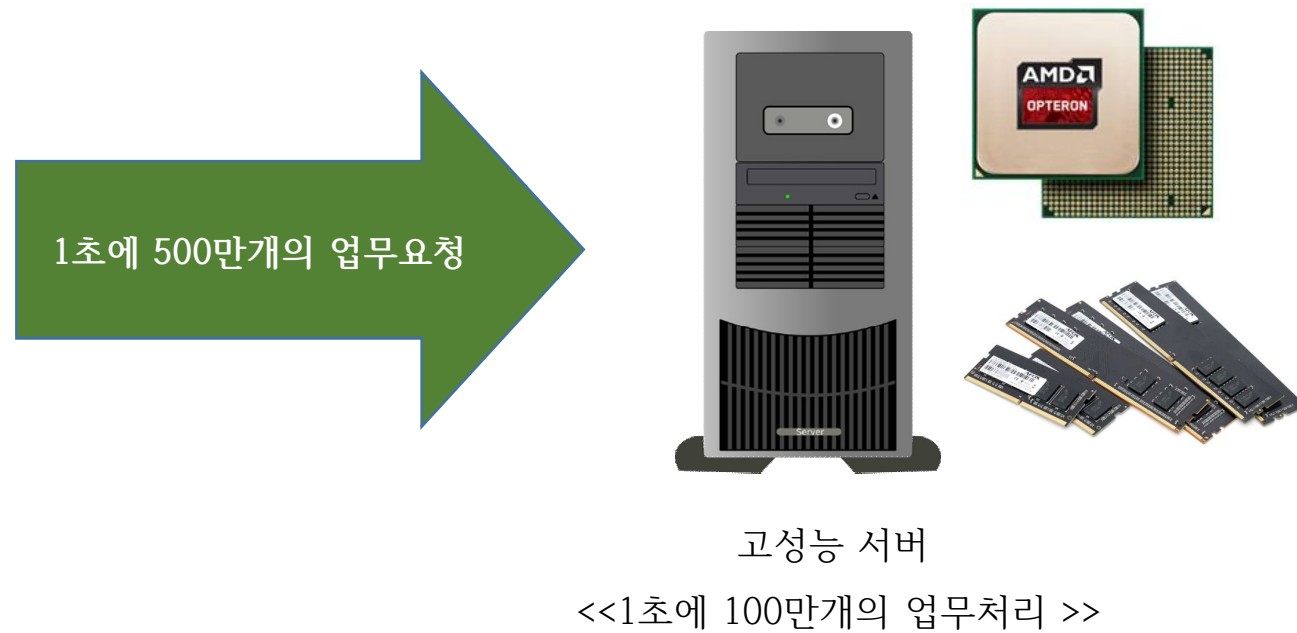
UTP Cable(1G)



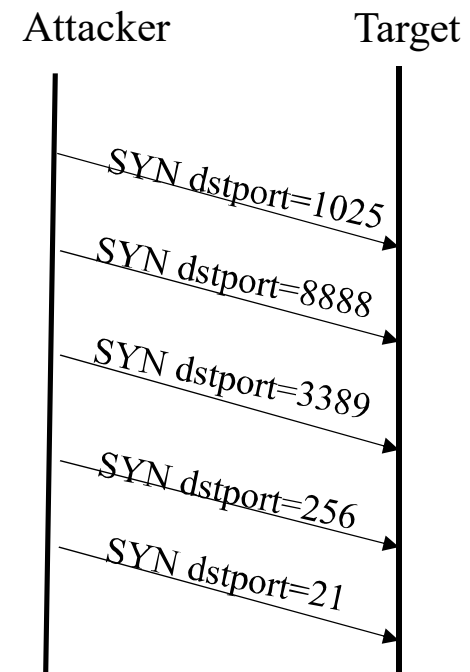
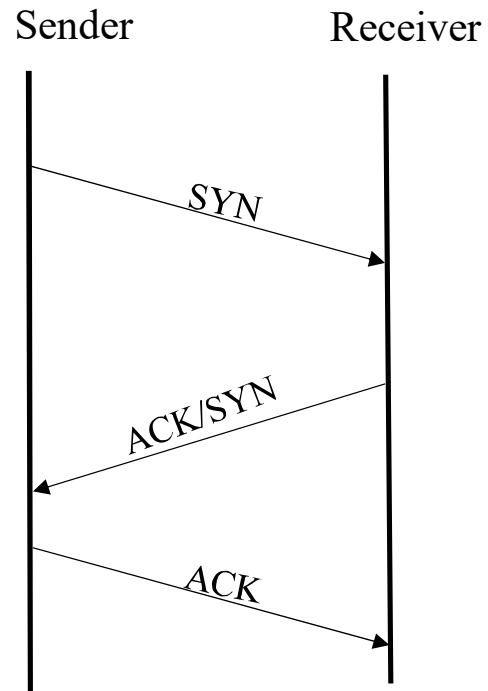
Optical Cable(10G)



- 각 정보원이 **처리 가능한 성능 이상의 요청이 발생할 경우** 이를 처리하지 못하여 정상적인 통신이 불가능해진다.



- 프로토콜의 허점을 이용하여 운영체제 또는 설치된 애플리케이션이 비정상적 상태에 빠지게 한다.



미존재IP  
IP:??????

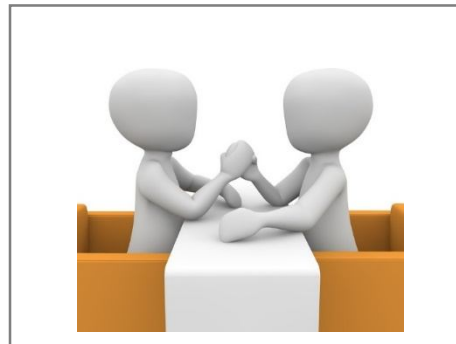
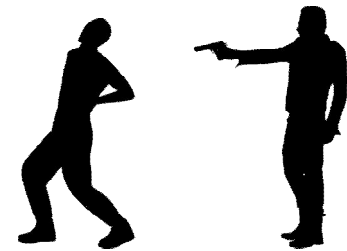
### 3) DDoS 공격 목적

- 특정 시스템의 취약점을 이용하여 시스템에 침투하거나 파일을 유출 또는 변조하는 행위

금전요구



개인적 원한



경쟁상에 의한 공격/청부



해커비즘

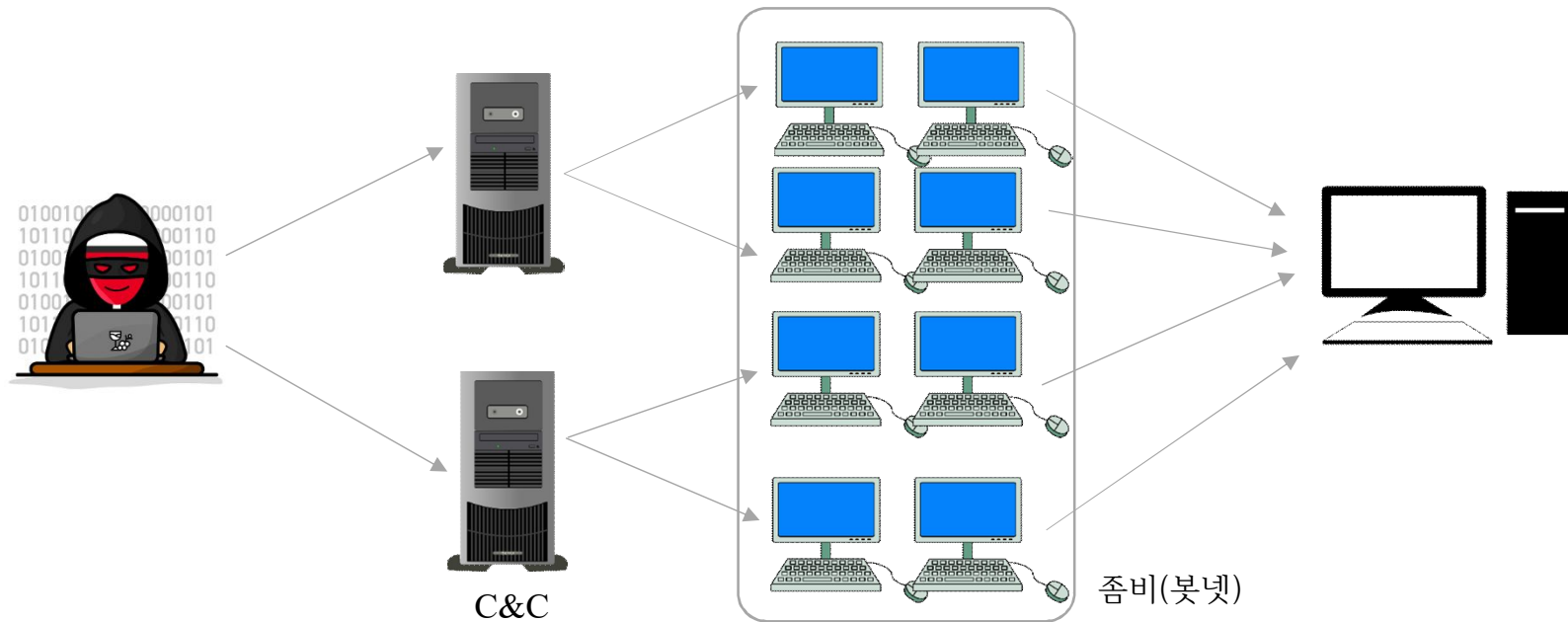
## DoS (Denial of Service, 서비스 거부 공격)

- 특정 공격 PC 또는 서버 1대에서 공격 대상 서버 1대로 과도한 트래픽 또는 패킷은 전송하는 1:1 형태

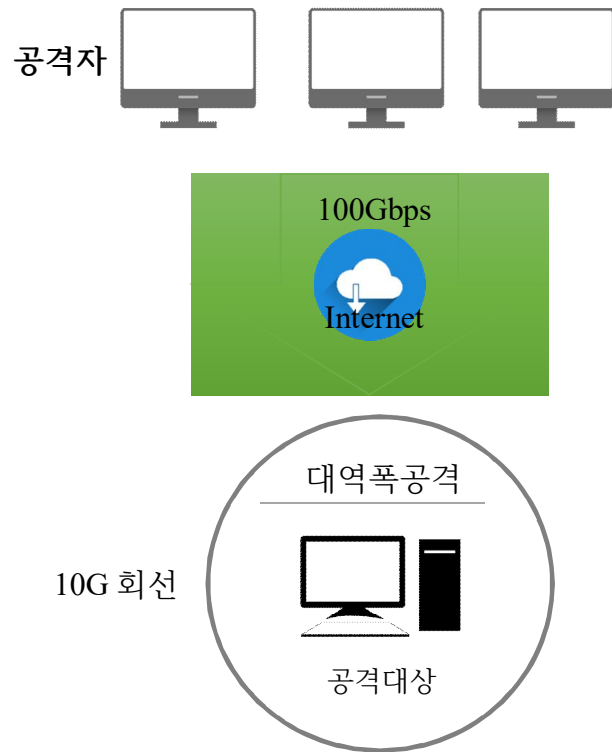


## DDoS (Distributed Denial of Service, 분산서비스거부공격)

- 서버와 차단 장비의 성능이 높아짐에 따라 DoS와 같은 1:1 공격은 더 큰 효과를 낼 수 없게 되었음
- 공격 성능을 증대 시키기 위해 탄생한 것
- 악성코드에 감염된 여러 대의 좀비들을 이용하여 동시에 공격하므로 N:1 형태를 띠



# DDoS 대역폭 공격



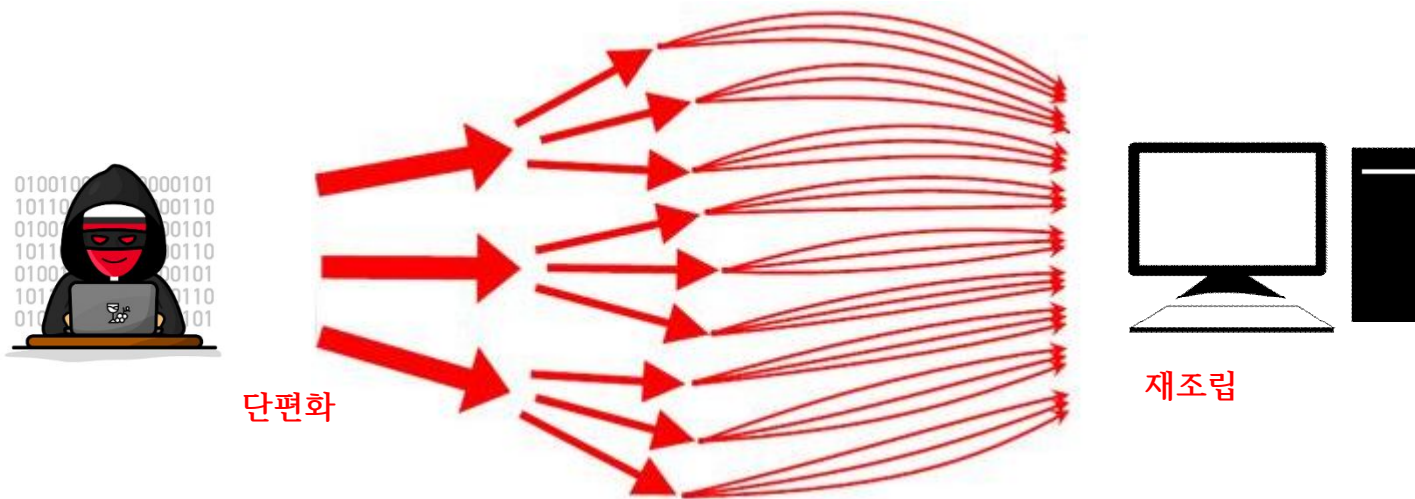
- ❶ 목적 : 대용량의 트래픽 전송으로 인한 네트워크 회선 대역폭 고갈
- ❷ 영향 : 회선 대역폭 고갈로 인한 정상 사용자 접속 불가
- ❸ 주요 프로토콜 : UDP, ICMP
- ❹ 특징 : 주로 위조된 큰 크기의 패킷과 위조된 출발지 IP 사용

구분	내용
대표적인 공격유형	UDP flooding ICMP flooding Fragment Flooding
공격목적	회선 대역폭 잠식
공격기법	bps(bit per second)
공격계층	네트워크 계층(layer 3/4)

# Fragmentation Flooding Attack

\* 문자열 65000 바이트로 네트워크에 ping전송

**hping3 --icmp --rand-source 192.168.10.20 -d 65000 --flood**



- 네트워크 기기가 전송할 수 있는 최대전송단위 MTU이상의 크기의 패킷을 전송 시, 패킷이 분할되는 단편화(fragmentation)의 특징을 이용한 공격 유형

# DDoS 자원 고갈 공격



- ❶ 목적 : 정상 혹은 비정상적인 TCP flag 가 설정된 패킷을 서버 또는 네트워크 장비로 전송하여 장비의 자원 고갈
- ❷ 영향 : 장비의 특정 자원이 고갈되어 정상적인 운영 불가
- ❸ 주요 프로토콜 : TCP
- ❹ 특징 : TCP flag를 이용하여, 위조된 IP를 사용

구분	내용
대표적인 공격 유형	SYN flooding ACK flooding Fragment Flooding
공격목적	서버 및 네트워크 장비의 자원 고갈로 인한 장비 운영 불가
공격기법	PPS (Packet Per Second)
공격계층	네트워크 계층(layer 3/4)

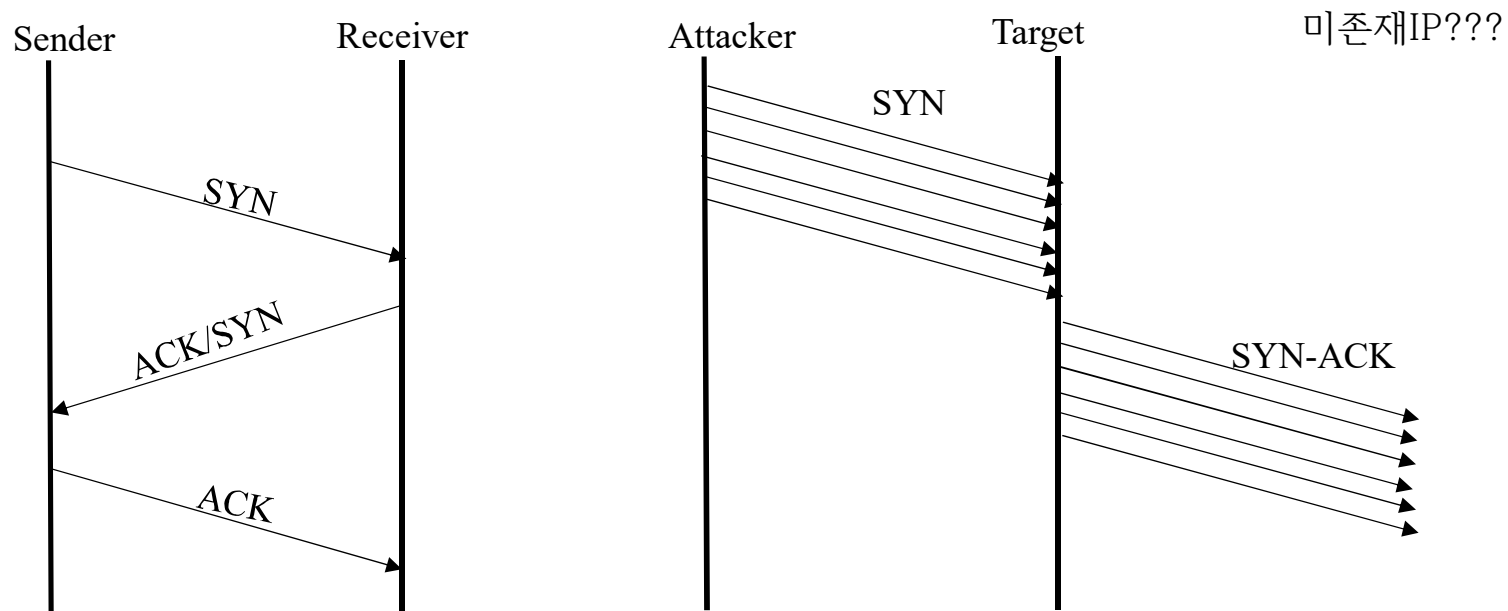


# SYN Flooding Attack

- SYN Flooding 공격을 이용하여 Web Server의 HTTP 서비스를 지연 또는 정지시킴

\*10초 내에 패킷 50여만개의 SYN 전송

**hping3 --rand-source 192.168.10.20 -p 80 -S --flood**



- TCP의 3-way-handshake 과정에서 발생 가능한 취약점을 이용한 공격 유형