

# 스니핑 방지 대책

## ■ 정적 매핑 테이블 사용

- 각 주소 매핑 테이블이 별다른 인증 절차 없이 동적으로 변경이 가능하다는 점에서 발생하는 보안 취약성

→ 동적으로 변경을 불가능하게(정적 주소 매핑 테이블)

- 호스트의 MAC주소, 라우팅 테이블 → ARP Snooping, ICMP Snooping
- 스위칭 허브의 동적 MAC 주소 매핑 테이블 → Jamming Attack

## ■ 탐지/차단

- 네트워크에서 스니핑을 탐지하여 문제가 되는 호스트를 네트워크로부터 격리.

# Sniffing Detection

Sniffing 을 탐지하는 몇 가지 방법들

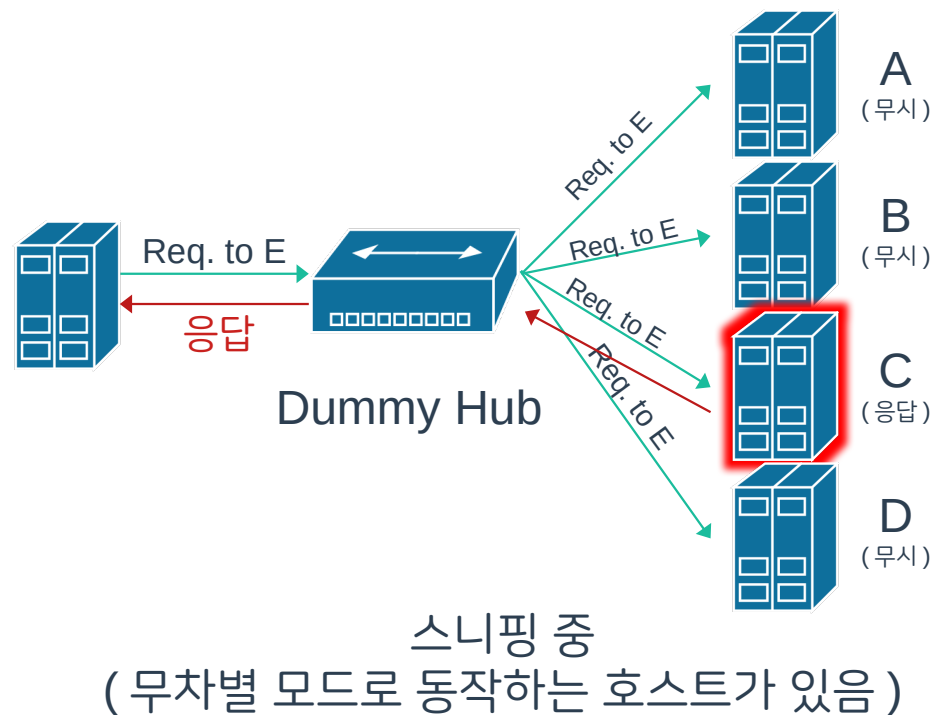
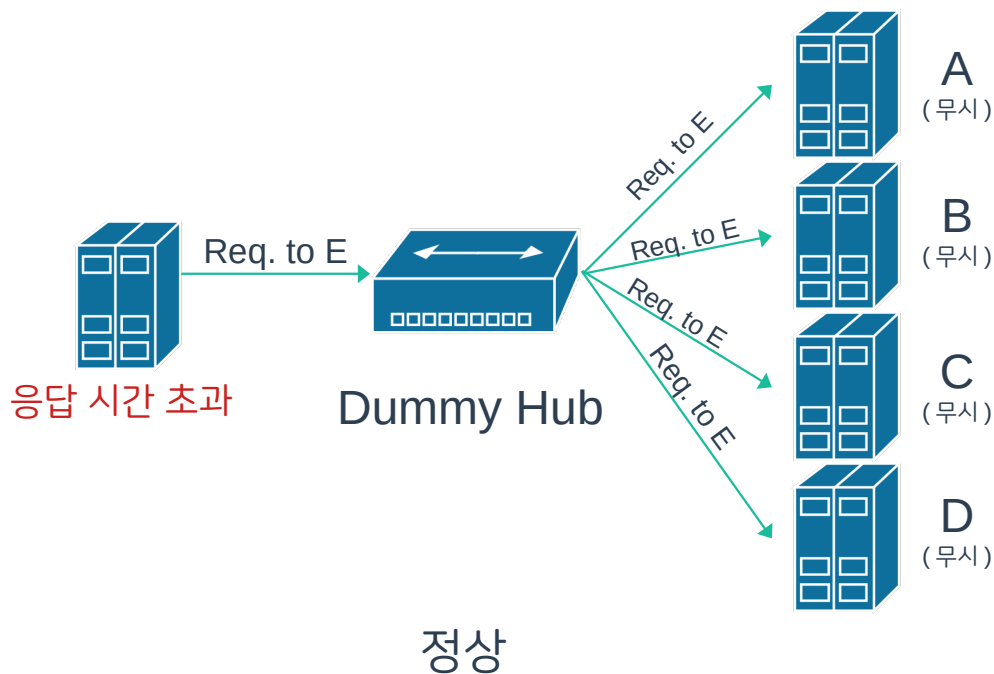
# ICMP 질의 메시지를 통한 탐지

- 대부분의 호스트는 별다른 설정을 하지 않았을 경우 **ICMP Echo Request** 메시지에 **ICMP Echo Reply** 메시지로 응답하게 되어있음 .
- 네트워크에 어떤 호스트가 있는지 알고 있을 때 , 존재하지 않는 호스트를 목적으로 하는 **ICMP Echo Request** 를 전송했을 때 응답이 있는지 확인 .

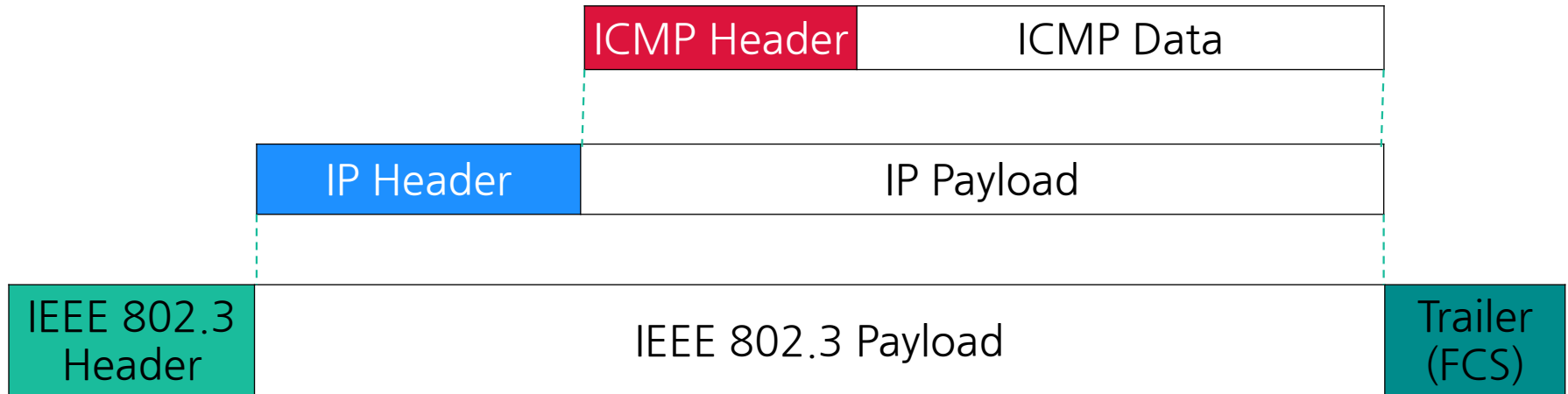
# ICMP 질의 메시지를 통한 탐지

- 존재하지 않는 호스트에 대해 응답이 돌아온다면 어떤 방법으로든 해당 요청을 다른 호스트가 비정상적으로 수신, 응답했다고 판단할 수 있음.
- 현재 통신 매체에 **Snipping** 하는 호스트가 있음.

# ICMP 질의 메시지를 통한 탐지



# ICMP 질의 메시지를 통한 탐지



# ICMP 질의 메시지를 통한 탐지

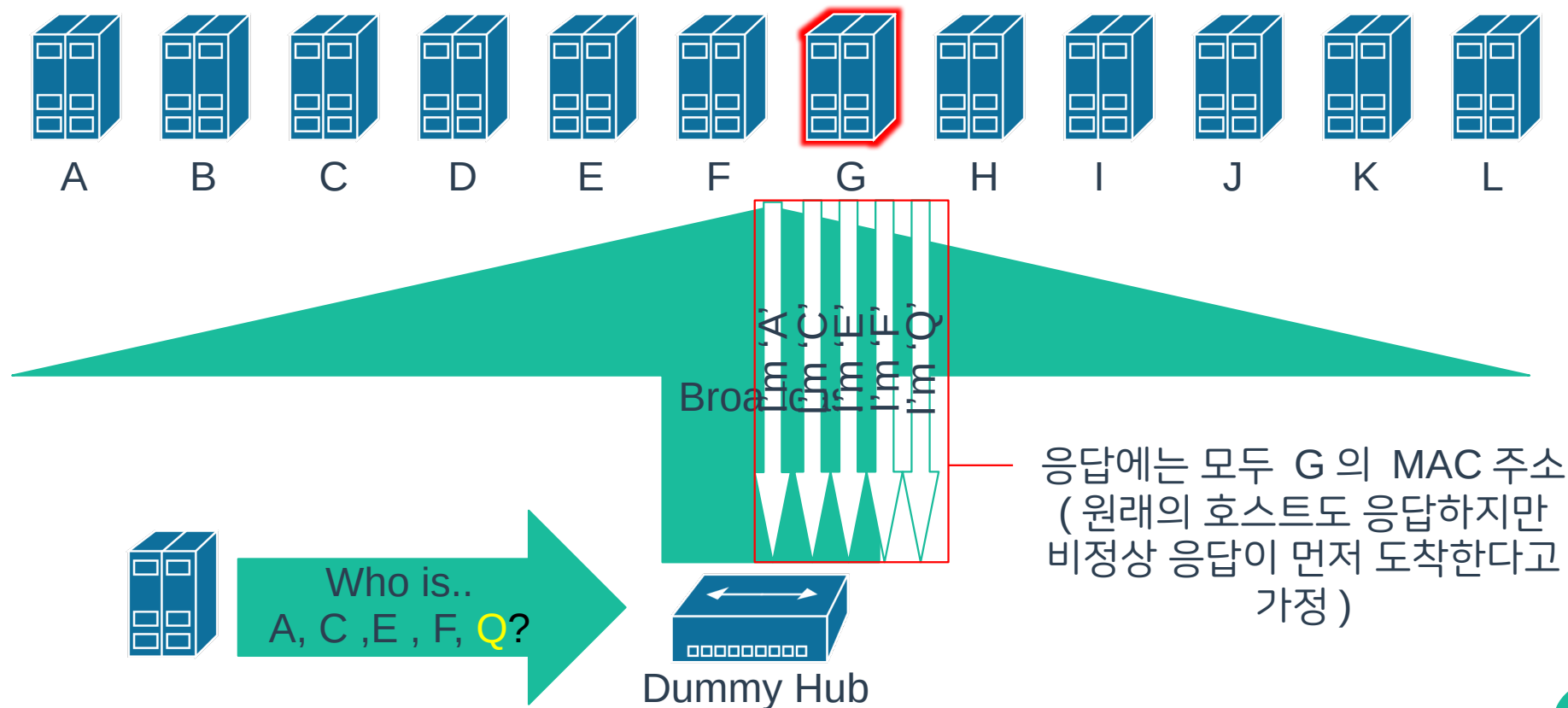


# ARP 질의를 통한 탐지

- **ICMP** 질의 메시지를 통한 탐지와 유사한 방법.
- 존재하지 않는 호스트가 있는지 질의하고 응답하는지 검사.



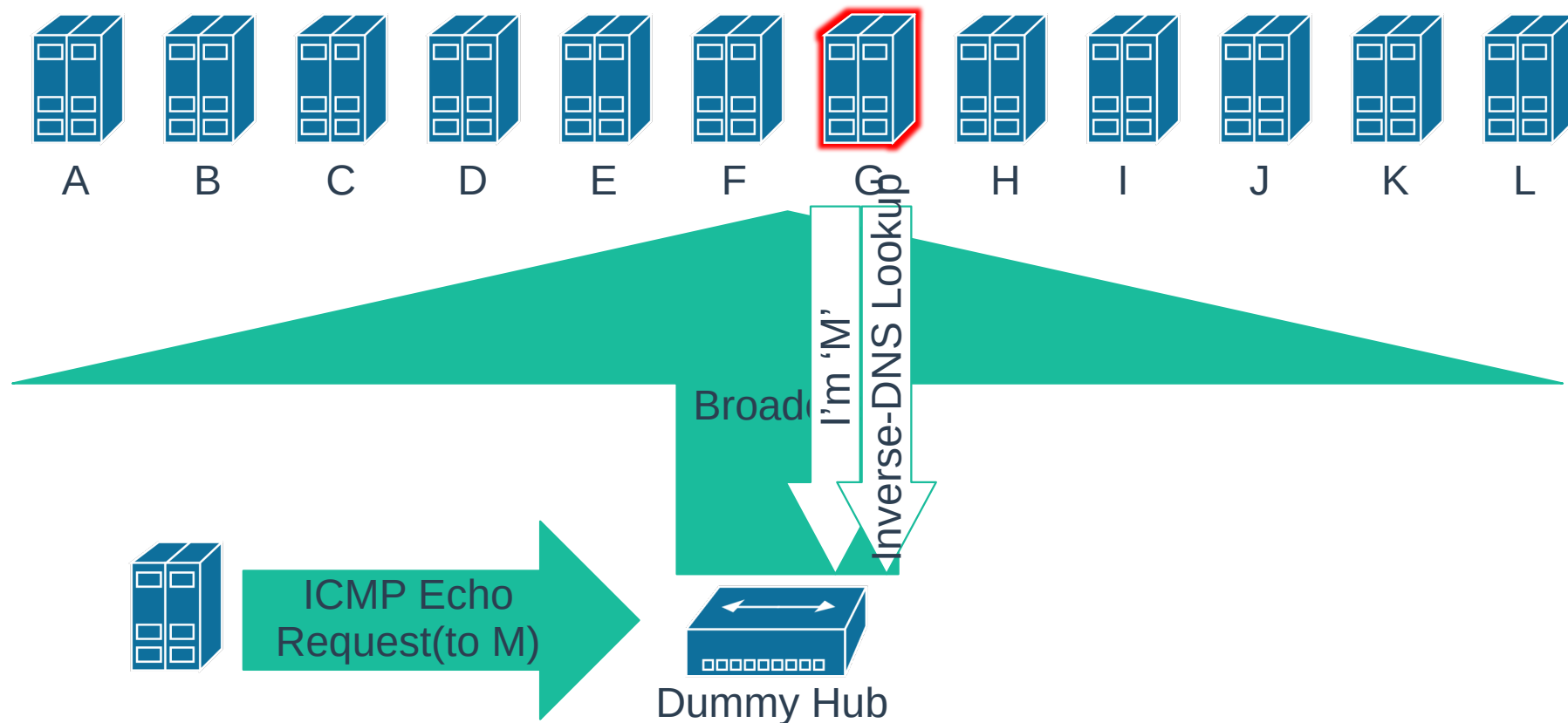
# ARP 질의를 통한 탐지



# Inverse DNS Lookup 을 통한 탐지

- 거의 대부분의 스니핑 도구는 편의를 위해 감지한 호스트의 호스트 이름을 표시해줌 .
- **IP** 주소로 부터 호스트 이름을 알아내려면 **Inverse-DNS Lookup** 요청을 해야함 .

# Inverse DNS Lookup 을 통한 탐지



# Inverse DNS Lookup 을 통한 탐지

No.	Time	Source
1	0.000000	
2	0.025276	
3	0.999754	
4	1.458644	
5	1.505537	
6	1.783866	
7	1.785212	
8	1.956291	
9	1.956521	
10	1.963245	1.1.1.1
11	1.963245	1.1.1.1



No.	Time	Source
1	0.000000	
2	0.025276	
3	0.999754	
4	1.458644	
5	1.505537	
6	1.783866	
7	1.785212	
8	1.956291	
9	1.956521	
10	1.963245	one.one.one.one
11	1.963245	one.one.one.one

네트워크 분석 도구인 Wireshark 에서 Sniffing 이후 Inverse-DNS Lookup 으로 IP 주소에서 호스트 이름으로 자동 치환되는 모습

# 공격자 유인을 통한 탐지

- **Sniffing** 의 목적이 네트워크 분석 등과 같은 합법적 목적일 수 있지만 합법적 **Sniffing** 의 경우는 탐지할 이유가 없다.
- 따라서 인가되지 않은 공격자가 있고, 악의적 목적을 가지고 네트워크를 **Sniffing** 하고 있다는 것이다.
- 공격자는 패킷으로 부터 유의미한 정보를 얻기 위한 목적일 것이고, 그 정보는 **ID/PW** 일 수도 있다.

## 공격자 유인을 통한 탐지

- 네트워크에 거짓 **ID/PW** 를 노출 시키고 공격자가 해당 **ID/PW** 로 어떤 행동을 하는지 추적하고 분석.
- **Sniffing** 탐지 및 공격 벡터 분석 이외 사용되지 않는 거짓 **ID/PW** 가 제 3 자에 의해 사용되었다는 사실은 공격자가 공격을 시도한다는 증거.

# 호스트 기반 탐지

- **Sniffing** 행위가 의심되는 호스트에서 직접 확인하는 방법
- 직접 의심되는 호스트에 물리적으로 접근, 호스트의 **Network Interface Card** 가 무차별 모드로 동작하고 있는지 검사.
- 무차별 모드만 탐지하고, 의심되는 호스트를 직접 확인해야 한다는 단점이 존재함.

# 호스트 기반 탐지

Windows 10, 11 환경에서 Powershell 을 통한 무차별 모드 확인

```
Get-NetAdapter | Format-List -Property PromiscuousMode, Name
```

```
PromiscuousMode : False  
Name           : 로컬 영역 연결
```

평상시

```
PromiscuousMode : True  
Name           : 로컬 영역 연결
```

Wireshark 에 의해 무차별  
모드로 동작 중일 때 ( 예시 )



# 호스트 기반 탐지

GNU/Linux 환경에서 무차별 모드 확인

```
$ ifconfig
```

```
ubuntu@Ubuntu-Virtual-Machine:~$ ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether [redacted] brd ff:ff:ff:ff:ff:ff
```

최근 배포판은 net-tools 가 iproute2 으로 대체된 경우가 많기에 직접 net-tools 를 설치하거나 iproute2 의 \$ ip link 명령어를 사용 (아마도 될 것으로 예상 ..?)

# 네트워크 기반 탐지

- 네트워크를 지속적 모니터링 하여 이상을 감지하는 방법
- 교재에서는 “**ARP watch**” 라는 도구를 사용해서 **ARP** 정보를 모니터링 하는 경우를 설명함

# 네트워크 기반 탐지

Index	IP	MAC
1	10.123.4.2	B4:4E:90:FA:D2:5B
2	10.123.4.3	B0:A9:44:A1:38:C8
3	10.123.4.4	E3:A0:75:80:C6:B5
4	10.123.4.65	7C:DF:E7:8F:F6:26
5	10.123.4.68	2D:91:E6:30:39:DD
6	10.123.4.233	DB:3A:B3:08:CA:89
7	10.123.4.112	0E:BA:0E:A8:50:50
8	10.123.4.109	E8:54:44:56:2C:A0
9	10.123.4.200	93:50:C2:B1:26:BE
10	10.123.4.203	B2:97:CC:2A:AC:42

정상

Index	IP	MAC
1	10.123.4.2	E3:A0:75:80:C6:B5
2	10.123.4.3	E3:A0:75:80:C6:B5
3	10.123.4.4	E3:A0:75:80:C6:B5
4	10.123.4.65	E3:A0:75:80:C6:B5
5	10.123.4.68	E3:A0:75:80:C6:B5
6	10.123.4.233	E3:A0:75:80:C6:B5
7	10.123.4.112	E3:A0:75:80:C6:B5
8	10.123.4.109	E3:A0:75:80:C6:B5
9	10.123.4.200	E3:A0:75:80:C6:B5
10	10.123.4.203	E3:A0:75:80:C6:B5

ARP Spoofing 을 통한 Sniffing 후

# Spoofing

호스트를 속이다 , Spoofing

# Spoofing

- 사전적 의미는 “속이다”
- 무결성이 보장되지 않거나 취약한 프로토콜에서 정보 조작을 통해 공격

# Spoofing

- 공격자가 수신자 행세를 함
  - 원래 수신자에게 가야 할 메시지를 가로챈
  - 수신자는 메시지를 전달 받지 못함 .
- 공격자가 송신자 행세를 함
  - 송신자 인척 메시지를 전송함 .
  - 수신자에게 조작된 메시지를 전송할 수 있음 .

# ARP Spoofing

주소 변환 프로토콜 , ARP 를 속이다 .

# ARP 개념

- 논리 주소인 **IP** 주소를 물리 주소인 **MAC** 으로 동적 매핑 하는데 사용되는 프로토콜



# ARP 구조

Hardware Type		Protocol Type
Hardware Length	Protocol Length	Operation
Sender Hardware Address		
Sender Protocol Address		
Target Hardware Address		
Target Protocol Address		

Hardware Type:

LAN의 유형, Ethernet의 경우는 1

Protocol Type:

프로토콜의 유형, IPv4의 경우는 2048

Hardware Length:

물리 주소의 길이 (Byte 단위)

Protocol Length:

프로토콜 주소의 길이 (Byte 단위)

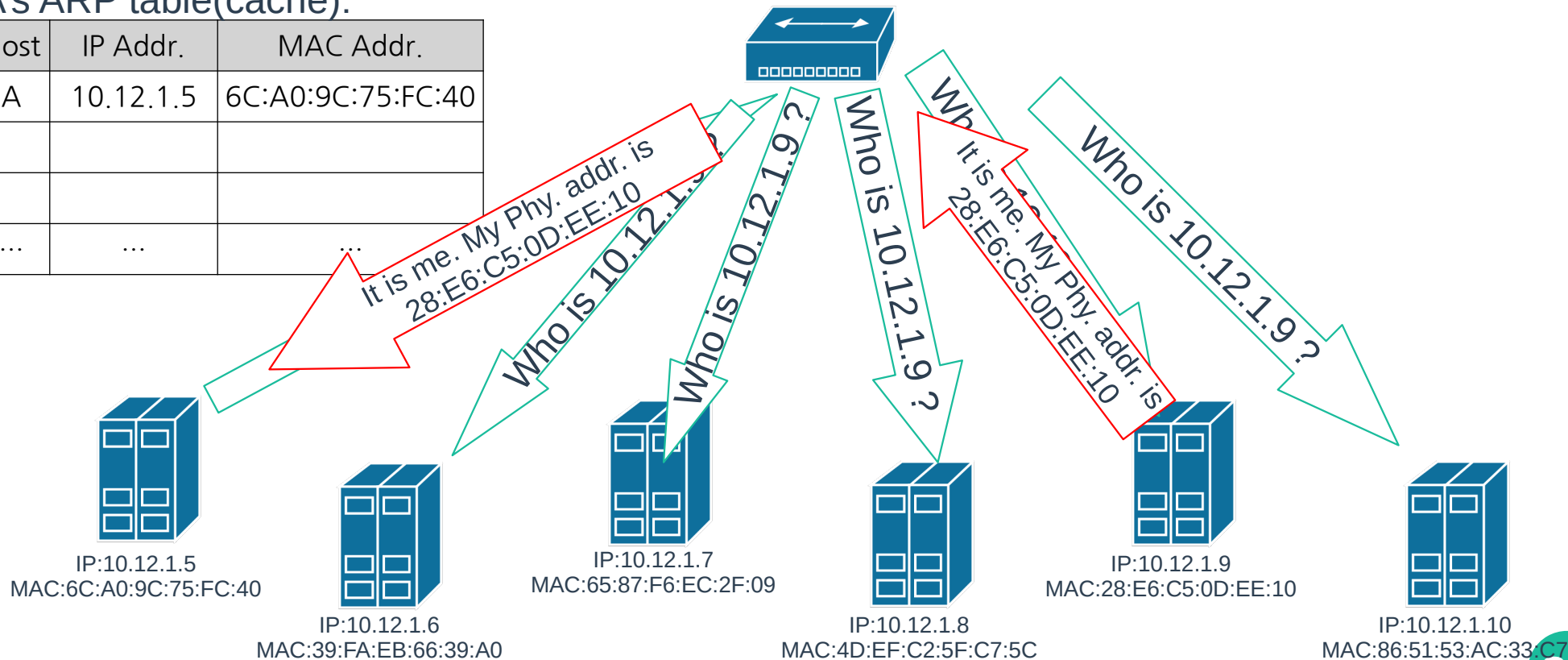
Operation:

패킷 유형, 요청 (1)/ 응답 (2)

# ARP 동작

A's ARP table(cache):

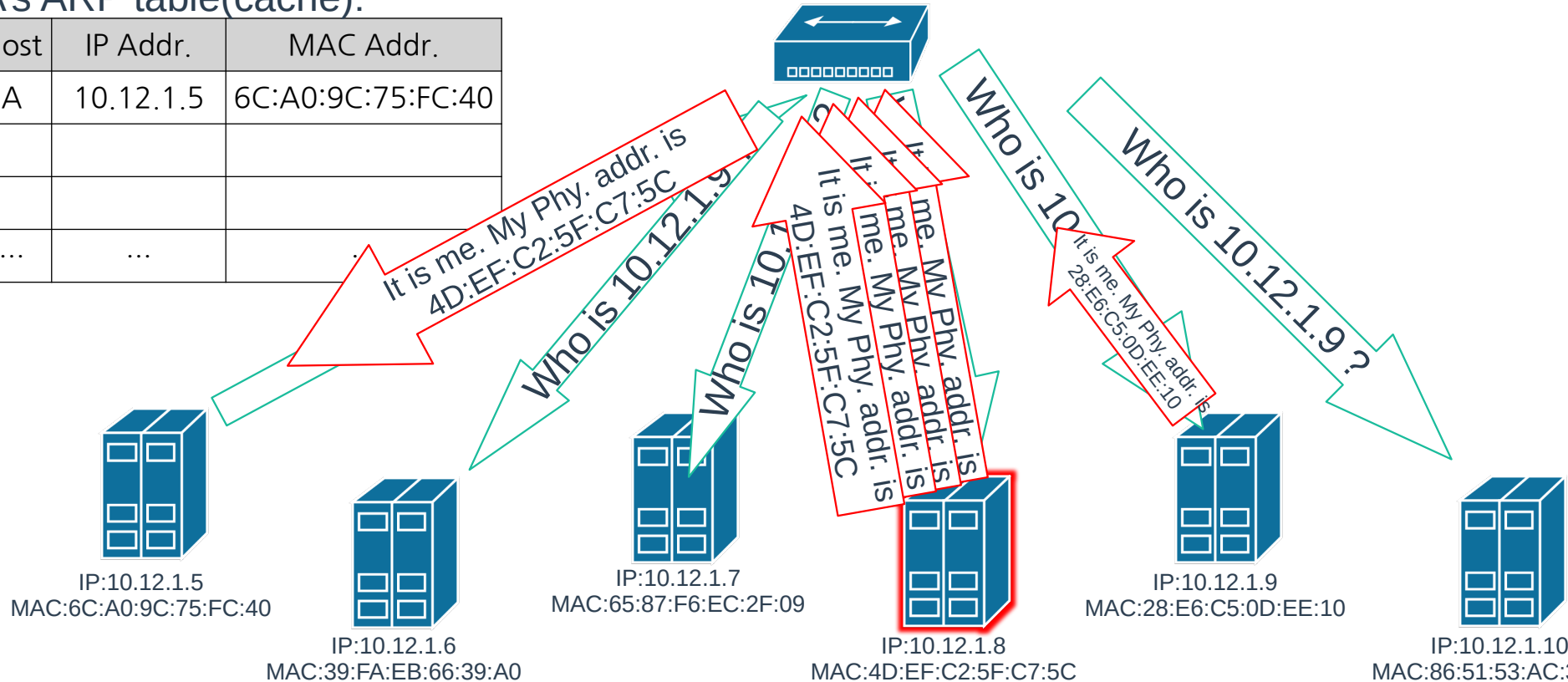
Host	IP Addr.	MAC Addr.
A	10.12.1.5	6C:A0:9C:75:FC:40
...	...	...



# ARP Spoofing

A's ARP table(cache):

Host	IP Addr.	MAC Addr.
A	10.12.1.5	6C:A0:9C:75:FC:40
...	...	

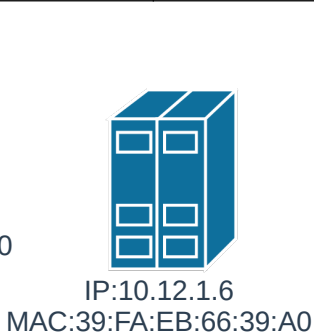


# ARP Spoofing

Host	IP Addr.	MAC Addr.
local host	Local IP	Local MAC
A	10.12.1.5	6C:A0:9C:75:FC:40
B	10.12.1.6	39:FA:EB:66:39:A0
C	10.12.1.7	65:87:F6:EC:2F:09
D	10.12.1.8	4D:EF:C2:5F:C7:5C
E	10.12.1.9	28:E6:C5:0D:EE:10
F	10.12.1.10	86:51:53:AC:33:C7

공격 후

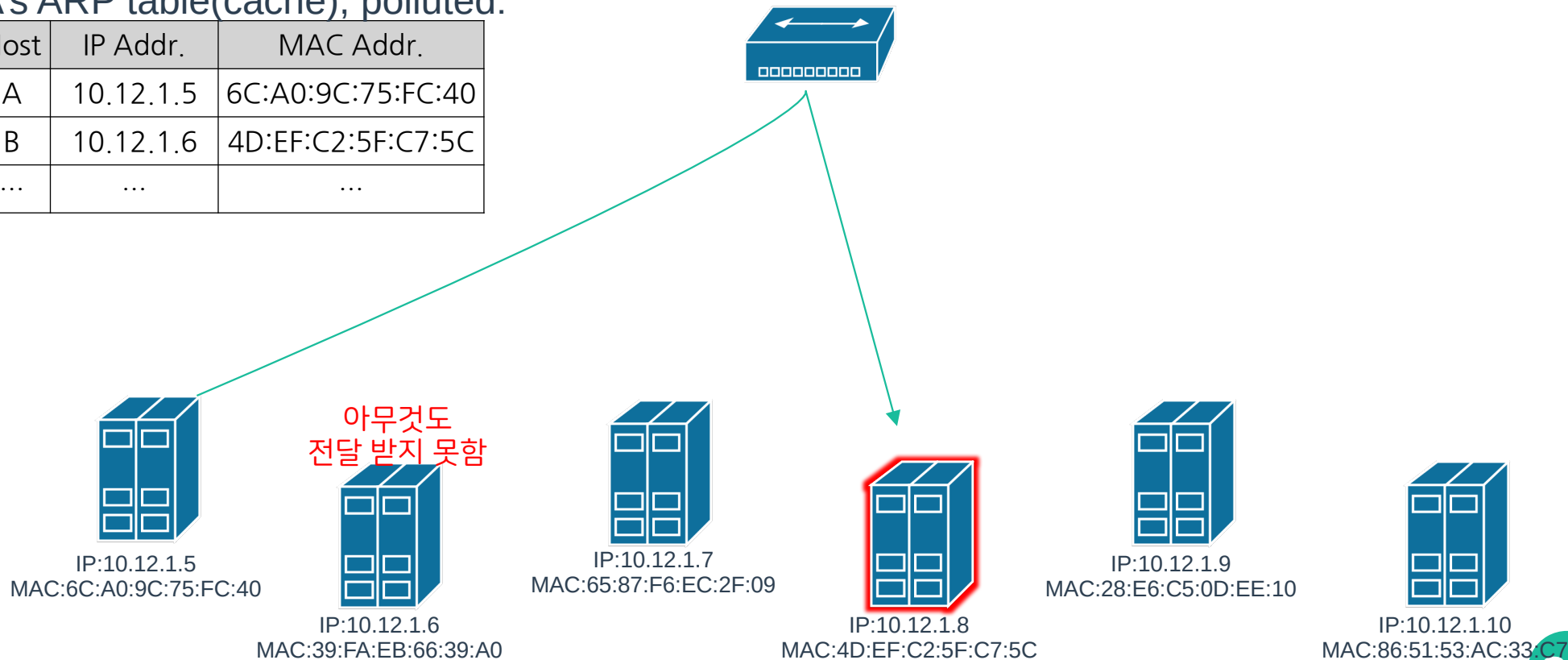
Host	IP Addr.	MAC Addr.
local host	Local IP	Local MAC
A	10.12.1.5	4D:EF:C2:5F:C7:5C
B	10.12.1.6	4D:EF:C2:5F:C7:5C
C	10.12.1.7	4D:EF:C2:5F:C7:5C
D	10.12.1.8	4D:EF:C2:5F:C7:5C
E	10.12.1.9	4D:EF:C2:5F:C7:5C
F	10.12.1.10	4D:EF:C2:5F:C7:5C



# ARP Spoofing

A's ARP table(cache), polluted:

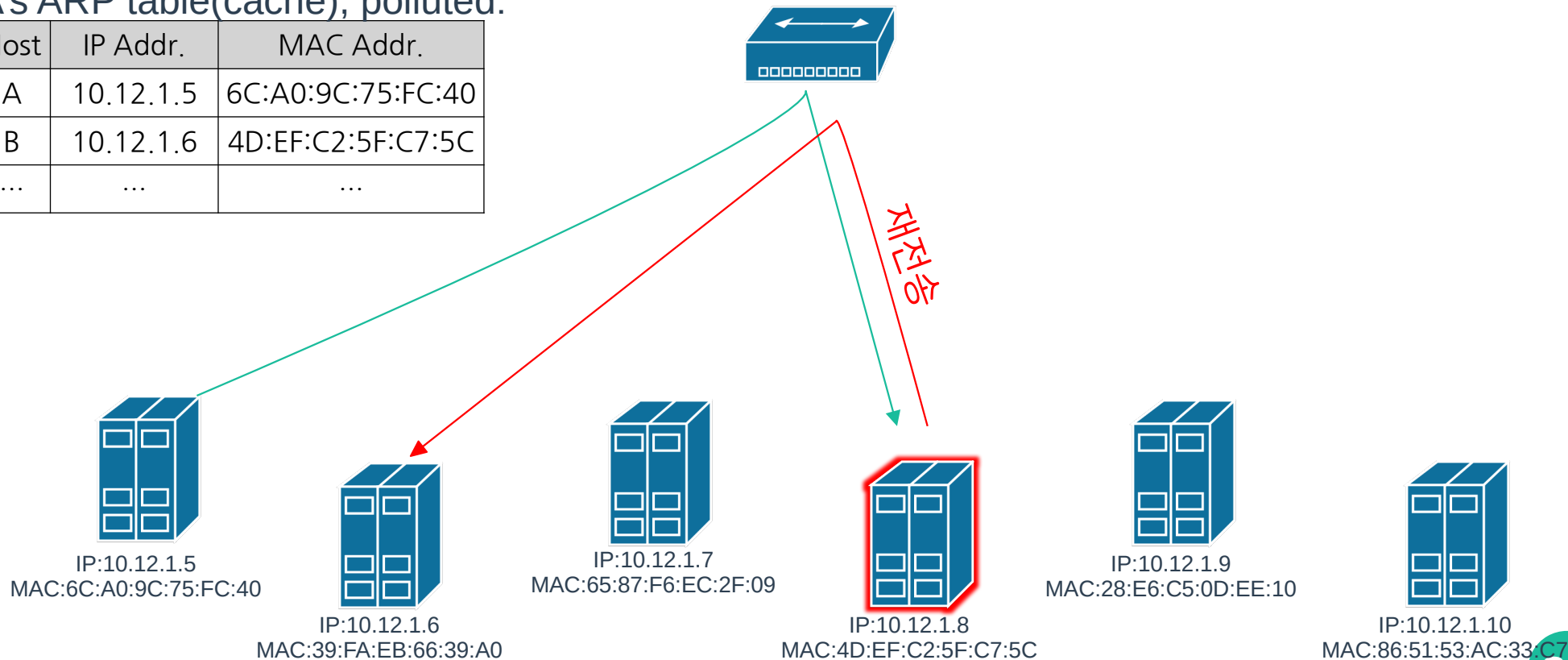
Host	IP Addr.	MAC Addr.
A	10.12.1.5	6C:A0:9C:75:FC:40
B	10.12.1.6	4D:EF:C2:5F:C7:5C
...	...	...



# ARP Spoofing

A's ARP table(cache), polluted:

Host	IP Addr.	MAC Addr.
A	10.12.1.5	6C:A0:9C:75:FC:40
B	10.12.1.6	4D:EF:C2:5F:C7:5C
...	...	...



# ARP Spoofing Detection

ARP Spoofing 을 감지하는 몇 가지의 방법들

# ARP Spoofing 시 발생하는 이상 증상

- 지속적인 **ARP** 응답 발생
- **ARP** 테이블에서 중복된 **MAC** 주소 확인
  - ARP 테이블 감시 프로그램 활용
- 네트워크 속도 저하



## 지속적인 **ARP** 응답 발생

- 공격자는 정상 **ARP** 응답보다 조작된 응답이 희생자에 컴퓨터에 먼저 도달하도록 해야 공격에 성공할 가능성이 증가함.
- 그렇기 때문에 공격자는 **ARP** 응답을 주기적으로 그리고 많이 전송함.
- 비 정상적으로 많은 **ARP** 패킷이 탐지되면 공격자가 **ARP Spoofing** 공격을 수행하고 있다고 의심해 볼 수 있음.

# ARP 테이블에서 중복된 MAC 주소 확인

- 특별한 상황이 아니거나 **IP** 주소와 **MAC** 주소는 **1:1** 부합.
- 여러 개의 **IP** 주소가 하나의 **MAC** 주소를 가리킨다면 각각의 호스트로 가야 할 메시지들이 하나의 호스트로 전송되고 있는 것일 수 있음.
- 중복되는 하나의 **MAC** 주소가 **Spoofing** 중인 호스트의 **MAC** 주소.

# ARP 테이블에서 중복된 MAC 주소 확인

- Windows 환경에서 **arp -a** 명령을 통해 ARP 테이블 확인이 가능하다.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 기능 및 개선 사항에 대한 최신 PowerShell을 설치하세요! https://aka.ms/PSWindows

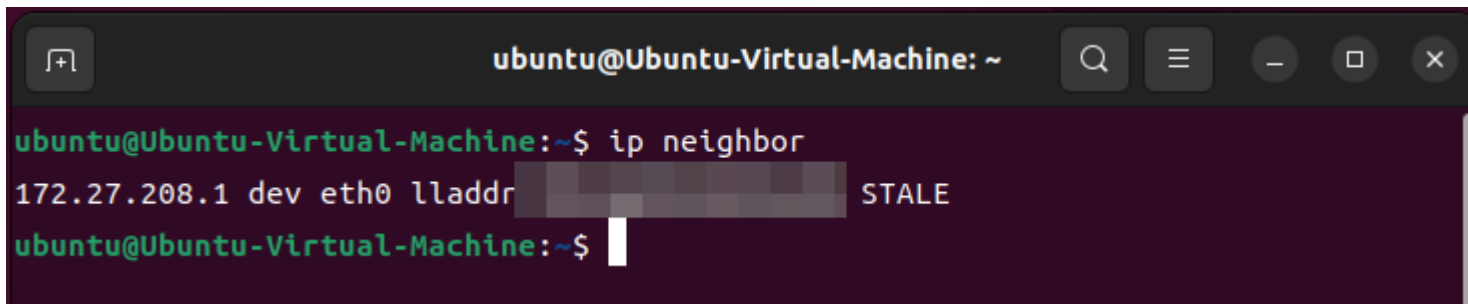
PS C:\Users\> arp -a

인터페이스: 10.34.248.180 --- 0x1d
인터넷 주소      물리적 주소      유형
10.34.248.180     [REDACTED]        동적
10.34.248.255     [REDACTED]        정적
224.0.0.2         [REDACTED]        정적
224.0.0.22        [REDACTED]        정적
224.0.0.251       [REDACTED]        정적
239.100.0.1       [REDACTED]        정적
239.255.255.250   [REDACTED]        정적

인터페이스: 192.168.0.14 --- 0x24
인터넷 주소      물리적 주소      유형
192.168.0.1       [REDACTED]        동적
192.168.0.2       [REDACTED]        동적
192.168.0.6       [REDACTED]        동적
192.168.0.8       [REDACTED]        동적
192.168.0.255     [REDACTED]        정적
224.0.0.2         [REDACTED]        정적
224.0.0.22        [REDACTED]        정적
224.0.0.251       [REDACTED]        정적
224.0.0.252       [REDACTED]        정적
224.0.1.187       [REDACTED]        정적
239.100.0.1       [REDACTED]        정적
```

# ARP 테이블에서 중복된 MAC 주소 확인

- **GNU/Linux** 환경에서 “**arp -a**”(net-tools) 명령 또는 “**ip neighbour**”(iproute2) 명령으로 **ARP** 테이블 확인이 가능하다.

A terminal window titled 'ubuntu@Ubuntu-Virtual-Machine: ~' with standard window controls. The prompt is 'ubuntu@Ubuntu-Virtual-Machine:~\$'. The command 'ip neighbor' has been entered, and the output is '172.27.208.1 dev eth0 lladdr [redacted] STALE'. The prompt is now 'ubuntu@Ubuntu-Virtual-Machine:~\$' with a cursor.

```
ubuntu@Ubuntu-Virtual-Machine: ~
ubuntu@Ubuntu-Virtual-Machine:~$ ip neighbor
172.27.208.1 dev eth0 lladdr [redacted] STALE
ubuntu@Ubuntu-Virtual-Machine:~$
```

# ARP 테이블에서 중복된 MAC 주소 확인

- 사람이 매번 **ARP** 테이블을 모니터링 하는 건 상당히 불편하고 실수가 발생할 가능성이 높음  
→ **ARP** 테이블 모니터링을 자동화 하자!
- 교재에서 소개된 도구
  - Xarp
  - arpwatch

# 네트워크 속도 저하

- **ARP Spoofing** 시 지속적인 **ARP** 응답 발생으로 인한 트래픽 증가
- 공격 호스트를 거쳐서 **PDU** 가 전달되기 때문에 그 만큼의 지연시간 증가

# ARP Spoofing 방지 대책

- 정적 **ARP Table** 관리

- ARP Spoofing 은 MAC 주소 변조가 가능하기에 할 수 있는 공격  
MAC 주소 변조가 불가능하게 정적으로 관리 .
  - 장점 :ARP Spoofing 을 해결할 수 있는 간단한 방법  
단점 : 네트워크를 구성하는 호스트에 변경이 생길 경우 네트워크에 속한 모든 호스트의 ARP Table 수정이 필요함 .

- 네트워크 내 호스트 보안 수준 강화

- 네트워크 내 호스트가 공격자에게 탈취당해 Bot 으로 동작할 경우 해당 Bot 이 연결된 네트워크는 ARP Spoofing 공격이 가능해짐 .

# IP Spoofing

IP 를 속이다



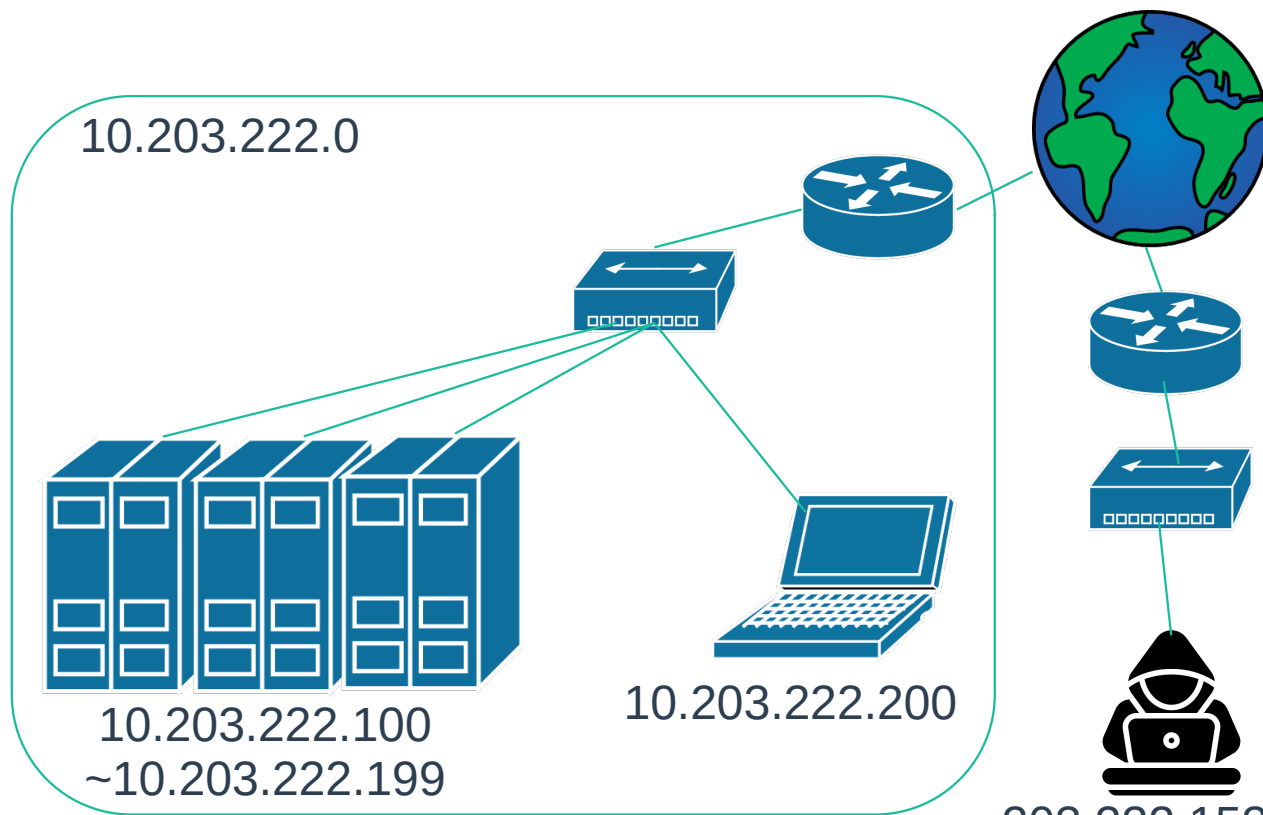
# IP Spoofing

- **IP** 주소를 변조하는 공격 방법
  - IP 헤더의 주소 필드를 변조
  - 2 계층의 MAC 주소를 변조하는 ARP Spoofing 보다 공격 가능 범위가 넓음 .

# IP Header

0		3		7		15		31	
Version		Header Length		DS/ECN		Total Length			
Identifier				0	D F	M F	Fragment Offset		
Time To Live		Protocol ID		Header Checksum					
Source IP Address									
Destination IP Address									
Options (if available)									
Data (if available)									

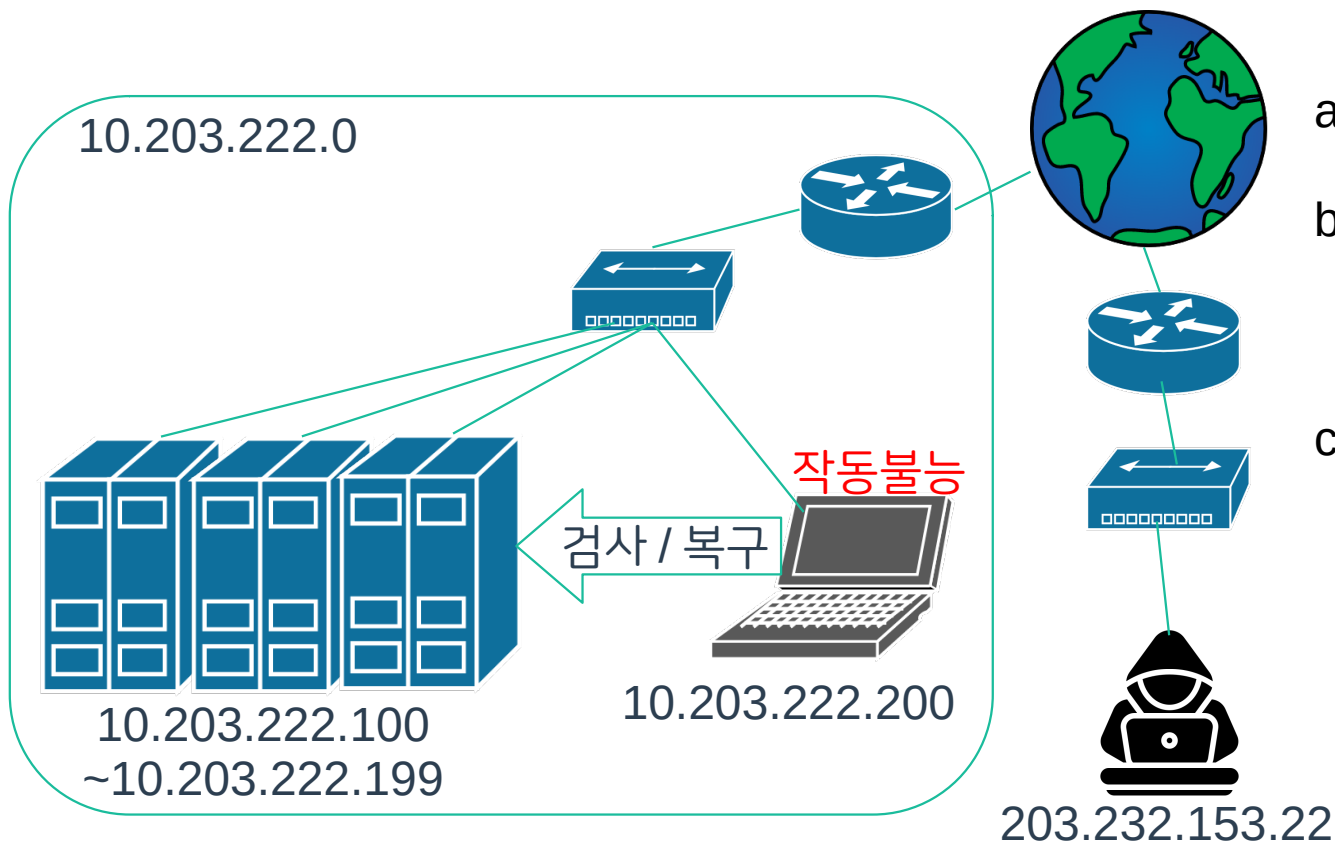
# IP Spoofing



[ 공격 시나리오 ( 예 ) ]

- 어느 기업의 서버들은 관리를 위한 Dumb Terminal 이 사용되고 있다 .
- 관리자 편의를 위해 암호 기반의 인증 대신 IP 기반의 인증을 사용하고 있고 서버는 10.203.222.200 로 부터 도착하는 관리자 명령을 신뢰하고 받아들이게 되어있다 .
- 공격자는 Source IP 주소 필드를 Dumb Terminal 의 IP 주소로 변조하여 기업의 네트워크로 전송한다 .

# IP Spoofing



[ 공격 시나리오 ( 예 ) ]

- 이전 슬라이드와 비슷한 환경
- Dumb Terminal 에서 주기적으로 보안 정책을 확인하고 이상을 발견하면 원래 정책으로 덮어쓰기 .
- 공격자는 DoS 공격을 통해 Dumb Terminal 의 보안 정책 복구 기능이 작동하지 않게 한 뒤 IP Spoofing 을 통해 원격 명령 실행 .

# IP Spoofing 의 합법적 사용

- 웹 성능 테스트

- HP(Hewlett-Packard) 의 LoadRunner 는 IP Spoofing 을 사용해서 서로 다른 IP 주소를 가진 다수의 여러 사용자가 접속하는 듯한 효력을 내고 이 결과를 토대로 웹 서비스의 성능을 평가합니다 .

# IP Spoofing 방어

- **IP 기반 Trust 사용하지 않기**

- IP 주소는 Spoofing 이 가능하기 때문에 취약 , 사용하지 않음 .
- 그럼에도 트러스트를 사용해야 한다면 트러스트 관계의 호스트들의 보안 수준을 강화해야 함 .( 보안 업데이트 , 정기점검 , 모니터링 )

- **Packet Filtering**

- 네트워크 경계 (Border) 를 기준 , 인바운드 패킷 중 소스 IP 주소가 내부 IP 주소 대역으로 설정된 패킷이 있다면 해당 패킷을 Drop.
  - 이 방법은 내부에서 발생하는 IP Spoofing 에 대응하지 못함 .

## HTTPS IP Spoofing 이 사용되는 예 ( 번외 )

Index	Deny Domain
0	XsecurityVideos.com
1	NetworkSecurityHub.com
...	...

# Match!

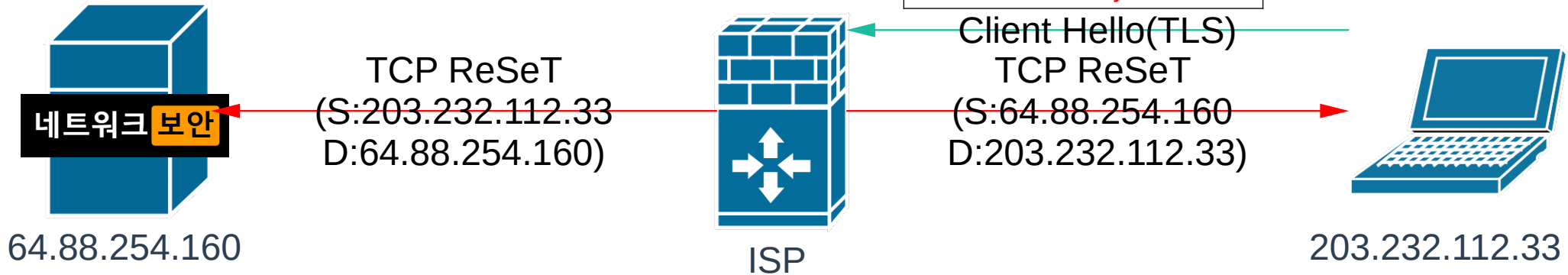
```
server_name
```

- NetworkSecurityHub.com

## Client Hello(TLS)

# TCP ReSeT

(S:64.88.254.160  
D:203.232.112.33)



# ICMP Spoofing

ICMP 를 속이다



# ICMP

- 정의

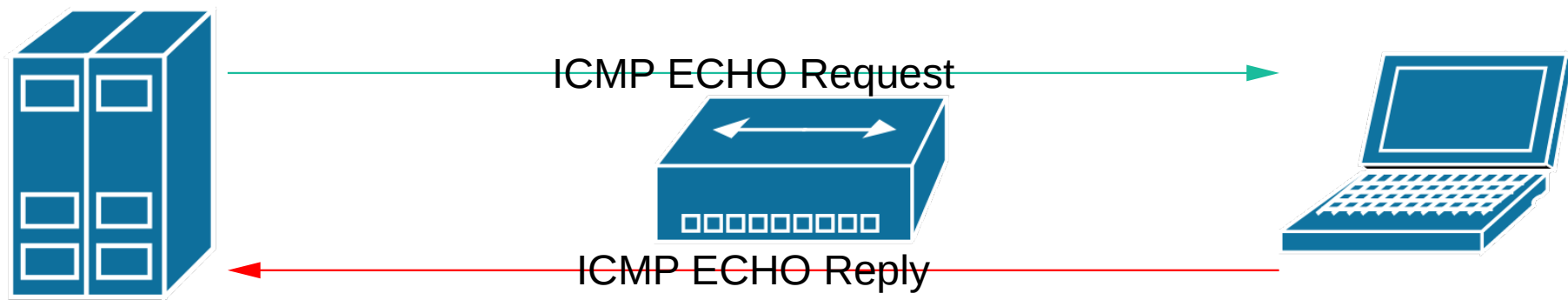
- Internet Control Message Protocol ( 인터넷 제어 메시지 프로토콜 )

- 용도

- 네트워크 진단
- 네트워크 흐름 통제

# ICMP

- 네트워크 진단

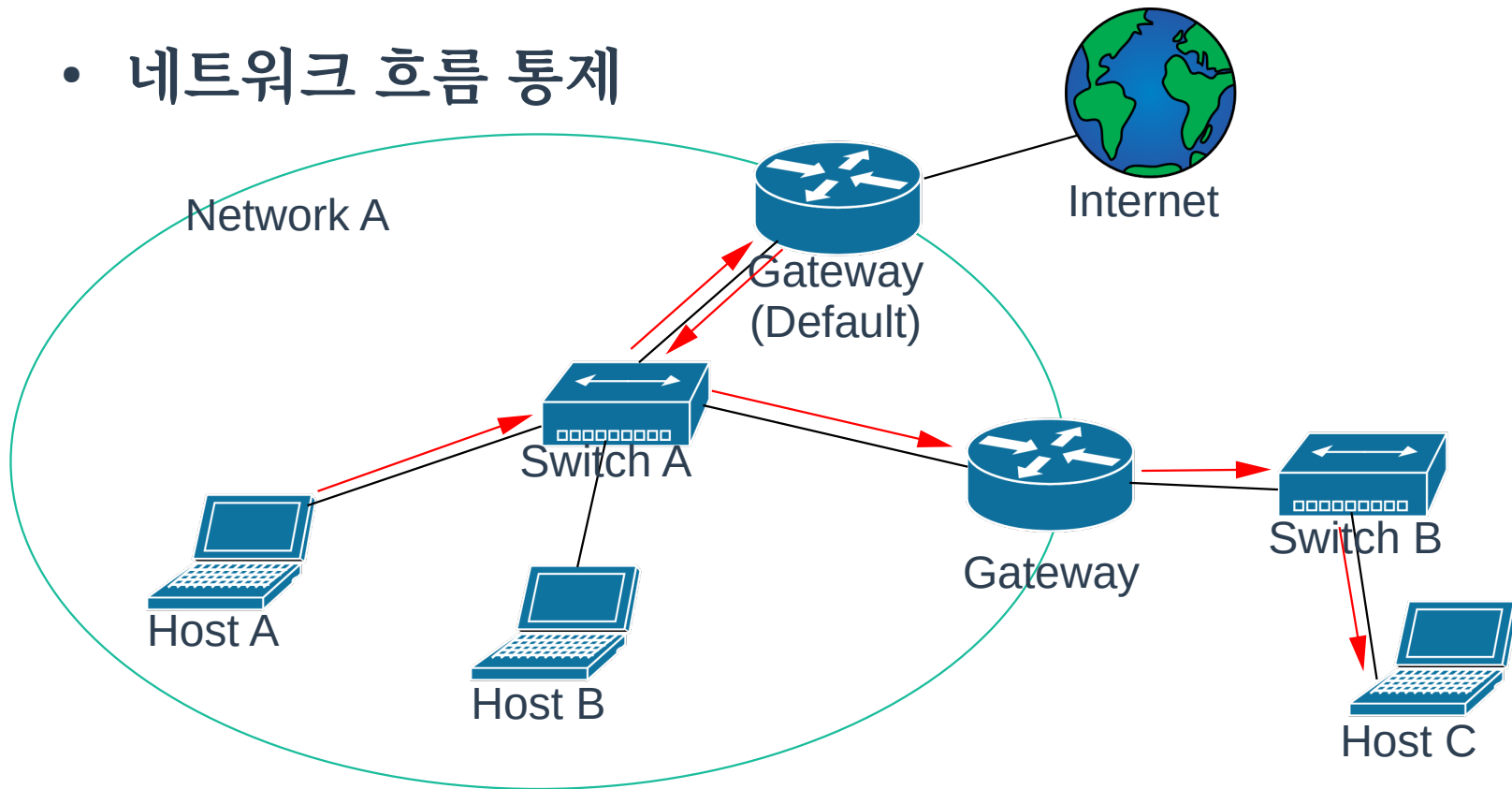


# ICMP ECHO



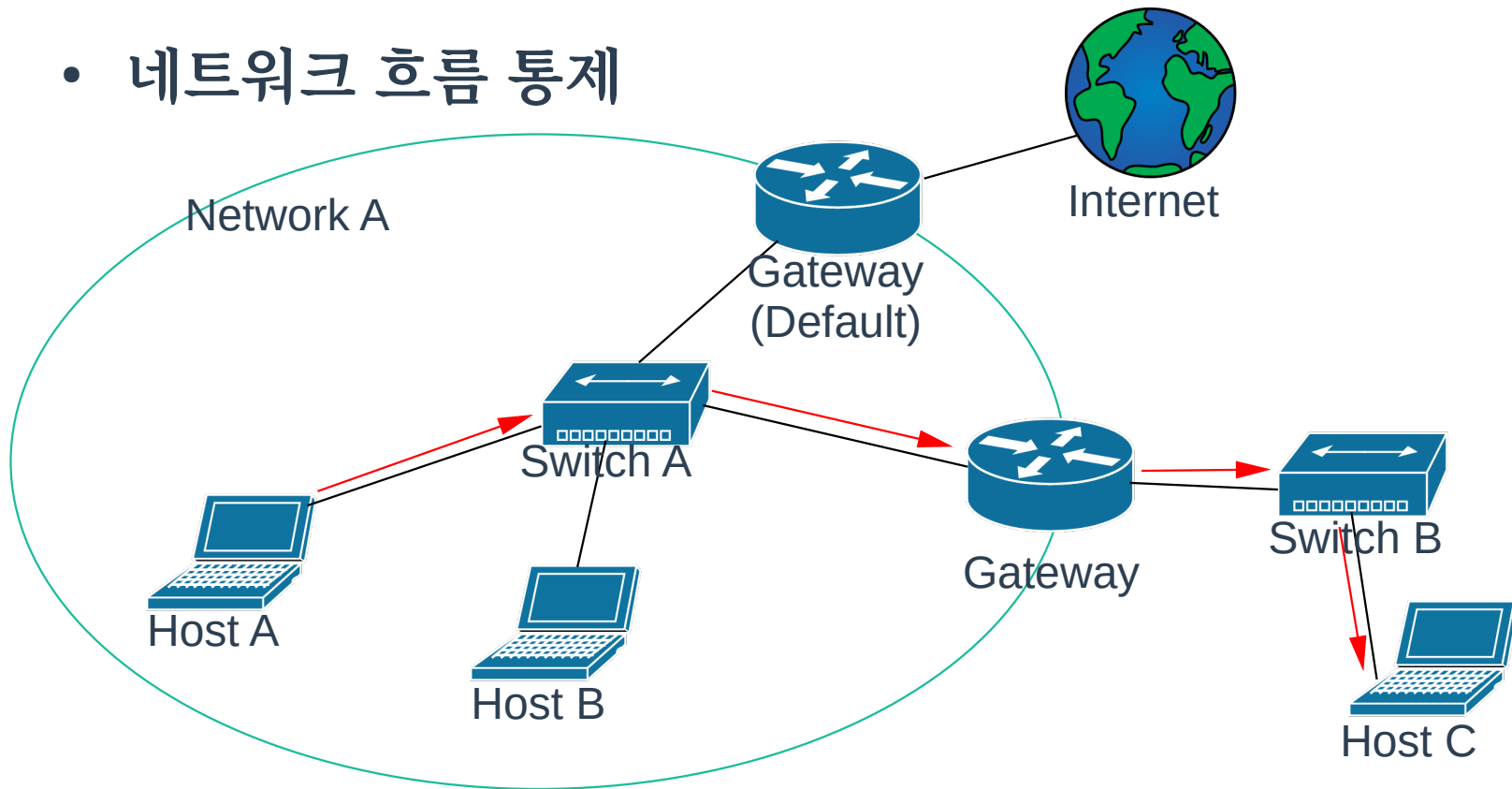
# ICMP

- 네트워크 흐름 통제

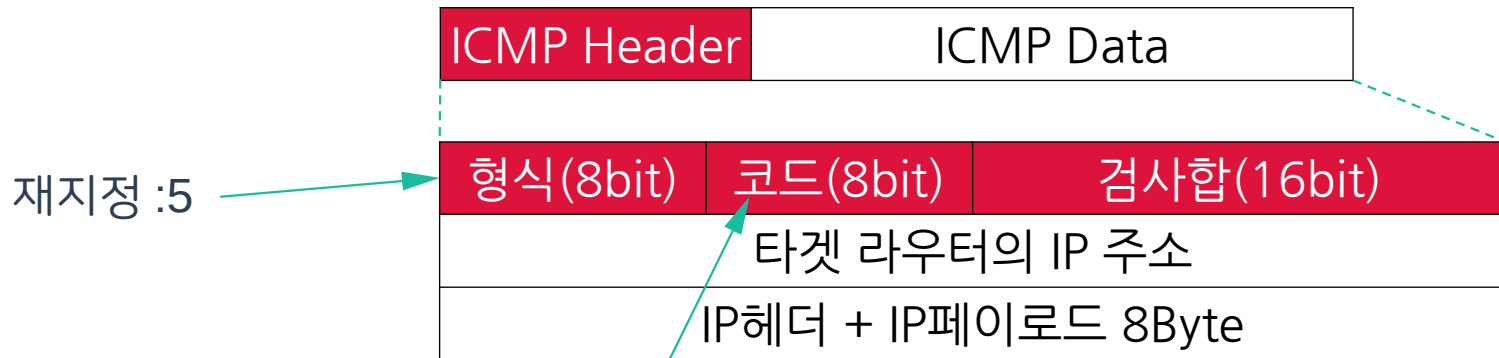


# ICMP

- 네트워크 흐름 통제



# ICMP Redirection



0: 네트워크 지정 경로를 위한 재지정

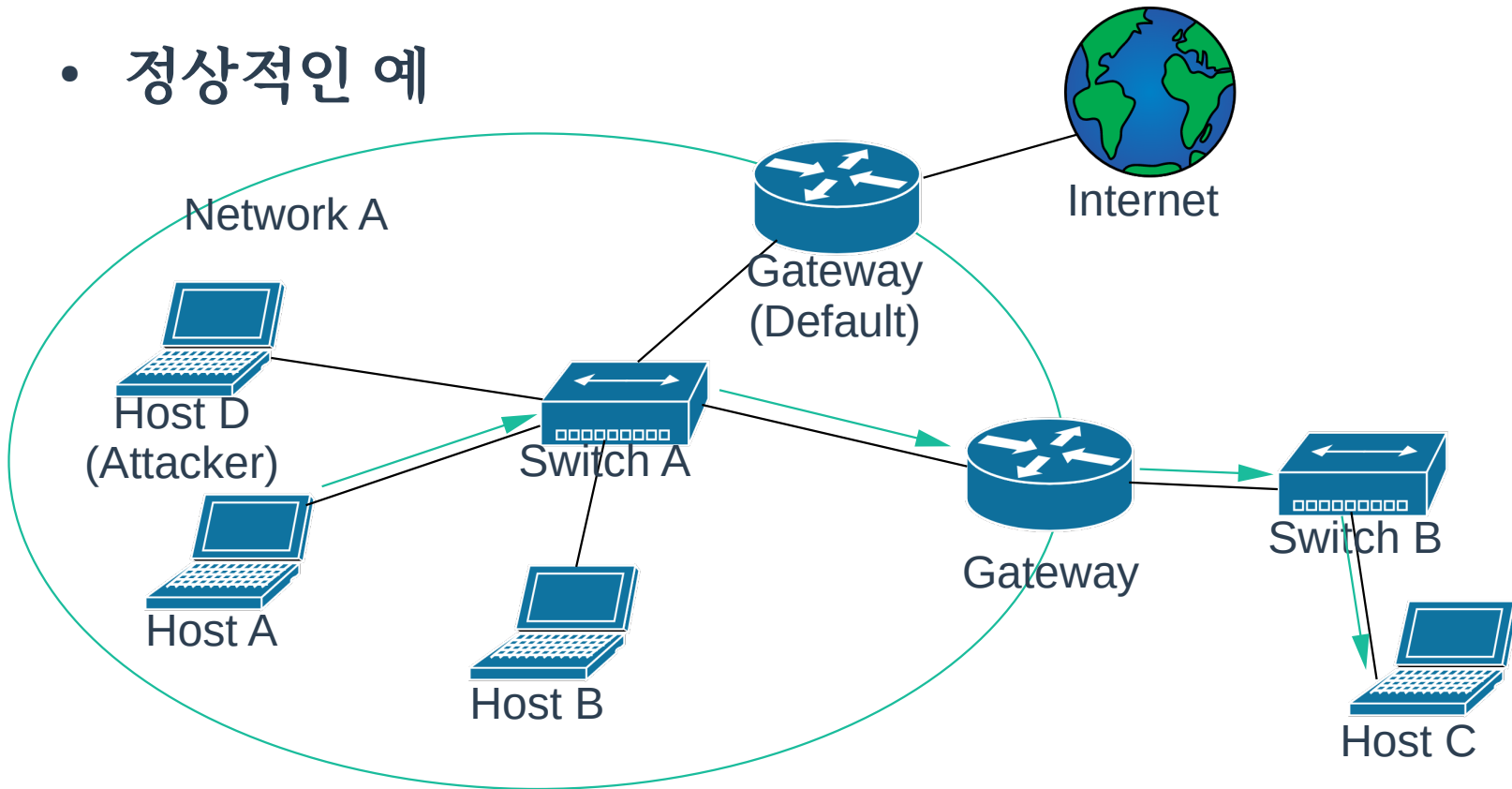
1: 호스트 지정 경로를 위한 재지정

2: 특정한 서비스 유형에 기초한 네트워크 지정 경로를 위한 재지정

3: 특정한 서비스 유형에 기초한 호스트 지정 경로를 위한 재지정

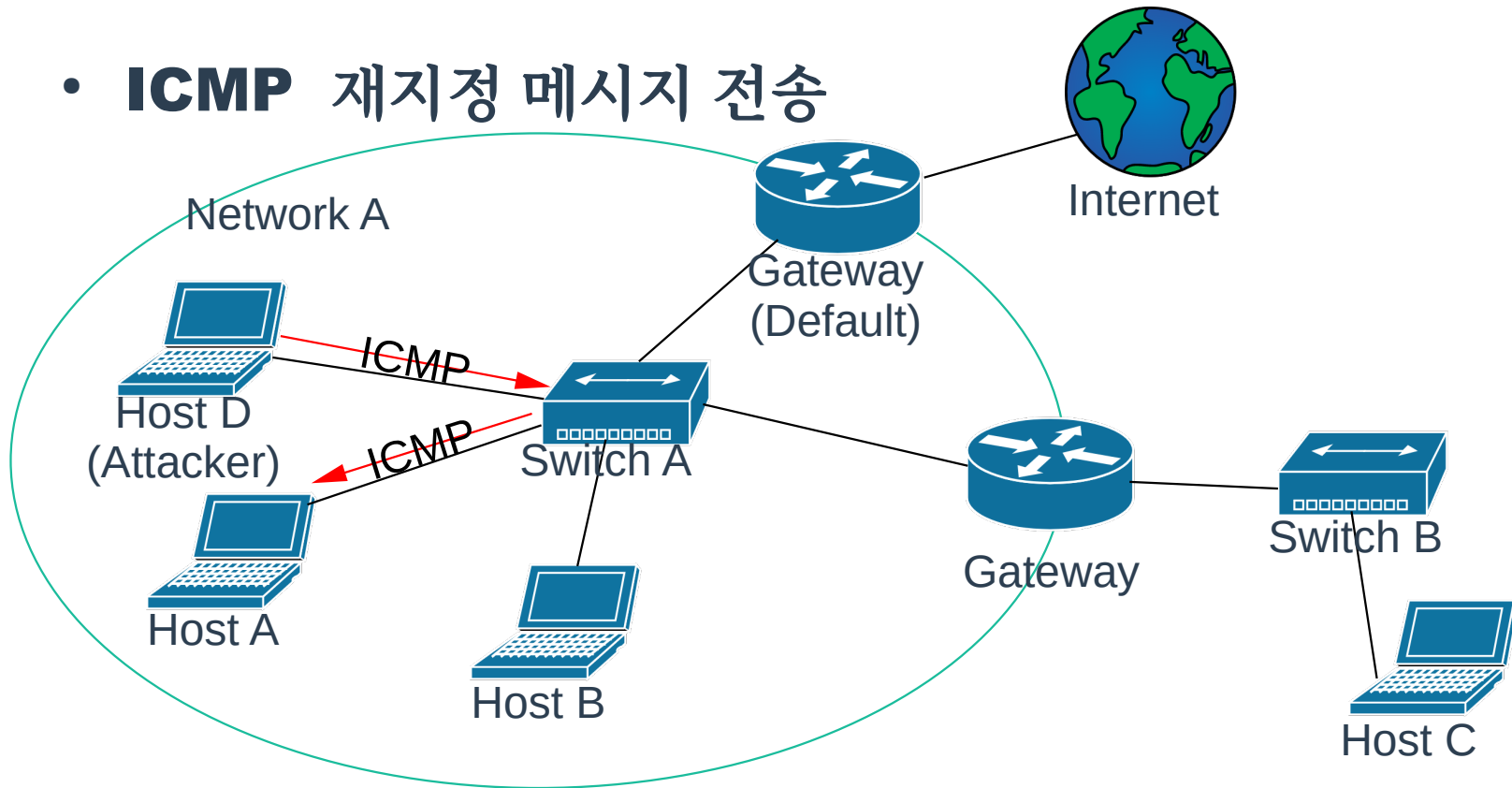
# ICMP Spoofing

- 정상적인 예



# ICMP Spoofing

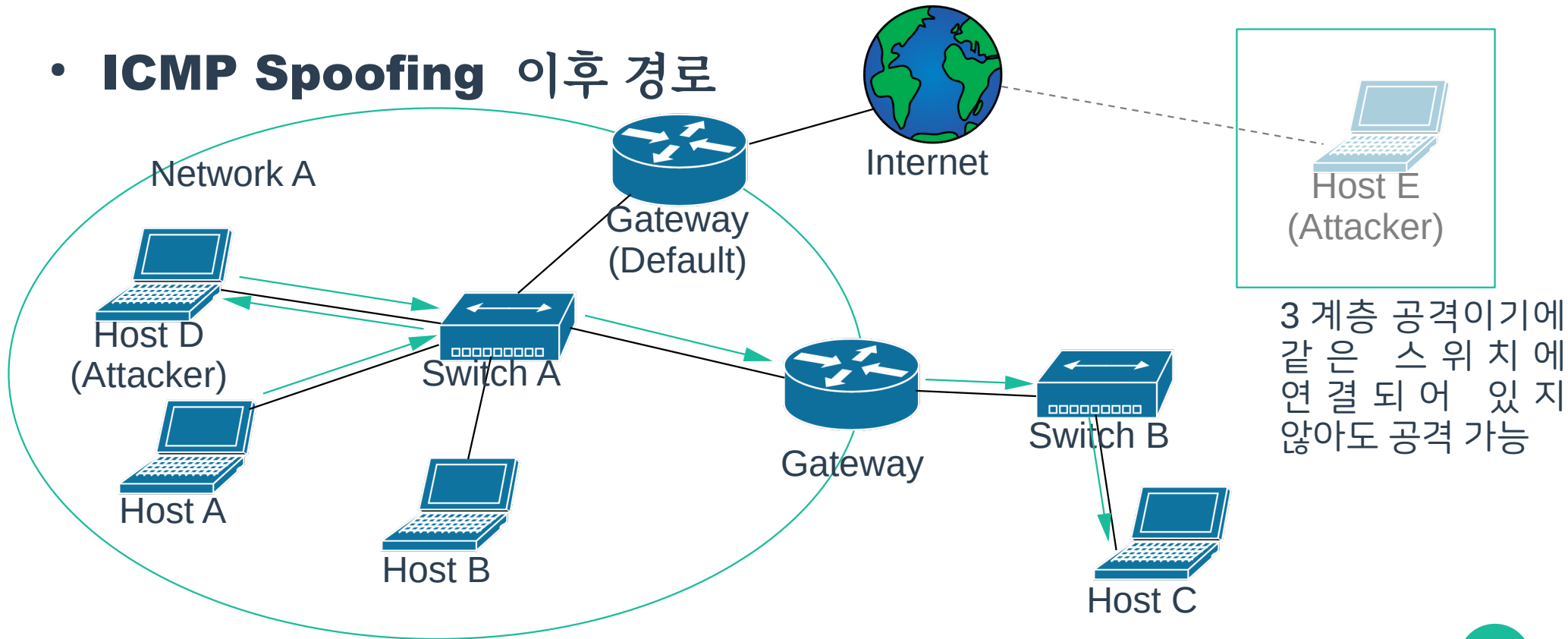
- **ICMP** 재지정 메시지 전송





# ICMP Spoofing

## • ICMP Spoofing 이후 경로



3 계층 공격이기에  
같은 스위치에  
연결되어 있지  
않아도 공격 가능

# ICMP Spoofing 방어

- **ICMP Redirection** 메시지 차단

- ICMP Redirection 메시지에 의해서 기본 게이트웨이가 변경되지 않도록 함 .

# DNS Spoofing

DNS 를 속이다

# DNS

- **DNS**

- Domain Name System
- 문자로 된 이름을 숫자인 주소 체계로 변환하는 시스템

# DNS Message

조회 메시지  
0

0																16																32															
Identification																Q R	OPCode				A A	T C	R D	R A	0	0	0	rCode																			
질문 레코드 수																응답 레코드 수																															
권한 레코드 수																추가 레코드 수																															
조회 이름(가변적)																																															
조회 유형																조회 클래스																															

QR(Query/Response):

OpCode:

AA(Authoritative Answer):

TC(Truncated):

RD(Recursion Desired):

RA(Recursion Available):

Query(0)/Response(1)

표준 (0)/ 역조회 (1)/ 서버상태요청 (2)

(0)/ 권한인정 (1)

512 바이트 이내 (0), 512 바이트 이상 , 잘림 (1)

(0)/ 재귀 응답 요청 (1)

(0)/ 반복 응답 가능 (1)

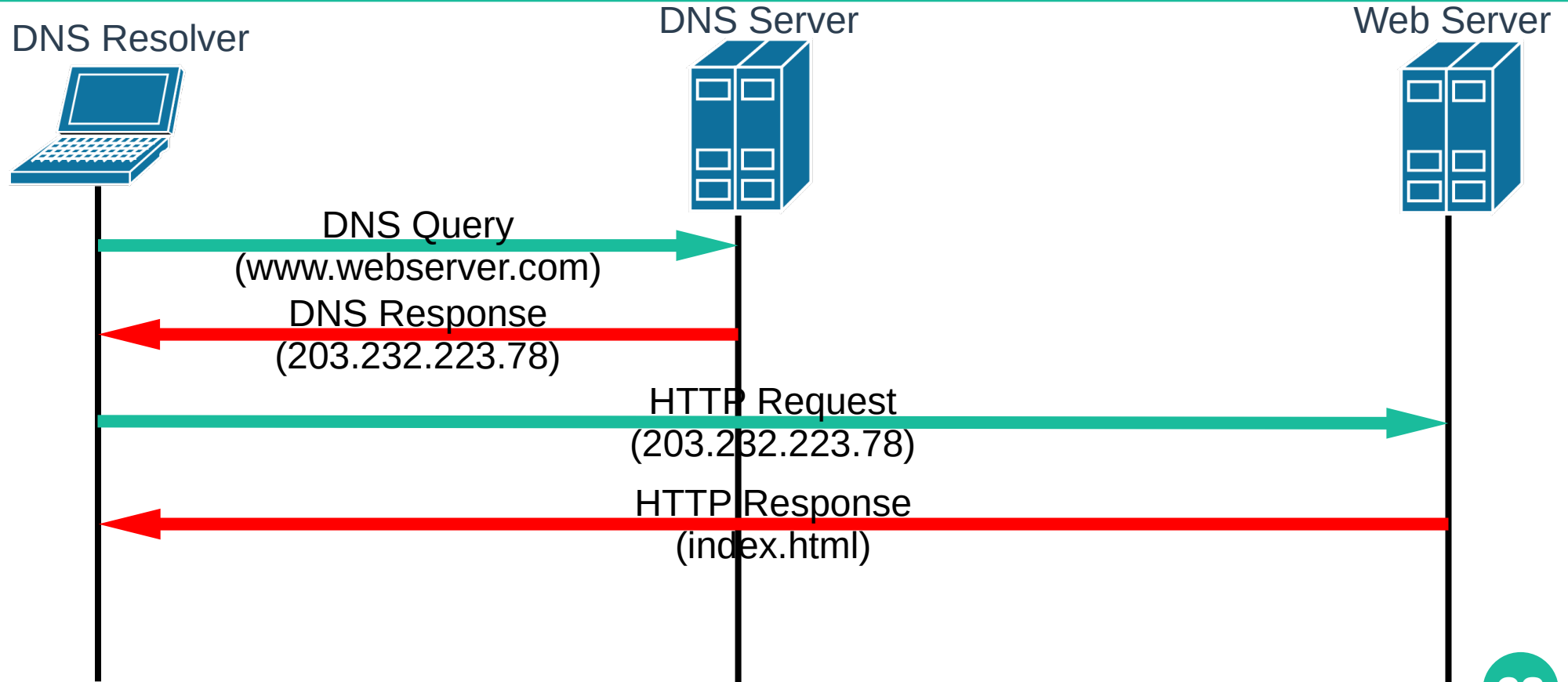
# DNS Message

응답 메시지  
0

0

16																32															
Identification																Q R	OPCode				A A	T C	R D	R A	0	0	0	rCode			
질문 레코드 수																응답 레코드 수															
권한 레코드 수																추가 레코드 수															
도메인 이름 이름(가변적)																															
도메인 유형																도메인 클래스															
수명(Time To Live)																															
자원 데이터 길이 (Resource data length)																															
자원 데이터(Resource Data)																															

# DNS



# DNS Spoofing

- **Spoofing**

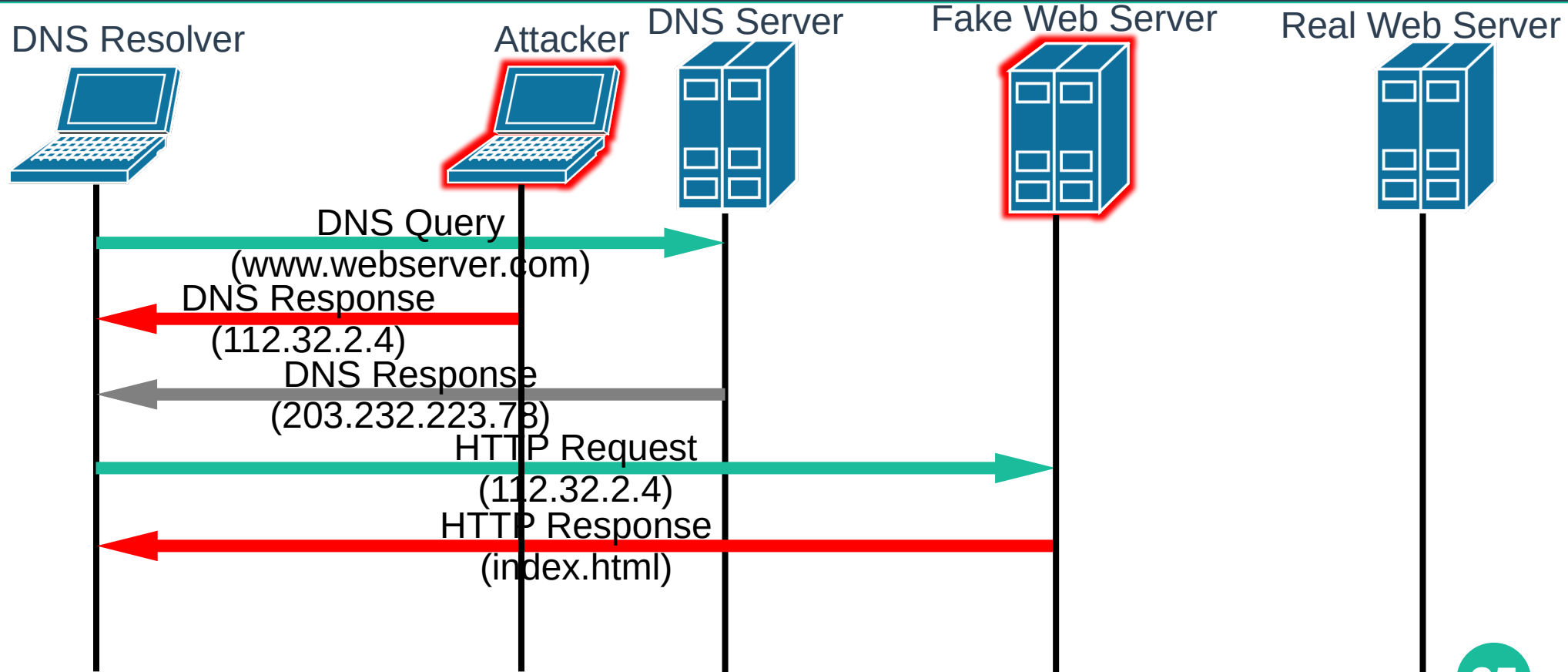
- Spoofing 으로 DNS 요청 메시지를 확인하면 정상 DNS 응답 보다 먼저 변조된 DNS 응답이 희생 호스트에 전송되도록 하는 방법

- **DNS Cache Poisoning**

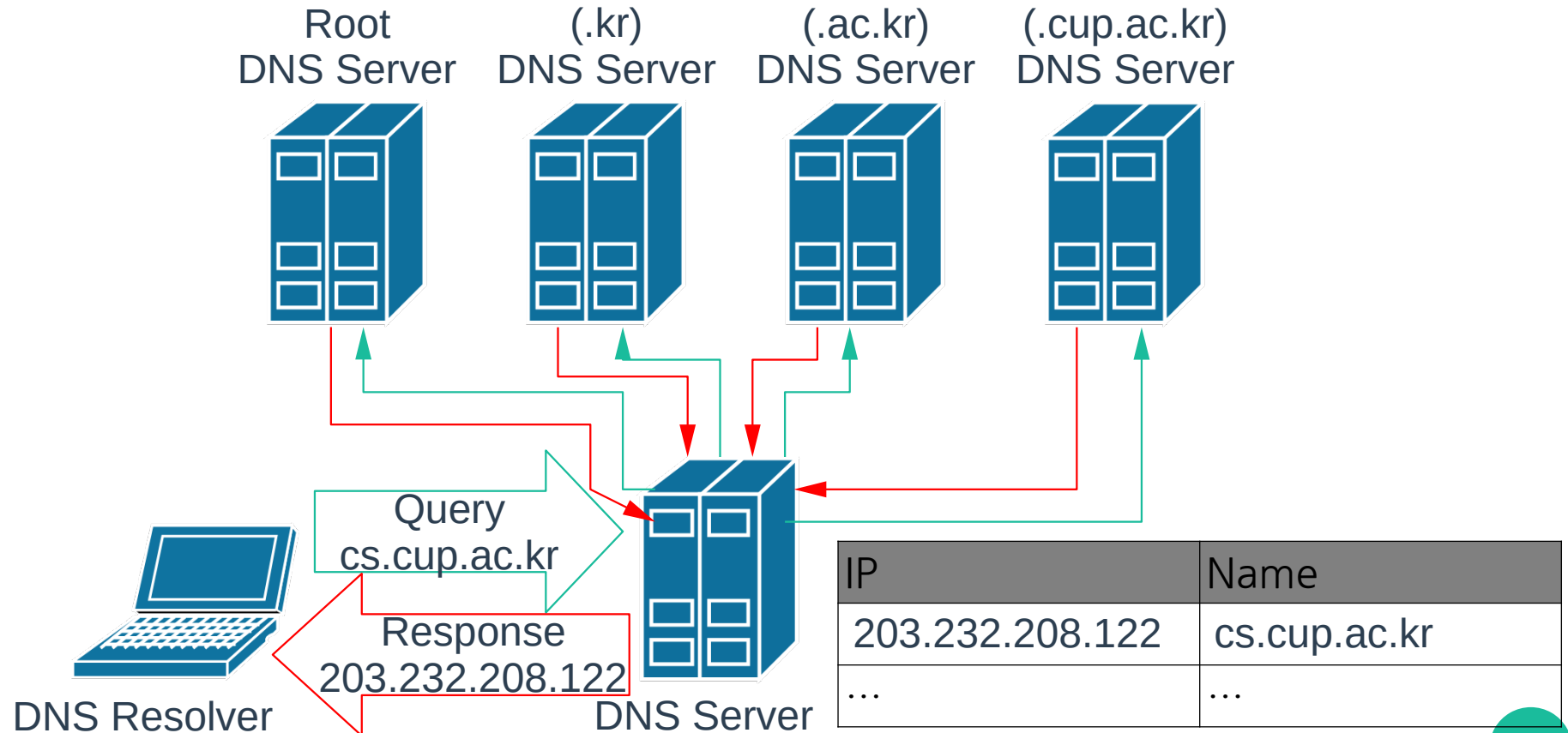
- DNS 서버의 순환 질의 동작 과정을 공격하여 DNS 캐시를 오염시키는 방법
  - Spoofing
  - 무작위 ID 생성 기반의 DNS 캐시 포이즈닝



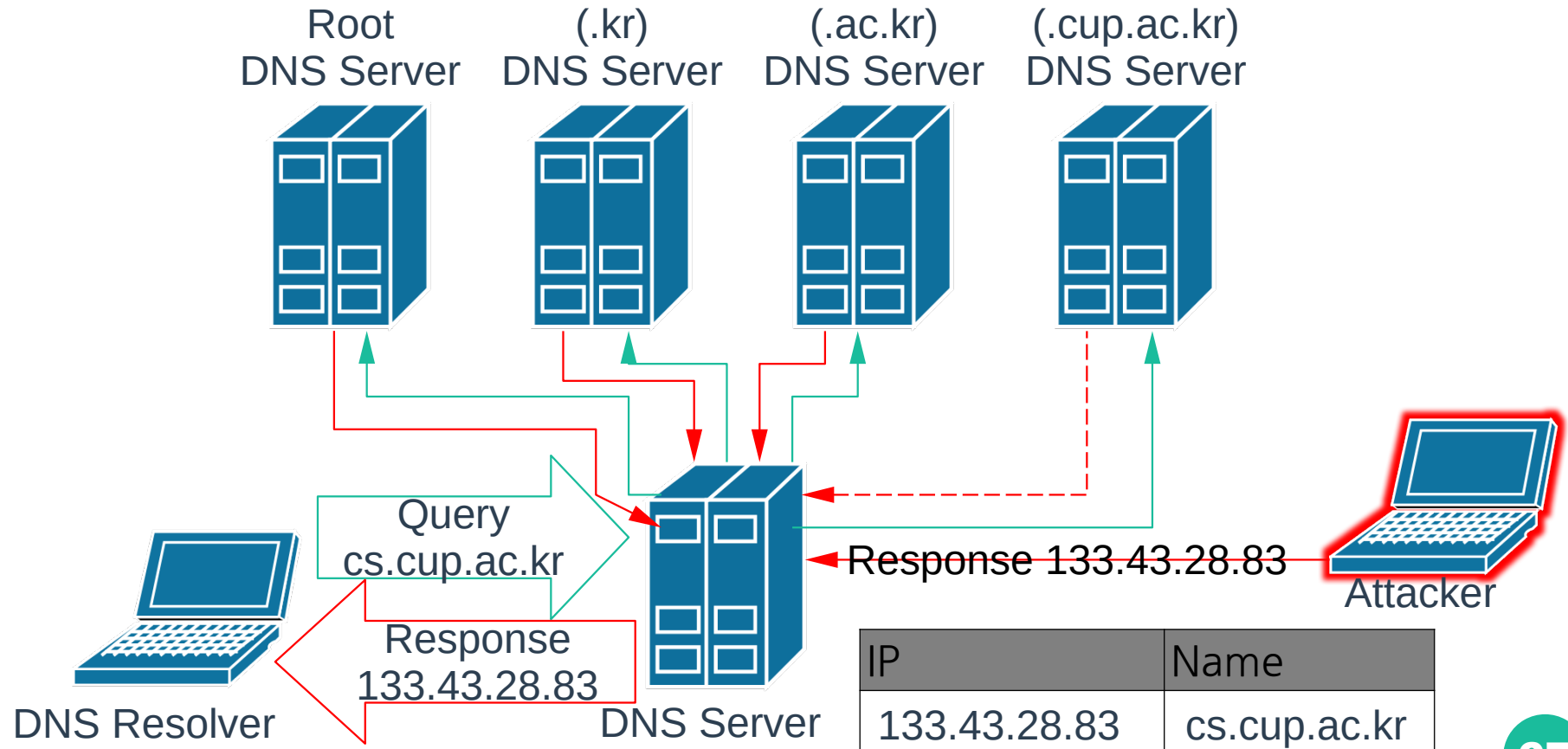
# DNS Spoofing



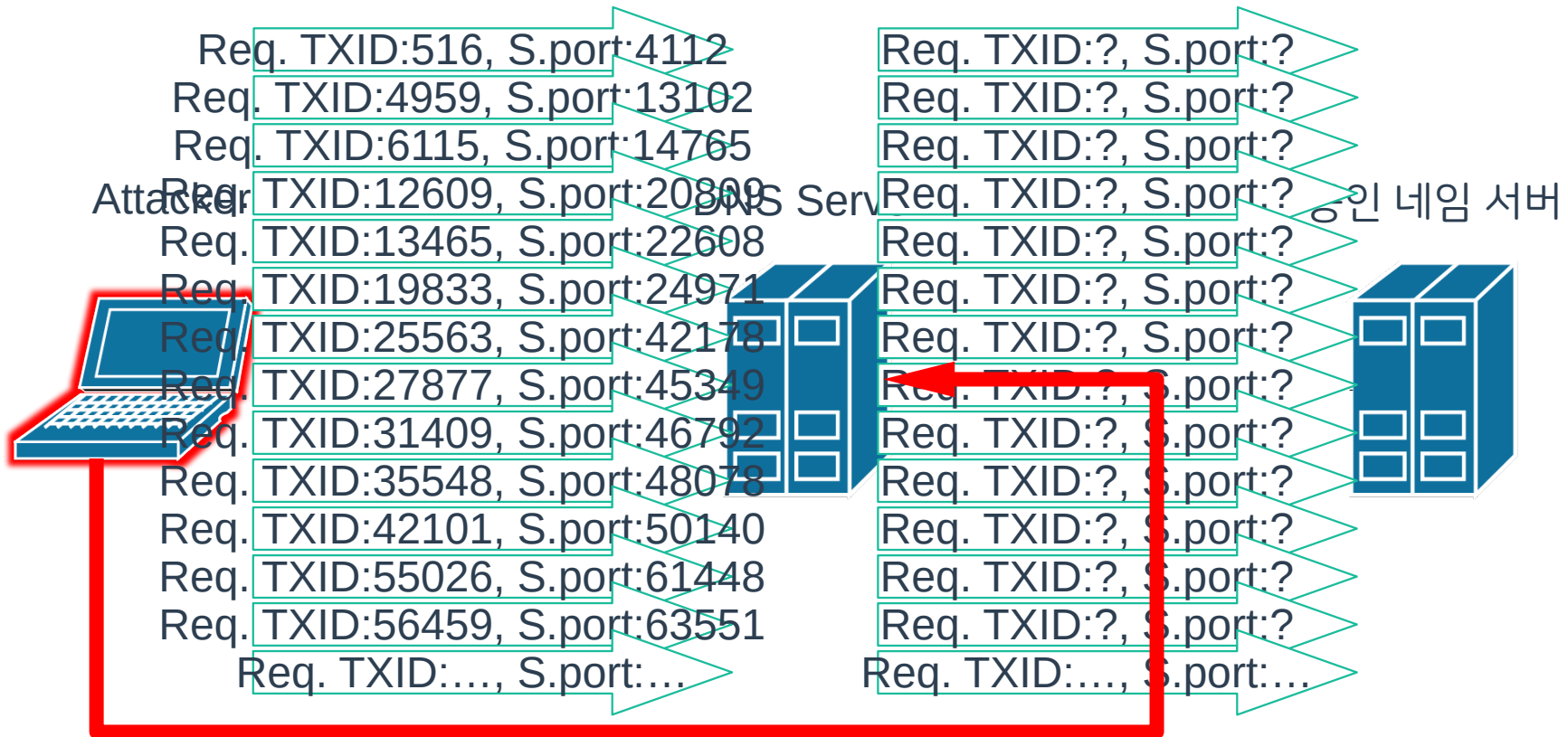
# DNS Recursive Query



# DNS Cache Poisoning



# DNS Cache Poisoning



# DNS Cache Poisoning 방어

- **DNS 서버 소프트웨어 보안 수준 강화**
  - 보안 취약점이 수정된 최신 버전으로 업데이트
- **순환 질의를 사용하지 않음**
  - 순환 질의를 사용하지 않거나, 신뢰 가능한 경우에만 제한적 허용
- **암호화**
  - 요청 / 응답을 암호화

# 비둘기 집 원리

- **$n+1$**  마리의 비둘기를  **$n$**  개의 비둘기 집에 모두 넣었다고 했을 때 비둘기 집 중 하나는 **2** 마리가 들어가야 한다는 원리



[https://ko.wikipedia.org/wiki/%EB%B9%84%EB%91%98%EA%B8%B0%EC%A7%91\\_%EC%9B%90%EB%A6%AC](https://ko.wikipedia.org/wiki/%EB%B9%84%EB%91%98%EA%B8%B0%EC%A7%91_%EC%9B%90%EB%A6%AC)

# 생일 문제

- 임의의 사람들 중에 생일이 같은 사람이 존재할 확률을 구하는 문제.
- 윤일을 포함한 1 년은 366 일 일때, 366 명을 초과하는 사람들이 모여야 100% 로 생일이 겹치게 된다.( 비둘기 집 원리 )
  - 실제로는 366 명 보다 많이 적어도 생일이 겹칠 확률이 아주 높아지게 된다.

- [https://ko.wikipedia.org/wiki/%EC%83%9D%EC%9D%BC\\_%EB%AC%B8%EC%A0%9C](https://ko.wikipedia.org/wiki/%EC%83%9D%EC%9D%BC_%EB%AC%B8%EC%A0%9C)

# 생일 공격

- 생일 문제의 정리 결과를 기반으로 암호를 찾아내거나 해시 충돌을 찾아내는 등의 수학적 확률에 기반을 둔 공격.



# DoS Attack

서비스 거부 공격

# Denial of Service Attack

- **DoS Attack**

- Denial of Service Attack, 서비스 거부 공격
- 서비스가 정상적으로 제공되지 못하도록 시스템의 가용성을 저하시키는 공격
  - 주로 대규모의 가짜 요청을 만들어서 공격 대상 시스템에 과부하를 유발함 .
- 공격 종류
  - 대역폭 소진 공격
    - 네트워크 자원을 소진 시키는 공격
  - 서버 마비 공격
    - 서버의 소프트웨어 또는 하드웨어 자원을 소진 시키는 공격

# Denial of Service Attack 유형

	대역폭 소진 공격	서버 마비 공격
공격의 형태	TCP SYN flooding ICMP/UDP flooding IP flooding: LAND, Teardrop	HTTP GET flooding
공격 대상	네트워크 인프라	웹 서버, 정보보호 장비 등
증상	네트워크 대역폭 고갈	공격 대상 시스템만 피해

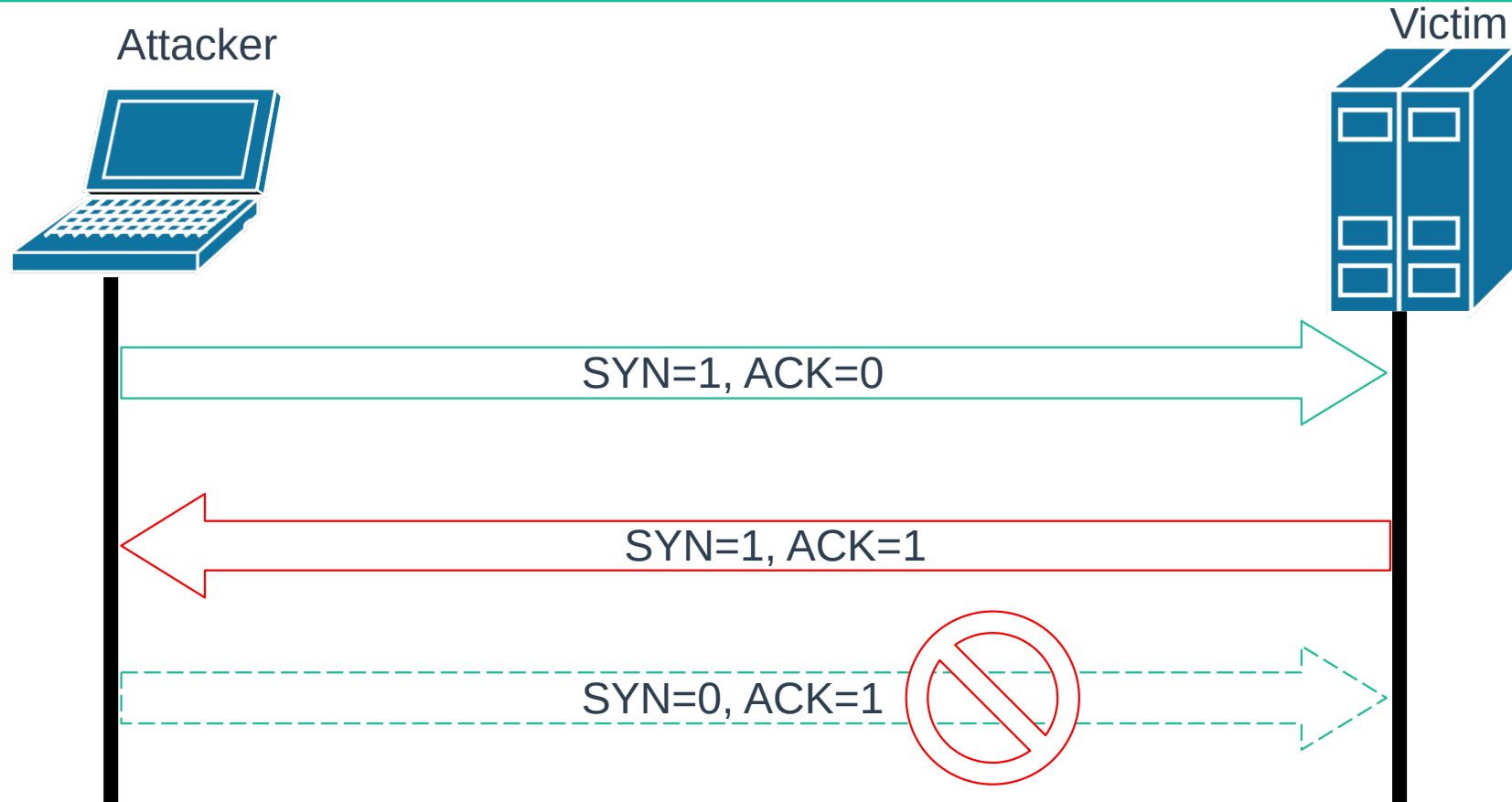
# TCP SYN flooding

반 (Half) 개방 공격

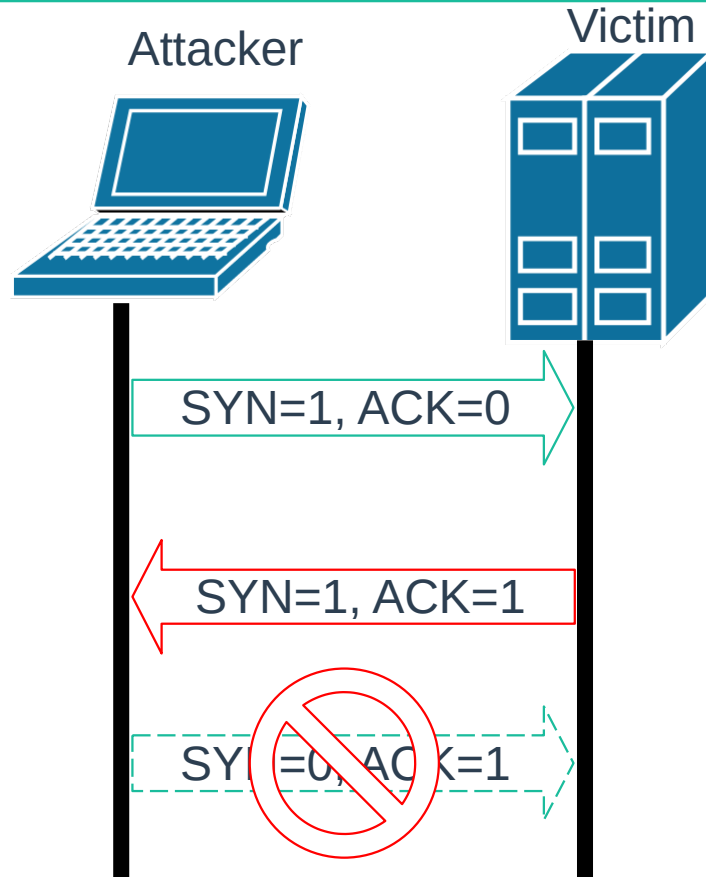
# TCP SYN flooding

- **TCP** 의 연결 과정을 공격하는 방법
  - TCP 는 3-Way Handshake, 연결 설정이 3 단계에 걸쳐 이루어짐
    - SYN, SYN+ACK, ACK
  - TCP 연결을 절반만 열기 때문에 반 (half) 개방 공격이라고도 불림

# TCP SYN flooding



# TCP SYN flooding



희생자 호스트의 대기 큐 :

Request	Status
0	Waiting ACK
1	Waiting ACK
2	Waiting ACK
3	Waiting ACK
4	Waiting ACK
...	...
n	Waiting ACK

# TCP SYN flooding

- **TCP SYN flooding 방어**

- 대기 큐의 크기 늘리기
  - 대기 큐의 크기를 늘림으로써 공격자가 발생하는 요청보다 많은 요청을 수용할 수 있게 함.
- 최대 접속 대기 시간 줄이기
  - 느린 네트워크 환경의 정상 호스트와 연결에 문제가 있을 것으로 예상
- 보안 솔루션 사용
  - 비정상 연결 요청을 탐지하고 차단함



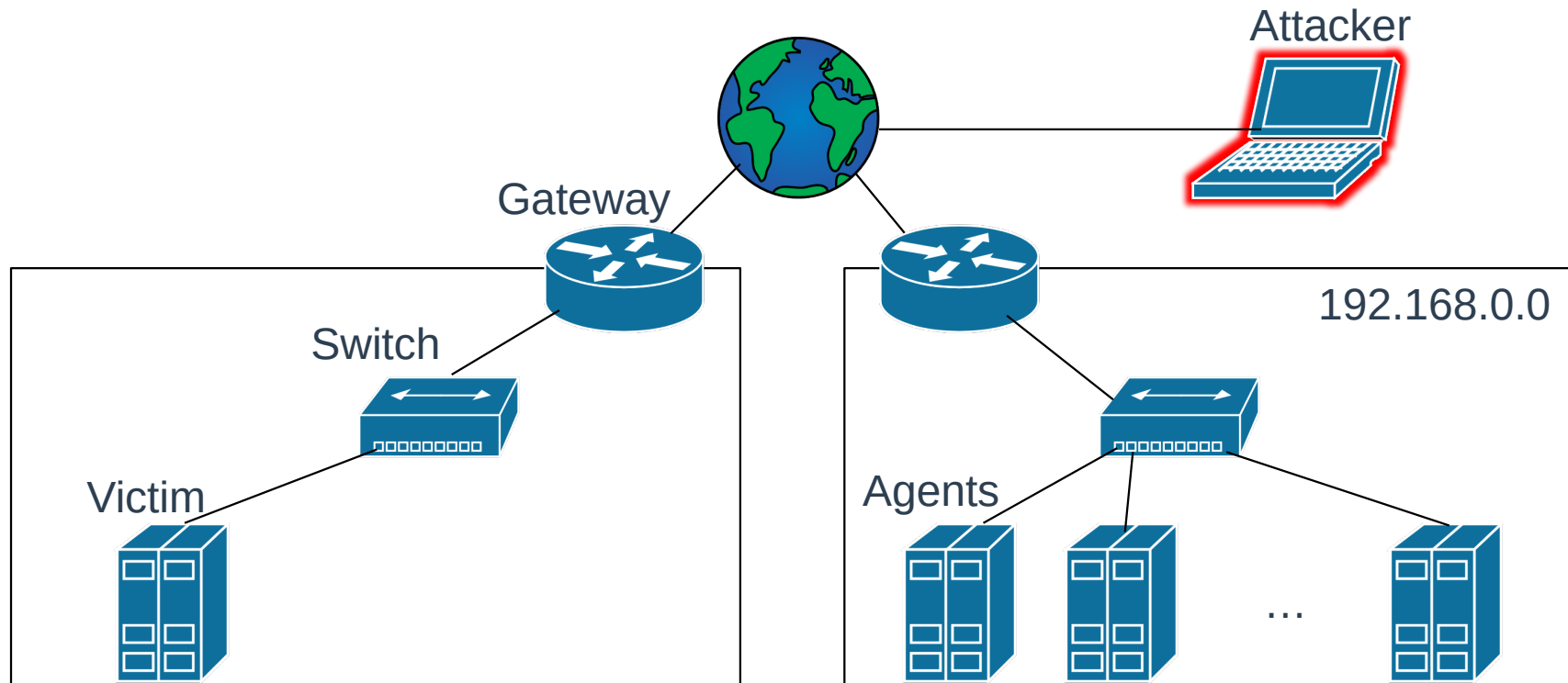
# ICMP flooding

스머프 (Smurf) 공격

# ICMP flooding

- **Internet Control Message Protocol Flooding**
  - 발신지 IP 주소가 희생자의 IP 주소로 변조된 ICMP Echo Request 를 특정 네트워크의 브로드캐스트 주소로 보내서 다수의 ICMP Echo Reply 가 희생자에게 전송 되도록 하는 공격

# ICMP flooding



# UDP flooding

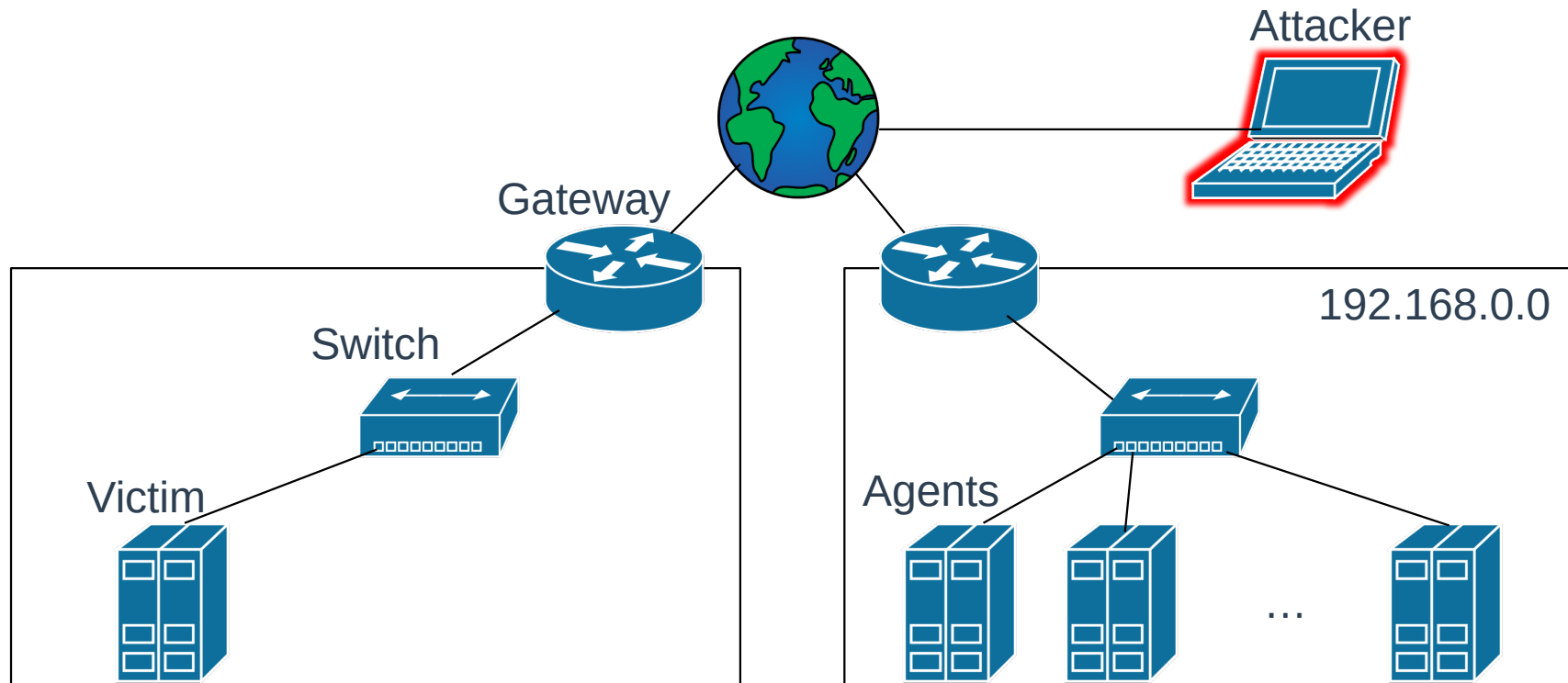
프레글 (Fraggle) 공격

# UDP flooding

- **User Datagram Protocol Flooding**

- ICMP Flooding 과 유사한 형태의 공격 .
- UDP Flooding 의 경우는 Agent 의 7 번 포트에 UDP PDU 전송 .
- 7 번 포트는 Echo 로 수신된 데이터를 송신측에 전송

# UDP flooding



# Broadcasting 기반의 DoS 공격 방어

- 패킷 필터링

- 외부에서 내부로 전달되는 브로드캐스트 (directed broadcast) 패킷을 차단

# Distribute Denial of Service

- **DDoS**

- Distribute Denial of Service: 분산 서비스 거부 공격
  - 공격자가 여러 곳에서 동시에 서비스 거부 공격을 하는 방법
  - ICMP Flooding, UDP Flooding에서는 게이트웨이나 라우터가 Master 역할을 함 .
  - 최근의 DDoS 공격은 악성코드를 활용해서 불특정 다수의 호스트를 Agent 로 하여 공격을 수행함
  - 이 악성코드들은 대부분 자체적으로 증식하고 전파하는 기능을 포함하고 있어 네트워크에 하나의 호스트만 감염되어도 쉽게 전파됨
  - 공격자 → [Master] → Agent → 희생자 로 공격이 이루어지기 때문에 Agent 를 찾는 것은 비교적 쉽지만 실질적 공격자를 찾는 것은 어려움
- Master: 공격자로부터 직접 공격 명령을 전달 받아 각 Agent 에게 공격 명령을 전달 .
- Agent: 공격 대상자에게 실제 공격을 하는 주체

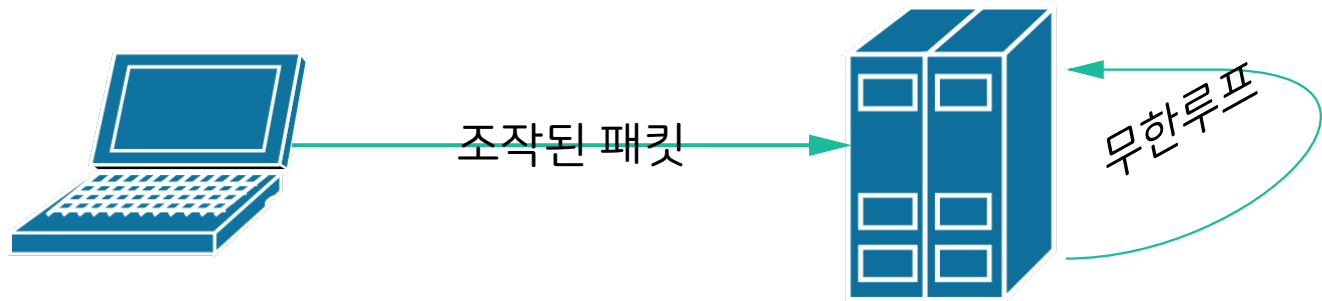


# IP Flooding

LAND, Teardrop 공격

# LAND:Local Area Network Denial

- 송신 **IP** 주소와 수신 **IP** 주소를 희생자 호스트의 **IP** 주소로 조작한 패킷을 전송하는 공격 방법
  - 희생 호스트는 패킷의 송신 IP 주소 필드의 주소로 응답하지만 이 주소는 스스로의 주소이므로 스스로 메시지를 무한정 주고받는 루프상태에 빠짐
  - 현재 대부분의 네트워크 장비와 운영체제에서는 소스 IP 주소와 목적지 IP 주소가 동일한 패킷에 대해서 예외 처리하기에 공격이 유효하지 않음 .



# Teardrop

- **IP Header 의 Fragment Offset field 의 값을 조작하여 수신 호스트에서 재조립 시 중복 / 생략이 발생하도록 하는 공격**
  - 재조립 과정에서 Fragment Offset 값으로 위치를 계산하고 단편을 복사하는데 Offset 값을 조작하여 계산 결과가 음수가 되게 하면 memcpy() 함수의 데이터 타입 차이로 인해서 잘못된 메모리 접근이 발생함 .
  - [https://news.sbs.co.kr/news/endPage.do?news\\_id=N0311200908](https://news.sbs.co.kr/news/endPage.do?news_id=N0311200908)

# HTTP GET Flooding

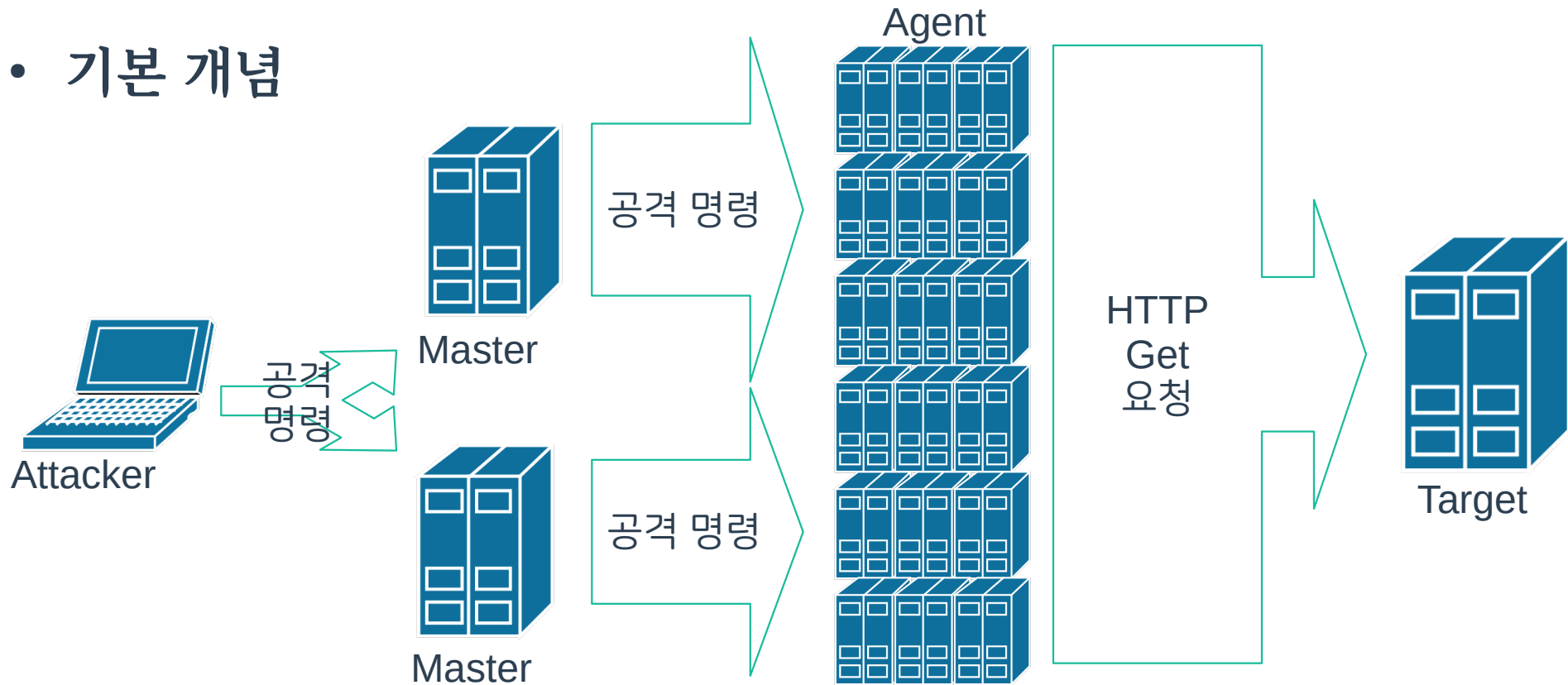
웹 서버 소프트웨어를 대상으로 한 공격

# HTTP GET Flooding

- 다수의 **Agent** 를 이용하여 웹 서버에 **Get** 요청을 보내 과부하를 유발하는 공격 방법
  - 기본적인 HTTP Get Flooding 도 있지만 변종 공격이 있음
    - HTTP CC(Cache-Control) 공격
    - 동적 HTTP 요청 공격

# HTTP GET Flooding

- 기본 개념



# HTTP CC(Cache-Control) 공격

- **HTTP** 헤더의 **Cache-Control** 지시문으로 공격 목표의 처리 부하량을 증가.
  - no-cache
    - 서버 유효성 재검사 없이 응답을 재사용하는 것을 방지
  - no-store
    - 아무것도 저장하지 않음 ( 캐시 사용하지 않음 )
  - must-revalidate
    - 만료된 캐시만 서버에 유효성 재검사 후 사용

# 동적 HTTP Get 공격

- **Get** 요청하는 **URL** 을 무작위 또는 인간을 모사
  - 정상 요청과 구별하기 어렵게 하여 탐지하고 방어하기 어렵게 함 .



# HTTP GET Flooding 방어

- 요청 임계치 기반

- 특정 IP 주소로 부터 오는 요청이 임계치를 넘을 경우 해당 IP 주소의 요청을 위해로 판단하고 일시적 또는 영구적 차단.
- Cache-Control 헤더 옵션 별로 다르게 임계치를 설정할 수 있으면 더 효과적 .



## **Question & Answer**

**Connection Closed**

