



네트워크야 놀자!

Level 4. 방화벽

김기동 프로
광운대-시스코 이노베이션 센터

2024.05.08.

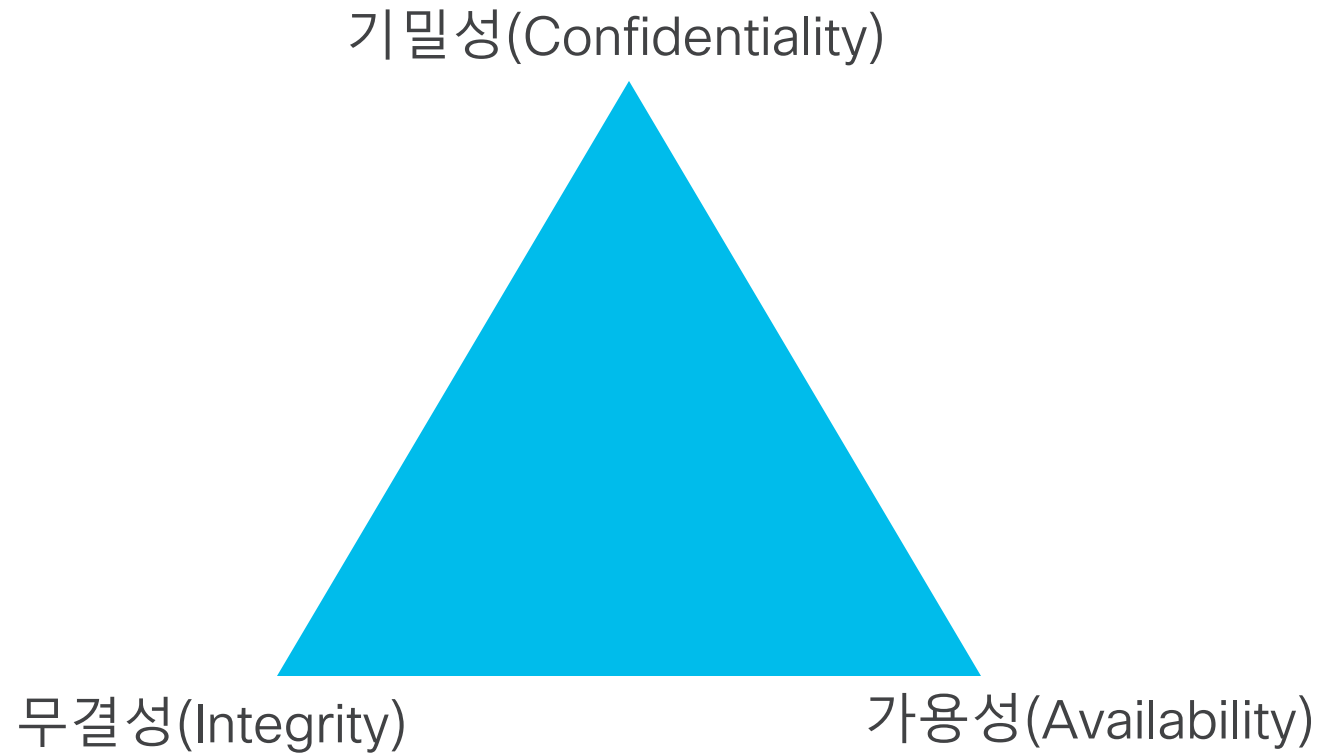
목차

1. 정보보안이란?
2. 공격 기법
3. 방화벽이란?
4. ACL(Access Control List)
5. 방화벽 ACL

1. 정보보안이란?

정보보안의 3요소

- 보호하고자 하는 요소



기밀성 (Confidentiality)

- 허가 받지 않은 사람들에게 정보가 노출되지 않도록 보호하는 것
- 정보에 접근할 수 있는 사람을 제한하여, 데이터를 안전하게 유지
- 사용 기술: 패스워드, 암호화

```
Router#show run
*May 3 07:29:55.290: %SYS-5-CONFIG_I: Configured from console by console
Building configuration...

Current configuration : 4251 bytes
!
! Last configuration change at 07:29:55 UTC Fri May 3 2024
!
version 15.9
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname Router
!
boot-start-marker
boot-end-marker
!
enable secret 9 $9$l5oPUeqcy9YAqv$l.VVGieeIqxnxA93zHcRwM5Vtg.sg75dQmir38mC2Bo
!
```

네트워크 장비에서 enable 패스워드를 설정하고 암호화

기밀성 (Confidentiality)

- Telnet 사용 시

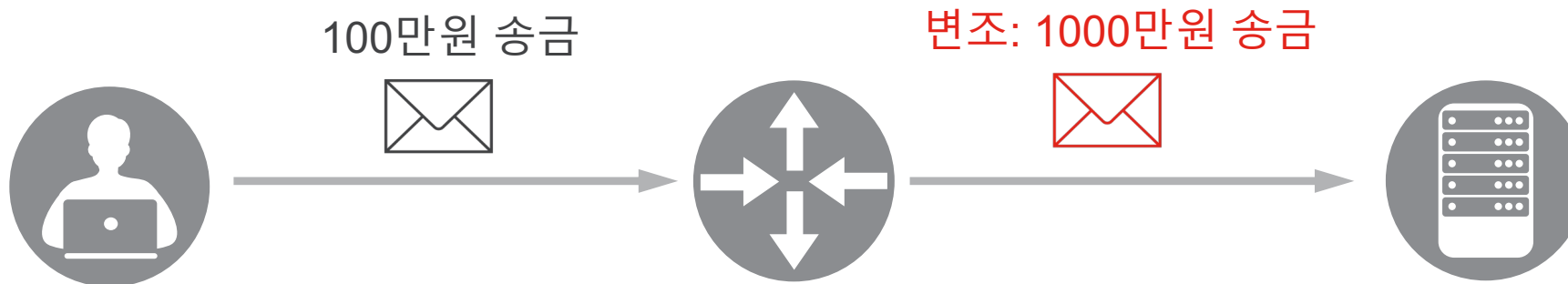
```
▶Frame 41: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)
▶Ethernet II, Src: RealtekU_09:e2:65 (52:54:00:09:e2:65), Dst: RealtekU_11:53:13 (52:54:00:11:53:13)
▶Internet Protocol Version 4, Src: 10.10.10.11, Dst: 10.10.10.1
▶Transmission Control Protocol, Src Port: 46230, Dst Port: 23, Seq: 43, Ack: 686, Len: 2
▼Telnet
  Data: co 평문으로 전송
```

- SSH 사용 시

```
▶Frame 47: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits)
▶Ethernet II, Src: RealtekU_09:e2:65 (52:54:00:09:e2:65), Dst: RealtekU_11:53:13 (52:54:00:11:53:13)
▶Internet Protocol Version 4, Src: 10.10.10.11, Dst: 20.20.20.21
▶Transmission Control Protocol, Src Port: 33670, Dst Port: 22, Seq: 1794, Ack: 1758, Len: 84
▼SSH Protocol
  ▼SSH Version 2 (encryption:chacha20-poly1305@openssh.com mac:<implicit> compression:none)
    Packet Length (encrypted): f6caa72f
    Encrypted Packet: a1875412662de3101b208fc29c5428815302989ea94b2f92\xe2\x80\xa6
    MAC: d2e8916940426790cf1999d8bfbcb0a51
    Direction: client-to-server 암호화되어 전송
```

무결성 (Integrity)

- 데이터의 정확성과 일관성을 유지하고, 권한 없는 변경으로부터 보호하는 것
- 원래의 목적대로 정확하게 유지되어야 한다는 원칙
- 사용 기술: 해시 알고리즘, 디지털 서명

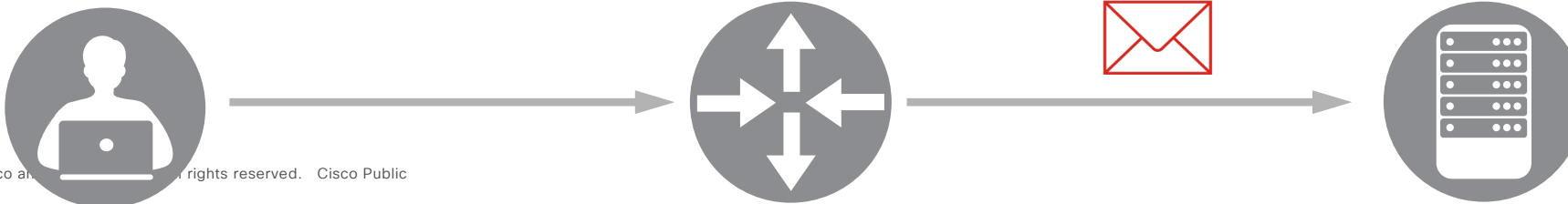


무결성 (Integrity)

- MAC: Message Authentication Code
- 전달하려는 데이터로 MAC을 계산하고 이를 추가로 전달
- 수신자 측에서 수신한 데이터로 계산한 MAC과 수신한 MAC과 비교하여 검증

```
▶Frame 47: 150 bytes on wire (1200 bits), 150 bytes captured (1200 bits)
▶Ethernet II, Src: RealtekU_09:e2:65 (52:54:00:09:e2:65), Dst: RealtekU_11:53:13 (52:54:00:11:53:13)
▶Internet Protocol Version 4, Src: 10.10.10.11, Dst: 20.20.20.21
▶Transmission Control Protocol, Src Port: 33670, Dst Port: 22, Seq: 1794, Ack: 1758, Len: 84
▼SSH Protocol
  ▼SSH Version 2 (encryption:chacha20-poly1305@openssh.com mac:<implicit> compression:none)
    Packet Length (encrypted): f6caa72f
    Encrypted Packet: a1875412662de3101b208fc29c5428815302989ea94b2f92\xe2\x80\xa6
    MAC: d2e8916940426790cf1999d8bfbc0a51
    Direction: client-to-server
```

송신 MAC: d2e8.....a51 \neq 계산한 MAC: 12d3.....14dd



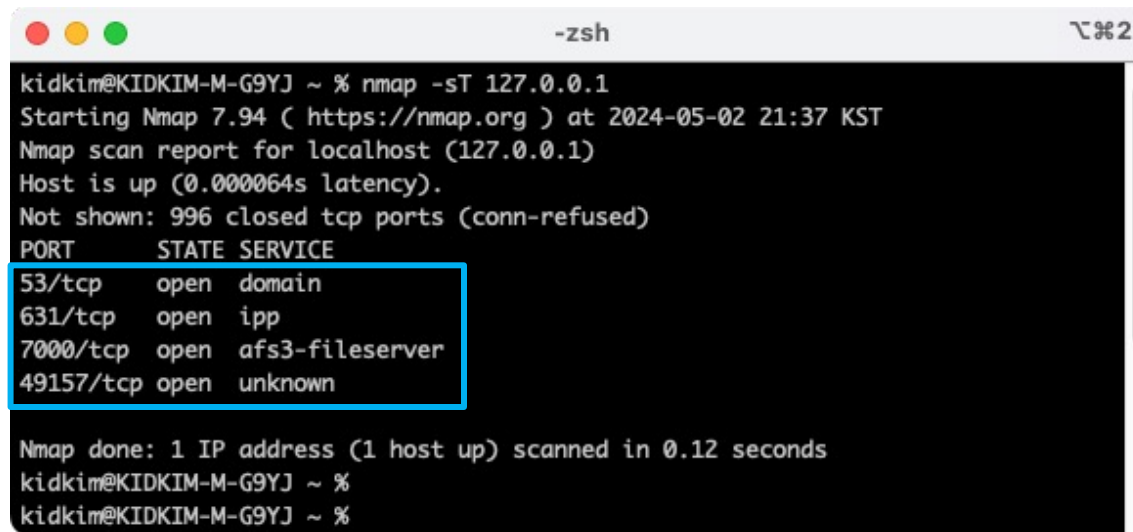
가용성 (Availability)

- 정보가 필요할 때 사용자가 접근할 수 있도록 하는 것
 - 사용 기술: RAID, 게이트웨이 이중화
-
- 기밀성을 높이기 위해 암호화 키 길이를 늘이고, 알고리즘을 복잡하게 구성
-> 복호화 하는데 시간이 오래 걸려서 가용성이 훼손

2. 공격 기법

정찰

- 실제 액세스나 DoS와 같은 공격을 시도하기 전에 공격 대상에 대해 파악하려는 시도
- IP 주소, TCP UDP 포트, 운영 체제, 프로그램 버전 등의 정보를 탐색

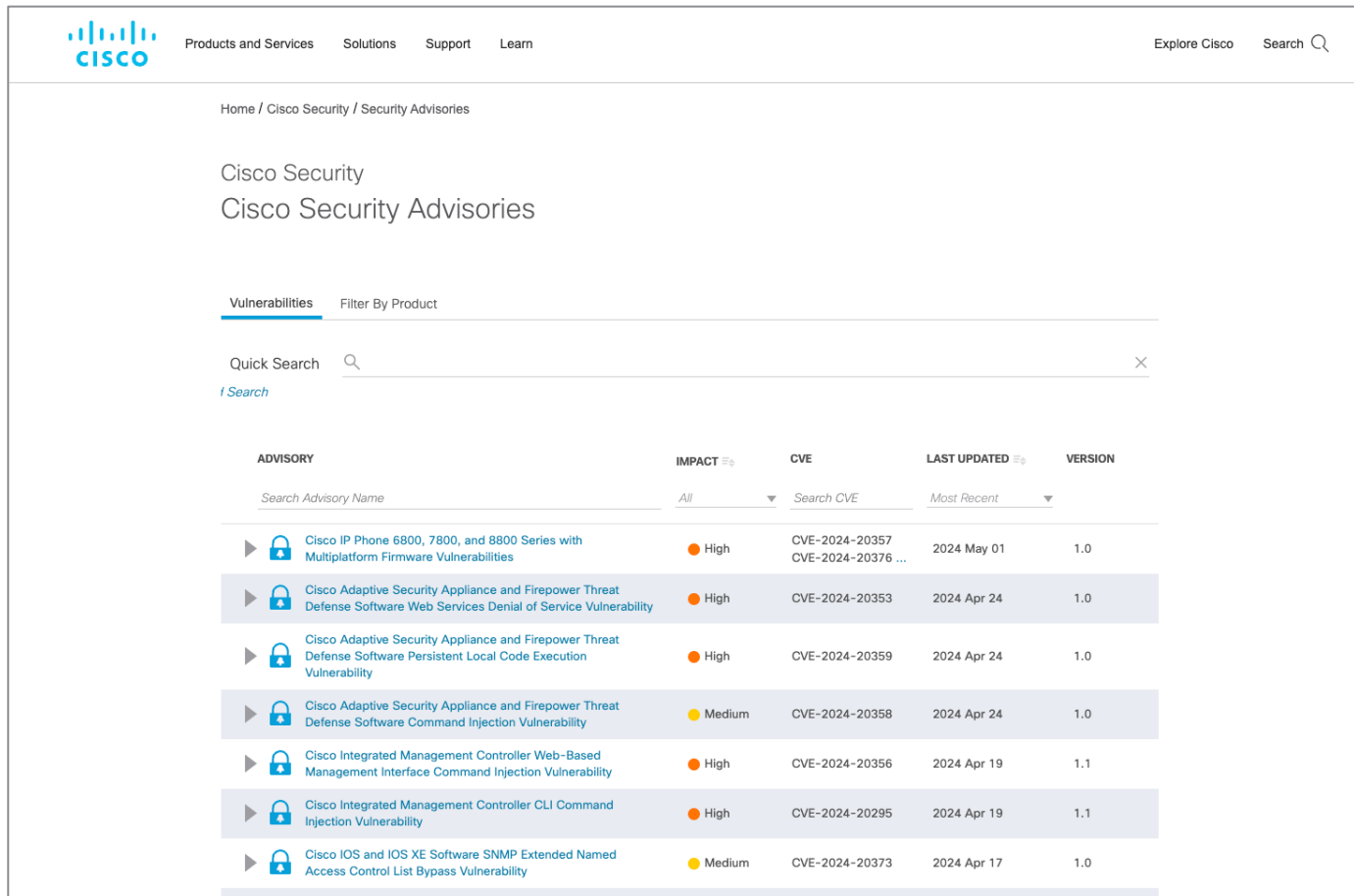


```
kidkim@KIDKIM-M-G9YJ ~ % nmap -sT 127.0.0.1
Starting Nmap 7.94 ( https://nmap.org ) at 2024-05-02 21:37 KST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000064s latency).
Not shown: 996 closed tcp ports (conn-refused)
PORT      STATE SERVICE
53/tcp    open  domain
631/tcp    open  ipp
7000/tcp   open  afs3-fileserver
49157/tcp  open  unknown








Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
kidkim@KIDKIM-M-G9YJ ~ %
kidkim@KIDKIM-M-G9YJ ~ %
```

취약점 예시


- <https://sec.cloudapps.cisco.com/security/center/publicationListing.x>




The screenshot shows the Cisco Security Advisories page. The header includes the Cisco logo and navigation links: Products and Services, Solutions, Support, Learn, Explore Cisco, and Search. The breadcrumb trail is Home / Cisco Security / Security Advisories. The main heading is Cisco Security Advisories. Below this, there is a 'Vulnerabilities' tab and a 'Filter By Product' link. A 'Quick Search' bar is present with a magnifying glass icon and a close button. Below the search bar is a table of advisories. The table has columns for ADVISORY, IMPACT, CVE, LAST UPDATED, and VERSION. The ADVISORY column includes a search bar and a list of advisories. The IMPACT column shows severity levels (High, Medium). The CVE column shows CVE IDs. The LAST UPDATED column shows dates. The VERSION column shows version numbers.

| ADVISORY | IMPACT | CVE | LAST UPDATED | VERSION |
|---|--------|--------------------------------------|--------------|---------|
| Search Advisory Name | All | Search CVE | Most Recent | |
|  Cisco IP Phone 6800, 7800, and 8800 Series with Multiplatform Firmware Vulnerabilities | High | CVE-2024-20357 CVE-2024-20376 ... | 2024 May 01 | 1.0 |
|  Cisco Adaptive Security Appliance and Firepower Threat Defense Software Web Services Denial of Service Vulnerability | High | CVE-2024-20353 | 2024 Apr 24 | 1.0 |
|  Cisco Adaptive Security Appliance and Firepower Threat Defense Software Persistent Local Code Execution Vulnerability | High | CVE-2024-20359 | 2024 Apr 24 | 1.0 |
|  Cisco Adaptive Security Appliance and Firepower Threat Defense Software Command Injection Vulnerability | Medium | CVE-2024-20358 | 2024 Apr 24 | 1.0 |
|  Cisco Integrated Management Controller Web-Based Management Interface Command Injection Vulnerability | High | CVE-2024-20356 | 2024 Apr 19 | 1.1 |
|  Cisco Integrated Management Controller CLI Command Injection Vulnerability | High | CVE-2024-20295 | 2024 Apr 19 | 1.1 |
|  Cisco IOS and IOS XE Software SNMP Extended Named Access Control List Bypass Vulnerability | Medium | CVE-2024-20373 | 2024 Apr 17 | 1.0 |

취약점 예시

Products and ServicesSolutionsSupportLearnExplore Cisco

Home / Cisco Security / Security Advisories

 Cisco Security Advisory

Cisco Adaptive Security Appliance Software and Firepower Threat Defense Software SNMP Access Control Vulnerability

Medium

Advisory ID:

First Published:

Version 1.0:

Workarounds:

Cisco Bug IDs:

CVSS Score:

cisco-sa-asaftd-snmpaccess-M6yOweq3


2021 October 27 16:00 GMT


Final

No workarounds available

[CSCw49739](#)
[CSCw31710](#)
[CSCw51436](#)

Base 5.3

 Download CSAF

 Email

Cisco Security Vulnerability Policy

To learn about Cisco security vulnerability disclosure policies and publications, see the [Security Vulnerability Policy](#). This document also contains instructions for obtaining fixed software and receiving security vulnerability information from Cisco.

Subscribe to Cisco Security Notifications

Subscribe

Related to This Advisory

[Cisco Event Response: October 2021 Cisco ASA, FMC, and FTD Software Security Advisory Bundled Publication](#)

Your Rating:

Summary

A vulnerability in the Simple Network Management Protocol version 3 (SNMPv3) access control functionality of Cisco Adaptive Security Appliance (ASA) Software and Cisco Firepower Threat Defense (FTD) Software could allow an unauthenticated, remote attacker to query SNMP data.

This vulnerability is due to ineffective access control. An attacker could exploit this vulnerability by sending an SNMPv3 query to an affected device from a host that is not permitted by the SNMPv3 access control list. A successful exploit could allow the attacker to send an SNMP query to an affected device and retrieve information from the device. The attacker would need valid credentials to perform the SNMP query.

Cisco has released software updates that address this vulnerability. There are no workarounds that address this vulnerability.

This advisory is available at the following link:
<https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-asaftd-snmpaccess-M6yOweq3>

ASA Software

| Cisco ASA Software Release | First Fixed Release for This Vulnerability |
|-------------------------------|--|
| Earlier than 9.8 ¹ | Not vulnerable. |
| 9.8 | Not vulnerable. |
| 9.9 ¹ | Not vulnerable. |
| 9.10 ¹ | Not vulnerable. |
| 9.12 | Not vulnerable. |
| 9.13 ¹ | Not vulnerable. |
| 9.14 | 9.14.2.4 |
| 9.15 | 9.15.1.7 |
| 9.16 | Not vulnerable. |

1. Cisco ASA Software releases 9.7 and earlier as well as releases 9.9, 9.10, and 9.13 have reached **end of software maintenance**. Customers are advised to migrate to a supported release that includes the fix for this vulnerability.

Man In The Middle

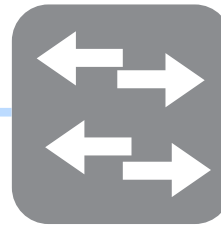
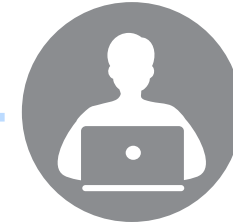
- MITM 공격은 두 목적지 간의 통신을 가로채는 공격
- 공격자는 두 목적지 중간에서 패킷을 가져간 후, 다시 네트워크에 전송
- MITM예시
 - ARP spoofing
 - DNS, ICMP redirection
 - DHCP 기반 공격: DHCP 쿼리 및 응답을 가로챈

Man In The Middle - ARP Spoofing

IP: 1.1.1.10
MAC: aaaa:aaaa:aaaa



IP: 1.1.1.20
MAC: bbbb:bbbb:bbbb



ARP
IP: 1.1.1.20
MAC: cccc:cccc:cccc



ARP
IP: 1.1.1.10
MAC: cccc:cccc:cccc



MAC: cccc:cccc:cccc

Man In The Middle - ARP Spoofing

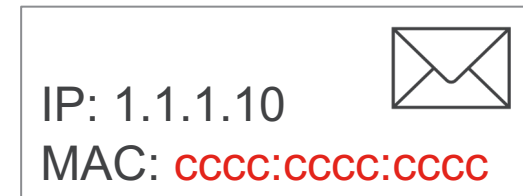
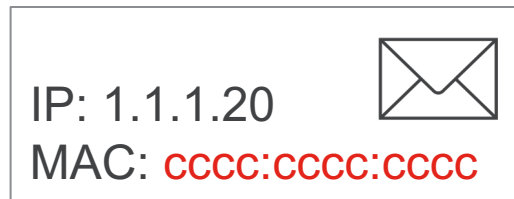
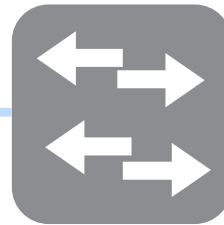
IP: 1.1.1.10

MAC: aaaa:aaaa:aaaa



IP: 1.1.1.20

MAC: bbbb:bbbb:bbbb



MAC: **cccc:cccc:cccc**

DoS (Denial of Service) 공격

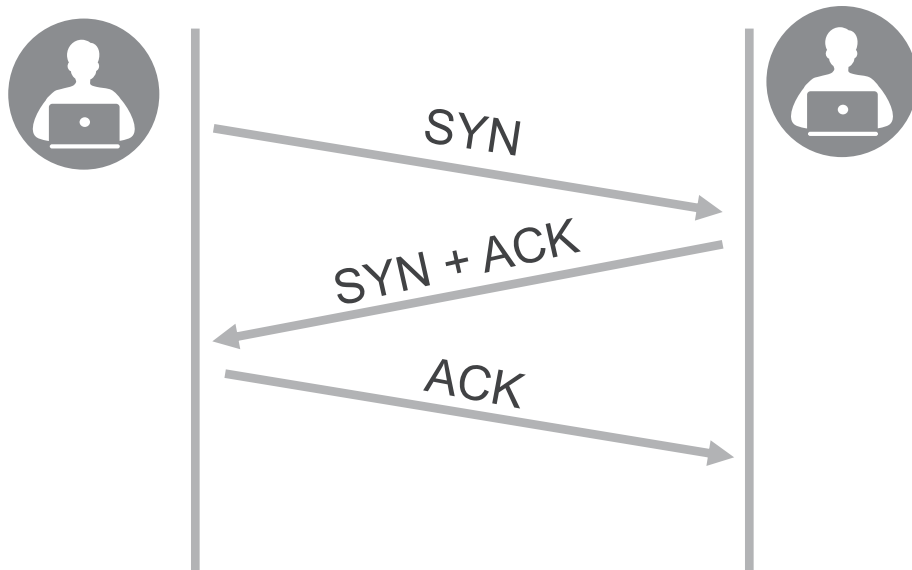
- 서비스를 제공하는 대상이 정상적인 서비스 제공할 수 없게 방해하는 공격
- 대상 시스템의 자원을 과부하 상태로 만들거나, 시스템이 서비스 요청을 처리하지 못하도록 함
- DoS 공격은 한 개의 공격자가 단일 소스에서 공격을 수행하는 경우
- 공격 예시
 - 네트워크: 많은 양의 데이터를 보내 네트워크 대역폭을 소모
 - 시스템: 서버가 처리할 수 없는 양의 연결 요청을 보내 정상적인 요청을 처리할 수 없게 함
 - 응용 프로그램: 특정프로그램의 취약점을 이용하여 CPU, 메모리 등의 리소스를 과도하게 사용

DoS 공격 - Ping of Death

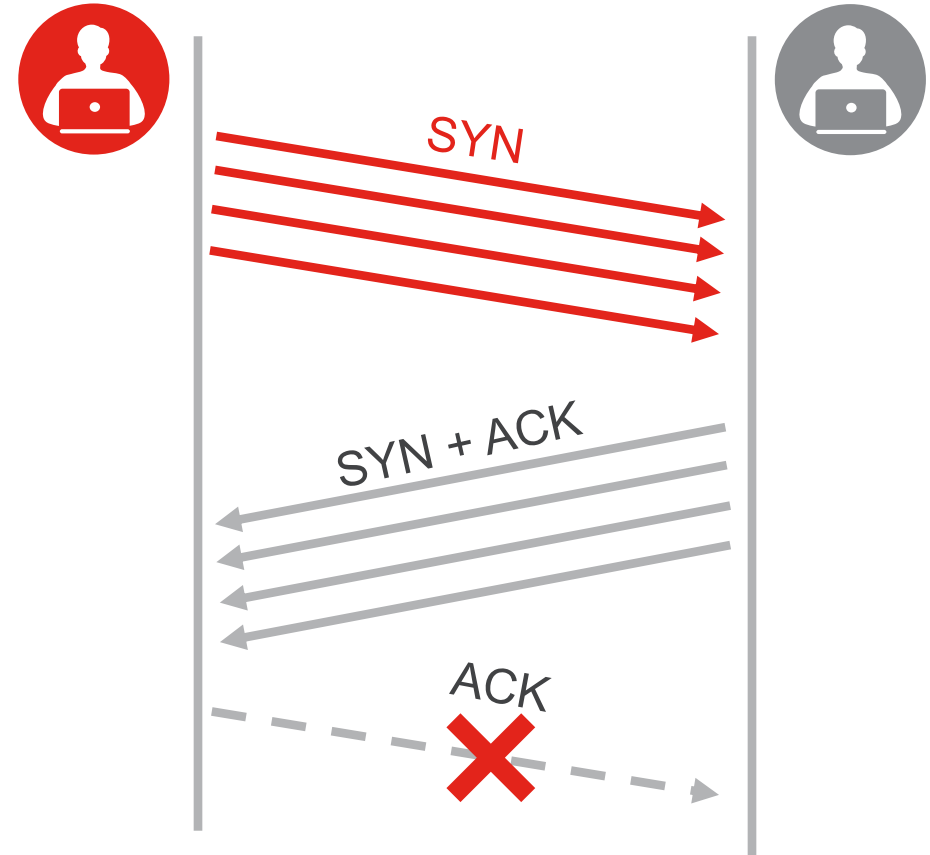
- 대부분 네트워크 장비의 기본 MTU(Maximum Transmission Unit) 1500바이트 이하로 설정되어 있는 점을 이용
- IPv4에서는 하나의 패킷이 가질 수 있는 최대 크기는 65,535 바이트
- 65,535 바이트 크기로 ping을 보내면 여러 패킷으로 분할되며 수신 측에서 과대한 패킷을 처리하려고 할 때 버퍼 오버플로우 등 문제가 발생할 수 있음

DoS 공격 - SYN Flooding

- TCP 3way handshake



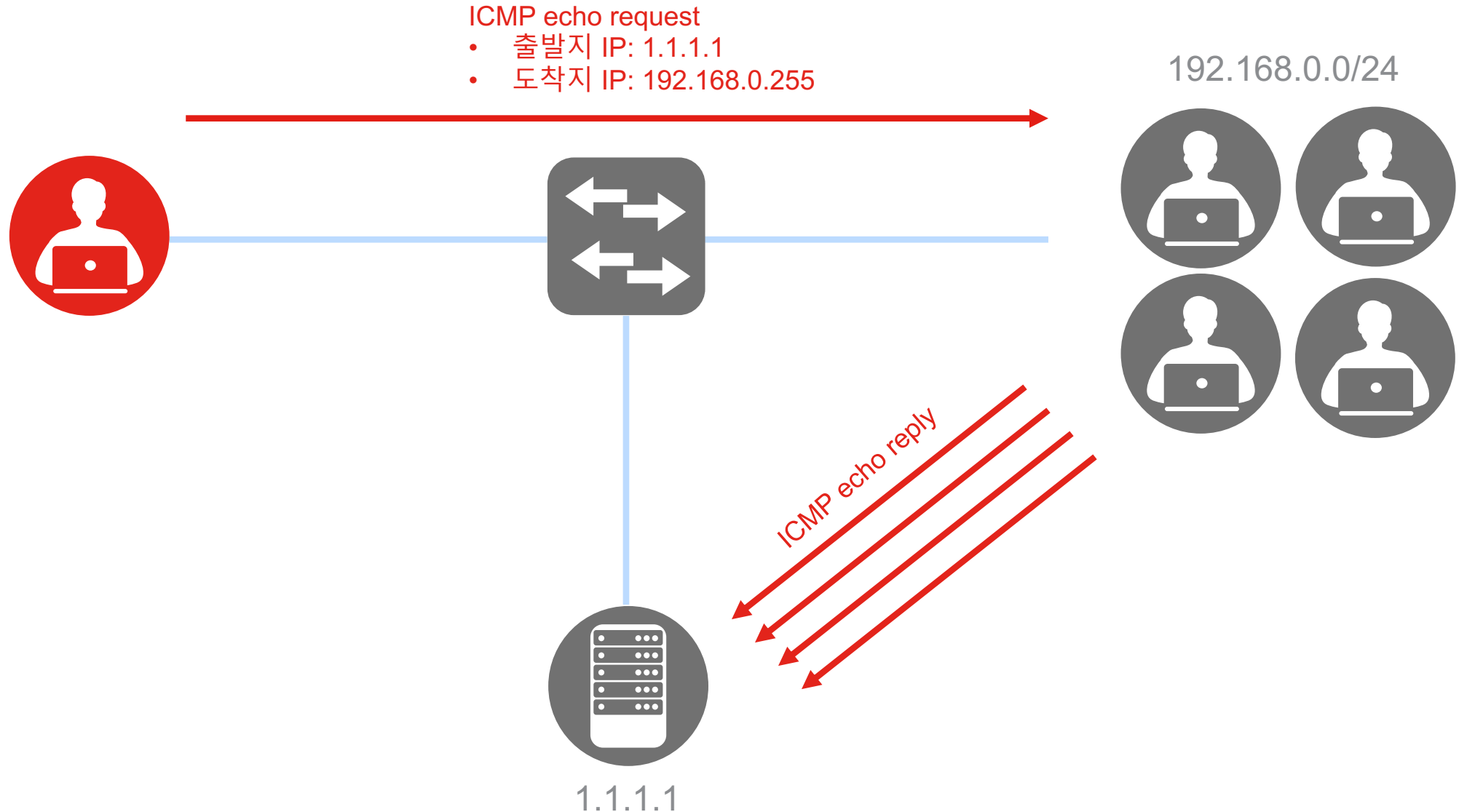
- SYN flooding



DDoS (Distributed Denial of Service) 공격

- 여러 대의 컴퓨터가 하나의 시스템이나 네트워크 서비스에 대해 동시에 DoS공격을 수행함으로써 정상적인 서비스 제공할 수 없게 방해하는 공격
- 전 세계에 분산된 수백, 수천 대의 감염된 기기(좀비 컴퓨터, 봇넷)로부터 발생
- 많은 수의 감염된 기기로부터 발생하는 막대한 양의 트래픽을 생성하여 대상 서비스를 마비
- 다수의 출처에서 공격이 이루어지기 때문에 차단이 어려움

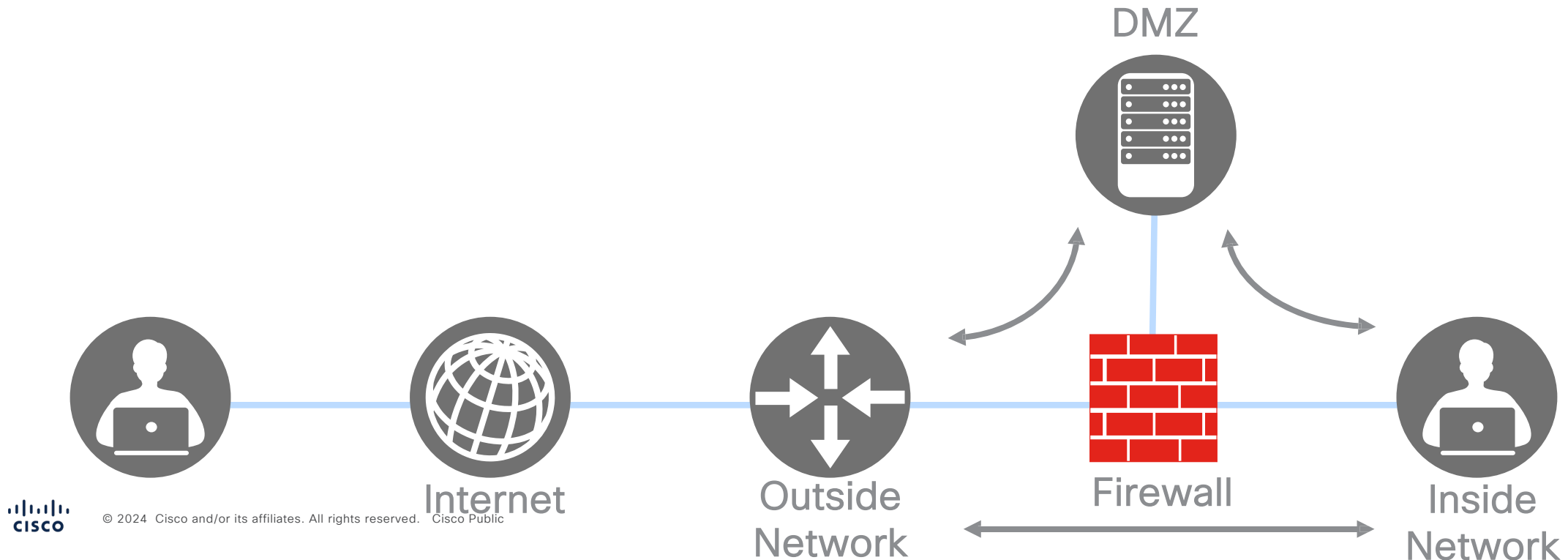
DDoS 공격 - Smurf



3. 방화벽이란?

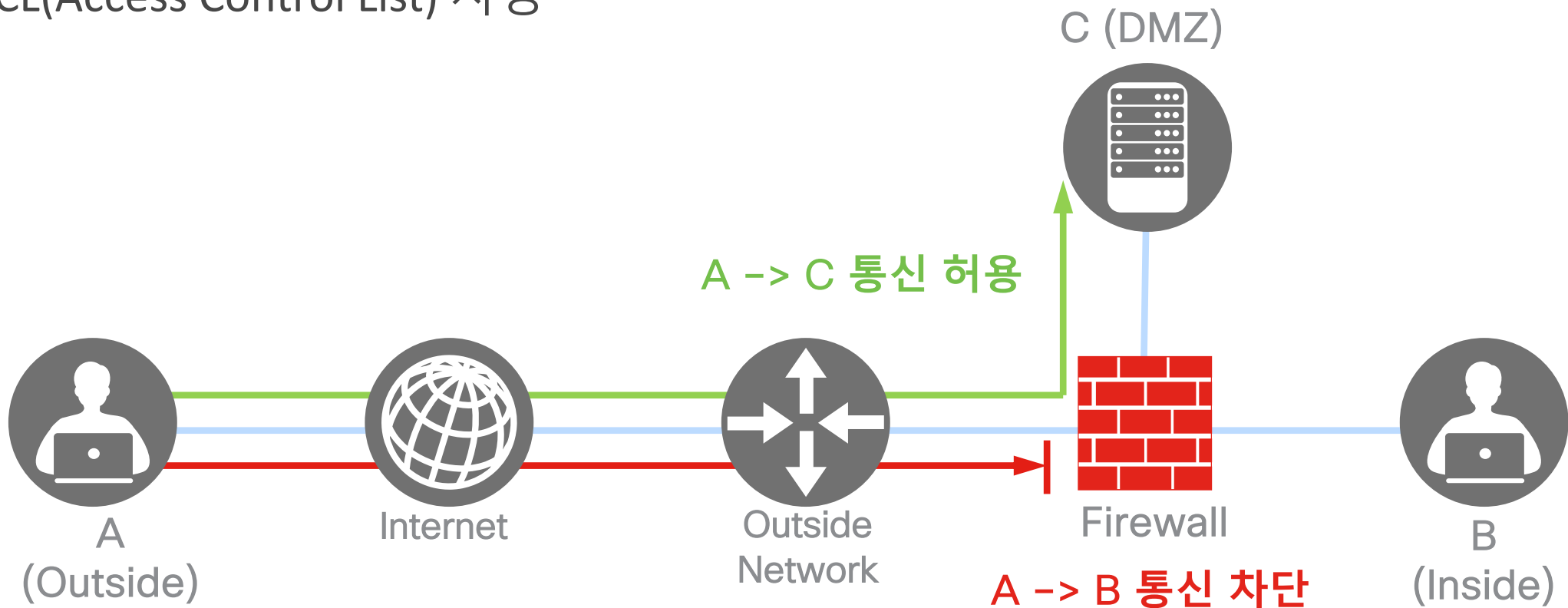
방화벽이란?

- 들어오고 나가는 네트워크 트래픽을 모니터링하고 정의된 보안 규칙 세트를 기반으로 특정 트래픽을 허용할지 또는 차단할지를 결정하는 네트워크 보안 장치
- 신뢰할 수 있는 내부 네트워크와 인터넷과 같은 신뢰할 수 없는 외부 네트워크 사이에 보안성 있고 통제된 장벽을 구축



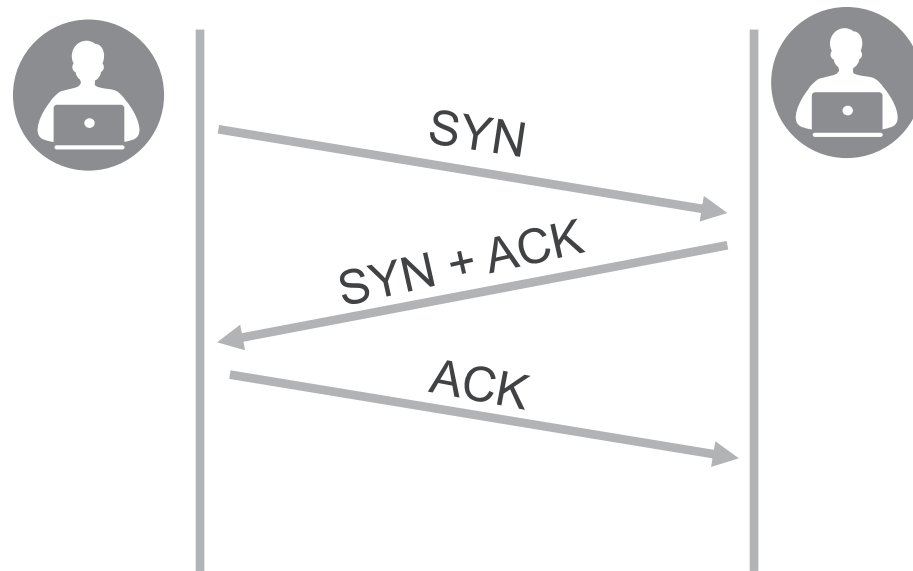
Packet Filtering

- 목적지 및 출처 주소/포트를 기준으로 네트워크로 허용되는 정보를 제한
- ACL(Access Control List) 사용

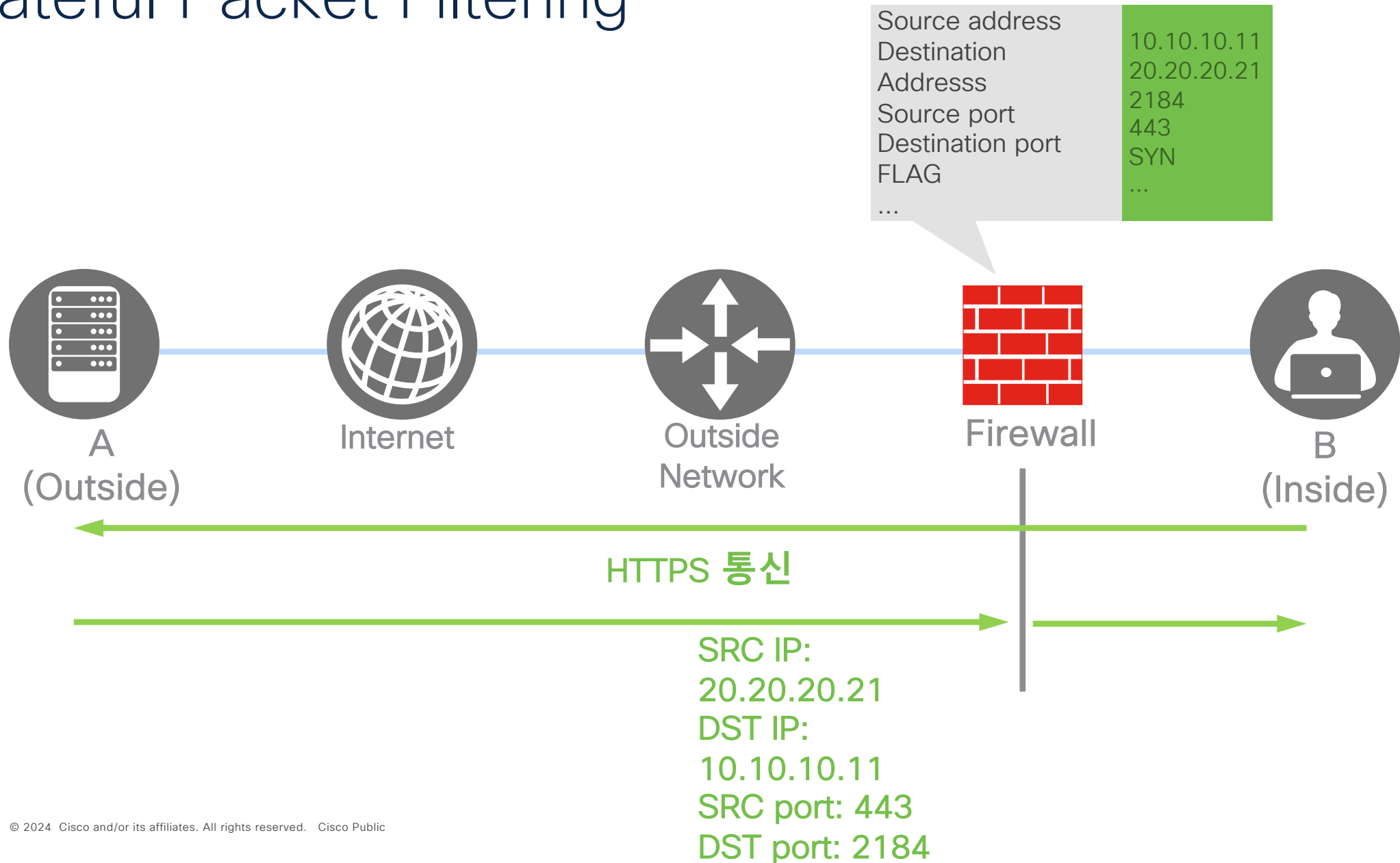


Stateful Packet Filtering

- 네트워크 연결 상태에 대한 정보를 기억하고 이를 바탕으로 패킷을 필터링
- TCP 연결을 설정하는 초기 핸드셰이크에서 SYN, ACK 등의 플래그를 추적하고, 해당 연결이 활성화되어 있는 동안에만 데이터 패킷을 통과
- 더 많은 리소스와 처리 능력을 요구함



Stateful Packet Filtering



4. ACL(Access Control List)

ACL (Access Control List)

- 네트워크 트래픽을 허용 또는 거부할지 정한 규칙들로 구성된 목록
- 목록에 포함된 규칙(Access Control Entry) 들을 순차적으로 확인하여, 네트워크 트래픽 허용 여부를 정함
- ACL을 만들고, 이 ACL을 인터페이스에 적용하며 인터페이스 하나의 액세스 목록만 허용
- ACL에서 패킷을 거부하게 되면 패킷을 삭제하고, 호스트에 연결할 수 없다는 ICMP 메시지를 반환함
- 종류: Standard ACL, Extended ACL

보안적합성 검증항목

3. 정보흐름 통제

■ 3.1 관리자가 설정한 ACL 규칙에 따라 트래픽을 제어하는 기능

3.1.1

필수



장비 별로 다음과 같은 물리적 인터페이스 기반의 ACL 기능을 제공해야 한다.

요구항목

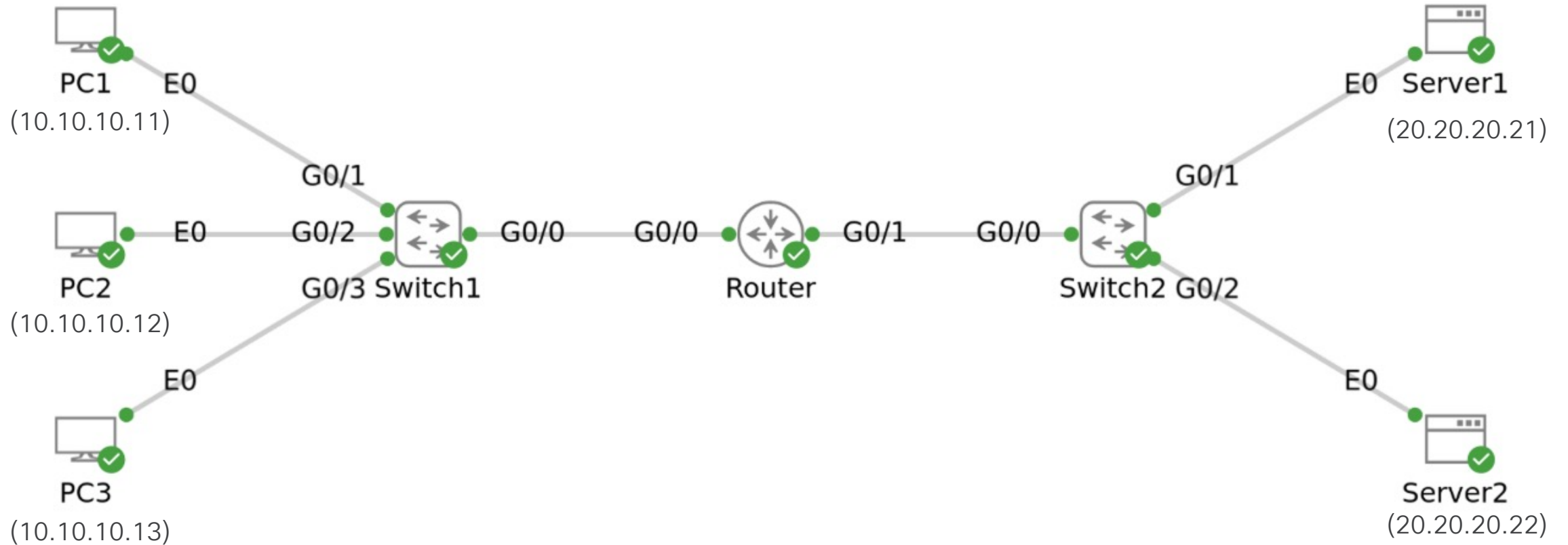
- ① L3 장비 : IP Address(Source, Destination)
- ② L4 장비 : IP Address(Source, Destination), Port(Source, Destination)
- ③ L5 이상 장비 : IP Address(Source, Destination), Port(Source, Destination), Protocol

참고 사항

- ① 물리적 인터페이스는 대상 장비와 외부 장비(또는 호스트) 간의 통신을 위해 케이블(UTP, 광섬유) 등을 직접 연결하는 인터페이스를 의미한다.
- ② 통상적으로 통신 용량에 따라, △ethernet △fast-ethernet △gigabit-ethernet 포트로 지칭된다.

국가용 보안요구사항 v3.0 - 네트워크 장비 p.13

예제 토폴로지



Standard ACL 예제 1 – ACL 설정

1. ACL 정의

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#permit 10.10.10.11 log
Router(config-std-nacl)#deny 10.10.10.12 log
Router(config-std-nacl)#permit any log
```

2. 인터페이스 적용

```
Router(config)#interface GigabitEthernet 0/0
Router(config-if)#ip access-group 1 in
```

Standard ACL 예제 1 – PC1(10.10.10.11) ping 테스트

```
PC1:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
64 bytes from 20.20.20.21: seq=0 ttl=42 time=1.625 ms
64 bytes from 20.20.20.21: seq=1 ttl=42 time=1.907 ms
64 bytes from 20.20.20.21: seq=2 ttl=42 time=2.380 ms
64 bytes from 20.20.20.21: seq=3 ttl=42 time=1.923 ms
^C
--- 20.20.20.21 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.625/1.958/2.380 ms
PC1:~$
```

Router#

*May 1 05:38:03.013: %SEC-6-IPACCESSLOGNP: list 1 permitted 0 10.10.10.11 -> 20.20.20.21, 4 packets

Router#

Router#show ip access-lists 1

Standard IP access list 1

10 permit 10.10.10.11 log (4 matches)

20 deny 10.10.10.12 log

30 permit any log

Router#

Standard ACL 예제 1 – PC2(10.10.10.12) ping 테스트

```
PC2:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
^C
--- 20.20.20.21 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
PC2:~$
```

Router#

*May 1 05:43:46.654: %SEC-6-IPACCESSLOGNP: list 1 denied 0 10.10.10.12 -> 20.20.20.21, 1 packet

Router#

Router#show ip access-lists 1

Standard IP access list 1

10 permit 10.10.10.11 log (4 matches)

20 deny 10.10.10.12 log (10 matches)

30 permit any log

Router#

Standard ACL 예제 1 – PC3(10.10.10.13) ping 테스트

```
PC3:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
64 bytes from 20.20.20.21: seq=0 ttl=42 time=3.339 ms
64 bytes from 20.20.20.21: seq=1 ttl=42 time=1.614 ms
64 bytes from 20.20.20.21: seq=2 ttl=42 time=2.024 ms
64 bytes from 20.20.20.21: seq=3 ttl=42 time=1.957 ms
^C
--- 20.20.20.21 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.614/2.233/3.339 ms
PC3:~$
```

Router#

*May 1 06:16:04.115: %SEC-6-IPACCESSLOGNP: list 1 permitted 0 10.10.10.13 -> 20.20.20.21, 1 packet

Router#

Router#show ip access-lists 1

Standard IP access list 1

10 permit 10.10.10.11 log (4 matches)

20 deny 10.10.10.12 log (10 matches)

30 permit any log (4 matches)

Router#

Standard ACL 예제 2 – Sequence number

```
Router#show ip access-lists 1
Standard IP access list 1
 10 permit 10.10.10.11 log
 20 deny 10.10.10.12 log
 30 permit any log
Router#
```

- 가장 첫 항목에 Sequence number를 지정하지 않는 경우 10이 할당됨
- Sequence number 없이 항목을 추가할 때는 마지막 Sequence number에서 10 추가되어 목록 끝에 배치
- Sequence number를 제외하고 기존 항목과 일치하는 항목을 입력하면 무시함
- 최대 Sequence number : 2147483647

Standard ACL 예제 2 - 삭제

1. ACE 삭제

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#no permit 10.10.10.11
Router(config-std-nacl)#no 30
Router(config-std-nacl)#
```

Before

```
Router#show ip access-lists 1
Standard IP access list 1
  10 permit 10.10.10.11 log
  20 deny 10.10.10.12 log
  30 permit any log
Router#
```

After

```
Router#show ip access-list 1
Standard IP access list 1
  20 deny 10.10.10.12 log
Router#
```

Standard ACL 예제 2 - 추가 Before

2. ACE 추가

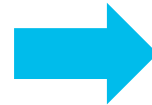
```
Router#show ip access-list 1
Standard IP access list 1
  20 deny 10.10.10.12 log
Router#
```

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#deny 10.10.10.13 log
Router(config-std-nacl)#
```



```
Router#show ip access-list 1
Standard IP access list 1
  20 deny 10.10.10.12 log
  30 deny 10.10.10.13 log
Router#
```

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#5 permit 10.10.10.11 log
Router(config-std-nacl)#
```



```
Router#show ip access-list 1
Standard IP access list 1
  5 permit 10.10.10.11 log
  20 deny 10.10.10.12 log
  30 deny 10.10.10.13 log
Router#
```

Standard ACL 예제 2 - 추가

2. ACE 추가

Before

```
Router#show ip access-list 1
Standard IP access list 1
  5 permit 10.10.10.11 log
 20 deny   10.10.10.12 log
 30 deny   10.10.10.13 log
Router#
```

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#50 deny 10.10.10.12 log
Router(config-std-nacl)#
```



```
Router#show ip access-list 1
Standard IP access list 1
  5 permit 10.10.10.11 log
 20 deny   10.10.10.12 log
 30 deny   10.10.10.13 log
Router#
```

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#2 permit 10.10.10.12 log
Router(config-std-nacl)#
```



```
Router#show ip access-list 1
Standard IP access list 1
  5 permit 10.10.10.11 log
 20 deny   10.10.10.12 log
  2 permit 10.10.10.12 log
 30 deny   10.10.10.13 log
Router#
```

Standard ACL 예제 2 – PC2(10.10.10.12) ping 테스트

```
PC2:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
^C
--- 20.20.20.21 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss
PC2:~$
```

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#no 5
Router#
Router#show ip access-list 1
Standard IP access list 1
 20 deny 10.10.10.12 log (12 matches)
 2 permit 10.10.10.12 log
 30 deny 10.10.10.13 log
Router#
```

Standard ACL 예제 2 – PC1(10.10.10.11) ping 테스트

```
PC1:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
^C
--- 20.20.20.21 ping statistics ---
7 packets transmitted, 0 packets received, 100% packet loss
PC1:~$
```

Router#

Router#show ip access-list 1

Standard IP access list 1

20 deny 10.10.10.12 log (12 matches)

2 permit 10.10.10.12 log

30 deny 10.10.10.13 log

Router#

Router#

모든 항목과 일치하지 않는 경우 차단!
(마지막에 deny any가 암시적으로 있다고 생각)

Standard ACL 예제 3 - IP 대역 설정

```
Router(config)#ip access-list standard 1  
Router(config-std-nacl)#permit 10.10.10.11 log  
Router(config-std-nacl)#deny 10.10.10.0 0.0.0.255 log  
Router(config-std-nacl)#
```

Wildcard mask 로 대역 지정

Standard ACL 예제 3 - Wildcard mask

- wildcard mask를 2진수로 나타냈을 때,
 - 비트 0은 해당 비트 값을 확인한다는 의미
 - 비트 1은 해당 비트 값을 무시한다는 의미
- 각 옥텟을 10진수로 표기했을 때 0은 유효, 255이면 무시

| | | | | | |
|---------------|----|-----|-----|------|----|
| 10.10.10.11 | 10 | .10 | .10 | .11 | |
| wildcard mask | 0 | .0 | .0 | .255 | |
| 결과 | 10 | .10 | .10 | .0 | 무시 |

| | | | | | |
|---------------|----|-----|-----|------|----|
| 20.20.20.21 | 20 | .20 | .20 | .21 | |
| wildcard mask | 0 | .0 | .0 | .255 | |
| 결과 | 20 | .20 | .20 | .0 | 무시 |

Standard ACL 예제 3 - Wildcard mask

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#permit 10.10.10.11 log
Router(config-std-nacl)#deny 10.10.10.0 0.0.0.255 log
```

- ACL에 있는 주소(10.10.10.0) 연산

| | |
|---------------|---|
| 10.10.10.0 | 0000 1010 . 0000 1010 . 0000 1010 . 0000 0000 |
| wildcard mask | 0000 0000 . 0000 0000 . 0000 0000 . 1111 1111 |
| 결과 | 0000 1010 . 0000 1010 . 0000 1010 . 0000 0000 |

- PC2(10.10.10.12)가 보내는 경우

| | |
|---------------|---|
| 10.10.10.12 | 0000 1010 . 0000 1010 . 0000 1010 . 0000 1100 |
| wildcard mask | 0000 0000 . 0000 0000 . 0000 0000 . 1111 1111 |
| 결과 | 0000 1010 . 0000 1010 . 0000 1010 . 0000 0000 |

결과가
서로 일치하기 때문에 Deny 됨

Standard ACL 예제 3 - Wildcard mask

```
Router(config)#ip access-list standard 1
Router(config-std-nacl)#permit 10.10.10.11 log
Router(config-std-nacl)#deny 10.10.10.0 0.0.0.255 log
```

- ACL에 있는 주소 (10.10.10.0) 연산

| | |
|---------------|---|
| 10.10.10.0 | 0000 1010 . 0000 1010 . 0000 1010 . 0000 0000 |
| wildcard mask | 0000 0000 . 0000 0000 . 0000 0000 . 1111 1111 |
| 결과 | 0000 1010 . 0000 1010 . 0000 1010 . 0000 0000 |

- Server1(20.20.20.21)가 보내는 경우

| | |
|---------------|---|
| 20.20.20.21 | 0001 0100 . 0001 0100 . 0001 0100 . 0001 0101 |
| wildcard mask | 0000 0000 . 0000 0000 . 0000 0000 . 1111 1111 |
| 결과 | 0001 0100 . 0001 0100 . 0001 0100 . 0000 0000 |

결과가 서로 다르기 때문에
다음 항목으로 넘어감

Standard ACL 예제 3 - Wildcard mask

- Subnet mask는 가장 좌측부터 1 비트가 연속되어야 하며, 연속된 1 비트 중간에 0이 들어갈 수 없으나 Wildcard mask는 상관 없음
 - 1111 1111 . 1111 1111 . 1111 1111 . 0000 0000 (Subnet mask 사용 가능)
 - 1111 1111 . 1111 1001 . 1111 1111 . 0000 0000 (Subnet mask 사용 불가: 중간에 0비트)
 - 0011 1111 . 1111 1111 . 1111 1111 . 0000 0000 (Subnet mask 사용 불가: 시작에 0비트)

Standard ACL 예제 3 – Wildcard mask

- 연속된 0 비트 이후 연속된 1 비트가 사용된 wildcard mask의 경우, 비트를 반전하여 마치 Subnet 인 것처럼 네트워크 대역을 계산할 수 있음

```
Router(config)#ip access-list standard 1  
Router(config-std-nacl)#permit 10.10.10.11 log  
Router(config-std-nacl)#deny 10.10.10.0 0.0.0.255 log
```



비트 반전하면 255.255.255.0 Subnet mask인 것처럼 계산
호스트 주소인 10.10.10.0 – 10.10.10.255까지 deny됨

Standard ACL 예제 3 – ping 테스트

- PC1(10.10.10.11)에서 ping

```
PC1:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
64 bytes from 20.20.20.21: seq=0 ttl=42 time=1.817 ms
64 bytes from 20.20.20.21: seq=1 ttl=42 time=1.785 ms
64 bytes from 20.20.20.21: seq=2 ttl=42 time=1.366 ms
64 bytes from 20.20.20.21: seq=3 ttl=42 time=1.919 ms
^C
--- 20.20.20.21 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.366/1.721/1.919 ms
PC1:~$
```

- PC2(10.10.10.12)에서 ping

```
PC2:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
^C
--- 20.20.20.21 ping statistics ---
4 packets transmitted, 0 packets received, 100% packet loss
PC2:~$
```

Router#

*May 1 07:43:03.010: %SEC-6-IPACCESSLOGNP: list 1 permitted 0 10.10.10.11 -> 20.20.20.21, 3 packets

*May 1 07:44:03.010: %SEC-6-IPACCESSLOGNP: list 1 denied 0 10.10.10.12 -> 20.20.20.21, 8 packets

Router#

Router#show ip access-lists 1

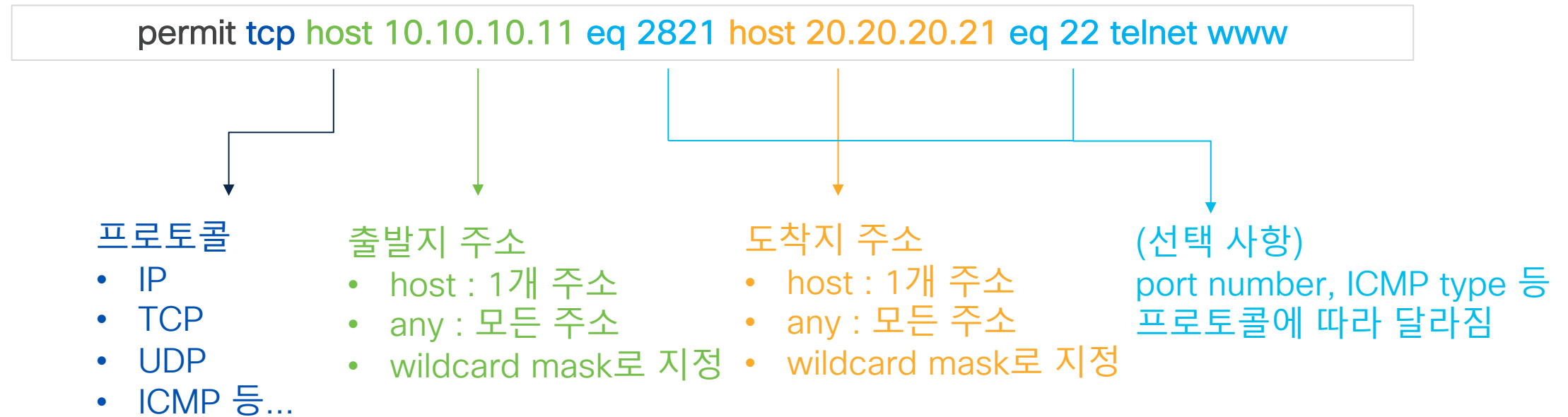
Standard IP access list 1

10 permit 10.10.10.11 log (4 matches)

20 deny 10.10.10.0, wildcard bits 0.0.0.255 log (8 matches)

Router#

Extended ACL



ACL 생성시 number 또는 name 지정

- Standard ACL은 1-99, 1300-1999의 숫자를 사용하여 지정
- Extended ACL은 100-199, 2000-2699의 숫자를 사용하여 지정

Standard
ACL

```
Router(config)#ip access-list standard ?  
<1-99> Standard IP access-list number  
<1300-1999> Standard IP access-list number (expanded range)  
WORD Access-list name
```

Extended
ACL

```
Router(config)#ip access-list extended ?  
<100-199> Extended IP access-list number  
<2000-2699> Extended IP access-list number (expanded range)  
WORD Access-list name
```

ACL 생성시 number 또는 name 지정

- Extended ACL을 숫자로 지정하는 경우 TCP, UDP 포트를 여러 개 지정할 수 없음

```
Router(config)#ip access-list extended 2000
Router(config-ext-nacl)#permit udp any any eq snmp ntp
% Multiple values are allowed on named ACLs only
Router(config-ext-nacl)#
```

- 숫자로 대신 이름으로 지정하는 것을 권장

```
Router(config)#ip access-list extended ext_acl
Router(config-ext-nacl)#permit udp any any eq snmp ntp
Router(config-ext-nacl)#
```

Extended ACL – IP ACE 예제

- Standard ACL에서 사용한 Entry를 그대로 Extended ACL에 사용하려는 경우

Standard
ACL

```
deny 10.10.10.0 0.0.0.255
```



Extended
ACL

```
deny ip 10.10.10.0 0.0.0.255 any
```

Extended ACL – IP ACE 예제

| 설명 | Action | protocol | Source address | Destination address |
|-------------------------------------|--------|----------|----------------------|----------------------|
| 10.10.10.11에서 출발하는 모든 트래픽 차단 | deny | ip | host 10.10.10.10.11 | any |
| 10.10.10.11로 도착하는 모든 트래픽 차단 | deny | ip | any | host 10.10.10.11 |
| 10.10.10.0/24에서 20.20.20.0/24 통신 차단 | deny | ip | 10.10.10.0 0.0.0.255 | 20.20.20.0 0.0.0.255 |
| 10.10.10.0/24에서 20.20.20.21 통신 차단 | deny | ip | 10.10.10.0 0.0.0.255 | host 20.20.20.21 |

Extended ACL – TCP ACE 예제

- Source port, Destination port는 선택사항

| 설명 | Action | proto col | source address | Source port | Destination address | Destination port |
|-------------------------------|--------|--------------|----------------|---------------|---------------------|------------------|
| 모든 SSH 접속 차단 | deny | tcp | any | | any | eq 22 |
| 200-300범위를 Source port로 사용 차단 | deny | tcp | any | range 200 300 | any | |
| ntp, snmp 사용 차단 | deny | udp | any | | any | eq ntp snmp |

Extended ACL – ICMP ACE 예제

| 설명 | Action | protocol | source address | Destination address | ICMP message type |
|------------|--------|----------|----------------|---------------------|-------------------|
| 모든 핑 요청 차단 | deny | icmp | any | any | echo |
| 모든 핑 응답 차단 | deny | icmp | any | any | echo-reply |

Extended ACL

```
Router(config)#ip access-list extended 100
Router(config-ext-nacl)#permit tcp host 10.10.10.11 host 20.20.20.21 eq 22 log ①
Router(config-ext-nacl)#deny icmp host 10.10.10.11 any echo log ②
Router(config-ext-nacl)#permit icmp 10.10.10.0 0.0.0.255 any echo log ③
Router(config-ext-nacl)#
Router(config-ext-nacl)#
Router(config-ext-nacl)#
Router(config-ext-nacl)#
Router(config-ext-nacl)#interface GigabitEthernet 0/0
Router(config-if)#ip access-group 100 in
```

① PC1(10.10.10.11)에서 Server1(20.20.20.21)로 SSH 허용

② PC1(10.10.10.11)에서 모든 핑 차단

③ 10.10.10.0/0에서 모든 핑 허용

이외 모든 트래픽 차단

Extended ACL 예제 3 - ping 테스트

- PC1(10.10.10.11)에서 Server1로 SSH 성공, ping 실패

```
PC1:~$ ssh cisco@20.20.20.21
cisco@20.20.20.21's password:
( '>')
/) TC (\   Core is distributed with ABSOLUTELY NO WARRANTY.
(/-_-_-\)      www.tinycorelinux.net

cisco@Server1:~$
```

```
PC1:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
^C
--- 20.20.20.21 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
PC1:~$
```

```
*May  1 10:29:55.232: %SEC-6-IPACCESSLOGP: list 100 permitted tcp 10.10.10.11(51396) -> 20.20.20.21(22), 1
packet
*May  1 10:30:42.096: %SEC-6-IPACCESSLOGDP: list 100 denied icmp 10.10.10.11 -> 20.20.20.21 (8/0), 1 packet
*May  1 10:31:32.677: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 10.10.10.12 -> 20.20.20.21 (8/0), 1 packet
Router#
Router#show ip access-list 100
Extended IP access list 100
 10 permit tcp host 10.10.10.11 host 20.20.20.21 eq 22 log (39 matches)
 20 deny icmp host 10.10.10.11 any echo log (5 matches)
 30 permit icmp 10.10.10.0 0.0.0.255 any echo log (4 matches)
Router#
```


Extended ACL 예제 3 - ping 테스트

- PC2(10.10.10.11)에서 Server1로 ping 성공, SSH 실패

```
PC2:~$ ping 20.20.20.21
PING 20.20.20.21 (20.20.20.21): 56 data bytes
64 bytes from 20.20.20.21: seq=0 ttl=42 time=2.018 ms
64 bytes from 20.20.20.21: seq=1 ttl=42 time=2.237 ms
64 bytes from 20.20.20.21: seq=2 ttl=42 time=2.196 ms
64 bytes from 20.20.20.21: seq=3 ttl=42 time=1.699 ms
^C
--- 20.20.20.21 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.699/2.037/2.237 ms
PC2:~$
```

```
PC2:~$ ssh cisco@20.20.20.21
ssh: connect to host 20.20.20.21 port 22: Host is unreachable
PC2:~$
```

*May 1 10:29:55.232: %SEC-6-IPACCESSLOGP: list 100 permitted tcp 10.10.10.11(51396) -> 20.20.20.21(22), 1 packet

*May 1 10:30:42.096: %SEC-6-IPACCESSLOGDP: list 100 denied icmp 10.10.10.11 -> 20.20.20.21 (8/0), 1 packet

*May 1 10:31:32.677: %SEC-6-IPACCESSLOGDP: list 100 permitted icmp 10.10.10.12 -> 20.20.20.21 (8/0), 1 packet

Router#

Router#show ip access-list 100

Extended IP access list 100

10 permit tcp host 10.10.10.11 host 20.20.20.21 eq 22 log (39 matches)

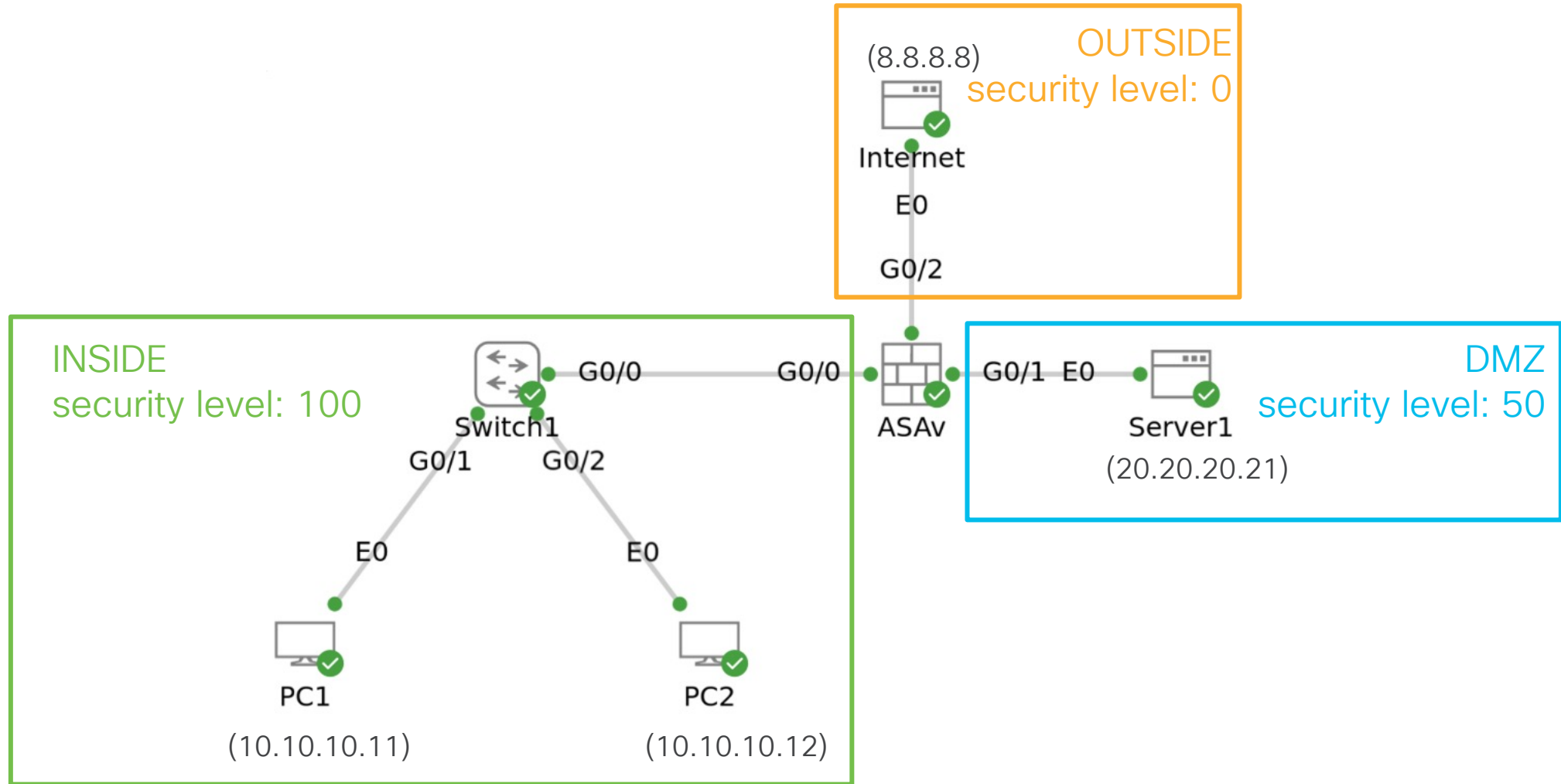
20 deny icmp host 10.10.10.11 any echo log (5 matches)

30 permit icmp 10.10.10.0 0.0.0.255 any echo log (4 matches)

Router#

방화벽 ACL

ASA 테스트 토폴로지

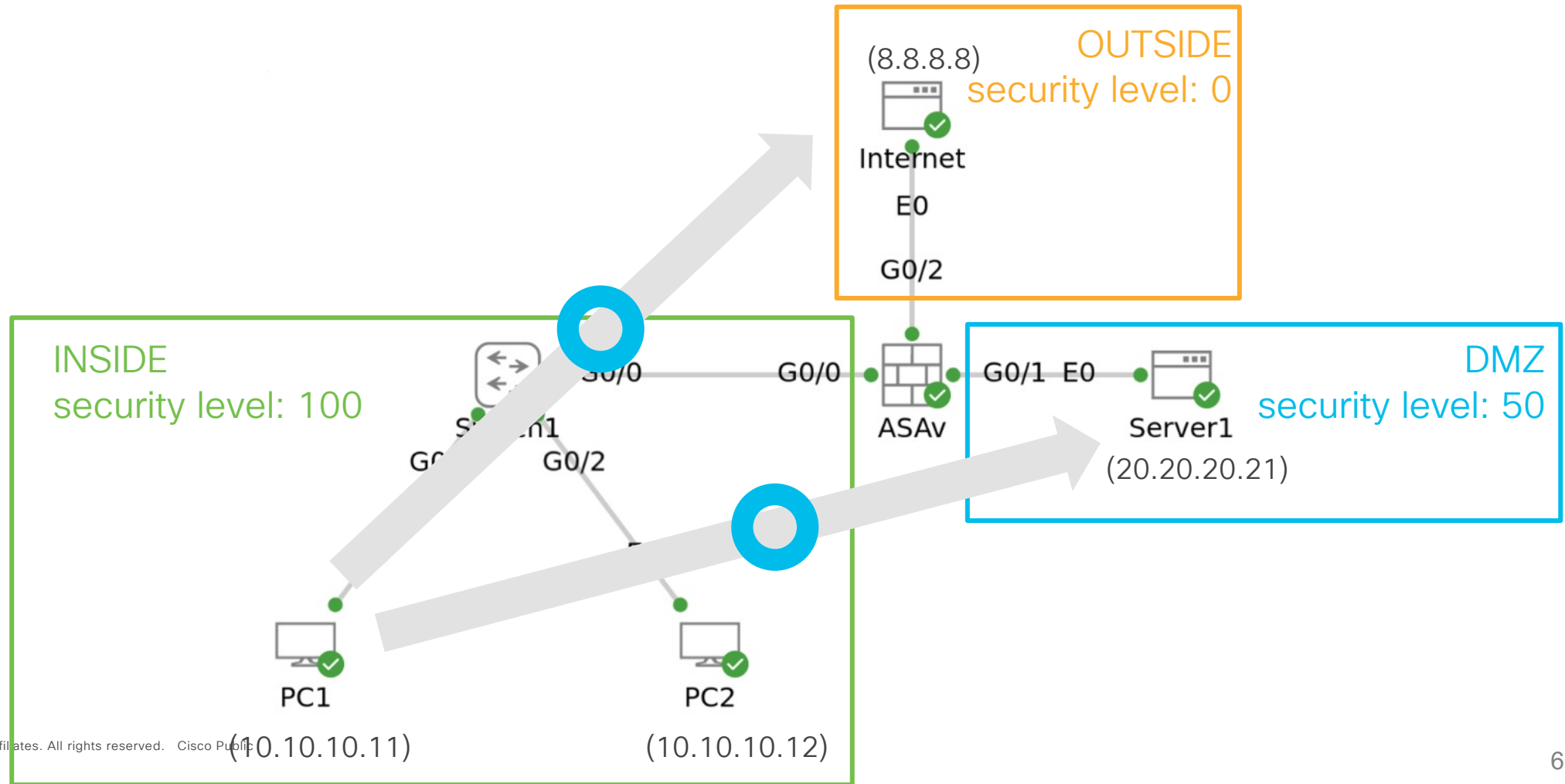


Security Level

- 인터페이스에 할당된 값으로서, 해당 인터페이스의 신뢰도를 나타냄
- 0에서 100 사이 값을 지정할 수 있으며, 높은 숫자가 더 높은 신뢰도를 의미
- 높은 보안 수준에서 낮은 보안 수준으로의 트래픽 흐름은 허용
- 낮은 보안 수준에서 높은 보안 수준으로의 트래픽 흐름은 차단
 - 이를 통과시키려면 명시적인 접근 제어 정책(ACL: Access Control List)을 구성 필요
- 같은 보안 수준을 가진 인터페이스 간의 트래픽 흐름은 차단
 - "same-security-traffic permit inter-interface" 명령을 사용하여 이러한 트래픽을 허용

Security Level 예시

- Security level이 100인 PC1에서 Internet, Server1 접속 성공



Security Level 예시

- Security level이 100인 PC1에서 Internet, Server1 접속 성공

```
PC1:~$ ssh cisco@8.8.8.8
cisco@8.8.8.8's password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

Internet:~$ exit
Connection to 8.8.8.8 closed.
PC1:~$
```

```
PC1:~$ ssh cisco@20.20.20.21
cisco@20.20.20.21's password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

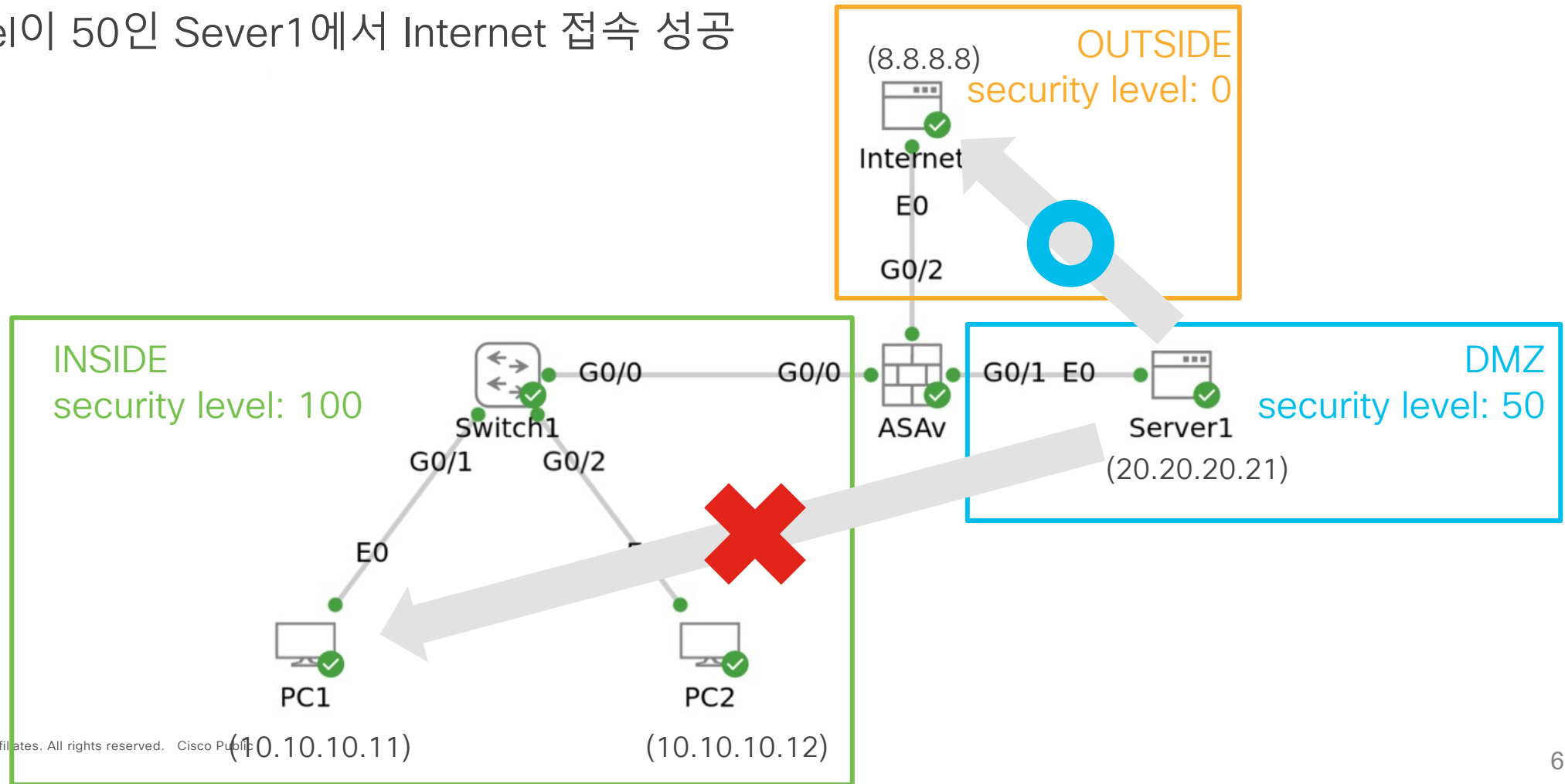
You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

Server1:~$
```

Security Level 예시

- Security level이 50인 Sever1에서 PC1 접속 실패
- Security level이 50인 Sever1에서 Internet 접속 성공



Security Level 예시

- Security level이 50인 Server1에서 PC1 접속 실패
- Security level이 50인 Server1에서 Internet 접속 성공

```
Server1:~$ ssh cisco@10.10.10.11  
^C  
Server1:~$
```

```
Server1:~$ ssh cisco@8.8.8.8  
cisco@8.8.8.8's password:  
Welcome to Alpine!  
  
The Alpine Wiki contains a large amount of how-to guides and general  
information about administrating Alpine systems.  
See <http://wiki.alpinelinux.org/>.  
  
You can setup the system with the command: setup-alpine  
  
You may change this message by editing /etc/motd.  
  
Internet:~$ ^C
```


nameif

- nameif 명령어로 인터페이스 이름을 지정
 - inside 로 지정 시, security level 은 100으로 설정
 - inside 이외로 지정 시, security level은 0으로 설정(DMZ, outside 포함)

```
Firewall(config)# interface gi 0/1
Firewall(config-if)# nameif DMZ
INFO: Security level for "DMZ" set to 0 by default.
Firewall(config-if)#
Firewall(config-if)# security-level 50
Firewall(config-if)# ip addr 20.20.20.1 255.255.255.0
Firewall(config-if)# no shut
```

ASA에서 ACL 지정하기

1. configuration mode 에서 ACE를 설정함

```
Firewall(config)# access-list DENY_ALL extended deny ip any any log  
Firewall(config)#
```

2. configuration mode 에서 ACL을 적용하며, 적용할 방향과 인터페이스 지정

```
Firewall(config)# access-group DENY_ALL in interface INSIDE  
Firewall(config)#
```

ASA에서 ACL 지정하기

- INSIDE 인터페이스와 연결된 PC1에서 DMZ, Outside로 SSH 통신 불가

```
PC1:~$ ssh cisco@8.8.8.8
ssh: connect to host 8.8.8.8 port 22: Connection refused
PC1:~$
PC1:~$
PC1:~$ ssh cisco@20.20.20.21
ssh: connect to host 20.20.20.21 port 22: Connection refused
PC1:~$
```

- DMZ 인터페이스와 연결된 Server1에서 Outside로 SSH 통신 성공

```
Server1:~$ ssh cisco@8.8.8.8
cisco@8.8.8.8's password:
Welcome to Alpine!

The Alpine Wiki contains a large amount of how-to guides and general
information about administrating Alpine systems.
See <http://wiki.alpinelinux.org/>.

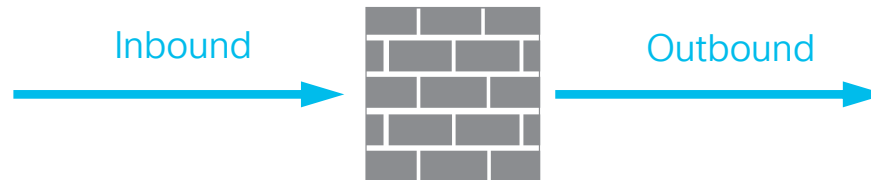
You can setup the system with the command: setup-alpine

You may change this message by editing /etc/motd.

Internet:~$
```

ACL 적용 방향

- inbound: 인터페이스로 들어오는 트래픽에 ACL 적용
- outbound: 인터페이스에서 나가는 트래픽에 ACL 적용
- global: interface에 적용한 inbound, outbound 설정에 걸리지 않는 경우



Global ACL 예시

1. ACE를 설정 방법은 이전과 동일함

- INSIDE인 10.10.10.11가 도착지인 핑 요청 트래픽을 허용함
- INSIDE인 10.10.10.11가 출발지인 핑 응답 트래픽을 허용함

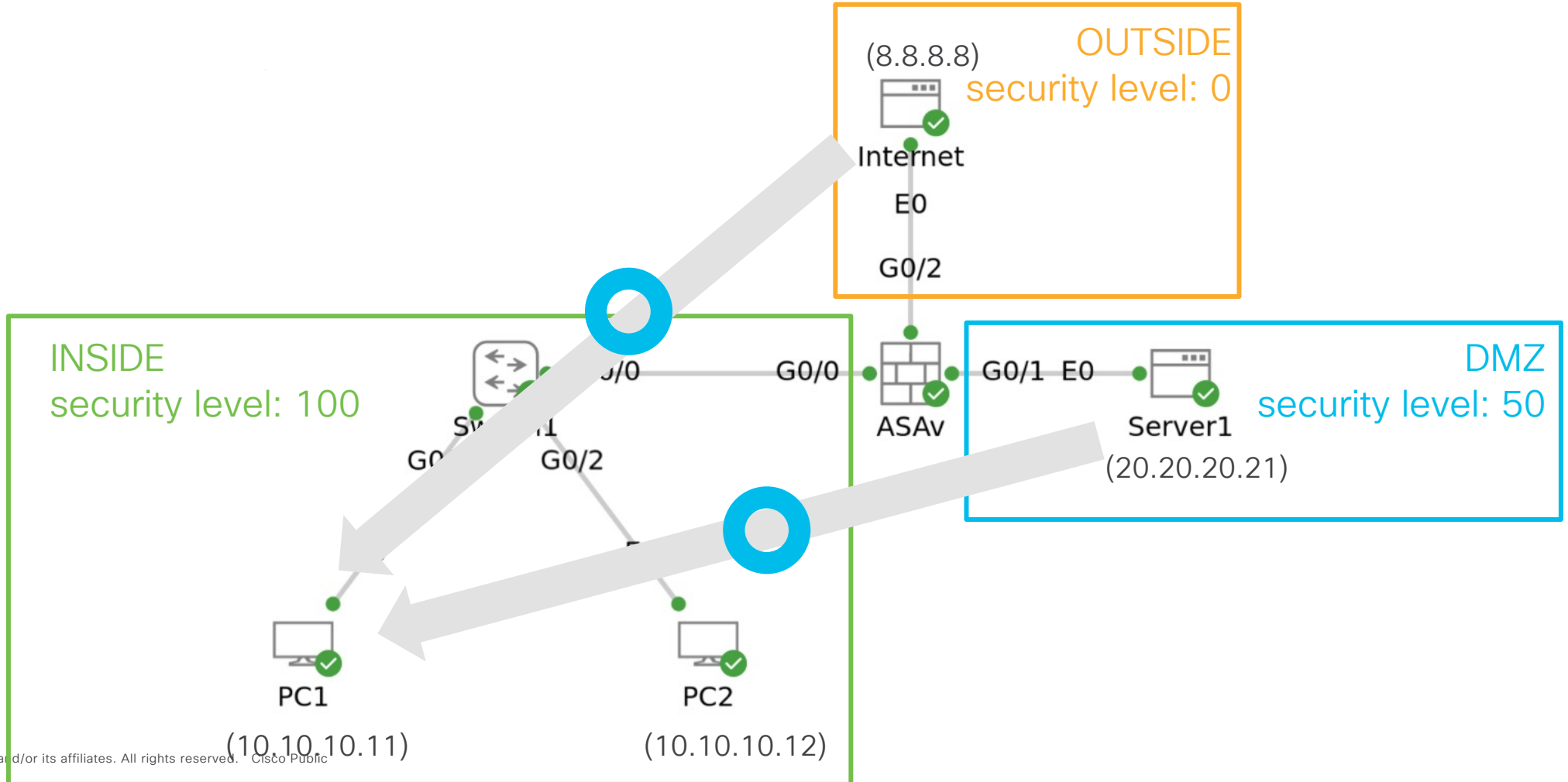
```
Firewall(config)# access-list GLOBAL_ACL extended permit icmp any host 10.10.10.11 echo
Firewall(config)# access-list GLOBAL_ACL extended permit icmp host 10.10.10.11 any echo-reply
Firewall(config)#
```

2. ACL 적용 시 global 키워드를 사용하며 인터페이스는 지정하지 않음

```
Firewall(config)# access-group GLOBAL_ACL global
Firewall(config)#
```

Global ACL 예시

- Outside, DMZ 인터페이스에 각각 설정하지 않아도 적용됨



Global ACL 예시

- Outside, DMZ 에서 PC1으로 ping 통신 성공

```
Server1:~$ ping 10.10.10.11
PING 10.10.10.11 (10.10.10.11): 56 data bytes
64 bytes from 10.10.10.11: seq=0 ttl=42 time=1.691 ms
64 bytes from 10.10.10.11: seq=1 ttl=42 time=1.534 ms
64 bytes from 10.10.10.11: seq=2 ttl=42 time=1.772 ms
64 bytes from 10.10.10.11: seq=3 ttl=42 time=1.698 ms
^C
--- 10.10.10.11 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.534/1.673/1.772 ms
Server1:~$
```

```
Internet:~$ ping 10.10.10.11
PING 10.10.10.11 (10.10.10.11): 56 data bytes
64 bytes from 10.10.10.11: seq=0 ttl=42 time=1.749 ms
64 bytes from 10.10.10.11: seq=1 ttl=42 time=1.631 ms
64 bytes from 10.10.10.11: seq=2 ttl=42 time=1.427 ms
64 bytes from 10.10.10.11: seq=3 ttl=42 time=1.677 ms
^C
--- 10.10.10.11 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.427/1.621/1.749 ms
Internet:~$ ^C
```

- Inside인터페이스와 연결된 PC1에서로 Outside로 ping 통신 실패

```
PC1:~$ ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
^C
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 0 packets received, 100% packet loss
PC1:~$
```

ASA에서 ACL 카운터 확인

```
Firewall# show access-list GLOBAL_ACL  
access-list GLOBAL_ACL; 2 elements; name hash: 0xa23a7bf2  
access-list GLOBAL_ACL line 1 extended permit icmp any host 10.10.10.11 echo-reply (hitcnt=18) 0xb85a78c6  
access-list GLOBAL_ACL line 2 extended permit icmp host 10.10.10.11 any echo (hitcnt=18) 0x73e58003
```




The bridge to possible