SNORT 실습

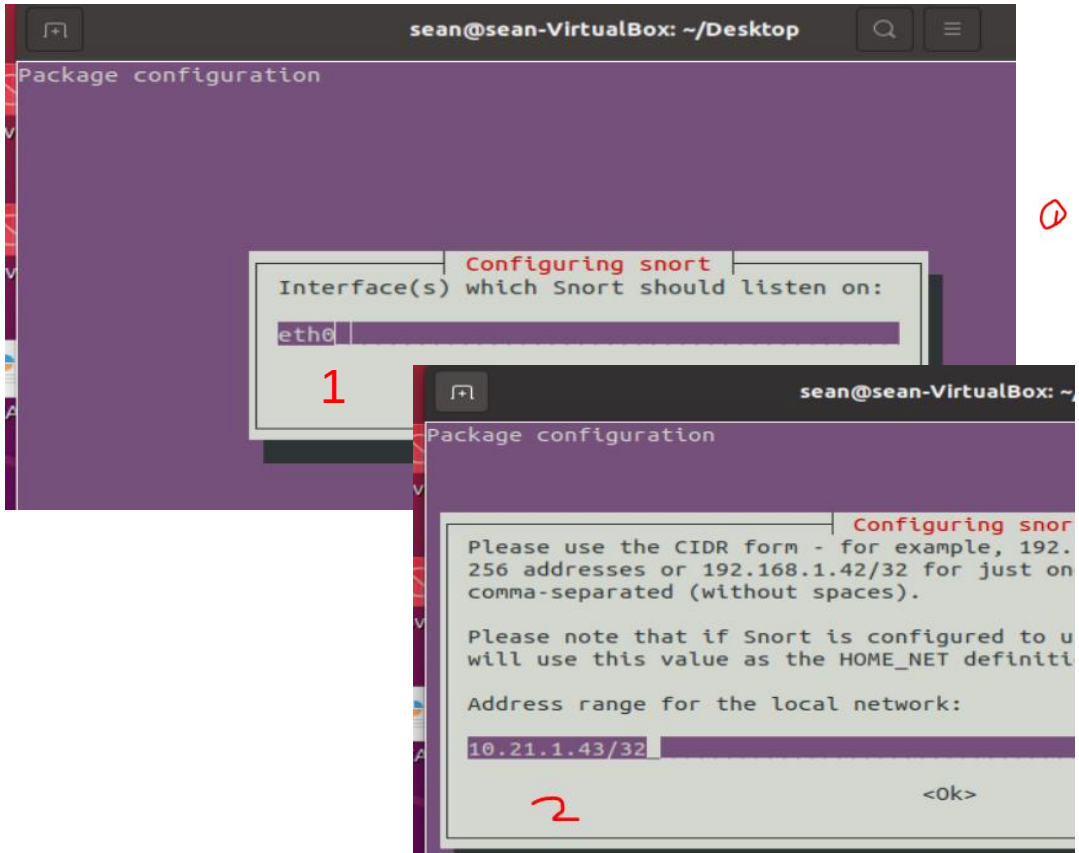
SNORT

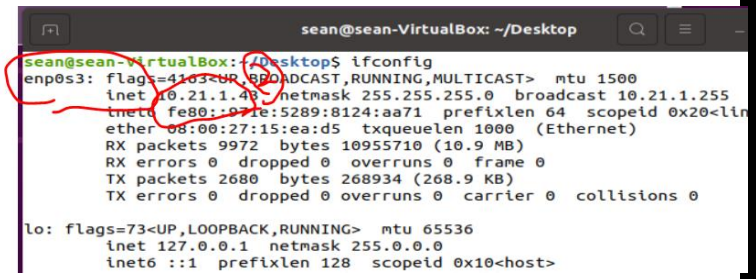
- Small : ~800k source download
- Portable : Linux, Windows, MacOS X, Solaris, BSD, IRIX, Tru64, HP-UX, etc)
- Fast : High probability of detection for a given attack on 100Mbps networks
- Configurable : Easy rules language, many reporting/logging options
- Free : GPL/Open Source Software

SNORT 설치

$ sudo apt install snort

1. interface **지정** : $ ifconfig **참조**
2. target network **지정** : ip addr/32 (4byte **주소**)





SNORT 실행

▶ 버전 보기

$ sudo su

snort –V

환경 설정

- /etc/snort/snort.debian.conf
  - **설치 시 설정**
  - **자기 ip 주소**
  - network interface

```
DEBIAN_SNORT_STARTUP="boot"
DEBIAN_SNORT_HOME_NET="10.21.1.43/32"
DEBIAN_SNORT_OPTIONS=""
DEBIAN_SNORT_INTERFACE="enp0s3"
DEBIAN_SNORT_SEND_STATS="true"
DEBIAN_SNORT_STATS_RCPT="root"
DEBIAN_SNORT_STATS_THRESHOLD="1"
~
~
```

- /etc/snort/snort.conf

```
#
ipvar HOME_NET 10.21.1.43/32

# Set up the external network add
ipvar EXTERNAL_NET any
```

TEST

- **로컬 룰 설정** : /etc/snort/rules/local.rules
  - icmp **탐지**

```
# additions here.
#

alert icmp any any -> any any ( msg :"icmp detected"; sid:1000001;)
~
```

snort **실행**

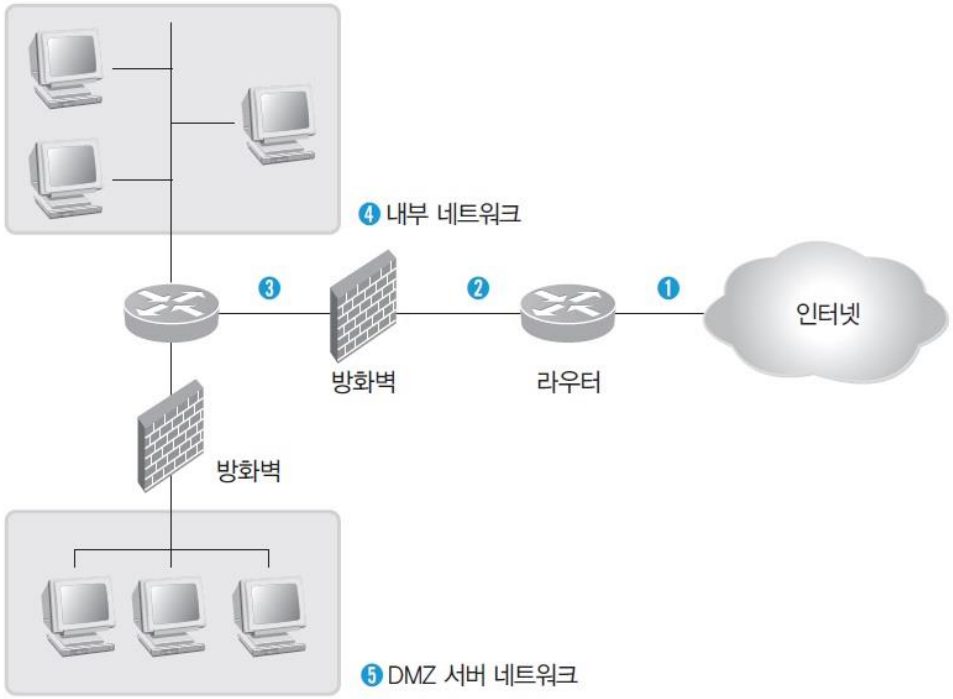
- PC**에서** ping

```
C:₩Users₩Sean>ping 10.21.1.43

Ping 10.21.1.43 32바이트 데이터 사용:
10.21.1.43의 응답: 바이트=32 시간<1ms TTL=64
10.21.1.43의 응답: 바이트=32 시간<1ms TTL=64
10.21.1.43의 응답: 바이트=32 시간<1ms TTL=64
10.21.1.43의 응답: 바이트=32 시간<1ms TTL=64
```

- **중지** : ctrl + c
  - **안되면** ctrl + z **후**, kill -9 pid

IDS의 효과적인 설치 위치



① 패킷이 라우터로 들어오기 전(①)
네트워크에 실행되는 모든 공격 탐지 가능
너무 많은 공격에 대한 데이터를 수집하여 정작 치명적인 공격에는 대처가 어려움.

② 라우터 뒤(②)
①보다 좀더 적은 수의 공격을 탐지하며, 좀더 강력한 의지가 있는 공격자 탐지 가능

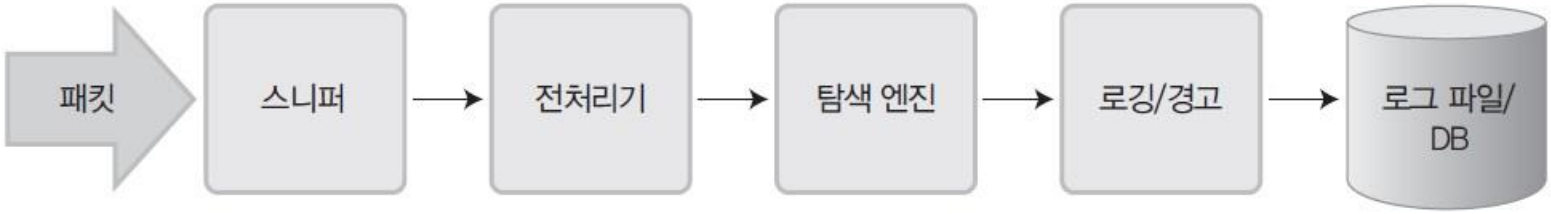
③ 방화벽 뒤(③)
내부 공격자 어느 정도 탐지 가능
만약 침입 탐지 시스템을 한 대만 설치할 수 있다면 이곳에 설치해야 함.

④ 내부 네트워크(④)
내부 클라이언트에 의한 내부 네트워크 해킹을 감시할 때 설치

⑤ DMZ(⑤)
외부와 내부의 공격자에 의한 데이터 손실이나 서비스 중단을 막기 위해 설치

SNORT 구조

구성



rule id

번호	의미
0 ~ 99	고유 목적을 위해 예약된 sid
100 ~ 1,000,000	Snort.org에서 공식 배포한 Rule의 sid
1,000,000 ~	사용자에 의해 작성된 Rule의 sid

SNORT RULE 체계

- snort.conf + local.rules
- # ls /etc/snort/rules
- **중복 제거** : /etc/snort/snort.conf **에서** comment out

```
include $RULE_PATH/ftp.rules
#include $RULE_PATH/icmp-info.rules
#include $RULE_PATH/icmp.rules
include $RULE_PATH/imap.rules
```

- **또는**, local.rules **만 실행**

```
snort -q -A console -b -c /etc/snort/rules/local.rules
```

SNORT RULE

```
# This file intentionally does not come with signatures.  Put your loc
# additions here.
#

alert icmp any any -> any any ( msg :"icmp detected"; sid:1000001;)
~
```

Rule 헤더							Rule 옵션
처리 방법	프로토콜	송신자 IP	송신자 포트	패킷 방향	수신자 IP	수신자 포트	
1	2	3	4	5	6	7	8

번호	의미	예
1	처리 방법	alert, log, pass, activate, dynamic
2	프로토콜	TCP, UDP, ICMP, IP
3	송신자 IP	any, 192.168.0.10, 172.16.0.0/16 등
4	송신자 포트	any, 22, 25, 80, 8080 등
5	패킷 방향	–〉, 〈〉
6	수신자 IP	3과 동일
7	수신자 포트	4와 동일
8	옵션	content, msg, sid 등

RULE 변경

- **방향** : **들어오는** request **만 탐지**

```
alert icmp any any -> 10.21.1.43/32 any ( msg :"icmp request"; sid:1000001;)
```

- Action **유형**

명령어	내용
alert	경고 발생 및 로그 기록
log	로그 기록
pass	패켓 무시
drop	패켓 차단 및 로그 기록 (IPS 기능으로 사용됨, 단 인라인 구조가 되어야 한다.)
reject	패켓 차단 및 로그 기록(TCP - TCP RST 응답, UDP - ICMP Unreachable 응답)
sdrop	패켓 차단 및 로그 기록 없음

  - reject : console**에서는** A **옵션으로** alert **만 함**
  - IPS (intrusion prevention system) : IDS + firewall

```
reject icmp any any -> 10.21.1.43/32 any ( msg :"icmp request"; sid:1000001;)
```

RULES

- http request outbound

```
alert tcp 10.21.1.43/32 any -> any 80 ( msg :"http request"; sid:1000002;  )
```

  - **브라우저 띄워보기**

- ftp login inbound
  - ftp **데몬 설치**

$ sudo apt install vsftpd

$ sudo systemctl start vsftpd

$ sudo systemctl enable vsftpd

  - **룰 설정**

```
alert tcp any any -> 192.168.121.128/32 21 ( msg: "FTP"; content:"root"; sid:1000003;)
```

  - **윈도우에서** ftp **실행**

```
C:\Users\Sean>ftp 192.168.121.128
192.168.121.128에 연결되었습니다.
220 (vsFTPd 3.0.3)
200 Always in UTF8 mode.
사용자(192.168.121.128:(none)): root
331 Please specify the password.
암호:
```

과제 1

- FTP content "**자기이니셜" 탐지 화면 캡처** :1.jpg

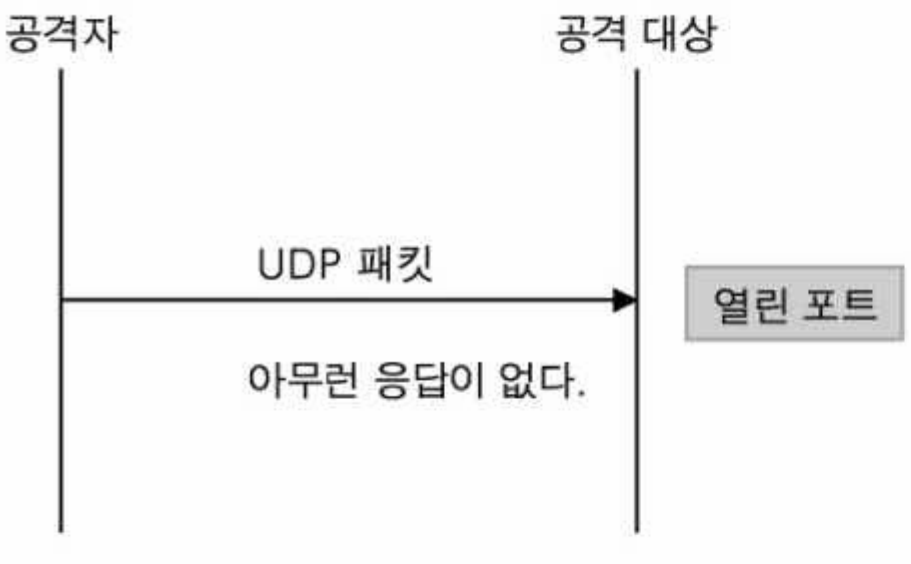
스캔

스캔

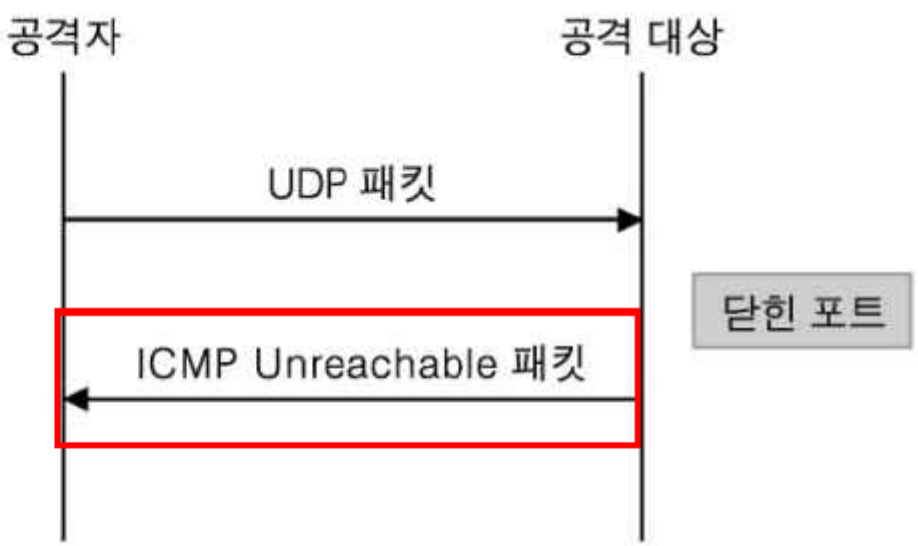
- 스캔은 서비스를 제공하는 서버의 작동여부와 제공하고 있는 서비스를 확인
- TCP 기반의 프로토콜 질의(Request) 응답(Response) 메커니즘
- 열려있는 포트, 제공하는 서비스, 동작중인 데몬의 버전, 운영체제의 버전, 취약점 등 다양한 정보 획득가능
- 일반적으로nmap 사용

Ping & ICMP Scan

UDP 스캔

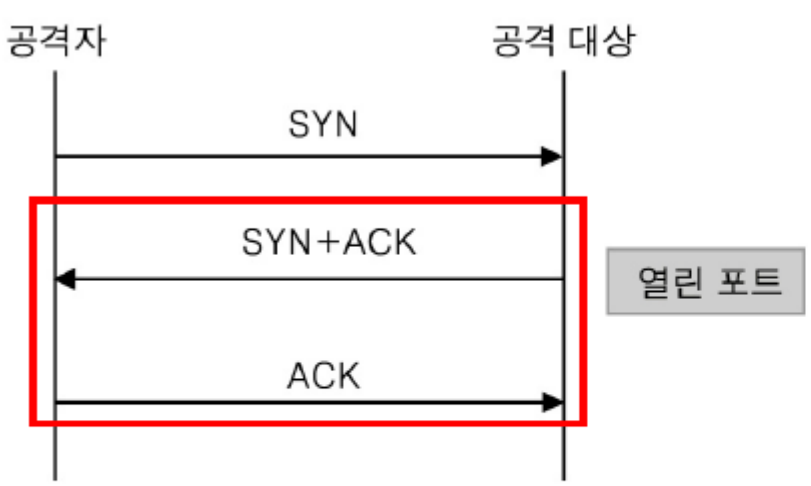


포트가 열려 있을 경우

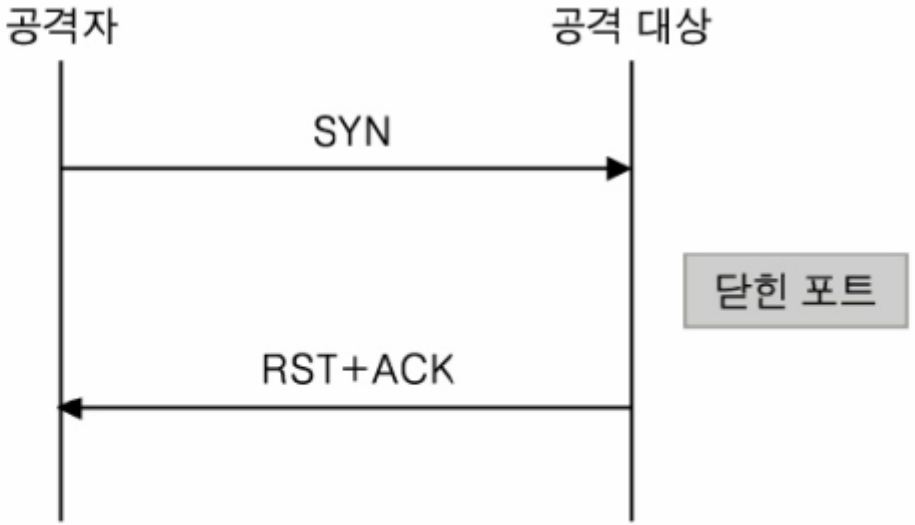


포트가 닫혀 있을 경우

TCP 스캔

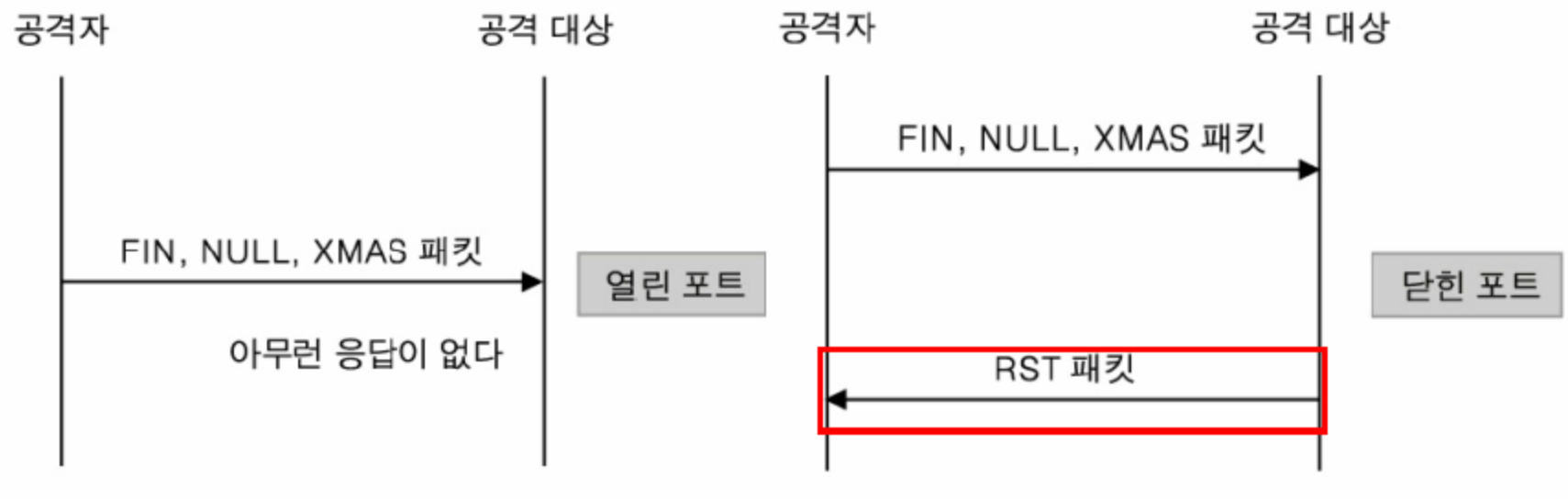


포트가 열려 있을 경우



포트가 닫혀 있을 경우

STEALTH 스캔 : FIN, XMAX, NULL 스캔



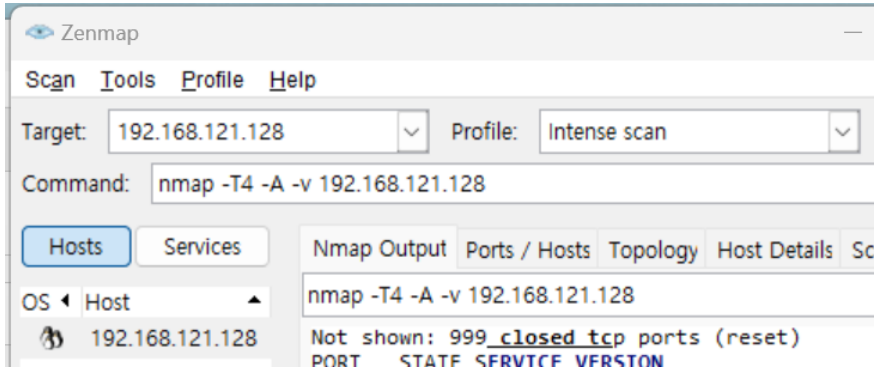
포트가 열려 있을 경우

포트가 닫혀 있을 경우

※ XMAS = ACK,FIN,RST,SYN, URG의 패킷 묶음

포트 스캐닝 탐지

- **스캐너 nmap 윈도우에 설치**
  - http://nmap.org/download.html **설치**
- /etc/snort/snort.conf rule **확인**




```
# Portscan detection.  For more information, see README.sfportscan
preprocessor sfportscan: proto  { all } memcap { 10000000 } sense_level { low }
```

- **결과**

```
root@sean-VirtualBox:~# snort -q -A console -b -c /etc/snort/snort.conf
11/29-23:37:22.949568  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempte
d Information Leak] [Priority: 2] {TCP} 192.168.56.104:64566 -> 192.168.56.101:161
11/29-23:37:22.960476  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: A
ttempted Information Leak] [Priority: 2] {TCP} 192.168.56.104:64566 -> 192.168.56.101:7
05
11/29-23:37:23.886439  [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted I
nformation Leak] [Priority: 2] {TCP} 192.168.56.104:40967 -> 192.168.56.101:1
11/29-23:37:26.033506  [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted I
nformation Leak] [Priority: 2] {TCP} 192.168.56.104:40967 -> 192.168.56.101:1
11/29-23:37:28.174986  [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted I
nformation Leak] [Priority: 2] {TCP} 192.168.56.104:40967 -> 192.168.56.101:1
11/29-23:37:31.810474  [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted I
nformation Leak] [Priority: 2] {TCP} 192.168.56.104:40967 -> 192.168.56.101:1
11/29-23:37:33.945535  [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted I
nformation Leak] [Priority: 2] {TCP} 192.168.56.104:40967 -> 192.168.56.101:1
```

과제 2

- scan **탐지 결과 캡처** : 2.jpg