

Homework 7

Due date: July 22 @6PM

100 Points

Dmytro Poliak

Student Number: 100443432

Section: S50

Date: 22.07.2024

Save as PDF then Upload to Moodle

Class Demo: July 22 : 5-6PM (0 if no Demo)
Cedar Room 2075

WRITE BELOW ALL KALI LINUX COMMANDS YOU USED TO INSTALL AND CONFIGURE SNORT V3 (10 POINTS)

- `sudo -s`
- `apt update && apt full-upgrade -y`
- `mv /etc/apt/sources.list /etc/apt/sources.list.bak`
- `find /var/lib/apt/lists -type f -exec rm {} \;`
- `wget https://gist.githubusercontent.com/ishad0w/788555191c7037e249a439542c53e170/raw/3822ba49241e6fd851ca1c1cbcc4d7e87382f484/sources.list -O /etc/apt/sources.list`
- `apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 3B4FE6ACC0B21F32`
- `apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 871920D1991BC93C`
- `apt update && apt full-upgrade -y`
- `apt install snort`
- `find /var/lib/apt/lists -type f -exec rm {} \;`

WRITE BELOW ALL KALI LINUX COMMANDS YOU USED TO INSTALL AND CONFIGURE SNORT V3 (10 POINTS)

- `mv /etc/apt/sources.list /etc/apt/ubuntu_sources.list`
- `mv /etc/apt/sources.list.bak /etc/apt/sources.list`
- `apt update && apt full-upgrade -y`

Paste a screen shot of running: **snort --version** command

```
(dmytropoliak@kali)-[~]  
$ snort --version  
  
,,_      -*> Snort++ <*-  
o"  )~   Version 3.1.82.0  
'    '   By Martin Roesch & The Snort Team  
         http://snort.org/contact#team  
         Copyright (C) 2014-2024 Cisco and/or its affiliates. All rights reserved.  
         Copyright (C) 1998-2013 Sourcefire, Inc., et al.  
         Using DAQ version 3.0.12  
         Using LuaJIT version 2.1.1700206165  
         Using OpenSSL 3.2.2 4 Jun 2024  
         Using libpcap version 1.10.4 (with TPACKET_V3)  
         Using PCRE version 8.39 2016-06-14  
         Using ZLIB version 1.3.1  
         Using LZMA version 5.6.2  
  
(dmytropoliak@kali)-[~]  
$
```

30 Points

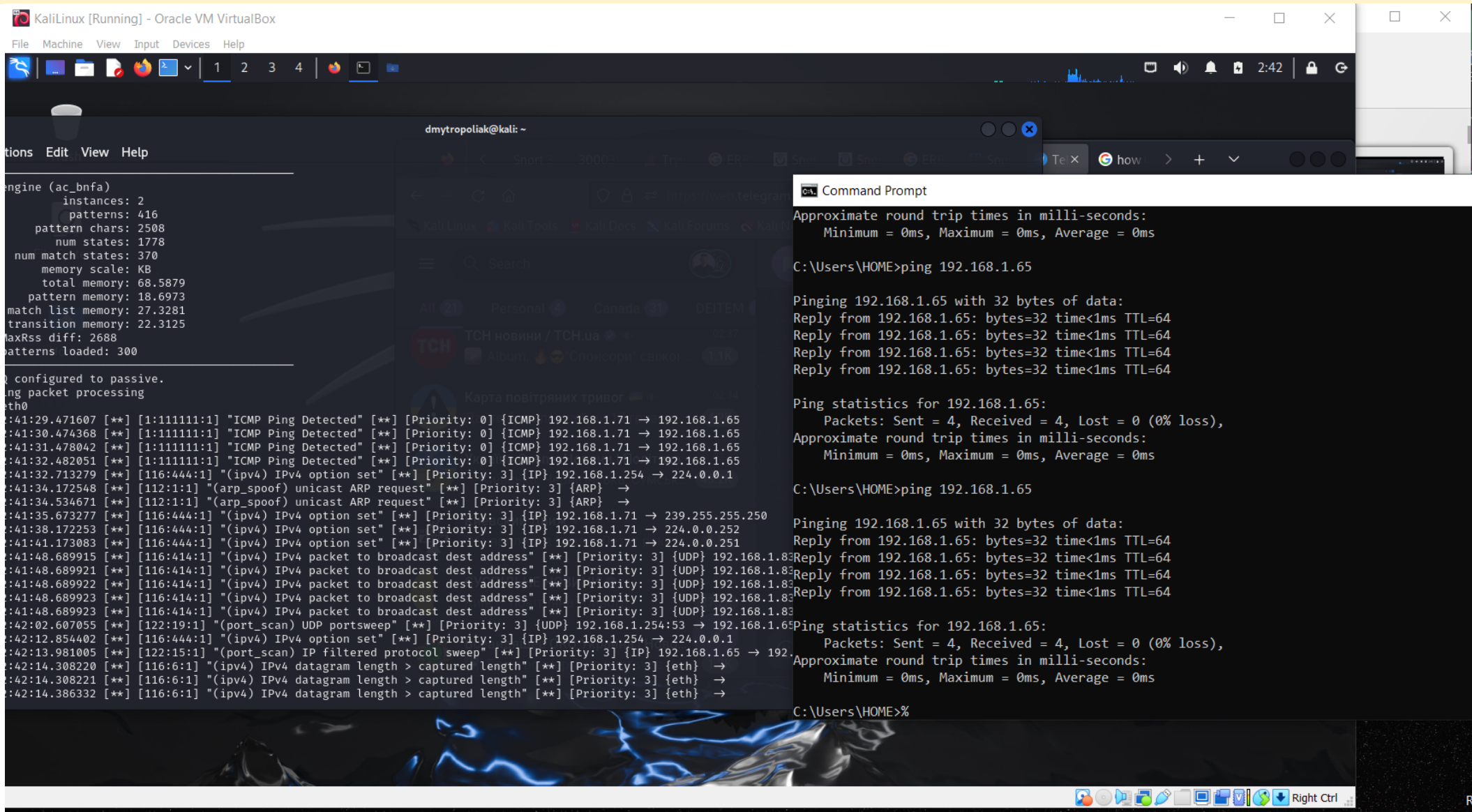
Generate ICMP Requests Alerts

1. stop snort
2. Write the rule
3. Start snort

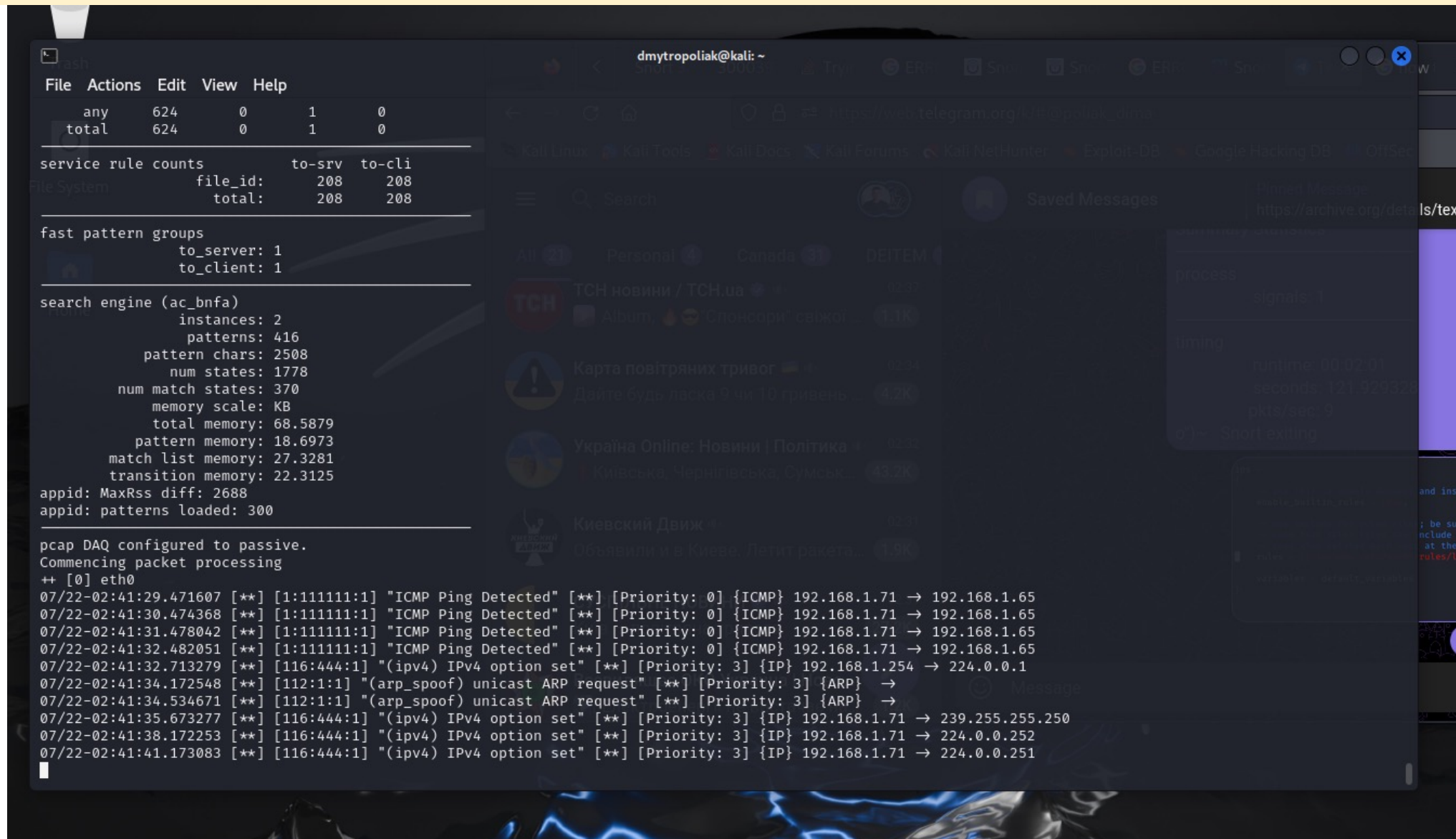
Write a rule to generate an alert when any host pings KALI machine use 111111 as the signature id

```
alert icmp any any -> 192.168.1.65 any (msg:"Ping detected"; itype:8; sid:111111; rev:1;)
```

Show a screenshot of executing the ping command on your Windows/MAC computer



Paste a screenshot that shows the rule is working (only correct if it matches the signature id)



The screenshot displays a Kali Linux desktop environment. On the left, a terminal window shows the output of a rule processing command. The output includes statistics for service rules, fast pattern groups, and search engine details. Below these, it shows the configuration of the pcap DAQ and a list of captured packets with their corresponding rule matches.

```
File Actions Edit View Help
any 624 0 1 0
total 624 0 1 0

service rule counts
file_id: 208
total: 208
to-srv 208
to-cli 208

fast pattern groups
to_server: 1
to_client: 1

search engine (ac_bnfa)
instances: 2
patterns: 416
pattern chars: 2508
num states: 1778
num match states: 370
memory scale: KB
total memory: 68.5879
pattern memory: 18.6973
match list memory: 27.3281
transition memory: 22.3125
appid: MaxRss diff: 2688
appid: patterns loaded: 300

pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
07/22-02:41:29.471607 [**] [1:111111:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65
07/22-02:41:30.474368 [**] [1:111111:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65
07/22-02:41:31.478042 [**] [1:111111:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65
07/22-02:41:32.482051 [**] [1:111111:1] "ICMP Ping Detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65
07/22-02:41:32.713279 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.254 → 224.0.0.1
07/22-02:41:34.172548 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] {ARP} →
07/22-02:41:34.534671 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] {ARP} →
07/22-02:41:35.673277 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.71 → 239.255.255.250
07/22-02:41:38.172253 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.71 → 224.0.0.252
07/22-02:41:41.173083 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.71 → 224.0.0.251
```

On the right, a Telegram web interface is visible, showing a list of messages and a search bar. The interface is in Russian and includes various navigation and search options.

Paste a screenshot that shows the rule is working (only correct if it matches the signature id)

```
transition memory: 23.3281
fast pattern only: 1
appid: MaxRss diff: 2688
appid: patterns loaded: 300

pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
07/22-17:56:23.369293 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65
07/22-17:56:24.374720 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65
07/22-17:56:25.386637 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65
07/22-17:56:26.402282 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65
07/22-17:56:28.171703 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] {ARP} →
07/22-17:56:28.491581 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] {ARP} →
07/22-17:56:33.136190 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.89 → 239.255.255.250
07/22-17:56:35.813267 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 232.239.0.17
07/22-17:56:36.617409 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.82 → 232.239.0.17
07/22-17:56:37.952618 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 → 239.255.255.250
07/22-17:56:38.662584 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.74 → 232.239.0.17
07/22-17:56:39.182939 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:39.182941 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:39.182943 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:40.000133 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:40.000134 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:40.000137 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:40.000138 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:40.000479 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:40.091984 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
07/22-17:56:41.633762 [would_reset] [**] [1:222222:1] "ICMP request rejected" [**] [Priority: 0] {ICMP} 192.168.1.67 → 192.168.1.65
```

25 Points

Reject ICMP Requests

1. stop snort
2. Write the rule
3. Start snort

**Write a rule to reject ICMP requests from any host
use 222222 as the signature id**

```
reject icmp any any -> $HOME_NET any (msg:"ICMP request rejected";  
itype:8; content:"|08|"; sid:222222; rev:1;)
```

Paste a screenshot that shows the rule is working (only correct if it matches the signature id)

```
07/22-17:56:40.000479 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:40.091984 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:41.633762 [would_reset] [**] [1:222222:1] "ICMP request rejected" [**] [Priority: 0] {ICMP} 192.168.1.67 → 192.168.1.65  
07/22-17:56:55.531684 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 232.239.0.17  
07/22-17:56:57.915288 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
```

```
Transition memory: 23.3281  
fast pattern only: 1  
appid: MaxRss diff: 2688  
appid: patterns loaded: 300  
  
pcap DAQ configured to passive.  
Commencing packet processing  
++ [0] eth0  
07/22-17:56:23.369293 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65  
07/22-17:56:24.374720 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65  
07/22-17:56:25.386637 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65  
07/22-17:56:26.402282 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65  
07/22-17:56:28.171703 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] {ARP} →  
07/22-17:56:28.491581 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] {ARP} →  
07/22-17:56:33.136190 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.89 → 239.255.255.250  
07/22-17:56:35.813267 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 232.239.0.17  
07/22-17:56:36.617409 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.82 → 232.239.0.17  
07/22-17:56:37.952618 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 → 239.255.255.250  
07/22-17:56:38.662584 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.74 → 232.239.0.17  
07/22-17:56:39.182939 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:39.182941 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:39.182943 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:40.000133 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:40.000134 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:40.000137 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:40.000138 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:40.000479 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:40.091984 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →  
07/22-17:56:41.633762 [would_reset] [**] [1:222222:1] "ICMP request rejected" [**] [Priority: 0] {ICMP} 192.168.1.67 → 192.168.1.65
```


Paste a screenshot that shows the rule is working (only correct if it matches the signature id)

docs.snort.org/start/rules

Command Prompt

Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\HOME>ping 192.168.1.65

Pinging 192.168.1.65 with 32 bytes of data:

Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.65:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\HOME>ping 192.168.1.65

Pinging 192.168.1.65 with 32 bytes of data:

Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.65:

Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\HOME>

File Actions Edit View Help

total memory: 69.8154

pattern memory: 18.7373

match list memory: 27.375

transition memory: 23.3281

fast pattern only: 1

appid: MaxRss diff: 2688

appid: patterns loaded: 300

pcap DAQ configured to passive.

Commencing packet processing

++ [0] eth0

07/22-17:56:23.369293 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65

07/22-17:56:24.374720 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65

07/22-17:56:25.386637 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65

07/22-17:56:26.402282 [**] [1:111111:1] "Ping detected" [**] [Priority: 0] {ICMP} 192.168.1.71 → 192.168.1.65

07/22-17:56:28.171703 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] {ARP} →

07/22-17:56:28.491581 [**] [112:1:1] "(arp_spoof) unicast ARP request" [**] [Priority: 3] {ARP} →

07/22-17:56:33.136190 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.89 → 239.255.255.250

07/22-17:56:35.813267 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 232.239.0.17

07/22-17:56:36.617409 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.82 → 232.239.0.17

07/22-17:56:37.952618 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 → 239.255.255.250

07/22-17:56:38.662584 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.74 → 232.239.0.17

07/22-17:56:39.182939 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:39.182941 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:39.182943 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:40.000133 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:40.000134 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:40.000137 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:40.000138 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:40.000479 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:40.091984 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:56:41.633762 [would_reset] [**] [1:222222:1] "ICMP request rejected" [**] [Priority: 0] {ICMP} 192.168.1.67 → 192.168.1.65

07/22-17:56:55.531684 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 232.239.0.17

07/22-17:56:57.915288 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →

07/22-17:57:01.601854 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 232.239.0.17

07/22-17:57:03.754595 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.81 → 224.0.0.251

07/22-17:57:04.469524 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 232.239.0.17

07/22-17:57:04.571297 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 → 224.0.0.251

07/22-17:57:11.535240 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 239.255.255.250

Writing Snort Rules

The Basics

Rule Headers

Rule Actions

Protocols

IP Addresses

If users want to include rules from other files

ips =

{

rules = [

include

include

...

]

}

Activate Windows

10 Points

SSH – Access

1. No Rule Writing

Paste a screenshot: Accessing KALI from windows/Mac command prompt using ssh before writing the ssh rule

```
C:\Users\HOME>ssh dmytropoliak@192.168.1.65
The authenticity of host '192.168.1.65 (192.168.1.65)' can't be established.
ECDSA key fingerprint is SHA256:cKQrCdKetI0RmINFxFhn1o6LhZwXiW/mnqXB6mR9KQU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.65' (ECDSA) to the list of known hosts.
dmytropoliak@192.168.1.65's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
(dmytropoliak@kali)~$
```

25 Points

SSH Rejection

1. Stop snort
2. Write the rule
3. Start snort

Write a rule to reject SSH requests from your windows host to KALI : use 333333 as a signature id (enter the rule below)

```
reject tcp any any -> $HOME_NET 22 (msg:"SSH request rejected from Windows host"; sid:333333; rev:1;)
```

Paste a screenshot that shows the rule is working (only correct if it matches the signature id)

```
transition memory: 23.3281
fast pattern only: 1
appid: MaxRss diff: 2688
appid: patterns loaded: 300

pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
07/22-18:33:09.032587 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 → 224.0.0.251
07/22-18:33:09.645053 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.254 → 224.0.0.1
07/22-18:33:09.673571 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.71 → 224.0.0.251
07/22-18:33:09.673575 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.71 → 224.0.0.252
07/22-18:33:09.954051 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.74 → 239.255.255.250
07/22-18:33:10.362513 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.74 → 232.239.0.17
07/22-18:33:17.620279 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.82 → 232.239.0.17
07/22-18:33:18.145188 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 → 239.255.255.250
07/22-18:33:33.627208 [would_reset] [**] [1:333333:1] "SSH request rejected from Windows host" [**] [Priority: 0] {TCP} 192.168.1.71:49785 → 192.168.1.65:22
07/22-18:33:51.604182 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.82 → 239.255.255.250
07/22-18:33:55.355449 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.74 → 232.239.0.17
07/22-18:33:57.147624 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 → 239.255.255.250
07/22-18:33:57.766310 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.80 → 232.239.0.17
07/22-18:33:57.773614 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 → 224.0.0.251
07/22-18:33:59.657552 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.82 → 232.239.0.17
07/22-18:34:12.895616 [**] [122:15:1] "(port_scan) IP filtered protocol sweep" [**] [Priority: 3] {IP} 192.168.1.65 → 192.168.1.254
07/22-18:34:13.339529 [**] [116:6:1] "(ipv4) IPv4 datagram length > captured length" [**] [Priority: 3] {eth} →
```

Paste a screenshot that shows the rule is working (only correct if it matches the signature id)

```
docs.snort.org/start/rules
dmytropoliak@kali: ~
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\HOME>ping 192.168.1.65
Pinging 192.168.1.65 with 32 bytes of data:
  Reply from 192.168.1.65: bytes=32 time<1ms TTL=64
  Reply from 192.168.1.65: bytes=32 time<1ms TTL=64
  Reply from 192.168.1.65: bytes=32 time<1ms TTL=64
  Reply from 192.168.1.65: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.65:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\HOME>ssh dmytropoliak@192.168.1.65
dmytropoliak@192.168.1.65's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jul 22 03:14:21 2024 from 192.168.1.71
dmytropoliak@kali:~$

File Machine View Input Devices Help
service rule counts      to-srv  to-cli
                        file_id  208    208
                        total:   208    208

fast pattern groups
                        any: 2
                        to_server: 1
                        to_client: 1

search engine (ac_bnf)
  instances: 3
  patterns: 417
  pattern chars: 2510
  num states: 1780
  num match states: 371
  memory scale: KB
  total memory: 69.8154
  pattern memory: 18.7373
  match list memory: 27.375
  transition memory: 23.3281
  fast pattern only: 1
  appid: MaxRss diff: 2688
  appid: patterns loaded: 300

pcap DAQ configured to passive.
Commencing packet processing
++ [0] eth0
07/22-18:33:09.032587 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 -> 224.0.0.251
07/22-18:33:09.645053 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.254 -> 224.0.0.1
07/22-18:33:09.673571 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.71 -> 224.0.0.251
07/22-18:33:09.673575 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.71 -> 224.0.0.252
07/22-18:33:09.954051 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.74 -> 239.255.255.250
07/22-18:33:10.362513 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.74 -> 232.239.0.17
07/22-18:33:17.620279 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.82 -> 232.239.0.17
07/22-18:33:18.145188 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.88 -> 239.255.255.250
07/22-18:33:33.627208 [would_reset] [**] [1:333333:1] "SSH request rejected from Windows host" [**] [Priority: 0] {TCP} 192.168.1.71:49785 -> 192.
07/22-18:33:51.604182 [**] [116:444:1] "(ipv4) IPv4 option set" [**] [Priority: 3] {IP} 192.168.1.82 -> 239.255.255.250
```

Paste Windows/Mac screenshot that shows connection is denied

```
C:\Users\HOME>ssh dmytropoliak@192.168.1.65
dmytropoliak@192.168.1.65's password:
Linux kali 6.8.11-amd64 #1 SMP PREEMPT_DYNAMIC Kali 6.8.11-1kali2 (2024-05-30) x86_64

The programs included with the Kali GNU/Linux system are free software;the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Jul 22 03:14:21 2024 from 192.168.1.71
❏(dmytropoliak@ kali)-[~]
❏$
```