Project 5

Task 1: Install OSSEC

OSSEC is one of the well known open-source host-based IDS. It analyzes the system and provides logs to detect misuses of system etc. It helps to detect intrusion to the system.

Installation

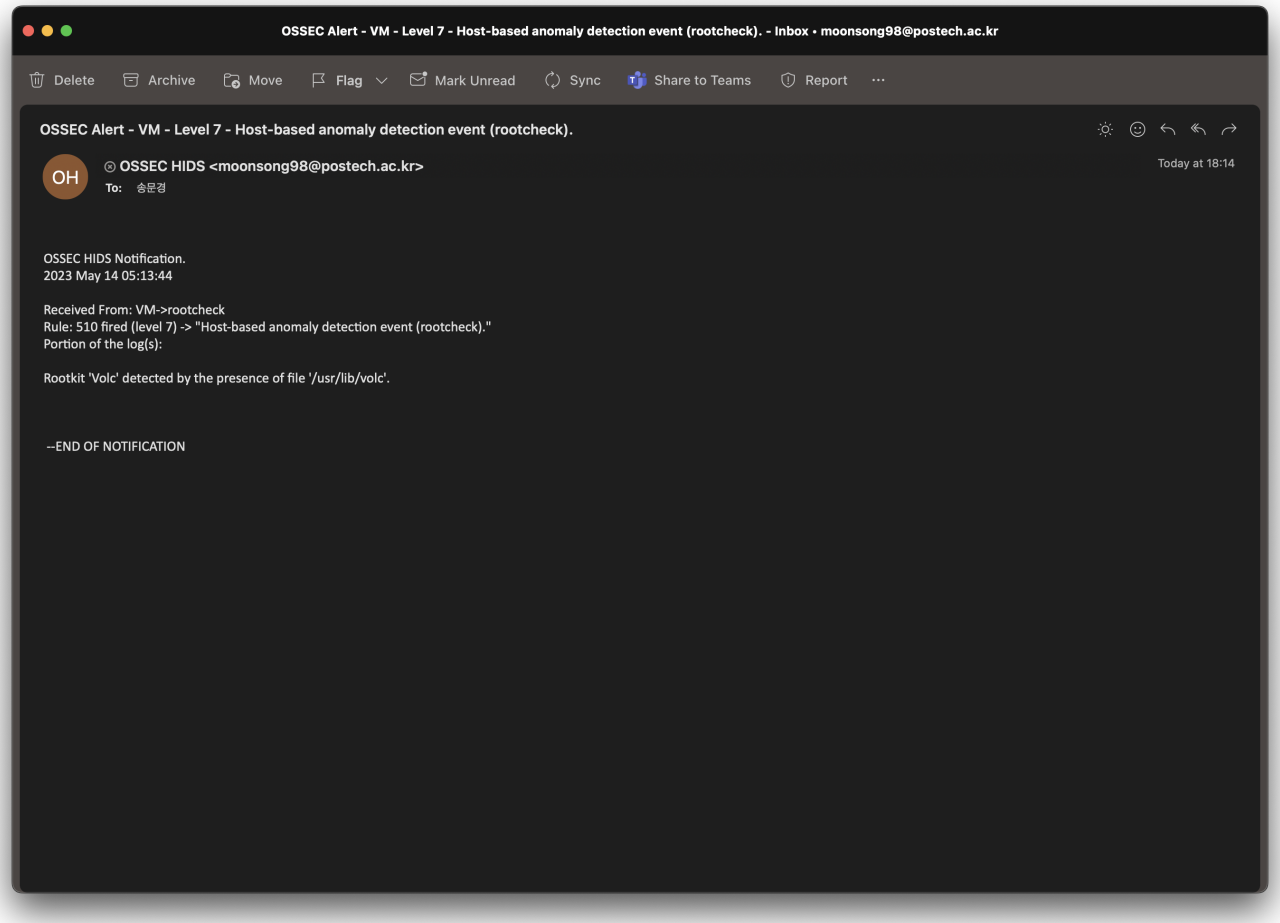
Installed on Ubuntu 20.04 provided by seed lab.

```
 - System is Debian (Ubuntu or derivative).
 - Init script modified to start OSSEC HIDS during boot.

 - Configuration finished properly.

 - To start OSSEC HIDS:
     /var/ossec/bin/ossec-control start

 - To stop OSSEC HIDS:
     /var/ossec/bin/ossec-control stop

 - The configuration can be viewed or modified at /var/ossec/etc/ossec.conf


    Thanks for using the OSSEC HIDS.
    If you have any question, suggestion or if you find any bug,
    contact us at contact@ossec.net or using our public maillist at
    ossec-list@ossec.net
    ( http://www.ossec.net/main/support/ ).

    More information can be found at http://www.ossec.net

    ---  Press ENTER to finish (maybe more information below). ---


[05/13/23]seed@VM:~/ossec-hids-2.9.0$ /var/ossec/bin/ossec-control start
-bash: /var/ossec/bin/ossec-control: Permission denied
[05/13/23]seed@VM:~/ossec-hids-2.9.0$ sudo /var/ossec/bin/ossec-control start
Starting OSSEC HIDS v2.9.0 (by Trend Micro Inc.)...
Started ossec-maild...
Started ossec-execd...
Started ossec-analysisd...
Started ossec-logcollector...
Started ossec-syscheckd...
Started ossec-monitord...
Completed.
```

Detection

Signature-based Detection

OSSEC manages well known rootkits in `src/rootcheck/db/rootkit_files.txt` [LINK].
AS shown in that file, volc is one of the well known rootkit.
Since, direct installaion could be endanger my system, I just created a file `/usr/lib/volc`.
Since then, OSSEC detected it and sent me an email.

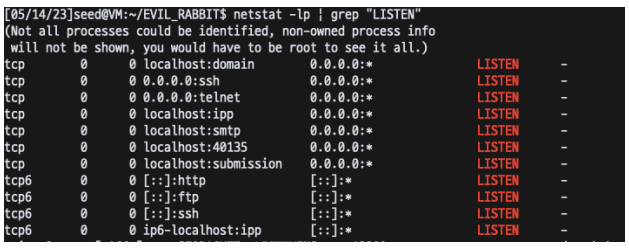


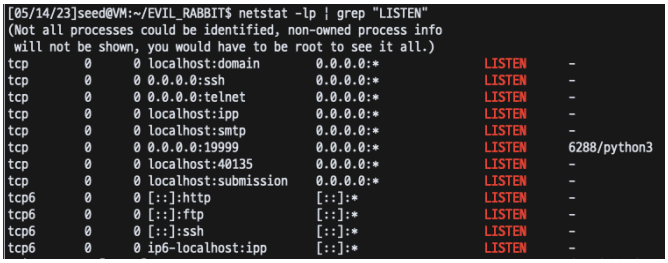
Anomaly-based Detection

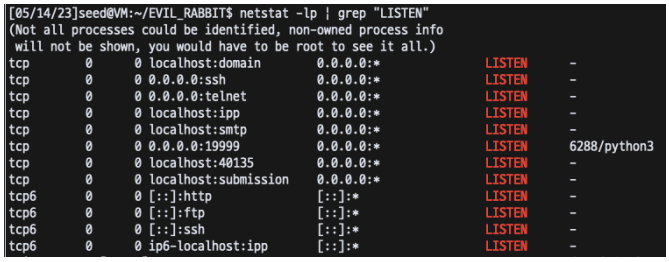
I tested anomaly-based detection by using `EVIL_RABBIT` [LINK].

When it is installed, it behaves as below.

- Conceal itself and in general any file specified on the filesystem (including GNOME file manager - nautilus)
- Posses a payload of TCP bind shell which is activated only if a /tmp directory contains a file named .snow_valley (i.e. /tmp/.snow_valley).



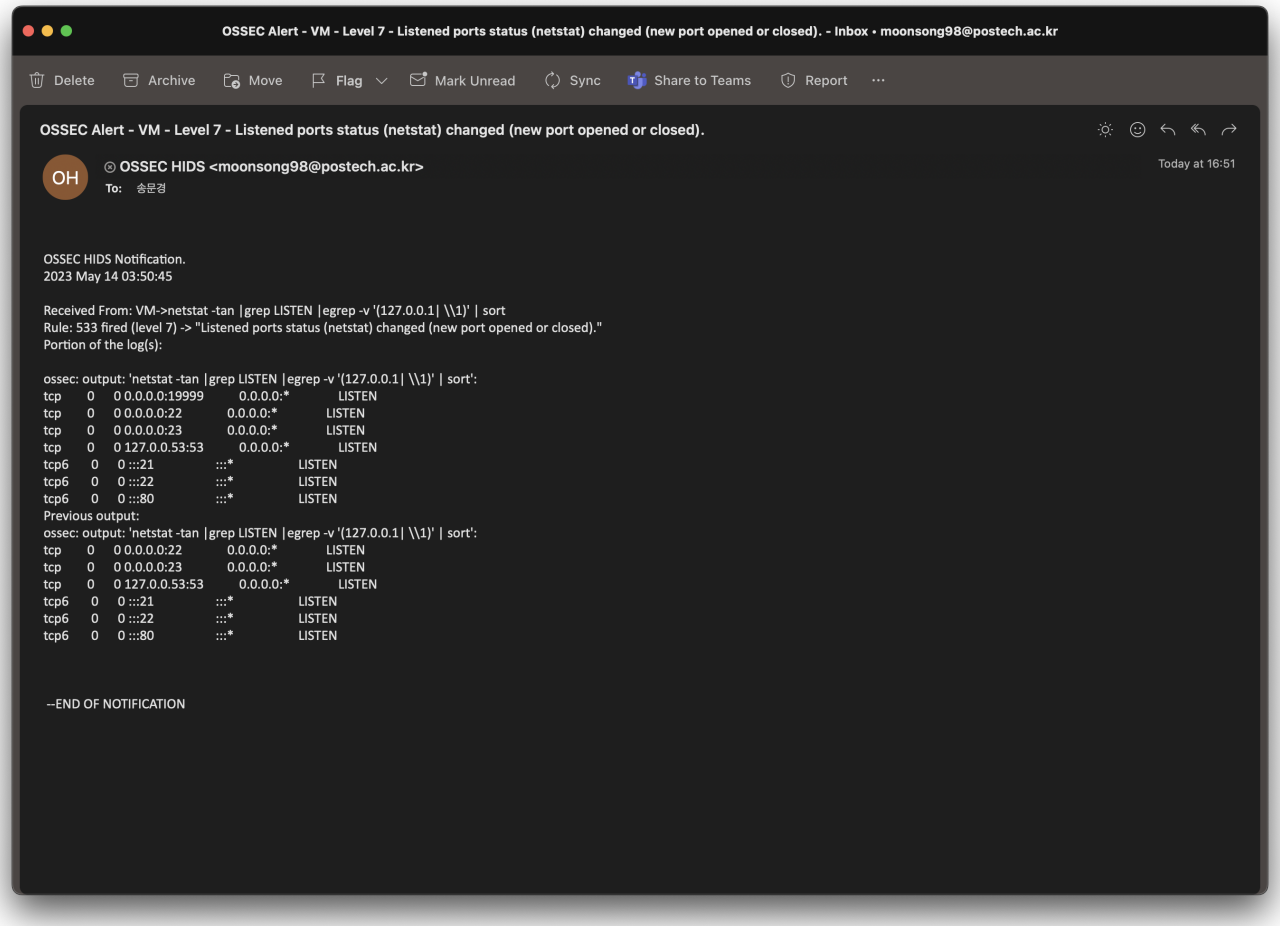




Before Infected After Infected

As shown pics above, new connection is established with port num 19999.

OSSEC detected this new connection and sent me an email.



Task 2: Install Snort

Snort is a packet analysis tool which is used for network traffic analysis.

Installation

Installed on Ubuntu 20.04 provided by seed lab.

```
[05/14/23]seed@VM:~$ snort -V

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11
```

Modes

Sniffer Mode

Following command display the packet data as well as the headers.

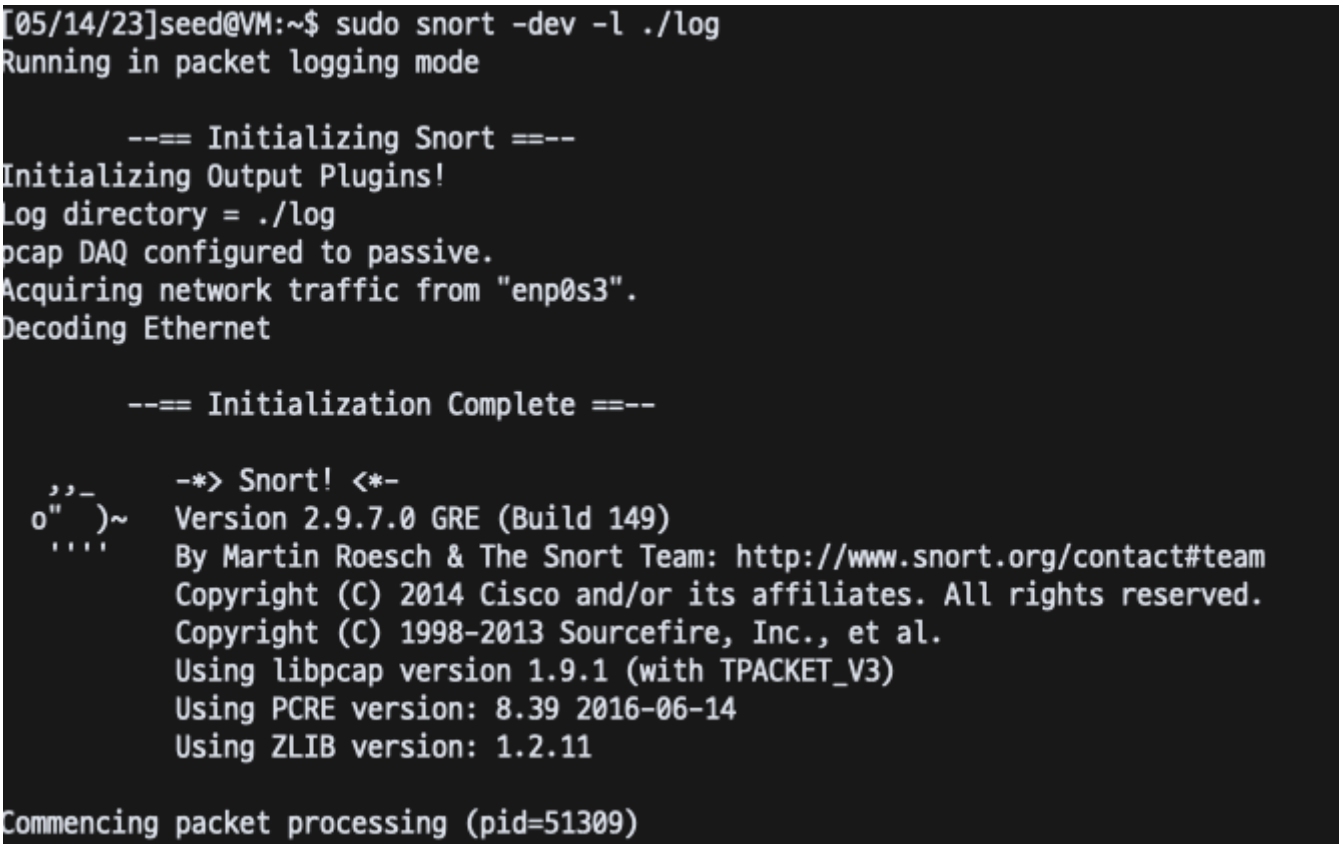
```
sudo snort -vde
```

```
[05/14/23]seed@VM:~$ sudo snort -vde
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "enp0s3".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.7.0 GRE (Build 149)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.9.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=51239)
05/14-13:49:32.130874 08:00:27:57:29:70 -> 52:54:00:12:35:02 type:0x800 len:0x6A
10.0.2.15:22 -> 10.0.2.2:57243 TCP TTL:64 TOS:0x10 ID:60605 IpLen:20 DgmLen:92 DF
***AP*** Seq: 0xA575C06F  Ack: 0x58CF3F  Win: 0xFFFF  TcpLen: 20
F6 04 B8 C8 F8 E4 4C A4 F5 DB E9 A0 09 68 5C A8  ......L......h\.
12 D6 24 0D 1E 21 A4 0D D3 E8 63 23 F6 B2 77 29  ..$..!....c#..w)
63 46 BB 0F BA 8B 3D FD 1A 8A B8 15 BD 79 BF E6  cF....=......y..
55 DA 53 53                                      U.SS

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
05/14-13:49:32.131310 52:54:00:12:35:02 -> 08:00:27:57:29:70 type:0x800 len:0x3C
10.0.2.2:57243 -> 10.0.2.15:22 TCP TTL:64 TOS:0x0 ID:40249 IpLen:20 DgmLen:40
***A**** Seq: 0x58CF3F  Ack: 0xA575C0A3  Win: 0xFFFF  TcpLen: 20

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

05/14-13:49:32.131855 08:00:27:57:29:70 -> 52:54:00:12:35:02 type:0x800 len:0x62
10.0.2.15:22 -> 10.0.2.2:57243 TCP TTL:64 TOS:0x10 ID:60606 IpLen:20 DgmLen:84 DF
***AP*** Seq: 0xA575C0A3  Ack: 0x58CF3F  Win: 0xFFFF  TcpLen: 20
F3 53 73 70 61 D2 1A 66 0B BA 4A 8B 46 0E 47 9E  .Sspa..f..J.F.G.
80 09 A0 60 71 79 8D 75 84 02 27 6F 77 C9 C5 1E  ...`qy.u..'ow...
47 27 6B 67 DB 78 48 EF FE E2 5D 7F              G'kg.xH...].

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Packet Logger Mode

Following command record the packets to the disk, if logging directory is specified.
Snort will automatically know to go into packet logger mode

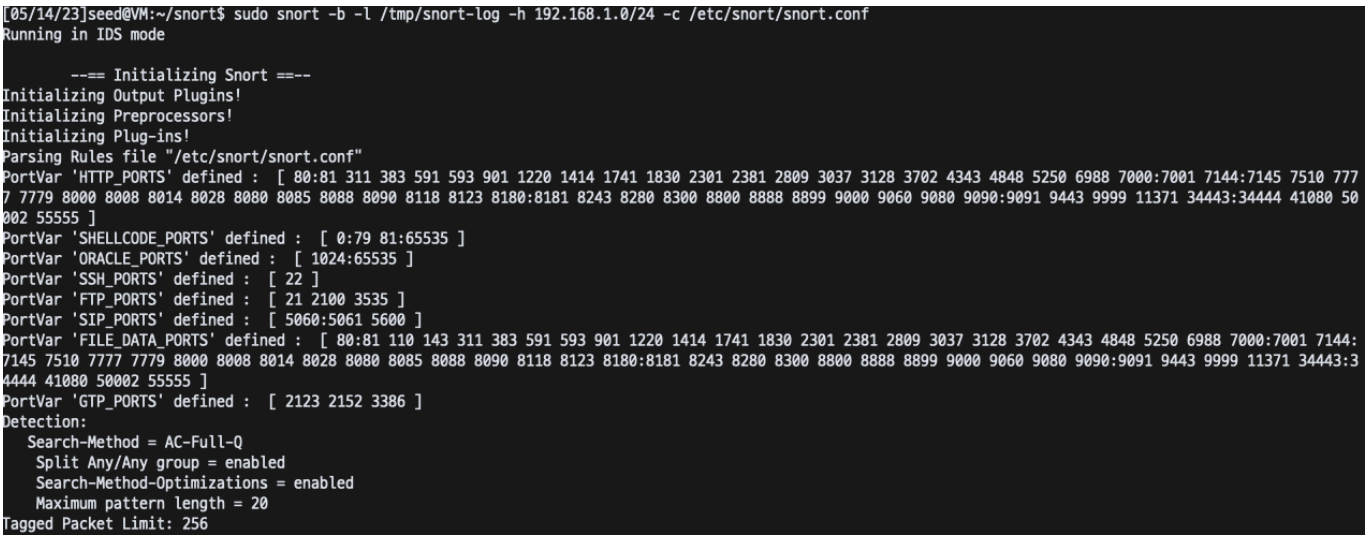
```
sudo snort -dev -l ./log
```



NIDS Mode

Following command enable NIDS mode so that each packet would be recorded by rules given by rules file after -c option.

```
sudo snort -b -l /tmp/snort-log -h 192.168.1.0/24 -c /etc/snort/snort.conf
```



Detection

Signature-based Detection

I tested signature-based detection via DDOS attack. Direct attack would endanger my system, so I alternatively sent packet whose packet is 15104.
In ddos.rules, it treats packet whose port is from 15104 as DDOS atack. I sent a packet via command below

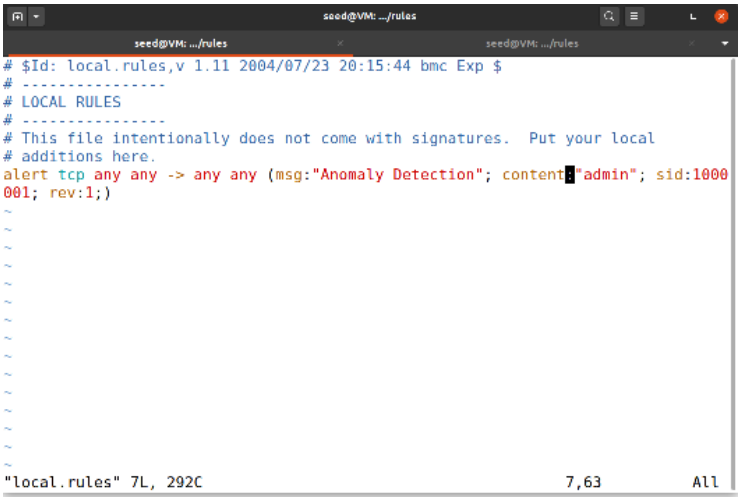
```
nc 172.30.1.76 15104
```

Then snort detects it as a DDOS attack.

```
[05/14/23]seed@VM:~$ sudo snort -q -A console -b -c /etc/snort/snort.conf
05/14-14:34:31.640889  [**] [1:249:8] DDOS mstream client to handler [**] [Class
ification: Attempted Denial of Service] [Priority: 2] {TCP} 172.30.1.19:58749 ->
_172.30.1.76:15104
```

Anomaly-based Detection

To test anomaly based detection, I created a local rule at `/etc/snort/rules/local.rules`.
When a TCP packet which contains "admin" arrives, it treats as a intrusion.
Below pics shows the result.



Create a Local Rule | Send "Admin"

```
[05/14/23]seed@VM:.../rules$ sudo snort -q -A console -b -c /etc/snort/snort.con
f
05/14-15:07:46.109847  [**] [1:1000001:1] Anomaly Detection [**] [Priority: 0] {
TCP} 172.30.1.19:59023 -> 172.30.1.76:11111
```

Anomaly Detected