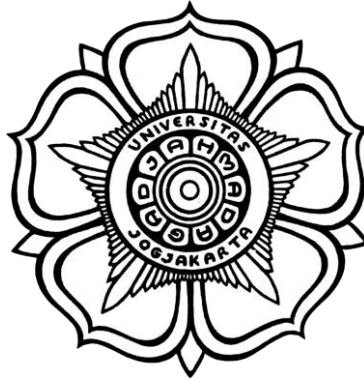


LAPORAN PRAKTIKUM KEAMANAN INFORMASI 1

Pertemuan 6 – *Snort* dan *Firewall Rules*



DISUSUN OLEH

Nama : Sofiyanatul Munawaroh
NIM : 21/474781/SV/19035
Hari, Tanggal : Selasa, 21 Maret 2023
Kelas : RI4AA

**LABORATORIUM PERANGKAT KERAS DAN LUNAK
PROGRAM SARJANA TERAPAN (DIV) TEKNOLOGI
REKAYASA INTERNET
DEPARTEMEN TEKNIK ELEKTRO DAN INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
2023**

Praktikum Keamanan Informasi 1

Pertemuan 6 – *Snort dan Firewall Rules*

I. Tujuan

- Mempersiapkan lingkungan virtual.
- *Firewall* dan *Log IDS*.
- Hentikan dan hapus proses *mininet*.

II. Latar Belakang

Dalam topologi jaringan yang aman, peringatan jaringan dihasilkan oleh berbagai jenis perangkat seperti peralatan keamanan, *firewall*, perangkat IPS, *router*, *switch*, *server*, dan lain-lain. Permasalahan terdapat pada tidak semua peringatan dibuat sama. Misalnya, peringatan yang dihasilkan oleh *server* dan peringatan yang dihasilkan oleh *firewall* akan berbeda dan bervariasi dalam konten dan format.

Pengertian *firewall* sendiri adalah sebagai pencegah berbagai macam hal yang mampu menyebabkan kerusakan pada komputer. *Firewall* merupakan bentuk pengamanan ketika mengakses informasi, terutama yang sifatnya publik (*non private*) dan rentan menjadi target utama para *hacker* ketika komputer tidak disematkan *firewall*.

Cara kerja *firewall* untuk mengamankan jaringan komputer terbagi menjadi tiga metode. Pertama, cara kerja *firewall* adalah data akan diakses terlebih dahulu melalui proses penyaringan (*filter*). Cara kerja *firewall* yang kedua menggunakan layanan *proxy* dapat diterapkan untuk memudahkan proses pendistribusian dari internet ke sistem pusat, begitu pula sebaliknya. Metode atau cara ketiga yaitu inspeksi mengizinkan sistem melakukan proses penyesuaian data yang sudah dianggap aman untuk didistribusikan. Ketiga metode ini dapat dijalankan satu-satu ataupun dikombinasikan satu sama lain.

Dari segi lalu lintas pada sebuah jaringan, terdapat *Snort* sebagai pendeteksi. *Snort* merupakan aplikasi yang ampuh dalam sistem pencegahan dan mampu melakukan *real-time* analisis lalu lintas serta paket *logging* pada jaringan IP. *Snort* adalah program khusus dalam Deteksi Penyusupan (*Intrusion Detection*).

Dalam mengoperasikan *Snort* terdapat tiga mode. Mode pertama adalah *sniffer mode* untuk melihat paket yang lewat di jaringan. Kedua yaitu *packet logger mode* untuk mencatat semua paket yang lewat di jaringan untuk dianalisa di kemudian hari. Selanjutnya *intrusion detection mode* untuk mendeteksi serangan yang dilakukan melalui jaringan komputer.

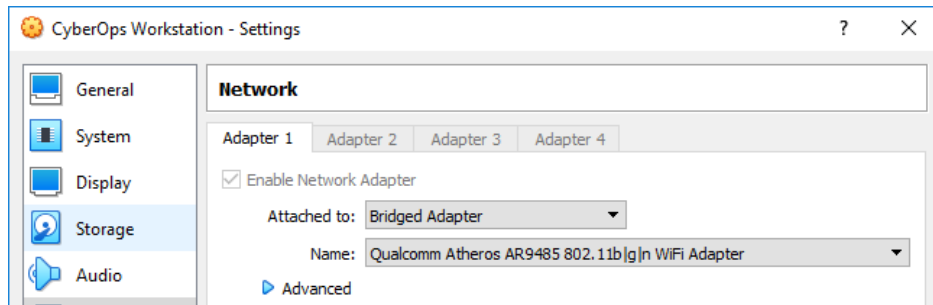
III. Alat dan Bahan

- *Cyberops Workstation Virtual Machine*
- Laptop
- Koneksi Internet

IV. Instruksi Kerja

A. Mempersiapkan *Virtual Machine*

1. Jalankan VM *CyberOps Workstation*. Ubah mode koneksi menjadi *bridged adapter* jika jaringan WiFi tidak menggunakan *proxy* atau NAT jika jaringan WiFi menggunakan *proxy*.



Karena menggunakan jaringan *hotspot* pribadi, maka gunakan mode *Bridged Adapter*.

2. Konfigurasi agar mendapatkan IP secara otomatis, lakukan konfigurasi DHCP.

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help

[analyst@secOps ~]$ sudo ./lab.support.files/scripts/configure_as_dhcp.sh
[sudo] password for analyst:
Configuring the NIC to request IP info via DHCP...
Requesting IP information...
IP Configuration successful.

[analyst@secOps ~]$
```

Cek IP dengan perintah *ifconfig*.

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help

[analyst@secOps ~]$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.25.20 netmask 255.255.255.0 broadcast 192.168.25.255
    inet6 fe80::a00:27ff:fe1a:19b2 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:1a:19:b2 txqueuelen 1000 (Ethernet)
    RX packets 7512 bytes 10321344 (9.8 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2048 bytes 168678 (164.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
```

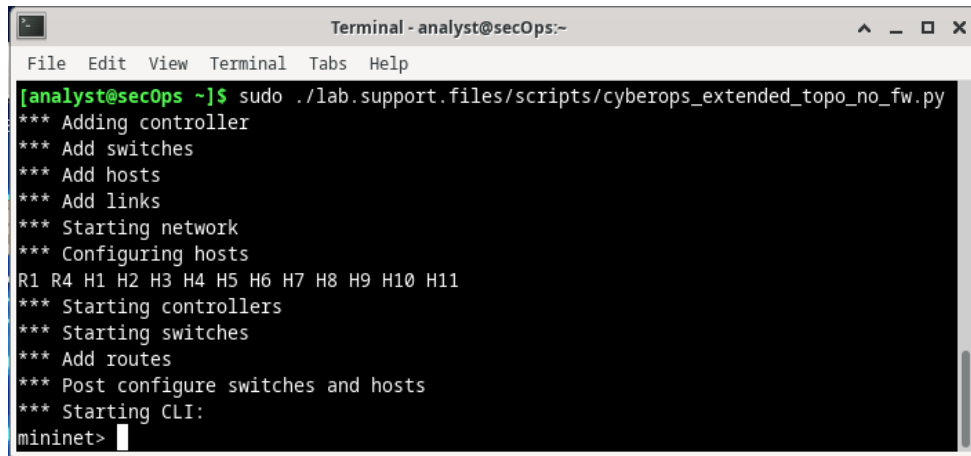
Cek koneksi PING ke *webserver public* seperti www.cisco.com.

```
Terminal - analyst@secOps:~
File Edit View Terminal Tabs Help

[analyst@secOps ~]$ ping www.cisco.com
PING e2867.dsca.akamaiedge.net (104.93.203.251) 56(84) bytes of data.
64 bytes from 104.93.203.251: icmp_seq=1 ttl=55 time=26.8 ms
64 bytes from 104.93.203.251: icmp_seq=2 ttl=55 time=584 ms
64 bytes from 104.93.203.251: icmp_seq=3 ttl=55 time=23.9 ms
64 bytes from 104.93.203.251: icmp_seq=4 ttl=55 time=46.3 ms
^C
--- e2867.dsca.akamaiedge.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 12039ms
rtt min/avg/max/mdev = 23.940/170.250/583.986/239.024 ms
[analyst@secOps ~]$
```

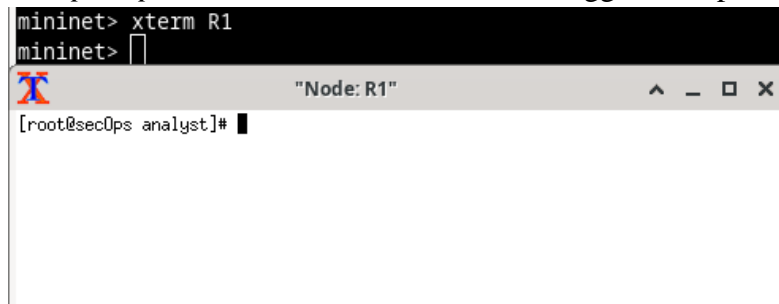
B. Firewall & IDS Logs

1. Dari VM *CyberOps Workstation*, jalankan skrip untuk memulai **mininet**.



```
Terminal - analyst@secOps:-
File Edit View Terminal Tabs Help
[analyst@secOps ~]$ sudo ./lab.support.files/scripts/cyberops_extended_topo_no_fw.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
R1 R4 H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11
*** Starting controllers
*** Starting switches
*** Add routes
*** Post configure switches and hosts
*** Starting CLI:
mininet>
```

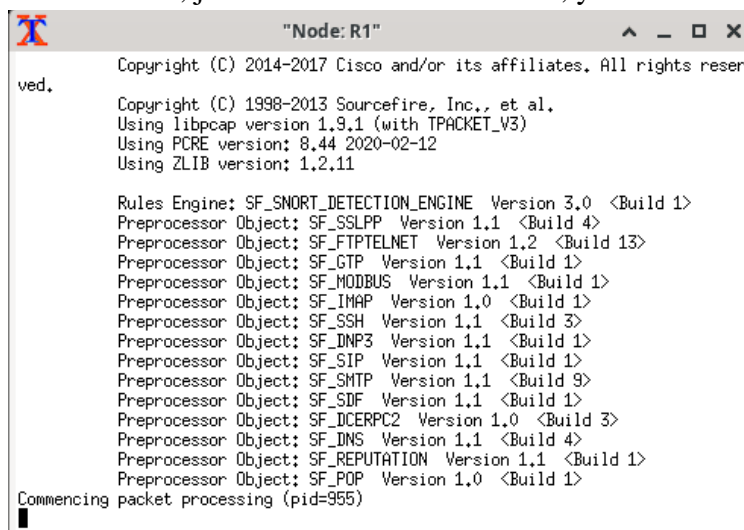
2. Dari *prompt mininet*, buka *shell* dari R1 menggunakan perintah *xterm R1*.



```
mininet> xterm R1
mininet>
"Node: R1"
[root@secOps analyst]#
```

Maka akan muncul jendela *Shell R1* yang masuk sebagai *Super User* dengan indikator *root* pada *username* yang digunakan.

3. Dari *shell R1*, jalankan IDS berbasis Linux, yaitu *Snort*.

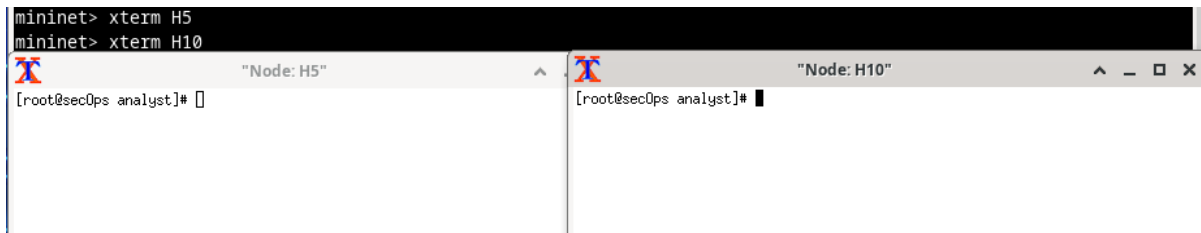


```
"Node: R1"
Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using libpcap version 1.9.1 (with TPACKET_V3)
Using PCRE version: 8.44 2020-02-12
Using ZLIB version: 1.2.11

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.0 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Commencing packet processing (pid=955)
```

4. Dari *prompt mininet CyberOps Workstation VM*, buka *shell* untuk *host H5* dan *H10*.

```
mininet> xterm H5
mininet> xterm H10
```

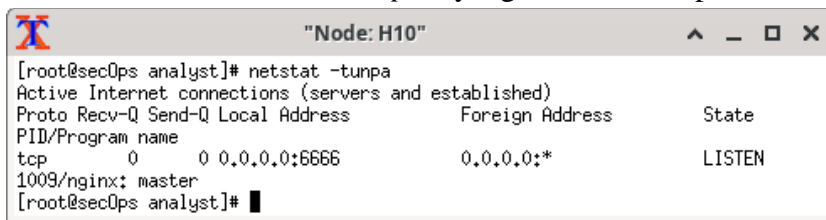


5. H10 akan mensimulasikan *server* di internet yang meng-*hosting malware*. Pada H10, jalankan skrip `mal_server_start.sh` untuk memulai server.



```
[root@secOps analyst]# ./lab.support.files/scripts/mal_server_start.sh
[root@secOps analyst]#
```

6. Pada H10, gunakan *netstat* dengan opsi `-tunpa` untuk memverifikasi bahwa *server web* sedang berjalan. Saat digunakan seperti yang ditunjukkan di bawah ini, *netstat* mencantumkan semua *port* yang saat ini ditetapkan ke layanan.

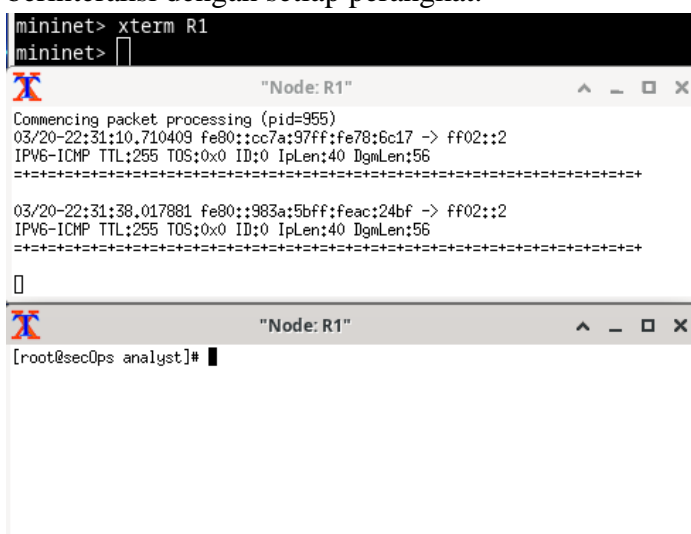


```
[root@secOps analyst]# netstat -tunpa
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:6666             0.0.0.0:*               LISTEN
1009/nginx: master
[root@secOps analyst]#
```

Seperti yang terlihat pada *output* di atas, *nginx server web* ringan sedang berjalan pada koneksi pada *port* TCP 6666.

7. Di jendela terminal R1, sebuah *instance* dari *Snort* sedang berjalan. Untuk memasukkan lebih banyak perintah di R1, buka terminal R1 lain dengan memasukkan `xterm R1` lagi di jendela terminal VM *CyberOps Workstation*. Anda mungkin juga ingin mengatur jendela terminal sehingga dapat melihat dan berinteraksi dengan setiap perangkat.

```
mininet> xterm R1
mininet>
```



8. Di *tab terminal* R1 baru, jalankan perintah *tail* dengan opsi -f untuk memantau *file* /var/log/snort/alert secara *real-time*. *File* ini adalah tempat *snort* dikonfigurasi untuk merekam peringatan.

```
"Node: R1"
[root@secOps analyst]# tail -f /var/log/snort/alert
```

9. Dari H5, gunakan perintah *wget* untuk mengunduh *file* bernama W32.Nimda.Amm.exe. Dirancang untuk mengunduh konten melalui HTTP, *wget* adalah alat yang hebat untuk mengunduh *file* dari *server web* langsung dari baris perintah.

```
"Node: H5"
[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe
--2023-03-20 22:43:54-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... connected.
HTTP request sent, awaiting response... 200 OK
Length: 345088 (337K) [application/octet-stream]
Saving to: 'W32.Nimda.Amm.exe.1'

W32.Nimda.Amm.exe.1 100%[=====>] 337.00K --.-KB/s in 0.01s
2023-03-20 22:43:54 (27.1 MB/s) - 'W32.Nimda.Amm.exe.1' saved [345088/345088]

[root@secOps analyst]#

"Node: R1"
***A*** Seq: 0x38E1B9D2 Ack: 0x552D3E76 Win: 0x36E TcpLen: 32
TCP Options (3) => NOP NOP TS: 3696864482 248353799
=====
03/20-22:43:54.711919 209.165.200.235:51166 -> 209.165.202.133:6666
TCP TTL:63 TOS:0x0 ID:56355 IpLen:20 DgmLen:52 DF
***A*** Seq: 0x38E1B9D2 Ack: 0x552D3E76 Win: 0x36E TcpLen: 32
TCP Options (3) => NOP NOP TS: 3696864487 248353799
=====
03/20-22:43:54.711977 209.165.202.133:6666 -> 209.165.200.235:51166
TCP TTL:63 TOS:0x0 ID:16153 IpLen:20 DgmLen:52 DF
***A*** Seq: 0x552D3E76 Ack: 0x38E1B9D3 Win: 0x55 TcpLen: 32
TCP Options (3) => NOP NOP TS: 248353804 3696864487
=====

"Node: R1"
[root@secOps analyst]# tail -f /var/log/snort/alert
03/20-22:43:54.684470 [**] [1:1000003:0] Malicious Server Hit! [**] [Priority:
0] {TCP} 209.165.200.235:51166 -> 209.165.202.133:6666
```

- *Port* yang digunakan adalah 6666 dengan indikator :6666 setelah IP Address dari *webserver*.
 - *File* diunduh sepenuhnya dengan indikator 100%.
 - IDS memberikan peringatan ditandai dengan *alert* pada jendela *Node R1* yang kedua.
10. Saat *file* berbahaya sedang transit R1, IDS, *Snort*, dapat memeriksa muatannya. *Payload* cocok dengan setidaknya satu tanda tangan yang dikonfigurasi di *Snort* dan memicu peringatan di jendela terminal R1 kedua (*tab* tempat *tail -f* berjalan). Entri peringatan ditunjukkan di bawah ini. Stempel waktu Anda akan berbeda:

```
"Node: R1"
[root@secOps analyst]# tail -f /var/log/snort/alert
03/20-22:43:54.684470  [**] [1:1000003;0] Malicious Server Hit! [**] [Priority:
0] {TCP} 209.165.200.235:51166 -> 209.165.202.133:6666
```

- Alamat Ipv4 sumber dan tujuan yang digunakan dalam transaksi adalah 209.165.200.235 dan 209.165.202.133.
- Berdasarkan *Alert*, *port* sumber dan tujuan yang digunakan dalam transaksi adalah 51166 dan 6666.
- Berdasarkan peringatan yang ditunjukkan, pengunduhan dilakukan pada tanggal 20 Maret 2023 pukul 22:43:54.
- Berdasarkan peringatan yang ditunjukkan pesan yang direkam adalah *Malicious Server Hit*.

11. Pada H5, gunakan perintah tcpdump untuk merekam peristiwa dan mengunduh *file malware* lagi sehingga Anda dapat merekam transaksi. Keluarkan perintah berikut untuk mulai pengambilan paket.

```
"Node: H5"
[root@secOps analyst]# tcpdump -i H5-eth0 -w nimda.download.pcap &
[1] 1088
[root@secOps analyst]# tcpdump: listening on H5-eth0, link-type EN10MB (Ethernet
), capture size 262144 bytes
[root@secOps analyst]#
```

12. Perintah di atas menginstruksikan tcpdump untuk menangkap paket pada antarmuka H5-eth0 dan menyimpan tangkapan ke *file* bernama nimda.download.pcap. Simbol & di bagian akhir memberitahu *shell* untuk mengeksekusi tcpdump di latar belakang. Tanpa symbol ini, tcpdump akan membuat terminal tidak dapat digunakan saat sedang berjalan. Perhatikan [1] 5633; itu menunjukkan suatu proses dikirim ke latar belakang dan ID prosesnya (PID) adalah 5366. PID Anda kemungkinan besar akan berbeda.

13. Tekan ENTER beberapa kali untuk mendapatkan kembali kendali atas *shell* saat tcpdump berjalan di latar belakang.

14. Sekarang tcpdump menangkap paket, unduh *malware* lagi. Pada H5, jalankan kembali perintah atau gunakan panah atas untuk memanggilnya kembali dari fasilitas riwayat perintah.

```
"Node: H5"
[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe
--2023-03-20 22:53:07-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... connected.
HTTP request sent, awaiting response... 200 OK
Length: 345088 (337K) [application/octet-stream]
Saving to: 'W32.Nimda.Amm.exe.2'

W32.Nimda.Amm.exe.2 100%[=====>] 337.00K --.-KB/s in 0.02s

2023-03-20 22:53:07 (20.2 MB/s) - 'W32.Nimda.Amm.exe.2' saved [345088/345088]

[root@secOps analyst]#
```

15. Hentikan pengambilan dengan membawa tcpdump ke latar depan dengan perintah fg. Karena tcpdump adalah satu-satunya proses yang dikirim ke latar belakang, PID tidak perlu ditentukan. Hentikan proses tcpdump dengan Ctrl+C. Proses tcpdump berhenti dan menampilkan ringkasan tangkapan. Jumlah paket mungkin berbeda untuk pengambilan Anda.

```
"Node: H5"
[root@secOps analyst]# fg
tcpdump -i H5-eth0 -w nimda.download.pcap
^C68 packets captured
68 packets received by filter
0 packets dropped by kernel
[root@secOps analyst]#
```

16. Pada H5, Gunakan perintah ls untuk memverifikasi file pcap sebenarnya disimpan ke disk dan memiliki ukuran lebih besar dari nol.

```
"Node: H5"
[root@secOps analyst]# ls -l
total 27712
drwxr-xr-x 2 analyst analyst 4096 May 20 2020 Desktop
drwxr-xr-x 3 analyst analyst 4096 Apr 2 2020 Downloads
-rw-r--r-- 1 root root 12639885 Feb 20 21:05 httpdump.pcap
-rw-r--r-- 1 root root 14323077 Feb 20 21:24 httpsdump.pcap
drwxr-xr-x 9 analyst analyst 4096 Jul 15 2020 lab.support.files
-rw-r--r-- 1 analyst analyst 351038 Mar 20 22:56 nimda.download.pcap
drwxr-xr-x 2 analyst analyst 4096 Mar 21 2018 second_drive
-rw-r--r-- 1 analyst analyst 345088 Mar 15 22:29 W32.Nimda.Amm.exe
-rw-r--r-- 1 root root 345088 Mar 23 2018 W32.Nimda.Amm.exe.1
-rw-r--r-- 1 root root 345088 Mar 23 2018 W32.Nimda.Amm.exe.2
[root@secOps analyst]#
```

C. Menyetel Aturan Firewall Berdasarkan IDS Alerts.

1. Di VM CyberOps Workstation, mulai jendela terminal R1 ketiga.

```
mininet> xterm R1
mininet>
"Node: R1"
[root@secOps analyst]#
```

2. Di jendela terminal R1 baru, gunakan perintah iptables untuk membuat daftar rantai dan aturannya yang sedang digunakan.

```
"Node: R1"
[root@secOps analyst]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
[root@secOps analyst]#
```

Saat ini belum ada chain yang digunakan oleh R1.

3. Koneksi ke server menghasilkan paket yang harus melintasi firewall iptables di R1. Paket yang melintasi firewall ditangani oleh aturan FORWARD dan oleh

karena itu, rantai itulah yang akan menerima aturan pemblokiran. Agar komputer pengguna tidak terhubung ke *server* yang diidentifikasi di Langkah 1, tambahkan aturan berikut ke rantai FORWARD di R1.

```
"Node: R1"
[root@secOps analyst]# iptables -I FORWARD -p tcp -d 209.165.202.133 --dport 6666 -j DROP
[root@secOps analyst]#
```

Di mana:

- -I FORWARD: menyisipkan aturan baru dalam rantai FORWARD.
- -p tcp: menentukan protokol TCP.
- -d 209.165.202.133: menentukan tujuan paket.
- --dport 6666: menentukan *port* tujuan.
- -j DROP: atur aksi ke *drop*.

4. Gunakan perintah *iptables* lagi untuk memastikan aturan telah ditambahkan ke rantai FORWARD. VM *CyberOps Workstation* mungkin memerlukan beberapa detik untuk menghasilkan *output*.

```
"Node: R1"
[root@secOps analyst]# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination
0 0 DROP tcp -- any any anywhere 209.165.202.133 tcp dpt:6666

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target prot opt in out source destination

[root@secOps analyst]#
```

5. Pada H5, coba unduh *file* lagi.

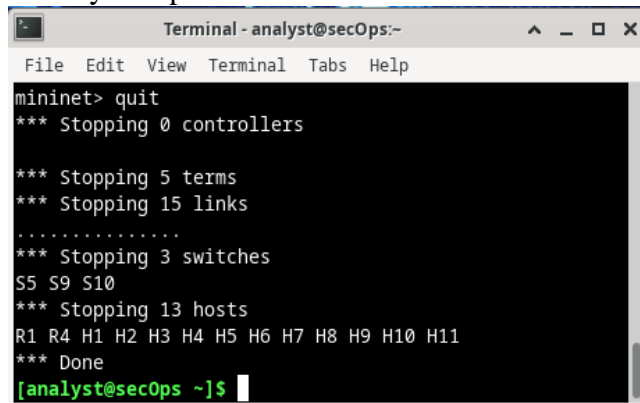
```
"Node: H5"
[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe
--2023-03-20 23:07:37-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... failed: Connection timed out.
Retrying.

--2023-03-20 23:09:47-- (try: 2) http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... failed: Connection timed out.
Retrying.

--2023-03-20 23:11:58-- (try: 3) http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... ^C
[root@secOps analyst]#
```

- Unduhan tidak berhasil karena terdapat *firewall* yang akan melakukan *DROP* paket yang telah ditentukan, dalam hal ini adalah paket dengan tujuan ke 209.165.202.133 dengan *port* 6666.
- Kita dapat melakukan *block* terhadap IP *Server* tujuan. Hal ini dapat sepenuhnya memotong akses ke *server* tujuan dari jaringan internal, sehingga kita tidak perlu menentukan IP, *Port*, maupun protokol dalam sebuah *server*.

6. Hentikan proses *mininet* dengan mengarahkan ke terminal yang digunakan untuk memulai Mininet. Hentikan Mininet dengan memasukkan *quit* di jendela terminal VM CyberOps utama.

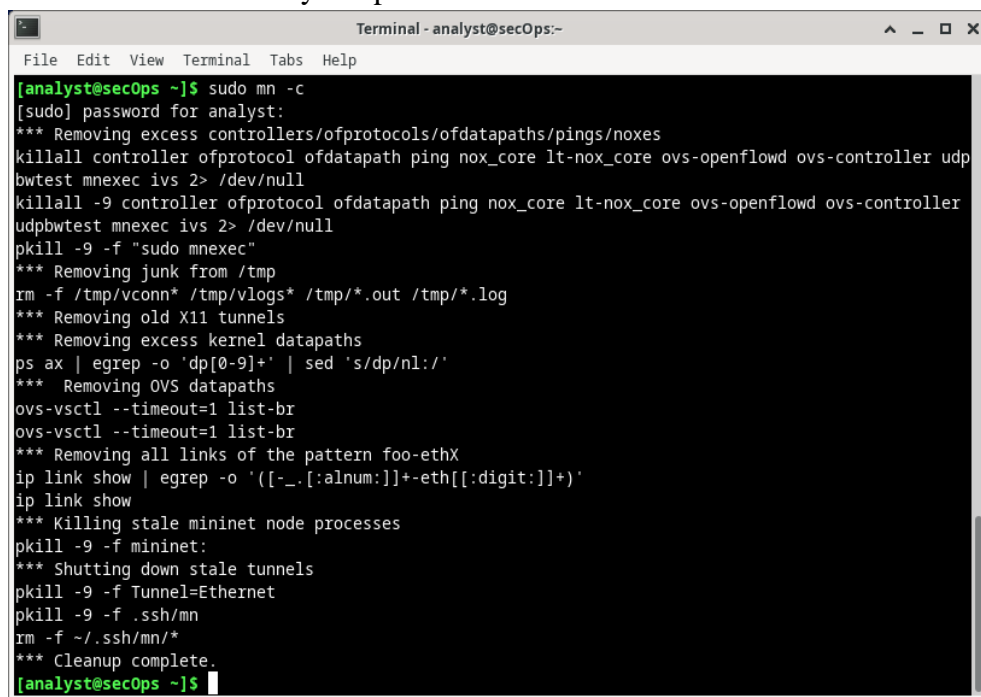


```
Terminal - analyst@secOps:-
File Edit View Terminal Tabs Help

mininet> quit
*** Stopping 0 controllers

*** Stopping 5 terms
*** Stopping 15 links
.....
*** Stopping 3 switches
S5 S9 S10
*** Stopping 13 hosts
R1 R4 H1 H2 H3 H4 H5 H6 H7 H8 H9 H10 H11
*** Done
[analyst@secOps ~]$
```

7. Setelah keluar dari Mininet, bersihkan proses yang dimulai oleh Mininet. Masukkan kata sandi cyberops saat diminta.



```
Terminal - analyst@secOps:-
File Edit View Terminal Tabs Help

[analyst@secOps ~]$ sudo mn -c
[sudo] password for analyst:
*** Removing excess controllers/ofprotocols/ofdatapaths/pings/noxes
killall controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udp
bwtest mnexec ivs 2> /dev/null
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller
udpbwtest mnexec ivs 2> /dev/null
pkill -9 -f "sudo mnexec"
*** Removing junk from /tmp
rm -f /tmp/vconn* /tmp/vlogs* /tmp/*.out /tmp/*.log
*** Removing old X11 tunnels
*** Removing excess kernel datapaths
ps ax | egrep -o 'dp[0-9]+' | sed 's/dp/nl:/'
*** Removing OVS datapaths
ovs-vsctl --timeout=1 list-br
ovs-vsctl --timeout=1 list-br
*** Removing all links of the pattern foo-ethX
ip link show | egrep -o '([_.:alnum:]]+-eth[[:digit:]]+)'
ip link show
*** Killing stale mininet node processes
pkill -9 -f mininet:
*** Shutting down stale tunnels
pkill -9 -f Tunnel=Ethernet
pkill -9 -f .ssh/mn
rm -f ~/.ssh/mn/*
*** Cleanup complete.
[analyst@secOps ~]$
```

V. Hasil dan Pembahasan

Pada praktikum ini, mahasiswa akan melakukan pengamatan atau pendeteksian terhadap aktivitas lalu lintas dalam suatu jaringan komputer menggunakan IDS berbasis Linux yaitu *Snort*. Dalam praktikum ini, terdapat *tools emulator* yang digunakan yaitu mininet. Pada mininet dapat mensimulasikan kinerja antara *end-host*, *switch*, *router*, kotroler, dan *link* dalam sebuah kernel Linux. Mininet merupakan sebuah sistem virtualisasi yang dapat menggambarkan jaringan yang besar.

Pada mininet terdapat beberapa *node* yang digunakan, diantaranya adalah R1, H5, dan H10. R1 diatur sebagai *router* yang menjalankan *snort* serta *firewall*, H10 sebagai *web server*, dan H5 digunakan sebagai *host* yang akan menjadi *client* dari *web server*. Dalam *web server* terdapat *file* berbahaya bernama W32.Nimda.Amm.exe. Ketika *host*

H5 melakukan pengunduhan *file* tersebut, maka *snort* pada R1 akan memeriksa muatan dalam paket yang diunduh. Adanya *payload* yang cocok dengan setidaknya satu *signature* memicu *alert* pada R1. *Alert* yang ditampilkan dari hasil praktikum adalah berupa *signature Malicious Sever Hit* dengan alamat sumbernya yaitu 209.165.200.235 dengan *port* 51166 serta alamat tujuannya yaitu 209.165.202.133 dengan *port* 6666. Dari *alert* tersebut ditampilkan juga waktu pengunduhan *file* tersebut, yaitu tanggal 20 Maret 2023 pukul 22:43:54.

Untuk pencegahan terhadap ancaman berikutnya dapat dilakukan dengan membuat aturan *firewall* sesuai dengan IDS *alert*. Caranya adalah membuat aturan *firewall* yang akan melakukan DROP paket yang melewati *router* ke suatu jaringan luar yang telah terindikasi berbahaya, dalam kasus ini adalah sumber dari *file* W32.Nimda.Amm.exe. *Chain* yang digunakan adalah *forward* dengan IP, *port*, serta protokol yang sebelumnya telah terdeteksi pada R1. Untuk mencegah terdapat *file* berbahaya lain dalam *server* tersebut, dapat juga dilakukan pendekatan yang lebih agresif dan valid dengan melakukan *block* terhadap IP *Server* tujuan tanpa menentukan IP, *port*, serta protokol yang digunakan. Dengan ini, akses ke *server* tujuan akan sepenuhnya terpotong sehingga *host* tidak akan dapat mengakses semua hal yang ada pada *server* yang telah diblok.

VI. Kesimpulan

1. *Snort* mendeteksi aktivitas lalu lintas dalam suatu jaringan.
2. Mininet merupakan sebuah sistem virtualisasi yang dapat menggambarkan jaringan yang besar.
3. *Snort* akan memberikan peringatan (*alert*) saat setidaknya terdapat 1 *signature* yang cocok dengan *payload*.
4. Untuk pencegahan terhadap ancaman berikutnya dapat dilakukan dengan membuat aturan *firewall* sesuai dengan IDS *alert*.
5. Untuk mencegah *file* berbahaya lain diunduh dari *server* yang telah terindikasi sebelumnya, dapat dilakukan *block* terhadap IP *Server* tujuan secara penuh tanpa menentukan IP, *port*, serta protokol yang digunakan.
6. Pemblokiran terhadap IP *Server* akan memotong semua akses yang ada pada *server* yang telah diblok.

VII. Daftar Pustaka

- Jagoan Hosting Team. (2021). *Apa itu Firewall? Fungsi, Manfaat, Cara Kerja dan Jenisnya*. Diakses pada 18 Maret 2023 dari <https://www.jagoanhosting.com/blog/apa-itu-firewall/>
- Mochlis Budiono. (Tanpa Tahun). *SNORT*. Diakses pada 18 Maret 2023 dari <https://mochlisbudiono.wordpress.com/snort/#:~:text=Snort%20merupakan%20program%20khusus%20dalam%20Deteksi%20Penyusupan%20%28Intrusion,secara%20cepat%20dengan%20menggunakan%20program%20khusus%20yang%20otomatis.>