

**LAPORAN PRAKTIKUM
KEAMANAN INFORMASI 1
UNIT 8**



DI SUSUN OLEH:

Nama : Bintang Nur K
NIM : 21/481453/SV/19790
Kelas : RI4AA
Hari, tanggal : Selasa, 21 Maret 2023
Dosen Pengampu : Anni Karimatul Fauziyyah, S.Kom., M.Eng
Asisten Praktikum : Gabriella Alvera Chaterine

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

UNIT 8

SNORT DAN FIREWALL RULES

I. TUJUAN

- Mempersiapkan Lingkungan Virtual
- Firewall dan Log IDS
- Hentikan dan Hapus Proses Mininet

II. LATAR BELAKANG

Intrusion Detection System atau IDS adalah sebuah sistem yang memonitor trafik jaringan untuk mendeteksi aktivitas-aktivitas mencurigakan. Jika aktivitas mencurigakan tersebut ditemukan, IDS akan melaporkannya dalam bentuk peringatan

Snort merupakan salah satu aplikasi *firewall* yang dikonfigurasi dalam terminal linux, meliputi konfigurasi *snort*, *input rule snort*, dan hasil alert *snort* pada terminal linux.

Metode untuk menangani Snort berjalan pada *mode inline* dengan menggunakan modul *daq_afpacket* dalam snort itu sendiri, dan untuk melakukan blok ketika terjadi serangan, snort menggunakan *firewall iptables*. Alert diimplementasikan pada email menggunakan protokol *smtp* dan pada telegram menggunakan id dan api telegram. Hasil dari penelitian menyatakan pembuatan web *interface* dapat dengan mudah mengelola *rule* dan alert *snort*, serta dapat diaplikasikan dalam beberapa serangan yang diujikan.

Dalam topologi jaringan yang aman, peringatan jaringan dihasilkan oleh berbagai jenis perangkat seperti peralatan keamanan, firewall, perangkat IPS, router, switch, server, dan banyak lagi.

Masalahnya adalah 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.

III. ALAT DAN BAHAN

Alat dan Bahan yang dibutuhkan untuk melaksanakan praktikum adalah

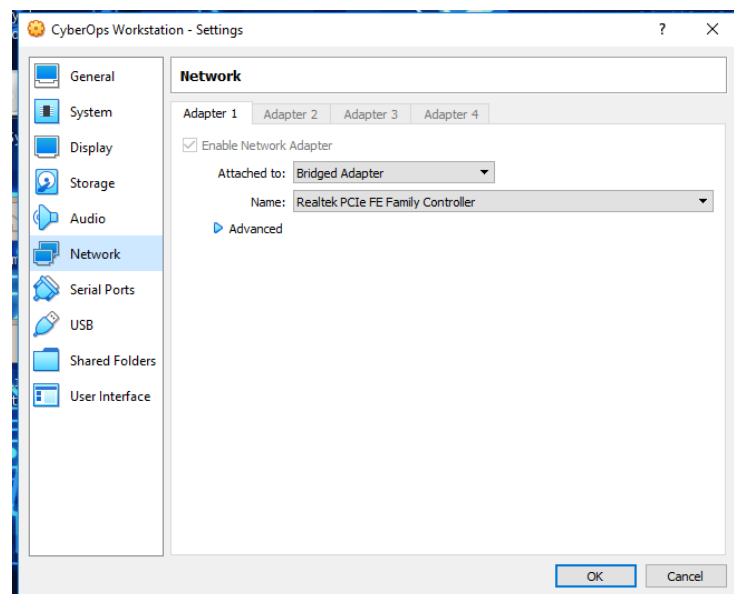
- Mesin Virtual CyberOps Workstation
- Koneksi Internet

IV. LANGKAH KERJA DAN HASIL

1. Bagian 1: Mempersiapkan Lingkungan Virtual

Luncurkan Oracle VirtualBox dan ubah CyberOps Workstation untuk mode Bridged, jika perlu.

Pilih Mesin > Pengaturan > Jaringan. Di bawah Attached To, pilih Bridged Adapter (atau jika Anda menggunakan WiFi dengan proxy, Anda mungkin memerlukan adaptor NAT) dan klik OK.



Luncurkan VM CyberOps Workstation, buka terminal dan konfigurasi jaringannya dengan menjalankan skrip `configure_as_dhcp.sh`. Karena skrip memerlukan hak pengguna super, berikan kata sandi untuk user analyst

```
[analyst@secOps~]$
```

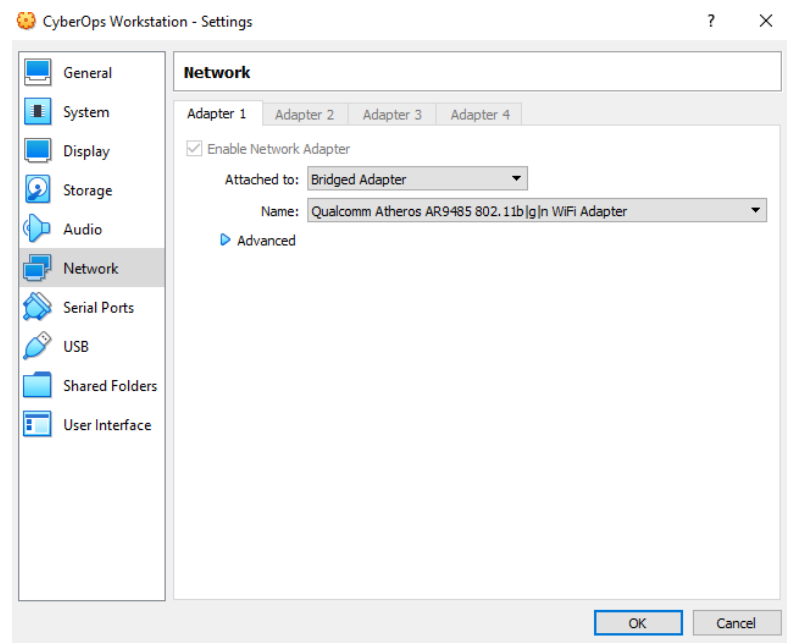
```
sudo./lab.support.files/scripts/configure_as_dhcp.sh
```

```
[sudo] password for analyst:
```

```
[analyst@secOps ~]$
```

```
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 ~]$
```

Lalu jika tersambung dengan jaringan WiFi



Gunakan perintah ifconfig untuk memverifikasi CyberOps Workstation VM sekarang memiliki alamat IP di jaringan lokal Anda. Anda juga dapat menguji konektivitas ke server web publik dengan melakukan ping ke www.cisco.com. Gunakan Ctrl+C untuk menghentikan ping.

```
[analyst@secOps ~]$ ping www.cisco.com
```

```
[analyst@secOps ~]$ ping www.cisco.com  
ping: www.cisco.com: Temporary failure in name resolution  
[analyst@secOps ~]$
```

Lalu apabila tersambung dengan jaringan WiFi sendiri maka akan berjalan program seperti ini

```
Terminal - analyst@secOps:~  
File Edit View Terminal Tabs Help  
[analyst@secOps ~]$ ping www.cisco.com  
PING e2867.dsca.akamaiedge.net (104.69.160.9) 56(84) bytes of data.  
64 bytes from a104-69-160-9.deploy.static.akamaitechnologies.com (104.69.160.9):  
  icmp_seq=1 ttl=56 time=60.2 ms  
64 bytes from a104-69-160-9.deploy.static.akamaitechnologies.com (104.69.160.9):  
  icmp_seq=2 ttl=56 time=72.6 ms  
64 bytes from a104-69-160-9.deploy.static.akamaitechnologies.com (104.69.160.9):  
  icmp_seq=3 ttl=56 time=56.2 ms  
64 bytes from a104-69-160-9.deploy.static.akamaitechnologies.com (104.69.160.9):  
  icmp_seq=4 ttl=56 time=53.5 ms  
64 bytes from a104-69-160-9.deploy.static.akamaitechnologies.com (104.69.160.9):  
  icmp_seq=5 ttl=56 time=72.7 ms  
64 bytes from a104-69-160-9.deploy.static.akamaitechnologies.com (104.69.160.9):  
  icmp_seq=6 ttl=56 time=58.7 ms  
64 bytes from a104-69-160-9.deploy.static.akamaitechnologies.com (104.69.160.9):  
  icmp_seq=7 ttl=56 time=75.6 ms  
64 bytes from a104-69-160-9.deploy.static.akamaitechnologies.com (104.69.160.9):  
  icmp_seq=8 ttl=56 time=65.3 ms  
^C  
--- e2867.dsca.akamaiedge.net ping statistics ---  
8 packets transmitted, 8 received, 0% packet loss, time 7010ms  
rtt min/avg/max/mdev = 53.518/64.357/75.598/7.879 ms  
[analyst@secOps ~]$
```

2. Bagian 2: Firewall & IDS Logs

- a. Dari VM CyberOps Workstation, jalankan skrip untuk memulai mininet.mininet.

```
[analyst@secOps ~]$ sudo ./lab.support.files/scripts/cyberops_extended_topo_no_fw.py  
[sudo] password for analyst:  
*** 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>
```

- b. Dari prompt mininet, buka shell di R1 menggunakan perintah di bawah ini

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

Jawab: pada shell ini ialah root user

- ```
./lab.support.files/scripts/start_snort.sh
```

```

"Node: R1"

WARNING: flowbits key 'file,rjs' is set but not ever checked.
WARNING: flowbits key 'NetDemon_OpenBrowser' is set but not ever checked.
WARNING: flowbits key 'file,flv' is set but not ever checked.
WARNING: flowbits key 'file,jmx' is set but not ever checked.
WARNING: flowbits key 'file,xls' is set but not ever checked.
WARNING: flowbits key 'file,upd' is set but not ever checked.
WARNING: flowbits key 'file,mng' is set but not ever checked.
WARNING: flowbits key 'file,collada' is set but not ever checked.
WARNING: flowbits key 'file,4xm' is set but not ever checked.
WARNING: flowbits key 'file,ses' is set but not ever checked.
WARNING: flowbits key 'file,jpeg' is set but not ever checked.
234 out of 1024 flowbits in use.

[Port Based Pattern Matching Memory]
+- [Aho-Corasick Summary] -----
| Storage Format : Full-Q
| Finite Automaton : DFA
| Alphabet Size : 256 Chars
| Sizeof State : Variable (1,2,4 bytes)
| Instances : 40
| 1 byte states : 33
| 2 byte states : 7
| 4 byte states : 0
| Characters : 34210
| States : 26401
| Transitions : 819777
| State Density : 12.1%
| Patterns : 2456
| Match States : 2407
| Memory (MB) : 13.80
| Patterns : 0.25
| Match Lists : 0.45
| DFA
| 1 byte states : 0.28
| 2 byte states : 12.63
4 byte states : 0.00
[Number of patterns truncated to 20 bytes: 14]
oap DAQ configured to passive.
Acquiring network traffic from "R1-eth0".
Reload thread starting...
Reload thread started, thread 0x7ffa579b1700 (852)
Decoding Ethernet
Set gid to 29
Set uid to 29

--== Initialization Complete ==--

o""~ -*) Short! <*-
 Version 2.9.11.1 GRE (Build 268)

```

```

==== Initialization Complete ====

--> Snort! <*-
Version 2,9,11,1 GRE (Build 268)
By Martin Roesch & The Snort Team; http://www.snort.org/contact#team
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_SHTP Version 1.1 <Build 9>
Preprocessor Object: SF_SDP 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_PDP Version 1.0 <Build 1>

Commencing packet processing (pid=839)

```

- d. Dari prompt mininet CyberOps Workstation VM, buka shell untuk host H5 dan H10

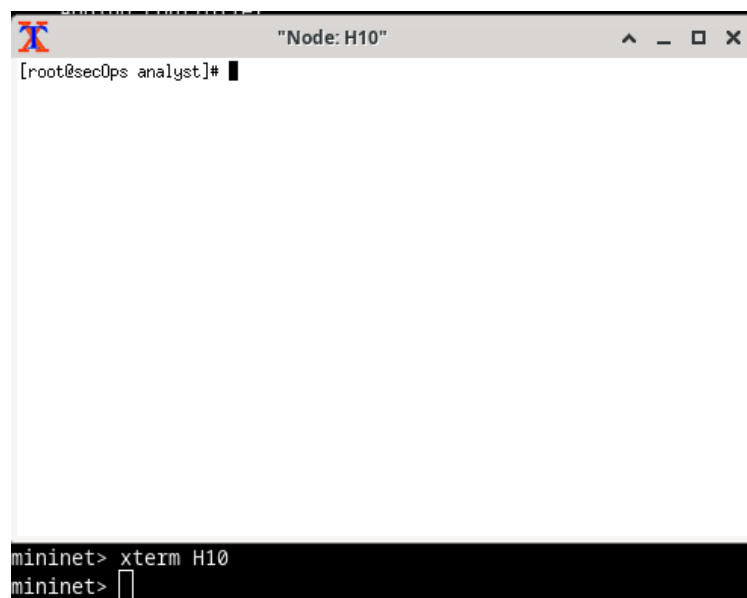
mininet> xterm H5

```

mininet> xterm H5
mininet>

```

mininet> xterm H10



```

"Node: H10"
[root@secOps analyst]#

```

mininet> xterm H10

mininet>

- e. H10 akan mensimulasikan server di Internet yang menghosting 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]# ./lab.support.files/scripts/mal_server_start.sh
2023/03/20 22:01:22 [emerg] 902#902: bind() to 0.0.0.0:6666 failed (98: Address
already in use)
2023/03/20 22:01:22 [emerg] 902#902: bind() to 0.0.0.0:6666 failed (98: Address
already in use)
2023/03/20 22:01:22 [emerg] 902#902: bind() to 0.0.0.0:6666 failed (98: Address
already in use)
2023/03/20 22:01:22 [emerg] 902#902: bind() to 0.0.0.0:6666 failed (98: Address
already in use)
2023/03/20 22:01:22 [emerg] 902#902: bind() to 0.0.0.0:6666 failed (98: Address
already in use)
2023/03/20 22:01:22 [emerg] 902#902: still could not bind()
[root@secOps analyst]#
```

- f. 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
```

```
[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
894/nginx: master p
[root@secOps analyst]#
```

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

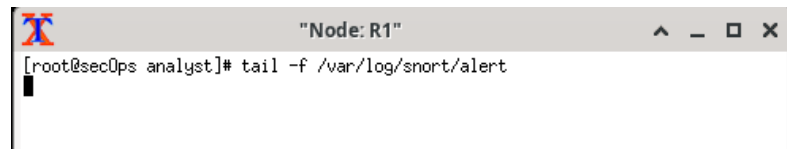
- g. Di jendela terminal R1, sebuah instance dari Snort sedang berjalan. Untuk memasukkan lebih banyak perintah di R1, buka terminal R1 lain dengan memasukkan xterm R1 lagi di jendela terminal VM CyberOps Workstation. Anda mungkin juga ingin mengatur jendela terminal sehingga Anda dapat melihat dan berinteraksi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

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



- h. 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.

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

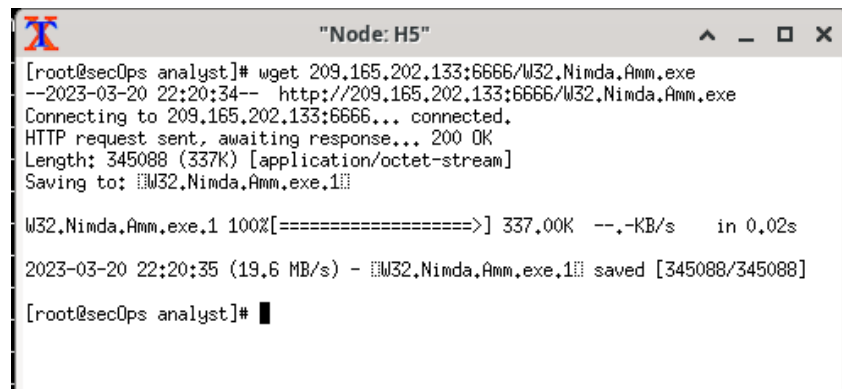


Karena belum ada peringatan yang direkam, log harus kosong. Namun, jika Anda telah menjalankan lab ini sebelumnya, entri peringatan lama mungkin ditampilkan. Dalam kedua kasus, Anda tidak akan menerima prompt setelah mengetik perintah ini. Jendela ini akan menampilkan peringatan saat itu terjadi.

- i. 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.

```
[root@secOps analyst]#
```

```
wget 209.165.202.133:6666/W32.Nimda.Amm.exe
```



Pertanyaan:

Port apa yang digunakan saat berkomunikasi dengan server web malware? Apa indikatornya?

Jawab:

Port 6666 dan indikatornya ditentukan di URL setelah tanda “:”

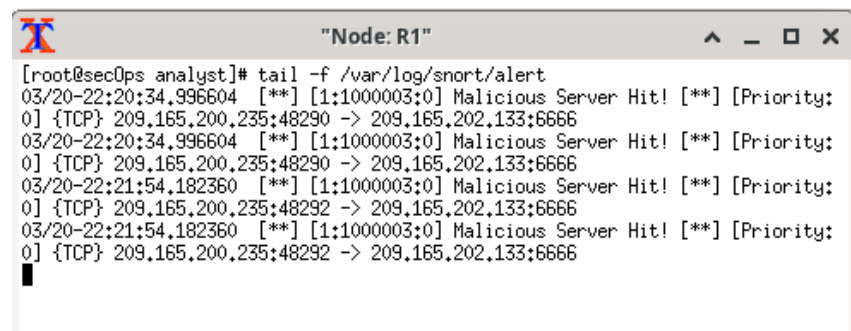
Apakah file telah diunduh sepenuhnya?

Jawab: Iya, sudah

Apakah IDS menghasilkan peringatan yang terkait dengan unduhan file?

Jawab: Iya, benar

- j. 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:



```
[root@secOps analyst]# tail -f /var/log/snort/alert
03/20-22:20:34.996604 [**] [1:1000003:0] Malicious Server Hit! [**] [Priority:
0] {TCP} 209.165.200.235:48290 -> 209.165.202.133:6666
03/20-22:20:34.996604 [**] [1:1000003:0] Malicious Server Hit! [**] [Priority:
0] {TCP} 209.165.200.235:48290 -> 209.165.202.133:6666
03/20-22:21:54.182360 [**] [1:1000003:0] Malicious Server Hit! [**] [Priority:
0] {TCP} 209.165.200.235:48292 -> 209.165.202.133:6666
03/20-22:21:54.182360 [**] [1:1000003:0] Malicious Server Hit! [**] [Priority:
0] {TCP} 209.165.200.235:48292 -> 209.165.202.133:6666
```

Pertanyaan:

Berdasarkan peringatan yang ditunjukkan di atas, apa alamat IPv4 sumber dan tujuan yang digunakan dalam transaksi?

Jawab: pada alamat IP sumber adalah 209.165.200.235 dan alamat IP tujuan adalah 209.165.202.133

Berdasarkan alert di atas, port sumber dan tujuan apa yang digunakan dalam transaksi?

Jawab: port sumber yaitu 48292 dan port tujuan yaitu 6666

Berdasarkan peringatan yang ditunjukkan di atas, kapan pengunduhan dilakukan?

Jawab: 20 Maret pukul 22.20

Berdasarkan peringatan yang ditunjukkan di atas, apa pesan yang direkam IDS signature?

Jawab: "Malicious Server Hit!"

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

```
[root@secOps analyst]# tcpdump -i H5-eth0 -w nimda.download.pcap &
```

```
[root@secOps analyst]# tcpdump -i H5-eth0 -w nimda.download.pcap &
[1] 1057
[root@secOps analyst]# tcpdump: listening on H5-eth0, link-type EN10MB (Ethernet
) , capture size 262144 bytes
[root@secOps analyst]# █
```

- k. 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 simbol ini, tcpdump akan membuat terminal tidak dapat digunakan saat sedang berjalan. Perhatikan [1] 5633; itu menunjukkan satu proses dikirim ke latar belakang dan ID prosesnya (PID) adalah 5366. PID Anda kemungkinan besar akan berbeda.

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

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

```
[root@secOps analyst]#
```

```
wget 209.165.202.133:6666/W32.Nimda.Amm.exe
```

```

[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe
--2023-03-20 22:21:54-- 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.2'

W32.Nimda.Amm.exe.2 100%[=====>] 337,00K --.-KB/s in 0,01s

2023-03-20 22:21:54 (26,3 MB/s) - 'W32.Nimda.Amm.exe.2' saved [345088/345088]

[root@secOps analyst]# tcpdump -i H5-eth0 -w nimda.download.pcap &
[1] 1057
[root@secOps analyst]# tcpdump: listening on H5-eth0, link-type EN10MB (Ethernet
), capture size 262144 bytes
[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe
--2023-03-20 22:27:43-- 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.3'

W32.Nimda.Amm.exe.3 100%[=====>] 337,00K --.-KB/s in 0,02s

2023-03-20 22:27:43 (17,7 MB/s) - 'W32.Nimda.Amm.exe.3' saved [345088/345088]

[root@secOps analyst]# █

```

- n. 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.

```
[root@secOps analyst]# fg
```

```

[root@secOps analyst]# fg
tcpdump -i H5-eth0 -w nimda.download.pcap
^C65 packets captured
65 packets received by filter
0 packets dropped by kernel
[root@secOps analyst]# █

```

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

```
[root@secOps analyst]# ls -l
```



Pertanyaan : Rantai apa yang saat ini digunakan oleh R1?

Jawab: INPUT, FORWARD, dan OUTPUT.

- c) 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:

```
[root@secOps ~]# iptables -I FORWARD -p tcp -d 209.165.202.133 --dport 6666 -j DROP
```

```
[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

- d) Gunakan perintah iptables lagi untuk memastikan aturan telah ditambahkan ke rantai FORWARD. VM CyberOps Workstation mungkin memerlukan beberapa detik untuk menghasilkan output:

```
[root@secOps analyst]# iptables -L -v
```

```
[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
 0 0 tcp dpt:6666

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

[root@secOps analyst]#
```

- e) Pada H5, coba unduh file lagi:

```
[root@secOps analyst]#
```

```
wget 209.165.202.133:6666/W32.Nimda.Amm.exe
```

```
[root@secOps analyst]# wget 209.165.202.133:6666/W32.Nimda.Amm.exe
--2023-03-20 22:40:02-- http://209.165.202.133:6666/W32.Nimda.Amm.exe
Connecting to 209.165.202.133:6666... ^C
[root@secOps analyst]#
```

Pertanyaan:

Apakah unduhan berhasil kali ini? Jelaskan

Jawab: tidak berhasil, karena koneksi yang terhubung ke server hosting malware diblokir oleh firewall

Apa pendekatan yang lebih agresif tetapi juga valid saat memblokir server yang melanggar?

Jawab: Dengan menggunakan firewall yang dikonfigurasi dengan benar untuk memblokir akses ke server yang melanggar. Dan dapat menghubungi penyedia layanan hosting untuk memberi tahu mereka tentang pelanggaran tersebut dan meminta bantuan mereka dalam menyelesaikan masalahnya.

- f) Hentikan dan Hapus Proses Mininet

1. Arahkan ke terminal yang digunakan untuk memulai Mininet.  
Hentikan Mininet dengan memasukkan quit di jendela terminal VM CyberOps utama.

```
mininet> quit
*** Stopping 0 controllers

*** Stopping 14 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 ~]$
```

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

```
[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 udprawtest mnexec ivs 2> /dev/n
ull
killall -9 controller ofprotocol ofdatapath ping nox_core lt-nox_core ovs-openflowd ovs-controller udprawtest mnexec ivs 2> /de
v/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. ANALISIS

Pada praktikum kali ini melakukan snort dan firewall rules menggunakan Virtual Machine yang di set untuk lingkungan Mininet yang ditunjukkan di Topologi. Jika terdapat kendala pada VM, tambah memori yang disediakan menjadi 2G.

IDS (Intrusion Detection System) adalah sebuah aplikasi perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. IDS digunakan untuk mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan.

Pada umumnya, cara kerja IDS adalah mendeteksi dan menemukan ancaman. Cara kerja ini tidak jauh berbeda dengan program-program *antivirus*, dimana sistem akan mendeteksi aktivitas berbahaya.

IDS memantau dan mencocokkan trafik dengan pusat data intrusi yang menyimpan kumpulan data berbagai jenis penyusupan atau serangan. Jika terdapat kecocokan, selanjutnya IDS akan mengidentifikasi sekaligus mengirimkan peringatan. Perlu diingat bahwa IDS hanya dapat mengirimkan peringatan, bukan mengambil tindakan secara aktif seperti menghapus atau memblokir ancaman

Firewall dan Intrusion Detection Systems (IDS) sering digunakan untuk mengotomatisasi sebagian tugas pemantauan lalu lintas. Baik firewall dan IDS mencocokkan lalu lintas masuk dengan aturan administratif.

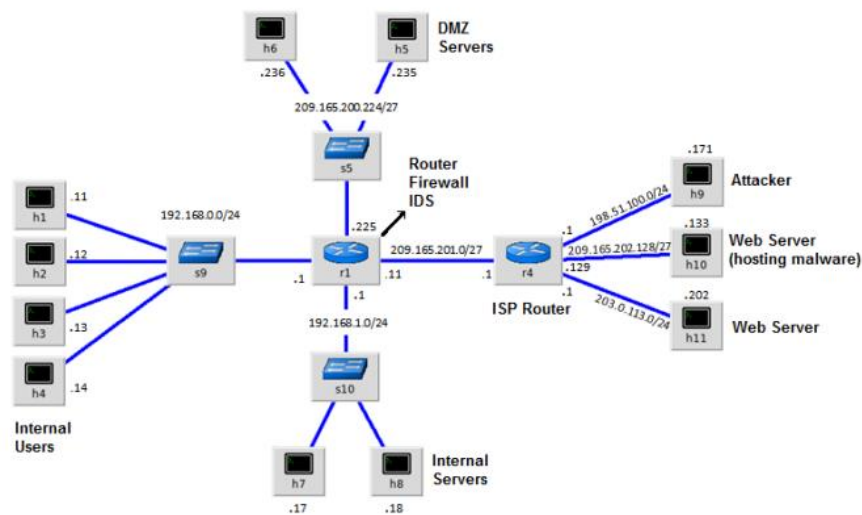


Firewall biasanya membandingkan header paket dengan kumpulan aturan sementara IDS sering menggunakan muatan paket untuk perbandingan kumpulan aturan. Karena firewall dan IDS menerapkan aturan yang telah ditentukan sebelumnya ke bagian yang berbeda dari paket IP, IDS dan aturan firewall memiliki struktur yang berbeda. Meskipun ada perbedaan dalam struktur aturan, beberapa kesamaan antara komponen aturan tetap ada. Misalnya, aturan firewall dan IDS berisi komponen yang cocok dan komponen tindakan.

Tindakan diambil setelah kecocokan ditemukan.

- Matching component - menentukan elemen paket yang diinginkan, seperti: sumber paket; tujuan paket; protokol dan port lapisan transport; dan data yang termasuk dalam paket payload.
- Action component - menentukan apa yang harus dilakukan dengan paket yang cocok dengan komponen, seperti: menerima dan meneruskan paket; drop paket; atau kirim paket ke kumpulan aturan sekunder untuk pemeriksaan lebih lanjut.

Desain firewall yang umum adalah mengirim paket secara default dan secara manual menentukan lalu lintas apa yang harus diizinkan. Dikenal sebagai drop-by-default, desain ini memiliki keuntungan melindungi jaringan dari protokol dan serangan yang tidak diketahui. Sebagai bagian dari desain ini adalah umum untuk mencatat peristiwa paket yang dikirimkan karena ini adalah paket yang tidak diizinkan secara eksplisit dan oleh karena itu, melanggar kebijakan organisasi. Peristiwa semacam itu harus direkam untuk analisis di masa mendatang.



## VI. KESIMPULAN

Setelah melaksanakan praktikum yang saya dapatkan adalah

- IDS diklasifikasikan menjadi lima jenis, yakni NIDS, HIDS, PIDS, APIDS, dan *Hybrid*
- IPS dan IDS adalah dua sistem berbeda dengan kelebihan dan kekurangannya masing-masing. Meski memiliki teknis yang sama, yakni memantau dan mendeteksi ancaman pada jaringan, namun keduanya tidak dapat disamaratakan. Dalam hal ini, IDS hanya dapat mendeteksi dan mengirimkan peringatan saja. Sedangkan *Intrusion Prevention System* (IPS) mampu mengambil tindakan aktif untuk memblokir ancaman yang terdeteksi
- IDS adalah sebuah sistem yang memantau trafik jaringan untuk mendeteksi intrusi atau aktivitas mencurigakan serta melaporkannya dalam bentuk peringatan.

Link Github:

[https://github.com/BintangNu/481453\\_Bintang-Nur\\_UNIT8](https://github.com/BintangNu/481453_Bintang-Nur_UNIT8)

## VII. DAFTAR PUSTAKA

Prak KI 1. (2023). Materi Pertemuan 6. Retrieved March 27, 2023, from Elok UGM

Huda, N. (2022, October 10). *APA ITU Intrusion Detection System (IDS)? Jenis Dan Cara Kerjanya*. Blog Dewaweb. Retrieved March 27, 2023, from <https://www.dewaweb.com/blog/ids-adalah/>

Sabekti, M. A. (n.d.). *Pembuatan web interface snort untuk manajemen firewall DENGAN OPERASI CRUD (create, read, update, DELETE) pada file system snort Dan Pengujian Web Dengan Serangan Serta notifikasi Pada email Dan Telegram*. IJAI (Indonesian Journal of Applied Informatics). Retrieved March 27, 2023, from <https://jurnal.uns.ac.id/ijai/article/view/27836>