



# ZEEK

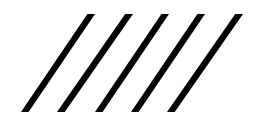
1805092- ISHIKA TARIN

1805114- AFROZA PARVIN DISA





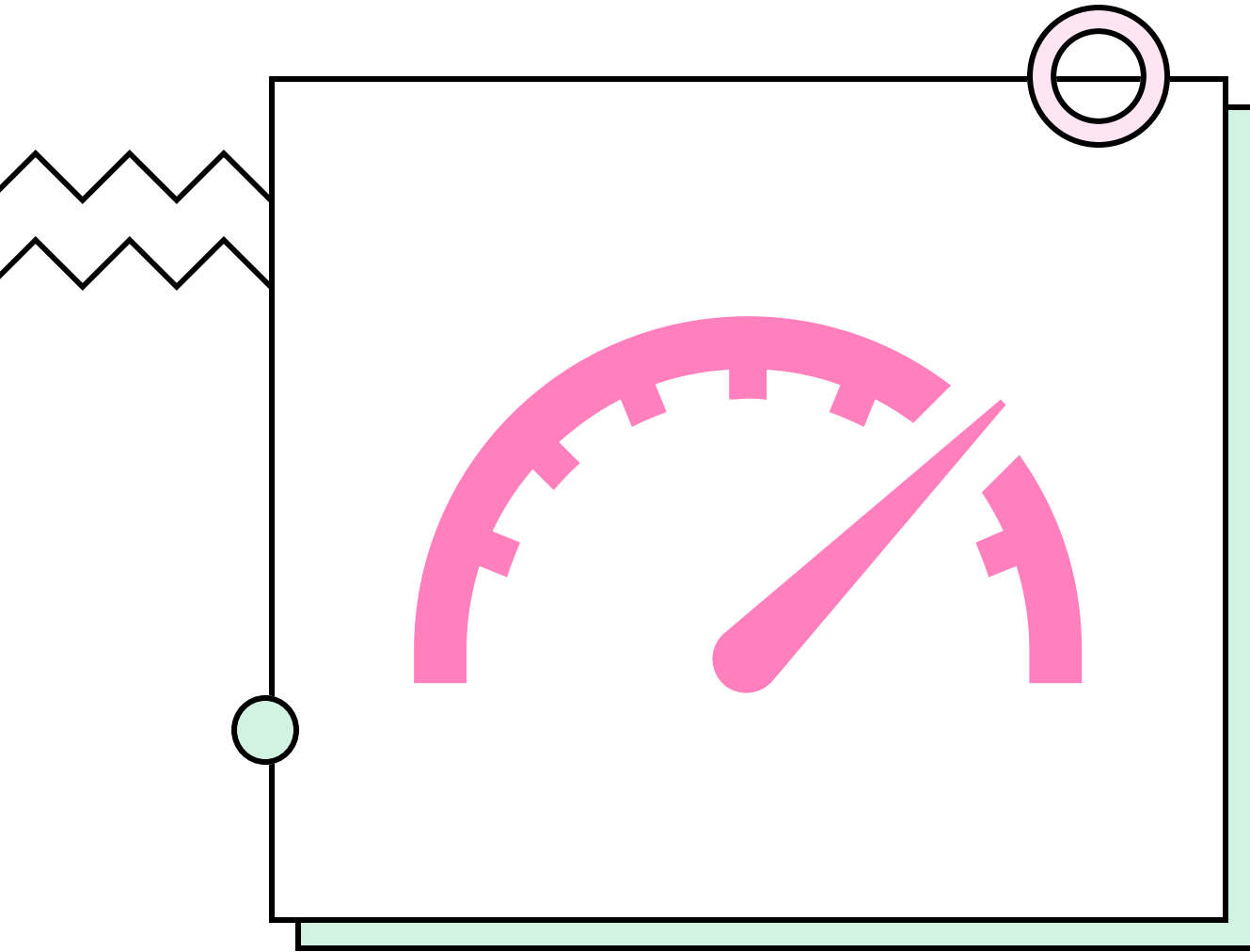
# What Is Zeek?

- A network security monitor (NSM) to support investigations of suspicious or malicious activity.
  - A fully customizable and extensible platform for traffic analysis.
  - A platform supporting extensive set of logs describing network activity, seen on the wire, but also application-layer transcripts.
- 



# **Zeek features**

- Live Monitoring of traffic
- Storing live traffics in Packet Capture (pcap) Files
- Command line utility with zeekcontrol
- Browsing Log Files
- Notification and log file update



# **LIVE TRAFFIC MONITORING & NETWORK ANALYSIS**





## Command Line: **ip a**

It lists all network interfaces (both physical and virtual) that are currently configured on system.

```
ime@ime-VirtualBox:/$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ac:8e:aa brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85779sec preferred_lft 85779sec
    inet6 fe80::a546:b26d:e38e:c91a/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

'**lo**' is the loopback interface with IPv4 address 127.0.0.1 and IPv6 address ::1

'**enp0s3**' is the physical interface with IPv4 address 10.0.2.15/24 and IPv6 address fe80::a546:b26d:e38e:c91a/64.





## Command Line: **sudo tcpdump -n -i enp0s3**

Tcpdump starts listening on the enp0s3 network interface.

```
ime@ime-VirtualBox:/$ sudo tcpdump -n -i enp0s3
[sudo] password for ime:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
16:18:00.768350 IP 10.0.2.15.44019 > 4.2.2.2.53: 47712+ [1au] A? push.services.mozilla.com. (54)
16:18:00.768454 IP 10.0.2.15.55102 > 4.2.2.2.53: 15301+ [1au] AAAA? push.services.mozilla.com. (54)
16:18:00.946699 IP 4.2.2.2.53 > 10.0.2.15.44019: 47712 2/0/1 CNAME autopush.prod.mozaws.net., A 34.117.65.55 (108)
16:18:03.338833 IP 10.0.2.15.45991 > 4.2.2.2.53: 22511+ [1au] AAAA? connectivity-check.ubuntu.com. (58)
16:18:03.399824 IP 4.2.2.2.53 > 10.0.2.15.45991: 22511 6/0/1 AAAA 2620:2d:4000:1::22, AAAA 2620:2d:4000:1::2a, AAAA 2620:2d:4000:1::2b, AAAA 2001:67c:1562::24, AAAA 2620:2d:4000:1::23, AAAA 2001:67c:1562::23 (226)
16:18:05.772731 IP 10.0.2.15.38407 > 8.8.8.8.53: 15301+ [1au] AAAA? push.services.mozilla.com. (54)
16:18:06.008659 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
16:18:06.009159 ARP, Reply 10.0.2.2 is-at 52:54:00:12:35:02, length 46
16:18:10.776208 IP 10.0.2.15.36902 > 8.8.8.8.53: 45728+ [1au] A? push.services.mozilla.com. (54)
16:18:10.776285 IP 10.0.2.15.34335 > 4.2.2.2.53: 15301+ [1au] AAAA? push.services.mozilla.com. (54)
16:18:10.851873 IP 8.8.8.8.53 > 10.0.2.15.36902: 45728 2/0/1 CNAME autopush.prod.mozaws.net., A 34.117.65.55 (108)
16:18:15.778792 IP 10.0.2.15.57710 > 8.8.8.8.53: 15301+ [1au] AAAA? push.services.mozilla.com. (54)
16:18:15.781465 IP 10.0.2.15.48597 > 8.8.8.8.53: 22952+ [1au] A? push.services.mozilla.com. (54)
16:18:16.022599 IP 8.8.8.8.53 > 10.0.2.15.57710: 15301 1/1/1 CNAME autopush.prod.mozaws.net. (174)
16:18:16.022880 IP 10.0.2.15.44503 > 8.8.8.8.53: 37826+ [1au] AAAA? autopush.prod.mozaws.net. (53)
16:18:16.029864 IP 8.8.8.8.53 > 10.0.2.15.48597: 22952 2/0/1 CNAME autopush.prod.mozaws.net., A 34.117.65.55 (108)
16:18:16.096202 IP 8.8.8.8.53 > 10.0.2.15.44503: 37826 0/1/1 (138)
16:18:16.097137 IP 10.0.2.15.36446 > 34.117.65.55.443: Flags [S], seq 1027016620, win 64240, options [mss 1460,sackOK,TS val 4101525700 ecr 0,nop,wscale 7], length 0
16:18:16.169543 IP 34.117.65.55.443 > 10.0.2.15.36446: Flags [S.], seq 1073344001, ack 1027016621, win 65535, options [mss 1460], length 0
16:18:16.169576 IP 10.0.2.15.36446 > 34.117.65.55.443: Flags [.], ack 1, win 64240, length 0
16:18:16.172654 IP 10.0.2.15.36446 > 34.117.65.55.443: Flags [P.], seq 1:651, ack 1, win 64240, length 650
16:18:16.173075 IP 34.117.65.55.443 > 10.0.2.15.36446: Flags [.], ack 651, win 65535, length 0
16:18:16.232064 IP 34.117.65.55.443 > 10.0.2.15.36446: Flags [P.], seq 1:213, ack 651, win 65535, length 212
16:18:16.232100 IP 10.0.2.15.36446 > 34.117.65.55.443: Flags [.], ack 213, win 64028, length 0
```

Each line represents an IPv4 packet captured at a specific time, showing the source and destination IP addresses, port numbers, TCP flags indicating an acknowledgment, window size, and length of the payload.







The line **14:38:10.400419 IP 10.0.2.15.39134 > 34.117.237.239.443: Flags [.], ack 1035, win 64028, length 0**

**14:38:10.400419:** This timestamp indicates the time at which the packet was captured.

**IP:** This indicates that the packet is an IPv4 packet.

**10.0.2.15.39134 > 34.117.237.239.443:** This part indicates the source and destination addresses along with their corresponding port numbers.

**10.0.2.15** is the source IP address., **39134** is the source port number.

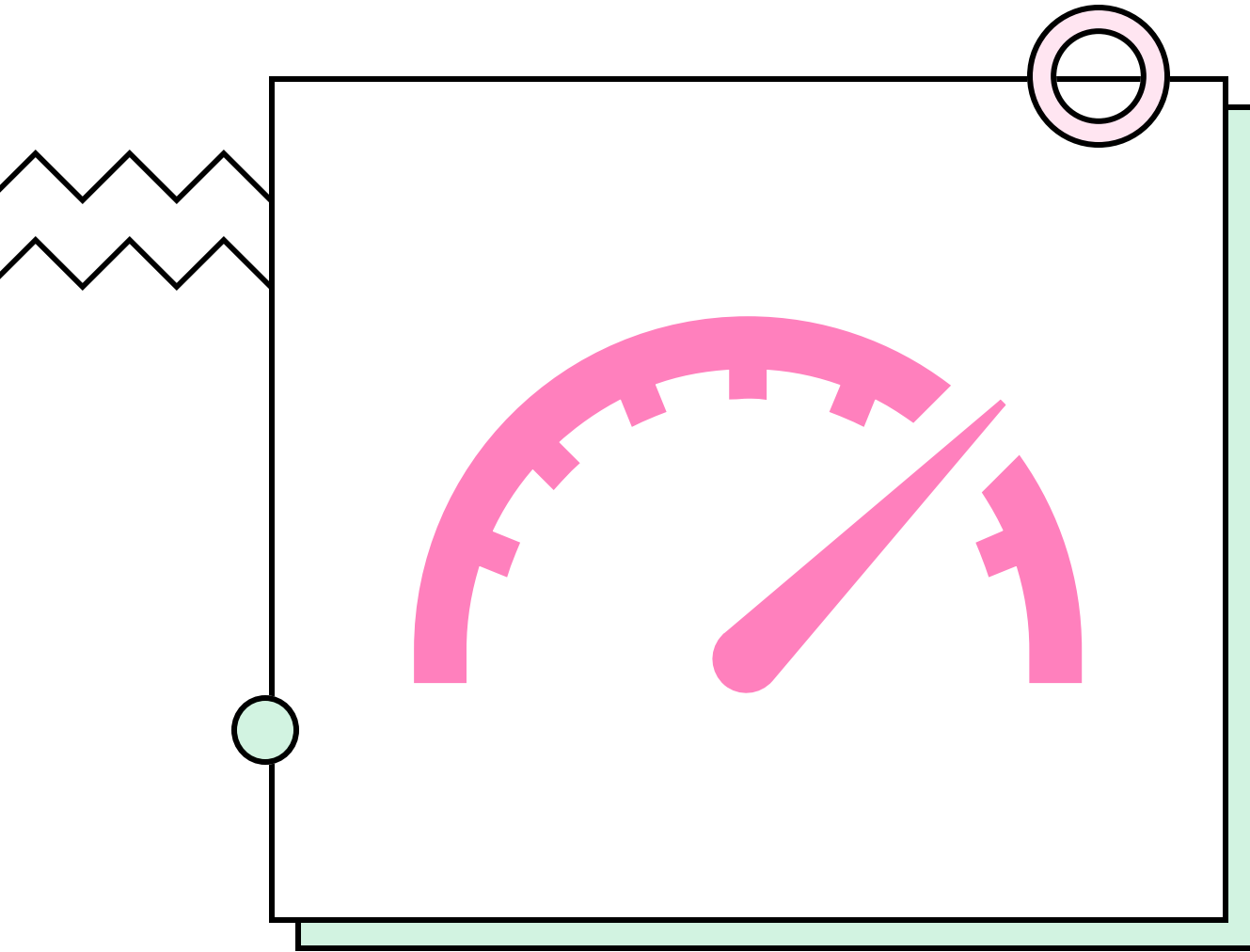
**34.117.237.239** is the destination IP address., **443** is the destination port number.

**Flags [.], ack 1035:** Here **[.]** indicates that this is an acknowledgment packet. **ack 1035** means the acknowledgment number is **1035**, indicating that the receiver has successfully received up to sequence number 1035.

**win 64028:** This shows the window size, which is 64028.

**length 0:** This indicates the length of the data payload in the packet. In this case, the length is 0, suggesting that this is a control or acknowledgment packet without any payload data.





# STORING LIVE TRAFFICS IN PACKET CAPTURE FILES







Opening a pcap file:

**ls -ld**

**sudo ls -ld pcap**

**sudo chmod +xrw pcap**

**sudo ls -ld pcap**

```
ime@ime-VirtualBox:/$ ls -ld .
drwxr-xr-x 20 root root 4096 আস্ট 25 11:45 .
ime@ime-VirtualBox:/$ sudo mkdir -m 700 pcap
ime@ime-VirtualBox:/$ ls
bin  cdrom  etc  lib  lib64  lost+found  mnt  pcap  root  sbin  srv  sys  usr
boot  dev  home  lib32  libx32  media  opt  proc  run  snap  swapfile  tmp  var
ime@ime-VirtualBox:/$ ls -ld
drwxr-xr-x 21 root root 4096 আস্ট 29 16:36 .
ime@ime-VirtualBox:/$ ls -l pcap
ls: cannot open directory 'pcap': Permission denied
ime@ime-VirtualBox:/$ sudo ls -l pcap
total 0
ime@ime-VirtualBox:/$ sudo ls -ld pcap
drwx----- 2 root root 4096 আস্ট 29 16:36 pcap
ime@ime-VirtualBox:/$ sudo chmod +x pcap
ime@ime-VirtualBox:/$ sudo ls -ld pcap
drwx--x--x 2 root root 4096 আস্ট 29 16:36 pcap
ime@ime-VirtualBox:/$ sudo chmod +rw pcap
ime@ime-VirtualBox:/$ sudo ls -ld pcap
```

Thus a packet capture file is opened where read , write and execution permission is given for the owner.





Writing at a pcap file:

**sudo tcpdump -i enp0s3 -w /pcap/captured\_traffic.pcap**

```
ime@ime-VirtualBox:/$ sudo tcpdump -i enp0s3 -w /pcap/captured_traffic.pcap
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C655 packets captured
655 packets received by filter
0 packets dropped by kernel
```

This command will capture network traffic on the "enp0s3" network interface and save it to the "/pcap/captured\_traffic.pcap" file within the "/pcap" directory.





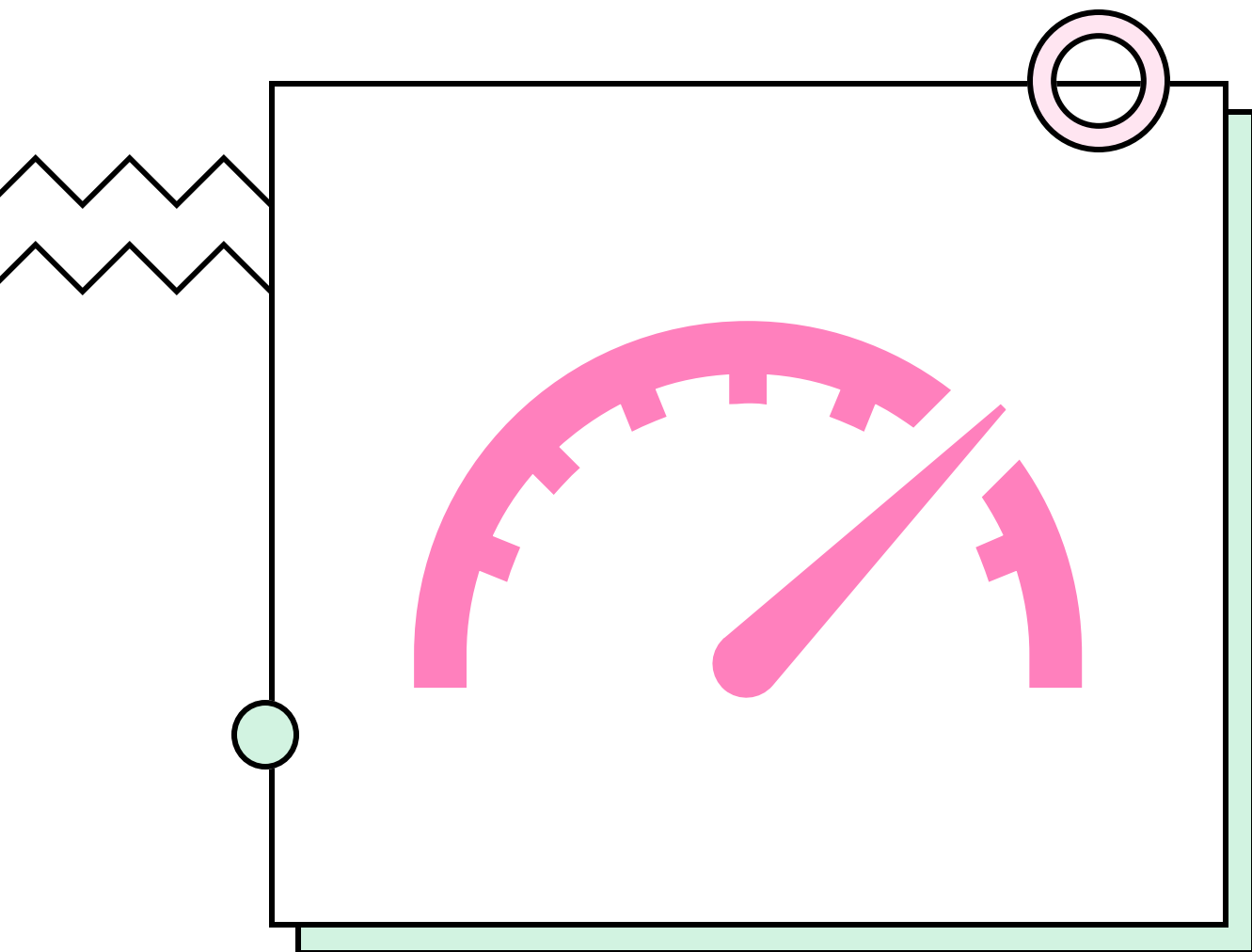
Reading from pcap file:

**sudo tcpdump -n -r /pcap/captured\_traffic.pcap**

```
ime@ime-VirtualBox:/$ sudo tcpdump -n -r /pcap/captured_traffic.pcap
reading from file /pcap/captured_traffic.pcap, link-type EN10MB (Ethernet), snapshot length 262144
16:40:48.467450 IP 10.0.2.15.36366 > 4.2.2.2.53: Flags [S], seq 3890530676, win 64240, options [mss 1460,sackOK,TS val 3965519187 ec
r 0,nop,wscale 7,tfo cookiereq,nop,nop], length 0
16:40:49.497187 IP 10.0.2.15.36366 > 4.2.2.2.53: Flags [S], seq 3890530676, win 64240, options [mss 1460,sackOK,TS val 3965520217 ec
r 0,nop,wscale 7], length 0
16:40:51.512520 IP 10.0.2.15.36366 > 4.2.2.2.53: Flags [S], seq 3890530676, win 64240, options [mss 1460,sackOK,TS val 3965522232 ec
r 0,nop,wscale 7], length 0
16:40:53.560334 ARP, Request who-has 10.0.2.2 tell 10.0.2.15, length 28
16:40:53.560547 ARP, Reply 10.0.2.2 is-at 52:54:00:12:35:02, length 46
16:40:55.607975 IP 10.0.2.15.36366 > 4.2.2.2.53: Flags [S], seq 3890530676, win 64240, options [mss 1460,sackOK,TS val 3965526328 ec
r 0,nop,wscale 7], length 0
16:40:58.476886 IP 10.0.2.15.46125 > 8.8.8.8.53: 44098+ [1au] AAAA? o33249.ingest.sentry.io. (52)
16:40:58.476971 IP 10.0.2.15.51933 > 8.8.8.8.53: 14957+ [1au] AAAA? chat.openai.com. (44)
16:40:58.477071 IP 10.0.2.15.60140 > 8.8.8.8.53: 20753+ [1au] A? chat.openai.com. (44)
16:40:58.477151 IP 10.0.2.15.60115 > 8.8.8.8.53: 45549+ [1au] A? o33249.ingest.sentry.io. (52)
16:41:03.482720 IP 10.0.2.15.35306 > 4.2.2.2.53: Flags [S], seq 876529738, win 64240, options [mss 1460,sackOK,TS val 3965534203 ecr
0,nop,wscale 7,tfo cookiereq,nop,nop], length 0
16:41:04.504194 IP 10.0.2.15.35306 > 4.2.2.2.53: Flags [S], seq 876529738, win 64240, options [mss 1460,sackOK,TS val 3965535224 ecr
0,nop,wscale 7], length 0
16:41:06.520694 IP 10.0.2.15.35306 > 4.2.2.2.53: Flags [S], seq 876529738, win 64240, options [mss 1460,sackOK,TS val 3965537240 ecr
0,nop,wscale 7], length 0
16:41:10.712503 IP 10.0.2.15.35306 > 4.2.2.2.53: Flags [S], seq 876529738, win 64240, options [mss 1460,sackOK,TS val 3965541432 ecr
0,nop,wscale 7], length 0
```

By running this command tcpdump will read the contents of the PCAP file and display the captured network traffic on your terminal, showing details like source and destination IP addresses, ports, packet sizes, timestamps, and more.





**COMMAND  
LINE  
UTILITY  
WITH ZEEK  
CONTROL**





Run Zeekctl:

**sudo ./zeekctl**

```
ime@ime-VirtualBox:/opt/zeek/bin$ ls -l zeekctl
-rwxr-xr-x 1 root root 28643 জানু 29 2015 zeekctl
ime@ime-VirtualBox:/opt/zeek/bin$ sudo ./zeekctl
```

Zeekctl (sometimes referred to as "**Zeek Control**") is a utility for managing and controlling Zeek, a powerful network security monitoring tool.

Zeekctl provides a set of commands and features that make it easier to configure, start, stop, and manage Zeek instances, especially in more complex deployment scenarios.

Zeek commands to Run:

- >**Zeek start** : zeek is started
- >**Zeek deploy** : zeek is started then it is stopped
- >**Zeek stop** : zeek is stopped
- >**Zeek diag**: to troubleshoot zeek installation







## zeek start, deploy and stop

```
[ZeekControl] > start
starting zeek ...
[ZeekControl] > deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
starting ...
starting zeek ...
[ZeekControl] > stop
stopping zeek ...
[ZeekControl] > █
```





## zeek diag

```
[ZeekControl] > diag
[zeek]

No core file found.

Zeek 5.0.9
Linux 6.2.0-26-generic

Zeek plugins: (none found)

==== No reporter.log

==== stderr.log
listening on enp0s3

==== stdout.log
max memory size          (kbytes, -m) unlimited
data seg size            (kbytes, -d) unlimited
virtual memory           (kbytes, -v) unlimited
core file size           (blocks, -c) unlimited

==== .cmdline
-i enp0s3 -U .status -p zeekctl -p zeekctl-live -p standalone -p local -p zeek l
ocal.zeek zeekctl zeekctl/standalone zeekctl/auto
```

If there are errors while trying to start the Zeek instance, you can view the details with the **diag** command. If started successfully, the Zeek instance will begin analyzing traffic according to a default policy and output the results in `$opt/zeek/logs/current` directory.





## zeek diag : compressing and storing the log files

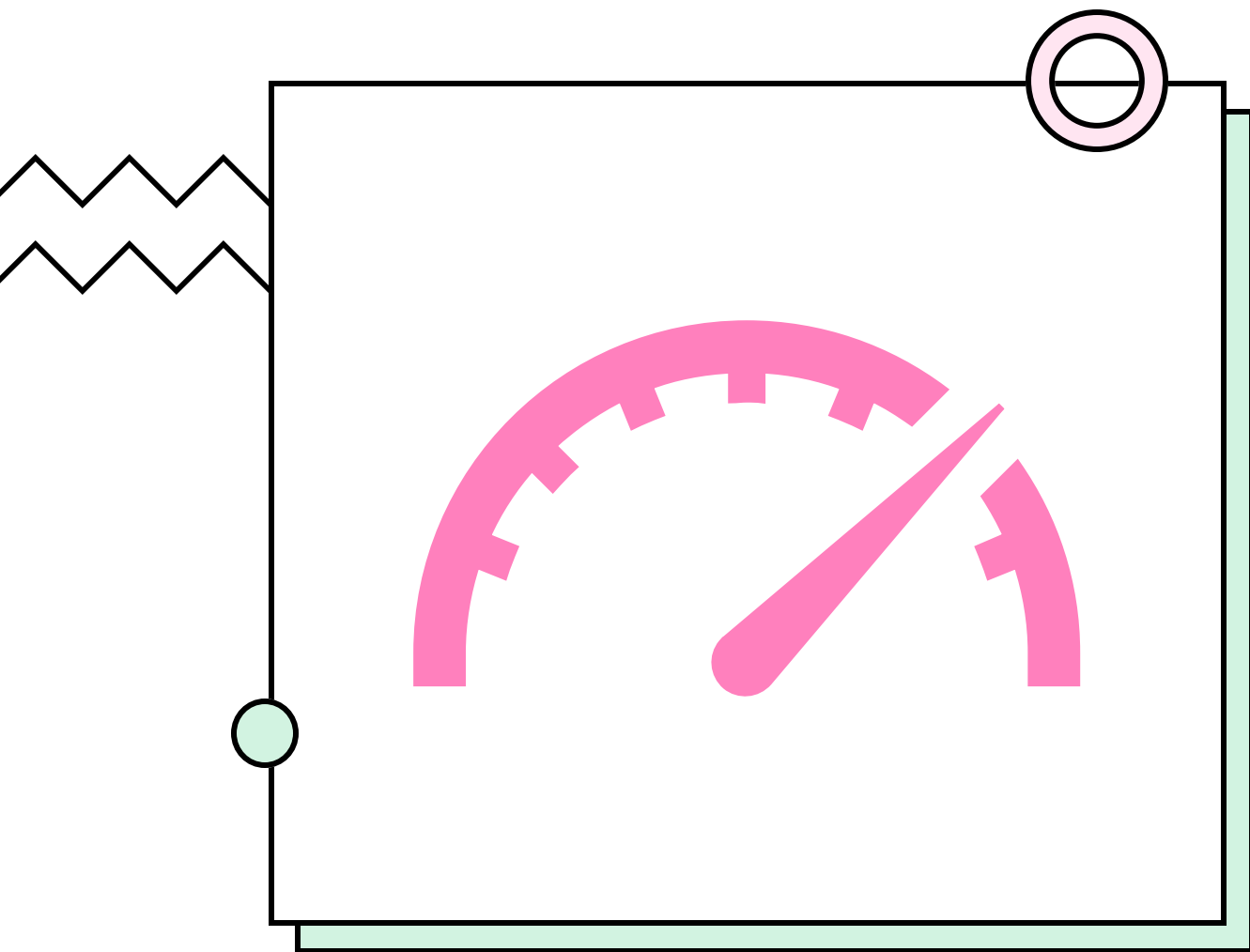
```
ime@ime-VirtualBox:/opt/zeek/bin$ cd ..
ime@ime-VirtualBox:/opt/zeek$ ls
bin  etc  include  lib  logs  share  spool  var
ime@ime-VirtualBox:/opt/zeek$ cd logs
ime@ime-VirtualBox:/opt/zeek/logs$ ls
2023-08-29  2023-08-31  2023-09-06  2023-09-07  current
ime@ime-VirtualBox:/opt/zeek/logs$
```

Here the current directory is empty before stopping the zeek instance using zeekctl. There are folders named with the date when the folder was created.

The log files in the `$opt/zeek/logs/current` directory are compressed and moved into the current day named folder **"2023-09-07"** inside the `$opt/zeek/logs` directory.

```
ime@ime-VirtualBox:/opt/zeek/logs$ ls 2023-09-07
capture_loss.01:00:00-01:37:22.log.gz  known_services.01:00:00-01:37:22.log.gz
capture_loss.01:37:22-01:37:47.log.gz  notice.01:00:00-01:37:22.log.gz
conn.01:00:00-01:37:22.log.gz           notice.01:37:22-01:37:47.log.gz
conn.01:37:22-01:37:47.log.gz           ntp.01:00:00-01:37:22.log.gz
conn-summary.01:00:00-01:37:22.log.gz   reporter.01:00:00-01:37:22.log.gz
conn-summary.01:37:22-01:37:47.log.gz   ssl.01:00:00-01:37:22.log.gz
dhcp.01:00:00-01:37:22.log.gz           stats.01:00:00-01:37:22.log.gz
dns.01:00:00-01:37:22.log.gz            stats.01:37:22-01:37:47.log.gz
http.01:00:00-01:37:22.log.gz           weird.01:00:00-01:37:22.log.gz
```

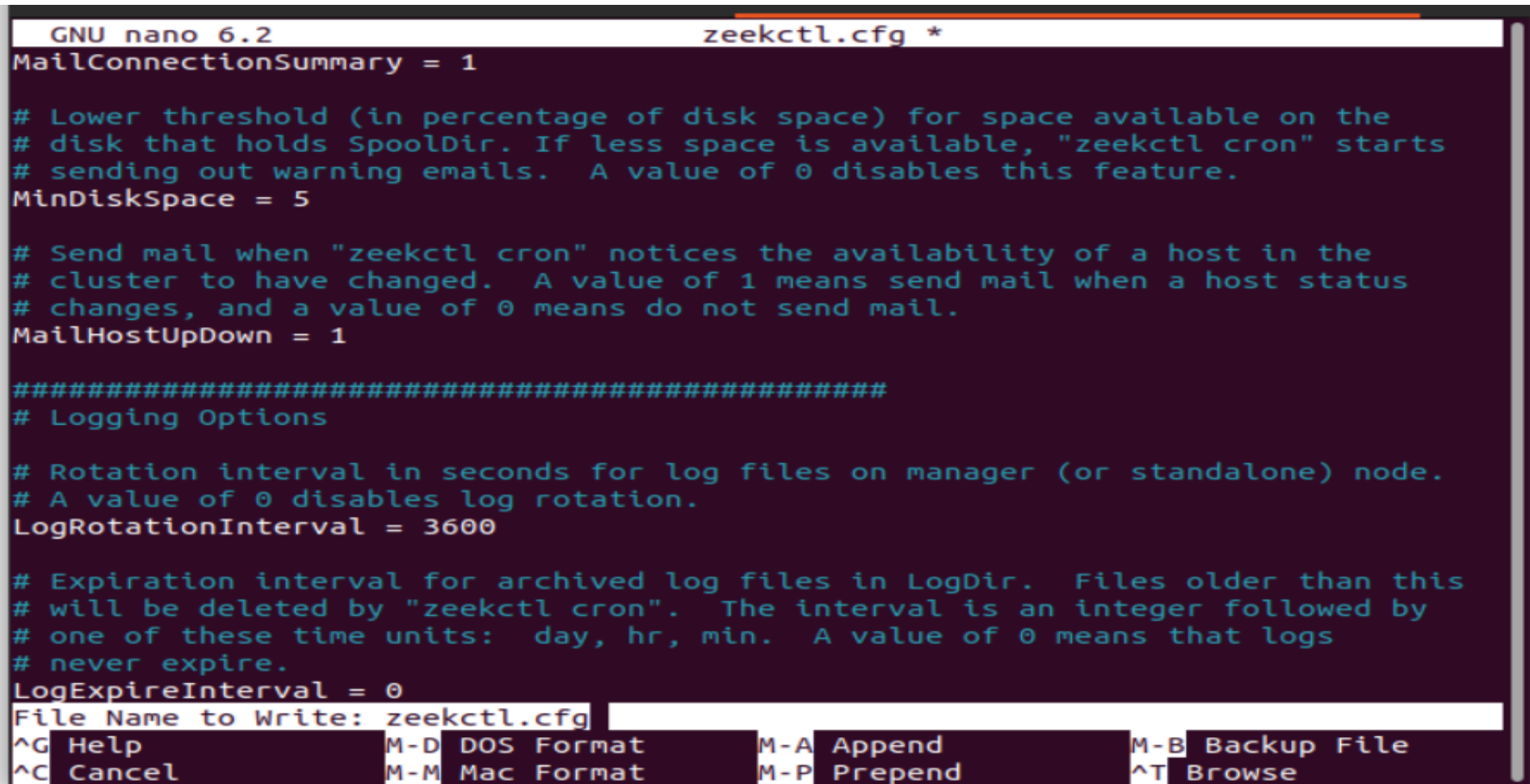




NOTIFICATION  
AND LOG  
FILE UPDATE



Open zeekctl.cfg > **nano zeekctl.cfg**



```
GNU nano 6.2 zeekctl.cfg *
MailConnectionSummary = 1

# Lower threshold (in percentage of disk space) for space available on the
# disk that holds SpoolDir. If less space is available, "zeekctl cron" starts
# sending out warning emails. A value of 0 disables this feature.
MinDiskSpace = 5

# Send mail when "zeekctl cron" notices the availability of a host in the
# cluster to have changed. A value of 1 means send mail when a host status
# changes, and a value of 0 means do not send mail.
MailHostUpDown = 1

#####
# Logging Options

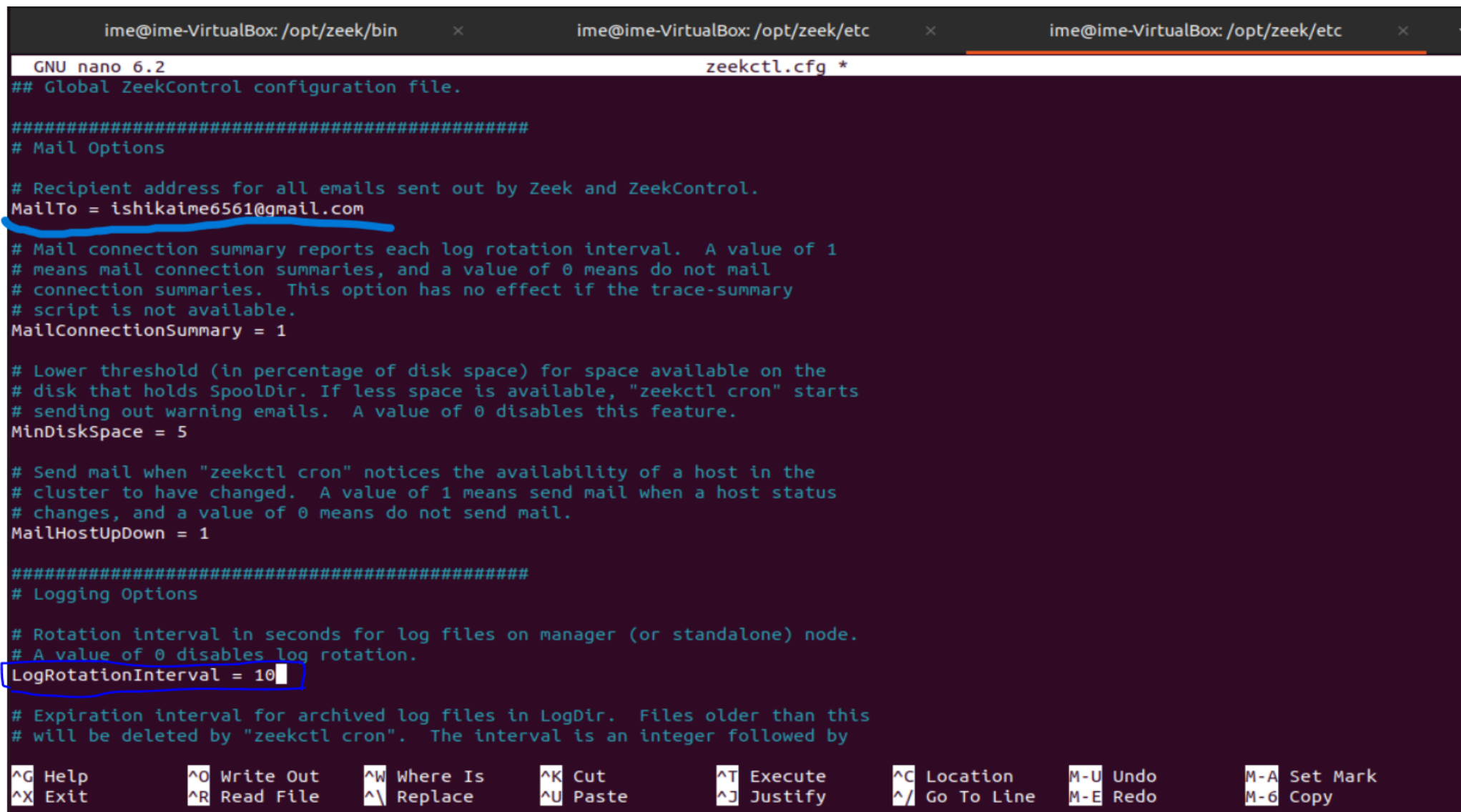
# Rotation interval in seconds for log files on manager (or standalone) node.
# A value of 0 disables log rotation.
LogRotationInterval = 3600

# Expiration interval for archived log files in LogDir. Files older than this
# will be deleted by "zeekctl cron". The interval is an integer followed by
# one of these time units: day, hr, min. A value of 0 means that logs
# never expire.
LogExpireInterval = 0
File Name to Write: zeekctl.cfg
^G Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend    ^T Browse
```

- MailTo:** We can change the email address to desired email address
- LogRotationInterval:** We can Adjust the log archival frequency to the desired value. This value is typically specified in seconds.



**MailTo : [ishikaime6561@gmail.com](mailto:ishikaime6561@gmail.com) and LogRotationInterval= 10**



The image shows a terminal window with three tabs: 'ime@ime-VirtualBox: /opt/zeek/bin', 'ime@ime-VirtualBox: /opt/zeek/etc', and 'ime@ime-VirtualBox: /opt/zeek/etc'. The active tab is the third one, showing the 'zeekctl.cfg' file being edited with 'GNU nano 6.2'. The file content is as follows:

```
## Global ZeekControl configuration file.

#####
# Mail Options

# Recipient address for all emails sent out by Zeek and ZeekControl.
MailTo = ishikaime6561@gmail.com

# Mail connection summary reports each log rotation interval. A value of 1
# means mail connection summaries, and a value of 0 means do not mail
# connection summaries. This option has no effect if the trace-summary
# script is not available.
MailConnectionSummary = 1

# Lower threshold (in percentage of disk space) for space available on the
# disk that holds SpoolDir. If less space is available, "zeekctl cron" starts
# sending out warning emails. A value of 0 disables this feature.
MinDiskSpace = 5

# Send mail when "zeekctl cron" notices the availability of a host in the
# cluster to have changed. A value of 1 means send mail when a host status
# changes, and a value of 0 means do not send mail.
MailHostUpDown = 1

#####
# Logging Options

# Rotation interval in seconds for log files on manager (or standalone) node.
# A value of 0 disables log rotation.
LogRotationInterval = 10

# Expiration interval for archived log files in LogDir. Files older than this
# will be deleted by "zeekctl cron". The interval is an integer followed by
```

The bottom of the terminal shows the nano editor's command palette with the following options:

<b>^G</b> Help	<b>^O</b> Write Out	<b>^W</b> Where Is	<b>^K</b> Cut	<b>^T</b> Execute	<b>^C</b> Location	<b>M-U</b> Undo	<b>M-A</b> Set Mark
<b>^X</b> Exit	<b>^R</b> Read File	<b>^_</b> Replace	<b>^U</b> Paste	<b>^J</b> Justify	<b>^/_</b> Go To Line	<b>M-E</b> Redo	<b>M-6</b> Copy



Go to define a rule:

**cat /opt/zeek/share/zeek/site/local.zeek**

```
ime@ime-VirtualBox:/opt/zeek/share/zeek$ ls
base  builtin-plugins  cmake  policy  python  site  test-all-policy.zeek  zeekctl  zeekygen
ime@ime-VirtualBox:/opt/zeek/share/zeek$ cd site
ime@ime-VirtualBox:/opt/zeek/share/zeek/site$ ls
local.zeek
ime@ime-VirtualBox:/opt/zeek/share/zeek/site$ cat local.zeek
##! Local site policy. Customize as appropriate.
##!
##! This file will not be overwritten when upgrading or reinstalling!

# Installation-wide salt value that is used in some digest hashes, e.g., for
# the creation of file IDs. Please change this to a hard to guess value.
redef digest_salt = "Please change this value.";

# This script logs which scripts were loaded during each run.
@load misc/loaded-scripts

# Apply the default tuning scripts for common tuning settings.
@load tuning/defaults

# Estimate and log capture loss.
@load misc/capture-loss

# Enable logging of memory, packet and lag statistics.
@load misc/stats
```

Inside local.zeek, all the policies of zeek are defined. We can define a rule and trigger an event that will alert us through email system.







## Define a rule:

```
# Extend email alerting to include hostnames
@load policy/frameworks/notice/extend-email/hostnames
event http_request(c: connection, msg: http_message) {
    if (c$http$user_agent == "EvilBot/1.0") {
        NOTICE([$note=HTTP::Request, $msg="HTTP request with EvilBot/1.0 User-Agent detected", $conn=c]);
    }
}
global notice_policy: table[string] of Notice::Info = {
    [HTTP::Request] = {
        $action = Notice::ACTION_EMAIL,
        $priority = Notice::PRIORITY_HIGH,
        $group = "HTTP",
        $email_subject = "Zeek Alert: HTTP Request with EvilBot/1.0 User-Agent",
        $email_body = "Zeek detected an HTTP request with the EvilBot/1.0 User-Agent.\n\nConnection details:\n%conn\n",
    },
};
```

To trigger an alert email with the modified Zeek policy and the changed email address, a network event needs to be created that matches the updated rule in the Zeek policy.

In this case, the modified policy is to generate an alert when an HTTP request contains the user agent string **"EvilBot/1.0"**





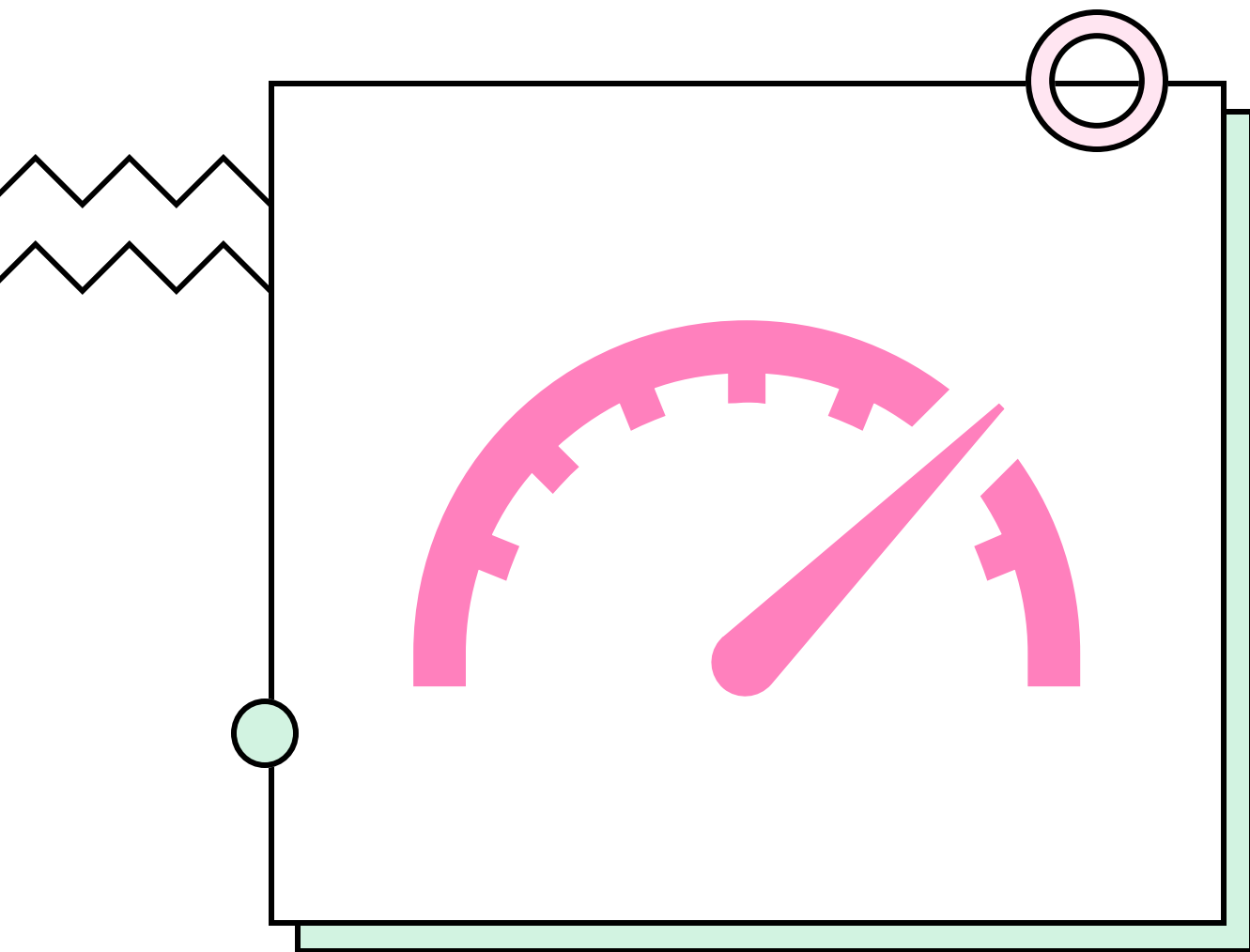
curl -A "Evilbot/1.0" http://google.com

```
</HTML>
ime@ime-VirtualBox:/$ curl -A "EvilBot/1.0" http://google.com
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
```

Zeek will send an email notification to the updated email address specified in the **MailTo** setting (In our case, [ishikaime6561@gmail.com](mailto:ishikaime6561@gmail.com)) of Zeek configuration.







B R O W S I N G  
L O G F I L E S





## Finding log files in /opt/zeek/bin

```
ime@ine-VirtualBox:/opt/zeek/bin$ ls
adtrace          btest-progress  mytrace.zeek    spicyz
bifcl            btest-rst-cmd   ntp.log          ssl.log
binpac           btest-rst-include ocsf.log         trace-summary
bro              btest-rst-pipe  packet_filter.log weird.log
bro-config       btest-setsid    pcap             zeek
broctl           conn.log        reporter.log     zeek-archiver
bro-cut          dhcp.log        rst              zeek-client
btest            dns.log         spicy-build      zeek-config
btest-ask-update files.log        spicyc           zeekctl
btest-bg-run     gen-zam         spicy-config     zeek-cut
btest-bg-run-helper hiltic         spicy-driver     zeek-wrapper
btest-bg-wait    hilti-config    spicy-dump       zkg
btest-diff        http.log        spicy-precompile-headers
```

Here all log files are located, conn.log, dhcp.log, ssl.log, notice.log, known\_services.log, dns.log, weird.log etc.



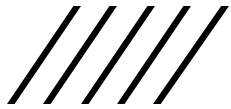


# cat http.log

```
ime@ime-VirtualBox:/opt/zeek/bin$ cat http.log
#separator \x09
#set_separator ,
#empty_field (empty)
#unset_field -
#path http
#open 2023-09-06-21-55-08
#fields ts uid id.orig_h id.orig_p id.resp_h id.resp_p trans_depth method host uri refe
rrer version user_agent origin request_body_len response_body_len status_code status_msg info_code i
nfo_msg tags username password proxied orig_fuids orig_filenames orig_mime_types resp_fuids resp_filenam
es resp_mime_types
#types time string addr port addr port count string string string string string string string count coun
t count string count string set[enum] string string set[string] vector[string] vector[string] vector[strin
g] vector[string] vector[string] vector[string]
1694015708.113351 Ceq4j12N99VFqAEuLi 10.0.2.15 40124 35.224.170.84 80 1 - - - - - 1
.1 - - 0 0 204 No Content - - (empty) - - - - -
- - - - - - - - - - - - - - - -
1694016009.028028 CaQv274dolInVdnZG6 10.0.2.15 46494 185.125.190.17 80 1 - - - - - 1
.1 - - 0 0 204 No Content - - (empty) - - - - -
- - - - - - - - - - - - - - - -
```

**http.log** is one of the log files generated by Zeek (formerly known as Bro) during network traffic analysis. This log file contains detailed information about HTTP (Hypertext Transfer Protocol) traffic observed on the network.

Logs that deal with analysis of a network protocol will often start like this: a timestamp, a unique connection identifier (UID), and a connection **4-tuple (originator host/port and responder host/port)**. The UID can be used to identify and correlate all logged activity (possibly across multiple log files) associated with a given connection 4-tuple over its lifetime.





## **conn.log**

Contains an entry for every connection seen on the wire, with basic properties such as time and duration, originator and responder IP addresses, services and ports, payload size, and much more. This log provides a comprehensive record of the network's activity.

## **notice.log**

Identifies specific activity that Zeek recognizes as potentially interesting, odd, or bad. In Zeek-speak, such activity is called a "notice".

## **known\_services.log**

This log file contains the services detected on the local network and are known to be actively used by the clients on the network. It helps in enumerating what all services are observed on a local network and if they all are intentional and known to the network administrator.

## **weird.log**

Contains unusual or exceptional activity that can indicate malformed connections, traffic that doesn't conform to a particular protocol, malfunctioning or misconfigured hardware/services, or even an attacker attempting to avoid/confuse a sensor.



**THANK  
YOU**

