comment-installer-zeek-network-security-monitoring-sur-debian-12

Zeek (formerly Bro) is a free and open-source platform for network security monitoring. It is a powerful passive network traffic analyzer to investigate suspicious or malicious activity. Zeek can be used as a network security monitor (NSM) and supports a wide range of traffic analysis, from the security domain to performance measurement and troubleshooting.

In this guide, I will show you how to install Zeek network Security Monitoring on the Debian 12 server step-by-step. You will install Zeek, and configure Zeek in cluster mode, then you will learn how to parse Zeek TSV log format via the zeek-cut command line. Furthermore, you will learn how to set up Zeek log output as JSON and parser Zeek JSON log via the jq command line.

Prerequisites

Before commencing, confirm that you have the following:

- A Debian 12 server.
- A non-root user with administrator privileges.

Adding Repository

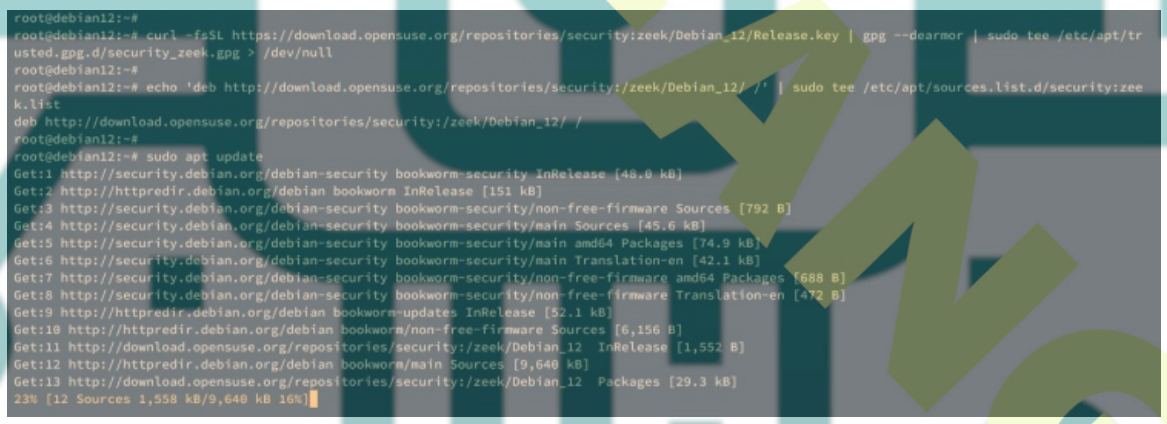
Zeek can be installed on the Linux system by compiling it manually or by using a third-party repository. In this guide, you will install Zeek using a third-party repository via APT.

First, run the following command to add the GPG key and repository for the Zeek package.

```
curl -fsSL https://download.opensuse.org/repositories/security:zeek/Debian_12/Release.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/security_zeek.gpg > /dev/null
echo 'deb http://download.opensuse.org/repositories/security:/zeek/Debian_12/ /' | sudo tee /etc/apt/sources.list.d/security:zeek.list
```

Now update and refresh your Debian repository by executing the following apt update command.

```
sudo apt update
```



Installing Zeek

After adding the Zeek repository, you can now start Zeek installation on your Debian machine. In the following step, you will install Zeek and add the Zeek installation directory to the system PATH.

Execute the apt install command below to install Zeek to your Debian machine. There are multiple versions of Zeek available, such as the *latest release, LTS, and nightly*. In this example, you will install *zeek-lts*.

```
sudo apt install zeek-lts
```

Type y to proceed with the installation.

```
root@debian12:~#
root@debian12:~# sudo apt install zeek-lts
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  dbus dbus-bin dbus-daemon dbus-session-bus-common dbus-system-bus-common dbus-user-session fontconfig-config fonts-dejavu-core git git-man
  javascript-common libabsl20220623 libaom3 libavif15 libbroker-lts-dev libc-ares2 libc-bin libc-dev-bin libc-devtools libc-l10n libc6
  libc6-dev libcrypt-dev libdav1d6 libdbus-1-3 libdbus-1-dev libde265-0 libdeflate0 liberror-perl libfontconfig1 libgav1-1 libgd3 libheif1
  libjbig0 libjpeg62-turbo libjs-jquery liblerc4 liblockfile1 libmaxminddb-dev libnsl-dev libnuma1 libpcap-dev libpcap0.8 libpcap0.8-dev
  libpkgconf3 librav1e0 libssl-dev libsvtav1enc1 libtiff6 libtirpc-dev libwebp7 libx11-6 libx11-data libx265-199 libxau6 libxcb1 libxdmcp6
  libxpm4 libyaml-0-2 libyuv0 linux-libc-dev locales lockfile-progs m4 make manpages-dev patch pkg-config pkgconf pkgconf-bin
  postfix-mta-sts-resolver python3-aiodns python3-aiohttp python3-aioredis python3-aiosignal python3-aiosqlite python3-async-timeout
  python3-attr python3-cffi-backend python3-deprecated python3-frozenlist python3-git python3-gitdb python3-hiredis python3-multidict
  python3-packaging python3-pycares python3-redis python3-semantic-version python3-smmap python3-typing-extensions python3-uvloop python3-wrapt
  python3-yaml python3-yarl rpcsvc-proto sendmail-base sendmail-bin sendmail-cf sgml-base xml-core zeek-lts-btest zeek-lts-btest-data
  zeek-lts-client zeek-lts-core zeek-lts-core-dev zeek-lts-spicy-dev zeek-lts-zkg zeekctl-lts zlib1g-dev
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn apache2 | lighttpd | httpd glibc-doc
  libnss-nis libnss-nisplus libgd-tools libssl-doc m4-doc make-doc ed diffutils-doc python-aiosqlite-doc python-attr-doc python-git-doc
  redis-server python-pycares-doc python-semantic-version-doc python3-nose sendmail-doc logcheck resolvconf sasl2-bin sgml-base-doc debhelper
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core git git-man javascript-common libabsl20220623 libaom3 libavif15 libbroker-lts-dev libc-ares2 libc-dev-bin
  libc-devtools libc6-dev libcrypt-dev libdav1d6 libdbus-1-dev libde265-0 libdeflate0 liberror-perl libfontconfig1 libgav1-1 libgd3 libheif1
  libjbig0 libjpeg62-turbo libjs-jquery liblerc4 liblockfile1 libmaxminddb-dev libnsl-dev libnuma1 libpcap-dev libpcap0.8 libpcap0.8-dev
  libpkgconf3 librav1e0 libssl-dev libsvtav1enc1 libtiff6 libtirpc-dev libwebp7 libx11-6 libx11-data libx265-199 libxau6 libxcb1 libxdmcp6
  libxpm4 libyaml-0-2 libyuv0 linux-libc-dev lockfile-progs m4 make manpages-dev patch pkg-config pkgconf pkgconf-bin postfix-mta-sts-resolver
  python3-aiodns python3-aiohttp python3-aioredis python3-aiosignal python3-aiosqlite python3-async-timeout python3-attr python3-cffi-backend
  python3-deprecated python3-frozenlist python3-git python3-gitdb python3-hiredis python3-multidict python3-packaging python3-pycares
  python3-redis python3-semantic-version python3-smmap python3-typing-extensions python3-uvloop python3-wrapt python3-yaml python3-yarl
  rpcsvc-proto sendmail-base sendmail-bin sendmail-cf sgml-base xml-core zeek-lts zeek-lts-btest zeek-lts-btest-data zeek-lts-client
  zeek-lts-core zeek-lts-core-dev zeek-lts-spicy-dev zeek-lts-zkg zeekctl-lts zlib1g-dev
The following packages will be upgraded:
  dbus dbus-bin dbus-daemon dbus-session-bus-common dbus-system-bus-common dbus-user-session libc-bin libc-l10n libc6 libdbus-1-3 locales
11 upgraded, 100 newly installed, 0 to remove and 23 not upgraded.
Need to get 124 MB of archives.
After this operation, 699 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

After zeek is installed, which is by default to the target directory */opt/zeek*. Execute the following command to add the */opt/zeek/bin* directory to the system **PATH** via *~/.bashrc* file.

```
echo "export PATH=$PATH:/opt/zeek/bin" >> ~/.bashrc
```

Next, reload the *~/.bashrc* file and check the system **PATH** variable using the following command. You should see the */opt/zeek/bin* directory within the system PATH.

```
source ~/.bashrc
echo $PATH
```

```
root@debian12:~#
root@debian12:~# ls /opt/zeek/
bin  etc  include  lib  logs  share  spool  var
root@debian12:~#
root@debian12:~# echo "export PATH=$PATH:/opt/zeek/bin" >> ~/.bashrc
root@debian12:~#
root@debian12:~# source ~/.bashrc
root@debian12:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/opt/zeek/bin
root@debian12:~#
```

Lastly, verify Zeek by executing the following command. With this, you will check the location of the Zeek binary file, check the Zeek version, and print the Zeek help message.

```
which zeek
zeek --version
zeek --help
```

In the following output, you should see that zeek **5.0** is installed at */usr/zeek/bin/zeek*. Also, you should options for the Zeek command.

```
root@debian12:~#
root@debian12:~# which zeek
/opt/zeek/bin/zeek
root@debian12:~#
root@debian12:~# zeek --version
zeek version 5.0.10
root@debian12:~#
root@debian12:~# zeek --help
zeek version 5.0.10
usage: zeek [options] [file ...]
usage: zeek --test [doctest-options] -- [options] [file ...]
    <file>                          | Zeek script file, or read stdin
    -a|--parse-only                 | exit immediately after parsing scripts
    -b|--bare-mode                  | don't load scripts from the base/ directory
    -c|--capture-unprocessed <file> | write unprocessed packets to a tcpdump file
    -d|--debug-script               | activate Zeek script debugging
    -e|--exec <zeek code>           | augment loaded scripts by given code
    -f|--filter <filter>            | tcpdump filter
    -h|--help                       | command line help
    -i|--iface <interface>          | read from given interface (only one allowed)
    -p|--prefix <prefix>            | add given prefix to Zeek script file resolution
    -r|--readfile <readfile>        | read from given tcpdump file (only one allowed, pass '-
    -s|--rulefile <rulefile>        | read rules from given file
    -t|--tracefile <tracefile>      | activate execution tracing
    -u|--usage-issues               | find variable usage issues and exit
       --no-unused-warnings          | suppress warnings of unused functions/hooks/events
    -v|--version                    | print version and exit
    -w|--writefile <writefile>      | write to given tcpdump file
    -C|--no-checksums               | ignore checksums
    -D|--deterministic              | initialize random seeds to zero
```

Configuring Zeek

Now that Zeek is installed, the next step is to configure Zeek installation. You can run Zeek in multiple modes, such as command-line mode, standalone mode, and cluster mode.

In the following example, you will learn how to run Zeek in cluster mode using a single server.

Before configuring Zeek, execute the following command to check your network interfaces and IP address.

```
ip a
```

You should the list available interfaces on your system with detailed information on a server IP address like the following:

```
root@debian12:~#
root@debian12:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNO
    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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_code
    link/ether 08:00:27:5f:91:71 brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic eth0
       valid_lft 81826sec preferred_lft 81826sec
    inet6 fe80::a00:27ff:fe5f:9171/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_code
    link/ether 08:00:27:86:7c:80 brd ff:ff:ff:ff:ff:ff
    altname enp0s8
    inet 192.168.10.15/24 brd 192.168.10.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe86:7c80/64 scope link
       valid_lft forever preferred_lft forever
root@debian12:~#
root@debian12:~#
```

Open the network configuration for Zeek */opt/zeek/etc/networks.cfg* using the following nano editor command.

```
sudo nano /opt/zeek/etc/networks.cfg
```

Insert your internal network subnet like the following. You can also add multiple subnets to it.

```
10.0.0.0/8          Private IP space
172.16.0.0/12       Private IP space
192.168.0.0/16      Private IP space
```

Save the file and exit the editor when you're done.

Now open zeek configuration */opt/zeek/etc/node.cfg* using the nano editor command below.

```
sudo nano /opt/zeek/etc/node.cfg
```

By default, Zeek is running in a standalone mode. Add the **#** to disable Zeek standalone mode.

```
#[zeek]
#type=standalone
#host=localhost
#interface=eth0
```

Insert the following configuration to run Zeek in the cluster mode with a single server. Be sure to change the server IP address with your information.

With the following configuration, you will be running Zeek in cluster mode, which has multiple components, such as *zeek-logger, zeek-manager, zeek-proxy, and zeek-worker*.

```
# logger
[zeek-logger]
type=logger
host=192.168.10.15

# manager
[zeek-manager]
type=manager
host=192.168.10.15

# proxy
[zeek-proxy]
type=proxy
host=192.168.10.15

# worker
[zeek-worker]
type=worker
host=192.168.10.15
interface=eth0

# worker localhost
[zeek-worker-lo]
type=worker
host=localhost
interface=lo
```

Save and close the fine when finished.

Next, run the following command to access the Zeek control shell.

```
zeekctl
```

Run the check command to validate your Zeek configuration.

```
check
```

If everything goes well, you should see each component of the Zeek script is ok:

```
Hint: Run the zeekctl "deploy" command to get started.

Welcome to ZeekControl 2.4.1

Type "help" for help.

[ZeekControl] > check
zeek-logger scripts are ok.
zeek-manager scripts are ok.
zeek-proxy scripts are ok.
zeek-worker scripts are ok.
zeek-worker-lo scripts are ok.
[ZeekControl] >
```

Next, run the *deploy* command to start and run Zeek on your machine. The *deploy* command is equivalent to the *install* and *start* command on Zeek.

```
deploy
```

You should see each component of the Zeek cluster is starting:

```
[ZeekControl] >
[ZeekControl] > deploy
checking configurations ...
installing ...
creating policy directories ...
installing site policies ...
generating cluster-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping workers ...
stopping proxy ...
stopping manager ...
stopping logger ...
starting ...
starting logger ...
starting manager ...
starting proxy ...
starting workers ...
[ZeekControl] >
[ZeekControl] >
```

Lastly, verify each component of your Zeek cluster by executing the status command below.

```
status
```

If your installation is successful, you should see each of the Zeek cluster components is running.

```
[ZeekControl] >
[ZeekControl] > status
Name          Type    Host             Status    Pid    Started
zeek-logger   logger  192.168.10.15    running   6760
zeek-manager  manager 192.168.10.15    running   6810
zeek-proxy    proxy   192.168.10.15    running   6864
zeek-worker   worker  192.168.10.15    running   6928
zeek-worker-lo worker  localhost        running   6935
[ZeekControl] >
```

Type *exit* to log out from the Zeek control shell.

At this point, the Zeek cluster is running. It also monitors the target network interface and subnet and generates log files to the */opt/zeek/logs* directory.

Guide to Zeek Logs

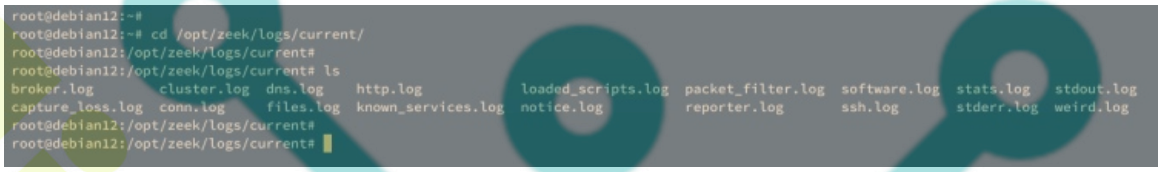
After configuring Zeek, the next step is to explore log files that are generated by Zeek, which is located at */opt/zeek/logs/current* directory. By default, zeek generates log files with the TSV (Tab-separated values) format.

When Zeek is running, it will monitor the target network interface on your system and generate log files to */opt/zeek/logs/current/* directory.

Move your working directory to */opt/zeek/logs/current/* directory and run the ls command below.

```
cd /opt/zeek/logs/current/
ls -ah
```

You should see multiple log files generated by Zeek. You may see some log files are missing on your system because the target service is not available.



Below are some of the important log files that you must know:

- **conn.log**: The connection log for both TCP and UDP. This log file provides the most useful information such as timestamp, connection duration, service, and many more.
- **dns.log**: The DNS (Domain Name System) log.
- **http.log**: The HyperText Transfer Protocol (HTTP) log.
- **ssh.log**: The Secure Shell (SSH) log for tracking SSH connections.
- **ssl.log**: The Secure Socket Layer (SSL) log that also contains the HTTPS log.

Analyzing Zeek Logs TSV (Tab-separated values) via Zeek-cut

By default, zeek generates log files with TSV (Tab-separated values) format. In the following step, you will analyze Zeek log files with TSV format via the zeek-cut command line.

Execute the cat command below to view the log file dns.log.

```
cat dns.log
```

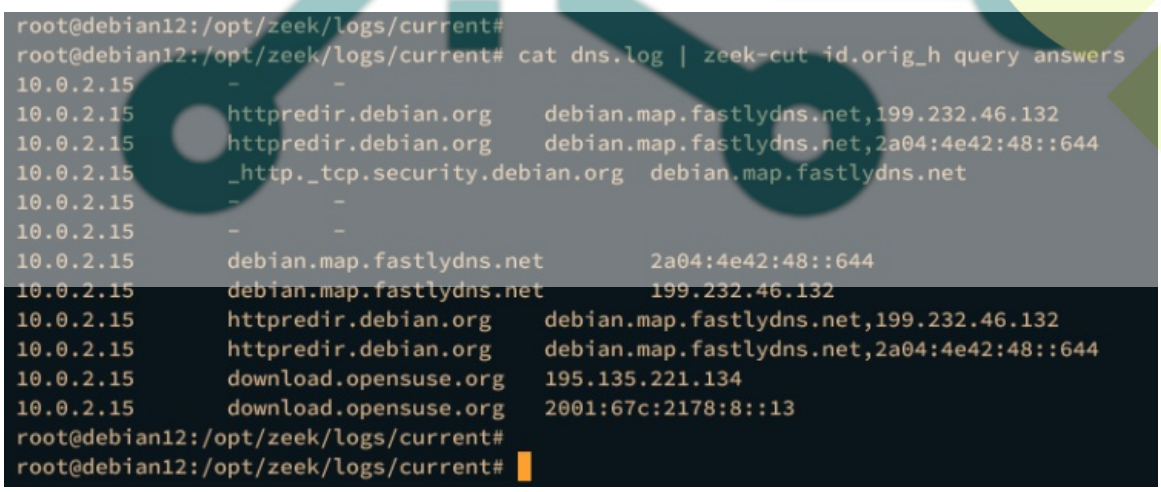
In the following output, you should see multiple fields such as *ts, uid, id.orig_p, id.resp_h, id.resp_p, proto*, and many more.

Next, execute the following command to parse the Zeek TSV log format. With this, you will send the output via pipe "|" to the *zeek-cut* command.

In this example, you will three fields from the log file, such as *id.orig_h, query,* and *answers*.

```
cat dns.log | zeek-cut id.orig_h query answers
cat dns.log | zeek-cut query answers id.orig_h
```

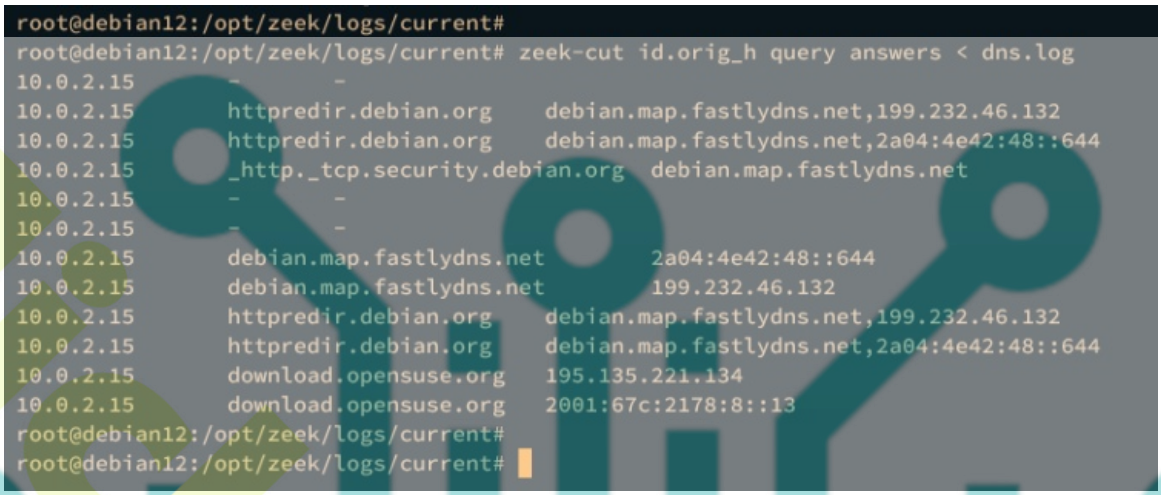
You should see the similar output like the following:



You can also use the redirect symbol to process the TSV log file via zeek-cut like the following command.

```
zeek-cut id.orig_h query answers < dns.log
```

The output should be similar.



Configuring Zeek Log Files to JSON

In the following step, you will configure Zeek to generate output log files with JSON format. To achieve that, you must modify *local.zeek* file and load the zeek script tuning/json-logs to your zeek installation.

Open the file */opt/zeek/share/zeek/site/local.zeek* using the following nano editor command.

```
sudo nano /opt/zeek/share/zeek/site/local.zeek
```

Insert the following configuration to the bottom of the line.

```
@load tuning/json-logs
```

Save and close the file when you're done.

Now run the zeekctl command below to redeploy your zeek installation.

```
zeekctl deploy
```

You should see zeek is now reinstalling. Once the process is finished, zeek will generate JSON log files.

Before analyzing the JSON log format, install jq to your machine by executing the following apt command.

```
sudo apt install jq -y
```

```
root@debian12:/opt/zeek/logs/current#
root@debian12:/opt/zeek/logs/current# sudo apt install jq -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libjq1 libonig5
The following NEW packages will be installed:
  jq libjq1 libonig5
0 upgraded, 3 newly installed, 0 to remove and 23 not upgraded.
Need to get 388 kB of archives.
After this operation, 1,165 kB of additional disk space will be used.
Get:1 http://httpredir.debian.org/debian bookworm/main amd64 libonig5 amd64 6.9.8-1 [188 kB]
Get:2 http://httpredir.debian.org/debian bookworm/main amd64 libjq1 amd64 1.6-2.1 [135 kB]
Get:3 http://httpredir.debian.org/debian bookworm/main amd64 jq amd64 1.6-2.1 [64.9 kB]
```

Once jq is installed, move to the */opt/zeek/logs/current/* directory. The directory */opt/zeek/logs/current* contains zeek log files in JSON format, and it's automatically generated by zeek.

```
cd /opt/zeek/logs/current/
```

Run the cat command below to view the log file dns.

```
cat dns.log
```

The JSON output will be displayed on your terminal screen.

Next, run the jq command below to process the JSON log dns.log.

```
jq . dns.log
```

```
root@debian12:/opt/zeek/logs/current#
root@debian12:/opt/zeek/logs/current# jq . dns.log
{
  "ts": 1696390143.652125,
  "uid": "CToVhK2jj5XSZG3bG4",
  "id.orig_h": "10.0.2.15",
  "id.orig_p": 56922,
  "id.resp_h": "10.0.2.3",
  "id.resp_p": 53,
  "proto": "udp",
  "trans_id": 16776,
  "rcode": 3,
  "rcode_name": "NXDOMAIN",
  "AA": false,
  "TC": false,
  "RD": false,
  "RA": false,
  "Z": 0,
  "rejected": false
}
{
  "ts": 1696390143.739642,
  "uid": "CA7HpO38JdJyDEvcgc",
  "id.orig_h": "10.0.2.15",
  "id.orig_p": 53920,
  "id.resp_h": "10.0.2.3",
  "id.resp_p": 53,
  "proto": "udp",
  "trans_id": 37313,
  "query": "_http._tcp.security.debian.org",
  "rcode": 0,
  "rcode_name": "NOERROR",
  "AA": false,
  "TC": false,
  "RD": false,
  "RA": true,
  "Z": 0,
  "answers": [
    "debian.map.fastlydns.net"
  ],
  "TTLs": [
    900
  ],
  "rejected": false
}
{
```


Or you can display the compact format via the *-c* option like the following.

```
jq . -c dns.log
```

Lastly, execute the following command to display a specific key/value from the JSON file dns.log.

```
jq -c '["id.orig_h", ."query", ."answers"]' dns.log
```

You should see the output like the following.

```
root@debian12:/opt/zeek/logs/current#
root@debian12:/opt/zeek/logs/current# jq -c '[."id.orig_h", ."query", ."answers"]' dns.log
["10.0.2.15",null,null]
["10.0.2.15","_http._tcp.security.debian.org",["debian.map.fastlydns.net"]]
["10.0.2.15",null,null]
["10.0.2.15","httpredir.debian.org",["debian.map.fastlydns.net","199.232.46.132"]]
["10.0.2.15","httpredir.debian.org",["debian.map.fastlydns.net","2a04:4e42:48::644"]]
["10.0.2.15","debian.map.fastlydns.net",["199.232.46.132"]]
["10.0.2.15","debian.map.fastlydns.net",["2a04:4e42:48::644"]]
["10.0.2.15","download.opensuse.org",["2001:67c:2178:8::13"]]
["10.0.2.15","download.opensuse.org",["195.135.221.134"]]
["10.0.2.15","google.com",["142.251.175.101","142.251.175.138","142.251.175.139","142.251.175.113","142.251.175.102","142.251.175.100"]]
["10.0.2.15","google.com",["2404:6800:4003:c1c::8b","2404:6800:4003:c1c::71","2404:6800:4003:c1c::64","2404:6800:4003:c1c::65"]]
["10.0.2.15","debian.com",["2603:400a:ffff:bb8::801f:3e","2001:67c:2564:a119::77"]]
["10.0.2.15","debian.com",["130.89.148.77","128.31.0.62"]]
["10.0.2.15","debian.org",["151.101.2.132","151.101.130.132","151.101.194.132","151.101.66.132"]]
["10.0.2.15","debian.org",["2a04:4e42:400::644","2a04:4e42::644","2a04:4e42:200::644","2a04:4e42:600::644"]]
["10.0.2.15",null,null]
root@debian12:/opt/zeek/logs/current#
```

Conclusion

Congratulations! You've now successfully installed the Zeek network monitoring tool on the Debian 12 server. You've installed Zeek, run Zeek in the cluster mode, learned some zeek log files, and also learned how to parse zeek log files with TSV format via zeek-cut. Furthermore, you've also changed the zeek log to JSON and learned how to parse Zeek log JSON format via jq command lines.
