Zeek

1. Install Zeek.

```
root@ntsapi:/opt/zeek/bin# ./zeek --version
./zeek version 6.0.5
root@ntsapi:/opt/zeek/bin#
```

2. Run Zeek to monitor traffic on the chosen interface.

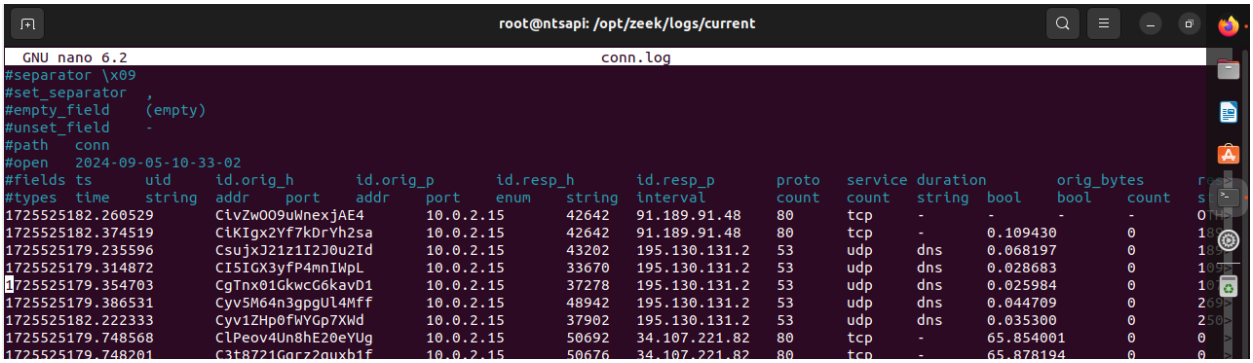
```
root@ntsapi:/opt/zeek/bin# ./zeekctl deploy
checking configurations ...
installing ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /opt/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
starting ...
starting zeek ...
root@ntsapi:/opt/zeek/bin#
```

```
root@ntsapi:/opt/zeek/etc# zeekctl status
Name         Type       Host          Status    Pid    Started
zeek         standalone localhost     running   6154   05 Sep 10:32:56
```

3. Zeek will create log files in the current directory. List the files to see the generated logs.

```
root@ntsapi:/opt/zeek/logs/current# ls
capture_loss.log  http.log            ocsp.log           stats.log      weird.log
conn.log          loaded_scripts.log  packet_filter.log  stderr.log
dns.log           notice.log          reporter.log       stdout.log
files.log         ntp.log             ssl.log            telemetry.log
root@ntsapi:/opt/zeek/logs/current#
```

4. Inspect the conn.log file. Use any text editor. Identify some key fields like ts, id.orig_h, id.resp_h, proto, duration, orig_bytes, resp_bytes.



Sample Entry 1

1725525182.260529 CivZwOO9uWnexjAE4 10.0.2.15 42642 91.189.91.48 80 tcp - - - - OTH>

ts: 1725525182.260529 (Unix timestamp)

id.orig_h: `10.0.2.15` (Client IP)

id.resp_h: `91.189.91.48` (Server IP)

proto: `tcp` (Protocol)

duration: `-` (No duration provided)

orig_bytes: `-` (No data sent by the client)

resp_bytes: `-` (No data sent by the server)

Sample Entry 2

1725525182.374519 CiKIgx2Yf7kDrYh2sa 10.0.2.15 42642 91.189.91.48 80 tcp - 0.109430 0 189>

ts: `1725525182.374519` (Unix timestamp)

id.orig_h: `10.0.2.15` (Client IP)

id.resp_h: `91.189.91.48` (Server IP)

proto: `tcp` (Protocol)

duration: `0.109430` seconds

orig_bytes: `0` (No data sent by the client)

resp_bytes: `189` (Bytes sent by the server)

5. Explore Sample Scripts. Zeek comes with several sample scripts located generally in the /usr/share/zeek/scripts directory. Explore these scripts to get a feel for the syntax and functionalities.

```
root@ntsapi:/opt/zeek/share/zeekctl/scripts# ls
archive-log   expire-logs         packet_filter.log  set-zeek-path
check-config  files.log           postprocessors     stats-to-csv
conn.log      helpers             post-terminate     test-conn.zeek
crash-diag    http.log            reporter.log       weird.log
delete-log    ipp.pcap            run-zeek           zeekctl-config.sh
dns.log       make-archive-name   run-zeek-on-trace
expire-crash  nohup.out           send-mail
root@ntsapi:/opt/zeek/share/zeekctl/scripts#
```

6. Create a Zeek script. Create a new file named test-conn.zeek. Write the script to print a message when a connection is established.

```
  GNU nano 6.2                        test-conn.zeek
# test-conn.zeek

event zeek_init() {
    print "Zeek Script initialized.";
}

event connection_established(c: connection) {
    print fmt("Connection established between %s and %s", c$id$orig_h, c$id$res>
}
```

7. Execute the script with Zeek on a pcap file or live traffic. If you don't have a pcap file, you can download a sample one from Wireshark Sample Captures or use a Wireshark pacap from your own capture.

   On live traffic:

```
root@ntsapi:/opt/zeek/share/zeekctl/scripts# zeek -i enp0s3 test-conn.zeek
listening on enp0s3

Zeek Script initialized.
1725545287.997304 warning in /opt/zeek/share/zeek/base/misc/find-checksum-offloa
ding.zeek, line 54: Your interface is likely receiving invalid TCP checksums, mo
st likely from NIC checksum offloading.  By default, packets with invalid checks
ums are discarded by Zeek unless using the -C command-line option or toggling th
e 'ignore_checksums' variable.  Alternatively, disable checksum offloading by th
e network adapter to ensure Zeek analyzes the actual checksums that are transmit
ted.
Connection established between 10.0.2.15 and 34.107.221.82
Connection established between 10.0.2.15 and 34.107.221.82
```

   On captured pcap files:

```
root@ntsapi:/opt/zeek/share/zeekctl/scripts# zeek -r ipp.pcap test-conn.zeek
Zeek Script initialized.
Connection established between 10.10.10.49 and 10.10.10.251
Connection established between 10.10.10.49 and 10.10.10.251
Connection established between 10.10.10.49 and 10.10.10.251
1210953939.492942 warning in /opt/zeek/share/zeek/base/misc/find-checksum-offloa
ding.zeek, line 54: Your trace file likely has invalid TCP checksums, most likel
y from NIC checksum offloading.  By default, packets with invalid checksums are
discarded by Zeek unless using the -C command-line option or toggling the 'ignor
e_checksums' variable.  Alternatively, disable checksum offloading by the networ
k adapter to ensure Zeek analyzes the actual checksums that are transmitted.
```

8. Modify the script test-conn.zeek script to log connection details to a custom log file.

```
  GNU nano 6.2                                                  test-conn.zeek
# test-conn.zeek

@load base/protocols/conn

module TestConn;

export {
    redef enum Log::ID += { LOG };
}

redef record Log::Info += {
    orig_h: addr &log;
    resp_h: addr &log;
    proto: string &log;
    duration: interval &log;
    orig_bytes: count &log;
    resp_bytes: count &log;
};

event zeek_init() {
    Log::create_stream(TestConn::LOG, [$columns=Log::Info, $path="test-conn"]);
}

event connection_established(c: connection) {
    local info: Log::Info = [
        $ts = network_time(),
        $orig_h = c$id$orig_h,
        $resp_h = c$id$resp_h,
        $proto = c$id$proto,
        $duration = c$duration,
        $orig_bytes = c$orig_bytes,
        $resp_bytes = c$resp_bytes
    ];
    Log::write(TestConn::LOG, info);
}
```

9. Execute the script on the same pcap file or live traffic.

```
root@ntsapi:/opt/zeek/share/zeekctl/scripts# zeek -r ipp.pcap test-conn.zeek
warning in ./test-conn.zeek, line 7: Can't generate zeekygen documentation for redef of Log::Info, identifier lookup failed
warning in ./test-conn.zeek, line 8: Can't generate zeekygen documentation for record field ts, unknown record: Log::Info
warning in ./test-conn.zeek, line 9: Can't generate zeekygen documentation for record field orig_h, unknown record: Log::Info
warning in ./test-conn.zeek, line 10: Can't generate zeekygen documentation for record field resp_h, unknown record: Log::Info
warning in ./test-conn.zeek, line 11: Can't generate zeekygen documentation for record field orig_port, unknown record: Log::Info
warning in ./test-conn.zeek, line 12: Can't generate zeekygen documentation for record field resp_port, unknown record: Log::Info
warning in ./test-conn.zeek, line 13: Can't generate zeekygen documentation for record field proto, unknown record: Log::Info
warning in ./test-conn.zeek, line 14: Can't generate zeekygen documentation for record field duration, unknown record: Log::Info
warning in ./test-conn.zeek, line 15: Can't generate zeekygen documentation for record field orig_bytes, unknown record: Log::Info
warning in ./test-conn.zeek, line 16: Can't generate zeekygen documentation for record field resp_bytes, unknown record: Log::Info
warning in ./test-conn.zeek, line 17: Can't generate zeekygen documentation for record field conn_state, unknown record: Log::Info
warning in ./test-conn.zeek, line 18: Can't generate zeekygen documentation for record field app_proto, unknown record: Log::Info
error in ./test-conn.zeek, line 7: unknown identifier (Log::Info)
error in ./test-conn.zeek, line 22: identifier is not exported: Log::Info
error in ./test-conn.zeek, line 26: identifier is not exported: Log::Info
error in ./test-conn.zeek, line 7 and ./test-conn.zeek, line 26: not a Zeek type (Log::Info)
error in ./test-conn.zeek, line 30: no such field in record (TestConn::c$id$orig_port)
error in ./test-conn.zeek, line 31: no such field in record (TestConn::c$id$resp_port)
error in ./test-conn.zeek, line 32: no such field in record (TestConn::c$id$proto)
error in ./test-conn.zeek, line 34: no such field in record (TestConn::c$orig_bytes)
error in ./test-conn.zeek, line 35: no such field in record (TestConn::c$resp_bytes)
error in ./test-conn.zeek, line 36: no such field in record (TestConn::c$state)
error in ./test-conn.zeek, line 37: unknown identifier app_proto, at or near "app_proto"
```

10. Verify that the custom log file test-conn.log is created and contains the expected connection information.

    File was not created