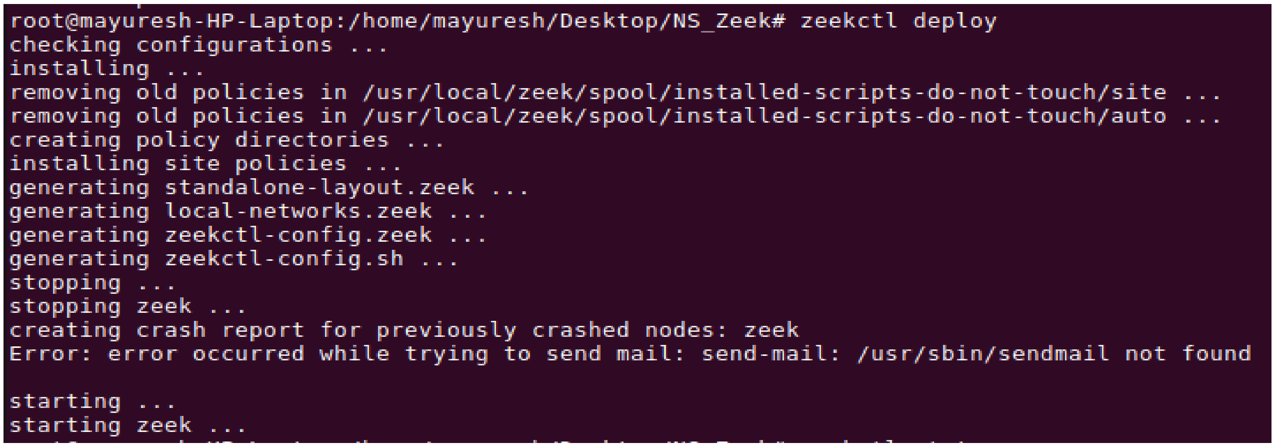
Name: Dindorkar Mayuresh Rajesh
Roll Number: CS23MTECH14007
Network Security Assignment 8: Zeek Hands-on

—---

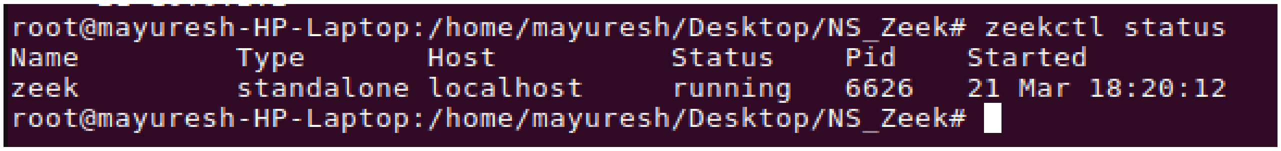
Task 1A

Started the Zeek:



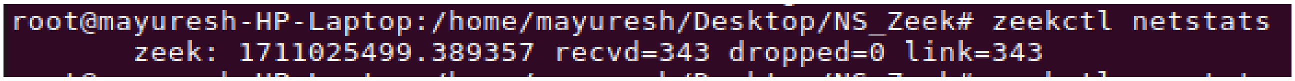
```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek# zeekctl deploy
checking configurations ...
installing ...
removing old policies in /usr/local/zeek/spool/installed-scripts-do-not-touch/site ...
removing old policies in /usr/local/zeek/spool/installed-scripts-do-not-touch/auto ...
creating policy directories ...
installing site policies ...
generating standalone-layout.zeek ...
generating local-networks.zeek ...
generating zeekctl-config.zeek ...
generating zeekctl-config.sh ...
stopping ...
stopping zeek ...
creating crash report for previously crashed nodes: zeek
Error: error occurred while trying to send mail: send-mail: /usr/sbin/sendmail not found

starting ...
starting zeek ...
```

Checking the status of Zeek:



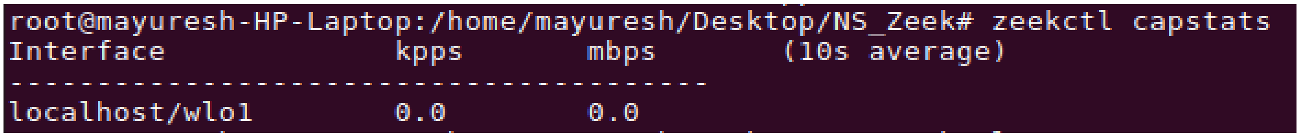
```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek# zeekctl status
Name         Type       Host          Status    Pid    Started
zeek         standalone localhost     running   6626   21 Mar 18:20:12
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek# 
```

Checking netstat status:



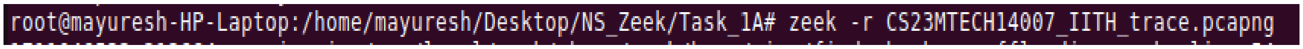
```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek# zeekctl netstats
        zeek: 1711025499.389357 recvd=343 dropped=0 link=343
```

Checking the capstats status output:



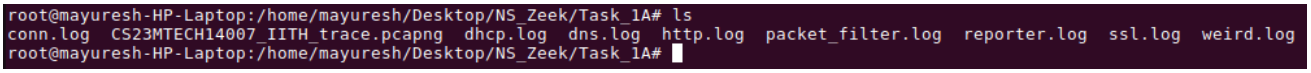
```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek# zeekctl capstats
Interface              kpps       mbps       (10s average)
----------------------------------------
localhost/wlo1         0.0        0.0
```

Providing the pcap file captured on personal laptop to zeek as input, for analysis:
$ zeek -r <pcap_file_name>



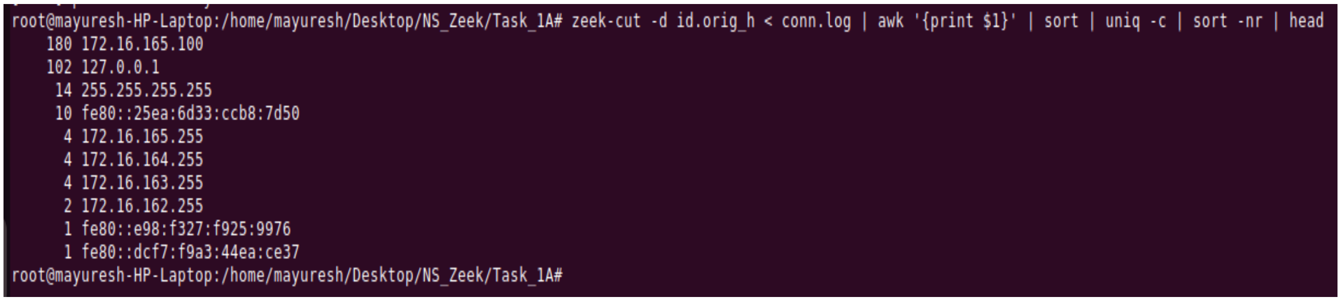
```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_1A# zeek -r CS23MTECH14007_IITH_trace.pcapng
```

- This command creates log files containing network traffic information in the present working directory.
- We can see them using the 'ls' command.



```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_1A# ls
conn.log  CS23MTECH14007_IITH_trace.pcapng  dhcp.log  dns.log  http.log  packet_filter.log  reporter.log  ssl.log  weird.log
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_1A# 
```

IP addresses that created most network traffic in DESC order:
$ zeek-cut -d id.orig_h < conn.log | awk '{print $1}' | sort | uniq -c | sort -nr | head

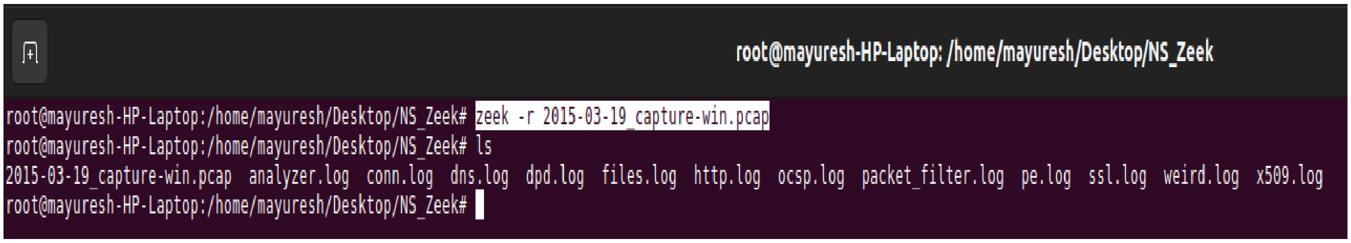


```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_1A# zeek-cut -d id.orig_h < conn.log | awk '{print $1}' | sort | uniq -c | sort -nr | head
    180 172.16.165.100
    102 127.0.0.1
     14 255.255.255.255
     10 fe80::25ea:6d33:ccb8:7d50
      4 172.16.165.255
      4 172.16.164.255
      4 172.16.163.255
      2 172.16.162.255
      1 fe80::e98:f327:f925:9976
      1 fe80::dcf7:f9a3:44ea:ce37
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_1A#
```

In the output, the first column represents the count, while the second column depicts the IP address.
—---

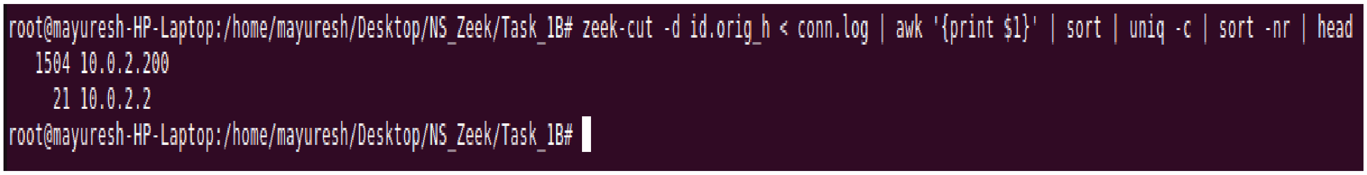
Task 1B

Providing the pcap file downloaded from https://www.stratosphereips.org/datasets-mixed (link 4: CTU-Mixed-Capture-4) to zeek as input, for analysis:
$ zeek -r <pcap_file_name>
This command creates 12 log files in the present working directory. We can see them using ‘ls’ command.



```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek# zeek -r 2015-03-19_capture-win.pcap
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek# ls
2015-03-19_capture-win.pcap  analyzer.log  conn.log  dns.log  dpd.log  files.log  http.log  ocsp.log  packet_filter.log  pe.log  ssl.log  weird.log  x509.log
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek#
```

IP addresses that created most network traffic in DESC order (using zeek-cut command):
$ zeek-cut -d id.orig_h < conn.log | awk '{print $1}' | sort | uniq -c | sort -nr | head



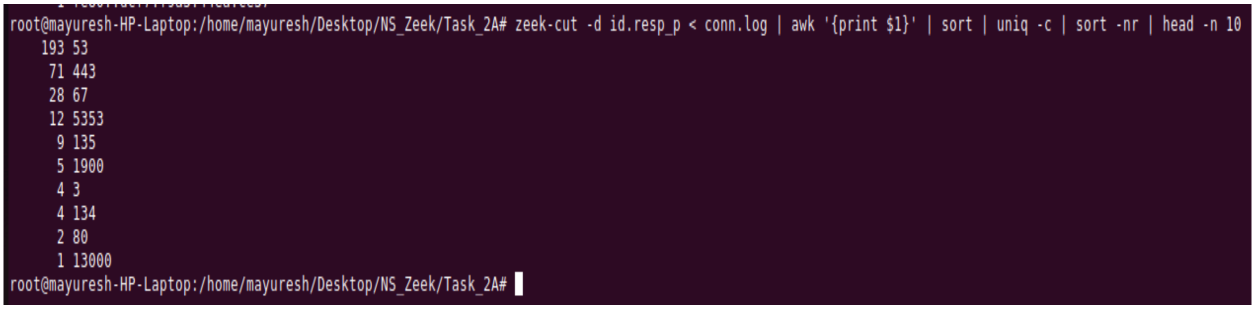
```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_1B# zeek-cut -d id.orig_h < conn.log | awk '{print $1}' | sort | uniq -c | sort -nr | head
   1504 10.0.2.200
     21 10.0.2.2
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_1B#
```

In the output, the first column represents the count, while the second column depicts the IP address.
—---

Task 2A

Top 10 destination ports which received most network traffic in DESC order for pcap captured on personal laptop:

$ zeek -r <pcap_file_name>
$ zeek-cut -d id.resp_p < conn.log | awk '{print $1}' | sort | uniq -c | sort -nr | head -n 10



```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_2A# zeek-cut -d id.resp_p < conn.log | awk '{print $1}' | sort | uniq -c | sort -nr | head -n 10
    193 53
     71 443
     28 67
     12 5353
      9 135
      5 1900
      4 3
      4 134
      2 80
      1 13000
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_2A# 
```

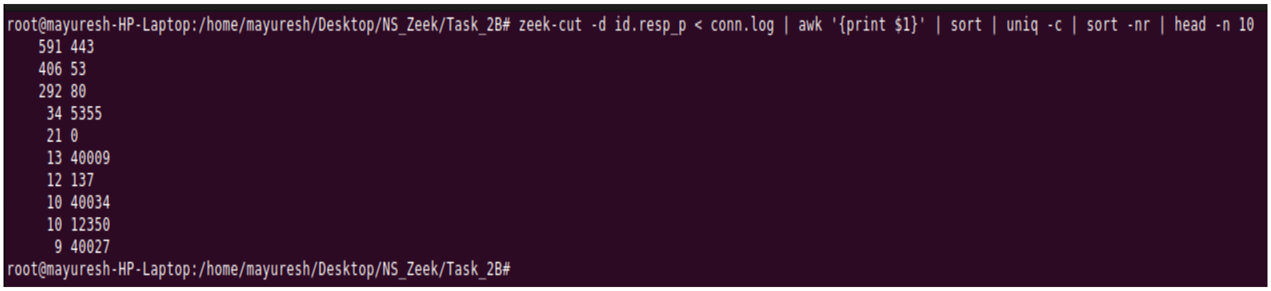
In the output, the first column represents the count, while the second column depicts the port number.

—--

Task 2B

Top 10 destination ports which received most network traffic in DESC order for pcap downloaded from https://www.stratosphereips.org/datasets-mixed
(link 4: CTU-Mixed-Capture-4):

$ zeek -r <pcap_file_name>
$ zeek-cut -d id.resp_p < conn.log | sort | uniq -c | sort -nr | head -n 10



```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_2B# zeek-cut -d id.resp_p < conn.log | awk '{print $1}' | sort | uniq -c | sort -nr | head -n 10
    591 443
    406 53
    292 80
     34 5355
     21 0
     13 40009
     12 137
     10 40034
     10 12350
      9 40027
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_2B#
```

In the output, the first column represents the count, while the second column depicts the port number.

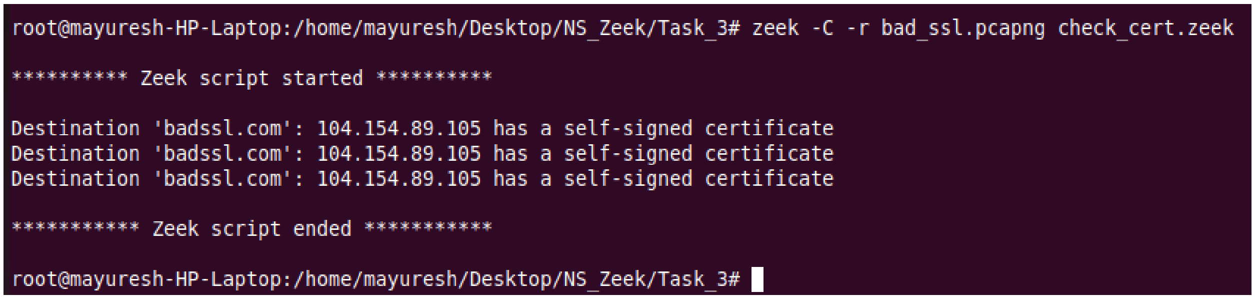
—--

Task 3

Zeek Script to detect self signed certificate of https://self-signed.badssl.com/ :

```
@load base/protocols/ssl
@load base/files/x509

# Gets called during initialization
event zeek_init() {
  print("");
  print fmt("********** Zeek script started **********");
  print("");
}

event ssl_established(c: connection) {

   local source_ip = c$id$orig_h;
   local dest_ip = c$id$resp_h;

  # If certificate does not contain chain
   if (!c?$ssl || !c$ssl?$cert_chain)
  {
    return;
  }

   local end_entity_certificate = c$ssl$cert_chain[0]$x509$certificate;
  if (end_entity_certificate$cn != "*.badssl.com") {
     print("Certificate does not belong to 'badssl.com'");
     return;
   }

  if (end_entity_certificate$issuer == end_entity_certificate$subject) {
    print fmt("Destination 'badssl.com': %s has a self-signed
certificate", dest_ip);
  }
}
# Called on end
event zeek_done() {
  print("");
  print fmt("*********** Zeek script ended ***********");
  print("");
}
```

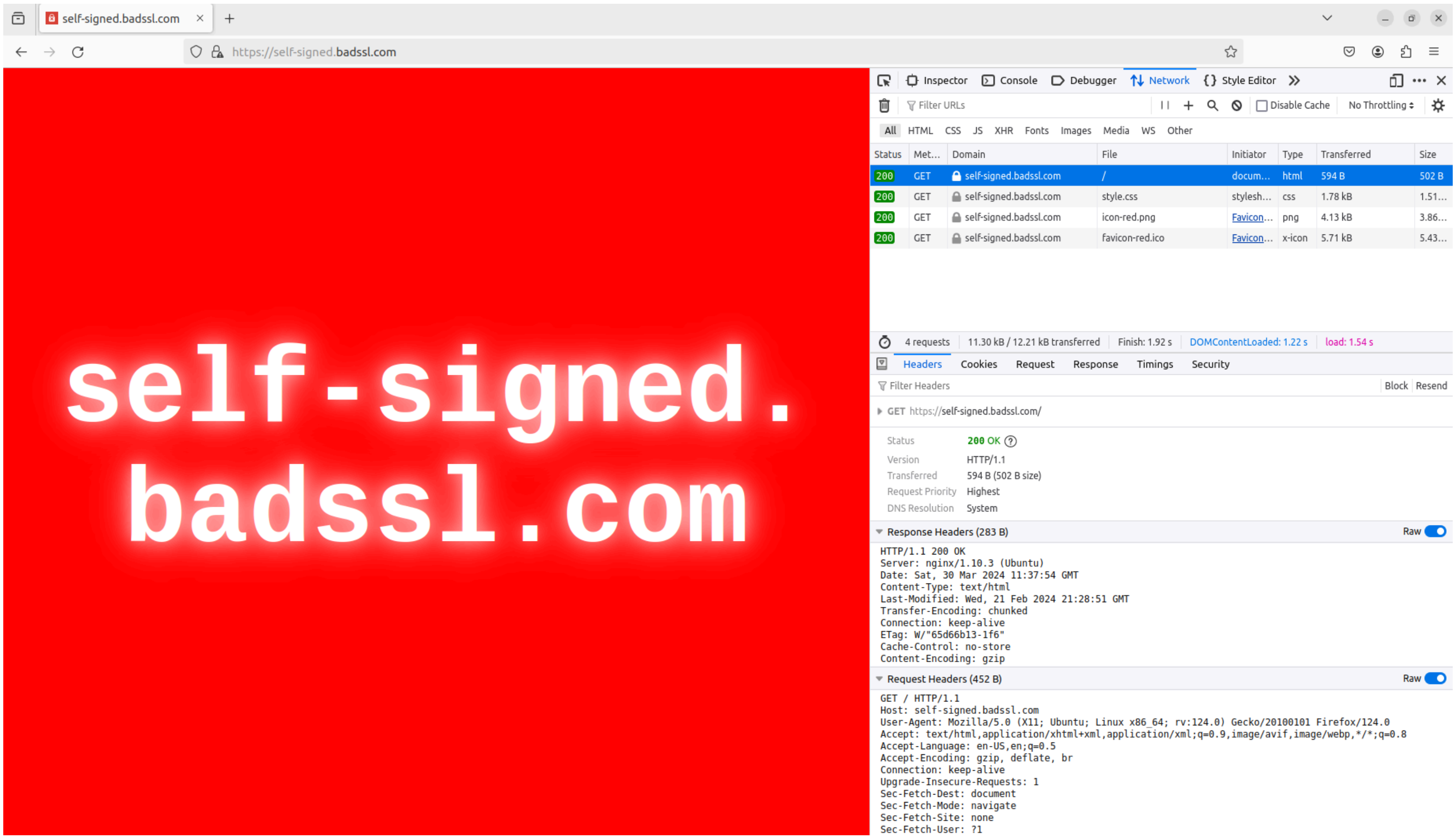
Output:



```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_3# zeek -C -r bad_ssl.pcapng check_cert.zeek

********** Zeek script started **********

Destination 'badssl.com': 104.154.89.105 has a self-signed certificate
Destination 'badssl.com': 104.154.89.105 has a self-signed certificate
Destination 'badssl.com': 104.154.89.105 has a self-signed certificate

*********** Zeek script ended ***********

root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_3#
```

Observations:

- We have captured the wireshark trace while visiting ‘badssl.com’.
- We can observe that the IP address of badssl.com is 104.154.89.105.
- For identifying the self-signed certificate, we compare the subject name and issuer of the X.509 certificate.



Status	Met…	Domain	File	Initiator	Type	Transferred	Size
200	GET	self-signed.badssl.com	/	docum…	html	594 B	502 B
200	GET	self-signed.badssl.com	style.css	stylesh…	css	1.78 kB	1.51…
200	GET	self-signed.badssl.com	icon-red.png	Favicon…	png	4.13 kB	3.86…
200	GET	self-signed.badssl.com	favicon-red.ico	Favicon…	x-icon	5.71 kB	5.43…

4 requests | 11.30 kB / 12.21 kB transferred | Finish: 1.92 s | DOMContentLoaded: 1.22 s | load: 1.54 s

GET https://self-signed.badssl.com/

Status	200 OK
Version	HTTP/1.1
Transferred	594 B (502 B size)
Request Priority	Highest
DNS Resolution	System

Response Headers (283 B)

```
HTTP/1.1 200 OK
Server: nginx/1.10.3 (Ubuntu)
Date: Sat, 30 Mar 2024 11:37:54 GMT
Content-Type: text/html
Last-Modified: Wed, 21 Feb 2024 21:28:51 GMT
Transfer-Encoding: chunked
Connection: keep-alive
ETag: W/"65d66b13-1f6"
Cache-Control: no-store
Content-Encoding: gzip
```

Request Headers (452 B)

```
GET / HTTP/1.1
Host: self-signed.badssl.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:124.0) Gecko/20100101 Firefox/124.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: none
Sec-Fetch-User: ?1
```

—--

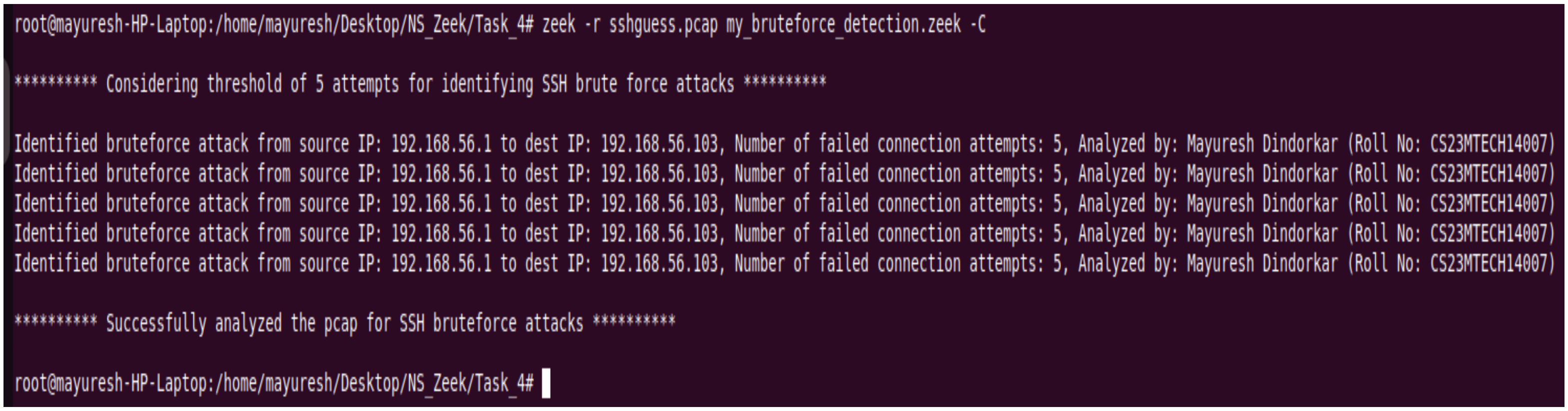
Task 4

Considering the threshold of 5 attempts for identifying the SSH bruteforce attack:
Zeek Script (my_bruteforce_detection.zeek):

```
@load base/protocols/ssh

# Dictionary to store <IP:count> pairs
global connection_attempts_dict: table[addr] of count = table();

# Threshold considered for identifying the brute force attack
global threshold: count;

# Gets called during initialization
event zeek_init() {
    print("");
    threshold = 5;
    print fmt("********** Considering threshold of %d attempts for
identifying SSH brute force attacks **********", threshold);
    print("");
}

# Event that gets triggered on each SSH connection attempt
event ssh_auth_attempted(conn: connection, authenticated: bool) {

    # If the connection attempt was unsuccessful
    if (!authenticated) {

        local source_host_ip = conn$id$orig_h;
        local dest_host_ip = conn$id$resp_h;

        # Increased the count of unsuccessful attempts for 'source IP' in
dictionary
        connection_attempts_dict[source_host_ip] = !(source_host_ip in
connection_attempts_dict) ? 1 : connection_attempts_dict[source_host_ip] +
1;

        # Checking whether threshold is exceeded or not
        if (threshold <= connection_attempts_dict[source_host_ip]) {

            local line1 = fmt("Identified bruteforce attack from source IP:
%s to dest IP: %s, Number of failed connection attempts: %d, ",
source_host_ip, dest_host_ip, connection_attempts_dict[source_host_ip]);
            local line2 = fmt("Analyzed by: Mayuresh Dindorkar (Roll No:
```

```
CS23MTECH14007)");
            print line1 + line2;

            # Resetting the count
            connection_attempts_dict[source_host_ip] = 0;
        }
    }
}

# Called on zeek stop
event zeek_done() {
    print("");
    print fmt("********** Successfully analyzed the pcap for SSH bruteforce
attacks **********");
    print("");
}
```

Output Screenshot for threshold = 5:



```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_4# zeek -r sshguess.pcap my_bruteforce_detection.zeek -C

********** Considering threshold of 5 attempts for identifying SSH brute force attacks **********

Identified bruteforce attack from source IP: 192.168.56.1 to dest IP: 192.168.56.103, Number of failed connection attempts: 5, Analyzed by: Mayuresh Dindorkar (Roll No: CS23MTECH14007)
Identified bruteforce attack from source IP: 192.168.56.1 to dest IP: 192.168.56.103, Number of failed connection attempts: 5, Analyzed by: Mayuresh Dindorkar (Roll No: CS23MTECH14007)
Identified bruteforce attack from source IP: 192.168.56.1 to dest IP: 192.168.56.103, Number of failed connection attempts: 5, Analyzed by: Mayuresh Dindorkar (Roll No: CS23MTECH14007)
Identified bruteforce attack from source IP: 192.168.56.1 to dest IP: 192.168.56.103, Number of failed connection attempts: 5, Analyzed by: Mayuresh Dindorkar (Roll No: CS23MTECH14007)
Identified bruteforce attack from source IP: 192.168.56.1 to dest IP: 192.168.56.103, Number of failed connection attempts: 5, Analyzed by: Mayuresh Dindorkar (Roll No: CS23MTECH14007)

********** Successfully analyzed the pcap for SSH bruteforce attacks **********

root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_4#
```

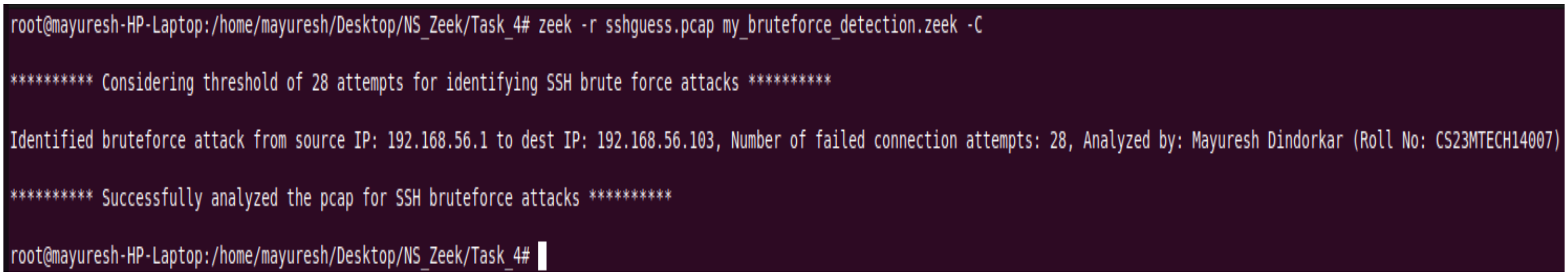
Observations:

- We can observe that there is an SSH bruteforce attempt from **source host ‘192.168.56.1’** to **destination host ‘192.168.56.103’**.

- When we kept the threshold as 5, we can observe the same results multiple times.

- When we increase the threshold to 28, there is only one result as shown in below screenshot. Hence, the source host performed the SSH connection attempt 28 times.

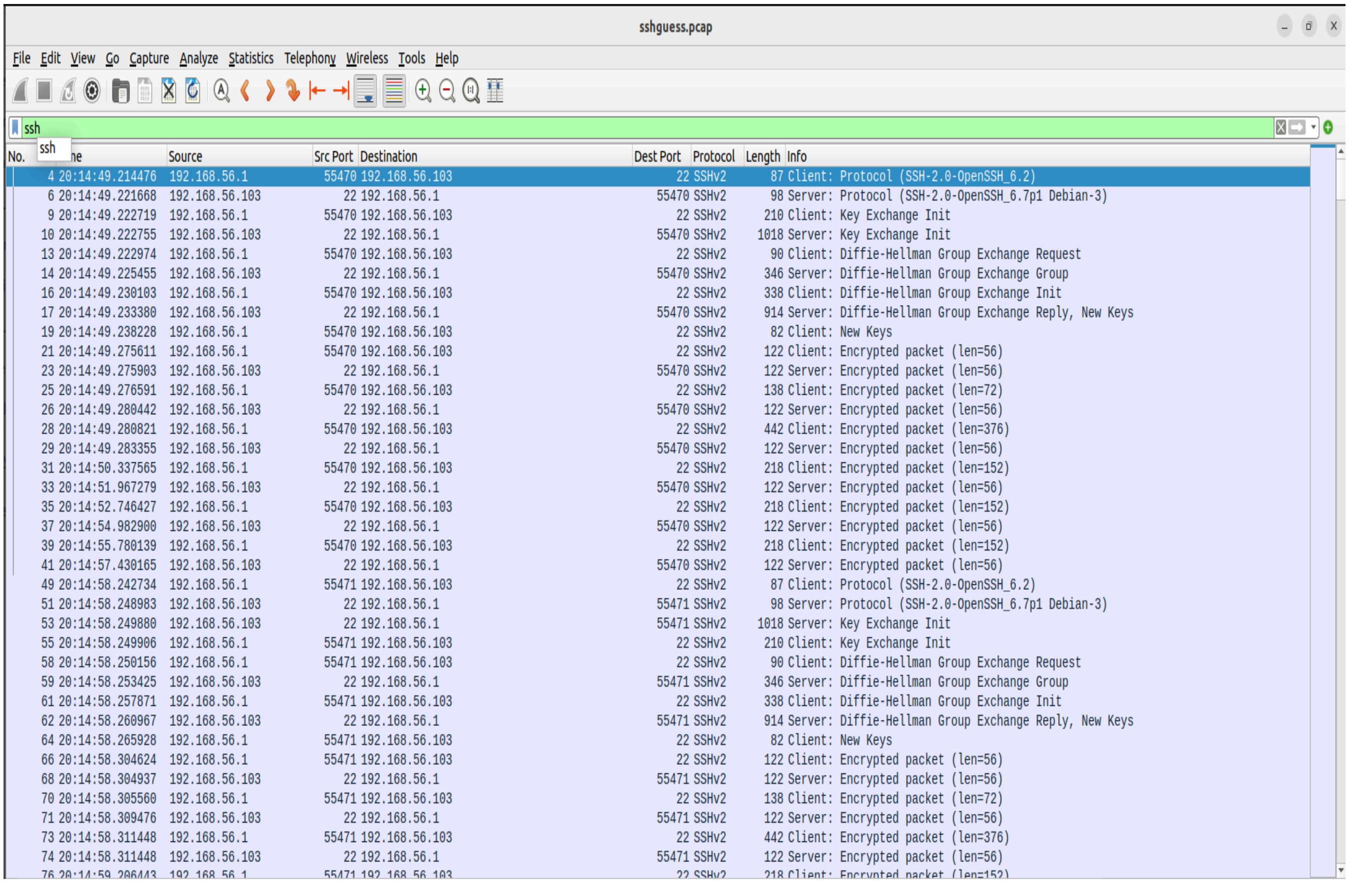
Output screenshot for threshold = 28:



```
root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_4# zeek -r sshguess.pcap my_bruteforce_detection.zeek -C

********** Considering threshold of 28 attempts for identifying SSH brute force attacks **********

Identified bruteforce attack from source IP: 192.168.56.1 to dest IP: 192.168.56.103, Number of failed connection attempts: 28, Analyzed by: Mayuresh Dindorkar (Roll No: CS23MTECH14007)

********** Successfully analyzed the pcap for SSH bruteforce attacks **********

root@mayuresh-HP-Laptop:/home/mayuresh/Desktop/NS_Zeek/Task_4# 
```

- **If we increase the threshold further**, we don't observe any result. Hence, the source has tried to establish the SSH connection exactly 28 times.

- We can cross check the results from wireshark. We can observe that after filtering the packets by ssh, all packets have '192.168.56.1' and '192.168.56.103' as source and destination IP and vice versa.



ssh

No.	ne	Source	Src Port	Destination	Dest Port	Protocol	Length	Info
4	20:14:49.214476	192.168.56.1	55470	192.168.56.103	22	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_6.2)
6	20:14:49.221668	192.168.56.103	22	192.168.56.1	55470	SSHv2	98	Server: Protocol (SSH-2.0-OpenSSH_6.7p1 Debian-3)
9	20:14:49.222719	192.168.56.1	55470	192.168.56.103	22	SSHv2	210	Client: Key Exchange Init
10	20:14:49.222755	192.168.56.103	22	192.168.56.1	55470	SSHv2	1018	Server: Key Exchange Init
13	20:14:49.222974	192.168.56.1	55470	192.168.56.103	22	SSHv2	90	Client: Diffie-Hellman Group Exchange Request
14	20:14:49.225455	192.168.56.103	22	192.168.56.1	55470	SSHv2	346	Server: Diffie-Hellman Group Exchange Group
16	20:14:49.230103	192.168.56.1	55470	192.168.56.103	22	SSHv2	338	Client: Diffie-Hellman Group Exchange Init
17	20:14:49.233380	192.168.56.103	22	192.168.56.1	55470	SSHv2	914	Server: Diffie-Hellman Group Exchange Reply, New Keys
19	20:14:49.238228	192.168.56.1	55470	192.168.56.103	22	SSHv2	82	Client: New Keys
21	20:14:49.275611	192.168.56.1	55470	192.168.56.103	22	SSHv2	122	Client: Encrypted packet (len=56)
23	20:14:49.275903	192.168.56.103	22	192.168.56.1	55470	SSHv2	122	Server: Encrypted packet (len=56)
25	20:14:49.276591	192.168.56.1	55470	192.168.56.103	22	SSHv2	138	Client: Encrypted packet (len=72)
26	20:14:49.280442	192.168.56.103	22	192.168.56.1	55470	SSHv2	122	Server: Encrypted packet (len=56)
28	20:14:49.280821	192.168.56.1	55470	192.168.56.103	22	SSHv2	442	Client: Encrypted packet (len=376)
29	20:14:49.283355	192.168.56.103	22	192.168.56.1	55470	SSHv2	122	Server: Encrypted packet (len=56)
31	20:14:50.337565	192.168.56.1	55470	192.168.56.103	22	SSHv2	218	Client: Encrypted packet (len=152)
33	20:14:51.967279	192.168.56.103	22	192.168.56.1	55470	SSHv2	122	Server: Encrypted packet (len=56)
35	20:14:52.746427	192.168.56.1	55470	192.168.56.103	22	SSHv2	218	Client: Encrypted packet (len=152)
37	20:14:54.982900	192.168.56.103	22	192.168.56.1	55470	SSHv2	122	Server: Encrypted packet (len=56)
39	20:14:55.780139	192.168.56.1	55470	192.168.56.103	22	SSHv2	218	Client: Encrypted packet (len=152)
41	20:14:57.430165	192.168.56.103	22	192.168.56.1	55470	SSHv2	122	Server: Encrypted packet (len=56)
49	20:14:58.242734	192.168.56.1	55471	192.168.56.103	22	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_6.2)
51	20:14:58.248983	192.168.56.103	22	192.168.56.1	55471	SSHv2	98	Server: Protocol (SSH-2.0-OpenSSH_6.7p1 Debian-3)
53	20:14:58.249880	192.168.56.103	22	192.168.56.1	55471	SSHv2	1018	Server: Key Exchange Init
55	20:14:58.249906	192.168.56.1	55471	192.168.56.103	22	SSHv2	210	Client: Key Exchange Init
58	20:14:58.250156	192.168.56.1	55471	192.168.56.103	22	SSHv2	90	Client: Diffie-Hellman Group Exchange Request
59	20:14:58.253425	192.168.56.103	22	192.168.56.1	55471	SSHv2	346	Server: Diffie-Hellman Group Exchange Group
61	20:14:58.257871	192.168.56.1	55471	192.168.56.103	22	SSHv2	338	Client: Diffie-Hellman Group Exchange Init
62	20:14:58.260967	192.168.56.103	22	192.168.56.1	55471	SSHv2	914	Server: Diffie-Hellman Group Exchange Reply, New Keys
64	20:14:58.265928	192.168.56.1	55471	192.168.56.103	22	SSHv2	82	Client: New Keys
66	20:14:58.304624	192.168.56.1	55471	192.168.56.103	22	SSHv2	122	Client: Encrypted packet (len=56)
68	20:14:58.304937	192.168.56.103	22	192.168.56.1	55471	SSHv2	122	Server: Encrypted packet (len=56)
70	20:14:58.305560	192.168.56.1	55471	192.168.56.103	22	SSHv2	138	Client: Encrypted packet (len=72)
71	20:14:58.309476	192.168.56.103	22	192.168.56.1	55471	SSHv2	122	Server: Encrypted packet (len=56)
73	20:14:58.311448	192.168.56.1	55471	192.168.56.103	22	SSHv2	442	Client: Encrypted packet (len=376)
74	20:14:58.311448	192.168.56.103	22	192.168.56.1	55471	SSHv2	122	Server: Encrypted packet (len=56)
76	20:14:59.206443	192.168.56.1	55471	192.168.56.103	22	SSHv2	218	Client: Encrypted packet (len=152)

—---

PLAGIARISM STATEMENT

I certify that this assignment/report is my own work, based on my personal study and/or research and that I have acknowledged all material and sources used in its preparation, whether they be books, articles, reports, lecture notes, and any other kind of document, electronic or personal communication. I also certify that this assignment/report has not previously been submitted for assessment in any other course, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that I have not copied in part or whole or otherwise plagiarized the work of other students and/or persons. I pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, I understand my responsibility to report honor violations by other students if I become aware of it.

Name: Mayuresh Dindorkar

Date: 30/03/2024

Signature: MD