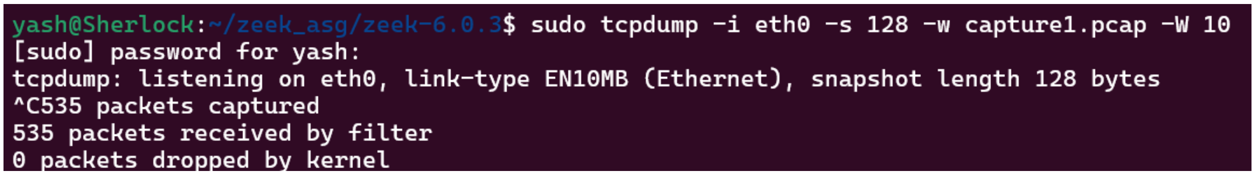
Assignment 8 : Hands-on with Zeek

cs23mtech14018 - Yash Shukla

Task 1A: Collect network traffic (only packet headers up to MAC layer to reduce the size of pcap file) using tcpdump or wireshark on your personal laptop for 10 mins and show the source IP addresses that generated the most network traffic, organized in descending order using zeek-cut. Deliverables: pcap file generated and relevant zeek log files; A screenshot of zeek-cut and its options used for answering this query and the output generated

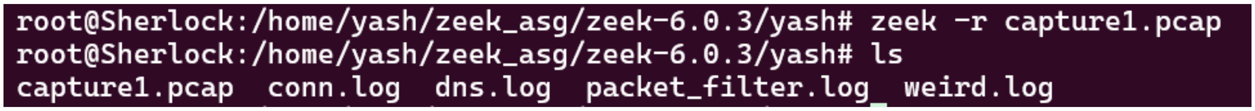
Solution :
: Command : sudo tcpdump -i eth0 -s 128 -w capture1.pcap -W 10



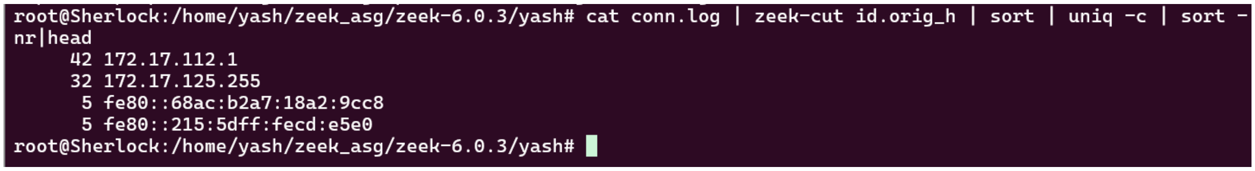
```
yash@Sherlock:~/zeek_asg/zeek-6.0.3$ sudo tcpdump -i eth0 -s 128 -w capture1.pcap -W 10
[sudo] password for yash:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 128 bytes
^C535 packets captured
535 packets received by filter
0 packets dropped by kernel
```

Run zeek and analyze the captured traffic in the capture1.pcap file.

Using zeek-cut to analyze zeek logs ->



```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash# zeek -r capture1.pcap
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash# ls
capture1.pcap  conn.log  dns.log  packet_filter.log  weird.log
```



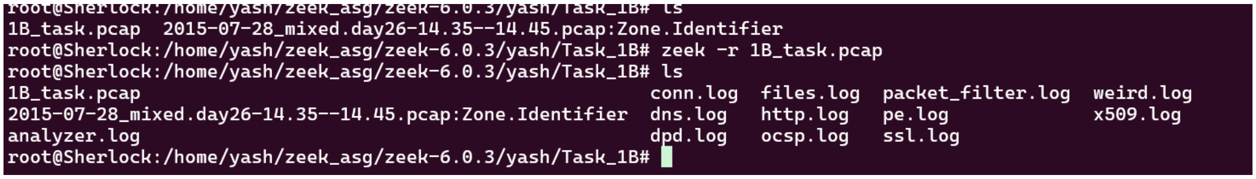
```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash# cat conn.log | zeek-cut id.orig_h | sort | uniq -c | sort -
nr|head
     42 172.17.112.1
     32 172.17.125.255
      5 fe80::68ac:b2a7:18a2:9cc8
      5 fe80::215:5dff:fecd:e5e0
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash#
```

Task 1B: Repeat Task 1A by using one of the pcap files from
https://www.stratosphereips.org/datasets-mixed or
https://www.honeynetproject.com/dataset.html
Deliverables: link of the pcap file used; A screenshot of zeek-cut and its options used for answering this query and the output generated.

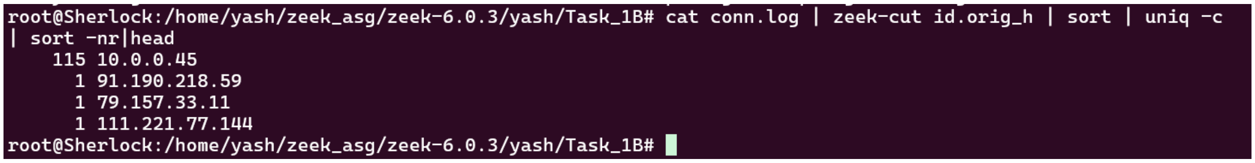
Solution :

Link of pcap file used :

https://mcfp.felk.cvut.cz/publicDatasets/CTU-Mixed-Capture-1/2015-07-28_mixed.day26-14.35--14.45.pcap



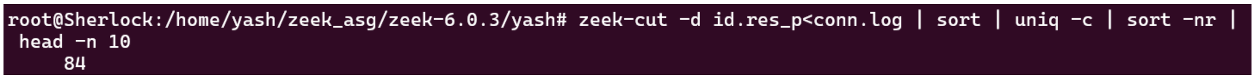
```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_1B# ls
1B_task.pcap  2015-07-28_mixed.day26-14.35--14.45.pcap:Zone.Identifier
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_1B# zeek -r 1B_task.pcap
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_1B# ls
1B_task.pcap                                             conn.log  files.log  packet_filter.log  weird.log
2015-07-28_mixed.day26-14.35--14.45.pcap:Zone.Identifier dns.log   http.log   pe.log             x509.log
analyzer.log                                             dpd.log   ocsp.log   ssl.log
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_1B#
```



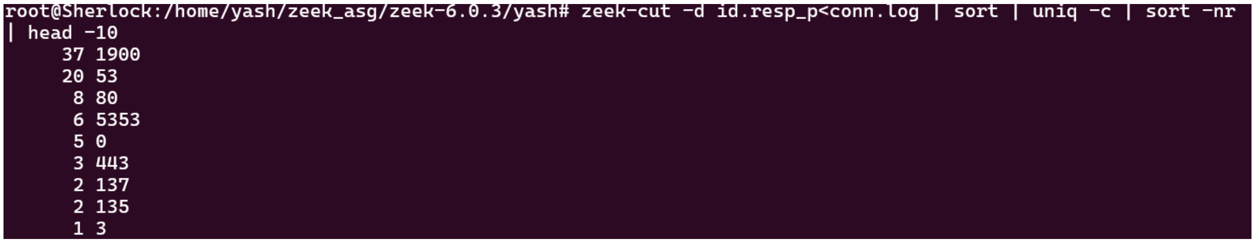
```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_1B# cat conn.log | zeek-cut id.orig_h | sort | uniq -c
| sort -nr|head
    115 10.0.0.45
      1 91.190.218.59
      1 79.157.33.11
      1 111.221.77.144
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_1B#
```

Task 2A: Show the 10 destination ports that received the most network traffic,organized in descending order using zeek-cut. Deliverables: Relevant zeek log files and a screenshot of zeek-cut and its options used for answering this query and the output generated.

Solution :



```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash# zeek-cut -d id.res_p<conn.log | sort | uniq -c | sort -nr |
 head -n 10
     84
```



```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash# zeek-cut -d id.resp_p<conn.log | sort | uniq -c | sort -nr
| head -10
     37 1900
     20 53
      8 80
      6 5353
      5 0
      3 443
      2 137
      2 135
      1 3
```

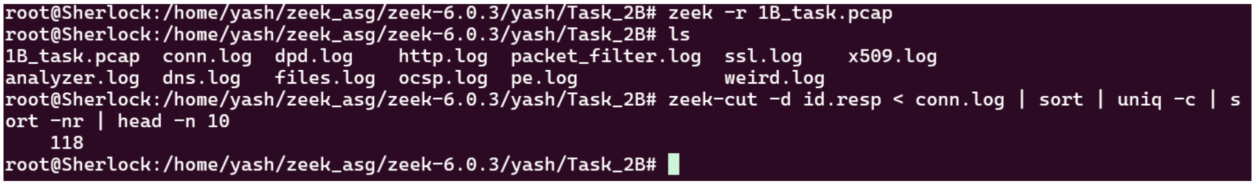
Task 2B: Repeat Task 2A by using one of the pcap files from https://www.stratosphereips.org/datasets-mixed or https://www.honeynetproject.com/dataset.html

Deliverables: link of the pcap file used for completing this task; Relevant zeek log files; A screenshot of zeek-cut and its options used for answering this query and the output generated

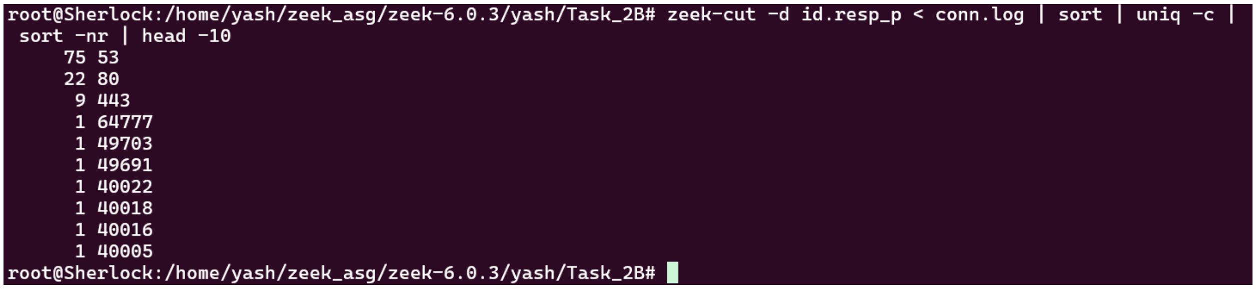
Solution :

Link of pcap file used :

https://mcfp.felk.cvut.cz/publicDatasets/CTU-Mixed-Capture-1/2015-07-28_mixed.day26-14.35--14.45.pcap



```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_2B# zeek -r 1B_task.pcap
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_2B# ls
1B_task.pcap  conn.log  dpd.log    http.log  packet_filter.log  ssl.log    x509.log
analyzer.log  dns.log   files.log  ocsp.log  pe.log             weird.log
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_2B# zeek-cut -d id.resp < conn.log | sort | uniq -c | s
ort -nr | head -n 10
    118
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_2B# 
```

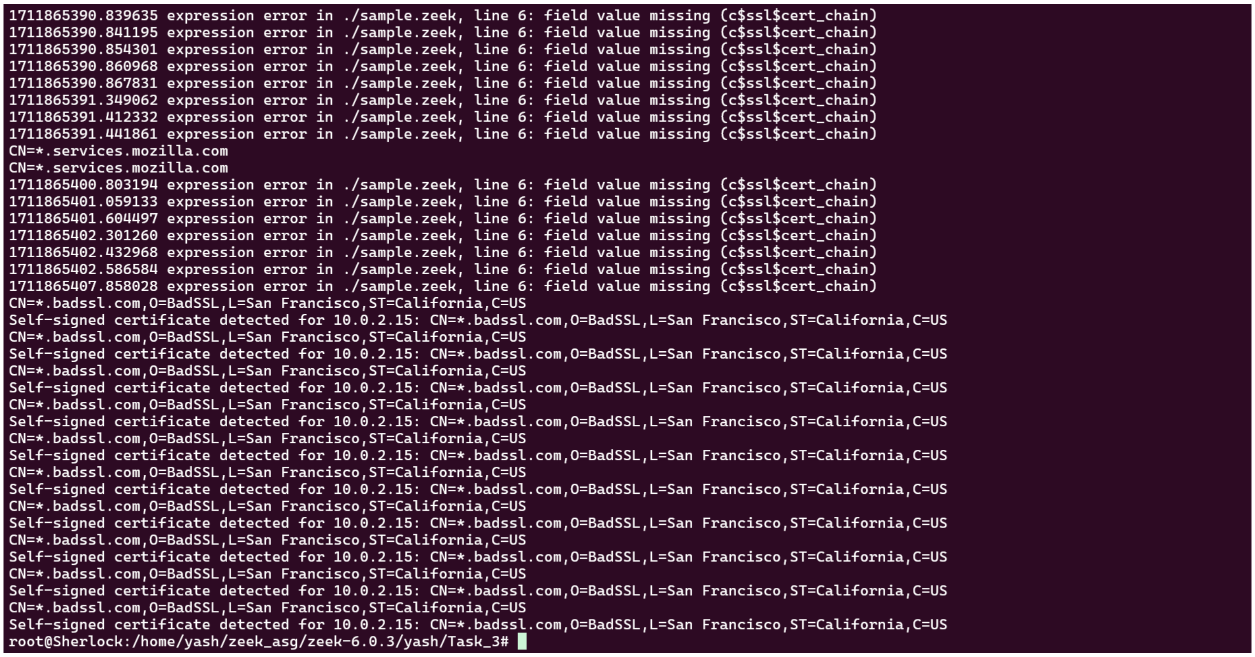


```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_2B# zeek-cut -d id.resp_p < conn.log | sort | uniq -c |
 sort -nr | head -10
     75 53
     22 80
      9 443
      1 64777
      1 49703
      1 49691
      1 40022
      1 40018
      1 40016
      1 40005
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_2B# 
```

Task 3: Write a Zeek script to identify the Self Signed Certificate of the website: https://self-signed.badssl.com/

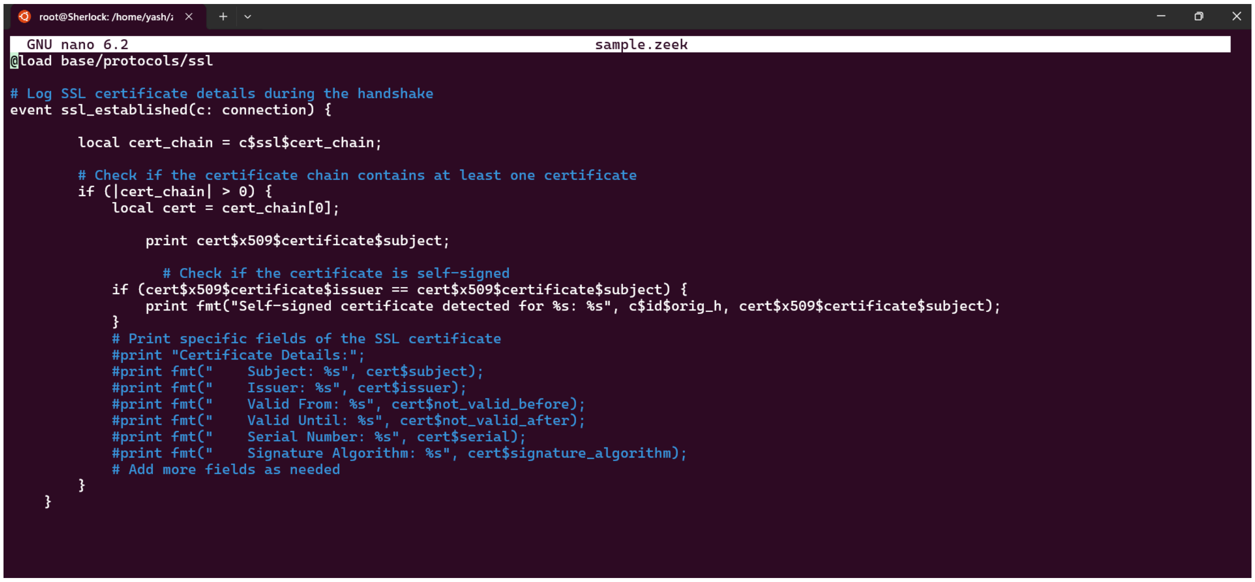
Deliverables: zeek script and a screenshot of the output generated by it when you visited this webpage.

Solution :



```
1711865390.839635 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865390.841195 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865390.854301 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865390.860968 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865390.867831 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865391.349062 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865391.412332 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865391.441861 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
CN=*.services.mozilla.com
CN=*.services.mozilla.com
1711865400.803194 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865401.059133 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865401.604497 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865402.301260 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865402.432968 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865402.586584 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
1711865407.858028 expression error in ./sample.zeek, line 6: field value missing (c$ssl$cert_chain)
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
Self-signed certificate detected for 10.0.2.15: CN=*.badssl.com,O=BadSSL,L=San Francisco,ST=California,C=US
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_3# 
```

Here is the Code Snippet used to identify the Self Signed Certificate of the Website :



```
  GNU nano 6.2                                                    sample.zeek
@load base/protocols/ssl

# Log SSL certificate details during the handshake
event ssl_established(c: connection) {

        local cert_chain = c$ssl$cert_chain;

        # Check if the certificate chain contains at least one certificate
        if (|cert_chain| > 0) {
            local cert = cert_chain[0];

                print cert$x509$certificate$subject;

                  # Check if the certificate is self-signed
            if (cert$x509$certificate$issuer == cert$x509$certificate$subject) {
                print fmt("Self-signed certificate detected for %s: %s", c$id$orig_h, cert$x509$certificate$subject);
            }
            # Print specific fields of the SSL certificate
            #print "Certificate Details:";
            #print fmt("    Subject: %s", cert$subject);
            #print fmt("    Issuer: %s", cert$issuer);
            #print fmt("    Valid From: %s", cert$not_valid_before);
            #print fmt("    Valid Until: %s", cert$not_valid_after);
            #print fmt("    Serial Number: %s", cert$serial);
            #print fmt("    Signature Algorithm: %s", cert$signature_algorithm);
            # Add more fields as needed
        }
    }
```

Zeek Script :

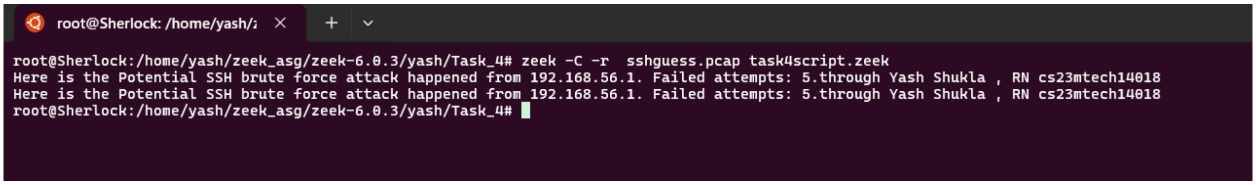
```
@load base/protocols/ssl

# Log SSL certificate details during the handshake
event ssl_established(c: connection) {

        local cert_chain = c$ssl$cert_chain;

        # Check if the certificate chain contains at least one certificate
        if (|cert_chain| > 0) {
            local cert = cert_chain[0];

                print cert$x509$certificate$subject;

                  # Check if the certificate is self-signed
            if (cert$x509$certificate$issuer == cert$x509$certificate$subject) {
                print fmt("Self-signed certificate detected for %s: %s", c$id$orig_h,
cert$x509$certificate$subject);
            }
            # Print specific fields of the SSL certificate
            #print "Certificate Details:";
            #print fmt("    Subject: %s", cert$subject);
            #print fmt("    Issuer: %s", cert$issuer);
            #print fmt("    Valid From: %s", cert$not_valid_before);
            #print fmt("    Valid Until: %s", cert$not_valid_after);
            #print fmt("    Serial Number: %s", cert$serial);
            #print fmt("    Signature Algorithm: %s", cert$signature_algorithm);
            # Add more fields as needed
        }
    }
```

Explanation :

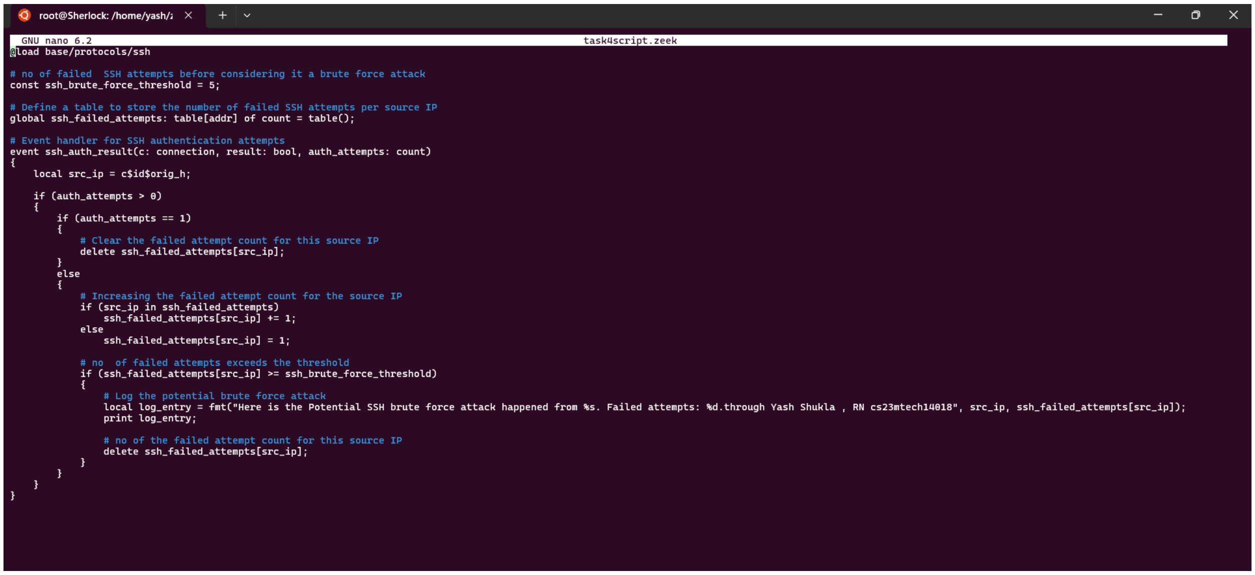
Task 4: Write a Zeek script to identify the ssh brute force password attacks in the following pcap file. Print the hosts that are guessing ssh passwords along with your name and RollNo in the generated log.
https://github.com/bro/bro/raw/master/testing/btest/Traces/ssh/sshguess.pcap

Solution :

ScreenShot :



```
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_4# zeek -C -r  sshguess.pcap task4script.zeek
Here is the Potential SSH brute force attack happened from 192.168.56.1. Failed attempts: 5.through Yash Shukla , RN cs23mtech14018
Here is the Potential SSH brute force attack happened from 192.168.56.1. Failed attempts: 5.through Yash Shukla , RN cs23mtech14018
root@Sherlock:/home/yash/zeek_asg/zeek-6.0.3/yash/Task_4#
```



```
@load base/protocols/ssh

# no of failed  SSH attempts before considering it a brute force attack
const ssh_brute_force_threshold = 5;

# Define a table to store the number of failed SSH attempts per source IP
global ssh_failed_attempts: table[addr] of count = table();

# Event handler for SSH authentication attempts
event ssh_auth_result(c: connection, result: bool, auth_attempts: count)
{
    local src_ip = c$id$orig_h;

    if (auth_attempts > 0)
    {
        if (auth_attempts == 1)
        {
            # Clear the failed attempt count for this source IP
            delete ssh_failed_attempts[src_ip];
        }
        else
        {
            # Increasing the failed attempt count for the source IP
            if (src_ip in ssh_failed_attempts)
                ssh_failed_attempts[src_ip] += 1;
            else
                ssh_failed_attempts[src_ip] = 1;

            # no  of failed attempts exceeds the threshold
            if (ssh_failed_attempts[src_ip] >= ssh_brute_force_threshold)
            {
                # Log the potential brute force attack
                local log_entry = fmt("Here is the Potential SSH brute force attack happened from %s. Failed attempts: %d.through Yash Shukla , RN cs23mtech14018", src_ip, ssh_failed_attempts[src_ip]);
                print log_entry;

                # no of the failed attempt count for this source IP
                delete ssh_failed_attempts[src_ip];
            }
        }
    }
}
```

Here is the Code Snippet used to identify the ssh brute force password attacks in the pcap file :

Zeek Script :

```
 GNU nano 6.2                                                   task4script.zeek

   @load base/protocols/ssh

# no of failed  SSH attempts before considering it a brute force attack
const ssh_brute_force_threshold = 5;

# Define a table to store the number of failed SSH attempts per source IP
global ssh_failed_attempts: table[addr] of count = table();

# Event handler for SSH authentication attempts
event ssh_auth_result(c: connection, result: bool, auth_attempts: count)
{
   local src_ip = c$id$orig_h;

   if (auth_attempts > 0)
   {
      if (auth_attempts == 1)
      {
         # Clear the failed attempt count for this source IP
         delete ssh_failed_attempts[src_ip];
      }
      else
      {
         # Increasing the failed attempt count for the source IP
         if (src_ip in ssh_failed_attempts)
            ssh_failed_attempts[src_ip] += 1;
         else
            ssh_failed_attempts[src_ip] = 1;

         # no  of failed attempts exceeds the threshold
         if (ssh_failed_attempts[src_ip] >= ssh_brute_force_threshold)
         {
            # Log the potential brute force attack
            local log_entry = fmt("Here is the Potential SSH brute force attack happened from %s.
Failed attempts: %d.through Yash Shukla , RN cs23mtech14018", src_ip,
ssh_failed_attempts[src_ip]);
            print log_entry;

            # no of the failed attempt count for this source IP
```

```
                delete ssh_failed_attempts[src_ip];
            }
        }
    }
}
```

Explanation :

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: Yash Shukla
Date: 30/03/2024
Signature: Yash Shukla