Assignment 8: Hands-on with Zeek

Submitted By : Yug Patel
Roll No. : CS23MTECH14019

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.

Step:1) Passively tapping the network traffic using tcpdump on interface **wlo1.**

```
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment# tcpdump -i wlo1 -w capture.pcap
tcpdump: listening on wlo1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C2024 packets captured
2024 packets received by filter
0 packets dropped by kernel
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment# ls
capture.pcap
```

Step:2) Making the zeek log files from the captured pcap file.

```
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment# zeek -r capture.pcap
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment# ls
capture.pcap  conn.log  dns.log  http.log  packet_filter.log  ssl.log
```

Step:3) Filtering the most frequent source ip using zeek-cut.

Command: zeek-cut -d -F, id.orig_h < conn.log | sort | uniq -c | sort -nr

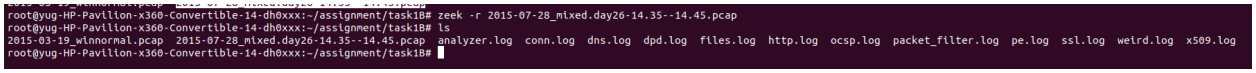
```
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment# zeek-cut -d -F, id.orig_h < conn.log | sort | uniq -c | sort -nr
     77 192.168.0.103
     15 192.168.0.113
     12 192.168.112.172
     10 192.168.113.83
     10 192.168.112.165
      5 192.168.113.224
      5 192.168.112.154
      5 192.168.112.123
      5 192.168.0.1
      2 192.168.112.215
      2 192.168.0.102
      1 fe80::4c5:a78d:2f59:cf49
      1 192.168.113.230
```

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

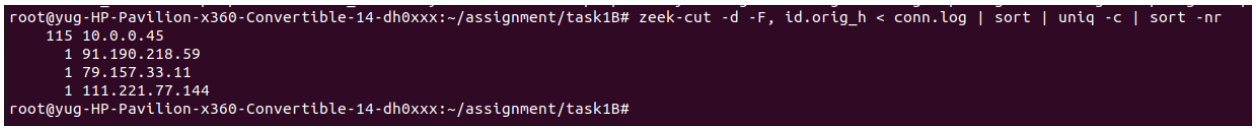
Link of pcap file: https://mcfp.felk.cvut.cz/publicDatasets/CTU-Mixed-Capture-1/2015-07-28_mixed.day26-14.35--14.45.pcap

Step:1) Downloading the required file.

Step:2) Forming the log files



Step:3) Filtering the conn.log files using the above mentioned command



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.

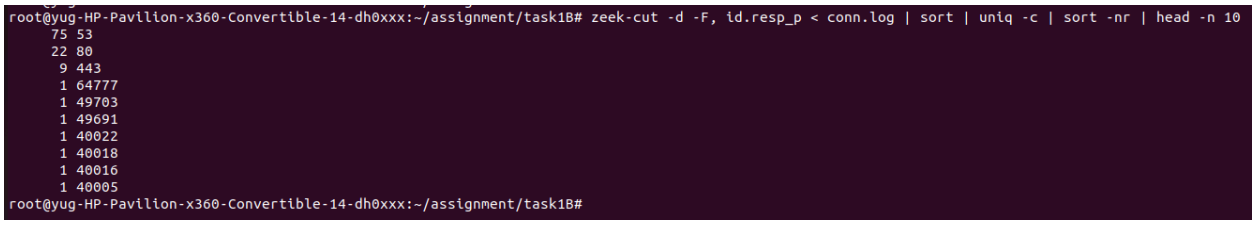
Step:1) Same as Task1A
Step:2) Same as Task1A
Step:3) Making a small change in the above mentioned command to get the desired output.

Command: zeek-cut -d -F, id.resp_p < conn.log | sort | uniq -c | sort -nr | head -n 10

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

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

Step:1) Same as Task1A
Step:2) Same as Task1A
Step:3) Same as Task2A



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

Step:1) Writing a zeek script that detects whether a connection's SSL connection possesses a certificate chain or not , then extract its leaf certificate and check whether it's self-signed or not.

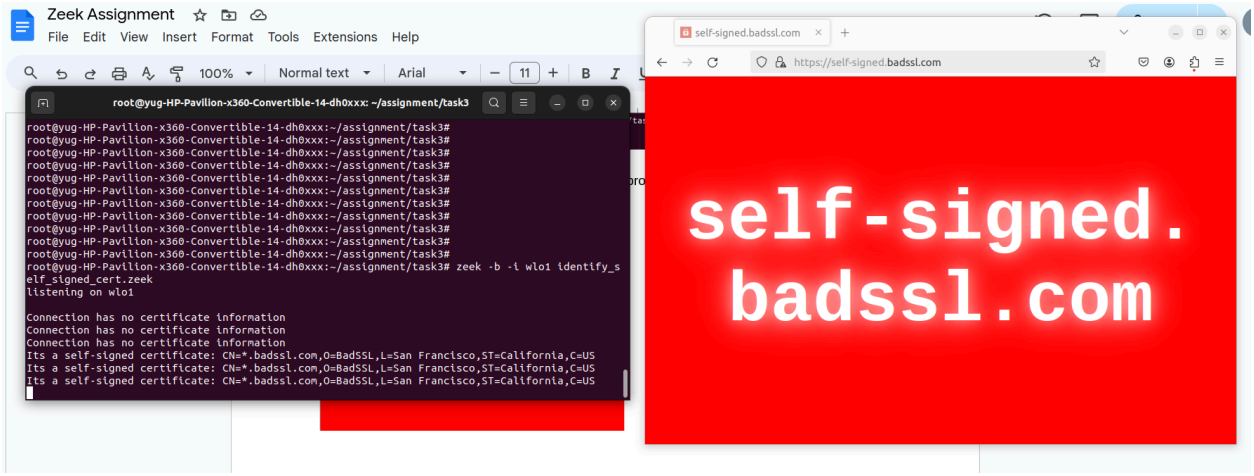
```
#identify_self_signed_cert.zeek
@load base/protocols/ssl

event ssl_established(c: connection) {
        if (c$ssl?$cert_chain && |c$ssl$cert_chain| > 0) {
        local leaf_cert = c$ssl$cert_chain[|c$ssl$cert_chain| - 1];

        # Check if the leaf certificate is self-signed
        if (leaf_cert$x509$certificate$subject == leaf_cert$x509$certificate$issuer) {
        print fmt("Its a self-signed certificate: %s", leaf_cert$x509$certificate$subject);
        } else {
        print "Its not a self-signed certificate";
        }
        } else {
        print "Connection has no certificate information";
        }
}
```

Step:2) Listening for connection on interface **wlo1**.

```
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment/task3#
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment/task3# zeek -b -i wlo1 identify_self_signed_cert.ze
ek
listening on wlo1
```

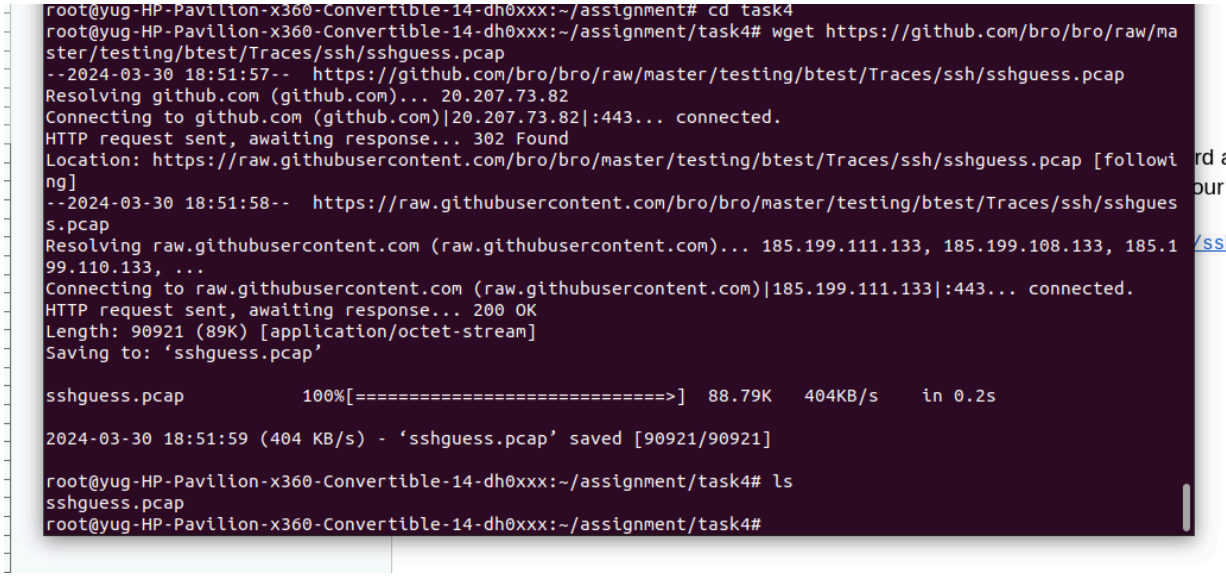
Step:3) Opening https://self-signed.badssl.com/ in browser.

Step:4) The script detected that the certificate of https://self-signed.badssl.com/ is self signed.



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

Step:1) Downloading the required pcap file.



Step:2) Write a zeek script which defines a tunable threshold for number of allowable failed attempts and print each of the failed attempts with a uid and host ip address who at this time might be guessing the password. If the number of failed attempts exceeds the threshold then the script classifies that host as an attacker.

#ssh_brute_force_detection.zeek

```
@load base/frameworks/notice
@load base/frameworks/sumstats

module SSH;

export {
        redef enum Notice::Type += {
        Exceeded_Failed_Login_Threshold,
        Attacker_Detected,
        };

        const failed_login_threshold: count = 5 &redef;
}
```

```
global failed_logins: table[addr] of count = table();
global attackers: set[addr] = set();

global detected_attackers: set[addr] = set();

event ssh_auth_failed(c: connection)
        {
        local id = c$id$orig_h;

        if (id !in failed_logins) {
        failed_logins[id] = 1;
        } else {
        failed_logins[id] += 1;
        }
        if (failed_logins[id] <= failed_login_threshold) {
        print fmt("Name: Yug, Roll No.: CS23MTECH14019,
        Connection UID %s: Host %s has attempted failed login (%d time).", c$uid, id,
        failed_logins[id]);
        }


        if (failed_logins[id] > failed_login_threshold && !(id in detected_attackers)) {
        print fmt("Name: Yug, Roll No.: CS23MTECH14019,
        Connection UID %s: Host %s has exceeded the failed login threshold (%d times) and is
        classified as an attacker.", c$uid, id, failed_logins[id]);
        attackers += {id};
        detected_attackers += {id};
        NOTICE([$note=Exceeded_Failed_Login_Threshold,
                $msg=fmt("Connection UID %s: Host %s has exceeded the failed login threshold
(%d times) and is classified as an attacker.", c$uid, id, failed_logins[id]),
                $src=id]);
        NOTICE([$note=Attacker_Detected,
                $msg=fmt("Host %s has been classified as an attacker.", id),
                $src=id]);
        }
}
```

Step:3) Run the **sshguess.pcap** with **ssh_brute_force_detection.zeek** to generate the required log files and to detect any ssh brute force attacks.

Command: zeek -C -r sshguess.pcap ssh_brute_force_detection.zeek

```
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment/task4# zeek -C -r sshguess.pcap ssh_brute_force_detection.zeek
Name: Yug, Roll No.: CS23MTECH14019, Connection UID CzrTb42IFkYh9T7V7: Host 192.168.56.1 has attempted failed login (1 time).
Name: Yug, Roll No.: CS23MTECH14019, Connection UID COA1gx3s3PtryGkHjc: Host 192.168.56.1 has attempted failed login (2 time).
Name: Yug, Roll No.: CS23MTECH14019, Connection UID CjPPkv2XeecklezTDl: Host 192.168.56.1 has attempted failed login (3 time).
Name: Yug, Roll No.: CS23MTECH14019, Connection UID CTcY9A2PJchTWmr5Uk: Host 192.168.56.1 has attempted failed login (4 time).
Name: Yug, Roll No.: CS23MTECH14019, Connection UID Cg5mzR2sq4HNw1pblg: Host 192.168.56.1 has attempted failed login (5 time).
Name: Yug, Roll No.: CS23MTECH14019, Connection UID CPqlScjuhYHOa0itl: Host 192.168.56.1 has exceeded the failed login threshold (6 times) and is classified as an attacker.
root@yug-HP-Pavilion-x360-Convertible-14-dh0xxx:~/assignment/task4#
```

It has classified host **192.168.56.1** as a brute force attacker as it has exceed the threshold i.e. **5**

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: Yug Patel
Date: 30/03/2024
Signature: Yug Patel