CS6903 : Network Security Term Project Report
KeyEscrow

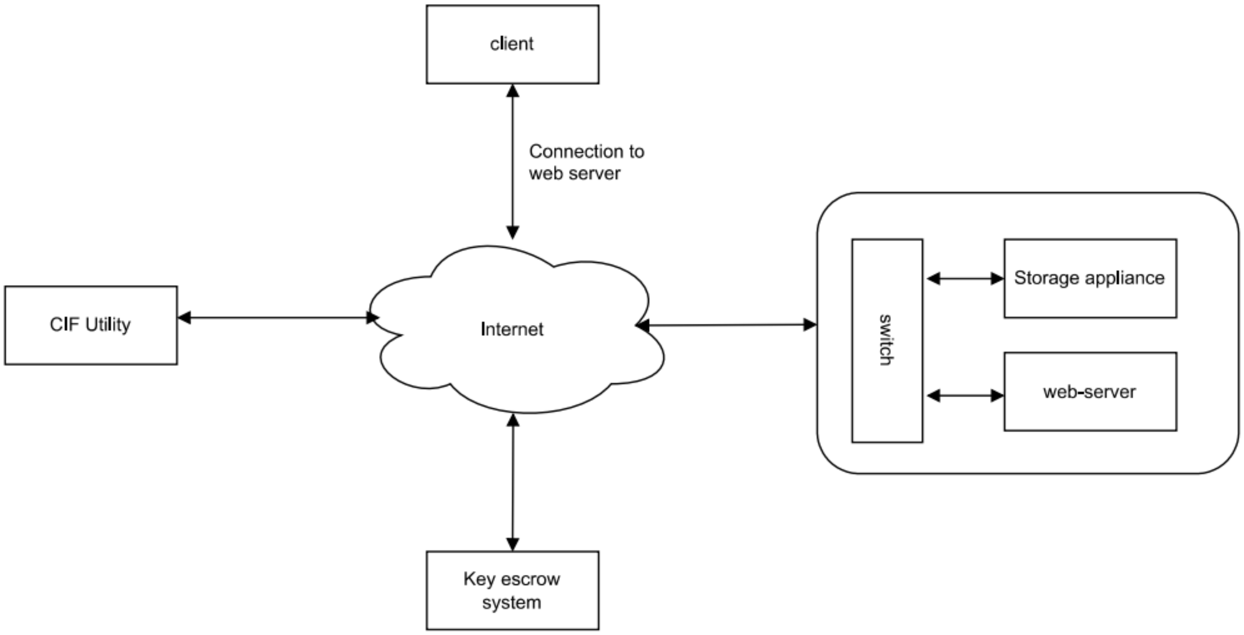
Team Members:

Vijay Varma : AI20BTECH11012

Jeevan S : CS20BTECH11047

Problem Statement

- In the current scenario of Cyber World, Knowledge is available for free to learn. This led to an increase in Attackers who used this knowledge for bad purposes.
- The Secret Keys of any company are very important as they protect confidential data. So the Secret keys need to be given equal importance as of the confidential data.
- When a breach in any company happens, these keys are targeted by the Attackers as they provide valuable information.
- So managing the above Keys is very important and the Key Escrow system does the above job securely.
- And also in case of criminal activities in any organization, with the Permission from Law and Order, one can decrypt the confidential information using the Secret Keys stored in the Key Escrow System.
- Key Escrow System provides valuable information in Continuous Internet Forensics(CIF).
- Build Key Escrow System to store cryptographic keys of web-servers securely.
- On request of authorities one can access the stored keys and decrypt the encrypted data for criminal investigation, Malware Analysis.
- Key Escrow helps immensely during Continuous Internet Forensics (CIF), and Post-incident Internet Forensics.



Problem Description

LI Functionality

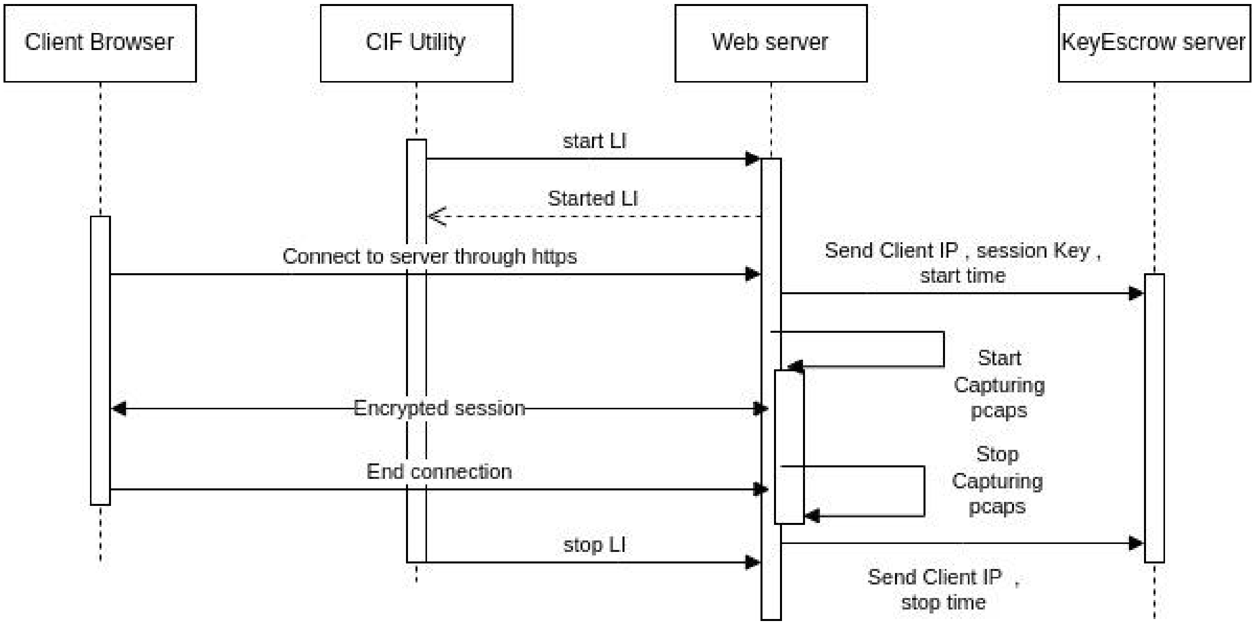
- We have set up a testbed and implemented the keyEscrow system. We have built HTTPS server, KeyEscrow server, CIF utility. In an experiment, we have started HTTPS web server, KeyEscrow server, CIF Utility.
- CIF Utility server can send “start LI " command to server for starting LI, then server starts sending newly connected clients session metadata (client IP address, session keys, start time, additional data) to keyEscrow server and also starts storing network packets in Storage Appliance. KeyEscrow server will store the session metadata of all clients who are connected to that web server after LI is started. When CIF utility sends “stop LI” web server sends connected clients end time to keyEscrow server and will not send any newly connected clients data to KeyEscrow. Now KeyEscrow has client metadata [client IP address, session keys, start time, end time].

Forensic Analysis

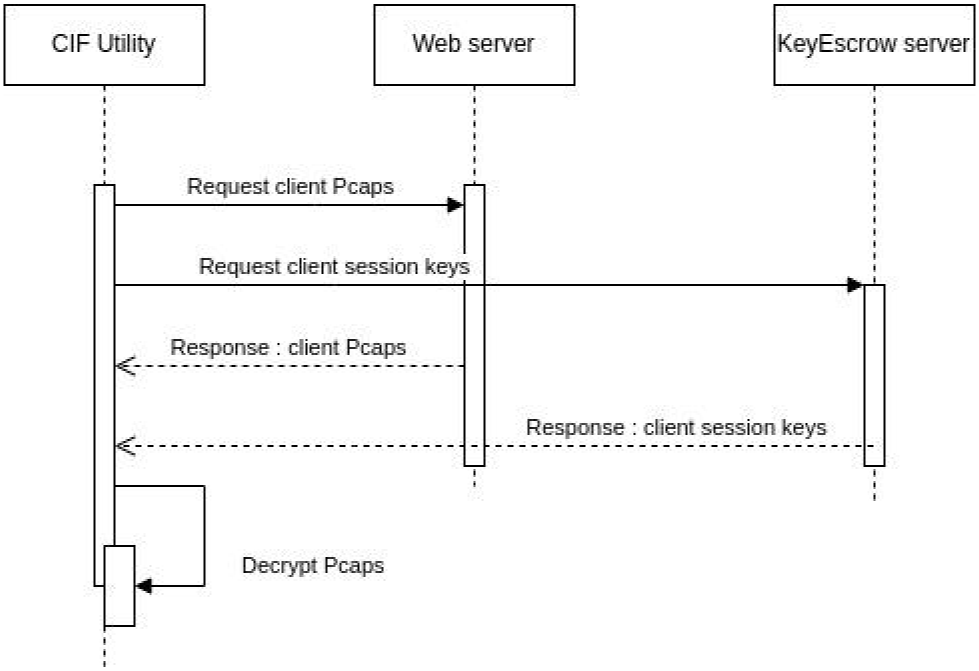
- Now CIF utility can retrieve client metadata from KeyEscrow server and capture pcaps from web server storage appliance. Now CIF will use the client session keys to decrypt the captured pcaps. CIF can use the wireshark tool or its command line tools to decrypt client encrypted sessions.

Illustrations

Sequence diagram for for LI



Sequence diagram for Forensic Analysis



3.Implementation :

- Used Python environment for all programs
- Every utility has their own certificates.

Web Server

- First we have built a simple http server and used an ssl socket library to create an ssl socket.
- So every client connection will be encrypted.
- Web servers will be able to determine if a normal client is connected or cif utility is connected.
- If normal user connect it will send a html form to client
- It receives client responses through post responses.
- If cif utility connects it will parse the request body, verify password for authentication and check whether to start LI or stop LI.
- If it is “start LI” command the web server (run a tcp dump script) will start capturing packets and it will send client session key, start time, client IP to keyEscrow server using keyEscrow post url.
- session keys are extracted using the sslkeylog module.
- If the client disconnects, then the web server will send client end time to keyEscrow and stop capturing packets.
- If the web server receives ‘stop LI’ it will terminate capturing packets and send time to keyEscrow.

KeyEscrow

- The KeyEscrow server was implemented with a python request module.
- It can receive “GET” and “POST” requests. web server can send POST requests with client data to keyEscrow server. KeyEscrow will store client data in its file system.
- CIF utility can send “GET” requests to keyEscrow server. When the keyEscrow server receives a “GET” request with an authentication password it will send client session details to the CIF utility.

CIF Utility

- CIF utility was developed using python https server and request module.
- It will send a GET request with ‘start/stop LI’ and password for authentication.
- Now it receives successful started LI response from web server.
- Now for forensic analysis, it will send get request to keyEscrow with password authentication. Once keyEscrow authenticates, CIF utility will receive client IP, session keys, start and end time.
- CIF utility can access the storage folder where pcaps are getting stored.

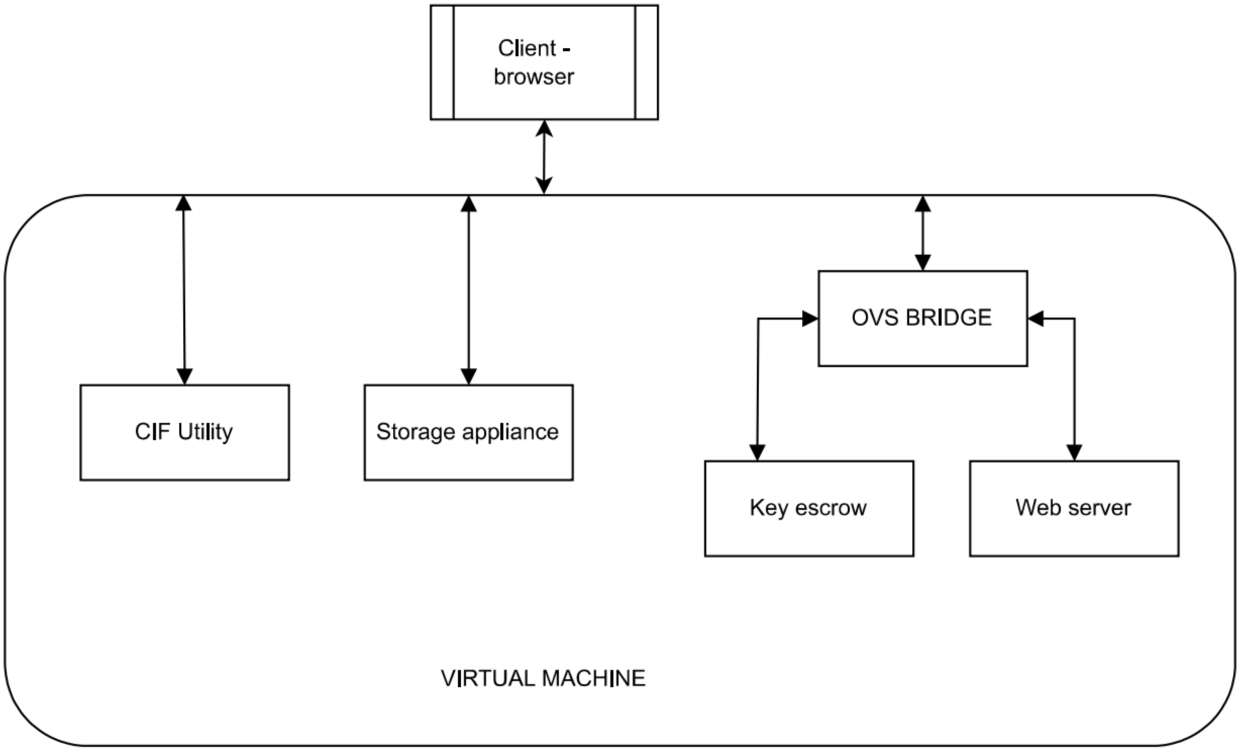
Forensic Analysis :

- Now we have session keys and pcaps .
- Using wireshark commands we can decrypt and see all the decrypted data.
- All the necessary commands are kept in one bash file.

Challenges faced

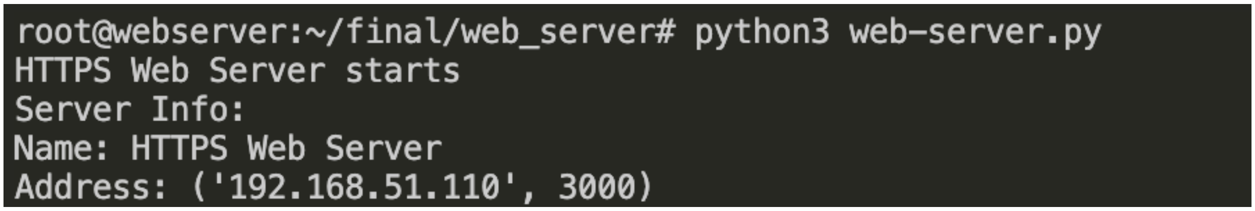
- Extracting session keys for clients. sending and receiving get and post requests with authentication password.

Simulation Setup



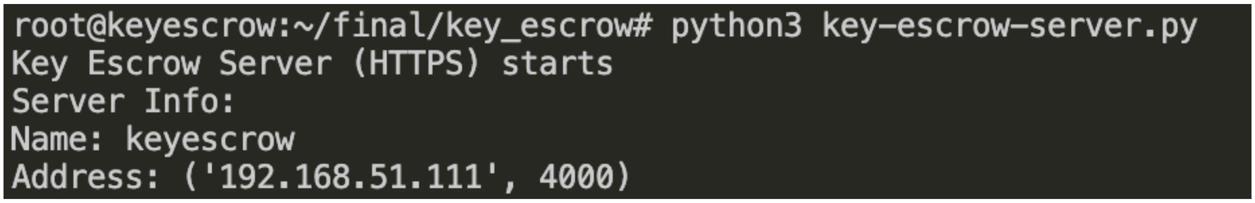
Steps to Follow for Simulation

- Starting the Web Server



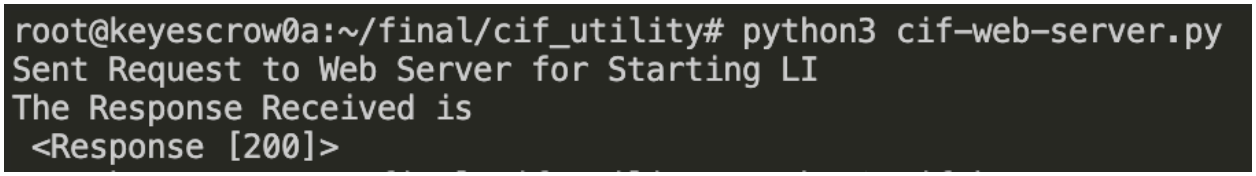
```
root@webserver:~/final/web_server# python3 web-server.py
HTTPS Web Server starts
Server Info:
Name: HTTPS Web Server
Address: ('192.168.51.110', 3000)
```

- Starting the Key Escrow Server



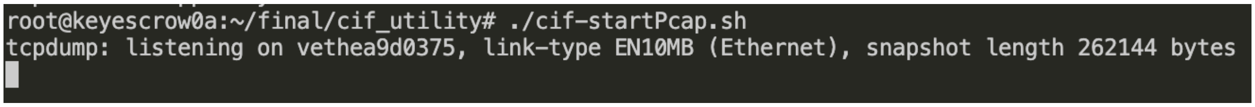
```
root@keyescrow:~/final/key_escrow# python3 key-escrow-server.py
Key Escrow Server (HTTPS) starts
Server Info:
Name: keyescrow
Address: ('192.168.51.111', 4000)
```

- Sending the StartLI from CIF to Web Server



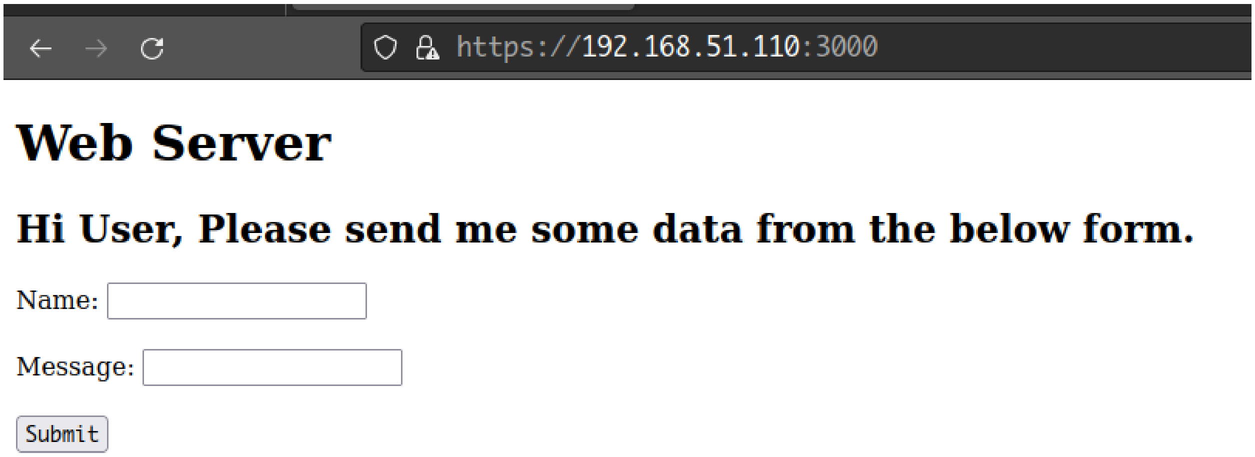
```
root@keyescrow0a:~/final/cif_utility# python3 cif-web-server.py
Sent Request to Web Server for Starting LI
The Response Received is
 <Response [200]>
```

- Start Capturing the In and Out Packets from the Web Server from the CIF



```
root@keyescrow0a:~/final/cif_utility# ./cif-startPcap.sh
tcpdump: listening on vethea9d0375, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

- Client Connected



https://192.168.51.110:3000

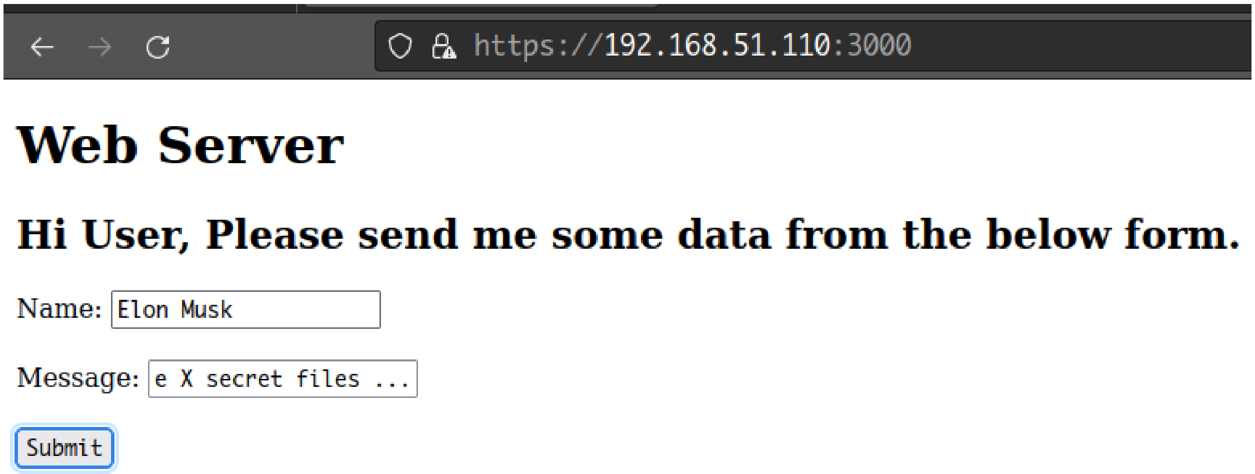
Web Server

Hi User, Please send me some data from the below form.

Name:

Message:

- Client Entered Some Data



https://192.168.51.110:3000

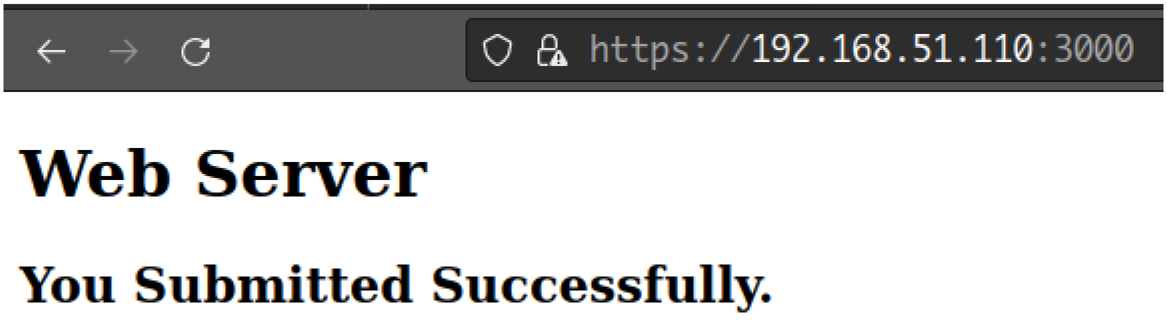
Web Server

Hi User, Please send me some data from the below form.

Name: Elon Musk

Message: e X secret files ...

- Client Submitted the Data to the Web Server.

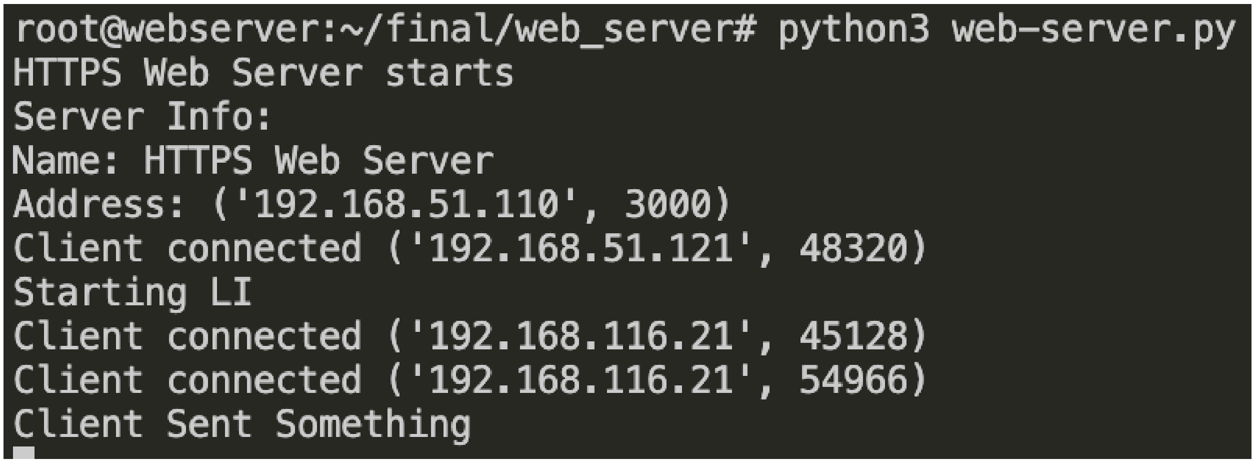


https://192.168.51.110:3000

Web Server

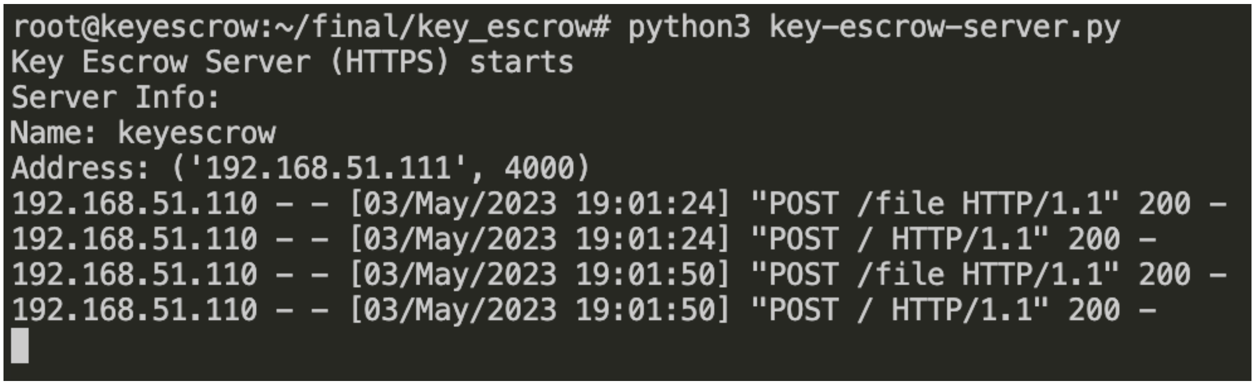
You Submitted Successfully.

- Web Server after Client Sent Something to it.



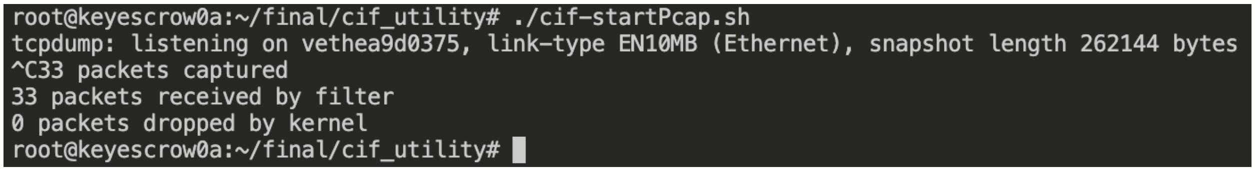
```
root@webserver:~/final/web_server# python3 web-server.py
HTTPS Web Server starts
Server Info:
Name: HTTPS Web Server
Address: ('192.168.51.110', 3000)
Client connected ('192.168.51.121', 48320)
Starting LI
Client connected ('192.168.116.21', 45128)
Client connected ('192.168.116.21', 54966)
Client Sent Something
```

- Key Escrow after Client Sent Something to the Web Server



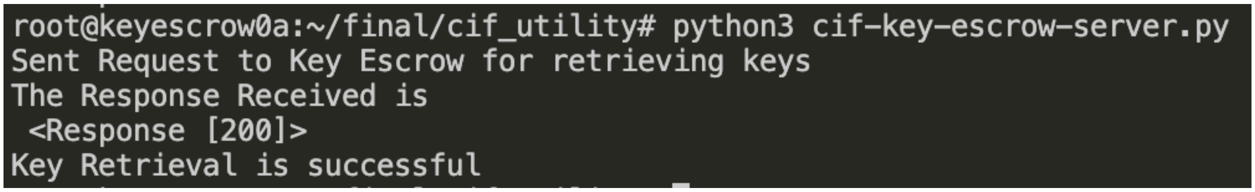
```
root@keyescrow:~/final/key_escrow# python3 key-escrow-server.py
Key Escrow Server (HTTPS) starts
Server Info:
Name: keyescrow
Address: ('192.168.51.111', 4000)
192.168.51.110 - - [03/May/2023 19:01:24] "POST /file HTTP/1.1" 200 -
192.168.51.110 - - [03/May/2023 19:01:24] "POST / HTTP/1.1" 200 -
192.168.51.110 - - [03/May/2023 19:01:50] "POST /file HTTP/1.1" 200 -
192.168.51.110 - - [03/May/2023 19:01:50] "POST / HTTP/1.1" 200 -
```

- Stop the Packet Capturing in CIF



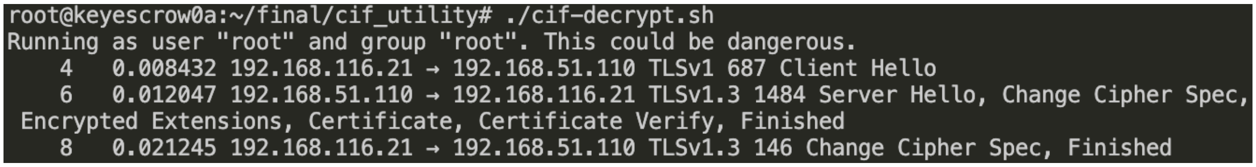
```
root@keyescrow0a:~/final/cif_utility# ./cif-startPcap.sh
tcpdump: listening on vethea9d0375, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C33 packets captured
33 packets received by filter
0 packets dropped by kernel
root@keyescrow0a:~/final/cif_utility#
```

- Get Keys from the Key Escrow Server from CIF

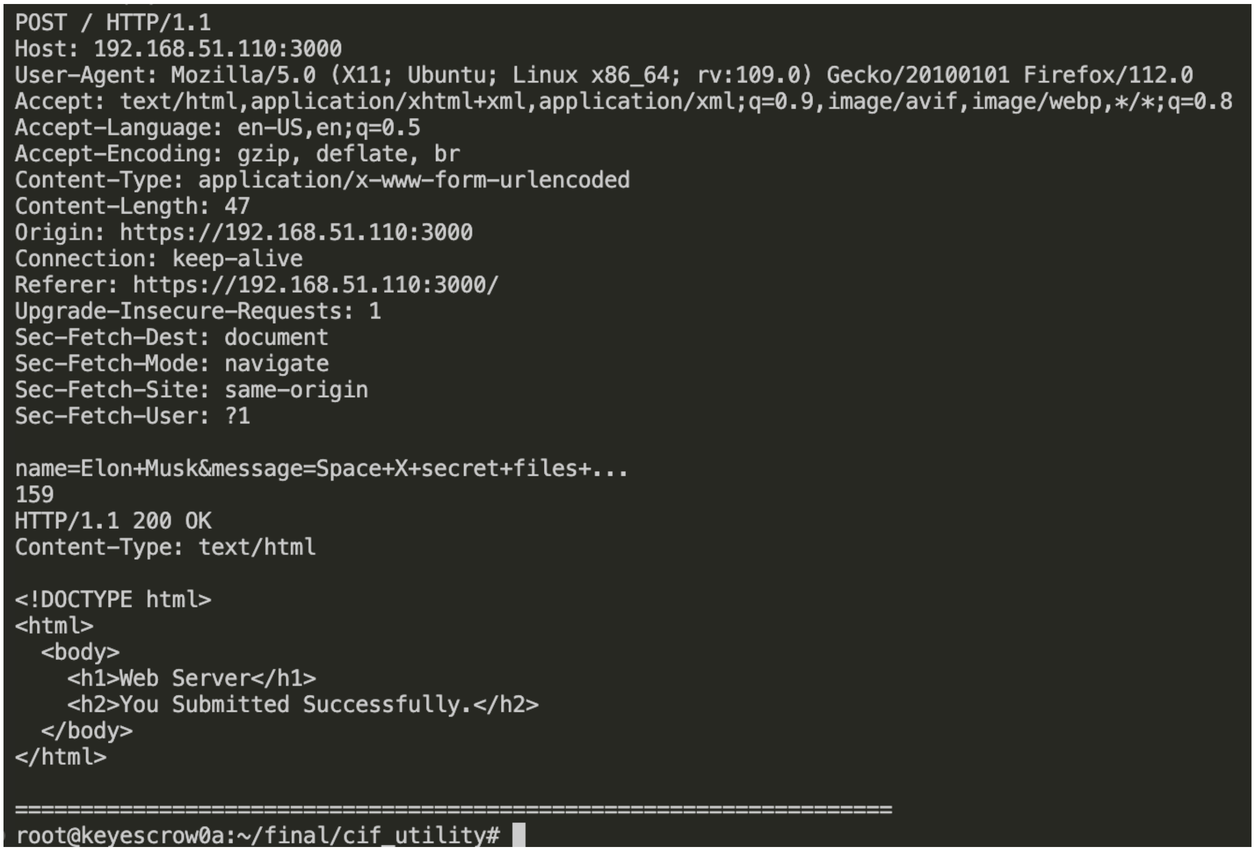


```
root@keyescrow0a:~/final/cif_utility# python3 cif-key-escrow-server.py
Sent Request to Key Escrow for retrieving keys
The Response Received is
 <Response [200]>
Key Retrieval is successful
```

- Decrypt the Keys using the keylogfile.txt and webserver.pcapng with tshark (Terminal Wireshark) or Wireshark.

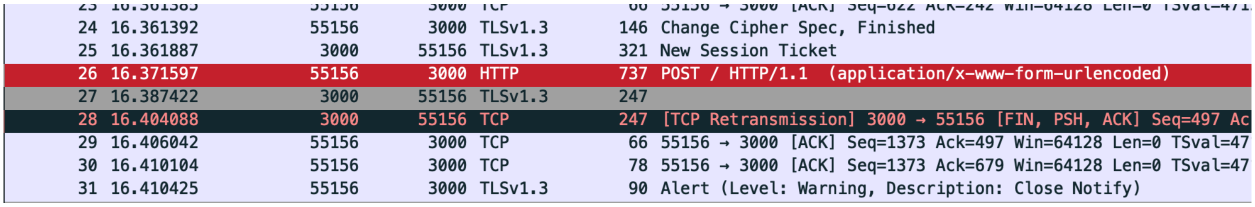


```
root@keyescrow0a:~/final/cif_utility# ./cif-decrypt.sh
Running as user "root" and group "root". This could be dangerous.
    4   0.008432 192.168.116.21 → 192.168.51.110 TLSv1 687 Client Hello
    6   0.012047 192.168.51.110 → 192.168.116.21 TLSv1.3 1484 Server Hello, Change Cipher Spec,
 Encrypted Extensions, Certificate, Certificate Verify, Finished
    8   0.021245 192.168.116.21 → 192.168.51.110 TLSv1.3 146 Change Cipher Spec, Finished
```

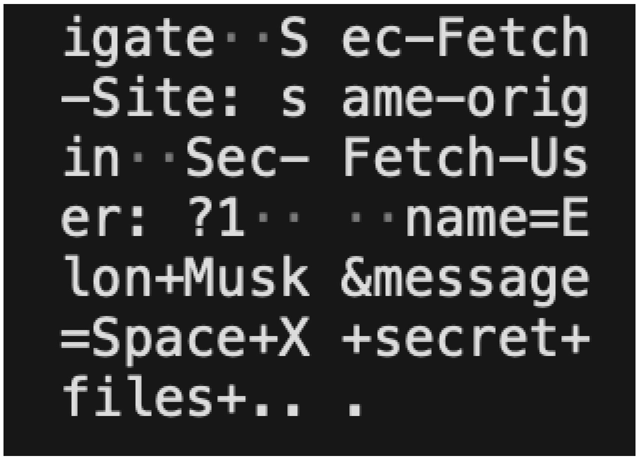


```
POST / HTTP/1.1
Host: 192.168.51.110:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/112.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 47
Origin: https://192.168.51.110:3000
Connection: keep-alive
Referer: https://192.168.51.110:3000/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1

name=Elon+Musk&message=Space+X+secret+files+...
159
HTTP/1.1 200 OK
Content-Type: text/html

<!DOCTYPE html>
<html>
  <body>
    <h1>Web Server</h1>
    <h2>You Submitted Successfully.</h2>
  </body>
</html>

===================================================================
root@keyescrow0a:~/final/cif_utility# 
```

- Wireshark Decryption



```
     23  16.361385          55156        3000  TCP         66  55156 → 3000 [ACK] Seq=622 Ack=242 Win=64128 Len=0 TSval=471
     24  16.361392          55156        3000  TLSv1.3    146  Change Cipher Spec, Finished
     25  16.361887           3000       55156  TLSv1.3    321  New Session Ticket
     26  16.371597          55156        3000  HTTP       737  POST / HTTP/1.1  (application/x-www-form-urlencoded)
     27  16.387422           3000       55156  TLSv1.3    247
     28  16.404088           3000       55156  TCP        247  [TCP Retransmission] 3000 → 55156 [FIN, PSH, ACK] Seq=497 Ac
     29  16.406042          55156        3000  TCP         66  55156 → 3000 [ACK] Seq=1373 Ack=497 Win=64128 Len=0 TSval=47
     30  16.410104          55156        3000  TCP         78  55156 → 3000 [ACK] Seq=1373 Ack=679 Win=64128 Len=0 TSval=47
     31  16.410425          55156        3000  TLSv1.3     90  Alert (Level: Warning, Description: Close Notify)
```



```
igate··S ec-Fetch
-Site: s ame-orig
in··Sec- Fetch-Us
er: ?1··  ··name=E
lon+Musk &message
=Space+X +secret+
files+.. .
```

Conclusion

- The Key Escrow Server developed works in stealth mode, and stores the TLS Session Keys of the Client and Web Server Session.
- The CIF Utility can capture the packets between Client and Web Server and can ask for the Keys of the Session using LI(Lawful Introspection) and can finally decrypt the communication between the Client and the Web Server.

Possible Future Direction

- **Scalability:** The Developed Key Escrow Server is not scalable for multiple clients. So we can work on making the Key Escrow Server and also the CIF Utility to be scalable for many users so that Key Escrow is able to store the Session Keys of many Clients at a time efficiently and can easily retrieve for a given Client IP and Session Times.
- **Security:** The Key Escrow Server and the CIF Utility were built with basic Security Measures. We can work on making the Key Escrow and the CIF Utility to be more secure.

References

- Mengbo Hou, Qiuliang Xu, Tao Ban, “Perfect Forward Secure Two-Party Key Agreement Protocol with Key Escrow”, IEEE Access, 2009.
- Pooja Bharadwaj, Harshita Pal, Bhawna Narwal, “Proposing a Key Escrow Mechanism for Real-Time access to End-to-End encryption systems in the Interest of Law Enforcement”, IEEE Access, 2018.
- Maruthi Seshidhar Inukonda, Sai Harsha Kottapalli, Bheemarjuna Reddy Tamma, Sparsh Mittal, “FENCE: A Real-Time Privacy-Preserving Solution for Enterprise Internet Forensics at Scale”, IEEE Access, 2023.
- Project Github Link - Key-Escrow

Contributions

Student 1 (Vijay)	**Student 2 (Jeevan)**
Concept Overview, Literature Review.	Concept Overview, Literature Review.
Basic HTTPS Server and Basic Setup of Containers.	Basic HTTPS Server and Basic Setup of Switches.
Key Escrow Server and LI API’s	Key Escrow Server and Capturing Packets
CIF Utility (API Handling)	CIF Utility (Packet Decoding)

Anti-plagiarism statement

We certify that this assignment/report is our own work, based on our personal study and/or research and that we have acknowledged all material and sources used in its preparation, whether they be books, articles, ChatGPT tips, packages, datasets, reports, lecture notes, and any other kind of document, electronic or personal communication. We also certify that this assignment/report has not previously been submitted for assessment/project in any other course lab, except where specific permission has been granted from all course instructors involved, or at any other time in this course, and that we have not copied in part or whole or otherwise plagiarized the work of other students in this group. We pledge to uphold the principles of honesty and responsibility at CSE@IITH. In addition, We understand my responsibility to report honor violations by other students if we become aware of it.
Names: K B Vijay Varma, Jeevan S.
Date: 03/05/23.
Signature: KBVV, JS.