Saumyaa Prajapat
21BCI0265

ASSIGNMENT 2

Tools of Kali Linux

WIRESHARK:
Wireshark is a network protocol analyzer or application that collects packets from a network connection, such as from your computer to your home office or the Internet. A packet is the name given to a separate unit of data in a typical Ethernet network.

Wireshark is the most widely used packet sniffing tool in the world. Like any other packet sniffer, Wireshark does three things:

1. Packet Capture:

Wireshark listens for network connections in real time, then retrieves the entire traffic stream – most likely tens of thousands of packets at once.

2. Filtering:

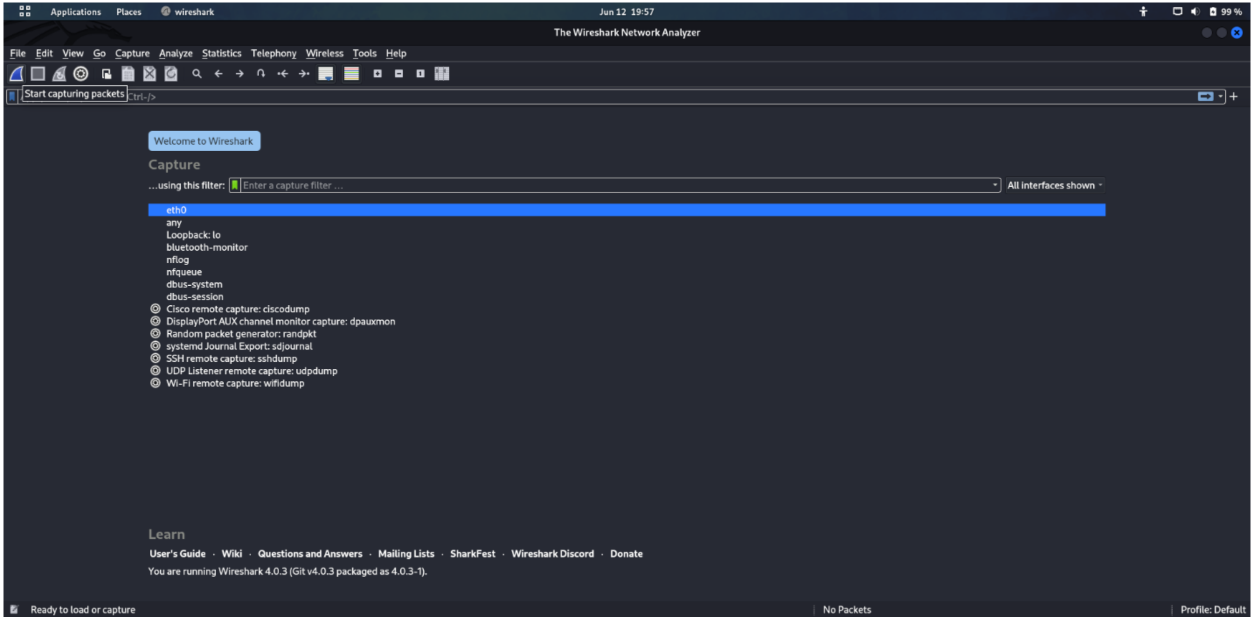
Wireshark can break down all this live random data using filters. By applying filters, you can get only the information you need.

3. Visualization:

Wireshark, like any good packet sniffer, lets you drill down to the heart of network packets. It also allows you to view entire conversations and network streams.

Step 1:

Open the Wireshark application and start capturing interfaces (the eth0 network connection in this case – check to see if some signal is present in the interface).



...using this filter: Enter a capture filter ... All interfaces shown

eth0
any
Loopback: lo
bluetooth-monitor
nflog
nfqueue
dbus-system
dbus-session
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal
SSH remote capture: sshdump
UDP Listener remote capture: udpdump
Wi-Fi remote capture: wifidump

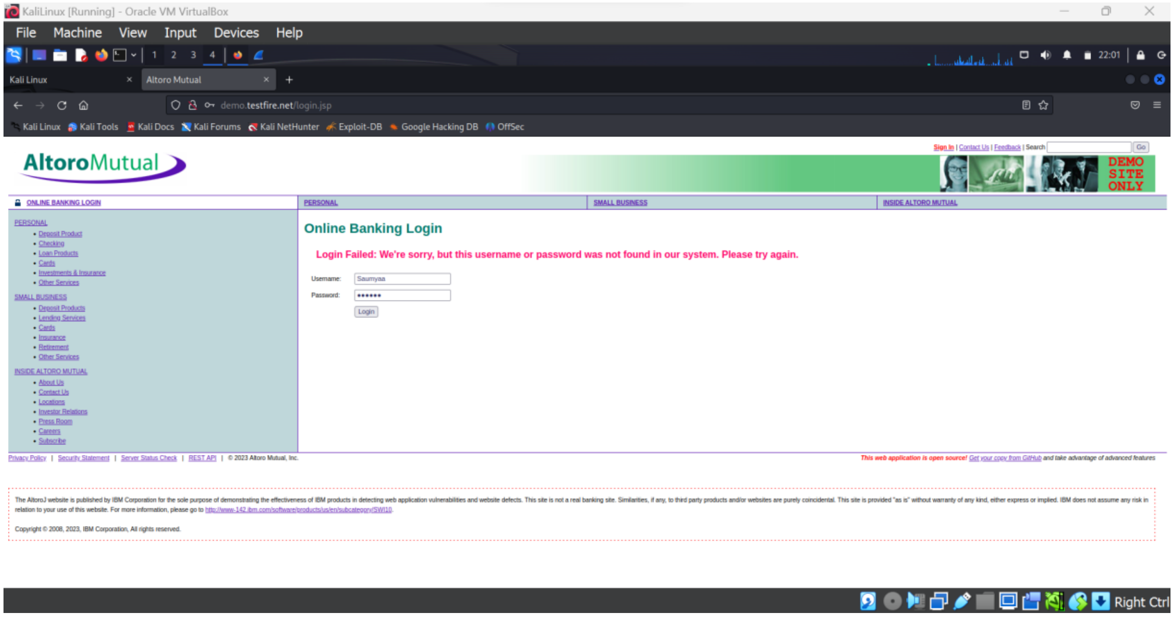
You are running Wireshark 4.0.3 (Git v4.0.3 packaged as 4.0.3-1).

Ready to load or capture | No Packets | Profile: Default

Step 2:

Saumyaa Prajapat
21BCI0265

Open a web browser and Make sure your browser's cache is cleared. Go to http://demo.testfire.net Note that this web page uses HTTP. Click on the ONLINE BANKING LOGIN link on top left.
Enter your First name as the Username, and your Reg no as the password.



Online Banking Login

Login Failed: We're sorry, but this username or password was not found in our system. Please try again.

Username: Saumyaa
Password: ••••••
Login

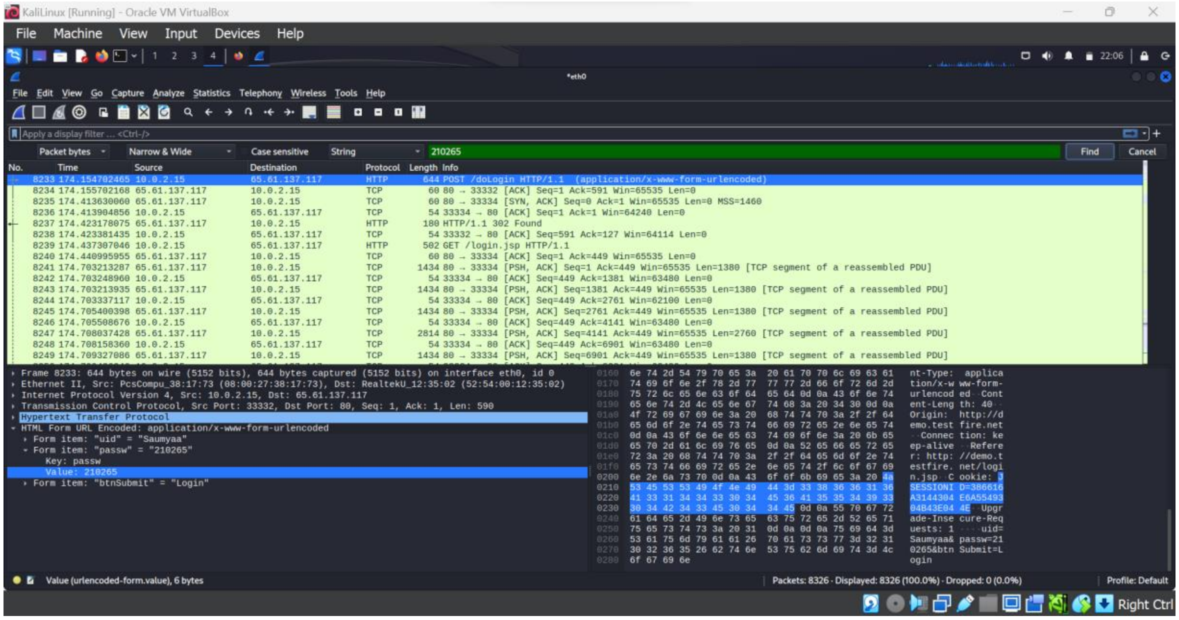
This web application is open source! Get your copy from GitHub and take advantage of advanced features

The AltoroJ website is published by IBM Corporation for the sole purpose of demonstrating the effectiveness of IBM products in detecting web application vulnerabilities and website defects. This site is not a real banking site. Similarities, if any, to third party products and/or websites are purely coincidental. This site is provided "as is" without warranty of any kind, either express or implied. IBM does not assume any risk in relation to your use of this website. For more information, please go to http://www-142.ibm.com/software/products/us/en/subcategory/SWI10.

Copyright © 2008, 2023, IBM Corporation, All rights reserved.

Step 3:

Stop capturing traffic on Wireshark by clicking on the top. Save the captured data into a file for analysis.


KaliLinux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
22:06
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Find
Cancel
Right Ctrl

Packet bytes · Narrow & Wide · Case sensitive · String · 210265

```
No.  Time            Source          Destination     Protocol Length Info
8233 174.154702465 10.0.2.15      65.61.137.117   HTTP   644 POST /doLogin HTTP/1.1  (application/x-www-form-urlencoded)
8234 174.155702168 65.61.137.117  10.0.2.15       TCP     60 80 → 33332 [ACK] Seq=1 Ack=591 Win=65535 Len=0
8235 174.413630060 65.61.137.117  10.0.2.15       TCP     60 80 → 33334 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
8236 174.413904856 10.0.2.15      65.61.137.117   TCP     54 33334 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
8237 174.423178075 65.61.137.117  10.0.2.15       HTTP   180 HTTP/1.1 302 Found
8238 174.423381435 10.0.2.15      65.61.137.117   TCP     54 33332 → 80 [ACK] Seq=591 Ack=127 Win=64114 Len=0
8239 174.437307046 10.0.2.15      65.61.137.117   HTTP   502 GET /login.jsp HTTP/1.1
8240 174.440995955 65.61.137.117  10.0.2.15       TCP     60 80 → 33334 [ACK] Seq=1 Ack=449 Win=65535 Len=0
8241 174.703213287 65.61.137.117  10.0.2.15       TCP   1434 80 → 33334 [PSH, ACK] Seq=1 Ack=449 Win=65535 Len=1380 [TCP segment of a reassembled PDU]
8242 174.703248960 10.0.2.15      65.61.137.117   TCP     54 33334 → 80 [ACK] Seq=449 Ack=1381 Win=63480 Len=0
8243 174.703213935 65.61.137.117  10.0.2.15       TCP   1434 80 → 33334 [PSH, ACK] Seq=1381 Ack=449 Win=65535 Len=1380 [TCP segment of a reassembled PDU]
8244 174.703337117 10.0.2.15      65.61.137.117   TCP     54 33334 → 80 [ACK] Seq=449 Ack=2761 Win=62100 Len=0
8245 174.705400398 65.61.137.117  10.0.2.15       TCP   1434 80 → 33334 [PSH, ACK] Seq=2761 Ack=449 Win=65535 Len=1380 [TCP segment of a reassembled PDU]
8246 174.705508676 10.0.2.15      65.61.137.117   TCP     54 33334 → 80 [ACK] Seq=449 Ack=4141 Win=63480 Len=0
8247 174.708037428 65.61.137.117  10.0.2.15       TCP   2814 80 → 33334 [PSH, ACK] Seq=4141 Ack=449 Win=65535 Len=2760 [TCP segment of a reassembled PDU]
8248 174.708158360 10.0.2.15      65.61.137.117   TCP     54 33334 → 80 [ACK] Seq=449 Ack=6901 Win=63480 Len=0
8249 174.709327086 65.61.137.117  10.0.2.15       TCP   1434 80 → 33334 [PSH, ACK] Seq=6901 Ack=449 Win=65535 Len=1380 [TCP segment of a reassembled PDU]
```

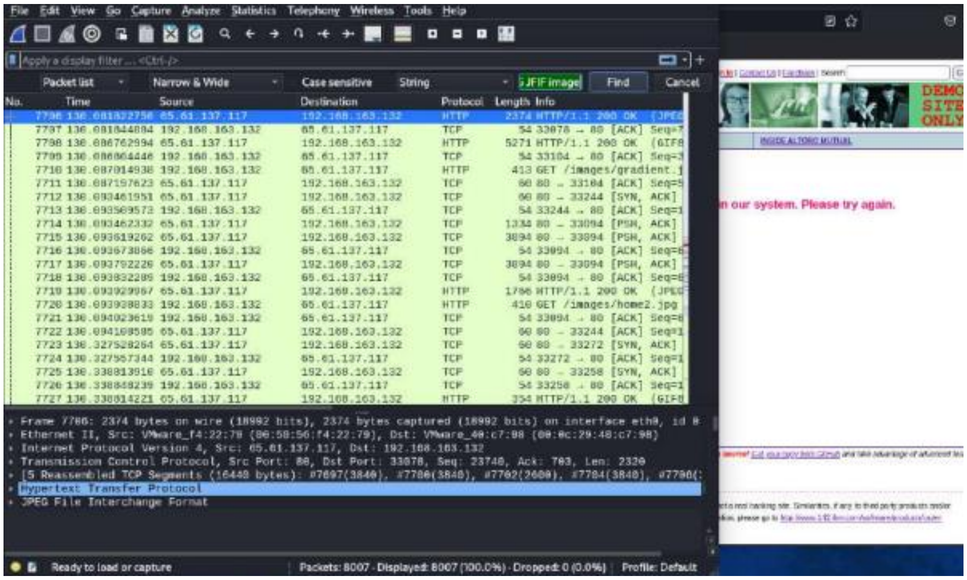
```
› Frame 8233: 644 bytes on wire (5152 bits), 644 bytes captured (5152 bits) on interface eth0, id 0
› Ethernet II, Src: PcsCompu_38:17:73 (08:00:27:38:17:73), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
› Internet Protocol Version 4, Src: 10.0.2.15, Dst: 65.61.137.117
› Transmission Control Protocol, Src Port: 33332, Dst Port: 80, Seq: 1, Ack: 1, Len: 590
› Hypertext Transfer Protocol
⌄ HTML Form URL Encoded: application/x-www-form-urlencoded
  › Form item: "uid" = "Saumyaa"
  ⌄ Form item: "passw" = "210265"
      Key: passw
      Value: 210265
  › Form item: "btnSubmit" = "Login"
```

```
0160  6e 74 2d 54 79 70 65 3a  20 61 70 70 6c 69 63 61   nt-Type:  applica
0170  74 69 6f 6e 2f 78 2d 77  77 77 2d 66 6f 72 6d 2d   tion/x-w ww-form-
0180  75 72 6c 65 6e 63 6f 64  65 64 0d 0a 43 6f 6e 74   urlencod ed··Cont
0190  65 6e 74 2d 4c 65 6e 67  74 68 3a 20 34 30 0d 0a   ent-Leng th: 40··
01a0  4f 72 69 67 69 6e 3a 20  68 74 74 70 3a 2f 2f 64   Origin:  http://d
01b0  65 6d 6f 2e 74 65 73 74  66 69 72 65 2e 6e 65 74   emo.test fire.net
01c0  0d 0a 43 6f 6e 6e 65 63  74 69 6f 6e 3a 20 6b 65   ··Connec tion: ke
01d0  65 70 2d 61 6c 69 76 65  0d 0a 52 65 66 65 72 65   ep-alive ··Refere
01e0  72 3a 20 68 74 74 70 3a  2f 2f 64 65 6d 6f 2e 74   r: http: //demo.t
01f0  65 73 74 66 69 72 65 2e  6e 65 74 2f 6c 6f 67 69   estfire. net/logi
0200  6e 2e 6a 73 70 0d 0a 43  6f 6f 6b 69 65 3a 20 4a   n.jsp··C ookie: J
0210  53 45 53 53 49 4f 4e 49  44 3d 33 38 36 36 31 36   SESSIONI D=386616
0220  41 33 31 34 34 33 30 34  45 36 41 35 35 34 39 33   A3144304 E6A55493
0230  30 34 42 34 33 45 30 34  34 45 0d 0a 55 70 67 72   04B43E04 4E··Upgr
0240  61 64 65 2d 49 6e 73 65  63 75 72 65 2d 52 65 71   ade-Inse cure-Req
0250  75 65 73 74 73 3a 20 31  0d 0a 0d 0a 75 69 64 3d   uests: 1 ····uid=
0260  53 61 75 6d 79 61 61 26  70 61 73 73 77 3d 32 31   Saumyaa& passw=21
0270  30 32 36 35 26 62 74 6e  53 75 62 6d 69 74 3d 4c   0265&btn Submit=L
0280  6f 67 69 6e                                        ogin
```

Value (urlencoded-form.value), 6 bytes · Packets: 8326 · Displayed: 8326 (100.0%) · Dropped: 0 (0.0%) · Profile: Default

Step 4:

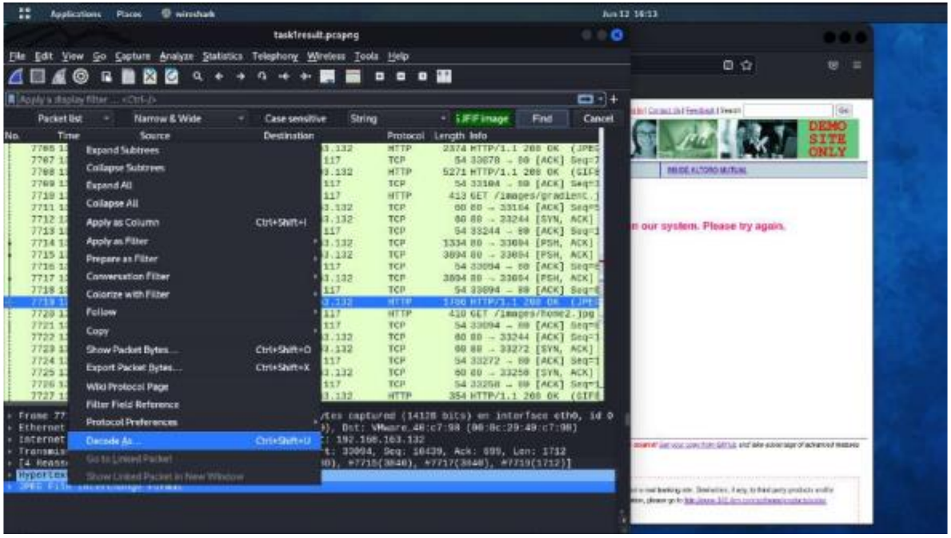
Filter "http" packets. Now, go to Edit -> Find Packet. Now search for Packet list with the string "JPEG JFIF image" without quotes.
Photo extraction –

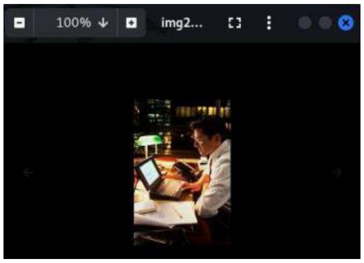

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Packet list
Narrow & Wide
Case sensitive
String
JFIF image
Find
Cancel
DEMO SITE ONLY
INSIDE ALTORO MUTUAL
n our system. Please try again.

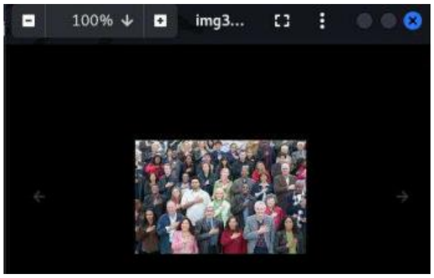
```
No.   Time           Source           Destination      Protocol Length Info
 7706 136.081822756  65.61.137.117    192.168.163.132  HTTP     2374 HTTP/1.1 200 OK  (JPEG
 7707 136.081844884  192.168.163.132  65.61.137.117    TCP        54 33078 → 80 [ACK] Seq=7
 7708 136.086762994  65.61.137.117    192.168.163.132  HTTP     5271 HTTP/1.1 200 OK  (GIF8
 7709 136.086864446  192.168.163.132  65.61.137.117    TCP        54 33104 → 80 [ACK] Seq=3
 7710 136.087014938  192.168.163.132  65.61.137.117    HTTP      413 GET /images/gradient.j
 7711 136.087197623  65.61.137.117    192.168.163.132  TCP        60 80 → 33104 [ACK] Seq=5
 7712 136.093461951  65.61.137.117    192.168.163.132  TCP        60 80 → 33244 [SYN, ACK]
 7713 136.093569573  192.168.163.132  65.61.137.117    TCP        54 33244 → 80 [ACK] Seq=1
 7714 136.093462332  65.61.137.117    192.168.163.132  TCP      1334 80 → 33094 [PSH, ACK]
 7715 136.093619262  65.61.137.117    192.168.163.132  TCP      3894 80 → 33094 [PSH, ACK]
 7716 136.093673866  192.168.163.132  65.61.137.117    TCP        54 33094 → 80 [ACK] Seq=6
 7717 136.093792226  65.61.137.117    192.168.163.132  TCP      3894 80 → 33094 [PSH, ACK]
 7718 136.093832209  192.168.163.132  65.61.137.117    TCP        54 33094 → 80 [ACK] Seq=6
 7719 136.093929967  65.61.137.117    192.168.163.132  HTTP     1786 HTTP/1.1 200 OK  (JPEG
 7720 136.093928833  192.168.163.132  65.61.137.117    HTTP      410 GET /images/home2.jpg
 7721 136.094023619  192.168.163.132  65.61.137.117    TCP        54 33094 → 80 [ACK] Seq=6
 7722 136.094169585  65.61.137.117    192.168.163.132  TCP        60 80 → 33244 [ACK] Seq=1
 7723 136.327528264  65.61.137.117    192.168.163.132  TCP        60 80 → 33272 [SYN, ACK]
 7724 136.327567344  192.168.163.132  65.61.137.117    TCP        54 33272 → 80 [ACK] Seq=1
 7725 136.338813916  65.61.137.117    192.168.163.132  TCP        60 80 → 33258 [SYN, ACK]
 7726 136.338848239  192.168.163.132  65.61.137.117    TCP        54 33258 → 80 [ACK] Seq=1
 7727 136.338814221  65.61.137.117    192.168.163.132  HTTP      354 HTTP/1.1 200 OK  (GIF8

› Frame 7706: 2374 bytes on wire (18992 bits), 2374 bytes captured (18992 bits) on interface eth0, id 0
› Ethernet II, Src: VMware_f4:22:79 (00:50:56:f4:22:79), Dst: VMware_40:c7:98 (00:0c:29:40:c7:98)
› Internet Protocol Version 4, Src: 65.61.137.117, Dst: 192.168.163.132
› Transmission Control Protocol, Src Port: 80, Dst Port: 33078, Seq: 23740, Ack: 703, Len: 2320
› [5 Reassembled TCP Segments (16440 bytes): #7697(3840), #7700(3840), #7702(2600), #7704(3840), #7706(
› Hypertext Transfer Protocol
› JPEG File Interchange Format

Ready to load or capture    Packets: 8007 · Displayed: 8007 (100.0%) · Dropped: 0 (0.0%)    Profile: Default
```

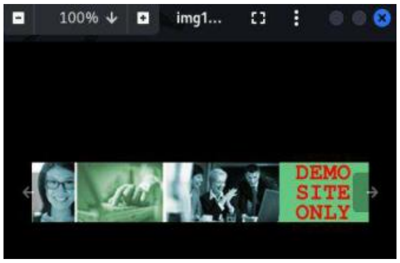
Step 5:

Go to the Packet Details area. Right click on the "JPEG File Interchange Format" and click on the “Export Packet Bytes” option. Save the packet as img1.jpg on the desktop. Repeat this step for any three packets and check whether these images were the ones you saw in the website. Save them as img2.jpg and img3.jpg, respectively.



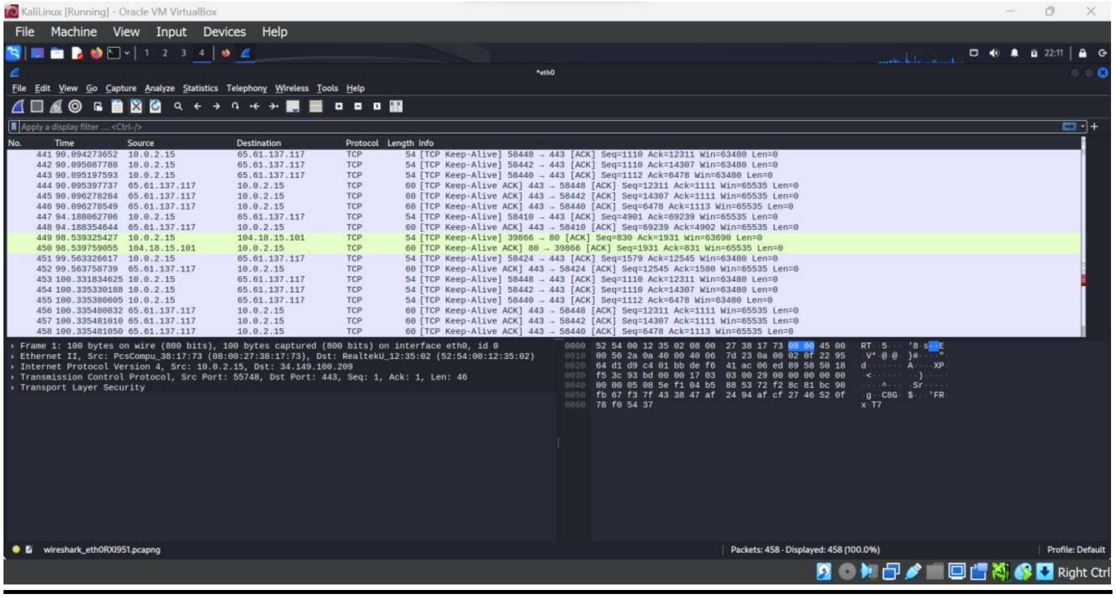






With HTTPS – https://demo.testfire.net/

Saumyaa Prajapat
21BCI0265

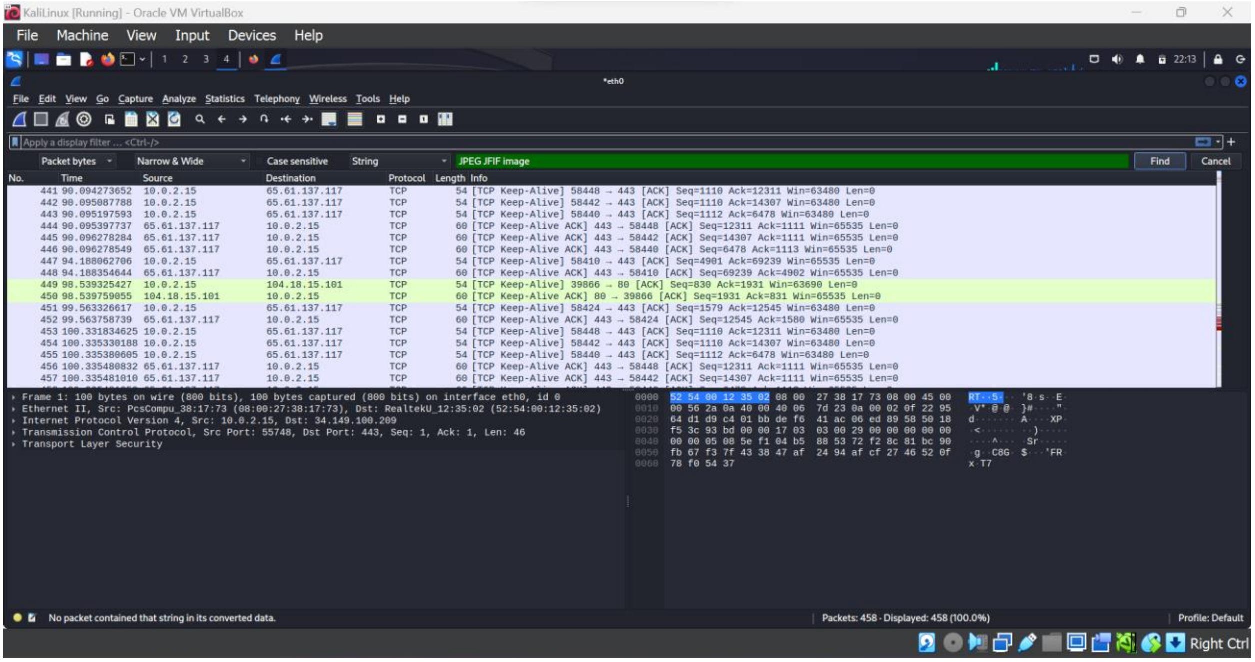

KaliLinux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
22:11
Right Ctrl

```
No.  Time          Source          Destination     Protocol Length Info
 441 90.094273652  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58448 → 443 [ACK] Seq=1110 Ack=12311 Win=63480 Len=0
 442 90.095087788  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58442 → 443 [ACK] Seq=1110 Ack=14307 Win=63480 Len=0
 443 90.095197593  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58440 → 443 [ACK] Seq=1112 Ack=6478 Win=63480 Len=0
 444 90.095397737  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58448 [ACK] Seq=12311 Ack=1111 Win=65535 Len=0
 445 90.096278284  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58442 [ACK] Seq=14307 Ack=1111 Win=65535 Len=0
 446 90.096278549  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58440 [ACK] Seq=6478 Ack=1113 Win=65535 Len=0
 447 94.188062706  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58410 → 443 [ACK] Seq=4901 Ack=69239 Win=65535 Len=0
 448 94.188354644  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58410 [ACK] Seq=69239 Ack=4902 Win=65535 Len=0
 449 98.539325427  10.0.2.15       104.18.15.101   TCP      54 [TCP Keep-Alive] 39866 → 80 [ACK] Seq=830 Ack=1931 Win=63690 Len=0
 450 98.539759055  104.18.15.101   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 80 → 39866 [ACK] Seq=1931 Ack=831 Win=65535 Len=0
 451 99.563326617  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58424 → 443 [ACK] Seq=1579 Ack=12545 Win=63480 Len=0
 452 99.563758739  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58424 [ACK] Seq=12545 Ack=1580 Win=65535 Len=0
 453 100.331834625 10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58448 → 443 [ACK] Seq=1110 Ack=12311 Win=63480 Len=0
 454 100.335330188 10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58442 → 443 [ACK] Seq=1110 Ack=14307 Win=63480 Len=0
 455 100.335380605 10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58440 → 443 [ACK] Seq=1112 Ack=6478 Win=63480 Len=0
 456 100.335480832 65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58448 [ACK] Seq=12311 Ack=1111 Win=65535 Len=0
 457 100.335481010 65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58442 [ACK] Seq=14307 Ack=1111 Win=65535 Len=0
 458 100.335481050 65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58440 [ACK] Seq=6478 Ack=1113 Win=65535 Len=0
```

```
› Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface eth0, id 0
› Ethernet II, Src: PcsCompu_38:17:73 (08:00:27:38:17:73), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
› Internet Protocol Version 4, Src: 10.0.2.15, Dst: 34.149.100.209
› Transmission Control Protocol, Src Port: 55748, Dst Port: 443, Seq: 1, Ack: 1, Len: 46
› Transport Layer Security
```

```
0000  52 54 00 12 35 02 08 00  27 38 17 73 08 00 45 00   RT··5···'8·s··E·
0010  00 56 2a 0a 40 00 40 06  7d 23 0a 00 02 0f 22 95   ·V*·@·@·}#····"·
0020  64 d1 d9 c4 01 bb de f6  41 ac 06 ed 89 58 50 18   d·······A····XP·
0030  f5 3c 93 bd 00 00 17 03  03 00 29 00 00 00 00 00   ·<········)·····
0040  00 00 05 08 5e f1 04 b5  88 53 72 f2 8c 81 bc 90   ····^····Sr·····
0050  fb 67 f3 7f 43 38 47 af  24 94 af cf 27 46 52 0f   ·g··C8G·$···'FR·
0060  78 f0 54 37                                        x·T7
```

wireshark_eth0RXI951.pcapng
Packets: 458 · Displayed: 458 (100.0%)
Profile: Default


KaliLinux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
*eth0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
22:13
Find
Cancel
Right Ctrl

Packet bytes | Narrow & Wide | Case sensitive | String | JPEG JFIF image

```
No.  Time          Source          Destination     Protocol Length Info
 441 90.094273652  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58448 → 443 [ACK] Seq=1110 Ack=12311 Win=63480 Len=0
 442 90.095087788  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58442 → 443 [ACK] Seq=1110 Ack=14307 Win=63480 Len=0
 443 90.095197593  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58440 → 443 [ACK] Seq=1112 Ack=6478 Win=63480 Len=0
 444 90.095397737  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58448 [ACK] Seq=12311 Ack=1111 Win=65535 Len=0
 445 90.096278284  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58442 [ACK] Seq=14307 Ack=1111 Win=65535 Len=0
 446 90.096278549  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58440 [ACK] Seq=6478 Ack=1113 Win=65535 Len=0
 447 94.188062706  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58410 → 443 [ACK] Seq=4901 Ack=69239 Win=65535 Len=0
 448 94.188354644  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58410 [ACK] Seq=69239 Ack=4902 Win=65535 Len=0
 449 98.539325427  10.0.2.15       104.18.15.101   TCP      54 [TCP Keep-Alive] 39866 → 80 [ACK] Seq=830 Ack=1931 Win=63690 Len=0
 450 98.539759055  104.18.15.101   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 80 → 39866 [ACK] Seq=1931 Ack=831 Win=65535 Len=0
 451 99.563326617  10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58424 → 443 [ACK] Seq=1579 Ack=12545 Win=63480 Len=0
 452 99.563758739  65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58424 [ACK] Seq=12545 Ack=1580 Win=65535 Len=0
 453 100.331834625 10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58448 → 443 [ACK] Seq=1110 Ack=12311 Win=63480 Len=0
 454 100.335330188 10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58442 → 443 [ACK] Seq=1110 Ack=14307 Win=63480 Len=0
 455 100.335380605 10.0.2.15       65.61.137.117   TCP      54 [TCP Keep-Alive] 58440 → 443 [ACK] Seq=1112 Ack=6478 Win=63480 Len=0
 456 100.335480832 65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58448 [ACK] Seq=12311 Ack=1111 Win=65535 Len=0
 457 100.335481010 65.61.137.117   10.0.2.15       TCP      60 [TCP Keep-Alive ACK] 443 → 58442 [ACK] Seq=14307 Ack=1111 Win=65535 Len=0
```

```
› Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface eth0, id 0
› Ethernet II, Src: PcsCompu_38:17:73 (08:00:27:38:17:73), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
› Internet Protocol Version 4, Src: 10.0.2.15, Dst: 34.149.100.209
› Transmission Control Protocol, Src Port: 55748, Dst Port: 443, Seq: 1, Ack: 1, Len: 46
› Transport Layer Security
```

```
0000  52 54 00 12 35 02 08 00  27 38 17 73 08 00 45 00   RT··5···'8·s··E·
0010  00 56 2a 0a 40 00 40 06  7d 23 0a 00 02 0f 22 95   ·V*·@·@·}#····"·
0020  64 d1 d9 c4 01 bb de f6  41 ac 06 ed 89 58 50 18   d·······A····XP·
0030  f5 3c 93 bd 00 00 17 03  03 00 29 00 00 00 00 00   ·<········)·····
0040  00 00 05 08 5e f1 04 b5  88 53 72 f2 8c 81 bc 90   ····^····Sr·····
0050  fb 67 f3 7f 43 38 47 af  24 94 af cf 27 46 52 0f   ·g··C8G·$···'FR·
0060  78 f0 54 37                                        x·T7
```

No packet contained that string in its converted data.
Packets: 458 · Displayed: 458 (100.0%)
Profile: Default

JOHN THE RIPPER:

John the Ripper, commonly known as "John", is in fact available in Kali Linux and offers three main methods to crack passwords:

Single crack mode:

- In single crack mode, John the Ripper attempts to crack passwords by guessing the password directly from the hash function without using word lists or rules.
- It uses built-in rules and techniques to make educated guesses based on common password patterns and weaknesses.

- Single Crack mode is usually the fastest method, but it may not be as effective as using a word list for difficult-to-guess passwords.

Word list mode:

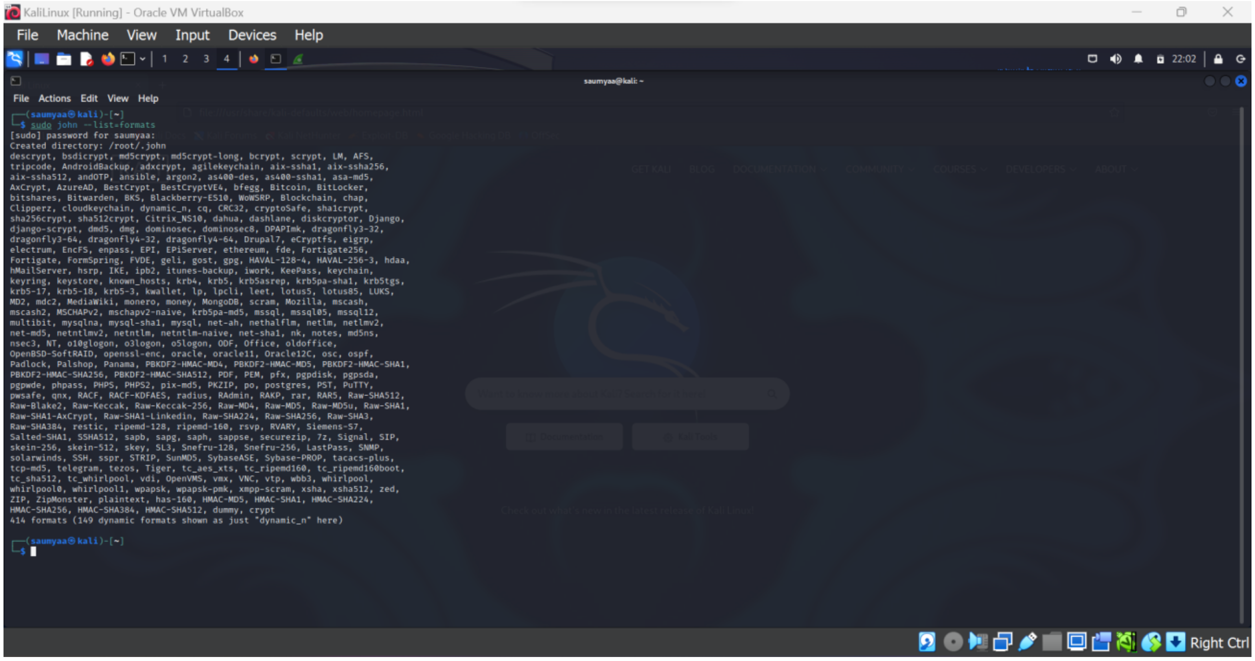
- Word List mode is one of the most used methods with John the Ripper. This involves providing a list of potential passwords (called a word list or dictionary) and John will attempt to decrypt the passwords in the hash file by comparing them with the words in the list.
- John supports different wordlist formats and users can also create custom wordlists.

Incremental mode:

- Incremental mode is the brute force method used by John the Ripper. It systematically generates and tests all possible password combinations, starting with the minimum length and gradually increasing to the maximum length.
- Incremental mode is the longest and most resource-intensive method and is often used as a last resort when other methods fail.
- You can customize the character set, minimum and maximum password length, and other settings for incremental mode.

To see the list of hashes that John can crack:

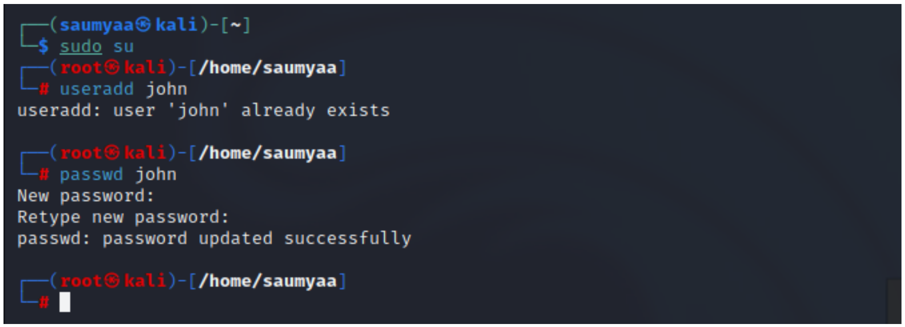
sudo john --list=formats


KaliLinux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
File Actions Edit View Help
Right Ctrl

```
┌──(saumyaa㉿kali)-[~]
└─$ sudo john --list=formats
[sudo] password for saumyaa:
Created directory: /root/.john
descrypt, bsdicrypt, md5crypt, md5crypt-long, bcrypt, scrypt, LM, AFS,
tripcode, AndroidBackup, adxcrypt, agilekeychain, aix-ssha1, aix-ssha256,
aix-ssha512, andOTP, ansible, argon2, as400-des, as400-ssha1, asa-md5,
AxCrypt, AzureAD, BestCrypt, BestCryptVE4, bfegg, Bitcoin, BitLocker,
bitshares, Bitwarden, BKS, Blackberry-ES10, WoWSRP, Blockchain, chap,
Clipperz, cloudkeychain, dynamic_n, cq, CRC32, cryptoSafe, sha1crypt,
sha256crypt, sha512crypt, Citrix_NS10, dahua, dashlane, diskcryptor, Django,
django-scrypt, dmd5, dmg, dominosec, dominosec8, DPAPImk, dragonfly3-32,
dragonfly3-64, dragonfly4-32, dragonfly4-64, Drupal7, eCryptfs, eigrp,
electrum, EncFS, enpass, EPI, EPiServer, ethereum, fde, Fortigate256,
Fortigate, FormSpring, FVDE, geli, gost, gpg, HAVAL-128-4, HAVAL-256-3, hdaa,
hMailServer, hsrp, IKE, ipb2, itunes-backup, iwork, KeePass, keychain,
keyring, keystore, known_hosts, krb4, krb5, krb5asrep, krb5pa-sha1, krb5tgs,
krb5-17, krb5-18, krb5-3, kwallet, lp, lpcli, leet, lotus5, lotus85, LUKS,
MD2, mdc2, MediaWiki, monero, money, MongoDB, scram, Mozilla, mscash,
mscash2, MSCHAPv2, mschapv2-naive, krb5pa-md5, mssql, mssql05, mssql12,
multibit, mysqlna, mysql-sha1, mysql, net-ah, nethalflm, netlm, netlmv2,
net-md5, netntlmv2, netntlm, netntlm-naive, net-sha1, nk, notes, md5ns,
nsec3, NT, o10glogon, o3logon, o5logon, ODF, Office, oldoffice,
OpenBSD-SoftRAID, openssl-enc, oracle, oracle11, Oracle12C, osc, ospf,
Padlock, Palshop, Panama, PBKDF2-HMAC-MD4, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1,
PBKDF2-HMAC-SHA256, PBKDF2-HMAC-SHA512, PDF, PEM, pfx, pgpdisk, pgpsda,
pgpwde, phpass, PHPS, PHPS2, pix-md5, PKZIP, po, postgres, PST, PuTTY,
pwsafe, qnx, RACF, RACF-KDFAES, radius, RAdmin, RAKP, rar, RAR5, Raw-SHA512,
Raw-Blake2, Raw-Keccak, Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-MD5u, Raw-SHA1,
Raw-SHA1-AxCrypt, Raw-SHA1-Linkedin, Raw-SHA224, Raw-SHA256, Raw-SHA3,
Raw-SHA384, restic, ripemd-128, ripemd-160, rsvp, RVARY, Siemens-S7,
Salted-SHA1, SSHA512, sapb, sapg, saph, sappse, securezip, 7z, Signal, SIP,
skein-256, skein-512, skey, SL3, Snefru-128, Snefru-256, LastPass, SNMP,
solarwinds, SSH, sspr, STRIP, SunMD5, SybaseASE, Sybase-PROP, tacacs-plus,
tcp-md5, telegram, tezos, Tiger, tc_aes_xts, tc_ripemd160, tc_ripemd160boot,
tc_sha512, tc_whirlpool, vdi, OpenVMS, vmx, VNC, vtp, wbb3, whirlpool,
whirlpool0, whirlpool1, wpapsk, wpapsk-pmk, xmpp-scram, xsha, xsha512, zed,
ZIP, ZipMonster, plaintext, has-160, HMAC-MD5, HMAC-SHA1, HMAC-SHA224,
HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, dummy, crypt
414 formats (149 dynamic formats shown as just "dynamic_n" here)

┌──(saumyaa㉿kali)-[~]
└─$ 
```

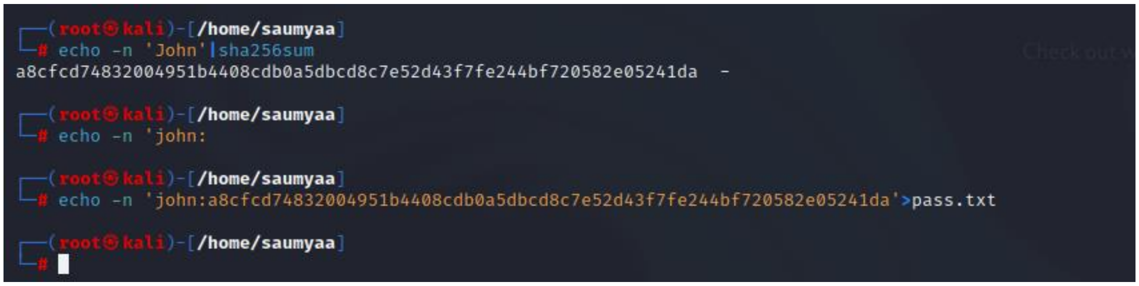
A. Single Crack Mode –

1. Add new username as John and create a password.



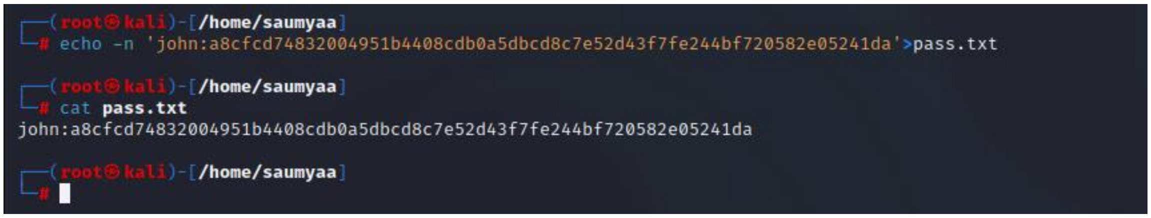
```
┌──(saumyaa㉿kali)-[~]
└─$ sudo su
┌──(root㉿kali)-[/home/saumyaa]
└─# useradd john
useradd: user 'john' already exists

┌──(root㉿kali)-[/home/saumyaa]
└─# passwd john
New password:
Retype new password:
passwd: password updated successfully

┌──(root㉿kali)-[/home/saumyaa]
└─# ▮
```

2. Finding the SHA-256-hashed password (of the password set above)



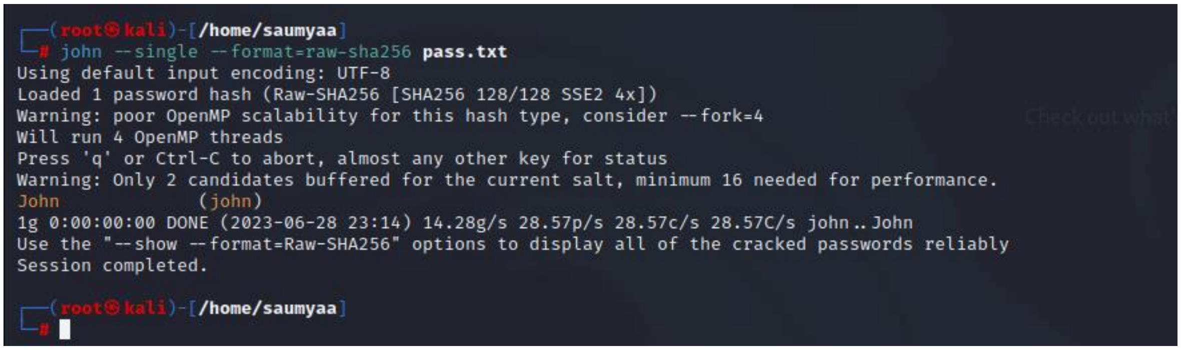
```
┌──(root㉿kali)-[/home/saumyaa]
└─# echo -n 'John'|sha256sum
a8cfcd74832004951b4408cdb0a5dbcd8c7e52d43f7fe244bf720582e05241da  -

┌──(root㉿kali)-[/home/saumyaa]
└─# echo -n 'john:

┌──(root㉿kali)-[/home/saumyaa]
└─# echo -n 'john:a8cfcd74832004951b4408cdb0a5dbcd8c7e52d43f7fe244bf720582e05241da'>pass.txt

┌──(root㉿kali)-[/home/saumyaa]
└─# ▮
```

3. Save the username and password hashvalue into a file pass.txt -



```
┌──(root㉿kali)-[/home/saumyaa]
└─# echo -n 'john:a8cfcd74832004951b4408cdb0a5dbcd8c7e52d43f7fe244bf720582e05241da'>pass.txt

┌──(root㉿kali)-[/home/saumyaa]
└─# cat pass.txt
john:a8cfcd74832004951b4408cdb0a5dbcd8c7e52d43f7fe244bf720582e05241da

┌──(root㉿kali)-[/home/saumyaa]
└─# ▮
```

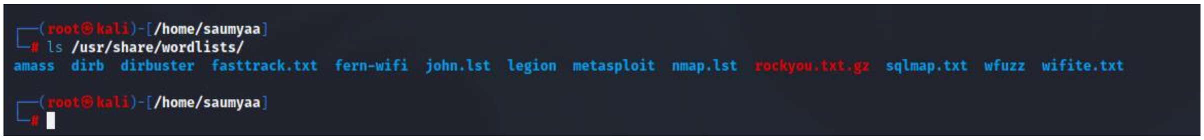
4. Run pass.txt through John the Ripper’s single crack Mode :



```
┌──(root㉿kali)-[/home/saumyaa]
└─# john --single --format=raw-sha256 pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 SSE2 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=4
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 2 candidates buffered for the current salt, minimum 16 needed for performance.
John             (john)
1g 0:00:00:00 DONE (2023-06-28 23:14) 14.28g/s 28.57p/s 28.57c/s 28.57C/s john..John
Use the "--show --format=Raw-SHA256" options to display all of the cracked passwords reliably
Session completed.

┌──(root㉿kali)-[/home/saumyaa]
└─# ▮
```

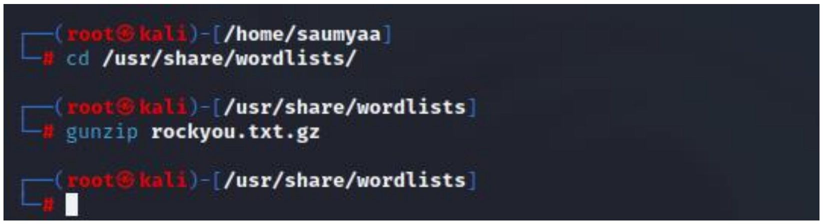
B. <u>Wordlist Mode –</u>

1. Locate the file rockyou.txt.gz :



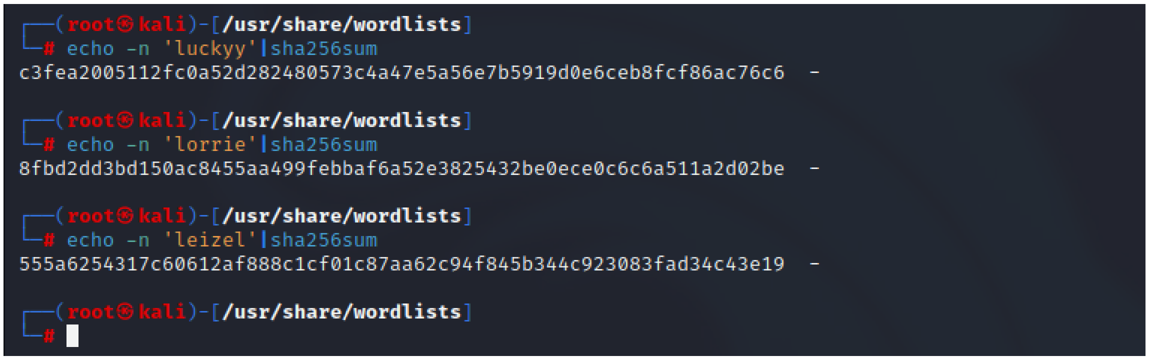
```
┌──(root㉿kali)-[/home/saumyaa]
└─# ls /usr/share/wordlists/
amass  dirb  dirbuster  fasttrack.txt  fern-wifi  john.lst  legion  metasploit  nmap.lst  rockyou.txt.gz  sqlmap.txt  wfuzz  wifite.txt

┌──(root㉿kali)-[/home/saumyaa]
└─# ▮
```

2. Navigate to directory containing file and exract it :



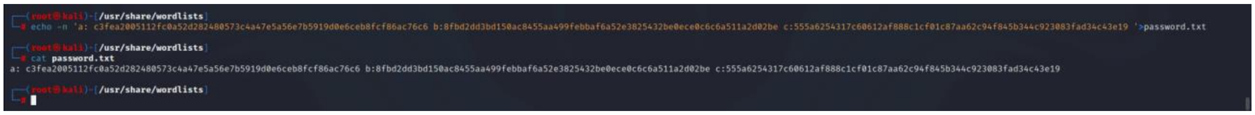
```
┌──(root㉿kali)-[/home/saumyaa]
└─# cd /usr/share/wordlists/

┌──(root㉿kali)-[/usr/share/wordlists]
└─# gunzip rockyou.txt.gz

┌──(root㉿kali)-[/usr/share/wordlists]
└─# 
```

3. Choosing 3 passwords :



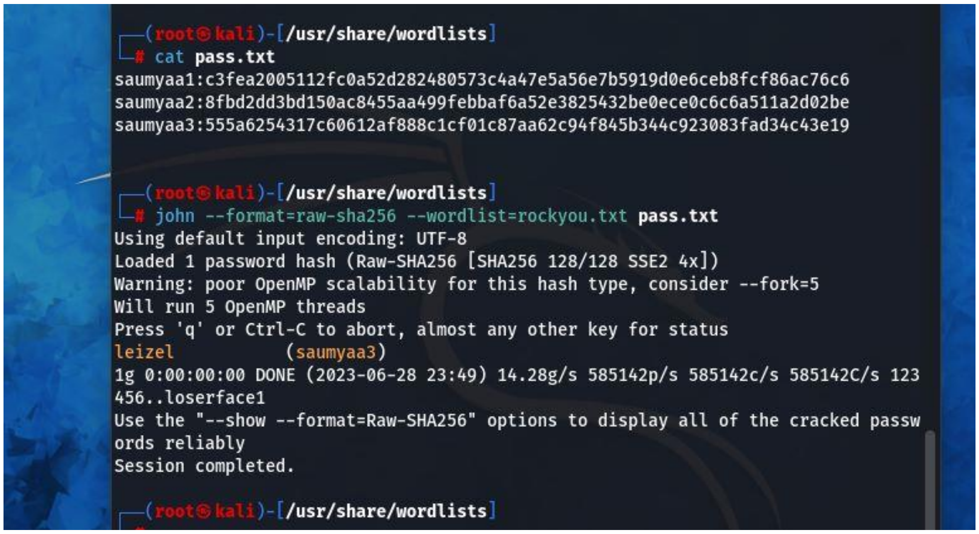
```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'luckyy'|sha256sum
c3fea2005112fc0a52d282480573c4a47e5a56e7b5919d0e6ceb8fcf86ac76c6  -

┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'lorrie'|sha256sum
8fbd2dd3bd150ac8455aa499febbaf6a52e3825432be0ece0c6c6a511a2d02be  -

┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'leizel'|sha256sum
555a6254317c60612af888c1cf01c87aa62c94f845b344c923083fad34c43e19  -

┌──(root㉿kali)-[/usr/share/wordlists]
└─# 
```

4. Save 3 usernames :



```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'a: c3fea2005112fc0a52d282480573c4a47e5a56e7b5919d0e6ceb8fcf86ac76c6 b:8fbd2dd3bd150ac8455aa499febbaf6a52e3825432be0ece0c6c6a511a2d02be c:555a6254317c60612af888c1cf01c87aa62c94f845b344c923083fad34c43e19 '>password.txt

┌──(root㉿kali)-[/usr/share/wordlists]
└─# cat password.txt
a: c3fea2005112fc0a52d282480573c4a47e5a56e7b5919d0e6ceb8fcf86ac76c6 b:8fbd2dd3bd150ac8455aa499febbaf6a52e3825432be0ece0c6c6a511a2d02be c:555a6254317c60612af888c1cf01c87aa62c94f845b344c923083fad34c43e19

┌──(root㉿kali)-[/usr/share/wordlists]
└─# 
```

5. Run John in wordlist mode:

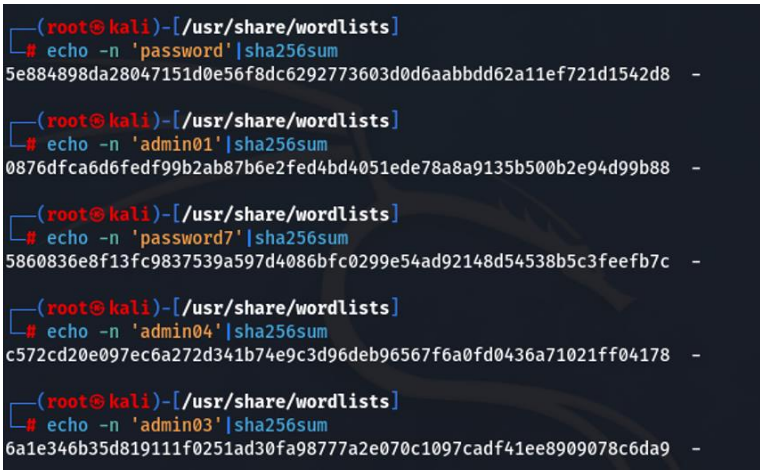


```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# cat pass.txt
saumyaa1:c3fea2005112fc0a52d282480573c4a47e5a56e7b5919d0e6ceb8fcf86ac76c6
saumyaa2:8fbd2dd3bd150ac8455aa499febbaf6a52e3825432be0ece0c6c6a511a2d02be
saumyaa3:555a6254317c60612af888c1cf01c87aa62c94f845b344c923083fad34c43e19


┌──(root㉿kali)-[/usr/share/wordlists]
└─# john --format=raw-sha256 --wordlist=rockyou.txt pass.txt
Using default input encoding: UTF-8
Loaded 1 password hash (Raw-SHA256 [SHA256 128/128 SSE2 4x])
Warning: poor OpenMP scalability for this hash type, consider --fork=5
Will run 5 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
leizel           (saumyaa3)
1g 0:00:00:00 DONE (2023-06-28 23:49) 14.28g/s 585142p/s 585142c/s 585142C/s 123
456..loserface1
Use the "--show --format=Raw-SHA256" options to display all of the cracked passw
ords reliably
Session completed.

┌──(root㉿kali)-[/usr/share/wordlists]
```

C. Incremental Mode –

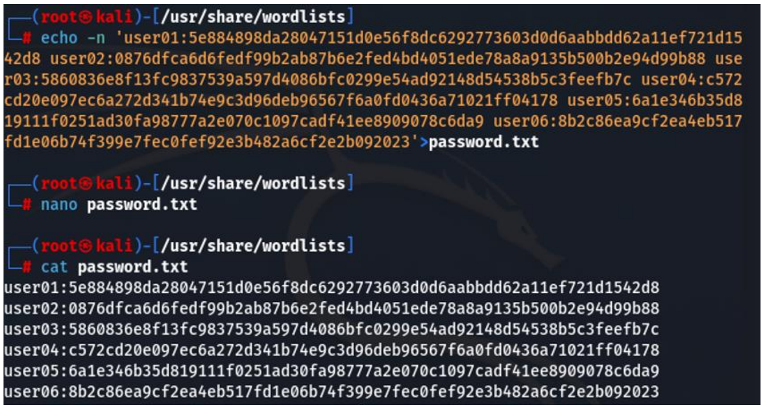
1. Choose passwords –
password,admin01,password7,admin04,admin03,password5

Hashing them –

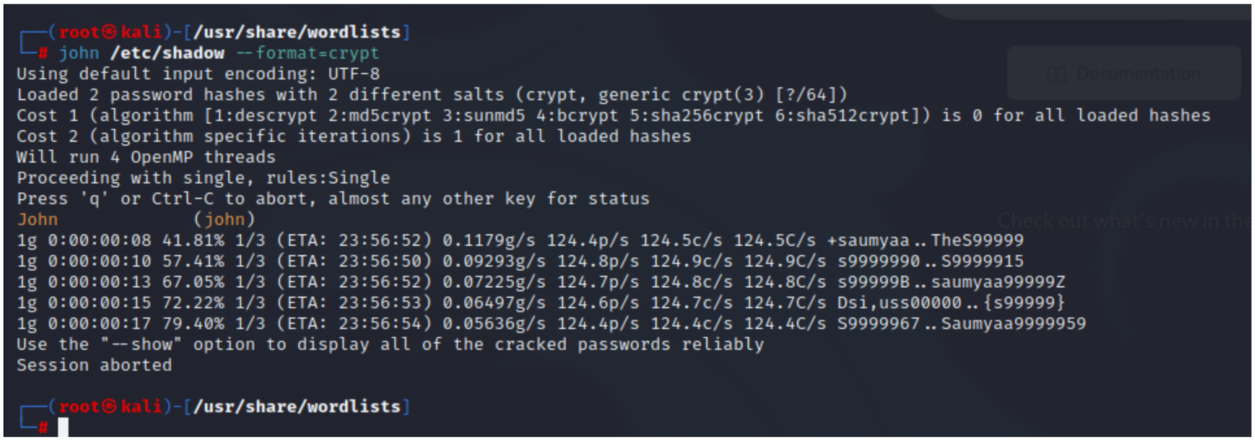


```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'password'|sha256sum
5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8  -

┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'admin01'|sha256sum
0876dfca6d6fedf99b2ab87b6e2fed4bd4051ede78a8a9135b500b2e94d99b88  -

┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'password7'|sha256sum
5860836e8f13fc9837539a597d4086bfc0299e54ad92148d54538b5c3feefb7c  -

┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'admin04'|sha256sum
c572cd20e097ec6a272d341b74e9c3d96deb96567f6a0fd0436a71021ff04178  -

┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'admin03'|sha256sum
6a1e346b35d819111f0251ad30fa98777a2e070c1097cadf41ee8909078c6da9  -
```

2. Saving usernames in the document –



```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# echo -n 'user01:5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d15
42d8 user02:0876dfca6d6fedf99b2ab87b6e2fed4bd4051ede78a8a9135b500b2e94d99b88 use
r03:5860836e8f13fc9837539a597d4086bfc0299e54ad92148d54538b5c3feefb7c user04:c572
cd20e097ec6a272d341b74e9c3d96deb96567f6a0fd0436a71021ff04178 user05:6a1e346b35d8
19111f0251ad30fa98777a2e070c1097cadf41ee8909078c6da9 user06:8b2c86ea9cf2ea4eb517
fd1e06b74f399e7fec0fef92e3b482a6cf2e2b092023'>password.txt

┌──(root㉿kali)-[/usr/share/wordlists]
└─# nano password.txt

┌──(root㉿kali)-[/usr/share/wordlists]
└─# cat password.txt
user01:5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8
user02:0876dfca6d6fedf99b2ab87b6e2fed4bd4051ede78a8a9135b500b2e94d99b88
user03:5860836e8f13fc9837539a597d4086bfc0299e54ad92148d54538b5c3feefb7c
user04:c572cd20e097ec6a272d341b74e9c3d96deb96567f6a0fd0436a71021ff04178
user05:6a1e346b35d819111f0251ad30fa98777a2e070c1097cadf41ee8909078c6da9
user06:8b2c86ea9cf2ea4eb517fd1e06b74f399e7fec0fef92e3b482a6cf2e2b092023
```



```
┌──(root㉿kali)-[/usr/share/wordlists]
└─# john /etc/shadow --format=crypt
Using default input encoding: UTF-8
Loaded 2 password hashes with 2 different salts (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
John             (john)
1g 0:00:00:08 41.81% 1/3 (ETA: 23:56:52) 0.1179g/s 124.4p/s 124.5c/s 124.5C/s +saumyaa..TheS99999
1g 0:00:00:10 57.41% 1/3 (ETA: 23:56:50) 0.09293g/s 124.8p/s 124.9c/s 124.9C/s s9999990..S9999915
1g 0:00:00:13 67.05% 1/3 (ETA: 23:56:52) 0.07225g/s 124.7p/s 124.8c/s 124.8C/s s99999B..saumyaa99999Z
1g 0:00:00:15 72.22% 1/3 (ETA: 23:56:53) 0.06497g/s 124.6p/s 124.7c/s 124.7C/s Dsi,uss00000..{s99999}
1g 0:00:00:17 79.40% 1/3 (ETA: 23:56:54) 0.05636g/s 124.4p/s 124.4c/s 124.4C/s S9999967..Saumyaa9999959
Use the "--show" option to display all of the cracked passwords reliably
Session aborted

┌──(root㉿kali)-[/usr/share/wordlists]
└─# 
```