University of Sheffield

Lab Report For COM6015- 2022

Group Number 5 :

Jagpreet Jakhar, Rohanshu Sharma

Department of Computer Science

May 17, 2023

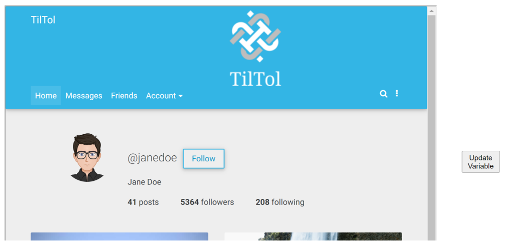
Contents

List of Figures

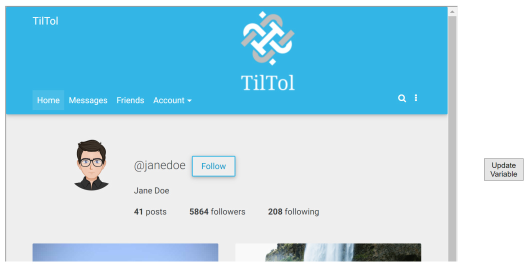
List of Tables

Chapter 1

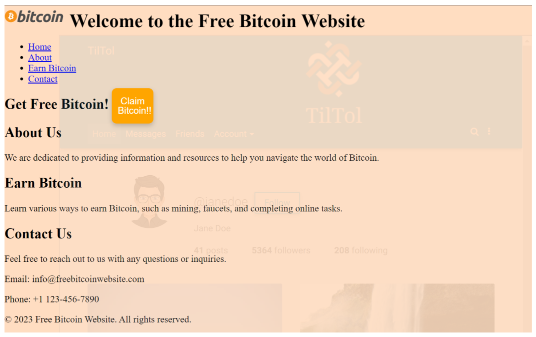
Question 1:Clickjacking



(a) *Before Update: followers:5364*



(b) *After update: followers:5864 (+500 on every press)*



Welcome to the Free Bitcoin Website

Get Free Bitcoin!

About Us

We are dedicated to providing information and resources to help you navigate the world of Bitcoin.

Earn Bitcoin

Learn various ways to earn Bitcoin, such as mining, faucets, and completing online tasks.

Contact Us

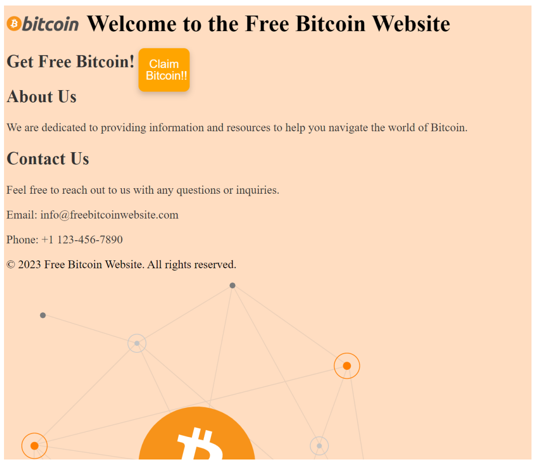
Feel free to reach out to us with any questions or inquiries.

Email: info@freebitcoinwebsite.com

Phone: +1 123-456-7890

© 2023 Free Bitcoin Website. All rights reserved.

(c) *Pre-Final*



Welcome to the Free Bitcoin Website

Get Free Bitcoin!

About Us

We are dedicated to providing information and resources to help you navigate the world of Bitcoin.

Contact Us

Feel free to reach out to us with any questions or inquiries.

Email: info@freebitcoinwebsite.com

Phone: +1 123-456-7890

© 2023 Free Bitcoin Website. All rights reserved.

(d) *Final Clickbait*

Figure 1.1: *Scam Chain*

The code files are attached as test.html for clickbait webpage, and joan.html for Joan Doe's tiltolpage.

```
<!DOCTYPE html>
/**
 * Retrieves and updates followers count of Jane Doe
 *
 * The access() function retrieves a reference to the iframe element
   containing Jane Doe's Webpage.
 *
 * The updateVariable() function accesses the document object of the iframe
   and then retrieves the element with the ID "followers_no", which is the
   number of followers  and
 * it updates its value by incrementing the current value by 500.
 * Clicking the button triggers the updateVariable() function.
 */
<script>
    function access() {
      var iframe = document.getElementById("victim_website");

      }
      function updateVariable() {
      var iframe = document.getElementById("jane");
      var iframeDocument = iframe.contentDocument || iframe.contentWindow.
   document;

      // Access and update the variable within the iframe
        var followers_no = iframeDocument.getElementById("followers_no");
        if (followers_no) {
          var currentValue = parseInt(followers_no.textContent);
          var newValue = currentValue + 500;
          followers_no.textContent = newValue;
        }
      }
    </script>
<iframe id="jane" src="joan.html" onload="access()" class="scam_frame"></
   iframe>
    <button onclick="updateVariable()" class="scam">Claim Bitcoin!!</button>

```

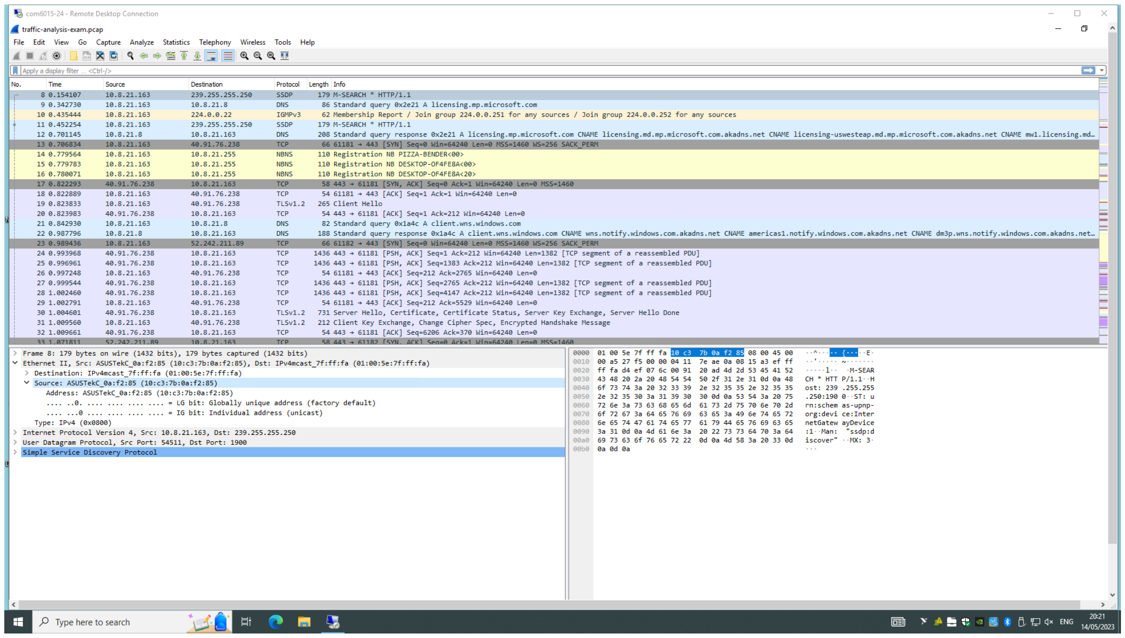
Code 1.1: Clickbait Code

The Clickbait puts the Joan Doe's page in an Iframe and then using JavaScript extracts the variable "followers_no" from the joan.html. Everytime the Claim Bitcoin button is clicked the number of followers increase by 500.

Chapter 2

Question 2: Wireshark

2.1 What is the victim MAC address?



```
No.  Time      Source         Destination      Protocol Length Info
  8 0.154107  10.8.21.163    239.255.255.250  SSDP      179 M-SEARCH * HTTP/1.1
  9 0.342730  10.8.21.163    10.8.21.8        DNS        86 Standard query 0x2e21 A licensing.mp.microsoft.com
 10 0.435444  10.8.21.163    224.0.0.22       IGMPv3     62 Membership Report / Join group 224.0.0.251 for any sources / Join group 224.0.0.252 for any sources
 11 0.452254  10.8.21.163    239.255.255.250  SSDP      179 M-SEARCH * HTTP/1.1
 12 0.701145  10.8.21.8      10.8.21.163      DNS       208 Standard query response 0x2e21 A licensing.mp.microsoft.com CNAME licensing.md.mp.microsoft.com.akadns.net CNAME licensing-uswesteap.md.mp.microsoft.com.akadns.net CNAME mw1.licensing.md…
 13 0.706834  10.8.21.163    40.91.76.238     TCP        66 61181 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
 14 0.779564  10.8.21.163    10.8.21.255      NBNS      110 Registration NB PIZZA-BENDER<00>
 15 0.779783  10.8.21.163    10.8.21.255      NBNS      110 Registration NB DESKTOP-OF4FE8A<00>
 16 0.780071  10.8.21.163    10.8.21.255      NBNS      110 Registration NB DESKTOP-OF4FE8A<20>
 17 0.822293  40.91.76.238   10.8.21.163      TCP        58 443 → 61181 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
 18 0.822889  10.8.21.163    40.91.76.238     TCP        54 61181 → 443 [ACK] Seq=1 Ack=1 Win=64240 Len=0
 19 0.823833  10.8.21.163    40.91.76.238     TLSv1.2   265 Client Hello
 20 0.823983  40.91.76.238   10.8.21.163      TCP        54 443 → 61181 [ACK] Seq=1 Ack=212 Win=64240 Len=0
 21 0.842930  10.8.21.163    10.8.21.8        DNS        82 Standard query 0x1a4c A client.wns.windows.com
 22 0.987796  10.8.21.8      10.8.21.163      DNS       188 Standard query response 0x1a4c A client.wns.windows.com CNAME wns.notify.windows.com.akadns.net CNAME americas1.notify.windows.com.akadns.net CNAME dm3p.wns.notify.windows.com.akadns.net…
 23 0.989436  10.8.21.163    52.242.211.89    TCP        66 61182 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
 24 0.993968  40.91.76.238   10.8.21.163      TCP      1436 443 → 61181 [PSH, ACK] Seq=1 Ack=212 Win=64240 Len=1382 [TCP segment of a reassembled PDU]
 25 0.996961  40.91.76.238   10.8.21.163      TCP      1436 443 → 61181 [PSH, ACK] Seq=1383 Ack=212 Win=64240 Len=1382 [TCP segment of a reassembled PDU]
 26 0.997248  10.8.21.163    40.91.76.238     TCP        54 61181 → 443 [ACK] Seq=212 Ack=2765 Win=64240 Len=0
 27 0.999544  40.91.76.238   10.8.21.163      TCP      1436 443 → 61181 [PSH, ACK] Seq=2765 Ack=212 Win=64240 Len=1382 [TCP segment of a reassembled PDU]
 28 1.002460  40.91.76.238   10.8.21.163      TCP      1436 443 → 61181 [PSH, ACK] Seq=4147 Ack=212 Win=64240 Len=1382 [TCP segment of a reassembled PDU]
 29 1.002791  10.8.21.163    40.91.76.238     TCP        54 61181 → 443 [ACK] Seq=212 Ack=5529 Win=64240 Len=0
 30 1.004601  40.91.76.238   10.8.21.163      TLSv1.2   731 Server Hello, Certificate, Certificate Status, Server Key Exchange, Server Hello Done
 31 1.009560  10.8.21.163    40.91.76.238     TLSv1.2   212 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
 32 1.009661  40.91.76.238   10.8.21.163      TCP        54 443 → 61181 [ACK] Seq=6206 Ack=370 Win=64240 Len=0
 33 1.071811  52.242.211.89  10.8.21.163      TCP        58 443 → 61182 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
```

```
> Frame 8: 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits)
v Ethernet II, Src: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  > Destination: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
  v Source: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85)
      Address: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85)
      .... ..0. .... .... .... .... = LG bit: Globally unique address (factory default)
      .... ...0 .... .... .... .... = IG bit: Individual address (unicast)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.8.21.163, Dst: 239.255.255.250
> User Datagram Protocol, Src Port: 54511, Dst Port: 1900
> Simple Service Discovery Protocol
```

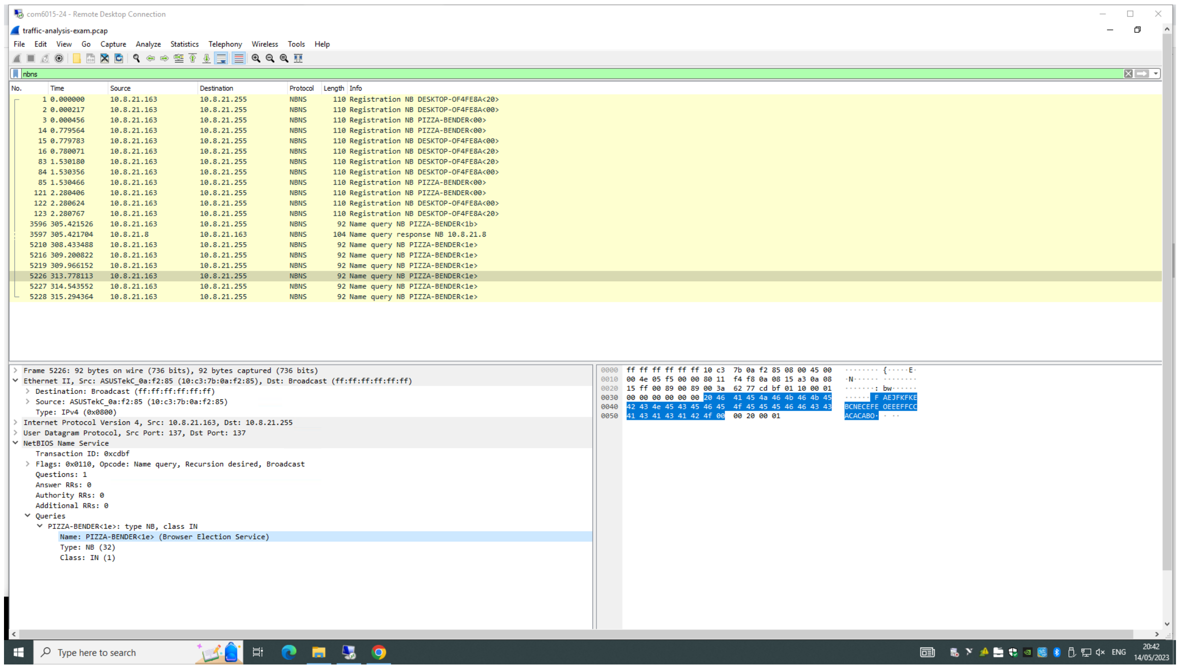
```
0000  01 00 5e 7f ff fa 10 c3  7b 0a f2 85 08 00 45 00   ··^····· {·····E·
0010  00 a5 27 f5 00 00 04 11  7e ae 0a 08 15 a3 ef ff   ··'····· ~·······
0020  ff fa d4 ef 07 6c 00 91  20 ad 4d 2d 53 45 41 52   ·····l··  ·M-SEAR
0030  43 48 20 2a 20 48 54 54  50 2f 31 2e 31 0d 0a 48   CH * HTT P/1.1··H
0040  6f 73 74 3a 20 32 33 39  2e 32 35 35 2e 32 35 35   ost: 239 .255.255
0050  2e 32 35 30 3a 31 39 30  30 0d 0a 53 54 3a 20 75   .250:190 0··ST: u
0060  72 6e 3a 73 63 68 65 6d  61 73 2d 75 70 6e 70 2d   rn:schem as-upnp-
0070  6f 72 67 3a 64 65 76 69  63 65 3a 49 6e 74 65 72   org:devi ce:Inter
0080  6e 65 74 47 61 74 65 77  61 79 44 65 76 69 63 65   netGatew ayDevice
0090  3a 31 0d 0a 4d 61 6e 3a  20 22 73 73 64 70 3a 64   :1··Man:  "ssdp:d
00a0  69 73 63 6f 76 65 72 22  0d 0a 4d 58 3a 20 33 0d   iscover" ··MX: 3·
00b0  0a 0d 0a                                           ···
```

com6015-24 - Remote Desktop Connection
traffic-analysis-exam.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Type here to search
ENG
20:21
14/05/2023

Figure 2.1: *Victim MAC Address*

As can be seen here the MAC address of the victim with the IP address - 10.8.21.163 is 10:C3:7B:0A:F2:85. This was found under the Source field of Ethernet.

2.2 What is the victim Host Name?


com6015-24 - Remote Desktop Connection
traffic-analysis-exam.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
nbns
Type here to search

```
No.  Time        Source       Destination  Protocol Length Info
   1 0.000000    10.8.21.163  10.8.21.255  NBNS     110 Registration NB DESKTOP-OF4FE8A<20>
   2 0.000217    10.8.21.163  10.8.21.255  NBNS     110 Registration NB DESKTOP-OF4FE8A<00>
   3 0.000456    10.8.21.163  10.8.21.255  NBNS     110 Registration NB PIZZA-BENDER<00>
  14 0.779564    10.8.21.163  10.8.21.255  NBNS     110 Registration NB PIZZA-BENDER<00>
  15 0.779783    10.8.21.163  10.8.21.255  NBNS     110 Registration NB DESKTOP-OF4FE8A<00>
  16 0.780071    10.8.21.163  10.8.21.255  NBNS     110 Registration NB DESKTOP-OF4FE8A<20>
  83 1.530180    10.8.21.163  10.8.21.255  NBNS     110 Registration NB DESKTOP-OF4FE8A<20>
  84 1.530356    10.8.21.163  10.8.21.255  NBNS     110 Registration NB DESKTOP-OF4FE8A<00>
  85 1.530466    10.8.21.163  10.8.21.255  NBNS     110 Registration NB PIZZA-BENDER<00>
 121 2.280406    10.8.21.163  10.8.21.255  NBNS     110 Registration NB PIZZA-BENDER<00>
 122 2.280624    10.8.21.163  10.8.21.255  NBNS     110 Registration NB DESKTOP-OF4FE8A<00>
 123 2.280767    10.8.21.163  10.8.21.255  NBNS     110 Registration NB DESKTOP-OF4FE8A<20>
3596 305.421526  10.8.21.163  10.8.21.255  NBNS      92 Name query NB PIZZA-BENDER<1b>
3597 305.421704  10.8.21.8    10.8.21.163  NBNS     104 Name query response NB 10.8.21.8
5210 308.433488  10.8.21.163  10.8.21.255  NBNS      92 Name query NB PIZZA-BENDER<1e>
5216 309.200822  10.8.21.163  10.8.21.255  NBNS      92 Name query NB PIZZA-BENDER<1e>
5219 309.966152  10.8.21.163  10.8.21.255  NBNS      92 Name query NB PIZZA-BENDER<1e>
5226 313.778113  10.8.21.163  10.8.21.255  NBNS      92 Name query NB PIZZA-BENDER<1e>
5227 314.543552  10.8.21.163  10.8.21.255  NBNS      92 Name query NB PIZZA-BENDER<1e>
5228 315.294364  10.8.21.163  10.8.21.255  NBNS      92 Name query NB PIZZA-BENDER<1e>
```

```
> Frame 5226: 92 bytes on wire (736 bits), 92 bytes captured (736 bits)
v Ethernet II, Src: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  > Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  > Source: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85)
    Type: IPv4 (0x0800)
> Internet Protocol Version 4, Src: 10.8.21.163, Dst: 10.8.21.255
> User Datagram Protocol, Src Port: 137, Dst Port: 137
v NetBIOS Name Service
    Transaction ID: 0xcdbf
  > Flags: 0x0110, Opcode: Name query, Recursion desired, Broadcast
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v PIZZA-BENDER<1e>: type NB, class IN
        Name: PIZZA-BENDER<1e> (Browser Election Service)
        Type: NB (32)
        Class: IN (1)
```

```
0000  ff ff ff ff ff ff 10 c3  7b 0a f2 85 08 00 45 00   ········ {·····E·
0010  00 4e 05 f5 00 00 80 11  f4 f8 0a 08 15 a3 0a 08   ·N······ ········
0020  15 ff 00 89 00 89 00 3a  62 77 cd bf 01 10 00 01   ·······: bw······
0030  00 00 00 00 00 00 20 46  41 45 4a 46 4b 46 4b 45   ······ F AEJFKFKE
0040  42 43 4e 45 43 45 46 45  4f 45 45 45 46 46 43 43   BCNECEFE OEEEFFCC
0050  41 43 41 43 41 42 4f 00  00 20 00 01               ACACABO· · ··
```

Figure 2.2: *Victim Host Name*

The host name is PIZZA-BENDER. This was found by checking the NetBIOS which is used by Microsoft Windows for its name resolution function.

2.3 What is the client operating system, and what day and time the malware was executed?

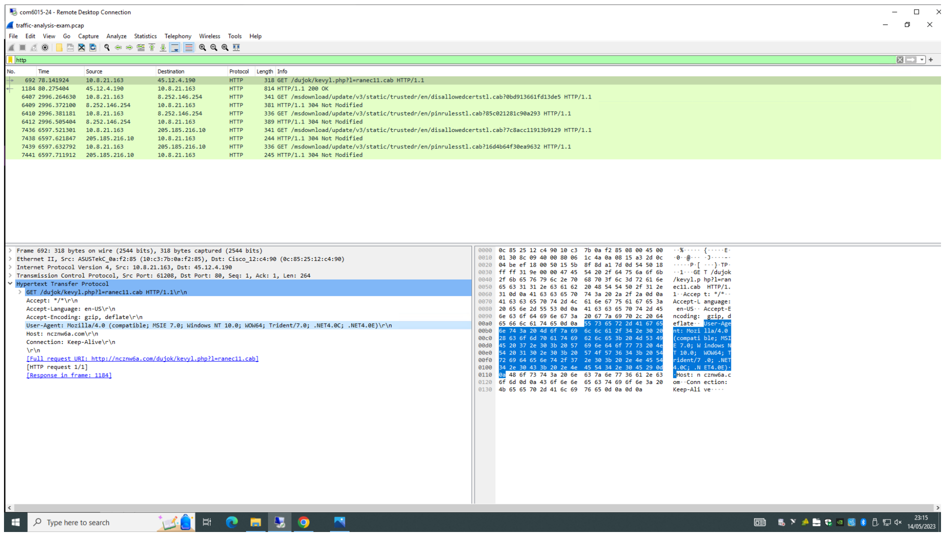

```
com6015-24 - Remote Desktop Connection
traffic-analysis-exam.pcap
http

No.  Time         Source          Destination     Protocol Length Info
 692 78.141924    10.8.21.163     45.12.4.190     HTTP     318 GET /dujok/kevyl.php?l=ranec11.cab HTTP/1.1
1184 80.275404    45.12.4.190     10.8.21.163     HTTP     814 HTTP/1.1 200 OK
6407 2996.264630  10.8.21.163     8.252.146.254   HTTP     341 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?0bd913661fd13de5 HTTP/1.1
6409 2996.372100  8.252.146.254   10.8.21.163     HTTP     381 HTTP/1.1 304 Not Modified
6410 2996.381181  10.8.21.163     8.252.146.254   HTTP     336 GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?85c021281c90a293 HTTP/1.1
6412 2996.505404  8.252.146.254   10.8.21.163     HTTP     389 HTTP/1.1 304 Not Modified
7436 6597.521301  10.8.21.163     205.185.216.10  HTTP     341 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?7c8acc11913b9129 HTTP/1.1
7438 6597.621847  205.185.216.10  10.8.21.163     HTTP     244 HTTP/1.1 304 Not Modified
7439 6597.632792  10.8.21.163     205.185.216.10  HTTP     336 GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?16d4b64f30ea9632 HTTP/1.1
7441 6597.711912  205.185.216.10  10.8.21.163     HTTP     245 HTTP/1.1 304 Not Modified

> Frame 692: 318 bytes on wire (2544 bits), 318 bytes captured (2544 bits)
> Ethernet II, Src: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85), Dst: Cisco_12:c4:90 (0c:85:25:12:c4:90)
> Internet Protocol Version 4, Src: 10.8.21.163, Dst: 45.12.4.190
> Transmission Control Protocol, Src Port: 61208, Dst Port: 80, Seq: 1, Ack: 1, Len: 264
v Hypertext Transfer Protocol
  > GET /dujok/kevyl.php?l=ranec11.cab HTTP/1.1\r\n
    Accept: */*\r\n
    Accept-Language: en-US\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E)\r\n
    Host: ncznw6a.com\r\n
    Connection: Keep-Alive\r\n
    \r\n
    [Full request URI: http://ncznw6a.com/dujok/kevyl.php?l=ranec11.cab]
    [HTTP request 1/1]
    [Response in frame: 1184]

0000  0c 85 25 12 c4 90 10 c3  7b 0a f2 85 08 00 45 00   ··%····· {·····E·
0010  01 30 8c 09 40 00 80 06  1c 4a 0a 08 15 a3 2d 0c   ·0··@··· ·J·····-·
0020  04 be ef 18 00 50 15 5b  8f 8d a1 7d 0d 54 50 18   ·····P·[ ···}·TP·
0030  ff ff 31 9e 00 00 47 45  54 20 2f 64 75 6a 6f 6b   ··1···GE T /dujok
0040  2f 6b 65 76 79 6c 2e 70  68 70 3f 6c 3d 72 61 6e   /kevyl.p hp?l=ran
0050  65 63 31 31 2e 63 61 62  20 48 54 54 50 2f 31 2e   ec11.cab  HTTP/1.
0060  31 0d 0a 41 63 63 65 70  74 3a 20 2a 2f 2a 0d 0a   1··Accep t: */*··
0070  41 63 63 65 70 74 2d 4c  61 6e 67 75 61 67 65 3a   Accept-L anguage:
0080  20 65 6e 2d 55 53 0d 0a  41 63 63 65 70 74 2d 45    en-US·· Accept-E
0090  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 2c 20 64   ncoding:  gzip, d
00a0  65 66 6c 61 74 65 0d 0a  55 73 65 72 2d 41 67 65   eflate·· User-Age
00b0  6e 74 3a 20 4d 6f 7a 69  6c 6c 61 2f 34 2e 30 20   nt: Mozi lla/4.0
00c0  28 63 6f 6d 70 61 74 69  62 6c 65 3b 20 4d 53 49   (compati ble; MSI
00d0  45 20 37 2e 30 3b 20 57  69 6e 64 6f 77 73 20 4e   E 7.0; W indows N
00e0  54 20 31 30 2e 30 3b 20  57 4f 57 36 34 3b 20 54   T 10.0;  WOW64; T
00f0  72 69 64 65 6e 74 2f 37  2e 30 3b 20 2e 4e 45 54   rident/7 .0; .NET
0100  34 2e 30 43 3b 20 2e 4e  45 54 34 2e 30 45 29 0d   4.0C; .N ET4.0E)·
0110  0a 48 6f 73 74 3a 20 6e  63 7a 6e 77 36 61 2e 63   ·Host: n cznw6a.c
0120  6f 6d 0d 0a 43 6f 6e 6e  65 63 74 69 6f 6e 3a 20   om··Conn ection:
0130  4b 65 65 70 2d 41 6c 69  76 65 0d 0a 0d 0a         Keep-Ali ve····
```

Figure 2.3: *Victim Client OS*

Here HTTP filter was used and as can be seen that there is only one request that gave code 200(success). So the malware might have been downloaded from here. Now as can be seen the HTTP field in the User-Agent that the client operating system is Windows NT 10.0.

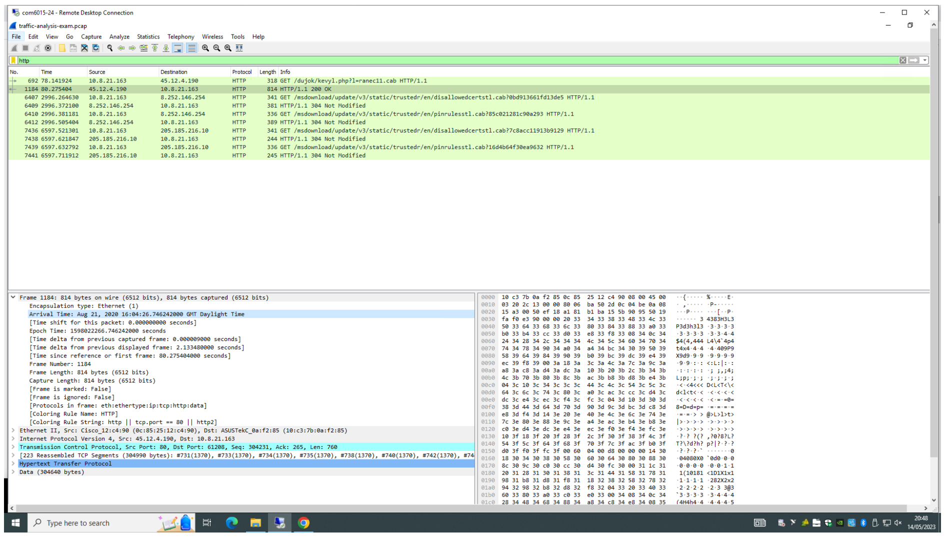
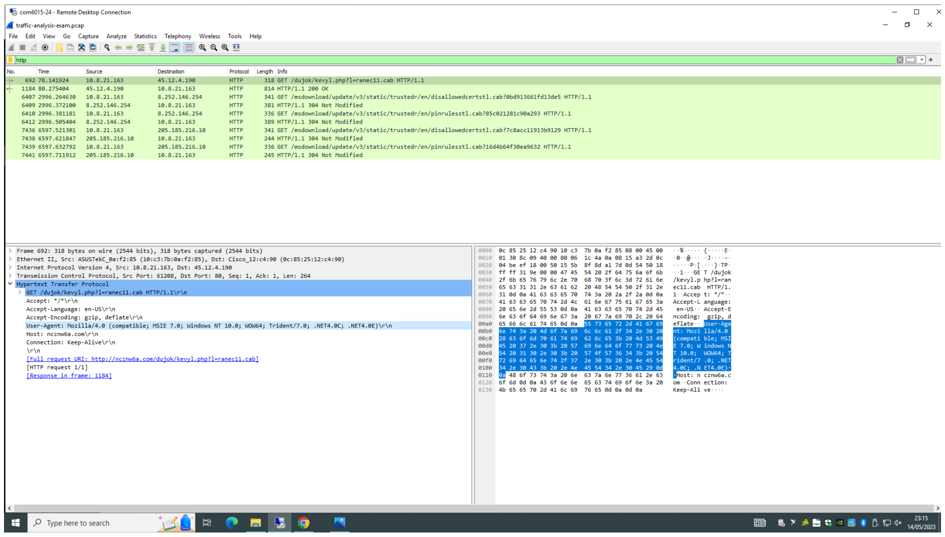

```
com6015-24 - Remote Desktop Connection
traffic-analysis-exam.pcap
http

No.  Time         Source          Destination     Protocol Length Info
 692 78.141924    10.8.21.163     45.12.4.190     HTTP     318 GET /dujok/kevyl.php?l=ranec11.cab HTTP/1.1
1184 80.275404    45.12.4.190     10.8.21.163     HTTP     814 HTTP/1.1 200 OK
6407 2996.264630  10.8.21.163     8.252.146.254   HTTP     341 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?0bd913661fd13de5 HTTP/1.1
6409 2996.372100  8.252.146.254   10.8.21.163     HTTP     381 HTTP/1.1 304 Not Modified
6410 2996.381181  10.8.21.163     8.252.146.254   HTTP     336 GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?85c021281c90a293 HTTP/1.1
6412 2996.505404  8.252.146.254   10.8.21.163     HTTP     389 HTTP/1.1 304 Not Modified
7436 6597.521301  10.8.21.163     205.185.216.10  HTTP     341 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?7c8acc11913b9129 HTTP/1.1
7438 6597.621847  205.185.216.10  10.8.21.163     HTTP     244 HTTP/1.1 304 Not Modified
7439 6597.632792  10.8.21.163     205.185.216.10  HTTP     336 GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?16d4b64f30ea9632 HTTP/1.1
7441 6597.711912  205.185.216.10  10.8.21.163     HTTP     245 HTTP/1.1 304 Not Modified

v Frame 1184: 814 bytes on wire (6512 bits), 814 bytes captured (6512 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Aug 21, 2020 16:04:26.746242000 GMT Daylight Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1598022266.746242000 seconds
    [Time delta from previous captured frame: 0.000009000 seconds]
    [Time delta from previous displayed frame: 2.133480000 seconds]
    [Time since reference or first frame: 80.275404000 seconds]
    Frame Number: 1184
    Frame Length: 814 bytes (6512 bits)
    Capture Length: 814 bytes (6512 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http:data]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
> Ethernet II, Src: Cisco_12:c4:90 (0c:85:25:12:c4:90), Dst: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85)
> Internet Protocol Version 4, Src: 45.12.4.190, Dst: 10.8.21.163
> Transmission Control Protocol, Src Port: 80, Dst Port: 61208, Seq: 304231, Ack: 265, Len: 760
> [223 Reassembled TCP Segments (304990 bytes): #731(1370), #733(1370), #734(1370), #735(1370), #738(1370), #740(1370), #742(1370), #74
> Hypertext Transfer Protocol
> Data (304640 bytes)

0000  10 c3 7b 0a f2 85 0c 85  25 12 c4 90 08 00 45 00   ··{····· %·····E·
0010  03 20 2c 13 00 00 80 06  ba 50 2d 0c 04 be 0a 08   · ,····· ·P-·····
0020  15 a3 00 50 ef 18 a1 81  b1 ba 15 5b 90 95 50 19   ···P···· ···[··P·
0030  fa f0 e3 90 00 00 20 33  34 33 38 33 48 33 4c 33   ······ 3 4383H3L3
0040  50 33 64 33 68 33 6c 33  80 33 84 33 88 33 a0 33   P3d3h3l3 ·3·3·3·3
0050  b0 33 b4 33 cc 33 d0 33  e8 33 f8 33 08 34 0c 34   ·3·3·3·3 ·3·3·4·4
0060  24 34 28 34 2c 34 34 34  4c 34 5c 34 60 34 70 34   $4(4,444 L4\4`4p4
0070  74 34 78 34 90 34 a0 34  a4 34 bc 34 30 39 50 39   t4x4·4·4 ·4·409P9
0080  58 39 64 39 84 39 90 39  b0 39 bc 39 dc 39 e4 39   X9d9·9·9 ·9·9·9·9
0090  ec 39 f8 39 00 3a 18 3a  3c 3a 4c 3a 7c 3a 9c 3a   ·9·9·:·: <:L:|:·:
00a0  a8 3a c8 3a d4 3a dc 3a  10 3b 20 3b 2c 3b 34 3b   ·:·:·:·: ·; ;,;4;
00b0  4c 3b 70 3b 80 3b 8c 3b  ac 3b b8 3b d8 3b e4 3b   L;p;·;·; ·;·;·;·;
00c0  04 3c 10 3c 34 3c 3c 3c  44 3c 4c 3c 54 3c 5c 3c   ·<·<4<<< D<L<T<\<
00d0  64 3c 6c 3c 74 3c 80 3c  a0 3c ac 3c cc 3c d4 3c   d<l<t<·< ·<·<·<·<
00e0  dc 3c e4 3c ec 3c f4 3c  fc 3c 04 3d 10 3d 30 3d   ·<·<·<·< ·<·=·=0=
00f0  38 3d 44 3d 64 3d 70 3d  90 3d 9c 3d bc 3d c8 3d   8=D=d=p= ·=·=·=·=
0100  e8 3d f4 3d 14 3e 20 3e  40 3e 4c 3e 6c 3e 74 3e   ·=·=·> > @>L>l>t>
0110  7c 3e 80 3e 88 3e 9c 3e  a4 3e ac 3e b4 3e b8 3e   |>·>·>·> ·>·>·>·>
0120  c0 3e d4 3e dc 3e e4 3e  ec 3e f0 3e f4 3e fc 3e   ·>·>·>·> ·>·>·>·>
0130  10 3f 18 3f 20 3f 28 3f  2c 3f 30 3f 38 3f 4c 3f   ·?·? ?(? ,?0?8?L?
0140  54 3f 5c 3f 64 3f 68 3f  70 3f 7c 3f ac 3f b0 3f   T?\?d?h? p?|?·?·?
0150  d0 3f f0 3f fc 3f 00 60  04 00 d8 00 00 00 14 30   ·?·?·?·` ·······0
0160  18 30 34 30 38 30 58 30  60 30 64 30 80 30 88 30   ·04080X0 `0d0·0·0
0170  8c 30 9c 30 c0 30 cc 30  d4 30 fc 30 00 31 1c 31   ·0·0·0·0 ·0·0·1·1
0180  20 31 28 31 30 31 38 31  3c 31 44 31 58 31 78 31    1(10181 <1D1X1x1
0190  98 31 b8 31 d8 31 f8 31  18 32 38 32 58 32 78 32   ·1·1·1·1 ·282X2x2
01a0  94 32 98 32 b8 32 d8 32  f8 32 04 33 20 33 40 33   ·2·2·2·2 ·2·3 3@3
01b0  60 33 80 33 a0 33 c0 33  e0 33 00 34 08 34 0c 34   `3·3·3·3 ·3·4·4·4
01c0  28 34 48 34 68 34 88 34  a8 34 c8 34 e8 34 08 35   (4H4h4·4 ·4·4·4·5
```

Figure 2.4: *Date and Time of attack*

Also the day and time on which the attack was executed, which can be seen in the particulars of the frame, is August 21, 2020 at 16:04.

2.4 Where was the intruder file location?

As can seen in the full request URI field the HTTP request is as follows : http://ncznw6a.com/dujok/kevyl.php?1=ranec11.cab .



```
No.  Time         Source          Destination     Protocol Length Info
 692 78.141924    10.8.21.163     45.12.4.190     HTTP     318 GET /dujok/kevyl.php?l=ranec11.cab HTTP/1.1
1184 80.275404    45.12.4.190     10.8.21.163     HTTP     814 HTTP/1.1 200 OK
6407 2996.264630  10.8.21.163     8.252.146.254   HTTP     341 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?0bd913661fd13de5 HTTP/1.1
6409 2996.372100  8.252.146.254   10.8.21.163     HTTP     381 HTTP/1.1 304 Not Modified
6410 2996.381181  10.8.21.163     8.252.146.254   HTTP     336 GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?85c021281c90a293 HTTP/1.1
6412 2996.505404  8.252.146.254   10.8.21.163     HTTP     389 HTTP/1.1 304 Not Modified
7436 6597.521301  10.8.21.163     205.185.216.10  HTTP     341 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?7c8acc11913b9129 HTTP/1.1
7438 6597.621847  205.185.216.10  10.8.21.163     HTTP     244 HTTP/1.1 304 Not Modified
7439 6597.632792  10.8.21.163     205.185.216.10  HTTP     336 GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?16d4b64f30ea9632 HTTP/1.1
7441 6597.711912  205.185.216.10  10.8.21.163     HTTP     245 HTTP/1.1 304 Not Modified
```

```
> Frame 692: 318 bytes on wire (2544 bits), 318 bytes captured (2544 bits)
> Ethernet II, Src: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85), Dst: Cisco_12:c4:90 (0c:85:25:12:c4:90)
> Internet Protocol Version 4, Src: 10.8.21.163, Dst: 45.12.4.190
> Transmission Control Protocol, Src Port: 61208, Dst Port: 80, Seq: 1, Ack: 1, Len: 264
v Hypertext Transfer Protocol
  > GET /dujok/kevyl.php?l=ranec11.cab HTTP/1.1\r\n
    Accept: */*\r\n
    Accept-Language: en-US\r\n
    Accept-Encoding: gzip, deflate\r\n
    User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; WOW64; Trident/7.0; .NET4.0C; .NET4.0E)\r\n
    Host: ncznw6a.com\r\n
    Connection: Keep-Alive\r\n
    \r\n
    [Full request URI: http://ncznw6a.com/dujok/kevyl.php?l=ranec11.cab]
    [HTTP request 1/1]
    [Response in frame: 1184]
```

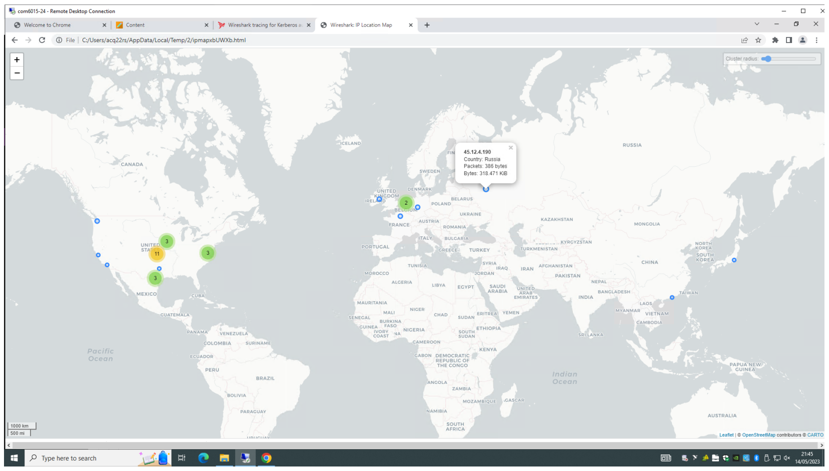
```
0000  0c 85 25 12 c4 90 10 c3  7b 0a f2 85 08 00 45 00   ··%····· {·····E·
0010  01 30 8c 09 40 00 80 06  1c 4a 0a 08 15 a3 2d 0c   ·0··@··· ·J····-·
0020  04 be ef 18 00 50 15 5b  8f 8d a1 7d 0d 54 50 18   ·····P·[ ···}·TP·
0030  ff ff 31 9e 00 00 47 45  54 20 2f 64 75 6a 6f 6b   ··1···GE T /dujok
0040  2f 6b 65 76 79 6c 2e 70  68 70 3f 6c 3d 72 61 6e   /kevyl.p hp?l=ran
0050  65 63 31 31 2e 63 61 62  20 48 54 54 50 2f 31 2e   ec11.cab  HTTP/1.
0060  31 0d 0a 41 63 63 65 70  74 3a 20 2a 2f 2a 0d 0a   1··Accep t: */*··
0070  41 63 63 65 70 74 2d 4c  61 6e 67 75 61 67 65 3a   Accept-L anguage:
0080  20 65 6e 2d 55 53 0d 0a  41 63 63 65 70 74 2d 45    en-US·· Accept-E
0090  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 2c 20 64   ncoding:  gzip, d
00a0  65 66 6c 61 74 65 0d 0a  55 73 65 72 2d 41 67 65   eflate·· User-Age
00b0  6e 74 3a 20 4d 6f 7a 69  6c 6c 61 2f 34 2e 30 20   nt: Mozi lla/4.0 
00c0  28 63 6f 6d 70 61 74 69  62 6c 65 3b 20 4d 53 49   (compati ble; MSI
00d0  45 20 37 2e 30 3b 20 57  69 6e 64 6f 77 73 20 4e   E 7.0; W indows N
00e0  54 20 31 30 2e 30 3b 20  57 4f 57 36 34 3b 20 54   T 10.0;  WOW64; T
00f0  72 69 64 65 6e 74 2f 37  2e 30 3b 20 2e 4e 45 54   rident/7 .0; .NET
0100  34 2e 30 43 3b 20 2e 4e  45 54 34 2e 30 45 29 0d   4.0C; .N ET4.0E)·
0110  0a 48 6f 73 74 3a 20 6e  63 7a 6e 77 36 61 2e 63   ·Host: n cznw6a.c
0120  6f 6d 0d 0a 43 6f 6e 6e  65 63 74 69 6f 6e 3a 20   om··Conn ection: 
0130  4b 65 65 70 2d 41 6c 69  76 65 0d 0a 0d 0a         Keep-Ali ve····
```

com6015-24 - Remote Desktop Connection
traffic-analysis-exam.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
http
Type here to search
23:15
14/05/2023

Figure 2.5: *File Location*

2.5 What is the intruder IP address? Show the geographical location of the intruder using the Wireshark endpoint feature.

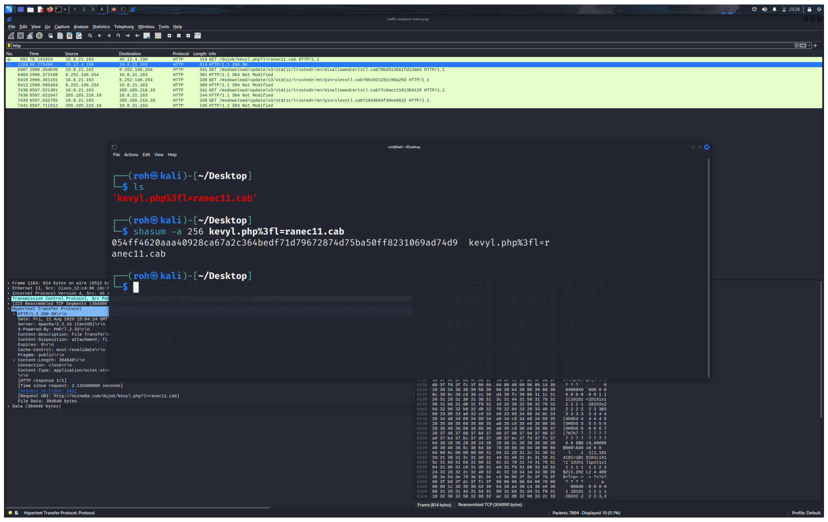
As we can see in the above images, the IP address is 45.12.4.190. Now using the Wireshark endpoint feature we got the geographical location as:



45.12.4.190
Country: Russia
Packets: 386 bytes
Bytes: 318.471 KiB

Figure 2.6: *IP Address*

2.6 Extract the malicious file (in your VM) and extract its hash key using Linux terminal/ windows cmd (SHA 256).



```
No.  Time         Source          Destination     Protocol Length Info
 692 78.141924    10.8.21.163     45.12.4.190     HTTP     318 GET /dujok/kevyl.php?l=ranec11.cab HTTP/1.1
1184 80.275404    45.12.4.190     10.8.21.163     HTTP     814 HTTP/1.1 200 OK
6407 2996.264630  10.8.21.163     8.252.146.254   HTTP     341 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?0bd913661fd13de5 HTTP/1.1
6409 2996.372100  8.252.146.254   10.8.21.163     HTTP     381 HTTP/1.1 304 Not Modified
6410 2996.381181  10.8.21.163     8.252.146.254   HTTP     336 GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?85c021281c90a293 HTTP/1.1
6412 2996.505404  8.252.146.254   10.8.21.163     HTTP     389 HTTP/1.1 304 Not Modified
7436 6597.521301  10.8.21.163     205.185.216.10  HTTP     341 GET /msdownload/update/v3/static/trustedr/en/disallowedcertstl.cab?7c8acc11913b9129 HTTP/1.1
7438 6597.621847  205.185.216.10  10.8.21.163     HTTP     244 HTTP/1.1 304 Not Modified
7439 6597.632792  10.8.21.163     205.185.216.10  HTTP     336 GET /msdownload/update/v3/static/trustedr/en/pinrulesstl.cab?16d4b64f30ea9632 HTTP/1.1
7441 6597.711912  205.185.216.10  10.8.21.163     HTTP     245 HTTP/1.1 304 Not Modified
```

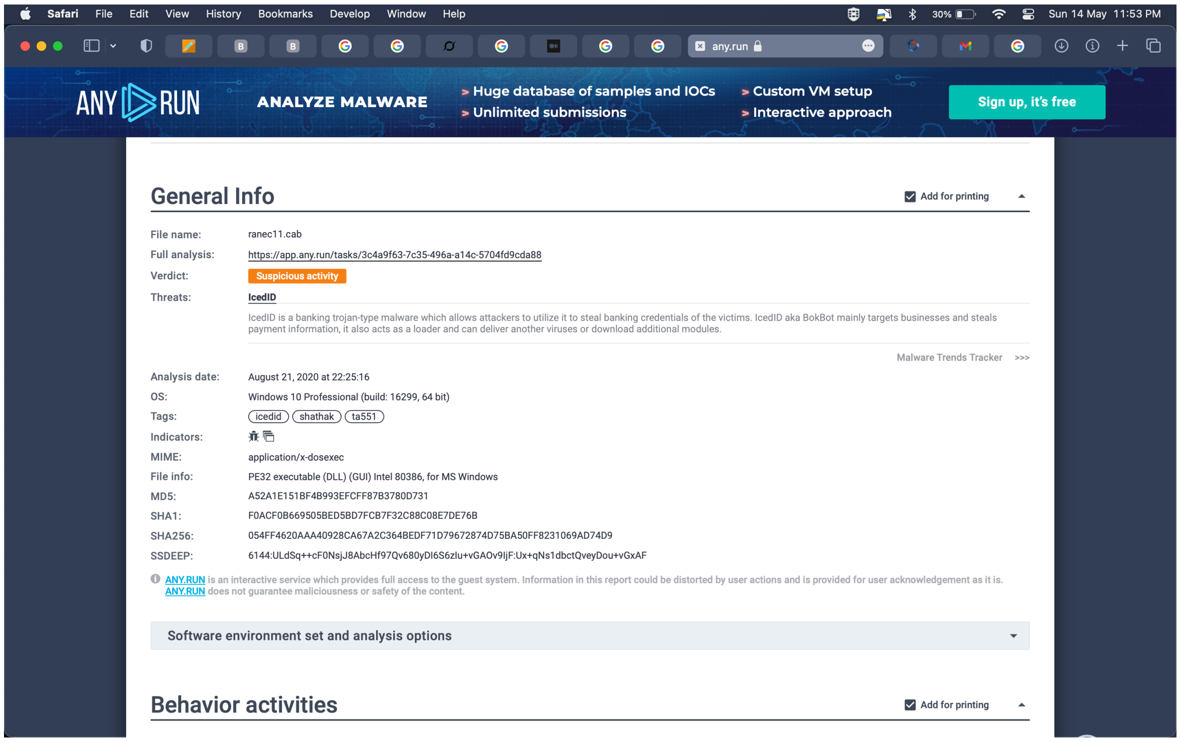
```
┌──(roh㉿kali)-[~/Desktop]
└─$ ls
'kevyl.php%3fl=ranec11.cab'

┌──(roh㉿kali)-[~/Desktop]
└─$ shasum -a 256 kevyl.php%3fl=ranec11.cab
054ff4620aaa40928ca67a2c364bedf71d79672874d75ba50ff8231069ad74d9  kevyl.php%3fl=r
anec11.cab

┌──(roh㉿kali)-[~/Desktop]
└─$ 
```

Figure 2.7: *Malicious File*

The pcap was opened in Kali to see the hash as the as the vm system did not have rights to download malicious files. The file was exported from Wireshark and the hash(SHA 256) as seen was extracted.

2.7 Use the extracted SHA 256 code, search on the Internet and provide the malware description.

We can see that when the hash was entered on the internet the following details were found. The malware's name is IcedID and it is a trojan that is designed to steal banking credentials of the victims. It steals payment information and can also deliver another viruses. This explains the unauthorised withdrawal of $5000.



General Info

Add for printing

File name:	ranec11.cab
Full analysis:	https://app.any.run/tasks/3c4a9f63-7c35-496a-a14c-5704fd9cda88
Verdict:	Suspicious activity
Threats:	IcedID
	IcedID is a banking trojan-type malware which allows attackers to utilize it to steal banking credentials of the victims. IcedID aka BokBot mainly targets businesses and steals payment information, it also acts as a loader and can deliver another viruses or download additional modules.
	Malware Trends Tracker >>>
Analysis date:	August 21, 2020 at 22:25:16
OS:	Windows 10 Professional (build: 16299, 64 bit)
Tags:	icedid shathak ta551
Indicators:	
MIME:	application/x-dosexec
File info:	PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
MD5:	A52A1E151BF4B993EFCFF87B3780D731
SHA1:	F0ACF0B669505BED5BD7FCB7F32C88C08E7DE76B
SHA256:	054FF4620AAA40928CA67A2C364BEDF71D79672874D75BA50FF8231069AD74D9
SSDEEP:	6144:ULdSq++cF0NsjJ8AbcHf97Qv680yDI6S6zlu+vGAOv9IjF:Ux+qNs1dbctQveyDou+vGxAF

ANY.RUN is an interactive service which provides full access to the guest system. Information in this report could be distorted by user actions and is provided for user acknowledgement as it is. ANY.RUN does not guarantee maliciousness or safety of the content.

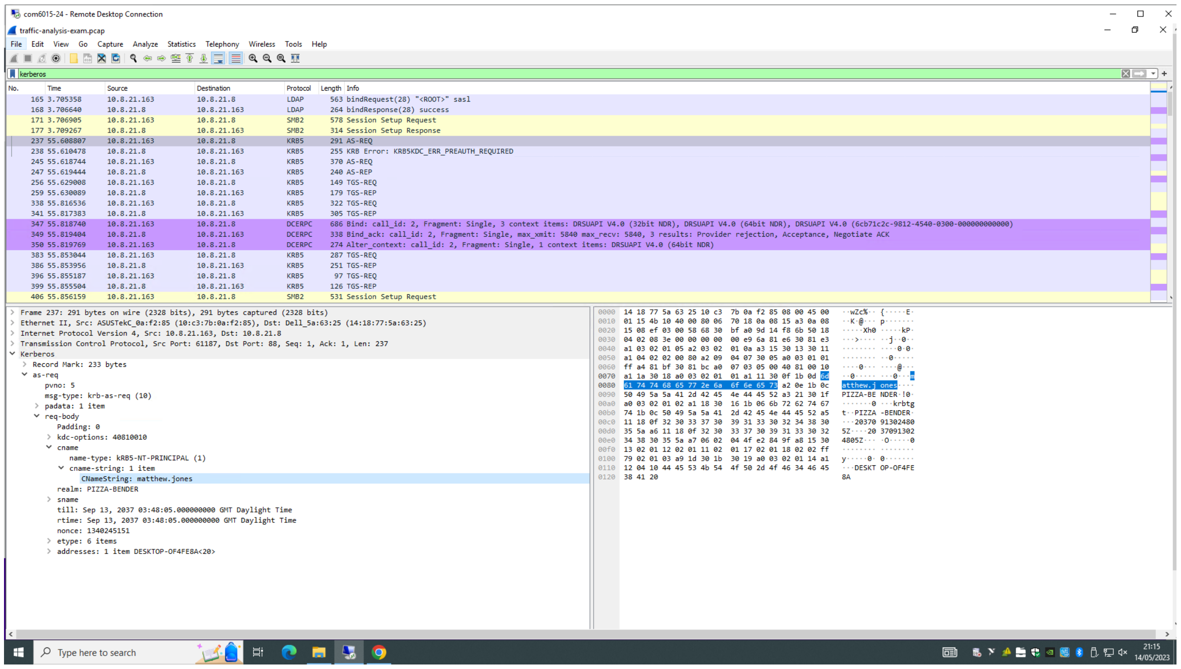
Software environment set and analysis options

Behavior activities

Add for printing

Figure 2.8: *Malicious File*

2.8 Which user is accountable for downloading the malicious malware from the Internet? provide Account Name by searching inside Kerberos packets (hint: find the answer in the KRBS packet (237) 'CNameString' section)



com6015-24 - Remote Desktop Connection

traffic-analysis-exam.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

kerberos

```
No.  Time       Source       Destination  Protocol Length Info
 165 3.705358   10.8.21.163  10.8.21.8    LDAP      563 bindRequest(28) "<ROOT>" sasl
 168 3.706640   10.8.21.8    10.8.21.163  LDAP      264 bindResponse(28) success
 171 3.706905   10.8.21.163  10.8.21.8    SMB2      578 Session Setup Request
 177 3.709267   10.8.21.8    10.8.21.163  SMB2      314 Session Setup Response
 237 55.608807  10.8.21.163  10.8.21.8    KRB5      291 AS-REQ
 238 55.610478  10.8.21.8    10.8.21.163  KRB5      255 KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
 245 55.618744  10.8.21.163  10.8.21.8    KRB5      370 AS-REQ
 247 55.619444  10.8.21.8    10.8.21.163  KRB5      240 AS-REP
 256 55.629008  10.8.21.163  10.8.21.8    KRB5      149 TGS-REQ
 259 55.630089  10.8.21.8    10.8.21.163  KRB5      179 TGS-REP
 338 55.816536  10.8.21.163  10.8.21.8    KRB5      322 TGS-REQ
 341 55.817383  10.8.21.8    10.8.21.163  KRB5      305 TGS-REP
 347 55.818740  10.8.21.163  10.8.21.8    DCERPC    686 Bind: call_id: 2, Fragment: Single, 3 context items: DRSUAPI V4.0 (32bit NDR), DRSUAPI V4.0 (64bit NDR), DRSUAPI V4.0 (6cb71c2c-9812-4540-0300-000000000000)
 349 55.819404  10.8.21.8    10.8.21.163  DCERPC    338 Bind_ack: call_id: 2, Fragment: Single, max_xmit: 5840 max_recv: 5840, 3 results: Provider rejection, Acceptance, Negotiate ACK
 350 55.819769  10.8.21.163  10.8.21.8    DCERPC    274 Alter_context: call_id: 2, Fragment: Single, 1 context items: DRSUAPI V4.0 (64bit NDR)
 383 55.853044  10.8.21.163  10.8.21.8    KRB5      287 TGS-REQ
 386 55.853956  10.8.21.8    10.8.21.163  KRB5      251 TGS-REP
 396 55.855187  10.8.21.163  10.8.21.8    KRB5       97 TGS-REQ
 399 55.855504  10.8.21.8    10.8.21.163  KRB5      126 TGS-REP
 406 55.856159  10.8.21.163  10.8.21.8    SMB2      531 Session Setup Request
```

```
> Frame 237: 291 bytes on wire (2328 bits), 291 bytes captured (2328 bits)
> Ethernet II, Src: ASUSTekC_0a:f2:85 (10:c3:7b:0a:f2:85), Dst: Dell_5a:63:25 (14:18:77:5a:63:25)
> Internet Protocol Version 4, Src: 10.8.21.163, Dst: 10.8.21.8
> Transmission Control Protocol, Src Port: 61187, Dst Port: 88, Seq: 1, Ack: 1, Len: 237
v Kerberos
  > Record Mark: 233 bytes
  v as-req
      pvno: 5
      msg-type: krb-as-req (10)
    > padata: 1 item
    v req-body
        Padding: 0
      > kdc-options: 40810010
      v cname
          name-type: kRB5-NT-PRINCIPAL (1)
        v cname-string: 1 item
            CNameString: matthew.jones
        realm: PIZZA-BENDER
      > sname
        till: Sep 13, 2037 03:48:05.000000000 GMT Daylight Time
        rtime: Sep 13, 2037 03:48:05.000000000 GMT Daylight Time
        nonce: 1340245151
      > etype: 6 items
      > addresses: 1 item DESKTOP-OF4FE8A<20>
```

```
0000  14 18 77 5a 63 25 10 c3  7b 0a f2 85 08 00 45 00   ··wZc%·· {·····E·
0010  01 15 4b 10 40 00 80 06  70 18 0a 08 15 a3 0a 08   ··K·@··· p·······
0020  15 08 ef 03 00 58 68 30  bf a0 9d 14 f8 6b 50 18   ·····Xh0 ·····kP·
0030  04 02 08 3e 00 00 00 00  00 e9 6a 81 e6 30 81 e3   ···>···· ··j··0··
0040  a1 03 02 01 05 a2 03 02  01 0a a3 15 30 13 30 11   ········ ····0·0·
0050  a1 04 02 02 00 80 a2 09  04 07 30 05 a0 03 01 01   ········ ··0·····
0060  ff a4 81 bf 30 81 bc a0  07 03 05 00 40 81 00 10   ····0··· ····@···
0070  a1 1a 30 18 a0 03 02 01  01 a1 11 30 0f 1b 0d 6d   ··0····· ···0···m
0080  61 74 74 68 65 77 2e 6a  6f 6e 65 73 a2 0e 1b 0c   atthew.j ones····
0090  50 49 5a 5a 41 2d 42 45  4e 44 45 52 a3 21 30 1f   PIZZA-BE NDER·!0·
00a0  a0 03 02 01 02 a1 18 30  16 1b 06 6b 72 62 74 67   ·······0 ···krbtg
00b0  74 1b 0c 50 49 5a 5a 41  2d 42 45 4e 44 45 52 a5   t··PIZZA -BENDER·
00c0  11 18 0f 32 30 33 37 30  39 31 33 30 32 34 38 30   ···20370 91302480
00d0  35 5a a6 11 18 0f 32 30  33 37 30 39 31 33 30 32   5Z····20 37091302
00e0  34 38 30 35 5a a7 06 02  04 4f e2 84 9f a8 15 30   4805Z··· ·O·····0
00f0  13 02 01 12 02 01 11 02  01 17 02 01 18 02 02 ff   ········ ········
0100  79 02 01 03 a9 1d 30 1b  30 19 a0 03 02 01 14 a1   y·····0· 0·······
0110  12 04 10 44 45 53 4b 54  4f 50 2d 4f 46 34 46 45   ···DESKT OP-OF4FE
0120  38 41 20                                           8A
```

Figure 2.9: *User Accountable*

As can be seen the kerberos packets were explored. In the cname string the name Matthew.jones is seen. It can be inferred that he is responsible for downloading the malicious malware.

Chapter 3

Question 3 : SQL Injection

3.1 Extract the column names of the table that contains the user data

3.2 Attempt the previous injection in medium security, update your query to bypass the new security measures

3.3 Locate the location of the database on the remote system (File path the database is stored)

3.4 Read a file (e.g. passwords) from the discovered path

3.5 Discover the users with the highest and lowest salaries

3.6 Who has the insurance number: 53779132

Chapter 4

Question 4 : XSS

4.1 Set your DVWA to HIGH security. Analyse the PHP source code and perform an attack that works on Reflected, Stored and DOM XSS pages. Provide a description for each relating to the source code as to why your given attack works.

Chapter 5

Question 5 : Defence

5.1 In a language of your choice, write a short function which handles input from a user which would prevent an SQL injection attack and a Stored XSS attack. Provide a short description of where/why your function is safe against these vulnerabilities.

```
   function sanitizeInput(input) {
  // to remove single quotes, double quotes, and backslashes from the input
    string.
  const sanitizedInput = input.replace(/['"\\]/g, '');

  //  prevents the browser from interpreting the input as HTML and executing
    potentially harmful scripts.
  const safeInput = document.createElement('div');
  safeInput.appendChild(document.createTextNode(sanitizedInput));
  const escapedInput = safeInput.innerHTML;

  return escapedInput;
}
```

Code 5.1: Code Sanitizer

Chapter 6

Question 6: Buffer Overflow

6.1 Choose one of the known buffer overflow vulnerabilities and write a half-page description of it. Aspects that you should cover are:

The FORCEDENTRY buffer overflow vulnerability, also known as CVE-2021-30860, was an integer overflow vulnerability when collating referenced segments[1] resulting in triggering a heap buffer overflow in the ImageIO JBIG2 decoder, used by ImageIO library in CoreGraphicsAPI used in Apple. The entry point for attack was a fake gif containing JBIG2(decoder codec) for PDF which was sent as iMessage, the JBIG2 codec contained code which was Turing Complete i.e. operators like AND, OR, XOR, and XNOR, enabled the attacker to perform bit-level operations on memory regions at arbitrary offsets. Tallowing for arbitrary memory access and computation of any computable function.

6.1.1 Which systems were affected by the vulnerability?

The vulnerability affected all versions of iOS and macOS prior to iOS 14.8 and macOS Big Sur 11.3.1. It is estimated that over 1 billion devices were vulnerable to the attack at the time it was discovered.

6.1.2 When was it discovered, reported and fixed?

This vulnerability was discovered in March 2021 by Citizen Lab, The vulnerability was patched by Apple in iOS 14.8 and macOS Big Sur 11.3.1 updates released for its devices.

6.1.3 What were the known attacks exploiting it, and what were their consequences?

This vulnerability was used by NSO Group for its zero-click iMessage exploit to spy on Saudi activist and Al-jazeera journalists[2].The affected phone was analysed by Citizen Lab and further analysis was done by Ian Beer of Project Zero[1] from Google.NSO was added the "Entity List" by the US government, severely restricting the ability of US companies to do business with NSO.

6.2 Propose a modification of the C programming language that would mitigate buffer overflow vulnerabilities, and discuss the implications of this modification.

Buffer Overflow vulnerabilities have plagued C language since the 1980s, as C allows for direct access to memory and lacks strong object typing.

One Modification that would help in mitigating buffer overflows would be the use of canaries or stack cookies[3] during runtime, They are random values which are placed on stack after each buffer to check for buffer overflows.They may contain other bytes, such as newline characters, that frequently terminate the copying responsible for string-based buffer overflows[4] or some bits unknown to the attacker to prevent return-address clobbering with an integer overflow.

However, they cannot protect against a direct overwrite (by exploiting an indexing error) as they only check for corruption at function exit.

Implications:

- Overhead: Adding stack canaries may add some overhead to the program and therefore slow execution.
- Compatibility issues may arise due to so many legacy code not using stack canaries.
- Using Stack Canaries leads to Reduced Vulnerability Surface and Enhanced Security

Other modifications that are more robust and can guard against more sophisticated attacks are :

1. Address Space Layout Randomisation(ASLR), it provides most comprehensive protection against attacks and is only vulnerable to information leakage attacks.
2. Using Data execution Prevention(DEP) for non-executable stacks, However this approach is vulnerable to Return-Oriented Programming Attacks.
3. Rearrangement of local variables, so that scalar variables are above array variables, so that in case of overflow, static variables are not affected.
4. Bound Checking to check for variables are in appropriate buffers
5. Null termination for handling string buffers
6. Using Static Analysis tools and using secure alternatives to vulnerable functions such as strcpy(), cand writing secure and high quality code.

Bibliography

[1] Google Project Zero. A deep dive into nso zero-click exploits. `https://googleprojectzero.blogspot.com/2021/12/a-deep-dive-into-nso-zero-click.html`, December 2021.

[2] Citizen Lab. Forcedentry: Nso group imessage zero-click exploit captured in the wild. *Citizen Lab Research Publication*, September 2021. URL `https://citizenlab.ca/2021/09/forcedentry-nso-group-imessage-zero-click-exploit-captured-in-the-wild/`.

[3] Laszló Szekeres, Mathias Payer, Tao Wei, and Dawn Song. SoK: Eternal war in memory. In *2013 IEEE Symposium on Security and Privacy*, pages 48–62. IEEE, 2013. URL `https://people.eecs.berkeley.edu/~dawnsong/papers/Oakland13-SoK-CR.pdf`.

[4] Steve Anderson. Low-level security by example. *University of Pennsylvania*, 2019. URL `https://www.cis.upenn.edu/~sga001/classes/cis331f19/resources/low-level-security-by-example.pdf`.

Appendices