Exp04

Matteo Di Fabio: 264339
Luca Genovese: 264364

1) To identify all the involved systems it is necessary to use the following instruction command line: *"tshark -r traffic1.pcap -T fields -e ip.dst ip.src | sort | uniq"* that extract and sort, uniquely, all the IP addresses into pick-up file from the provided pcap file.
   Here there are all the detected addresses:
   - 189.126.11.82
   - 192.168.115.238
   - 200.149.77.224
   - 66.7.200.69
   - 66.7.200.72
   - 8.8.4.4

   An alternative way to find them is using Wireshark's file menu bar option *"Statistics -> Endpoints"*
2) To identify the geolocalisation we decided to use an online tool ("https://whatismyipaddress.com/ip-lookup") to try to obtain these information. The obtained results are the following:
   - 189.126.11.82

     ```
     ISP: Calcontec Tel.ecomunicacoes E Informatica Ltda
     Organization: Calcontec Tel.ecomunicacoes E Informatica
     Ltda
     Services: None detected
     Assignment: Likely Static IP
     Continent: South America
     Country: Brazil
     Latitude: -22.8305  (22° 49′ 49.80″ S)
     Longitude: -43.2192  (43° 13′ 9.12″ W)
     ```

   - 200.149.77.224

     ```
     ISP: Oi Internet
     Organization: Oi Internet
     Services: None detected
     Type: Broadband
     Assignment:     Likely Static IP
     Continent: South America
     Country: Brazil
     State/Region: Rio de Janeiro
     City: Niterói
     Latitude: -22.922  (22° 55′ 19.20″ S)
     Longitude: -43.1025  (43° 6′ 9.00″ W)
     ```

   - 66.7.200.69

```
ISP: HostDime.com
Organization:   HostDime.com
Services:  None detected
Type: Corporate
Assignment: Likely Static IP
Continent: North America
Country: United
Latitude: 37.751  (37° 45′ 3.60″ N)
Longitude: -97.822  (97° 49′ 19.20″ W)
```

- 66.7.200.72

```
ISP: HostDime.com
Organization: HostDime.com
Services: None detected
Type: Corporate
Assignment: Likely Static IP
Continent: North America
Country: United
Latitude: 37.751  (37° 45′ 3.60″ N)
Longitude: -97.822  (97° 49′ 19.20″ W)
```

To identify the location of the hosts, it is also possible to use a bash command or a tool in Wireshark *"Statistics -> Endpoints".* To use this tool, it is necessary to install download a geolocalisation database (GeoLite2-ASN and GeoLite2-City from http://www.maxmind.com) and then insert it in Wireshark. The extracted informations from the captured packet are the following:

Ethernet · 2 | IPv4 · 6 | IPv6 | TCP · 13 | UDP · 5

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number	AS Organization
8.8.4.4	8	782	4	464	4	318	United States	—	15169	GOOGLE
66.7.200.69	424	390 k	255	380 k	169	9,335	United States	—	33182	DIMENOC
66.7.200.72	18	2,394	8	1,528	10	866	United States	—	33182	DIMENOC
189.126.11.82	36	4,094	14	1,476	22	2,618	Brazil	—	270398	CALCONTEC TEL.ECOMUNICACOES E INFORMATICA LTDA
192.168.115.238	1,762	1,627 k	672	38 k	1,090	1,588 k	—	—	—	—
200.149.77.224	1,276	1,230 k	809	1,204 k	467	25 k	Brazil	Niterói	7738	Telemar Norte Leste S.A.

The IP from which the victim download the malware is '66.7.200.69'.

3) To identify the number of TCP session it is possible to use a bash command (tshark and/or snort) or a tool in Wireshark *"Statistics -> Conversation".* The obtained results are 9 TCP sessions with a total of 1756 exchanged packets:
   - 192.168.115.238 on port 1126 <-> 200.149.77.224 on port 80 packets: 1276
   - 192.168.115.238 on port 1127 <-> 66.7.200.69 on port 80 packets: 424
   - 192.168.115.238 on port 1128 <-> 66.7.200.72 on port 80 packets: 9
   - 192.168.115.238 on port 1129 <-> 189.126.11.82 on port 80 packets: 3
   - 192.168.115.238 on port 1130 <-> 66.7.200.72 on port 80 packets: 9
   - 192.168.115.238 on port 1131 <-> 189.126.11.82 on port 80 packets: 11
   - 192.168.115.238 on port 1132 <-> 189.126.11.82 on port 80 packets: 12
   - 192.168.115.238 on port 1133 <-> 189.126.11.82 on port 80 packets: 3
   - 192.168.115.238 on port 1136 <-> 189.126.11.82 on port 80 packets: 7

screenshot in wireshark

Ethernet · 1 | IPv4 · 5 | IPv6 | TCP · 9 | UDP · 4

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration
192.168.115.238	1126	200.149.77.224	80	1,276	1,230 k	467	25 k	809	1,204 k	1.546345	65.7965
192.168.115.238	1127	66.7.200.69	80	424	390 k	169	9,335	255	380 k	22.997932	7.7395
192.168.115.238	1128	66.7.200.72	80	9	1,197	5	433	4	764	25.550195	0.7188
192.168.115.238	1129	189.126.11.82	80	3	186	3	186	0	0	27.065005	9.9025
192.168.115.238	1130	66.7.200.72	80	9	1,197	5	433	4	764	35.900375	0.7514
192.168.115.238	1131	189.126.11.82	80	11	1,372	6	726	5	646	37.187016	36.5656
192.168.115.238	1132	189.126.11.82	80	12	1,482	6	834	6	648	49.704174	24.0238
192.168.115.238	1133	189.126.11.82	80	3	186	3	186	0	0	58.670457	8.9377
192.168.115.238	1136	189.126.11.82	80	7	868	4	686	3	182	73.716120	0.1064

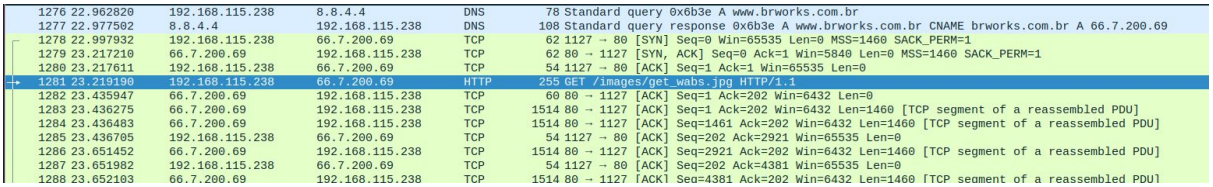
screenshot of result using snort

```
===================================
Stream statistics:
             Total sessions: 13
               TCP sessions: 9
               UDP sessions: 4
              ICMP sessions: 0
                IP sessions: 0
                 TCP Prunes: 0
                 UDP Prunes: 0
                ICMP Prunes: 0
                  IP Prunes: 0
TCP StreamTrackers Created: 9
TCP StreamTrackers Deleted: 9
               TCP Timeouts: 0
               TCP Overlaps: 0
        TCP Segments Queued: 1073
      TCP Segments Released: 1073
        TCP Rebuilt Packets: 100
          TCP Segments Used: 1073
               TCP Discards: 1
                   TCP Gaps: 2
       UDP Sessions Created: 4
       UDP Sessions Deleted: 4
               UDP Timeouts: 0
               UDP Discards: 0
                     Events: 0
            Internal Events: 0
            TCP Port Filter
                   Filtered: 0
                  Inspected: 0
                    Tracked: 1754
            UDP Port Filter
                   Filtered: 0
                  Inspected: 0
                    Tracked: 4
===================================
```

4) The attack that we detected is a DNS spoofing (DNS Shadow server) on Google DNS *"8.8.4.4"* after the query request of "*www.brworks.com.br*" that the victim (*"192.168.115.238")* does. As this kind of attack is characterized by providing an

answer before the correct answer (from Google DNS in this case) and in this way the answer that arrives late will be discarded from the client as duplicate.

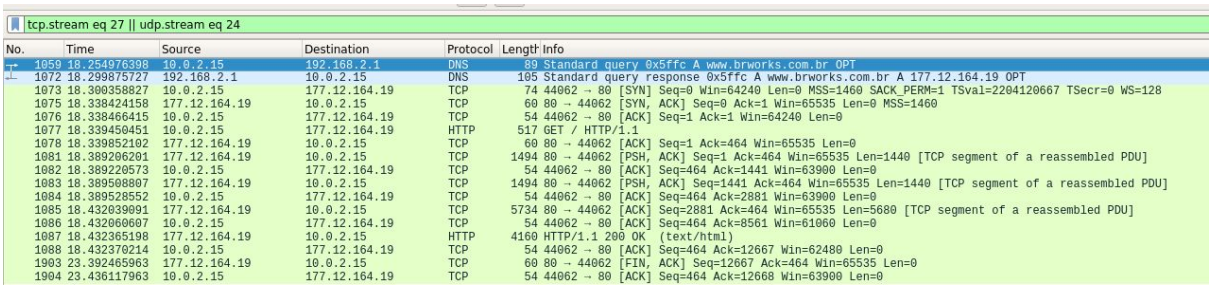
The attack lasts the time needed by the attacker to answer before Google DNS so as



1276	22.962820	192.168.115.238	8.8.4.4	DNS	78	Standard query 0x6b3e A www.brworks.com.br
1277	22.977502	8.8.4.4	192.168.115.238	DNS	108	Standard query response 0x6b3e A www.brworks.com.br CNAME brworks.com.br A 66.7.200.69
1278	22.997932	192.168.115.238	66.7.200.69	TCP	62	1127 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1
1279	23.217210	66.7.200.69	192.168.115.238	TCP	62	80 → 1127 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460 SACK_PERM=1
1280	23.217611	192.168.115.238	66.7.200.69	TCP	54	1127 → 80 [ACK] Seq=1 Ack=1 Win=65535 Len=0
1281	23.219190	192.168.115.238	66.7.200.69	HTTP	255	GET /images/get_wabs.jpg HTTP/1.1
1282	23.435947	66.7.200.69	192.168.115.238	TCP	60	80 → 1127 [ACK] Seq=1 Ack=202 Win=6432 Len=0
1283	23.436275	66.7.200.69	192.168.115.238	TCP	1514	80 → 1127 [ACK] Seq=1 Ack=202 Win=6432 Len=1460 [TCP segment of a reassembled PDU]
1284	23.436483	66.7.200.69	192.168.115.238	TCP	1514	80 → 1127 [ACK] Seq=1461 Ack=202 Win=6432 Len=1460 [TCP segment of a reassembled PDU]
1285	23.436705	192.168.115.238	66.7.200.69	TCP	54	1127 → 80 [ACK] Seq=202 Ack=2921 Win=65535 Len=0
1286	23.651452	66.7.200.69	192.168.115.238	TCP	1514	80 → 1127 [ACK] Seq=2921 Ack=202 Win=6432 Len=1460 [TCP segment of a reassembled PDU]
1287	23.651982	192.168.115.238	66.7.200.69	TCP	54	1127 → 80 [ACK] Seq=202 Ack=4381 Win=65535 Len=0
1288	23.652103	66.7.200.69	192.168.115.238	TCP	1514	80 → 1127 [ACK] Seq=4381 Ack=202 Win=6432 Len=1460 [TCP segment of a reassembled PDU]

it is possible to observe it takes 0.0014682 seconds.

In this second case instead we try to emulate the same query with our PC and as we



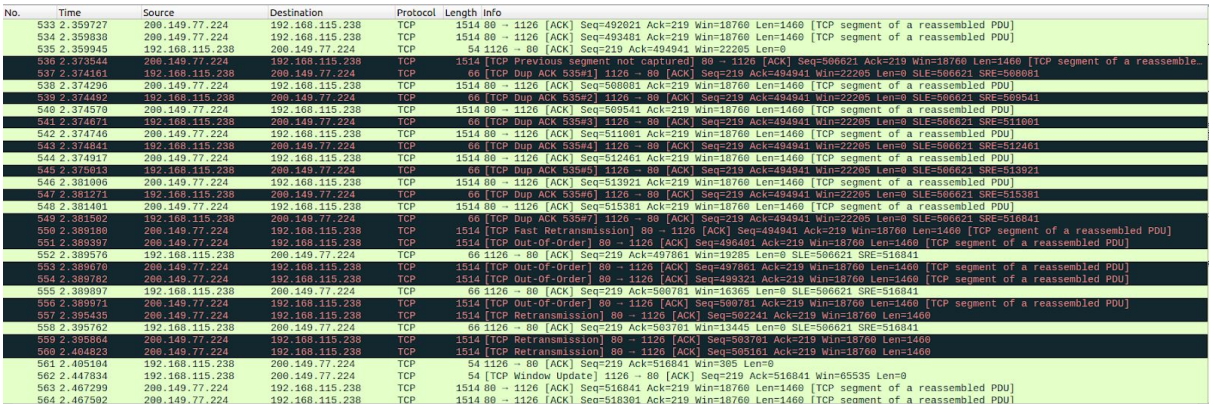
tcp.stream eq 27 || udp.stream eq 24

No.	Time	Source	Destination	Protocol	Length	Info
1059	18.254976398	10.0.2.15	192.168.2.1	DNS	89	Standard query 0x5ffc A www.brworks.com.br OPT
1072	18.299875727	192.168.2.1	10.0.2.15	DNS	105	Standard query response 0x5ffc A www.brworks.com.br A 177.12.164.19 OPT
1073	18.300358827	10.0.2.15	177.12.164.19	TCP	74	44062 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2204120667 TSecr=0 WS=128
1075	18.338424158	177.12.164.19	10.0.2.15	TCP	60	80 → 44062 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
1076	18.338466415	10.0.2.15	177.12.164.19	TCP	54	44062 → 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
1077	18.339450451	10.0.2.15	177.12.164.19	HTTP	517	GET / HTTP/1.1
1078	18.339852102	177.12.164.19	10.0.2.15	TCP	60	80 → 44062 [ACK] Seq=1 Ack=464 Win=65535 Len=0
1081	18.389206201	177.12.164.19	10.0.2.15	TCP	1494	80 → 44062 [PSH, ACK] Seq=1 Ack=464 Win=65535 Len=1440 [TCP segment of a reassembled PDU]
1082	18.389220573	10.0.2.15	177.12.164.19	TCP	54	44062 → 80 [ACK] Seq=464 Ack=1441 Win=63900 Len=0
1083	18.389508807	177.12.164.19	10.0.2.15	TCP	1494	80 → 44062 [PSH, ACK] Seq=1441 Ack=464 Win=65535 Len=1440 [TCP segment of a reassembled PDU]
1084	18.389528552	10.0.2.15	177.12.164.19	TCP	54	44062 → 80 [ACK] Seq=464 Ack=2881 Win=63900 Len=0
1085	18.432039091	177.12.164.19	10.0.2.15	TCP	5734	80 → 44062 [ACK] Seq=2881 Ack=464 Win=65535 Len=5680 [TCP segment of a reassembled PDU]
1086	18.432060607	10.0.2.15	177.12.164.19	TCP	54	44062 → 80 [ACK] Seq=464 Ack=8561 Win=61060 Len=0
1087	18.432365198	177.12.164.19	10.0.2.15	HTTP	4160	HTTP/1.1 200 OK (text/html)
1088	18.432370214	10.0.2.15	177.12.164.19	TCP	54	44062 → 80 [ACK] Seq=464 Ack=12667 Win=62480 Len=0
1903	23.392465963	177.12.164.19	10.0.2.15	TCP	60	80 → 44062 [FIN, ACK] Seq=12667 Ack=464 Win=65535 Len=0
1904	23.436117963	10.0.2.15	177.12.164.19	TCP	54	44062 → 80 [ACK] Seq=464 Ack=12668 Win=63900 Len=0

can see we receive a different IP (the right one): '177.12.164.19'.

The duration of the total connection and consequently download the file *'get_wabs.jpg'* (that is the malware that we identify as malicious) and finally close the session with '66.7.200.69': so 7,7395 seconds.

5) To identify the attack we have analyzed all the packet using the default view and the flow graph using *"Statistics->FLow Graph"*.



No.	Time	Source	Destination	Protocol	Length	Info
533	2.359727	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=492021 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
534	2.359838	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=493481 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
535	2.359945	192.168.115.238	200.149.77.224	TCP	54	1126 → 80 [ACK] Seq=219 Ack=494941 Win=22205 Len=0
536	2.373544	200.149.77.224	192.168.115.238	TCP	1514	[TCP Previous segment not captured] 80 → 1126 [ACK] Seq=506621 Ack=219 Win=18760 Len=1460 [TCP segment of a reassemble…
537	2.374161	192.168.115.238	200.149.77.224	TCP	66	[TCP Dup ACK 535#1] 1126 → 80 [ACK] Seq=219 Ack=494941 Win=22205 Len=0 SLE=506621 SRE=508081
538	2.374296	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=508081 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
539	2.374492	192.168.115.238	200.149.77.224	TCP	66	[TCP Dup ACK 535#2] 1126 → 80 [ACK] Seq=219 Ack=494941 Win=22205 Len=0 SLE=506621 SRE=509541
540	2.374570	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=509541 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
541	2.374671	192.168.115.238	200.149.77.224	TCP	66	[TCP Dup ACK 535#3] 1126 → 80 [ACK] Seq=219 Ack=494941 Win=22205 Len=0 SLE=506621 SRE=511001
542	2.374746	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=511001 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
543	2.374841	192.168.115.238	200.149.77.224	TCP	66	[TCP Dup ACK 535#4] 1126 → 80 [ACK] Seq=219 Ack=494941 Win=22205 Len=0 SLE=506621 SRE=512461
544	2.374917	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=512461 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
545	2.375013	192.168.115.238	200.149.77.224	TCP	66	[TCP Dup ACK 535#5] 1126 → 80 [ACK] Seq=219 Ack=494941 Win=22205 Len=0 SLE=506621 SRE=513921
546	2.381006	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=513921 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
547	2.381271	192.168.115.238	200.149.77.224	TCP	66	[TCP Dup ACK 535#6] 1126 → 80 [ACK] Seq=219 Ack=494941 Win=22205 Len=0 SLE=506621 SRE=515381
548	2.381401	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=515381 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
549	2.381502	192.168.115.238	200.149.77.224	TCP	66	[TCP Dup ACK 535#7] 1126 → 80 [ACK] Seq=219 Ack=494941 Win=22205 Len=0 SLE=506621 SRE=516841
550	2.389180	200.149.77.224	192.168.115.238	TCP	1514	[TCP Fast Retransmission] 80 → 1126 [ACK] Seq=494941 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
551	2.389397	200.149.77.224	192.168.115.238	TCP	1514	[TCP Out-Of-Order] 80 → 1126 [ACK] Seq=496401 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
552	2.389576	192.168.115.238	200.149.77.224	TCP	66	1126 → 80 [ACK] Seq=219 Ack=497861 Win=19205 Len=0 SLE=506621 SRE=516841
553	2.389670	200.149.77.224	192.168.115.238	TCP	1514	[TCP Out-Of-Order] 80 → 1126 [ACK] Seq=497861 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
554	2.389782	200.149.77.224	192.168.115.238	TCP	1514	[TCP Out-Of-Order] 80 → 1126 [ACK] Seq=499321 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
555	2.389897	192.168.115.238	200.149.77.224	TCP	66	1126 → 80 [ACK] Seq=219 Ack=500781 Win=16365 Len=0 SLE=506621 SRE=516841
556	2.389971	200.149.77.224	192.168.115.238	TCP	1514	[TCP Out-Of-Order] 80 → 1126 [ACK] Seq=500781 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
557	2.395435	200.149.77.224	192.168.115.238	TCP	1514	[TCP Retransmission] 80 → 1126 [ACK] Seq=502241 Ack=219 Win=18760 Len=1460
558	2.395762	192.168.115.238	200.149.77.224	TCP	66	1126 → 80 [ACK] Seq=219 Ack=503701 Win=13445 Len=0 SLE=506621 SRE=516841
559	2.395864	200.149.77.224	192.168.115.238	TCP	1514	[TCP Retransmission] 80 → 1126 [ACK] Seq=503701 Ack=219 Win=18760 Len=1460
560	2.404823	200.149.77.224	192.168.115.238	TCP	1514	[TCP Retransmission] 80 → 1126 [ACK] Seq=505161 Ack=219 Win=18760 Len=1460
561	2.405104	192.168.115.238	200.149.77.224	TCP	54	1126 → 80 [ACK] Seq=219 Ack=516841 Win=305 Len=0
562	2.447834	192.168.115.238	200.149.77.224	TCP	54	[TCP Window Update] 1126 → 80 [ACK] Seq=219 Ack=516841 Win=65535 Len=0
563	2.467299	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=516841 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]
564	2.467502	200.149.77.224	192.168.115.238	TCP	1514	80 → 1126 [ACK] Seq=518301 Ack=219 Win=18760 Len=1460 [TCP segment of a reassembled PDU]

In this first analysis we just note some packets loss, that were detected by WireShark and after a while restransmited as expected for the TCP protocol.

At this point the only weird behaviour is that the first connection was closed just after a long time, after all the other connection were closed. Another anomaly was in the SYN packet (1712) for the website "trabucar.com.br" in the last connection *'189.126.11.82'*. As after a long time the client did not received a response, the SYN is retransmitted by the client in the packet 1713 (after 3 seconds) and then again after 6 seconds in 1726 (from the second retransmission).

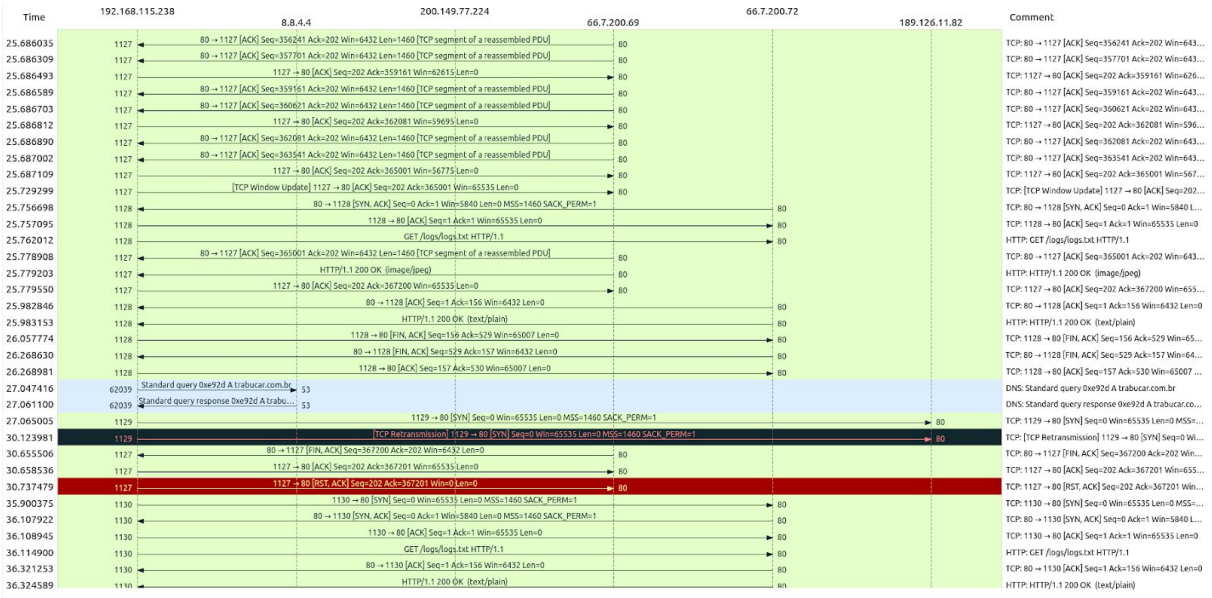
After that preliminary analyses we didn't find any anomaly, so at this point we decided to used the tool *"Analysis->Expert information"* and as we can observe at the packet 1703 on 66.7.200.69 connection, is detected a malformed packet.

Severity	Summary	Group	Protocol	Count
Warning	Connection reset (RST)	Sequence	TCP	2
Warning	ACKed segment that wasn't captured (common at capture s...	Sequence	TCP	16
Warning	This frame is a (suspected) out-of-order segment	Sequence	TCP	6
Warning	Previous segment(s) not captured (common at capture start)	Sequence	TCP	4
Warning	TCP Zero Window segment	Sequence	TCP	6
Warning	TCP window specified by the receiver is now completely full	Sequence	TCP	12
Note	HTTP body subdissector failed, trying heuristic subdissector	Malformed	HTTP	1
1703	HTTP/1.1 200 OK (image/jpeg)	Malformed	HTTP	
Note	This frame is a (suspected) fast retransmission	Sequence	TCP	3
Note	This frame is a (suspected) retransmission	Sequence	TCP	11
Note	Duplicate ACK (#1)	Sequence	TCP	14
Chat	Connection finish (FIN)	Sequence	TCP	9
Chat	TCP window update	Sequence	TCP	38
Chat	GET /DATA-FILES/ARQUIVO12.XLS HTTP/1.1\r\n	Sequence	HTTP	12
Chat	Connection establish acknowledge (SYN+ACK): server port 80	Sequence	TCP	7
Chat	Connection establish request (SYN): server port 80	Sequence	TCP	13

Now, knowing that it is possible to make some specific analysis on these packet (around 1703) with flow graph and it is possible to note that the client make a query request to download an image: *'get_wabs.jpg'.*

As it is explained in answer 6 we export this file to be able to analyse it in VirusTotal website and this file was identified as a malware.

Knowing that we continued our analysis and because of this image is the malware, downloaded on client (victim's machine), now it is necessary to understand why the victim downloaded this file.



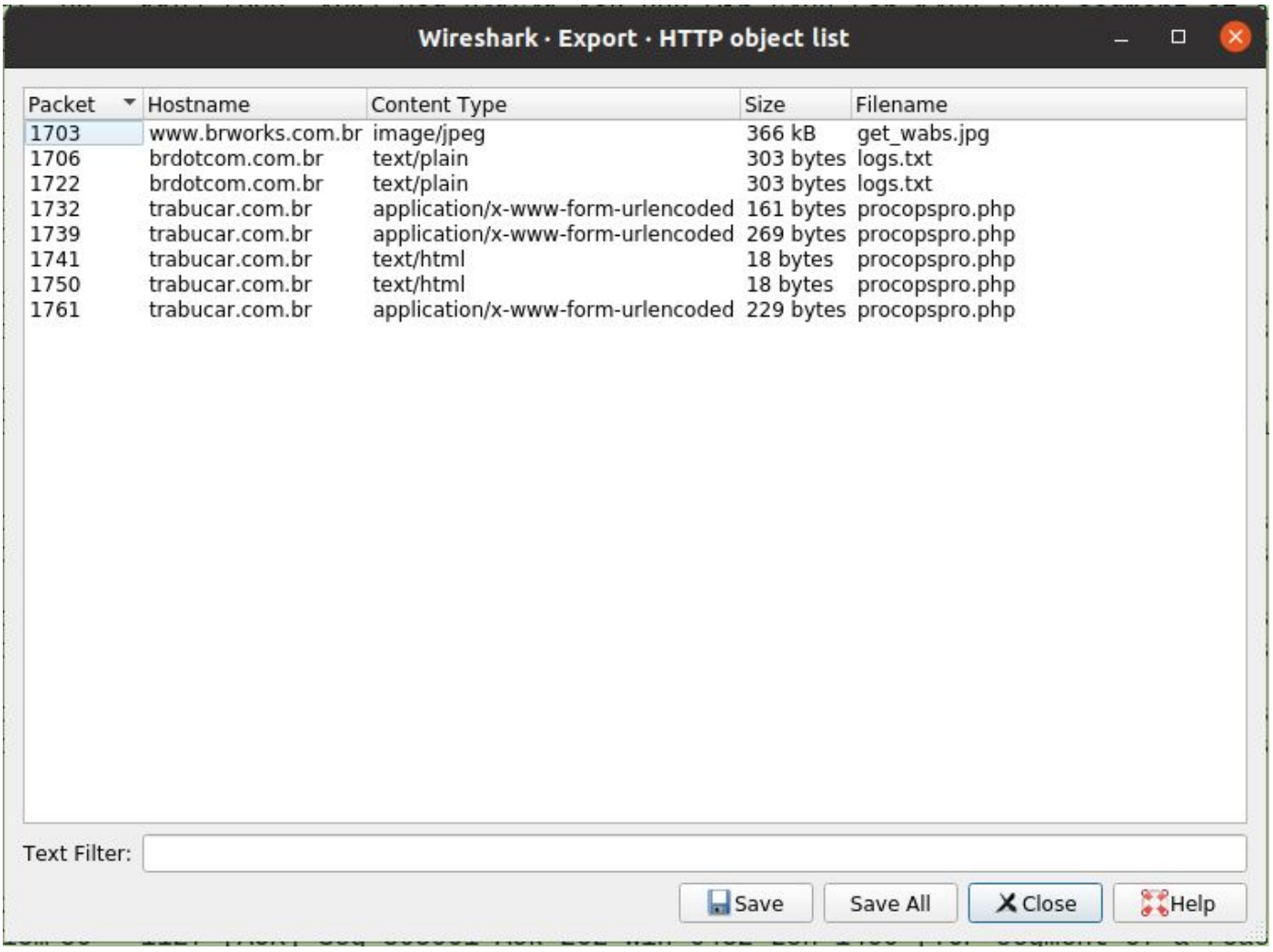
As it is possible to observe the client/victim, make a query request to access on "https://www.brworks.com.br/" and he received back *"66.7.200.69"*. Now as it is explained in answer 4 the attacker (that is a man in the middle) carry out a DNS spoofing (Shadow server) attack.

We can just suppose that this is the attack, as we didn't identify an evident prove that this kind of attack has been carried out by a man in the middle. In fact for this reason it is much better to sniff not on the client-local NS but on geographical route,

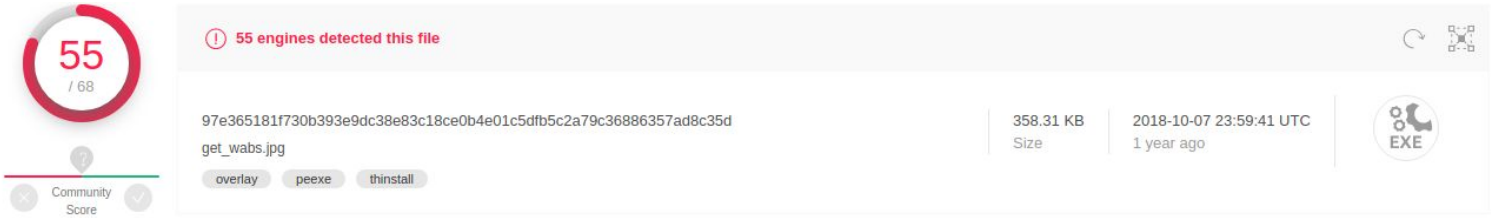
for example, between the local-NS and the root-NS. So if a wrong answer is provided to local-NS, this will be cached and therefore will be provided to all clients on the local network that will make this query.

Note: we can't know if this IP is changed in these years as the capture is made in the year 2010 and the emulation in 2020.

6) Yes, we identify the file *'get_wabs.jpg'* as malicious file through the webapp virustotal.com that analyse an uploaded file using a lot different antivirus in parallel.
   We have followed the following steps to recognize the malicious file:
   a) In Wireshark we have downloaded all the files downloaded during the traffic.pcap capture using the window bar function *"File->Export Objects->HTTP"* and clicked on save all to take all the files.



b) Upload all files on virustotal.com and find out if the uploaded file is a malware. This is the result with the file *'get_wabs.jpg'* downloaded from 66.7.200.69:



DETECTION | DETAILS | RELATIONS | BEHAVIOR | COMMUNITY 1

Ad-Aware	Trojan.Generic.KDV.604151	AegisLab	Trojan.Win32.Homa.a!c
AhnLab-V3	Trojan/Win32.Banki.C2231696	ALYac	Trojan.Generic.KDV.604151
Antiy-AVL	Trojan/Win32.TGeneric	Arcabit	Trojan.Generic.KDV.D937F7
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira (no cloud)	TR/Dropper.Gen	AVware	Trojan.Win32.Generic!BT
BitDefender	Trojan.Generic.KDV.604151	Bkav	W32.eHeur.Virus02
Cybereason	Malicious.ef0bed	Cylance	Unsafe
Cyren	W32/Risk.MJIC-1164	DrWeb	Trojan.PWS.Banker.51815
Emsisoft	Trojan.Generic.KDV.604151 (B)	Endgame	Malicious (high Confidence)
eScan	Trojan.Generic.KDV.604151	ESET-NOD32	Win32/Spy.Banker.SWH
F-Prot	W32/MalwareF.FMWV	F-Secure	Trojan.Generic.KDV.604151
Fortinet	W32/Homa.AXO!tr.dldr	GData	Trojan.Generic.KDV.604151
Ikarus	Trojan-Spy.Agent	Jiangmin	TrojanDownloader.Homa.apy
K7AntiVirus	Spyware (004d0dfc1)	K7GW	Spyware (004d0dfc1)

As it is possible to observe the *'get_wabs.jpg'* is detected by 55 over 68 antivirus like a malware: some identify it like a trojan other like a Spyware, but in general like a malware.