

# Janus

The goal of the project was to be able to scan a networked environment for potential security violations through a mobile application.

# Wireshark Output

test.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.159.110.99	23.2.42.218	TCP	54	60874 → 443 [FIN, ACK] Seq=1 Ack=1 Win=258 Len=0
2	0.000321	10.159.110.99	23.2.42.218	TCP	54	60873 → 443 [FIN, ACK] Seq=1 Ack=1 Win=258 Len=0
3	0.000386	10.159.110.99	23.2.42.218	TCP	54	60872 → 443 [FIN, ACK] Seq=1 Ack=1 Win=258 Len=0
4	0.000493	10.159.110.99	23.2.42.218	TCP	54	60875 → 443 [FIN, ACK] Seq=1 Ack=1 Win=258 Len=0
5	0.004510	23.2.42.218	10.159.110.99	TLSv1.2	85	Encrypted Alert
6	0.004545	10.159.110.99	23.2.42.218	TCP	54	60872 → 443 [RST, ACK] Seq=2 Ack=32 Win=0 Len=0
7	0.004813	23.2.42.218	10.159.110.99	TCP	54	443 → 60872 [FIN, ACK] Seq=32 Ack=2 Win=245 Len=0
8	0.004813	23.2.42.218	10.159.110.99	TLSv1.2	85	Encrypted Alert
9	0.004813	23.2.42.218	10.159.110.99	TCP	54	443 → 60874 [FIN, ACK] Seq=32 Ack=2 Win=245 Len=0
10	0.004843	10.159.110.99	23.2.42.218	TCP	54	60874 → 443 [RST, ACK] Seq=2 Ack=32 Win=0 Len=0
11	0.005410	23.2.42.218	10.159.110.99	TLSv1.2	85	Encrypted Alert
12	0.005429	10.159.110.99	23.2.42.218	TCP	54	60873 → 443 [RST, ACK] Seq=2 Ack=32 Win=0 Len=0
13	0.005688	23.2.42.218	10.159.110.99	TCP	54	443 → 60873 [FIN, ACK] Seq=32 Ack=2 Win=245 Len=0
14	0.007516	23.2.42.218	10.159.110.99	TLSv1.2	85	Encrypted Alert
15	0.007535	10.159.110.99	23.2.42.218	TCP	54	60875 → 443 [RST, ACK] Seq=2 Ack=32 Win=0 Len=0
16	0.007796	23.2.42.218	10.159.110.99	TCP	54	443 → 60875 [FIN, ACK] Seq=32 Ack=2 Win=245 Len=0
17	23.328922	10.159.110.99	52.177.166.224	TLSv1.2	98	Application Data
18	23.338450	52.177.166.224	10.159.110.99	TLSv1.2	180	Application Data
19	23.391218	10.159.110.99	52.177.166.224	TCP	54	55480 → 443 [ACK] Seq=45 Ack=127 Win=258 Len=0
20	28.211106	LiteonTe_70:57:4b	Cisco_ff:fd:e0	ARP	42	Who has 10.159.0.1? Tell 10.159.110.99
21	28.212675	Cisco_ff:fd:e0	LiteonTe_70:57:4b	ARP	60	10.159.0.1 is at 00:08:e3:ff:fd:e0
22	39.555922	151.139.128.14	10.159.110.99	TCP	54	80 → 60883 [FIN, ACK] Seq=1 Ack=1 Win=237 Len=0
23	39.556032	10.159.110.99	151.139.128.14	TCP	54	60883 → 80 [ACK] Seq=1 Ack=2 Win=255 Len=0
24	39.556065	10.159.110.99	151.139.128.14	TCP	54	60883 → 80 [FIN, ACK] Seq=1 Ack=2 Win=255 Len=0
25	39.559885	151.139.128.14	10.159.110.99	TCP	54	80 → 60883 [ACK] Seq=2 Ack=2 Win=237 Len=0
26	39.574183	151.139.128.14	10.159.110.99	TCP	54	80 → 60884 [FIN, ACK] Seq=1 Ack=1 Win=119 Len=0
27	39.574272	10.159.110.99	151.139.128.14	TCP	54	60884 → 80 [ACK] Seq=1 Ack=2 Win=255 Len=0
28	39.574396	10.159.110.99	151.139.128.14	TCP	54	60884 → 80 [FIN, ACK] Seq=1 Ack=2 Win=255 Len=0
29	39.579416	151.139.128.14	10.159.110.99	TCP	54	80 → 60884 [ACK] Seq=2 Ack=2 Win=119 Len=0

> Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0  
> Ethernet II, Src: LiteonTe\_70:57:4b (3c:95:09:70:57:4b), Dst: Cisco\_ff:fd:e0 (00:08:e3:ff:fd:e0)  
> Internet Protocol Version 4, Src: 10.159.110.99, Dst: 23.2.42.218  
> Transmission Control Protocol, Src Port: 60874, Dst Port: 443, Seq: 1, Ack: 1, Len: 0

0000 00 08 e3 ff fd e0 3c 95 09 70 57 4b 08 00 45 00 .....<-pWk-E-  
0010 00 28 04 f6 40 00 80 06 3a fc 0a 9f 6e 63 17 02 -(.-@.-.-.-nc--

test.pcapng

Packets: 29 · Displayed: 29 (100.0%) Profile: Default

Mail has new messages

Type here to search

9:00 PM 12/12/2019

# Three Parts of The Design

- The first part of the design was a sniffer to collect data and tag the location that the data was collected from.
- The second part was the storage and the analysis of the data for potential security threats.
- The last phase is the ability to take action.

# Sniffer

- The sniffer collects the following information:
  - MAC address of the device. ( allows for the tracking of the device if it changes locations in the network)
  - GPS location that the device was located at.
  - IP addresses visited.
  - Date/Time

# Server

- The server provides the following functions:
  - Stores information in case there is an incident, and the information needs to be reviewed
  - Allows the network administrator to flag potential threats for the next stage.
  - Allows for the data to be fairly anonymous

# Action

- This module provides the following functions:
  - Sends a “last known GPS location”, a MAC address and the security violation to be addressed.

# Results of the Project

## Why No One Else Did This

- In order for this project to reach its full potential the device has to be rooted and the PCAP libraries have to be added to the device.
- Then code has to be added to the App to call a shell on the device and to execute the code for the PCAP libraries.
- The results of these calls then have to be translated back to Android, and then interpreted.
- All of these calls and interpretation have to occur in real time.

# What Started This Project

- There is a company called PwnieExpress
- A couple of years back they made the prototype phone for a TV series called “Mr Robot”
- I was wondering how far I could push this design into a real functioning network testing tool.





# Summary of the Results

- While a mobile device for the testing of a network may sound appealing, the reality of the design is not.
- The better option is to use a laptop running COTS software packages that have years of testing and research behind them.
- These results can then be passed to the server, and analyzed.
- From this point the results of the analysis could be passed to the mobile devices for remediation

# Questions

## Better Options for Network Security Testing

