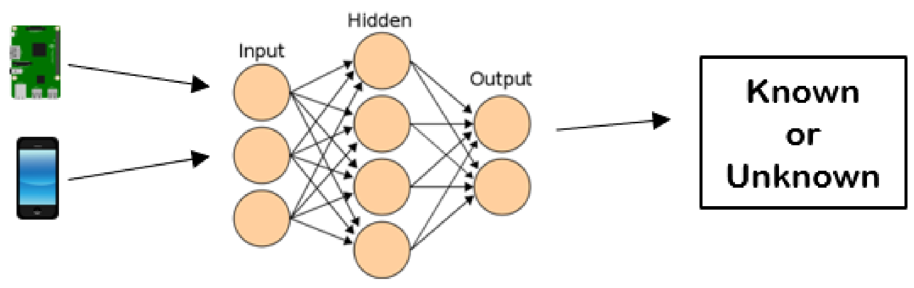
DEVICE FINGERPRINTING

PI: Prof. Brejesh Lall, CO-PI: Dr. Sk Subidh Ali, PS: Suman Kalyan Maiti

Detecting a device is known or unknown is an important issue for a local network. Devices are connected in the network via the access point so any device which has the credentials can be connected with spoofing a legitimate device. Device fingerprinting is the solution to detect a device known or unknown based on the device's inherent properties which are varies device to device and also unique and difficult to change.

INTRODUCTION

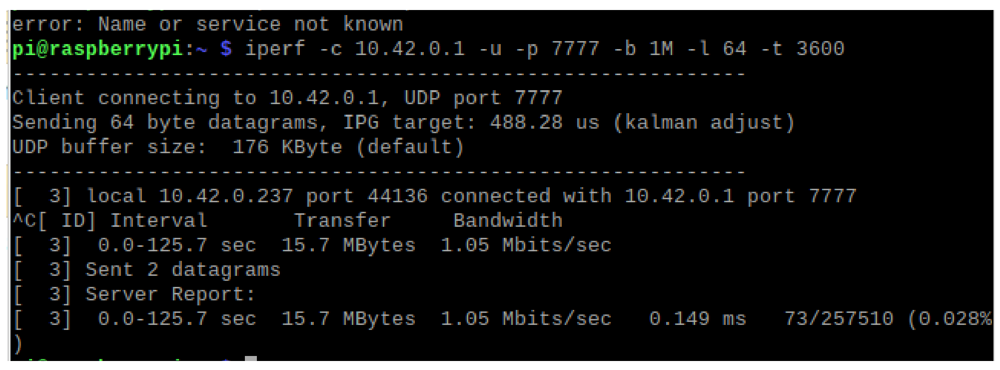
- Device fingerprinting ensure that a device can be uniquely identified based on its unique inherent features.
- Each device has a different hardware configuration (e.g., processor, DMA controller, memory, clock skew, etc.). These small changes make a device different from others.
- When a device connected to a network it has unique IP and MAC address with inherit network packet sending pattern which depends on the hardware configuration.
- The major components that affect the creation of packets are: the CPU, L1/L2 cache, physical memory, the DMA controller, the NIC, etc.
- Here the Inter Arrival Time(IAT) of network traffic packets outgoing from the device is considered to create unique, reproducible device signatures.
- We collect traffic between the access point (AP) and device in a local network to generate IAT signatures and use Artificial Neural Network (ANN) model to identify devices as known or unknown.



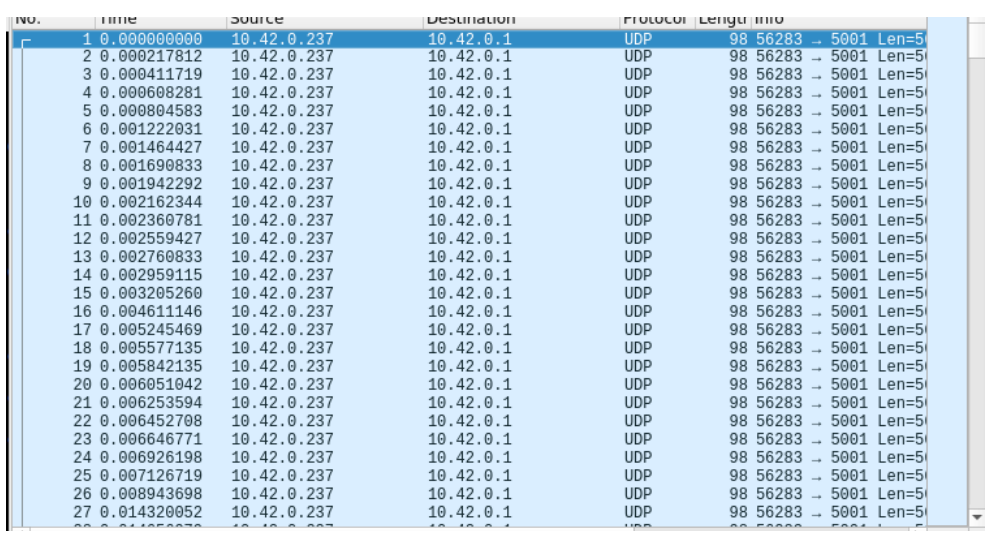
USE-CASE SETUP

1. Raspberry pi as a server to accept UDP packets.
2. Device sending UDP packets to the server.
3. Capturing Packets on the access point.

```
pi@raspberrypi:~ $ iperf -s -u -p 7777
------------------------------------------------------------
Server listening on UDP port 7777
Receiving 1470 byte datagrams
UDP buffer size:  176 KByte (default)
------------------------------------------------------------
[  3] local 10.42.0.1 port 7777 connected with 10.42.0.237 port 44136
[ ID] Interval       Transfer     Bandwidth        Jitter   Lost/Total Datagrams
[  3]  0.0-125.7 sec  15.7 MBytes  1.05 Mbits/sec   0.149 ms   73/257510 (0.028%
)
```



```
error: Name or service not known
pi@raspberrypi:~ $ iperf -c 10.42.0.1 -u -p 7777 -b 1M -l 64 -t 3600
------------------------------------------------------------
Client connecting to 10.42.0.1, UDP port 7777
Sending 64 byte datagrams, IPG target: 488.28 us (kalman adjust)
UDP buffer size:  176 KByte (default)
------------------------------------------------------------
[  3] local 10.42.0.237 port 44136 connected with 10.42.0.1 port 7777
^C[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-125.7 sec  15.7 MBytes  1.05 Mbits/sec
[  3] Sent 2 datagrams
[  3] Server Report:
[  3]  0.0-125.7 sec  15.7 MBytes  1.05 Mbits/sec   0.149 ms   73/257510 (0.028%
)
```

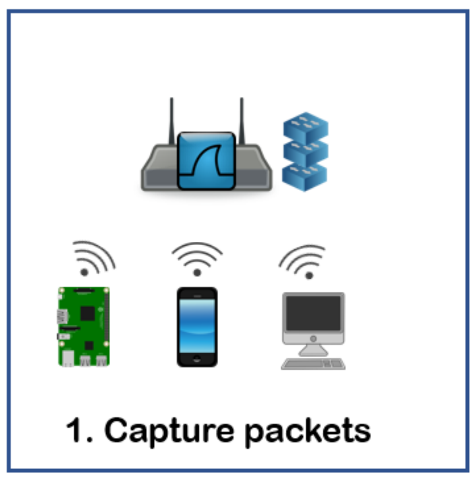


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
2	0.000217812	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
3	0.000411719	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
4	0.000608281	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
5	0.000804583	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
6	0.001222031	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
7	0.001464427	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
8	0.001690833	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
9	0.001942292	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
10	0.002162344	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
11	0.002360781	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
12	0.002559427	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
13	0.002760833	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
14	0.002959115	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
15	0.003205260	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
16	0.004611146	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
17	0.005245469	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
18	0.005577135	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
19	0.005842135	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
20	0.006051042	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
21	0.006253594	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
22	0.006452708	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
23	0.006646771	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
24	0.006926198	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
25	0.007126719	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
26	0.008943698	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5
27	0.014320052	10.42.0.237	10.42.0.1	UDP	98	56283 → 5001 Len=5

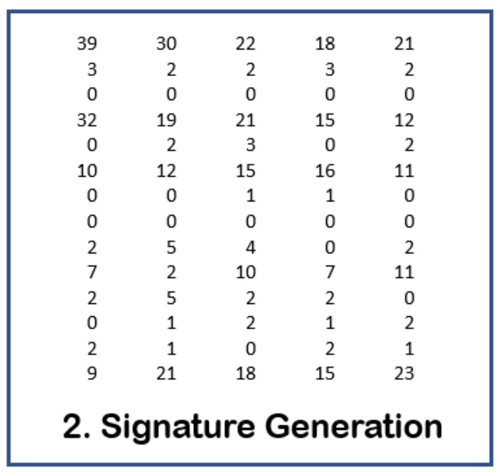
METHODOLOGY

Pre-processing:

1. Capture Packets of connected devices.
2. Extract arrival time, protocol, source, port etc. from each packet
3. Separate each device traffic based on device MAC address.
4. Generate Inter Arrival time of each packet.
5. Separate Inter Arrival Time for each traffic type.
6. Generate Signature using 2500 IATs and each device has 1000 signatures of size 300.



1. Capture packets

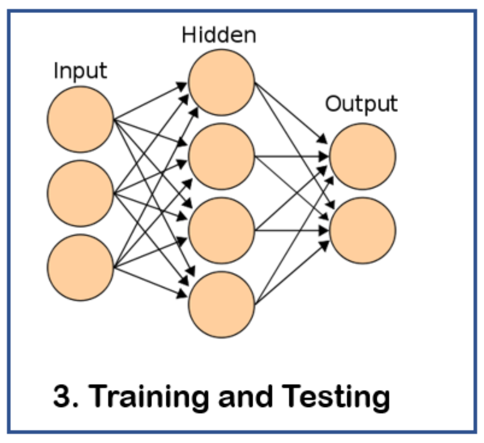


39	30	22	18	21
3	2	2	3	2
0	0	0	0	0
32	19	21	15	12
0	2	3	0	2
10	12	15	16	11
0	0	1	1	0
0	0	0	0	0
2	5	4	0	2
7	2	10	7	11
2	5	2	2	0
0	1	2	1	2
2	1	0	2	1
9	21	18	15	23

2. Signature Generation

Training:

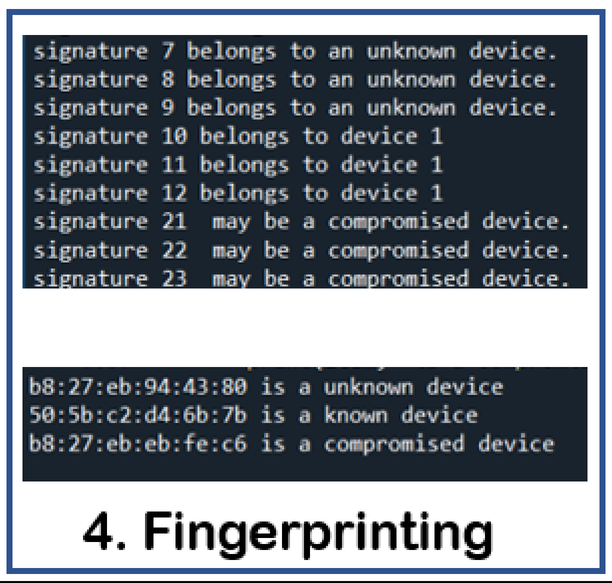
1. Make an ANN model with 3 hidden layers with sigmoid activation and loss as binary cross-entropy.
2. Train the model with (signature, device id).
3. Save the model with True Positive probabilities for each device.



3. Training and Testing

Fingerprinting

1. For a selected device do pre-process step to generate n unknown signatures.
2. Take true positive probabilities for each trained device from the saved model.
3. Feed the unknown signatures to the trained model to collect the predicted class and its probability.
4. Take the saved true positive probabilities of predicted class and device id captured from the network packet.
5. i) If the predicted class and device id captured from the network packet is different: It is an unknown device.
   ii) Else if same then if the predicted probability is x percentile of true positive probability for the class: It is known device.



```
signature 7 belongs to an unknown device.
signature 8 belongs to an unknown device.
signature 9 belongs to an unknown device.
signature 10 belongs to device 1
signature 11 belongs to device 1
signature 12 belongs to device 1
signature 21  may be a compromised device.
signature 22  may be a compromised device.
signature 23  may be a compromised device.
```

```
b8:27:eb:94:43:80 is a unknown device
50:5b:c2:d4:6b:7b is a known device
b8:27:eb:eb:fe:c6 is a compromised device
```

4. Fingerprinting