

Effective IoT Device Identification at the Edge

Coseners 2020

32nd Multi-Service Networks workshop (MSN 2020)

Roman Kolcun, Anna Maria Mandalari, Yiming Xie,
Hamed Haddadi (Imperial College London)
Diana Andreea Popescu, Vadim Safronov,
Richard Mortier (Cambridge University)
Poonam Yadav (University of York)

July 9, 2020

Motivation

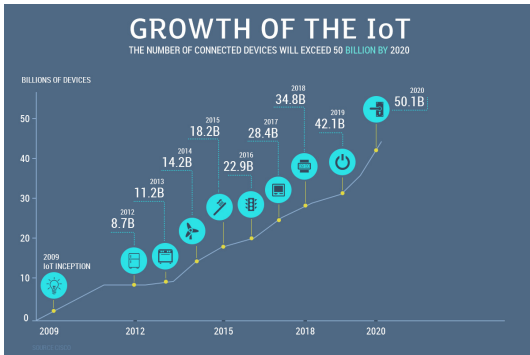


Figure: Number of IoT devices is rising

Motivation

```

In this howto I am explaining restoration of a full opcode
of the stripped ELF in ARM arch. With the method explained in
the MMD ELF Workshop. The skeleton will
help to define which function are those fcn.xxx & we have only
reverse the rest of value, then good C code will
be created beautifully. by @unixfreaxjp of #MalwareMustDie

// cracking a stripped ARM ELF malware

0x0000ccf8      2f0d00eb      bl fcn.000101bc
0x0000ccfc      010770e3      cmn r0, 1
0x0000cc00      0040a0a1      mov r4, r0
0x0000cc04      0a00000a      mov 0xc034
0x0000cc08      451aa0e3      mov r1, 0x45000
0x0000cc0c      0130a0e3      mov r3, 1
0x0000cc10      071c31e2      add r1, r1, 0x700
0x0000cc14      462a8de2      add r2, sp, 0x1180
0x0000cc18      015a8de2      add r5, sp, 0x1090
0x0000cc1c      121181e2      add r1, r1, 0x80000004
0x0000cc20      14282e2      add r2, r2, 0x11
0x0000cc24      94318de5      str r3, [r5, 0x194]
0x0000cc28      040d00eb      bl fcn.00010140
0x0000cc2c      040a00e1      mov r0, r4
0x0000cc30      e80c00eb      bl fcn.000100d8
: JMP XNEF from 0x0000cc04 (main)
-> 0x0000cc34      7c089fe5      ldr r0, [pc, 0x07fc]
0x0000cc38      db0c00eb      bl fcn.000100ac

// using skeleton to force rollback all function names
// MMD skeleton:
char *stripped-functions-mmd[] = {
    fcn.000101bc = open()
    fcn.00010140 = exec()
    fcn.000100d8 = close()
    fcn.000100ac = chdir()
    (void*)0
};

// restore the all value into its place ;) & CRACKZ!!
// source code reversed :
cmd1 = open("/dev/watchdog", 2, 0, 0);
if (cmd1 != -1) //if watchdog opened
{
    exec("/dev/watchdog", "-2147199228, 't', 1);
    close(v6);
    chdir("");
}

#MalwareMustDie! Linux/Mirai ELF ARM Malware Reversing
@unixfreaxjp, Aug 29 2016
    
```

Figure: Apparently a part of Mirai source code

Motivation

IoT vulnerability exploit

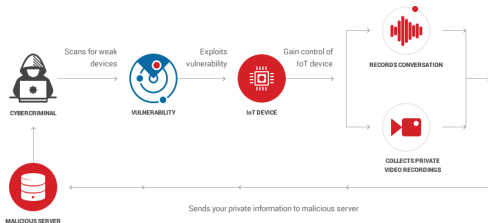


Figure: How vulnerabilities are exploited

Motivation

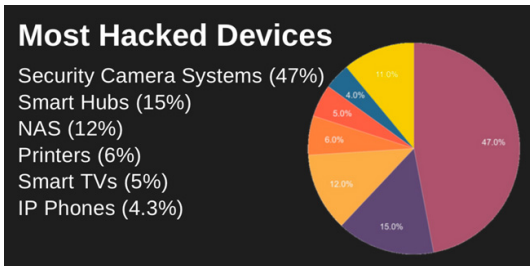


Figure: A pie chart

Motivation

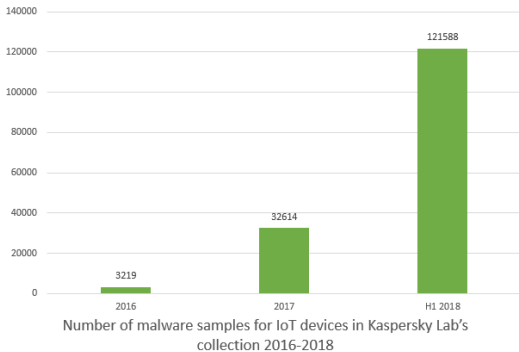


Figure: A bar chart

└ Motivation

- IoT are penetrating our households and the number is ever rising
- IoT are source of large number of security threats
- They would benefit from automated management
- This requires identification of devices
- The most natural way is to identify them by their network traffic at the home router

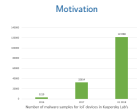


Figure: A bar chart

Research Question

- ▶ Which ML models are best suited for this task?
- ▶ Once trained, do these models stay accurate?
- ▶ Is it feasible to run inference of the models at the edge?
- ▶ If needed, can these models be trained at the edge?

Test-beds



- ▶ Large Test-bed - 43 devices
- ▶ Small Test-bed - 9 devices (a subset)

All devices were split into 6 categories: Surveillance, Media, Audio, Hub, Appliance, Home Automation

Datasets

Two types of data were collected from both test-beds:

- ▶ Idle, i.e. no interaction with test-beds (3 weeks)
- ▶ Active, i.e. automated interaction with test-beds (1 week)

Evaluation

- ▶ Five different types of models:
 - ▶ Fully Connected Neural Network
 - ▶ LSTM Network
 - ▶ 1D Convolutional Network
 - ▶ Random Forest Classifier
 - ▶ Decision Tree Classifier
- ▶ Four different groups of models:
 - ▶ One model for all devices
 - ▶ One model per device
 - ▶ One model for all categories
 - ▶ One model per category

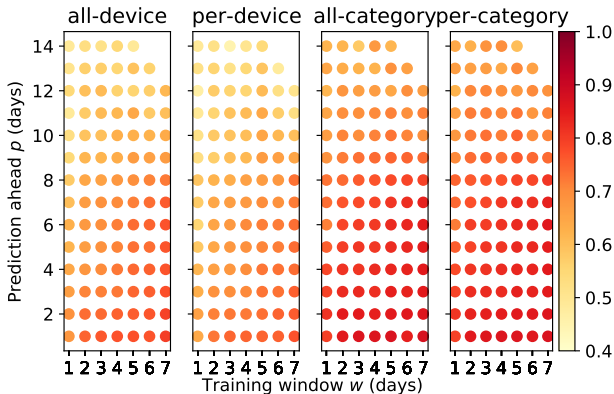
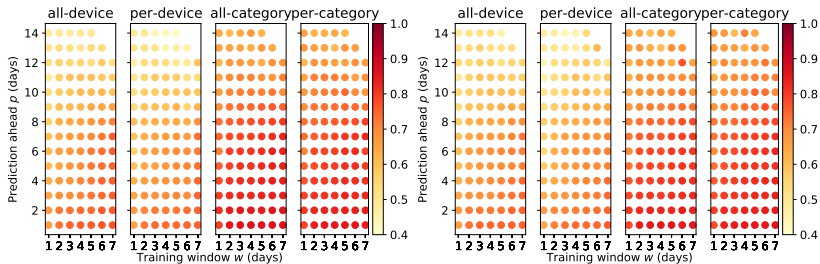
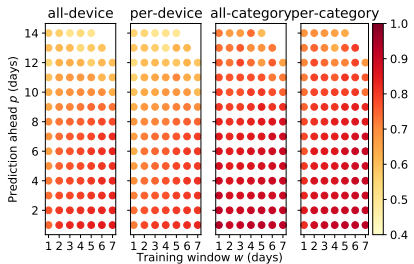


Figure: Fully Connected Network

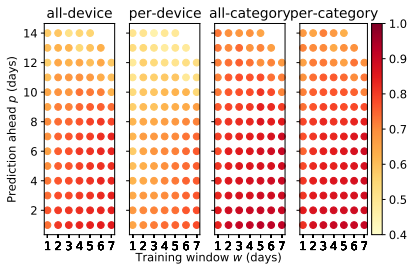


(a) Long short-term memory model

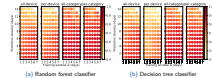
(b) 1D convolutional model



(a) Random forest classifier



(b) Decision tree classifier



- Models performance is rather similar
- The accuracy of the models decrease over time
- The longer the training time, the longer the model remains accurate
- None of the models can reliably classify devices more than 2 weeks ahead
- Retraining of models is necessary

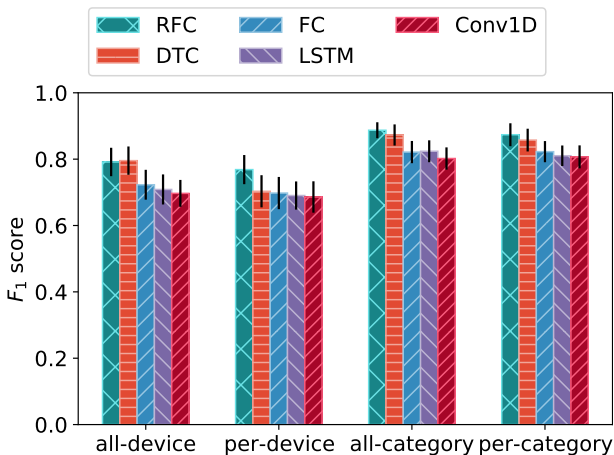


Figure: Average F_1 score of various models using a training window of various sizes (1-7) over the prediction of up to 7 days ahead.

Effective IoT Device Identification at the Edge

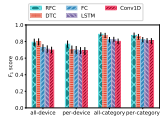
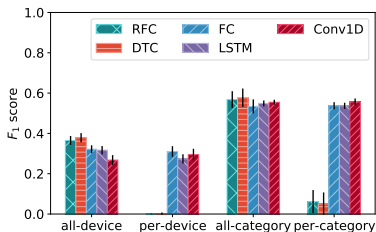
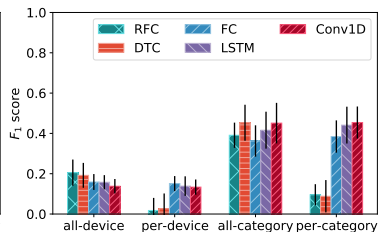


Figure: Average F_1 score of various models using a training window of various sizes (1-7) over the prediction of up to 7 days ahead.

- RFC & DTC models slightly outperform neural network based models
- Performance of neural network based models is virtually the same
- Device classification is less accurate than category classification
- Single multi-classification model outperforms multiple binary classification models



(a) Large Test-bed



(b) Small Test-bed

Figure: Average F_1 score of models trained on 7 day window of idle data and tested on active data of the large and the small test-bed.

Effective IoT Device Identification at the Edge

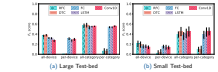
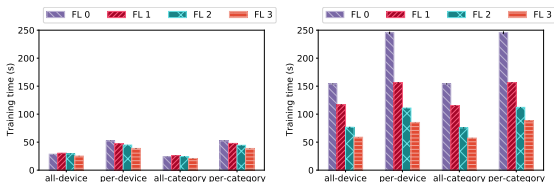


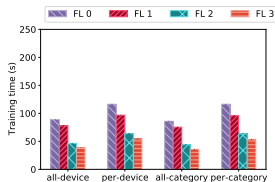
Figure: Average F_1 score of models trained on 7 day window of idle data and tested on active data of the large and the small test-bed.

- Models trained on an idle dataset are not accurate on active data
- Models trained on one test-bed are less accurate on the other test-bed
- Models updated with data from one test-bed increase accuracy on the same test-bed but have very small impact on the other test-bed
- Models need to be updated with local data



(a) Fully connected model

(b) LSTM model



(c) 1D conv. model

Figure: Average training time on RPi4 with different numbers of frozen layers (0, 1, 2 or 3 layers).

Effective IoT Device Identification at the Edge

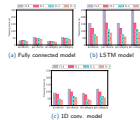


Figure: Average training time on RPi4 with different numbers of frozen layers (0, 1, 2 or 3 layers).

- Model retraining is feasible at the edge
- The improvement in training time largely depends on the type and the architecture of the neural network
- Layer freezing more than halves the training time for LSTM and Conv1D models
- Fully connected layer freezing reduces the training time modestly

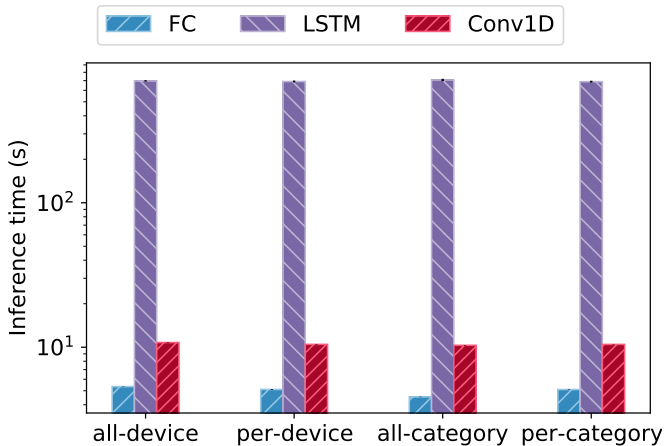


Figure: Average inference time on RPi4 of 100K samples using TensorFlow Lite.

Effective IoT Device Identification at the Edge

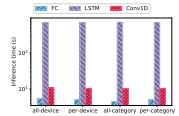


Figure: Average inference time on RPi4 of 100K samples using TensorFlow Lite.

- Results are similar across the all groups of models and model type
- The inference time for LSTM models is considerably larger
- Fully connected model is the fastest
- Inference time of 1D Convolutional model is double the time of FC model

Conclusion

- ▶ All models lose accuracy over time
- ▶ Models need to be updated with local data
- ▶ It is feasible to run model inference at the edge
- ▶ It is feasible to update the models at the edge

Is There A Way I Can Help?

Is There A Way I Can Help?

Would you like to know how chatty your IoT devices are?

Is There A Way I Can Help?

Would you like to know how chatty your IoT devices are?

Would you like to help researchers to collect data?

Is There A Way I Can Help?

Would you like to know how chatty your IoT devices are?

Would you like to help researchers to collect data? (or not)

Is There A Way I Can Help?

Would you like to know how chatty your IoT devices are?

Would you like to help researchers to collect data? (or not)

Good news: we will be running an experiment where you can install a Raspberry Pi 4 at your home which will act as a router for smart devices and send TCP/UDP headers (i.e. no payload) to our server.

Is There A Way I Can Help?

Would you like to know how chatty your IoT devices are?

Would you like to help researchers to collect data? (or not)

Good news: we will be running an experiment where you can install a Raspberry Pi 4 at your home which will act as a router for smart devices and send TCP/UDP headers (i.e. no payload) to our server.

Anyone interested should contact me at roman.kolcun@imperial.ac.uk or any of the aforementioned researchers.