Heterogeneous Network Inspection in IoT Environment with FPGA based Pre-Filter and CPU based LightGBM

Zhenguo Hu¹, Hirokazu Hasegawa², Yukiko Yamaguchi¹, Hajime Shimada¹

Nagoya University¹
National Institute of Informatics²

2023/9/28
Contents

- Background
- Heterogeneous Malicious Traffic Detection System Design
  - FPGA based Pre-Filter
  - Machine Learning based Traffic Detection
- Experiment & Evaluation
  - Evaluation on Training Stage
  - Evaluation on Inference Stage
- Conclusion
Contents

- Background
  - Heterogeneous Malicious Traffic Detection System Design
    - FPGA based Pre-Filter
    - Machine Learning based Traffic Detection

- Experiment & Evaluation
  - Evaluation on Training Stage
  - Evaluation on Inference Stage

- Conclusion
Nowadays, with the development of modern society, IoT has entered many aspects of our daily lives.

Both consumer IoT connections and enterprise IoT connections will increase in recent years.

IoT connections are predicted to experience the majority growth in the Asia Pacific region.

[1] IoT for Development: Use cases delivering impact, GSMA Intelligence
Cyber Attack

With the development of IoT technologies, IoT devices and connections will suffer many various malicious attacks on privacy and security.

- Bruteforce Attack
- DDoS Attack
- SQL Injection Attack
- ...

How to protect the network security and detect malicious traffic under IoT environment has become the common goal of researchers in the whole world.
NIDS is a hardware device or software application which is deployed to identify network threats.

- Malicious attacks against the IoT network are gradually increasing.
- NIDS can be deployed to execute complex malicious traffic detection.

In order to detect malicious traffic especially in IoT environment, it is important to build an effective defense system to prevent attacks.
Contents

- Background
- Heterogeneous Malicious Traffic Detection System Design
  - FPGA based Pre-Filter
  - Machine Learning based Traffic Detection
- Experiment & Evaluation
  - Evaluation on Training Stage
  - Evaluation on Inference Stage
- Conclusion
The heterogeneous malicious traffic detection system is designed to detect packet-level malicious traffic.

It mainly consists of two parts:
- FPGA based Pre-Filter
- Machine Learning based Traffic Detection
The FPGA based Pre-Filter is used to filter the truly malicious traffic by setting a blacklist in the FPGA board.

**Diagram:**
- Traffic → QSFP 28 → Ethernet Port
- Ethernet Port → CMAC Module
- CMAC Module → Bloom Filter
  - Decoder → Hash
  - 0 1 0 1
- Bloom Filter → TX Path
- TX Path → PCIe Module
- PCIe Module

**Steps:**
1. Ethernet Port
2. CMAC Module
3. Bloom Filter
4. PCIe Module
FPGA based Pre-Filter (Bloom Filter)

- Bloom filter is used as the IP blacklist implementation and to filter the “Source IPv4 Address”.
  - Cyclic Redundancy Check (CRC) hash function
  - $False Positive Probability = (1 - e^{-\frac{kn}{m}})^k$

**K=4, m/n=16, false positive probability = 0.239%**
Machine Learning based Traffic Detection

Machine learning based traffic detection leverages the data-driven insight ability of machine learning to analyze the malicious traffic on the CPU side.

Two modules are implemented to detect the attack behaviors:

- Traffic Capturer and Parser Module
- LightGBM Classifier
Feature Extraction

- We focus on extracting packet-level traffic features.
- We encode the label for Layer, Source IP Address and Destination IP Address to convert the category format into the number format.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4 Length</td>
<td>The length of an IPv4 packet</td>
</tr>
<tr>
<td>IPv4 ID</td>
<td>The identification of an IPv4 packet</td>
</tr>
<tr>
<td>IPv4 TTL</td>
<td>The time to live of an IPv4 packet</td>
</tr>
<tr>
<td>Layer</td>
<td>The type of protocol</td>
</tr>
<tr>
<td>Source Port</td>
<td>The source port</td>
</tr>
<tr>
<td>Destination Port</td>
<td>The destination port</td>
</tr>
<tr>
<td>Source IP Address</td>
<td>The source IP address</td>
</tr>
<tr>
<td>Destination IP Address</td>
<td>The destination IP address</td>
</tr>
</tbody>
</table>

LightGBM Classifier

- We train a LightGBM model in advance and then instantiate it as the classifier implementation.
- The extracted features are combined into a feature vector and sent to the LightGBM to execute the prediction.
Contents

1. Background

2. Heterogeneous Malicious Traffic Detection System Design
   - FPGA based Pre-Filter
   - Machine Learning based Traffic Detection

3. Experiment & Evaluation
   - Evaluation on Training Stage
   - Evaluation on Inference Stage

4. Conclusion
The experiment and evaluation dataset comes from Kitsune Mirai. It is presented in 2018 and it is captured from an IoT network, where the Mirai malware begins to infect other devices and scans for new victims network. It consists of 642,516 pieces of malicious data and 121,621 pieces of benign data. We select 80% (611,309) as the training set and 20% (152,828) as the test set.

<table>
<thead>
<tr>
<th>Number</th>
<th>Num of Malicious</th>
<th>Num of Benign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num of Train</td>
<td>642,516</td>
<td>121,621</td>
</tr>
<tr>
<td>Num of Test</td>
<td>611,309</td>
<td>152,828</td>
</tr>
</tbody>
</table>


## Evaluation Metric

- False Positive (FP), False Negative (FN), Accuracy (ACC), Precision, Recall, F1-score

### Evaluation Results

- We set four other models including Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest and Decision Tree as the comparisons.

- LightGBM achieves the highest evaluation in all the classifiers on ACC, Precision, Recall and F1-score.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>ACC</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>0.9104</td>
<td>0.9706</td>
<td>0.9214</td>
<td>0.9453</td>
</tr>
<tr>
<td>KNN</td>
<td>0.9494</td>
<td>0.9740</td>
<td>0.9656</td>
<td>0.9698</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.9525</td>
<td>0.9766</td>
<td>0.9668</td>
<td>0.9716</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>0.9556</td>
<td>0.9773</td>
<td>0.9697</td>
<td>0.9735</td>
</tr>
<tr>
<td>LightGBM</td>
<td>0.9589</td>
<td>0.9774</td>
<td>0.9736</td>
<td>0.9755</td>
</tr>
</tbody>
</table>

(a) Decision Tree  
(b) LightGBM
Experiment & Evaluation (Inference Stage)

Experiment Environment

- The assumed experimental IoT network
  - By adjusting the bloom array in the FPGA based pre-filter, we build four situations of irregular traffic which includes different types of source IPv4 addresses.

<table>
<thead>
<tr>
<th>Source IPv4 Address Range</th>
<th>Clients</th>
<th>Assumed Irregular Traffic filtered by FPGA based Pre-Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.168.2.108</td>
<td>30</td>
<td>Situation1</td>
</tr>
<tr>
<td>192.168.2.108 192.168.2.1</td>
<td></td>
<td>Situation2</td>
</tr>
<tr>
<td>192.168.2.108 192.168.2.1</td>
<td></td>
<td>Situation3</td>
</tr>
<tr>
<td>192.168.2.108 192.168.2.1 192.168.2.113</td>
<td></td>
<td>Situation4</td>
</tr>
</tbody>
</table>

- The experiment environment setup
  - Cisco TRex is used as the traffic generator to generate and send the traffic to the testing server.
  - On the testing server, the Xilinx Alveo U50 accelerator card is our FPGA platform.
Experiment & Evaluation (Inference Stage)

Evaluation Results

- **FPGA Resource Utilization**
  - The resource consumption of LUT, Register and BRAM Tile comes from the Xilinx Vivado Design Suite 2020.2.
  - From the result we can see, the filter module consumes 5,279 LUTs and 5,209 registers which occupies a reasonable resource consumption.

<table>
<thead>
<tr>
<th>Module Name</th>
<th>LUT</th>
<th>Register</th>
<th>BRAM Tile</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMAC Module</td>
<td>9,793</td>
<td>31,504</td>
<td>0</td>
</tr>
<tr>
<td>Filter Module</td>
<td>5,279</td>
<td>5,209</td>
<td>0</td>
</tr>
<tr>
<td>PCIe Module</td>
<td>79,576</td>
<td>84,544</td>
<td>94</td>
</tr>
<tr>
<td>Proportion</td>
<td>10.9%</td>
<td>7.0%</td>
<td>7.0%</td>
</tr>
</tbody>
</table>

- **Throughput**
  - We evaluate the throughput under various situations. Baseline indicates that there are no rules being enabled in the pre-filter.
  - The FPGA based pre-filter can block a portion of malicious traffic which reduces the burden on ML based traffic detection, and improve the detection performance.
Contents

- Background

- Heterogeneous Malicious Traffic Detection System Design
  - FPGA based Pre-Filter
  - Machine Learning based Traffic Detection

- Experiment & Evaluation
  - Evaluation on Training Stage
  - Evaluation on Inference Stage

- Conclusion
Conclusion

We present a realtime heterogeneous malicious traffic detection method using FPGA based pre-filter and machine learning based traffic detection especially in IoT environment.

We design an experiment to evaluate the proposed system and the results show that it has better performance with low FPGA resource usage and effective throughput improvement.

In the future, we will explore to add other models to the system to further improve the detection efficiency.
Thank you for your listening.