ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

DEPARTMENT

UNIVERSITY OF CALIFORNIA, IRVINE

NETWORKING RESEARCH GROUP
PROGRAMMING LANGUAGE RESEARCH GROUP

PINGPONG 1.0

DOCUMENTATION

Table of Contents

Getting Started

Please read the PingPong paper here or here to first understand the big picture of the system.

Code Download

Please download the source code here or use the following command on a command prompt with Git.

```
$git clone git://plrg.eecs.uci.edu/pingpong.git
```

PingPong can be opened on a code editor (the code was developed using IntelliJ and it was tested on IntelliJ IDEA 2018.3.2). To ease the running process, we created a set of scripts in the folder

```
$<absolute-path>/smart_home_traffic/Code/Projects/
PacketLevelSignatureExtractor/
```

Datasets Download

Please download the datasets here.

Then we need to unzip (using **tar**) the file by running the following command or use your favorite compression program.

```
$tar -xvzf PingPong.tar.gz <destination-folder>
```

Note:
The following command is to compress a folder using **tar**.

```
$tar -zcvf <file-name>.tar.gz <source-folder>
```

Data Collection

- The scripts for data collection can be found in

  ```
  $<absolute-path>/smart_home_traffic/automation/
  ```

- The script `browser.sh` was run on a MacBook to generate browsing traffic (as background traffic).

- The scripts `clicker*.sh` were used to trigger events from devices; the relevant commands with “`adb`” need to be uncommented to run the scripts. They are set up to generate 100 triggers (typically binary) with a 131-second interval in between triggers.

- Please pay attention that certain devices only exist in certain clicker script. At some point, we had to run the triggers in parallel for 2 devices at once so we put certain sets of adb commands in certain scripts (see `clickers1.sh` and `clickers2.sh`). Do not forget to open the necessary Android phone app before starting the script as the script will generate click events.

- To run the scripts please run the following (the command will save the generated timestamps into a file). We need to run “`adb devices`” to see the Android device ID of connected phones.

  ```
  $<absolute-path>/smart_home_traffic/automation/clicker.sh >
  <file-name>

  $<absolute-path>/smart_home_traffic/automation/clickers.sh
  <device-id> > <file-name>

  $<absolute-path>/smart_home_traffic/automation/clickers1.sh
  <device-id> > <file-name>

  $<absolute-path>/smart_home_traffic/automation/clickers2.sh
  <device-id> > <file-name>
  ```

- For now the triggering commands for the WeMo Insight plug are uncommented in the scripts. In general, if any triggering commands do not work, you can adjust the x and y coordinates in the `adb` commands (or even delete/add more commands). To get the right coordinates, please run “`adb shell getevent -l`” and click on the desired button on the phone app.

General Instructions

- These instructions are applicable to the scripts in `smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor`.

- To run PingPong to generate signatures for a certain device and event type, please uncomment the necessary "gradlew" line(s).

- There are 3 different triggering setups in the script: local-phone (labeled "LOCAL"), remote-phone (labeled "REMOTE"), IFTTT (labeled "IFTTT"). There are also separate sections labeled "`same-vendor`" and "`public-dataset`" in the scripts.

- Please replace <*sub-folder*> with one of the following: "`local-phone`", "`remote-phone`", "`ifttt`", "`same-vendor`", or "`public-dataset`" depending on which setup you want to extract signatures for (please study the scripts and see the directory structure in `PingPong/evaluation-datasets/`).

- We have the following scripts discussed in the next sections.

  - execute_layer2_smarthome_all_detection_results_analysis.sh
  - execute_layer2_smarthome_all_detection.sh
  - execute_layer2_unb_all_detection.sh
  - execute_layer2_unsw_all_detection.sh
  - execute_layer2_yourthings_all_detection.sh
  - execute_layer3_smarthome_all_detection_results_analysis.sh
  - execute_layer3_smarthome_all_detection.sh
  - execute_layer3_unb_all_detection.sh
  - execute_layer3_unsw_all_detection.sh
  - execute_layer3_yourthings_all_detection.sh
  - execute_signature_generation_imc_dataset.sh
  - execute_signature_generation.sh
  - execute_signature_validation_results_analysis.sh
  - execute_signature_validation.sh
  - execute_vpn_smarthome_all_detection.sh

- Please use the right script for each section (the scripts' names are fairly descriptive).

Signature Generation

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSi
  gnatureExtractor/execute_signature_generation.sh
  <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/
  <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standa
  lone/
  ```

- The signatures and cluster analyses files will then be generated in the following folders.

  ```
  <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standa
  lone/<device-name>/signatures
  <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standa
  lone/<device-name>/analyses
  ```

- **Note:** The output log file will also contain maximum signature duration values that we can use as the signature duration for signature validation and detection. For now, the duration is only used to constrain the Layer 2 detection since it does not enforce signature packets to appear consecutively.

- We also have a script that we used to extract signatures from a public dataset. This script can be found in

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSi
  gnatureExtractor/execute_signature_generation_imc_dataset.sh
  ```

- **Note:** We did not have signature validation scripts to validate signatures generated by this script because of the problems the dataset has. However, we still validated the generated signatures by excluding problematic PCAP files and assuming to have less number of events, e.g., if the total number of events is 40 and 20 PCAP files are problematic, we would just generate the signatures from only 20 PCAP files (20 events). If we could find clusters/sequences with 20 members then we validated the generated signatures.

Signature Validation

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

```
$<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor/execute_signature_validation.sh
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standalone/
```

- The results will then be generated in the following folders.

```
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standalone/<device-name>/
```

- Run on the command prompt the following script from the PingPong code directory to analyze the results.

```
$<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor/execute_signature_validation_results_analysis.sh
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standalone/
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standalone/
```

- The results will then be generated in the following folders.

```
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standalone/<device-name>/
```

Signature Detection - Layer 3

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

```
$<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor/execute_layer3_smarthome_all_detection.sh
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/
```

- The results will then be generated in the following folders.

```
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/<device-name>/
```

- Run on the command prompt the following script from the PingPong code directory to analyze the results.

```
$<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor/execute_layer3_smarthome_all_detection_results_analysis.sh
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/
```

- The results will then be generated in the following folders.

```
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/<device-name>/
```

Signature Detection - Layer 2

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

```
$<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor/execute_layer2_smarthome_all_detection.sh
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/
```

- The results will then be generated in the following folders.

```
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/<device-name>/
```

- Run on the command prompt the following script from the PingPong code directory to analyze the results.

```
$<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor/execute_layer2_smarthome_all_detection_results_analysis.sh
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/
<absolute-path>/PingPong/evaluation-datasets/<sub-folder>/smarthome/
```

- The results will then be generated in the following folders.

```
<absolute-path>/PingPong/evaluation-datasets/smarthome/<device-name>/
```

Negative Control Experiment

UNSW

Please first read the README file in
`<absolute-path>/PingPong/negative-datasets/UNSW/`

Layer 3

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSi
  gnatureExtractor/execute_layer3_unsw_all_detection.sh
  <absolute-path>/PingPong/negative-datasets/UNSW/
  <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standa
  lone/ <absolute-path>/PingPong/negative-datasets/UNSW/result/
  ```

- The results will then be generated in the following folders.

  ```
  <absolute-path>/PingPong/negative-datasets/UNSW/result/
  ```

Layer 2

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSi
  gnatureExtractor/execute_layer2_unsw_all_detection.sh
  <absolute-path>/PingPong/negative-datasets/UNSW/
  <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standa
  lone/ <absolute-path>/PingPong/negative-datasets/UNSW/result/
  ```

- The results will then be generated in the following folders.

  ```
  <absolute-path>/PingPong/negative-datasets/UNSW/result/
  ```

UNB

Please first read the README file in
`<absolute-path>/PingPong/negative-datasets/UNB/`

Layer 3

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor/execute_layer3_unb_all_detection.sh <absolute-path>/PingPong/negative-datasets/UNB/Monday-WorkingHours.pcap <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standalone/ <absolute-path>/PingPong/negative-datasets/UNB/result/
  ```

- The results will then be generated in the following folders.

  ```
  <absolute-path>/PingPong/negative-datasets/UNB/result/
  ```

Layer 2

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSignatureExtractor/execute_layer2_unb_all_detection.sh <absolute-path>/PingPong/negative-datasets/UNB/Monday-WorkingHours.pcap <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standalone/ <absolute-path>/PingPong/negative-datasets/UNB/result/
  ```

- The results will then be generated in the following folders.

  ```
  <absolute-path>/PingPong/negative-datasets/UNB/result/
  ```

YourThings

Please first read the README file in
`<absolute-path>/PingPong/negative-datasets/YourThings/`

Layer 3

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSi
  gnatureExtractor/execute_layer3_yourthings_all_detection.sh
  <absolute-path>/PingPong/negative-datasets/YourThings/2018/
  <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standa
  lone/
  <absolute-path>/PingPong/negative-datasets/YourThings/result/
  ```

- The results will then be generated in the following folders.

  ```
  <absolute-path>/PingPong/negative-datasets/YourThings/result/
  ```

Layer 2

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSi
  gnatureExtractor/execute_layer2_yourthings_all_detection.sh
  <absolute-path>/PingPong/negative-datasets/YourThings/2018/
  <absolute-path>/PingPong/evaluation-datasets/<sub-folder>/standa
  lone/
  <absolute-path>/PingPong/negative-datasets/YourThings/result/
  ```

- The results will then be generated in the following folders.

  ```
  <absolute-path>/PingPong/negative-datasets/YourThings/result/
  ```

Packet-padding Experiments

- The scripts used for the packet padding experiments (i.e., TLS and VPN-based paddings) can be found in

  ```
  $<absolute-path>/smart_home_traffic/packet-padding/
  ```

- Please generate the usage summary by running

  ```
  $python3 <script-name> -h
  ```

- Please note that the scripts were written and run using Python 3.6.

VPN-based STP

- Please note that the PCAP files for the VPN-based STP are regenerated using the existing PCAP files. We created a pseudo-VPN PCAP files by using the layer-2 detection and consider everything a single big stream of packets (no flow separation). The scripts used to generate the PCAP files for the VPN-based STP can be found in

  ```
  $<absolute-path>/smart_home_traffic/vpn/
  ```

- We used the "`tcpreplay`" command to generate the pseudo-VPN PCAP files (see `local-phone/smarthome/<device-name>/vpn/`) by injecting individual event files (see `local-phone/smarthome/<device-name>/event/`).

- Run on the command prompt the following script from the PingPong code directory. Please adjust <absolute-path> with the actual absolute path in your system.

  ```
  $<absolute-path>/smart_home_traffic/Code/Projects/PacketLevelSi
  gnatureExtractor/execute_vpn_smarthome_all_detection.sh
  <absolute-path>/PingPong/evaluation-datasets/local-phone/
  <absolute-path>/PingPong/evaluation-datasets/local-phone/smarth
  ome/
  ```

- The results will then be generated in the following folders.

  ```
  <absolute-path>/PingPong/evaluation-datasets/local-phone/smarth
  ome/<device-name>/
  ```

- For this experiment, we replayed and regenerated the PCAP files from the previous experiments and injected event packets. Therefore, all the packets got new timestamps and we do not have the list of event timestamps. To check the recall and false positive rates, we had to check them manually.

Contact Us

- Please contact us should you find any issues or have any questions.

- Our contact emails can be found in the paper (here or here).