Project phase 4

WPI CS4516 Spring 2019 D term

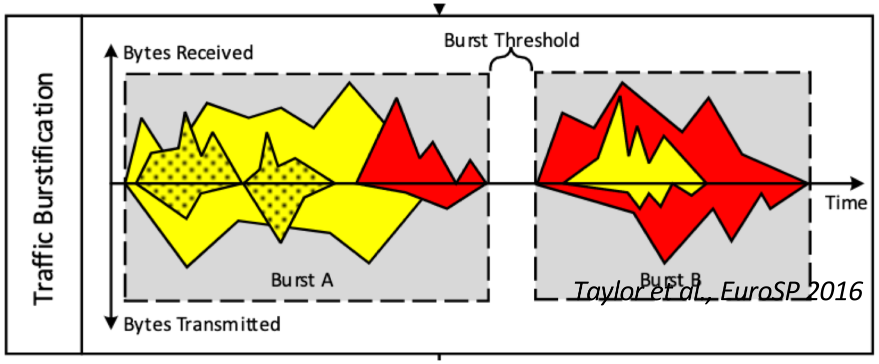
Instructor: Lorenzo De Carli (ldecarli@wpi.edu)

Phase 4 overview

- Implement an on-line classifier for encrypted traffic
  - Put phases 2 and 3 together
  - Phase 2: online traffic logging
  - Phase 3: offline traffic classification
  - Phase 4: online traffic classification
  - In a nutshell, incorporate the classification model you create for phase 3 into the traffic analysis tool you created for phase 2

Traffic analysis process

- 1 – Traffic burstification
  - Packet burst definition:



A packet burst includes all packets transmitted and received between the end of the previous burst, and a period of time when no packet is transmitted/received for 1s

  - In other words, segment the flow of captured packets in sections separated by 1s of silence

Traffic analysis process

- 2 – flow generation
  - Extract all flows from a given burst
  - Flows can be defined in different ways:
    - All packets in a burst part of the same TCP connection
    - All packets in a burst part of the same direction within a TCP connection
    - Different notions of source and destination (Android VM, connection originator, etc.)
    - Design choice does not matter as long as you describe it

Traffic analysis process

- 3 – flow classification
  - At the end of each burst, apply classifier you developed in phase 3 to each flow
  - If flow matches any action defined in phase 3 (e.g. ”start app X” etc.), label the flow

Deliverable specifications

- Upload a copy of the gateway VM with the following files in /home/tc:
  - A Python script named analyzeFlows. When executed, the script must print out a list of bursts, flows in every burst, and the label of each flow that originated a certain action (if any). Output example:

```
➢ ./analyzeFlows
Burst 1:
<timestamp> <src addr> <dst addr> <src port> <dst port> <proto>\
<#packets sent> <#packets rcvd> <#bytes send> <#bytes rcvd> <label>
```

Deliverable specifications/2

- Upload a copy of the gateway VM with the following files in /home/tc:
  - A file named readme.txt describing:
    - The specific definition of flow that you used
    - A brief description of the features that you used
    - Anything we need to know in order to run and grade your work
    - Anything else you want us to be aware of (limitations, problems, etc.)

Phase 4 evaluation

- What we will do:
  - Execute actions on the Android VM
  - Observe if:
    - Actions that are part of the evaluation set (ref. phase 3) are identified correctly (true positives evaluation)
    - Actions that are not part of the evaluation set do not get labeled (true negative evaluation)

Some suggestions

- TinyCore has better support for Python 2 than Python 3
  - Models exported from Python 2 won’t work with Python 3 (in general) and vice-versa
  - If you have a Python 2 model, you are probably good to go
  - If you have a Python 3 model, you will need to either re-create in Python 2, or install Python 3 in TinyCore (clunky, but not impossible)

Some suggestions/2

- The default TinyCore VM is configured with a fairly low amount of RAM
- If you find this create issues, feel free to change the configuration

Some suggestions/3

- Do not start at the last minute
- Most people got through the previous phases fine, but the combination VirtualBox+TinyCore is finnicky
  - If you run into issues due to specific configurations, Python versions, etc. we can try to help but only if you contact us early