**Steps to Use Anomaly Detection Program**

This document lists steps for using the anomaly detection program, including software that must be installed and other files to download.

1. Install the latest version of Python from [www.python.org](http://www.python.org). Use the default installation.
2. When Python is installed, it installs pip, which is used to install Python libraries. However, the version of pip that gets installed is an older version. To get the latest version of pip, get into Windows PowerShell. Type “pip --version” to see the current version installed, and use “python -m pip install --upgrade pip”.
3. If you want to develop code in an IDE (Interactive Development Environment) download and install PyCharm from here: <https://www.jetbrains.com/pycharm>
4. If you are not familiar with the OSI Seven-Layer Network Model, read this paper for a simple overview: <http://www.electronicdesign.com/what-s-difference-between/what-s-difference-between-osi-seven-layer-network-model-and-tcpip>
5. Install Wireshark from wireshark.org. WinPCap (winpcap.org) gets installed as part of Wireshark. WinPCap is the industry-standard tool for link-layer network access in Windows environments.
6. Download some sample PCAP files from <https://wiki.wireshark.org/SampleCaptures> and view them in Wireshark to become familiar with Wireshark and some header fields from layers 3 and 4 in the OSI model.
7. Read the whitepaper “Network Traffic Anomaly Detection in Embedded Systems” in the docs directory of the Anomaly-Detection repository to understand the basic anomaly detection algorithm.
8. Read the code in anomaly\_detect\_fake\_data.py to see how it implements the basic algorithm in the whitepaper. You can also run it in PyCharm or on the command line – it doesn’t take any command line arguments.
9. Install the Python library pypcapfile to be able to parse PCAP files from a Python script on Windows. Use “pip install pypcapfile” in Windows PowerShell to install it. The code will get installed under <python install directory>\Lib\site-packages\pcapfile.
10. Download the Python files ip.py, tcp.py, ethernet.py, udplite.py and vlan.py from the pypcapfile directory in the Anomaly-Detection repository. These 5 Python files supplement the Python library pypcapfile. The files ip.py, tcp.py and ethernet.py are edited versions of files that come with the library. The files udplite.py and vlan.py are new files for parsing packets of protocols UDPLite and VLAN, respectively. The edited files should replace the existing files in the library. File udplite.py should be placed in directory pcapfile\protocols\transport. File vlan.py should be placed in directory pcapfile\protocols\linklayer.
11. Download the Python scripts parse\_pcap.py and trim\_file.py from the python directory in the Anomaly-Detection repository. The script parse\_pcap.py parses PCAP files using the supplemented pypcapfile Python library and prints information about the first 15 packets in each PCAP file. The PCAP filenames are hardcoded into the script, but the comments in the script tell you where to find these files. The script trim\_file.py can be used to trim a large PCAP file so it can be parsed by parse\_pcap.py without running out of memory. It was used as follows to produce the PCAP file maccdc2012\_00000\_trim.pcap used by parse\_pcap.py: python trim\_file.py maccdc2012\_00000.pcap maccdc2012\_00000\_trim.pcap 10000000.
12. Run the script parse\_pcap.py on some PCAP files and compare its output with what you see in Wireshark for the same data packets. You can modify the script to change the names of PCAP files and the number of packets for which information is output. Keep in mind that the supplemented pypcapfile library currently only handles six protocols (Ethernet, VLAN, IP, TCP, UDP, UDPLite), so any other protocol found in a PCAP file won’t be parsed, and any layers under it won’t be parsed.