CSC/ECE 574 - Project 2*
Network Security
Assigned September 23, 2019; Due 11:59pm on October 9, 2019
Total Points: 100

Prof. William Enck

Goal: The goal of this project is to learn how to analyze network traces (PCAP files) with the intention to discover interesting characteristics and phenomena. The project is mostly open-ended (that is, there is no specific correct answer, and I do not have a working "solution."). In doing so, you will be required to be creative and learn new tools. Students will turn in project reports that describe both their findings and techniques used to determine the findings. Any custom scripts or tools built for the project will also be submitted.

Collaboration: Each student must submit their own solution. That said, students are encouraged to discuss tools and techniques with each other. The only rule is that the answer to the "findings" discussion question should be unique. One way to ensure uniqueness is for collaborating students to work on different pcap network trace files. Finally, if you discuss tools, techniques, or any insights with another student, **you must list their name** on your solution PDF in a clearly denoted "Acknowledgements" section. Such collaboration is highly encouraged and will not be viewed as a weakness of the solution.

Dataset: For this assignment, you will be analyzing publicly available pcap files from the Mid-Atlantic Collegiate Cyber Defense Competition. The CCDC is a defensive take on the concept of "Capture-the-Flag" competitions. The competitors all take a defensive posture. Each team is given a collection of hosts that contain vulnerabilities. The competition organizers act as a Red Team to exploit those vulnerabilities. The competitors must defend against attacks while maintaining service availability. You may choose any of the pcap files listed on the NETRESEC website (be sure to list the pcap filename on in your solution description): http://www.netresec.com/?page=MACCDC
Note: Some of these pcap files may consume a significant amount of RAM when opening in Wireshark. If you run into resource problems, consider splitting the pcap file with editcap.

Tools: You are permitted to use any network trace analysis tools that you would like, as long as you list the tools and the sources (i.e., URL) in the "Tools" section of your report. Custom tools and scripts must be turned in with the assignment. There are many great tools to analyze pcap files. You are encouraged to research pcap analysis tools on your own. However, here are several good starting places. You may find yourself using a combination of tools.

*Last revised on September 23, 2019.

- Wireshark is commonly known for sniffing network traffic, but it can also read pcap files. It has parsers for many different network protocols. Wireshark is a great way to look at network traffic and its contents. However, analysis with Wireshark is fairly manual.

- The Bro network security monitor is well known for intrusion detection. However, it is also a powerful network analysis framework that can operate on pcap files. The Bro documentation has many great resources for using Bro. Essentially, you run `bro -r file.pcap` and it generates a series of `.log` files. These files can be processed using a combination of the `bro-cut` command and `awk` (see these videos and these slides for examples).

- Scapy is a Python module for interactive packet manipulation; however, your Python program can also consume a pcap file. If you prefer to build your own custom tools, strongly consider using Scapy, as it provides a simple API for accessing different packet features and data.

- Pyshark is a Python wrapper for tshark, allowing python packet parsing using Wireshark dissectors.

- Dpkt is a Python module for fast, simple packet creation / parsing, with definitions for the basic TCP/IP protocols.

- CapTCP is a free and open source program for TCP analysis of PCAP files. It also integrates with GnuPlot.

- Gnuplot is a command line graphing utility. While Gnuplot has nothing to do with pcap analysis, it is the *de facto* graphing tool used by network security researchers. Strongly consider using Gnuplot to create your graphs. It allows you to write a graph template and reference data files that are generated by your custom scripts and tools. Pro tip: you can even automate the creation of Gnuplot files as part of your analysis framework.

Inspiration: If you are having trouble getting started, the following articles might be helpful.

- "Finding a needle in a PCAP" by Shiva Persaud - The author shares how he deconstructed traffic and constructed visualization to correlate a bigger perspective on what was happening: https://blogs.cisco.com/security/finding-a-needle-in-a-pcap

- "The Inside Story Behind MS08-067" by John Lambert - A great article from the Microsoft SRT that used their error-reporting telemetry to discover the exploit behind Conficker: https://blogs.technet.microsoft.com/johnla/2015/09/26/the-inside-story-behind-ms08-067/

Grading

Project 2 is worth 100 points. Each question will have its own grading requirements shown in its grading table. Your appendex can be used to hold additional data tables that are too big. At a minimum, the appendex should contain the filename and download link of your network pcap file. All figures and tables should have a caption with a label of Figure X. or Table X. These labels are really helpful to use during discussion as a reference.

Requirements for each question may or may not have the following subsections: Tools, Experimental, Data, and Discussion. The only sections that are required for each individual question is described in each question's own grading table.

- Tools - a list of tools or custom scripts used to answer the respective question.
- Experimental - brief description of your methodology to reproduce your results.
- Data - contains your figures and tables.
- Discussion - a discussion of your data, results, and interesting findings.

Section	**Points**
Question 1	23
Question 2	23
Question 3	23
Question 4	23
Acknowledgement	4
Appendex	4

Table 1: Point value for each section.

Question 1: Network Topology Reconstruction {23 points}

The CCDC consists of a Red Team, multiple competing teams, and supporting network infrastructure. Extract as much information as possible about the network topology from the pcap file. At bare minimum, you should identify the IP address of all of the hosts. Next, classify the IP addresses into "Red Team," "Competitors" (clustering into teams if possible), "Service Requests," "Infrastructure," and "Unknown." You may also be able to derive a partial network topology. While switches do not modify packets, routers decrement the TTL field in the IP header. If you can derive a partial topology, present it as an undirected graph (consider using Graphviz).

Grading

Section	**Points**
Tools	2
Experimental	4
Data	17

Table 2: Requirements for Question 1.

Question 2: Network Traffic Graphs {23 points}

Application layer protocols are frequently identifiable by the transport layer port number used to connect to the server. Use this port information, along with the packet timestamps, to reconstruct a network traffic graph showing the frequency of packets per time interval. For example, create a line-graph with the x-axis as 1-minute (second) intervals and the y-axis is the number of packets of that type that occurred in the 1-minute (second) interval. The graph could have one line for each major protocol and a "catch-all" category for "Other" traffic. Be creative.

Hint: Running `bro -r file.pcap` produces log files that are already split by major protocols. Therefore, Bro may be a good starting point for this question, even if you use other tools for the other questions.

Grading

Section	**Points**
Tools	2
Experimental	4
Data	17

Table 3: Requirements for Question 2.

Question 3: Attack Identification and Extraction {23 points}

Use the graph in Question 2 to identify a phenomena that appears to be an attack. Extract the attack payload (e.g., using Wireshark) and investigate what occurred. Provide a description of the attack, the target service, and any CVEs that may correspond to the vulnerability. Also provide a partial attack payload in the report if feasible.

Grading

Section	**Points**
Tools	2
Experimental	4
Discussion	17

Table 4: Requirements for Question 3.

Question 4: Interesting Findings {23 points}

Continue your investigation of the network trace. Find something interesting. Report what you found and how you found it. Examples of interesting findings include attack trends (e.g., frequencies and durations of specific types of attacks), and defense postures (e.g., certain teams were faster at defending against certain attacks). Be creative. Use graphs whenever possible to help explain your findings.

Grading

Section	**Points**
Tools	2
Experimental	4
Data	7
Discussion	10

Table 5: Requirements for Question 4.

Submission Instructions

Submit your solution as two separate files using WolfWare (one `.pdf`, one `.tar.gz`). To upload your assignment, navigate to the CSC574 course. Use the "Project 2" assignment under "Mini-Projects."

The first file should be a single PDF document with your report. **Writeups submitted in Word, PowerPoint, Corel, RTF, Pages, and other non-PDF or ASCII formats will not be accepted.** Consider using LaTeX to format your homework solutions. (For a good primer on LaTeX, see the Not So Short Introduction to LATEX.) The second file should be a tarball (`.tar.gz`) or Zip (`.zip`) of any custom tools or attack captures that are relevant to your report.

Please post questions (especially requests for clarification) about this homework to Piazza.