ECEN 4133 March 2, 2021
Computer Security Fundamentals Project 3: Network Security

Project 3: Network Security

This project is due on **Thursday, March 18** at **11:59 p.m.** and counts for 8% of your course grade. Late work will not be accepted after 24 hours past the deadline. If you have a conflict due to travel, interviews, etc., please plan accordingly and turn in your project early.

You may work individually, or in **teams of two** and submit one project per team. Please find a partner as soon as possible.

The code and other answers your group submits must be entirely your own work, and you are bound by the Honor Code. You may consult with other students about the conceptualization of the project and the meaning of the questions, but you may not look at any part of someone else's solution or collaborate with anyone outside your group. You may consult published references, provided that you appropriately cite them (e.g., with program comments), as you would in an academic paper.

Solutions must be submitted electronically via Canvas, following the submission checklist below.

Introduction

This project will introduce you to common network protocols, the basics behind analyzing network traces from both offensive and defensive perspectives, and several local network attacks.

Objectives

- Gain exposure to core network protocols and concepts.
- Understand offensive techniques used to attack local network traffic.
- Learn to apply manual and automated traffic analysis to detect security problems.

Part 1. Network Attacks

In this part of the project, you will experiment with network attacks by man-in-the-middling an HTTP connection to a website we control, and replacing some of its content.

Setup

This part can **optionally** use a VM, if you need it. You are allowed to write a script and run it on your own computer, but if you run into trouble installing scapy/pcap libraries, try the VM.

(optional) VM setup

1. Download VirtualBox from https://www.virtualbox.org/ and install it on your computer. VirtualBox runs on Windows, Linux, and Mac OS.
2. Get the VM file at https://file.ecen4133.org/4133-vm.ova. This file is 3 GB, so we recommend downloading it from campus.
3. Launch VirtualBox and select File ▷ Import Appliance to add the VM.
4. Start the VM. There is a user named `ubuntu` with password `ubuntu`.
5. `cd project3`
6. Run `./getkey.py` to see that it outputs a key.
7. In this project, you will edit `./attack.py` to attack the output of `./getkey.py`.

Attacking

You can download getkey.py and attack.py from here: https://ecen4133.org/static/proj3/getkey.py https://ecen4133.org/static/proj3/attack.py

We have set up the website http://freeaeskey.xyz/, which is a website that provides "random" AES-256 keys for free to anyone who visits. Professor Vuln has decided to use this website for encrypting his research. To do this, he has created a program that first fetches a fresh key from freeaeskey.xyz, and uses it to encrypt the private data. Your goal is to get Professor Vuln to encrypt the secret research under a key known to you.
To do this, you are able to get a program to run on Professor Vuln's network. For the purposes of this assignment, you will run your program as root on the same machine that Professor Vuln uses to download his key (i.e. the provided VM). Your task is to edit the `./attack.py` Python program to watch for requests to freeaeskey.xyz, and replace the key provided with one known to you:

```
4d6167696320576f7264733a2053717565616d697368204f7373696672616765
```

The rest of the web page should remain un-modified to avoid suspicion. When Professor Vuln runs the `./getkey.py` script, it should output this key every time. You are not allowed to modify the `getkey.py` script.
Your script will run as root, and any other users on the same machine that visit freeaeskey.xyz while it is running should receive this injected key. We will grade this project using the same VM given to you.

You are welcome to use any of the following libraries. If you believe you need additional ones, please ask on Piazza.
https://pypi.python.org/pypi/scapy
https://pypi.python.org/pypi/dpkt
https://pypi.python.org/pypi/dnet

Bonus: Attack HTTPS [Extra credit]
Professor Vuln has realized it is unwise to download keys over HTTP, and has switched to using HTTPS to download his keys, from https://freeaeskey.xyz. Make a new script (`attack_https.py` that caries out the same attack as before against HTTPS, this time fooling `./getkey-secure.py`. (Hint: what is secure about getkey-secure?)

What to submit Submit a Python script named `attack.py` that performs the attack when run as root on the local machine. For the (optional) bonus, submit `attack_https.py` as well.

Part 2. Anomaly Detection

In this part, you will programmatically analyze trace data to detect suspicious behavior. Specifically, you will be attempting to identify port scanning.

Port scanning is a technique used to find network hosts that have services listening on one or more target ports. It can be used offensively to locate vulnerable systems in preparation for an attack, or defensively for research or network administration. In one port scan technique, known as a `SYN` scan, the scanner sends TCP `SYN` packets (the first packet in the TCP handshake) and watches for hosts that respond with `SYN+ACK` packets (the second handshake step).

Since most hosts are not prepared to receive connections on any given port, typically, during a port scan, a much smaller number of hosts will respond with `SYN+ACK` packets than originally received `SYN` packets. By observing this effect in a packet trace, you can identify source addresses that may be attempting a port scan.

Your task is to develop a Python program that analyzes a PCAP file in order to detect possible `SYN` scans. You should use a library for packet manipulation and dissection, such as `scapy`. To learn about `scapy`, visit https://scapy.readthedocs.io/en/latest/usage.html.

Your program will take one argument, the name of the PCAP file to be analyzed, e.g.:

```
  python3 detector.py capture.pcap
```

The output should be the set of IP addresses (one per line) that sent more than 3 times as many `SYN` packets as the number of `SYN+ACK` packets they received. Your program should silently ignore packets that are malformed or that are not using Ethernet, IP, and TCP.

A sample PCAP file captured from a real network can be downloaded at https://file.ecen4133.org/proj3.pcap. (You can examine the packets manually by opening this file in Wireshark.) For this input, your program's output should be these lines, in any order:

```
128.3.23.2
128.3.23.5
128.3.23.117
128.3.23.158
128.3.164.248
128.3.164.249
```

What to submit Submit a Python program that accomplishes the task specified above, as a file named `detector.py`. You should assume that `scapy` 2.4 is available, and you may use standard Python system libraries, but your program should otherwise be self-contained. We will grade your detector using a variety of different PCAP files.

Submission Checklist

Upload to Moodle a gzipped tarball (`.tgz`) named `project3.`*`identikey1`*`.`*`identikey2`*`.tgz`. The tarball should contain only the files below:

Part 1: Network Attacks

`attack.py`	A Python script that caries out the attack specified in Part 1.
`attack_https.py*`	A Python script that does the HTTPS attack (extra credit)

Part 2: Anomaly Detection

`detector.py`	Your Python program for `SYN` scan detection.