Project 3: Network Security

This project is due on **Friday, March 11** at **6 p.m.** and counts for 8% of your course grade. Late submissions will be penalized by 10% plus an additional 10% every 5 hours until received. Late work will not be accepted after 20.5 hours past the deadline. If you have a conflict due to travel, interviews, etc., please plan accordingly and turn in your project early.

This is a group project; you will work in **teams of two** and submit one project per team. Please find a partner as soon as possible. If have trouble forming a team, post to Piazza's partner search forum. The final exam will cover project material, so you and your partner should collaborate on each part.

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 CTools and Gradescope, 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.
- Learn to apply manual and automated traffic analysis to detect security problems.
- Understand offensive techniques used to attack local network traffic.

Read this First

This project asks you to perform attacks, with our permission, against a target network that we are providing for this purpose. Attempting the same kinds of attacks against other networks without authorization is prohibited by law and university policies and may result in *fines, expulsion, and jail time*. **You must not attack any network without authorization!** There are also severe legal consequences for unauthorized interception of network data under the Electronic Communications Privacy Act and other statutes. Per the course ethics policy, you are required to respect the privacy and property rights of others at all times, *or else you will fail the course.* See "Ethics, Law, and University Policies" on the course website.

Part 1. Exploring Network Traces

Security analysts and attackers both frequently study network traffic to search for vulnerabilities and to characterize network behavior. In this section, you will examine a network packet trace (commonly called a "pcap") that we recorded on a sample network we set up for this assignment. You will search for specific vulnerable behaviors and extract relevant details using the Wireshark network analyzer, which is available at https://www.wireshark.org.

Download the pcap from https://www.eecs.umich.edu/courses/eecs388/static/project3/part1.pcap, and examine it using Wireshark. Familiarize yourself with Wireshark's features. and try exploring the various options for filtering and for reconstructing data streams.

Concisely answer the questions below. Each response should require at most 2–3 sentences.

1. Multiple devices are connected to the local network. What are their MAC and IP addresses?

2. What type of network does this appear to be (e.g., a large corporation, an ISP backbone, etc.)? Point to evidence from the trace that supports this.

3. One of the clients connects to an FTP server during the trace.

   (a) What is the DNS hostname of the server it connects to?

   (b) Is the connection using Active or Passive FTP?

   (c) Based on the packet capture, what's one major vulnerability of the FTP protocol?

   (d) Name at least two network protocols that can be used in place of FTP to provide secure file transfer.

4. The trace shows that at least one of the clients makes HTTPS connections to sites other than Facebook. Pick one of these connections and answer the following:

   (a) What is the domain name of the site the client is connecting to?

   (b) Is there any way the HTTPS server can protect against the leak of information in (a)?

   (c) During the TLS handshake, the client provides a list of supported cipher suites. List the cipher suites and name the crypto algorithms used for each.

   (d) Are any of these cipher suites worrisome from a security or privacy perspective? Why?

   (e) What cipher suite does the server choose for the connection?

5. One of the clients makes a number of requests to Facebook.

   (a) Even though logins are processed over HTTPS, what is insecure about the way the browser is authenticated to Facebook?

   (b) How would this let an attacker impersonate the user on Facebook?

   (c) How can users protect themselves against this type of attack?

   (d) What did the user do while on the Facebook site?

What to submit Submit a PDF named `pcap.pdf` containing your answers. Make sure each answer is formatted as a single line, and that the file you submit is in PDF format. Format your file using this template:

```
# Question 1

1. [Answer ...]

# Question 2

2. [Answer ...]

# Question 3

3a. [Answer ...]

3b. [Answer ...]

3c. [Answer ...]

3d. [Answer ...]

# Question 4

4a. [Answer ...]

4b. [Answer ...]

4c. [Answer ...]

4d. [Answer ...]

4e. [Answer ...]

# Question 5

5a. [Answer ...]

5b. [Answer ...]

5c. [Answer ...]

5d. [Answer ...]
```

Part 2. Anomaly Detection

In Part 1, you manually explored a network trace. Now, you will programmatically analyze a pcap file 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 kind of 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: We suggest `dpkt`. It is available in most package repositories. You can find more information about `dpkt` at https://github.com/kbandla/dpkt and view documentation by running `pydoc dpkt`, `pydoc dpkt.ip`, etc.; there's also a helpful tutorial here: https://jon.oberheide.org/blog/2008/10/15/dpkt-tutorial-2-parsing-a-pcap-file/.

Your program will take the path of the pcap file to be analyzed as a command-line parameter, e.g.:

```
python2.7 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 ftp://ftp.bro-ids.org/enterprise-traces/hdr-traces05/lbl-internal.20041004-1305.port002.dump.anon. (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 `dpkt` 1.8 and `scapy` 2.2 are 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.

Part 3. Penetration Test Pre-Engagement Agreement

What's going on?

The fictional company MegaCorp has contracted with the University of Michigan to provide Penetration Testing services to it in exchange for free hugs and cat videos. The University has nominated this semester's EECS 388 class to provide a thorough penetration test of its networks and exposed systems. What follows is the Pen Test Pre-Engagement Agreement, which covers goals, scope, compensation, and authorization to begin the Pen Test. You must agree to this document in writing before you begin.

Introduction

MegaCorp has recently set up a remote office in the Bob and Betty Beyster (BBB) Building for its employees to work at. MegaCorp is concerned that its remote office may be more vulnerable than its headquarters since it uses a wireless network to provide networking access to its remote employees.
MegaCorp has shared a brief description of its infrastructure with us, as shown in Figure 1.

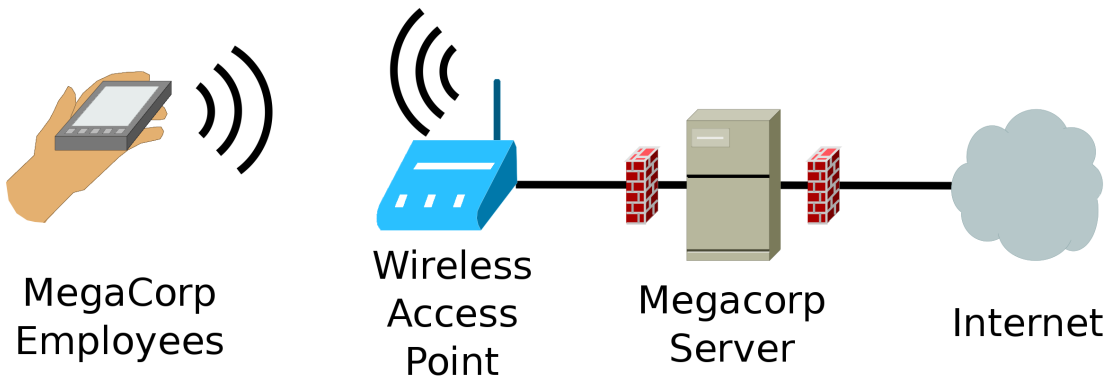

Figure 1: Infrastructure overview of MegaCorp's remote office.

MegaCorp employees generally connect to the **EECS388_MegaCorp** wireless network using WPA2-PSK security settings. From there, they can access the MegaCorp server, which allows MegaCorp employees to log in with their *UM Kerberos credentials* and gain access to company proprietary information.
Your objective is to test the real-world security of MegaCorp's networks and systems. In comparison with other security audits such as automated scanning or Vulnerability Assessments, in this Penetration Test you will be authorized to break in to MegaCorp's systems and explore any vulnerabilities you may find, subject to the Rules of Engagement in Section . As in a real-world pen test, you will be expected to use your ingenuity to discover clues and techniques for meeting your objectives.

Deliverables

This contract stipulates that by **Friday, March 11**, close of business (6:00PM), your pen test team will submit to GradeScope a penetration test report designed for a technical audience containing the following sections:

- Overview - A 2-3 sentence description of the objective of the pen test.
- Methodology - A 1-2 paragraph description of the work you performed.
- Findings - A description of any findings you may have made. Specifically, include details about the following:
  - Hostnames of any machines you gain access to over the course of the pen test.
  - Any encryption keys for networks you gain access to.
  - Any username/passwords you are able to obtain (not including your own).
- Remediation - A 1-2 paragraph description of what MegaCorp must do to secure its remote office. Be sure to specifically address each of your findings.

This pen test report outline is loosely based on the SANS outline [**sans**].

Rules of Engagement

There will be certain systems and networks that are *in scope* for this project. Everything else should be considered *out of scope*. If you have any questions about what is in or out of scope for this project, make sure you get clarification from one of the course instructors or TAs *before* you act.

Specifically, the following things are **in scope** for this pen test:
Hint: These are all things you should be doing.

- Capturing network traffic on the MegaCorp wired and wireless networks. Note: *eth0* on the MegaCorp server is a UM network and out of scope. *eth1* is a MegaCorp network and in scope.
- Using automated network scanning tools on the MegaCorp network.
- Connecting to the MegaCorp wireless network.
- Logging in to MegaCorp systems with your own credentials.
- Logging in to MegaCorp systems with MegaCorp employee credentials.
- Sending wireless attack packets to the MegaCorp access point (BSSID: 60:e3:27:4a:c4:bc) or MegaCorp wireless client (MAC: 00:0f:60:08:10:a5).

Let's go through a few examples of activities that are **out of scope** for this project:

- Sending wireless attack packets to any other users or devices besides those designated as in scope. **Do not deauth any other users or networks!** This would be illegal.
- Network scanning the MegaCorp server from the UM network. (From the wireless network, it's fine.)
- In general, actions that cause other users difficulties or interfere with the project infrastructure (Denial of Service) are prohibited.
- Attempting to elevate your shell privileges on the MegaCorp server. Use it as you would the CAEN computers.
- Anything else not specifically designated as in scope. If you're unsure, please ask for help.

A note about cheating: There may be backdoors you discover along the way. If these are shared with you before you get a chance to discover them for yourself, DO NOT USE THEM. We will be auditing the progress of each team as you complete the pen test and if we see you skip steps, we will consider this cheating and grounds for failing the course. If/when you discover a backdoor on your own, you may use it. If you have any questions about whether you may use a given backdoor, post a private question to the instructors on Piazza *before* you use it.

Compensation

This part is worth 3% of your course grade, of the 8% this project is worth, and will itself be graded out of 60 points. It will be graded as follows:

- 30 pts. Complete pen test report with all sections as described.
  - 3 pts. Overview.
  - 10 pts. Methodology.
  - 2 pts. Findings, not including points for specific findings.
  - 15 pts. Remediation.
- 10 pts each (maximum 3). Notable findings as described in the findings section.

All deserved points will be awarded after delivery of the report and subsequent processing by the MegaCorp security team.

Authorization

This document authorizes you, subject to the terms and conditions herein, to begin the Penetration Test for MegaCorp on behalf of the University of Michigan's EECS388 class. To accept this agreement and begin, you must email your acceptance ("I accept the EECS388 Pen Testing Agreement") to *eecs388-staff@umich.edu* along with the maximum number of years in **jail** that you could face under 18 USC § 2511 for intercepting traffic on an encrypted wifi network without permission.
Note: Both team members must email us their acceptance.

You **must** email us your acceptance before you begin.

Click here for a mailto: link with prepopulated fields.

Contact Information

General project questions: Post to Piazza. We encourage giving each other help, but please do not post spoilers (detailed step-by-step instructions).
Questions involving potential rule-breaking: Email us at eecs388-staff@umich.edu

Misc. things you will need

- This project is meant to be accompanied by discussion slides detailing equipment setup and several networking tools and techniques. You will need these to complete the project.
- Wireless network attacks are often very dependent on the wireless hardware you use. We strongly recommend you use an Alfa wireless adapter (Amazon). You can borrow one from us. We have a pinned piazza post about when and where you can borrow one.
- Once you crack the wireless network, you shouldn't need the Alfa wireless adapter anymore. Please return it so other groups can use it.
- Once you reach the MegaCorp server, you will discover you no longer need to use the MegaCorp wireless network at all. Please consider disconnecting from it to reduce wireless congestion for your classmates.
- Capturing network traffic is much easier on a wired network than a wireless one. Unless you need to capture network traffic wirelessly (say, to capture wireless authentication packets), we recommend using a wired interface.

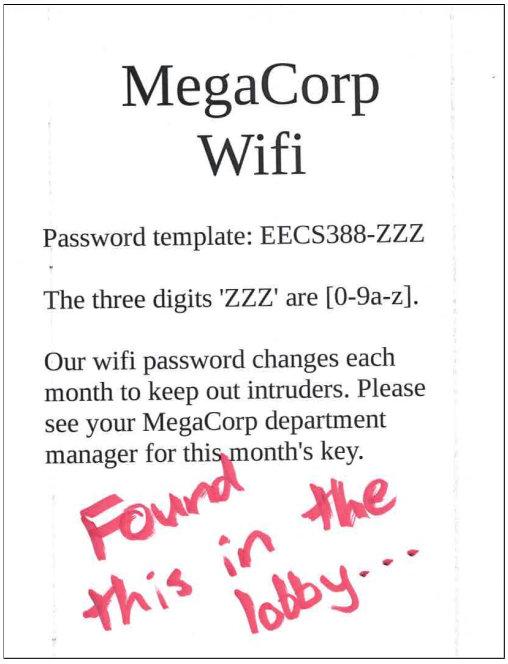

Figure 2: A flyer that was found in the BBB lobby. Looks important.

Tool List

What follows is a partial list of tools that may be helpful for this project. This list is not complete. You will need to use tools besides the ones listed here.

- **man**: The manual. Use this commmand to read the manual for other commands, including their options.
- Aircrack suite:
  - **aircrack-ng**: Cracks wireless passwords. Generally requires a network traffic capture.

  - **aireplay-ng**: Injects wireless packets into the air. Can reply captured packets and deauthenticate (deauth) other clients.
  - **airodump-ng**: Dumps, or captures, wireless network traffic. Can filter packets from specific targets. Can pin wireless adapter to specific channel.

- **nmap**: Network exploration tool and port scanner. Can scan network to find hosts, find open ports, even detect software versions in come cases.
- **tcpdump**: Network traffic analysis tool. Can capture traffic and save to a file. Can view traffic in real time. The -A and/or -w options may be helpful for this project.
- **wireshark**: Graphical network traffic analysis tool. Network traffic captured with tcpdump can be viewed with wireshark.
- **ssh**: Login to servers remotely.
- **scp**: Secure copy. Uses the ssh protocol to transfer files between hosts.

Submission Checklist

Submit to Gradescope the following files as answers to Parts 1 and 3. Submit to CTools the Python file for Part 2. Make sure Parts 1 and 3 render correctly as PDFs on Gradescope, as we cannot grade them if they do not render.
We will download and autograde your Python program from Part 2.

Part 1: Exploring Network Traces

`pcap.pdf`	A PDF containing your answers to the questions in Part 1.

Part 2: Anomaly Detection

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

Part 3: Penetration Test

`report.pdf`	A PDF with the contents specified in Part 3.