Project 3: Network Security

This project is split into two parts, with the first checkpoint due on **Wednesday, October 14** at **6:00pm** and the second checkpoint due on **Friday, October 23** at **6:00pm**. The first checkpoint is worth 2% of your total grade, and the second checkpoint is worth 10%. We strongly recommend that you get started early. Each semester everyone will be given ONE late extension that allows you to turn in up to one assignment up to 24 hours after the due date. Extensions are not automatic. So, if you want to use your late extension, you MUST send an e-mail to **ece422-staff@illinois.edu**. Late work will not be accepted after 24 hours past the due date.

This is a group project; you SHOULD work in **teams of two** and if you are in teams of two, you MUST 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. Some hardware does not work with some of the tools needed for this project such as Aircrack (Checkpoint 2). Build your teams such that at least one member of the team can run the required tools.

The code and other answers your group submits must be entirely your own work, and you are bound by the Student Code. You MAY consult with other students about the conceptualization of the project and the meaning of the questions, but you MUST 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 in any one of the group member's svn directory, following the submission checklist given at the end of each checkpoint. Details on the filename and submission guideline is listed at the end of the document.

"You can't defend. You can't prevent. The only thing you can do is detect and respond."
– Bruce Schneier

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.

Guidelines

- You SHOULD work in a group of 2.
- Your answers may or may not be the same as your classmates'.
- All the necessary files to start the project will given under the folder called "mp3" in your SVN directory. We've also generated some empty files for you to submit your answers in. You MUST submit your answers in the provided files; we will only grade what's there!

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.

3.1 Checkpoint 1 (20 points)

3.1.1 Exploring Network Traces (15 points)

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 packet trace from 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 (`http://www.wireshark.org`).

Get the network trace from `https://subversion.ews.illinois.edu/svn/fa15-cs461/_shared/mp3/3_1.pcap` and examine it using Wireshark. Provide concise answers to the following questions using submission format.

3.1.1.1 MAC, IP address (5 points)

Multiple hosts sent packets on the local network.

1. What are their MAC addresses?
2. What are their IP addresses?

What to submit: Submit a text file named `3.1.1.1_mac.txt` that contains the MAC addresses of hosts, and a text file named `3.1.1.1_ip.txt` contains the hosts' IP addresses. Write one address per line in the same order for both MAC address and IP address. Refer to solution format and example under *Checkpoint 1 Submission Checklist*.

3.1.1.2 FTP server (5 points)

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

1. What is the DNS hostname of the server it connects to?
2. Is the connection using Active or Passive FTP?

What to submit: Submit a text file named `3.1.1.2_dns.txt` containing DNS hostname, and a text file named `3.1.1.2_ftp.txt` containing whether it is Active or Passive FTP.

3.1.1.3 HTTPS connection (5 points)

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:

1. What is the domain name of the site the client is connecting to?

   What to submit: Submit a text file named `3.1.1.3_domain.txt` containing your answer.

2. During the TLS handshake, the client provides a list of supported cipher suites. List the cipher suites. Refer to the website with a list of known cipher suites table `http://www.thesprawl.org/research/tls-and-ssl-cipher-suites/`. Double check whether cipher suite name matches from the given page.

   What to submit: Submit a text file named `3.1.1.3_client.txt` where each line contains each cipher suite's name. Refer to the website with a list of known cipher suites table `http://www.thesprawl.org/research/tls-and-ssl-cipher-suites/`. Double check whether cipher suite name matches from the given page.

3. What cipher suite does the server choose for the connection?

   What to submit: Submit a text file named `3.1.1.3_server.txt` containing the corresponding cipher name.

3.1.1.4 Facebook traffic analysis (5 points)

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

1. Even though logins are processed over HTTPS, what is insecure about the way the browser is authenticated to Facebook?

   What to submit: Submit a text file named `3.1.1.4_insecurity.txt` containing your answer.

2. How would this let an attacker impersonate the user on Facebook?

   What to submit: Submit a text file named `3.1.1.4_impersonate.txt` containing your answer.

3. How can users protect themselves against this type of attack?

   What to submit: Submit a text file named `3.1.1.4_protect.txt` containing your answer.

4. What did the user do while on the Facebook site?

   What to submit: Submit a text file named `3.1.1.4_user.txt` containing your answer.

Checkpoint 1: Submission Checklist

Inside your mp3 directory svn, you will have the auto-generated files named as below. Make sure that your answers for all tasks up to this point are submitted in the following files before **Wednesday, October 14** at **6:00pm**:

SVN Directory

`https://subversion.ews.illinois.edu/svn/fa15-cs461/NETID/mp3`

Team Members

`partners.txt` : a text file containing netIDs of both members, one netid per line. Place the student's netID, whose directory contain your project submission, at the top of the file.

example content of partners.txt

```
netid1
netid2
```

Solution Format

example content of 3.1.1.1_mac.txt

```
0f:0f:0f:0f:0f:0f
1e:1e:1e:1e:1e:1e
```

example content of 3.1.1.1_ip.txt

```
1.2.3.4
127.0.0.1
```

example content of 3.1.1.2_dns.txt

```
dns1.illinois.edu
```

example content of 3.1.1.3_domain.txt

```
illinois.edu
```

example content of 3.1.1.3_client.txt

```
TLS_NULL_WITH_NULL_NULL
TLS_RSA_WITH_NULL_MD5
TLS_RSA_WITH_NULL_SHA
```

List of solution files that must be submitted for checkpoint 1

- `partners.txt`
- `3.1.1.1_mac.txt`
- `3.1.1.1_ip.txt`
- `3.1.1.2_dns.txt`
- `3.1.1.2_ftp.txt`
- `3.1.1.3_domain.txt`
- `3.1.1.3_client.txt`
- `3.1.1.3_server.txt`
- `3.1.1.4_insecurity.txt`
- `3.1.1.4_impersonate.txt`
- `3.1.1.4_protect.txt`
- `3.1.1.4_user.txt`