Ira A. Fulton Schools of Engineering
Arizona State University

CSE 543: Information Assurance and Security

TCP Spoofing

Purpose

The purpose of this assignment is to familiarize you with the most traditional type of TCP spoofing attack. You will learn how to create and use raw sockets, send raw IP packets with forged source IPs, manually establish TCP connections, and manually create TCP packets.

Objectives

Students will be able to:

- Program with raw sockets.
- Manually and programmatically create TCP packets.
- Programmatically simulate TCP connections from a forged IP.
- Send TCP packets with spoofed source IP addresses.

Technology Requirements

While students can use any programming languages, Python is strongly recommended since it saves your time of coding.

Note: The course team will not be able to help you if you choose any language that is not Python, Java, or C#; therefore, to create the best learning experience, Python is strongly recommended.

Project Description

A *TCP* service FlagIt is running at flagit.cse543.rev.fish:13337. This TCP service receives a target IP address from the user, and if the user is authenticated, it will happily send a flag (a special string) via UDP to port 13337 of the target IP. Your job is to write a program that retrieves the flag.

FlagIt employs THE BEST AUTHENTICATION METHOD IN THE WORLD: Source-IP-based authentication, which means it authenticates all users based on their source IP addresses. If a user's source IP address is trusted, FlagIt will send out the flag to the specified destination (repeat: via UDP). Otherwise, it will send an error message back to the untrusted user (via UDP, too).

The only trusted IP is 10.2.4.10. Your task is to break or bypass this source-IP-based authentication scheme and steal the flag.

To keep the internet a secure place, flagit.cse543.rev.fish points to a private IP that is only accessible through VPN. You will receive an OpenVPN configuration file. You will need to install OpenVPN and use it to connect to the VPN before talking to FlagIt.

Directions

You are strongly encouraged to use Python 3 and the scapy package to solve this problem. Here is a step-by-step guide:

1. Since the goal is to conduct TCP spoofing, you may want to revisit the lecture about TCP spoofing.
2. The key of TCP spoofing is correctly guessing the sequence number that the server sends back. It is worth noting that the server is not using a secure TCP/IP stack. Specifically, the server does not properly choose *random* initial sequence numbers for each TCP connection. Therefore, you should collect several sequence numbers from talking to the server and observe what is special about these sequence numbers.
3. Assuming you figured out the secret in how the server generates sequence numbers, the next step is to implement a TCP client that talks to the server using scapy.
4. Since your TCP client will send TCP segments with forged source IPs, you must work with raw sockets.
5. The only thing difficult with implementing a TCP client using raw sockets is correctly computing the checksum of each TCP segment. You may find online how TCP checksum works.
6. Make sure the source code of your TCP client is properly documented.
7. Use your TCP client to attack the server and get the flag. Remember that the flag will be sent back to you via UDP.
8. Submit your flag (as `flag.txt`), a file (`readme.txt`) that describes your thought process and your solution, and your code (as a separate zip file).

Evaluation

Your submission will be first automatically graded then manually graded. You will earn 100 points if your stolen flag.txt is correct. Otherwise the instructor will grade your submission and assign partial scores: You will earn 40 points (out of 100) if your code for predicting TCP sequence numbers is reasonable and correct; In addition, you will earn 30 points if your attacking logic is reasonable and correct. You will not be able to explain to the instructor what your submitted source code does or intends to do. Therefore, it is important to ensure that your code is properly commented, especially when your code does not fully work.

Submission Directions for Project Deliverables

You will submit three files for this assignment:

1. A txt file flag.txt with the stolen flag inside,
2. A txt file readme.txt describing your thought process or your solution to this problem.
3. Your code (a Python script or source code in any programming languages) that attacks the service and obtains the flag. Submit your code as a ZIP file.