Lab 2: TCP/IP Treasure Hunt

Here we dust off our knowledge of TCP/IP network programming by writing some simple clients and servers, and we learn how to capture flags in the lab's virtual network environment.

Due: 10pm Thursday, Apr 17, 2014

Challenge Problems

Problem 1: A Simple UDP Server

Follow the example code at python.org (https://wiki.python.org/moin/UdpCommunication) to write a UDP server. Listen on UDP port 2001 for a packet containing the flag.

Question: How else could you have captured this flag, without using a socket?

Problem 2: A Simple UDP Client

Follow the example code at python.org (https://wiki.python.org/moin/UdpCommunication) to write a UDP client. Send a packet to IP address 192.168.14.10 on UDP port 2002 and listen on the same UDP port (2002) for a response packet containing the flag.

Problem 3: A Simple TCP Server

Follow the example code at python.org (https://wiki.python.org/moin/TcpCommunication) to write a TCP server. Listen on TCP port 2003 for a message containing the flag.

Question: Would your alternate approach for capturing the flag in Problem 1 still work here? Why or why not?

Problem 4: A Simple TCP Client

Follow the example code at python.org (https://wiki.python.org/moin/TcpCommunication) to write a TCP client. Connect to IP address 192.168.14.20 on TCP port 2004. Send it the message "GET FLAG", and the server will send you the flag.

Problem 5: TCP Client with Password

Follow the example code at python.org (https://wiki.python.org/moin/TcpCommunication) to write a TCP client. Connect to IP address 192.168.14.30 on TCP port 2005 and the server will ask you for the password. If you provide the correct password, you may demand the flag as in Problem 4, and the server will send it to you.

Note: The password was unwisely chosen based on a dictionary word. Here (dictionary.txt)'s the dictionary.

Problem 6: TCP Client with Challenge Response

Follow the example code at python.org (https://wiki.python.org/moin/TcpCommunication) to write a TCP client. Connect to IP address 192.168.14.40 on TCP port 2006 and the server will send you a math puzzle. Send the correct answer back, and then you may demand the flag.

Problem 7: TCP Wild Goose Chase

Follow the example code at python.org (https://wiki.python.org/moin/TcpCommunication) to write a TCP client. Connect to IP address 192.168.14.10 on TCP port 2007 and demand the flag. The server will reply in one of two ways. Either it will give you the flag, or it will point you to another server that can help you find the flag. Follow the clues until you find the flag!

Problem 8: IMAP Client

Alice sent the flag to Bob in an email. Bob's IMAP server is 192.168.14.24. Unfortunately for Bob, he didn't pick a very good password. In fact, his password is about the worst password that he might have chosen for his password. Log in to Bob's email using his password and retrieve the flag.

Hint: The Subject line of the email is "FLAG".

Problem 9: Spear Phishing

Bob thinks that maybe he should get a new flag from Alice. He has a sneaking suspicion that the old one may have been compromised somehow. Forge an email to Alice from Bob (bob@example.com) by connecting to Alice's SMTP server at 192.168.14.48. The subject of your email should be "FLAG", and in the body it should say "GOTO (your IP) (your port)". Alice will connect via TCP to send you the new flag.

Hint: You don't need to know Alice's full email address for this one. In the SMTP connection, "rcpt to: alice" will work just fine.

Problem 10: Hidden TCP Service with Challenge Response

There is a server running on a high-numbered port at IP address 192.168.14.50. Find the open port and retrieve the flag.

Note: This server is running the same protocol as in Problem 06, so you'll need to solve a math problem to get the flag.