Networking and Disassembly Exercise

UNO CTF

1 Exercise Introduction

There is a file being served on a machine (IP will be provided) that you will need to find and disassemble with GDB in order to understand its purpose. It is a Linux executable. Nmap is highly suggested, and keep in mind that there are 65,535 possible ports to search!

2 Useful utilities/protocols for connections

2.1 Netcat

Netcat is somewhat of a Swiss army knife for testing network connections. It is a utility that can be used to connect to ports that may run a number of different networking protocols. You can use it to connect to any port, and it is often a boost when determining what type of service might lie on that port. If an initial connection does not provide you with enough information to determine the nature of the service on the port with which you connected, you might want to just try typing some commands to see how it responds.

2.2 Telnet

Telnet is not considered secure, but you might see backdoors placed on machines using Telnet. It may also often be used for administration of embedded devices. Unlike *Netcat*, Telnet is an application protocol as well as a commonly available application–you are able to connect to Telnet servers with Netcat!

2.2.1 Telnet basic usage

1. Connection

   ```
   telnet <host> [port]
   ```

2.3 SSH

SSH is a more secure shell protocol (as in the name). It's somewhat of a more secure version of Telnet. You can connect to SSH servers with Netcat as well, though you may have some difficulties. Netcat provides you with basic communication, but SSH clients abstract away much of the required work when establishing a connection.

3 Useful file transfer utilities/protocols

3.1 SCP

SCP is a network file transfer protocol based on SSH. It is generally used for simple file transfer, and generally as-needed per file or folder. SCP does not typically use a shell. As SCP is based upon SSH, the typical port used for a client connection to a server is port 22.

3.1.1 SCP Basic Usage

1. Local to remote transfer

   ```
   scp <localfilename> <user@desthost:destfile>
   ```

2. Remote to local transfer

   ```
   scp <user@sourcehost:sourcefile> <localfilename>
   ```

3. Remote to remote transfer

   ```
   scp <user@sourcehost:sourcefile> <user@desthost:destfile>
   ```

3.2 FTP

FTP is a network file transfer protocol. It is insecure, but still frequently used. FTP traffic is plaintext, and FTP even may allow passwordless anonymous logins, with the username "anonymous". FTP traffic flows according to two possible modes: in active mode, the server establishes the data connection with the client on a client-supplied port, but in passive mode, the client establishes the data connection on the server–passive mode may need to be used if errors result due to firewalls, though some clients handle this mode switch automatically.

3.2.1 FTP basic usage

1. Create connection

   ```
   ftp <host or IP address> [port]
   ```

2. Switch between active and passive mode while in FTP shell

   ```
   passive
   ```

3. List files while in FTP shell

   ```
   ls
   ```

4. Download file while in FTP shell

   ```
   get <filename>
   ```

4 Useful GDB commands for disassembly

4.1 `info functions`

This will list all functions in the program.

4.2 `disass <function>`

This will show the disassembly for the given function.

4.3 `x/s <address>`

Examine string at address

4.4 `break <function>`

Set a breakpoint on <function>

4.5 `break *<address>`

Set a breakpoint at <address>. Note that the asterisk (*) is used as a prefix when specifying addresses.

4.6 `delete [breakpoints]`

Delete [breakpoints] or, if no breakpoints are specified, delete all breakpoints