Discovering and exploiting a remote buffer overflow vulnerability in an FTP server – PART 1

Hello all, in this tutorial we will learn how to identify a vulnerability in an FTP server through the process of “Fuzzing” which could lead to a DoS or Buffer Overflow vulnerability identification. In this specific part we will use FTPFuzz to crash FileCOPA and identify a vulnerability in the LIST command.

First of all to get started, everything you will need to follow along this tutorial is downloadable in this .rar: http://sharingmatrix.com/file/713096/FileCOPAtutorial.rar

1) Set up FileCOPA and add a user with username “test” and password “test” (or whatever you want the username and password to be) and set the root folder to anything you like (here I used c:\ftp).

2) Just leave the server running, open Immunity Debugger and attach it to the process and press the play button to let it run.
3) Open FTPFuzz and enter the host. Now press “Discover” to discover what FTP commands are available on this server. FTPFuzz will automatically edit this.

*Note: if this were a real-life test on a server with no known vulnerabilities, we would leave it as it is after FTPfuzz discovered the commands for us, but for this example I’ll only leave on “USER” “PASS” and “LIST” (the vulnerability is in LIST. USER and PASS are just to login on the test account).

4) Go to USER and PASS and edit the arguments to fit the username and pass you specified when setting up the server (in my case this is just test:test).
5) Let **FTPFuzz** run and wait for **FileCOPA** to crash… and jup… it crashes!

6) Inside Immunity Debugger we can now see the **EIP** and **EBP** have been overwritten with 0x20412041 (which is “A A” — note the spaces — in reverse).

*We have now identified a vulnerability when using the LIST command, followed many times “A ” (\x20\x41)*

Take a moment to examine the following screenshot closely:
I hope you enjoyed this tutorial and that it has given you some insight on how to discover DoS or buffer overflow vulnerabilities.
In the next part, we will look at how we can exploit this vulnerability to execute remote shellcode.

Diggs, StumbleUpon’s, bookmarks at Delicious, … are always appreciated!

Many thanks go out to corelanc0d3r for his great tips.

Don’t forget to check out my twitter for more updates: http://twitter.com/Raykoid666

Original tutorial at my blog: http://raykoid666.wordpress.com