Fuzzing a Local Network Service with SPIKE

Write a Spike Script to Crash the Fuzzme.jar Application

In this Lab we want to write some a spike script in order to crash a simple fake network service. For this lab fuzzme will be used, which is a simple network service that speaks a plaintext protocol.

The basic protocol spec are:

Authentication is done

USER <username>\r\n

Commands

ls\r\n (no arguments)
whoami (no arguments)
cat <filename>\r\n

Notes:

- Use Kali or any Linux OS to work this lab
- Fuzzme.jar could be downloaded from here.
- Fuzzme.jar runs locally (127.0.0.1) on port 81
- Connect to fuzzme.jar to make sure its working using telnet/nc
- Check the log file for output (successful buffer overflow discovery)

Deliverables (1):

1. Write a SPIKE script to find buffer overflow bugs in fuzzme.jar
2. What was the size of the buffer and command that caused a buffer overflow?

Basic Walkthrough

You will receive some guidance in this lab, as the labs go on, you will receive less guidance from me, as I want you to work things yourself (still here to help, so ask me if you need my help).

Task #1

Start by running the fuzzme.jar application using any of the commands below:

```
# chmod +x runme.sh
# ./runme.sh
```

OR

```
# java -jar fuzzme.jar
```

For us to learn what data is being sent back and forth, we will be using a network traffic capturing tool "**Wireshark**". Start the tool and do the proper configurations required to capture the traffic going to the fuzzme.jar application.

```
# wireshark
```

Note: wireshark should be working before proceeding.

Before fuzzing the application, let us test what this application does by connecting to it. We will do that using **netcat**. What IP Address will you be using for the command below?

```
# nc <ip-address> 81
```

Task #2

A) Go back to page #1 of this document and read the specs. Then send an authentication command followed by other commands.
B) Check the wireshark captured packets. What did you find? Explain.

Task #3
Now let us start writing our SPIKE script to fuzz this application. Open any text editor and write the code below, then save it as "**fuzzme.spk**".

```
s_string("user Ali\r\n");
s_string("cat ");
s_string_variable("secretfile.txt\r\n");
```

A) Why did we use the first two lines "s_string" while "s_string_variable" in the last?

Before running the script below, restart Wireshark again and then run the spike script:

```
# generic_send_tcp   127.0.0.1    81 fuzzme.spk    0    0
```

B) What was the total number of strings sent?

C) From wireshark, show an example of the first fuzz sent and then any other stream of data sent too. What is the difference?

Check the output.log file using grep, then:

D) Was the application vulnerable to a buffer overflow?

E) Which command caused the overflow?

F) What was the size of the buffer that caused the overflow?

Task #4 - Reflection
Please reflect on what you have learned from this lab.