REVERSE ENGINEERING AND EXPLOITATION

Session 2

Introduction

When fuzzing, it's usual to use two workstations: one for launching the Sulley fuzzing run; the other to host the target (eg. WarFTPD) server to be fuzzed. We'll call these workstations *fuzz control* and *fuzz target* respectively. Normally you would need to know the IP address of the VMWare virtual machine on *fuzz target*. Then by pinging this IP address from *fuzz control*, you can ensure that a TCP/IP network connection may be made. However, this week all code will be running on the same machine, thus simplifying parts of our setup.

This weeks practical code is taken from chapter 9 of Gray Hat Python.

Questions

1. Enter and save the following Sulley FTP protocol skeleton into a file named *ftp.py* on *fuzz control*:

```
from sulley import *

s_initialize(“user”)
s_static(“USER”)
s_delim(“ “)
s_string(“cis2376”)
s_static(“\r\n”)

s_initialize(“pass”)
s_static(“PASS”)
s_delim(“ “)
s_string(“security workshop”)
s_static(“\r\n”)

s_initialize(“cwd”)
s_static(“CWD”)
s_delim(“ “)
s_string(“c: “)
s_static(“\r\n”)

s_initialize(“dele”)
s_static(“DELE”)
s_delim(“ “)
```

```
s_string(“c:\\test.txt”)
s_static(“\r\n”)

s_initialize(“mdtm”)
s_static(“MDTM”)
s_delim(“ “)
s_string(“c:\\boot.ini”)
s_static(“\r\n”)

s_initialize(“mkd”)
s_static(“MKD”)
s_delim(“ “)
s_string(“c:\\testdir”)
s_static(“\r\n”)
```

2. What FTP commands, if any, has our protocol skeleton missed out? Refer to RFC959 to work this out:

   http://www.faqs.org/rfcs/rfc959.html

3. Before fuzzing WarFTPD, let’s investigate how fuzzing works in Sulley. If you had a Sulley request named *example*, then the following code would allow you to list **all** possible fuzzed strings that the request can generate:

   ```
   import time
   request = s_get(“example”)
   mutations = request.num_mutations()
   for count in range(mutations):
     print request.render()
     time.sleep(2)
     request.mutate()
   ```

   Use this code to see what fuzzed strings the Sulley requests (in *ftp.py*) *user* and *mkd* generate.

4. Enter and save the following Sulley FTP session code in a file named *ftp_session.py* on *fuzz control*:

   ```
   from sulley import *
   import ftp

   def receive_ftp_banner(sock):
     sock.recv(1024)

   sess = sessions.session(session_filename=”C:\\warftpd.session”)
   target = sessions.target(“xx.xx.xx.xx”, 21)
   target.netmon = pedrpc.client(“xx.xx.xx.xx”, 26001)
   target.procmon = pedrpc.client(“xx.xx.xx.xx”, 26002)
   target.procmon_options = { “proc_name”: “war-ftpd.exe”,
   ```

```
                              “start_commands”: [“C:\\Documents and Set-
tings\\Administrator\\My Documents\\war-ftpd.exe”] }

sess.pre_send = receive_ftp_banner
sess.add_target(target)
sess.connect(s_get(“user”))
sess.connect(s_get(“user”), s_get(“pass”))
sess.connect(s_get(“pass”), s_get(“cwd”))
sess.connect(s_get(“pass”), s_get(“dele”))
sess.connect(s_get(“pass”), s_get(“mdtm”))
sess.connect(s_get(“pass”), s_get(“mkd”))

sess.fuzz()
```

Note: in the above code, the IP address xx.xx.xx.xx should be replaced by the IP address of *fuzz target*. Within the labs, we shall run be running WarFTP on the same machine as the client, so xx.xx.xx.xx can be replaced by 127.0.0.1 (ie. the localhost)

5. Assuming your code in questions 1 and 4 has been entered correctly, let’s set up a Sulley fuzzing run against WarFTPD[1]! The following command should be ran on *fuzz control*:

   ```
   python “C:\Program Files\sulley\process_monitor.py” -c C:\\warftpd.crash -p
   war-ftpd.exe
   ```

   This command setups and runs a code monitor. Should WarFTPD crash during the fuzzing run, Sulley will automatically restart it *and* save the crash image!

   We may now start the fuzzing run with the following command (issued on *fuzz control*):

   ```
   python ftp_session.py
   ```

   The fuzzing run may be monitored by using Sulley’s built in web interface. To access this, launch Firefox on *fuzz control* and navigate to:

   http://127.0.0.1:26000

6. Using the site:

   http://www.milw0rm.com

   what exploits for WarFTPD 1.65 can you find? Do any of these exploits bare any resemblance to the crash data you have generated with your Sulley fuzzing runs?

[1] **Note:** obviously, you’ll first need to have downloaded and unpacked an appropriate version of WarFTP, for example:

http://www.warftp.org/files/1.6_Series/ward165.exe

Please ensure that you unpack WarFTP in the same directory as your ftp.py and ftp_session.py file is located. Here we assume that this location is:

`C:\Documents and Settings\Administrator\My Documents`