Exploiting Simple Buffer Overflows on Win32

Vivek Ramachandran

SWSE, SMFE, SPSE, SISE, SLAE, SGDE Course Instructor

Certifications: http://www.securitytube-training.com

Pentester Academy: http://www.PentesterAcademy.com

Automating Bad Character Detection

Challenges in Bad Character Detection

- Painfully manual

- Visual Inspection
  - characters can be transformed which one might overlook

What if we had a tool …

- Could generate character sets
- Compare memory location with the generated character sets used
- Finds bad characters using the above

Enter Mona.py!

Mona.py

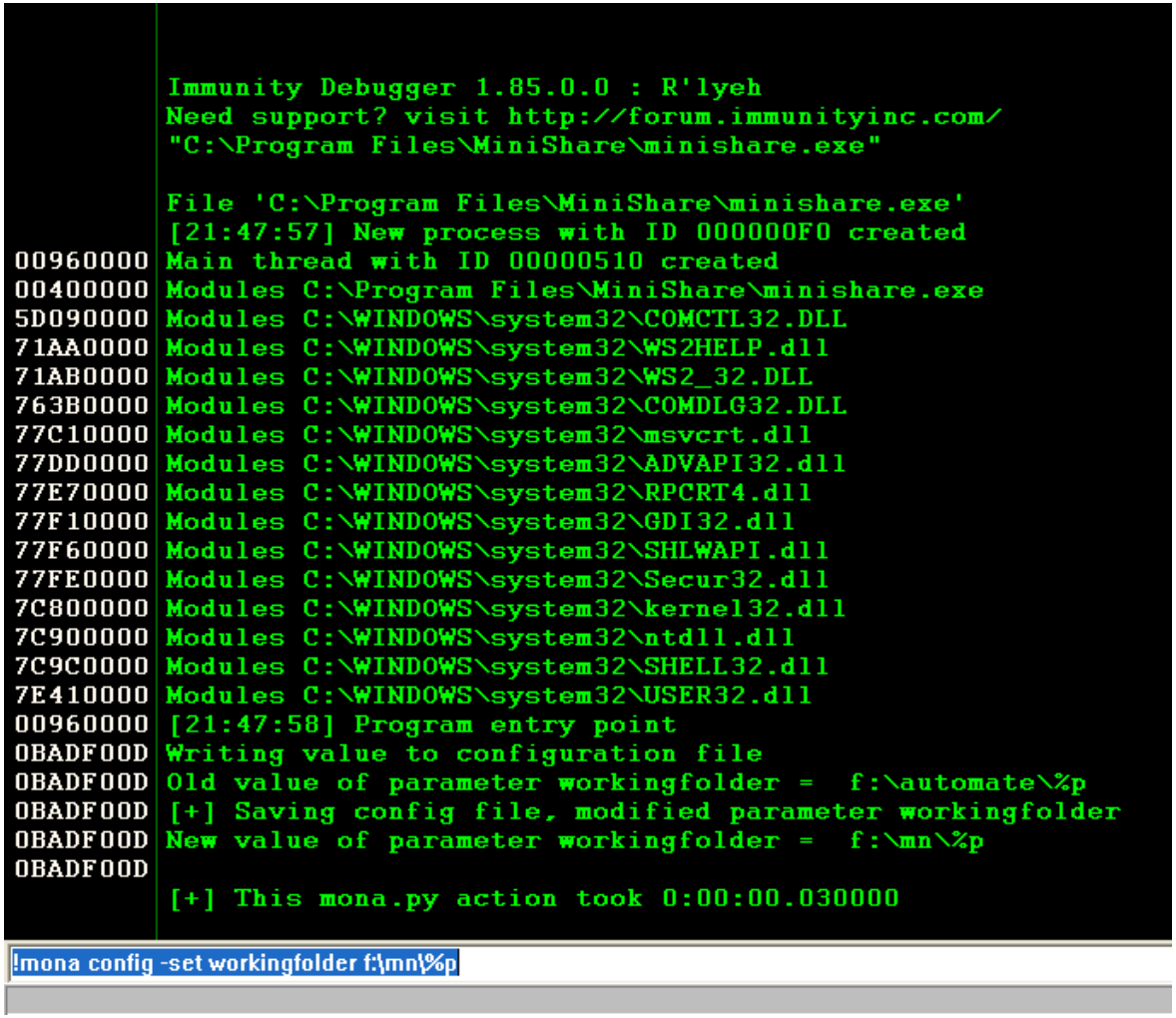
- http://redmine.corelan.be/projects/mona
- Created by the Corelan Team (Peter)
- Plugin for Immunity Debugger
- https://www.corelan.be/index.php/2011/07/14/mona-py-the-manual/ (Documentation)

Minishare

- Bad Character Detection using Mona.py

- Minishare

Step 1: Set Working Folder for Mona

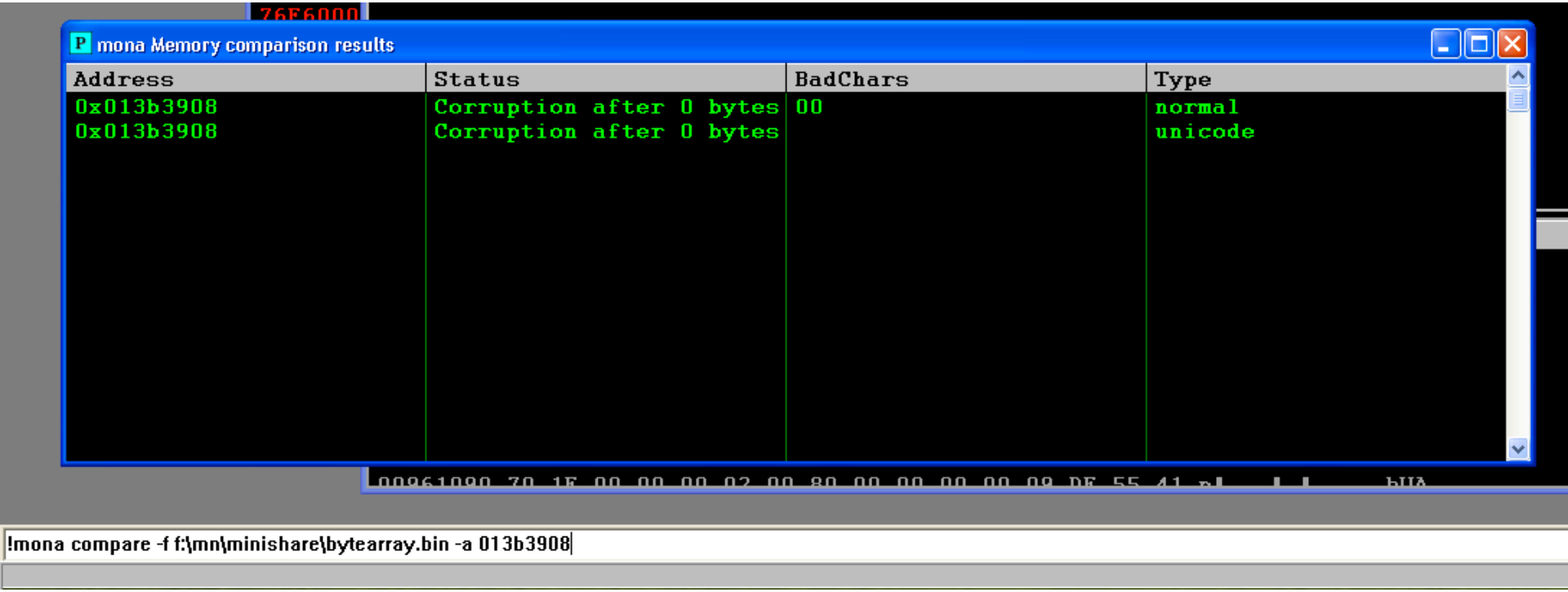


Step 2: Generate Bytearray

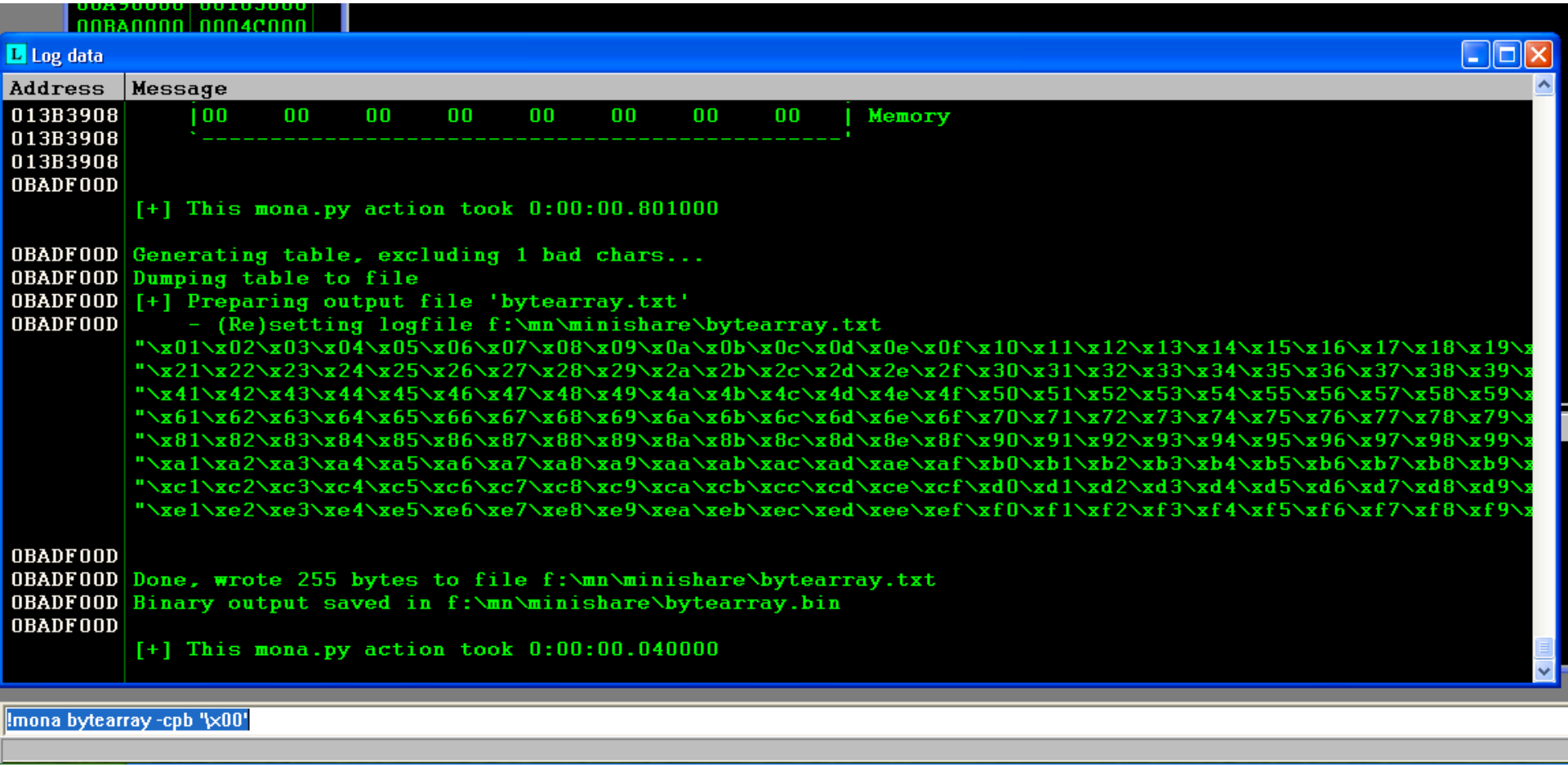
```
7E410000 Modules C:\WINDOWS\system32\USER32.dll
0BADF00D Generating table, excluding 0 bad chars...
0BADF00D Dumping table to file
0BADF00D [+] Preparing output file 'bytearray.txt'
0BADF00D     - Creating working folder f:\mn\Echo-Server-Bad-Char-Special
0BADF00D     - Folder created
0BADF00D     - (Re)setting logfile f:\mn\Echo-Server-Bad-Char-Special\bytearray.txt
         "\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"

0BADF00D
0BADF00D Done, wrote 256 bytes to file f:\mn\Echo-Server-Bad-Char-Special\bytearray.txt
0BADF00D Binary output saved in f:\mn\Echo-Server-Bad-Char-Special\bytearray.bin
0BADF00D
         [+] This mona.py action took 0:00:00.040000
```

```
!mona bytearray
```

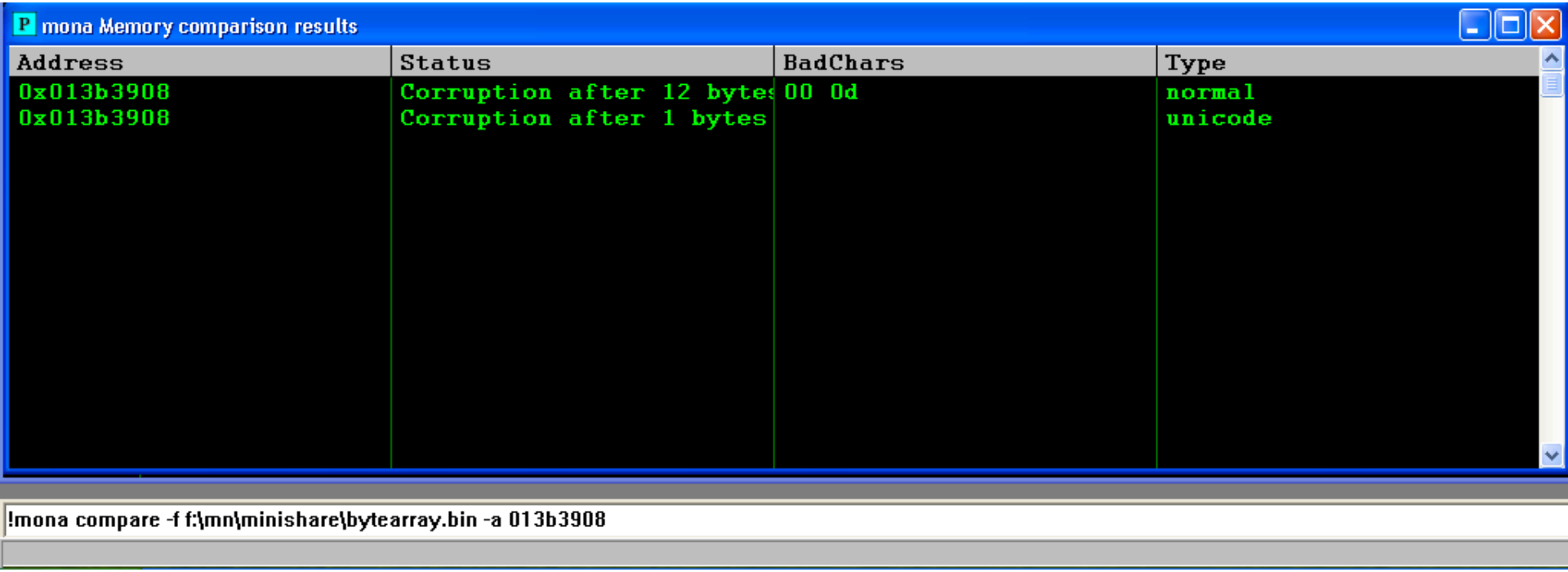
Step 3: Find Bad Character



Step 4: Create Bytearray with Exclusions



Step 5: Repeat and Find All Bad Chars



Pentester Academy

PentesterAcademy | a SecurityTube.net initiative

TOPICS PRICING WHY SUBSCRIBE MEMBER ACCESS

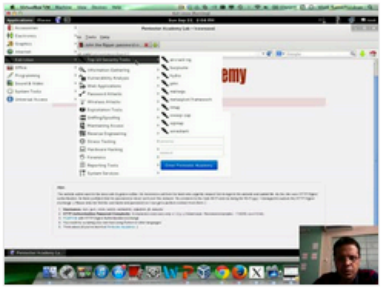
Revolutionizing Infosec Training

Highly Technical, Hands-on, Affordable

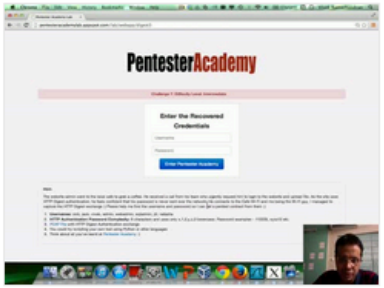
Start Learning Today!

Latest Videos

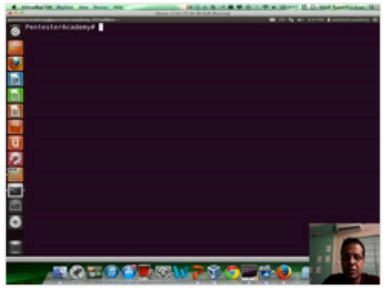
New content added weekly!



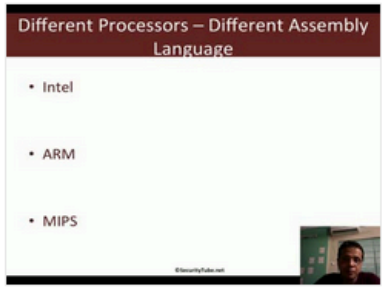
Challenge 7: Cracking Digest Authentication Solution in WAP Challenges



Challenge 7: Cracking Digest Authentication in WAP Challenges



Module 1: GDB Test Solution in x86_64 Assembly Language and Shellcoding on Linux



Module 1: CPU Information in x86_64 Assembly Language and Shellcoding on Linux