Hunting bad characters with mona.py

This document is how-to guide to identify bad characters when developing a buffer overflow exploit, what i like about this technique is it's fool-proof. Here's the steps

- In Immunity Debugger set where mona.py will be logging all of its output

```
!mona config -set workingfolder c:\logs\%p
```

- Now we need to generate a byte array for comparison. In Immunity Debugger run the command

```
!mona bytearray -cpb "\x00"
```

Note: I always exclude null byte right off the bat.

- Go to logging directory set in first step and copy list of bad characters generated in '**bytearray.txt**' to your exploit. Please note that exploit MUST match '**bytearray.txt**' for this to work properly.

- Now run exploit , again make sure it's '**badchars**' variable matches '**bytearray.txt**'

- After the crash, run the command in Immunity Debugger

```
!mona compare -f C:\logs\<program_name>\bytearray.bin -a
<memory_address_where_badchars_start>
```

This will compare characters in file '**bytearray.txt**' with those in memory (pushed by exploit) and spits ones that cause the program to act funny. Please Note that memory address needs to be accurate, for instance if '**badchars**' in exploit start 2 bytes after left-hand side address in Immunity debugger you'd need to increment that by two bytes in above command

- If you get a bad character you do have to include it in step 2 command and repeat the process until there is no bad characters.

I've used this technique three times thus far and it works like a charm.