

BUFFER OVERFLOW FOR BEGINNERS by JOAS

Introduction and Concept

- <https://www.triaxiomsecurity.com/introduction-to-buffer-overflow-attacks/>
- <https://betterprogramming.pub/an-introduction-to-buffer-overflow-vulnerability-760f23c21ebb>
- https://www.youtube.com/watch?v=qSnPayW6F7U&ab_channel=TheCyberMentor
- https://www.youtube.com/watch?v=liOQ-XD3uDw&ab_channel=DionTraining
- https://www.youtube.com/watch?v=54ivEIVZk8I&ab_channel=BittenTech
- https://www.youtube.com/watch?v=4rUNIF6_Mhk&ab_channel=NakerahNetwork
- <https://d0nut.medium.com/week-13-introduction-to-buffer-overflows-5f15c0d5b5c1>
- <https://www.imperva.com/learn/application-security/buffer-overflow/>
- <https://ijournals.in/wp-content/uploads/2017/07/14.3425-Nilesh.compressed.pdf>
- <https://searchsecurity.techtarget.com/definition/buffer-overflow>
- <https://medium.com/techloop/understanding-buffer-overflow-vulnerability-85ac22ec8cd3>
- <https://www.sans.org/white-papers/481/>
- <https://www.hackingarticles.in/a-beginners-guide-to-buffer-overflow/>
- <https://www.coengodegebure.com/buffer-overflow-attacks-explained/>

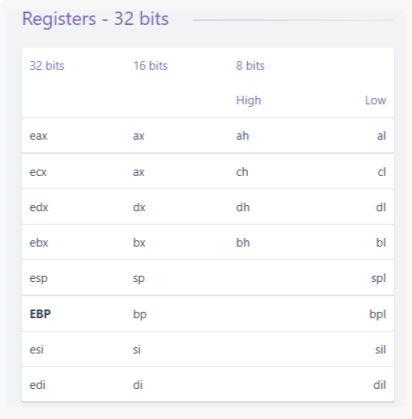
Practice Laboratory

- <https://mrmoddom.github.io/ctf/walkthrough-Stack-Overflows-for-Beginners-1/>
- <https://www.vulnhub.com/entry/stack-overflows-for-beginners-101,290/>
- <https://medium.com/@iwhoam047/stack-overflows-for-beginners-1-41acf75d230>
- <https://medium.com/dsc-sastra-deemed-to-be-university/buffer-overflow-vulnserver-4951a4318966>
- <https://z3r0th.medium.com/a-simple-buffer-overflow-using-vulnserver-86b011eb673b>
- https://www.youtube.com/watch?v=yJFOYPd8lDw&ab_channel=JohnHammond
- https://www.youtube.com/watch?v=OOkU7to0Ty4&ab_channel=JesseK
- https://www.youtube.com/watch?v=dCWgoSlni6s&ab_channel=NobodyAtall
- https://www.youtube.com/watch?v=k7EOkalfSKw&ab_channel=timthetinkerer
- https://www.youtube.com/watch?v=qyoALEIAJqo&ab_channel=drprventura
- <https://princerohit8800.medium.com/buffer-overflow-exploiting-smail-email-server-f90b27459911>
- <https://esseum.com/win-32-buffer-overflow-walkthrough-exploiting-smail-5-5/>
- <https://github.com/CyberSecurityUP/Buffer-Overflow-Labs>
- <https://medium.com/@rafaelrenovaci/exploit-to-pcman-ftp-server-2-0-7-remote-buffer-overflow-cffa8b8faddb>
- <https://medium.com/@mtucunduva98/buffer-overflow-pcman-ftp-server-2-0-7-el43ff3473c>
- <https://www.hebunilhanli.com/wonderland/bof-101/>
- http://www.computersecuritystudent.com/SECURITY_TOOLS/BUFFER_OVERFLOW/WINDOWS_APPS/lesson1/index.html
- https://www.youtube.com/watch?v=h5N0aOZjXDE&ab_channel=raulcpcop

Fundamentals Assembly (Sec4US)

Register is a small space used by CPU to store information.

Register Description — sp 16 bits — ESP 32 bits — rsp 64 bits



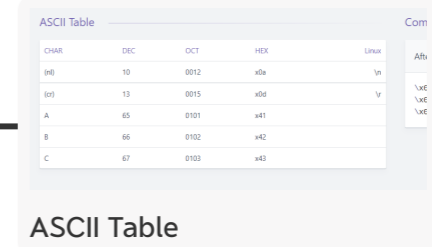
Registers - 32 bits		
32 bits	16 bits	8 bits
High		
Low		
eax	ax	ah
ecx	cx	ch
edx	dx	dh
ebx	bx	bh
esp	sp	sp
ebp	bp	bp
esi	si	si
edi	di	di

Register 32 Bits

IP - Instruction Pointer — Points to next instruction to be executed

- ip 16 bits
- EIP 32 bits
- rip 64 bits

Register Description



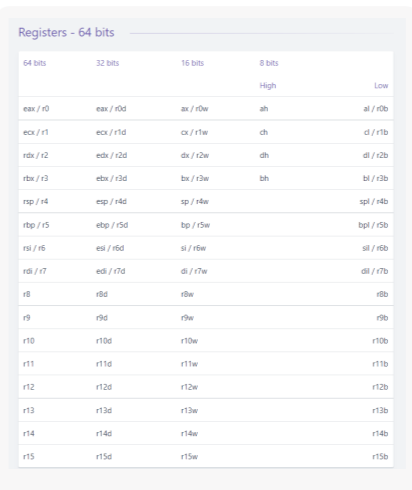
Code	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0x	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
1x	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
2x	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
3x	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
4x	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
5x	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
6x	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
7x	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
8x	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
9x	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
Ax	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
Bx	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
Cx	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
Dx	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
Ex	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
Fx	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

ASCII Table

Badchars

<https://sec4us.com.br/cheatsheet/bufferoverflow-windows>

<https://sec4us.com.br/cheatsheet/bufferoverflow-egghunting>

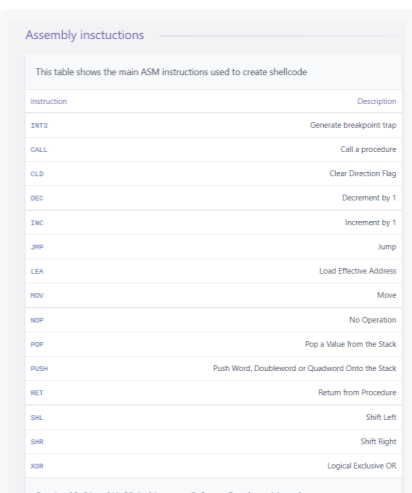


Registers - 64 bits			
64 bits	32 bits	16 bits	8 bits
High			
Low			
rax	rax	ax	ah
rcx	rcx	cx	ch
rdx	rdx	dx	dh
rbx	rbx	bx	bh
rsp	rsp	sp	sp
rbp	rbp	bp	bp
rsi	rsi	si	si
rdi	rdi	di	di
r8	r8	r8	r8
r9	r9	r9	r9
r10	r10	r10	r10
r11	r11	r11	r11
r12	r12	r12	r12
r13	r13	r13	r13
r14	r14	r14	r14
r15	r15	r15	r15

Register 64 Bits

SP Stacker

- sp 16 bits
- esp 32 bits
- rsp 64 bits



Instruction	Description
ADD	Arithmetic Add
AND	Bitwise AND
CALL	Call a procedure
CMOV	Compare and Move
DEC	Decrement by 1
INC	Increment by 1
JMP	Jump
LEA	Load Effective Address
MOV	Move
MOVQ	Move Quadword
POP	Pop a value from the stack
PUSH	Push Word, Quadword or Quadword onto the stack
RET	Return from procedure
ROL	Rotate Left
ROR	Rotate Right
SET	Logical Set/Reset CR

Assembly Instructions

- https://en.wikipedia.org/wiki/X86_instruction_listings
- <https://flint.cs.yale.edu/cs421/papers/x86-asm/asm.html>
- <https://software.intel.com/content/www/us/en/develop/articles/introduction-to-x64-assembly.html>
- https://cs.brown.edu/courses/cs033/docs/guides/x64_cheatsheet.pdf
- <https://www.felixcloutier.com/x86/>