Buffer Overflow Guide 1

Joas Antonio

https://www.linkedin.com/in/joas-antonio-dos-santos

What is Exploit?

- It is a piece of script designed to explore a determined security breach;
- Exploits consist of shellcodes and a piece of code to insert in a vulnerable application;

What is Shellcode?

- Shellcode is defined as a set of instructions injected and then executed by an exploit. Shellcode is used to directly manipulate the logs and functionality of an exploit, even ensuring a shell on the target machine, being its main purpose and many harnessing the codename Shell to refer to this, but maybe it turns out to be just an idea.
- In buffer overflow are codes used in exploration, used in development of exploits to exploit Buffer Overflow. who ever analyzed buffer overflow exploits already seen them, shellcodes are built only with the hexadecimal values of the opcodes of the target architecture, or that is, the instructions of the processor itself, so the understanding of the assembly language, which to some extent, has a 1 to 1 relationship with the machine language, if necessary. The shellcode is the code that will be actually executed while exploiting a buffer overflow. are called 'shellcodes' as your goal is usually to get a shell.

Exploit Development and Buffer Overflow Requeriments

- Knowledge in C Language;
- Knowledge in Python Language;
- Knowledge in Assembly x86 and x64;
- Memory and system address management;
- Knowledge in Buffer Overflow;
- Knowledge in Reverse Engineering;
- Knowledge in Registers;
- Concepts of Software Protection Mechanisms (DEP, NX and ASLR);
- Fuzzing Concepts;

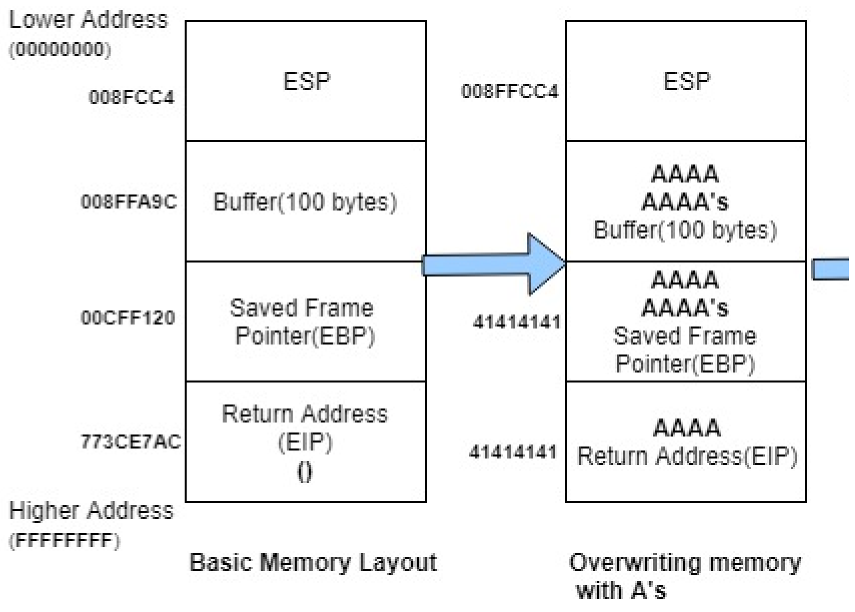
Stack Based Buffer Overflow

- Stack-based buffer overflow exploits are likely the shiniest and most common form of exploit for remotely taking over the code execution of a process. These exploits were extremely common 20 years ago, but since then, a huge amount of effort has gone into mitigating stack-based overflow attacks by operating system developers, application developers, and hardware manufacturers, with changes even being made to the standard libraries developers use. Below, we will explore how stack-based overflows work and detail the mitigation strategies that are put in place to try to prevent them.
- https://www.rapid7.com/blog/post/2019/02/19/stack-based-buffer-overflow-attacks-what-you-need-to-know/

Stack Based Buffer Overflow

The first thing that we need to do is send more data that the buffer can handle which overwrites the EIP Address as shown in the following figure.

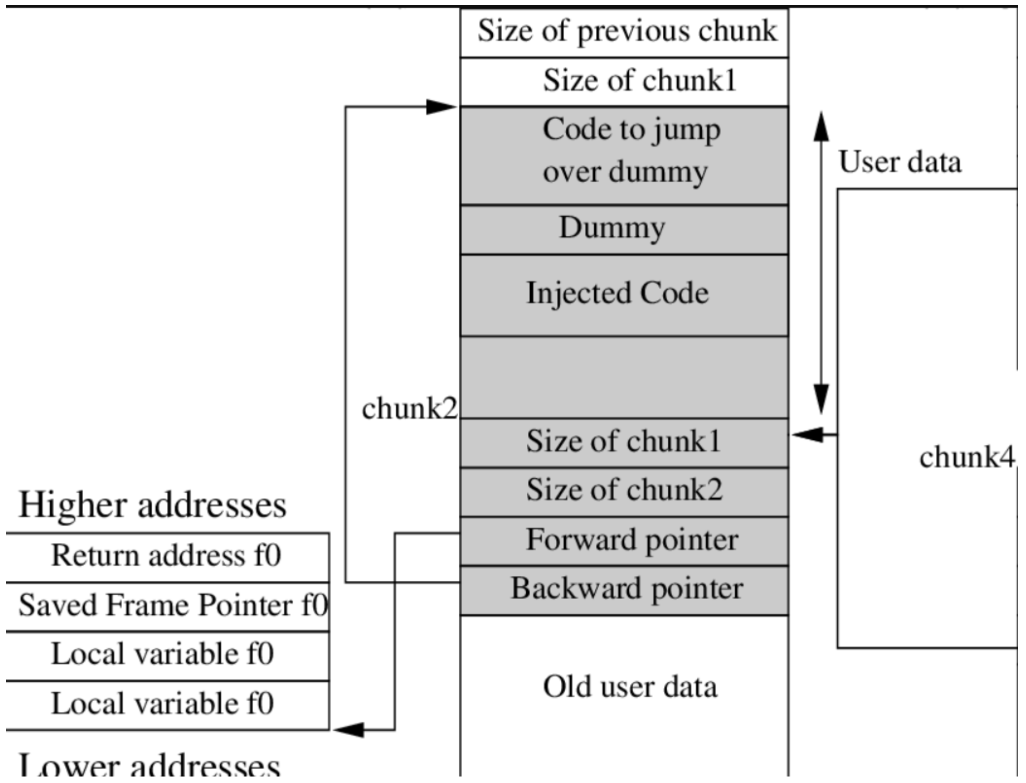
https://priyasloka.wordpress.com/2018/04/13/buffer-overflow-exploit-part-3/



HEAP CORRUPTION EXPLOITS

- Heap corruption occurs when the heap memory area is not has enough space for the data being written to it
- Heap memory is used dynamically by the application in time of execution

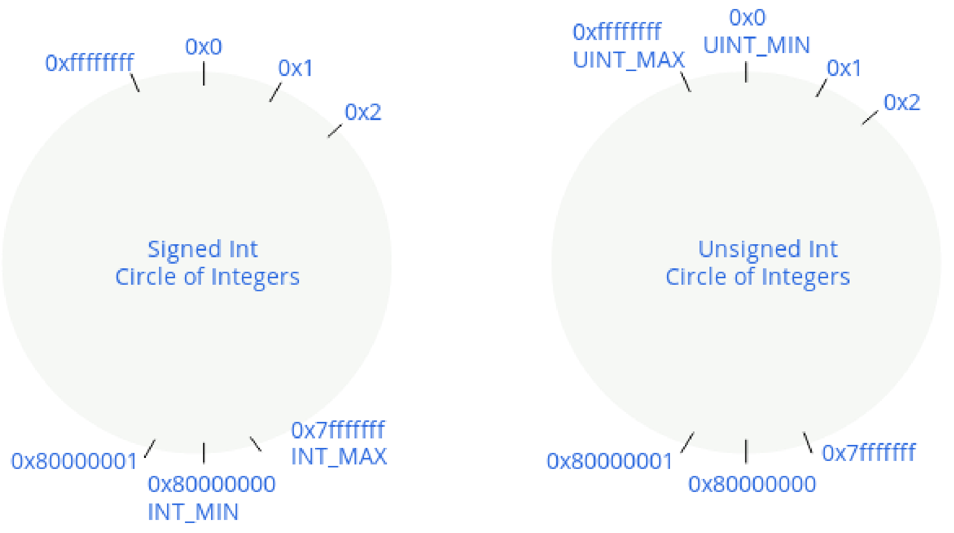
https://www.researchgate.net/figure/Heap-based-buffer-overflow-in-dlmalloc_fig1_244152148



Integer Overflow Attack

- Integer bugs are exploited by passing an integer oversized to a variable integer
- This may cause program control data to be overwritten valid, resulting in malicious code execution

https://resources.infosecinstitute.com/topic/defeating-integer-overflow-attack/



Race Condition

- Race condition is a software vulnerability that occurs when multiple accesses to the shared resource are not properly controlled;
- Types of Race Condition Attacks:
  - File race condition: Occurs when the attacker exploits a non atomic condition in a timed manner by creating, writing, reading and deleting a file in a temporary directory;
  - Signal Race Condition: The treatment race conditions signal can occur whenever an installed function such as a signal handler is not reentrant, which means it keeps some internal state or call another function that does;

  https://en.wikipedia.org/wiki/Race_condition

Socket Binding Exploits

- Client-side socket: Involves writing code to connecting the application to a remote server;
- Server-side socket: Involves writing code to listen on a port and process incoming connections;

https://realpython.com/python-sockets/
https://pymotw.com/2/socket/tcp.html
https://wiki.python.org.br/SocketBasico
https://www.embarcados.com.br/socket-server-tcp-em-c-inteledison/

https://www.geeksforgeeks.org/socket-programming-cc

Basic step-by-step development an exploit

1. Identify and analyze application bugs;
2. Write code to manipulate and control the target's memory;
3. Redirect the execution flow;
4. Inject the Shellcode;
5. Encrypt your socket communication;

Differences between Exploits on Windows and Linux

Windows:

- Explores call functions exported by link libraries Dynamic;
- Exploits written for Windows override return addresses in stack with an address containing “jmp reg” instruction where reg represents record;

Linux:

- Linux exploits use system calls;
- Exploits replace saved return addresses with a address stack where a user supplied data can be found;

Types of Shellcode

- Shellcodes are sets of instructions used by exploit programs to perform the desired function;
- They run after a vulnerability is exploited;
- Machine instructions are used to directly process the instruction desired in the memory location;
- These machine instructions consist of opcodes;

Remote Shellcode:

- Port Binding Shellcode
- Socket Descriptor Reuse Shellcode

Shellcode location:

- execve shellcode
- setuid shellcode
- chroot shellcode
- Windows shellcode

Shellcode Development

- Write code in assembly language or in C language and do
a disassembler;
- Collect the args and syscall id;
- Convert your assembly code into opcodes;
- Eliminate null bytes;
- Start a shell;
- Compile and run;
- Trace the code;
- Inject shellcode at program startup;

https://www.linkedin.com/pulse/development-um-simples shellcode-e-explorando-uma-dos-santos/

https://seedsecuritylabs.org/Labs_16.04/Software/Shellcode/

https://aayushmalla56.medium.com/shellcode-development-4590117a26bf

Shellcode Development Problems

- Address problem;
- Null Bytes Problem;
- Implementation of System Call;
- https://reverseengineering.stackexchange.com/questions/9184/the-addressing-problem
- https://reverseengineering.stackexchange.com/questions/8504/problem-finding-returnaddress-for-shellcode
- https://null-byte.wonderhowto.com/how-to/writing-64-bit-shellcode-part-2-removingnull-bytes-0161591/
- https://stackoverflow.com/questions/9776889/null-bytes-in-shellcode
- http://books.gigatux.nl/mirror/kerneldevelopment/0672327201/ch05lev1sec4.html
- https://cs155.stanford.edu/papers/traps.pdf
- http://courses.cms.caltech.edu/cs124/lectures/CS124Lec14.pdf

Assembly Basic

- https://caloni.com.br/basico-do-basico-assembly
- http://www.inf.furb.br/~maw/arquitetura/aula16.pdf
- https://paginas.fe.up.pt/~jmf/mp0506/dwnlds/mp1-0506-print.pdf
- https://www.tutorialspoint.com/assembly_programming/assembly_registers.htm
- https://www.eecg.utoronto.ca/~amza/www.mindsec.com/files/x86regs.html
- https://medium.com/@tirkarp/understanding-x86-assembly-5d7d637efb5
- https://www.cs.uaf.edu/2017/fall/cs301/lecture/09_11_registers.html
- https://cs.brown.edu/courses/cs033/docs/guides/x64_cheatsheet.pdf
- https://www.amazon.com.br/Programa%C3%A7%C3%A3o-Baixo-N%C3%ADvelProgramas-Arquitetura/dp/8575226673
- http://www.mathemainzel.info/files/x86asmref.html
- http://www.jegerlehner.ch/intel/opcode.html
- https://medium.com/@FreeDev/cdm-x86-parte2-8542566d6586

Reverse Engineering

- https://mentebinaria.gitbook.io/engenharia-reversa/assembly/funcoes-e-pilha
- https://ic.unicamp.br/~ducatte/mc404/Slides/mc404_07_2s06.pdf
- https://www.youtube.com/watch?v=IN9ElO90uLc&ab_channel=PapoBin%C3%A1rio
- https://www.youtube.com/watch?v=eakAd9_5LwI&ab_channel=RespondeemC%C3%B3digos
- https://kienmanowar.wordpress.com/r4ndoms-beginning-reverse-engineering-tutorials/tutorial-15-using-the-call-stack/
- https://revers.engineering/applied-re-the-stack/
- https://www.begin.re/assignment-2
- https://github.com/wtsxDev/reverse-engineering
- https://github.com/0xZ0F/Z0FCourse_ReverseEngineering
- https://github.com/GeoSn0w/Reverse-Engineering-Tutorials
- https://github.com/OpenToAllCTF/REsources
- https://github.com/alphaSeclab/awesome-reverse-engineering/blob/master/Readme_en.md

Reverse Engineering #2

- https://github.com/uwdata/rev
- https://medium.com/bugbountywriteup/bolo-reverse-engineering-part-1-basic-programming-concepts-f88b233c63b7
- https://medium.com/@vignesh4303/reverse-engineering-resources-beginners-to-intermediate-guide-links-f64c207505ed
- https://fahrishih.medium.com/getting-started-with-reverse-engineering-a68a5b8bb6ef
- https://medium.com/@Andromeda./basics-of-anti-reverse-engineering-9173826f1914
- https://medium.com/@danielabloom/bolo-reverse-engineering-part-2-advanced-programming-concepts-b4e292b2f3e
- https://medium.com/secjuice/the-road-to-reverse-engineering-malware-7c0bc1bda9d2
- https://medium.com/@Andromeda./introduction-to-reverse-engineering-251d6432f7ec
- https://medium.com/swlh/intro-to-reverse-engineering-part-2-4087a70104e9
- https://www.foo.be/cours/dess-20122013/b/Eldad_Eilam-Reversing__Secrets_of_Reverse_Engineering-Wiley(2005).pdf

Windows Internals

- https://mentebinaria.gitbook.io/engenharia-reversa/windows-api
- https://docs.microsoft.com/en-us/windows/win32/apiindex/windows-api-list
- https://docs.microsoft.com/en-us/windows/win32/api/
- https://docs.microsoft.com/en-us/sysinternals/resources/windows-internals
- https://www.youtube.com/watch?v=YqfMpoOKEkA&ab_channel=TheSourceLens
- https://www.youtube.com/watch?v=4AkzIbmI3q4&ab_channel=TheSourceLens
- https://www.youtube.com/watch?v=vz15OqiYYXo&ab_channel=JasmineRice
- https://medium.com/@0xdeadbeefJERKY/windows-internals-course-review7001bfdf335e
- https://scorpiosoftware.net/2020/01/03/next-windows-internals-remotetraining/
- https://www.youtube.com/watch?v=qMWvqdtlbkQ

Buffer Overflow

- https://www.youtube.com/watch?v=oS2O75H57qU
- https://owasp.org/www-community/vulnerabilities/Buffer_Overflow
- https://www.imperva.com/learn/application-security/buffer-overflow/
- https://www.youtube.com/watch?v=VX27nq6EcjI&list=PLcKsaFvYl4l87C8_HhxkcFaoNhzC9bRw6&ab_channel=Vin%C3%ADciusVieira
- https://www.youtube.com/watch?v=59_gjX2HxyA&ab_channel=RicardoLongAtto
- https://www.youtube.com/watch?v=HrFZ6ry6roQ&ab_channel=Sec4US
- https://www.youtube.com/watch?v=wLi-dGphpdg
- https://medium.com/@mzainkh/how-it-works-buffer-overflow-attack-4dcae8fa2630
- https://resources.infosecinstitute.com/seh-exploit/

Buffer Overflow 2

- https://medium.com/better-programming/an-introduction-to-buffer-overflow-vulnerability-760f23c21ebb
- https://medium.com/@musyokaian/buffer-overflow-101-356904169d94
- https://medium.com/@n0auth/buffer-overflows-0x01-67664959a256
- https://medium.com/@d0nut/week-13-introduction-to-buffer-overflows-5f15c0d5b5c1
- https://medium.com/@mtucunduva98/buffer-overflow-pcman-ftp-server-2-0-7-E143ff3473c
- https://medium.com/dev-genius/buffer-overflow-tutorial-part1-efc6b9f3e4ee
- https://medium.com/@rafaelrenovaci/buffer-overflow-slmail-5-5-fad2a67316dc
- https://medium.com/@chawdamrunal/lets-talk-about-buffer-overflow-54764101030b
- https://medium.com/@sghosh2402/understanding-exploiting-stack-based-buffer

Buffer Overflow 3

- https://github.com/gh0x0st/Buffer_Overflow
- https://github.com/justinsteven/dostackbufferoverflowgood
- https://github.com/muhammet-mucahit/Security-Exercises
- https://github.com/freddiebarrsmith/Buffer-Overflow-Exploit-Development-Practice
- https://github.com/V1n1v131r4/OSCP-Buffer-Overflow
- https://github.com/npapernot/buffer-overflow-attack
- https://github.com/hyperreality/OSCP-Buffer-Overflow-in-30-minutes
- https://github.com/yehiahesham/Buffer-Overflow-Attack
- https://github.com/hackutk/overflow-example
- https://github.com/JasonPap/Buffer-Overflows
- https://github.com/dievus/bufferoverflow
- https://github.com/EmreOvunc/Buffer-Overflow-PoC

Buffer Overflow 4

- https://www.youtube.com/watch?v=-KEN0I-G3qk
- https://datacellsolutions.com/2020/09/23/exploiting-a-simple-stack-based-buffer-overflow-vulnerability/
- https://techterms.com/definition/lifo#:~:text=Stands%20for%20%22Last%20In%2C%20First,order%20they%20have%20been%20entered
- https://www.geeksforgeeks.org/lifo-last-in-first-out-approach-in-programming/
- https://www.geeksforgeeks.org/fifo-vs-lifo-approach-in-programming/
- http://www-di.inf.puc-rio.br/~endler/courses/inf1612/aula-6.pdf
- https://www.youtube.com/watch?v=g7VazjrUWJ0&ab_channel=FernandoFresteiro
- https://www.youtube.com/watch?v=mRkb9BxRu4o&ab_channel=FernandoFresteiro
- https://sec4us.com.br/cheatsheet/

Buffer Overflow 5

- https://blog.rapid7.com/2019/06/12/heap-overflow-exploitation-on-windows-10-explained
- https://www.lume.ufrgs.br/bitstream/handle/10183/36924/000819136.pdf
- https://www.youtube.com/watch?v=oS2O75H57qU
- https://www.youtube.com/watch?v=TfJrU95q1J4&ab_channel=LiveOverflow
- https://www.youtube.com/watch?v=1S0aBV-Waeo&ab_channel=Computerphile
- https://www.youtube.com/watch?v=L8Ya7fBgEzU&ab_channel=BillyEllis
- https://en.wikipedia.org/wiki/Stack_buffer_overflow#:~:text=A%20stack%20buffer%20overflow%20can,is%20a%20potential%20security%20vulnerability.
- https://blog.rapid7.com/2019/02/19/stack-based-buffer-overflow-attacks-what-you-need-to-know/
- https://www.youtube.com/watch?v=hJ8IwyhqzD4
- https://stackoverflow.com/questions/30547811/read-a-non-atomic-variable-atomically
- https://preshing.com/20130618/atomic-vs-non-atomic-operations
- https://www.youtube.com/watch?v=5g137gsB9Wk
- https://www.youtube.com/watch?v=1hScemFvnzw
- https://www.youtube.com/watch?v=pTYHloclHEQ&ab_channel=TurkyGary

Dev Exploit

- https://medium.com/@fahri.shihab/structured-exception-handling-seh-buffer-overflow-e809cb7d0e5d
- https://www.exploit-db.com/docs/english/17505-structured-exception-handler-exploitation.pdf
- https://m0chan.github.io/2019/08/21/Win32-Buffer-Overflow-SEH.html
- https://www.corelan.be/index.php/2009/07/25/writing-buffer-overflow-exploits-a-quick-and-basic-tutorial-part-3-seh/
- https://www.youtube.com/watch?v=UVtXaDtIQpg
- https://www.youtube.com/watch?v=Znrvsf8Trvg&ab_channel=StefanMolls
- https://www.coalfire.com/the-coalfire-blog/march-2020/the-basics-of-exploit-development-2-seh-overflows
- https://fuzzysecurity.com/tutorials/expDev/3.html
- https://sec4us.com.br/cheatsheet/bufferoverflow-seh
- https://www.rapid7.com/resources/structured-exception-handler-overwrite-explained/
- https://techjoomla.com/blog/beyond-joomla/seh-buffer-overflow-exploitation-using-egghunter-payload
- https://medium.com/@notsoshant/windows-exploitation-egg-hunting-117828020595

Dev Exploit 2

- https://www.corelan.be/index.php/2010/01/09/exploit-writing-tutorial-part-8-win32-egg-hunting/
- https://www.corelan.be/index.php/2019/04/23/windows-10-egghunter/
- https://www.exploit-db.com/docs/english/18482-egg-hunter---a-twist-in-buffer-overflow.pdf
- https://blog.rapid7.com/2012/07/06/an-example-of-egghunting-to-exploit-cve-2012-0124/
- https://github.com/rhamaa/Binary-exploit-writeups
- https://www.shogunlab.com/blog/2017/09/02/zdzg-windows-exploit-3.htm

Shellcode Dev

- https://www.ime.usp.br/~adao/CI1.pdf
- https://silva97.gitbook.io/assembly-x86/a-base/instrucoes
- https://ivanitlearning.wordpress.com/2018/10/13/windows-32-bit-shellcoding-101/
- http://mcdermottcybersecurity.com/articles/windows-x64-shellcode
- https://www.tophertimzen.com/blog/windowsx64Shellcode/
- https://www.exploit-db.com/exploits/40549
- https://nytrosecurity.com/2019/06/30/writing-shellcodes-for-windows-x64/
- https://github.com/peterferrie/win-exec-calc-shellcode
- https://resources.infosecinstitute.com/shellcode-analysis-on-linux-x86-32bit/
- https://stackoverflow.com/questions/61023648/how-to-execute-32-bit-shellcode-on-a-64-bit-linux-system
- https://vividmachines.com/shellcode/shellcode.html
- http://www.lia.deis.unibo.it/Courses/SicurezzaM1920/lab/lab03_getting_a_shell.pdf
- https://pen-testing.sans.org/resources/papers/gcih/stage-attack-one-way-shellcode

Shellcode Dev 2

- https://www.offensive-security.com/metasploit-unleashed/payload-types/
- https://buffered.io/posts/staged-vs-stageless-handlers/
- https://zerosum0x0.blogspot.com/2014/12/x64-egg-hunter-shellcode.html
- https://securityboulevard.com/2020/02/evading-antivirus-with-better-meterpreter-payloads/
- https://www.exploit-db.com/papers/35646
- https://www.blackhat.com/presentations/bh-usa-08/Miller/BH_US_08_Ty_Miller_Reverse_DNS_Tunneling_Shellcode.pdf
- https://www.ired.team/offensive-security/code-injection-process-injection/loading-and-executing-shellcode-from-portable-executable-resources
- https://slazarus.net/analysing-msfvenom-shellcode/
- https://www.researchgate.net/figure/Architecture-of-an-emulation-based-shellcode-detector_fig1_274570378
- https://www.virtuesecurity.com/evading-antivirus-with-better-meterpreter-payloads

Shellcode Dev 3

- https://netsec.ws/?p=331
- http://slae-581.blogspot.com/p/assignment5-metasploit-shellcodes.html
- https://medium.com/@vikrant.navalgund/encoded-obfuscated-shellcode-securitytube-linux-assembly-expert-32-bit-exercise-4-568c5a18149a
- https://n3k00n3.github.io/blog/04052017/Shellcode.html
- https://rastating.github.io/creating-a-custom-shellcode-encoder/
- https://medium.com/syscall59/writing-a-custom-shellcode-encoder-31816e767611
- https://www.rcesecurity.com/2015/01/slae-custom-rbix-shellcode-encoder-decoder/
- https://snowscan.io/custom-encoder/
- https://github.com/Potato-Industries/custom-shellcode-encoder-decoder

Debuggers

- https://www.mentebinaria.com.br/noticias/gdb-de-gente-grande-conhe%C3%A7a-o-gef-r331/
- https://darkdust.net/files/GDB%20Cheat%20Sheet.pdf
- http://users.ece.utexas.edu/~adnan/gdb-refcard.pdf
- http://www.ollydbg.de/odbg64.html
- https://error4hack.com/x64dbg-tutorial/
- https://www.corelan.be/index.php/search/Cheat+Sheet/
- https://www.sans.org/reading-room/whitepapers/malicious/paper/36982
- http://index-of.es/Varios-2/Using%20Immunity%20Debugger%20to%20Write%20Exploits.pdf
- https://blog.hackerenv.com/buffer-overflow-tutorial-step-by-step-with-immunity-debugger-3

Extra Resources

- https://www.exploit-db.com/shellcodes
- https://securitycafe.ro/2015/10/30/introduction-to-windows-shellcode-development-part1
- https://www.youtube.com/watch?v=74Y_w2_MgpY&ab_channel=T3jv1l
- https://www.youtube.com/watch?v=FmCDVwA5kYQ&ab_channel=GuidedHacking
- https://bufferoverflows.net/developing-custom-shellcode-x64-linux/
- https://medium.com/@aayushmalla56/shellcode-development-4590117a26bf
- https://www.youtube.com/watch?v=2giDgeXpJE0&list=PLWHiAJhsj4eXi1AF6N5MYz61RcwSCoVO8 (Assembly)
- https://www.youtube.com/watch?v=oZeezrNHxVo&list=PLIfZMtpPYFP5qaS2RFQxcNVkmJLGQwyKE

Extra Resources 2

- https://www.youtube.com/watch?v=RF3-qDy-xMs&list=PLIfZMtpPYFP6_YOrfX79YX79I5V6mS0ci
- https://pt.wikipedia.org/wiki/Gerenciamento_de_mem%C3%B3ria
- https://null-byte.wonderhowto.com/how-to/exploit-development-everything-you-need-know-0167801/
- https://www.tutorialspoint.com/assembly_programming
- https://www.cs.virginia.edu/~evans/cs216/guides/x86.html
- http://spot.pcc.edu/~wlara/asmx86/asmx86_manual_4.pdf