FUNDAMENTOS DE DESENVOLVIMENTO DE EXPLOITS - OVERVIEW

SOBRE O LIVRO

- Um overview dos principais conceitos de desenvolvimento de exploits;
- Conceitos fundamentais;
- Material de estudo para se aprofundar;

SOBRE O AUTOR

- Joas Antonio
- Entusiasta em Cibersegurança
- https://www.linkedin.com/in/joas-antonio-dos-santos/

CONCEITOS E FUNDAMENTOS

O QUE É UM EXPLOIT?

- É um pedaço de script desenvolvido para explorar uma determinada brecha de segurança;
- Exploits consiste em shellcodes e um pedaço de código para inserir em uma aplicação vulnerável;

SHELLCODE

- O Shellcode é definido como um conjunto de instruções injetadas e depois executadas por um exploit. O Shellcode é usado para manipular diretamente os registros e as funcionalidade de um exploit, garantindo até mesmo uma shell na máquina alvo, sendo seu principal próposito e muitos atrelando o codename Shell para se referir a isso, mas talvez passe a ser apenas uma ideia.
- No bufffer overflow são códigos utilizados na exploração, utilizados no desenvolvimento de exploits para exploração de Buffer Overflow. Quem já analisou os exploits de buffer overflows já os viu, shellcodes são construídos apenas com os valores em hexadecimal dos opcodes da arquitetura alvo, ou seja, as instruções do próprio processador, por isso o entendimento da linguagem assembly, que até certo ponto, possui relação de 1 para 1 com a linguagem de máquina, se faz necessária. O shellcode é o código que será de fato executado durante a exploração de um buffer overflow. São chamados de 'shellcodes' pois geralmente o seu objetivo é a obtenção de uma shell.

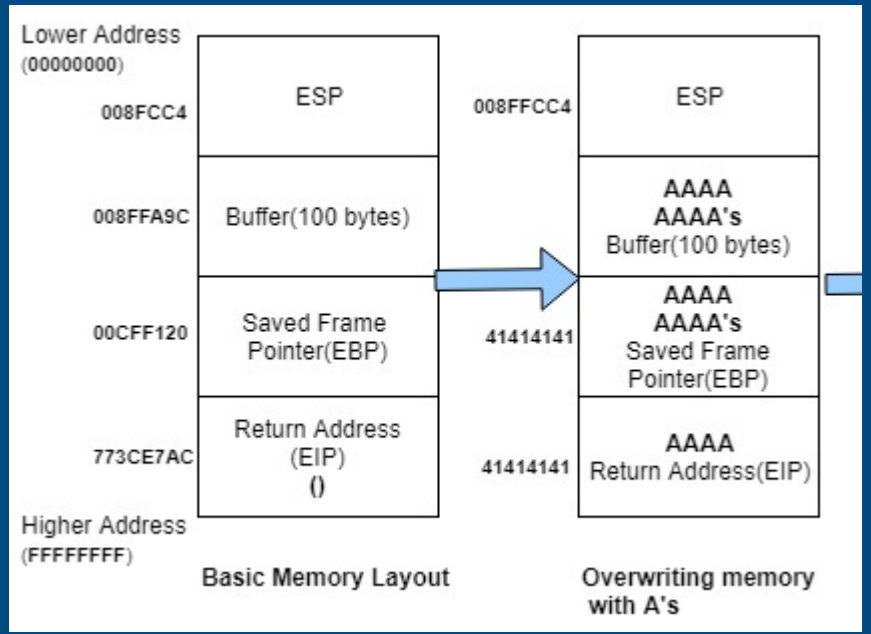
Pré-requisitos de desenvolvimento de exploits e shellcode

- Conhecimentos em Linguagem C;
- Conhecimentos em Assembly x86 and x64;
- Gerenciamento de memória e endereços de sistemas;
- Conhecimentos em Buffer Overflow;
- Conhecimentos em Engenharia Reversa;
- Conhecimentos em Registradores;
- Conceitos de mecanismos de proteção de Software (DEP and ASLR);
- Conceitos de Fuzzing;

STACK OVERFLOW EXPLOITS

- Um ataque de Buffer Overflow ocorre quando um dado superdimensionado é escrito no stack buffer de um processador;

- O Buffer Overflow pode substituir os dados de fluxo do programa e outras variáveis;

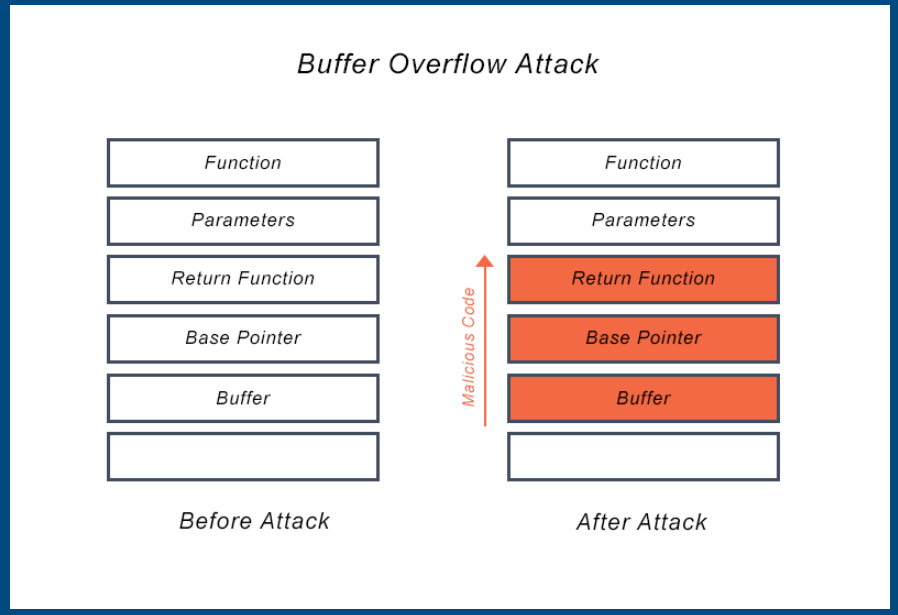


https://priyasloka.wordpress.com/2018/04/13/buffer-overflow-exploit-part-3/

HEAP CORRUPTION EXPLOITS

- A corrupção de heap ocorre quando a área de memória heap não tem espaço suficiente para os dados sendo gravados nela

- A memória heap é usada dinamicamente pelo aplicativo em tempo de execução



https://avinetworks.com/glossary/buffer-overflow/

Format String Attack

Código Seguro

A linha `printf("%s", argv[1]);`no exemplo é segura, se você compilar o programa e executá-lo:

```
./example "Hello World %s%s%s%s%s%s"
```

O `printf`na primeira linha não interpretará "% s% s% s% s% s% s" na string de entrada, e a saída será: "Hello World% s% s% s% s% s% s"

Código Vulnerável

A linha `printf(argv[1]);`no exemplo é vulnerável, se você compilar o programa e executá-lo:

```
./example "Hello World %s%s%s%s%s%s"
```

O `printf`na segunda linha interpretará o `%s%s%s%s%s%s`na string de entrada como uma referência a ponteiros de string, então ele tentará interpretar cada% s como um ponteiro para uma string, começando da localização do buffer (provavelmente na pilha). Em algum ponto, ele obterá um endereço inválido e a tentativa de acessá-lo causará o travamento do programa.

Diferentes cargas úteis

Um invasor também pode usar isso para obter informações, não apenas para travar o software. Por exemplo, executando:

```
./example "Hello World %p %p %p %p %p %p"
```

Irá imprimir as linhas:

```
Hello World %p %p %p %p %p %p
Hello World 000E133E 000E133E 0057F000 CCCCCCCC CCCCCCCC CCCCCCCC
```

```c
#include  <stdio.h>
void main(int argc, char **argv)
{
    // This line is safe
    printf("%s\n", argv[1]);

    // This line is vulnerable
    printf(argv[1]);
}
```

https://owasp.org/www-community/attacks/Format_string_attack

Integer Bug Exploits

- Bugs inteiros são explorados passando um inteiro superdimensionado para um inteiro variável

- Isso pode causar a substituição de dados de controle de programa válidos, resultando em execução de códigos maliciosos

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

Race Conditions

- Condição de corrida é uma vulnerabilidade de software que ocorre quando múltiplas acessos ao recurso compartilhado não são controlados adequadamente;
- Tipos de ataques de condição de corrida:
  - **Condição de corrida de arquivo:** Ocorre quando o atacante explora um condição não-atômica de maneira cronometrada criando, escrevendo, lendo e apagando um arquivo e etc, em um diretório temporário;
  - **Signal Race Condition**: As condições de corrida de tratamento de **sinal** podem ocorrer sempre que uma função instalada como um manipulador de **sinal** não é reentrante, o que significa que ela mantém algum estado interno ou chama outra função que o faz;

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

Socket Binding Exploits

- **Socket do lado do cliente**: Envolve escrever o código para conectando o aplicativo a um servidor remoto;
- **Socket do lado do servidor**: Envolve escrever o código para escutar em uma porta e processar as entrada de conexões;

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-intel-edison/

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

Metasploit

- O software Metasploit ajuda a identificar problemas de segurança, verificando vulnerabilidades, auxiliando na mitigações e na avaliações de segurança orientadas;
- O Metasploit framework fornece soluções avançadas para pesquisadores que precisam de um alternativa para desenvolver suas próprias ferramentas de PenTest;

https://www.offensive-security.com/metasploit-unleashed/exploit-development/

https://www.offensive-security.com/metasploit-unleashed/exploit-development-goals/

https://www.youtube.com/watch?v=0iXX5W85HXo&ab_channel=GusKhawaja

https://www.youtube.com/watch?v=YQj8FaQ8B5w&ab_channel=KacperSzurekEN

https://www.offensive-security.com/metasploit-unleashed/building-module/

Passo a passo básico para desenvolver um exploit

1. Identificar e analisar os bugs da aplicação;
2. Escreva o código para manipular e controlar a memória do alvo;
3. Redirecione o fluxo de execução;
4. Injete o Shellcode;
5. Criptografe a comunicação do seu socket;

Diferenças entre Exploits Windows e Linux

Windows:

- Explora funções de chamada exportadas por bibliotecas de vínculo dinâmico;
- Exploits escritos para Windows sobrescreve o retorno endereços na pilha com um endereço que contém instrução “jmp reg” onde reg representa registro;

Linux:

- Exploits Linux usa system calls;
- Os exploits substituem os endereço salvos de retorno com uma pilha de endereço onde os dados fornecidos de um usuário podem ser encontrados;

Shellcodes e seus tipos

- Shellcodes são conjuntos de instruções usadas por programas de exploração para realizar a função desejada;
- Eles são executados depois que uma vulnerabilidade é explorada;
- As instruções da máquina são usadas para processar diretamente a instrução desejada no local da memória;
- Estas instruções de máquina consistem em opcodes;

Remote Shellcode:

- Port Binding Shellcode
- Socket Descriptor Reuse Shellcode

Local Shellcode:

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

Shellcodes Tools

- NASM (https://www.nasm.us/);
- Sickle (https://github.com/wetw0rk/Sickle)
- https://github.com/tophertimzen/shellcodeTester
- https://github.com/helviojunior/shellcodetester
- https://github.com/xapax/shellcode-tools
- https://github.com/MarioVilas/shellcode_tools
- https://www.immunityinc.com/products/debugger/
- https://x64dbg.com/#start

Passo a passo para escrever um shellcode

- Escreva o código em linguagem Assembly ou em linguagem C e faça um disassembler;
- Colete os args e syscall id;
- Converta seu código assembly em opcodes;
- Elimine os null bytes;
- Inicie uma shell;
- Compile e execute;
- Rastreie o código;
- Injete o shellcode na inicialização do programa;
- https://www.linkedin.com/pulse/desenvolvendo-um-simples-shellcode-e-explorando-uma-dos-santos/

Problemas envolvidos no desenvolvimento de shellcodes

- Problema de endereços;
- Problema de Null Bytes;
- Implementação do System Call;
- https://reverseengineering.stackexchange.com/questions/9184/the-addressing-problem
- https://reverseengineering.stackexchange.com/questions/8504/problem-finding-return-address-for-shellcode
- https://null-byte.wonderhowto.com/how-to/writing-64-bit-shellcode-part-2-removing-null-bytes-0161591/
- https://stackoverflow.com/questions/9776889/null-bytes-in-shellcode#:~:text=The%20problem%20with%20null%20bytes,copying%20shall%20stop%20on%20it.&text=It%20anyway%20is%20null%2Dsafe,but%20exploitable%20in%20another%20way).
- 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 básico – Conceitos e fundamentos

- 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%ADvel-Programas-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

Engenharia Reversa – Conceitos e fundamentos

- https://mentebinaria.gitbook.io/engenharia-reversa/assembly/funcoes-e-pilha
- https://ic.unicamp.br/~ducatte/mc404/Slides/mc404_07_2s06.pdf
- https://www.techtudo.com.br/noticias/noticia/2015/02/o-que-e-o-codigo-ascii-e-para-que-serve-descubra.html#:~:text=O%20nome%20ASCII%20vem%20do,e%20alguns%20c%C3%B3digos%20de%20controle.
- https://www.youtube.com/watch?v=IN9EIO90uLc&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

Engenharia Reversa – Conceitos e fundamentos

- 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://medium.com/@fahri.shihab/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 – Conceitos e fundamentos

- 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-review-7001bfdf335e
- https://scorpiosoftware.net/2020/01/03/next-windows-internals-remote-training/
- https://www.youtube.com/watch?v=qMWvqdtIbkQ

Buffer Overflow – Conceitos & Exercícios

- https://drive.google.com/file/d/1poocO7AOMyBQBtDXvoaZ2dgkq3Zf1Wlb/view?usp=sharing (my ebook)
- 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=PLcKsaFvYI4I87C8_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 - Conceitos & Exercícios

- 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-overflows-acf9b8659cba

Buffer Overflow - Conceitos & Exercícios

- 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 - Conceitos & Exercícios

- 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/

Dev Exploit - Conceitos & Exercícios

- 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 - Conceitos & Exercícios

- 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.html

Shellcode Dev - Conceitos & Exercícios

- 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-106319

Shellcode Dev - Conceitos & Exercícios

- 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 - Conceitos & Exercícios

- 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/

Evasion AV

- https://null-byte.wonderhowto.com/how-to/evading-av-software/
- https://www.hackers-arise.com/evading-av-with-shellter/
- https://securityboulevard.com/2020/02/evading-antivirus-with-better-meterpreter-payloads/
- https://medium.com/@offs3cg33k/antivirus-evasion-bypass-techniques-b547cc51c371
- https://blog.rapid7.com/2018/05/03/hiding-metasploit-shellcode-to-evade-windows-defender/
- https://searchsecurity.techtarget.com/feature/Antivirus-evasion-techniques-show-ease-in-avoiding-antivirus-detection
- https://www.blackhat.com/docs/us-16/materials/us-16-Bulazel-AVLeak-Fingerprinting-Antivirus-Emulators-For-Advanced-Malware-Evasion.pdf
- https://www.blackhat.com/docs/us-15/materials/us-15-Xenakis-ROPInjector-Using-Return-Oriented-Programming-For-Polymorphism-And-Antivirus-Evasion.pdf
- https://www.blackhat.com/docs/us-17/thursday/us-17-Jung-AVPASS-Leaking-And-Bypassing-Anitvirus-Detection-Model-Automatically.pdf

Evasion AV

- https://i.blackhat.com/us-18/Thu-August-9/us-18-Bulazel-Windows-Offender-Reverse-Engineering-Windows-Defenders-Antivirus-Emulator.pdf
- https://www.blackhat.com/docs/us-14/materials/arsenal/us-14-Schroeder-The-Veil-Framework-Slides.pdf
- https://www.mike-gualtieri.com/posts/modifying-empire-to-evade-windows-defender
- https://www.blackhillsinfosec.com/getting-powershell-empire-past-windows-defender/
- https://github.com/ciccio-87/Python-AV-Evasion
- https://github.com/nccgroup/metasploitavevasion
- https://www.youtube.com/watch?v=MO11gJ-WJqY&ab_channel=BlackHat https://www.youtube.com/watch?v=2HNuzUuVyv0&ab_channel=BlackHat
- https://www.youtube.com/watch?v=wDNQ-8aWLO0&ab_channel=BlackHat

CONTEÚDOS EXTRAS

- 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=pTYHlocIHEQ&ab_channel=TurkyGary

CONTEÚDOS EXTRAS 2

- 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

CONTEÚDOS EXTRAS 3

- 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

Conclusão

- Esses foram alguns conceitos e materiais para se aprofundar na área de desenvolvimento de exploits;
- Eu recomendo que antes de tudo você pegue uma base, entenda como o sistema funciona e como ele interage com recursos físicos e lógicos;
- Pratique bastante e faça exercícios para melhorar as suas habilidades em desenvolvimento de exploits;
- A base é essencial para quem está começando, caso você queira pegar outros conceitos, possuo alguns e-books que pode auxiliar: https://drive.google.com/drive/folders/12Mvq6kE2HJDwN2CZhEGWizyWt87YunkU
- Caso procure algum curso, eu recomendo a Sec4us, Acadi-TI, Offensive Security, eLearn Security, EC-COUNCIL, Gohacking que possuem conteúdos mais avançados;

Obrigado