Shellcode Development

Joas Antonio

Details

- This is a basic pdf about shellcode development, the aim is to provide an overview
- https://www.linkedin.com/in/joas-antonio-dos-santos

Introduction

- O shellcode é um código executado na exploração de vulnerabilidades, como por exemplo buffer overflow. Tradicionalmente, o shellcode dá ao atacante acesso à uma shell no computador explorado por meio de conexão reversa. Porém, ultimamente existem shellcodes que fazem ações mais complexas e variadas.
- Shellcode is code designed to exploit vulnerabilities such as buffer overflow. Traditionally, shellcode gives an attacker access to the shell on the exploited computer via a reverse connection. However, lately there are shellcodes that perform more complex and varied actions.

Introduction

- https://en.wikipedia.org/wiki/Shellcode
- https://searchsecurity.techtarget.com/answer/What-is-the-relationship-between-shellcode-and-exploit-code
- https://www.sentinelone.com/blog/malicious-input-how-hackers-use-shellcode/
- https://www.pcmag.com/encyclopedia/term/shellcode
- https://www.contextis.com/en/blog/a-beginners-guide-to-windows-shellcode-execution-techniques

Shellcode Types - Local

- *Local* shellcode is used by an attacker who has limited access to a machine but can exploit a vulnerability, for example a buffer overflow, in a higher-privileged process on that machine. If successfully executed, the shellcode will provide the attacker access to the machine with the same higher privileges as the targeted process.

Shellcode Types - Remote

- *Remote* shellcode is used when an attacker wants to target a vulnerable process running on another machine on a local network, intranet, or a remote network. If successfully executed, the shellcode can provide the attacker access to the target machine across the network. Remote shellcodes normally use standard TCP/IP socket connections to allow the attacker access to the shell on the target machine. Such shellcode can be categorized based on how this connection is set up: if the shellcode establishes the connection, it is called a "reverse shell" or a *connect-back* shellcode because the shellcode *connects back* to the attacker's machine. On the other hand, if the attacker establishes the connection, the shellcode is called a *bindshell* because the shellcode *binds* to a certain port on the victim's machine. There's a peculiar shellcode named *bindshell random port* that skips the binding part and listens on a random port made available by the operating system. Because of that the *bindshell random port* became the smallest and stable bindshell shellcode for x86_64 available to this date. A third, much less common type, is *socket-reuse* shellcode. This type of shellcode is sometimes used when an exploit establishes a connection to the vulnerable process that is not closed before the shellcode is run. The shellcode can then *re-use* this connection to communicate with the attacker. Socket re-using shellcode is more elaborate, since the shellcode needs to find out which connection to re-use and the machine may have many connections open.[6]
- A firewall can be used to detect outgoing connections made by connect-back shellcode as well as incoming connections made by bindshells. They can therefore offer some protection against an attacker, even if the system is vulnerable, by preventing the attacker from connecting to the shell created by the shellcode. This is one reason why socket re-using shellcode is sometimes used: it does not create new connections and therefore is harder to detect and block.

Shellcode Types – Download and Execute

- *Download and execute* is a type of remote shellcode that *downloads and executes* some form of malware on the target system. This type of shellcode does not spawn a shell, but rather instructs the machine to download a certain executable file off the network, save it to disk and execute it. Nowadays, it is commonly used in drive-by download attacks, where a victim visits a malicious webpage that in turn attempts to run such a download and execute shellcode in order to install software on the victim's machine. A variation of this type of shellcode downloads and loads a library.[7][8] Advantages of this technique are that the code can be smaller, that it does not require the shellcode to spawn a new process on the target system, and that the shellcode does not need code to clean up the targeted process as this can be done by the library loaded into the process.

Shellcode Types – Staged

- When the amount of data that an attacker can inject into the target process is too limited to execute useful shellcode directly, it may be possible to execute it in stages. First, a small piece of shellcode (stage 1) is executed. This code then downloads a larger piece of shellcode (stage 2) into the process's memory and executes it.

Shellcode Types – Egghunt

- This is another form of *staged* shellcode, which is used if an attacker can inject a larger shellcode into the process but cannot determine where in the process it will end up. Small *egg-hunt* shellcode is injected into the process at a predictable location and executed. This code then searches the process's address space for the larger shellcode (the *egg*) and executes it.

Shellcode Types – Omelette

- This type of shellcode is similar to *egg-hunt* shellcode, but looks for multiple small blocks of data (*eggs*) and recombines them into one larger block (the *omelette*) that is subsequently executed. This is used when an attacker can only inject a number of small blocks of data into the process.
- https://en.wikipedia.org/wiki/Shellcode
- https://security.stackexchange.com/questions/189399/what-is-the-use-of-different-types-of-shellcode

Shellcode Encode

- https://lastlistener.github.io/Courses--SLAE--4_Shellcode_Encoding_Scheme.html
- https://snowscan.io/custom-encoder/
- https://cybersecpolitics.blogspot.com/2019/03/the-lost-art-of-shellcode.html
- https://www.rcesecurity.com/2015/01/slae-custom-rbix-shellcode-encoder-decoder/
- https://modexp.wordpress.com/2020/06/26/shellcode-encoding-null-bytes-faster/
- https://securitychops.com/2018/05/27/slae-assignment-4-custom-encoder.html
- https://www.programmersought.com/article/27852345566/
- https://github.com/Potato-Industries/custom-shellcode-encoder-decoder
- https://github.com/DarkCoderSc/slae32-xor-encoder
- https://github.com/rcx/shellcode_encoder
- https://github.com/SkyLined/alpha3
- https://rastating.github.io/creating-a-custom-shellcode-encoder/
- https://github.com/EAugustoAnalysis/Automatic-ASCII-Shellcode-Subtraction-Encoder
- https://github.com/Arno0x/ShellcodeWrapper
- https://github.com/dhrumil29699/Printable-Encoder
- https://github.com/wr34k/shellcode-encode

Shellcode Encode - Types

- Because most processes filter or restrict the data that can be injected, shellcode often needs to be written to allow for these restrictions. This includes making the code small, null-free or alphanumeric. Various solutions have been found to get around such restrictions, including:
- Design and implementation optimizations to decrease the size of the shellcode.
- Implementation modifications to get around limitations in the range of bytes used in the shellcode.
- Self-modifying code that modifies a number of the bytes of its own code before executing them to re-create bytes that are normally impossible to inject into the process.
- Since intrusion detection can detect signatures of simple shellcodes being sent over the network, it is often encoded, made self-decrypting or polymorphic to avoid detection.

Shellcode Encode – Percent Encoding

- https://en.wikipedia.org/wiki/Percent-encoding
- https://www.liquisearch.com/shellcode/shellcode_encoding/percent_encoding
- https://dbpedia.org/page/Percent-encoding

Shellcode Encode – Null-free shellcode

- https://www.exploit-db.com/exploits/37362
- https://stackoverflow.com/questions/6853945/null-free-shellcode
- https://nets.ec/Shellcode/Null-free
- https://packetstormsecurity.com/files/156478/Windows-x86-Null-Free-WinExec-Calc.exe-Shellcode.html
- https://security.stackexchange.com/questions/91969/removing-null-bytes-from-shell-code
- https://iphelix.medium.com/corelan-tutorial-9-exercise-solution-6c1ba682bf?source=post_internal_links---------0----------------------

Shellcode Encode – Alphanumeric shellcode

- https://en.wikipedia.org/wiki/Alphanumeric_shellcode
- https://www.offensive-security.com/metasploit-unleashed/alphanumeric-shellcode/
- https://dl.packetstormsecurity.net/papers/shellcode/alpha.pdf
- https://blackcloud.me/Linux-shellcode-alphanumeric/
- https://www.usenix.org/system/files/woot20-paper-patel_0.pdf
- https://nets.ec/Alphanumeric_shellcode
- https://nets.ec/Ascii_shellcode
- https://packetstormsecurity.com/files/155844/Linux-x86-Execve-Alphanumeric-Shellcode.html
- https://vishnudevtj.github.io/notes/arm-alphanumeric-shellcode
- https://h0mbre.github.io/LTER_SEH_Success/
- https://flylib.com/books/en/1.545.1.58/1/
- https://www.youtube.com/watch?v=5mzdfzMKah0

Shellcode Encode – Unicode proof shellcode

- http://phrack.org/issues/61/11.html
- https://ieeexplore.ieee.org/document/6013556
- https://paper.seebug.org/papers/old_sebug_paper/pst_WebZine/pst_WebZine_0x01/html/%5BPSTZine_0x01%5D%5B0x02%5D%5BAn-improvement-on-mixed-case-alphanumeric-shellcode-decoder%5D.html
- https://www.corelan.be/index.php/2009/11/06/exploit-writing-tutorial-part-7-unicode-from-0x00410041-to-calc/
- https://www.securitysift.com/windows-exploit-development-part-7-unicode-buffer-overflows/
- https://www.semanticscholar.org/paper/Unicode-proof-code-injection-attack-on-Windows-CE-%E2%80%94-Song-Zhang/bad47d590e073540ca2277c791e127d2a3fbb482/figure/2
- https://www.coalfire.com/the-coalfire-blog/july-2020/the-basics-of-exploit-development-4-unicode-overfl
- https://movaxbx.ru/category/shellcode/

Generating Shellcode

- https://www.offensive-security.com/metasploit-unleashed/generating-payloads/
- https://www.offensive-security.com/metasploit-unleashed/alphanumeric-shellcode/
- https://www.rapid7.com/blog/post/2018/05/03/hiding-metasploit-shellcode-to-evade-windows-defender/
- https://medium.com/@PenTest_duck/offensive-msfvenom-from-generating-shellcode-to-creating-trojans-4be10179bb86
- https://blog.f-secure.com/dynamic-shellcode-execution/
- https://blog.cobaltstrike.com/2014/02/12/modifying-metasploits-stager-shellcode/
- https://resources.infosecinstitute.com/topic/shellcode-and-metasploit/
- https://www.trendmicro.com/en_us/research/20/j/metasploit-shellcodes-attack-exposed-docker-apis.html
- https://github.com/r00t-3xp10it/venom
- https://www.hackingloops.com/venom-shellcode-payload-generator/
- https://netsec.ws/?p=331
- https://pentesttools.net/venom-1-0-15-metasploit-shellcode-generator-compiler-listener/

Generating Shellcode

- http://www.ethicalpentest.com/2018/01/generating-shellcode-using-msfvenom.html
- https://www.youtube.com/watch?v=rssv0rZp9p8
- https://www.youtube.com/watch?v=PSdyrkOeuNY
- https://www.youtube.com/watch?v=K864mSQbsdQ
- https://www.youtube.com/watch?v=qSjxR8tfokg
- https://www.youtube.com/watch?v=OqxSD2Lnqq0
- https://www.youtube.com/watch?v=q6f-uvIlXq0
- https://www.youtube.com/watch?v=m10xuTv02f8
- https://www.youtube.com/watch?v=uwLbJM64Rkg

Writer Shellcode

- https://www.youtube.com/watch?v=rvZsvSH2pXo
- https://hackerculture.com.br/?p=1059
- https://medium.com/syscall59/writing-a-custom-shellcode-encoder-31816e767611
- https://www.youtube.com/watch?v=nSR3U0Pzsb8
- http://www.securitytube.net/video/7042
- https://www.youtube.com/watch?v=EJJI1AW3sfQ
- https://medium.com/syscall59/a-trinity-of-shellcode-aes-go-f6cec854f992
- https://www.exploit-db.com/docs/english/17065-manual-shellcode.pdf

Writer Shellcode

- http://www.dmi.unipg.it/bista/didattica/sicurezza-pg/buffer-overrun/hacking-book/0x2a0-writing_shellcode.html
- https://null-byte.wonderhowto.com/how-to/writing-64-bit-shellcode-part-1-beginner-assembly-0161593/
- https://medium.com/@songchai.d01/basics-of-windows-shellcode-writing-7465f329cf19
- https://medium.com/@zlkidda/day-0-quest-for-my-first-zero-day-writing-my-shell-code-847f493b71d0
- https://www.corelan.be/index.php/2010/02/25/exploit-writing-tutorial-part-9-introduction-to-win32-shellcoding/
- https://github.com/reyammer/shellnoob

Assembly x86

- https://rudamoura.com/x86.html
- https://en.wikipedia.org/wiki/X86_assembly_language
- https://www.cin.ufpe.br/~eaa3/Arquivos/Assembly/Assembly%20x86%20NASM.pdf
- https://www.youtube.com/watch?v=wLXIWKUWpSs
- https://www.youtube.com/watch?v=HgEGAaYdABA
- https://www.youtube.com/watch?v=75gBFiFtAb8
- https://www.youtube.com/watch?v=dkjfZyJvO0I
- https://www.cs.virginia.edu/~evans/cs216/guides/x86.html
- https://www.revista-programar.info/artigos/iniciacao-ao-assembly-x86-aspectos-teoricos/
- https://bitismyth.wordpress.com/2019/05/10/dicas-de-assembly-para-o-x86-64/

Assembly x86

- https://en.wikibooks.org/wiki/X86_Assembly
- https://www.comp.uems.br/~ojacques/LM/Livros/Linguagem%20Assembly%20para%20i386%20e%20x86-64%20v0.8.5.pdf
- https://gitbook.ganeshicmc.com/engenharia-reversa/hello-world-em-x86
- https://github.com/Silva97/livro-assembly-x86

Assembly x64

- https://software.intel.com/content/www/us/en/develop/articles/introduction-to-x64-assembly.html
- https://www.youtube.com/watch?v=rxsBghsrvpI
- https://www.youtube.com/watch?v=Dh7GQ_joeE4
- https://sonictk.github.io/asm_tutorial/
- https://cs.brown.edu/courses/cs033/docs/guides/x64_cheatsheet.pdf
- https://docs.microsoft.com/pt-br/cpp/assembler/masm/masm-for-x64-ml64-exe?view=msvc-160
- https://gpfault.net/posts/asm-tut-0.txt.html
- https://github.com/0xAX/asm
- https://github.com/Apress/beginning-x64-assembly-programming
- https://exercism.io/tracks/x86-64-assembly
- https://www.cs.uaf.edu/2017/fall/cs301/reference/x86_64.html
- https://www.youtube.com/watch?v=guru397zg2g

Assembly – MOV AV, BX

- http://www.facom.ufu.br/~gustavo/OC1/Apresentacoes/Assembly.pdf
- https://stackoverflow.com/questions/65517145/what-mov-ax-bx-actually-does
- http://www.sce.carleton.ca/courses/sysc-3006/s13/Lecture%20Notes/Part5-SimpleAssembly.pdf
- http://www.telecom.uff.br/orgarqcomp/Slides/OrgComp_pt5_Conjunto_de_Instrucoes_8086-88.pdf
- https://web.uettaxila.edu.pk/CMS/AUT2010/cpMSbs/notes/Lecture%203.pdf
- https://www.tutorialspoint.com/microprocessor/microprocessor_8086_addressing_modes.htm
- http://marco.uminho.pt/~joao/Computacao2/node33.html
- https://ruc.edu.iq/wp-content/uploads/2021/02/L3.pdf
- http://www.husseinsspace.com/teaching/udw/1996/asmnotes/chaptwo.htm

Assembly – Instructions

- https://www.tutorialspoint.com/assembly_programming/assembly_logical_instructions.htm
- https://www.tutorialspoint.com/assembly_programming/assembly_logical_instructions.htm
- https://www.youtube.com/watch?v=Wz_xJPN7IAY
- https://www.youtube.com/watch?v=nHq0E7jt7fU
- https://flint.cs.yale.edu/cs421/papers/x86-asm/asm.html
- https://docs.oracle.com/cd/E19120-01/open.solaris/817-5477/ennby/index.html
- https://www.allaboutcircuits.com/technical-articles/how-to-write-assembly-basic-assembly-instructions-ARM-instruction-set/
- https://www.keil.com/support/man/docs/armasm/armasm_dom1361289863017.htm
- http://www.ece.utep.edu/courses/web3376/Notes_files/ee3376-assembly.pdf

Assembly – Operators

- https://www.zippia.com/assembly-operator-jobs/what-does-an-assembly-operator-do/
- https://www.jobhero.com/job-description/examples/production/assembly-operator
- https://www.automotiveip.co.uk/wp-content/uploads/sites/7/2016/01/Operator-Assembly-Generic-JD.pdf
- https://docs.oracle.com/cd/E19120-01/open.solaris/817-5477/eoqjy/index.html
- http://www.inf.furb.br/~maw/arquitetura/aula16.pdf
- http://www.estgv.ipv.pt/paginaspessoais/acarv/200607/AC/Docs/act_OpL_ExL_OpR.pdf
- https://insper.github.io/Z01.1/Util-Resumo-Assembly/
- https://www.docsity.com/pt/assembly-cap7/4806802/
- https://multilogica-shop.com/Referencia/OperadoresDeBits_AND_OR_XOR

Memory Management

- https://en.wikipedia.org/wiki/Memory_management#:~:text=Memory%20management%20is%20a%20form,reuse%20when%20no%20longer%20needed.
- https://www.tutorialspoint.com/operating_system/os_memory_management.htm
- https://en.wikipedia.org/wiki/Memory_management_(operating_systems)
- https://whatis.techtarget.com/definition/memory-management
- http://www.idc-online.com/technical_references/pdfs/information_technology/Memory_Management_Concepts.pdf
- https://www.guru99.com/os-memory-management.html
- https://isaaccomputerscience.org/concepts/sys_os_memory_management
- https://www.studytonight.com/operating-system/memory-management
- https://ecomputernotes.com/fundamental/disk-operating-system/what-is-memory-management
- https://www2.latech.edu/~box/os/ch08.pdf

Shellcode 64 bits

- https://zerosum0x0.blogspot.com/2014/12/x64-linux-bind-shellcode-81-bytes-96.html
- https://www.youtube.com/watch?v=njaQE8Q_Ems
- https://www.youtube.com/watch?v=9q1VL8UU8h0
- https://zerosum0x0.blogspot.com/2014/12/x64-linux-reverse-tcp-connect-shellcode.html
- https://nytrosecurity.com/2019/06/30/writing-shellcodes-for-windows-x64/
- http://shell-storm.org/shellcode/files/shellcode-806.php
- https://www.exploit-db.com/exploits/42179
- https://www.exploit-db.com/exploits/41750

Shellcode 64 bits

- https://www.tophertimzen.com/blog/windowsx64Shellcode/
- https://epi052.gitlab.io/notes-to-self/blog/2018-08-04-x64-linux-metasploit-execve-bin-sh-shellcode-analysis/
- https://wajid-nawazish.medium.com/developing-custom-shellcode-in-x64-57172a885d77
- https://xoban.info/blog/2019/01/29/shellcode-egg-hunter-x64/
- https://zerosum0x0.blogspot.com/2014/12/detect-x86-or-x64-assembly-stub.html
- https://axcheron.github.io/linux-shellcode-101-from-hell-to-shell/
- https://eo-security.com/slae64-assignment-5-msfvenom-shellcode-analysis/
- https://silviavali.github.io/blog/2019-05-01-blog-SLAE51/
- https://mmquant.net/analysis-of-metasploit-linux-x64-exec-shellcode/
- https://www.youtube.com/watch?v=ySKEF8MHcZA

Shellcode x86 bits

- http://shell-storm.org/shellcode/files/shellcode-827.php
- https://www.exploit-db.com/exploits/44321
- https://www.mentebinaria.com.br/forums/topic/397-menor-shellcode-execvebinsh-x86/
- https://packetstormsecurity.com/files/154870/Linux-x86-execve-bin-sh-Shellcode.html
- https://rayoflightz.github.io/shellcoding/linux/x86/2018/11/15/Shellcoding-for-linux-on-x86.html
- https://www.pentesteracademy.com/video?id=115
- https://vegardw.medium.com/haiku-x86-assembly-simple-shellcode-7c76ae614b08
- https://medium.com/@chaudharyaditya/slae-0x2-linux-x86-reverse-shellcode-d7126d638aff
- https://www.programmersought.com/article/32927595383/
- https://modexp.wordpress.com/2017/06/07/x86-trix-one/
- https://www.youtube.com/watch?v=6yv3h1t-58s

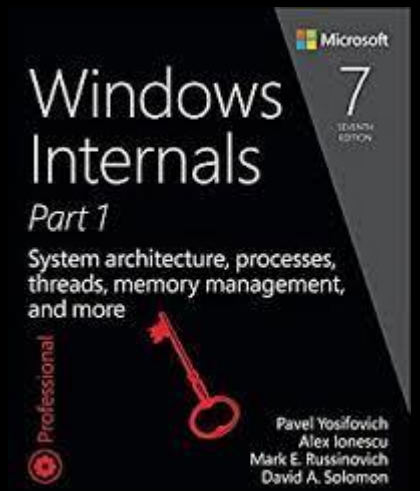
Shellcode x86 bits

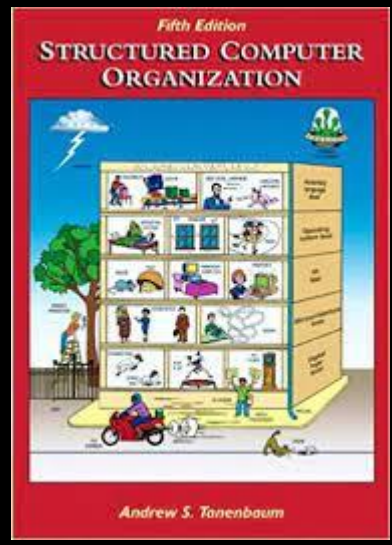
- https://medium.com/@chaudharyaditya/slae-0x5-part-i-analyzing-msfvenom-linux-x86-exec-shellcode-3aef6aad4f70
- https://burmat.gitbook.io/security/hacking/msfvenom-cheetsheet
- https://barriersec.com/2018/12/linux-x86-msfvenom-exec-shellcode-analysis/
- https://cheats.philkeeble.com/exploit-dev/shellcode
- https://ihack4falafel.github.io/Disecting-Msfvenom-Shellcode-~-Linux-x86/
- https://xn4k.github.io/pentest/ethical%20hacking/MSFVenom-Cheetsheet/
- https://book.hacktricks.xyz/shells/shells/untitled
- https://github.com/PacktPublishing/Penetration-Testing-with-Shellcode

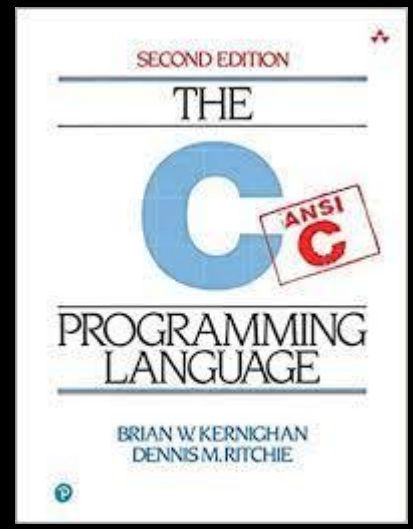
Shellcode Tester

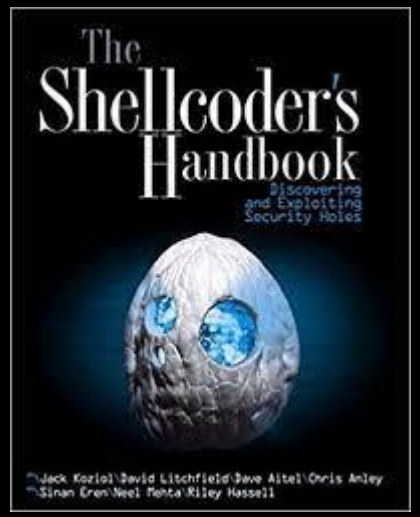
- https://github.com/helviojunior/shellcodetester
- https://github.com/hellman/shtest
- https://github.com/NullByteGTK/Shellcode-Tester
- https://github.com/emptymonkey/drinkme
- https://github.com/fuzboxz/SLAE
- https://sec4us.com.br/cheatsheet/shellcoding
- https://github.com/danielhenrymantilla/shellcode-factory
- https://github.com/NytroRST/ShellcodeCompiler

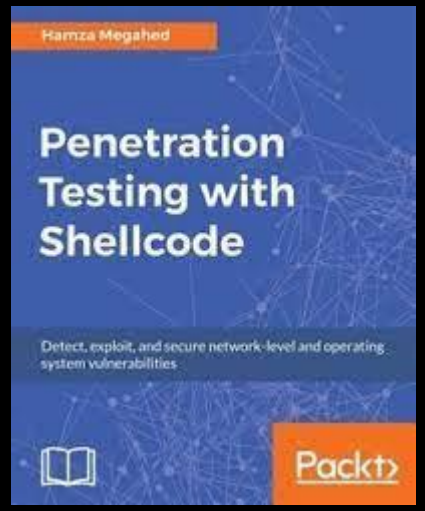
Books

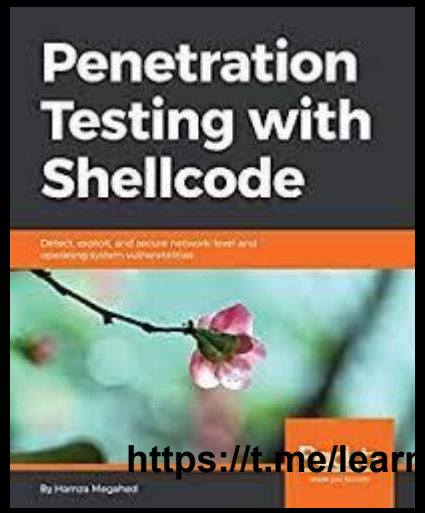


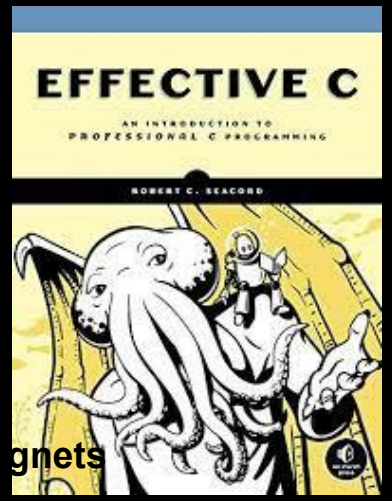


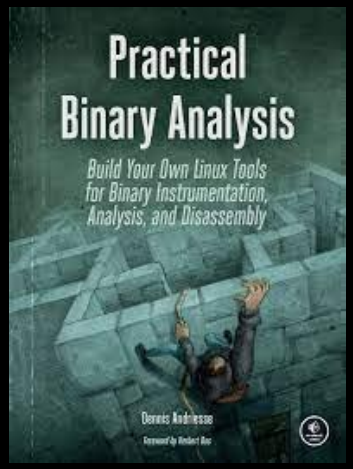


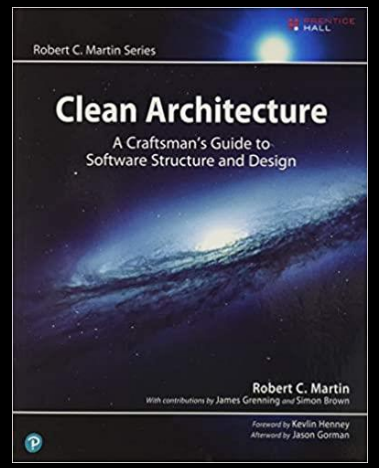












https://t.me/learn gnets