IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Issue 8, August 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008078 |

Automation Module to Prevent a Buffer Overflow Vulnerability

Snehal Jagdhane [1], Shubhanand Hatkar[2]

P.G. Student, Department of CNIS, Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, India[1]

HOD, Department of CNIS, Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, India[2]

ABSTRACT: Buffer overflows were the maximum not unusual place shape of safety vulnerability for the ultimate ten years. Moreover, buffer overflow vulnerabilities dominate the region of far-off community penetration vulnerabilities, wherein a nameless Internet consumer seeks to benefit partial or overall manage of a host. If buffer overflow vulnerabilities can be efficaciously eliminated, a totally massive part of the maximum severe safety threats might additionallybeeliminated.Wesurvey thediversevarietiesofbufferoverflowvulnerabilitiesandassaultsandsurvey the diverseprotectingmeasuresthatmitigatebufferoverflowvulnerabilities,togetherwithourpersonalStackGuardmethod. We then keep in mind which mixtures of strategies can get rid of the trouble of buffer overflow vulnerabilities, at the same time as retaining the capability and overall performance of current systems. To check what protection techniques has been enabled on executables using tools: Checksec for Linux and PESecurity& Win Checksec for Windows. In this paper I have implemented the automation module for both Windows and Linux Environment that will check whether Buffer Overflow Vulnerability is prevented ornot.

KEYWORDS: Buffer Overflow, Automation, Security Mitigation Flags, Checksec, PESecurity& Win Checksec.

I. INTRODUCTION

In maximum facts generation circles those days, the time period buffer overflows turned out to be synonymous with vulnerabilities or in a few cases, exploits. It isn't simplest a horrifying phrase that could maintain you thinking if you got the first-class firewalls, configured your new host-primarily totally based intrusion prevention machine correctly, and feature patched your whole environment, however, can input the safety water-cooler discussions quicker than McAfee's new depraved anti-virus software program or Symantec's brand-new acquisition. Buffer overflows are evidence that the laptop science, or software program programming, networkneverthelessdoes now no longer have an understanding (or, greater importantly, corporation knowledge) of a way to design, create, and enforce stable code. Likeitornownolonger,allbufferoverflowsareamadeofpoorlybuiltsoftwareprogramprograms.

These packages may also have a couple of deficiencies together with stack overflows, heap corruption, layout string bugs, and race conditions—the primary 3 usually being known as truly buffer overflows. Buffer overflows may be as small as one out of place individual in a million-line application or as complicated as a couple of individual arrays which might be inappropriately handled. Some buffer overflows may be observed in nearby packages together with calendar applications, calculators, games, and Microsoft Office applications, while others will be resident in faraway softwareprogramtogetherwithemailservers,FTP,DNS,andtheever-famousInternetWebservers.

Buffer overflow is taken into consideration one of the maximum risky vulnerabilities due to the fact if whilst exploited, itoffersanattackerabigdiplomaofmanipulateoverasoftware'sexecutionandpermitsexecution ofattacker-furnished malicious instructions and code. Buffers are the reminiscence garage areas that quickly maintain information. A buffer overflow vulnerability takes place whilst a software tries to position extra information in a buffer than it could maintain. The extra information corrupts close by area withinside the reminiscence and can modify different information. As a result, it could corrupt information, crash the software, or purpose the execution of maliciouscode.

Someprogramminglanguagesareextrasusceptibletobufferoverflowissues,includingCandC++.Thesearelow-stage languages that depend on the developer to allocate reminiscence and that they don't have integrated safeguards towards overwriting or having access to information of theirreminiscence.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008078 |

Languages including PERL, Java, JavaScript, and C# use integrated protection mechanisms that limit the chance of buffer overflow. However, they're now no longer safe. Some of them permit direct reminiscence manipulation and that they regularly use center capabilities which might be written in C/C++.Paper is organized as follows. Section II. Problem statement Section III. Problem Solution IV. Implementation. Finally, Section V presents conclusion.

II. PROBLEM STATEMENT

In a regular buffer overflow vulnerability exploit, the attacker sends big quantities of facts to an inclined program, which it shops in an undersized stack buffer. The result is that facts on the decision stack is overwritten, together with the feature's go back pointer. The facts units the fee of the go back pointer so that after the feature returns, it transfers manipulate to malicious code contained withinside the attacker's facts.

Types of Buffer Overflow

1. Stack-based: These are extra not unusual place buffer overflows and exploited the use of stack reminiscence that simplest exists at some stage in the execution time of a function. A stack buffer overflow or stack buffer overrun takes place whilst an application writes to a reminiscence cope with at the application's name stack out of doors of the supposed information structure, which is mostly a fixed-period buffer. Stack buffer overflow insects are induced whilst an application writes extra information to a buffer positioned at the stack than what's genuinely allotted for that buffer. This nearly continually effects in corruption of adjoining information at the stack, and in instances in which the overflow changed into prompted via way of means of mistake, will frequently purpose this system to crash or function incorrectly. Stack buffer overflow is a sort of the extra trendy programming malfunction referred to as buffer overflow (or buffer overrun). Overfilling a buffer at the stack is much more likely to derail application execution than overfilling a buffer at the heap duetothefactthestackincorporatesthegobackaddressesforalllivelycharacteristiccalls.

2. Heap-primarily based totally: In this sort of buffer overflow, an attacker floods the reminiscence area allotted for a software past reminiscence used for contemporary runtime operations (heap primarily based totally dynamic reminiscence allocation). A heap overflow or heap overrun is a sort of buffer overflow that takes place withinside the heap records area. Heap overflows are exploitable in a distinctive way to that of stack-primarily based totally overflows. Memory at the heap is dynamically allotted at runtime and usually carries application records. Exploitation is finished with the aid of using corrupting this record in particular approaches to reason the software to overwrite inner systems including connected listing pointers. The canonical heap overflow method overwrites dynamic reminiscence allocation linkage (including malloc metadata)andmakesuseoftheensuingpointeralternatetooverwriteaapplicationfeaturepointer.

Enable buffer overflow safety techniques:

1. Randomization of deal with space: These are extra not unusual place buffer overflows and exploited the use ofstackreminiscencethatsimplestexistsatsomestageintheexecutiontimeofafunction.
Address Space Layout Randomization (ASLR) in Windows: The /DYNAMICBASE alternative in Windows modifies the header of an executable image, a .dll or .exe file, to signify whether or not the utility has to be randomly rebased at load time and permits digital cope with allocation randomization Position Independent Executable (PIE) in Linux: In Linux device -pie alternative produces a dynamically connected function impartial executable on objectives that aid it. This alternative comes into play whilst the compiler hyperlinks item documents into an executable output file.

2. Non-executable stack: This approach is used to save you code execution in stack memory. DEP (Data Execution Prevention) in Windows: DEP is a gadget putting at runtime and packages ought to claim themselves likeminded or not. /NXCOMPAT linker choice suggests that an executable is like minded with the Windows Data Execution Prevention feature. NX (Non-Execute) in Linux: In Linux, NX is a kernel choice, so that is a gadget putting atruntime.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008078 |

3. Stack Canary & Control Flow Guard (CFG): In Linux structures whilst going for walks a program, compilers regularly create random values referred to as canaries and whilst a stack-buffer overflows into the characteristic go back cope with the canary is overwritten. During characteristic go back the canary price is checked and if the price has modified this system isterminated.
   Control Flow Guard (CFG) in Windows: The /guard:cf collect choice is used to permit CFG protection. The Compiler analyses manage float for oblique name objectives at collect time, after which inserts code to confirm the objectives at runtime ii. Canary in Linux: This affects performance, so it needs to be attempted with unique values. Prefer -stack-protector-robust and song the --param ssp-buffer-size=X (song X, that's eight with the aid of using default: Lower is better, however affectsperformance).

III. PROBLEM SOLUTION

Automation module script reduced the manual scanning and checking of the security flag mitigations also the script will perform the data cleaning on the executables which are in C/C++.The algorithm automatically generates result of flag mitigation values for all executables.

IV. IMPLEMENTATION

Automation for Windows buffer overflow mitigation: ASLR, DEP and ControlFlowGuard these three flag values will be checked to prevent buffer overflow vulnerability in windows. The scan will be performed on the specified executables on the windows Environment machine and accordingly it will generate the result with the Flag values along with data cleaning.
Algorithm:

1. Login to themachine
2. Execute the commands to perform the scan on the specified scanresult
3. Check the scan result (PowerShellscript)
4. Perform the data cleaning (created the pythonscript)
5. Generate the scan result with the flag mitigation values for all theexecutables

Path = Scan Path			
FileName	ASLR	DEP	ControlFlowGuard
Executale path	FALSE	FALSE	FALSE
Executale path	TRUE	TRUE	FALSE
Executale path	FALSE	FALSE	FALSE
Executale path	TRUE	TRUE	FALSE
Executale path	TRUE	TRUE	FALSE

Fig 1. Result of some executables in Windows

Abovefig1.showstheflagmitigationvaluesmitigatedflagvaluesaremarkedwiththegreenandthosearenotmitigated marked with red color, accordingly we will get an idea of the status of Buffer Overflow Vulnerability.

PowerShell Commands for Windows Automation

New-PSSession - The New-PSSessioncmdlet creates a PowerShell session (PSSession) on a nearby or faraway computer. When you create a PSSession, PowerShell establishes a chronic connection to the faraway computer. Use aPSSessiontorunmorethanoneinstructionthatproportiondata,whichincludesafeatureorthefeeofavariable.

Invoke-Command - The Invoke-Command cmdlet runs a Get-Host command at the faraway computer systems. It makes use of dot notation to get the Version belongings of the PowerShell host. These instructions run one at a time. When the instructions complete, the output of the instructions from all the computer systems is stored in the $model variable.

| e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 7.569|

|| Volume 10, Issue 8, August 2021 ||

| DOI:10.15680/IJIRSET.2021.1008078 |

Automation for Linux buffer overflow mitigation: Stack Canary, NX, PIE and Symbols flag values will be checked to prevent buffer overflow vulnerability in Linux. The scan will be performed on the specified executables on the Linux Environment machine and accordingly it will generate the result with the Flag values along with data cleaning.
Algorithm:

1. Login to themachine
2. Execute the commands to perform the scan on the specified scanresult
3. Check the scan result (PowerShellscript)
4. Perform the data cleaning (created the pythonscript)
5. Generate the scan result with the flag mitigation values for all theexecutables

Path = Scan Path				
FileName	**STACK CANARY**	**NX**	**PIE**	**Symbols**
Executale path	canary found	NX enabled	No PIE	No symbols
Executale path	No canary found	NX enabled	PIE	77 Symbols
Executale path	canary found	NX Disabled	PIE	977 Symbols
Executale path	No canary found	NX enabled	PIE	3492 Symbols

Fig 2. Result of some executables in Linux

Abovefig2.showstheflagmitigationvaluesmitigatedflagvaluesaremarkedwiththegreenandthosearenotmitigated marked with red colour, accordingly we will get an idea of the status of Buffer OverflowVulnerability.

PowerShell Commands for Linux Automation

New-SSHSession - SSH-Sessions module offers SSH consultation creation, control and interplay from PowerShell. Invoke-SSHCommand – This command provides the facility to execute the shell command on the remote Linux machine.

V. CONCLUSION

We have implemented an automation module that will check whether Buffer Overflow Vulnerability is prevented or not. Our algorithm successfully checks the various security flag values are mitigated as per the requirement or not. We have applied our algorithm on many executables and found that it successfully detects the Flag mitigation values.

REFERENCES

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Buffer Overflow: Proof Of Concept Implementation”, in Proc. SIGGRAPH, pp. 417–424, 2000.
[2] S.Bellovin,“BufferOverflowsandRemoteRootExploits,”PersonalCommunications,vol.4,pp.613–615,1999.
[3] R. Seacord, “Secure Coding in C and C++,” Addison-Wesley:2005.
[4] J. Foster, V. Osipov, N. Bhalla, and N. Heinen, “Buffer Overflow Attacks: Detect, Exploit, Prevent,” Syngress Publishing:2005.
[5] C. Kil, J. Jun, C. Bookholt, J. Xu, P. Ning. "Address Space Layout Permutation (ASLP): Towards Fine-Grained Randomization of Commodity Software," Proceedings of the 22nd Annual Computer Security Applications Conference(ACSAC’06).
[6] W. Robertson, C. Kruegel, D. Mutz, and F. Valeur. "Run-time Detection of Heap-based Overflows," Proceedings of the 17th Large Installation Systems Administration (LISA XVII),2003.
[7] C. Cowan, C. Pu, D. Maier, H. Hintony, J. Walpole, P. Bakke, S. Beattie, A. Grier, P. Wagle, and Q. Zhang. "Stackguard: automatic adaptive detection and prevention of buffer-overflow attacks," SSYM’98: Proceedings of the 7th conference on USENIX Security Symposium, pp. 63–78,1998.
[8] M. Zitser, “Securing software: An evaluation of static source code analyzers,” Master’s thesis, Massachusetts Institute of Technology, 2003.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details