INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

SIGNATURE AND SIGNATURE-FREE MALICIOUS CODE BLOCKER SYSTEM FOR IMAGE FILES

PROF. R. R. SAWANT[1], MS. S. D. PANDAO[2], MR. K. D. BODHE[3]

Khurana Sawant Institute of Engineering and Technology , Hingoli

Accepted Date: 15/02/2014 ; Published Date: 01/04/2014

Abstract: Internet threat have different form of attacks, considering individual users to obtain control over data and network. The Buffer Overflow which is one of the most frequently occurring security vulnerabilities on network. Buffer Overflow occurs while writing data to a buffer and it overruns the buffer's threshold and overwrites it to neighboring memory. The techniques to avoid buffer overflow vulnerability vary per architecture, Operating system and memory region. The Signature based buffer overflow detection finds the particular Signature and if that found it blocks it to protect form malicious attack. The remaining request are consider for checking against the buffer size and grant to server if the buffer of request is less than or equal to defined threshold value of buffer. Signature free first filters and extracts instruction sequences from a request. Finally it compares the number of useful instructions to a threshold to determine if this instruction sequence contains code. Signature free thus it can block new and unknown buffer overflow attacks, Signature free is also immunized from most attack-side code obfuscation methods. Since Signature free is transparent to the servers that protected, it is efficient for economical Internet wide deployment with very low deployment and maintenance cost. I will demonstrate techniques for preventing buffer overflow during the transmission of images of different formats. I will discuss and evaluate certain tools and techniques which prevent buffer overflows.

Keywords: Buffer-Overflow, Signature, Signature free, Malicious code, Intrusion, vulnerabilities

PAPER-QR CODE

Corresponding Author: PROF. R. R. SAWANT

Access Online On:

www.ijpret.com

How to Cite This Article:

RR Sawant, IJPRET, 2014; Volume 2 (8): 145-151

INTRODUCTION

Computer Security includes the protection of information and property from hacker, corruption while allowing the information to remain accessible to its intended user. Computer security means valuable information and services are protected from access, collapse by unauthorized activities or events. A buffer overflow occurs when data written to a fixed sized buffer, due to insufficient bound checking, corrupts data values in memory addresses adjacent to the allocated buffer.

Software security has become an increasing necessity for guarantying, as much as possible, the correctness of computer systems. Unfortunately, software vulnerabilities are omnipresent. In the last few decades, much new vulnerability has been discovered, and old ones have been continuously exploited. Memory corruption in C and C++ programs has been known for decades and is one of the oldest classes of software vulnerabilities.

Computer Security includes the protection of information and property from hacker, corruption while allowing the information to remain accessible to its intended user. Computer security means valuable information and services are protected from access, collapse by unauthorized activities or events. A buffer overflow occurs when data written to a fixed sized buffer, due to insufficient bound checking, corrupts data values in memory addresses adjacent to the allocated buffer.

Software security has become an increasing necessity for guarantying, as much as possible, the correctness of computer systems. Unfortunately, software vulnerabilities are omnipresent. In the last few decades, much new vulnerability has been discovered, and old ones have been continuously exploited. Memory corruption in C and C++ programs has been known for decades and is one of the oldest classes of software vulnerabilities.

2. EXISTING STUDY

Xinran Wang, Chi-Chun Pan, Peng Liu, and Sencun Zhu proposed work on “Signature free: A Signature-Free Buffer Overflow Attack Blocker” [1]. There experimental study shows that the dependency-degree-based Signature free could block all types of code-injection attack packets (above 750) tested in their experiments with very few false positives. Moreover, Signature free causes very small extra latency to normal client requests when some requests contain exploit code.

Eric Haugh and Matt Bishop discussed work on “Testing C Programs for Buffer Overflow Vulnerabilities” [2] this evaluation shows that the tool is useful for finding buffer overflow flaws that it has a low false positive rate, and compares well with other techniques.

Crispin Cowan, Perry Wagle, Calton Pu, Steve Beattie, and Jonathan Walpole proposed work on “Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade” [3]. They consider which combinations of techniques can eliminate the problem of buffer overflow vulnerabilities, while sustaining the functionality and performance of existing systems.

Hassen Sallay, Khalid A. AlShalfan, Ouissem Ben Fred proposed work on " A scalable distributed IDS Architecture for High speed Networks" [4]. They worked on switch-based splitting approach that supports intrusion detection on high-speed links by balancing the traffic load among different sensors running Snort.

Wenhua Wang, Yu Lei, Donggang Liu, Devid Kung, proposed work on "A Combinatorial Approach to detecting Buffer Overflow Vulnerabilities [13]. They present a black-box testing approach to detecting buffer overflow vulnerabilities. it is motivated by a reflection on how buffer overflow vulnerabilities are exploited in practice.

Ashish Kundu, Elisa Bertino proposed work on " A New Class of Buffer Overflow Attacks[14].They focus on a class of buffer overflow vulnerabilities that occur due to the " placement new" expression in C++.it showed how the "placement new" expression in C++ can be used to carry out buffer overflow attacks-on the stack as well as heap/data.

This works deals with finding and detection of buffer overflow occurs during transmission of image file format over network. The signature and signature free detection mechanism is implemented for detection of buffer overflow during transmission. First the signature based detection finds a particular signature after the signature free technology is implanted for testing the buffer overflow. Each time during receiving end we check for buffer overflow and if found then we are blocking that images and forwarding the remaining images to server for processing.

To overcome the problem of buffer overflow we proposed the SigFree attack blocker technique. The background behind the SigFree is motivated by an important observation that "the nature of communication to and from network services is predominantly or exclusively data and not executable code" [12].

The aim is to develop an effective security solution for today's Computer's Society i.e.to develops Signature and signature free buffer-overflow detection system for jpg and gif files, which blocks known and unknown attack. The signature and signature free detection blocks the unwanted request and supply pure code to server for maintaining security. The analysis of signature and signature free detection based on the threshold value set by programmer.

3. METHODOLOGY

3.1 Network-based attacks Detection and Prevention Techniques:-

Network-based attacks can be protected by using Firewall, Intrusion Prevention System and Buffer Overflow Exploit Prevention. One of the newest host protection technologies available is buffer overflow exploit prevention, also known as memory protection. As a highlevel rule, code should never be executed from writable areas of system memory, by observing the use of Stack and Heap system memory.

A personal firewall will block known and unknown attacks against the ports and services you don't need. IPS will filter out known and unknown attacks against known vulnerabilities. Intrusion detection systems (IDS) provide deep packet inspection capabilities that examine the traffic allowed through by personal firewall rules and alert the user to an attack on the host system. IPS technology is with the ability to identify good traffic from malicious traffic in real time. It is categorized into either signature-based methods or protocol analysis-based methods. Signature-based techniques are effective at stopping known exploits, but are often too reactive.

3.2 Principles of Buffer Overflow Attacks:-

In a typical buffer overflow attack, the attacker injects aninstruction sequence into the victim application and transfers the control of the application to the injected code. As an application's text segment is typically read-only, the only way to hijack the control of an application is to dynamically modify the target address of its branch instructions whose target is not fixed at compilation time. Such dynamic branch instructions include function returns, pointer-based function calls, and C style switch statements.

3.3 Block diagram

Figure 1 shows the architecture of our study. The request from client is verified on Signature and Signature free and the valuable request are send to server.

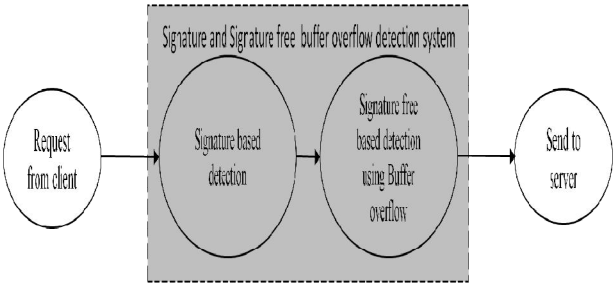

Signature and Signature free buffer overflow detection system
Request from client
Signature based detection
Signature free based detection using Buffer overflow
Send to server

Fig. 1: Block Diagram of System.

3.4 Signature based Detection techniques for buffer Overflows:-

The request generated by client is encoded and Signature based detection decode the request from client. Signature based detection finds defined signature in the request generated by client which block a particular request contain defined signature(here IP address is considered as defined signature).The request containing defined signature is block by signature based technique by using patter matching, there by implementing security for network.

3.5 Signature Free Detection techniques for buffer overflows:-

The request not containing defined signature from signature based detection are forwarded to signature free detection system where checking for the buffer overflow by setting threshold value of buffer. If the buffer requirement of request is more than defined threshold value of buffer then it exploits the buffer overflow which helps to find malicious code embedded in request when it is send over network.

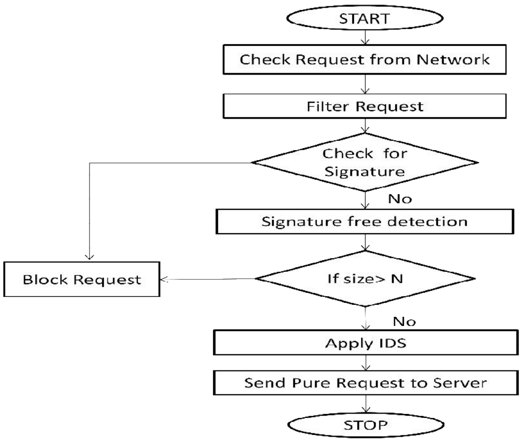
3.6 Algorithm:-

```
Input: Data from network
Step 1:  Check request from Network
Step 2:  Check for signature
Step 3:  Signature found
Step 4:  Block request
Step 5:  Pure request send to signature free detection
Step 6:  Check for signature free
Step 7:  Check size of Data
Step 8:  If size > N (Buffer overflow occurred)
Step 9:  Block Request
Step 10: Supply pure code to Server.
Step 11: Stop
```

Generate request from client and send to the server. On Server receive request from client and remove unwanted request (other than gif and jpg format). In signature based detection firstly signature verified on the basis of pattern matching method check the signature which is defined on server and IP address of request is consider as signature, if the signature is found block that request and filter request send to the signature free detection.

In Signature free detection Select the buffer size if the buffer size is greater than that selected buffer size block that request less than or equal buffer size request is send to the server.

3.7 Flowchart:-



4. Conclusion

We have proposed Signature and Signature-free malicious code blocker system with image file format the can filter code-injection buffer overflow attack, one of the most serious cyber security paradigm. Sigfree does not require any signatures, thus it can block new malicious code and provide security for the system. Sigfree is less affected from malicious attack and economical for deployment with little maintenance cost and low performance overhead.

REFERENCES:

1. Xinran Wang, Chi-Chun Pan, Peng Liu, and Sencun Zhu, "SigFree: A Signature-Free Buffer Overflow Attack Blocker", Ieee Transactions On Dependable And Secure Computing, Vol. 7, No. 1, January-March 2010.

2. Eric Haugh and Matt Bishop, "Testing C Programs for Buffer Overflow Vulnerabilities".

3. Crispin Cowan, Perry Wagle, Calton Pu, Steve Beattie, and Jonathan Walpole "Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade"

4. Hassen Sallay, Khalid A. AlShalfan, Ouissem Ben Fred j," A scalable distributed IDS Architecture High speed Networks", IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.8, August 2009

5. E. Barrantes, D. Ackley, T. Palmer" Randamized Instruction Set Emulation to Disrupt Binary Code Injection attacks, Proc.10th ACM Conf. Computer and Comm. Security Oct.2003

6. J. Newsome and D. Song, "Dynamic Taint Analysis for Automatic Detection, Analysis, and Signature Generation of Exploits on Commodity Software," Proc. 12th Ann. Network and Distributed System Security Symp. (NDSS), 2005.

7. B.A. Kuperman, C.E. Brodley, H. Ozdoganoglu, T.N. Vijaykumar, and A. Jalote, "Detecting and Prevention of Stack Buffer Overflow Attacks," Comm. ACM, vol. 48, no. 11, 2005.

8. M. Costa, J. Crowcroft, M. Castro, A. Rowstron, L. Zhou, L. Zhang, and P. Barham, "Vigilante: End-to-End Containment of Internet Worms," Proc. 20th ACM Symp. Operating Systems Principles (SOSP), 2005.

9. J. Pincus and B. Baker, "Beyond Stack Smashing: Recent Advances in Exploiting Buffer Overruns," IEEE Security and Privacy, vol. 2, no. 4,2004.

10. G. Kc, A. Keromytis, and V. Prevelakis, "Countering Code-Injection Attacks with Instruction-Set Randomization," Proc. 10th ACM Conf. Computer a Comm. Security (CCS '03), Oct. 2003.

11. S.-X. Wu and W. Banzhaf, "The Use of Computational Intelligence in Intrusion Detection Systems: A Review," Elsevier Applied Soft Computing, vol. 10, issue 1, Jan.2010, pp1-35.

12. M H. T. Elshoush and I. M. Osman, "Reducing False Positives through Fuzzy Alert Correlation in Collaborative Intelligent Intrusion Detection Systems — A Review," Prof. IEEE Int'l.Conf. Fuzzy Systems, July 2000, pp. 1–8.

13. Wenhua Wang, Yu Lei,Donggang Liu,Devid Kung, "A Combinatorial Approach to detecting Buffer Overflow Vulnerabilities" 978-1-4244-9233-6/11/$26.00 ©2011 IEEE

14. Ashish Kundu, Elisa Bertino "A New Class of Buffer Overflow Attacks", 2011, 31st International Conference on Distributed Computing Systems

15. Crispin Cowan, Perry Wagle, Calton Pu, Steve Beattie, and Jonathan Walpole "Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade", http://www.cse.ogi.edu/DISC/projects/immunix 2002.