Buffer Overflow attack avoiding Signature free technique

Umesh Deshmukh
Student of Comuter Engineering
S.E.C.O.E.Kopargaon ,A"Nagar Maharastra,India

Prof.P.N.Kalawadekar
Department of Computer Engineering
S.E.C.O.E.Kopargaon ,A"Nagar Maharastra,India

Abstract- Now a days, The web technologies provide to scatters data of digital information worldwide real-time but today any computer system not secure from the attacker. There are many attackers or hackers to attack and access our private data in our private account. So these problem can overcome in these paper. In Paper system can secure private data by using the intrusion and detection algorithm to overcome the blockers and most serious cyber threats. In paper, system any user or authorized user can give to user to access their account but it only gives the access on a particular file and it is a time limited process. So another user can not access the private data of that particular user's account. So in paper system is basically ly security purpose to secure the private data from the hackers or attackers. In paper Signature free technique, a real time ,out-of the- box, avoiding signature to use, application layer blocker for prevent buffer overflow attacks by hackers, one of the most serious cyber security coercion. Signature free technique can filter out code-injection buffer overflow attack messages or code targeting at various Internet services such as web service. In these paper starting blindly dissembles and extract order sequences from a request which is upload or access. After that then code abstraction technique is used, which helpful data flow anomaly to prune useless instructions in an instruction sequence by defines url. At end it differ the number of useful information to a threshold to determine if this instruction series contains program code or not. In paper system will block new and unknown buffer overflow attacks which is done by hackers.; Hence Signature free technique is transparent to the servers being protected from hacker or attacker, System will be good for costly Internet wide deployment with very low deployment and maintenance cost. Hence it will be beneficial to networking system to obtained pure data rather than code. That is system is secure from hacker which they are not apply any program (code) in respective buffer.

Keywords – Intrusion detection, Buffer overflow attacks, Code injection attacks, Cyber security, Obufaction

I. INTRODUCTION

First a general question is arise that what is buffer? Buffer is temporary storage area usually in Ram. It act as holding area enabling the CPU to manipulate data before transferring to device. Because the process of reading and writing data to a disk relatively slow many programs track of data changes in buffer and buffer to disk.[12] A buffer overflow is similar a glass of water which hold full capacity of glass. But when it will happen that the water overflow from glass indicate that capacity of glass has full and overflow the water from glass. So consider that glass is buffer and water is a information or code. There are two types of buffer one is 'stack' and another is 'Heap'. In these paper consider only stack. Normally, One type involves overwriting (and thus changing) security sensitive variables or control flags stored in memory adjacent to the unchecked buffer. The most common type of stack overflow involves the overwriting of purpose pointers that can be used to change program flow or gain elevated privileges within the operate by system environment. The more complex heap over run involves dynamic allocations, or memory allocated at run time by an application.[11] In this paper will place our focus on the Stack Buffer Overflow. Mostly some cases must have not vary bad understanding of how to allocate memory for operating system and similarly, how to use application in allocation of memory. So that stack are in realistic applications.Generally,the worm that has unleashed on network which bombarding a buffer overflow vulnerability on network with some software. However, the production versions of the executables may not slot in such protection for performance reasons. It has much constraints, like that those that apply to small embedded systems, may identify for particularly tiny executables, meaning executables without the protection against buffer overflow.[5][11]

The rest of the paper is organized as follows. Proposed embedding and extraction algorithms are explained in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. PROPOSED SYSTEM

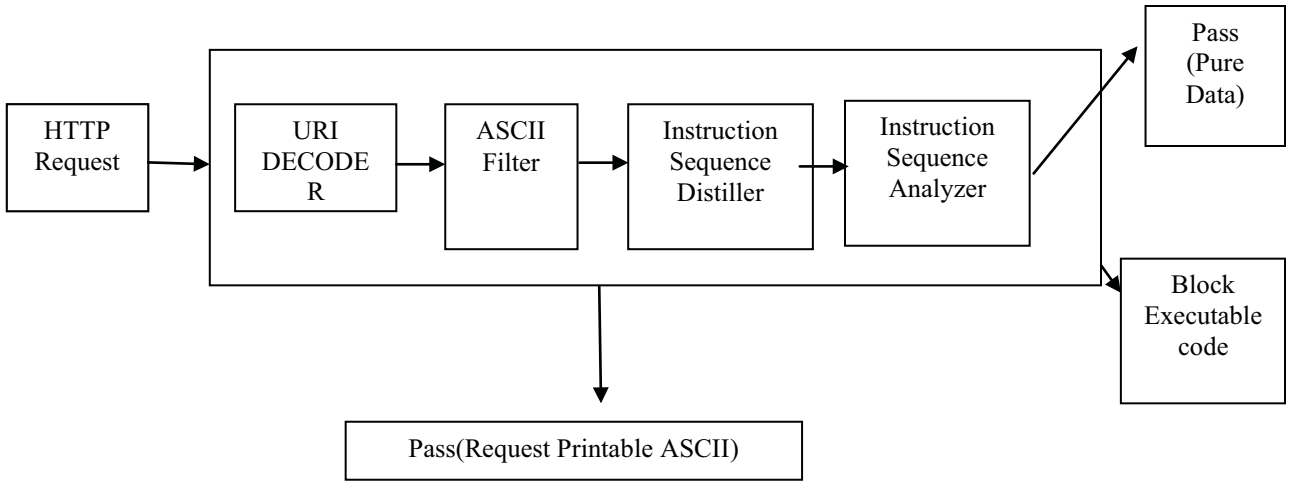


Figure 1: Architecture of System.

The breakdown structure mainly focuses on following areas-

1. Module 1: Responsible for user control; restricts unauthorized users.
2. Module 2: Encode and decode ASCII filter.
3. Module 3: Instruction Sequence Distiller.
4. Module 4: Instruction sequence Analyzer.

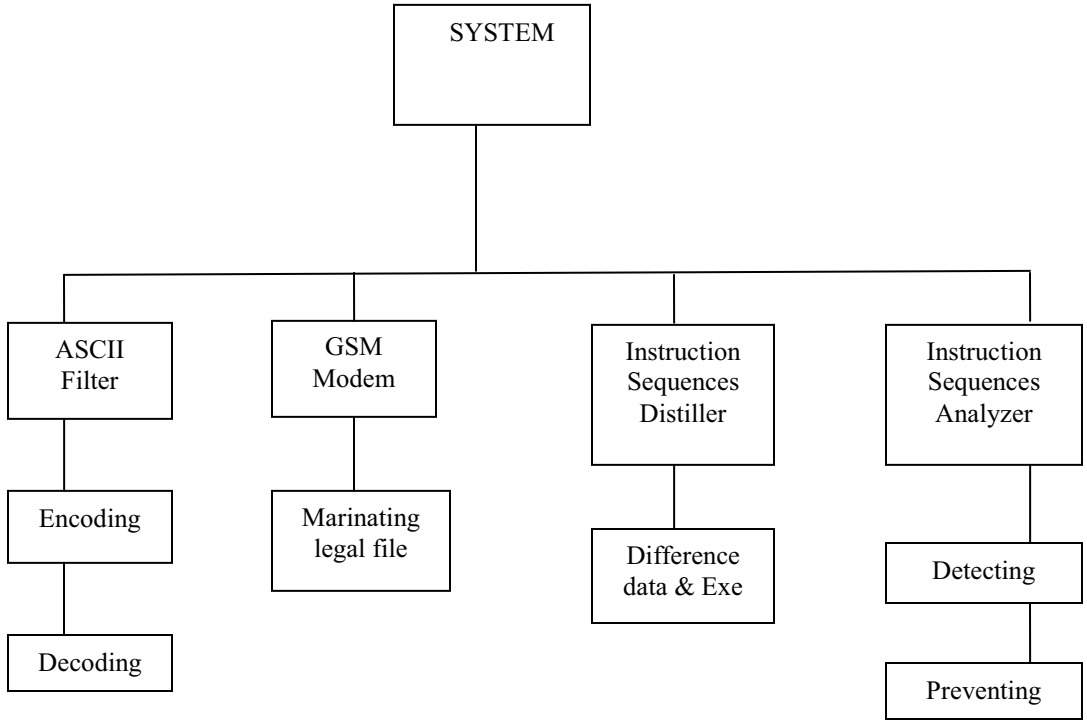


Fig: 2 Breakdown Structure Modules

Module 1: User Control
Input: User name and password as input.
Output: Successful or unsuccessful login.
Algorithm:

1. Get user name and password.
2. Logs into the system.
3. Starts his new session.
4. After completion of session user logs out.

The above algorithm shows how exactly the login module will provide security to the entire system to prevent unauthorized access of system. When user clicks on save button all his information get inserted into the database.

Now this user has its own username and password. By clicking "click here to login" link he will redirect to login page. Here user will login into the system with his personal username and password. If user enters correct username and password as filled in the registration form; a "Login Successful" message will displays else if he enters wrong username or password then the system will displays "Invalid username or password message". Thus this module gives security and provides user control to the system.

5.2 Module 2: ASCII Encode and Decode
Input: Accept HTTP request
Output: Generate ASCII values
Algorithm:

1. Check HTTP request.
2. Encode files.
3. Decode files.
4. Generate ASCII values.
   The above algorithm shows how exactly the check HTTP request through database then encode decode the files then generate ASCII values within range 20-7E. When user login the page then it transmit to Accept HTTP request page. Then generate encode decode files for security purpose and then doing generate ASCII values.

5.3 Module 3: Instruction sequence distiller
Input: Attach file
Output: Check exe code
Algorithm:

1. Attach file.
2. Check exe code.

This algorithm used to distill an instruction sequence; we first assign an address to every byte of a request. Then, we disassemble the request from a certain address until the end of the request is reached or an illegal instruction op-code is encountered. There are two traditional disassembly algorithms*:* linear sweep and recursive traversal .The linear sweep algorithm begins disassembly at a certain address, and proceeds by decoding each encountered instruction. The recursive traversal algorithm also begins disassembly at a certain address, but it follows the control flow of instructions.

5.4 Module 4: Instruction Sequence Analyzer

Input: Upload file.
Output: Check legal or illegal file.

Algorithm:

1. Upload file
2. legal or illegal file to pass to server.

This algorithm is to check if the number of useful instructions in an execution path exceeds a threshold. The algorithm involves a search over an EISG in which the nodes are visited in a specific order derived from a depth first

search. The algorithm assumes that an EISG G and the entry instruction of the instruction sequence are given, and a push down stack is available for storage. During the search process, the visited node (instruction) is abstractly executed to update the states of variables, find data flow anomaly, and prune useless instructions in an execution path.

- In our proposed system use the Public key algorithm to generate the public key and private key.
- In our system key is generated randomly using a Random Number Generator and Pseudorandom number generator (PRING). PRING is a computer algorithm that produces data that appears random analysis

RESULT:-

TID	Exp.Result	Actual Result	(Pass/Fail)
1	Login	check valid or not	P
2	Upload	Put file Name,No.	P
3	Browse	File upload	P
4	Encode	check valid or not	P
5	Decode	check valid or not	P
6	ASCII values	check valid or not	P
7	Legal /Illegal file	Pass or not	P

Table 1: Experiment result with Actual result

MODULE REPRESENTATION USING SEQUENCE DIAGRAM

A sequence diagram is graphical view of scenario that shows object interaction in time based sequence. What happens first, what happens next? Sequence Diagram establishes the roles of objects and help to provide essential information to determine class responsibilities and interfaces. This type of diagram is best used early analysis phases in design because they are simple and easy to comprehend. Here is the sequence diagram of 'Login Module'.

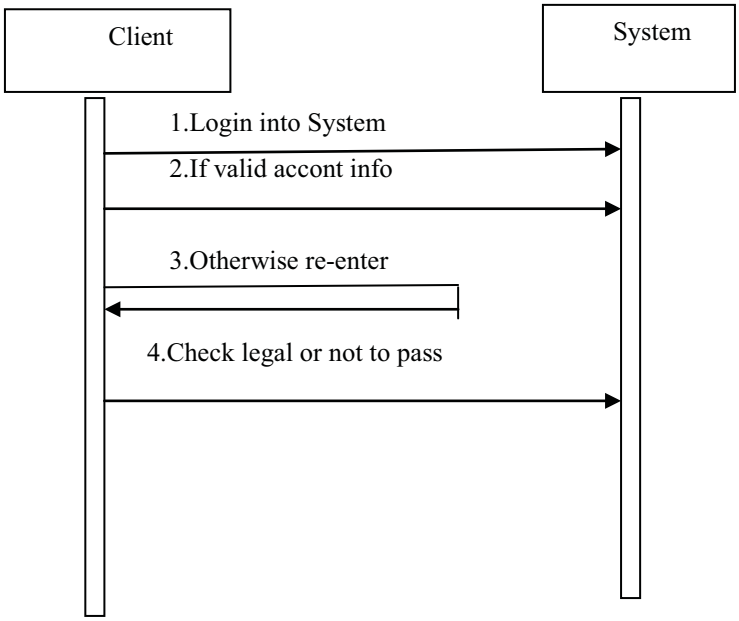
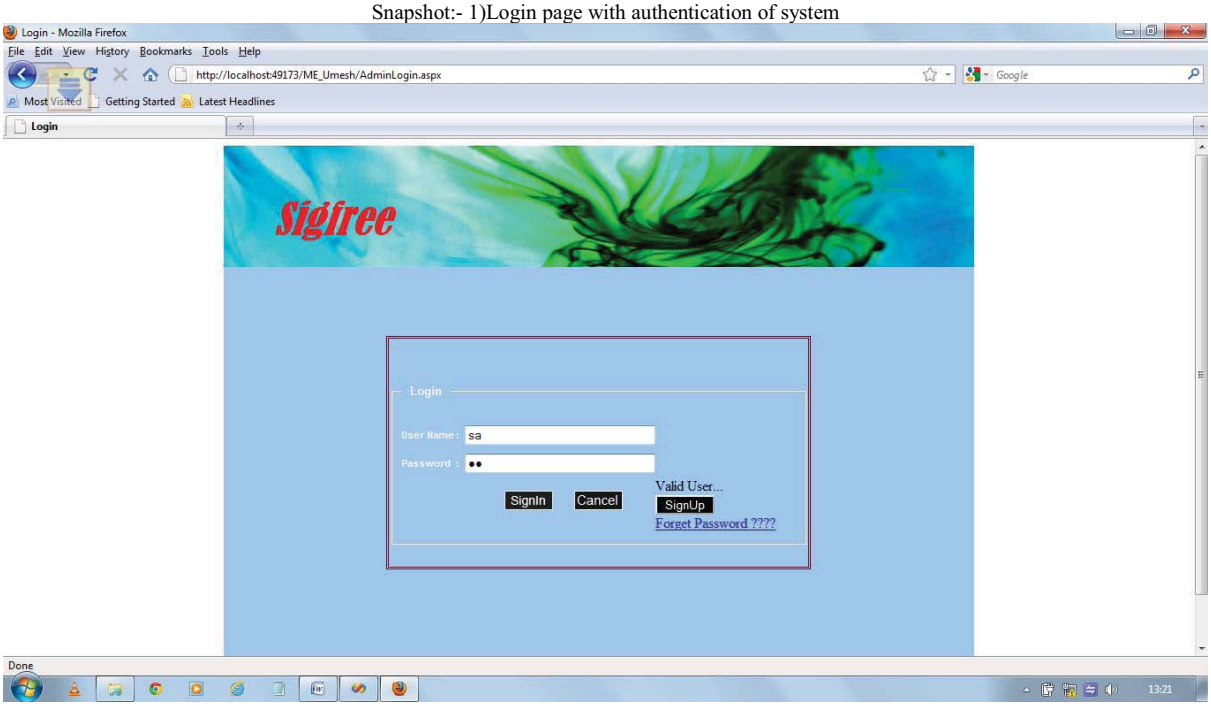
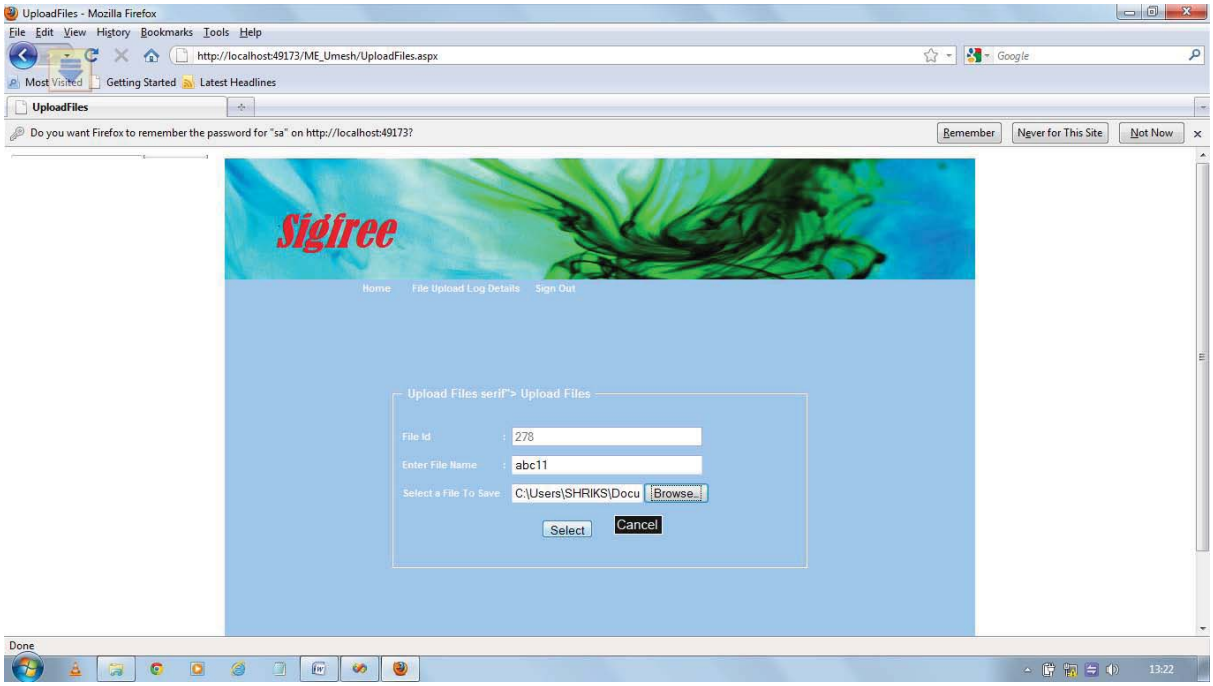


Fig:3 Module sequence Diagram

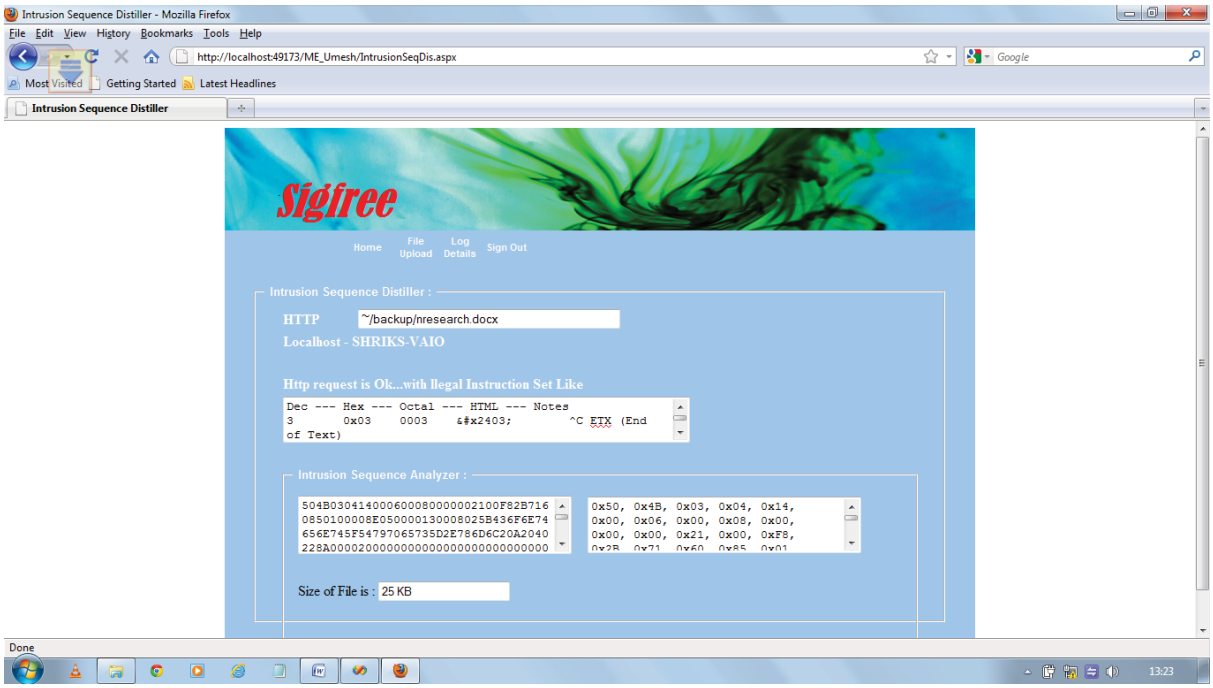
Snapshot:- 1)Login page with authentication of system



2) Upload the file



3) Final Check to legal or Illegal with size .



IV.CONCLUSION

In these papers A Signature free, a real time, out of- the box blocker that can scan code-introduce buffer overflow attack messages, one of the important cyber security threats, to various Web services. It provides Signature free technique so does not require any define signatures, thus it can block new, unknown attacks which is done by hacker. Signature Free technique is immunized from most attack-side code obfuscation methods, good for economical Internet wide deployment with minimum maintenance cost and minimum throughput degradation.

REFERENCES

[1] Citeseer: Scientific Literature Digital Library,*http://citeseer.ist. psu.edu*, 2007. .
[2] J. Pincus and B. Baker,*Beyond Stack Smashing: Recent Advances in Exploiting Buffer Overruns IEEE Security and Privacy* vol. 2,no. 4, 2004.
[3] 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.
[4] D. Evans and D. Larochelle, "Improving Security Using Extensible Lightweight Static Analysis," IEEE Software, vol. 19, no. 1, 2002.
[5] J. Newsome, B. Karp, and D. Song,*Polygraph: Auto matic Signature Generation for Polymorphic Worms*, Proc. IEEE Symp. Security and Privacy , 2005.
[6] Intel ia-32 architecture software developer's manual volume 1: Basic architecture.
[7] Security advisory: Acrobat and adobe reader plug-in buffer overflow. *http://www.adobe.com/supprot/techdocs/321644.html*.
[8] Symantec security response: back door. hesive.*http://securityresponse.symantec.com*
[9] J. Huang, *" Detection of Data Flow Anomaly through Program Instrumentation "*, IEEE Trans. Software Eng., vol. 5, no. 3, May 1979.
[10] Xinran *"Sigfree:A Signature free Buffer Overflow Attack Blocker"*, IEEE,VOL. 5, NO. 4,2008.
[11] Bhatkar *"Efficient techniques for comprehensive prtection from memory error exploits"*,In USENIX Scurity(2005).
[12] Krerk piromsopa *"Buffer-Overflow Protection: The Theory"*,IEEE.