NOVATEUR PUBLICATIONS
INTERNATIONAL JOURNAL OF INNOVATIONS IN ENGINEERING RESEARCH AND TECHNOLOGY [IJIERT]
ISSN: 2394-3696
VOLUME 6, ISSUE 6, June-2019

DYNAMIC MALWARE ATTACK DETECTION AND PREVENTION IN REAL TIME IoT WITH HYBRID SIGNATURE FREE METHOD

DR.R.SRİDEVİ
Associate Professor, Department of Computer Science and Engineering,
K Ramakrishnan College of Engineering Samayapuram, Trichy – 621 112.
sridevivelon@gmail.com

S.SRİMATHİ
Assistant Professor, Department of Computer Science and Engineering,
K Ramakrishnan College of Engineering Samayapuram, Trichy – 621 112.

ABSTRACT
In today's information age the major issue to address is cyber security. In Iot Cloud, usually a client targets a server and causes plenty of problems. In present IDS are used to detect only known service level attacks which leaves them vulnerable to original and novel malicious attacks. The primary objectives of this study is to develop an efficient security model which detects malicious attacks like code injection, ping, worm, flood etc. done through various layers by malicious clients and prevent them from causing damage apart from blocking them, thereby saving both the clients and their valuable data stored in the server. The proposed model named HSIGFREE is signature independent and hence able to detect any attacks in the network on the server without any problems and prevents damages to the server caused by such malicious clients. Hence the proposed HSIGFREE model is a viable, safe and secure data and application layer protection and attack prevention model in dynamic server environments.
KEYWORDS: Malware Attacks, HSigfree, Attack Prevention, intrusion detection, cyber security

1 INTRODUCTION
Any network which present today in this technology era needs a potential strength in significant situations. With the rise of internet access and plethora of internet of Things services security and cyber-attacks have also proportionally increased. The malicious web pages do lots of attacks to the victim. Some of the attacks are like Impersonation, Being a third party Desktop support Shoulder surfing Dumpster diving attacks apart from Phishing, Baiting and On-line scams The proposed model involves the prevention of such malicious pages before the damage is caused to the users.

In reality most existing models are content-based and require extensive scrutiny methods adopted to detect malicious websites. But they fail to recognize script based attacks on the victims. Some uses DNS based approaches for detecting the attacks and they were incapable in providing deeper understanding. Usually use fixed features which can realizes the victim damage before it is going to happen and let the customer either compromised or loses money or data.

Most existing systems are Signature Based or Anomaly Based meaning they require any pattern or model to train in order to detect. Here signature means a model or pattern and the existing models today tuned to detect only known network attacks. The models have to sift through large quantum of data and hence before the model even has gone through half the data, the attack is over and valuable data is lost. Some models are Hybrid Based – Combo of signature and anomaly but are known to suffer from computational overheads. Some drawbacks of existing system are signature dependent and in case of absence of signature it does not detect. Depending on the training pattern or classification model, detection takes place after attack or damage is done which has huge computational cost and overheads and generates very high false positives and false negatives, which were vulnerable to novel attacks and can be addressed well by using the Data mining techniques.

In today's modern information age IDS are usually detect only known attacks which leaves them vulnerable. In real world the data found to be is high that too is growing rapidly which raise the data overload problem and also have the problem of deciding an analyst to include how much data can be analyzed efficiently that all depends on the intrusion detection tools available in hand. For false positives which generated by IDS indicates when the normal attack is erroneously categorized as malicious and treated accordingly and in case of false negatives, where an IDS does not generate an alert when an intrusion is actually taking place.

The proposed model first detects suspicious activity. The activity is sandboxed and the URL is blacklisted. Next the actions are blocked and alerts are issued to the user. Further the attackers are revoked by the proposed kayo model. Thus the adversaries are neutralized and a database is thus built. During the next visit by a member client they are sufficiently blocked and warned about the type of attack carried out.
The proposed model named HSIGFREE is signature independent and hence able to detect any attacks in the network on the server without any problems, being signature independent. The model acts as a filter, since it filters out any instructions and sequences causing damage to the server by the codes by sandboxing them and monitoring them. If they are found to be out of sync or causing attacks the HSIGFREE immediately detects them and blocks the executions. The commands are then dropped and the entire sequence is quarantined. This is how the damage is prevented by HSIGFREE before it spreads.

1.1 SIGFREE INTRODUCTION

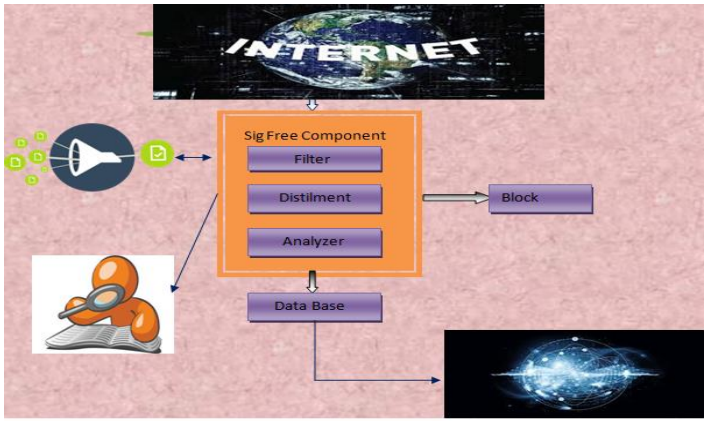


Fig.1.0 Signature free intrusion detection system.

In figure 1.0 depicts the architecture of Sigfree intrusion detection system, which includes the router deployment. The data coming from numerous host systems is first filtered for analyzing the possibility of attacker in the network. There are different component present were filter, Sequence instruction distiller and Sequence instruction analyzer in which the data coming from various host system is allowed to get into its component in a stepwise. The instruction sequence distiller can allow instructions that are not filtered are passed with the help of recursive traversal algorithm that can block the illegal and external address source. Even though some of the data anomaly instructions in it will enter the sequence instruction analyzer where code abstraction technique adopted which check the data anomaly during the flow of instruction, the data separated from executables is blocked and allowed to the destination host.

Algorithm 1

1. Initialize the instruction address array
2. Get the starting address and ending address
3. While (instruction address)
   Begin
   If the address is in the range
   Add the instruction sequentially
   Else
   If the address already visited
   Return
   Else
   Mark instruction as visited
4. If instruction is illegal then
   Mark its type illegal
   Else
   If legal instruction available
   Add Instruction address and perform operation

2 RELATED WORKS

Saurabh Chakradeo et al in their work developed MAST which can drastically lessen the costs involved in destroying malicious activity with different size malware composition of about 36,710, found 95% of malware at the cost of analyzing 13% of the non-malicious applications, and shown that successful triage extremely reduce the costs involved in removing malicious applications. Alireza Saberi et al in their work used three dissimilar methods to learn and detect phishing and developed ensemble methods for scam detection mechanism with successive detection rate of about 94.4% scam emails, while only 0.08% emails detected as scams. Sebastian Gajek et al in their work concentrated on identifying phishing attacks and scarcely counter the new malware phishing attacks. Sujata Garera et al in their work focused on the structure of URLs involved in various phishing attacks, used the filter to perform measurements on numerous URLs and decided whether it's a phishing attack without requiring any knowledge and extracted many features to discriminate a phishing URL from a good URL and found a model with high accuracy in detecting phishing sites. Ali Ikinci et al in their seminal work proposed an approach to build a database of threats found in Client-side attacks and malicious websites and evaluated the system by analyzing different crawls performed for three months period and present the lessons learned. Anh Le et al in their work developed a phishing detection system for that first selected lexical features that are resistant to attackers, second lexical features were used to evaluate the classification accuracy, third compared several classification algorithms and proposed an online method to overcome noisy training data, proposed PhishDef, a phishing detection system which is a highly accurate method compared to all existing. San Diego et al in their work described an automated URL classification, by means of statistical methods extracted suspicious URLs by repeatedly analyze and get lots of features with 95-99% accuracy, with only modest false positives. Niels et al in their work "All Your iFRAMEs Point to Us" studied the relationship exists in user browsing behavior and experience to malware.

3 PROPOSED WORK

The primary objectives of this project dealt in developing a significant security system which detects malicious attacks like code injection, ping, worm, flood etc. done through various layers by malicious clients and prevent them from causing damage apart from blocking them, thereby saving both the clients and their valuable data stored in the server. The scope of the study is to develop secure models which detect various attacks dynamically by sandboxing using HSIGFREE and then limit their actions with minimal false positives. The architecture diagram of the proposed system is depicted in fig 3.1.

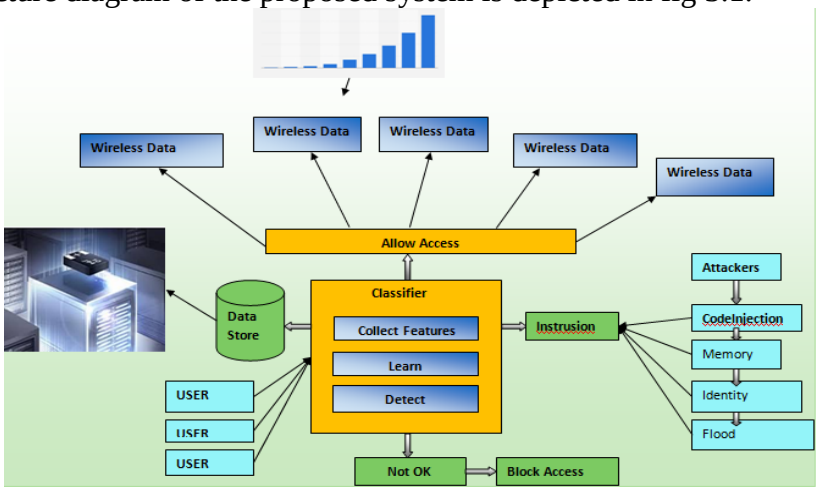


Fig: 3.1 Architecture diagram

The probability of various attacks emerges while using sensor nodes in unsecured environment increases that too in code injection. The objective of this work is to see the consequences of Code injection with the help of different parameters and which can be easily identified with the help of secure HSIGFREE IDS and even block them. In this paper some attacks on network with the issue of code injection, how to provide security against the code injection attack analyzed discussed.

The attacks which have been detected are as follows

- **Data Stealing -** Steal client data from the host computers and make it public for hacking purposes
- **Flooding -** The system is flooded with requests and folders are used to fill the empty spaces of the client drive so that storage is reduced.
- **Code Injection -** Code is injected into the client code so that new programs are started in the background to occupy Ram and processor and thus slow down the system.
- **Worm Hole Trojan Attack -** Here the files are destroyed by trojans causing damage to the client data.

Fig 3.2 shows the data flow diagram for the proposed work

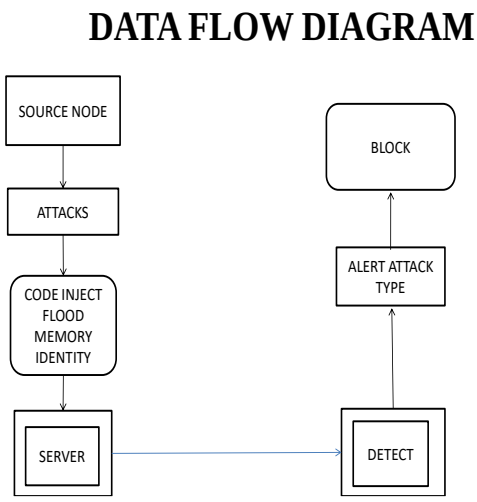


Fig 3.2 data flow diagram

The work carried out in this paper can be described below in stages:

3.1 CLIENT SERVER SETUP

In this module nodes are setup with information collection and dissemination to Server with the IP address of the server. The Nodes are placed based on the number of cluster heads. When the data transmission takes place, each node is assumed to send requests independently. During the transmission, the nodes can transmit the information and they also lose energy during transmission. When the network density is high the network performance and its lifetime extending can be carried out by the multi-channel.

3.2 ATTACKS MODULE

3.2.1 CODE INJECTION

In this type of attack the intruder injects the code into a batch file and then embeds it in the server without its knowledge. This code then starts executing automatically to start different programs in the server. So the server unknown and responds to the intruder's requests and various programs hog a lot of the memory space.

This causes immense strain to the server and ultimately it buckles under the pressure. Finally the server succumbs to the pressure and the intruder has the upper hand. The adversary blocks the started program from the server.

Code injection also causes false data to be given as parameters to the started programs which are in the self-executable batch files. This causes the programs to crash in the future.

3.2.2 IDENTITY ATTACKS

Here the intruders purpose is to steal the data from the server using masquerading techniques. The data may range from ordinary files to system files, user sensitive transaction data and also passwords. All the retrieved

data will be misused causing huge losses to both the integrity of the data server as well as the users of the server.

3.2.3 PING ATTACK

A ping is generally a command used to test whether the remote host can be reached from computer and useful for analyzing network connection with request message to a specified system and receiving a reply.

3.2.4 FLOOD ATTACK

A flood attack is a diagnostic check between two computers to identify the system without disruptions which usually using many computers.

3.3 SIGFREE DETECTIONMODULE

Once an intruder wish has been flagged as bad, an action is performed by the response system executed. The conventional actions are not severe actions, log the anomaly details or send an alert, but they do not dynamically prevent an intrusion.

Forceful actions, on the other hand, are capable of preventing an intrusion dynamically by dropping the request, disconnecting the user or revoking/denying the necessary privileges. Fine-grained response actions are neither too conservative nor too aggressive. A suspended request is simply put on hold, until some specific actions are executed by the user, such as the execution of further authentication steps. A tainted request is simply marked as a potential suspicious request resulting in further monitoring of the user and possibly in the suspension or dropping of subsequent requests by the same user.

3.4 REVOCATION

When the two digests are different, the intruder node's IP address will be blocked and the intruder's actions will also be prevented by passing the witness node with same claim Digest.

3.5 DETECTION

The proposed HSIGFREE algorithm detects any activity outside and first sandboxes the activities within a confined space. In case of any suspicious activity alerts are issued and the activity is contained and not allowed to execute thus preventing damage from the above type of attacks.

HSIGFREE ALGORITHM

Training phase

Step 1: First get the total layers.

Step 2: Do features selection separately for each layer.

Step 3: For each features selected train the layers repeatedly using step 2 with HSIGFREE.

Step 4: Accept the trained models linearly so that only normal can enter the next layer.

Testing phase

Step 5: Testing process repeat from steps 6 through 9.

Step 6: Each instance is tested and labeling done based on either it is attack or normal.

Step 7: When particular instance identified as an attack, block that specific instance with its layer name and go to Step 5. Else pick the next instance and repeat the same process to the next layer.

Step 8: Test the instance till the last layer in the system, and go to Step 7 if the current layer to be tested is found to be last then go to Step 9.

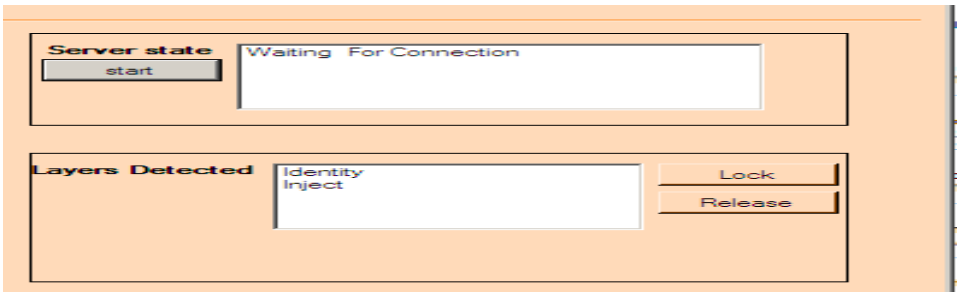
Step 9: Perform the same test as in step 7 and if the instance is labeled as an attack, block it with corresponding layer name.

4 RESULTS AND DISCUSSION

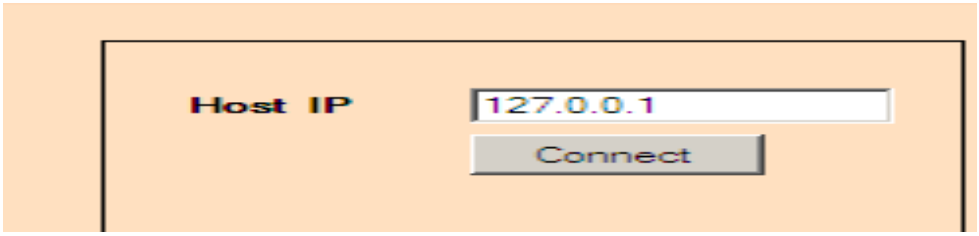
The primary objectives have been used to prevent breach of security model by detecting malicious attacks like code injection, ping, worm, flood etc. done through various layers by malicious clients and prevent them from causing damage apart from blocking them, thereby saving both the clients and their valuable data stored in the server. The model is effective as it is signature free and does not have prior knowledge about the type of attacks unlike traditional systems. The proposed model is signature independent and hence can

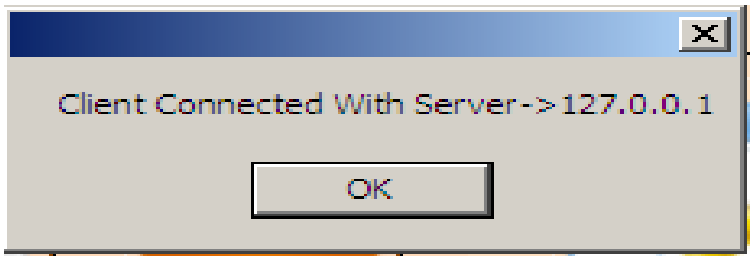
detect any type of attacks. First attack is prevented independent of the layer it comes from. Then only warnings are issued and attackers are revoked. Has less or insignificant computational cost and even lesser overheads. Does not have false positives or false negatives. The following were the screenshots of the results obtained.

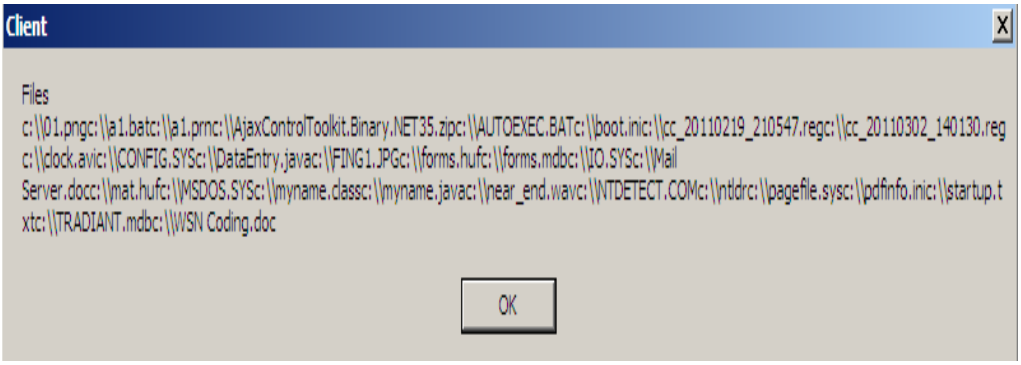
SERVER

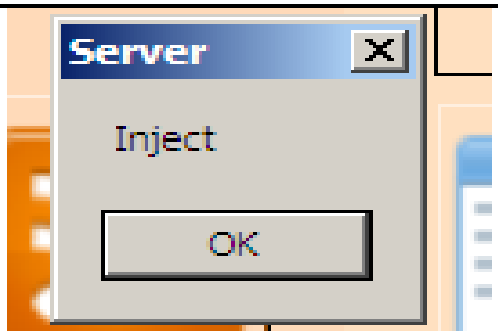


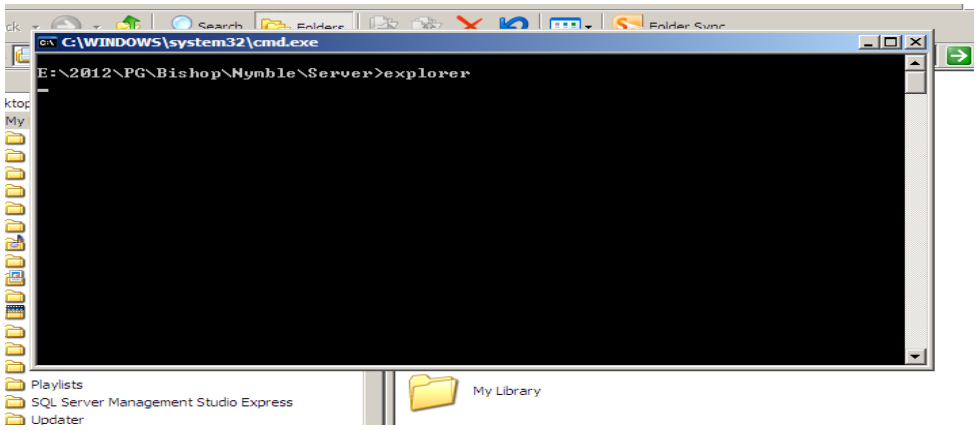
CLIENT





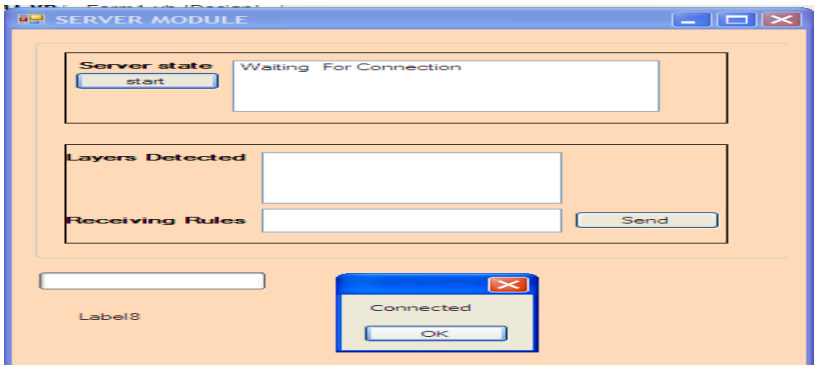


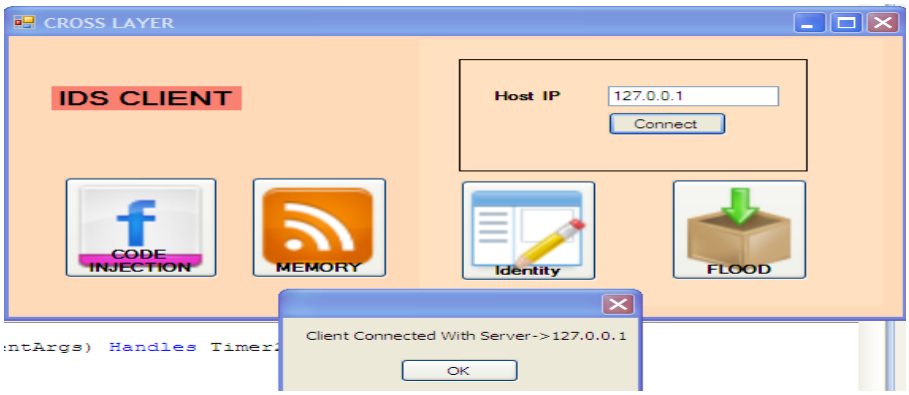

Server
Inject
OK

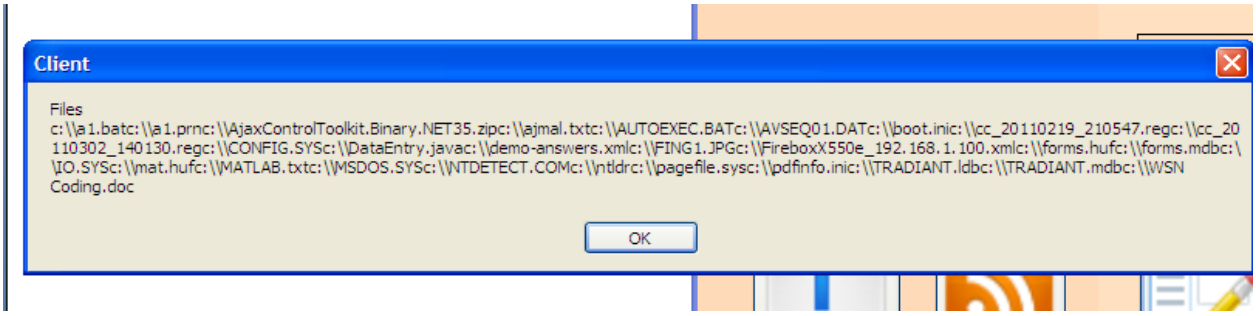

C:\WINDOWS\system32\cmd.exe
E:\2012\PG\Bishop\Nymble\Server>explorer
Playlists
SQL Server Management Studio Express
Updater
My Library

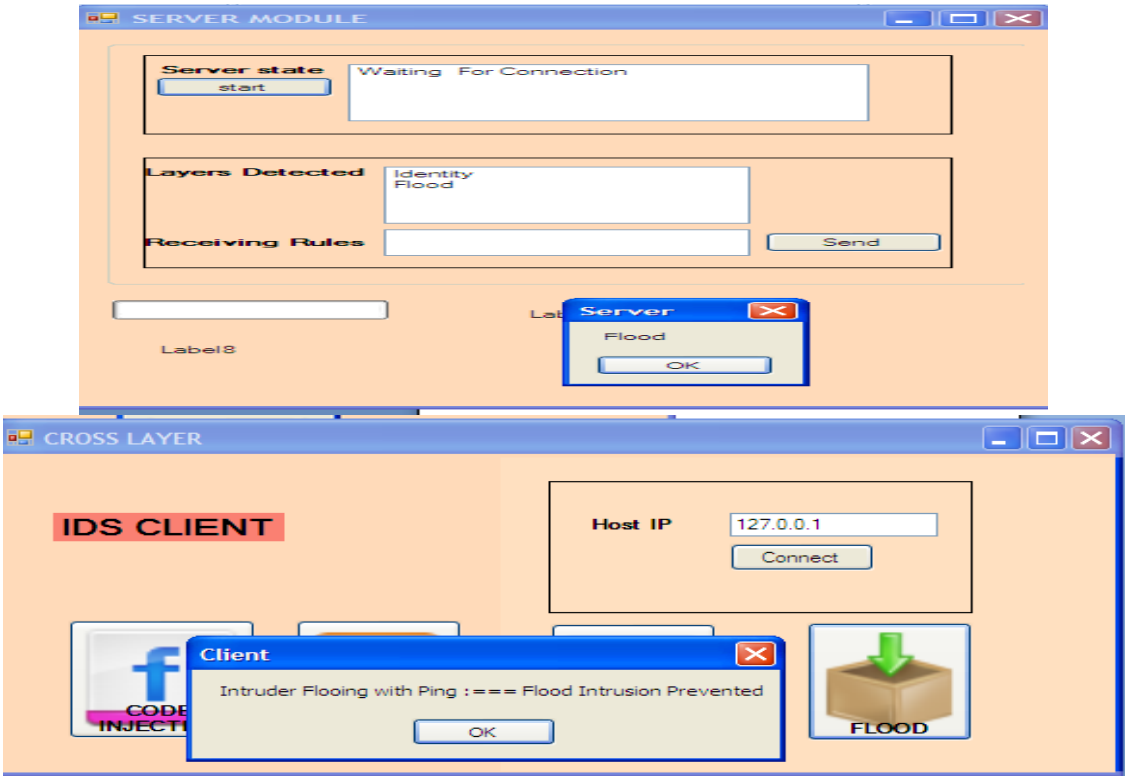

SERVER MODULE
Server state
start
Waiting For Connection
Layers Detected
Receiving Rules
Send
Label7
Label8


CROSS LAYER
IDS CLIENT
Host IP
127.0.0.1
Connect
CODE INJECTION
MEMORY
Identity
FLOOD


SERVER MODULE
Server state
start
Waiting For Connection
Layers Detected
Receiving Rules
Send
Label8
Connected
OK







5 CONCLUSIONS

The proposed model named HSIGFREE is signature independent and hence able to detect any attacks in the network on the server without any problems and prevents damages to the server caused by such malicious clients. Hence the proposed HSIGFREE model is a viable, safe and secure data and application layer protection and attack prevention model in dynamic server environments. The areas for future research include the use of our method for extracting features that can aid in the development of signatures for signature-based systems. This can further be extended to implement pipelining of layers in multicore processors, which is likely to result in very high performance.

6 REFERENCES

1) A. Ikinci, T. Holz, and F. Freiling. Monkey-spider: Detecting maliciouswebsites with low-interaction honeyclients. In Proceedings of Sicherheit,Schutz und Zuverlassigkeit, 2008.
2) L. Invernizzi, S. Benvenuti, M. Cova, P. M. Comparetti, C. Kruegel, andG. Vigna. Evilseed: A guided approach to finding malicious web pages. InProceedings of the 2012 IEEE Symposium on Security and Privacy, 2012.
3) P. Kolari, T. Finin, and A. Joshi. Svms for the blogosphere: Blog identification and splog detection. In Proceedings of AAAI Spring Symposium onComputational Approaches to Analysing Weblogs, 2006.
4) A. Le, A. Markopoulou, and M. Faloutsos. Phishdef: Url names say it all.In Proceedings of IEEE International Conference on Computer Communications(INFOCOM), 2011.
5) C. Lever, M. Antonakakis, B. Reaves, P. Traynor, and W. Lee. The core ofthe matter: Analyzing malicious traffic in cellular carriers. In Proceedingsof Network and Distributed System Security Symposium (NDSS), 2013.
6) P. Likarish, E. Jung, and I. Jo. Obfuscated malicious javascript detectionusing classification techniques. In Proceedings of Malicious and UnwantedSoftware (MALWARE), 2009.
7) C. Ludl, S. Mcallister, E. Kirda, and C. Kruegel. On the effectiveness oftechniques to detect phishing sites. In Proceedings of the 4th InternationalConference on Detection of Intrusions and Malware, and Vulnerability Assessment(DIMVA), 2007.
8) J. Ma, L. K. Saul, S. Savage, and G. M. Voelker. Beyond blacklists: Learningto detect malicious web sites from suspicious URLs. In Proceedings of theSIGKDD Conference, 2009.
9) D. K. McGrath and M. Gupta. Behind phishing: an examination of phishermodi operandi. In Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats (LEET), 2008.
10) E. Medvet, E. Kirda, and C. Kruegel. Visual-similarity-based phishing detection. In Proceedings of International Conference on Security and Privacy in Communication Netowrks (SecureComm), 2008.
11) Y. min Wang, D. Beck, X. Jiang, R. Roussev, C. Verbowski, S. Chen, and S. King. Automated web patrol with strider honeymonkeys: Finding websites that exploit browser vulnerabilities. In Proceedings of the Networkingand Distributed Systems Security (NDSS), 2006.
12) R.Sridevi, Rajan chattemvelli,"Genetic algorithm and Artificial Immune System – A combinational approach for network intrusion detection",IEEE,2012.
13) SigFree: A Signature-Free Buffer Overflow Attack Blocker Xinran Wang, Chi-Chun Pan, Peng Liu, and Sencun Zhu, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 5, NO. 4, OCTOBER-DECEMBER 2008.
14) P. H. Rathod , S. N. Dhage," Sigfree: Buffer Overflow Attack Detection", Second International Conference on Emerging Trends in Engineering (SICETE),IOSR- Journal of Electronics and Communication Engineering, PP: 54-58.