Int. j. inf. tecnol. (March 2022) 14(2):837–846
https://doi.org/10.1007/s41870-018-0182-4

ORIGINAL RESEARCH

HONEYDOS: a hybrid approach using data mining and honeypot to counter denial of service attack and malicious packets

Pratima Sharma[1] · **Bharti Nagpal**[1]

Received: 27 June 2017 / Accepted: 13 April 2018 / Published online: 20 April 2018

Abstract Honeypots and data mining are the major methods used as a safeguard of assets or for classifying the data. Each technique has positives and disadvantages of its own and is best applied in a particular position. Hybrid approach uses aspects of both techniques to upgrade performance and shortcomings. In this paper, we propose a hybrid approach based on Honeypot and Data Mining based Support Vector Machine technique, implemented in the dot net framework for preventing Denial of Service Attack. The proposed approach, HoneyDos tested in three interfaces. This paper presents an empirical comparison of the hybrid approach to the earlier methods used for preventing Denial of Service attack and draw useful conclusions upon their performance.

Keywords Denial of Service · HoneyDos · Interface · Malicious · SVM

1 Introduction

With the massive use of cyberspace and the increasing execution of distributed systems, most of the methods and mesh presently are sensitive to rise of security risk. In this respect, there are very few administrators who are not agreed that security has turned into a top option for them over the last decade. Presently in the corporate sector, there are all kinds of tools and solution to preserve the organization mesh safe from the various attacks. Apart from many freely available tools and software is being developed with new risks emerging every day.

Most of the large network organizations incorporated with firewalls and IDS in order to preserve their resources from illegitimate and intruder's activities. IDS trace the attacker's activities for identifying the unauthorized activity, but sometimes it runs out to describe the new methods, virus and worms used by the attackers because of complex encryption used. Therefore, Security's experts started the efforts for novel solutions; they desired to attract bad people into decoy server and attempt to hide their activities. This resulted into Honeypot technology.

The Honeypot is a decoy server or real system that presents in the network and exposed to the Internet that simulates utilities like HTTP, FTP, SSH, SMTP and many others. The concept behind is to show up the system to the attacker to attack, probe and compromised the system. It also logs the methods, packets or activity used by the aggressor to attack the system. Because the idea was initiated a few years back, therefore, very few corporate and freely tools are available in the marketplace.

The field of data mining and knowledge discovery is emerging as a new, fundamental research area with important applications to science, engineering, medicine, business, and education. Data mining attempts to formulate, analyze and implement basic induction processes that facilitate the extraction of meaningful information and knowledge from unstructured data.

Therefore, in this paper, a hybrid approach based on Honeypot and Data mining based Support Vector Machine technique is applied for separating the normal packet from the network traffic. The proposed approach, HoneyDos uses the SVM technique for identifying the malicious packets and also provides the counteractive step for preventing Denial of Service attack.

✉ Pratima Sharma
pratima.sharma1491@gmail.com

[1] CSE Department, GGSIPU University, Geeta Colony, Delhi, India

2 Related work

The application of honeypot falls into three categories: detection of new attacks, preventing system resources and preventing Dos attack. A number of researchers used the concept of honeypot to counter the new and known attacks.

Rajalakshmi Selvaraj et al. [1] have proposed an ant based DDoS detection technique using roaming virtual honeypots. In ADTRVH, a multi-level secure architecture is used for collecting information of various attackers with the help of virtual roaming honeypot at different network levels. Based on ant colony optimization, the collected information used by the multi level architecture to restrict the further connection of the attacker with honeypot or stop further spread of intruders.

Supeno Djanali et al. [2] have proposed a low-interaction honeypot for emulating vulnerabilities that can be exploited using XSS and SQL injection attacks. It also used browser exploitation techniques for exposing the identity of the attacker. Some attackers would use techniques to hide their identity, thus they couldn't be tracked. This honeypot was trying to overcome this problem by giving them malicious JavaScript codes. The malicious JavaScript codes will be run when an attacker open the honeypot's website.

Buvaneswari et al. [3] have proposed an iHoneycol, a collaborative solution for the early detection of flooding DDoS attacks by making use of "FirecolIPS" system and "Honeypot-IDS" system. It prevents the attack as close to the source and as far from destination, providing a protection to sub-scribed customers and saving valuable network resources.

Mitsuaki Akiyama et al. [4] have proposed a client honeypot system designed for achieving honeypot multiplication and implemented it in a field trial to evaluate its effectiveness. This system uses the proposed multi-OS and multi-process approaches. In particular, our process sandbox mechanism solved problems of process multiplication by providing a virtually isolated execution environment.

Almotairi et al. [5] have presented a low-interaction based honeypot technique for detecting new attacks in network traffic. The proposed detection method has performed in two parts. Firstly, based on IP addresses traffic flows are grouped and then PCA profile of honeypot traffic is built. Secondly, new traffic vectors are projected onto the residual space of the PCA attack model and the square prediction error (SPE) statistic is used to flag new attacks based on their large deviations from the attack model.

3 Problem specification

The goal of the proposed research is to prevent the resources of the system from malicious packets and Denial of Service Attack. As we all know network security is very prominent for all computer systems, because any unprotected machine in a network can be compromised in any moment. One may lose all the secret and prominent data of a company, which can be a big deprivation, and it is, likewise, really dangerous that someone else knows your personal information. Thus, we tried to resolve these problems by developing SVM based honeypot application that is freely used by the users to protect their private information. The intention of engendering a new application is to dispense with the current ongoing problem of Denial of Service attack and supply an efficient GUI based application primarily for Windows operating system because open source tools are very complex to configure and use.

4 Proposed approach

In this paper, we proposed a new HoneyDos Application with Support Vector machine technique to prevent the organization from malicious packets and Denial of Service attack. HoneyDos is a hybrid application that combine the two different technology into one platform that is Honeypot and Data mining SVM technique.

HoneyDos application is designed to catch the network traffic of the system on which it is deployed to prevent the resources of the system by capturing the malicious packets [6] and block them by stopping their sources using a blocker module. It is simply contrived to supply the good environment and continues up the blacklist by examining the approaching traffic and filter the traffic according to the blacklist maintained by the blocker module of the application. Support Vector based technique plays an important role by creating the vector class for each protocol and generates the threshold value for each vector class whenever the limit exceeds it starts discarding the packets. In this way, SVM used for preventing the Denial of Service Attack [7].

Figure 1 below illustrates the architecture of HoneDos Application and also give the detailed description of the overall flow of the application by using a modular approach.

The overall flow of our plan, which consists of four main phases as follows:

- *Packet analyzer* During this phase, network traffic is captured by the application and analyses it using Support Vector Machine based approach.

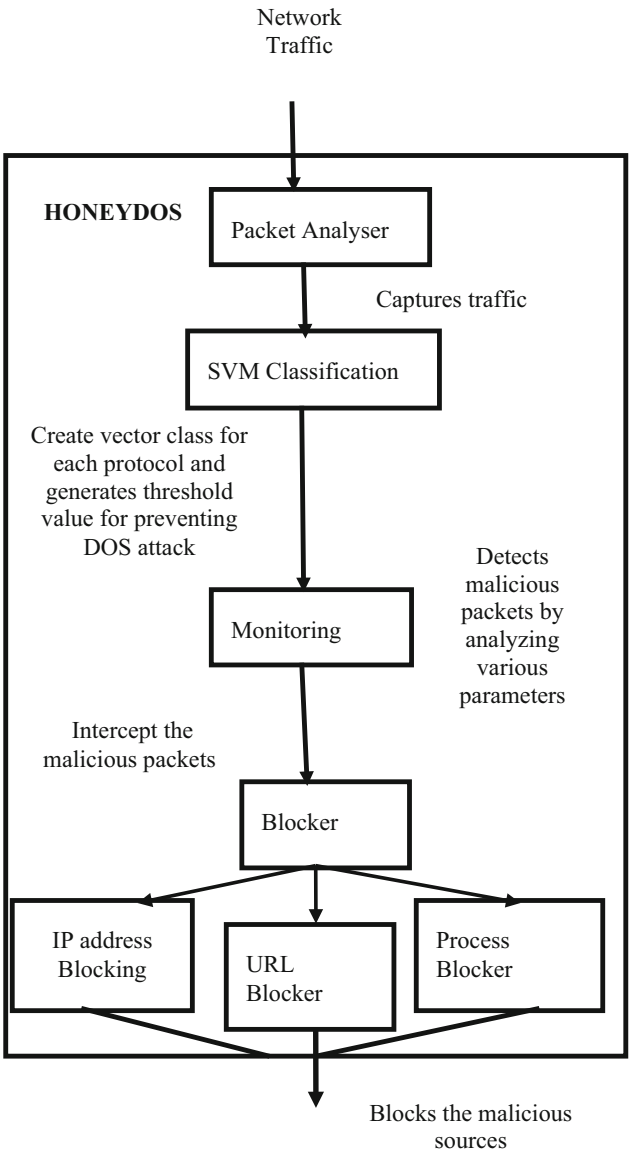

Fig. 1 Overall architecture of HoneyDos

- *SVM classification* In this phase, SVM classification algorithm is used for classifying the malicious and normal packets. It also creates the vector class for each protocol and set threshold value for each class whenever the limit exceeds it starts discarding the packets.
- *Monitoring* It detects the malicious packets by analyses the identification number and protocol ID and intercepts them to prevent the system.
- *Blocker* This stage offers the alternative for blocking the suspected IP address, URL, processes and services.

We will be discussing in detail each of the above-mentioned stages in the residual of this paper. Specifically, Sect. 3 outlines the Application Design features of HoneyDos application. Section 4 describes the implementation process and modules of the new plan. Section 5 presents the results and discussions. Lastly, Sect. 6 makes some closing remarks.

5 Application design

In this part, we explained the plan structure of HoneyDos application. We outline our decision of choosing a dot net framework and C# programming language for getting new and efficient application.

5.1 Choice of developmental language

We opted C# language as a developing language for a number of causes. Dot net technology provides soft to use and user friendly environment, mainly for Windows Operating System. It also provides excellent library for developing the application.

5.2 Choice of framework

A visual studio is an open-source framework that provides a modern environment required for developing innovative and effective application. It supplies all the features like code execution, error generation and management and software management.

5.3 Design decisions

In this section, we explained our thinking under design decisions that is employed for developing HoneyDos application. The Designing path represents an easy, simple and prolong able Honeypot application that is strong enough to execute many protocols and handles numerous utilities simultaneously.

5.4 DOS attack prevention

HoneyDos is designed with a support vector machine that is utilized to create the array of vector for each connected IP address and results notification if they deviated from normal traffic. This application design helps the users to deal with the denial of service attackers [8]. This application acts as a safeguard because the clients connecting to the HoneyDos application are not genuine one and may send malicious packets or launch fake connections [9].

6 Implementation of HoneyDos

HoneyDos is a hybrid application used for preventing the system resources from Denial of Service attack and malicious packets. It is designed to be exceedingly simple and

easily extendable. Our design structure reflects the desire to create an efficient application that applies the idea of honeypot and data mining technique usable for anyone with limited technical skill to elongate the functionality of the application according to the requirements of user.

6.1 Implementation features

HoneyDos supports the following basic characteristics:

- *Backup for multiple protocols* This application supports the multiple protocols and allows the user to access the several utilities smoothly. It, likewise, provides the readily extendable environment that facilitates the users to add on more roles as their prerequisite.
- *No limits to the number of client connections* HoneyDos support various nodes and likewise, leave the scheme to accept the multiple connections at the same time.
- *Graphical interface* HoneyDos provides graphical user interface for easy accessibility.
- *Configurable* The application is easy to configure to attach more options.

6.2 Working of modules

HoneyDos application is divided into four segments, and each module provides different functions to deal with the malicious and intruded packet. The first module is utilized for capturing the web traffic and redirects them to SVM module. The second module classifies the normal and fake packets and intercepts them in order to prevent the system from Denial of Service attack [10]. Then Blocker module is used for stopping the malicious source addresses and processes of the system that unknowingly executed by the parcels within the organization.

Figure 2 below depicts the overflow of working of Modules. It also identifies the role of each of the modules separately along with input and yield.

6.2.1 Packet analyzer and SVM classification modules

Packet analyzer is the main module of HoneyDos Application used for capturing the network packets. The support vector machine technique applied at the next step for creating the vector class of each protocol. Packet analyzer captures the network traffic and pulls up the features of captured packets. Then Support Vector Machine [11] applies the classification technique and create the vector class for each protocol. SVM is a form of statistical learning based on a structural risk minimization principle.

In this Module, packet analyzer captures the network traffic and extracts the features of each packet by analyzing

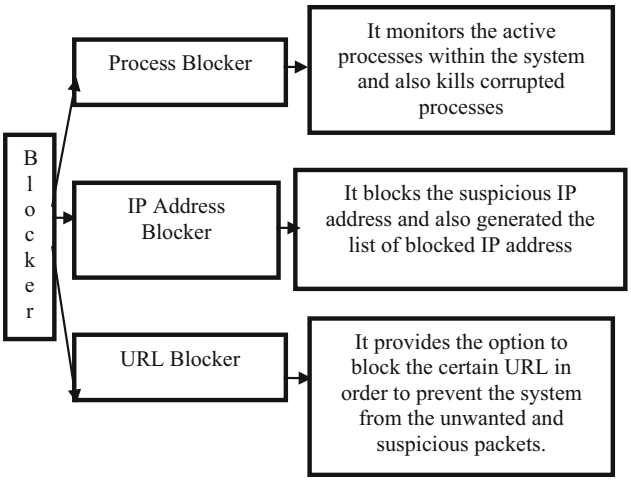

Fig. 2 Blocker sub modules

the header format of the packets. Then data mining based classification approach Support Vector Machine works on the extracted features of packets and creates the separate vector class on the basis of different features of packets. SVM uses real time parameters as shown in Table 1 to differentiate between the regular traffic and deviated traffic parameters in order to increase the efficiency of an analysis operation, we added three more parameters to increase the efficiency of the technique these parameters are Identification ID, Port ID and Transaction ID. SVM first calculates these parameters and set the threshold value for each protocol on the basis of number of bytes captured per unit time. The threshold value is the limiting factor and the interface crossing these limiting values are considered as attack suspect and will be intercepted by the monitoring module. So, when Denial of Service Attack launched by the attacker SVM technique is used for creating the threshold values for each protocol and intercepts the packets whenever the limit exceeds by the traffic. In this way, SVM technique is used for stopping Denial of Service attack and also for classifying between the malicious and normal traffic by examining the parameters like Identification ID, Port ID and Transaction ID these parameters must be dissimilar for different packets.

The Table 1 shows the calculation of SVM parameters. Every interface is monitored, and list of array vector is created by SVM for each interface by using the parameters defined in the above table. Then analyze the traffic along with the defined parameters determines the deviation from vectors created by SVM and reports that diverged packet as malicious one and intercept that packet.

Algorithm 1 shown below describes the Packet Analyses using Support Vector Machine, which is initialized with training data sets that contain the detailed packet format of each protocol along with parameters defined in the above table. In this algorithm, we have added three new

parameters in order to increase the efficiency of the detection process that parameters are Identification Number, Transaction ID and Port ID these are used for identifying the malicious packets and detecting the Denial of Service attack. SVM first creates individual classes for each protocol and the interfaces that deviate from these classes are considered as attack suspect and that interface is blocked by the administrator.

Algorithm 1: Packet Analyses using Support Vector Machine

Input: Network Packets

Output: Support vector classes

Parameters:

Packets_capture: It is a string that captures the network packets

Feature_extraction: It extracts the features of each captured packet

Vector_class: Svm creates multiple class stores them in vector_class parameter

Training_datasets: Contain predefined features or packet format of each protocol

```
Begin
1.  Loop
2.  Packet_capture <= network traffic
3.  Foreach (protocol in packet_capture)
4.    Create vector_class <= packet_capture          //create vector class using SVM logic
5.    Feature_extraction <= vector_class
6.    Compare (Feature_extraction, training_datasets)
      {
7.   If (training_datasets matches with Feature_extraction)
8.  Analyze (identification_number and transaction_id)   //must be different for different protocols
9.  Normal traffic                                       // consider as normal packets
10. Else
11. Redirect them to monitor module
12. End if   }
13. End For Loop
14. Return Vector_class
End
```

6.2.2 *Monitoring module*

In this Module, HoneyDos Application monitors the captured packets and examines them in order to get out the malicious packets [12]. It passes on the details close to the length of the packets along with protocol, source address, destination address and total packets intercepted by the HoneyDos Application.

Algorithm 2 shown below depicts the Identification of malicious packets from vector class and intercepts by redirecting them blocker module in order to add sources of suspected packets in the black list.

Algorithm 2: Monitored the vector_class and intercept the packets

Input: Vector classes

Output: Intercepted packets

Parameters:

Suspected_packets: It stores the intercepted packets

```
Begin
1. Monitored_packets <= vector_class
2. Foreach (Packet in vector_class. GetPacket())        //get packets from vector class created by SVM
3.    Examine(packets)                                  // examine each packet against set of parameters
4.    Suspected_packets <= Add(examine_packet)
5. End For loop
6. Foreach(Suspected_packets  in GetPacket())
7.    Identify(protocol, length, source, time){
8.    Intercepted_packets <= addfield()  }              // add the identified fields in intercepting packet list
9. End For loop
10. Return intercepted_packets
End
```

6.2.3 Blocker module

Blocker contains the four subparts as given under:

- *Process monitor* It monitors the active processes within the organization and also pops up the window whenever a new process plant in the organization. It, likewise, offers the option of voting down the dynamic process. It depends upon the administrator to identify the corrupted process and forcibly kill it by using process monitor option.
- *IP address blocker* It stops the suspicious IP address and also generated the list of blocked IP addresses and does not allow the packets to enter the system with the block IP address.
- *URL blocker* It provides the option to blank out the certain URL in order to prevent the arrangement from the undesired and suspicious packets. It further guaranteed the bandwidth utilized by the unusual packets by blocking them and also furnishes the option of preventing the Denial of Service attack by not leaving the unsolicited packets to introduce into the scheme.
- *URL manager* It is used for handling the list of out of use URL by the URL blocker. It offers the choice to save, deleted and updates the list of the blocked URL.

Algorithm 4 shown below links to the monitoring module and extracted the intercepted packets.

Table 1 SVM parameters

Parameters	Estimation
Total number of packets/duration	No. of packets reaching the server in particular duration
Maximum number of ICMP packets/duration	Maximum No. of ICMP packets reaching the server in particular duration
Maximum number of TCP packets/duration	Maximum No. of TCP packets reaching the server in particular duration
Maximum number of UDP packets/duration	Maximum No. of UDP packets reaching the server in particular duration
Identification number in each protocol	Analyze the identification number in each protocol that must be different
Transaction ID	Analyze the Transaction ID for each packet. Transaction ID must be dissimilar
Port ID	Analyze the Port ID for each packet. Port ID may be same

Algorithm 3: Blocks the suspected IP addresses and maintains the black list

Input: Intercepted packets

Output: Black list of suspected IP addresses

Parameters:

Suspected_packets: intercepted packets from the monitoring module is taken as input

Black_iplist: list of blocked IP addresses

Kill_process: list of forcibly killed processes

Begin

```
1.  Suspected_packets <= intercepted_packet
2.  Foreach (suspected_packets)
3.  Identify_ip (IP address in suspected_packets)
4.  { Black_iplist <= Add  ip_address }                    //add suspected IP address in the blacklist
5.  End For loop
6.  void addnewprocess()
        {
7.        List.Items.Clear()
8.   Foreach (Process proc in System.Diagnostics.Process.GetProcesses())      //identify the system processes
9.        List.Items.Add (proc.ProcessName )
10. End For loop
          }
11.  Foreach (Process proc   in Process.GetProcesses()
12.     if (proc.ProcessName == suspicious_process)
13.     Kill_process <=  proc.Kill()                                  //kill the suspicious process
14.    End if
15.  fillprocess()
16. End For loop
```

End

7 Results and analysis

In this section, we represent the results derived from the proposed HoneyDos application. Firstly, Table 2 gives the list of interfaces that are analyzed using the application. Then Table 3 gives the number of packets captured by each interface within the time duration 2 min and also represents them in a graphical form.

Table 2 lists the number of IP addresses within the range of network along with the operating system detail. It too indicates the reaction time from each IP address. It also indicates the number of interfaces monitored along with the IP addresses and interface code used for different IP address. In this application, we have monitored three interfaces and analyzed the packets on each interface. Interface 0 represents having contact with 192.168.0.104; interface 1 represents having contact with 192.168.0.105;

Table 2 Identification of IP addresses within the scope

Interface number	IP address	Response time (s)	Operating system
0	192.168.0.102	2	Window 8 professional
1	192.168.0.104	4	Window 8 professional
2	192.168.0.105	1	Window 8 professional

Table 3 Traffic monitoring

Interface	Interface 0	Interface 1	Interface 2
No. of Packets within the given time duration (2 min)			
TCP	2	81	50
UDP	67	220	143
ICMP	0	11	4
IGMP	3	7	5
Total packets	72	319	202
Total intercepted packets	65	289	176
Entire length of intercepted packets (bytes)	22257	78912	65542

interface 2 represents having contact with 192.168.0.102, respectively.

While monitoring the network, the ICMP, TCP, UDP, IGMP and total number of packets arrived are given in Table 3 with the duration in minutes. It reads the traffic from various IP's. It takes the total number of packets, ICMP, TCP, UDP and total number of intercepted packets for specific time duration of 2 min from the respective IP's. The traffic analysis for three interfaces can be shown below in Fig. 3.

In Table 4, we represent the list of blocked IP addresses, URL and processes. It gives the list of blacklisted IP addresses, URL and Processes. This list is used by the HoneyDos application for filtering the net packets.

7.1 Analysis

In this segment, we are comparing the proposed approach with existent protection solutions by considering the various parameters like technique used, attack detection, attack prevention, positives and negatives. The motive of comparing the existing approaches with HoneyDos is to reach the concluding remark about the efficient approach in terms of services supplied by them. In the following table, we compare the features of various techniques (Table 5).

Table 4 Blocked lists

URL	IP address	Process
http://www.innobuzz.com	189.14.65.162	OSPPSVC
http://www.rapidshare.com	210.101.131.231	Safetynut
http://www.cricinfo.com	104.131.66.240	Nvtray
http://www.wallpaper.com	120.236.148.113	alg

8 Conclusion and future work

In this paper, proposed application explained in detail, and also demonstrates the result of a hybrid approach, HoneDos Application at three interfaces. Proposed application is designed to catch the network traffic of the system on which it is deployed in order to prevent the resources of the system by capturing the malicious packets and halt them by stopping their sources. Various societies are using honeypot systems to protect the whole organization's network security, and researchers are making academic experiments on them. As we all know network security is very significant for all computer systems, because any unprotected machine in a network can be compromised in any moment. One may lose all the secret and influential data of a company, which can be a big deprivation, and it is, likewise, really dangerous that someone else knows your important

Fig. 3 Traffic analysis

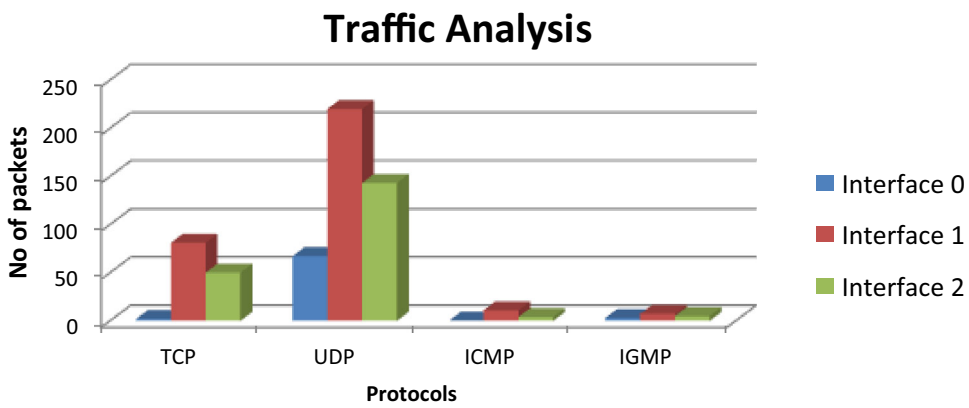

Table 5 Comparison between HoneyDOS and various techniques

S. no	Techniques	Year	Environment	Method used	Attack detection	Attack prevention	Type of honeypot	Positives	Negatives
1	Proposed approach (HoneyDos)	2017	Designed for user environment	Support vector machine based classification technique is used for identifying malicious packets and preventing DDoS attack	SVM creates the threshold values for each protocol and intercepts the packets whenever the limit exceeds by the traffic	Honeypot simulates the behavior of network and prevents the resources of the system	Low Interaction Honeypot	Suitable for preventing system from malicious packets	Require Administrator for handling blocking feature
2	Roaming Virtual Honeypot based DDoS detection technique [1]	2016	Multi-level architecture	Roaming virtual honeypot for collecting information about intruders	Ant-based optimization for detecting intruders	Multi-level architecture to create defense system	Virtual Honeypot	Reducing number of false positives and negatives	Complex solution
3	Analysis of Malicious traffic using Low Interaction Honeypot [8]	2015	VoIP Environment	Implementation of low interaction honeypot for monitoring illegal activities	Low interaction honeypot implemented in VoIP environment for detecting attacks	Prevent resources from malicious packets	Low interaction honeypot	Improve the defense mechanism such as firewall and intrusion detection system	Designed for specific environment
4	Web based Low interaction honeypot for preventing XSS and SQL Injection Attack [2]	2014	Web based application	Get the attacker's information using javascript code sent by honeypot at attacker's browser	Expose attacker's identity by using likejacking technique	Honeypot simulate the attacks and send the response as if the attacks succeeded	Low interaction honeypot	Prevent XSS and SQL injection attack	Skilled attackers bypass the honeypot security
5	IHoneycol for mitigation DDoS attack [3]	2013	Designed for corporate environment	Integration of Firecol and Honeypot technology	Overcome twin attack and ping of death attack	Integration of Firecol and Honepot helps in mitigation DDoS attack	Honeycol	Prevent Twin attack and Ping of Death attack	All clients register themselves with ISP and also send their IP address and location
6	Efficient client honeypot on high interaction system [4]	2012	High interaction system	Multi OS and multi process honeypot multiplication approach	Sandbox mechanism creates a virtually isolated environment for each web browser	Identification of malicious websites	Client honeypot	Improves performance efficiency and in depth analysis on high interaction system	Complex solution
7	Audit Data gathering based IDS [13]	2009	Designed for client side system	Auditing technique	Yes	No	Intrusion detection system	Simple approach for auditing packets	Poor performance

Table 5 continued

S. no	Techniques	Year	Environment	Method used	Attack detection	Attack prevention	Type of honeypot	Positives	Negatives
8	Misuse based IDS technique [13]	2009	Designed for client side system	Signature based approach	Yes	No	Intrusion detection system	Accurately generate much fewer false alarm	Cannot detect novel and unknown attack
9	Anomaly based IDS technique [14]	2009	Designed for client side system	Profile based approach	Yes	No	Intrusion detection system	Identify unknown attacks	High false alarm and limited by training data

personal information. Thus, we tried to resolve these problems by developing a hybrid application that combines the benefits of Honeypot and SVM techniques to protect the private information. HoneyDos is extremely elementary and easy to extend. One can prolong it by adding more security function according to the desire of users. This application can also be implemented in other languages.

References

1. Selvaraj R, Kuthadi VM, Marwala T (2016) Ant based distributed denial of service detection technique using roaming virtual honeypots. IET J. https://doi.org/10.1049/iet-com.2015.0497
2. Djanali S, Arunanto FX, Pratomo AB et al (2014) Aggressive web application honeypot for exposing attacker's identity. 1st International Conference on Information Technology 212–216
3. Buvaneswari M, Subha T (2013) IHONEYCOL: A distributed collaborative approach for mitigation of Ddos attack. International Conference on Information Communication and Embedded Systems (ICICES). https://doi.org/10.1109/icices.2013.6508281
4. Akiyama M, Kawakoya Y, Hariu T (2012) Scalable and performance-efficient client honeypot on high interaction system. 12th International Symposium on Applications and the Internet 40–50. https://doi.org/10.1109/saint.2012.15
5. Almotairi S, Clark A, Mohay G, Zimmermann J (2009) A technique for detecting new attacks in low-interaction honeypot traffic. Fourth International Conference on Internet Monitoring and Protection, pp 7–13. https://doi.org/10.1109/icimp.2009.9
6. Zhan Z, Xu M, Xu S (2013) Characterizing honeypot-captured cyber attacks: statistical framework and case study. IEEE Trans Inf Forens Secur 8(11):1775–1798
7. Agarwal P.K., Gupta B.B., Jain Satbir (2011) SVM Based scheme for Predicting Number of Zombies in a DDoS Attack. In: European Intelligence and Security Informatics Conference, 178–182. https://doi.org/10.1109/EISIC.2011.19
8. Vargas IRJS, Kleinschmidt JH (2015) Capture and analysis of malicious Traffic in VoIP environments using a low interaction honeypot. IEEE Latin Am Trans 13(3):777–783
9. Singh G, Sharma S, Singh P (2013) Design and develop a honeypot for small scale organization. Int J Innov Technol Explor Eng (IJITEE) 2(3):170–174
10. Wang J, Zeng J (2011) Construction of large scale honeynet based on honeyd. Advanced in Control Engineering and Information Science, Elsevier, pp 3260–3264
11. Li Zhu Yu, Ruixi Gaun Xiaohong (2007) Accurate classification of the internet traffic based on the SVM method. IEEE Int Conf. https://doi.org/10.1109/ICC.2007.231
12. Poongothai M, Sathyakala M (2012) Simulation and Analysis of DDoS Attacks. International Conference on Emerging Trends in Science, Engineering and Technology 78–85
13. Leu F-Y, Li Z-Y (2009) Detecting DoS and DDoS attacks by using an intrusion detection and remote prevention system. Fifth International Conference on Information Assurance and Security. https://doi.org/10.1109/ias.2009.294
14. Garcia-Teodoro P, Diaz-Verdejo J et al (2009) Anomaly-based network intrusion detection: techniques systems and challenges. Elsevier J. Comput Secur 28:18–28