Intelligent IDS: Venus Fly-trap Optimization with Honeypot Approach for Intrusion Detection and Prevention

Movva Sai Chaithanya
National Institute of Technology Goa
Suresh Nikudiya
National Institute of Technology Goa
Varsha S Basanaik
National Institute of Technology Goa
Damodar Reddy (dr.reddy@nitgoa.ac.in)
National Institute of Technology Goa https://orcid.org/0000-0002-5040-0745
Hanumanthu Bhukya
Kakatiya Institute of Technology and Science

Research Article

Keywords: Honeypot , IDS , IPS , Intruder , Malware , Venus Flytrap , Carnivorous plants

Posted Date: July 24th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-670906/v1

Noname manuscript No.
(will be inserted by the editor)

Intelligent IDS: Venus fly-trap Optimization with Honeypot approach for Intrusion Detection and Prevention

Movva Sai Chaithanya · Suresh Nikudiya · Varsha S Basanaik · Damodar Reddy Edla · Hanumanthu Bhukya

Received: date / Accepted: date

Abstract Intrusion Detection Systems and Intrusion Prevention Systems are used to detect and prevent attacks/malware from entering the network/system. Honeypot is a type of Intrusion Detection System which is used to find the intruder, study the intruder and prevent the intruder to access the original system. It is necessary to build a strong honeypot because if it is compromised, the original system can be easily targeted by the attacker. To overcome such challenges an efficient honeypot is needed that can shut the attacker after extracting his attack technique and tools. In this paper, a Venus fly-trap optimization algorithm has been used for implementing the honeypot system along with Intrusion Detection System. Venus plants are a type of carnivorous plants that catch their prey intelligently. By adopting this feature we make an effective honeypot system that will intelligently interact with the attacker. A new fitness function has been proposed to identify size of the attacker. The effectiveness of the proposed fitness function has been evaluated by comparing it with state of the art. For comparison, remote-to-local attacks, probing attacks and DOS attacks are performed on both proposed and existing models. The proposed model is significant to catch/block all the intruders which were caught by the art and also the proposed model reduces the time of in-

National Institute Of Technology Goa
Ponda - 403401, Goa, India E-mail: m.s.chaithanya1997@gmail.com

National Institute Of Technology Goa
Ponda - 403401, Goa, India E-mail: sureshnikuliya@gmail.com

National Institute Of Technology Goa
Ponda - 403401, Goa, India E-mail: twinvarsha@gmail.com

National Institute Of Technology Goa
Ponda - 403401, Goa, India E-mail: dr.reddy@nitgoa.ac.in

Kakatiya Institute of Technology & Science
Warangal, Telangana-506 015 E-mail: bh.cse@kitsw.ac.in

teraction between the attacker and honeypot system thereby giving minimum information to the attacker.

Keywords Honeypot · IDS · IPS · Intruder · Malware · Venus Flytrap · Carnivorous plants

1 Introduction

Security against cyber-criminals have become a very important issue as more and more new technologies are being invented. The attackers are finding new vulnerabilities to exploit the data or cause harm to the system. A vulnerability in a system/network can occur due to faulty design, coding error, improper protocol or due to backdoor function. To prevent the attacks [1] on any system/network, there is a need to understand and improve the security of the network/system. Hence, security tools such as intrusion detection systems (IDS) [3], firewalls [2], etc. help us to prevent most of the malicious activities from entering into the network/system.

A firewall [2] is the most widely used security tool for safeguarding against attackers on the internet. It is a physical device or software installed in the network/system which will check the incoming and outgoing network/system traffic for blacklisted and white-listed IP addresses and do the required action. All the blacklisted IP addresses are blocked by the firewall and all white-listed IP addresses are allowed to make the connection. But bypassing the firewall by either using IP spoofing or sending the malicious data in the data part of the packet can be an easy attempt to violate the firewall. These attacks are not identified by the firewall as it only checks the header part of the packet. On other hand, network Intrusion Detection/ Prevention System [3] is also a device or software used to identify/ prevent malicious activity from entering the network. They scan the header as well as the data part of packets entering the network for malicious activity. Intrusion prevention systems are an extension to the IDS[13], they can block the attacker or drop the malicious packets without alerting the administrator. Based on deployment in the network, design structure IDS [3] is divided into different types. Since we want to analyze all the network traffic entering the LAN we use Network Intrusion Detection System(NIDS). There are two types of intrusion detection systems based on the detection technique they are:

- Signature based detection: The signature based detection system just searches for the previously defined signatures in the packets based on the rules generated by using signatures. Limitations for using these detection system is that it takes a lot of storage space and the database needs to be updated always with all possible permutations of the signature.
- Anomaly-based detection: The anomaly based detection system have a previously defined behavior of data packets and if the packet deviates from this behavior it is identified as an attack by this system. Since there are many types of protocols involved for data transmission, it is really hard to classify the malicious and non-malicious data.

These systems can't give 100 percent accuracy due to design issues. Even if there is a 2 percent false positive rate the IDS system will generate 200 false alarms for every 10000 packets scanned and there will be around 12,000 packets entering through a 10mbps bandwidth connection per second. It is not easy to handle all these false alarms. Though we can achieve 100 percent detection accuracy for known attacks using signature-based IDS [3]. It has it's own drawbacks like if the signature of the attack deviates even slightly from the defined one this type of IDS can't detect it and also it increases the delay as it has to search for the signature of the packet in all the rules to classify a packet as good or bad. For this purpose, we intent to use honeypot. Honeypot [4] is another security tool kept as bait to lure attackers away from original systems towards the honeypot system by providing some dummy information and trap for them and to learn their techniques.

Many models have been proposed using IDS and honeypot in combination to improve the security strength of a network. Saurabh Kulkarni *et al.* [10] have created a new honeypot system called honeydoop. Honeydoop is a honeypot which uses IDS to identify the IP address on which the attacker is interested and creates a virtual honeypot with that IP address. It redirects the attacker to the newly created honeypot. The basis of their model is that the on-demand allocation of the honeypots at the right time and at the right place would make the network more secure and harder to sneak. But the problem with honeydoop is that the unknown attacks are not identified at all, requires a lot of virtual machines if there are a lot of attacks each performed on different IP address and also the false positive attacks of IDS are redirected to honeypot which might cause loss of important connections. Babak K *et al.* [11] have given a similar model of redirecting the attacker towards honeypot using routers for further analysis of the attacker. Their main aim was to reduce the false positive rate of the IDS. If it was a false alarm then traffic would again send to its original location. But there might be loss of some packets when the original user is redirected to the honeypot. Though the traffic at honeypot is reduced, the traffic at IDS has not. Georgios P. *et al.* [12] had created SweetBait which uses Sweetspot (a low interaction honeypot), Argos (a high interaction honeypot), HIDS, NIDS and NIPS systems for intrusion capture and containment. The main aim of their project is to automatically identify signatures of zero-day worms without human intervention which will reduce the damage caused by zero-day worms, reduce false alarms of IDS, continuously refine the worm signatures to provide automated signature revision. The worms aggressiveness is predicted by continuously monitoring its activity level which helps to sort the signatures in IDS based on the urgency level.

Bio-inspired algorithms like genetic algorithms, particle swarm optimization algorithms, etc. have been used to improve the performance of IDS. Vajiheh H. *et al.* [13] had proposed a new hybrid classification algorithm using Artificial Bee Colony algorithm and Artificial Fish Swarm algorithms for anomaly detection. Their model has improved the performance of IDS by decreasing false positive rate but computational overhead and time complexity is almost similar to other approaches. Wei Li [14] has described an approach for using

genetic algorithms in IDS. For identifying the complex anomalous behaviors, he had used both temporal and spatial information of the network connections for generating IDS[13] rules. Although there are many models in the art, here, a new model which uses Venus flytrap optimization [6] has been proposed with a new fitness function for identifying the size of an attacker. Venus Flytrap scientifically known as *Dionaea muscipula* is a carnivorous plant that captures insects. The Venus plant leaves contain two heart-shaped lobes each containing 2-3 hairs on its surface as shown in figure 1 [22]. On the surface of these

Fig. 1 Venus Flytrap [22]

lobes, the plant secretes honey like enzyme to attract the insects. When any prey comes in contact with the hairs present on lobes, it causes the trap to get into a semi-closed state and if the prey moves it will stimulate the hair again which will make the trap tighter and the trap goes into a completely closed state where the prey is digested. Semi Lehtinen [5] has provided the first mathematical cost-benefit model using the carnivorous behaviour of the Venus flytrap plants. He has analyzed the dynamics of prey capture, costs, and benefits of catching and digesting prey. Ruoting Y. *et al.* [12] have done mathematical modeling on the opening and closing behavior of Venus plants. They have analyzed the time taken by the trap to open, close and also time taken by the plant to transition from one state to another(open state, semi-closed state, closed state) mathematically. Ruoting Y. *et al.* [9] have also mathematically explained the opening and closing mechanism of Venus plants. Venus plant's behavior as an optimization technique has also been used by R. Gowri *et al.* [6], by mimicking the rapid closure behavior of Venus flytrap to capture the prey. The authors have proposed a type of Venus flytrap Optimization algorithm which was applied in [7] [8]. Venus plants enter into a semi-closed state

when the trigger hairs are touched once. When it is triggered again within 30 seconds of the first touch, it enters into a completely closed state. This behavior is called the rapid closure behavior of Venus plants. In their model when the hair has been touched, some charge is generated causing the Venus plant to enter into a completely closed state. The sum of charge generated during the first touch and after the second touch within certain time should be greater than some threshold, and the threshold is met only when the hair is touched twice within a certain period (30 seconds in the case of Venus plants trap to close).

In this paper, we have improved Venus flytrap optimization algorithm [6] by proposing a new fitness function that can be used in network security to analyze the attackers who are worth catching by the honeypot [4]. The rest of the paper is structured as section 2 presents related work. In section 3 the preliminary details are presented. The proposed method is presented in section 4. In section 5 the experimental results and in section 6 conclusion and future scope is presented.

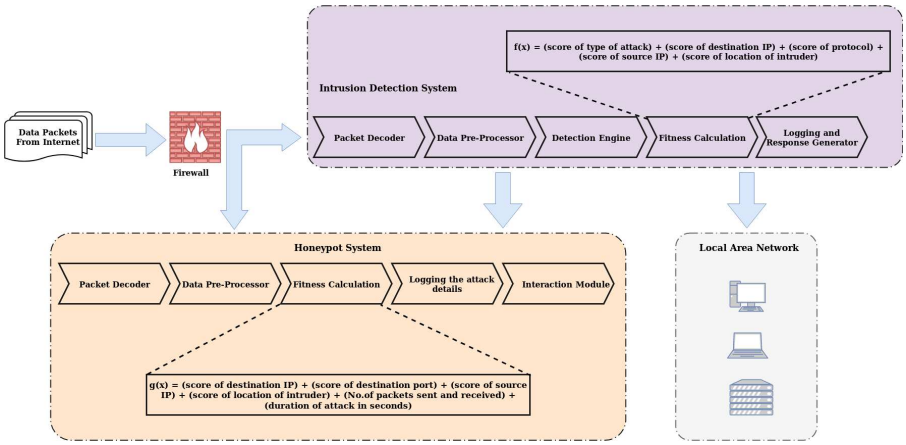


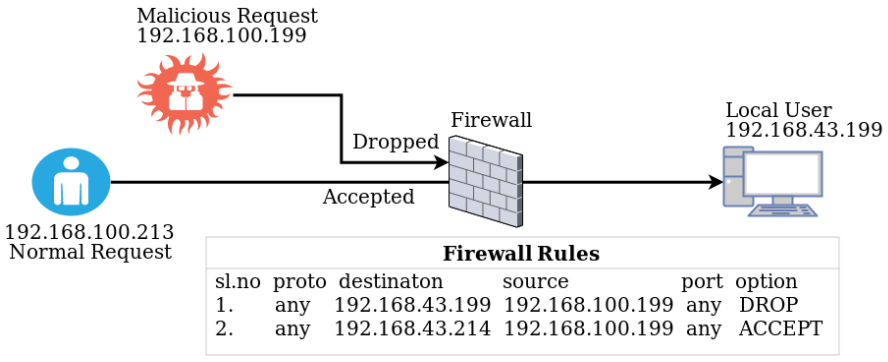
Fig. 2 Network Architecture of proposed honeypot system with Venus Flytrap Optimization

2 Intelligent Intrusion Detection System

The prey selection of Venus flytrap is mimicked in proposed algorithm. The algorithm has been made so that the honeypot can catch attackers who seemed to be potential, that is interacting with those attackers might give us some new information about the vulnerability/attack tools. The proposed network architecture is shown in figure 2. The process has been divided into 3 phases.

2.1 IP Blacklisting and White-listing Phase

First, the data packets coming from the internet will be checked at the firewall for the blacklisted and white-listed packets. If the packets are found to be blacklisted then those packets are either dropped or blocked by the firewall. Else the packets are allowed inside the network/system. By using the firewall we are blocking all the unwanted connections from the internet. The firewall contains a rule set of blacklisted (which are to be blocked) and white-listed (which are to be allowed) IP addresses. Whenever a packet comes to the firewall, it checks in the IP header part of the packet for the rules. If any of the rules are matched then it does accordingly. After the firewall, based on the destination IP address of the packet, it goes to either the intrusion detection phase or the honeypot interaction phase. For example, let us consider the situation shown in figure 3. Here, there is a connection request coming from IP address 192.168.100.199, 192.168.100.213 to our system at IP address 192.168.43.199. Now if the firewall contains the rules, "block any connection coming from source IP address 192.168.100.199 to any destination IP address" and "allow any connection coming from source IP address 192.168.100.213 to destination IP address 192.168.43.199" then the connection from 192.168.100.199 is blocked and 192.168.100.199 is considered blacklisted and connection 192.168.100.213 is allowed and it is considered white-listed.



Firewall Rules

sl.no	proto	destinaton	source	port	option
1.	any	192.168.43.199	192.168.100.199	any	DROP
2.	any	192.168.43.214	192.168.100.199	any	ACCEPT

Fig. 3 Example of IP blacklisting and white-listing phase.

2.2 Intrusion Detection Phase

In the intrusion detection phase each of the incoming packet is checked for malicious content in both header part and payload part. If no malicious content is found in a packet then only that packet is sent to its destination IP address located in the local area network. If any packet contains malicious content the fitness of that packet is calculated based on the fitness function f(x) given below and if the calculated fitness f(x) for a packet is found to be greater than lower

bound X1 and less than the upper bound X2 then the connection to which that particular packet belongs is redirected to the honeypot interaction phase. If the fitness is found to be greater than or equal to X2 (which represents big attack) then the administrator is alerted about the attack and the connection is blocked after entering the details of that packet into log file. And if the fitness is less than or equal to X1 (which represents small attack) then the connection is blocked after entering the details without alerting the administrator. IDS contains 5 components they are:

- Packet Decoder: Here the incoming packets are decoded into readable format and sent to next component
- Data Pre-Processor: It collects and formats the decoded packets which are sent to detection engine for analysis.
- Detection Engine: Here the packets are checked/analyzed for malicious content based on the provided rules.
- Fitness Calculation: Here the fitness of the malicious packets are calculated and entered into the log file.
- Logging and Response Generator: Here the malicious packet details are logged and corresponding response is generated. If there is no malicious content in the packet then it is sent to local area network.

The fitness of the attacker is calculated using the following formula, where scores are obtained from tables 1 and 2.

$f(x) = (scoreoftypeofattack)+(scoreofdestinationIP)+(scoreofprotocol)+(scoreofsourceIP) + (scoreoflocationofintruder)$

The fitness scores are given based on network vulnerabilities which can be changed as per the network requirements. Here the scores are set as per our network. We have given high priority to U2R and R2L than DOS, probing because in DOS, probing attacks the attacker generally doesn't interact with the system, it only tries to send unlimited requests in case of DOS or get system information in case of probing. So, they are not preferred over U2R or R2L attacks. Most of the packets use TCP or UDP protocol for normal message transmission. ICMP is mostly used to send error messages by network devices like routers. So, we gave ICMP least score.

The Score of source IP is obtained by searching the log file if the attacker's IP address is already present in the log file then he is a known attacker to us. He might know some information about the security of our network from the previous attack so we give him high priority over the new attacker. If the destination IP is admin/office system then it is given high priority over the normal user, since admin systems might contain valuable information. For the location of the intruder if the attacker is an insider(that is attacker is a local user) we give high score than external attacker as he might know some vulnerabilities and he need not go through the firewall.

For example, let us consider the situation in figure 4. Here the attacker having source address 192.168.100.199 (we consider attacker is unknown) is using RemoteToLocal attack on destination address 192.168.43.199 (we consider user at destination as normal user) using TCP protocol through destination port

21. At IDS(192.168.43.1) the fitness value of this connection is calculated as following,
score of type of attack = 3;
score of protocol = 3;
score of destination IP address = 1;
score of source IP address = 1;
score of location of intruder = 1;
fitness f(x) = 3 + 1 + 3 + 1 + 1 = 9;
And let X1 = 7, X2 = 14. Then the connection from attacker system to 192.168.43.199 is blocked and the attacker is redirected towards honeypot system having IP address 192.168.43.213 (as X1 < f(x) < X2). If f(x) is less than or equal to 7 then we just block the connection from attacker to local user. Else if f(x) is greater than or equal to X2 then we block the connection from attacker to local user and also alert the administrator.

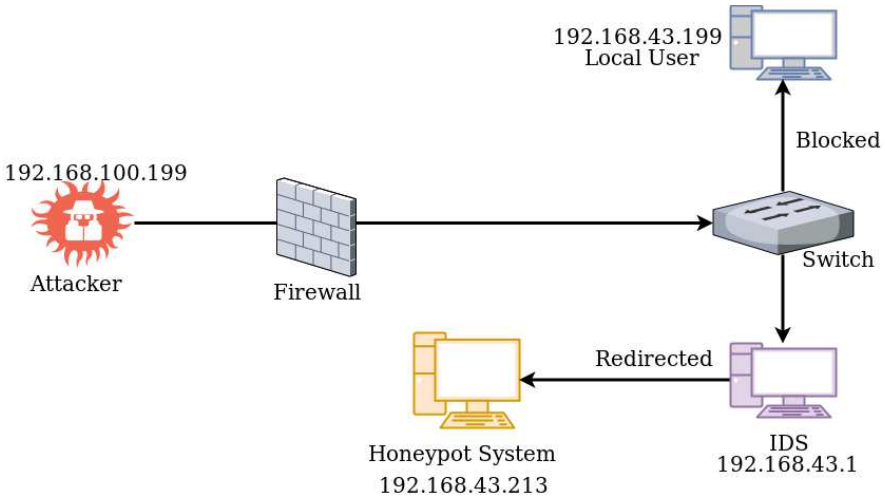


Fig. 4 Example of intrusion detection phase.

2.3 Honeypot Interaction Phase

In the Honeypot interaction phase every packet is considered as malicious. The fitness of the packet is calculated based on the fitness function g() given below and if the calculated fitness g(x) for a packet is found to be greater than lower bound X1 and less than the upper bound X2 then the connection to which that particular packet belongs is allowed to interact with the honeypot. If the fitness is found to be greater than or equal to X2 (which represents big attack) then the administrator is alerted about the attack and the connection is blocked

Table 1 Fitness scores

Type of attack	score	protocol	score	Dst Port	score
UserToRoot	5	tcp	3	ftp	3
RemoteToLocal	4	udp	2	http	2
DOS	3	icmp	1	irc	1
Probing attack	2				
Unspecified attack	1				

Table 2 Fitness scores

Dst IP	score	location	score	Source IP	score
Admin/Office	2	Internal	2	Known	2
Normal user	1	External	1	New attacker	1

after entering the details of that packet into log file. And if the fitness is less than or equal to X1 (which represents small attack) then the connection is blocked after entering the details without alerting the administrator. Honeypot

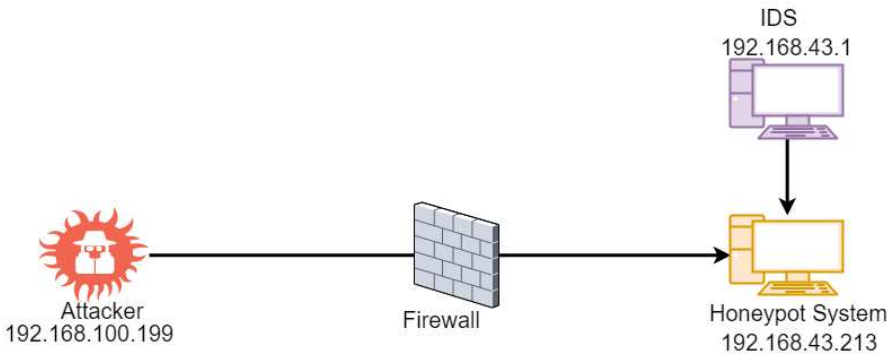


Fig. 5 Example of honeypot interaction phase.

contains 5 components they are:

- Packet Decoder: Here the incoming packets are decoded into readable format and sent to next component.
- Data Pre-Processor: It collects and formats the decoded packets which are sent to detection engine for analysis.
- Fitness Calculation: Here the fitness of the malicious packets are calculated.
- Logging and Response Generator: Here the malicious packet details are logged into the log file.

– Interaction Module: Here the response messages are generated and sent to the attacker (if necessary).

The flow chart of the process is given in figure 6. The fitness of the attacker at honeypot is calculated using the following formula, where scores are obtained from tables 1 and 2.

$g(x) = (scoreofdestinationIP)+(scoreofdestinationport)+(scoreofsourceIP)+$
$(scoreoflocationofintruder)+(No.ofpacketssentandreceived)+(durationofattackinseconds)$

Here type of the attack is not checked. Attacker interacts with the honeypot through the open ports, in our system we have kept FTP(21), HTTP(80), IRC(6667) ports open to lure the attacker. We can also provide more services like SSH, TELNET, etc. but for now, we are using these three services. As we interact with the attacker the no.of packets sent and received, time of interaction will keep on increasing. The attacker might compromise the honeypot if he keeps on interacting with the system so we use some parameters to know when to stop interacting like no. of packets sent and received, duration of the attack.

Algorithm 1 Improved Venus Flytrap Optimization Algorithm

```
for i = 1 : n all n preys do
    calculate fitness of prey g(x), x = (x1, ..., xd)
    Log prey details.
    while true do
        if (g(x) ≤ X1) then
            break
        if (g(x) ≥ X2) then
            • alert and break
        Interact with the attacker
        update fitness g(x)
return r
```

For example, let us consider the situation in figure 5. Here the attacker having source address 192.168.100.199 (we consider attacker is unknown) is performing an attack on destination address 192.168.43.213 (honeypot system) using TCP protocol through destination port 21. At honeypot(192.168.43.213) the fitness value of this connection is calculated as following,
score of destination port = 3;
score of protocol = 3;
score of destination IP address = 1;
score of source IP address = 1;
score of location of intruder = 1;
no.of packets sent and received = 2;
duration of attack in seconds = 0;
fitness g(x) = 3 + 1 + 3 + 1 + 1 + 2 + 0 = 11;
And let X1 = 10, X2 = 200. Then the connection from the attacker system to 192.168.43.213 to the honeypot system is allowed. If g(x) is less than or

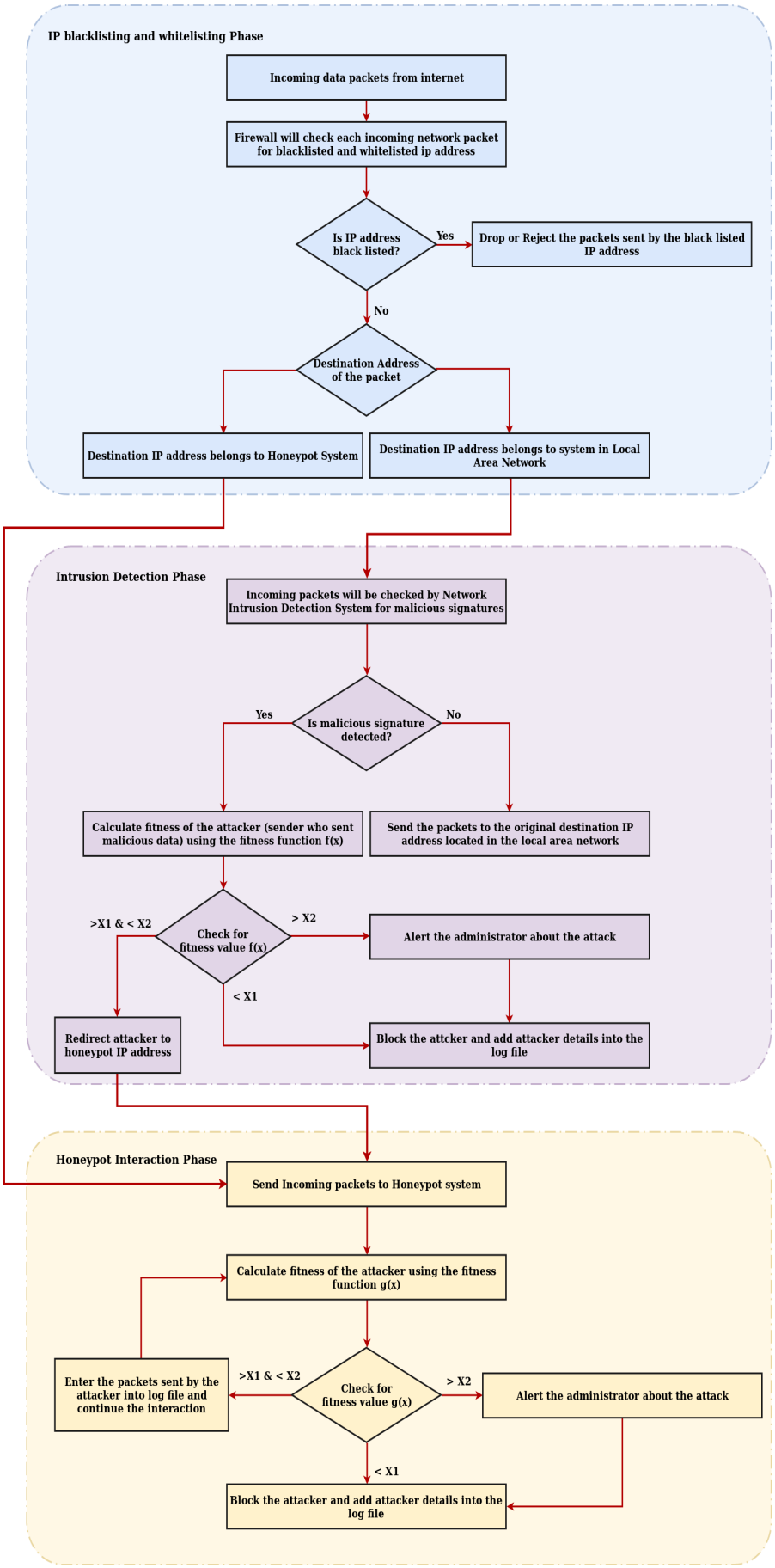


Fig. 6 Flow chart of proposed honeypot system with Venus Flytrap Optimization

equal to X1 then we just block the connection from the attacker. Else if g(x) is greater than or equal to X2 then we block the connection from the attacker and also alert the administrator. Here we allow the connection (as X1 < g(x) < X2) and allow honeypot to interact with attacker. As time passes and interaction goes on, the duration of the attack and no.of packets sent and received increases which increases the fitness of the attacker. Once the fitness of the attacker reaches X2 the connection with the attacker is blocked and the admin is alerted. The connection can also be stopped by the attacker or by the honeypot before fitness reaches X2 then we just log the attacker details without alerting the admin.

3 Experimentation and Results

3.1 Components Used

All the experiments are performed using the following components,

3.1.1 Honeypot System

"HoneyRJ" a low interaction honeypot has been used for the experiment. It requires an eclipse IDE(release version 4.11) to run. A system with Kali Linux OS with pre-installed eclipse IDE as Honeypot System has been utilized.

3.1.2 NIDS

"Snort", a signature based IDS has been used with a system having Ubuntu Linux OS. We can easily install Snort in any linux machine using the following command line,

sudo apt-get install snort \tab to install snort.
And to run Snort in NIDS mode the following command is used,

snort -A console -q -c etc/snort/snort.conf -l /var/log/snort/ -i wlan0 \tab to run snort in IDS mode and log packets

3.1.3 Local Area Network

Virtual box on the IDS system to simulate a LAN connected to a switch is used.

3.1.4 Firewall

An ip-tables firewall, which is an inbuilt firewall for all Linux machines is utilized. The following the syntax for appending a rule into iptables to block

an incoming connection,
iptables -t filter -A INPUT -s src_ipaddress -d dst_ipaddress -p protocol –dport src_port -j DROP

3.1.5 Attacker

Malicious pcap files are used for testing the IDS. For testing HoneyRJ different attacks are performed using a system with Kali linux OS (as it contains all the penetration testing tools).

Snort IDS is used at switch for listening on mirror port in NIDS mode. So, whenever snort identifies an attacker with fitness greater than X1 and less than X2, we will redirect that attacker to HoneyRJ using ip-tables by port forwarding.

3.2 Testing IDS

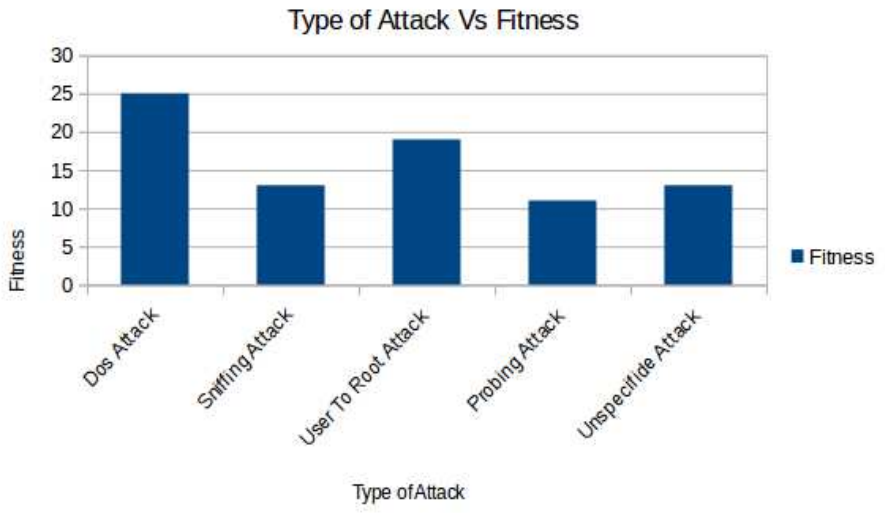


Fig. 7 Graph with DOS attacks given highest priority over other attacks

The graph in figure 7 shows the range of the fitness values vs the type of attacks in which the priority order is DOS attacks, U2R attacks, sniffing attacks, probing attacks, unspecified attacks and figure 8 shows the same, but the priority order of the attacks is changed to U2R attacks, R2L attacks, DOS attacks, probing attacks, unspecified attacks. Set the X1 value to 7 and X2 value to 14 at snort for redirecting attackers which captures most of the harmful attacks but again these values can be changed based on administrator preference. Scores for the type of attack is given based on the vulnerability of

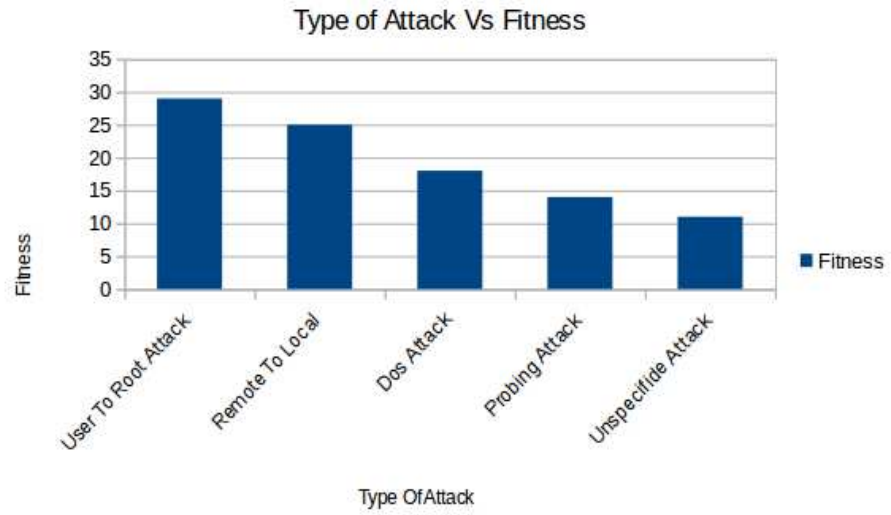


Fig. 8 Graph for which fitness scores for type of attack is shown in table I.

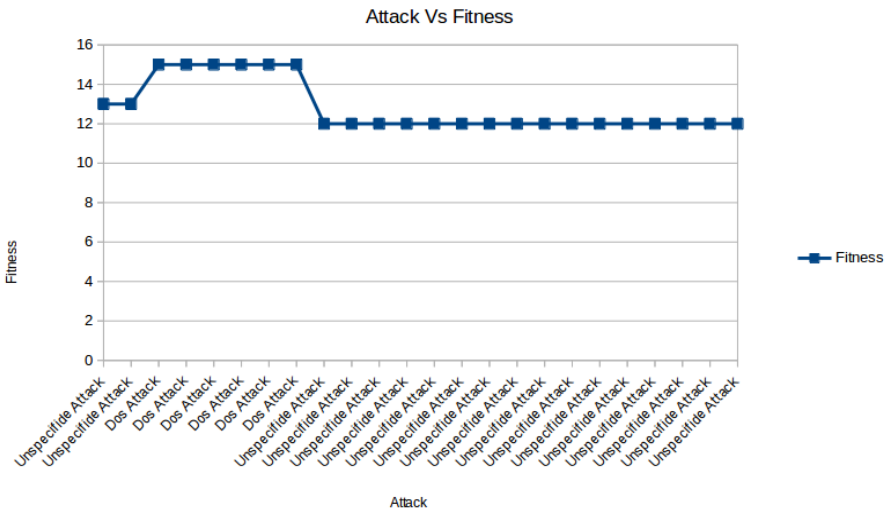


Fig. 9 Attack vs fitness graph for sample data shown in figure 10

the system/network which we want to protect. The pcap files of MACCDC [17] are used to test the fitness scores whose output is shown in figures 9 and 10. The U.S. National CyberWatch Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC) is a unique experience for college and university students to test their cybersecurity knowledge and skills in a competitive environment.

	Protocol	Type Of Attack	Attack	Source Address	destination Address	Source Port	Destination Port	Fitness
2	TCP	Unspecifide Attack	MasterCard number detected in clear text	192.168.47.171	192.168.47.134	1061	25	13
3	TCP	Unspecifide Attack	GIF in email	192.168.47.171	192.168.47.134	2826	25	13
4	TCP	Dos Attack	DOS flood denial of service attempt	66.201.189.32	130.68.81.253	48065	80	15
5	TCP	Dos Attack	DOS flood denial of service attempt	66.201.189.32	130.68.81.253	48073	80	15
6	TCP	Dos Attack	DOS flood denial of service attempt	66.201.189.32	130.68.81.253	48079	80	15
7	TCP	Dos Attack	DOS flood denial of service attempt	66.201.189.32	130.68.81.253	48066	80	15
8	TCP	Dos Attack	DOS flood denial of service attempt	66.201.189.32	130.68.81.253	48080	80	15
9	TCP	Dos Attack	DOS flood denial of service attempt	66.201.189.32	130.68.81.253	48149	80	15
10	TCP	Unspecifide Attack	PDF	212.227.84.95	172.16.121.162	80	8040	12
11	TCP	Unspecifide Attack	GIF	212.227.84.95	172.16.121.162	80	8040	12
12	TCP	Unspecifide Attack	GIF	77.72.118.168	192.168.47.171	80	2642	12
13	TCP	Unspecifide Attack	GIF	77.72.118.168	192.168.47.171	80	2644	12
14	TCP	Unspecifide Attack	PDF	212.227.84.95	172.16.121.162	80	8274	12
15	TCP	Unspecifide Attack	GIF	212.227.84.95	172.16.121.162	80	8274	12
16	TCP	Unspecifide Attack	PDF	212.227.84.95	172.16.121.163	80	1440	12
17	TCP	Unspecifide Attack	PDF	212.227.84.95	172.16.121.162	80	7595	12
18	TCP	Unspecifide Attack	GIF	98.139.134.174	192.168.47.171	80	2577	12
19	TCP	Unspecifide Attack	GIF	77.72.118.168	192.168.47.171	80	2739	12
20	TCP	Unspecifide Attack	GIF	77.72.118.168	192.168.47.171	80	2741	12
21	TCP	Unspecifide Attack	PDF	212.227.84.95	172.16.121.162	80	7666	12
22	TCP	Unspecifide Attack	GIF	212.227.84.95	172.16.121.162	80	7666	12
23	TCP	Unspecifide Attack	PDF	212.227.84.95	172.16.121.162	80	8097	12
24	TCP	Unspecifide Attack	GIF	212.227.84.95	172.16.121.162	80	8097	12
25	TCP	Unspecifide Attack	PNG	212.227.84.95	172.16.121.162	80	8097	12

Fig. 10 Snort log file for different attacks

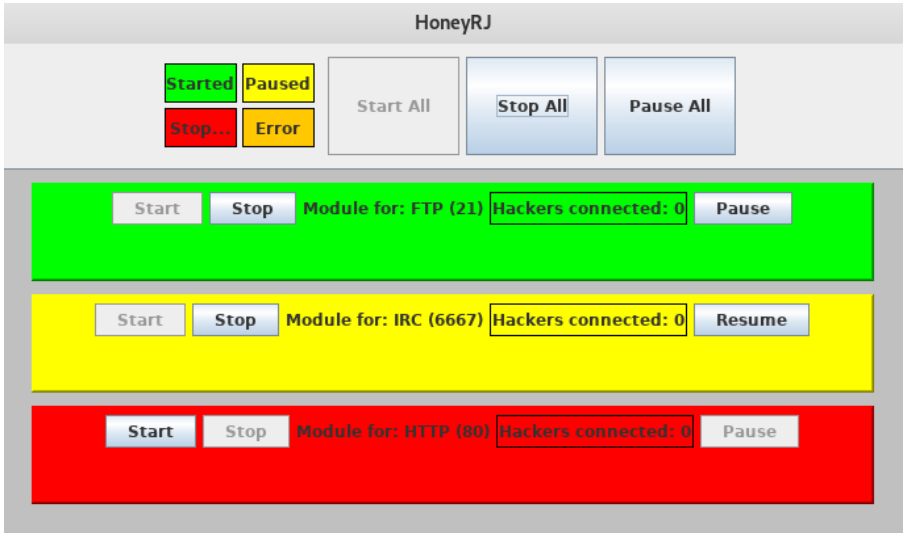


Fig. 11 HoneyRJ

3.3 Testing Honeypot

HoneyRJ is an open source low interaction honeypot written in Java for implementing the proposed honeypot algorithm. It provides only two services FTP and IRC. We have added HTTP, Sample Client First protocol, Sample Server First protocol services as well to this so that it can provide more services. It has GUI built into it which makes it more user-friendly. We can start, stop,

pause individual service or all the services using the GUI as shown in figure 11.

When HoneyRJ starts, it open the ports 21(FTP), 6667(IRC), 80(HTTP), 65001(Sampl Client First Protocol), 65000(Sample Server First Protocol) and starts listening to these ports for attacks. When an attacker tries to make connection, HoneyRJ will calculate fitness. If the fitness is greater than X1(10) and less than X2(150) then the reply message is sent based on the interaction module in HoneyRJ and if it's fitness is not in the range of (X1,X2) then the connection is rejected/blocked. The following are the interaction modules which are present in the HoneyRJ.

– FTP service Interaction: FTP service runs on dedicated port 21. So, when a user connects to HoneyRJ through 21 this module will start to interact with the attacker. The interaction process is shown in figure 12. Here the attacker is 192.168.43.232 and his fitness after the connection has ended is 47. It has increased from 14 to 47. The interaction was stopped because the attacker has entered into quit connection state in the interaction module.

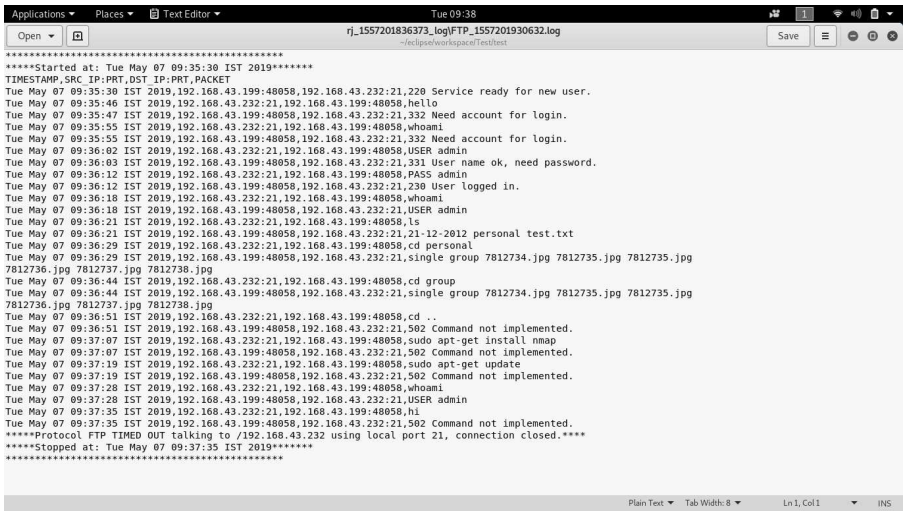


Fig. 12 FTP interaction process

– IRC service Interaction: IRC service runs on dedicated port 6667. So, when a user connects to HoneyRj through 6667 this module will start to interact with the attacker. The interaction process is shown in figure 13. Here the attacker is 192.168.43.232 and his fitness after connection has ended is 27. It has increased from 13 to 27. The interaction was stopped because the attacker has entered into quit connection state in the interaction module.
– HTTP service Interaction: HTTP service runs on dedicated port 80. So, when a user connects to HoneyRJ through 80 this module will start to interact with the attacker. The interaction process is shown in figure 14. Here

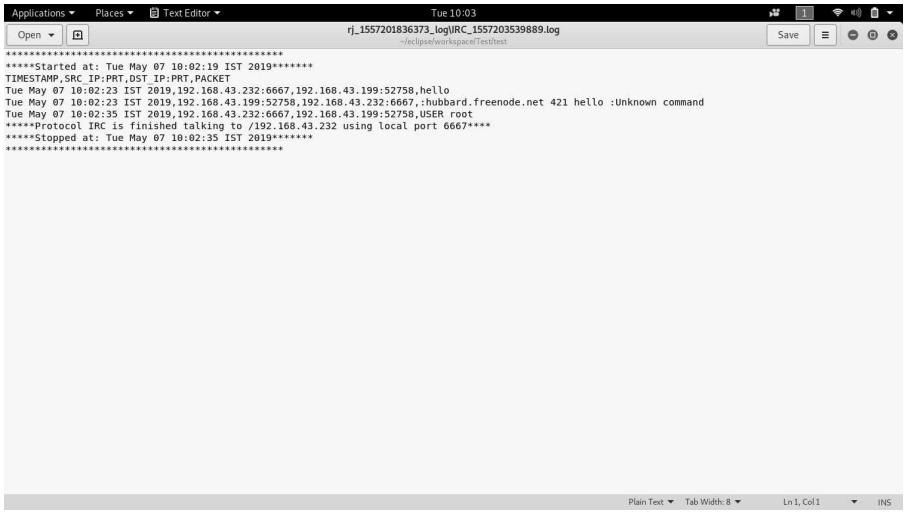


Fig. 13 IRC interaction process

the attacker is 192.168.43.232 and his fitness after connection has ended is 24. It has increased from 12 to 24. The interaction was stopped because the attacker has entered into quit connection state in the interaction module.

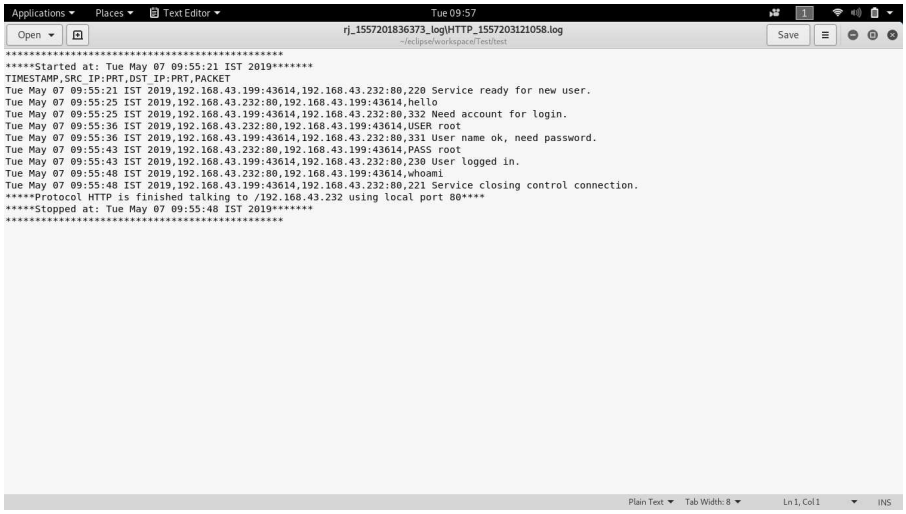


Fig. 14 HTTP interaction process

– Sample Client First Protocol Interaction: Sample Client First protocol service is given port 65000. So, when a user connects to HoneyRJ through 65000 this module will start to interact with the attacker. The interaction

process is shown in figure 15.

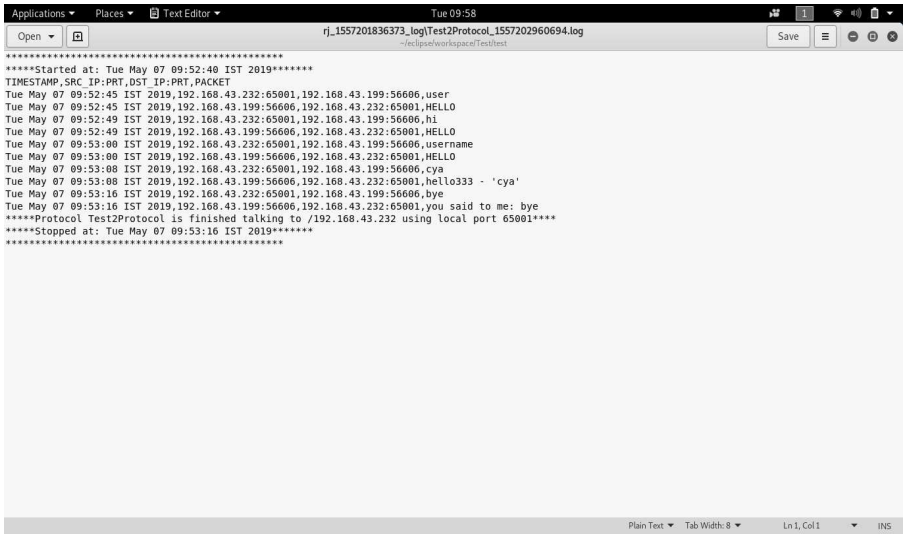


Fig. 15 Sample Client First protocol

– Sample Server First Protocol Interaction: Sample Server First protocol service is given port 65001. So, when a user connects to HoneyRJ through 65001 this module will start to interact with the attacker. The interaction process is shown in figure 16.

The Sample Server First Protocol, Sample Client First Protocol are testing protocols used in the HoneyRJ software for testing the interaction modules. As you can see in interaction modules, HoneyRJ is interacting with the attacker at the same time fitness is being calculated, so that we know whether to continue interaction or block it. The following attacks have been performed on the honeypot system to test the working of the proposed model:

– Traceroute: This attack can be classified as a probing attack. A traceroute attack allows you to find out precisely how a data transmission (like a Google search) occurred from your computer to another. Quite simply, the traceroute outputs a list of the systems on the network that are involved with specific internet activity. So, by using this attack the attacker can discover a route to another host. **Terminal Command: Nmap -A 192.168.100.199(IP Address of Honeypot Machine).**
– Remote System Access: Remote System Access is used to remotely operate your system from another system using your login credentials from any location. But this feature is being exploited by the attacker to access your computer by guessing the login credentials or by using a brute force attack. This is due to the use of weak/common login credentials. **Terminal**

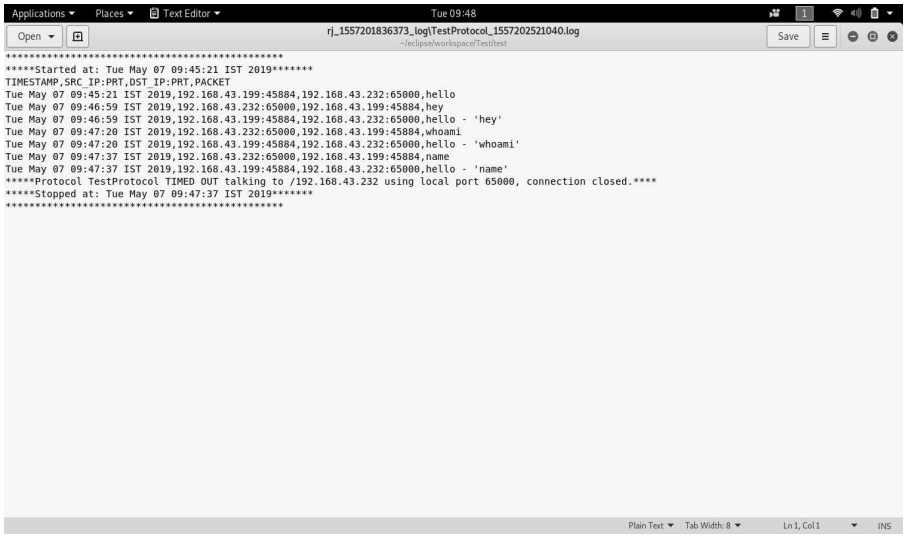


Fig. 16 Sample Server First Protocol

Command: Nmap -A 192.168.100.199(IP Address of Honeypot Machine).

- Script Scanning: It is a type of probing attack. It is used by the attackers to discover information about your network, detects more sophisticated and accurate OS version, identifies vulnerabilities in your network/system. **Terminal Command: Nmap -sC 192.168.100.199 //(IP Address of Honeypot).**
- Version/Service Scanning: It is a type of probing attack. By using this attack the attacker can get to know about the version of a particular service or software running on your system. **Terminal Command: Nmap -sV 192.168.100.199 //(IP Address of Honeypot).**

Table 3 shows the no.of packets captured when we test the HoneyRJ with our fitness function and without our function. And the graphical representation of

Table 3 Table showing no.of packets sent and received when performed an attack.

Type of attack	**No.of Packets With Fitness**	**No.of Packets Without Fitness**
Traceroute	2	16
Remote System Access	4	14
Script Scanning	7	22
Version/Service Scanning	2	14

the same is shown in figure 17. As it is clear from the figure less interaction can

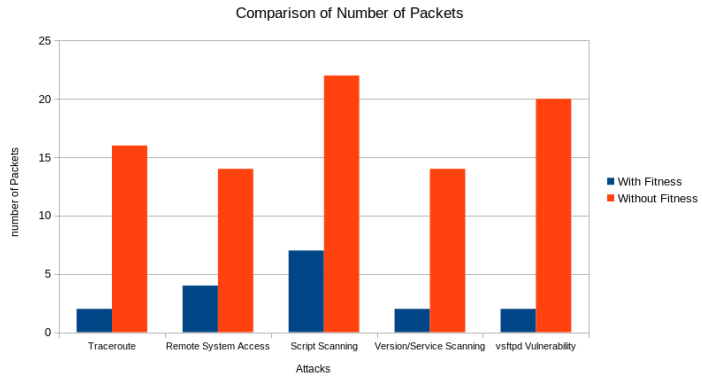


Fig. 17 Graphical representation of data present in table 3

be visible with the attacker when the proposed optimization algorithm is used as compared to the existing model. When an attack is performed whose fitness is below X1, the attacker will be allowed to interact with the honeypot system using the existing models, but using the proposed optimization technique the attacker will be stopped immediately as shown in figure 18.

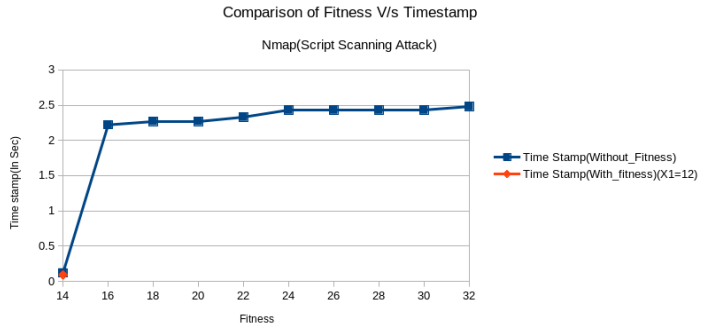


Fig. 18 Graph showing the change in fitness value for Script Scanning attack for proposed(red) and existing(blue) models.

Here, a Script Scanning Attack is performed which will not harm the system but will give information about the vulnerabilities of the system/network to the attacker which might be useful to perform an active attack. Interacting with this type of attack will not provide us with any useful information as in this attack empty/request/acknowledge packets are sent to identify the vulnerabilities. When an is performed attack whose fitness is in the range of X1 and X2 then the interaction process of the proposed optimization technique and existing model is similar, until the fitness value reaches X2. When the fitness value reaches X2 our proposed model will stop the attacker and alert the

administrator while the existing model will keep on interacting with attacker until the attacker stops the interaction as shown in figure 19.

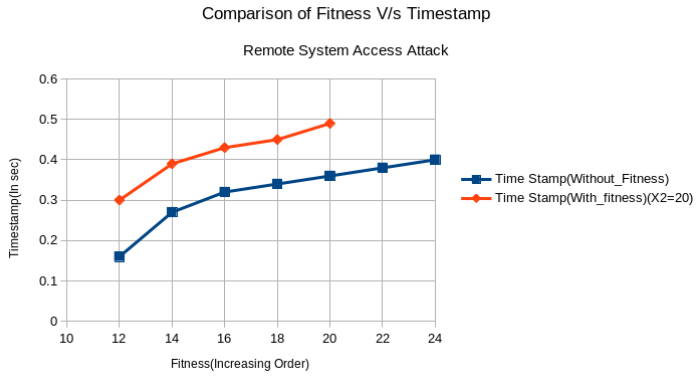


Fig. 19 Graph showing the change in fitness value for Remote System Access attack for proposed(red) and existing(blue) models.

A Remote System Access attack has been performed through open port 21. In proposed model, the interaction has been stopped because further interaction might cause damage to the honeypot system or the attacker might compromise the honeypot system and use it as a bot to attack other systems.

4 Conclusion

In this paper, Venus flytrap optimization technique has been adopted for the honeypot system. To perform this, a new fitness function is proposed which uses features like destination IP address, source IP address, destination port number, protocol, type of attack, no.of packets sent and received, duration of attack and location of the intruder. The interaction is established with only the effective attackers, skipping the small and the large attacks. As a result of several experiments, it is observed that the proposed model is performing well than the existing model. In proposed model, attacks such as nmap scanning, script scanning were blocked and attacks like remote system access were allowed to interact for some time whereas, in the existing model all the attacks were allowed to interact with the honeypot system until the attacker manually disconnects from the system. When we compared the no.of packets exchanged between honeypot and attacker, the proposed model was able to get information about attacker with less data exchange than the existing models. The interaction process is improved and the honeypot system is used effectively without wasting time on small prey. It is able to protect itself before an attacker causes serious damage to it. By redirecting the attacks to the honeypot system, we are able to safeguard the original system and also get to learn more

details about the attacker. Though the proposed model show us good results, it can be improved further by adding more features to obtain the size of an attacker more accurately.

Acknowledgment

The authors would like to thank Ms. Annushree Bablani, Mr. Kannadasan K., and Dr. Venkatanareshbabu Kuppili for their help in executing this research.

Declarations

Funding: Not applicable
Conflicts of interest/Competing interests: Not applicable
Availability of data and material: Not applicable
Code availability: Not applicable
Ethics approval: Not applicable
Consent to participate: Not applicable
Consent for publication: Not applicable

References

1. Goyal, Priyanka, Sahil Batra, and Ajit Singh. "A literature review of security attack in mobile ad-hoc networks." International Journal of Computer Applications 9.12 (2010): 11-15.
2. Ingham, Kenneth, and Stephanie Forrest. "A history and survey of network firewalls." University of New Mexico, Tech. Rep (2002).
3. Sabahi, Farzad, and Ali Movaghar. "Intrusion detection: A survey." In 2008 Third International Conference on Systems and Networks Communications, pp. 23-26.
4. Snehil vidwarshi, Atul Tyagi, Rishi Kumar, "A Discussion about Honeypots and different models based on Honeypots", Journal of Advanced Computational Engineering and Networking, ISSN: 2320-2106, Volume-3, Issue-8, Aug.-2015: 32-40.
5. Lehtinen S., "Understanding the Venus flytrap through mathematical modelling." Journal of theoretical biology 444, 2018. 1-10.
6. Gowri, R., and Rathipriya, R., "Venus flytrap optimization." Computational Intelligence, Cyber Security and Computational Models. Springer, Singapore, 2016. 519-531.
7. Gowri, R., Sivabalan, S., and Rathipriya, R., "Biclustering using venus flytrap optimization algorithm." Computational Intelligence in Data Mining—Volume 1. Springer, New Delhi, 2016. 199-207.
8. Sivabalan, S., R. Gowri, and R. Rathipriya. "Optimizing Energy Efficient Path Selection Using Venus Flytrap Optimization Algorithm in MANET." Computational Intelligence in Data Mining—Volume 1. Springer, New Delhi, 2016. 191-198.
9. Yang, R., Lenaghan, S.C., Zhang, M. and Xia, L.,"A mathematical model on the closing and opening mechanism for Venus flytrap" Plant signaling & behavior, 5(8), 2010, pp.968-978.
10. Kulkarni, S., Mutalik, M., Kulkarni, P. and Gupta, T. "Honeydoop-a system for on-demand virtual high interaction honeypots". In 2012 International Conference for Internet Technology and Secured Transactions, pp. 743-747, 2012.
11. Khosravifar, B., and Bentahar, J., "An experience improving intrusion detection systems false alarm ratio by using honeypot." 22nd International Conference on Advanced Information Networking and Applications (aina 2008), pp. 997-1004, 2008.

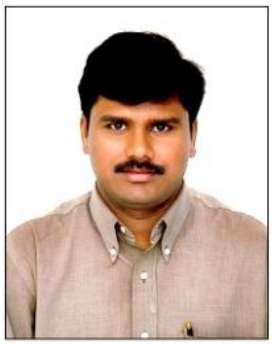
12. Portokalidis, G. and Bos, H., "SweetBait: Zero-hour worm detection and containment using low-and high-interaction honeypots." Computer Networks, 51(5), pp.1256-1274, 2007.
13. Hajisalem, Vajiheh, and Shahram Babaie. "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection." Computer Networks 136 (2018): 37-50.
14. Li, Wei. "Using genetic algorithm for network intrusion detection." Proceedings of the United States Department of Energy Cyber Security Group 1 (2004): 1-8.
15. Paliwal, Swati, and Ravindra Gupta. "Denial-of-service, probing & remote to user (R2L) attack detection using genetic algorithm." International Journal of Computer Applications 60.19 (2012): 57-62.
16. Jeya, P. Gifty, M. Ravichandran, and C. S. Ravichandran. "Efficient classifier for R2L and U2R attacks." International Journal of Computer Applications 45.21 (2012): 29.
17. https://www.netresec.com/?page=MACCDC [Accessed on May 1st 2019.]
18. Shiue, Lai-Ming, and Shang-Juh Kao. "Countermeasure for detection of honeypot deployment." In International Conference on Computer and Communication Engineering, 2008: 595-599.
19. Pedram Hayati, Vidyasagar Potdar. "Spammer and Hacker, Two Old Friends", 3rd IEEE International Conference on Digital Ecosystems and Technologies, 2009: 290-294.
20. Winn, Michael, et al. "Constructing cost-effective and targetable industrial control system honeypots for production networks." *International Journal of Critical Infrastructure Protection* 10 (2015): 47-58.
21. Wagener, Gérard, Alexandre Dulaunoy, and Thomas Engel. "Heliza: talking dirty to the attackers." *Journal in computer virology* 7.3 (2011): 221-232.
22. Noah Elhardt, "Venus Flytrap showing trigger hairs" Wikipedia the free encyclopedia, wikimedia foundation, 13 April 2006. https://en.wikipedia.org/wiki/Venus_flytrap [Acessed on 15th May 2019]

Movva Sai Chaithanya has received the B.Tech degree in Computer Science and Engineering from National Institute of Technology, Goa. His research interest includes Soft Computing, Wireless Sensor Networks, Data Mining, and Neural Networks.

Suresh Nikudiya has received the B.Tech degree in Computer Science and Engineering from National Institute of Technology, Goa. His research interest includes Wireless Sensor Networks, Soft Computing, Data Mining, and Neural Networks.

Varsha S Basanaik has received the B.Tech degree in Computer Science and Engineering from National Institute of Technology, Goa. Her research interest includes Wireless Sensor Networks, Soft Computing, Data Mining.

Damodar Reddy Edla is an assistant professor in the Department of computer science and engineering department, National Institute of Technology Goa. He received M.Sc degree from the University of Hyderabad in 2006, M.Tech and Ph. D. degree in computer science and engineering from Indian School of Mines Dhanbad in 2009 and 2013 respectively. His research interest includes Wireless Sensor Networks, Cognitive Neuroscience, and Data Mining. He has published more than 70 research papers articles in reputed journal and International conferences. He is a senior member of IEEE and IACSIT. He is also Editorial Board member of several International journals.



Hanumanthu Bhukya is an Assistant Professor, Department of Computer Science and Engineering, Kakatiya Institute of Technology & Science Warangal. He has published several research articles in peer-reviewed International journals and conferences. His research focuses on data analysis, web security and information security. He is a member of ISTE.