International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 9, September - 2013

Modeling and Detection of Camouflaging Worm by Using Machine Learning Technique

T.Malyadri(M.Tech), Assistant Professor,
L.Roshini(M.Tech), Assistant Professor,
S.Lavanya Reddy(M.Tech), Associate Professor,

Abstract:

Active worms has been major security threat to the Internet. This is due to the ability of active worms to propagate in an automated fashion as they continuously compromise computers on the internet. Active worms develop gradually during their propagation and thus pose a great challenges to protect from danger against them. In this paper we study a new class of active worm, referred as a camouflaging worm. The camouflaging worms is different from the normal traditional worms because of it has the ability to intelligently manipulates its scan traffic volume over time. So it is very difficult to detect the camouflaging worm by using normal worm detection technique. We propose a new worm detection technique based on Machine Learning Technique. This approach captures dynamic program behavior to provide accurate and efficient detection against both seen and unseen worms. This particular software is developed based on the data set given by DARPA. The data set contains lakhs of transactions that include all attacks observed by the DARPA. Based on the transaction behavior we made software. Our experimental results clearly demonstrate the effectiveness of our approach to detect new worms in terms of very high detection rate and a low false positive rate. This process not only detect camouflaging worms also detects the denial-of-service (DOS) attacks , probe attacks and etc.

Keywords: Worm,Camouflaging Worm,Power Spectral Density,Spectral Flatness Measure.

I INTRODUCTION

Malwares are malicious software. The malwares are used to designed to damage the computer systems without the knowledge of the user of the system. Malware is a collective terms for any malicious software which enters system without authorization of user of the system. The term is created from merging the words “malicious” and “software”. Malware is a very big threat in today computing world. Most of malware enters into our system by downloading the files from the internet. Once malicious software finds it way into the system, it scans all the vulnerabilities of operating system and performs unintended actions of the system, finally slows down the performance of the system.

Malware has the ability to infect other executable code , data/system files, boot partitions of drivers and creates excessive traffic on network leading to denial of service when ever a user executes a infected afterwards. If the operating system has the vulnerability, malware can

also take control of system and infect other systems on the network.

Some malwares are easy to detect and remove through the antivirus software. The antivirus software maintains a repository of virus signatures i.e. binary pattern characteristics of malicious code. By using the signatures, we check the infected files for presence of any virus signatures. This method is well until the malware writer started writing polymorphic and metamorphic malware. Malware writers continuously make use of new obfuscation methods so that the malware could evade or escape detection.

Malware can be broadly classified into different types. Virus, Worm, Trojan horse, Spyware, Ad ware etc.

Virus is a piece of malware code with harmful intent and has ability to self replicate. Mode of operation is appending virus code to an executable file and when a file runs, the virus code gets executed and spread from the infected computer to other computers through network.

Trojan horse, at first glance appears as a useful software but actually do damage once installed or run on computer.

Spyware is a collective term for software which monitors and gathers personal information about users like pages frequently visited , email address, credit card numbers and key pressed by the user.

Adware or advertising supported software automatically plays, displays or downloads advertisements' to a computer after malicious software installed. This piece of code is generally embedded into free software.

Worm is a program that can run independently and can propagate a full working version of it to other machines. Worms are self replicating programs which uses network to send copies of itself to other systems invisibly without user authorization. Worms may cause harm to network by consuming the bandwidth. Unlike the virus they do not need the support of the executable file.

active worm refers to a malicious software program that broadcast itself on the Internet to infect other computers. The broadcast of the worm is based on utilize vulnerabilities of computers on the Internet. Many real-world worms have origin notable damage on the Internet. These worms contain "Code-Red" worm in 2001, "Slammer" worm in 2003 , and "Witty"/ "Sasser" worms in 2004 . Many energetic worms are used to infect a large number of computers and recruit them as bots or zombies, which are networked mutually to form botnets. These botnets can be used to:

1. Launch massive Distributed Denial-of-Service (DDoS) attacks that interrupt the Internet utilities
2. Access confidential in sequence that can be misused through large-scale traffic sniffing, key logging, identity theft, etc.,
3. Destroy data that has a high fiscal value
4. Distribute large-scale unsolicited advertisement emails (as spam) or software (as malware).

There is evidence showing that contaminated computers are being rented out as "Botnets" for creating an entire black-market industry for renting, trading, and managing "owned" computers, leading to economic enticements for attackers. Researchers also showed possibility of "super botnets," networks of independent botnets that can be synchronized for attacks of unprecedented scale. For an adversary, super botnets would also be tremendously versatile and resistant to countermeasures. Due to the considerable damage caused by worms in the past years, there have been momentous efforts on developing detection and defense mechanisms beside worms. A network- based worm detection system plays a foremost role by monitoring, collecting, and analyzing the scan traffic (messages to identify vulnerable computers) produce during worm attacks.

The influence of the network characteristics on the virus extend is analyzed in a new the intertwined Marko chain model, whose only estimate lies in the application of mean field theory. The mean field approximation is quantized in detail. The intertwined replica has been compared with the exact 2-state Markov replica and with previously projected "homogeneous" or "local" models. The sharp epidemic threshold, which is a outcome of mean field theory, is rigorously shown to be equal to1, where max is the largest eigen value the spectral radius—of the adjacency matrix. A sustained fraction expansion of the steady-state infection prospect at node is presented as well as several upper bound The model belongs to the class of susceptible infected susceptible (SIS) models that, mutually with the susceptible infected removed (SIR) models, are the standard models for computer virus infections. Each node in the network is moreover infected or healthy. An infected node can infect its neighbors with an contagion rate, but it is cured with curing rate .However, once cured and healthy, the node is again prone to the virus. Both infection and curing processes are independent. Refinements like the existence of an incubation period, an infection rate that depends on the number of neighbours, a

remedial process. That takes a certain amount of time, and other sophistications are not considered.

The theory of the spreads of epidemics during a network can be applied to the spread of email worms and other computer viruses, the broadcast of faults or failures, and, more generally, the spread of information (e.g., news, rumours, brand awareness, and marketing of new products) and epidemic dissemination or/and routing in ad hoc and peer-to-peer networks.

II. RELATED WORK

A. In the circumstance of the Internet, net-work fragmentation is well known and occurs in several situations, including an increasing preponderance of network address conversion, firewalls, and virtual private networks. Recently, however, new threats to Internet consistency have received media concentration. The issues fall into two categories: Conflict regarding naming and the use of geo location to restrict access to resources. First, a number of nations have raised formal opposition to the oversight of ICANN by the United States, and a number of private organizations such as Unified Root have materialized to offer alternative name spaces. Global agreement on Internet governance is becoming increasingly complicated, which means the potential for inconsistency in naming resulting from various Domain Name Service (DNS) roots or addresses that are not globally unique will only increase. To a significant extent, the Internet depends upon everyone having contact to the same set of names. The threats, therefore, are a) the same name does not subsist in both of two locations (lack of global consistency), and b) the same name refers to dissimilar resources in different locations (lack of global uniqueness).Second, a professed increase in online criminal activity has created viable business models for businesses that provide geo position services marketed for their ion about how a user is connected to the Internet (such as IP address and Internet service provider (ISP) data) to conclude whether the user is likely to be fraudulent. This has caused a number of legitimate online dealings to be denied when users are not connected at their usual point of accessory. Finally, various governments and service providers around the world have organized network technology that (accidentally or intentionally) restricts contact to certain Internet content.

B. These services are sometimes referred to as conference oriented services because they manage on traffic flowing between pairs of source and purpose nodes. In the case of a stub autonomous system (AS), some of these source and target nodes are likely to be edge routers connected to the hosts, whereas in the case of a large transit AS, these nodes are two border routers in the ingress and egress of the AS. The serviced traffic negotiates the shortest path from the source to the overhaul gateway and then the shortest path from the gateway to the destination. Traditionally, such overhaul gateways have been placed on the boundary of an AS since all inter domain traffic passes. However, there is a emergent trend to place network services inside the AS. It was first shown by that FTP traffic can be significantly condensed by placing caches in strategic locations inside the AS backbone. Since then, there has been a large quantity of work that demonstrates the benefits of well considered placement strategies in a variety of service contexts. Such strategies take into description the distribution of traffic as well as the topology of the AS. Research on service placement has determined mainly on placing the service gateways in a way that minimizes the average length of the traversed routes. Therefore, each flow always selects the service gateway that imposes the shortest probable route. However, this loom does not take into account the reciprocal effect of individual flows, the load obligatory on the network links, and the possible survival of hotspots (congested areas) in the network.

C. In this paper we here a scalable routing protocol for ad hoc networks. The protocol is based on geographic location organization strategy that keeps the overhead of routing packets relatively small. Nodes are allocating home regions and all nodes within a home region know the estimated location of the registered nodes. As nodes travel, they send location renews messages to their home regions and this in sequence is used to route data packets. In this paper, we obtain theoretical performance results for the protocol and confirm that the control overhead scales linearly with node speed and as N3=2 with increasing number of nodes. These results point out that our protocol is well suited to moderately large ad hoc networks where nodes travel at high speed. In this paper we here a new routing protocol and derive a theoretical bound that characterizes its scalability. Our routing protocol relies on a location update method that maintains approximate location information for all nodes in a disseminated fashion. As nodes move, this estimated location information is constantly updated. To maintain the location information in a decentralized way, we map node IDs to a geographic sub-region of the network. In this paper we obtainable a new routing protocol for large networks and developed a theoretical model for predicting its scalability with reverence to increased node speeds as well as increasing network sizes. One disadvantage of this protocol is, it does

not determining a lower bound for routing overhead.

D. Networks while focusing only on one phase of the configuration the one related to the configuration of the Border Gateway Protocol (BGP). BGP is the most frequently deployed inter-domain routing protocol. It allows networks to join to each other. measured BGP configuration errors that were noticeable from routing updates at the Oregon Route Views servers over the course of 21 days, and found that misconfigurations were pervasive. About 75% of all new advertised routes were speciously announced during that time, which was a conservative approximate according to the authors. In addition to their occurrence, network resulted in the proliferation of routes, leading to“ misdirected/lost traffic for tens of thousands of networks” .Several solutions have been projected to deal with the router misconfiguration problem. All but one of them compare configurations with a list of constraints or common best practices that a network ought to follow to function correctly. This loom makes the assumption that rules violations are misconfigurations , and is very effective in detecting certain types of clear-cut problems, such as checking that internal BGP speakers form a full mesh, verifying that all IP addresses within a network are distinctive, and determining whether referenced routing policies are really defined. However, the identification and definition of the restriction scan be a challenging task. What constitutes an error sometimes depends on the network what is an error for one network can be common follow for another. This relativism of error definition is echoed by others. Studied configuration files from networks.

III ALGORITHM EXPLINATION

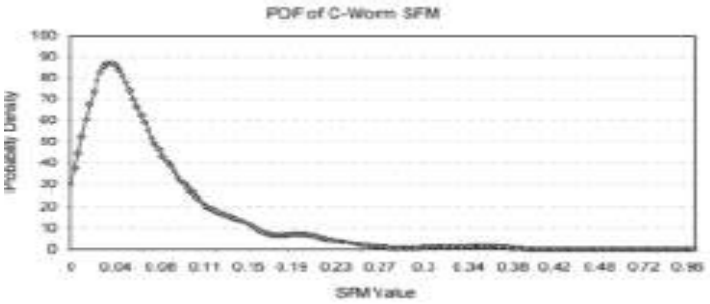
Spectrum Based Analysis
1.Power Spectral Density
2.Spectral Flatness Measure

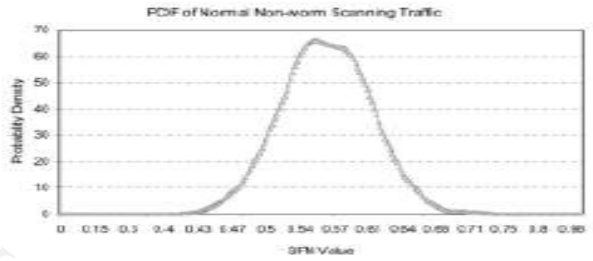
Power Spectral Density:

Transfer data from time province Scan traffic determined the discrete fourier transform. Compare c-worm traffic and standard worm traffic Transform data from time domain into occurrence domain Random process x(t),t€[0,n], X(t)- source count in time period PSD SCAN traffic information is determined using Discrete fourier Transform Compare non worm traffic and c-worm traffic.

Spectral flatness measure:

Distinguish the scan traffic of c-worm and customary non worm traffic Spectral flatness determines concentration of information at narrow frequency range Higher concentration at small spectrum comparatively.





Compare c-worm traffic and non worm traffic.

IV PROPOSED SYSTEM ARCHITECTURE

In the existing system we can find the camouflaging worms by using spectrum based analysis. By using this analysis it is not possible to find all the camouflaging worms and one more thing is it takes lot of time to find these worms.

Now we proposed one technique to effectively identify the camouflaging worms and all other network attacks with in less time name is Machine Learning Technique. Machine Learning focuses on the predictions based on known properties ,learned from training data. Now –a-days most of intrusion detection system are using the Machine Learning Technique because of its ability to adopt the changing conditions and infer patterns from data. In the proposed system we take both training and testing data from Defense Advance Research Project Agency (DARPA). The proposed system work based on Support Vector Mechanism(SVM) can be defined as which use hypothesis space of linear function in a high dimensional feature space , trained with a learning algorithm.

Generally all the network attacks is divided into different categories like probe attack, DOS attack ,local to

remote attack and remote to local attack. so the camouflaging worms identify in the remote to local attacks.

Hierarchical Approach for Camouflaging Worm:

Going into the details of the Layer-based Camouflaging worm System. Hierarchical approach which depends availability, confidentiality, and integrity of data and (or) services over a network. Reducing computation and overall time required in detecting anomalous event is achieved by using hierarchical model. Reduction in time to detect an intrusive event is possible by reducing the communication overhead among different hierarchical levels and this can be done by making the hierarchical levels autonomous and self-sufficient in blocking an worm without the aid of a central decision-maker. All the levels in the LIDS framework are trained separately and then sequentially deployed. After defining the four hierarchical levels, Probe level, Denial of Service level, REMOTE-TO-LOCAL level, and USER-TO-ROOT level corresponding to the four worm groups mentioned in the data set and then each level is separately trained with a small set of relevant features. Feature selection is very important for Hierarchical approach we examine it in the next section. For making the hierarchical levels independent, some of the features may be present in more than one of the levels. These hierarchical levels are vital as such they act as filters that block any anomalous connection, thus preventing the need of further processing at subsequent hierarchical levels making quick response to intrusion. The net result of that sequence of hierarchical levels is that the anomalous events are recognized and blocked at the very moment they are detected.

Improving the speed of operation of the system is the next goal. For that, the LIDS is put into effect and a small set of features are selected for every level rather than using all the 41 features. The result is that there is significant performance improvement of both the training and the testing of the system. In several situations, there is a trade-off between efficiency and accuracy of the system and various avenues are there to enhance system performance. Methods like naive Bayes presume independence among the observed data and this surely increases system efficiency, but it may adversely affect the accuracy. In order to balance this trade-off, we use the SVM's that are more precise, though expensive, but we implement the Hierarchical approach to enhance overall system performance. This performance of the proposed system, Hierarchical level SVM learning, is on par with that of the decision trees and the naive Bayes, and this system has higher warm detection accuracy.

Experiments:

Standard KDD '99 intrusion data set is used in this experiments.1998 DARPA Intrusion Detection and Camouflaging worm evaluation program, which is prepared and managed by the MIT Lincoln Laboratory, is the original version of this dataset. Five million connection records as the training data and about two million connection records as the test data are contained in this data set. 10 percent of the total training data and 10 percent of the test data (with corrected labels), which are provided separately are used in these experiments and this leads to about 494,020 training and 311,029 test instances. T the data set in each of these records indicates a connection between two IP addresses, one starting and the other ending at some well defined times along with a well-defined protocol. Also, every record is typified by 41 different features. A separate connection is represented by each record. Therefore each and every record is distinct.

We provide the Precision, Recall, and F-Value and not the accuracy alone because it is easy to attain very high accuracy by conscientiously choosing the size of the sample. As per the dataset opted, it is identified that the number of instances for camouflaging worms in USER-TO-ROOT, Probes, and REMOTE-TO-LOCAL layers is very low. So, the system can be prejudiced and can attain a precision of more than 99 percent for camouflaging worms in USER-TO-ROOT layer , if at all accuracy is utilized to evaluate and test the perform the execution of the system. Whatsoever, Precision, Recall, and F-Value are independent of the size of the training and the test samples. They are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$F\text{-Value} = \frac{(1 + \beta^2) * \text{Recall} * \text{Precision}}{\beta^2 * (\text{Recall} + \text{Precision})},$$

Here TP, FP, and FN indicate the number of True Positives, False Positives, and False Negatives, respectively, and relates to the relative significance of precision versus recall and is generally set to 1.

Normal, Probe, Denial-Of-Service, REMOTE-TO-LOCAL, and USER-TO-ROOT are the various layers into which we divide the trained data and also we divided the test data alike. 10 experiments for each worm class are done after haphazardly choosing data corresponding to that

warm class and normal data only. It is explained here with an example. In order to check the camouflaging worms in Probe layer, training and testing of the system is done using only Probe layer and normal data. Even the Denial-Of-Service, REMOTE-TO-LOCAL, and USER-TO-ROOT layers are not used data in this process. Excluding them provides system to better acquire the knowledge of features for camouflaging worms in Probe layer and normal events. Such a system when it is used camouflaging worm can either be seen as normal or as Probes. Camouflaging worms in Denial-Of-Service layer, if at all they are tracked as normal, they are anticipated to be detected as warm at other hierarchical levels in the system. On the other hand if the camouflaging worms in Denial-Of-Service layer are tracked at Probe layer, it is perceived as a gain as the warm is tracked at an early stage. Going by the same fashion, the camouflaging worm at probe layer that are not tracked at that layer may be detected at further hierarchical levels. This is the reason for having independent models for all the 4 classes, which are then trained separately with different features to find camouflaging worm, which confine to a specific group. The best, the average, and the worst cases are reported here.

Detecting camouflaging worms in Denial-Of-Service layer with All 41 Features:

Choosing haphazardly 20,000 normal records and around 4,000 records with camouflaging worms at DOS layer from the training data and then we have utilized every normal and with camouflaging worms at DOS layer from the test data for testing. Thus 24,000 training instances and 290,446 test instances are listed. From table 4,290,446 test instances were labeled in a testing time span of 64.42 seconds. From the observations it is observed decision trees have a slight edge when test time efficiency is considered, in contrast to the observations that all the 3 methods employed possess analogous warm detection accuracy.

Here the feature selection is performed using the data that is utilized in the previous experiment. Table 5 illustrates the outcomes. It is noticed that 290,446 test instances were labeled in 15.17 seconds. Only a moderate betterment was observed when compared with the previous experiment. It is found that hierarchical decision trees are a better option when testing time is considered. It can also be noticed that there is a tiny increase in the detection precision when feature selection is done, but this increase is imperceptible. The perceptible gain is seen in the reduced time for testing, which subsides by about four folds.

Detecting camouflaging worms in REMOTE-TO-LOCAL Layer with All 41 Features:

In this experiment 1,000 normal records and all the records with camouflaging worm at REMOTE-TO-LOCAL layer from the training data as the training data for detecting camouflaging worm at REMOTE-TO-LOCAL layer are chosen. Then all the normal and records with camouflaging warm at REMOTE-TO-LOCAL layer from the test data are utilized for testing purpose. In all, 2,000 training instances and 76,942 test instances are used.

From the experiments, it can be noticed that 76,942 test instances were labeled in 17.16 seconds. Observations also reveal that though the decision trees have a higher F-Value, considering the count of the number of false alarms, it can be inferred that SVM execute better and have high Precision than that of the decision trees and the naive Bayes.

In this particular experiment feature selection is done in order to detect Detecting camouflaging worms in REMOTE-TO-LOCAL layer. Considering the observations, it can be noticed that 76,942 instances are tested in a time span of 5.96 seconds. It can also be noticed that, the Hierarchical SVM rendered better performance than the SVM (increase is approximately 60 percent), hierarchical decision trees (increase is approximately 125 percent), decision trees (increase is approximately 17 percent), hierarchical naive Bayes (increase is approximately 250 percent), and naive Bayes (increase is approximately 250 percent) and therefore they are the pick of the choices for detecting the Camouflaging worms in REMOTE-TO-LOCAL layer. Though the Hierarchical SVM takes a little more time, it doesn't matter as higher detection accuracy is obtained.

Detecting camouflaging warms in USER-TO-ROOT Layer with All 41 Features:

In this experiment 1,000 normal records which are chosen haphazardly and every USER-TO-ROOT record from the training data is taken as the training data to detect the User to Root warms. Along with them all the normal and records with camouflaging worm at USER-TO-ROOT layers from the test data are also utilized for testing purpose. In all there are 1,000 training instances and 60,661 test instances. From the experiments, it is noticed that 60,661 test instances are labeled in a time span of 13.45 seconds. It is observed that the SVM have advantage over the other two methods. The F-Value of SVM 150 percent more than that of decision trees and in the case of naive Bayes it is more than 600%. The advantage of SVM is that

they can be utilized to reliably detect The camouflaging worms in USER-TO-ROOT layer, which in general, are very perplexing to detect and most of the contemporary Intrusion Detection and Camouflaging worm systems fail to detect these with acceptable reliability. .

In the present experiment, the instances that were used in the earlier experiment were only used. The difference is that feature selection is executed here.

From the experiments, it can be noticed that 60,661 test instances were labeled in 2.67 seconds. The inference is that Hierarchical SVM is the most valuable choice when it comes to the detecting Camouflaging worm in USER-TO-ROOT layer and are far better than SVM (an increase of about 8% percent), hierarchical decision trees (approximately 30% increase), decision trees (approximately 184% increase), hierarchical naive Bayes (approximately 38% increase), and naive Bayes (approximately 675% increase).Further it can also be observed that the warm detection capability also increases for both the decision trees and the naive Bayes. It is also clear from the results that the correctness of Hierarchical SVM is notably higher for the Camouflaging worm in USER-TO-ROOT, REMOTE-TO-LOCAL, and the Probe Layer, but he difference in accuracy is, however, inconspicuous for the Denial-Of-Service warms. It is also pellucid that irrespective of the method employed and especially for the SVM, by executing feature selection, the time required for training and testing the system is reduced to a great extent. Further it can be noticed that increase in detection accuracy is not perceptible in the case of the hierarchical decision trees and the hierarchical naive Bayes for the DOS group of worm. Their accuracy of detection also decreases in the case of the Probe and Camouflaging worm in REMOTE-TO-LOCAL layer while it increases for the Camouflaging worm in USER-TO-ROOT layer. Whatsoever, it was found that in all the cases, when a small set of specific features for training are utilized, the Hierarchical SVM has executed strikingly better than any other one**.**

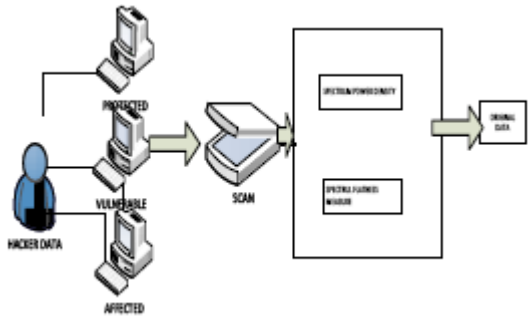
Implementing the System in Real Life:

In general, the category of an worm is not known to us. We are engaged in knowing the warm category only when the system detects an event as anomalous. As every layer in a Hierarchical Approach is trained to detect only a specific category of warm, this approach helps to enhance the warm and also to recognize the type of the warm. So, if an warm is recognized at the Camouflaging worm in USER-TO-ROOT layer, it is very plausible that the warm is of "USER-TO-ROOT" type, thus facilitating to execute quick recovery and take precautions to prevent such warms. In this experiment, we combine the4 models (along with feature selection) in order to evolve the final system and the data utilized is the same that was used for training the individual models the earlier experiments, except for the fact the data in the test set is relabeled either as normal or as warm and all the data from the test set is passed though the system starting from the first layer. A connection which is identified as an warm by the layer 1 is blocked and labeled as "Probe." The events which are labeled as "Normal" are only allowed to go to the next layer. Similar process is redone at the next hierarchical levels where an worm is blocked and labeled as "DODS," "REMOTE-TO-LOCAL," or "USER-TO-ROOT" at level 2, level 3, and level 4, respectively .All the experiments are carried out 10 times and their average is disclosed. ason that in the real environment it is crucial to detect a worm very early in order to its impact. It is also significant to note that most of the Camouflaging worm in "USER-TO-ROOT" layer are traced in the third level itself and hence labeled as "REMOTE-TO-LOCAL." However, in the case if the third level is removed, the fourth level can still recognize these worms with analogues precision.

For confining the worm traffic to the starting hierarchical levels in the system, Hierarchical approach is very powerful .Experiments are executed even without implementing the Hierarchical Approach. In this case only a single system is considered which is trained with two classes (normal and warm), all the Probes, DOS, Camouflaging worm in REMOTE-TO-LOCAL, and USER-TO-ROOT layer are labeled as "layer." Experiments both with and without feature selection are also done. For the process of feature selection, 21 features are taken into consideration, which are selected by applying the union operation on the feature sets of all the four warm types. These results are then compared with the Hierarchical Approach as shown in Table 13.It can be noticed that a system that implements a Hierarchical Approach with feature selection is more efficient and more accurate in detecting warms, especially with respect to the Camouflaging worm in USER-TO-ROOT, the REMOTE-TO-LOCAL, and the Probes.

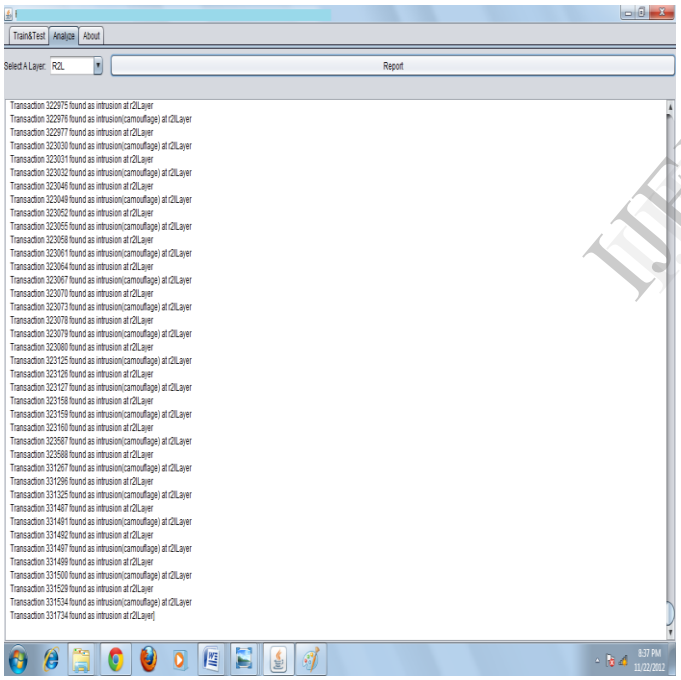
It is very important to understand that the time should be considered in relative terms rather than absolute terms this is for the comfortable utilization of the scripts. However, in real environment, high speed can be achieved by executing the entire system in languages with proficient compilers. An example is the "C Language." In addition, pipelining can be put into effect in multicourse processors,

where each core may represent a single level, and because of pipelining; multiple I/O operations can be replaced by a single I/O operation providing very high speed of operation.

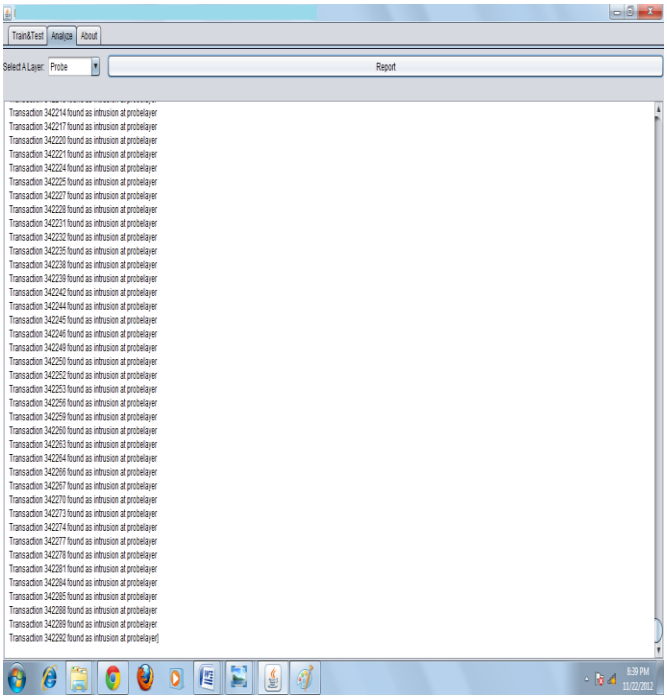


RESULTS:

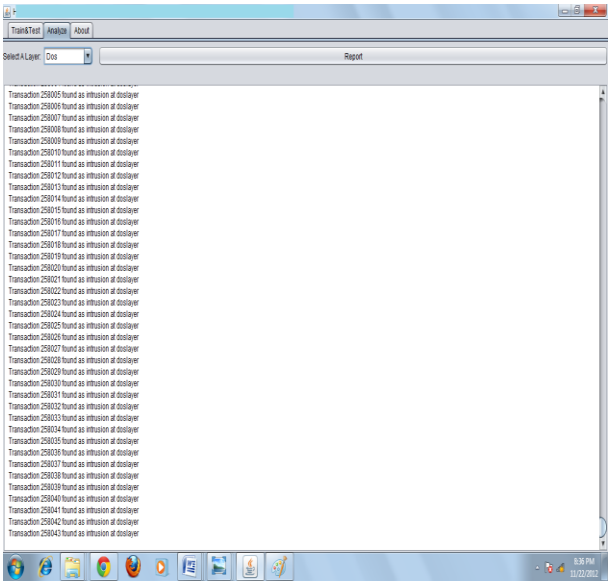
The experimental results are shown as follows. It shows the different ypes of attacks like camouflaging worms ,probe attacks and DOS attacks…….



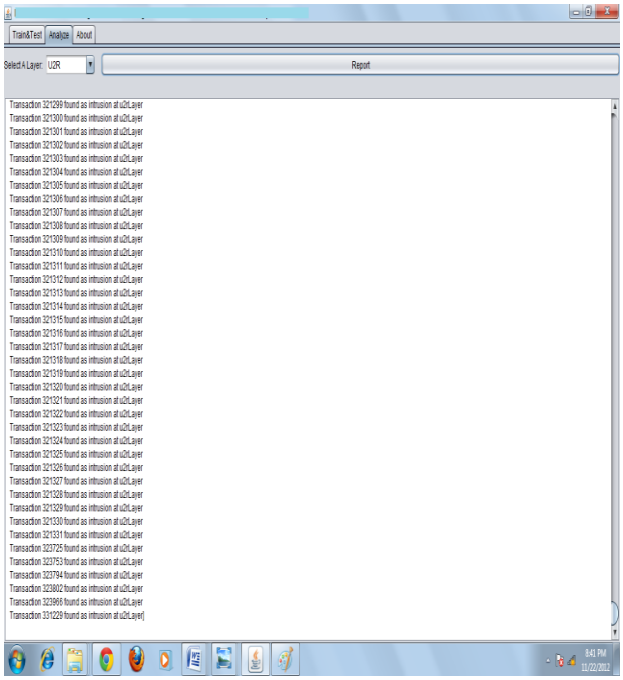
Camouflging worms



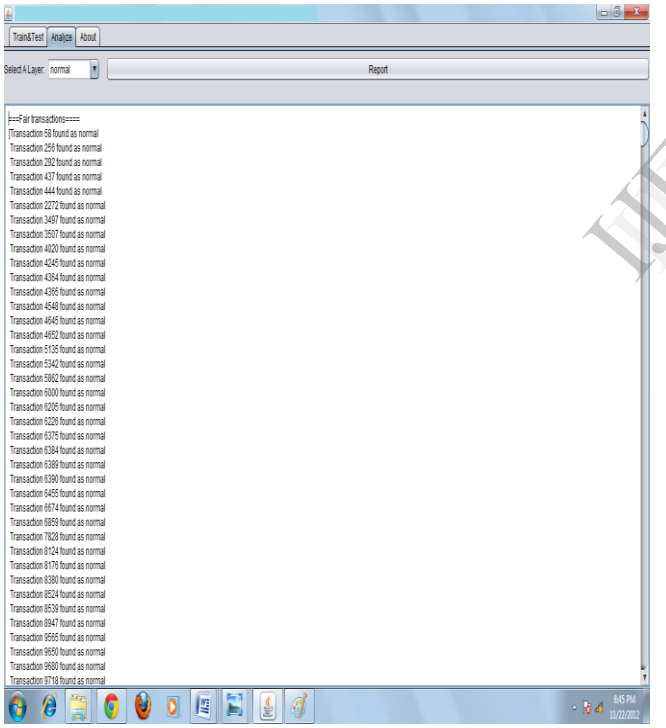
Probe attacks



Dos attacks



R2L attacks



Normal transaction

By using our proposed system not only identify the camouflaging worms we can identify all other attacks like probe, Dos and local to remote attacks .The proposed system saves time and cost also because no need to purchase and install extra hardware and software and it is effective one.

V CONCLUSION

C-Worm could use based on the restricted network and computing resources available during its propagation. Incorporating the Peer-to-Peer techniques to broadcast information through secured channels Actually a worm could take benefit of the knowledge that an infection endeavor was a new hit reaching a previously uninfected vulnerable computer and replica hit reaching a previously infected vulnerable computer. The loom used by the "self-stopping" worms that do not require a global overlay control network for Realizing their behavior in practice. We call our approach to approximation the Distributed Co-ordination method. In this method, there is no centralized coordination among the C-Worm instances to obtain feedback in sequence about the value. By scanning vulnerable computers and obtaining the water-marks in sequence during the scanning.

VI REFERENCES

[1] W. Gong, and D. Towsley C.C. Zou, , Nov. 2010 "Code-Red Worm Propagation Modeling and Analysis," Proc. Ninth ACM Conf. Computer and Comm. Security (CCS).

[2] Cert, cert/cc advisories, may/june 2010 388 iesee transactions on dependable and secure computing, vol. 8, no. 3

[3] S. Coull, and F. Monrose C. Wright,), Feb. 2009 , "Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis," Proc. 15th IEEE Network and Distributed System Security Symp. (NDSS).

[4] J. Brown D. Moore, C. Shannon, and, "Code- Red), Nov. 2010: A Case Study on the Spread and Victims of an Internet Worm," Proc. Second Internet Measurement Workshop (IMW).

[5] W.B. Gong, and L.X. Gao Towsley, C. Zou, D. Oct. 2010, "Monitoring and Early Detection for Internet Worms," Proc. 10th ACM Conf. Computer and Comm. Security (CCS),.

[6] D. Moore, V. Paxson, and S. Savage, , July 2010 "Inside the Slammer Worm," Proc. IEEE Magazine of Security and Privacy.

[7] J. Ma, and S. Savage, , G.M. Voelker Nov. 2010 "Self-Stopping Worms," Proc. ACM Workshop Rapid Malcode (WORM),

[8] M. Garetto, W.B. Gong, and D. Towsley, , Mar. 2010 "Modeling Malware Spreading Dynamics," Proc. IEEE INFOCOM

[9] , J. Caballero, M.G. Kang and D. Song,), July 8, 2011 "Distributed Evasive Scan Techniques and

Countermeasures," Proc. Int'l Conf. Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA.

[10] P.R. Roberts, Zotob.asp, 2010 Arrest Breaks Credit Card Fraud Ring, http:// www.eweek.com/article2/0,1895,1854162,00.

[11] R. Naraine, asp, 2010 Botnet Hunters Search for Command and Control Servers, http://www.eweek.com/article2/0,1759,1829347,00..

[12] R. Vogt, J. Aycock, and M. Jacobson, , Oct. 2009 "Quorum Sensing and Self-Stopping Worms," Proc. Fifth ACM Workshop Recurring Malcode (WORM).

[13] , V. Paxson S. Staniford, and N. Weaver, Aug 2009 "How to Own the Internet in Your Spare Time," Proc. 11th USENIX Security Symp. (SECURITY),