Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/59

A NEURAL NETWORK CLASSIFIER FOR PAYLOAD-BASED ONLINE WORMS DETECTION

Bhasker Pant[1], Dibyahash Bordoloi[2], Durgaprasad Gangodkar[3]

[1]Department of Computer Science & Engineering,Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002
[2]Head of the Department, Department of Computer Science & Engineering,Graphic Era Hill University, Dehradun, Uttarakhand India, 248002
[3]Department of Computer Science & Engineering,Graphic Era Deemed to be University, Dehradun, Uttarakhand India, 248002

ABSTRACT

Worms pose a significant risk to the Internet because they may infect hundreds of thousands of hosts simultaneously. Internet worm detection is a major research frontier. Worms disseminate themselves throughout the Internet and represent a significant risk to data integrity. Signature-based anti-worm methods are ineffective against zero-day exploits. This study is moving its emphasis from detecting Internet worms by their characteristic patterns to recognising their malevolent activity. Using a data mining approach, such as a neural network classifier, this research provides the innovative notion of collecting flow level characteristics that may distinguish worms from clean systems. The detection rate of Internet worms for which no data was utilised in the model building process was 97.90%, demonstrating the effectiveness of our method.

Keywords: Worm detection on the internet, flow characteristics, neural network, and novel detection

INTRODUCTION

The Internet has become a battleground for attackers and defenders as computer and communication networks grow ubiquitous. The Internet worm is a devastating tool for cybercriminals [1]. More specifically, a worm is a malicious computer programme that spreads itself by infecting other computers and then using those infected computers as a breeding ground for more worms. This has led to hundreds of thousands of hosts being infected by worms like Code Red, Slammer, and Witty, which poses a serious challenge to network administrators. Worms' creators are becoming more and more creative with their attacks, which creates new difficulties for those tasked with keeping them at bay. Attacks from viruses and worms are only two of the numerous varieties that constantly menace the Internet. A worm is a self-replicating piece of software that spreads from infected host to host on a network by exploiting a security hole in those hosts. A virus, on the other hand, is a little bit of code that spreads only when a user opens a certain file. A significant problem in networking is the computational complexity of early detection of emerging worms and viruses during their first phases of spread [2]. In the last two decades, scientists have created hundreds of new worms. Some of these worms have severely impacted international computer systems. Morris, Code Red and Code Red II, Nimda, Slapper, Sapphire/Slammer, and more recently So- Big.F, MSBlast, and Mydoom are among the most infamous worms in history. Ever since the first worm, the Morris worm, was published in 1988, detecting worms on the Internet has been a major field of study. One must be familiar with the numerous worms, payloads, and attackers in order to fully grasp the worm threat. If a programme on one computer can trigger its own (potentially modified) copy to run on another computer, we call that programme a "worm" in the network [3]. Typically, worms spread independently across networks by exploiting holes in the security or policy of frequently used network services. One way in which worms and viruses differ is that the latter attach themselves to files and wait for users to open them before spreading. This is why viral reproduction lags behind that of worms. However, worms may rapidly disperse and cause widespread damage. In less than 14 hours, more than 359,000 machines were infected with the Code Red I version 1 internet worm in 2001 [4]. Within 10 minutes of the more aggressive Slammer internet worm assault in 2003 [5], more than 90% of 75,000 susceptible hosts were infected. Internet systems may be infected by a worm if it were well-designed.

with increased haste [6]. Worms are a serious problem that may severely impact the availability of a network. Many researchers are interested in finding ways to automate defences against these threats.

Here is how the remainder of the paper is structured. The second section elaborates on how computer worms operate. In Section 3, we'll talk about the different worm detection methods, what properties of worms they use, and where they fall short. In Section 4, we will briefly recap the current state of worm detection.

CONNECTED TEXTS

Viruses and worms spread around the Internet by means of malicious communication. Faster and more reliable transmission is achieved by traffic monitoring and detection of harmful activity. Monitoring traffic flows inside a network may help identify and take advantage of unlawful traffic generated by internet worms rather than doing a manual payload inspection of each packet. Here are some of the methods that have been presented for identifying these worms in cyberspace:

In a recent paper, Chen offered a new method, arguing that it will reduce future internet epidemics via the use of the efficient Divide-and-conquer-scanning worm. When compared to the slow and obvious random-scanning worm, this method is far more efficient and covert. The author also discussed two types of defensive strategies in this paper: eliminating sick hosts and using active honey nets. When it comes to finding this particular worm in networks, Kong presented a new approach. Before anything else, an automated Semantics Aware statistical method creates the signatures. This is used with a hidden Markov model to build worm signatures automatically by removing the non-critical bytes. Another data mining strategy, [7], used three distinct feature classes in addition to many classifiers to identify dangerous software. They used a main dataset that included 3,265 harmful applications and 1,001 clean ones. A rule-based framework called RIPPER was used on the DLL dataset. A Naive Bayes classifier was fit to string data, and an n-gram-based Multi-Naive Bayes classifier with a voting approach was trained. There was no mention of an n-gram decreasing method. Instead, six Naive-Bayes classifiers were trained on separate subsets of the data obtained by splitting the original dataset. Since each classifier was constructed using a unique set of characteristics, a direct comparison between them is problematic. Their model performed best when based on Naive-Bayes techniques applied to strings. Examined to identify the susceptible host. Here, the author uses the gradual hybrid anti-worm strategy. Both aggressive and passive anti-worm measures were used in this strategy. When an active anti-worm is deployed, it searches for hosts that are susceptible on the network and applies updates to them. Passive anti-worm took care of the listening process by attacking the worm from inside the host system once it had been patched. The worm tree methodology was suggested to study the evolutionary history of internet worm infections. Captures the essential features of internet worm identification and applies them to bot detection using mathematical analysis.

To lessen the economic damage caused by network worms, developed a delay-based strategy. The paper derives one such value. A worm will be removed from the system if the delay in time is higher than a certain threshold value. Anti-worm system WSRMAS has been implemented. This method, which uses a multi-agent system to prevent or halt the transmission of worms in network routers, is very successful.

Various techniques for identifying infected networks by Internet worms have been presented in the aforementioned papers. According to the data, they are able to spot suspicious activity by keeping tabs on the payload and the flow of traffic. When worms are encrypted, payload detection fails to identify them. By observing traffic patterns, we can only find out about them after they've already spread. The suggested method identifies Internet worms by monitoring the traffic flow information, hence overcoming the aforementioned restrictions.

DETECTION SYSTEM

System for Detection

The suggested method employs a neural network classification algorithm to identify the malicious internet worm flow traffic payload based on features of the network flow payload. Examining TCP and UDP traffic, segmenting it into time periods, and extracting an associated vector allows for the classification of Internet worms. Attribute vectors are used to distinguish between malicious and benign communications. The term "network traffic" is used to describe the total quantity of data being sent over a network at any particular moment. Network packets, which contain bits of data, are primarily responsible for the network's workload. Measurement, control, and simulation of network traffic all revolve on the flow of data over a network. Quality of service may be guaranteed in a network if traffic is managed properly. This flow of information between nodes in a network is referred to as network traffic.

Flows indicate the total number of bytes and packets transferred across a network, providing holistic picture of network activity. So, flows significantly lessen the mountain of data that has to be examined. A group of Internet Protocol (IP) packets that go over a network at the same time is called a flow. There is a shared set of characteristics among all packets in a flow. The following variables may be used to characterise a flow: (Source IP Address, Destination IP Address, Source Port, Destination Port, Protocol)

Obtaining Features

For our experimental analysis, we use the CAIDA dataset and the KDD CUP99 data set from MIT, both of which are well-known in the field of off-line worm intrusion detection. The whole dataset, a 10% subset, plus a test dataset with verified proper labels (called correct.gz) make up this data set. In particular, we conduct tests using subsamples of the KDD CPU99 data set, sampling 1% and 2%, which yields 49402 and 98804 samples, respectively. A denial of service assault is abbreviated as DoS. As a result of the attacks, the computer is unable to process genuine requests and may even deny access to legitimate users. Having unauthorised access to the local super user002E is represented by the U2R symbol (2). The attackers leverage a security flaw to get root access through a user with less privileges, then log in as root to do illicit actions. Three, R2L stands for "remote user attack." The hackers get access to the computer remotely, log in using the user name and password, and then do malicious acts. Scanning for open ports and potential security flaws is represented by number four (Probe). By scanning for open ports and scanning for vulnerable machines, attackers may gather a wealth of information and launch a comprehensive assault.
Each observation in the dataset is represented as a vector, which contains information on simulated intrusions into a network.

Normalization

Higher valued input variables may have a tendency to overshadow lower valued ones during training of the neural network. In addition, the simulated neurons may approach the saturated conditions if the raw data is applied directly to the network. Saturation of the neurons means that changes in the input value have little effect on the output value, if any effect at all. Training of the network is profoundly impacted by this. The input data is normalised before being given to the neural network, reducing the impact of magnitude differences across inputs and preventing saturation of the neuron activation function. Min-max normalisation is one approach of adjusting the range of a feature x. Maintaining the original connection between values, min-max normalisation uses a linear transformation to produce a new set of values.

Selecting Features

Worm identification relies heavily on the accurate selection and ranking of features. The term "feature selection" refers to the procedure of determining the quality of a set of candidate features by calculating their scores and then identifying the best 'k' features. A scoring function is obtained by independently counting the occurrences of a feature in training samples from the positive and negative classes. For worm identification, it is necessary to keep an eye on a wide variety of traits, some of which may be fruitful while others may be worthless. Improved performance may be attained by reducing computing time and increasing accuracy by eliminating unnecessary characteristics. In this study, we use a feature-selection criteria based on the chi-square statistic.

To accomplish feature selection through discretization, a Chi-Square strategy is a straightforward and generic procedure that may choose an appropriate 2 value, statistic to discretize numerical features iteratively until any inconsistencies are detected in the data. The chi-squared statistic is used to analyse characteristics with regard to the classes in the 2 method. The procedure necessitates that the intervals be chosen carefully when discretizing the range of a numerical property. A characteristic's 2 value is characterised as:

Oij = number of samples in the ith intervals of the jth class, where m = number of intervals, n = number of classes.
amount of samples taken. The number of possible outcomes in a Chi-squared test is (m – 1). (n– 1).

There is a direct correlation between the quantity of features and the processing power and time required by the classifiers.

Units (neurons) in a neural network's layers translate an input vector into a desired output. Each node receives an input, processes it using a function (typically a nonlinear one), and sends the result on to the next layer. This kind of network is known as a "feed-forward" network because each node sends its output to the nodes in the next tier without receiving any information from the layer below it. In order to train a neural network to solve a specific issue, it is necessary to apply weights to the signals that go from one unit to another. At the moment, we are at the teaching stage [8].

Problems where all relevant data may be supplied to the neural network at once are well suited to Multilayer feed forward neural networks. During the training phase, the neural network receives input from a training set and repeatedly modifies network weights and biases to create an output that matches, within some degree of accuracy, a previously known result. A fresh input is given to the network in the testing phase, and a result is produced using the network parameters that were determined in the training phase. An acceptable learning approach for training multilayer feed forward networks for vector classification is back propagation learning, which is used to train the network in this study [9]. There are six neurons in the input layer, one each for the input vector's dimensions, and two neurons in the output layer. In a feed-forward neural network, the performance function is the mean square error, hence the number of neurons in the hidden layer is chosen experimentally so as to reduce this error by a factor of 30.

RESULTS AND DISCUSSIONS

In order to develop a reliable Internet worm detection system, we do extensive tests on the NSL-KDD dataset and the CAIDA dataset, both of which include 60438 training examples, 22544 testing instances, the 15 most salient features, and a random forest classification. We have measured the effectiveness of our classifier using many metrics, including true positive rate, F-measure, and accuracy.
Number of detected Internet worms expressed as a percentage.
Accuracy= TP+TN
TP+FP+FN+TN
Number of samples that were accurately identified as worms (True Positive, TP).
Number of samples wrongly identified as worms = False Positives (FP). Number of samples for which a negative prediction was made is called the "true negative" (TN). FN = the number of outlier samples that were wrongly classified as normal.
Precision is the percentage of correctly identified worm classes in the test data.
We can see that TP FP We may evaluate how well we did in identifying each worm class by looking at the recall percentage.

Webology, Volume 18, Number 5, 2021
ISSN: 1735-188X
DOI: 10.29121/WEB/V18I5/59

One way to evaluate a test's reliability is using the F-Measure, which takes into account both the level of detail provided by the results and the overall reliability.
Measurement in terms of the F-Term = Number 2 * Accurate recall

CONCLUSION

In this research, we provided a data mining architecture for discovering these malicious programmes. The classifier relied heavily on flow level payload characteristics extracted from network flow data to complete the procedure. To eliminate the potential for bias introduced by features that only appear in one class, we relied on the flow characteristics shared by worms and clean code. Our detection rate was 97.90%, with a false-positive rate of just 0.057%

REFERENCES

1. Aldwairi, M., Mardini, W., & Alhowaide, A. (2018). Anomaly payload signature generation system based on efficient tokenization methodology. *International Journal on Communications Antenna and Propagation (IRECAP)*, *8*(5), 421-429.
2. Min, E., Long, J., Liu, Q., Cui, J., & Chen, W. (2018). TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest. *Security and Communication Networks*, *2018*.
3. Ahmad, M. A. (2019). Early detection and containment of network worm. *Science World Journal*, *14*(2), 118-124.
4. Jin, X., Cui, B., Li, D., Cheng, Z., & Yin, C. (2018). An improved payload-based anomaly detector for web applications. *Journal of Network and Computer Applications*, *106*, 111-116.
5. Swarnkar, M. (2019). Deep packet inspection applications for traffic classification and security monitoring.
6. Pratomo, B. A., Burnap, P., & Theodorakopoulos, G. (2018, June). Unsupervised approach for detecting low rate attacks on network traffic with autoencoder. In *2018 international conference on cyber security and protection of digital services (Cyber Security)* (pp. 1-8). IEEE.
7. Jing, X., Yan, Z., & Pedrycz, W. (2018). Security data collection and data analytics in the internet: A survey. *IEEE Communications Surveys & Tutorials*, *21*(1), 586-618.
8. Liu, X., Tang, Z., & Yang, B. (2019, May). Predicting network attacks with CNN by constructing images from NetFlow data. In *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS)* (pp. 61-66). IEEE.
9. Khammas, B. M., Ismail, I., & Marsono, M. N. (2019). Pre-filters in-transit malware packets detection in the network. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *17*(4), 1706-1714.