Birlesik Dünya Arastırma
BD CENTER
Innovasyon ve Yayıncılık Merkezi

International Journal of Innovative Research in Education

Volume 08, Issue 1, (2021) 27-35

Semi supervised machine learning approach for DDOS detection

Sai Ramya Akula [*]**,** Devineni Venkata Ramana & Dr. Hima Sekhar MIC College of Technology, Andhra Pradesh 521180, India

Suggested Citation:

Akula, S. R. (2021). Semi supervised machine learning approach for DDOS detection. *International Journal of Innovative Research in Education, 8*(1), 27–35. https://doi.org/10.18844/ijire.v8i1.6445

Received from March 25, 2021; revised from May 20, 2021; accepted from June 20, 2021.
Selection and peer review under responsibility of Assoc. Prof. Dr. Zehra Ozcinar Teacher Training Academy.

Abstract

The appearance of malicious apps is a serious threat to the Android platform. In this paper, we propose an effective and automatic malware detection method using the text semantics of network traffic. In particular, we consider each HTTP flow generated by mobile apps as a text document, which can be processed by natural language processing (NLP) to extract text-level features. Later, the use of network traffic is used to create a useful malware detection model. We examine the traffic flow header using the N-gram method from the NLP. Then, we propose an automatic feature selection algorithm based on the Chi-square test to identify meaningful features. It is used to determine whether there is a significant association between the two variables. We propose a novel solution to perform malware detection using NLP methods by treating mobile traffic as documents. We apply an automatic feature selection algorithm based on the N-gram sequence to obtain meaningful features from the semantics of traffic flows. Our methods reveal some malware that can prevent the detection of antiviral scanners. In addition, we design a detection system to drive traffic to your own-institutional enterprise network, home network, and 3G/4G mobile network. Integrating the system connected to the computer to find suspicious network behaviors.

Keywords: Semi supervised, machine, learning approach, detection, android platform.

[*] ***ADDRESS FOR CORRESPONDENCE: Sai Ramya Akula, Devineni Venkata Ramana & Dr. Hima Sekhar MIC College of Technology, Andhra Pradesh 521180, India.

E-mail address: ramyaakula2602@gmail.com

1. Introduction

1.1. What is machine learning?

Machine learning is an application of artificial intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly.

1.2. What is distributed denial-of-service (DDOS) attack

In computing, a denial-of-service attack is a cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily distributing services of a host connected to the Internet. A DDoS attack occurs when multiple systems flood the bandwidth or resources of a targeted system, usually one or more web servers. Such an attack is often the result of multiple compromised systems (for example, a botnet) flooding the targeted system with traffic. A botnet is a network of zombie computers programmed to receive commands without the owners' knowledge. When a server is overloaded with connections, new connections can no longer be accepted.

The major advantages to an attacker of using a DDOS attack are that multiple machines can generate more attack traffic than one machine, multiple attack machines are harder to turn off than one attack machine, and the behavior of each attack machine can be stealthier, making it harder to track and shut down. These attacker advantages cause challenges for defense mechanisms. For example, merely purchasing more incoming bandwidth than the current volume of the attack might not help, because the attacker might be able to simply add more attack machines. This, after all, will end up completely crashing a website for periods of time.

1.3. Types of DDOS attack

Broadly speaking, DoS and DDoS attacks can be divided into three types:

1.3.1. Volume based attacks

Includes User Datagram Protocol (UDP) floods, ICMP floods, and other spoofed-packet floods. The attack's goal is to saturate the bandwidth of the attacked site, and magnitude is measured in bits per second.

1.3.1.1. UDP floods: A UDP flood, by definition, is any DDoS attack that floods a target with UDP packets. The goal of the attack is to flood random ports on a remote host. This causes the host to repeatedly check for the application listening at that port, and (when no application is found) reply with an ICMP 'Destination Unreachable' packet. This process saps host resources, which can ultimately lead to inaccessibility.

1.3.1.2. TCP-based attack: TCP SYN flood (a.k.a. SYN flood) is a type of DDOS attack that exploits part of the normal TCP three way handshake to consume resources on the targeted server and render it unresponsive. Essentially, with SYN flood DDoS, the offender sends TCP connection requests faster than the targeted machine can process them, causing network saturation.

1.3.2. Protocol attacks

Includes SYN floods, fragmented packet attacks, Ping of Death (POD), Smurf DdoS, and more. This type of attack consumes actual server resources, or those of intermediate communication equipment, such as firewalls and load balancers, and is measured in packets per second (Pps).

1.3.3. Ping of Death (POD)

A 'POD' attack involves the attacker sending multiple malformed or malicious pings to a computer. The maximum packet length of an IP packet (including header) is 65,535 bytes. However, the Data Link Layer usually poses limits to the maximum frame size – for example, 1,500 bytes over an Ethernet network. In this case, a large IP packet is split across multiple IP packets (known as fragments), and the recipient host reassembles the IP fragments into the complete packet. In a POD scenario, following malicious manipulation of fragment content, the recipient ends up with an IP packet that is larger than 65,535 bytes when reassembled. This can overflow memory buffers allocated for the packet, causing a denial of service for legitimate packets.

1.4. Impact of DDOS attack

DoS does not usually try to steal information or lead to a security breach, but the loss of reputation for the affected company can still cost a large amount of time and money. Often customers also decide to switch to an alternative provider, as they fear future security issues, or simply cannot afford to have an unavailable service. A DoS attack lends itself to activists and blackmailers – not really the best situation for companies to find themselves in.

1.5. Real time examples of DDOS attack

The most well-known and spectacular DoS attacks in the last few years.

- Summer 2014: A massive 300 Gbps DDoS attack exploited flaws of 100,000 unpatched servers, joined together as a botnet. An unidentified data centre was faced with the extremely huge scale of a DDoS attack.

- Spring 2015: UK-based phone carrier Carphone Warehouse gets targeted by a DDos attack – while hackers steal millions of customers' data

- July 2015: The New York Magazine gets hit by a DDoS attack just after publishing interviews of 35 women accusing Bill Cosby of sexual assault.

- December 2015: Threats of a DDos attack on Microsoft's Xbox Live service claim to take down both the XBox Live and PlayStation network over the Christmas period for up to a week. The attackers are trying to highlight the continued weak security of Microsoft's services.

- January 2016: The latest target of a sophisticated DDos attack saw some of the HSBC customers losing access to their online banking accounts two days before the tax payment deadline in the United Kingdom.

1.6. How does DDOS attack works?

DDoS attacks are one of the most common forms of cyber attack, with the number of global DDoS attacks increasing to 50 million annually, according to VeriSign. DDOS or DDoS in short, refers to a cyber attack resulting in victims being unable to access systems and network resources, essentially disrupting Internet services. The DDoS attack will attempt to make an online service or website unavailable by flooding it with unwanted traffic from multiple computers. For a DDoS attack to be successful, an attacker will spread malicious software to vulnerable computers, mainly through infected emails and attachments. This will create a network of infected machines which is called a

botnet. The attacker can then instruct and control the botnet, commanding it to flood a certain site with traffic: so much that its network ceases to work, taking the site offline.

1.7. Purpose of DDOS attack

DDoS attacks are usually done by individuals calling themselves hacktivists. Their purpose is to crash a website server by overwhelming it with activity by 'bots' and is generally an aim with a political purpose. These attacks can take up 111 to 179 gbps, and in April of 2013, took 144 million Pps. If the hacktivists are against a company's stance on a specific issue or multiple issues, the hacktivist will attack in an attempt to show opposition and illustrate the company's weakness. A hacktivist may also attack a company, such as a bank or finance company, which they believe makes money off of the downtrodden of society. Some hackers have used DDoS attacks for extortion in an attempt to make companies pay a designated amount before their servers can go back online. A DDoS attack can even be carried out by hackers who deem the company in competition with another company the hacker supports. It is also beginning to look like DDoS attacks are being used in fraud techniques and many alerts have been issued to companies to be aware of these possible attacks.

2. Requirement analysis

Requirement analysis is the first and important phase of the software developing activity in developing any kind of project effectively. I started to list out all the functionalities that my application should provide. There have been some minor changes with respect to the functionalities over the course of development. Following are the requirements that have been implemented in this project. The existing Machine Learning-based DDoS detection approaches can be divided into two categories.

2.1. Existing system

Our approach constitutes of two main parts unsupervised and supervised. The unsupervised part includes entropy estimation, co-clustering, and information gain ratio. The supervised part is the Extra-Trees ensemble classifiers. The unsupervised part of our approach allows one to reduce the irrelevant and noisy normal traffic data, hence reducing false-positive rates and increasing the accuracy of the supervised part. Whereas, the supervised part is used to reduce the false-positive rates of the unsupervised part and to accurately classify the DDoS traffic. To better evaluate the performance of the proposed approach three public network traffic datasets are used in the experiment, namely the NSL-KDD, the UNB ISCX IDS 2012 dataset, and the UNSW-NB15. The experimental results are satisfactory when compared with the state-of-the-art DDoS detection methods.

The first phase of their approach consists of dividing the incoming network traffic into three types of protocols TCP, UDP, or Other. Then classify it into normal or anomaly traffic. In the second stage, a multi-class algorithm classify the anomaly detected in the first phase to identify the attacks class in order to choose the appropriate intervention. Two public datasets are used for experiments in this paper namely the UNSW-NB15 and the NSL-KDD Several approaches have been proposed for detecting DDoS attack. Information theory and machine learning are the performances of network intrusion detection approaches, in general, rely on the distribution characteristics of the underlying network traffic data used for assessment. The DDoS detection approaches in the literature are under two main categories unsupervised approaches and supervised approaches. Depending on the benchmark datasets used, unsupervised approaches often suffer from high false-positive rate and the supervised approach cannot handle a large amount of network traffic data and their performances are

often limited by noisy and irrelevant network data. Therefore, the need of combining both supervised and unsupervised approaches arises to overcome DDoS detection issues.

2.1.1. Supervised approach

Supervised ML approaches that use generated labeled network traffic datasets to build the detection model.

2.1.2. Unsupervised approach

The unsupervised approaches no labeled dataset is needed to build the detection model. The DDoS and the normal traffics are distinguished based on the analysis of their underlying distribution characteristics.

2.2. Proposed system

This section introduces our methodology to detect the DDoS attack. The five-fold steps application process of data mining techniques in network systems discussed in characterises the followed methodology. The main aim of combining algorithms used in the proposed approach is to reduce noisy and irrelevant network traffic data before preprocessing and classification stages for DDoS detection while maintaining high performance in terms of accuracy, false-positive rate and running time, and low resources usage. Our approach starts with estimating the entropy of the FSD features over a time-based sliding window. When the average entropy of a time window exceeds its lower or upper thresholds the co-clustering algorithm split the received network traffic into three clusters.

When the average entropy of a time window exceeds its lower or upper thresholds the co-clustering algorithm split the received network traffic into three clusters. Entropy estimation over time sliding windows allows to detect abrupt changes in the incoming network traffic distribution which are often caused by DDoS attacks. Incoming network traffic within the time windows having abnormal entropy values is suspected to contain DDoS traffic.

The focus only on the suspected time windows allows to filter the important amount of network traffic data, therefore, only relevant data are selected for the remaining steps of the proposed approach. Also, important resources are saved when no abnormal entropy occurs. In order to determine the normal cluster, we estimate the information gain ratio based on the average entropy of the FSD features between the received network traffic data during the current time window and each one of the obtained clusters.

3. Methodology

3.1. Algorithm

An algorithm is a detailed series of instructions for carrying out an operation or solving a problem. In a non-technical approach, we use algorithms in everyday tasks, such as a recipe to bake a cake or a do-it-yourself handbook. This can be measured with the help of graphical notations such as pie chart, bar chart, and line chart. The data can be given in dynamical data.

Computers use algorithms to list the detailed instructions for carrying out an operation. For example, to compute an employee's paycheck, the computer uses an algorithm. To accomplish this task, appropriate data must be entered into the system. In terms of efficiency, various algorithms are able to accomplish operations or problem solving easily and quickly. The Algorithm used here is Extra Trees algorithm. The Extra-Trees algorithm builds an ensemble of un-pruned decision or regression trees according to the classical top-down procedure. Its two main differences with other tree-based

ensemble methods are that it splits nodes by choosing cut-points fully at random and that it uses the whole learning sample (rather than a bootstrap replica) to grow the trees. The Extra-Trees splitting procedure for numerical attributes uses two parameters: K, the number of attributes randomly selected at each node and nmin, the minimum sample size for splitting a node. It is used several times with the (full) original learning sample to generate an ensemble model (we denote by M the number of trees of this ensemble). The predictions of the trees are aggregated to yield the final prediction, by majority vote in classification problems and arithmetic average in regression problems. In the algorithm, co-clustering method plays an important role. Coclust provides both a Python package that implements several diagonal and non-diagonal co-clustering algorithms, and a ready to use script to perform co-clustering. Co-clustering (also known as bi clustering), is an important extension of cluster analysis since it allows to simultaneously groups objects and features in a matrix, resulting in both row and column clusters. The script enables the user to process a dataset with co-clustering algorithms without writing Python code. Co-Clustering is a machine learning algorithm that can be used for both classification and regression challenges. Co-clustering algorithm performs a simultaneous clustering of rows and columns of a data matrix based on a specific criterion. It produces clusters of rows and columns that represent sub-matrices of the original data matrix with some desired properties.

Steps involved in algorithm are as follows:

Our approach starts with estimating the entropy of the FSD features over a time-based sliding window.

When the average entropy of a time window exceeds its lower or upper thresholds the co-clustering algorithm split the received network traffic into three clusters.

Entropy estimation over time sliding windows allows to detect abrupt changes in the incoming network traffic distribution which are often caused by DDoS attacks.

Incoming network traffic within the time windows having abnormal entropy values is suspected to contain DDoS traffic.

The focus only on the suspected time windows allows to filter the important amount of network traffic data, therefore only relevant data is selected for the remaining steps of the proposed approach.

Also, important resources are saved when no abnormal entropy occurs. In order to determine the normal cluster, we estimate the information gain ratio based on the average entropy of the FSD features between the received network traffic data during the current time window and each one of the obtained clusters.

As discussed in the previous section during a DDoS period the generated amount of attack traffic is largely bigger than the normal traffic. Hence, estimating the information gain ratio based on the FSD features allows one to identify the two clusters that preserve more information about the DDoS attack and the cluster that contains only normal traffic. Therefore, the cluster that produces a lower information gain ratio is considered normal and the remaining clusters are considered as anomalous clusters .after that ddos traffic detection using an extra tree algorithm is done on anomalous traffic this gives the required results.

4. Implementation

4.1. Modules

There are four modules that can be divided here for this project they are listed as below:

4.1.1. User apps

User handling for some various times of smartphones, desktops laptops, and tablets. If any kind of devices attacks for some unauthorised Malware software's. In this Malware on threats for user personal dates includes for personal contact, bank account numbers and any kind of personal documents are hacking in possible.

4.1.2. Classifications of DDOS attack

- Here, we compare the classification performance of co-clustering algorithm with other popular machine learning algorithms.
- We have selected several popular classification algorithms. For all algorithms, we attempt to use multiple sets of parameters to maximise the performance of each algorithm.
- Using Co-clustering algorithm algorithms classification for malware bag-of-words weightage

5. Testing

5.1. Introduction

Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies, and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

5.2. Types of tests

5.2.1. Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages, and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5.2.2. Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as shown by successful unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components. Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test results: All the test cases mentioned above passed successfully. No defects encountered.

5.2.3. Acceptance testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test results: All the test cases mentioned above passed successfully. No defects encountered.

6. Conclusion

Android is a new and fastest growing threat to malware. Currently, many research methods and antivirus scanners are not hazardous to the growing size and diversity of mobile malware. As a solution, we introduce a solution for mobile malware detection using network traffic flows, which assumes that each HTTP flow is a document and analyses HTTP flow requests using natural language processing string analysis. The N-Gram line generation, feature selection algorithm, and SVM algorithm are used to create a useful malware detection model. Our evaluation demonstrates the efficiency of this solution, and our trained model greatly improves existing approaches and identifies malicious leaks with some false warnings. The harmful detection rate is 99.15%, but the wrong rate for harmful traffic is 0.45%. Using the newly discovered malware further verifies the performance of the proposed system.

When used in real environments, the sample can detect 54.81% of harmful applications, which is better than other popular anti-virus scanners. As a result of the test, we show that malware models can detect our model, which does not prevent detecting other virus scanners. Obtaining basically new malicious models Virus Total detection reports are also possible. Added, Once new tablets are added to training samples, we will please re-train and refresh and update the new malware.

For future work, we are planning to perform real-world deployment of the proposed approach and evaluate it against several DDoS tools.

REFERENCES

Ahmed, M., & Mahmood, A. N. (2014). Network traffic pattern analysis using improved information theoretic co-clustering based collective anomaly detection. In *International conference on security and privacy in communication systems* (pp. 204–219). Springer.

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2015). An empirical evaluation of information metrics for low-rate and high-rate ddos attack detection. *Pattern Recognition Letters, 51*, 1–7.

Boroujerdi, A. S., & Ayat, S. (2013). A robust ensemble of neurofuzzy classifiers for ddos attack detection. In *2013 3rd International conference on computer science and network technology (ICCSNT)* (pp. 484–487). IEEE.

Chang, R. K. C. (2002). Defending against flooding-based dis- tributed denial-of-service attacks: a tutorial. *IEEE Communications Magazine, 40*(10), 42–51.

Lin, S. C., & Tseng, S. S. (2004). Constructing detection knowledge for ddos intrusion tolerance. *Expert Systems with Applications, 27*(3), 379–390.

Papalexakis, E. E., Beutel, A., & Steenkiste, P. (2014). *Network anomaly detection using co clustering. In: Encyclopedia of social network analysis and mining* (pp. 1054–1068). Springer.

Saied, A., Overill, R. E., & Radzik, T. (2016). Detection of known and unknown ddos attacks using artificial neural networks. *Neurocomputing, 172*, 385–393

Wikipedia (2016) 2016 dyn cyber attack. Wikipedia.

Yu, S. (2014). Distributed denial of service attack and defense. Springer.