An Approach of DDOS Attack Detection Using Classifiers

Chapter · January 2015

DOI: 10.1007/978-81-322-2550-8_41

Johnson kh
National Institute of Technology, Manipur
11 PUBLICATIONS **26** CITATIONS

SEE PROFILE

Tanmay De
National Institute of Technology, Durgapur
71 PUBLICATIONS **164** CITATIONS

SEE PROFILE

An Approach of DDOS Attack Detection Using Classifiers

Khundrakpam Johnson Singh and Tanmay De

Abstract To defend and protect web server from the attack, it is important to know the nature and the behaviour of legitimate and illegitimate clients. It is also important to provide access to the legitimate clients and provide a defence system against illegitimate clients. The Distributed Denial of Service (DDoS) attack is a critical threat to the Internet. By using its application layer protocol DDoS can cause a massive destruction by silently making an entrance to the web server as it act as one of the legitimate clients. The paper uses parameter of the network packet like http GET, POST request and delta time to compute the accuracy in finding out the possible attack. We use different classifiers like Naive Bayes, Naive Bayes Multinomial, Multilayer Perception, RBF network, Random Forest etc. to classify the attack generated dataset. We compare the accuracy, true positive rate, false positive rate of each algorithm by finding the confusion matrix.

Keywords Ddos · Http · Naive bayes · Attack tools · Weka · Monitoring tools

1 Introduction

Millions of people depend on internet for discussion and sharing knowledge with the outside world; Websites are porn to malicious attacks thereby preventing legitimate people from accessing it. Out of numerous ways of malicious behaviour of attacking websites, overloading a website with unwanted traffic is a common one. When someone maliciously uses several computers to do this, it is termed as distributed denial of service attack (DDoS) [1]. To build a DDoS attack, the attacker will often start by building network of infected machine known as botnet. It is done by spreading malicious software through email, websites and social media. The botnet can be control and commanded like an army to send an overwhelming volume of traffic that can take target website offline. Botnet can also be hired or sold

K.J. Singh (✉) · T. De
Department of CSE, National Institute of Technology, Durgapur, India

N.R. Shetty et al. (eds.), *Emerging Research in Computing, Information, Communication and Applications*, DOI 10.1007/978-81-322-2550-8_41

out to people who want to bring someone's site offline. These happen mostly in political and competition among the business organization irrespective of the size. DDoS attack are easy to initiate, anyone anywhere in the world can take a site down no matter the size. These attacks are caused due to competition, ransoms, fun etc.

These attacks are a nightmare for small and individual site as they can be flooded by a small network of botnet traffic. They lack the resources and the infrastructure to tackle the problem. DDoS attack is broadly classified into two category network layer attack and application layer attack [2]. The main aim of the layer 3 DDoS attack is to overwhelm the server and used up the bandwidth with floods. The motives of the application layer attack are crashing the server by low and slow connections. Some of the typical layer 7 attack tools are slowloris [3], RUDY [4] etc.

1.1 Are You Dead Yet?

It is categorized as low and slow attack tools as it generates a slow rate and low volume of traffic. It is difficult to detect by standard anti DoS mitigation system. The Rudy tool exploits vulnerabilities in the http protocol. When a user fills a web-form its web browser sends data to the web server using http POST request. When a legitimate user sends data, it uses one or two packets to the web server. The server then closes the connection and moves to handle the next request.

However when an attacker use the Rudy tool the data from the form is broken into many packets, where each packet contains only one byte of the data. The Rudy tools then sends the packets at random time intervals preventing the web server from closing the connection, posting it to wait for the request to complete. A few thousand requests generated by the Rudy tools over several minutes would cost the web server to stop handling new request and prevent the server from the legitimate users, hence achieving denial of service. Any website that contain web form such as login information, feedback forms or web mail that accept http POST request is susceptible to Rudy attack.

1.2 Slowloris

Slowloris tool is a low and slow attack that generates a denial of service attack. Slowloris is categorized as a low and slow attack tool because it generates a slow rate and low volume of traffic and therefore it is difficult to detect by standard and dos mitigation system. It is one of the first known slow rate dos tools developed. The slowloris attack tool exploits a weakness in the http protocol, requiring every http request to be terminated by a sequence of new line characters. A legitimate http GET request normally contained in one packet and is terminated quickly by the sequence of newline at the end of the message.

However when an attacker is using the slowloris tool the http request is sent without the termination sequence. This causes the web server to leave the connection open and to allocate resources that are waiting for the termination sequence. The typical web server allocates limited resources for handling open connections and it expects the connection to be short and terminated quickly. Slowloris take advantage of this and generates thousand of request over several minutes, where none of the request is terminated. This consumes the available open connection resources of the web server causing it to stop handling new request and prevent the service from legitimate users.

The rest of the paper is organized as follows. Section 2 provides related works. The proposed method with the comparison with other methods is presented in Sect. 3. Section 4 describes experimental results and a discussion for detecting DDoS attacks. Finally, we present the concluding remarks in Sect. 5.

2 Related Work

Detection and prevention of duplicate request attacks, accounting the client history is important [5]. This paper discusses technique about cookie based on accounting model. It keeps a record of every client request in the cookie. The paper tries to detect the client's abnormal behaviour by considering the hash value of the cookie in the server database. This process includes modifying the cookie information or resending the prior request cookie with the current request.

A new approach to detect botnet activity based on traffic behaviour analysis by classifying network traffic behaviour using machine learning is discussed [6]. Traffic behaviour analysis methods are independent of packets payload. It means that they can work with encrypted network communication protocols. This paper discusses about the possibilities of detecting botnet activity without having seen a complete network flow by classifying behaviour based on time intervals.

Cross Layer Design techniques [7] for detecting and identifying the attackers is discussed. It helps to mitigate the attack by reducing the unwanted impact on the network. This paper focused on the powerful DoS attack in wireless networks using IEEE 802.11 Distributed Coordination Function (DCF) protocols. Two-dimensional Markov Chain is analyzed to get the maximum throughput and identify the DoS attackers.

Distributed divide-and-conquer approach [8] is designed for new data dissemination that efficiently tracks attack sources. This paper consider three different categories of the problem which can be named as construction of attack tree, frequency detection of attack path, and association of packet to path. Hundreds of victim clients which are capable of thousands of high-bandwidth flows are supported. This method can truly achieve single packet trace back guarantees with minimum overhead and high efficiency.

Re-Traffic Pricing (RTP) [9] provides technique against application-level DDoS attacks. It prevents the server from overloading by encouraging all the users to spend re-traffic. It is assumed that legitimate users have greater re-traffic value than the attackers to contest for the service resources. The motives of RTP architecture are to allocate service resources in an approximate proportion to the users' re-traffic. As a benefit of this mechanism, legitimate clients with greater re-traffic will receive the bulk of the service.

Edge based Capabilities (EC) [10] make use of its framework by combining end to end authentication with network-based control. The EC applications try to authenticate legitimate clients and provide capabilities to tag their packets filtering out untagged packets. It can be incrementally deployed because it helps the users, servers and ISPs by providing with the right spur.

A new Botnet 2.0 [11], which attempts to exploit the flexibility of Web 2.0 to maintain and expand the botnet, is introduced. The proposed method applies a unique identifier of the computer, an encryption algorithm with session keys and CAPTCHA verification.

3 Proposed Method

The proposed method tries to detect the entire possible Layer Seven DDoS attack or application layer DDoS attacks on the web server. We extract the parameters like http count, delta time of the packet captured. Layer seven DDoS attack are low volume and act itself as a legitimate transaction thus are not able to detect by firewall or IDS systems. Our proposed method in its early stage capture all the packets from the attack source thereby enabling us to select the parameters like number of http GET or POST request from a single IP address. We also select the parameter like delta time, which can be defined as the time interval between any two consecutive http requests sent by a single IP address.

Since Layer Seven or application layer DDoS attack uses the http protocol to use up the recourses in the web server, we consider the IP addresses having maximum number of http count towards a single IP destination address. As a normal human user will not be able to send http requests one after another at high speed, we consider the delta time between any two consecutive requests. The smaller the delta time value the possibility of carrying out the attack is greater.

The above parameters are considered to detect whether an IP address has the potential of carrying out DDoS attack. The dataset having these four parameters are fed into the classifiers like NaiveBayes Multinomial [12], NaiveBayes [13], MultiLayerPerceptron [14], Random Forest [15], RBF Network [16] Logistic [17] and find out the possibilities.

The pseudo code of Naive Bayes Multinomial algorithm is divided into two parts the training part and the testing part.

```
Algorithm: Training NaiveBayes Multinomial (P,Q)
$\lambda \leftarrow$ ExtractVocabularly(Q)
$N \leftarrow$ CountDocs(Q)
for each $c \in P$
do $N_c \leftarrow$ CountDocsInClass(Q,c)
prior[c] $\leftarrow N_c/N$
text$_c \leftarrow$ ConTextODatasetsInClass (Q,c)
for each t □ $\lambda$
do $T_{ct} \leftarrow$ CountTokensOfTerm (text$_c$, t)
for each $t \in \lambda$
do cond prob[t][c]$\leftarrow \frac{T_{ct}+1}{\sum_{t'}(T_{ct'}+1)}$
return V, prior, condprob
```

```
Algorithm: TEST NaiveBayes Multinomial (P, $\lambda$, prior, cond prob, d)
$\mu \leftarrow$ ExtractTockensFromDataset($\lambda$, d)
for each $c \in P$
do score[c]$\leftarrow$ log prior[c]
for each $t \in \mu$
do score[c]+=log condprob[t][c]
return arg max$_{c□P}$ score[c]
```

We find out the confusion matrix (contingency table), a machine learning field to visualize the performance of the algorithm used in the classification. It is a supervised learning where each column of the matrix provides the instances in a predicted category and each row provides the instances in an actual category.

The descriptions of A, B, C and D of the confusion matrix given in Table 1 are:

1. A provides the correct predictions that an instance is negative,
2. B provides the incorrect predictions that an instance is positive,
3. C provides the incorrect of predictions that an instance is negative, and
4. D provides the correct predictions that an instance is positive.

$$\text{Accuracy} = \frac{A + D}{A + B + C + D} \tag{1}$$

$$\text{True Positive} = \frac{D}{C + D} \tag{2}$$

$$\text{False Positive} = \frac{B}{A + B} \tag{3}$$

The computed confusion matrix for each algorithm is illustrated in Table 2. The confusion matrix of each algorithm used is calculated using the DDoS attack dataset that is generated using the experimental setup in Sect. 4. For calculating the confusion matrix we consider only 79 items and their associated http count and delta time. The algorithms try to find out the possible attack by considering their respective combination of inputs.

Using the confusion matrix and the associated values we compute the accuracy as in Eq. 1, true positive as in Eq. 2 and false positive as in Eq. 3 of each algorithm

Table 1 A general confusion matrix

Actual	Predicted	
	Negative	Positive
Negative	A	B
Positive	C	D

Table 2 Confusion matrix of multiple classifiers

Sl. no	Algorithm	Confusion matrix
1.	NaiveBayes	$\begin{bmatrix} 28 & 4 \\ 3 & 44 \end{bmatrix}$
2.	NaiveBayes multinomial	$\begin{bmatrix} 31 & 1 \\ 4 & 43 \end{bmatrix}$
3.	MLP	$\begin{bmatrix} 28 & 4 \\ 5 & 42 \end{bmatrix}$
4.	Random forest	$\begin{bmatrix} 28 & 4 \\ 3 & 44 \end{bmatrix}$
5.	Logistic	$\begin{bmatrix} 29 & 3 \\ 3 & 44 \end{bmatrix}$
6.	RBFNetwork	$\begin{bmatrix} 26 & 6 \\ 2 & 45 \end{bmatrix}$

used for classification. The computed value of accuracy, false positive and false negative are given in Table 3.

The comparison shows that Naive Bayes Multinomial has the highest accuracy of 93.67 % with lowest false positive rate of 03.10 %. We consider NaiveBayes Multinomial for the final classification of the dataset. It is given that Naive Bayes algorithm uses limited resources in terms of CPU and memory. It also takes less training time and train quickly [18].

4 Experimental and Discussion

To evaluate the performance of the proposed solution, we adopt a factual internet connected victim server with DDoS attack discovery setting. We install a victim machine running apache tomcat service of version 2.4.9 on window operating

Table 3 Comparison of the different classifiers with the input dataset

Sl. no	Algorithm	Accuracy (%)	True positive (%)	False positive (%)
1	NaiveBayes	91.14	93.62	12.50
2	NaiveBayesMultinomial	93.67	91.49	03.10
3	MLP	88.61	89.36	12.50
4	Random forest	91.14	93.62	12.50
5	Logistic	92.41	93.62	12.50
6	RBFNetwork	89.87	95.74	18.75

system (window 7). The running of apache tomcat service on window is made possible by installing the Wamp server and enabling the apache services. The www directory will be created automatically and the default web page will be available on the URL http://localhost.

The attack scenario is created in the CentOS version 6.1. We virtually installed the CentOS on the VMware by providing the RAM of 1 GB, 2.7 GHz Intel CPU and 40 GB hard disk. In order to carry out the attack we installed some slowloris package called sowhttptest 1.6 on CentOS. We also install a few RUDY tools and low IONS tools in Ubantu OS, which is already installed in the VMware. This attack machines are configured to access the Wamp server running apache tomcat service. We create some legitimate machine on CentOS and are also configured to access the web server.

The web server must me reachable from the attacker machines and the legitimate clients. We further check the status of all the machines using ping command. This command is mainly used to verify a computer whether it can communicate over the network with another computer or network device. We provide an Ubantu Operating system of 4 GB RAM, 3.33 GHz Intel(R) Core (TM) i5 CPU, 80 GB hard disk as an intermediate node between the web server and the attacks, clients. All the access to the web server has to be passed through this Ubantu operating system. We virtually installed the Ubantu OS in the VMware and provide with two NIC card Vmnet1 as incoming interface and Vmnet 2 as outgoing interface. In

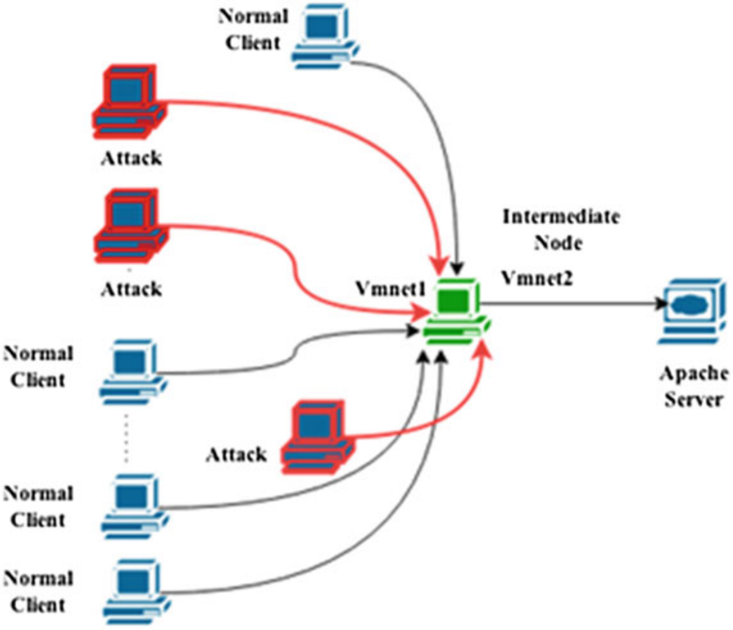

Fig. 1 A configuration of the DDoS attack scenario

order to trap the packets from all the connected nodes accessing the server, we install the wireshark 1.12.2 on to the intermediate node.

In our experimental setup we provide almost 300 slowloris attack hosting machine and 100 legitimate machines. The configuration of the above scenario is given in Fig. 1. We use weka 3.6 for the classification of the dataset and computing the threshold.

5 Conclusion

It is not possible to achieve 100 % accuracy in detecting the DDoS attack in a network or achieve a complete defence against these attacks at a single stage. From the practically acquired DDoS attack dataset, we found out that Naive Bayes Multinomial achieve 93.67 % accuracy in detecting the attack and a small false positive rate of 03.10 %. Setting up the firewall rule to block such IP addresses will be accurate measures for preventing against the application layer DDoS attack. We consider only a medium scale attack scenario with 1000 clients and a web server; these can be extended to a larger network of DDoS attack and compute the accuracy of detection using different new classifiers.

References

1. McGregory, S.: Preparing for the next DDoS attack. Netw. Secur. **2013**(5), 5–6 (2013). ISSN:1353-4858
2. Mansfield-Devine, S.: DDoS: threats and mitigation. Netw. Secur. **2011**(12), 5–12 (2011). ISSN:1353-4858
3. Hoque, N., Monowar, H., Bhuyan, R.C., Baishya, D.K., Bhattacharyya, J., Kalita, K.: Network attacks: taxonomy, tools and systems. J. Netw. Comput. Appl. **40,** 307–324. ISSN:1084-8045
4. McGregory, S.: Preparing for the next DDoS attack. Netw. Secur. **2013**(5), 5–6 (2013). ISSN:1353-4858
5. Venkatesan, S., Saleem Basha, M.S., Chellappan, C., Dhavachelvan, A.V.P.: Analysis of accounting models for the detection of duplicate requests in web services. J. King Saud Univers.–Comput. Inform. Sci. **25**(1), 7–24 (2013). ISSN:1319-1578
6. Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., Garant, D.: Botnet detection based on traffic behavior analysis and flow intervals. Comput. Secur. **39**(Part A), 2–16 (2013). ISSN 0167-4048
7. Soryal, J., Saadawi, T.: IEEE 802.11 DoS attack detection and mitigation utilizing cross layer design. Ad Hoc Netw. **14,** 71–83 (2014). ISSN:1570-8705
8. Muthuprasanna, M., Manimaran, G.: Distributed divide-and-conquer techniques for effective DDoS attack defenses. In: The 28th International Conference on, Distributed Computing Systems, ICDCS ‘08, pp. 93, 102, 17–20 June 2008
9. Shen, Y.-Y., Fan, F.-Q., Xie, W.-X., Mo, L.-F.: Re-traffic pricing for fighting against DDoS. In: ISECS International Colloquium on, Computing, Communication, Control, and Management, CCCM ‘08, vol. 2, pp. 332, 336, 3–4 Aug 2008

10. Karrer, R.P., Kuehn, U., Huehn, T.: Joint application and network defense against DDoS flooding attacks in the future internet. In: Second International Conference on, Future Generation Communication and Networking, FGCN'08, vol. 1, pp. 11, 16, 13–15 Dec 2008
11. Vo, N.H., Pieprzyk, J.: Protecting Web 2.0 Services from Botnet Exploitations. Cybercrime and Trustworthy Computing Workshop (CTC), 2010, vol. 2, pp. 18, 28, 19–20 July 2010
12. Bermejo, P., Gámez, J.A., Puerta, J.M.: Improving the performance of Naive Bayes multinomial in e-mail foldering by introducing distribution-based balance of datasets. Expert Syst. Appl. **38**(3), 2072–2080 (2011). ISSN:0957-4174
13. Peng, J., Chan, P.P.K.: Revised Naive Bayes classifier for combating the focus attack in spam filtering. In: International Conference on, Machine Learning and Cybernetics (ICMLC), vol. 2, pp. 610, 614, 14–17 July 2013
14. Zhang, Z., Shao, W., Zhang, H.: A learning algorithm for multilayer perceptron as classifier. In: International Joint Conference on Neural Networks, IJCNN '99, vol. 3, pp. 1681, 1684 (1999)
15. Aung, W.T., Saw Hla, K.H.M.: Random forest classifier for multi-category classification of web pages. In: Services Computing Conference, APSCC 2009. IEEE Asia-Pacific, vol. pp. 372, 376, 7–11 Dec 2009
16. Xu, R., An, R., Geng, X.F.: Research intrusion detection based PSO-RBF classifier. In: 2011 IEEE 2nd International Conference on, Software Engineering and Service Science (ICSESS), pp. 104, 107, 15–17 July 2011
17. Kostadinov, D., Bogdanova, S.: Logistic regression classifier for palmprint verification. In: 2012 19th International Conference on, Systems, Signals and Image Processing (IWSSIP), pp. 413, 416, 11–13 April 2012
18. Huang, J., Lu, J., Ling, C.X.: Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. In: Third IEEE International Conference on, Data Mining, 2003. ICDM 2003, pp. 553, 556, 19–22 Nov 2003