Proceedings of 12th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2022 Month: October Volume: 12

Long Short Term Memory Based Web Application Firewall to Detect Different Types of Web Attacks

Bibek Adhikari [a], Babu R. Dawadi [b]

[a,b] *Department of Electronics and Computer Engineering, Pulchowk Campus, IOE, TU, Nepal*
[a] biwek.adhikaryma@gmail.com, [b] baburd@ioe.edu.np

Abstract
With the emerging new technologies, security is a challenging part that has bigger concerns with the increasing cyber threat in the modern world of computing technologies. New techniques and tactics are being used to take unauthorized access to the web and harm, steal, and destroy the information. Protecting the system from many threats like DDoS, SQL injection, cross-site scripting, etc. is very challenging. This research work makes a comparative analysis between normal HTTP traffic and attack traffic, identifies the attack-indicating parameters and features, and develops a layered architecture model for DDoS, XSS, and SQL injection attack detection using data collected from the simulation environment. In an LSTM-based layered architecture, the first layer is the DDoS detection model with an accuracy of almost 97% and the second is the XSS and SQL injection layer with an accuracy of almost 89%. The higher rate HTTP traffic is checked first, filtered out, and then only passed to the second layer. In this way, the performance of the attack detection system can increase.

Keywords
LSTM,DDoS, WAF, XSS, SQL Injection, Web security

1. Introduction

One of the common difficulties in various disciplines of computer science is protecting computers and networks from infiltration, theft, and disturbance. The importance of a security system increases as the number of internet users increases. Many attempts have been made to build various security solutions, such as intrusion detection systems and firewalls. In most cases, network layer firewalls and intrusion detection systems do not inspect HTTP packets in the application layer[1]. As a result, they are incapable of fully safeguarding Web servers. Web applications, especially those in the cloud, are one of the most appealing targets for attackers looking to break into an organization's information infrastructure. Internal data leaks, financial and credit losses, and website manipulation can all result from an organization's failure to implement web security. A WAF is a tool to identify and prevent many types of attacks, such as SQL injection, XSS, and DDoS [2]. WAFs use IDS methods in the application layer to secure web applications.

Many users depend on web applications for education, banking, social media, information, etc. However, when using these applications, the existence of security vulnerabilities can bring risks. Attackers can use these vulnerabilities to get access to this sensitive information and send bad HTTP requests or install malware, redirect unsuspecting users to malicious websites, and engage in other malicious activities by controlling, filtering, and monitoring HTTP traffic between a web application and client on the Internet. A web application firewall helps to secure online applications. It usually defends online applications against threats like cross-site scripting (XSS), cross-site forgery, SQL injection, DDOS, etc[3][4][5]. A web application firewall is a web security measure that works on protocol layer 7 defense and is designed to fight against different forms of attack in the application layer. This type of attack minimization is usually part of a larger set of technologies that work together to provide comprehensive protection against a variety of threats[2].

Hence, we introduce a layered architecture of Web application firewall. Based on the traffic rate, the higher detection rate is filtered in the first layer, and only filtered traffic from the first layer is processed in the second layer. Since the nature of the attack is different for different attacks, extracting the required

parameter based on the nature of the attack and predicting the new request using a pre-trained model will increase the performance and accuracy of the web application firewall.

2. Background and related work

2.1 A Web application firewall

A web application firewall acts as a barrier between a web application and the client on the internet when it is deployed in front of a web application[2]. A web application firewall is a type of reverse proxy that protects the web server from exposure to the client by detecting bad traffic in the web application firewall. Where a proxy server acts as an intermediary to protect a client machine's identity. A WAF is controlled by a set of rules known as policies and the pre-trained module to predict the new incoming requests[1]. These policies try to guard against application vulnerabilities by filtering out harmful communications. A WAF's usefulness is derived partly from the speed and ease with which policy modifications may be deployed, allowing for a faster reaction to various attack vectors.

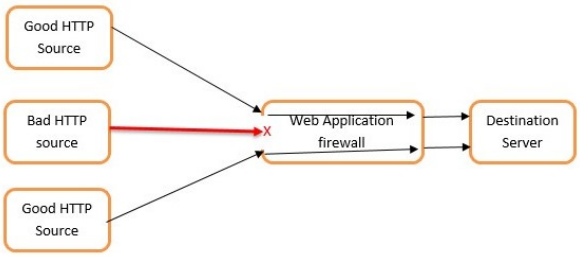

Figure 1: WAF architecture

2.2 Cross-Site Scripting

Cross-site Scripting is an injection attack that occurs when attackers use vulnerabilities in trusted websites to inject any malicious code, this code can be implemented to steal personal information from the site, such as login information, session cookies, and sensitive information, and it can even remain on the website permanently to continue targeting multiple users[5].

The two approaches for attackers to insert malicious code into a webpage are:

- Reflected XSS
  A reflected or non-persistent XSS attack is passed to the victim by the attacker through a different medium, such as an email, message, or another website, and this allows an attacker to send malicious code as part of a server request that is directed to a vulnerable site and then returns to the user's browser.
- Stored XSS
  A stored or persistent XSS attack occurs when malicious content is stored on the targeted server, and when a user requests the stored information from that malicious web server, which may be a web page with a malicious script, the code will be returned as part of the message.

2.3 SQl Injection

SQL injection is a sort of online security issue that allows an attacker to manipulate database queries in a web application. It gives an attacker access to data they wouldn't normally have access to. This could include data from other users as well as any other information that has access to. An attacker can change or erase this data in many cases, causing the application's content or behavior to be permanently changed[4].

2.4 DDoS

DDoS attacks are deliberate attempts to interrupt the normal traffic of a targeted server, service, or network by flooding the target or its surrounding infrastructure with Internet traffic. Since DDoS attacks leverage numerous compromised computer systems as attack traffic sources, so they are effective[3]. A DDoS attack is similar to unanticipated traffic congestion that prevents regular traffic from reaching its target.

2.5 Related work

Gustavo et al.[6] explored deep learning techniques for the analysis of HTTP traffic. The author used a transformer encoder to analyze the HTTP traffic for easier classification of HTTP traffic.

Moradi et al. [1] used the n-gram feature extraction model in web application firewall to detect anomalies. The author used three different Machine learning models and compare the performance of these models.

Pen et al. [7] analyzed the XSS and SQL injection in supervised and unsupervised learning models and proposed the auto-encoder-based model for the detection of such attacks.

Rajesh et al.[3] proposed the analysis of different

features for DDoS attack detection. The author also presents the comparative analysis result of the system in different machine learning methods.

Lente et al.[5] used the LSTM model by converting the word into the vector and implementing and evaluating the performance in the different batch sizes of the LSTM model.

Keralan et al.[8] performed analysis in different data sets to analyze the HTTP traffic for the web attacks. The authors used tokenization, data argumentation, and k-fold validation to train the model for the prediction of normal and malicious traffic.

In this research work, we focus on analyzing the standard and generated data sets to find the suitable features/parameters to detect the attack. The goal is to implement a deep learning-based web application firewall in a layered architecture that detects DDoS, XSS, and SQL injection.

3. Proposed Model

3.1 System architecture

Web application firewall exists between the web server and client. Incoming HTTP traffic is parsed and analyzed in the request processing unit WAF. The WAF is trained with the training data set, it predicts whether the new incoming HTTP traffic is good or bad. Since the nature of DDoS attacks is different from the other two XSS and SQL injection, so the system is trained with a separate appropriate data set. The new HTTP request is parsed and extracted the parameters required for the prediction by the module. Then it applies to the pre-trained module for the prediction, if the HTTP traffic is classified as a Bad request the request is dropped otherwise it is passed to the second module for testing the SQL injection and XSS. In the same way, this module identifies whether the HTTP traffic is good or Bad. If the HTTP traffic is predicted as good traffic the HTTP session is passed to the Web Server, Otherwise, the HTTP session is discarded/dropped in the WAF itself. Since the rate of DDoS requests is very high if we checked DDoS in the first layer of the WAF the accuracy and performance of WAF system increase.

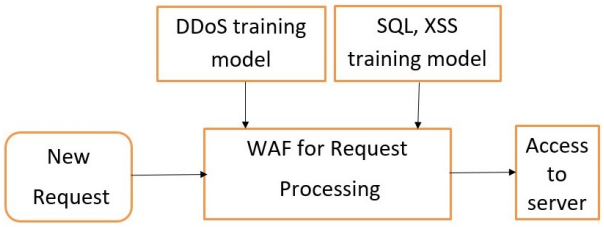

Figure 2: Model Development framework

3.2 Data collection

Cross-site and SQL injection data is collected with help of the DVWA application using 5000 different payloads related to XSS and SQL injection. DVWA tool provides the platform to test the different types of attack, Payloads passed through the respective attack field in the DVWA and forwarded HTTP traffic is collected in the proxy tool Burp suite. Both normal HTTP traffic and traffic with payloads are collected and extracted the specific parameters/features.

Similarly, for the correlative analysis of DDoS, SQL injection, and XSS the same data is collected with the help of the Wire shark tool. The traffic which is collected by inserting payloads one by one should be categorized as normal traffic in the DDoS attack detection model.

Also for the DDoS attack data collection set Low Orbit Ion Cannon (LOIC) in a virtual environment. LOIC is a DDoS attack tool used for flooding the TCT, UDP, and HTTP DoS traffic. We used 4 instances of LOIC to send the HTTP traffic toward a test website. Also, the normal traffic is collected while visiting the site normally. The DDoS and normal traffic is collected in a Wire shark tool. The collected raw log is processed to extract the useful parameters for the model development by training the system.

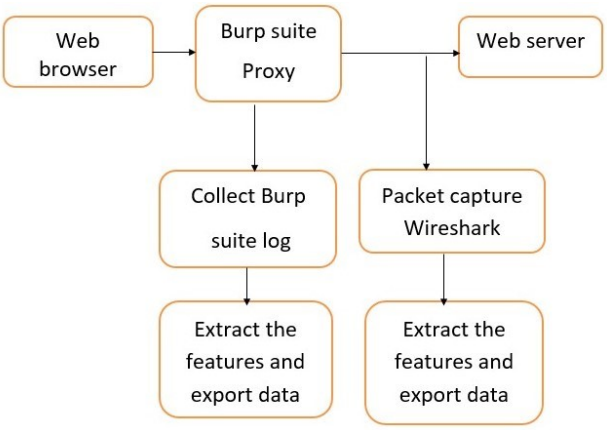

Figure 3: Data collection Methodology

4. Result and analysis

4.1 Standard Data set analysis

We Used the CISC2010 HTTP traffic to analyze the normal HTTP traffic. Find the character present in the HTTP traffic and the number of such characters present in the normal traffic. Also, HTTP traffic with payload is parsed to find the characters and count present in it. During the analysis, the standard character like #,”,’,,=, space, and words like select, script, delete, drop, etc. are the most distinguishing parameters to differentiate between normal traffic and attack traffic.

Also, we used IDS ISCX2012 and CIC2019 DDoS data sets to analyze and find out the parameters and their nature in normal conditions and DDoS attack conditions. Figure out the most distinguishable parameter to differentiate between normal traffic and attack traffic. During the analysis the parameters like header length, don’t fragment flag, Time to live value, IP protocol used, Acknowledgement flag, ports used, and flow rate are the most useful parameters to distinguish normal and DDoS attack traffic.

4.2 Generated Dataset

We collected XSS and SQL injection HTTP traffic in the simulation environment with and without payloads. By parsing the collected data we extracted the character and special words present in the raw data.

Similarly, by parsing the DDoS traffic collected in Wire shark, extracted the numeric value of the features like segment length, packet length, windows scaling factor, windows size, don’t fragment flag, flow rate, Delta time from the previous capture, reserve bit, etc.

4.3 Accuracy and Loss in XSS, SQL injection module

While implementing the generated data set for the XSS and SQL injection detection in the LSTM model the train and test accuracy was obtained as shown in figure 4. The accuracy obtained from the module is almost 89%.

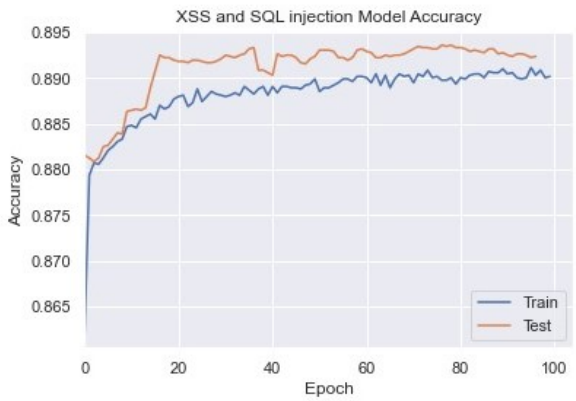

Figure 4: Accuracy and loss of SQL injection, XSS model

4.4 Accuracy and Loss in DDoS detection Module

While implementing generated DDoS data set in the LSTM model the train and test accuracy was obtained in the DDoS Detection module as shown in the figure below.

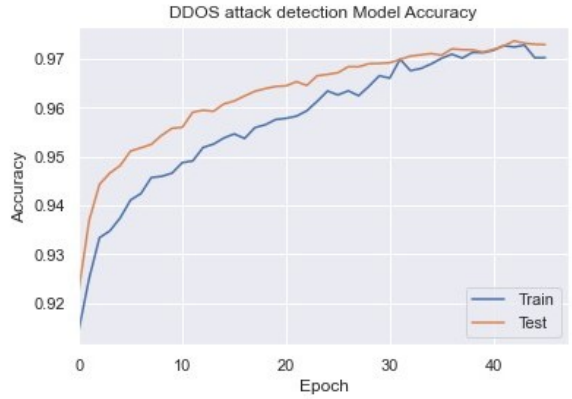

Figure 5: Accuracy and loss of DDoS detection model

The accuracy obtained from the model is almost 97%

4.5 Flow in the Model

In a pre-trained model, weights are adjusted as per the training data set. Once the new traffic is passed through the WAF the features of incoming traffic are extracted and applied to the DDoS detection module. If it is detected as good, it is passed to the XSS, SQL injection module otherwise drops in this layer. Similarly passed traffic is checked for XSS and SQL injection in the second layer, only traffic classified as good reaches to the web server.

4.6 Limitation

The nature of HTTP traffic toward the different sites may differ, so studied features may not be sufficient to identify the good or bad traffic. For the DDoS data preparation, we have used only 4 instances of LOIC which might not be sufficient because in a real environment thousands of such bots are used.

5. Conclusion

The proposed model detects DDoS, XSS, and SQL injection attacks using LSTM deep learning modules with good accuracy. We analyzed and used additional features and parameters for attack detection which reduce the false positive during the traffic filtering in the WAF. Since DDoS traffic is at a higher rate than normal, it improves the system's performance when we checked the traffic in the layered format ie first check for DDoS then SQL injection, and XSS.

6. Recommendation

This research is limited to three types of attack DDoS, XSS, and SQL injection only further, we can add other types of attacks like Remote code execution, Brute force, path traversal, etc. We can analyze the correlation between similar types of attacks and implement them in a single module as here we have used XSS and SQL injection in a single module because it has similar attack-indicating features and parameters.

References

[1] Ali Moradi Vartouni, Mohammad Teshnehlab, and Saeed Sedighian Kashi. Leveraging deep neural networks for anomaly-based web application firewall. *IET Information Security*, 13(4):352–361, 2019.

[2] Michiaki Ito and Hitoshi Iyatomi. Web application firewall using character-level convolutional neural network. In *2018 IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA)*, pages 103–106. IEEE, 2018.

[3] Shriram Rajesh, Marvin Clement, Sooraj SB, Al Shifan SH, and Jyothi Johnson. Real-time ddos attack detection based on machine learning algorithms. *Available at SSRN 3974241*, 2021.

[4] Manar Hasan Ali AL-Maliki and Mahdi Nsaif Jasim. Review of sql injection attacks: Detection, to enhance the security of the website from client-side attacks. *International Journal of Nonlinear Analysis and Applications*, 13(1):3773–3782, 2022.

[5] Caio Lente, Roberto Hirata Jr, and Daniel Macêdo Batista. An improved tool for detection of xss attacks by combining cnn with lstm. In *Anais Estendidos do XXI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 1–8. SBC, 2021.

[6] Nicolás Montes, Gustavo Betarte, Rodrigo Martínez, and Alvaro Pardo. Web application attacks detection using deep learning. In *Iberoamerican Congress on Pattern Recognition*, pages 227–236. Springer, 2021.

[7] Yao Pan, Fangzhou Sun, Zhongwei Teng, Jules White, Douglas C Schmidt, Jacob Staples, and Lee Krause. Detecting web attacks with end-to-end deep learning. *Journal of Internet Services and Applications*, 10(1):1–22, 2019.

[8] Hacer Karacan and Mehmet Sevri. A novel data augmentation technique and deep learning model for web application security. *IEEE Access*, 9:150781–150797, 2021.