Literature Review

Jacobus Marthinus (Koot) Becker

Abstract

The goal of this project is to implement a heuristics-based anomaly detection algorithm to inspect JSON Web Tokens (JWT's) which extends the functionality of a Web Application Firewall (WAF) to prevent authorisation bypass [1]. What follows is a literature review of what has previously been achieved in this field.

A. Next Generation Firewall for Network Security: A Survey [2]

The authors of this paper attempt to understand what differentiates next-generation firewalls (WAF's) from traditional firewalls we are generally used to. They go into depth to see which features these firewalls use that make them desirable and what traditional firewalls lack in the modern age. These authors list commonly occurring cyber attacks and how each firewall deals with each attack. The authors also compare the advantages and disadvantages of each firewall.

The performance benchmarks used by the authors are Advanced Evasion Techniques, Targeted Cyber-Attack, Web Application Attacks, and Data Focused Attacks. The authors also compare the differences between the two firewalls in the following areas: Preventing Advanced Persistent Attacks, Inspecting SSL Traffic, Controlling Web Applications, Managing Users & Use Policy and Trade off Security vs Performance.

Overall this paper does not go into too much technical detail about what WAF's do and how they operate, but the authors make the point that it is necessary because of its advantages over traditional firewalls. Finally, the authors look at an example of the Palo Alto Next-Generation Firewall and how it manages to extend its functionality over traditional firewalls.

B. Critical Analysis on Web Application Firewall Solutions [3]

This paper compares existing WAF solutions to see which gaps exist in which solutions. The authors make the argument that certain solutions have certain features that make them a better fit for certain scenarios. Solutions that are being compared include Mod Security, Imperva's Secure Sphere, Barracuda network application gateway etc.

The paper goes into some technical detail on how and where WAF's operate. WAF's work on OSI layer 7, the application layer by deeply inspecting the HTTP packet and directing and analyzing it for malicious strings and configuration error problems. The authors compare all the options using metrics like Time efficiency, how well its organised, Effectiveness, if it has Monitoring, Blinking, Response Filtering, Attack Prevention, Web Site Cloaking, Authentication and Web SSO, Deep Inspection, Session Protection and finally evaluate the Overall Security Performance.

The researchers also compare the management interface of the WAF applications using the following metrics: Web Based, Command Line, Desktop Based, Ease-of-use, Comprehensiveness and Flexibility. The authors make a final observation that WAF's have become essential for each website and can become highly customized, but depending on your business needs a best fit solution can be found.

C. Fast Pattern Matching in Compressed Data Packages [4]

The authors of this paper explore Fast Pattern Matching which can significantly speed up Firewalls and intrusion prevention systems by leveraging deep packet inspection of compressed data packages in high-speed systems. The authors attempt to leverage hardware Huffman encoding by only decompressing data once and then filtering similar packets since the system does not need to decompress packets with the same source and destination address as the first decompressed packet.

The authors explain that this is also an important step in ad filtering since keyword searches for ad-related words in the decompressed packages can also be searched for. According to the authors, this method has enormous speed increases compared to pure software encoding.

D. A Fully Automated Deep Packet Inspection Verification System with Machine Learning [5]

The authors of this paper created a deep packet inspection (DPI) verification system for growing network density and Quality of Service. The authors use mobile automation tools like Monkey Runner, UI Automator, Monkey Talk, Appium and GUITAR to simulate application web traffic. This is easier than generating application web traffic using a mobile web browser since not a lot of tools for this exist.

The authors explore doing Automated Application Signature Verification for the HTTP Packets. The solution works by running a command-line packet sniffer and network protocol analyzer to see all the HTTP traffic. From this an include/exclude list which contains specific keywords that are either blocked or allowed. The packets are inspected and allowed or blocked based on the specific keyword. This is done differently for HTTP, SSL, TCP and UDP.

The proposed script generates a list of relevant flows based on the SSL, TCP or UDP flow size and rate. The verification system detects the flow rates and flags any suspicious flow rates. The final result is a fully automated signature verification system of the flow rates.

E. A survey of network anomaly detection techniques [6]

This paper proposes an in-depth analysis of classification, statistics, information theory and clustering in anomaly detection techniques. The authors describe the generic framework for network anomaly detection and how it is utilized in most solutions. They also indicate that security incidents had exponential growth in 2009-2014 and anomaly detection has become more important.

The authors describe types of anomalies like point anomalies, contextual anomalies and collective anomalies, how these different anomalies can be detected and the output of different anomaly detection techniques. The authors have a heuristic score where if the anomaly score is above a certain threshold it gets flagged. The types of network attacks like Denial of Service (DoS), Probe, User to Root (U2R) and Remote to User (R2U) are described and how these attacks can be measured by heuristics and how sophisticated the attacks can get.

The paper then follows by describing classification-based techniques, statistical anomaly detection techniques, Information theory and cluster-based techniques and their respective complexities. In conclusion, depending on the attack preference, different techniques can be used to detect the anomalies.

F. Deep learning methods in network intrusion detection: A survey and an objective comparison [7]

This paper investigates the efficiency of different deep learning models on different internet traffic datasets. The paper lists supervised instance learning, supervised sequence learning, semi-supervised instance learning and other learning paradigms as deep learning models. The Internet traffic datasets are of different sizes and are real-world data from web servers with the datasets varying in size from 100 000 instances to 16 million instances.

The weighted macro average of the metrics is taken as the heuristic of the neural network and the results are based on the accuracy of the neural network. The authors explain how they do pre-processing on the data and hyper-parameter configuration. The authors list using legacy datasets as one of the several weaknesses, as well as inadequate details of models leading to inaccurate results that cannot be reproduced.

The results show that deep feed-forward neural networks perform the best overall datasets and are perfect for WAF's that have a large number of packets incoming since the ANN gets more efficient on more data. Since this is application-specific, the ANN does not do well on small web servers.

I. Core Problems

A. Primary Problems

1) **JWT Inspection:**
   The product needs to parse incoming JWT tokens to extract claims and verify their integrity and authenticity, validate JWT signatures to ensure they haven't been tampered with and check the expiration time, issuer, and audience of the JWT token to prevent token replay attacks. It also needs to monitor JWT usage patterns to detect abnormal JWT behaviour, such as excessive token creation or usage from unexpected locations.
2) **Heuristic-Based Anomaly Detection:**
   The product needs heuristics to identify abnormal patterns in incoming HTTP requests which may include request rate, payload size, header analysis, HTTP method, user agent analysis, etc. It also needs to implement algorithms to detect anomalies based on these heuristics. Machine learning algorithms such as clustering, decision trees, or neural networks can be used for anomaly detection.
3) **Integration with Web Application Firewall:**
   The product needs ensure the WAF can intercept incoming HTTP requests before they reach the application server and implement rules to block or allow requests based on the results of anomaly detection and JWT inspection.

B. Secondary Problems

1) **Continuous Monitoring and Updates:** Continuously monitor the performance of the WAF and adjust heuristics and detection algorithms as needed.
2) **Logging and Reporting:** Log all WAF actions, including blocked requests, allowed requests, and detected anomalies.
3) **Testing and Deployment:** Thoroughly test the WAF in a staging environment, monitoring its performance and effectiveness in real-world scenarios.
4) **Documentation and Training:** Document the WAF's configuration, rules, and operational procedures to train administrators and developers on how to use and configure the WAF effectively.

References

[1] G. Patil, "Jwt authentication bypass," 2022. [Online]. Available: https://redfoxsec.com/blog/jwt-authentication-bypass/

[2] K. Neupane, R. Haddad, and L. Chen, "Next generation firewall for network security: A survey," in *SoutheastCon 2018*, 2018, pp. 1–6.

[3] A. Razzaq, A. Hur, S. Shahbaz, M. Masood, and H. F. Ahmad, "Critical analysis on web application firewall solutions," in *2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS)*, 2013, pp. 1–6.

[4] M. S. Berger and B. B. Mortensen, "Fast pattern matching in compressed data packages," in *2010 IEEE Globecom Workshops*, 2010, pp. 1591–1595.

[5] U. Trivedi and M. Patel, "A fully automated deep packet inspection verification system with machine learning," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2016, pp. 1–6.

[6] M. Ahmed, A. Naser Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804515002891

[7] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *Journal of Network and Computer Applications*, vol. 169, p. 102767, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520302411