IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DETECTING WEB ATTACKS WITH END TO END DEEP LEARNING

Mrs A. Deepika [1] , Vuppala Harika [2], Venteru Mounica [3], Mannala Kruthika [4]

[1]Assistant Professor Department of CSE(AIML), CMR College of Engineering and Technology,
Hyderabad, Telangana, India.

[2, 3, 4] Student, Department of CSE (AIML), CMR College of Engineering and Technology,
Hyderabad, Telangana, India.

ABSTRACT— Due to their network accessibility and frequent weaknesses, web applications are common targets for cyber-attacks. When a hacking attempt is discovered, a system for intrusion detection monitors online applications and sends out alerts. Intrusion detection systems that are now in use often take features from input strings or network packets that are carefully chosen to be pertinent to attack analysis. However, picking characteristics manually takes a lot of time and requires in-depth understanding of the security area. Furthermore, in order for supervised learning algorithms to differentiate between normal and aberrant behaviours, they require a significant amount of labelled legitimate or attack request data, which is frequently prohibitively expensive for use in production web services. This work adds three new insights to the field on autonomic intrusion detection systems research. Using the Robust Software Modelling Tool (RSMT), which autonomously monitors and characterises the runtime behaviour of web applications, as a basis, we first assess the viability of an unsupervised/semi-supervised strategy for web attack detection. In the second section, we walk through how RSMT trains a stacked blurring auto encoder to encode as well as reconstruct a call graph for end-to-end deep learning. The reconstruction error of the request data is computed to identify anomalies using a low-dimensional representation of the feature set with unlabelled sample data. The findings of our empirical testing of RSMT on purposefully vulnerable production applications as well as artificial datasets are examined in our third section. According to our findings, the suggested method requires little domain expertise and a little amount of labelled training data in order to effectively and accurately detect attacks such as SQL injection, cross-site scripting, and deserialization.

Key Words: Deep Learning, RSTM , SQL

INTRODUCTION

Web apps are appealing targets for online criminals.Common attacks which may disable web services, steal confidential user data, and result in large financial losses for both service providers and consumers include SQL injection, cross-site scripting (XSS), and remote code execution. It is difficult to defend online applications from attacks. Web attacks still pose a significant risk, despite the fact that developers and researchers have created numerous countermeasures and defensive programming best practices) to safeguard web applications. Researchers, for instance, discovered that, in a 2015–2016 scan, more than half of the online applications had serious security flaws like SQL Injection or XSS. Additionally, the typical American business loses $15.4 million annually due to hacking attacks. Sensitive personal data belonging to approximately 143 million American consumers was compromised in the 2017 Equifax data breach, which was caused by an Apache Struts vulnerability. Despite the vulnerability being discovered and fixed in March 2017, Equifax did not respond for another four months, resulting in a projected insured loss of one hundred and twenty-five million dollars. For a variety of reasons, traditional systems for intrusion detection do not perform as effectively as anticipated.

RELATED WORK

A classification of SQL-injection attacks and countermeasures:

Because SQL injection attacks provide attackers free access to the databases that underpin Web applications and the potentially sensitive data they contain, they constitute a severe security risk to these systems. While a number of alternatives have been presented by researchers and practitioners to address the SQL injection issue, the current options either do not fully address the problem or have drawbacks that hinder their adoption and implementation. Many academics and practitioners were familiar with only a subset of the vast range of strategies accessible to attackers who are seeking ways to take advantage of the SQL injection vulnerabilities. Consequently, many of the solutions put forth in the literature only deal with a portion of the problems associated with SQL injection. We provide a thorough analysis of the various SQL injection attack types that are currently known in order to solve this issue. We offer explanations and illustrations of possible attack methods for each kind of attack. Our analysis and presentation of current methods for detecting and preventing SQL injection attacks also include this. We go over each technique's advantages and disadvantages for dealing with SQL injection attacks in general.

Static detection of cross-site scripting vulnerabilities:

Many of what we do every day are supported by web applications, yet despite their widespread use and ease of exploitation, they frequently include security flaws. An attacker uses cross-site scripting (XSS) to execute an injected script on a browser with the privileges of the server by taking advantage of the confidence that a Web client (browser) have for a trusted server. XSS was the most common class of new reported vulnerabilities in 2006, and it continues to be the most common class of assaults. Web applications are vulnerable to cross-site scripting (XSS) attacks because the verification that they undertake on input from untrusted sources is insufficient to stop that input from triggering the JavaScript interpreter of a browser. This validation is especially challenging to do correctly if the input must include some HTML mark-up. Since most methods already in use to identify XSS vulnerabilities generally taint-based and presume that input validation mechanisms are sufficient, they either fail to identify actual vulnerabilities or produce a large number of false positives. This study directly targets inadequate or non-existent input validation by presenting a static analysis for XSS vulnerability detection. Our method combines string analysis with studies on contaminated information flow. The numerous ways to call the JavaScript interpreter make proper input validation challenging. We encounter the same challenge when checking over vulnerabilities statically, but we resolve it by formalising an approach based on a W3C recommendation, a Firefox source code, and online guides to closed-source browsers. Based on our policy, we provide efficient checking algorithms. We put our method into practice and offer a thorough analysis that identifies known and undisclosed vulnerabilities in real-world web apps.

Defensive programming: It is Using an annotation toolkit to build dos-resistant software:

This paper describes a toolbox to help strengthen code resilience against denial-of-service attacks. It is our observation that programmers prioritise functionality above all else while creating software. Many times, intrusion detection systems, firewalls, and the operating system are blamed for defending code against attacks. Consequently, many Do’s vulnerabilities remain undiscovered until an attack occurs and significant harm is inflicted upon the system. This paper argues that making software defensive by periodically adding protection measures into the code itself is a superior approach than responding to assaults after the fact. We offer an API in our toolkit so that programmers can annotate their code. During execution, these annotations function as both sensors and actuators, keeping an eye out for instances of resource misuse and responding appropriately if such instances are identified. The toolkit's design, implementation, and efficacy assessment using three widely-used network services are presented in this study.

Effects of cyber security knowledge on attack detection:

It takes cognitive skills to identify potential threats from vast volumes of network data, and maintaining cyber security was a challenging undertaking that depends on domain knowledge. The purpose of this study is to determine how information security and network operations expertise affect the ability to identify breaches in a basic network. In order to investigate how people, regardless of experience with cyber security, identify malicious activity and proclaim an assault based on a series of network events, we created a rudimentary version of the Intrusion Detection System (IDS). Our findings show that increased cyber security expertise reduced the incorrect categorization of innocuous events as malicious and enhanced the accurate identification of malicious events. Nonetheless, information was less useful in determining if a series of occurrences represented a cyber-attack. Although awareness of cyber security aids in the identification of malicious activities, precise detection judgements require contextual knowledge of the particular network in question. Individuals with cyber security expertise were able to differentiate between various cyber-attack kinds based on their responses, while individuals without any experience had difficulty identifying the various sorts of attacks. We describe the relationship between these results and cognitive processes, as well as the implications for enhancing cyber security.

METHODOLOGY:

Upload RSMT Traces Dataset:

Upload RSMT Traces Dataset is the first module of our project and it is used to upload attack traces.

Generate Train & Test Model:

Generate Train & Test Model module is used to generate train and test data. All deep learning algorithms will take 80% dataset as training and 20% dataset for testing.

Run SVM Algorithm:

Run SVM Algorithm module is used to generate SVM model on train data and calculate precision, recall on test data.

Run Naive Bayes algorithm:

Run Naive Bayes Algorithm module is used to generate Naive Bayes model on train data and calculate precision, recall on test data.

Run Propose Auto encoder Deep Learning Algorithm:

Run Propose Auto Encoder module to run propose algorithm, Auto Encoder got 90% accuracy.

Run Extension LSTM Algorithm:

Run Extension LSTM Algorithm module is used to run LSTM and we got values for LSTM algorithm.

Precision comparison Graph:

Precision Comparison graph x-axis represents algorithm name and y-axis represents precision value. In all algorithm propose Auto Encoder showing good performance.

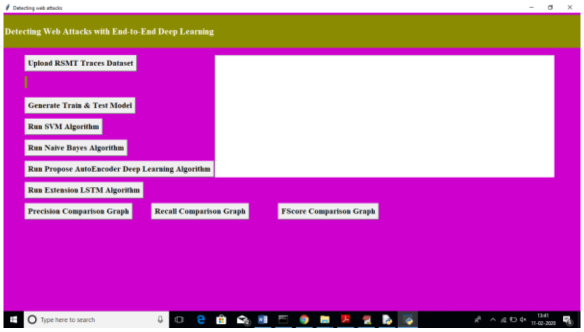
Recall Comparison Graph:

Recall Comparison graph x-axis represents algorithm name and y-axis represents recall value. In all algorithm Extension LSTM showing good performance.

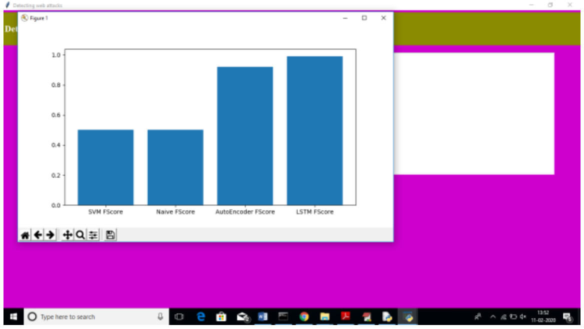
Score Comparison Graph:

Score comparison graph x-axis represents algorithm name and y-axis represents Score value. In all algorithm Extension LSTM showing good performance.

RESULT AND DISCUSSION:


Detecting web attacks
Detecting Web Attacks with End-to-End Deep Learning
Upload RSMT Traces Dataset
Generate Train & Test Model
Run SVM Algorithm
Run Naive Bayes Algorithm
Run Propose AutoEncoder Deep Learning Algorithm
Run Extension LSTM Algorithm
Precision Comparison Graph
Recall Comparison Graph
FScore Comparison Graph
Type here to search
13:41
11-02-2020

In above screen click on 'Upload RSMT Traces Dataset' button to upload attack traces



In above graph x-axis represents algorithm name and y-axis represents Score value. In all algorithm Extension LSTM showing good performance

CONCLUSION:

Through our research, we have proven that reinforcement learning (RL) can be a useful method for solving the difficult dual-objective problem of steering a vessel to follow a predetermined route with known waypoints while avoiding obstacles without using a map. We have demonstrated, more precisely, that in the presence of immobile impediments obstructing and encircling the intended path, the most advanced PPO algorithm merges to a strategy that produces intelligent guided behaviours. Several innovative concepts, such as the Feasibility Pooling method for intelligent real-time sensing package dimensionality reduction, have to be designed and implemented in order to engineer the agent's observation vector and reward function. Through experiments, we have shown that the agent can modify its guidance strategy (i.e., prefer path-adherence over collision avoidance) according to the value fed by the observation vector by augmenting it with the incentive trade-off parameter, allowing the representative to adapt to alterations in its reward function. After doing comprehensive testing, we have found that the agent's rate of success is in the mid-80s when it is configured to cause a tight path adherence bias and nearly 100% when an additional defensive approach is selected—even in difficult test scenarios with high obstacle densities. Finally, it is crucial to acknowledge that deep neural networks—which learn enormous amounts of trained parameters—are a major component of DRL algorithms. These networks are currently uninterrupted by humans, which is a barrier to the algorithms' widespread adoption in safety-critical applications. Nonetheless, the present research showcases the potential for integrating intelligence into these vital safety applications.

REFERENCES

1.Hal fond WG, Viegas J, Orso A. A classification of SQL-injection attacks and countermeasures. In: Proceedings of the IEEE International Symposium on Secure Software Engineering. IEEE; 2006. p. 13–5. 2. Wassermann G, Su Z. Static detection of cross-site scripting vulnerabilities. In: Proceedings of the 30th International Conference on Software Engineering. ACM; 2008. p. 171–80.

3.Di Pietro R, Mancini LV. Intrusion Detection Systems vol. 38: Springer; 2008.

4.Qie X, Pang R, Peterson L. Defensive programming: Using an annotation toolkit to build dos-resistant software. ACM SIGOPS Oper Sits Rev. 2002;36(SI):45–60.

5.https://doi.org/https://www.acunetix.com/acunetix-web-applicationvulnerability-report-2016. Accessed 16 Aug 2017.

6.https://doi.org/http://money.cnn.com/2015/10/08/technology/cybercrime-cost-business/index.html. Accessed 16 Aug 2017.

7.https://doi.org/https://www.consumer.ftc.gov/blog/2017/09/equifaxdata-breach-what-do. Accessed 16-August-2017.

8.https://doi.org/https://theconversation.com/why-dont-big-companieskeep-their-computer-systems-up-to-date-84250. Accessed 16 Aug 2017.

9. Ben-Asher N, Gonzalez C. Effects of cyber security knowledge on attack detection. Compute Hum Behave. 2015; 48:51–61.

10.Japonica N, Stephen S. The class imbalance problem: A systematic study. Intel Data Anal. 2002;6(5):429–49.

11.Liu G, Yi Z, Yang S. A hierarchical intrusion detection model based on the pica neural networks. Neurocomputing. 2007;70(7):1561–8.

12.Xu X, Wang X. An adaptive network intrusion detection method based on pica and support vector machines. Advanced Data Mining and Applications. 2005; 3584:696–703.

13.Pietraszek T. Using adaptive alert classification to reduce false positives in intrusion detection. In: Recent Advances in Intrusion Detection. Springer; 2004. p. 102–24.

14.Goodfellow I, Bengio Y, Courville A. Deep Learning: MIT press; 2016.

15.Hrushevsky A, Subsieve I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2012. p. 1097–105.

16.Amodei D, Anantha Narayanan S, Manubhai R, Bai J, Battenberg E, Case C, Casper J, Catanzaro B, Cheng Q, Chen G, et al. Deep speech 2: End-to-end speech recognition in English and mandarin. In:

International Conference on Machine Learning. New York: PMLR; 2016. p. 173–82.

17.Subsieve I, Vinyl's O, Le QV. Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2014. p. 3104–12.

18.Sun F, Zhang P, White J, Schmidt D, Staples J, Krause L. A feasibility study of autonomically detecting in-process cyber-attacks. In: Cybernetics (CYBCON), 2017 3rd IEEE International Conference On. IEEE; 2017. p. 1–8.