Chapter 8

Conclusion

8.1 Concluding Remarks

The Web is an integral part of our day-to-day lives. Everyday activities be it social or financial make use of the Web. With the advent of technology, Web applications and sites are now able to provide essential services to its users. In doing so, the users may share some sensitive personal information with the applications. From the application's perspective, the service that it provides should always be reliable, secure and available to its user base. On the other hand, from the user's perspective, sensitive information shared should be protected at all costs and the services accessible when needed. However, all day everyday might not be as smooth as anticipated. This is because there exist a section of ill-intending users who try to disrupt the normal functioning of applications by exploiting existing vulnerabilities in applications and protocols. Such nefarious users may be motivated on various grounds such as financial, political, and espionage. Interestingly, the ultimate aim may be to disrupt the ongoing services of the application, or gain unauthorized access to the systems, or steal sensitive personal information of users. Consequences of aforementioned dishonorable activities at times may be severe and as such detecting Web-based attacks with high accuracy is of paramount importance. The Web-based attacks considered in this thesis are: Cross-site Scripting attacks, HTTP Flooding attacks and Attacks in Critical Infrastructure. Precisely detecting these attacks in real time still remains a challenging task.

This thesis presents methods to detect the selected Web-based attacks using machine learning techniques. Specifically, the aim in developing such detection methods is four fold: i) Detection with high accuracy, ii) Cost-effective detection,

iii) Detection with minimum false alarms and iv) Detection in near real-time. Three noteworthy tasks to be mentioned when developing such detection methods are:

1. Determine the features or characteristics which help distinguish normal activity from malicious activity,
2. Out of all the features determined, identify the most informative ones. In other words, identify the most relevant features.
3. Determine an optimal subset of relevant features to detect an attack.

The three tasks mentioned above can be accomplished by a preprocessing task in machine learning known as feature selection. Feature selection inherently chooses a subset of features based on some criteria which is then given to a learning model. Significance of feature selection techniques have already been established for detection of Web-based attacks. It is imperative here to note that, testing the effectiveness of a detection method necessitates the use of datasets.

Initially, a dataset preparation pipeline for XSS attack is presented. It emphasizes how features can be extracted from raw scripts and URLs which aid in differentiating malicious and benign instances. To understand the ensemble learning methods applicable in security datasets, 16 datasets have been analyzed with four ensemble learning methods namely, Bagging, Boosting, Bagging-Boosting and Stacking. Next the thesis presents, three feature selection methods namely, MICC-UD. INFS-MICC and FSRA to detect the selected Web-based attacks. The following enumerated points summarize the work done along with a few observations based on the theoretical and experimental studies conducted in this evolving field of research.

1. Efficacy of a learning model can be tested with the help of a dataset. In Chapter 3, along with discussing some of the benchmark datasets, a dataset preparation pipeline is also proposed which outputs the dataset named XSSD. XSSD is a cross-site scripting attack dataset containing both attack and benign samples. To the best of our knowledge, XSSD is the first publicly available feature dataset for XSS attacks.
2. In Chapter 5, a traditional feature selection method named MICC-UD is proposed to detect XSS attacks. This method selects a subset of highly ranked features by considering feature-class mutual information for identifying relevant features and feature-feature correlation for determining irredundant

features. Effectiveness of the method is established in terms of high accuracy, precision and recall on the proposed XSS attack dataset as well as several other benchmark security datasets.

3. An incremental feature selection method named INFS-MICC is proposed in Chapter 6 to detect HTTP Flooding attacks in the application layer. The main highlight of INFS-MICC is that it selects a subset of highly relevant and irredundant features incrementally which avoids re-computation of the whole dataset from scratch. This cost-effective nature of the proposed method is useful for detecting HTTP-Flooding attacks in real-time. Effectiveness of the method is established with three benchmark datasets and five ensemble learners.

4. The primary idea behind ensemble learning is that decision given by a group of learners is better than the decision given by an individual learner. An ensemble learning method called FSRA is proposed in Chapter 7 for the detection of attacks in critical infrastructures. This method takes into account three benchmark ranker algorithms and combines their individual ranks with the help of the proposed score. From comparative analysis based on three datasets, it is seen that FSRA performs better than the considered base feature selection methods in terms of f1-score.

5. To conduct the experimental studies in Chapter 5,6 and 7, five well-known ensemble learners are considered. The learners are configured with 10-fold cross validation and a recursive elimination setting to find the optimal number of features. Results and analysis show that Extreme Gradient Boosting (XGB), Gradient Boosting (GB) collectively perform better compared to the other learners.

8.2 Future Work

With evolving trends and sophisticated techniques, new attack vectors to launch Web-based attacks are also emerging on a regular basis. As future work, the following issues are aimed to address to strengthen the defense mechanisms of Web-based attacks.

1. The proposed method in Chapter 5 can be further strengthened to handle

multi-vector attacks. These attacks simultaneously try to overwhelm the system through multiple sophisticated attack methods. In other words, several lines of different attacks are generated to attack the same target. For instance, an attacker may try to carry out different types of DDoS attacks consecutively on the same target. Such attacks may also be launched from different layers of the OSI model. For example, a TCP-SYN Flooding attack followed by an HTTP Flooding attack. Such attacks require high levels of co-ordination among the compromised entities but if performed flawlessly can be far more damaging than single vector attacks [1]. From the victim's perspective on the other hand, such attacks are challenging to detect because-

   (a) Victim may not be aware that attacks are launched using multiple vectors, and

   (b) Even if the multiple vectors are identified, all may not be successfully mitigated.

2. Vulnerable Web applications are common victims to zero-day attacks as attackers exploit the vulnerability even before the developers fix it. Even though such attacks are difficult to detect using traditional techniques, sophisticated analysis mechanisms that includes behavioral analysis and unsupervised anomaly detection techniques could be explored to counter such attacks.

3. In recent times, attacks have evolved to such an extent that even machine learning models are thwarted. Compromising a model prohibits it from successfully defending against an attack. Such attacks are called adversarial attacks and key areas include appropriate adversarial training instance generation, robust feature extraction, feature selection and robust model building.

4. The computational complexity of the proposed methods in Chapter 5 and 6 could be reduced with the help of CPU-GPU implementations. Such implementations can handle attack detection in real time and as such are vital for any defense system.

5. A Web tool needs to be developed which could automatically streamline several stages in the attack detection framework. Stages such as feature extraction, feature selection, analyzing the most important attack features and

[1] `https://www.checkpoint.com/cyber-hub/cyber-security/what-is-cyber-attack/what-is-a-multi-vector-attack/`

model building can be incorporated. The reports generated by such a tool can then be shared with the security community.

6. Incorporation of Explainable AI is another future direction. Such techniques increase model interpretability because of which the reasons behind the decisions given by models are better understood. Subsequently, it helps building a trustworthy detection system.

7. All the frameworks proposed in the thesis are supervised learning techniques. However, unsupervised learning approaches can also be explored to enhance web-based attack detection capabilities by making use of large amounts of unlabeled data, generating useful representations and identifying hidden structures in the data to reveal novel patterns.