A Majority Voting Technique for Wireless Intrusion Detection Systems

Bandar Alotaibi, Advisor: Prof. Khaled Elleithy
Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT.

Abstract

This poster aims to build a misuse Wireless Local Area Network Intrusion Detection System (WIDS), and to discover some important fields in WLAN MAC-layer frame to differentiate the attackers from the legitimate devices. We tested several machine-learning algorithms, and found some promising ones to improve the accuracy and computation time on a public dataset. The Bagging classifier and our customized voting technique have good results (about 96.25% and 96.32% respectively) when tested on all the features.

Introduction

Wireless networks have dominated in recent years over the wired networks that have been dominant for decades. Nowadays, Wireless Local Area Networks (WLANs) are the first choice for local area connectivity because of the mobility and the low cost that they provide. Unfortunately, the mobility and the low cost do not come free; it comes with debatable security.

There are a wide range of security measures in use, such as encryption mechanisms, authentication methods, and access control techniques, but many intrusions remain undetected. Thus, there is a demand to automate the monitoring of WLAN activities to detect intrusions. There are two known Intrusion Detection methods: anomaly detection and misuse detection. Anomaly detection identifies attacks through deviation from the normal behavior, by the devices that generate these attacks. Misuse detection recognizes suspicious activities regarding patterns matching previous known attacks.

Research Problem

WLANs are exposed to several attacks because of the shared medium that wireless devices utilize to communicate with one another. WLANs attacks can be classified as:

Injection Attacks flood the wireless network with encrypted data frames smaller in size than the normal frame. ARP injection attack is an attack in which the attacker launches to speed up the process of collecting Initialization Vectors (IVs) from the targeted wireless device or AP.

Flooding Attacks usually generate an increase in the number of frames in a WLAN-management frames in particular. Some examples are de-authentication attack and authentication flooding attack.

Impersonation Attacks masquerade a legitimate device in a WLAN by changing one or more of its characteristics. The Evil-twin AP is one example, where the attacker can change the MAC address and Service Set Identifier (SSID) of the device to be the same as the MAC address and SSID of an existing AP.

Misuse Detection Framework

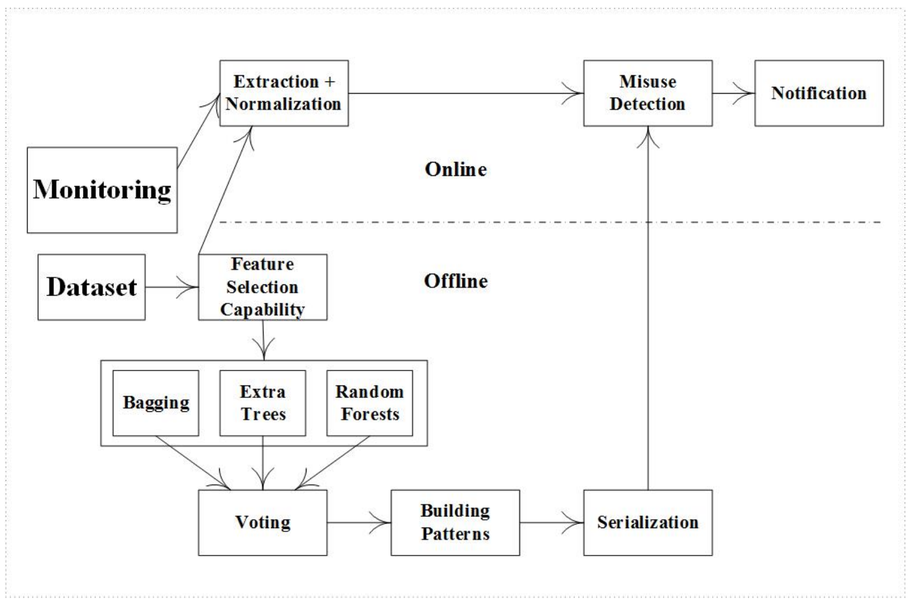


Figure 1: The proposed framework

Implementation and Test Plan

The only public data-set that we know for WLANs is Aegean Wi-Fi Intrusion Dataset (AWID) published by Kolias et al. [1] in 2015. The data-set consists of four classes and fifteen classes, respectively. The four classes are categories that launched attacks belong to, including flooding, injection, and impersonation, and the normal class, while the other reduced data-set consists of the names of the launched attacks and the normal class. The number of training samples of each reduced data-set is 1,795,575, and the number of test samples is 575,643. The number of features is 156, representing the WLAN frame fields along with physical layer meta-data.

Results and Discussion

The best machine learning algorithms that we used in our experiments are Decision Trees [2], Extra Trees [3], and Random Forests [4]. Decision Trees is not stable. We ran the test several times and it gave us different results every time. The three classifiers did not achieve better results than the J48 classifier that the Kolias et al. used in their experiments. We decided to use the Bagging classifier [5] of minimum Decision Trees as a base estimator to be more robust and to have minimum time. The Bagging classifier yields slightly better results and has better timing. We then used the voting classifier that utilized Extra Trees of 20 trees, Random Forests of 20 trees, and the Bagging classifier of 10 Decision Trees as base estimator, and got better results and reduced time.

Detection Accuracy and Time

Table II : Using all the features

	Accuracy	Time
Extra Trees	96.06	18.1
Random Forests	95.89	22.4
Bagging	96.25	154
Our method	**96.32**	390

Table III: Using 20 features (our Features)

	Accuracy	Time
Extra Trees	96.31	8.03
Random Forests	96.31	9.95
Bagging	96.25	35.7
Our Method	**96.32**	107

Bagging

Table I: Bagging Confusion Matrix

Normal	Flooding	Injection	Impersonation	Classified as
530383	343	0	59	Normal
2585	5512	0	0	Flooding
2	0	**16680**	0	Injection
18606	2	0	**1471**	Impersonation

Random Forests

Table IV : Random Forests Confusion Matrix

Normal	Flooding	Injection	Impersonation	Classified as
530775	6	0	4	Normal
2536	**5561**	0	0	Flooding
41	0	16641	0	Injection
18645	0	0	1434	Impersonation

Extra Trees

Table IV : Extra Trees Confusion Matrix

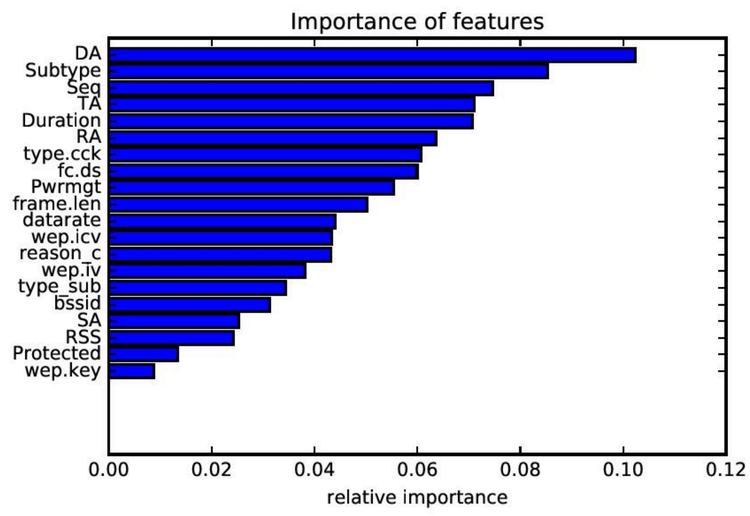
Normal	Flooding	Injection	Impersonation	Classified as
530773	2	0	10	Normal
2601	5496	0	0	Flooding
2	0	16680	0	Injection
18619	0	0	1460	Impersonation

Our method

Table IV : Our method Confusion Matrix

Normal	Flooding	Injection	Impersonation	Classified as
530778	0	0	0	Normal
2589	5508	0	0	Flooding
5	0	16677	0	Injection
18609	0	0	1470	Impersonation

Most important 20 features



Conclusion

We improved the accuracy and the time on the AWID data-set using a classifier that votes on the output of the carefully picked three classifiers: Extra Trees, Random Forests, and Bagging with ten Decision Trees as base estimators. This performs well in both accuracy and time. The best performing classifier is the voting classifier which improved accuracy and time to 96.32% and 390 seconds when we used all the features. We also used a data mining technique to choose the best 20 features to decrease time and improve accuracy of the best performing classifiers. We maintain the same accuracy, but improved the time by about 107 seconds.

References

[1] C. Kolias, G. Kambourakis, A. Stavrou and S. Gritzalis, "Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset," in *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 184-208, Firstquarter 2016.
[2] Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). Classification and regression trees. CRC press.
[3] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine learning, 63(1), 3-42.
[4] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.
[5] Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.