A Comparison Between Signature and Machine Learning Based Detectors

Adetokunbo Makanju, Patrick LaRoche, A. Nur Zincir-Heywood
Faculty of Computer Science
Dalhousie University
Halifax, Nova Scotia
B3H 1W5
Canada
makanju@cs.dal.ca,plaroche@cs.dal.ca,zincir@cs.dal.ca

***Abstract*— Data Link layer attacks on WiFi networks are known to be one of the weakest points of WiFi networks. While these attacks are very simple in implementation, their effect on WiFi networks can be devastating. To this end, several Intrusion Detection Systems (IDS) have been employed to detect these attacks. In this paper, we compare the ability of Snort-Wireless and a genetic programming (GP) based intrusion detector, in the detection of a particular data link layer attack, namely the deauthentication attack. We focus particularly on a scenario where the attacker stealthily injects the attack frames into the target network. Results show that the GP based detection system is much more robust against the different versions of the attack compared to Snort-Wireless and can achieve a detection rate in average 100% and a false positive rate in average 0.1%.**

I. Introduction

The wireless network protocol IEEE 802.11, also referred to as Wireless Fidelity (WiFi), is a protocol which has been deployed in a growing number of locations and environments. The security vulnerabilities of networks based on the IEEE 802.11 wireless network standard have been widely attested in literature [1]. These security vulnerabilities are not necessarily peculiar to WiFi networks but to all wireless communication protocols. Data transmission through open air waves is a characteristic of all wireless communication protocols, where this fact is responsible for their seeming openness to intrusions. Particular emphasis is however placed on WiFi networks due to the pervasiveness of their deployment. IEEE 802.11 is by far the most widely used wireless networking standard in the world today and its popularity increases by the day.

Wireless Data Link layer attacks target the lower layers of the Open System Interconnect (OSI) protocol stack and their goal is to render the network unusable. They are therefore of particular importance in any discussion on the vulnerabilities of WiFi networks. Apart from being designed specifically for WiFi networks, most of the security features incorporated into the WiFi protocol such as data encryption and client authentication are not able to guard against these attacks. The deauthentication attack which is investigated in this work is a data link layer attack, which causes a Denial-of-Service(DoS)by injecting a subset of the IEEE 802.11 Management Frames i.e. the deauthentication frame into the network traffic.

In this paper, we compare the detection capabilities of signature based IDSs against a machine learning, namely genetic programming (GP), based detection solution. Snort-Wireless, is selected for this work as a signature based detector since it is an open source solution, which is widely used. It detects the deauthentication attack through the measurement of certain metrics, whose values are provided by the network administrator. The problem with using such a method arises in a scenario where an attacker injects attack frames into the network in a controlled and stealthy manner in order to beat the signatures in the intrusion detection database. Based on the past success in the use of Genetic Programming based solutions in detecting the deauthentication attack [2], our objective is to test the suitability and robustness of such an approach in detecting the deauthentication attack when signature based solutions like Snort-Wireless fail.

The remainder of this paper is organised as follows. Section 2 discusses WiFi networks in detail and how data link layer attacks affect them. Section 3 discusses the methods of detecting data link layer attacks investigated. Section 4 outlines the experiments and explains our approach. Section 5 presents the results and conclusions are given in Section 6.

II. Data Link Layer Attacks and WiFi Networks

A. WiFi Networks

WiFi networks generally consist of one or more Access Points (APs) and a number of clients, which can be any device from laptop computers to wireless Personal Digital Assistants(PDAs), which communicate over a wireless medium using the IEEE 802.11 standard. Network technologies based on the IEEE 802.11 standard include 802.11b, 802.11g and so on. These technologies differ from each other, amongst other things, by the frequency at which they operate and the bandwidth that they are able to deliver. In this paper, we deal specifically with 802.11b networks [3].

WiFi APs act as base stations or servers for wireless Local Area Networks (WLANs). Using Beacon Frames, they periodically broadcast their Service Set Identifier (SSID), a character string, which identifies the AP. This way, any authorised client

machine that is within the range of the AP and that can pick up the SSID signal can choose to join the network of the AP.

WiFi networks have many advantages, one of which is their ease of deployment. This has made WiFi technology one of the fastest growing wireless technologies to reach its consumers. WiFi technologies reached 25% of the North American population within 12 years of its release, compared to televisions and wired telephones, which took 40 and 50 years respectively to achieve the same feat [4].

However, security is of great concern in WiFi networks. WiFi networks are particularly susceptible to attacks, which their wired counterparts are not susceptible. In particular, transmitting data over open airwaves is responsible for this. Data transmitted in this fashion can easily be intercepted. Indeed, research suggests that security is the major inhibitor to the future growth of the wireless network market. Several protocols, which use authentication and cryptographic techniques like Wireless Encryption Protocol(WEP), WiFi Protected Access(WPA)and wireless Virtual Private Networks(VPN)have been proposed to ameliorate these vulnerabilities. These protocols, however, do not deal with attacks that target the physical and data link layers of the OSI protocol stack. Most of these attacks are DoS attacks, which usually exploit Media Access Control (MAC) frames, and their end effect results in the network being unusable or inaccessible to legitimate clients.

B. Data Link Layer Management Frames

The 802.11 standard defines various frame types that stations (clients and access points) use for communication, as well as for management and control of their connections [3]. This gives rise to three broad classes of frames i.e. management frames, control frames and data frames. Management frames are used by stations to establish and maintain connections. This makes them the target of most attacks, which aim to make a WiFi network unusable. Types of management frames include: Association, Disassociation, Authentication, Deauthentication, Beacon and Probe frames. Full discussion on the uses of these frames is beyond the scope of this paper, we however briefly discuss the Association, Disassociation, Authentication and Deauthentication frame subtypes below.

- **Authentication frame:** This frame is used by clients to enable an AP to identify them as legitimate stations on a WiFi network. The client sends an authentication request and the AP replies with an authentication response, which either accepts or rejects the identity of the client.
- **Deauthentication frame:** A station sends a deauthentication frame to another station if it wishes to terminate secure communications. The station can either be the client or the AP.
- **Association request frame:** This frame is used by clients to associate with an AP. When a client is associated with an AP, the AP allocates resources for and synchronizes with the client. Association frames can either be requests (from the client to the AP) or responses (from the AP to client).
- **Disassociation frame:** This is sent when a station wishes to terminate an association between itself and another station. The station can either be the client or the access point.

C. Deauthentication Attack

As mentioned earlier, this paper focuses on the Deauthentication attack. This attack, like other MAC layer attacks is very easy to implement. An attacker simply eavesdrops on a network and gathers information about the stations on the network. The attacker then uses this information to spoof the MAC address of a station on the network. If the attacker targets a specific client, it creates a deauthentication frame with the MAC address of the target as the source and the MAC address of the AP as the destination. This frame causes the AP to send a deauthentication frame back at the client (target); this prevents the target from communicating any further as a legitimate client on the network. This scenario is outlined in Fig. 1.

Apart from the scenario outlined above the attacker can vary the scope of the attack i.e. focusing on the AP to take down the entire network, targeting a specific client or group of clients, as well.

D. Void11

Void11 is a free software implementation of some common 802.11b attacks [6]. The basic implementation works in a command line Linux/Unix environment (though it has a GUI implementation called gvoid11, too). For void11 to work on a computer, the computer must have a prism based wireless Network Interface Card (NIC) and must have hostap drivers installed. The hostap drivers allow the machine to act as a wireless AP [7].

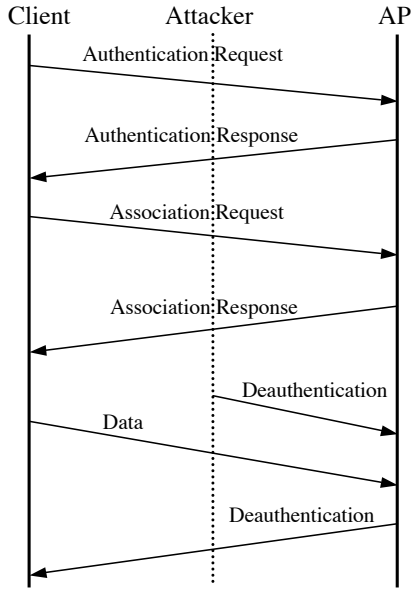

Fig. 1. Deauthentication Attack [5]

Void11 implements three data link layer attacks, which use management frames. They are Deauthenticate Flood (default mode), Authentication Flood and the Association Flood[1]. The basic goal of each of the attacks is to flood the network with management frames causing random clients to loose their connection with the AP. The end result of each of the attack types differs based on the rate of injection of the frames and on the type of client involved.

The delay (-d) switch is of particular interest to our work. This switch controls the rate at which management frames are injected into the network. The default value for the delay parameter is $10000\mu s$ [6]. Assigning a different value to this switch can be used to stealthily inject frames into the target network.

III. Detecting Data Link Layer Attacks

Traditionally Intrusion Detection Systems (IDSs)are used to detect attacks against the integrity, confidentiality and availability of computer networks [2], [8]. They are analogous to burglar alarms, which monitor the premises to find evidence of break-ins. These operations aim to catch attacks and log information about the incidents such as source and nature of the attack. An IDS can be a combination of software and hardware, which collects and analyzes data collected from a network(s) or a host(s). IDSs are generally analyzed from two aspects:

- **Deployment:** Whether to monitor incoming traffic or host information.
- **Detection:** Whether to employ the signatures of known attacks or to employ the models of normal behavior.

The use of machine learning and artificial intelligence techniques in the building of IDSs is relatively new. Hitherto, building IDSs required a human expert to construct a set of rules, which when triggered, would indicate malicious activity. In this section, we briefly discuss intrusion detection systems compared in this work i.e. Snort-Wireless and GP based IDS and how they detect the de-authentication attack. Snort-Wireless is a signature based technique, which uses rules constructed by a human expert. On the other hand, GP based detection is a machine learning based technique, which works by a data-driven approach.

A. Snort-Wireless Based Data Link Layer Attack Detection

There are several open source and commercial IDSs available in the market today but Snort stands out as being one of the most popular. Developed in 1998 by Martin Roesch, Snort is an open source, real-time intrusion detection system [9]. Using signature and anomaly based metrics it detects and prevents attacks by utilizing a rule-driven language. It is the most widely deployed open source IDS in industry and research.

With the appropriate patches applied, Snort can be transformed into Snort-Wireless [10]. These patches enable Snort (Snort-Wireless, after patches are applied) to detect WiFi specific attacks. Signatures that detect the deauthentication attack (and other WiFi MAC Layer attacks) are among the patches included in Snort-Wireless.

The most important metrics used by Snort-Wireless to detect the deauthentication attack are the number of deauthentication frames to be considered as an attack and the time frame within which that number of frames need to be detected. The default values for these parameters in Snort-Wireless are 20 frames and 60 seconds respectively[10]. While this setup can detect most attacks effectively, an attacker who injects only 19 frames in every 60 second period will go undetected with such a signature.

B. GP Based Data Link Layer Attack Detection

GP is an extension of the Genetic Algorithm (GA); which is an evolutionary computation (EC) method proposed by John H. Holland [11]. GP extends the GA to the domain of evolving complete computer programs[12]. Using the Darwinian concepts of natural selection and fitness proportional breeding, populations of programs are genetically bred to solve problems. These populations of programs can either be represented as tree like LISP structures or as binary strings, which represent integers . These integers are then mapped onto an instruction set and a set of source and destination registers. Each individual can thus be decoded into a program, which takes the form of assembly language type code for a register machine. This is known as the Linear Page Based GP (L-GP)[13]. Our work utilizes the L-GP approach, alongside the Random Subset Selection - Dynamic Subset Selection (RSS-DSS) algorithm [14], detailed below. L-GP has been used successfully by other researchers in the realm of IDSs [2], [8], [15]. Based on its successful use in detecting the deauthentication attack [2] and other higher level attacks [15], it is also employed in this work.

C. Linear Page Based GP

L-GP consists of a sequence of integers that once decoded, form the basis of a program in which the output is taken from the best performing register, as defined by the fitness function. In order to decode this linear set of instructions, each integer is mapped to a valid instruction from the defined instruction set. The instruction set consists of operands and either a source or destination register. The operands in our work are a set of register arithmetic functions, while the source and destinations are a set of valid general-purpose registers. The decoding of a sequence then creates a program that consists of simple register level transformation. Once the execution is completed, the output is taken from the best performing register. The sequences of integers are grouped in pages, each page consisting of the same number of integers (therefore the same number of instructions). The crossover operation performs a crossover on an entire page, preserving the total number of pages in an individual. The mutation operator selects one instruction with uniform probability and performs an Ex-OR operation between this and a bit-sequence

[1]The command syntax for using the void11 tool to launch an attack is: ***void11penetration -D -t[type of attack] -d[delay] -s[station MAC] -B[BSSID] [interface]***

created with uniform probability. A second crossover operator performs a swap of two instructions within the same individual (selected again with uniform probability). The page size itself, which controls the number of instructions per individual, is dynamically modified depending on the fitness level of the population. If the fitness level has not changed for a specified window, the page size is increased. This pattern will continue until a maximum page size is reached, at which point the page size is dropped back down to the initial starting page size. This entire process is continued until the GP has reached either optimal fitness, or some sort of previously set stopping criteria.

D. Random Subset Selection - Dynamic Subset Selection Algorithm

The Random Subset Selection - Dynamic Subset Selection (RSS-DSS) algorithm is a technique implemented in order to reduce the computational overhead involved with applying GPs to large data sets. To do so, the RSS-DSS algorithm utilizes a hierarchical sampling of training exemplars, dividing the problem into two levels. We present here a brief overview of how the algorithm functions for completeness, as we have implemented it in our GP, but it is not the focus of our work. The first level of RSS-DSS performs the RSS step. It divides the training set into blocks of equal size, the second level chooses (stochastically) a block and places it in memory. Level 2 performs the DSS step, as it dynamically selects a subset of the set in memory (the tournament selection). The dynamic selection is based on two metrics the GP maintains, the age of the exemplar and the apparent difficulty of the exemplar. The tournament individuals are then trained on the current subset, genetic operators are applied, and then placed back in the subset. This DSS is continued until a maximum number of DSS iterations or a stopping criteria is met, then the algorithm returns to the RSS step, selecting another block to place in memory and repeats DSS. This entire process continues until a maximum number of RSS iterations or the stopping criteria have been met. The RSS-DSS algorithm removes the requirement to train on the entire data set, instead using only a small subset of the data set that represents the more difficult or least recently encountered exemplars. This allows the GP to train more efficiently then standard techniques on big datasets.

IV. EXPERIMENTS

Our experiments require that we have appropriate datasets for the training and testing of the GP based IDS as well as for the testing of Snort-Wireless. These datasets have to be in tcpdump format for replaying through Snort-Wireless. Moreover, the tcpdump files need to be labeled for the training and testing of the GP based IDS. In order to generate such datasets, we attack a test network, outlined in Table I, using void11. All the clients are connected to an AP via 802.11 connection on channel 6. The data was collected on the monitoring machine using Kismet Wireless [16].

The deauthentication attack implemented is directed at the AP. From the attack machine, using void11, a stream of deauthentication frames with the source set to the MAC address of the AP and the target to that of the broadcast address **(ff:ff:ff:ff:ff:ff)** are intermittently released into the network stream. Normal traffic is also generated using our web crawling implementation, which is developed using the Java 2 Platform, Micro Edition (J2ME). The web crawler insures a continuous stream of web browsing requests from the clients as the network data is logged.

A. Feature Selection

The tcpdump traffic files collected by Kismet wireless could be automatically replayed through Snort-Wireless but needed further processing before they could be used on the GP based IDS. To this end, an appropriate feature set had to be selected from the features within the frames. 802.11 frames consist of several features but not all of them are related to this attack. Based on the feature selection in previous work [2],the following subset of features were selected for this purpose:

1) **Frame Control** - Defines the protocol version, type/subtype of the frame and any flags
2) **Destination Address** - MAC address of the destination of the frame
3) **Source Address** - MAC address of the source of the frame
4) **Basic Service Set Identifier (BSSID)**- Ethernet Address of the Access Point
5) **Fragment Number** - Defines the fragment number in a particular sequence of the frames
6) **Sequence Number** - Defines the sequence number of the frame
7) **Channel** - The transmission channel used for communication

B. Data Set Generation

A total of 16 different datasets are generated and employed in the following experiments. Table II details these datasets, where void11 is employed to generate attacks and Kismet wireless is employed for logging traffic.

In Table II, the datasets marked with attack type "Original" are those, which are generated by implementing the deauthentication attack with the default value of the "delay" parameter. The default setting of the "delay" parameter is $10000\mu S$ [6], after some tuning of the "delay" parameter, we were able to set the smallest value for the delay parameter at which we could continuously sustain the attack (without recourse to timing control). This value was $3,250,000\mu S$. All the datasets which

TABLE I
TEST NETWORK COMPONENTS

Type	Description
Clients	Palm Tungsten C (5x)
	HP IPAQ 4700 (3x)
	Dell Inspiron 9300 Laptop
	Macintosh Mini
AP	Airport Base Station Extreme
Monitoring Machine	Intel Based Desktop

TABLE II

DATASET CHARACTERISTICS

File	Size	Attack	Type	% Attack
1	23960	3302	Original	13.78
2	20960	3521	Original	16.80
3	7600	42	Modified	0.55
4	6800	42	Modified	0.62
5	7240	46	Modified	0.63
6	6880	43	Modified	0.62
7	7240	40	Modified	0.55
8	6440	44	Modified	0.68
9	6880	39	Modified	0.56
10	5600	48	Modified	0.85
11	6200	50	Modified	0.81
12	7200	36	Modified	0.50
13	5960	43	Modified	0.72
14	6400	42	Modified	0.66
15	5960	39	Modified	0.65
16	4960	38	Modified	0.77

TABLE III

GP PARAMETERS

Parameter	Setting
Population Size	125
Maximum Number of Pages	32
Page Size	8 Instructions
Maximum Working Page Size	8 Instructions
Crossover Probability	0.9
Mutation Probability	0.5
Swap Probability	0.9
Tournament Size	4
Number of Registers	8
Function Set	(+,-,*,/)
Terminal Set	(0,...,255) $\cup$ (*r0*,...,*r7*)
RSS Subset Size	5000
DSS Subset Size	50
RSS Iteration	1000
DSS Iteration	100

have an attack type of "Modified" are generated using this value for the delay parameter.

All the datasets we generated have approximately 30 minutes worth of WiFi traffic data logged in them. During this 30 minute period, we attacked the network twice. Each attack lasting for two minutes for the Original Attack and 10 minutes for the Modified Attack.

C. GP Based IDS Configuration

The parameter settings for the GP in all cases are given in Table III. In addition to the GP parameters, the fitness function utilised in this work is the switching fitness function [2]. The switching fitness function assigns credit to a member of the population depending on whether the execution of the individual on an exemplar produces a false positive (1) or a false negative (2). A higher credit value assignment at the end of the run indicates a poor performing individual.

$$Fitness \ += \frac{1}{TotalNumberofNormalConnections} \quad (1)$$

$$Fitness \ += \frac{1}{TotalNumberofAttackConnections} \quad (2)$$

V. RESULTS

In intrusion detection, two metrics are typically used in order to quantify the performance of the IDS, Detection Rate (DR) and False Positive Rate (FP), (3) and (4)respectively, a high DR and low FP rate would be the desired outcomes. In the instance of an unbalanced data set (more of one type of exemplar then the other, in this case more normal then attack) an evolved solution can survive by simply learning to label all of the exemplars as the larger type in the data set. This survival technique will provide a high DR, but also a high FP rate, an undesirable result. Undesirable results of this kind are referred to as *outlier solutions*.

$$DR = 1 - \frac{\#FalseNegativeClassifications}{TotalNumberOfAttackConnections} \quad (3)$$

$$FP = \frac{\#FalsePositiveClassifications}{TotalNumberofNormalConnections} \quad (4)$$

A. Snort-Wireless Results

When all the "Modified" datasets are replayed through Snort-Wireless, it is seen that Snort-Wireless cannot detect the attacks in them. It is worthy of note that Snort-Wireless with default parameters is only able to detect the attack in the traffic dump files if the attack is run in its default form, i.e. original attack scenario, otherwise it cannot detect the deauthentication attack if it is modified as described above.

During our experiments with Snort-Wireless, we observed that while the modified deauthentication attack was taking place, most of the clients continuously tried to renew their connections with the AP after being deauthenticated. They did this by sending an Authentication request to the AP. These authentication requests were sent each time the clients were deauthenticated, this lead to a flood of authentication requests directed at the AP during the attacks. This flood of authentication requests were erroneously detected by Snort-Wireless as an authentication attack[2]. The presence of these alerts in the snort alert logs for both the original and modified attacks is an example of proof of the effect of the attack in both the Original and Modified scenarios.

B. GP based IDS Results

In order to compare the performance of the GP based IDS against the performance of Snort-Wireless in detecting the attacks in the modified datasets, the GP based IDS is first trained on the "original attack" scenario datasets. To this end, the original datasets are both used as training and testing files in 40 runs of the GP, each time using a different initial seeding. This enabled us to determine which seeding produced the best individual,the seeding which produced the best individual is then used in the GP runs which were tested on the modified attack datasets. The results of these experiments are presented in the following.

[2]This is an example authentication flood alert: *06/23-15:14:41.810091 [**] [212:1:1](sppauthflood)Authflood detected! Addr src:00:07:e0:15:5c:5a -> Addr dst: 00:03:93:ec:64:55, Bssid: 00:03:93:ec:64:55. [**]*

C. Original Attack Datasets

The summary of the results of running the GP based IDS 40 times against the training/test original attack pairs are shown in Table IV.

Out of a total of 40 runs, nine outlier solutions are produced and are not included in the analysis. Based on these results, the seeding which produced the best overall individual, which is given in Table V, is used for GP runs on the "modified attack" scenario datasets.

D. Modified Attack Datasets

Each of the "modified" attack datasets is tested using a solution that was trained on each of the two "original" attack datasets, i.e. default deauthentication attack scenario. This way a fair comparison to Snort-Wireless is achieved as well, i.e. . GP is trained on the "original" but tested on the "modified" attack. The testing is done once using a seeding, which produced the best individual that is able to get 91% detection rate and 0.1% false positive in the training phase of our experiments, Table V. The results for testing are shown in Table VI.

The results show that the GP based IDS is able to detect the attacks in the files even though Snort-Wireless was unable to detect them. The results also show that individual produced by training on dataset 2 is a lot more accurate than the individual trained on dataset 1. While both individuals had good detection rates, the individual produced by training on dataset 1 produced an unusually high false positive rate with dataset 8. It is likely that the individual produced by dataset 2 is more accurate because the file contains a higher attack percentage than dataset 1. Having a more even distribution of exemplars in a training dataset increases the likelihood of producing a more robust solution.

E. Datasets With Increased Attack Percentages

The attack percentages in datasets created with the controlled attack are all well below 1%. It could be argued that this was responsible for the very good performance of the GP based IDS, as the low number of attacks reduced the number of points where the GP based IDS could go wrong. It should be noted however that this low attack percentages are characteristic of such datasets. Moreover, it is more difficult to detect an attack for a machine learning based algorithm in such unbalanced datasets, i.e. ˜99.9% normal and ˜0.1% attack data.

Kismet wireless, the data logging tool used in our work, provides an interface that allow for real-time monitoring of the data logging process. Information on the rate at which packets are logged is available from this interface. During our data collection we could ascertain from this interface that operation of the network, the average rate at which packets were been logged is 30 packets/sec (which leads to about 1800 packets/min). During periods when the modified attack was run, only about 19 packets/min of the total 1800 packets/min could be considered as part of the attack, all others would be normal. This corresponds to about ˜0.1% of the packets logged.

Notwithstanding, we also generated datasets with an increased attack percentage. The steps taken to increase the attack percentages (without unduly biasing the dataset) are highlighted below and the resultant datasets are summarised in Table VII.

1) Reduce the number of normal packets logged by reducing the length of time the clients spent retrieving pages from the internet.
2) Sustaining the attack for longer durations.
3) Not including the Beacon Frames in the attack. The beacon frames make up a large portion of frames logged on a wireless network and have no implication on the attack.

As can be seen from the results shown in Table VIII, the GP was able to achieve 100% detection rate in both files. However the individual produced by dataset 1 produced a high false positive rate in File 2, while the individual produced by dataset 2 did not. Again this underscores our initial assertion about solutions produced by training on dataset 2.

TABLE IV
STATISTICS ON GP BASED IDS ON ORIGINAL ATTACK DATASETS

	Max	Min	Mean
FP	0.5244	0	0
DR	1	0.2400	1
Time	47.7038	26.5402	38.5583

TABLE V
THE BEST INDIVIDUAL GIVEN BY THE GP BASED IDS

FP	DR	Time
0.0941	0.9125	48.8773

TABLE VI
RESULTS FOR GP-BASED IDS ON THE MODIFIED ATTACK

	Dataset 1			Dataset 2		
File	**FP**	**DR**	**Time**	**FP**	**DR**	**Time**
3	4.84E-05	1	35.4177	0	1	42.2258
4	0.000194	1	33.0050	0.000803	1	39.8475
5	0	1	35.0290	0.006078	0.9783	51.4397
6	0.129877	1	31.9577	0.09508	1	47.4222
7	0.000484	1	36.3893	0.006078	1	51.4573
8	0.494433	1	33.7363	0	1	42.6153
9	0	1	35.4747	0	1	41.554
10	0	1	31.0705	0	1	42.7578
11	0	1	33.3265	0	1	48.3118
12	0	1	34.7407	0	1	47.3905
13	0	1	32.6878	0	1	50.379
14	0.088537	1	33.6962	0	1	50.4942
15	0.119518	1	34.8012	0	1	50.0882
16	0	1	32.1827	0	1	35.242

TABLE VII
DATASETS WITH INCREASED ATTACK %

File	Size	Attack	Type	% Attack
1	1471	19	Modified	1.30
2	1160	26	Modified	2.24

TABLE VIII

RESULTS ON THE DATASETS WITH INCREASED ATTACK PERCENTAGES

	Dataset 1			Dataset 2		
File	**FP**	**DR**	**Time**	**FP**	**DR**	**Time**
1	0	1	48.8773	0	1	48.6145
2	0.3513	1	26.2513	0.0012	1	38.478

VI. CONCLUSION AND FUTURE WORK

In this work we designed, developed and tested a L-GP based IDS on Data Link Layer attacks on Wifi networks. Based on previous work [2], we implemented a feature set and employed a switching fitness function for our IDS as well as generated a data set to train and test it. Moreover, we compared the performance of the GP based IDS with Snort-Wireless under the same datasets. In these datasets, the deauthentication attack was run in two forms: (i) Original Attack Scenario, where the attack was run using default parameters as in the void11 attack tool employed. In a way, this corresponds to a new attacker just finding and employing the tool to attack. (ii) Modified Attack Scenario, where the attack was run by changing the parameters given in the vioid11 tool. This corresponds to a more experienced attacker, who is crafting a more difficult attack. To the best of our knowledge this is the first time such a comparison has been performed between a machine learning based IDS system, namely GP based IDS, and a signature based system, namely the well known Snort-Wireless.

The results show that both systems can detect the deauthentucation attack under the original attack scenario scenario but only the GP based IDS can detect the attacks under the second scenario, modified attack. In this case, the GP based IDS provides high detection rates while still maintaining a low false positive rate. The more consistent results of the GP based IDS does indicate that it encourages the evolving of solutions that can handle the modified attacks and the unbalanced nature of our data sets . Compared to implementing Snort-Wireless for detecting this DoS attack, the GP based IDS does not require a user to set a threshold count of de-authentication frames nor a maximum time window size for this count to be met. It eliminates this requirement, providing a more robust tool for detecting the DoS attack. Our future work will explore the use of larger data sets for training and testing our L-GP based IDS using more than one AP. This will allow us to verify the effectiveness of our work over larger networks as well as a varied number and length of DoS attacks.

Furthermore, we plan on applying the same approach described here on other WiFi attacks, with the goal of developing an IDS that can be used to detect a variety of attacks.

ACKNOWLEDGEMENTS

This research is supported by the NSERC Discovery and the CFI New Opportunities grants. This work is conducted as part of the NIMS project at http://www.cs.dal.ca/projectx/.

REFERENCES

[1] W. A. Arbaugh, N. Shankar, Y. Wan, and K. Zhang, "Your 802.11 wireless network has no clothes," *IEEE Wireless Communications*, vol. 9, pp. 44 – 51, December 2002.

[2] P. LaRoche and A. N. Zincir-Heywood, "Genetic programming based wifi data link layer attack detection," in *CNSR 2006*. Los Alamitos, CA 90720-1314: IEEE Computer Society, May 2006, pp. 285 – 292.

[3] IEEE-SA, *ANSI/IEEE Std. 802.11*, 1993rd ed., IEEE, New York, NY, USA, 2003.

[4] M. Maxim and D. Pollino, *Wireless Security*. McGraw Hill, 2002.

[5] J. Bellardo and S. Savage, "802.11 dos attacks: Real vulnerabilites and practical solutions," *USENIX Security Symposium*, pp. 15 –18, 2003.

[6] R. Floeter, "Void11 main page, www.wirelessdefence.org/contents/void11main," Retrieved from the Web., August 2006.

[7] J. Malinen, "Host ap driver for intersil prism2/2.5/3, hostapd, and wpa supplicant, http://hostap.epitest.fi," Retrieved from the Web., 2006.

[8] M. Crosbie and E. Spafford, "Applying genetic programming to intrusion detection," in *AAAI Symposium on Genetic Programming*, J. K. E.V. Siegel, Ed., AAAI. Cambridge, MA, USA: MIT, 1995, pp. 1 – 8.

[9] Sourcefire-Inc, "Snort - the de facto standard for intrusion detection/prevention, http://www.snort.org," Retrieved from the Web., 2006.

[10] A. Lockhart, "Snort wireless, http://www.snort-wireless.org," Retrieved from the web., 2005.

[11] J. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbour, Michigan, USA: University of Michigan Press, 1975.

[12] J. Koza, "Genetic programming: A paradigm for genetically breeding populations of computer programs to solve problems," Computer Science Department , Stanford University, Tech. Rep., 1990.

[13] M. Heywood and A. Zincir-Heywood, "Page-based linear genetic programming," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 5, pp. 3823 – 3828, 2000.

[14] C. Gathercole and P. Ross, "Dynamic training subset selection for supervised learning in genetic programming," *Parallel Problem Solving from Nature III*, vol. 866, pp. 312 – 321, 1994.

[15] D. Song, M. Heywood, and A. Zincir-Heywood, "Training genetic programming on half a million patterns: an example from anomaly detection," *IEEE Transactions on Evolutionary Computation*, pp. 225 – 239, 2005.

[16] M. Kershaw, "Kismet wireless, http://www.kismetwireless.net," Retrieved from the Web., 2006.