Adaptability of a GP Based IDS on Wireless Networks

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

***Abstract*— Security and Intrusion detection in WiFi networks is currently an active area of research where WiFi specific Data Link layer attacks are an area of focus; particularly recent work has focused on producing machine learning based IDSs for these WiFi specific attacks. These proposed machine learning based IDSs come in addition to the already deployed signatures which are already in use in conventional intrusion detection systems like Snort-Wireless and Kismet. In this paper, we compare the detection capability of Snort-Wireless and a Genetic Programming (GP) based intrusion detector, based on the ability to adapt to modified attacks, ability to adapt to similar unknown attacks and infrastructure independent detection. Our results show that the GP based detection system is much more robust against modified attacks compared to Snort-Wireless. Moreover, by focusing on the method(s) used in feature pre-processing for presentation to learning algorithms, GP based IDSs can achieve infrastructure independent detection and can adapt to similar unknown attacks too. On the other hand, even though Snort-Wireless is an infrastructure independent detector, it cannot adapt to unknown attacks even if they are similar to others for which it has signatures on.**

I. Introduction

The goal of any Intrusion Detection System (IDS) is to protect a network or host from malicious activity and traffic. With the widespread use of networks based on the IEEE 802.11 networking standard (also known as Wireless Fidelity (WiFi) networks) and the known security vulnerabilities of this protocol [1], it is not surprising that most conventional commercial and open source IDSs now have signatures to tackle WiFi specific attacks. 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; 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.

Several classes of WiFi specific attacks exist. Denial of Service (DoS) attacks, which exploit Management Frames at Media Access Control (MAC) or Data Link layer are an example. They 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. Attack types in this class of attacks are many; our work focuses on three well-known attacks in this class: de-authentication flood attack, authentication flood attack and association flood attack. Recent research has proposed machine learning based solutions for these data link layer attacks, where [2] proposes Genetic Programming (GP) based solutions while [3] proposed a solution based on Artificial Neural Networks (ANNs). It is therefore important that these machine learning based IDS be compared against existing technology i.e. Conventional Signature based IDSs, which are already been used in detecting these attacks.

In this paper, we compare the detection capabilities of Snort-Wireless, a signature based IDS, against a machine learning, namely 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. On the other hand, based on past success of the use of Genetic Programming based solutions for Intrusion Detection [2], [4], [5] it is chosen as our machine learning paradigm. Our comparison criteria include adaptability in the face of modified attacks, infrastructure independent detection and adaptability to similar unknown attacks.

Snort-Wireless detects data link layer attacks 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. This forms our basis for comparison in adaptability to modified attacks. On the other hand by infrastructure independent detection we refer to the ability to seamlessly port an IDS signature from one physical network to another with no change to the signature without a drop in performance. With adaptability to similar unknown attacks, we mean the ability for the detection signatures for one attack type to detect a different but similar attack. Conventional Signature based IDSs e.g. Snort-Wireless and Kismet that can be used for detecting data link layer attacks are infrastructure independent but their signatures do not detect unknown similar attacks. Thus, our objective is to

investigate how far a GP based IDS can go under the same circumstances.

The remaining sections of this paper is organised as follows. Section 2 gives an overview of wireless networks and exploits that target their data link layer. Section 3 discusses the intrusion detection investigated in work. Section 4 outlines the experiments and explains our approach. Section 5 presents the results and conclusions are given in Section 6.

II. MAC Layer Denial of Service (DoS) Attacks on WiFi Networks

Infrastructure WiFi networks generally consist of a backbone and a number of clients, which can be any device from laptop computers to wireless Personal Digital Assistants (PDAs). The backbone consists of one or more Access Points (APs) which the clients connect to and who in turn are connected to a wireline network. A WiFi network which consists of only clients is referred to as an Ad-Hoc network. We only deal with Infrastructure networks in our work.

These networks communicate over a wireless medium using the IEEE 802.11 standard. Variants based on the IEEE 802.11 standard include 802.11b, 802.11g and others. These variants 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 but our results can be generalized to the other variants of the standard, as the difference between these variants exist more at physical layer of the 802.11 protocol.[6].

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[7]. However, security is of great concern in WiFi networks. WiFi networks are susceptible to attacks, to which their wired counterparts are not susceptible. Data transmitted using open airwaves as a transmission medium can easily be intercepted. Several protocols like Wireless Encryption Protocol (WEP), WiFi Protected Access (WPA) and wireless Virtual Private Networks (VPN), have been proposed to ameliorate these vulnerabilities but so far these protocols do not prevent attacks that target the physical and data link layers of the OSI protocol stack. Attacks at these layers are usually DoS attacks, DoS attacks work to make a network unusable or inaccessible to legitimate clients.

A. WiFi Network Management Frames

The IEEE 802.11 standard defines three broad classes of frames i.e. management frames, control frames and data frames. Management frames are the focus of our work, types of management frames include: Association, Disassociation, Authentication, De-authentication, Beacon and Probe frames. Management frames are used by stations to establish and maintain connections [6], this makes them the target of most attacks, which aim to make a WiFi network unusable.

The process of joining a WiFi network for any client is a two step process. The first step is an “Authentication”, the client sends an authentication request and the AP replies with an authentication response. The authentication request and authentication response are sent using authentication frames. The next step is an “Association” with an AP using association frames, the process is similar to the authentication. Either step can be revoked at anytime using De-authentication or Disassociation frames. The WiFi attacks used in our work exploit the Authentication, Association and De-authentication attacks.

B. MAC Layer DoS Attacks

WiFi specific MAC layer DoS attacks which exploit management frames are very easy to implement. The first step is an information gathering stage, an attacker simply eavesdrops on a network using a passive wireless network monitoring tool and logs frames emanating from the target network. With this logged traffic data, the attacker is able to filter information about the stations on the network. The attacker then uses this information to create forged management frames with a spoofed MAC address of a station or an AP on the network. If the attacker chooses to target a specific client, it creates a management frame with the MAC address of the target client as the destination and the MAC address of the AP as the source. The attacker can also choose to vary the scope of the attack i.e. by focusing on the AP to take down the entire network, or by targeting a group of clients. Our work utilises three MAC layer attacks i.e. De-authentication, Authentication and Association attacks. These attacks are named after the management frames which they exploit, see Section II-A.

C. Attack Generation Tool

Void11 is the attack generation tool used in our work. It is a free software implementation of some common 802.11b attacks [8]. The basic implementation works in a command line Linux/Unix environment. Void11 requires a prism based wireless Network Interface Card (NIC) and hostap drivers installed on the computer on which is to be deployed. The hostap drivers allow the machine to act as a wireless AP [9].

Void11 implements the three data link layer attacks that we utilize in this work. 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 or keep the AP busy dealing with client requests, which slows down the network. The end result of each of these attack types differs based on the rate of injection of the frames and on the client involved. All the MAC layer attacks, which are launched to create the datasets used in our experiments, are executed using void11 and the default values of its command line arguments, except in the case of the modified de-authentication attack. It should be noted that void11 does not simulate the DoS attacks

it implements but mounts actual attacks which can make a WiFi network unusable.

The delay (-d) switch (in the command syntax used to launch void11) [8], 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 $10000\mu s$ [8]. Assigning a different value to this switch can be used to stealthily inject frames into the target network.

III. Data Link Layer Intrusion Detection

Intrusion Detection Systems (IDSs) are used to detect attacks against the integrity, confidentiality and availability of computer networks [2], [4]. 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 an 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:

1) **Deployment:** Whether to monitor incoming traffic or host information.
2) **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. Traditionally, developing IDSs required a human expert to construct a set of rules, which when triggered, would indicate malicious activity. In this section, we briefly discuss the intrusion detection systems utilised in this work i.e. Snort-Wireless and Genetic Programming (GP) based IDSs. 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 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 [10]. Using signature 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 [11]. These patches enable Snort (Snort-Wireless, after patches are applied) to detect WiFi specific attacks. Signatures that detect WiFi MAC Layer attacks are among the patches included in Snort-Wireless.

We setup physical networks and attacked them using void11. The traffic from these physical networks was logged using the traffic logging features of Kismet [12]. This logged traffic data files were then replayed in their raw tcpdump format to Snort-Wireless and to the GP based IDS after processing. Thus, the experiments with the detectors were done off-line.

The most important metrics used by Snort-Wireless to detect the de-authentication attack are the number of de-authentication 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 in Snort-Wireless are 20 frames and 60 seconds respectively [11]. While this setup can detect most attacks effectively, an attacker who injects only 19 frames every 60 seconds will go undetected with such a signature. This fact was used in setting up our stealth attacks. The values of these metric are the same for authentication attack and the association attack, the only difference is of course the frame type been monitored.

B. GP Based Detection

Recent research has been focused on the use of machine learning solutions in the detection of 802.11 MAC layer attacks [2], [3]. Our work focuses on GP based detectors, based on its successful use in detecting the de-authentication attack [2] and other higher level attacks [5]. GP is an extension of the Genetic Algorithm (GA); which is an evolutionary computation (EC) method proposed by John H. Holland [13]. GP extends the GA to the domain of evolving complete computer programs [14]. Using the Darwinian concepts of natural selection and fitness proportional breeding, populations of programs are genetically bred to solve problems. In tune with the fitness proportional breeding paradigm of GP, a fitness function is required, the fitness function assigns a value to the performance of an individual in the environment (the problem we hope to solve). This value is then used to determine which individuals can breed to produce the members of the next generation. The fitness function utilised in this work is the switching fitness function [2]. The switching fitness function assigns a credit value (fitness value) to a member of the population depending on the kind of error the execution of the individual on an exemplar generates if any, i.e. it either produces a false positive, Eq. (1) or a false negative, Eq. (2). During each generation, the variables $Fitness(n)$ and $Fitness(n+1)$ represent the fitness value of an individual before an evaluation and its fitness after an evaluation respectively. A generation proceeds by consecutively testing each member of the population against the exemplars in the traffic log dataset, if the individual incorrectly classifies an exemplar, its fitness value is incremented using either Eq. (1) or Eq. (2). A run of the GP would consist of a predetermined number of generations. A higher credit value assignment at the end of the run indicates a poor performing individual.

$$Fitness(n+1) = Fitness(n) + \frac{1}{TotalNo.OfNormalConnections} \quad (1)$$

$$Fitness(n+1) = Fitness(n) + \frac{1}{TotalNo.OfAttackConnections} \quad (2)$$

The populations of programs been bred by the GP can either be represented as tree like LISP structures or as binary strings, which represent integers. In the binary string representation, 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, these instructions 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. This is known as the Linear Page Based GP (L-GP) [15].

The GP algorithm is computationally intensive and this fact can be multiplied considerably when dealing with large data sets. 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 [16]. To do so, the RSS-DSS algorithm utilizes a hierarchical sampling of training exemplars, dividing the problem into two levels, a RSS level and then a DSS level[5]. The RSS level divides the training set into blocks of equal size, the second level chooses (stochastically) a block and places it in memory and then 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.

Our work utilizes the L-GP approach, alongside the Random Subset Selection - Dynamic Subset Selection (RSS-DSS) algorithm [16], detailed below. L-GP has been used successfully by other researchers in the realm of IDSs [2], [4], [5]. The parameter settings for the GP in all cases are given in Table I.

TABLE I
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

IV. EXPERIMENTS

Our experiments require that we have appropriate datasets. In order to generate such datasets, we set up three separate physical networks i.e. Network-I, Network-II and Network-III, see Figure 1. The components of these networks include 3 APs, 8 PDAs, 2 laptops and 2 desktop machines. While Network-I and Network-II are setup in the same manner, see Figure 1 (a), Network-III is setup as an ESS (Extended Service Set) with two APs, see Figure 1 (b). All the clients on all three networks are connected to the APs via 802.11 connections on channel 6. Attacks are generated on all networks using void11 installed on the attack machine. Data is collected on the monitoring machine using the data logging features of Kismet Wireless [12]. The only difference between Network-I and Network-II is the APs. In Network I, an Airport based AP is employed, whereas in Network II a Cisco based AP is employed. In doing so, our aim is to simulate two different network environments. An AP is central to any infrastructure based wireless network, creating two networks with different APs simulates different network environments. By setting up Network-III, our aim is to simulate a network environment, which involves more than one AP, and therefore represents more closely a real-world enterprise network.

The attack traffic consisted of the intermittent release of a stream of management frames into the network traffic. In all cases the source and the target MAC addresses of the frames are set to that of an AP, a client or the broadcast address **(ff:ff:ff:ff:ff:ff)**, depending on the scope of the attack. To ensure that normal traffic is also generated on our test networks, a web crawler is implemented , using the Java 2 Platform, Micro Edition (J2ME). This web crawler ensures a continuous stream of web browsing requests from the clients as the background normal traffic.

A. GP Training/Testing Features

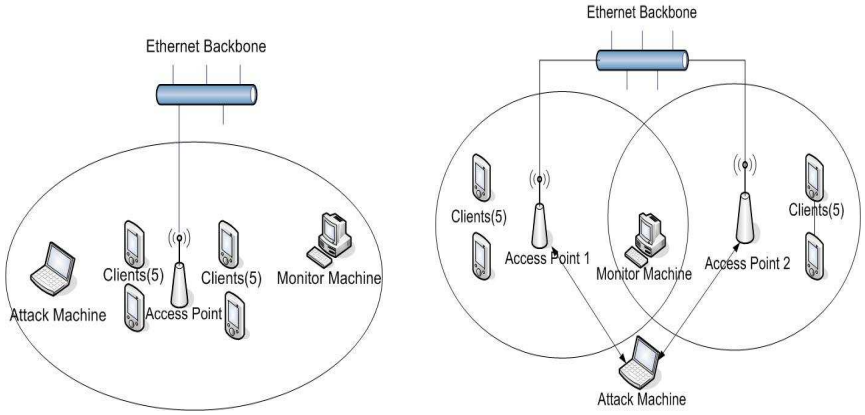
An IEEE 802.11 frame contains several fields. Out of this number of fields, an appropriate subset was selected for use as the feature set for training/testing our GP based IDS, as not all of the fields are relevant to the attacks. Table II outlines the fields employed as features input to the GP. Three of them being MAC addresses and all others being numeric.

TABLE II
GP TRAINING/TESTING FEATURES

Feature	Type	Range
Frame Control	Numeric	0 - 47
Destination Address	Nominal	00:00:00:00:00:00 - FF:FF:FF:FF:FF:FF
Source Address	Nominal	00:00:00:00:00:00 - FF:FF:FF:FF:FF:FF
Basic Service Set Ident. (BSSID)	Nominal	00:00:00:00:00:00 - FF:FF:FF:FF:FF:FF
Fragment Number	Numeric	≥ 0
Sequence Number	Numeric	≥ 0
Channel	Numeric	0 - 13

B. GP Training/Testing Data Sets

A total of 40 tcpdump traffic log files were collected during the course of our work. These log files were then passed through several processing stages which include feature extraction, feature mapping for appropriate data types and the grouping of individual frames to form sessions. The processing of the dataset files was achieved by passing the tcpdump files through a number of scripts. Table III gives a summary of the resulting 40 dataset files, the table outlines the attack type, the physical network on which the original log files were collected, the number of files collected and the form of the attack i.e. Original or Modified.



(a) Network-I and Network-II (b) Network-III

Fig. 1. Setup of Test Networks (a) Network-I and Network-II (b) Network-III.

TABLE III
DATASET SUMMARY

Attack Type	Network	# of Files	Attack Form
De-authentication	I	14	Modified
De-authentication	I	10	Original
De-authentication	II	10	Original
Authentication	III	2	Original
Association	III	2	Original

In Table III, the datasets marked with attack form "Original" are those, which are generated by implementing the attack with the default value of the "delay" parameter. The default setting of the "delay" parameter is $10000\mu S$ [8], 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 have an attack type of "Modified" are generated using this value for the delay parameter.

V. RESULTS

In intrusion detection, two metrics are typically used to quantify the performance of the IDS,

(i) Detection Rate (DR)
(ii) False Positive Rate (FP)

which are Eq. (3) and Eq. (4) respectively. A high DR and a low FP rate would be the desired outcomes. Evaluation of our results is based on the above criteria.

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

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

In the instance of an unbalanced data set (significantly 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*. Outlier solutions are classified as failed experiments when they occur and are excluded from the final results presented here.

A. Adaptability to Modified Attacks

Most network exploits follow a predefined pattern or number of steps, a fact which is used in designing conventional signatures for detecting them. Sometimes an attacker who is aware of this fact may choose to deviate from the normal pattern of an attack, in a bid to evade detection signatures for the attack. The "Modified" datasets in Table III were created with this mindset, an IDS which can adapt to such modified attacks is desirable.

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 de-authentication attack if it is modified as described previously.

On other hand the results for the GP based IDS give interesting 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 two of the "original" de-authentication files collected on Network-I. Each of the "modified" de-authentication attack datasets is then tested using a solution that was trained on each of the two "original" de-authentication attack datasets. This way a fair comparison to Snort-Wireless is achieved, i.e. GP is trained on the "original" attack but tested on the "modified" attack. The results for testing are shown in Table IV.

These 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.

B. Infrastructure Independent Detection

It is safe to state that no two physical networks are identical in setup and composition. An IDSs needs to be able to work on any network irrespective of the physical setup of the network. An IDS which can operate on any network irrespective of its physical setup can be said to be "Infrastructure Independent". Conventional IDSs like Snort-Wireless are capable of infrastructure independent attack detection, for a machine learning based IDS to be infrastructure independent it must be able to work on a network other than the one on which it was trained.

Recent research has shown that machine learning based IDS for 802.11 MAC layer attacks unlike their conventional counterparts are indeed susceptible to diminished performance when used on network other than that on which they were trained if no attention is paid to the representation of features employed [17]. In [17], it was shown that from one network to another, the performance of both Artificial Neural Network(ANN) and GP based IDSs can drop to 46% and 75% respectively, from a high of 99%, if no attention is paid to the feature representation technique, especially with the MAC addresses. As a solution we propose an Improved Role-Based MAC address mapping technique, Algorithm 1. This new Role-Based mapping technique maps the MAC addresses based on 5 recoqnised roles, i.e. Broadcast, Access Point, Station/Client, Host and Other.

While the numeric representations assigned to MAC addresses which fall into the Broadcast, Access Point, Host or Other role by the mapping technique are not derived from the actual MAC addresses themselves, the numeric representations assigned to MAC addresses of type Station/Client is derived from a hashing function, which we call Decimal-Sum. The steps in this function are visualized in Figure 2. The decimal-sum hashing technique is designed to achieve a balance between mapping the MAC addresses to large integers and the need to achieve a perfect hash. The scheme will map the addresses to integers in the range of 0 - 765,765 and has a $2 \cdot 90 \times 10^{-12}$ chance of a collision occurring.

Using this new Role-Based mapping technique, our work in investigating infrastructure independent detection was carried out by performing a 20-fold cross validation using the 20 original de-authentication datasets collected on Network-I and Network-II, see Table III. This means that results were produced from all possible combinations of training and testing pairs of the datasets. The final results are divided into four groups:

- Group-1: Results of Testing On Network-I datasets using solutions trained on Network-I datasets
- Group-2: Results of Testing On Network-II datasets using solutions trained on Network-II datasets
- Group-3: Results of Testing On Network-I datasets using solutions trained on Network-II datasets
- Group-4: Results of Testing On Network-II datasets using solutions trained on Network-I datasets

In these experiments, Groups (1) and (2) are called within-platform results, whereas (3) and (4) are cross-platform results. A good performance on cross-platform results should indicate acceptable infrastructure independent detection.

The mean DRs are presented in Table V. We see that FP rates remain pretty much constant for all solutions within or across platforms. We however do not notice the significant drop in performance when comparing DRs within-platform with those across platform. In other words, the average DRs remain relatively constant whether solutions are used within-platform or across platform. These results show that a GP based IDS is capable of being an infrastructure independent detector such as Snort-Wireless.

TABLE IV
PERFORMANCE OF GP BASED IDS ON MODIFIED DE-AUTHENTICATION ATTACK DATASETS

	FP	DR	TIME
1st Quartile	0.00	0.77	21.32
Median	0.01	0.99	38.84
3rd Quartile	0.32	1.00	54.53

TABLE V
MEAN PERFORMANCE USING GENETIC PROGRAMMING

Within Platform		
FP	DR	TIME
0.02	0.97	26.68

Across Platform		
0.02	0.96	26.69

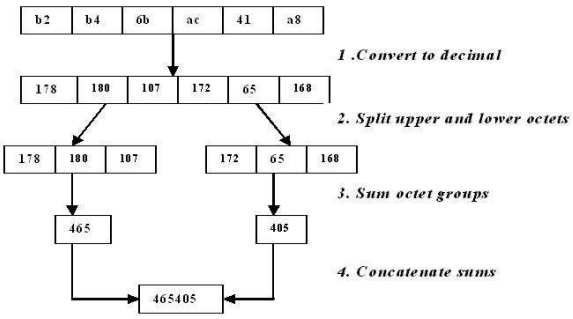


Fig. 2. Decimal-Sum Hashing Technique.

C. Adaptability to Similar Unknown Attacks

Sometimes two or more technically different attacks may have similar modes of implementation or very similar effects on their target network. Despite the similarity of such attacks,

Algorithm 1 Improved Role-Based Mapping

Input: Array $X[]$ containing all MAC addresses in dataset .
Output: Array $Y[]$ containing integer mappings of the MAC addresses in $X[]$. {Mapping of $X[i] = Y[i]$}

```
Sort(X)
i = 1
ap_no = 0 {Stores a count of the number of Access Points seen so far}
for every macaddr in X do
    Role = DetermineRole(macaddr) {Role can either be Broadcast, Access_Point, Host, Station or Other}
    if Role = Broadcast then
        Y[i] = 1
        i++
    end if
    if Role = Access_Point then
        Y[i] = 2.ap_no {Y[i] is an integer formed from the concatenation of the string equivalents of 2 and ap_no}
        i++
        ap_no++
    end if
    {Next IF is included only if data is been processed for a Host Based IDS}
    if Role = Host then
        Y[i] = 3
        i++
    end if
    if Role = Station then
        Y[i] = DecimalSum(macaddr)
        i++
    else
        Y[i] = 5 {Assumed that Role = Other}
        i++
    end if
end for
Return(Y)
```

a conventional IDS needs to create separate attack signatures for them.

To give an example, the Association Flood and the Authentication Flood are very similar attacks, differing only in the management frame exploited. In order to detect each one of these attacks, Snort-Wireless needs to have separate signatures for each attack, since one of the features is different. In a bid to eliminate this need, we also investigate whether a GP based IDS solution trained on the Authentication attack can detect the Association attack and vice versa. The results presented here followed the same training and testing procedure outlined in our previous set of experiments, the only difference been that they were carried out on the 4 Association and Authentication datasets presented in Table III.

Table VI shows the results for the FP and DRs, using the Role-Based mapping scheme. We are able to achieve above 98% DR when employing the authentication solution against the association attack and 71% DR vice versa. Keeping in tune with idea of paying attention to feature presentation, the Authentication and Association subtypes were represented as 11 and 13 (numbers which cluster close to each other) respectively, in the datasets used in these experiments. These results serve to further reinforce similar results which were seen with the De-authentication and Dissociation attacks in [2]. As discussed earlier, conventional detectors like Snort-Wireless are incapable of this kind of detection. Conventional detection signatures will not detect any other exploit other than that which they were designed for no matter how similar the attack is. Indeed, further experiments are needed to improve and analyse the implications of these results.

TABLE VI
MEAN PERFORMANCE USING GENETIC PROGRAMMING WITH ROLE-BASED MAPPING

Within Platform			
	FP	DR	TIME
Auth. vs. Assoc.	0.01	0.98	33.85
Assoc. vs. Auth.	0.00	0.67	32.85
Across Platform			
Auth. vs. Assoc.	0.01	0.94	36.81
Assoc. vs. Auth.	0.00	0.71	32.05

D. Analysis of GP Solutions

GP based IDSs as compared to other Machine Learning algorithms have the advantage of producing solutions that can be deciphered. In this section, we present the results of our preliminary analysis of the solutions produced by GP. The results are from a randomly chosen set of 20 best performing individuals for the de-authentication attack. Figure 3 and Figure 4 visually present an analysis of the information contained in the GP solutions based on frame, feature, operand and constant value usages.

Each exemplar presented to the GP algorithm during training and testing consists of eight individual frames, which are grouped together in temporal order to form a session. Exemplars with the last frame (8th) being part of the attack are labeled as attack exemplars, while others are labeled as normal. From the results in Figure 3 (a), we can see that the algorithm produces solutions that sample more features from the last frame (23%), the reason for this is obvious, given that the last frame represents the attack. However we are of the opinion that the distribution of frame use is more "balanced" than expected, as it is logically possible to discriminate the exemplars by focusing only on the last frame in each exemplar. The GP solutions also pay attention to the state of the network (represented by the earlier frames) in their detection. It is also pertinent to mention that we did have a best performing GP solution that did not sample features from the 8th frame at all. This implies that a solution was able to work solely on identifying the attack based on the state of the network and

not based on the frame injection. We believe this can provide a very successful generalization for the classifier.

In Figure 3 (b), we can see the distribution of features used. The GP solutions focus mainly on the fragment number, BSSID, frame type and the sequence number. The focus on these features is understandable;

- **BSSID:** The BSSID helps to differentiate legitimate traffic, the BSSID identifies the network of a frame. Any frame with a "foreign" BSSID should be ignored.
- **Frame Type:** Our attacks exploit management frames, it is logical to except that an effective IDS should be able identify the management frame types used in the attack from other management frames and frames in general.
- **Fragment and Sequence Numbers:** The management frames generated by our attack tool are forged. An analysis of the fragment and sequence numbers can be used to differentiate between the forged frames used in the attack and the legitimate frames been used on the network. It is not surprising therefore that the GP based IDS focuses on them.

.

On the other hand Figure 3 (c) shows the operand distribution for the GP solutions. The results show a tilt toward the use of subtraction and multiplication for the GP solutions. In addition to the feature values presented to a GP solution, a GP solution is allowed to use randomly generated integer values in its calculations. The range of values for these constants is 0 - 255. Figure 4, shows the distribution for these constants in ranges of width 30. The results show an affinity for values in the 0 - 29 and 90 - 119 range.

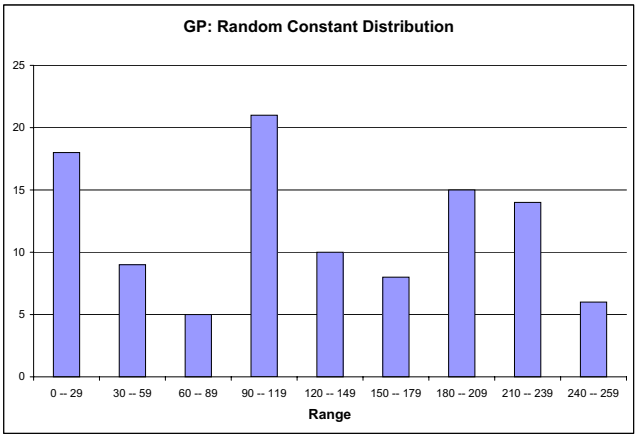


Fig. 4. Constant Usage Distribution for GP Solutions.

VI. Conclusion and Future Work

In this work we have not only showed that GP based IDSs are more adaptable than conventional IDSs in the detection of modified attacks which are crafted to evade detection signatures. We have also shown that GP based IDSs have the capacity for infrastructure independent detection and are able to detect similar unknown attacks. This is possible, if attention is paid to the representation of the features in the training and testing datasets presented to the learning algorithm.

We compared the performance of the GP based IDS with Snort-Wireless under different attack scenarios of the de-authentication attack i.e.: (i) Original Attack Scenario, where the attack was run using the default parameters of void11 and (ii) Modified Attack Scenario, where the attack was run using modified parameters. The results show that both systems can detect the de-authentication attack under the original attack scenario but only the GP based IDS can detect the attacks under the modified scenario. The more consistent results of the GP based IDS does indicate that it encourages the evolving of solutions that can handle the modified attacks. Unlike Snort-Wireless, 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, to detect the attack. Thus, GP based IDSs eliminate this requirement, providing a more robust tool for detecting the DoS attack.

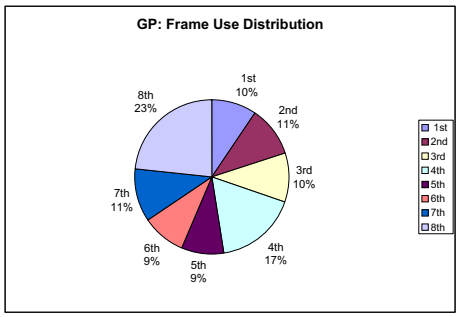
Carrying these results further, using the de-authentication attack again, we showed that by using a Role-Based paradigm for mapping the MAC addresses in our feature set, infrastructure independent detection for GP-Based IDSs can be enhanced. Moreover, we also show that a GP based IDS is capable of adapting in the face of similar unknown attacks. The results show that a GP based IDS, trained on one type of DoS attack can detect a different DoS attack other than the one it is trained on. This is type of adaptivity is not observed with the conventional detection signatures of Snort-Wireless, conventional signatures are incapable of detecting unknown attacks even if the new attack is similar. These results show that a machine learning based IDS, such as GP, not only has high dependability but also has survivability capabilities unlike conventional IDS.

Finally we present a preliminary analysis of the composition of the GP solutions evolved. Future work will explore applying this approach on other WiFi attacks, with the goal of developing an IDS that can be used to detect a variety of attacks. Moreover, we also believe that the Role-Based paradigm can also be used to enhance infrastructure independent detection for network names at higher levels of the network stack.

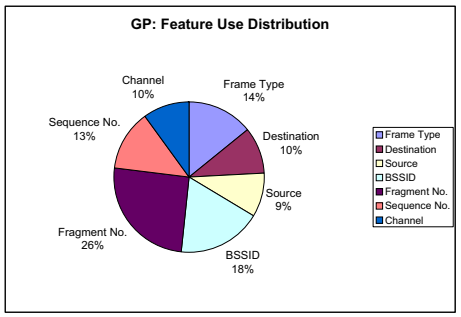
Acknowledgements

This research is supported by the NSERC Discovery and the CFI New Opportunities grants. The authors would also like to acknowledge the staff of Palomino System Innovations Inc., based in Toronto, Ontario and Telecoms Applications Research Alliance (TARA), based in Halifax, Nova Scotia for their support in completing this work.

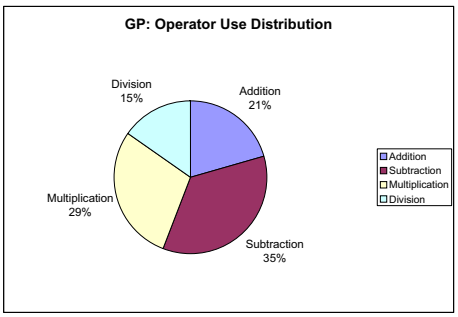
This work is conducted as part of the Dalhousie NIMS Lab at http://www.cs.dal.ca/projectx/.



(a) Frame Usage



(b) Feature Usage



(c) Operand Usage

Fig. 3. Distribution Analysis for GP (a) Frame Usage (b) Feature Usage (c) Operand Usage.

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] Y. Liu, D. Tian, and B. Li, "A wireless intrusion detection method based on dynamic growing neural network," *1st International Multi-Symposium on Computer and Computational Sciences*, 2006.

[4] 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.

[5] 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.

[6] I.-S. S. Board, *ANSI/IEEE Std. 802.11*, 1999th ed., IEEE, New York, NY, USA, 2003.

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

[8] R. Floeter, "Void11: A free implementation of some basic 802.11b attacks, http://www.wirelessdefence.org/contents/void11main.htm," Retrieved from the Web., September 2007.

[9] J. Malinen, "Host AP Driver , http://hostap.epitest.fi," Retrieved from the Web., September 2007.

[10] Sourcefire-Inc, "Snort - the de facto standard for intrusion detection/prevention, http://www.snort.org," Retrieved from the Web., September 2007.

[11] A. Lockhart, "Snort wireless, http://www.snort-wireless.org," Retrieved from the web., September 2007.

[12] M. Kershaw, "Kismet wireless, http://www.kismetwireless.net," Retrieved from the Web., September 2007.

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

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

[15] 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.

[16] 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.

[17] A. Makanju and A. N. Zincir-Heywood, "Investigating cross-platform robustness in machine learning based idss for 802.11 networks." *International Journal of Computer Science and Network Security*, pp. 1 –9, June 2007.