CORE
Metadata, citation and similar papers at core.ac.uk
Provided by UEL Research Repository at University of East London

Intelligent Detection of MAC Spoofing Attack in 802.11 Network

Chafika Benzaïd*
Division Sécurité Informatique
CERIST, Algérie
cbenzaid@cerist.dz

Abderrahman Boulgheraif
Dept. of Computer Science
USTHB, Algérie
b-abderrahmane@outlook.com

Fatma Zohra Dahmane
Dept. of Computer Science
USTHB, Algérie
fzdahmane@gmail.com

Ameer Al-Nemrat
ACE, UEL
United Kingdom
ameer@uel.ac.uk

Khaled Zeraoulia
Dept. of Computer Science
USTHB, Algérie
kzeraoulia@usthb.dz

ABSTRACT

In 802.11, all devices are uniquely identified by a Media Access Control (MAC) address. However, legitimate MAC addresses can be easily spoofed to launch various forms of attacks, such as Denial of Service attacks. Impersonating the MAC address of a legitimate user poses a big challenge for cyber crime investigators. Indeed, MAC spoofing makes the task of identifying the source of the attack very difficult. Sequence number analysis is a common technique used to detect MAC spoofing attack. Existing solutions relying on sequence number analysis, adopt a threshold-based approach where the gap between consecutive sequence numbers is compared to a threshold to decide the presence of a MAC spoofing attack. Nevertheless, threshold-based approach may lead to a high rate of false alerts due to lost or duplicated frames.

To overcome the limitations of threshold-based approach, this paper proposes a detection method that relies on a machine learning approach, namely Artificial Neural Network (ANN). ANNs provide the potential to identify and classify network behavior from limited, noisy, incomplete and non-linear data sources. The experimentation results showed the effectiveness of the proposed detection technique. Moreover, we proposed a user-friendly graphical representation of information to support the interpretation of quantitative results.

*Dept. of Computer Science, USTHB, Algérie. email: cbenzaid@usthb.dz

ACM 978-1-4503-3738-0.
http://dx.doi.org/10.1145/0000000.0000000

CCS Concepts

•Applied computing → Network forensics; •Networks → Mobile and wireless security; •Computing methodologies → Neural networks;

Keywords

802.11; MAC spoofing; Artificial Neural Networks

1. INTRODUCTION

The advent of wireless technology and the proliferation of portable devices have led to a wide adoption of Wi-Fi based networks in many domains, ranging from enterprises to homes. This adoption is justified by the greater flexibility of deploying and using these networks, in addition to ensuring continuous access to services and resources while moving. However, these emerging networks are increasingly sensitive to cybercrime attacks and identity theft by hackers who maliciously exploit wireless LANs and Wi-Fi access points. Due to the open nature of wireless medium, anyone within the range of an access point can potentially get access to the signals and possibly misuse them to carry out unlawful acts.

In 802.11, all devices are uniquely identified by a physic address known as Media Access Control (MAC) address. A MAC address is a physical identifier stored in the Network Interface Card (NIC). Unless the MAC address is overridden by the user, this address is unique [**?**]. A 48-bit MAC address consists of 6 bytes (48 bits) and is usually expressed as six pairs of hexadecimal digits. The address is divided into two sections, as shown in Figure 1. The three bytes at the left side represent the *Organizationally Unique Identifier* (OUI) which is assigned by the IEEE to a NIC card manufacturer. The remaining three bytes represent the *Universally Administered Address* (UAA) which is assigned by the manufacturer to uniquely identify the NIC.

MAC spoofing is a technique of masquerading a MAC address. It can be used for some legitimate purposes, including protection of user privacy by hiding their real MAC addresses in the wireless network, troubleshooting network

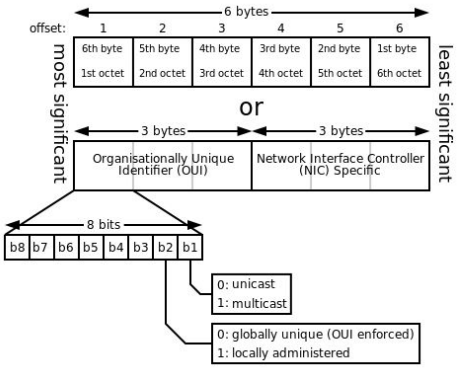

Figure 1: Structure of 48-bit MAC address

problems, ...etc. However, MAC spoofing can be exploited by an attacker to launch various forms of attacks, such as Denial of Service [**?**], session hijacking, access point (AP) / station (STA) spoofing, Man In The Middle (MITM).

Impersonating the MAC address of a legitimate user poses a big challenge for cyber crime investigators. Indeed, MAC spoofing makes the task of identifying the source of the attack very difficult. Therefore, identifying the presence of anomaly of having spoofed MAC address is of crucial importance. Sequence number analysis is a common technique for MAC spoofing detection. Existing sequence-based detection techniques [**?**, **?**] rely on the assumption that a legitimate device produces a linear sequence of sequence numbers. Thus, those techniques check if the gap between consecutive sequence numbers exceeds a threshold, which could be a sign of a MAC spoofing attack. However, this approach may lead to a high rate of false alerts due to lost or duplicated frames. To overcome the limitations of threshold-based approach, we propose a detection method that relies on a machine learning approach, namely Artificial Neural Network (ANN). ANNs provide the potential to identify and classify network behavior from limited, noisy, incomplete and non-linear data sources [**?**]. The high-speed processing of a large amount of information is another advantage of ANNs [**?**].

The rest of the paper is organized as follows. Section 2 describes existing MAC spoofing detection methods. Section 3 presents the principle of artificial neural networks. Section 4 details the proposed detection method. Implementation details and validation results are presented, respectively, in Section 5 and Section 6. Finally, Section 7 concludes the paper.

2. MAC SPOOFING DETECTION TECHNIQUES

2.1 OUI-based Spoofing Detection

This technique checks if the OUI part of a MAC address is a valid identifier assigned by IEEE. However, most algorithm used to generate MAC addresses take into account this constraint, which allows the attacker to easily evade detection.

2.2 Fingerprinting-based Spoofing Detection

This method relies on the fact that the behavior of each NIC has unique technical characteristics, which can be used to create the NIC's fringerprint. The detection approach checks that NIC's fingerprint corresponds to the manufacturer address. However, this method is hard to set up; collecting the NICs fingerprints is time-consuming, not to mention the fact that the behavior study of different NICs is needed.

2.3 Signal Strength-based Spoofing Detection

Information on the received signal strength (RSS) can be extracted from the radiotap header. The signal strength varies depending on the distance from the access point (AP), the presence of obstructions between the AP and the device, as well as the transmit power of the WiFI NIC [**?**].

The basic idea of the detection method is to monitor changes in the RSS during a session. Indeed, a wireless device does not often change its transmission power, thus a sudden change in RSS values from the same MAC address is a sign of a possible spoofing attack.

The downside of this method lies in the fact that the electronic components of the WiFi antenna have a quite unstable behavior, which may lead to an increase in the number of false positives.

2.4 Sequence Number-based Spoofing Detection

This method analyzes the sequence number (SN) field in the MAC header of 802.11 management and data frames. This number is assumed to increase monotonically as it should be incremented by one for every outgoing data and management frame, starting at 0, modulo 4096. Since the presumed linear behavior of the SN, any abnormal SN gaps within the frame sequence from the same MAC address can be a sign of a spoofing attack [**?**, **?**]. However, this approach may lead to a high rate of false alerts due to lost or duplicated frames. To overcome the limitations of threshold-based approach, we propose a detection method that relies on a machine learning approach, namely Artificial Neural Network (ANN).

3. ARTIFICIAL NEURAL NETWORK PRINCIPLE

An Artificial Neural Network (ANN) is an information processing paradigm simulating the way biological nervous systems process information [**?**]. An ANN consists of a set of neurons that are highly interconnected and convert a set of inputs to a set of desired outputs [**?**]. ANNs have the ability of learning-by-example and generalization from learned data, which make them able to detect unknown and even variation of known attacks. ANNs provide the potential to identify and classify network behavior from limited, noisy, incomplete and non-linear data sources [**?**]. The high-speed processing of a large amount of information is another advantage of ANNs [**?**].

Neurons are partitioned into different layers, with an *input* layer, an *output* layer, and several intermediary layers called

hidden layers.

The first step in implementing an ANN consists in obtaining data. During the learning phase, this information allows to learn about normal system's behavior or types of attacks. During the operation phase, they allow the detection of an attack. Nevertheless, the challenge is to identify the amount of data required for a correct model; that is to achieve a desired error rate. Moreover, selecting the data to be considered will depend on the type of learning [**?**].

Achieving the detection of an attack using an ANN is equivalent to implement a classification algorithm with two possible outputs; attack present or absent. The first step in this process is to create a training set containing traffic traces labeled as normal or malicious. Those traces contain traffic captured over several simulations conducted in a test environment. The inputs to the defined ANN is a set of features identifying the attack's behavior. The features are extracted from the traffic traces. The training set is split into two parts: the first part, representing almost 75% of the dataset, is dedicated to the learning phase. The remaining 25% data is dedicated to the testing phase whose aim is to assess the effectiveness of the algorithm in terms of detection accuracy.

4. INTELLIGENT DETECTION OF MAC SPOOFING ATTACK

To detect the MAC spoofing attack, the process begins by extracting the required features from the traffic traces. These features are the sequence numbers of frames sent by each node for a given BSSID. The extracted features are saved in a matrix with N lines, where N is the number of active nodes within the network. The ith line contains an ordered list of frame sequence numbers sent by node i. This step corresponds to lines $5-9$ in algorithm given in Figure 2.

According to experiments, we noticed that an entry with less than 100 sequence numbers yields to an inconclusive result regarding the presence or absence of MAC spoofing attack. Thus, we decided to keep in the matrix only lines with more than 100 frame sequence numbers (See lines $12-14$ of pseudo-code in Figure 2)

To maintain the sequential dependency between sequence numbers, each list of frame sequence numbers is converted into a list of gaps between consecutive sequence numbers (See lines $20-31$ of pseudo-code in Figure 2). Let's $gaps[j]$ denotes the sequence number gap between the jth and the $j-1$th frames. Since s sequence number is modulo 4096, the value of $gaps[j]$ will fall within the range $[0-4095]$. Afterward, the distribution of calculated gaps is determined by computing the percentage of each possible gap value as shown in line 37 of the pseudo-code.

Finally, the calculated percentages are fed to the ANN's input layer. This step corresponds to lines 39 of the pseudo-code given in Figure 2. Two values y_1 and y_2 will be returned through the ANN's output layer, where y_1 is the percentage that a machine has been spoofed and y_2 is the percentage that this machine has not been spoofed.

The pseudo-code of the proposed detection technique is given in Figure 2.

```
1  Extract_features(file.cap, BSSID):
2          Matrix nodes_SNs;
3
4          #Extracting the sequence numbers from the traffic trace file
5          for packet in file.cap:
6                  if packet.address not in nodes_SNs:
7                          nodes_SNs.add_nodes(packet.address)
8
9                  nodes_SNs[packet.address].add_SequenceNumber(packet.SN)
10
11         #Deleting nodes that don't have enough packets to give a conclusive result
12         for node in nodes_SNs:
13                 if len(node.SNs) < 100:
14                         delete(node)
15
16         Matrix Gaps;
17
18         #Convert sequence numbers to gaps
19
20         Gaps = Sequence_number_to_gaps(nodes_SNs)
21
22         return Gaps
23
24 Sequence_number_to_gaps(SNs):
25
26         matrix gaps;
27         for SN in SNs: # for each node in all nodes
28                 for number in SN: # for each sequence number
29                         gaps[i] = SN[i-1] - SN[i]
30
31         return gaps
32
33 Compute_distibution(gaps):
34         #This function returns the percentage of every possible element (0-4095)
35     #of the gaps list
36
37         distribution = percentage(gaps)
38
39         Neural_network.run(gaps)
```

Figure 2: Pseudo-code of the proposed MAC spoofing detection method

5. ANN IMPLEMENTATION

A C function was implemented to extract features from network traffic traces. The function takes a capture file as input and return an .CSV file containing gaps between sequence numbers relating to each node present in the capture. The .CSV file is then parsed by a python module to calculate the distribution of gaps. The result is finally passed to the ANN.

We used FANN (Fast Artificial Neural Network) library to implement the proposed ANN. FANN is a C library which implements multilayer artificial neural networks. FANN provides Python bindings that allow interaction with the FANN library from Python. The library offers unmatched execution speed, instantaneous results, and low memory use.

6. PROOF OF CONCEPT

As a proof of concept, we conducted experiments in an IEEE 802.11 network testbed deployed in a real-office environment. The network comprises one AP, three genuine nodes (STA1, STA2, STA3), two malicious nodes (STA4, STA5) spoofing the MAC addresses of STA2 and STA3, respectively. The network contains also a monitor node to capture traffic. The monitor's wireless NIC is set to operate in promiscuous mode in order to capture all frames exchanged within the network.

Using the created data set, we tried to optimize the ANN performance by adjusting the different parameters, namely: number of neurons within hidden layer, tolerable error rate, and maximum number of iterations. In fact, this tuning stage is very important to reduce the rate of false positives and negatives. The values of those parameters were set to 7, 0.0001 and 1000000, respectively. These values were chosen after several experiments and adjustments.

Around 175405 frames from the AP were captured and analyzed. The analysis revealed that 146 data frames were sent

using STA1's MAC address, 802 data frames were sent using STA2's MAC address, and 18094 data frames were sent using STA3's MAC address.

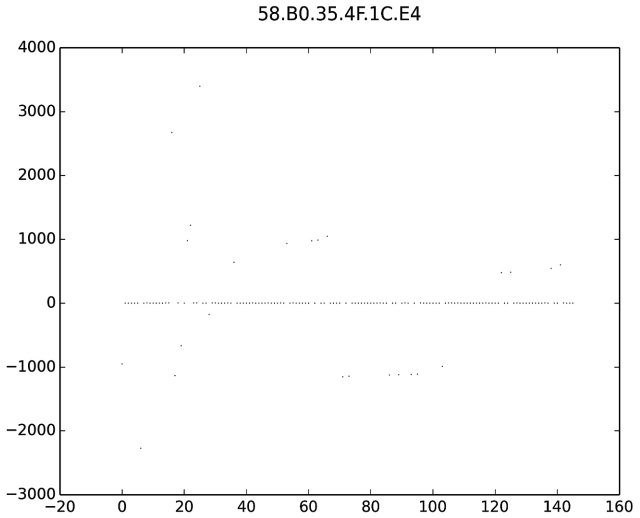

Figure 3: The pattern of gaps between the sequence numbers of frames sent using the STA1's MAC address (58.B0.35.4F.1C.E4)

In Figure 3, the gap between consecutive sequence numbers of frames sent using the STA1's MAC address (58.B0.35.4F.1C.E4) are shown over time. The depicted results show that 83% of gaps are equal to 1, demonstrating a normal behavior. Indeed, the ANN detected that STA1's MAC address is not spoofed with a probability close to 100%. Note that some gap values are greater than 1, which could be due to lost frames. Note also that some gap values are negative, revealing that some frames from the AP are transmitted out of order.

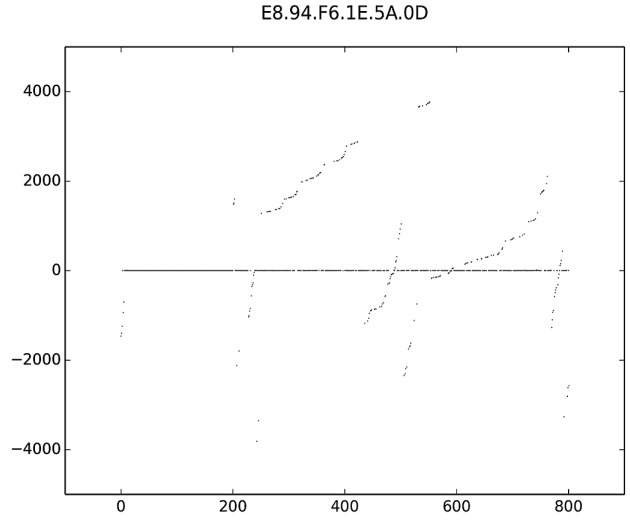

Figure 4: The pattern of gaps between the sequence numbers of frames sent using the STA2's MAC address (E8.94.F6.1E.5A.0D)

Figure 4 depicts the gap between consecutive sequence numbers of frames sent using the STA2's MAC address (E8.94.F6.1E.5A.0D). The results show a non-negligible number of gap values different from 1, which could be a sign of a spoofing attack. Recall that STA2's MAC address were spoofed by STA4. In fact, STA2's MAC address was detected spoofed with a probability close to 97%.

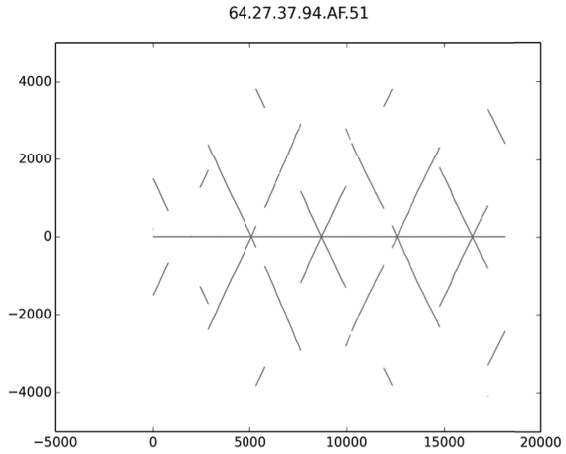

Figure 5: The pattern of gaps between the sequence numbers of frames sent using the STA3's MAC address (64.27.37.94.AF.51)

Finally, Figure 5 shows the gap between consecutive sequence numbers of frames sent using the STA3's MAC address (64.27.37.94.AF.51). Unlike Figure 4, the sign of a spoofing attack is much more apparent here. We found that this is due to the high activity of malicious node STA4, which was spoofing the STA3's MAC address. Indeed, the higher the activity is, the higher the gaps will be as it is hard for a malicious node to predict the next legitimate sequence number. STA3's MAC address was detected spoofed with a probability close to 98%.

In addition to high detection accuracy, the implemented ANN system provides a fast response time of the order of seconds.

Comparing the three graphs depicted in Figures 3, 4, and 5, we can notice the importance of a user-friendly graphical representation of information to support the interpretation of quantitative results. Using a visual tool can swiftly expose anomalies and amplifies cognition of investigators by tacking advantage of their human perceptual capabilities.

7. CONCLUSIONS

In this paper, we have proposed a MAC spoofing detection method that relies on Artificial Neural Networks (ANNs). ANNs provide the potential to identify and classify network behavior from limited, noisy, incomplete and non-linear data sources. The high-speed processing of a large amount of information is another advantage of ANNs. The proposed method was validated through prototype implementation. The validation results demonstrated that the proposed method achieves high detection rate for both legitimate and spoofed MAC addresses. The conducted research showed the importance of a user-friendly graphical representation of information to support the interpretation of quantitative results.

The proposed detection method can be integrated as a module in a wireless network forensic analysis framework, to allow swift and effective detection of MAC spoofing attack.