WISE GUARD

MAC Address Spoofing Detection System for Wireless LANs

Kai Tao, Jing Li and Srinivas Sampalli
Faculty of Computer Science, Dalhousie University, Halifax, Nova Scotia, B3H 1W5, Canada

Keywords: 802.11 wireless networks, access point parameters, intrusion detection system, MAC address spoofing, operating system fingerprinting, received signal strength, sequence number tracking.

Abstract: MAC (Medium Access Control) address spoofing is regarded as an important first step in a hacker's attempt to launch a variety of attacks on 802.11 wireless LANs. Unfortunately, MAC address spoofing is hard to detect. Most current spoofing detection systems mainly use the sequence number (SN) tracking technique, which has drawbacks. Firstly, it may lead to an increase in the number of false positives. Secondly, such techniques cannot be used in systems with wireless cards that do not follow standard 802.11 sequence number patterns. Thirdly, attackers can forge sequence numbers, thereby causing the attacks to go undetected. We present a new architecture called WISE GUARD (Wireless Security Guard) for detection of MAC address spoofing on 802.11 wireless LANs. It integrates three detection techniques – SN tracking, Operating System (OS) fingerprinting and tracking and Received Signal Strength (RSS) fingerprinting and tracking. It also includes the fingerprinting of Access Point (AP) parameters as an extension to the OS fingerprinting for detection of AP address spoofing. We have implemented WISE GUARD on a test bed using off-the-shelf wireless devices and open source drivers. Experimental results show that the new design enhances the detection effectiveness and reduces false positives, in comparison with current approaches.

1 INTRODUCTION

The popularity of wireless local area networks (WLANs) has increased tremendously with the advent of IEEE 802.11 series of standards (IEEE Wireless LAN Standards, n.d.). IEEE standards for WLANs include 802.11a, 802.11b, the currently popular 802.11g, and the upcoming high bandwidth standard 802.11n. These can be used in conjunction with 802.11e for quality of service and 802.11i for security.

The architecture of an IEEE 802.11 WLAN in the infrastructure mode consists of a number of wireless stations communicating with an access point (AP). The AP and the set of stations within its radio range form a basic service set (BSS). Several APs can be interconnected together by means of a distribution system (DS) to form an extended service set (ESS). The distribution system is typically, but not necessarily, a wired LAN. Two identifiers are significant in the architecture: the ID of the BSS or BSSID, which is the MAC address of the AP serving the BSS, and the ID of the ESS or ESSID, also known as SSID, which is a character string given to the WLAN by the administrator.

MAC address spoofing is considered as an important first step in an intruder's attempt to launch a variety of attacks on 802.11 WLANs, such as sniffing, session hijacking, man-in-the-middle, data modification and denial of service. MAC addresses are 48-bit hardware addresses. The manufacturer intending to produce network cards needs to obtain a three-byte organizationally unique identifier (OUI) from IEEE to be used as prefix for the MAC addresses of their products, which makes these addresses globally unique for all LAN-based devices in use today. Randomly generated MAC addresses are easy to detect by filtering through the OUI prefix information lists. However, adversaries can easily sniff legitimate MAC addresses on the wireless LAN and spoof these addresses. Many wireless cards permit setting/changing the MAC addresses by software, thereby enabling easy spoofing.

Since MAC address spoofing is such a threat to WLANs, many wireless developers have designed Intrusion Detection Systems (IDSs) to detect the MAC address spoofing. Most current IDSs – both

Tao K., Li J. and Sampalli S. (2007).
WISE GUARD - MAC Address Spoofing Detection System for Wireless LANs.
In *Proceedings of the Second International Conference on Security and Cryptography*, pages 140-147
DOI: 10.5220/0002123601400147

commercial and open source – use sequence number tracking to detect MAC address spoofing. Unfortunately, this technique has a number of drawbacks. Firstly, it may lead to an increase in the number of false positives in attack detection. Secondly, such techniques cannot be used in systems with wireless cards that do not follow 802.11 sequence number patterns. Thirdly, attackers can forge sequence numbers thereby causing the attacks to go undetected.

We propose a new architecture called WISE GUARD (Wireless Security Guard) for detection of MAC address spoofing on 802.11 wireless LANs. It integrates three detection techniques - Sequence Number Analysis, Operating System (OS) fingerprinting and tracking and Received Signal Strength (RSS) fingerprinting and tracking. It also has a novel AP fingerprinting and tracking as an extension to the OS fingerprinting and tracking mechanism. We have implemented WISE GUARD on a test bed using off-the-shelf wireless devices and open source drivers. The algorithm used in our system integrates the three detection techniques and reduces the false positives. The result shows that, while compared with WiFi-Scanner and Snort-Wireless, WISE GUARD has more accurate information and has less false positives. It should be noted that although the focus in our paper is on detection of MAC address spoofing in 802.11 wireless LANs, WISE GUARD is built on top of Snort-Wireless and can be extended to detect other kinds of attacks such as ICMP (Internet Control Message Protocol) flooding or buffer overflow attacks as well.

The paper is organized as follows. Section 2 introduces the background and related work. Section 3 presents the design and implementation of WISE GUARD. Section 4 gives the experiments and results of testing WISE GUARD. The final section provides concluding remarks.

2 BACKGROUND

2.1 Sequence Number Tracking

Sequence Number Tracking is a recent technique (Wright, 2003) for MAC address spoofing detection in wireless networks, and is popularly used in many IDSs. The IEEE specification (IEEE Wireless LAN Standards, n.d.) defines the sequence number as a 12-bit field indicating the sequence number of an MSDU (MAC service data units) or MMPDU (MAC management protocol data unit). Figure 1 shows the 802.11 frame header with an expanded sequence control field. Sequence numbers are assigned from a single modulo 4096 counter, starting at 0 and incrementing by 1 for each MSDU or MMPDU. The SN remains constant in all retransmissions of an MSDU or MMPDU. The fragment number is always zero unless the frame is a fragment of a larger packet. Without the ability to control the firmware functionality of wireless cards, and without the source code to develop custom firmware, an attacker does not have the ability to alter the value of the sequence control field in the 802.11 headers.

Most current IDS's – both commercial and open source – rely on tracking of sequence numbers for MAC address spoofing detection. Spoofing is said to have occurred when a jump in the sequence number is recorded (Haidong et al., 2004).

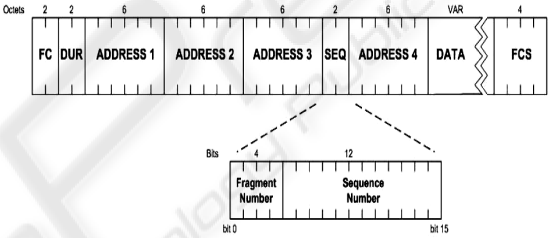

Figure 1: Format of 802.11 Frame Header (IEEE Wireless LAN Standard, n.d.).

2.2 OS Fingerprinting

Operating System (OS) fingerprinting technique was first proposed by Arkin (Arkin, 2000) and has been widely used by security professionals for mapping remote OSs on wired networks. There are two types of OS fingerprinting techniques: passive and active. Passive fingerprinting is the practice of determining a remote operating system by sniffing network packets without actively sending probes to any host while active fingerprinting is accomplished by sending carefully crafted packets to the target machine and analyzing the response that can be measured and compared to known fingerprints.

OS fingerprinting can serve as a useful tool in wireless LANs since many hacking tools only support Linux or FreeBSD systems. Thus, most attacks can only be launched from Linux or FreeBSD systems. According to (Arkin, 2000), wireless stations with different OSs have different features when they generate TCP, UDP, ARP and ICMP packets, and the OS can be detected by analyzing these features. OS fingerprinting and tracking has been used for the first time in this paper as one of the parameters for WLAN MAC address spoofing detection. In our design, we use "passive SYN-based OS fingerprinting" to track the wireless stations' OS fingerprints. This type of OS signature analysis is similar to the one used in P0f (Zalewski, n.d.).

Like SN tracking, passive OS fingerprinting and tracking also has its drawbacks if used alone. Firstly, management frames and control frames in 802.11 WLANs do not provide OS fingerprints. Secondly, in some cases, MAC address spoofing does not need to be run on Linux system. For example, in Windows system with service pack 2, the MAC address of the NIC can be changed with only a simple configuration. Thirdly, passive OS fingerprinting relies on SYN packets. However, during the network communication wireless stations do not always generate SYN packets, especially when the station is under passive monitor mode or when it only sends out management frames. In these cases, no OS fingerprints are tracked, thus no alert will be triggered by the IDS.

2.2 RSS Fingerprinting

Received Signal Strength (RSS) has been widely used in indoor geographical location (geolocation) and positioning systems in wireless LANs (e.g. the RARDA system of Microsoft (Bahl and Padmanabhan, 2000)). Some Intrusion Detection and Response Systems, as described in (Interlink Networks, 2000), use RSS to pinpoint the unauthorized 802.11 wireless station and APs. However, RSS itself has never been used as fingerprints of 802.11 wireless devices for the purpose of intrusion detection. It is very hard for an attacker to modify the signal strength of his or her wireless devices during the network transmission. For this reason, the RSS at the physical layer is a good signature or fingerprint for both wireless station and AP in IDS. There are four units of measurement to represent the RF signal strength (Bardwell, n.d.): mW (milliwatts), dB (decibels) and RSSI (Received Signal Strength Indicator), and a percentage measurement.

RSSI in IEEE 802.11 standard (IEEE Wireless LAN Standards, n.d.) is an integer value between 0-255 (a 1-byte value). No vendors have actually measured 256 different signal levels in their wireless devices, so each vendor's device will have a specific maximum RSSI value ("RSSI_Max"). For example, Cisco has 101 separate RSSI values for RF energy, and their RSSI_Max is 100. RSSI is internally used by the microcode on the wireless adapter or by the device driver. Roaming Threshold is the point when the wireless station is moving away from the AP and the received signal drops to a somewhat low value, which indicates the wireless station is roaming.

Different vendors use different RSSI values for the Roaming Threshold, and those threshold values are seldom released. We tested the roaming threshold of Cisco Aironet 1200 Wireless AP, and the RSSI value is around 97, which we implemented in our test bed as the roaming threshold for wireless station RSS fingerprinting and tracking. Some protocol analysis tools, such as AiroPeek (Airopeek, n.d.), measure RSSI as a percentage of RSSI_MAX.

2.3 Current IDSs

Many open source and commercial IDSs are available today. Snort-Wireless (Snort Wireless, n.d.) is a "lightweight", rule-based and real-time network IDS under UNIX OS. It is popular because of its open source, which can be customized for new detection by writing Snort rules or by adding new preprocessors and detection rules to reflect the latest attacks and exploits. Snort-Wireless adds several new features for 802.11 IDS functionality to the standard Snort distribution. These features allow one to specify custom rules for detecting specific 802.11 frames, rogue access points and Netstumbler like behavior (Wright, 2003). In order to accomplish this, Snort's rule engine has been augmented with support for Wi-Fi. The remaining features are implemented as preprocessors that can be configured and customized as desired according to the different requirements.

WiFi-Scanner (WiFi Scanner, n.d.) is an identification scanner program under UNIX OS. It changes the channel periodically, tries to find any received frame on every channel, and displays them. It uses the SN Tracking techniques discussed in the previous section as well as the timestamp fingerprinting technique for intrusion in the WLAN.

AirDefense (Air Defense Enterprise, n.d.) is a complete hardware and software system consisting of sensors deployed throughout the network, which are interfaced to a management appliance and administered by a management console. Their starter kit provides five sensors and can guard up to ten APs. AirDefense detects intruders and attacks and also diagnoses potential vulnerabilities in the network like mis-configurations.

Aruba Wireless Networks (Aruba Networks, n.d.) has released a complete software and hardware system consisting of switches, APs and its monitoring software. It is the first company to announce the installation of a secure wireless network based on the recently ratified 802.11i standard. One feature of Aruba networks is the ability to “lock the air” using wireless intrusion detection technology built into every Aruba switch and AP.

3 WISE GUARD

We propose a layered architecture called WISE GUARD. It uses off-the-shelf wireless devices and is built on the open source Linux drivers. WISE GUARD integrates OS and RSS fingerprinting and tracking techniques with SN tracking for MAC address spoofing detection. In addition, AP fingerprinting and tracking is used as an extension of OS fingerprinting and tracking to detect AP address spoofing. WISE GUARD can be a standalone solution to the MAC address spoofing detection or be integrated into large wireless IDSs like Snort. WISE GUARD can also be used to advantage in a wireless environment that is WEP- or WPA-enabled due to the fact that these methods can also be subject to MAC address spoofing since there is no authentication or encryption to protect MAC addresses.

3.1 Layered Architecture

The architecture of WISE GUARD, shown in Figure 2, integrates three techniques, which target different layers of the protocol stack in the detection engine: OS fingerprinting, Sequence number tracking and RSS fingerprinting. We also include the fingerprinting of Access Point (AP) parameters as an extension to the OS fingerprinting for the detection of AP address spoofing.

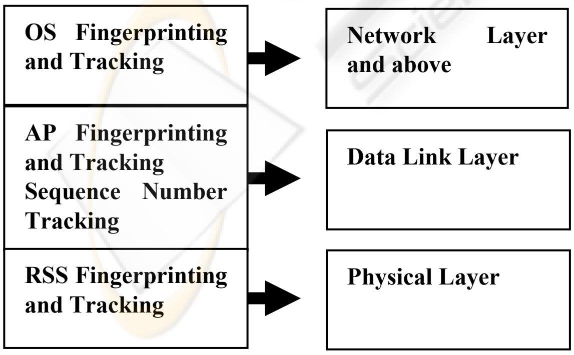
OS Fingerprinting and Tracking
Network Layer and above
AP Fingerprinting and Tracking Sequence Number Tracking
Data Link Layer
RSS Fingerprinting and Tracking
Physical Layer

Figure 2: Layered Architecture.

As mentioned earlier, at the network layer, we use “passive SYN-based OS fingerprinting” (Zalewski, n.d.) to track the OS fingerprints of wireless stations. However, this technique cannot deal with the situation when the wireless stations and APs only have management frame transmission. Hence we propose a new AP fingerprinting and tracking technique to extend OS fingerprinting and tracking.

This technique includes the fingerprints of Timestamps, Capability Information, Traffic Indication Map, and Tag Information (Vendor Information) in management frames. Here is the description of these fields (IEEE Wireless LAN Standards, n.d.).

Timestamp: The timestamp in the beacon frame is a 64-bit field counting in increments of microseconds. After receiving a beacon frame, a wireless station uses the timestamp value to update its local clock. This process enables synchronization among all stations that are associated with the same AP. So the timestamp is like the system clock of AP, it is very hard to spoof.

Capability Information: This signifies the requirements of wireless stations, which wish to belong to the wireless LAN that the beacon represents. For example, this information may indicate that all stations must use wired equivalent privacy (WEP) in order to participate on the network.

Traffic Indication Map (TIM): An AP periodically sends the TIM within a beacon to identify which stations using power saving mode have data frames waiting for them in the access point's buffer. The TIM identifies a station by the association ID that the access point assigned during the association process. We can set this value when configuring the AP.

Tag Information: This field includes the information about tag length and not interpreted vendor specification.

Tracking the change both the APs’ and wireless stations’ OS fingerprints can be done passively without generating additional traffic to the network.

At the data link layer, we still use the SN tracking technique. The rogue AP’s SN and the legitimate AP’s SN usually have a large gap, because the rogue AP and the legitimate AP turn on at different times. However, it is still possible for a rogue AP to have an SN similar to that of a legitimate AP in a short period, because the SN will restart from 0 after it reaches 4096. So the SN tracking technique has a threshold that indicates the times it can tolerate when the SN gap over a designated value. This value is called the “tolerate gap”. However, the retransmitted frames have a gap of 0. In Snort-Wireless, the retransmission frames is regarded as abnormal frames from rogue AP by using the formula:

Tolerate Gap = ((Current SN value – Previous SN value) + 4095) mod 4096

This approach eliminates the possibility for a hacker to spoof the SN number, but it will alert either AP has retransmitted frames or its SN value in the current frame is smaller than the SN value in previous frame, thus leading to false positives.

In our design, we changed this SN tracking scheme of Snort-Wireless. We assume that the possibility for a hacker to spoof the SN number is not high (even if he can, we still have the detection technique from other layers), and regard the retransmitted frames as legitimate frames from the AP. Thus if the retry bit is equal to 1 and the Tolerate Gap from the formula above is equal to 4095, we set the Tolerate Gap to 0. We also used the absolute value of the difference between current SN value and previous SN value to bypass the case of occurrence of the smaller SN value because of the transmission delay.

At the physical layer, we use RSS Fingerprinting and Tracking technique. Both (Bahl and Padmanabhan, 2000) and (Bahl et al, 2000) have established an indoor radio propagation model for its geolocation system, which indicate that RSS has some relationship, not linear, with the distance of the wireless devices. This may not be useful for distinguishing wireless stations and attackers. For example, if they are both on the edge of a circle, they will have same distance to the sensor, thus have the same signal strength, according to the propagation model. However, this is useful for detection of rogue APs. If we set the sensor right beside the AP, when the rogue AP turns on and is approaching the WLAN, the sensor can tell the difference immediately from the RSS.

Another reason we set up the sensor near the legitimate AP is because we can monitor the wireless station for roaming. If the RSS from a certain wireless station is going below the “Roaming Threshold”, the sensor will inform the Sequence Number tracking model and reset the tracking pattern. In this way, we can reduce the number of false positives of the SN tracking alert.

However, using RSS fingerprinting tracking alone is not accurate. According to the indoor Radio Propagation model in (Bahl et al, 2000), the signal propagation is dominated by reflections, diffraction, attenuation, and scattering of radio waves caused by structures within the building, e.g., when people moving in front of the AP will change the signal received by wireless stations, even when the wireless node is stationary.

3.2 Detection Components

There are three detection components in our design: *Sensor*, *Analyzer* and *Alert*. For a large-scale wireless network, the deployment should be centralized; with Sensors deployed all over the network to send back captured packets to a central server over a separate network, where the Analyzer and alert components are located. This separate network could typically be a secured wired network, e.g. a Virtual Private Network (VPN), and hence the detection traffic does not reduce the bandwidth of the wireless LAN. Furthermore, communication between the Sensor and the central server is secure. When frames with abnormal signatures are detected, the Analyzer will trigger an alert to be sent to the central server through the backbone and the Alert component decides the level. The alert can be simply displayed on the console of the central server or sent to the administrator by an e-mail or a page message.

In a small-scale wireless network, the three components can be integrated into one AP or into a standalone wireless device (acting like a sniffer).

3.3 Design Prototype

Our design is an extension of the Snort-wireless architecture. Snort has the three components that we require. However, it cannot intercommunicate between detection preprocessors or plug-ins, although Snort has defined pass rules, log rules and alert rules to tell the detection engine how to deal with a packet when rules have conflicts. We have added a postprocessor to process the outcome of the detection preprocessors or plug-ins and give a probability evaluation on the incoming packets. The parameter sets have the new OS fingerprinting feature values, RSSI tolerate gap, threshold values, authorized AP and wireless station lists or other parameters to initiate the detection engine.

The design prototype of WISE GUARD is shown in Figure 3. Like most of the WLAN discovery tools, Snort is built on widely available open source Linux drivers – HostAP (Malinen et al, n.d.) for 802.11 network cards utilizing the PRISM chipset. On top of the driver, the frame capture and decoder uses the libpcap or other open source Linux libraries to find and decode all the captured frames.

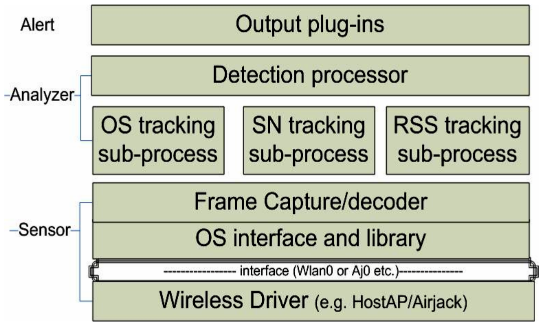

Figure 3: Design Prototype of Wise Guard.

After the frames have been decoded, the RSS, OS and Sequence Number fingerprints will be retrieved and sent to the detection processor (with three sub-processes), which is the core of the Analyzer. Results or outcomes will be generated between each sub-process. The main process waits and processes these outcomes, and then generates and sends out the status code based on the outcomes to the output plug-ins. The output plug-ins acts as the Alert; it can be an interface to any alert applications like e-mail system, page system or log system.

Based on different status codes, the output console will give three levels of alert as the report: high, medium, and low. In our implementation, we translate the status code into scores, which are used to calculate the indicators as percentages. For example, timestamp feature has a score of 5 out of the total score value 20. The administrator can decide the conversion scale between the scores and the alert levels.

Table 1: Status Code Descriptions for AP Detection.

Status Code	OS tracking		SN tracking		RSS tracking		Description	Level Of Alert
	0-normal	1-abnormal	0-normal	1-abnormal	0- in range	1- roam out		
000	0		0		0		Signal is good, no MAC spoofing	No Alert
001	0		0		1		Signal is weak, no MAC spoofing	No Alert
010	0		1		0		MAC address spoofing in progress, attacker using the same OS	Low
011	0		1		1		Wireless station is roaming out, no spoof	No Alert
100	1		0		0		Abnormal wireless station exists in range, packets forge exist	Medium
101	1		0		1		MAC address spoofing may be in progress, attacker is at the edge.	Medium
110	1		1		0		MAC address spoofing in progress, attacker is in range.	High
111	1		1		1		MAC address spoofing in progress, attacker is at the edge.	Medium

Table 1 and Table 2 give the status codes for AP detection and wireless station detection, respectively.

Table 2: Status Code Descriptions for Wireless Station Detection.

Status Code	AP tracking		SN tracking		RSS tracking		Description	Level Of Alert
	0-normal	1-abnormal	0-normal	1-abnormal	0-normail	1-abnormal		
000	0		0		0		No MAC spoofing	No Alert
001	0		0		1		No MAC spoofing (but may be spoofing SN)	No Alert
010	0		1		0		No MAC spoofing (retransmission or other special cases)	No Alert
011	0		1		1		MAC address spoofing in progress, spoofing in management frame	Medium
100	1		0		0		Abnormal AP exists in range, packets forge exist	Medium
101	1		0		1		MAC address spoofing may be in progress, attacker is at the edge.	High
110	1		1		0		MAC address spoofing in progress, attacker is in range.	High

4 EXPERIMENTAL RESULTS

We tested WISE GUARD and two other IDSs, namely, Snort-Wireless and WiFi-Scanner. Figure 4 shows the test bed. WISE GUARD was first launched on the detection server and then Snort-wireless was run on a laptop (K). The attacks were launched from another laptop. To launch the session hijacking attack, the attacker spoofs the BSSID and channel of the AP using the HostAP driver, and brings down the network. When performing the management frame DoS attack, de-authentication and disassociation frames were sent from the attacker’s laptop to the wireless client using the Libradiate tool. The attacks are stopped by resetting the HostAP driver or by terminating Libradiate. WiFi-Scanner was then run on the laptop K to detect the same attack. Furthermore, during the attack, the legitimate station was moved around the AP.

We observed that as soon as the MAC spoofing attack has been launched, WISE GUARD generated NEW ALERT, which indicates MAC address spoofing in progress.

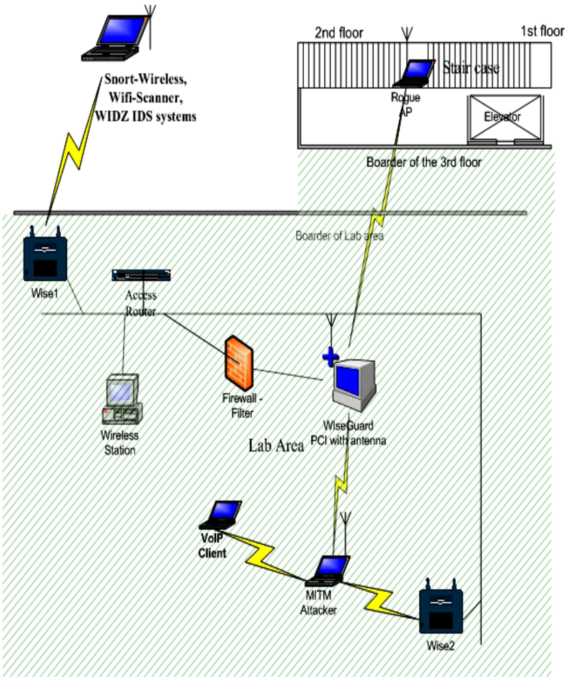

Figure 4: Test bed set up.

The output from WISE GUARD indicates the detection result of three techniques, the MAC address that is under attack, the time of launch of the attack, the level of alert and the percentage Indicator. It also indicates that two or more APs with same BSSID exist and have an abnormal gap in sequence numbers and signal strengths. The alert also displays two values in Capacity Info, TIM, and Tag Info, which were coming from the legitimate AP and the rogue AP, respectively. WISE GUARD generates alert with an increasing percentage indicator, till it reaches 100%. This is because SN and RSS tracking technique have a delay, while OS tracking does not. The increasing indicator shows a high probability of attacks in progress.

Table 3 summarizes the results based on the experiments that we performed using three types of attacks. Each attack was performed ten times, with a duration of 5 minutes each time. A false positive indicates that a legitimate MAC address was reported as being spoofed. A false negative indicates that the spoofed address is not reported. For example, 5/10 means we test the attack 10 times, the tested IDS has not detected the attacks, but during the attack time, it has reported a legitimate MAC address five times. Table 4 gives a summary of the comparisons with Snort-Wireless and WiFi-Scanner and other open source wireless IDS on detecting the three main MAC address spoofing attacks.

5 CONCLUSIONS

We proposed a novel wireless IDS, namely, WISE GUARD, to detect MAC address spoofing in wireless LANs. WISE GUARD integrates three detection techniques – SN tracking, OS fingerprinting and tracking and RSS fingerprinting and tracking. It also includes the fingerprinting of AP parameters as an extension to OS fingerprinting for detection of AP address spoofing. We implemented our system on a test bed using off-the shelf wireless devices and open source drivers. We tested our system and two other existing open source wireless IDSs for detecting session hijacking DoS, management frame DoS and man in the middle attacks. Experimental results show that our system performs better, especially in the effectiveness of detecting MAC address spoofing with less false positives.

Table 3: Summary of Results.

# of Tests	Attack Types	# of the results with false positives/false negatives		
		WISE GUARD	Snort-Wireless	WiFi-Scanner
10	Session Hijacking	0/0	10/0	2/0
10	De-authentication DoS	2/0	10/0	5/10
	Disassociation DoS	1/0	10/0	4/10
10	Man in the middle	2/0	10/0	6/0

Table 4: Comparison with other IDSs.

# of Tests	Attack Types	# of the results with false positives/false negatives		
		WISE GUARD	Snort-Wireless	WiFi-Scanner
10	Session Hijacking	0/0	10/0	2/0
10	De-authentication DoS	2/0	10/0	5/10
	Disassociation DoS	1/0	10/0	4/10
10	Man in the middle	2/0	10/0	6/0

REFERENCES

IEEE Wireless LAN Standards (n.d.), accessed March 2007, from http://standards.ieee.org/

Wright, J., 2003. Detecting Wireless LAN MAC Address Spoofing. Accessed March 2007 from http://home.jwu.edu/jwright/papers.htm

Haidong, X., Brustoloni, J., Mitrou, N., Kontovasilis, K., Rouskas, G., Iliadis, I., Merakos, L., 2004. Detecting and blocking unauthorized access in Wi-Fi networks in *Proceedings of the International Networking Conference*, Greece, May 2004, pp. 795-806.

Arkin, O., 2000. ICMP Usage in Scanning, Sys-Security Group Pub, July 2000, accessed March 2007 from http://www.syssecurity.com/archive/papers/ICMP_Scanning_v1.0.pdf

Zalewski, M., (n.d.) Passive OS fingerprinting tool", accessed March 2007 from http://www.networkintrusion.co.uk/osfp.htm.

Bahl, P., and Padmanabhan, V.N., 2000. Radar: An in-building rf-based user location and tracking system. In *Proceedings of the IEEE Infocom* 2000, Tel-Aviv, Israel, vol. 2, Mar. 2000, pp. 775--784.

Interlink Networks, 2002. A Practical Approach to Identifying and Tracking Unauthorized 802.11 cards and Access Points, *White Paper,* Interlink Networks, Inc., April 2002.

Bardwell, J., (n.d.) "WiFi Radio Characteristics and the Cost of WLAN implementation", *White Paper*, Connect802, accessed March 2007 from http://www.connect802.com/white_papers.htm.

Airopeek (n.d.), accessed March 2007, from http://www.wildpackets.com

Snort-wireless (n.d.), accessed March 2007, from http://snort-wireless.org

WiFi Scanner, (n.d.). Accessed March 2007 from http://wifiscanner.sourceforge.net.

Air Defense Enterprise, (n.d.), Accessed March 2007 from http://www.airdefense.net.

Aruba Networks, (n.d.). Accessed March 2007 from http://www.arubanetworks.com.

Bahl, P., Padmanabhan, V.N., and Balachandran, A., 2000. A Software System for Locating Mobile Users: Design, Evaluation, and Lessons, *Technical report* MSR-TR-2000-12, Feb 2000. Accessed March 2007 from http://citeseer.ist.psu.edu/bahl00software.html.

Malinen, J., and contributors (n.d.). Host AP driver for Intersil Prism2/2.5/3, hostapd, and WPA Supplicant. Accessed March 2007 from http://hostap.epitest.fi/.