Combating Wireless LAN MAC-layer Address Spoofing with Fingerprinting Methods

Güenther Lackner[1], Udo Payer[2], and Peter Teufl[2]

(Corresponding author: Guenther Lackner)

studio78.at, Feuerbachgasse 6, 8020 Graz, Austria[1] (Email: guenther.lackner@studio78.at)
Institute for Applied Information Processing and Communications, IAIK, University of Technology[2]
Inffeldgasse 16a, 8010 Graz, Austria (Email: udo.payer@iaik.tugraz.at)

(Received June 21, 2008; revised and accepted Dec. 4, 2008)

Abstract

Unwanted use of wireless networks has become a well-known problem in recent years. One attempt to solve this problem is the use of access control lists, which are associated with accredited MAC addresses. But since MAC addresses can be spoofed very easily, improved mechanisms are needed to attest the uniqueness of a dedicated wireless station. Today, all known approaches are based on the idea to generate NIC-specific profiles derived from invariant NIC-characteristics. In doing so, unique features are either extracted from RF-components or from the timing behavior of the MAC-chip. To give a review and to classify all proposed approaches, we start with a short introduction to all underlying ideas and will conclude with a comparison of these mechanisms.

Keywords: Wireless network security, MAC address spoofing, fingerprinting

1 Introduction

MAC address spoofing is a synonym for taking over the identity of network interface controllers (NIC). Every single networking device is equipped with a globally unique hardware address called MAC address. The uniqueness of MAC addresses is essential in all phases of network communication because they map all upper-layer identifiers, e.g. IP addresses, to particular network interfaces.

Spoofing a MAC address is basically identity theft and denotes the altering of the MAC address on a NIC [3]. This article is focused on possible attack scenarios due to MAC address spoofing in wireless networks based on IEEE's 802.11 standard, possible countermeasures and their practical applicability.

The exponential growth in the deployment of wireless access networks (WLAN), whether in enterprise or in home environments makes them an attractive target for attackers. Attacks that exploit vulnerabilities at the IP layer or superior network layers can readily be addressed by established intrusion detection systems. This is due to the fact that communications on these layers are independent of the underlying network-architecture. However, exploits involving the IEEE 802.11 link-layer protocol need to be addressed by novel methods and tools. Although next generation WLAN standards and equipment may support link-layer authentication, the vast legacy of currently installed systems will not be replaced in the near future [5, 11].

A common mistake is to believe that Wi-Fi Protected Access (WPA) or IEEE 802.11i (WPA2) can be used to prevent MAC address spoofing in all cases. Actually, WPA and WPA2 can provide data-frame authentication to prevent clients from being spoofed but unfortunately they do not provide authentication for management frames, leaving a significant gap for denial-of-service (DoS) attacks [13].

The methods described in this article are based on the detection of anomalies in different observations. Therefore, we will briefly discuss trivial approaches like sequence number analysis [5, 14]. But the main focus of this article is devoted to methods to generate NIC-fingerprints following three approaches, including one introduced by the authors.

1) Radio Frequency Fingerprinting by Neyanthis Hall et al. [8].

2) Passive Data Link Layer Fingerprinting by Jason Franklin et al. [4].

3) Acknowledge-Frame Delay Fingerprinting by Günther Lackner et al. [11].

Most of the presented approaches provide unsatisfying high false positive and/or false negative rates [4, 11]. Other limitations are due to the need for special purpose hardware to create and evaluate device fingerprints [8].

The remainder of the article is organized as follows. Section 2 describes possible MAC address spoofing attack scenarios. Section 3 describes and analyses spoofing-detection methods based on different approaches. Finally, Section 4 provides a conclusion and analysis of the practical applicability of the described methods.

2 Vulnerabilities and Attack Scenarios

The IEEE 802.11 MAC-layer was especially designed to meet all requirements of a wireless network. In particular the ability to discover networks, and coordinate access to the radio medium. Today, we know that most link-layer attacks on WLAN networks are DoS attacks based on these extended functionalities. These attacks mainly affect the availability of WLAN services - optionally for a dedicated target or the whole network. Sometimes, a DoS is only the first step in a more sophisticated attack that in the worst case could lead to the theft of authentication credentials like usernames and passwords [1]. The next sections describe the vulnerabilities and the attack scenarios that might arise.

2.1 Identity Vulnerabilities and Potential Attacks

As in wired Ethernet networks, all 802.11 nodes implicitly trust a sender's source address. For most WLAN management and control-frames, standard IEEE 802.11 networks do not provide any mechanism for verifying the correctness of the sender's identity. This allows an attacker to spoof other nodes and their messages. This fact leads to several vulnerabilities [1].

2.1.1 De-authentication Attack

To join a wireless network a client has to choose an access point and authenticate itself to it before any further communication may start. This authentication protocol also includes a message that allows nodes to de-authenticate from each other with one single message. Unfortunately this message is in no way protected against spoofing. So anybody can send this message with a forged identity. As a consequence the attacked client will not receive further messages unless it reestablishes authentication. With one single de-authentication message the attacker provokes six messages for the re-authentication between the attacked client and the access point. If this attack is replayed periodically a victim could be kept from joining the network indefinitely [11].

2.1.2 Disassociation Attack

In an environment with multiple access-points available, each client may be authenticated with more than one access point if they overlap. The state of association was introduced to allow the access points to agree who has the responsibility for forwarding packets to the client [1]. As with authentication, one single message allows the client or an attacker to disassociate. Exploiting this vulnerability is functionally identical to the de-authentication attack. The impact is slightly weaker due to the fact that the reestablishment of the association needs less effort than re-authentication [5].

2.1.3 Power-saving Attack

IEEE 802.11 power conservation functions also provide several vulnerabilities. A client, wishing to enter sleep mode, informs the access point (AP) so that it can buffer all inbound traffic for later transmission. Due to the timely synchrony of the clients and the AP all clients in power-saving mode know when to wake up to receive the traffic indication map (TIM). This TIM indicates if the AP has buffered packets for the client. Now the client may wake up and send a poll frame to signal the AP its readiness to receive the buffered packets. This mechanism offers two weak points for attackers. At first it is very easy to trick the AP into discarding the buffered traffic for a client in power-saving mode by simply spoofing the poll frame. Also by forging a TIM frame the client may be told that there are no buffered frames at all and the client will immediately return to power-saving mode. On the other hand an attacker may disturb the timely synchrony and consequently the client will wake up at the wrong time and will never receive a TIM resulting in the disruption of the network service [1].

2.1.4 Access-point Spoofing

Unlike the previous vulnerabilities the following two attacks do not directly rely on flaws in the IEEE 802.11 MAC layer specification but rather in completely faking the AP's identity. If an attacker is able to spoof the identity of an AP he might lure clients into connecting to the fake AP instead of the legitimate one.

The attacker only needs to emit a stronger signal than the legitimate AP. In many cases, public WLANs use web portals for user authentication. The attacker now might redirect the client to a faked web portal and steal the clients username and password. Alternatively the attacker can implement active man-in-the-middle attacks against SSH and HTTPS sessions by exploiting weak bindings in ad-hoc PKI [11]. The usage of IEEE 802.11i security mechanisms with integrated IEEE 802.1X would provide an effective protection against this attack.

2.1.5 Client Spoofing

By spoofing a legitimate wireless station (STA) an attacker may bypass an AP's MAC address-based access control list to gain access to a WLAN. This action is frequently the first step in infiltrating a network and followed by further attacks. Another possibility is to use the AP to decrypt traffic encrypted by WEP. In this attack an attacker impersonates a legitimate STA, captures WEP frames the STA sends, and retransmits these frames to the AP. The destination IP address in the WEP frames

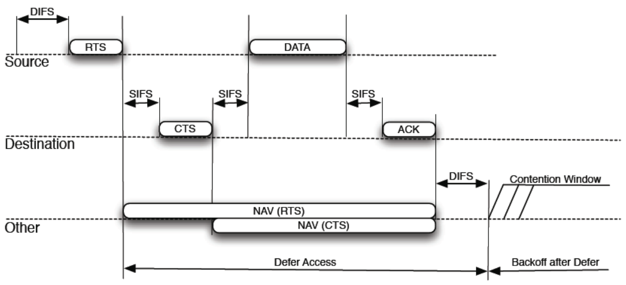

Figure 1: IEEE 802.11 carrier access scheme [8]

may be altered in order to trick the AP in transmitting the decrypted traffic to an Internet host controlled by the attacker [5]. The usage of IEEE 802.11i security mechanisms with integrated IEEE 802.1X would provide an effective protection against this attack.

2.2 Media Access Vulnerabilities

WLAN networks go through significant efforts to avoid transmission collisions. It is not feasible to implement perfect collision detection because of the possibility of hidden clients [2]. IEEE 802.11 implements the so-called distributed coordination function (DCF) which is a *carrier sense multiple access with collision avoidance* (CSMA/CA) technique [11]. It is a combination of physical and virtual carrier-sense mechanisms. Both of these mechanisms may be exploited by attackers [5]. A detailed description of these mechanisms would go beyond the scope of this article.

In the following passage only the basic ideas behind possible attacks are described.

As Figure 1 illustrates, different kinds of time windows are defined to control the carrier access in a WLAN. After a *Distributed Coordination Function Interframe Space* (DIFS) all STAs willing to transmit have to wait for a random time to minimize the risk of collisions. If a collision occurs the sending STA uses the *random exponential backoff algorithm* and retries transmission at a later time. The duration of this waiting period must at least be one *Short Interframe Space* (SIFS).

An attacker may now be able to completely monopolize the channel by sending a packet right before the end of the SIFS period. This approach is highly effective but due to the fact that about 50,000 packets per seconds are necessary to block the channel, the energy costs of this attack are rather high [1].

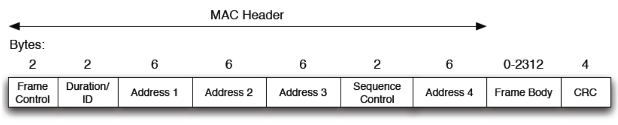

Figure 2: IEEE 802.11 data frame

The second media access vulnerability addresses the virtual carrier-sense mechanism. As can be seen in Figure 2, each IEEE 802.11 data frame contains a *duration field.*

This field is used to determine for how long the channel needs to be reserved in order to complete the upcoming transmission. Each STA uses this value to recompute its *Network Allocation Vector* (NAV). This NAV basically is a counter that, after reaching 0, tells the STA that it is allowed to access the medium.

An attacker now may exploit this feature by asserting large duration field values. The maximum the NAV may reach is 32767 or about 32 milliseconds on an IEEE 802.11b network. This means that an attacker only has to launch this attack about 30 times a second to block the channel completely.

3 State-of-the-Art

Although layer 2 address spoofing does not receive as much public notice as IP address spoofing it poses a permanent and serious threat to WLAN security. Even though recent advancements in IEEE 802.11 standards like IEEE 802.11i (WPA2) contributed some additional measurements for packet authentication, the defense against MAC address spoofing-based attacks has some loopholes [13]. To the best of our knowledge even future standards will not provide a solution for these problems.

In the last years the scientific IT-security community provided several ideas to face the threat of MAC address spoofing. Besides trivial approaches like OUI (Organizationally Unique Identifier) [9] respectively IAB (Individual Address Block) [10] plausibility checks [14] even more sophisticated ideas like sequence number analysis [5], and fingerprinting methods have been developed. Fingerprinting means identifying a device or software only by profiling its externally observable characteristics. The creation of these fingerprints might be complicated if the device under observation actively tries to thwart this effort.

This section will shortly describe the scientific background of three major approaches, the quality of their results and their practical applicability.

3.1 Radio Frequency Fingerprinting in Wireless Networks

his section describes a WLAN fingerprinting method presented by Hall [8] et al. As opposed to the two other methods described later in this work, we did not implement the method due to the lack of hardware which is needed for the approach. Thus, we only give a short overview and refer the reader to [7] and [8] for further details.

3.1.1 Background

The presented fingerprinting technique is based on the signal characteristics of turn-on transients of wireless transceivers. These transients are specific to each different transceiver and thus are perfectly suited as data source

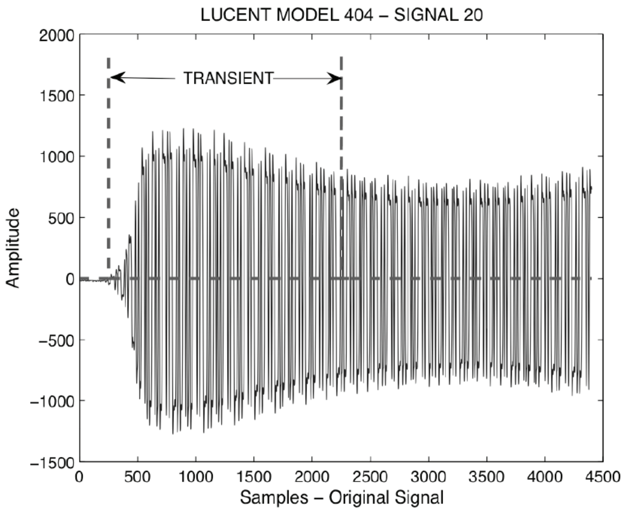

Figure 3: Signal from a 802.11b transceiver [8]

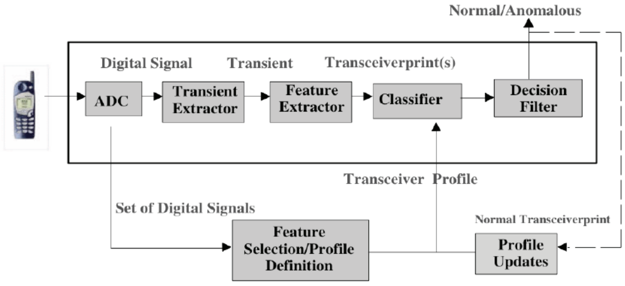

Figure 4: System overview [8]

for fingerprint generation. Figure 3 shows an example for the turn-on transient of an Orinoco chipset. In preceding work [7] the authors describe significant features that are extracted from the turn-on transient and are used for fingerprint creation. Transient capturing and analysis requires a special infrastructure for signal capturing, which is depicted in Figure 4.

The method extracts basic signal components - the DWT (Discrete Wavelet Transformation) coefficients, signal phase and signal amplitude - and generates features used for the classification process (see Figure 5) The extraction and the computation of the features and their further analysis is done with Matlab$^{\mathrm{TM}}$on a standard laptop. The fingerprint for each device is represented by these features. Fingerprint classification is based on a statistical classifier.

3.1.2 Evaluation

Hall et al. evaluated the performance of the fingerprinting method with 30 transceivers. For each transceiver 120 signals were captured and used for the performance evaluation. The results indicate that the method is capable of achieving a very low false positive rate (0% during the evaluation) and a high detection accuracy (95% during the evaluation). However, the biggest disadvantage of this method is the hardware which is needed for signal capturing. This drawback also limits the usability in intrusion detection systems.

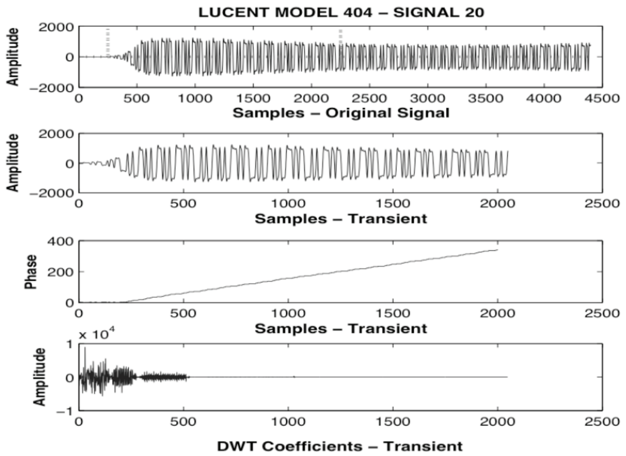

Figure 5: Signal components [8]

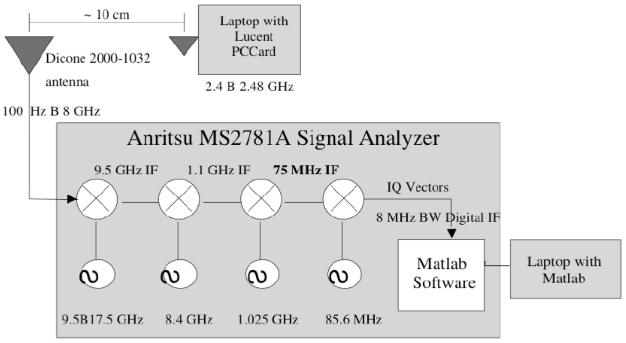

Figure 6: Evaluation setup [8]

3.2 Passive Data Link Layer Fingerprinting

This section describes the wireless NIC fingerprinting approach developed by Jason Franklin and his team, published in 2006 [4]. Franklin identified an imprecision in the IEEE 802.11 Media Access Control specification that was interpreted differently by wireless NIC firmware developers. The following section will explain this flaw and its use for fingerprinting in more detail.

3.2.1 Background

Typically, an activated wireless NIC instantly starts to look around for available wireless networks. This search is performed by broadcasting *probe-request frames*. The IEEE 802.11 standard describes this so-called *active scan function* as follows.

> For each channel, the client broadcasts a probe request and starts a timer. If the timer reaches *MinChannelTime* and the channel is idle, the client scans the next channel. Otherwise, the

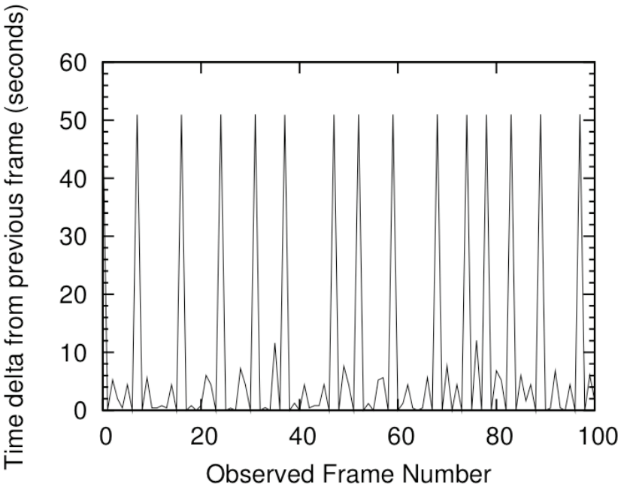

Figure 7: D-Link driver for the D-Link DWL-G520 (802.11b/g PCI wireless NIC [4]

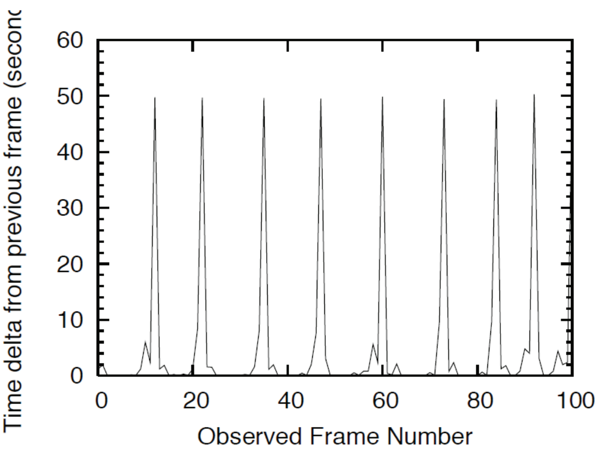

Figure 8: Cisco driver for the Aironet AIR-CB21AG-A-K9 (802.11 a/b/g) PCI wireless NIC [4]

> client waits until the timer reaches *MaxChannelTime*, processes the received probe response frames and then scans the next channel [4].

Due to this loose definition many drivers with slightly different probing techniques have been implemented. Jason Franklin and his team found out that these varieties are externally observable characteristics that allow the creation of fingerprints.

Figures 7 and 8 visualize the time difference between arriving probe frames as transmitted by two different wireless drivers. One can observe unique cyclic patterns with different time deltas between the probe requests for each wireless NIC. Small variations in these patterns which aggravate the creation of good fingerprints are due to two main reasons, packet loss caused by signal interference and the fact that wireless drivers by default constantly circle through all eleven channels in the 2.4 GHz ISM band in search of other access points. The first source of information loss can easily be avoided by using higher gain antennas while the second can be compensated by using statistical methods.

In order to create a fingerprint the presented method needs to *capture the trace* of a wireless NIC. This is done by capturing a series of probe request frames of a specific NIC with a WLAN sniffer. For characterizing the time deltas between the probe requests a binning approach has been chosen. Binning works by translating an interval of continuous data points into discrete bins. A bin is an internal value used in place of the true value of an attribute. The distributions of the observed deltas in these bins of equal size allow the creation of a stable signature [4].

By analyzing this collected data, Franklin et al. identified two attributes from the probing rate that are essential for fingerprinting the NIC respectively its driver. The first attribute is the bin frequency of delta arrival time values between probe request frames that characterizes the size of each bin. The second attribute was the average for each bin, of all actual (non-rounded) delta arrival time values of the probe request frames placed in that bin. This characterizes the actual mean of each bin. The next step was to create a signature for each driver. The authors decided to use a Bayesian model because it is simple and well tested [4].

Franklin et al. were now able to create signatures of 17 different NIC drivers which they named *master signatures*. Unknown signatures can now be compared to the master signatures in order to determine the closest matching NIC driver. This is done by calculating the closest distance between the captured signature and a master signature [4].

Let p_n be the percentage of probe request frames in the n-th bin of the signature T and let m_n be the mean of all probe request frames in the n-th bin. Let S be the set of all master signatures and let s be a single signature in this set. Let v_n be the percentage of probe request frames in the n-th bin of s and let w_n be the mean of all probe request frames in the n-th bin of s. The following equation was used to calculate the distance between the observed signature T and all known master signatures, assigning to C the distance value of the closest signature in S to T [4].

$$C = min(\forall s \in S \sum_{0}^{n} (|p_n - v_n| + v_n|m_n - w_n|)) \qquad (1)$$

3.2.2 Proof-of-Concept

To prove their concept empirically, Franklin et al. have chosen three different evaluation setups. The first two (named Test Set 1 and 2 in Table 1) were used to create the master signatures and evaluate them. These tests have been performed in a laboratory environment. No background traffic or other WLAN activity interfered with the measurement. The authors say that Test Set three (see Figure 9) could be seen as a *real world scenario*.

Table 1: Accuracy of fingerprinting technique by scenario [4]

Testset	Successful	Total	Accuracy [%]
1	55	57	96
2	48	57	84
3	**44**	**57**	**77**

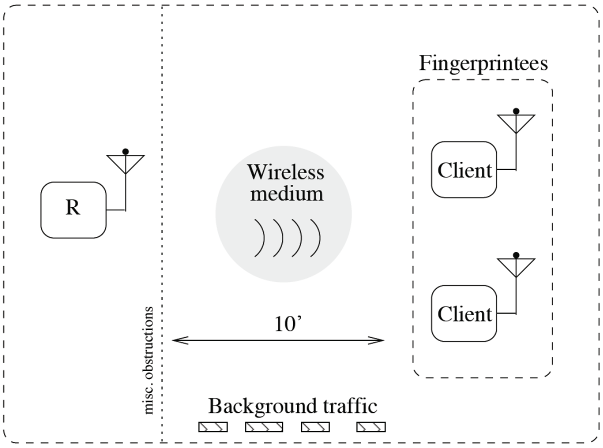

Figure 9: Evaluation setup [4]

These results may be obtained after 30 minutes of trace capturing per NIC. The authors also say that after one minute of scanning the accuracy of their technique is at least 60 % in all test cases [4]. The practical applicability and limitations to this approach are discussed in the next section.

3.2.3 Discussion

Franklin et al.'s approach use a uncertainty in the IEEE 802.11 specification. It is able to classify different firmware versions instead of the underlying hardware. For creating a meaningful fingerprint a large number of probe-requests need to be captured. Typically, a NIC - willing to join a network - usually just needs a hand-full of these requests. Consequently, it could take a rather long time to obtain a suitable amount of data. Another significant draw-back is that fingerprinting may easily be avoided by using passive-scanning or altering the device firmware.

3.3 Acknowledge-frame Delay Fingerprinting

Lackner et al. present a passive fingerprinting technique in [11], which identifies WLAN chipset by analyzing the distribution of delay values between 802.11 packets and the corresponding acknowledgement frames. Related work published by Guenther et al. [6] indicates that these delay values differ from chipset to chipset and thus could be used for chipset identification. The presented technique uses machine learning techniques to classify histograms which are created from delay time values extracted from passively observed WLAN traffic.

3.3.1 Background

Whenever a 802.11 packet is received, the receiver verifies its integrity by calculating a CRC checksum. Given the successful verification, the receiver acknowledges the receipt of the packet by sending an ACK packet (ACK) back to the sender [1]. The structure of such an ACK packet is depicted in Figure 10 and the communication process is shown in Figure 11.

The delay between the receipt of the packet and the transmission of the ACK is the basis for the fingerprinting method we have presented in [11]. This delay is independent of higher layers such as the network stack of the operating system, since only the WLAN chipset is responsible for calculating the CRC and transmitting the ACK packet. Our analysis of these delays suggests that the delay time is variable on the same chipset and that these variations are specific for each chipset. The variations can be used to create chipset specific fingerprints by storing the occurence of the delay time values for each chipset in histograms.

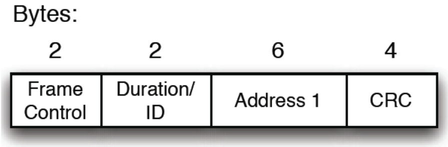

Figure 10: Structure of an ACK packet

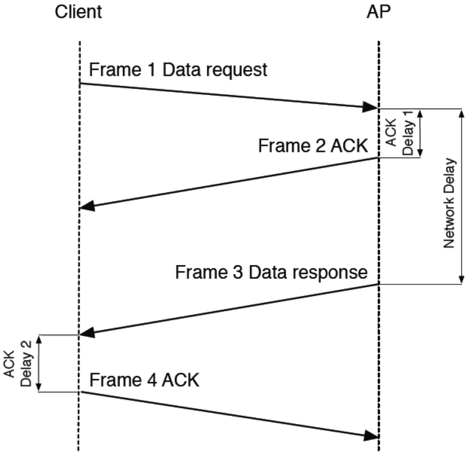

Figure 11: ACK delay

Obviously, the creation of significant histograms requires an adequate amount of delay times values. For the evaluation of our system we have been using between 50 and 150 delay time values for each histogram. An example for the histograms of two different WLAN chipsets

[1] There are some exceptions: The receiver does not acknowledge management frames, multi frames, and broadcast frames.

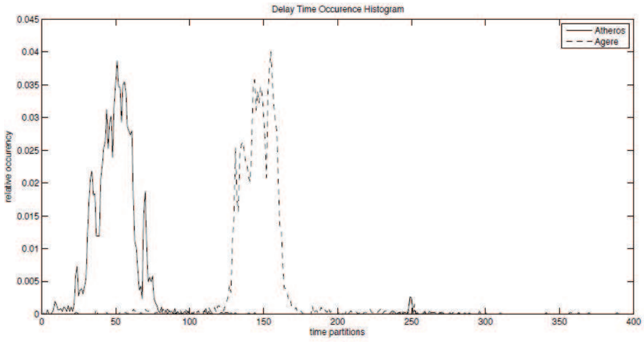

Figure 12: Delay histograms of an Atheros chipset and an Agere chipset

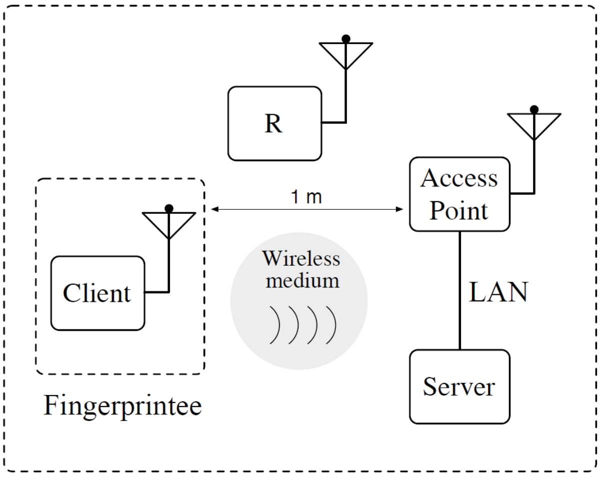

Figure 13: Evaluation setup

is given in Figure 12.

The fingerprinting technique relies on supervised machine learning algorithms to learn the typical histograms for each chipset. The applied algorithms are based on supervised versions of Self Organizing Maps and Neural Gas Maps. The supervised learning or training process requires labeled training data to create models which can then be used for the identification of traffic generated by unknown chipsets.

3.3.2 Proof-of-Concept

The evaluation setup is depicted in Figure 13.

The client uses the chipset which is the target of the analysis and creates traffic by communicating with another machine (labeled as server). The probe captures this traffic by using the monitor mode of its WLAN card. The traffic from the client to the server has been generated by using ICMP pings covering a large range of possible packet sizes. The captured data is used to create histograms needed for the generation of the training set and the test set. The SOM and Neural Gas based algorithms are trained and evaluated with these sets. Table 2 shows the chipset used for the evaluation process and Table 3 shows the classification accuracy of the system.

The results indicate that the proposed method is able to recognize different chipsets with an acceptable classification rate.

Table 2: Chipsets

Chipset
1: RoamAbout
2: Broadcom Corporation BCM4318
3: Linksys Broadcom 94306
4: Intel BG2200
5: Intel BG2100

Table 3: Results

	1	**2**	**3**	**4**	**5**
1	**80,4%**	0,0%	0,0%	0,0%	19,2%
2	0,0%	**97,2%**	0,0%	2,4%	0,4%
3	44,0%	0,0%	**56%**	0,0%	0,0%
4	0,0%	20,0%	0,0%	**74,0%**	3,0%
5	0,0%	4,0%	0,0%	13,0%	**79,0%**

Encouraged by these initial results, the techniques have been implemented in a tool called WIFINGER. The tool is written in C and classifies a chipset according to the captured delay time information. It forms the basic platform for the evaluation of further refinements of the technique. With the hope to achieve better classification results we added additional information to the delay histograms - the packet size. The rational behind this is based on the idea that the size might influence the delay time. In order to cope with the large amount of additional information added to the timing information, the delay values and packet sizes are arranged in groups.

Unfortunately, the analysis of the test results shows that including the packet size does not improve the classification accuracy. In fact, the delay time does not depend on the size of the packet.

3.3.3 Discussion

The proposed method uses the delay time between a data frame and the belonging ACK to identify chipsets. For an accurate classification result, 500 to 1000 values are needed. This values can be obtained by passive monitoring or by actively sending packets to the chipset that needs to be identified. In contrast to the method of Franklin et al. the amount of data needed for the accurate representation of the chipset fingerprint can be obtained quite easily, due to the fact that each packet needs to be acknowledged with an ACK packet.

The proposed fingerprinting method cannot differentiate between WLAN NICs containing the same chipset since in this case the extracted ACK delay information is the same. This represents a limitation to the broad use of this technique.

4 Evaluation of the Compared Methods

In the previous sections we describe three different WLAN fingerprinting methods, that exploit different characteristics of WLAN communication to create fingerprints for employed wireless NICs. Due to the different nature of the employed techniques each method comes with different strengths and weaknesses in terms of accuracy, required resources and the capability to differentiate between wireless NICs, employed drivers and firmware versions. Table 4 shows the differences of the presented methods.

- **Radio Frequency Fingerprinting in Wireless Networks by Hall et al.**: This method is capable of identifying single devices even if they use the same chipset, driver or firmware. Thus, this technique has the highest accuracy of the presented methods. However, the broad application is limited since special hardware is required for fingerprint identification.

- **Passive Data Link Layer Fingerprinting by Franklin et al.**: This method does not require any special hardware and is capable of identifying different WLAN chipsets. Due to the nature of this technique, an identification of the same chipsets with different driver versions is theoretically possible. The main downturn of this method is the time needed to capture enough data for accurate fingerprint creation in a real-world scenario.

- **Acknowledge-Frame Delay Fingerprinting by Lackner et al.**: This method does not require any special hardware and the data needed for the fingerprinting process can easily be captured. However, this method can only be used to identify WLAN NICs based on different chipsets.

5 Conclusions

In this paper we present the analysis of current WLAN fingerprinting techniques. The evaluation showed that all presented techniques are based on completely different approaches. However, all presented methods are well suited to enable such a identification capability.

Furthermore, each presented method is based on the usage of completely different features for the actual chip set classification. Due to the nature of different features and the fact that different data is used for the fingerprinting process, each of the presented methods has unique strengths and weaknesses. The analysis of current WLAN fingerprinting techniques leads to the conclusion that all presented methods have significant constraints. These constraints are the reasons for limited usability and missing acceptance of the proposed mechanisms. However, all presented methods provide valuable insight into different approaches of WLAN fingerprinting and can be used as basis for further improvements. From a practical point of view the best way to cope with MAC spoofing is to apply secure authentication methods such as 802.11i in combination with 802.1X and EAP-TLS [12] to ensure that MAC spoofing is at least not possible with dataframes. MAC control and management-frames are unfortunately left vulnerable, even by state-of-the-art security standards.

References

[1] J. Bellardo and S. Savage, "802.11 denial-of-service attacks: Real vulnerabilities and practical solutions", *USENIX Security Symposium*, Washington D.C., pp. 15-28, 2003.

[2] V. Bhargavan, A. J. Demers, S. Shenker, and L. Zhang, "Macaw: A media access protocol for wireless lan's", *ACM SIGCOMM Conference London*, 1994.

[3] E. D. Cardenas, *Mac Spoofing - An Introduction*, GIAC Security Essentials Certification (GSEC), 2003.

[4] D. McCoy, J. V. Randwyk, P. Tabriz, D. Sicker, V. Neagoe, and J. Franklin, "Passive data link layer 802.11 wireless device driver fingerprinting", *Proceedings of the 15th conference on USENIX Security Symposium*, 2006.

[5] F. Guo and T. C. Chiueh, *Sequence Number-based MAC Address Spoof Detection*, Technical report, Computer Science Department Stony Brook University, NY 11794, 2006.

[6] A. Günther and C. Hoene, *Measuring Round Trip Times to Determine the Distance Between WLAN Nodes*, Technical report, Telecommunication Networks Group, TU-Berlin, Germans, 2005.

[7] J. Hall, M. Barbeau, and E. Kranakis, "Detection of transient in radio frequency fingerprinting using signal phase", *Proceedings of the 3rd IASTED International Conference on Wireless and Optical Communications (WOC)*, 2003.

[8] J. Hall, M. Barbeau, and E. Kranakis, "Radio frequency fingerprinting for intrusion detection in wireless networks", *IEEE Transactions on Dependable and Secure Computing*, 2006.

[9] IEEE, *IEEE OUI License Page, IEEE OUI and Company ID Assignments*, 2006. (http://standards.ieee.org/regauth/oui/index.shtml)

[10] IEEE, *Request Form for an Individual Address Block*, 2006. (http://standards.ieee.org/regauth/oui/pilot-ind.html)

[11] G. Lackner, M. Lamberger, U. Payer, and P. Teufl, "Wifi fingerprinting", *DACH Mobility 2006*, Sep. 2006.

Table 4: Comparing the different methods

	Hall	**Franklin**	**Lackner**
Fingerprint for	chip	firmware	chipset
Resources	- -	+ +	+ +
Accuracy	>99%	70% (30%) 30% during our evaluation	60% - 90% depending on the environment
Biggest con	complicated	easy to circumvent	poor accuracy

[12] A. Mishra and W. A. Arbaugh, *An Initial Security Analysis of the IEEE 802.1x Standard*, Technical report, Department of Computer Science, university of Maryland, 2002.

[13] F. Robinson, "802.11i and wpa up close", *Network Computing*, vol. 4, no. 1, pp. 79-82, 2004.

[14] J. Wright, *Detecting wireless lan mac address spoofing*, Technical report, GCIH, CCNA, 2003.

Peter Teufl is a member of the *Institute for Applied Information Processing and Communications* (IAIK) at the University of Technology Graz. Currently, he is working on his Ph.D in the area of machine learning-based document classification and information extraction. During the last three years he was involved in different projects related to E-Government and network security.

Udo Payer graduated from a technical college for communication and electronics. Thereafter, he studied Telematics at the Graz University of Technology with special emphasis on networks and network security. Since 2001 he is responsible for all network security activities at IAIK. In 2005, he received his Ph.D in computer science for his work in the field of polymorphic code detection. His special interests are in the fields of intrusion- and malicious code detection based on machine learning techniques but also in problems of intrusive event correlation. At present, he is also working on secure P2P routing protocols.

Günether Lackner is currently working on his Ph.D in the area of WLAN security and anonymous authentication. He received his B.Sc and M.Sc degrees in Telematics at the University of Technology Graz, Austria. He collaborated in several network security-related projects during the last years as a member of the *Network Security Group* at the *Institute for Applied Information Processing and Communications* (IAIK) at the University of Technology Graz. Furthermore he is CIO and head of research at studio78.at.