On Security Vulnerabilities of Null Data Frames in IEEE 802.11 based WLANs

Wenjun Gu[‡], Zhimin Yang[‡], Can Que[§], Dong Xuan[‡], Weijia Jia[§]

[‡] *Department of Computer Science and Engineering*
The Ohio State University
Columbus, OH 43210, U.S.A.
{gu,yangz,xuan}@cse.ohio-state.edu

[§] *Department of Computer Science*
City University of Hong Kong, China
canque2@student.cityu.edu.hk
itjia@cityu.edu.hk

Abstract

Null data frames are a special but important type of frames in IEEE 802.11 based Wireless Local Area Networks (e.g., 802.11 WLANs). They are widely used for power management, channel scanning and association keeping alive. The wide applications of null data frames come from their salient features such as lightweight frame format and implementation flexibility. However, such features can be taken advantage of by malicious attackers to launch a variety of attacks. In this paper, we identify the potential security vulnerabilities in the current applications of null data frames. We then study two types of attacks taking advantage of these vulnerabilities in detail, and evaluate their effectiveness based on extensive experiments. Finally, we point out that our work has broader impact in that similar vulnerabilities exist in many other networks.

1 Introduction

IEEE 802.11 based WLANs (e.g., 802.11 WLANs) are widely deployed nowadays in airports, starbucks, universities, and even the whole city [7]. Such wide deployment is due to the fact that 802.11 WLANs allow the convenience of user mobility and the cost effectiveness/flexibility in network deployment. It is believed that 802.11 WLANs will keep playing a major role in wireless communications.

Null data frames are a special but important type of frames in 802.11 WLANs. They are special because their *Frame Body* fields are empty, and they are the only type of frame whose usage is not explicitly specified in the IEEE 802.11 standard [2]. However, in real network interface cards (NICs) implementations, they are used in a wide variety of applications, including power management, channel scanning and association keeping alive. When used for power management and channel scanning, a single bit in null data frame denotes the state switching of a station between active and sleeping states. On the other hand, when used for association keeping alive, the null data frame as a whole is used to notify the access point the existence of a station during idle period. Such wide applications of null data frames come from their salient features such as lightweight frame format and implementation flexibility. Null data frames are lightweight as they are short in size and un-encrypted. Therefore, it incurs little overhead in generating and processing these frames. Null data frames allow implementation flexibility due to the lack of standardization in the IEEE 802.11 standard [2]. Thus protocol designers have the freedom to apply them wherever applicable.

However, we find that these salient features do not come for free. Being short and un-encrypted, null data frames can easily be forged at low overhead to conduct a variety of spoofing attacks. On the other hand, being flexible in implementation, different vendors have applied null data frames in different applications, which allows a fingerprinting attack to track a device/user based on its unique set of behaviors in null data frame usage. In short, the salient features of null data frames act as a double-edged sword.

In this paper, we study two types of attacks on existing applications of null data frames in 802.11 WLANs based on the security vulnerabilities we identified above. First, we study *functionality based Denial-of-Service attacks*. In these attacks, the attacker spoofs the identity of the victim station, and sends fake null data frames to mess up with the intended functionalities of null data frames. Second, we study *implementation based fingerprinting attack*. In this attack, the attacker takes advantage of implementation variations of null data frames, and correlates the frames sent from seemingly different stations by the unique behaviors in using null data frames. Based on extensive experiments, we find that the above attacks are effective in achieving their respective goals. We also discuss preliminary defenses to alleviate the vulnerabilities. Finally, we point out that our work has broader impact in that similar vulnerabilities exist in many other networks.

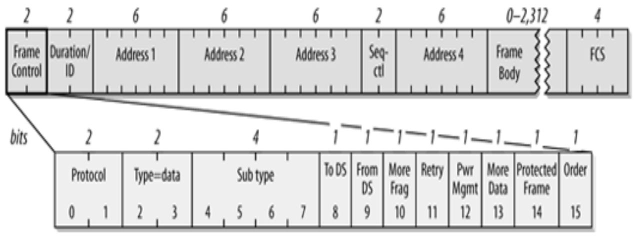


Figure 1. IEEE 802.11 data frame format [2]

2 Preliminary of Null Data Frames

2.1 Frame Format

In Fig. 1, we show the general IEEE 802.11 data frame format. Null data frames are a special type of data frame where the *Frame Body* field is empty. Depending on the situation, the *Duration/ID* field in data frames denotes either the period the medium is expected to be busy (e.g., NAV value) or the association ID. Among the four *Address* fields, null data frames only use two, that are, the addresses of the receiver (i.e., the access point) and the transmitter (i.e., the station). The *sequence control (Seq-ctl)* field consists of two parts, that are, sequence number and fragment number. The sequence number is used to detect lost or unordered frames. The fragment number is used to order the fragments of a single packet whose size exceeds the maximum allowed. The *frame check sequence (FCS)* field is used for frame integrity checking, and is calculated based on the whole frame. In the following, we discuss the *Frame Control* field in detail.

In the *Frame Control* field shown in Fig. 1, the $Protocol$ field denotes 802.11 MAC version. The $Type$ and $Subtype$ fields denote the frame type. The $ToDS$ and $FromDS$ fields denote whether the frame is sent to or from the distributed system respectively. The $MoreFrag$ field denotes whether there is more fragment in the current packet. The $Retry$ field denotes whether the frame is a retransmitted one. The $PwrMgmt$ field denotes the power saving state that will be detailed below. The $MoreData$ field denotes whether there are more buffered data. The $ProtectedFrame$ field denotes whether encryption is applied. The $Order$ field denotes whether strict frame ordering is applied.

2.2 Applications

Although IEEE 802.11 standard [2] does not explicitly specify the usage of null data frames. They are in fact widely used in reality. In the following, we classify the applications into two categories.

First, the station can use the $PwrMgmt$ field to inform the access point its state switching. In power management, the station sets $PwrMgmt$ field to 1 before it switches to sleeping state, and sets it to 0 otherwise. When the station is sleeping, the access point buffers all data frames for the station until it switches back to active state. Besides power management, the station can send a null data frame with $PwrMgmt$ bit set to 1 before scanning other channels. In this case, the access point also buffers the frames until the station switches back to the current channel.

Second, the station can use the null data frame as a whole to keep the association alive. When the station keeps idle for a long time, the access point cannot decide whether the station is simply idle, is currently out of its service area or becomes out of power. Keeping the associations for the stations out of service area or out of power will waste association ID resource. Thus, the idle station needs a mechanism to inform the access point of its existence. Null data frames are a natural choice.

3 Security Vulnerabilities

3.1 Functionality associated Vulnerability

As discussed above, the main functionalities of null data frames are state switching in power management and channel scanning, and association keeping alive during idle period. When null data frames are used for state switching, an attacker can spoof the identity of a station in sleeping state or scanning in another channel, and generate fake null data frames to fetch the buffered frames of the victim station. Since the access point deletes the buffered frames after successful transmission, such attack in effect results in continuous frame losses. On the other hand, when null data frames are used to keep association alive, an attacker first establishes a large number of dummy associations with the access point in public access 802.11 WLANs, and then use null data frames to keep those associations alive. As each association takes one unique association ID, and the total number of allowed association IDs for one access point is 2007, it is not difficult for the attacker to deplete all available association IDs. In this way, other stations cannot associate with the access point.

The main reason behind the above two DoS attacks is that the null data frames are lightweight. Null data frames are short. They have no frame body. Besides, null data frames are not encrypted. These salient features allow efficient generation and processing, and are useful in many situations. For example, when the station switches between sleeping and active states frequently, it is preferable for the station to retrieve the buffered frames at low cost. Null data frames are ideal for such usage. However, the lightweight feature of null data frames also allows malicious attacker to generate/manoeuver fake null data frames easily.

3.2 Implementation associated Vulnerability

Since the exact implementation of null data frames is not explicitly specified in IEEE 802.11 standard [2], NIC vendors have been applying null data frames in different ways. Some NICs use null data frames for power management, channel scanning and association keeping alive, while others implement only a subset of the above. Furthermore, for a single function such as power management, different vendors' implementations vary dramatically. Therefore, an attacker may leverage such implementation variations to determine that two seemingly unrelated sets of frames(i.e., frames with different MAC/IP addresses sent at different positions/time) in fact come from the same device/user. Based on this information, the attacker can track the victim device/user and compromise user privacy.

Clearly, the above attack comes as a result of the implementation flexibility. As null data frames are the only type of frame whose usage is not explicitly specified in the IEEE 802.11 standard [2], the NIC vendors have the freedom to use null data frames wherever they might be useful in whichever way appropriate. Such flexibility helps NIC vendors achieve efficiency in many applications, while on the other side allows the attacker to conduct fingerprinting.

4 Functionality based Denial-of-Service Attacks

4.1 Attack Model

The basic idea of functionality based Denial-of-Service (DoS) attacks is that the attacker spoofs null data frames to mess up with the intended functionalities of the genuine null data frames. There are three main usages of null data frames, that are, power management, channel scanning and association keeping alive. Since both power management and channel scanning use the same field for similar purposes, these two have the same functionality called state switching. In the following, we introduce two DoS attacks targeting state switching and association keeping alive.

In the *state switching based DoS attack*, an attacker spoofs a victim station in sleeping state or scanning in another channel. Thus, the attacker can fetch the buffered frames and cause continuous frame losses. In the *association keeping alive based DoS attack*, the attacker first establishes a large number of dummy associations with the access point in public access 802.11 WLANs, and then sends null data frames to keep those associations alive. As each association takes one unique association ID, and the total number of allowed association IDs for a single access point is 2007, the attacker can prevent other stations from associating with the access point.

Algorithm 1 State Switching based DoS Attack

```
1:  State_Switching_based_DoS_Attack
2:  while (TRUE)
3:    $capture\ a\ frame\ F$;
4:    if $F.sender\_MAC == victim\_MAC$ then
5:      if $F.PwrMgmt == 1$ then
6:        $generate\ fake\ null\ data\ frame\ F'$;
7:        $F'.sender\_MAC == victim\_MAC$;
8:        $F'.receiver\_MAC == AccessPoint\_MAC$;
9:        $F'.sequence\_number == F.sequence\_number + 1$;
10:       $F'.PwrMgmt == 0$;
11:       $send\ fake\ null\ data\ frame\ F'$;
12:       $victim\_is\_sleeping == TRUE$;
13:     else
14:       $victim\_is\_sleeping == FALSE$;
15:     end if
16:   else if $F.receiver\_MAC == victim\_MAC\ AND$
          $F.type == DATA\ AND$
          $victim\_is\_sleeping == TRUE$ then
17:     $send\ fake\ ACK\ frame\ to\ access\ point$;
18:   end if
19: end while
```

Due to space limitation, we focus on *state switching based DoS attack* in the paper. As the usages of null data frames in power management and channel scanning are similar, we assume the scenario is power management. However, our discussions apply to channel scanning as well.

An intuitive attack is that the attacker keeps flooding fake null data frames with $PwrMgmt$ field set to 0. This attack is simple to launch, but it involves large number of frame injections. This is not cost effective to the attacker, and results in easy detection. We design an attack that injects much fewer frames while achieving the same effect. The pseudocode of the attack is given in Algorithm 1. Specifically, the attacker captures and checks all frames related with the victim station. When the victim station sends a frame informing its intent to switch to sleeping state (lines 4 to 5), the attacker generates one fake null data frame (line 6). The attacker sets the MAC addresses of the sender and receiver and the sequence number appropriately (lines 7 to 9) to prevent immediate detection. Then the attacker sets the $PwrMgmt$ field to 0 (line 10) and sends out the fake null data frame (line 11). The attacker also sets a variable $victim\_is\_sleeping$ to TRUE (line 12) to track the victim station state. If the victim station sends a frame informing its intent to switch to active state (line 13), the attacker does nothing except for setting the variable $victim\_is\_sleeping$ to FALSE (line 14). When the access point sends a data frame to the victim station that is in sleeping state (line 16), the attacker sends a fake ACK frame back (line 17).

In Fig. 2, we illustrate the impact of such attack. In this example, the victim station wakes up in the beginning of every other beacon interval to check the availability of

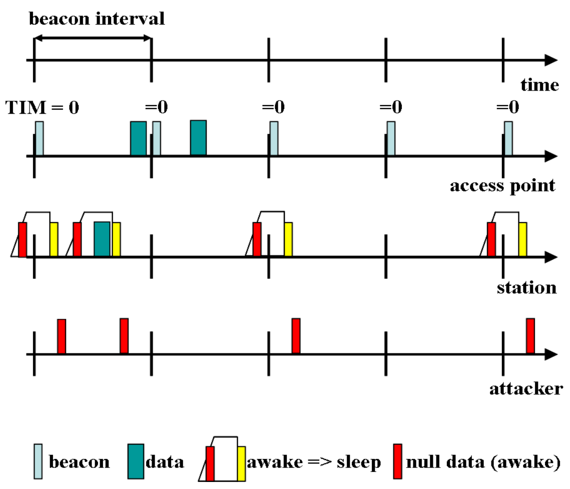
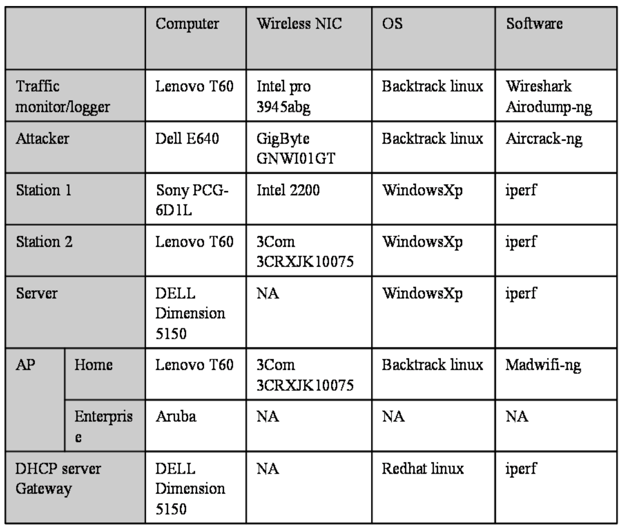


Figure 2. State switching based DoS attack



		Computer	Wireless NIC	OS	Software
Traffic monitor/logger		Lenovo T60	Intel pro 3945abg	Backtrack linux	Wireshark Airodump-ng
Attacker		Dell E640	GigByte GNWI01GT	Backtrack linux	Aircrack-ng
Station 1		Sony PCG-6D1L	Intel 2200	WindowsXp	iperf
Station 2		Lenovo T60	3Com 3CRXJK10075	WindowsXp	iperf
Server		DELL Dimension 5150	NA	WindowsXp	iperf
AP	Home	Lenovo T60	3Com 3CRXJK10075	Backtrack linux	Madwifi-ng
	Enterprise	Aruba	NA	NA	NA
DHCP server Gateway		DELL Dimension 5150	NA	Redhat linux	iperf

Figure 3. Equipments in our experiment

its buffered data frames at the access point. Such information is embedded in the beacons sent from the access point in the beginning of each beacon interval by setting a bit (called TIM) for the corresponding association ID. In particular, the TIM bit of an association ID is set to 1 if there are buffered frames for the corresponding station, and it is set to 0 otherwise. In the example, the station switches to sleeping state in the beginning of the first beacon interval since there is no buffered data frame for it ($TIM = 0$), and switches back to active state in the middle of this beacon interval since it has a data frame to send. After sending this data frame, the station switches back to sleeping state. Shortly after that, the attacker sends a fake null data frame, and notifies the access point that the victim station just switches back to active state again. Later on, the access point receives two data frames for the victim station, and sends them to the attacker. When the victim station switches to active state in the third beacon interval, it is notified that there is no buffered data frame for it. In effect, the attacker is able to delete two data frames from access point while the victim station is in sleeping state. In fact, the attacker is able to denial the service of the victim station by keeping deleting its data frames.

Remarks: In the following, we make some comments on our designed attack. We want to point out that the access point could detect the existence of attack from the inconsistency of sequence numbers when the victim station wakes up and sends data frames. However, the access point cannot tell whether a received null data frame is genuine or fake immediately after reception. This is because the sequence number in the fake null data frame can be modified to match the previous frame sent by the victim station. Even if the access point can eventually detect the attack by observing two frames with the same sequence number (one from the victim station and the other from the attacker), it cannot tell whether it is genuine or fake after receiving the first of these two frames. Therefore, it is hard for the access point to differentiate fake null data frames from the genuine ones in real time, and ignore fake frames to evade attack.

4.2 Experimental Evaluations

In order to determine the feasibility of the above attack and its effectiveness, we conduct extensive experiments on our testbed. The equipments used in our experiments are listed in Fig. 3. The *attacker* conducts the DoS attack on the two victim *stations*, and the *logger* passively logs all wireless communications for later analysis.

We use the Iperf [8] as traffic generator and performance measurement tool. The server is installed with Iperf to generate TCP/UDP traffic, while the stations are installed with Iperf to receive traffic and measure throughput. Each test lasts for 300 seconds. Initially, there is no attack in place. The attack is enabled at the 60^{th} second, lasts for 60 seconds, and is disabled at the 120^{th} second. Again, the attack is enabled at the 180^{th} second, lasts for 60 seconds, and is disabled at the 240^{th} second.

In Figs. 4 and 5, TCP and UDP traffic are tested on Intel NIC with power saving mode enabled and MADWiFi access point respectively. We can see in Fig. 4 that TCP throughput decreases to 0 immediately after attack comes at the 60^{th} second. This is because the attacker keeps deleting the data from the access point. Even worse, the TCP connection is disconnected during the attack, and the throughput remains 0 after attack is disabled at the 120^{th} second. This shows that consistent frame loss caused by the attack could cause TCP disconnection. In Fig. 5, we can see that

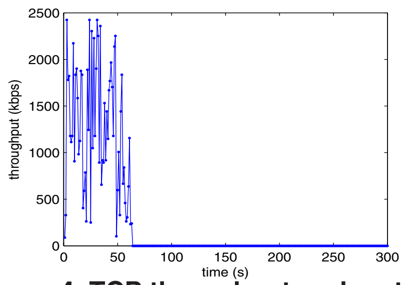


Figure 4. TCP throughput under attack

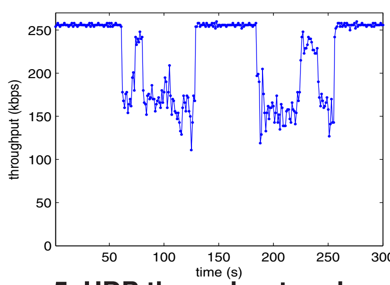


Figure 5. UDP throughput under attack

UDP throughput degrades significantly during the time the attack is enabled. This is also because of the attacker deleting frames from the access point. However, UDP traffic is able to resume its normal throughput when the attack is disabled due to the connectionless nature of UDP.

We conduct extensive experiments on various other application settings, NICs and access points. Due to space limitation, we cannot report all data here. Basically, the data we report are typical and representative, and our observations hold for all scenarios. In summary, we find that the state switching based DoS attack has significant impact on both TCP and UDP traffic. TCP traffic is disconnected, while UDP traffic suffers from throughput degradation under attack.

4.3 Preliminary Defense

For a defense mechanism of the above attack to be effective, the access point should be able to differentiate the fake null data frames from the genuine ones. However, such task is challenging. The null data frames are un-encrypted and easy to forge. Besides, the attacker could use the same hardware as the victim station, stay close to the victim station and control its transmission power so that the access point cannot tell the difference by RF fingerprinting techniques introduced in [4] [5].

Fortunately, we find that certain temporal information embedded in genuine null data frames with $PwrMgmt$ set to 0 could be leveraged by the access point. Based on the observation of the traces we collect, we find that there are two types of genuine null data frames with $PwrMgmt$ set to 0. The first type are sent just before the beacon interval when the station is supposed to wake up. The second type are sent just before the station sends the first data frame after it wakes up. In both cases, the station does not need to wake up too early to waste power. In particular, the station never wakes up more than one tenth of the beacon interval in advance. The above temporal information could help the access point to detect most fake null data frames and alleviate the attack impact. With such defense in place, the attacker falls into a dilemma between attack impact and detection. If the attacker sends fake null data frames too early, it will be detected. Thus, the attacker has to delay the sending of fake null data frames, which in effect limit the attack impact. We will study such defense mechanism and evaluate its effectiveness as our future work.

5 Implementation based Fingerprinting Attacks

5.1 Attack Model

In this section, we discuss how a fingerprinting attacker can take advantage of the implementation variations of null data frames to compromise user privacy. As we know, MAC address is an obvious identifier that can be used to correlate two sets of frames sent in different positions at different time. After obtaining a sequence of frames with the same MAC address, we are able to determine the position of a single device/user at certain time and even the whole movement pattern if enough number of frames are obtained. This could compromise the location privacy of wireless mobile devices and even the personal privacy of mobile users.

However, the user can change his MAC address between communication sessions to defend user tracking. Unfortunately, this is not enough. During the analysis of the traces we collect on a variety of typical NICs and the well known public traces, we find that the implementations of null data frames in NICs vary dramatically. In particular, we identify 7 fingerprinting rules that can help the attacker to correlate two sets of frames even if their MAC addresses are different, which are listed as follows.

Rule 1: Some NICs use null data frames for power management (**Y**) while others use PS-Poll control frames (**N**);

Rule 2: Some NICs send null data frames once per 10 seconds when there is no data communication to keep association alive (**Y**), while others do not (**N**);

Rule 3: Some NICs send null data frames before sending probe request frames during active channel scanning (**Y**), while others do not (**N**);

Rule 4: After switching from sleeping state to active state, some NICs send a null data frame ($PwrMgmt$ set

Confi-guration	System	NIC	Power Saving Mode
A	Dell 640m	Dell Wlan1390	Fast
B	Dell 640m	Dell Wlan1390	Disabled
C	Lenovo Thinkpad T60	Intel pro 3945abg	Default value
D	Dell E1405	Intel pro 3945abg	Default value
E	Sony PCG-6D1L	Intel 2200	Enabled
F	Lenovo Thinkpad T60	3Com 3CRXJK10075	Auto

Figure 6. Six configurations in our 802.11 WLAN traces

Configuration	R1	R2	R3	R4	R5	R6	R7
A	Y	N	N	N	N	N	N
B	N	Y	N	N/A	N/A	N/A	N/A
C	Y	N	Y	Y	N	Y	Y
D	Y	Y	Y	N	N	Y	Y
E	Y	N	Y	Y	Y	Y	N
F	Y	N	Y	N	N	Y	N

Figure 7. Behaviors of six configurations with respect to the seven fingerprinting rules

to 0) before sending the first data frame (**Y**), while others send the data frame directly (**N**);

Rule 5: Some NICs can still receive data frames for a short period of time after sending a null data frame ($PwrMgmt$ set to 1) (**Y**), while others switch to sleeping state immediately after sending such null data frame (**N**);

Rule 6: Some NICs send null data frames periodically (**Y**), while others send it aperiodically depending on the availability of data (**N**). For example, when there are continuous data communications, the station sends null data frames at a higher frequency. However, during the idle period, the station sleeps for longer time and occasionally wakes up to check the availability of buffered frames, which results in sending null data frames at a lower frequency;

Rule 7: Some NICs send null data frame ($PwrMgmt$ set to 1) and switch to sleeping state after sending out all the data frames to the access point even if there are more buffered data frames at the access point (**Y**), while others do so only when there is no frame in both directions (**N**).

5.2 Experimental Evaluations

In this section, we first apply the fingerprinting rules above to the 802.11 WLAN traces we collect with a variety of typical NICs with different power management modes.

MAC	Use PWR	R1	R3	R4	R5	R7
87C67ADBCE66	N					
1ED431D313E6	Y	Y	Y	Y	Y	Y
8E9ECB6C300B	Y	Y	Y	N	N	Y
A1782DD5AA38	Y	Y	Y	N	N	Y
19D41C6C3824	Y	Y	Y	Y	N	N
93AD8D75C406	Y	Y	Y	Y	N	N
F7DDB312CC5B	Y	Y	N	Y	N	N
6DD32345A7B0	Y	Y	N	Y	N	N
D955031F5CAA	Y	Y	Y	N	N	N
6AF471EA1CB0	Y	Y	Y	N	N	N
CD9A3A58E0C8	Y	Y	Y	N	N	N
9D0C3F1D9231	Y	Y	Y	N	N	N
4A3C00F92955	Y	Y	N	N	N	N
2DCCE6E9501C	Y	Y	N	N	N	N
459DE74BF915	Y	Y	N	N	N	N
A6F55AFA91B9	Y	Y	N	N	N	N

Figure 8. Behaviors of the NICs in Sigcomm 2004 trace [10]

Then we further apply these rules to the well known 802.11 WLAN trace collected in $Sigcomm$ 2004 [10].

In Fig. 6, we show the 6 configurations based on which we collect our 802.11 WLAN traces. In the first 5 configurations we use the NICs originally shipped with the laptops, while in the last one (i.e., configuration F) we use an external NIC on Lenovo Thinkpad T60. Except for the second configuration (i.e., configuration B), all other configurations are enabled with power management.

In Fig. 7, we show the behaviors of the 6 configurations with respect to the 7 fingerprinting rules. Each row corresponds to one configuration, while each column corresponds to one rule. As we can see, for any two configurations, there exists at least one rule that can differentiate them. In other words, the fingerprinting attack is effective in identifying each of the 6 configurations from others.

In order to evaluate the effectiveness of our fingerprinting attack in real life situation, we further apply our rules on the well known 802.11 WLAN trace collected in $Sigcomm$ 2004 [10]. In particular, we write scripts for 5 of our 7 fingerprinting rules (i.e., rules 1, 3, 4, 5 and 7) and obtain the behaviors of the NICs based on our scripts. We do not use rules 2 and 6 here due to the lack of idle periods in this trace. However, we believe these two rules are useful in general as they are observed in other public online traces. In order to obtain stable results, we only consider the NICs that send/receive more than 500 data frames (including null data frames) in the trace.

In Fig. 8, we show the behaviors of the NICs. Although all MAC addresses are camouflaged for privacy reason, these camouflaged MAC addresses can still help us to obtain the frames from the same NIC. An extra rule is added (second column), which states whether the NIC ever uses power management. As we can see, we can classify the NICs into 7 categories. In particular, 2 NICs are identified due to their unique behaviors, while the remaining share the same behaviors with a few others. Although in this case we may not be able to identify all NICs uniquely, our fingerprinting rules help to classify NICs into finer categories otherwise not possible. Besides, our fingerprinting rules can complement existing fingerprinting techniques to achieve finer classification and higher accuracy.

5.3 Preliminary Defense

In order to defend against the fingerprinting attack, decreasing the implementation variations seems to be a natural choice. However, it is not possible to completely eliminate implementation variations due to two reasons. First, there exists a tradeoff between security and power conservation. Although the user can disable power management to eliminate many null data frames, it comes at the cost of much higher power consumption. This is not preferable for wireless mobile users as the laptop battery still cannot sustain for more than only a few hours under current technology.

Second, we doubt a single implementation suffices for all situations. It is preferable for a single NIC to have multiple options for users to choose from depending on the scenarios and user preferences. There is no single implementation that can satisfy all users under all situations. Different implementations under different options certainly show different behaviors in certain ways, and can be taken advantage of by the fingerprinting attacker.

Another possible defense is frequently changing the usage of null data frames between communication sessions either manually by the user or automatically by the NIC driver. However, this defense has obvious drawbacks. First, forcing users manually change the usage options introduces extra work for the users in the best case, and annoys the users in the worst case. Second, letting NIC drivers dynamically change the usage options when the application scenario does not change obviously degrades the performance of associated functionalities.

Finally, we suggest the NICs vendors consider both security and functionality in NICs design. In particular, we would like to have as few unique options as necessary, and standardize the implementations of these options. This could make the fingerprinting attack much less effective.

6 Fundamental Tradeoff

In the above, we discussed the applications and features of null data frames in 802.11 WLANs and how the applications and features reveal security vulnerabilities to malicious attackers. Ironically, the fundamental reason of the existence of those attacks is that while the salient features of null data frames allow them to achieve many functionalities efficiently, such features at the same time allow malicious attackers to attack the functionalities equally efficiently.

The IEEE 802.11w working group [1] is planning to protect management frames such as deauthentication and deassociation frames to defend DoS attacks based on management frames spoofing. It might seem trivial to eliminate null data frame spoofing by simply protecting (i.e., encrypting) such frames. Unfortunately, frame protection may not work as well for our case. Null data frames are sent much more frequently under power management than the deauthentication and deassociation frames. Encrypting/decrypting null data frames at high frequency incurs too much cost. On the other hand, standardizing the null data frames implementation to defend fingerprinting attack is not trivial either. It clearly disables the possibility for users to choose different power management options based on user preference and/or application scenarios. Besides, it cannot help the large number of NICs that are already on the market. Thus, it remains an open issue as to efficiently eliminating the security vulnerabilities brought by null data frames.

In the above, we discussed the null data frames in 802.11 WLANs. In fact, in many other protocols, similar messages exist as well, such as TCP SYN message during connection establishment, BGP keepalive message for routing link maintenance, 802.11 RTS/CTS frames for medium reservation, etc. These messages share many similarities with null data frames in 802.11 WLANs, including frequent usage, being un-encrypted and being simple and lightweight. Besides, these common features are closed related. Due to the fact of frequent usage, it is preferable to use simple and unencrypted messages to reduce overhead. Not surprisingly, these messages suffer from the tradeoff between functionality and security as well. In TCP SYN DoS attack, the attacker floods SYN messages to the server to establish large number of open TCP connections, which could disallow valid users from establishing connections with the server. In BGP, an attacker can send fake keepalive messages to BGP peers at certain time to cause routing disruption. In 802.11 WLANs, it is possible to fake RTS/CTS frames with extremely large NAV values to prevent nearby stations from accessing the medium for a long time. We point out that it is important to consider both functionality and security during protocol design.

7 Related Work

In this section, we discuss literatures on Denial-of-Service and fingerprinting attacks in 802.11 WLANs.

We first discuss the works on Denial-of-Service (DoS) attacks in 802.11 WLANs [3] [6]. In [3], deauthentication and deassociation frames are sent to disrupt the communications between the station and the access point. In [3], control frames like RTS/CTS are spoofed to prevent other stations from accessing the medium when it is actually free. In IE poisoning attack [6], the attacker modifies some insignificant bits in IE elements so that the initial negotiation procedure fails due to the inconsistency of IE elements. In 4-way handshake blocking attack [6], the attacker spoofs the first message in 4-way handshake so that a memory DoS attack is launched to the station. Compared with the attacks above, the DoS attacks we identified have three salient features. First, our attacks are hard to detect since null data frames are not encrypted and the fake null data frames are sent only when the victim node is sleeping. Second, our attacks are cost effective as they do not require constant flooding. Third, our attacks do not have strict requirement on the time when the fake frames should be inserted.

One work related with part of our work is the PS-Poll control frame based DoS attack in [3]. In this attack, the attacker spoofs PS-Poll control frame ($PwrMgmt$ set to 0) to delete data frames of the victim station when it is sleeping. Albeit the same attack consequence, our work differs from that in [3] in the following ways. First, our attack is more efficient. Only one null data frame is needed to retrieve all buffered data frames, while one PS-Poll control frame is needed per buffered data frame. Second, our attack is more practical. Based on our traces and the well known public traces, we find that in reality almost all NICs use null data frames for power management instead of PS-Poll control frame as specified in the IEEE 802.11 standard [2]. Third, we have implemented our attack, evaluated its effectiveness in real testbed, and proposed preliminary defense, none of which is conducted in [3].

In the following, we discuss fingerprinting attacks on 802.11 WLANs. In [4], signal features such as aptitude, frequency and phase are used to differentiate different devices due to the minor variations in hardware. In [5], it is observed that the time a station waits before sending probe request frames shows different probability distribution depending on the device driver implementation. In [9], it is found that the set of broadcast frame sizes sent by one user are in general different from that sent by another user since different users probably use different sets of applications. Besides MAC address, it is discovered in [9] that several other fields in MAC header of 802.11 frames (called implicit identifiers) can be used for fingerprinting, such as SSID field, supported rate, etc. Our fingerprinting attack can complement the existing techniques to further enhance the effectiveness of fingerprinting.

8 Conclusion

Null data frames are widely used in IEEE 802.11 based WLANs for power management, channel scanning and association keeping alive. Such wide applications are due to the lightweight frame format and implementation flexibility of null data frames. However, these features can be taken advantage of by malicious attackers to launch a variety of attacks. In this paper, we identify the potential security vulnerabilities of current applications of null data frames in 802.11 WLANs, study two types of attacks in detail, evaluate the effectiveness of the attacks based on extensive experiments, and propose preliminary defense mechanisms against them. Finally, we point out that similar vulnerabilities exist in many other networks.

Acknowledgement

This work was supported in part by the US National Science Foundation (NSF) CAREER Award CCF−0546 668, the Army Research Office (ARO) under grant No. AMSRD−ACC−R 50521−CI, SAR Hong Kong RGC Competitive Earmarked Research Grant (CE RG) No. 9041129 (CityU 113906), CityU Strategic Research Grant (SRG) No. 7002102 and 7002214. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

References

[1] Ieee 802.11w: Wireless lan medium access control (mac) and physical layer (phy) specifications: Protected management frames. *IEEE Computer Society LAN MAN Standards Committee*.

[2] Ieee 802.11: Wireless lan medium access control and physical layer specifications. *IEEE Computer Society LAN MAN Standards Committee*, August 1999.

[3] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: real vulnerabilities and practical solutions. In *Proceedings of the 12th USENIX Security Symposium*, August 2003.

[4] D. B. Faria and D. R. Cheriton. Detecting identitybased attacks in wireless networks using signalprints. In *Proc. of ACM Workshop on Wireless Security*, September 2006.

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

[6] C. He and J. C. Mitchell. Security analysis and improvements for ieee 802.11i. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS)*, February 2005.

[7] Houston-WiFi. http://auscillate.com/wireless/houston/.

[8] Iperf. available at http://dast.nlanr.net/projects/iperf/.

[9] J. Pang, B. Greenstein, and R. Gummadi. 802.11 user fingerprinting. In *Proceedings of the 13th ACM International Conference on Mobile Computing and Networking (MOBICOM)*, September 2007.

[10] Sigcomm04-trace. available at http://crawdad.cs.dartmouth.edu/download/uw/sigcomm2004/kalahari.