But...my station is awake!
Power Save Denial of Service in 802.11 Networks

Leandro F. Meiners

September 2009

Abstract

The **IEEE 802.11** wireless network standard defines a power save mechanism that allows client stations to enter a sleep mode, during which they are unable to transmit or receive data, in order to conserve energy.

The power save mechanism defined in the standard is fairly simple. In order to enter sleep mode a station must inform this to the access point, which will start buffering inbound frames for the station. Periodically the station must wake up to verify if the access point has buffered frames for it, in which case it must poll the access point before returning to sleep mode.

In this paper we present a low bandwidth active targeted denial of service for wireless (IEEE 802.11) networks based on the power save features of the **IEEE 802.11** wireless standard.

Denial of service attacks, which are aimed at disrupting availability of a service or host, are generally based in flooding the victim. Denial of service against the **IEEE 802.11** protocol are no exception to this rule. Nevertheless, this attack departs from this rule since it doesn't require flooding the victim. Instead it abuses the power save features of the standard to partially disconnect a station from the network (it can still send frames).

This attack requires sending one frame to start the attack, and an additional frame after each frame exchange performed by the victim station (to maintain the attack over time).

This paper explains the attack, how and why it works, and presents the results obtained from the tests done using our implementation. We conclude this paper with possible mitigation strategies, one of which we implemented, tested and proved to be effective in our labs.

1 Introduction & Related Work

This paper presents an attack on the availability aspect of the 802.11 protocol. This class of attacks, generally referred to as denial of service attacks, are aimed at disrupting service either in a targeted manner (for a particular station belonging to the network) or in an unspecific manner (i.e. targeting the network

as a whole). The denial of service attack presented takes advantage of the power save features of the 802.11 standard, to disrupt service in a targeted manner.

Most of the research related to 802.11 security is aimed at the confidentiality and integrity aspects of the protocol. Nevertheless, research has been done regarding availability. In [1] the authors present several denial of service attacks against the 802.11 protocol, including an attack related to the power save features. The attack presented in [1] defers from the attack detailed in this paper in that it is aimed at tricking a station in sleep mode to "remain" in such a mode.

The attack presented below tricks the access point into dropping frames destined for the station under attack by making it believe the station is in sleep mode. Therefore, our attack doesn't require that the station already be in sleep mode prior to attacking it as does the attack documented in [1]. This has the important drawback that when the station wakes up, it can no longer be attacked, with their attack, until it decides to enter power save mode again.

The other denial of service attacks documented in [1], and those documented in [2], as well as layer one jamming (a.k.a, radio signal jamming or RF jamming) attacks ([7]) require the attacker to constantly flood the target, which makes maintaining the attack over time or attacking more than one station very expensive resource-wise (except for jamming attacks which degrade the medium for all stations, which has the drawback of not being targeted). The attack we present has the benefit of being both targeted and not requiring the attacker to flood the victim. The worst case scenario for this attack, as will be shown below, is at most one frame per station under attack, and an additional frame after each frame exchange performed by the a station under attack (to maintain the attack over time). Nonetheless, our attack is not as disruptive as the others (i.e. the frames sent by the station under attack may be processed by the access point and some frames might make it through to the station), which completely disconnect the station from the network. Not flooding the victim has the added advantage that the attack may pass as an "abnormal" functioning of the station rather than an attack in progress. This has the obvious benefit of helping the attacker remain undetected.

2 Brief Introduction to 802.11

The 802.11 standard ([5]) defines the physical layer (first layer of the **OSI** model), being a wireless protocol this means the radio wave modulation techniques and the segment of the spectrum to use, and the *Medium Access Control* (a.k.a **MAC**) layer , which is part of the data link layer (layer two) of the **OSI** model. In this paper we are only concerned with aspects related to the **MAC** layer of the protocol.

2.1 Frames in 802.11

The **MAC** layer of the protocol has three types of frames:

- Management: used for network management tasks.
- Control: used to mediate access to the medium.
- Data: used to send the upper layer data.

With respect to this paper we are mainly interested in the following frame subtypes:

Frame type	Subtype	Description
Management	Reassociation request	Request to rejoin the network (for example when changing from access points inside the same wireless network)
Management	Beacon	Access point information
Management	Probe request	Network availability request
Control	PS-Poll	Used by a station in power save mode to request pending frames buffered at the access point
Control	RTS	Request to send (medium reservation)
Data	Null Function (no data)	Empty data frame
Data	QoS Null (no data)	Empty data frame in a *QoS* network

A complete list of the different types of frames supported by the protocol is described in table 1 in section 6.

2.2 Power save in 802.11

The 802.11 standard ([5]) includes power management features, that allow energy conservation by the stations. In order to accomplish this, stations are allowed to enter a sleep mode (or "doze" mode as is referred to in the standard) that lowers power consumption.

A station that changes its power management mode must inform the access point of this fact. Until the station changes its power save mode to active and notifies this to the access point, the access point will buffer frames headed for the station, which can only be sent by the access point at designated times. A station shall remain in its current power save mode until it informs the access point of a change via a successful frame exchange sequence initiated by the station, not being able to do so during the frame exchange. The station informs the power change to the access point by setting the *Power Management* bit in the *Frame Control* field of the frames sent by the station as part of the frame exchange. The *Power Management* bit indicates the power management mode the station will be in **after** the successful completion of the entire frame exchange sequence.

When operating under "doze" mode stations are not allowed to transmit or receive frames and therefore consume less power. Instead, stations must

periodically awake for a brief lapse of time in order to listen for *Beacon* frames. The *Beacon* frames are sent by the access point and contain the **TIM** (*Traffic Indication Map*), which indicates the stations, if any, for which the access point has buffered frames. Stations in power save must interpret the **TIM**[1] received and, if there are buffered frames for it, send a *PS-Poll* to the access point. When the access point receives the *PS-Poll*, it must either send the buffered frames immediately or acknowledge its reception and send the buffered frames at a later time.

The time diagram 1 shows the power save procedure in operation.

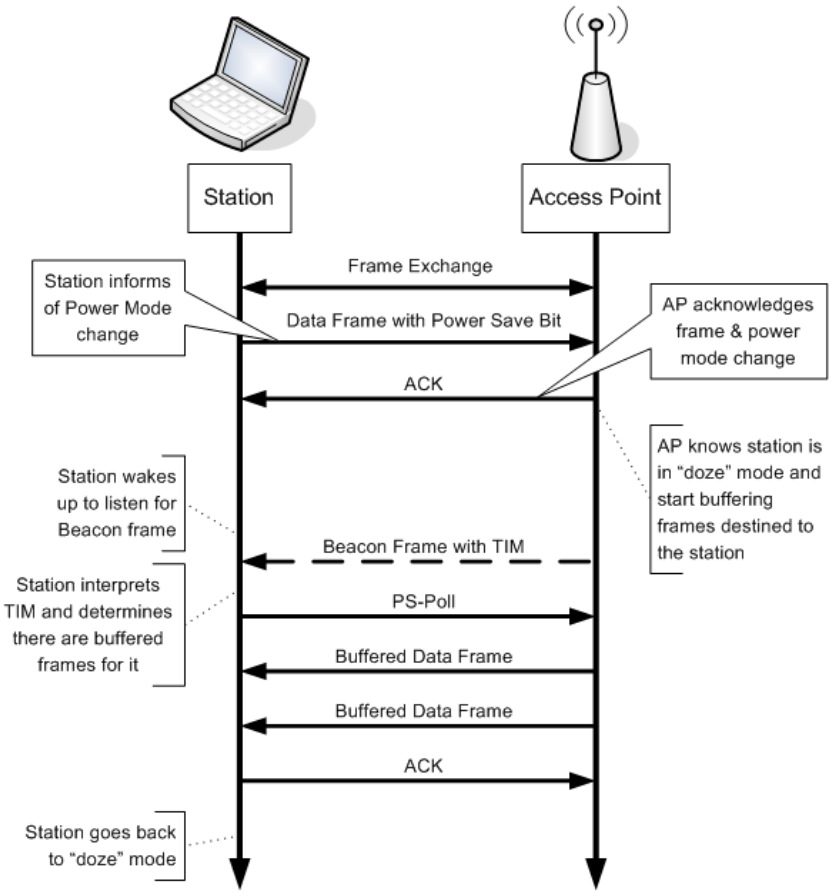

Figure 1: Power Save Time Diagram

[1] According to the standard stations operating in active mode don't have to do so.

2.3 Power Management bit

The *Power Management* bit indicates the power management mode the station will be in **after** the successful completion of the entire frame exchange, and remains constant during the exchange.

Figure 2 depicts the general **MAC** frame format of 802.11, which contains the *Frame Control* field.

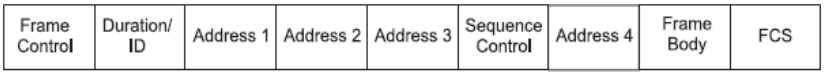

Figure 2: General 802.11 MAC frame format

Figure 3 depicts the *Frame Control* field, which contains the *Power Management* bit.

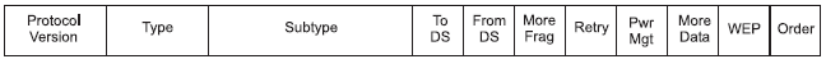

Figure 3: Frame Control field format

If the bit is set the station is indicating it will enter power save mode (i.e. "doze" mode).

3 Power Save Denial of Service

The idea behind the attack, as mentioned above, is to trick the access point into believing that the station under attack is in power save mode. Therefore, the access point will start buffering frames destined to the station, which results in a partial disconnection of the station from the network (the station can still transmit frames).

The beauty of this attack resides in the desynchronization that occurs between the station and the access point. Since the access point believes the station is in sleep mode it starts buffering inbound frames for it. Yet, the station, which isn't really in sleep mode, doesn't know the access point thinks it is in sleep mode and therefore, doesn't query the access point for its buffered frames.[2] The access point will eventually drop the buffered frames (the algorithm to do so is not specified in the standard, but it states that it must be done and refers to it as the "AP aging function").

The first step to carry out this attack is to find a way to trick the access point into thinking the station is entering power save mode. In order to accomplish this it's necessary to construct one or more frames that meet the following requirements and constitute a valid frame exchange sequence:

[2] According to the standard stations operating in active mode don't have to interpret the **TIM** in *Beacon* frames.

- Have the power save bit set.
- Have source MAC address as that of the victim station.
- Have destination MAC address as that of the access point.

These requirements imply that the attacker must be able to *spoof* ([4]) the frame(s) to appear as if they were sent by the victim station. To be able to spoof a frame it is preferable to find a frame which isn't a *data* frame and therefore doesn't have to be encrypted in a **WEP** or **WPA/WPA2** protected environment.[3] This reduces the potential candidates to be either *management* or *control* frames or empty data frames[4].

Another point to consider when selecting candidates is to select a frame sequence of one frame (i.e. consisting only of the frame being sent to the access point to notify it of the power mode change). If the sequence exchange consists of more than one frame, this increases the chances that a reply sent by the access point might allow the victim station to become aware of the power mode change or emit a reply thwarting the attack. Furthermore, since the power save mode can't be changed in the middle of a sequence the whole sequence must be carried out by the attacker. This has the consequence of making it infeasible to insert a frame with the power save bit set in the middle of a sequence exchange being carried out by the victim station and the access point.[5,6]

The following list contains the likely candidates, taking all the above into account:

- Request To Send (RTS)
- Reassociation Request
- Null function (no data)[7,8]
- Probe Request

All four frames will generate a response by the access point; the "Null function" frame will generate an **ACK**, the Reassociation Request an **ACK** and a Reassociation Response, the Probe Request an **ACK** and a Probe Response,

[3]Even though in **WEP** a frame can be replayed as is or with its header modified but its payload intact (since the header isn't encrypted) and therefore doesn't require knowing the secret key, **WPA/WPA2** has protection against replay attacks.

[4]Empty data frames aren't encrypted as their payload is empty.

[5]This is the reason why **ACK** frames were discarded as potential candidates.

[6]This is also the reason why **WEP** data frames aren't candidates, in spite of being possible to replay frames in a **WEP** protected environment. The procedure of listening for one sent by the victim station, setting the retry bit (so the frame is well formed as part of the sequence) and the power save bit and retransmitting it, would violate changing the power save mode in the middle of a frame exchange sequence.

[7]This frame is a data frame, yet it has no data and therefore is not encrypted.

[8]In the case that the wireless network is using the Quality of Service (**QoS**) feature supported by the standard a *QoS Null function (no data)* frame should be used instead, which is the equivalent in a networking supporting QoS functionality.

and the **RTS** a Clear To Send (**CTS**)[9]. The **ACK** responses will likely be ignored by the victim station, since they are handled directly by the network driver or the network card itself, and require little to no processing. However, the Reassociation response and Probe response might need to be acknowledged by the attacker (since the victim station might not do so).

It has been documented ([1]) that many access points ignore **RTS** and **CTS** (Clear To Send) frames. For example, many off-the-shelf access points, such as Linksys, include non-standard **RTS** thresholds. Furthermore, indicating that the station is going to enter power save mode in a request for medium reservation is a clear contradiction and might therefore be ignored by access points.

However, the other frame types are a crucial part of the connection process or data transmission and are therefore more likely to be complied to by an access point. Furthermore, the *Reassociation Request* frame has the advantage that it contains a listen interval field, which is used to indicate to the access point how often a station in power save mode wakes to listen to *Beacon* frames. According to the standard, "An access point may use the Listen Interval information in determining the lifetime of frames that it buffers for a station" [5]. Therefore, by setting a low value the attack might work even better because the access point might drop the frames faster.

It is also important to note that the 2007 version of the standard states "The Power Management bit shall not be set in any management frame, except an Action frame.". *Action frames* provide a mechanism for specifying extended management actions, and are therefore vendor-dependent making them a poor choice for this attack.

The pseudo-code for the attack is presented in figure 4.

The loop is performed to maintain the denial of service attack. The first *if* clause checks to see if the sniffed frame was sent by the victim station and doesn't have the power save bit set. This will be read by the access point as a power management mode change, and it will start forwarding future frames destined to the station once more. Therefore, it is necessary to make the access point believe that the station is going to go to "doze" mode again.

The second *if* clause checks to see if the access point is forwarding frames along to the victim. If this happens it means that the access point doesn't think the station is in sleep mode (otherwise it wouldn't send frames to it) or that a response for the frame used in the attack was sent by the access point. Therefore, as before, it is necessary to make the access point believe that the station is going to go to "doze" mode again or acknowledge the access point's response, respectively. This acknowledgment only occurs when using Reassociation Request and Probe Request frames for the attack, as RTS and Null Data frames don't elicit a response by the access point which must be acknowledged.

The time diagram 5 shows the power save denial of service, clearly distinguishing the two "views" of the power management state of the station; that of

[9]If the access point chooses to grant the requesting station usage of the medium. If not, no response will be sent.

- DoS_pkt = 80211_frame(src_mac = VictimMAC, dst_mac = BSSID, PowerBit = 1)
- DoS_pkt.setType(Data) // set appropriate frame type
- DoS_pkt.setSubType(Nullfunction) // set appropriate subtype
- ACK_pkt = 80211_ACK_frame(ra_mac = BSSID)
- sniffer_start()
- send(DoS_pkt)
- While True:
  - pkt = sniff_next_frame()
  - if (pkt.src_mac == VictimMAC && pkt.pwr_mgmt == 0):
    - send(DoS_pkt)
  - if (pkt.src_mac == BSSID && pkt.dst_mac == VictimMAC):
    - if (subtype(pkt).isResponse(subtype(DoS_pkt))):
      - send(ACK_pkt)
    - else:
      - send(DoS_pkt)

Figure 4: Pseudo-code for the Power Save Denial of Service

the access point and that of the station itself.

As mentioned before, all four frames present valid choices and the effectiveness of each will depend on the station and access point implementation. Yet, the best candidate seems the "Null function" frame since it doesn't require acknowledging a response (as do the Probe Request and Reassociation Request) and is more likely to be processed by the access point than an RTS frame. Furthermore, with the addition to the 2007 version of the standard (described above), only Null function frames seem viable as candidates for the attack.

From an attacker's perspective, in order to increase the chances of a successful attack, it could be wise to attempt with all four types, or simply construct all four frames and send them each time[10].

It is worth mentioning that due to the low overhead, for the attacker, of this attack, it can be extended to disrupt network services for the network by simply attacking each station that starts to transmit or joins the network.

[10]This has the drawback of making the attack easier to detect.

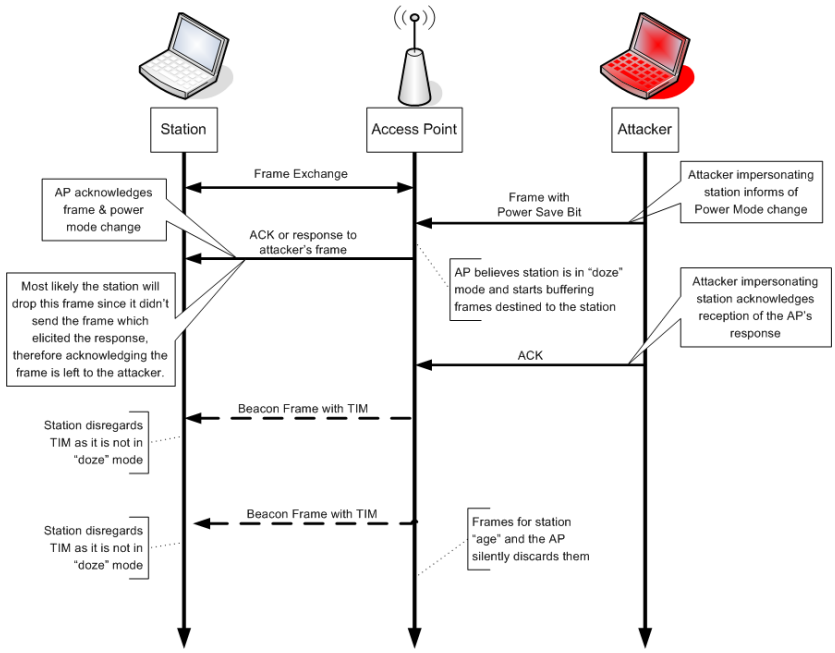

Figure 5: Power Save Denial of Service Time Diagram

3.1 Countermeasures

In order to resolve this class of attack, which are based on "identity" theft it is necessary to modify the 802.11 standard to include per-packet authentication. Since this is not going to happen in the short term, we propose modifying the standard in two different ways. The first implies changes to both the access point and the workstations, the other only to the station.

The first approach is for the access point to only process power change notification in non-empty data frames. This, coupled with the authentication and replay protection offered by **WPA/WPA2**, will mitigate this attack[11]. The problem with this approach is that in the case the station wants to go to sleep mode and has nothing to send, it will need to send a data frame for superfluous information, for example an *ICMP Echo request* or a *ARP Request* for the access point to be able to notify the access point of the power management mode change. The other issue with this approach is that it breaks access point compatibility with existing stations that send power changes with other frame types. The idea behind this approach is to eliminate all possible frame candidates for the attack.

The second approach would be to modify station's drivers so they interpret

[11] In open or **WEP** protected networks this might not be enough as valid data frames can be either constructed (only in open networks or if the attacker knows the **WEP** key) or altered and replayed.

the **TIM** even in active mode and, if there are frames buffered at the access point for the station, send the corresponding *PS-Poll* frame. This solution has the benefits that it doesn't require modifying existing access points and is backward compatible. Furthermore, this mitigation is independent of the encryption (which can be none) being enforced by the network. The idea behind this approach is to force the station to resynchronize with the access point. We implemented and successfully tested this countermeasure as detailed in section 4.

Both solutions would mitigate the attack, and serve as temporary solutions. Nevertheless, the second solution is clearly a wiser choice since it doesn't break current implementations and only requires changes in the stations. Therefore, we recommend the second as the course of action to take by driver developers . However, the standard should be modified to include per-packet authentication in order to completely thwart this attack as well as other denial of service attacks based on "identity" theft.

4 Results

We implemented the tool in GNU/Linux using *scapy* [3] and drivers patched for raw injection. We used the *Zydas 1211* kernel drivers with the patches developed by the *Aircrack-ng* ([6]) project.

The tool we implemented allows selecting the type of frame to send from the candidate frames discussed in section 3 entitled "Power Save Denial of Service".

We tested the all four frame types with the following three access point configurations:

- OPEN: No security settings.
- WEP: WEP encryption configured in the access point.
- WPA: WPA encryption in pre-shared key mode configured in the access point.
- WPA2: WPA2 encryption in pre-shared key mode configured in the access point.

In our tests, the only frame type that was consistently accepted by different access point vendors was the **Null function** frame. Our results also showed that the attack worked independently of the security settings of the access point.

We implemented our proposed countermeasure using the *r27t50* drivers for GNU/Linux and compared the attack efficiency in the same setups with and without the fix, which showed that our countermeasure was effective.

5 Conclusion

This paper addressed the availability aspect of the 802.11 protocol. We presented a denial of service attack against the protocol which, unlike previous denial of service attacks, doesn't require flooding the victim station. This makes the attack harder to detect and, since it requires fewer frames, permits the attacker to target multiple stations at once (i.e. it is more resource efficient from the attacker's perspective). In addition to detailing the attack, we presented a long term solution and a short term countermeasure, which we implemented and proved to be effective (with only a small lag being added to the responses to the station under attack, which was rarely noticiable).

6 Appendix I: Frame Types and subtypes of IEEE 802.11

The different types of frames supported by the protocol are described in table 1.

Frame type	Subtype	Description
Management	Association request	Request to join network
Management	Association response	Response to association request
Management	Reassociation request	Request to rejoin the network
Management	Reassociation response	Response to the reassociation request
Management	Probe request	Network availability request
Management	Probe response	Response to probe request
Management	Beacon	Access point information
Management	ATIM	For power save management in *ad hoc* networks
Management	Disassociation	Notification of "disconnection" from network
Management	Authentication	Authentication request or response
Management	Deauthentication	Notification of authentication relationship termination
Management	Action	Used to implement vendor specific management messages
Control	Block Ack Request (BlockAckReq)	Request to avoid using confirmation frames (ACK)
Control	Block Ack (BlockAck)	Response to Block Ack Request
Control	PS-Poll	Used by a station in power save mode to request pending frames buffered at the access point

continued on next page

continued from previous page		
Frame type	Subtype	Description
Control	RTS	Request to send (medium reservation)
Control	CTS	Positive response to RTS
Control	ACK	Reception acknowledgment
Control	CF-End	Signals the end of a contention-free period
Control	CF-End + CF-Ack	Signals the end of a contention-free period and confirms the reception of CF-Poll (this signals to the station that it can transmit a frame during a contention-free period)
Data	Data	Data frame
Data	Data + CF-Ack	Data frame with CF-Ack
Data	Data + CF-Poll	Data frame with CF-Poll
Data	Data + CF-Ack + CF-Poll	Data frame with CF-Ack y CF-Poll
Data	Null Function (no data)	Empty data frame
Data	CF-Ack (no data)	Empty data frame with CF-Ack
Data	CF-Poll (no data)	Empty data frame with CF-Poll
Data	CF-Ack + CF-Poll (no data)	Empty data frame with CF-Ack and CF-Poll
Data	QoS Data	Data frame which uses *QoS* functionality of the network
Data	QoS Data + CF-Ack	Data frame which uses *QoS* functionality of the network with CF-Ack
Data	QoS Data + CF-Poll	Data frame which uses *QoS* functionality of the network with CF-Poll
Data	QoS Data + CF-Ack + CF-Poll	Data frame which uses *QoS* functionality of the network with CF-Ack and CF-Poll
Data	QoS Null (no data)	Empty data frame which uses *QoS* functionality of the network
Data	QoS CF-Poll (no data)	Empty data frame which uses *QoS* functionality of the network with CF-Poll
Data	QoS CF-Ack + CF-Poll (no data)	Empty data frame which uses *QoS* functionality of the network with CF-Ack and CF-Poll

Table 1: Different frame types and subtypes

7 Acknowledgments

We would like to thank Diego Sor, for his discussions on the attack, and Gerardo Richarte and Ezequiel Gutesman for their valuable support and guidance in writing this paper.

References

[1] John Bellardo and Stefan Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the 12th USENIX Security Symposium*, pages 15–28. USENIX, August 2003.

[2] Massimo Bernaschi, Francesco Ferreri, and Leonardo Valcamonici. Access points vulnerabilities to DoS attacks in 802.11 networks. *Wireless Networks*, 14(2):159–169, 2008.

[3] Philippe Biondi. Scapy.

[4] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications: the insecurity of 802.11. In *MOBICOM*, pages 180–189, 2001.

[5] IEEE. *IEEE Std 802.11-1999, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, Edition 1999.

[6] The Aircrack ng team. Aircrack-ng project.

[7] Mika Sthlberg. Radio jamming attacks against two popular mobile networks, 2000.