Wireless Intrusion Detection and Response

A case study using the classic man-in-the-middle attack

Timothy R. Schmoyer, Yu Xi Lim and Henry L. Owen
School of Electrical and Computer Engineering
Georgia Institute of Technology
Altanta, GA 30332-0250
{schmoyer,owen}@ece.gatech.edu, yuxi@gmx.net

Abstract— Intrusion detection and countermeasures response is an active area of research. In this paper, we examine integrating an intrusion detection engine with an active countermeasure capability. We use a classic man in the middle attack as a case study to specify the integrated wireless intrusion detection capability with the active countermeasure response. We present the case study in dynamically defending against an example attack in an 802.11 infrastructure basic service set by combining the concepts for a distributed wireless intrusion detection and response system architecture with adaptive response strategies based on alarm confidence, attack frequency, assessed risks, and estimated response costs. We also include a description of a tool kit we have implemented to prototypically test and evaluate our concepts.

Keywords- Intrusion detection and active countermeasures, network Security, wireless security

I. INTRODUCTION

There has been a great deal of both research and commercial activity in wireless network security. The vulnerabilities associated with the IEEE 802.11 standards are now widely known and widely exploited. New standards activities and use of additional technologies such as virtual private networks and key rotation schemes make "secure" use of wireless networks in most networking environments possible. New commercial wireless devices and existing ones that allow firmware upgrades will likely be able to implement standardized secure techniques that result from these efforts. However, due to the prolific deployment of wireless devices at present, there will be a large number of legacy wireless devices that continue to be used even after the new standards are deployed in new and up-gradable devices. For example, the IEEE 802.11i draft standard includes capabilities to connect with legacy wireless infrastructure with the warning that the entire infrastructure security may possibly be compromised in such hybrid infrastructures. Thus, there is still a need for strengthening the security in the presently deployed wireless infrastructure.

Predictably, wireless management and Intrusion Detection Systems (IDS) have grown in popularity to provide visibility of the wireless infrastructure and enable administrators to perform network management and security functions. Like security and management systems for the wired network infrastructure, they attempt to monitor all the traffic on the wireless network, detect intrusions, block attackers, and maintain a level of service to authorized clients.

We are examining an architecture where every node uses an IDS agent to monitor local activity and respond to intrusions. Since local activity does not always provide sufficient data to detect or determine the type of an attack, local agents should be able to communicate securely and act collectively when an intrusion is suspected. By using a distributed and cooperative architecture, we can increase the effectiveness of wireless intrusion detection and response systems.

We have taken existing 802.11 threat matrix and explored what electronic countermeasure techniques may be employed against wireless attacks to minimize or prevent security breaches. Algorithms for recognizing attacks as well as the algorithms for countering those attacks are being explored [1]. An experimental implementation and demonstration prototype with limited proof of concept functionality is being used to experimentally evaluate the proposed counter measures techniques.

II. DISTRIBUTED INTRUSION DETECTION AND RESPONSE

A. System Architecture

Reference [2] proposed a distributed wireless intrusion detection and response system architecture, shown in Fig. 1, based on the needs of mobile ad hoc networks. The strength of this architecture is its modular design and ability for each node to detect signs of intrusion locally and independently, as well as allowing neighboring nodes to collaborate in detecting intrusions and developing response strategies.

While this distributed, decentralized architecture is targeted for ad hoc networks, there is no constraint preventing its use on infrastructure networks as well. This is especially attractive given the same device may act as a client in an infrastructure network and as a node in an ad hoc network at different times. The same intrusion detection agent can then be used in either role by applying appropriate security policies for that mode and service area.

The methods for collecting data, local detection, and local response are independent of each other and the other IDS agents in the network. A secure communications channel must be standardized among IDS agents to communicate and perform cooperative detection and coordinate global response

strategies. This means there must be agreement, or knowledge, of IDS type efficiencies, the meaning of confidence levels if shared, and the costs of attacks on the shared network resources.

The intrusion detection system discussed in [2] was anomaly based and provided experimental results exploring the performance of anomaly based detection using different ad hoc routing protocols. It is useful to note that the architecture is not limited to using anomaly or signature based intrusion detection systems, or a hybrid of both. In addition, the focus of [2] was primarily on detection and did not explore the response methodology in significant detail.

B. Adaptive Response to Minimize Risk

Reference [3] presented a network security model for dynamic intrusion detection and response. The model provided a mathematical approach to quantify intrusion detection efficiency, risk and cost. The measures can be used with thresholds either set by the security administrator or based on learned behavior so as to select predefined response strategies.

We use the same approach as in [3] but with different type of attacks on wireless networks. We include the concepts of local detection efficiency and efficiency improvement gained through cooperative detection, assessment of the risk to the network and client from the attack, and the cost to the network and client for available responses considering any residue risk after countermeasures have been applied.

This novel approach, combining a distributed wireless intrusion detection architecture with the dynamic intrusion response model, provides significant flexibility to study numerous attacks and their associated responses using different intrusion detection systems under different network conditions, including infrastructure and ad hoc wireless network modes.

Our case study begins using a classic man-in-the-middle (MITM) to demonstrate implementation requirements, how to rate the effectiveness of the system against attacks, and as a platform to present future areas of wireless intrusion detection and countermeasure response research.

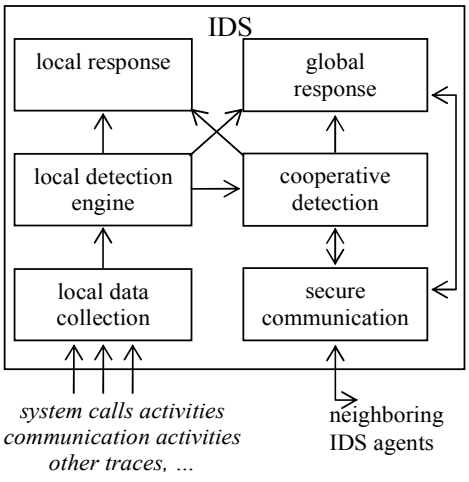


Figure 1. A conceptual model for an IDS agent [2]

III. THE CLASSIC MAN-IN-THE-MIDDLE ATTACK

Mike Lynn and Robert Baird demonstrated attacks on the 802.11 protocol at the 2002 Blackhat conference in Las Vegas, Nevada [4]. Among these attacks was the "Monkey-Jack" attack, which was a MITM attack to insert the attacker between a client and an AP in a wireless infrastructure basic service set (BSS).

The attack begins by sending an optional number of deauthentication frames to the victim client with the source address and BSSID set to the AP address. The attack also sends an optional number of disassociation frames to the victim client. The attack then changes the wireless interface to a different channel and transmits beacon frames spoofing the parameters for the targeted BSS.

If the victim client was successfully deauthenticated, it will begin scanning for APs. This MITM attack exploits the weak security of the joining, open authentication and association protocol in 802.11 as it exists today. In the joining process, the common criteria that client implementations use to choose which BSS to join is power level and signal strength. The attacker can locate closer to the victim client than the previously associated AP or use high gain antennas to have a higher received power level and signal strength at the victim client. The attack listens for an authentication request from the client on the new channel, followed by an association or re-association request and responds appropriately to each one.

After associating the victim client on a new channel, the attacker uses a second interface on the original channel to associate to the BSS, spoofing the victim client MAC as the source address. Packets are then passed from one interface, on one channel, to the other interface on a different channel. More details on the "Monkey-Jack" attack are available in [4].

IV. DETECTION

We were able to determine a simple set of rules to detect the MITM attack locally at the client and at the AP in order to study how cooperative detection and a secure communication channel can be used to improve the alarm confidence that a specific attack is in progress, and distinguish that attack from other attacks.

At the client, the first frame received from the attack will be either a deauthentication or disassociation frame. The sequence number of the frame will not match the sequence set used by the AP. This is expected since the sequence number is controlled by the firmware in available MAC implementations, and therefore not settable by the attacker [5]. As additional frames are received, the frequency of the frames can be analyzed to determine anomalous behavior and distinguish between a Rouge AP or MITM attack, and a deauthentication Denial of Service (DoS) attack [6].

The deauthentication frames being sent by the attack are also spoofing the AP's MAC address as the source address. If the detection engine shares the interface with the AP, the AP's MAC address should never be received as the source address by the AP since the transmit/receive switch on the transceiver would prevent it. If the detection engine is using a different interface, it would need to be able to query the firmware to

confirm whether the AP sent the frame. The byte fields in the 802.11 management frame that are tested in order to implement our client and AP detection rules are shown in Fig. 2. The initial stages of the MITM attack and the alarms generated locally at the client and AP are shown in Table I. In Table I, Seq# means sequence number.

Figure 2. 802.11 management frame

Up to this point the client and AP have detected the traits of the MITM attack independently. We can improve the alarm confidence at both the client and the AP by exchanging alarm information**.** For example, when the client generates a local alarm that a sequence number violation has occurred, the client can query the AP securely to verify the sequence number last sent by the AP. The client can also query the AP's authentication ID table to verify whether client is no longer associated or authenticated to the AP. Additionally, the client can update neighboring detection engines concerning the likelihood that an attack is in progress, which type of attacks are suspected with a confidence factor and which MAC addresses are involved.

The process of communicating to cooperatively detect an attack and generate cooperative alarms is shown in Table II. In Table II, Seq# violation (Deauth) means that a deauthentication frame was received by the client that did not match the sequence established previously by the AP. As a result, the client sends an update to other detection engines in the BSS ({w/ p%, client, Frame Insertion, AP}) providing the alarm confidence (w/ p%), the source node of the alarm (client), the type of alarm (Frame Insertion) and the source address of the frame causing the alarm (AP). More information concerning the information that might be exchanged between IDS agents is discussed in [2]. This enables the client and the AP to improve their confidence that an attack is in progress, what kind of attack(s) are suspected and notify neighboring IDSs of the level of risk in the operating environment. This also forms the foundation for local and global responses to defend against an attack. The ability to communicate securely is paramount to the usefulness of cooperative detection, and coordinated global responses. We are exploring the use of SNMP v3 as a means to communicate between agents. This method was chosen to ease integration with network management systems and take advantage of an existing protocol/system.

TABLE I. LOCAL DETECTION RULES

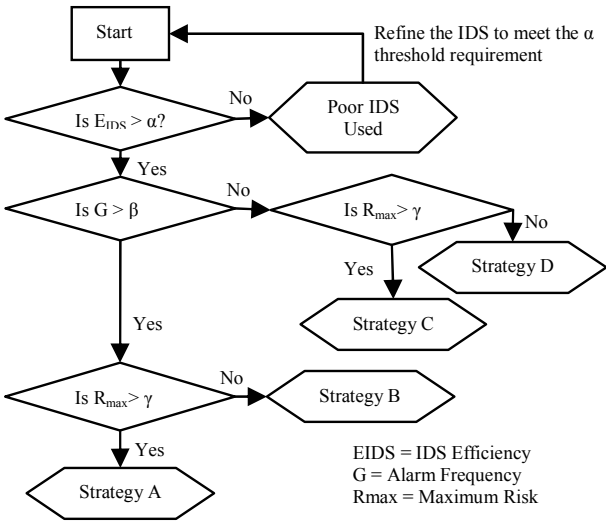
Monkey-Jack Attack Stages	Local Client alarms	Local AP alarms
Chosen number of Deauth and/or Disassoc frames sent to client	1. Seq# violation alarm on first frame 2. Anomalous # of Deauths/msec (DoS vs. MITM)	My MAC in SrcAddr (in my channel)
Auth_Resp Assoc_Resp Reassoc_Resp		My MAC in SrcAddr (in different channel)

TABLE II. COOPERATIVE DETECTION AND ALARMS

Local Alarms	Cooperative Detection	Cooperative Alarms
Seq# violation (Deauth)	{w/ p%, client, Frame Insertion, AP} GetRequest Seq# from AP	
	GetResponse Seq# from AP {w/ p%, client, Frame Insertion, AP} GetRequest AuthStatus from AP GetResponse AuthStatus from AP {w/ p%, client, Deauth DoS, AP} {w/ p%, client, Rogue AP, AP}	Seq# violation (Deauth)
Deauth(s)	{w/ p%, client, Deauth DoS, AP}	

V. RESPONSE STRATEGIES

The local response engine receives alarms from both the local detection engine and the cooperative detection engine as the attack progresses, and as the cooperative detection engine communicates with other agents in the BSS. The response engine selects a response strategy based on performance thresholds such as the performance of the IDS, the alarm confidence, and the maximum risk threshold, as shown in Fig. 3. The alarm frequency is used to determine between two groupings of response strategies. The strategy selected within a group is chosen based on the maximum risk threshold.



Selection among four intrusion response strategies where α, β, and γ are performance thresholds set by the IRS or by the security officer.

Figure 3. Response algorithm [3].

The efficiency of the IDS is derived based on all possible attacks and false alarms raised during an observation period, such that: [3]

$$E_{IDS} = QH / H + QM), \quad (1)$$

where E_{IDS} is the efficiency of the IDS,
Q is the Alarm Confidence,
H is the Average Detection Hit Rate, and
M is the Average Detection Miss Rate.

The alarm confidence, Q, is a quality indicator of the IDS system over the entire attack set defined as:

$$Q = x/G, \quad (2)$$

where x is the summation of all detection hits based on an empirically generated alarm matrix as shown in Fig. 4 and
G is the alarm frequency defined as the summation of all raised alarms in the monitoring period ($G = u + x + z$) denoted in the alarm matrix as false positives (u), hits (x) and confused alarms (z).

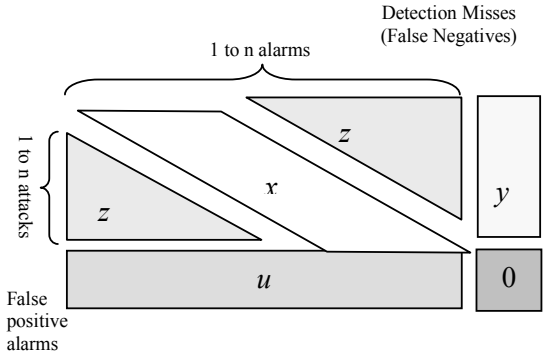


Figure 4. Algorithm matrix [3].

The average detection hit rate, H, is defined as:

$$H = x/F, \quad (3)$$

where F is defined as the attack frequency and is the summation of all real attacks within the monitoring period ($F = x + y + z$) including false negatives (y).

Finally, the average detection miss rate, M, is defined as:

$$M = y/F. \quad (4)$$

In Fig. 3, the alarm frequency, G, is used to determine between two groupings of response strategies. The strategy selected within a group is chosen based on the maximum risk, R_{max}. In order to determine the maximum risk, the average damage, D, must first be determined over all attack types. This can be expressed as a dollar amount by estimating the monetary loss for each attack type if successful, then calculating the average over all attack types. The maximum risk multiplies the attack damage by the corresponding attack frequency and is defined as:

$$R_{max} = D \, x \, F. \quad (5)$$

The response strategy can be implemented in a response matrix, as shown in Table III, where m responses are mapped to n attacks. The same response may be effective against multiple attacks, and an attack may have more than one effective response.

The IDS efficiency for each attack/response pair can then be measured and tabulated as the Hit efficiency and False/Miss efficiency based on learned behavior at the specific location, or previously tested results depending on the type of attack and response.

TABLE III. RESPONSE MATRIX.

Response	Attack	Hit Efficiency	False/Miss Effectiveness	Network Cost	Client Cost	p% of Attack	Response Threshold
Fake AP	Active Scanning	60%	60%	30%	20%	0%	70%
Fake Weak IV	WEP Cracking	0%	80%	30%	20%	0%	75%
Null SSID	Active Scanning	50%	0%	5%	5%	0%	40%
m responses	*n attack types*	*IDS Efficiency*		*Cost (One for Response and a second dual column for attack damages)*		*alarm confidence*	

A cost associated with each response must then be determined and recorded for each attack/response pair. A response which degrades access to the wireless network for a period of time would have an associated "Network cost" whereas a response which adds a MAC address to the client's access control list might have a minimal "Client cost" and no cost to the network. This cost can be used by the response engine and/or security officer to arrange response strategies based on a security policy, preference for response strategies and tolerance of risk.

The potential damage from the attack must also be calculated using the cost to the network and client. This allows the response engine, or security officer, to compare the cost of responding, or not responding, to a particular attack using a defined strategy. The response engine must always chose response strategies with a lower cost than the potential damage of a suspected attack; however the cost to the network and client can be weighted to reflect the security policy. For example, the response engine may choose not to respond to an attack with a high client cost if the response has a high network cost.

The alarm confidence is a variable that measures the probability of a "Hit" or real positive in detecting the attack versus a false positive or false negative ("Miss"). As the attack progresses, the local detection and cooperative detection engines will update the alarm confidence cells in the table for suspected attack(s).

Finally, the response threshold is used by the response engine to trigger a response strategy. The response threshold can be set by security officer or be based on a learning algorithm that establishes a "normal" threshold and triggers on anomalous behavior. For example, in a noisy BSS where frames are frequently lost, it may not be unusual to receive sequence number violations. Therefore, the response threshold may be set higher by the response engine based on this learned behavior than in a quieter static BSS. The algorithm for determining the response threshold should also take into account the effectiveness of the response, likelihood of a hit, cost of the response and potential damage of attack. Global response strategies are selected cooperatively by the intrusion response engines and use the same response table shown above. The cooperative detection engine updates or weights the alarm confidence to "turn on" a response that has been coordinated and "turn off" local responses that become ineffective during a global response.

VI. PROTOTYPE IMPLEMENTATION

A prototype was built upon a custom toolkit, shown in Fig. 5, which itself was intended as a low-level interface for the wireless hardware. Designed to run on a Linux PC, the toolkit uses readily available open source components. Host AP, Airjack and wlan-ng are freely available Linux drivers for 802.11b network cards utilizing the PRISM chipset. Libnet is a network library used to create data-link, network and application layer protocol headers and transmit the resultant frames. Since the library does not natively support IEEE 802.11b, it was modified to be able to generate the required 802.11b headers as well as hardware-specific PRISM headers. The modified version is identified as libnet-wireless. Finally, libpcap is a popular cross-platform packet capture library. It provides basic functionality for packet capture, time stamping, logging, and playback. Its wide support was the primary reason for its selection.

The toolkit was divided into separate modules, each designed to provide specific functions for intrusion detection and response. In addition, the multi-layer architecture allows for a hardware-independent API for the programmer. The WLAN control module provides functions for interacting with the hardware, such as setting the SSID, channel, or mode. The frame capture and decoding module interfaces with libpcap and gives the programmer a buffered packet capture system and is also able to extract 802.11b fields from all frames, including invalid frames that may have been generated by a malicious attacker. The frame builder module simplifies frame generation by providing templates for several common frames. It in turn passes the parameters on to libnet for frame creation and transmission. The bridging module interfaces with both the frame capture and builder modules. It is used primarily for MITM attacks and handles the bridging of frames between disparate wireless networks.

The toolkit did provide a reasonable level of abstraction from the drivers, but more work is required to support the wlan-ng drivers. The toolkit also eased the creation of several trivial attacks, such as DoS attacks, and simple attack detection programs. However, the more complicated MITM attack brought out the inadequacies in the current toolkit implementation. A significant deficiency was in the bridging module which was only able to handle data frames. Extending the module to handle all frames is complex because of the variety of possible frames and such including such functionality would make the bridging module no different from the hardware-specific driver it is trying to abstract. Another problem was the lack of control of all the fields in the 802.11b header – most importantly was the sequence numbers. Because of these two obstacles, the existing implementation of the MITM attack has an easily detectable signature.

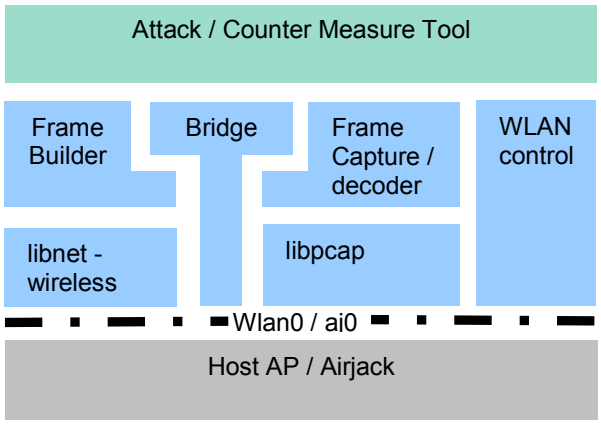


Figure 5. Protoype toolkit

Inadequate control of the lower-level access of the wireless hardware is the root cause of the problems. It is likely that migrating to thin MAC-based hardware such as newer

802.11a/b/g combo cards would alleviate the situation by allowing lower-level access to the hardware while simultaneously handling many higher-level functions automatically in the driver.

VII. CONCLUSION

By combining the concepts proposed in [2] and [3], we have developed a framework for a distributed, cooperative intrusion response engine that utilizes adaptive response strategies based on alarm confidence, attack frequency, assessed risks, and estimated response costs. We have defined a response matrix and described secure communications required between cooperative detection engines. We have developed a prototype environment by creating a customized toolkit for detecting attacks and sending 802.11 response frames.

We are still exploring how to use SNMP v3 as a means of communicating securely between the client and AP as part of theis wireless intrusion detection and response system. To communicate using SNMP v3 requires additional MIB data. We are looking into extending the 802.11 MIB to incorporate the Attack-Response policies and to possibly store specific packet information to supplement detection capabilities of the IDS. In the event the IDS is overloaded in the first stage of an attack, specific packet information could be retained and analyzed for evidence of an attack. We propose to extend the Station Management branch of the 802.11 MIB by adding an Attack-Response branch.

We are also working on how the local SNMP agent will be collocated with the IDS and how MIB data in the Attack-Response table can be used for secure communication. Our assumptions are that the IDS will write the shared data for the Attack-Response MIB to a flat file. Both the IDS and the SNMP agent need read-write access to the MIB data stored in the flat file.

We need to implement the agent for the Attack-Response MIB so that it stores the necessary data as well as performs backup analysis for the IDS. Once the attack confidence level crosses a predetermined threshold, the agent should respond by alerting the IDS and by sending a SNMP trap to other SNMP managers on the network.

To act cooperatively, managers should be able to update other agents' attack confidence using an SNMP set without overriding the local level, setting a global attack confidence level instead. The global attack confidence variable in the Attack-Response table allows this communication. The response to an attack should take into account both the local and global attack confidence levels.

Managers should also be able to query (SNMP getRequest) other managers for their local attack confidence. While the local attack confidence levels may be below the threshold, if a sufficient number of managers have higher than normal levels, some response may be warranted depending upon the specific attack.

One key component to wireless security is management and policy enforcement. This research does not examine this aspect of security even though management and policy are a major component of effective wireless security. Thus integration with Network Management Systems should be included in our proposed approach to security.

This research does not at present include an examination of new standards and implementations which are forecast to significantly increase security in wireless networks. These new standards typically assume that all existing infrastructure is to be updated or replaced as the new standards are implemented. Our research assumes that not all existing infrastructure will be upgradeable or replaced, thus there is still a need to find ways to increase existing infrastructure security.

In our case study of wireless network security, we do not at present examine location-based techniques. Thus techniques incorporating Global Positioning System, Line of Bearing/Triangulation, RF Mapping, Location Aware/Enabled Services are not examined. The use of spatial nulling using smart antenna arrays (SDMA), Frequency nulling using DSP filtering, and Code nulling (Change DSSS Barker code) are also not examined even though they are potential areas of research for increasing wireless security.

Finally, layer 3-7 detection and protection measures applicable to wired and/or wireless networks are not examined since they are not specifically layer one and layer two techniques which are the focus of this research.

The focus of this research is to examine how cooperative intrusion response engines with adaptive response strategies can increase wireless security. We are presently examining detection methods for wireless attacks, measuring response strategy effectiveness against specific attacks, measuring network and client costs, developing a wireless security MIB based on the response matrix and using SNMPv3 to communicate between cooperative detection engines.

REFERENCES

[1] Y. X. Lim, T. Schmoyer, J. Levine, H. Owen, "Wireless intrusion detection and response," IEEE 4th Annual Information Assurance Workshop, West Point N.Y., June 2003, pp. 68-75

[2] Y. Zhang, W. Lee, Y. Huang, "Intrusion detection techniques for mobile wireless networks", ACM WINET 2003 Vol. 9, No. 5, September 2003, pp. 545-556.

[3] S. Tanachaiwiwat, K. Hwang, and Y. Chen, "Adaptive Intrusion Response to Minimize Risks over Network Attacks", ACM Trans. on Information and System Security, unpublished.

[4] M. Lynn and R. Baird, "Advanced 802.11 Attack",Blackhat 2002, Las Vegas, NV, available online http://www.blackhat.com/html/bh-usa-02/bh-usa-02-speakers.html#Baird, 31 July 2002.

[5] J. Wright, "Detecting Wireless LAN MAC Address Spoofing", available online, http://home.jwu.edu/jwright/, 21 January 2003.

[6] J. Bellado, "802.11 Denial-of-Service Attacks: Real Vulneratilites and Practical Solutions", Proceedings of the USENIX Security Symposium, August 2003, pp. 15-28.