The Modeling and Comparison of Wireless Network Denial of Service Attacks

Martin Eian
Department of Telematics
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
martin.eian@item.ntnu.no

Stig F. Mjølsnes
Department of Telematics
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
stig.mjolsnes@item.ntnu.no

ABSTRACT

Mobile handhelds with wireless access are used in numerous safety critical applications. The wireless network protocols in use are vulnerable to a wide array of denial of service attacks. We propose a formal method for modeling semantic denial of service attacks against wireless network protocols. We then use our proposed model to find a new deadlock vulnerability in IEEE 802.11. The history of published denial of service attacks against wireless protocols indicates that formal methods can contribute to the construction of robust protocols.

1. INTRODUCTION

The use of mobile handhelds in safety critical applications is increasing. Such devices are used in life critical medical systems, intelligent transport systems (ITS), emergency communications and alarm systems. Furthermore, the current trend is to use standard commercial off the shelf (COTS) equipment and protocols in safety critical applications. Such safety critical applications require that the wireless networks used for communication by the mobile handhelds are available when needed.

The availability of a wireless network can be disrupted by denial of service (DoS) attacks. An adversary mounting a DoS attack on a wireless network used in safety critical applications could cause injury or death, as well as significant material damage.

MobiHeld '11, October 23, 2011, Cascais, Portugal.

We divide wireless network DoS attacks into four categories: *Jamming* attacks, *flooding* attacks, *semantic* attacks and *implementation specific* attacks. *Jamming* attacks are mounted by transmitting noise in the radio frequencies used by the wireless network. *Flooding* attacks exhaust resources by sending a large amount of messages to a protocol participant. *Semantic* attacks exploit protocol weaknesses by transmitting valid protocol messages with forged message fields. One example of a semantic attack is the deauthentication attack against IEEE 802.11 networks [1]. Finally, *implementation specific* attacks target implementation vulnerabilities in specific hardware or software. This category of attacks includes transmitting invalid protocol messages, since a correct implementation should discard all invalid messages. Implementation specific attacks are not related to the protocol design and will not be considered further in this paper.

The link layer protocols used in current wireless networks are vulnerable to semantic DoS attacks. Vulnerable protocols are used in 802.11 wireless local area networks (WLANs) [1, 2, 11, 3], 802.16 broadband networks [4], and GSM and UMTS mobile networks [10]. Due to functionality, performance or cost requirements, certain signaling messages used by these protocols are not integrity protected. One obvious solution to this problem is to integrity protect every message. However, a wireless network needs to exchange signaling messages, e.g. signal strength measurements, before it starts the authentication and key agreement (AKA) procedure. These initial signaling messages cannot be protected using the keys derived from the AKA procedure. Furthermore, small battery powered devices might not have the resources to verify the integrity of every message. The integrity protection of time critical signaling messages in wireless

networks might limit the usable bandwidth. The functionality of the wireless network might even depend on the use of unprotected messages, e.g. a network that has to provide service to clients that do not support any security mechanisms. As a consequence, these link layer protocols only protect a subset of all protocol messages. Any unprotected message could potentially be used by an adversary to disrupt the wireless network. Even the security mechanisms in the long term evolution (LTE) fourth generation (4G) mobile networks only protect a subset of the protocol messages [20].

A significant research effort has been invested in making network protocols more robust against *jamming* and *flooding* attacks. The research results are formal methods, models, tools and design principles to aid protocol designers [18, 16, 19, 21, 22, 13, 12]. The research effort on *semantic* DoS attack detection and prevention has been of a more informal nature. A multitude of semantic DoS attacks have been discovered in existing wireless protocols, practical attacks have been demonstrated, and ad hoc countermeasures have been proposed [1, 2, 11, 3]. The fact that new semantic vulnerabilities are routinely being discovered in proposed and operative protocols indicates that it is difficult to avoid protocol vulnerabilities. To improve the robustness of wireless networks used in safety critical applications, we need to verify that an adversary cannot exploit unprotected protocol messages to disrupt the service provided by the protocol.

Narayana et. al. proposed a formal model for evaluating *semantic* DoS attacks in 2006 [17]. To the best of our knowledge, this is the *only* proposed model of semantic DoS attacks in the literature prior to our work. Narayana et. al. use the temporal logic of actions (TLA+) to model the protocol and adversary, and to specify the model properties. They then apply the TLA+ model checker (TLC) to find DoS vulnerabilities. They define a DoS attack as a situation where the protocol participants cannot reach their final state. Our proposed model has three major contributions compared to their model. First, we propose a cost model. The cost model provides an objective quantification of the severity of the protocol vulnerabilities. Second, our model is able to detect scenarios where participants reach their final state, and are then desynchronized by a new attack. This kind of attack will not be detected in the model proposed by Narayana et. al., since the participants in this case are able to reach their final state. Third, we demonstrate the usefulness of our model through the detection and experimental validation of a previously unknown deadlock vulnerability in the 802.11 standard.

Figure 1: Protocol model with initiator and responder. The adversary is modeled as the network. The adversary can read, replay or forge every unprotected protocol message. The adversary cannot delete or intercept (i.e. read and delete) protocol messages.

In this paper we propose a formal method and model for evaluating wireless network protocol vulnerabilities to semantic DoS attacks. We analyze the adversary goals to find an appropriate quantification of the adversary cost. We then quantify the the protocol participant cost, and propose an attack efficiency definition. Finally, we use our model to discover a new deadlock vulnerability in the IEEE 802.11 family of standards. The proposed formal method is not protocol specific, it can be used to analyze any wireless protocol.

Our work complements the research efforts to make wireless networks more robust against jamming and flooding attacks. A network must be robust against all categories of DoS attacks. If one category of attacks is not addressed, then an adversary may still be able to disrupt the network.

The rest of this paper is structured as follows: Section 2 presents the formal protocol model and adversary model. Section 3 presents the cost model. Section 4 presents our model of IEEE 802.11 and Section 5 presents the experimental results. Finally, Section 6 gives the conclusions.

2. PROTOCOL AND ADVERSARY MODEL

We model a two party protocol $\boldsymbol{P}$ as illustrated in Figure 1. The protocol participants are the initiator $\boldsymbol{I}$ and responder $\boldsymbol{R}$.

Each participant has a set of local protocol states. We model $\boldsymbol{I}$ and $\boldsymbol{R}$ as *deterministic* finite state transducers defined by the 7-tuples

$$\boldsymbol{I} = (\Sigma, \mathrm{S}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{I}_0}, \delta_{\boldsymbol{I}}, \omega_{\boldsymbol{I}}, \gamma_{\boldsymbol{P}}, \mathrm{F}_{\boldsymbol{I}})$$

$$\boldsymbol{R} = (\Sigma, \mathrm{S}_{\boldsymbol{R}}, \mathrm{s}_{\boldsymbol{R}_0}, \delta_{\boldsymbol{R}}, \omega_{\boldsymbol{R}}, \gamma_{\boldsymbol{R}}, \mathrm{F}_{\boldsymbol{R}})$$

where:

- ϵ is the empty string
- $\Sigma = \{\sigma | \sigma \text{ is a protocol message}\} \cup \{\epsilon\}$

- $S_{\boldsymbol{I}} = \{s_{\boldsymbol{I}} | s_{\boldsymbol{I}} \text{ is a protocol initiator state}\}$
- $S_{\boldsymbol{R}} = \{s_{\boldsymbol{R}} | s_{\boldsymbol{R}} \text{ is a protocol responder state}\}$
- $s_{\boldsymbol{I}_0} \in S_{\boldsymbol{I}}$ is the protocol initiator initial state
- $s_{\boldsymbol{R}_0} \in S_{\boldsymbol{R}}$ is the protocol responder initial state
- $\delta_{\boldsymbol{I}} : S_{\boldsymbol{I}} \times \Sigma \rightarrow S_{\boldsymbol{I}}$ is the initiator state transition function
- $\delta_{\boldsymbol{R}} : S_{\boldsymbol{R}} \times \Sigma \rightarrow S_{\boldsymbol{R}}$ is the responder state transition function
- $\omega_{\boldsymbol{I}} : S_{\boldsymbol{I}} \times \Sigma \rightarrow \Sigma$ is the initiator output function
- $\omega_{\boldsymbol{R}} : S_{\boldsymbol{R}} \times \Sigma \rightarrow \Sigma$ is the responder output function
- $\gamma_{\boldsymbol{I}} : S_{\boldsymbol{I}} \times \Sigma \rightarrow \mathbb{R}^+$is the initiator cost function
- $\gamma_{\boldsymbol{R}} : S_{\boldsymbol{R}} \times \Sigma \rightarrow \mathbb{R}^+$is the responder cost function
- $F_{\boldsymbol{I}} \subseteq S_{\boldsymbol{I}} = \{s_{\boldsymbol{I}} | s_{\boldsymbol{I}} \in S_{\boldsymbol{I}} \text{ and } \boldsymbol{P} \text{ provides service}\}$
- $F_{\boldsymbol{R}} \subseteq S_{\boldsymbol{R}} = \{s_{\boldsymbol{R}} | s_{\boldsymbol{R}} \in S_{\boldsymbol{R}} \text{ and } \boldsymbol{P} \text{ provides service}\}$

The service provided by a protocol depends on the protocol's purpose. One common service is the transport of higher layer user data. The protocol $\boldsymbol{P}$ is in a state $(s_{\boldsymbol{I}}, s_{\boldsymbol{R}})$ where it provides service when $(s_{\boldsymbol{I}}, s_{\boldsymbol{R}}) \in (F_{\boldsymbol{I}} \times F_{\boldsymbol{R}})$.

We model an adversary $\boldsymbol{A}$ who can read, replay or forge every unprotected protocol message $\sigma_{\boldsymbol{A}} \in \Sigma_{\boldsymbol{A}}$. The adversary cannot delete or intercept (i.e. read and delete) messages. These capabilities correspond to a real world adversary who utilizes commercial off the shelf hardware and software. An example of such an adversary is someone with an 802.11 network interface card with driver support for monitor mode and frame injection. When not attacking, the adversary simply forwards all messages between $\boldsymbol{I}$ and $\boldsymbol{R}$. We model the attack behavior of the adversary as a *nondeterministic* finite state transducer defined by the 7-tuple

$$\boldsymbol{A} = (\Sigma_{\boldsymbol{A}}, S_{\boldsymbol{A}}, s_{\boldsymbol{A}_0}, \delta_{\boldsymbol{A}}, \omega_{\boldsymbol{A}}, \gamma_{\boldsymbol{A}}, F_{\boldsymbol{A}})$$

where:

- $\Sigma_{\boldsymbol{A}} \subseteq \Sigma = \{\sigma_{\boldsymbol{A}} | \sigma_{\boldsymbol{A}} \in \Sigma \text{ and } \boldsymbol{A} \text{ can forge } \sigma_{\boldsymbol{A}}\} \cup \{\epsilon\}$
- $S_{\boldsymbol{A}} = \{s_{\boldsymbol{A}} | s_{\boldsymbol{A}} \text{ is an adversary state}\}$
- $s_{\boldsymbol{A}_0} \in S_{\boldsymbol{A}}$ is the adversary initial state
- $\delta_{\boldsymbol{A}} : S_{\boldsymbol{A}} \rightarrow \mathcal{P}(S_{\boldsymbol{A}})$ is the adversary state transition function
- $\omega_{\boldsymbol{A}} : S_{\boldsymbol{A}} \rightarrow \Sigma_{\boldsymbol{A}} \times \Sigma_{\boldsymbol{A}}$ is the adversary output function
- $\gamma_{\boldsymbol{A}} : S_{\boldsymbol{A}} \rightarrow \mathbb{R}^+$is the adversary cost function
- $F_{\boldsymbol{A}} \subseteq S_{\boldsymbol{A}} = \{s_{\boldsymbol{A}} | s_{\boldsymbol{A}} \in S_{\boldsymbol{A}} \text{ and } \boldsymbol{A} \text{ has finished attack}\}$

The function $\omega_{\boldsymbol{A}}$ outputs a pair of messages. The first message is transmitted to $\boldsymbol{I}$, and the second message to $\boldsymbol{R}$. If the output is (ϵ, ϵ), then no messages are transmitted. The adversary mounts an attack by transmitting one or more messages $\sigma_{\boldsymbol{A}}$. A successful attack triggers a transition from a protocol state $(s_{\boldsymbol{I}}, s_{\boldsymbol{R}}) \in (F_{\boldsymbol{I}} \times F_{\boldsymbol{R}})$ to a protocol state $(s_{\boldsymbol{I}}, s_{\boldsymbol{R}}) \notin (F_{\boldsymbol{I}} \times F_{\boldsymbol{R}})$. We limit the number of messages that the adversary can transmit. Once this limit is reached, the adversary transitions to a state $s_{\boldsymbol{A}} \in F_{\boldsymbol{A}}$.

3. COST MODEL

The model presented in Section 2 can be used without a cost model to find deadlock vulnerabilities in a protocol. However, we have to define a cost model to find other semantic DoS vulnerabilities that do not cause a deadlock, since a protocol might be vulnerable to highly efficient attacks that cause temporary disruption. First, we define four additional functions. The function $\tau_{\mathrm{m}} : \Sigma \rightarrow \mathbb{R}^+$, with $\tau_{\mathrm{m}}(\epsilon) = 0$, represents the transmission time of a protocol message σ. The function $\tau_{\mathrm{o}} : \Sigma \rightarrow \mathbb{R}^+$, with $\tau_{\mathrm{o}}(\epsilon) = 0$, represents the protocol overhead time of a message σ, e.g. waiting for a time slot before message transmission. Finally, we define the functions:

$$\tau(\sigma) = \tau_{\mathrm{m}}(\sigma) + \tau_{\mathrm{o}}(\sigma)$$

$$\tau_{\mathrm{m_{sum}}}(\sigma_1, \sigma_2) = \tau_{\mathrm{m}}(\sigma_1) + \tau_{\mathrm{m}}(\sigma_2)$$

We assume that the adversary seeks to maximize network disruption and minimize the probability of being located. The adversary has to be in physical proximity of a wireless network in order to mount an attack. Location determination methods such as triangulation or trilateration can be used by the network operator to determine the physical location of the adversary. If an adversary is located, then an ongoing attack can be stopped and the adversary can be apprehended. The precision of location determination depends on the number of measurements. The longer an adversary transmits a wireless signal, the higher the probability of being located. An adversary would thus limit his transmission time to a minimum to avoid being located.

The time spent transmitting a signal is also strongly correlated with energy consumption. If an adver-

sary can mount an attack by very infrequent transmissions, then he could use a battery powered device to cause long term disruption of the network. Such long lived, low power devices are referred to as "cyber mines" [18]. The ability to use "cyber mines" reduces the risk to the adversary, since he no longer has to be physically present when the attack is mounted.

The adversary constraints are thus location determination time and energy usage. We propose that the transmission time of the messages used for the attack, measured in seconds, is the most appropriate quantification of the adversary cost $\Gamma_{\boldsymbol{A}}$. Formally, we first define $\gamma_{\boldsymbol{A}}$ as follows:

$$\gamma_{\boldsymbol{A}}(\mathrm{s}_{\boldsymbol{A}}) = \tau_{\mathrm{m_{sum}}}(\omega_{\boldsymbol{A}}(\mathrm{s}_{\boldsymbol{A}}))$$

We define the adversary cost $\Gamma_{\boldsymbol{A}}$ as the cumulative output of $\gamma_{\boldsymbol{A}}$, with $\Gamma_{\boldsymbol{A}} = 0$ in the initial state $\mathrm{s}_{\boldsymbol{A}_0}$. A more sophisticated adversary model could include the computational cost of constructing a message or breaking certain cryptographic primitives as part of the cost function $\gamma_{\boldsymbol{A}}$. We do not include these costs in our model, since we model an adversary that only transmits unprotected messages. The energy costs of computation are thus insignificant compared to the energy costs of transmission.

A wireless network provides one or more services. A DoS attack causes a time period where service is not provided. In our model, this is represented by the time spent in protocol states $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \notin (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$. We propose to use this time period, measured in seconds, to quantify the protocol cost $\Gamma_{\boldsymbol{P}}$. Formally, we first define $\gamma_{\boldsymbol{I}}$ and $\gamma_{\boldsymbol{R}}$ as follows:

$$\gamma_{\boldsymbol{I}}(\mathrm{s}_{\boldsymbol{I}}, \sigma) = \begin{cases} 0 & \text{if } \mathrm{s}_{\boldsymbol{I}} \in \mathrm{F}_{\boldsymbol{I}} \text{ and} \\ & \delta(\mathrm{s}_{\boldsymbol{I}}, \sigma) \in \mathrm{F}_{\boldsymbol{I}} \\ \tau(\omega(\mathrm{s}_{\boldsymbol{I}}, \sigma)) & \text{if } \mathrm{s}_{\boldsymbol{I}} \notin \mathrm{F}_{\boldsymbol{I}} \\ \tau(\omega(\mathrm{s}_{\boldsymbol{I}}, \sigma)) & \text{if } \mathrm{s}_{\boldsymbol{I}} \in \mathrm{F}_{\boldsymbol{I}} \text{ and} \\ & \delta(\mathrm{s}_{\boldsymbol{I}}, \sigma) \notin \mathrm{F}_{\boldsymbol{I}} \end{cases}$$

$$\gamma_{\boldsymbol{R}}(\mathrm{s}_{\boldsymbol{R}}, \sigma) = \begin{cases} 0 & \text{if } \mathrm{s}_{\boldsymbol{R}} \in \mathrm{F}_{\boldsymbol{R}} \text{ and} \\ & \delta(\mathrm{s}_{\boldsymbol{R}}, \sigma) \in \mathrm{F}_{\boldsymbol{R}} \\ \tau(\omega(\mathrm{s}_{\boldsymbol{R}}, \sigma)) & \text{if } \mathrm{s}_{\boldsymbol{R}} \notin \mathrm{F}_{\boldsymbol{R}} \\ \tau(\omega(\mathrm{s}_{\boldsymbol{R}}, \sigma)) & \text{if } \mathrm{s}_{\boldsymbol{R}} \in \mathrm{F}_{\boldsymbol{R}} \text{ and} \\ & \delta(\mathrm{s}_{\boldsymbol{R}}, \sigma) \notin \mathrm{F}_{\boldsymbol{R}} \end{cases}$$

We then define $\Gamma_{\boldsymbol{I}}$ as the cumulative output of $\gamma_{\boldsymbol{I}}$, with $\Gamma_{\boldsymbol{I}} = 0$ in the initial state $\mathrm{s}_{\boldsymbol{I}_0}$. Next, we define $\Gamma_{\boldsymbol{R}}$ as the cumulative output of $\gamma_{\boldsymbol{R}}$, with $\Gamma_{\boldsymbol{R}} = 0$ in the initial state $\mathrm{s}_{\boldsymbol{R}_0}$. Finally, we define $\Gamma_{\boldsymbol{P}} = \Gamma_{\boldsymbol{I}} + \Gamma_{\boldsymbol{R}}$.

We compare the cumulative cost functions $\Gamma_{\boldsymbol{A}}$ and $\Gamma_{\boldsymbol{P}}$ to a physical layer jamming attack as an illustration. In a constant jamming attack, the adversary transmits noise in the radio frequencies used by the wireless network. As long as the adversary is transmitting, the protocol does not provide service. Once the adversary stops transmitting, the protocol immediately provides service again. In our model, $\Gamma_{\boldsymbol{A}}$ represents the time spent transmitting by the adversary and $\Gamma_{\boldsymbol{P}}$ represents the time period where the protocol does not provide service. Thus, an equivalent semantic attack would have $\Gamma_{\boldsymbol{A}} = \Gamma_{\boldsymbol{P}}$.

To achieve his goals, an adversary would try to find attacks where $\Gamma_{\boldsymbol{P}} \gg \Gamma_{\boldsymbol{A}}$. Such attacks could be considered as an amplifier for the adversary. An attack where $\Gamma_{\boldsymbol{P}} = 2\Gamma_{\boldsymbol{A}}$ is twice as efficient as a constant jamming attack, i.e. it has an amplification of 2. We define this amplification as the *attack efficiency* E:

$$E = \frac{\Gamma_{\boldsymbol{P}}}{\Gamma_{\boldsymbol{A}}}$$

The adversary's goal is thus to find an attack with the highest possible attack efficiency E. A semantic attack with $E > 1$ could be considered a protocol weakness. However, it is up to the protocol designer to decide the acceptable attack efficiency threshold.

Note that the cost model can be replaced by modifying the cost functions of the protocol and adversary models. Our proposed cost model is simple, which makes it easy to implement and model check. A more sophisticated cost model might give more realistic results. The tradeoff between realism and practicality in the cost model is one possible avenue of future work.

4. MODELING IEEE 802.11

We validate the usefulness of our proposed model by model checking IEEE 802.11 [9]. IEEE 802.11 is a family of standards for wireless local area networks (WLANs). The standard specifies the WLAN medium access control (MAC) and physical (PHY) layers. The 802.11 specification is unclear on several points, which require interpretation. We use the open source hostapd [14] and wpa_supplicant [15] implementations of 802.11 as a guideline for how to interpret ambiguities. We model a subset of the IEEE 802.11 MAC layer with the 802.11i [8] and 802.11h [7] amendments. The cost parameters are based on the 802.11g [6] PHY layer with a 54 Mbit/s transfer speed. We use the Promela language to implement the model and the SPIN model checker to find protocol vulnerabilities [5]. Our model consists of three entities: An access point (AP), a station (STA) and an adversary. The AP acts as the re-

sponder $\boldsymbol{R}$ and the STA acts as the initiator $\boldsymbol{I}$. The entities are modeled as Promela processes. The cumulative cost functions $\Gamma_{\boldsymbol{A}}$ and $\Gamma_{\boldsymbol{P}}$ are global variables that are incremented during state transitions. The network is modeled as two asynchronous message channels in Promela, one in each direction. The STA initiates a connection setup whenever it is in state $\mathrm{s}_{\boldsymbol{I}_0}$.

Once the AP and STA have performed the 802.11 authentication, association and key agreement procedures, they alternately send and receive data frames. We model $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$ as the state where the STA receives a valid data frame from the AP and then transmits a data frame to the AP. One major challenge of using the proposed cost based model is how to avoid an infinite state space. If the protocol participants enter an infinite loop through protocol states $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \notin (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$, then $\Gamma_{\boldsymbol{P}}$, and thus the state space, will keep increasing. We solve this by letting only one of the participants initiate the transfer of data frames once the protocol transitions from a state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \notin (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$ to a state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$. While in $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$ the participants only transmit a data frame once they receive a data frame. If an attack causes a desynchronization of the participants, then the 802.11 resynchronization mechanisms will enable the AP and STA to resume communication. If, however, the resynchronization mechanisms fail, then the model will deadlock. This property makes checking for protocol deadlock vulnerabilities simple, since SPIN has a built-in test for deadlocks.

Having implemented the model in Promela, we then need to specify the properties to be checked. We use linear temporal logic (LTL) for this purpose. The LTL formula used for checking the cost based model is:

$$\Box((\Gamma_{\boldsymbol{A}} = 0) \vee (\frac{\Gamma_{\boldsymbol{P}}}{\Gamma_{\boldsymbol{A}}} < T))$$

The parameter T specifies the attack efficiency threshold. The most efficient published DoS attack against 802.11 is the quiet attack from [11].

5. EXPERIMENTAL RESULTS

We first set the threshold to the efficiency of the quiet attack to verify that the attack is found by the model checker. The result is that SPIN finds the quiet attack. We then proceed to set the threshold to slightly more than the efficiency of the quiet attack to see if a more efficient attack exists. The result is negative, the quiet attack is the most efficient attack against 802.11 in our model. We then gradually lower the threshold in order to find other less efficient attacks. The results are that the model checker is able to find the previously known semantic DoS attacks against 802.11, but we do not find any new attacks.

We then proceed to check for deadlocks. The result is that SPIN finds a new deadlock vulnerability in 802.11i. The adversary transmits a valid 802.11 Open System authentication request from the STA to the AP while the protocol participants are in a state $(\mathrm{s}_{\boldsymbol{I}}, \mathrm{s}_{\boldsymbol{R}}) \in (\mathrm{F}_{\boldsymbol{I}} \times \mathrm{F}_{\boldsymbol{R}})$. The AP deletes its 802.11i security association with the STA, but stays in 802.11 State 3. All data frames from the STA to the AP are dropped, since they are encrypted and integrity protected with a key that the AP no longer has access to. The AP cannot transmit any data frames to the STA, since it is unable to sign and encrypt them. The 802.11 resynchronization mechanisms are not triggered since both the AP and the STA are in 802.11 State 3.

We proceed to experimentally validate the vulnerability. We set up an 802.11 network using a wireless router with the hostapd software as the AP and a laptop computer with the wpa_supplicant software as the STA. The adversary causes a protocol deadlock by transmitting a single authentication request frame. In practice, the STA is able to recover after approximately 7 minutes due to an internal timeout. This timeout is not specified in the 802.11 standard, however, so there is no guarantee that other implementations would be able to recover. Even with this ability to recover, the attack is far more efficient than any published DoS attacks against 802.11. The author of hostapd and wpa_supplicant has been notified about the vulnerability.

6. CONCLUSIONS

The history of published attacks against existing wireless protocols shows that the design of robust protocols could greatly benefit from formal analysis tools. We have proposed a formal method for modeling semantic DoS attacks against wireless networks and shown how the model can be used to discover protocol vulnerabilities. By this, we have found a new deadlock vulnerability in 802.11 and experimentally validated it. Our proposed model can facilitate the design of robust protocols by discovering vulnerabilities during the design process.

7. REFERENCES

[1] Bellardo, J., and Savage, S. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the*

12th USENIX Security Symposium (Berkeley, CA, USA, 2003), USENIX Association.

[2] Eian, M. Fragility of the robust security network: 802.11 denial of service. In *Proceedings of the 7th International Conference on Applied Cryptography and Network Security* (2009), vol. 5536 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 400–416.

[3] Eian, M. A practical cryptographic denial of service attack against 802.11i TKIP and CCMP. In *Proceedings of the Ninth International Conference on Cryptology And Network Security* (2010), vol. 6467 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 62–75.

[4] Han, T., Zhang, N., Liu, K., Tang, B., and Liu, Y. Analysis of mobile WiMAX security: Vulnerabilities and solutions. In *Mobile Ad Hoc and Sensor Systems, 2008. MASS 2008. 5th IEEE International Conference on* (2008), pp. 828 –833.

[5] Holzmann, G. *Spin model checker, the: primer and reference manual*, first ed. Addison-Wesley Professional, 2003.

[6] IEEE. *IEEE Std 802.11g-2003*. New York, NY, USA, 2003.

[7] IEEE. *IEEE Std 802.11h-2003, IEEE 802.11-1999 Amendment 5: Spectrum and Transmit Power Management Extensions in the 5 GHz band in Europe*. New York, NY, USA, 2003.

[8] IEEE. *IEEE Std 802.11i-2004, IEEE 802.11-1999 Amendment 6: Medium Access Control (MAC) Security Enhancements*. New York, NY, USA, 2004.

[9] IEEE. *IEEE Std 802.11-2007, IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. New York, NY, USA, 2007.

[10] Kambourakis, G., Kolias, C., Gritzalis, S., and Hyuk-Park, J. Signaling-oriented DoS attacks in UMTS networks. In *Advances in Information Security and Assurance*, vol. 5576 of *Lecture Notes in Computer Science*. Springer-Verlag, 2009, pp. 280–289.

[11] Könings, B., Schaub, F., Kargl, F., and Dietzel, S. Channel switch and quiet attack: New DoS attacks exploiting the 802.11 standard. In *LCN 2009: Proceedings of the IEEE 34th Conference on Local Computer Networks* (2009), pp. 14–21.

[12] Lafrance, S., and Mullins, J. Using admissible interference to detect denial of service vulnerabilities. In *Sixth International Workshop in Formal Methods. Electronic Workshops in Computing (eWiC) by British Computer Society (BCS)* (2003), pp. 1–19.

[13] Mahimkar, A., and Shmatikov, V. Game-based analysis of denial-of-service prevention protocols. In *Proceedings of the 18th IEEE workshop on Computer Security Foundations* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 287–301.

[14] Malinen, J. hostapd: IEEE 802.11 AP, IEEE 802.1X / WPA / WPA2 / EAP / RADIUS Authenticator, 2011. http://hostap.epitest.fi/hostapd.

[15] Malinen, J. Linux WPA/WPA2/IEEE 802.1X Supplicant, 2011. http://hostap.epitest.fi/wpa_supplicant.

[16] Meadows, C. A formal framework and evaluation method for network denial of service. *IEEE Computer Security Foundations Workshop 00* (1999), 4.

[17] Narayana, P., Chen, R., Zhao, Y., Chen, Y., Fu, Z., and Zhou, H. Automatic vulnerability checking of IEEE 802.16 WiMAX protocols through TLA+. In *Secure Network Protocols, 2006. 2nd IEEE Workshop on* (2006), pp. 44 –49.

[18] Pelechrinis, K., Iliofotou, M., and Krishnamurthy, V. Denial of service attacks in wireless networks: The case of jammers. *Communications Surveys Tutorials, IEEE PP*, 99 (2010), 1 –13.

[19] Ramachandran, V. Analyzing DoS-resistance of protocols using a cost-based framework. Tech. rep., Yale University, 2002.

[20] Sankaran, C. Network access security in next-generation 3GPP systems: A tutorial. *Communications Magazine, IEEE 47*, 2 (2009), 84 –91.

[21] Smith, J. *Denial of Service: Prevention, Modelling and Detection*. Brisbane, Australia, 2007. PhD Thesis, Queensland University of Technology.

[22] Tritilanunt, S. *Protocol engineering for protection against denial-of-service attacks*. Brisbane, Australia, 2009. PhD Thesis, Queensland University of Technology.