WiFire: A Guardian Angel for Wireless Networks

Submitted to the ACM Student Research Competition Grand Finals 2012

Matthias Wilhelm
Distributed Computer Systems Lab
TU Kaiserslautern, Germany
(Advisor: Jens B. Schmitt)
wilhelm@cs.uni-kl.de

Abstract

Security in wireless networks is a notoriously difficult problem, mainly because medium access is hard to control: anyone in transmission range can easily inject packets into a network. The current solution strategy is to place the burden of packet filtering on each network node individually, leading to challenging administration and performance problems. We propose an alternative approach that provides the desired packet filtering remotely: the wireless firewall. The working principle is simple—if we cannot prevent the transmission of a malicious packet, we may still prevent its reception. The wireless firewall achieves this by content-based classification and selective interference that is just long enough to induce checksum errors in malicious packets. This way, the protection is fully transparent to the network: everything received without errors is trustworthy. We show the feasibility of our approach with WiFire, a software-defined wireless firewall system implemented on the USRP2, which achieves per-packet classification and selective destruction reliably: our evaluation shows that 99.9 % of adversarial traffic is successfully blocked without disturbing the legitimate operations of the network.

1 Problem and Motivation

Wireless networking technology is enabling a revolution of smart and connected devices around us that are becoming increasingly ubiquitous. However, this pervasiveness also increases the reliance on wireless-enabled devices in security-critical or privacy-sensitive applications such as healthcare [5, 7, 9], logistics [2, 25], home [14, 16, 33] or industrial automatization [25, 35], and public infrastructure (e.g., the power grid [18]). While such scenarios are a perfect motivation for using wireless communication technologies, they also provide an attractive playground for attackers to exploit.

Security in wireless networks is generally harder to achieve compared to wired networks. Due to the broadcast nature of radio frequency (RF) propagation, network access cannot be regulated physically; anyone in transmission range can eavesdrop or inject arbitrary messages.[1] The current response to this threat is to use strong cryptographic protection to ensure that messages remain confidential and that only authorized parties can participate in a network. This approach, however, is not always easily applicable because wireless devices have several unique characteristics: *(i)* the devices often have limited computational resources and are optimized for a particular application, *(ii)* they run on batteries and thus have the primary goal to maximize lifetime, *(iii)* they may have limited programmability or cannot be modified at all, *(iv)* they may be mobile and travel across different security domains, and *(v)* they are often personal belongings that are operated and configured by (possibly security-oblivious) end users. It is hard to imagine that devices such as sensor motes, RFID chips or implanted medical devices implement a full range of security measures despite these challenges. Their protection task is highly asymmetric because all security protocols must be implemented on each resource-limited device while the adversary can use high-performance systems.

Ideally, what you want is an external guardian system that supports devices in their task of protecting themselves. A wish list of its features may be:

- remote protection for several devices at the same time,
- generic and programmable security policies, and
- transparent operation; no changes to the existing devices should be necessary.

Remote protection helps to off-load security costs to an external security infrastructure, programmable security policies enable an easy adaptation to new technologies or threats, and transparent operation ensures that any wireless device can be protected this way. Recently published results show that such remote (or over-the-air) protection is feasible by using selective RF interference, intercepting malicious packets before they arrive at a receiver [8, 17, 34]. However, in contrast to related work, we aim for fully programmable security policies, similar to the ones found in network firewalls. There, a set of firewall rules is used to classify packets based on their content, and violating packets are dropped. While there are efforts to bring packet filtering to resource-constrained devices [11], this does not reduce the burden carried by them. In this research project, we aim to combine the two methods of over-the-air packet filtering and rule-based security policies to a unified protection system: the "wireless firewall." We present WiFire, a system that demonstrates the feasibility and applicability of our approach in IEEE 802.15.4 networks [13]. For each incoming packet, WiFire accesses its content in real-time and compares it to a set of filtering rules. If the packet matches one of these rules,

[1] For example, researchers were able to eavesdrop on Bluetooth phone calls from more than a mile away: `http://www.wired.com/politics/security/news/2004/08/64463`.

System	Application area	Maximum reaction time	Guard distance	Blocking criteria	Prototype evaluation[a]
IMD shield [8]	Implanted medical devices (IMDs)	10 ms	20 cm	Each packet is blocked and selectively forwarded	$\surd$ (USRP2)
IMDGuard [34]	IMDs	Tens of ms	20 cm	Guard notices a spoofing attack	$\sim^b$ (MICAz)
Blocker Tags [15]	RFID	300 µs	20 cm	Tag query to protected prefix	$\times$ (tag)
RFID Guardian [26, 27]	RFID	300 µs	1 m	Tag query to tag in ACL	$\times$ (handheld)
Jamming for Good [17]	Sensor networks	5 ms	2–3 m	Address+RSSI of registration packet	$\surd$ (MICAz)
[This work]	Sensor networks	64 µs	10–20 m	Per-packet decision (header+payload)	$\surd$ (USRP2)

[a] In parenthesis is the type of devices used for the (possibly envisioned) prototype implementation.
[b] The concept is evaluated on the IEEE 802.15.4 physical layer.

Table 1: Comparison of related protection systems using physical layer responses.

WiFire generates a short burst of interference to destroy the packet before it is received. Our implementation provides a easy-to-use interface in the spirit of the well-known Linux firewall `iptables` to define network access rules, and our system evaluation reveals that WiFire is able to block 99.9 % of malicious packets in a range of 18 m. With WiFire, we provide an external "guardian angel" that helps to protect wireless devices that otherwise may be unable to protect themselves.

2 Background and Related Work

The concept of using selective interference to reach security goals is applied in several application scenarios in the literature: guarding implanted medical devices (IMDs) from malicious readers, protecting the privacy of a person carrying RFID tags, and enabling authentic communications in wireless sensor networks (WSNs).

Gollakota et al. [8] use an external guardian system, called "IMD shield," to ensure the safety and privacy of patients with IMDs, because these devices may otherwise send out confidential information or even be reconfigured wirelessly against the patients wishes [7, 9]. They propose a system that concurrently receives and destroys each message linked to the IMD. Then, it selectively forwards commands to or replies from the IMD if they are authorized by the shield's security policy. IMDGuard, a similar system proposed by Xu et al. [34], supports an IMD during cryptographic operations, and uses selective interference when an attacker tries to disturb this operation. Both systems work transparently, i.e., there is no need to modify the IMD after it is implanted. However, in contrast to our work, the external guardian must be placed in close proximity to the protected device (approx. 20 cm), which constitutes a reader–device setting instead of a networked setting with several devices that we are aiming for. Additionally, they do not offer programmable security policies: the IMD shield is limited to classifying packets by the destination IMD serial number, and IMDGuard uses packet timing to detect malicious packets.

In the context of RFID privacy, Juels et al. introduce the "blocker tag" [15] to prevent malicious readers from discovering RFID tags carried by a person, which could allow for tracking or a disclosure of carried objects. The working principle is the following: each time a malicious reader queries the address space of the protected tags, the blocker tag generates an artificial collision on the medium. In this case, the reader assumes that several tags are present and that more specific queries are required; but as the collisions are triggered for all queries, the reader is forced to search the complete address space (e.g., 2^{64} addresses). Rieback et al. [26] extend this idea with the RFID Guardian, a battery-powered handheld device that enables the use of configurable access control lists (ACLs) for arbitrary sets of RFID tags. These ACLs specify which readers are allowed to interact with which tags. Again, this is a reader–device setting with small distances (up to 1 m for RFID Guardian). The main difference however is that these systems interfere the tag discovery protocol only. The query messages are actually received by the tags, only their answers are destroyed. With WiFire, we want to protect devices starting with the very first packet.

Closely related to WiFire is part of our previous work [17], which uses selective interference to enable authentic communication in WSNs. Sensor devices support each other to prevent impersonation attacks, i.e., an attacker pretending to be part of the network by spoofing its source address to look like an existing network device. When a sensor device wants to send data, it first sends a "reservation packet" to notify others of its transmission wish before it starts sending the data packet. Sensor devices in the vicinity compare the claimed source address and the physical signal fingerprint of the reservation (which depends on the device's location and is hard to spoof) with previously observed measurements. In case of a mismatch (i.e., a potential attack), the device schedules a concurrent transmission with the data packet, intercepting the spoofed packet. This approach explores the concept of over-the-air support to increase security, but also forces an expensive mode of operation on the network. WiFire explores the use of a dedicated security infrastructure to protect wireless devices in a fully transparent way: no modifications are required, the devices do not even need to know that WiFire is protecting them.

Wireless intrusion detection and prevention systems (WIPS [28, §5]) are also closely related to WiFire. However, commercial products for WLAN protection such as AirMagnet [6], AirDefense [20] or SpectraGuard [1] do not prevent the reception of packets; rather, they exploit the fact that communication is only possible after reaching an associated state with an access point, and repeatedly break this association to the adversary. This approach is not applicable to protect low-power wireless networks because their protocols do not use such association mechanisms. Thus, WiFire must operate on the physical layer to achieve its goal.

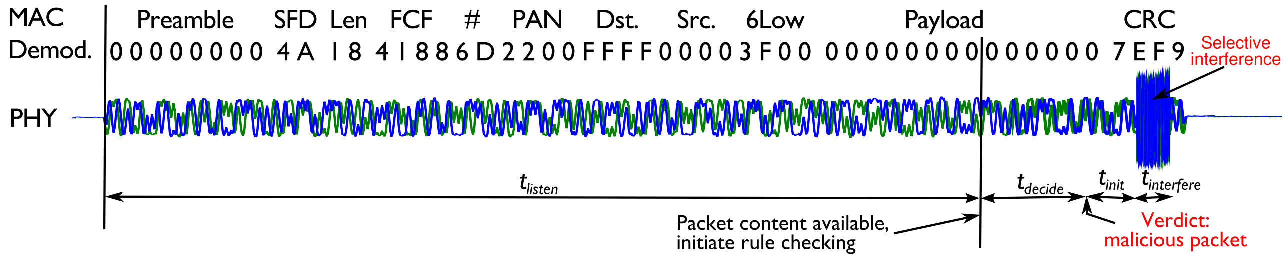


Figure 1: Operation of WiFire: first, the packet's signal is demodulated to access its content for classification. WiFire must wait for the payload bytes to arrive (t_{listen}) before the rule checker can start. When the packet is declared malicious (after t_{decide}), the transmission of interference is prepared (t_{init}), and the offending packet destroyed ($t_{\text{interfere}}$) before it may be received.

3 Approach and Uniqueness

In our approach, we aim to protect wireless devices from attacks over-the-air and on a per-packet basis, lifting the security burden from them. We are the first to combine selective interference and rule-based security policies to a generic protection mechanism for wireless networks.

3.1 WiFire's Concept

To illustrate the operation of WiFire and to identify the technical challenges that arise, we discuss the interception of a single malicious packet. Fig. 1 shows our packet, a 26 byte IEEE 802.15.4 data packet going to the broadcast address `0xFFFF` of sub-network `0x22`. The transmission duration of this packet is 832 µs, and it starts with a physical layer header, link layer header, and payload; it ends with a 16 bit checksum (CRC). WiFire operates as follows:

- It first detects the packet using the preamble and start-of-frame delimiter (SFD), which signals the beginning of the packet. Then, it proceeds to demodulate the content of the packet, gaining access to header fields and payload, which is subsequently used to decide whether the packet is malicious. The longer the reception period, denoted by t_{listen}, the more content is available to classify the packet; on the other hand, if WiFire listens for too long, it may be unable to still destroy the packet. If the decision is based on the header fields, t_{listen} is 480 µs, permitting a maximum system response time of 352 µs; if the full payload is considered, t_{listen} is 768 µs and only 64 µs remain. This illustrates that WiFire must fulfill very strict timing requirements to both classify and destroy the packet.
- As soon as the necessary content is available, the rule checker is started to compare the detected packet to the stored security policy. The execution time of the rule checker is denoted by t_{decide}.
- If the rule checker concludes that the packet is violating the security policy, it initiates the transmission of a burst of interference. The time required to set up this operation is denotes as t_{init}.
- Finally, the transmission of interference must reliably destroy the packet. In the IEEE 802.15.4 standard, a single bit error is sufficient to destroy the packet, even if the bit error is in the last CRC byte. The duration of interference ($t_{\text{interfere}}$) must be long enough to force at least one bit error, but not necessarily longer.
- The overall system response time is denoted by t_{response}, which is defined as the time from the start of classification to the end of the interference, i.e., $t_{\text{response}} = t_{\text{decide}} + t_{\text{init}} + t_{\text{interfere}}$. We aim for a response time below 64 µs.

In summary, the system must fulfill tight timing requirements; the overall time to listen to a packet and respond must be much smaller than the packet's duration.

3.2 WiFire's Implementation

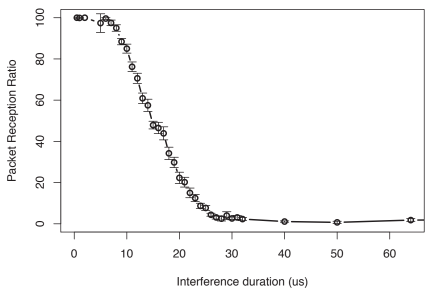
To fulfill these requirements, we need an implementation that is very close to the physical layer and offers high performance. We implemented WiFire on top of the widely used software-defined radio (SDR [19]) platform USRP2, which offers direct access to the physical layer by design. An SDR enables the implementation of receiver algorithms in software, providing a digitized representation of the RF waves (comparable to `.wav` files for raw audio) as input to the receiver. Similarly, arbitrary waveforms defined by digital sample sequences can be emitted, allowing full control over transmission parameters such as duration, physical shape, and output power. With the SDR architecture, we are able to build a receiver that supports real-time access to packet content, which is not feasible with off-the-shelf receivers [10, 22]. However, the conventional mode of operation of the USRP2 is not suited to support our desired operation. Normally, the USRP2 digitizes the RF channel and forwards the received samples to a host PC via Ethernet; on the host's CPU, the necessary digital signal processing is performed. If a transmission is initiated, the outgoing digital samples are sent to the USRP2 via Ethernet as well. This round-trip time accumulates to approximately 2 ms [21], which clearly makes a real-time detection and destruction infeasible, as discussed in the previous section.

To achieve its goals nevertheless, WiFire is fully implemented in FPGA logic on the USRP2.[2] WiFire consists of three main components:

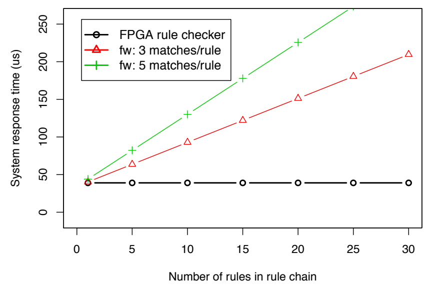
Packet detection. This subsystem continuously scans the RF medium to detect any packet that might be received by the network and demodulates and delivers the packet content to the subsequent decision subsystem. It consists of a receiver that detects the content bytes and a framer that interprets these bytes according to the IEEE 802.15.4 standard, providing access to header fields and payload bytes.

Rule decision. The decision system is triggered via interrupts by the packet framer when a pre-defined point in the packet is reached (e.g., when the full link layer header

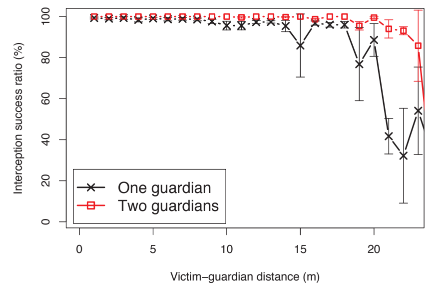
[2] Thankfully, the manufacturer of the USRP2, Ettus Research, provides all resources as open-source software.



(a) Minimum interference duration for IEEE 802.15.4 radios: interfering with 26 µs of a packet transmission is sufficient to trigger packet drops at the receiver reliably.



(b) System response time for two rule checker implementations: firmware with different rule configurations (flexible but slow) and FPGA (static rules but fast).



(c) The system's overall protection performance with close-proximity attacker (the attack always succeeds when WiFire is absent): detecting, classifying and destroying the attacker's packets.

Figure 2: Performance evaluation: minimum interference duration, response time, and overall protection performance.

is available) to trigger the decision process on whether the packet should be blocked; it classifies incoming packets according to a pre-defined policy. It is the critical component for real-time operation because the overall response time mainly depends on its execution time. Therefore, we implemented two different versions: *(i)* firmware code written in C and running on the USRP2's (soft-) micro-controller, which offers runtime reconfigurability but is comparatively slow, and *(ii)* an implementation in FPGA logic that reduces the response time, but the security policy must be specified at compile time. In both implementations, the rule checker notifies the interference subsystem via interrupts that a short burst of interference must now be generated to destroy a malicious packet. The firmware-based rule checker allows to define content-based rules in the style of `iptables`, defining rule chains that consist of one or more rules, each with zero or more matches (such as source or destination address). We implemented a command line tool (`wftables`), which generates a data structure that can be directly interpreted by the firmware rule checker. An example is the following rule definition with two matches (preventing the reception of all control packets going to the broadcast address in PAN `0x22`):

```
wftables -A -m dst --pan 0x22 --addr 0xFFFF
            -m type --ctrl -j DROP
```

This mechanism allows to define complex access policies and to deploy them on the distributed WiFire guardians. The FPGA rule checker uses hardware gates to compare detected packet bytes to a table of predefined values in parallel, such that the execution time is considerably reduced.

Selective interference. As an SDR enables arbitrary waveforms to be transmitted on the wireless channel, we are free to select an interference waveform with desirable properties such as spectral efficiency, limited damage to co-existing communication standards or matched waveforms that pass through a receiver's RF filters. We evaluated several interference waveforms and found that tone jamming [24] (a narrow-band continuous wave) is the most efficient one for IEEE 802.15.4 networks and is also limited to a narrow portion of the spectrum, having a negligible effect on other channels.

4 Results and Contributions

In this section, we show that WiFire reaches its goal of achieving over-the-air packet filtering and offering a reliable protection of wireless devices from a distance.

4.1 System Evaluation

We evaluate the system performance of WiFire; we are interested in the minimum interference duration to reliably destroy packets, the system response time and the resulting "depth" to look into packets, and the overall system performance of detection, classification, and destruction of malicious packets in an realistic indoor scenario.

Interference duration. We first evaluate how long WiFire must hit a packet to successfully destroy it. Our evaluation [29] shows that the necessary interference is a small fraction of the packet, only 26 µs, while the average packet duration is 1024 µs (see also Fig. 2a). This fact helps in two ways: first, we are able to observe large parts of a packet because t_{response} is small enough to observe the complete payload and still reliably destroy the packet. Second, it helps to reduce possible unintentional interference with co-existing networks, which is a critical point for real-world deployments of WiFire. From the view of a single channel, WiFire's behavior is comparable to frequency hopping systems such as Bluetooth. In fact, Bluetooth Power Class 1 devices [12, §7.2] use the same transmit power (100 mW) as WiFire and occupy a 2 MHz 802.15.4 channel for approximately 25 ms per second, which is comparable to the emissions of WiFire reacting to an attacker with maximum rate (1000 packets/s). This shows that WiFire can effectively control the wireless channel while using very limited emissions, comparable to licensed devices.

System response time. We proceed by evaluating the system response time t_{response} achieved by WiFire. As a reference value, if we want to read the complete payload and perform the classification and selective interference during the checksum at the end of the packet, this time must be less than 64 µs. The results of our evaluation are shown in Fig. 2b. For the firmware-based rule checker, the delay depends on the number and complexity of rules to be checked. The response time for a representative rule set is 160 µs, or

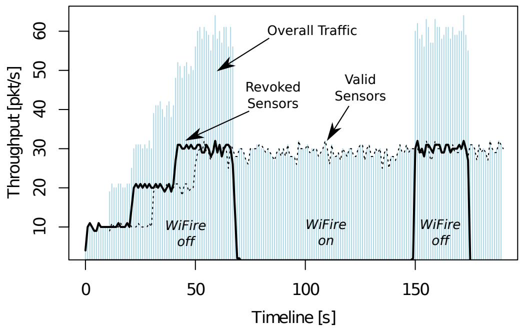


Figure 3: Central node revocation of three MICAz motes enforced by WiFire in a network consisting of six motes. After 70 seconds, WiFire is configured to selectively block traffic transmitted from revoked devices (3 motes).

5 byte before the end of the packet. For the FPGA implementation, the execution time is much shorter: even with a complex rule set, the response time does not exceed 39 µs. This enables rules using the complete packet payload and still ensures the destruction of the packet.

Overall system performance. This result signifies that we have achieved our timing goals, but how reliable can we classify and destroy packets over a distance? This is especially relevant because signals fade rapidly with increasing distance; both receiver and selective interference must still perform reliably. We use WiFire and two MICAz devices to evaluate the system performance in a demanding setting: the distance between attacker and victim is only 1 m, and the distance between the victim and WiFire is varied from 1 m to 25 m.

WiFire successfully prevents packet receptions in 98–99 % for distances up to 15 m, despite the close proximity of attacker and victim (see Fig. 2c). Two WiFire guardians located at the same position effectively counter the issue of 1–2 % packet misses, and achieve a combined 99.9 % protection rate up 18 m. They miss 19 out of 18000 packets up to this distance.

This system performance shows that we can indeed protect devices over a distance. We present an example application of WiFire in the next section.

4.2 Application Scenario: Node Revocation

In this section, we show results of an application scenario that we implemented to illustrate the feasibility to protect real networks. The goal is to remove compromised devices from the network, also referred to as node revocation. This usually arises if some nodes are already exploited, malfunctioning or misconfigured. In such a case, no traffic from revoked nodes is allowed; this is a notoriously difficult problem in WSNs because parts of the network may be under adversarial control, hindering the execution of node revocation protocols [4, 23]. Using WiFire, the solution is simply a security policy that is based on detecting revoked nodes by their MAC addresses and the network ID:

```
wftables -A -m src --addr 0x1111 --pan 0xACAC -j DROP
wftables -A -m src --addr 0x1112 --pan 0xACAC -j DROP
wftables -A -m src --addr 0x1115 --pan 0xACAC -j DROP
```

In this experiment, MICAz sensor devices are deployed indoor and consecutively start transmitting with a rate of 10 packets/s. After 70 seconds, three nodes are revoked for 90 seconds, then allowed again for 20 seconds, and finally revoked for the rest of the experiment. We are interested in packets from revoked nodes that are able to reach the network (false negatives) and the impact of WiFire on the legitimate traffic during policy enforcement (false positives). The results are shown in Fig. 3. The stepwise build-up of traffic is due to the consecutive start of the transmissions. As can be seen, WiFire immediately reacts by completely blocking the traffic from the revoked nodes. During the revocation phases, the amount of legitimate traffic equals the overall traffic, so that there are no false positives. The number of false negatives is one packet at the beginning and at the end of revocation phases (due to the transition of WiFire's policy enforcement).

4.3 Contributions

In this research project, we show that a remote protection system to increase the security of wireless devices from a distance is indeed feasible. The concept only required the emission of very limited interference (with a duration of 26 µs), operates very selectively with fully programmable per-packet classification and subsequent destruction of packets, and reliable operation over distances up to 18–20 m with 99.9 % of intercepted packets. This work was partially published in [29] and [30]. We made WiFire's software open-source to enable other researchers to experiment with selective interference [31]. This is because security is not the only potential use of our system: we also showed the feasibility of physical layer message manipulation attacks [32] and plan to augment wireless testbeds with selective interference to enable repeatable and controllable real-world experiments, in the spirit of Boano et al. [3]. In conclusion, the use of selective interference shows promise to enable the protection of wireless devices from a distance, and it may enable future wireless networks to operate more safely and reliably.

5 References

[1] AirTight Networks. Complete wireless security for your network: SpectraGuard Enterprise. Retrieved from www.airtightnetworks.com, Apr. 2011.

[2] D. J. A. Bijwaard, W. A. P. van Kleunen, P. J. M. Havinga, L. Kleiboer, and M. J. J. Bijl. Industry: using dynamic WSNs in smart logistics for fruits and pharmacy. In *Proc. of ACM SenSys 2011*, pages 218–231, Nov. 2011.

[3] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. Zuniga. JamLab: Augmenting sensornet testbeds with realistic and controlled interference generation. In *Proc. of ACM/IEEE IPSN 2011*, pages 175–186, Apr. 2011.

[4] H. Chan, V. Gligor, A. Perrig, and G. Muralidharan. On the distribution and revocation of cryptographic keys in sensor networks. *IEEE Transactions on Dependable and Secure Computing*, 2(3):233–247, Sept. 2005.

[5] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In *Proc. of ACM SenSys 2010*, pages 155–168, Nov. 2010.

[6] Fluke Corporation. AirMagnet solutions: Wi-Fi done right. Retrieved from http://www.airmagnet.com, March 2011.

[7] K. Fu. Inside risks: Reducing risks of implantable medical devices. *Communications of the ACM*, 52(6):25–27, June 2009.

[8] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu. They can hear your heartbeats: non-invasive security for implantable medical devices. In *Proc. of ACM SIGCOMM 2011*, pages 2–13, 2011.

[9] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Proc. of IEEE S&P 2008*, pages 129–142, May 2008.

[10] Z. He and T. Voigt. Precise packet loss pattern generation by intentional interference. In *Proc. of IEEE DCOSS 2011*, pages 1–6, June 2011.

[11] M. Hossain and V. Raghunathan. AEGIS: A lightweight firewall for wireless sensor networks. In *Distributed Computing in Sensor Systems*, volume 6131 of *LNCS*, pages 258–272. Springer, 2010.

[12] IEEE Computer Society. IEEE Standard 802 Part 15.1. `http://www.ieee802.org/15/`, June 2005.

[13] IEEE Computer Society. IEEE Standard 802 Part 15.4. `http://www.ieee802.org/15/`, Sept. 2006.

[14] X. Jiang, M. Van Ly, J. Taneja, P. Dutta, and D. Culler. Experiences with a high-fidelity wireless building energy auditing network. In *Proc. of ACM SenSys 2009*, pages 113–126, Nov. 2009.

[15] A. Juels, R. L. Rivest, and M. Szydlo. The Blocker Tag: selective blocking of RFID tags for consumer privacy. In *Proc. of ACM CCS 2003*, pages 103–111, Oct. 2003.

[16] J. Lu, T. Sookoor, V. Srinivasan, G. Gao, B. Holben, J. Stankovic, E. Field, and K. Whitehouse. The smart thermostat: using occupancy sensors to save energy in homes. In *Proc. of ACM SenSys 2010*, pages 211–224, Nov. 2010.

[17] I. Martinovic, P. Pichota, and J. B. Schmitt. Jamming for good: a fresh approach to authentic communication in WSNs. In *Proc. of ACM WiSec 2009*, pages 161–168, Mar. 2009.

[18] S. Massoud Amin and B. F. Wollenberg. Toward a smart grid: power delivery for the 21st century. *IEEE Power and Energy Magazine*, 3(5):34–41, 2005.

[19] J. Mitola. The software radio architecture. *IEEE Communications Magazine*, 33(5):26–38, May 1995.

[20] Motorola Solutions. Motorola AirDefense—security & compliance solutions. Retrieved from `http://www.airdefense.net`, March 2011.

[21] G. Nychis, T. Hottelier, Z. Yang, S. Seshan, and P. Steenkiste. Enabling MAC protocol implementations on software-defined radios. In *Proc. of USENIX NSDI 2009*, pages 91–105, Apr. 2009.

[22] C. P. O'Flynn. Message denial and alteration on IEEE 802.15.4 low-power radio networks. In *Proc. of IFIP NTMS 2011*, pages 1–5, Feb. 2011.

[23] B. Parno, A. Perrig, and V. Gligor. Distributed detection of node replication attacks in sensor networks. In *Proc. of IEEE S&P 2005*, pages 49–63, May 2005.

[24] R. A. Poisel. *Modern Communications Jamming: Principles and Techniques*. Artech House Publishers, Boston, MA, Nov. 2003.

[25] W.-B. Pöttner, L. Wolf, J. Cecilio, P. Furtado, R. Silva, J. Sá Silva, A. Santos, P. Gil, A. Cardoso, Z. Zinonos, J. M. do Ó, B. McCarthy, J. Brown, U. Roedig, T. O'Donovan, C. J. Sreenan, Z. He, T. Voigt, and A. Jugel. WSN evaluation in industrial environments first results and lessons learned. *Proc. of IEEE DCOSS 2011*, pages 1–8, 2011.

[26] M. Rieback, B. Crispo, and A. Tanenbaum. RFID Guardian: a battery-powered mobile device for RFID privacy management. In *Information Security and Privacy*, volume 3574 of *LNCS*, pages 259–273. Springer, 2005.

[27] M. Rieback, B. Crispo, and A. Tanenbaum. Keep on blockin' in the free world: Personal access control for low-cost RFID tags. In *Security Protocols*, volume 4631 of *LNCS*, pages 51–59. Springer, 2007.

[28] K. Scarfone and P. Mell. Guide to intrusion detection and prevention systems (IDPS). *NIST Special Publication*, 800(94):1–121, 2007.

[29] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders. Short paper: reactive jamming in wireless networks: how realistic is the threat? In *Proc. of ACM WiSec 2011*, pages 47–52, June 2011.

[30] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders. WiFire: A firewall for wireless networks. In *Proc. of ACM SIGCOMM 2011*, pages 456–457, Aug. 2011.

[31] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders. WiSec 2011 demo: RFReact—a real-time capable and channel-aware jamming platform. *SIGMOBILE Mobile Computing and Communications Review*, 15(1):41–42, Nov. 2011.

[32] M. Wilhelm, J. B. Schmitt, and V. Lenders. Practical message manipulation attacks in IEEE 802.15.4 wireless networks. In *MMB & DFT 2012 Workshop Proceedings*, pages 29–31, Mar. 2012.

[33] A. Wood, J. Stankovic, G. Virone, L. Selavo, Z. He, Q. Cao, T. Doan, Y. Wu, L. Fang, and R. Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *IEEE Network*, 22(4):26–33, 2008.

[34] F. Xu, Z. Qin, C. C. Tan, B. Wang, and Q. Li. IMDGuard: Securing implantable medical devices with the external wearable guardian. In *Proc. of IEEE INFOCOM 2011*, pages 1862–1870, Apr. 2011.

[35] C. Zhang, A. Syed, Y. Cho, and J. Heidemann. Steam-powered sensing. In *Proc. of ACM SenSys 2011*, pages 204–217, Nov. 2011.