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,
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.
IEEE 802.15.4 data packet going to the broadcast address 0xFFFF technical challenges that arise, we discuss the interception of 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 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 832µs, 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 \( t_{\text{listen}} \), 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, \( t_{\text{listen}} \) is 480µs, permitting a maximum system response time of 352µs; if the full payload is considered, \( t_{\text{listen}} \) is 768µs and only 64µs 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 \( t_{\text{decide}} \).
- 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 denoted as \( t_{\text{init}} \).
- 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 \( t_{\text{response}} \), 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 64µs.

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 frame 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.
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 (wtables), 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):

```bash
wtables -A -m dst --pan 0x22 --addr 0xFFFFFFFF
-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 26µs, while the average packet duration is 1024µs (see also Fig. 2a). This fact helps in two ways: first, we are able to observe large parts of a packet because $t_{\text{response}}$ 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 $t_{\text{response}}$ 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 64µs. 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 160µs, or
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 26µs), 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