Insecure Firmware and Wireless Technologies as "Achilles' Heel" in Cybersecurity of Cyber-Physical Systems

Andrei Costin

Abstract In this chapter, we analyze cybersecurity weaknesses in three use-cases of real-world cyber-physical systems: transportation (aviation), remote explosives and robotic weapons (fireworks pyrotechnics), and physical security (CCTV). The digitalization, interconnection, and IoT-nature of cyber-physical systems make them attractive targets. It is crucial to ensure that such systems are protected from cyber attacks, and therefore it is equally important to study and understand their major weaknesses.

1 Introduction

The exponential growth of the (industrial) Internet of Things and the increasing digitalization of everything in modern life will inherently lead to an increase in the number of cyber-physical system applications. Such systems, which have the most direct interaction with the physical world, are mainly in the medical, industry/energy, transportation, safety/surveillance, entertainment, and military sectors. This means that they have a somewhat direct connection or control over the (quality of) lives of many people. Often, however, such systems are not (completely) secure. In addition, as we demonstrate, in most cases, the main cause of insecurity lies in firmware or wireless communications.

This chapter is organized as follows. In Sect. 2, we analyze the cyber-physical system (CPS) related to the aviation transport. We will focus in particular on the next-generation ADS-B system, which will be used for radars, situational awareness, air-traffic control, and air-traffic management. In Sect. 3, we perform cybersecurity analysis and attack implementations for a wireless firing system used in fireworks pyrotechnics. In Sect. 4, we survey vulnerabilities in and attacks against CCTV and Video Surveillance Systems (VSS).

Andrei Costin
University of Jyväskylä, Jyväskylä, Finland, e-mail: ancostin@jyu.fi

2 ADS-B in Air Transport

In this section, our main focus is to demonstrate the ease, feasibility, and practicality of ADS-B compared to previous works that covered the theoretical aspects of ADS-B insecurity. To this end, we set up a practical, cost-effective, and moderately sophisticated attack against the next-generation ADS-B technologies, which are expensive and safety-critical. Although the use of a manual validation procedure [3] can partial mitigate attacks, conducting attacks on air traffic controllers and aircraft in continuous and/or decentralized manner greatly increases the potential for human error. For example, repeated erroneous messages on the air traffic control display and critical response time requirements affect the security of the entire system.

This section is based on the author's original work in [23].

2.1 ADS-B in General

Automatic Dependent Surveillance-Broadcast (ADS-B) is an Air Traffic Management and Control (ATM/ATC) surveillance system intended to replace traditional radar-based systems. It is expected to be an essential part of the next generation (NextGen) air transport system. The concept behind ATM, ATC and ADS-B is quite simple and can be summarized as follows. ADS-B avionics transmits plain text, unencrypted error-protected messages over radio transmission links approximately once per second. These messages include aircraft location, velocity, identification, and other air traffic control information.

ADS-B can be used for many purposes. It is useful in

- Improving air-traffic management and control security,
- Improving the detection and resolution of air-traffic conflicts,
- Optimizing and condensing air traffic.

ADS-B aims to dramatically improve pilots' situational awareness by providing them with access to real-time air-traffic information similar to that of air traffic controllers. For example, they receive information about other aircrafts as well as weather and terrain. ADS-B lets pilots know the position of the aircraft they are flying in relation to other aircraft without recourse to the infrastructure.

A traditional passive radar system has a relatively low resolution. Moreover, with traditional radars, the accuracy of the position depends on the distance to the plane. Radars are also usually unable to provide altitude information. ADS-B has much better coordinate accuracy and an effective range of 100–200 nautical miles [36]. Therefore, it is expected that ADS-B will allow for much better use of airspace by allowing the distance between planes to be reduced, especially near busy airports.

2.2 ADS-B in Detail

ADS-B operates on the following radio frequencies:

- 1030 MHz for active interrogation, for example from ATC towers, radars or other aircraft, and
- 978MHz/1090 MHz for active response or normal transmissions, for example from aircraft or, less frequently, from airport vehicles.

For interoperability, regulation, and tradition, ADS-B is supported by two different data connections, specifically 1090 MHz Mode-S Extended Squitter (1090ES) and 978 MHz Universal Access Transceiver (UAT). As part of the next generation ATM system, ADS-B will be developed and deployed in conjunction with Flight Information Services-Broadcast (FIS-B) and Traffic Information Service-Broadcast (TIS-B). Both FIS-B and TIS-B can be susceptible to attacks similar to those described in this section. However, such protocols are used for less critical data processing, so we have not investigated actual feasibility of the attacks but left it to be done in future work.

In terms of active response and normal broadcast, the role of the unit in the ADS-B architecture can be either a transmitter called ADS-B OUT or a transceiver called ADS-B IN. Currently, most aircraft are designated as transmitters and equipped with ADS-B OUT technology. Therefore, their role in ADS-B is to broadcast their location information for further analysis and compilation at ATC towers and ATM stations. ADS-B IN technology is currently used mainly in ATC towers. As one of the most advertised benefits of the ADS-B is the superior situational awareness it provides to the pilot of an aircraft, testing of the ADS-B IN has begun on aircraft. According to [35], SWISS is a pioneer in the use of ADS-B IN in Europe and one of only five airlines in the world to participate in the Airborne Traffic Situational Awareness (ATSAW) project. ADS-B IN is intended to enable ATSAW, spacing, separation and self-separation applications. However, from a security point of view, ADS-B IN technology in aircraft brings new challenges. Examples of challenges include verifying Online 2 reliably and real-time validation of identity, location, and flight route from a received broadcast. While the situation is well controlled at an ATC station on the ground where high-speed connection is not a problem, control is more difficult on an aircraft.

The ADS-B protocol is encapsulated in Mode-S frames. The ADS-B uses Pulse-Position Modulation (PPM) and the responses/transmissions are encoded in a certain number of pulses, each pulse being 1.0 μs long. Therefore, the data rate of the ADS-B is 1.0 Mbit/sec. The response/transmission frames consist of a preamble and a data-block. A preamble of length 8.0 μs is used to synchronize transmitters and receivers. It consists of four pulses, each of 0.5 μs in length, with intermediate space (relative to the first pulse) of 1.0 μs, 3.5 μs, and 4.5 μs. The ADS-B protocol does not specify whether Collision Detection (CD) or Collision Avoidance (CA) is used on medium radio frequency (especially considering that the transmission is plain text and digitally unsigned wireless transmission channel is used as the channel). The data blocks are either 56-bit or 112-bit and are used to encode various Downlink

Format (DF) messages. DF packets are used by a receiver, which is usually an airplane, Unmanned Aerial Vehicle (UAV), or Unmanned Aircraft System (UAS). Uplink Format (UF) messages are usually are sent by a ground station (e.g., air traffic control tower, UAT tower), but can also be sent by another airplane UAV, or UAS (e.g., Traffic Collision Avoidance System (TCAS), Airborne Collision Avoidance System (ACAS)). Related to this study, the most interesting DFs are DF11 (Mode S Only All-Call Reply) and DF17 (1090 Extended Squitter or 1090ES).

The secure Mode-S/ADS-B mode, used in the military, is encoded in DF19 Military Extended Squitter, DF22 for military use only (discussed in [44, 71]), and cryptographically coded Mode-5, which uses enhanced cryptography based on time of day and direct sequence spread spectrum modulation as specified in NATO STANAG 4193 [76] and ICAO's Annex 10 [4]. To our knowledge, the exact specifications of DF19, DF22, and Mode-5 are not public at the time of writing.

As the ADS-B is intended to support mission-critical automated and human decision-making and has a direct impact on overall air safety, it is imperative that the technology behind the ADS-B meets operational performance and security requirements. However, the main problem with ADS-B is the lack of security mechanisms, specifically lack of

- entity authentication to protect against messages send by unauthorized entities,
- message integrity checking (e.g., digital signatures, Message Authentication Codes (MAC)) to protect against message forgery or aircraft impersonation,
- message encryption to protect against eavesdropping,
- challenge-response mechanisms to protect against recurrence attacks,
- ephemeral identifiers to protect against privacy tracking attacks,
- prevention of jamming, although we did not include Denial of Service (DoS) (e.g., by jamming with radio signals) because it affects RF communication in general and is not specific to ADS-B alone.

Surprisingly, despite years of standardization [93, 90, 91, 89, 92], development, and thorough testing, the ADS-B protocol used in commercial air-traffic does not specify mechanisms to ensure that protocol messages are authentic and non-replayed, or that they comply with other security requirements.

2.3 ADS-B Attacker and Threat Models

Building the right attacker model is essential when assessing their potential actions in the system. In the ADS-B system, an attacker can be classified using several factors, such as his/her place in the system, physical position, and goals.

The attacker's place in the system can be *external* or *internal*. An external attacker is more likely. As an outsider, he/she does not require authentication or authorization, so he/she can easily execute low-cost attacks. This type of attacker can virtually belong to any group of the Classification III-A3. An internal attacker (insider) is a person the system trusts. For example, he/she could be a pilot, an air traffic controller,

an airport technician, etc. This type of attacker is encountered less frequently. He/she is mostly observed in intentional or unintentional prankster group, e.g., [29].

An attacker can be physically located on the ground or in the air. Ground attackers are most commonly analyzed. Various detection and mitigation techniques can be used against their attacks. Airborne attackers are still ignored, and such attacks may not be well understood and modeled. However, taking advantage of technological advances, they can use drones, UAVs, automatically activated luggage check-in, or passenger miniature devices capable of performing attacks.

The attacker's motivation/goal may be a prank, abuse, crime, or military intelligence. Pranks are usually considered the least offensive. However, the impact on safety can be significantly greater than expected. Attackers can include, for example, unaware pilots, "curious" and unaware technical experimenters. Abusers can have a variety of motivations, such as money, fame, message conveying. They can be invasive of privacy (e.g., paparazzi) or even pilots who intentionally abuse their access to ADS-B technology (e.g., by sending obscenities [29], drawing obscene trajectories [2]). Criminals usually have two main motivations: money and/or terror. Attackers conducting military intelligence may have state-level motivations, such as espionage, sabotage, etc. Attacks may target military intelligence agencies as well as nation states.

During the development and deployment of ADS-B, both academia and industry sought to create threat and vulnerability models to develop mitigation techniques and solutions. A wide range of identified and described threats can be found throughout the literature:

- jamming, denial of service,
- eavesdropping,
- spoofing, impersonation,
- message injection/replay
- message manipulation.

2.4 Implementation of a Wireless Attack

A similar hardware and software environment is required to trigger and demonstrate a potential attack. Next, we present the hardware and hardware settings, as well as the software modules we have used to carry out the attacks and exploits.

As the main hardware support, we used a radio device defined by USRP1 software [110]. The USRP was combined with an SBX transceiver daughter board [94] covering the frequency range of 400 MHz to 4.4 GHz. This was a good enough combination for 1030 MHz interrogation and 1090 MHz response frequencies. In addition, the transmission and reception chains could be controlled separately to provide greater flexibility for the scenarios being tested. To assess the correctness of our implementation and the effectiveness of the attacks, we used the PlaneGadget ADS-B virtual radar [85]. It is an enthusiast-level ADS-B receiver chosen for its

good price-quality ratio. However, a large number of similar ADS-B receivers are currently available and any of them could be used in such an experimental setup.

We used the open GNU Radio software package [1] as the main software base. GNU Radio is a FOSS implementation of several basic radio technologies that are useful for higher-level SDR design and applications. In particular, it provides very good software support for USRP1 and USRP2. We used USRP hardware in Universal Hardware Driver (UHD) mode, which is recommended because it supersedes the original hardware mode. In addition to the PlageGadget, we also used our USRP1 as a secondary ADS-B receiver as well as a backup device. Using USRP1 as an ADS-B IN device requires demodulation and decoder support. Fortunately, there are two public implementations of the Mode-S/ADS-B receiver module for GNU Radio. Eric Cottrell performed the historic first implementation of the Mode-S/ADS-B demodulator and decoder for pre-UHD-mode. The latest implementation for UHD-mode was done by Nick Foster [39]. Since the USRP1 was in UHD-mode, we used the gr-air-modes software module [39].

For reproducible attacks, we used the out-of-box functions of USRP1 and GNU Radio. Thus, our approach at the frame level is as follows:

- Capture ADS-B using uhd_rx_cfile at 1090 MHz;
- In UHD-mode, use TX samples to transmit reproducible captured data via GNU Radio;
- Or in pre-UHD mode, use usrp_replay_file.py to transmit reproducible captured data via GNU Radio.

(Please define TX samples!)

For message impersonation attacks, i.e. spoofing, it is necessary to implement ADS-B for PPM encoding and PPM modules. As usual, there are several ways to accomplish this. One of them is writing the original C/C++-based GNU Radio modulator and encoder [87]. Another approach we used is to perform most of the encoding and modulation in MatLab. In outline, we follow these steps:

1. Encode the detailed ADS-B data into a MatLab array as a bitstream;
2. Modulate it using PPM's `modulate()` function with a `ppm` argument;
3. Or read I/Q formatted data into MatLab (or Octave) using `read_float_binary.m` and modify the downloaded data;
4. Write the modulated data to I/Q format using `write_float_binary.m`;
5. In UHD-mode, use TX samples to transmit the modulated data via GNU Radio;
6. Or in pre-UHD-mode, use `usrp_replay_file.py` to transmit the modulated data via GNU Radio.

2.5 Key Results

Section 2 clearly verifies the inherent insecurity in the design of the commercial ADS-B protocol. Despite the fact that security vulnerabilities in ADS-B technology have been widely covered in previous academic studies and more recently in the

hacking community, fundamental problems in the architecture and design of ADS-B have never been addressed and fixed. Given the time and money invested so far and still to be invested, it is unclear why such a mission-critical safety protocol does not address safety at all and there is not even a security chapter in the main requirements specification document [93].

In conclusion, the most important and intended contribution of this study is to raise awareness among academia, industry, and policy makers that critical infrastructure technologies, such as ADS-B, require real security to operate safely and in accordance with requirements. We can do this by showing that a low-cost hardware setup combined with moderate software in multi-million dollar technology is enough to expose the system to dangerous security and operational failures while failing to take advantage of basic security mechanisms such as message authentication.

3 Wireless Firing Systems for Remote Explosives and Robotic Weapons

In this section, we examine the risks of the firing system. We describe our experience in discovering and exploiting a wireless firing system in a short period of time without prior knowledge of such systems. We demonstrate our methodology starting from firmware analysis to discovering vulnerabilities. Our static analysis helped us acquire a system suitable for the purpose, which we then analyzed in depth. This allowed us to confirm the presence of exploitable vulnerabilities in the actual hardware. Finally, we stress the security of hardware and software, as well as the need to monitor the safety of the use of pyrotechnic firing systems.

This section is based on the author's original work in [24].

3.1 Main Motivations

Fireworks are mainly explosives for entertainment purposes. A fireworks event, also called a pyrotechnic show or fireworks show, is a demonstration of the effects produced by fireworks devices. Fireworks devices are designed to produce, among other things, noise, light, smoke, and floating materials (e.g., confetti). Fireworks events and fireworks devices are controlled by fireworks firing systems. In addition to fireworks, firing systems often serve other primary industries as well. These include special effects production and military training or simulation.

Despite the fact that fireworks are intended for celebrations, their usage is often associated with a high risk of destruction, injury, and even death. Many recent news and studies show the dangers of fireworks [30, 83]. Sometimes fireworks are even used as real weapons in street clashes [108]. Fireworks accidents are often the result of improper handling of equipment, non-compliance with safety regulations, or poor quality fireworks. Fatal consequences can also occur when CPS-style systems with

software defects are connected to ammunition/explosive firing systems [14]. Another risk factor is that fireworks are generally intended to be displayed in densely populated areas. Accidents continue to occur despite the strict control of the distribution of fireworks and the mandatory professional license of a fireworks shooter.

Classically, fireworks firing systems consist of mechanical or electrical switches and electrical wires (often called shooting wires). This type of setup is simple, efficient and relatively safe [38]. However, it dramatically limits the effects, complexity, and implementation of fireworks systems and events. The development of software, embedded and wireless technologies can be fully utilized in fireworks systems. A modern (wireless) firing system is at the same time a complete Embedded Cyber-Physical System (ECPS) and a combination of Wireless Sensor/Actuator Network (WSAN). As fireworks firing systems increasingly rely on wireless, embedded, and software technologies, they are exposed to the same risks as other ECPS, WSAN, or computer systems. Recent research has shown that both critical and embedded systems have acquired a poor security reputation. For example, airplanes can be fooled by new radar systems [23], car control can be taken over [18, 65], car driving can be compromised by failure [55], an implanted insulin pump can be made to malfunction [86], or nuclear plant PLCs can be rendered inoperative [37, 68].

3.2 Overview of Fireworks and Pyrotechnics Systems

The pyrotechnics of fireworks is typically composed of:

- Remote control modules,
- Firing modules,
- Wired connections,
- Wireless transceivers,
- Igniter clips,
- Mortars,
- Pyrotechnic devices.

Remote control modules (sometimes also called main controls) control the entire show, which includes sequencing cues and the transmission of fire commands. They connect to the firing modules via wired or wireless connections. In simple systems, one remote control module is connected to all firing modules, while in more complex shows, there are several remote control modules, each of which is connected to a specific set of firing modules depending on the show. All remote control modules work independently. These devices rely on a micro-controller embedded in its own firmware.

The *firing modules* receive fire commands from the remote control modules and activate the minimum ignition current to the igniter clips. The firing modules are based on micro-controllers and have their own firmware. *Wired connections* are described here for completeness, but in our case, all remote control and firing modules were wireless. Classic fireworks firing systems consist of electrical wiring

between the remote control and the firing modules [38]. Simple connection cables with End-Of-Line (EOL) resistors are used for secure termination of wire loops. EOL resistors allow the remote control to detect wiring problems or tampering in short circuit situations while monitoring field wiring.

Wireless transceivers enable wireless connections between remote control modules and firing modules. These connections are usually implemented with 433.92 MHz modules (often capable of using rolling codes [10]), or 2.4 GHz ZigBee-compliant (IEEE 802.15.4) modules that support AES according to the standard.

The *igniter clips* connect the firing modules to the pyrotechnic devices inside the mortar. They ignite a fire when the firing module activates the minimum current. *Mortars* contain pyrotechnic devices. They also ensure the safe launching and firing of the pyrotechnic device into the sky. *Pyrotechnic devices* are actual pyrotechnic compositions that produce visual and sound effects in the sky after a firing.

3.3 Preliminary Analysis

First, we performed a large-scale firmware analysis by gathering firmware images from the Internet, reaching 172,000 firmware candidates [26]. Once the firmware images were unpacked, we processed each image with simple static analysis, correlation, and reporting tools, leading us to discover 38 previously unknown vulnerabilities. In the process, we accidentally discovered firmware images for the wireless firing system. We omit the name of the vendor and the system for safety and ethical reasons. Analysis of the firmware images for that system revealed components (strings, binary codes, configurations) that appeared insecure. The findings were convincing enough, so we acquired the devices for a detailed analysis. Another motivating factor for the acquisition was that, according to the vendor, this system is used by "over 1000 customers in over 60 countries". These systems seem to be particularly popular in fireworks companies.

3.3.1 Firmware Analysis

Our crawlers collected, among others, several Intel Hexadecimal Object File (Intel Hex) firmware images dedicated to the wireless firing system from the Internet. After unpacking, we used several heuristics, including keyword matching. Keyword matching searches for specific keywords such as backdoor, telnet, UART, shell, which often allows to find multiple vulnerabilities. The firmware images matched with the string Shell.

Based on this, we isolated those firmware images and further analyzed them using automated and manual approaches. We detected several security issues from the analyzed images. First, the Intel Hex format alone does not provide encryption or authentication, so the functionality is openly explorable by an attacker and thus likely to be open to malware. In addition, the Intel Hex format provides attackers

with mechanisms to insert code or data to memory regions that may not be designed to be accessed.

3.3.2 Wireless Communication Analysis

Wireless communication systems, like many others from other vendors, include a 2.4 GHz ZigBee (IEEE 802.15.4) CEL MeshConnect transceiver. Discovering, configuring, installing and pairing these units, as well as updating the firmware, is done through Synapse Portal [101]. We installed Synapse Portal and then ran a discovery and configuration query.

The wireless chipsets for the remote control, firing, and firmware reprogramming modules include AES-128-compatible firmware. However, encryption is not enabled, the encryption key is not present, and the AES-128 appears to be unused. In addition, the system documentation does not appear to support AES-128-secured configuration steps. Surprisingly, even if those devices conform to the standards and have AES-128 capabilities, message authentication or encryption is not used. This is likely due to difficulties in properly configuring key management and distribution. Thus, when used in this way, AES-128 carries the risk of functional failure to the fireworks rather than acts as a safety mechanism.

Further analysis revealed that it is possible to load Python application code into wireless remote chipsets. These scripts are executed in a Python interpreter on a wireless chipset microcontroller (MCU), see [100]. The provided interpreter framework is a subset of Python. Before downloading to target nodes, Synapse Portal compiles these Python scripts in binary format and stores them as SNAPpy files (with extension .spy), see [102]. The binary format is assigned to a specific MCU that drives a particular wireless chipset. These scripts expose the entry-points (functions) that other wireless nodes can call remotely (via RPC). Scripts can interact with the wireless chipset MCU or General-Purpose Input/Output (GPIO) ports. Usually, these GPIO ports are connected to the main MCU of the remote control or firing module. This allows interaction with main MCUs as well as IO peripherals such as buttons, displays and igniter clips. The typical use of script entry-points is as follows. The remote control module processes CSV orchestration scripts. When it decides that a fire command is required, it sends a ZigBee packet containing a higher-level message to a specific entry-point of a particular remote module.

The usual standard firing procedure is as follows:

1. Each firing module is connected to a specific remote control module.
2. The physical keys of the firing modules are turned to standby mode.
3. Staff move to the statutory safety distance to fire cues.
4. The keys for the remote controls are turned on.
5. After making sure everything is safe and ready, the staff presses the power button on the remote control. The remote, in turn, sends a wireless digital command to the firing module, which enters standby mode for incoming fire commands.
6. Staff begin the show by sending commands to each shooting module, either manually or according to the script.

Each firing module accepts arming, disarming, and firing commands only from its paired remote control. The pairing is forced by checking the remote control's 802.15.4 short address (similar to MAC address filtering).

3.4 Wireless Threats

The lack of encryption and mutual unit authentication opens up the system to multiple attacks, particularly sniffing, spoofing, and replaying. We describe a simple attack, however, that we consider the most dangerous to the fireworks show staff.

The attacker would proceed as follows. He/she eavesdrops on packets (broadcasts, multicasts, node-to-node) by learning from them the 802.15.4 addresses of each remote control and firing modules and the corresponding pairing. For each pair learned, the attacker spoofs the 802.15.4 addresses on the remote control and the digital arm command sent to the paired firing module, and immediately sends a fire command from all cues when the digital arm confirmation comes from the firing module. As a result of such an attack, when the show operator turns the physical key of a firing module to the arming position, that firing module immediately receives a series of digital arming and firing commands from all cues. This fires all pyrotechnic loads and in the worst case does not give the staff enough time to move to a safe distance. Thus, it overrides the security of the physical key and the separation of functions. We successfully implemented this attack on the systems we acquired using the components described in this section.

Alternatively, an attacker could easily replace the default Python functions responsible for firing cues with arbitrary malicious Python functions. For example, each malicious firing cue function could fire a firing module from all of the cues at once instead of its own cue, which could cause a massive chain explosion. Or it cannot fire from the cues at all or fires randomly, leaving the fireworks show below expectations. Last but not least, an attacker can remotely set random encryption keys on remote nodes. This would mean a denial of service to the legitimate user, as his/her devices would no longer be able to communicate with other devices used for fireworks. This can definitely ruin a holiday party or harm competitors in professional fireworks competitions.

3.5 Implementing a Wireless Attack

We implemented simple attacks, such as *message replay* and *unauthorized message injection* (e.g., the command "fire all"). However, it is obvious and trivial to extend the implementation to automatically and continuously sniffing out new firing modules and subsequently spoofing remote control sequences. Next, we present details of the software and hardware we used to carry out the attacks.

SNAP Stick SS200 The SNAP Stick SS200 [103] is firmware software primarily for remote control and firing modules and is based on Atmel's well-known ATmega128RFA1 chipset. Using the SNAP Portal utilities and its own firmware (Synapse ATmega128RFA1 Sniffer), the SNAP Stick SS200 can be converted to a SNAP-specific 802.15.4 sniffer that sniffs and decodes 802.15.4 packets based on Synapse's higher-level protocol semantics (e.g., multicast, broadcast, peer-to-peer or multicast RPC calls). We used it to sniff and record packets between the remote control and firing modules during their normal operation. Finally, we also used it to validate packet injection and replay attacks. If this sniffer received them, the remote control and firing modules would see our rogue packets. Otherwise, we had to fix our injector (regardless of whether our lower-level raw packet sniffer saw them) and then re-test the sniffed packets and the actual behavior of the devices.

GoodFET GoodFET [43] is an embedded bus adapter for various microcontrollers and radios, while also providing great open source support for advanced attacks. Its TelosB-compatible firmware allows sniffing, among other functionalities. We tested our attack with GoodFET firmware running on TelosB.

KillerBee KillerBee [64] is a framework and tools for exploiting ZigBee and 802.15.4 networks. It provides convenient pre-compiled GoodFET firmware for extra attack functionality. We tested our attack with such GoodFET firmware running on TelosB.

TelosB An sniffer based on SS200 is useful for SNAP protocols and visualization, but it filters and strips down the packets, which is largely limiting. We needed a lower level raw packet sniffer. We also needed an cheap and open source supported approach. Crossbow's TelosB [104] hardware and GoodFET firmware fit perfectly, so we used them as an additional, much more verbose and raw sniffer. After learning the SS200 higher-level packets for critical commands, we correlated them with the raw packets recorded by TelosB (running GoodFET firmware). Alternatively, Zigduino [118] could have been used for this task.

Econotag Redwire Econotag is an inexpensive and convenient open source platform for 802.15.4 networks. We assembled sequences of packets that sent commands to arm and fire from the remote control to the firing module. Finally, we encoded an infinite loop of these sequences in custom firmware. Once plugged, Econotag performs an attack on the firing module when its key is turned to the physical arm position. Alternatively, Zigduino [118] could have been used for this task as well.

3.6 Main Outcomes

We were able to quickly and automatically isolate the firmware of critical remote firing systems and identify several potential vulnerabilities using both automatic and manual static analysis. These vulnerabilities include unauthorized firmware updates, unauthenticated wireless communications, wireless communications sniffing and

spoofing, arbitrary code injection, functionality trigger, and temporary denial of service. We have successfully implemented and tested an unsophisticated attack that can have devastating consequences. Our conclusion is that, given the risk posed by use, the security of wireless firing systems should be taken very seriously. We also conclude that such systems need to be more rigorously certified and regulated.

We stress the need and urgency to introduce software and hardware compliance verification similar to that of the DO-178B and DO-254 respectively. We strongly believe that these small improvements, along with the suggested solutions, can definitely help improve the security and safety of wireless embedded systems. Last but not least, we discussed the issues with the vendor. The firmware update now deployed fixes most security issues. Unfortunately, due to more than 20 vendors, wireless firing systems may be vulnerable to similar attacks, especially those for which a firmware update is not available.

4 CCTV for Physical Security

Video surveillance, Closed-Circuit Television (CCTV), Digital/Network Video Recorder (DVR/NVR), and IP-camera (IPcam) systems[1] have become very common all over the world. Currently, the use of VSSs is central to most, if not all, areas of life in modern society. They are used very widely, from law enforcement and crime prevention to transport safety, traffic monitoring, and industrial process and retail control. Unfortunately, their unauthorized [116, 109], illegal [117], and even criminal [63] use is also common. Their number is incredibly large; in some reports it is estimated at 245 million cameras/systems [56]. It is expected that by 2021, there will be more than a billion CCTV cameras worldwide [20].

This section is based on the author's original work in [22, 21].

4.1 CCTV in General

Most of the concerns about video surveillance systems are related to privacy protection for obvious reasons. Improving the privacy of VSSs is particularly important in the light of global surveillance revelations, and specifically video surveillance scandals [31]. However, in addition to privacy issues, an insecure or compromised VSS can raise a myriad of other non-privacy issues. For example, data breaches were shown to endanger prison security [63], pose theft risks to money-based institutions such as banks [6] and casinos [117], emotionally affect people (especially children) [54], and interfere with police and law enforcement [80].

At a time when embedded devices are increasingly being analyzed on a large scale for security vulnerabilities [26, 27], it is no surprise that security researchers

[1] We call such a system a Video Surveillance System (VSS).

have dramatically increased their focus on VSSs [81, 70, 96, 52, 21]. These and similar studies found more than a handful of vulnerabilities [111, 7, 8, 73, 59, 16, 33, 34, 60, 17, 53, 77] that have a large-scale impact in real life [61, 114]. The number of vendors and the variety of vulnerabilities revealed in the investigations clearly indicate the unhealthy state of cyber-security in video surveillance systems.

4.2 Visual-layer Attacks

Compared to other embedded systems, video surveillance systems have an additional level of abstraction, i.e. *visual layer*. Therefore, it is possible to (ab)use this layer to carry out novel attacks on video surveillance systems that take advantage of imaging semantics and image recognition. Costin [21] first presented such an attack on CCTV cameras as the back door of the visual layer. Mowery et al. [75] carried out a similar attack on a full-body scanner as a secret knock-on image.

This attack is multi-stage and works on the visual layer as follows. In the first stage, the VSS is infected with a malicious component (e.g., hardware, firmware). In some scenarios, this can be achieved locally via a malicious firmware update over a USB port and remotely via a command injection or a malicious firmware update over a web interface. In other scenarios, a VSS or CCTV system with pre-installed malware could be sold through legitimate sales channel [113, 78]. In the second stage, the malicious component is triggered and controlled through the input of a malicious image that is "visualized" by the cameras and video sensors.

In the most general case, the trigger command can be coded in any arbitrary data-to-image encoding scheme[2]. First, a malicious component could be pre-programmed to blur an attacker's face or the license plates of an attacker's car, or to disable certain functions of a surveillance system (e.g., video recording functions or scanning a prohibited object such as a gun in a full-body scan [75]). Such malicious functions could be used for theft and other crime. Second, the malicious component could read commands from QR-like codes. Malicious images [62] could be printed on t-shirts, cars, or any accessory that is sufficiently visible to cameras. The command could be "stop recording", "blur the face of an attacker with a malicious image/QR-code", "contact the command and control center" or "update malicious components". A variation of such an attack was carried out in the hacking of Google Glass [45]. It used a specially crafted QR code as malicious image input to control (unauthorized and unattended) Google Glass and force it to visit a malicious URL.

Optical covert channel techniques could be used to hide the visual layer attack and the resulting load from human operators. Taking advantage of the camera's sensitivity to the infrared and near-infrared spectra, an attacker could send "invisible" information. An attacker could also use techniques similar to `VisiSploit` [46], except that the channel would be used to inject data and commands and not to exfiltrate data.

[2] QR-codes are a popular implementation of such data-to-image encoding schemes.

Finally, visual-layer attacks are certainly not far-fetched. Because visual layer information is processed at a certain point (e.g., image compression, face recognition, Optical Character Recognition (OCR)), both intentional and unintentional errors can occur. An infamous example of an unintentional error is Xerox scanners and copiers that randomly altered document numbers and data [66]. Because incredibly complex processing (e.g., image compression, face recognition, Automatic License Plate Reading (ALPR)) is built into modern video surveillance systems, it is reasonable to assume that similar (both intentional and unintentional) problems in the visual processing layer can allow an attack against them as well.

4.3 Covert-channel Attacks

In recent years, covert channels and data exfiltration (especially in air-gap environments) have been the subject of productive research. The channel used can be electromagnetic [67, 112, 48, 47], acoustic [79, 51, 50], thermal [49, 72], or optical [69, 95, 46, 88]. With regard to VSS and CCTV systems, we will introduce one novel covert channel and look more broadly at the use of several existing covert channels. Although the channels we present can mainly be used to exfiltrate data using the compromised VSS and CCTV component [113, 78], they can also be used for autonomous and distributed command-and-control functions.

4.3.1 Normal and Infrared LEDs

In modern electronic equipment, such as device status indicators, LED lights have been used repeatedly in covert channels and data exfiltration [69, 19, 95]. Smart LED bulbs have also recently been shown to pose similar threats [88]. Although LEDs are sometimes physically connected to hardware and cannot be controlled from software/firmware, recent attacks show that manipulating LEDs from software/firmware is becoming increasingly practical and feasible [13]. VSS and CCTV systems usually have plenty of status LEDs on both core equipment and outdoor CCTV cameras. Therefore, LEDs in VSS and CCTV systems could also be used in data exfiltration attacks.

There is one major drawback to (ab)using normal LEDs in such attacks. If the LEDs are handled in an eye-catching way (e.g., abnormal blinking frequencies, unusual luminosity levels), they are distinguishable to the human eye, allowing the covert channel to be exposed. Therefore, we propose the use of InfraRed (IR) LEDs in optical covert channels. IR-LED arrays are installed in almost any modern CCTV camera. IR LEDs are used for illumination and provide IR night vision for cameras and VSSs. One important characteristic of IR LEDs is that when they operate, they

are often invisible[3]. For example, another camera without IR cut-off filters (e.g., another IR-compatible CCTV camera) must be used to detect the operation of the IR LEDs. Therefore, IR-compatible CCTV cameras can use the intensity of IR LEDs (or their on and off mode) to modulate and exfiltrate data. Such exfiltration would be invisible to the human eye.

Ambient lighting can affect the success of an attack. When it is dark, changing the intensity/status of the IR LEDs is immediately reflected in the surveillance camera image, so operating personnel may notice that something is wrong. When the environment is lighted, changes to the IR LEDs would not be very visible in the surveillance camera image, but an attacker could still intercept the exfiltrated data remotely.

4.3.2 Covert Channels

Recently, Guri et al. [46] presented `VisiSploit`, a new type of optical covert channel that, taking advantage of the limitations of human visual perception, leaks data imperceptibly through the LCD display of a standard computer. Most VSS and CCTV systems are connected to screens that are fully or partially visible to the public. These screens display real-time images from one or more cameras in the system. For example, this is especially popular in supermarkets to deter shoplifting and to help staff early detect potential illegal or unethical activity. VSS and CCTV systems can also be seen to be used in this way in the operational centers of large car parks, in the reception lobbies of organizations (e.g., companies, hotels, elite residences) and in many other places. Therefore, the compromised VSS and CCTV component could use screens installed in this way in conjunction with `VisiSploit` techniques to exfiltrate the data.

Steganography is the art of hiding information inside other information (e.g., images, documents, media streams, or network protocols). Although many different "carrier media" can be used for this purpose, digital images are the most popular due to their prevalence on the Internet and their concealment efficiency. A comprehensive overview of image steganography is presented in [82, 74]. A special feature of VSS and CCTV systems is that virtually all systems provide both video and image streams [32]. Image streams can be either motion images (e.g., MJPEG) or still snapshots and can usually be accessed in URLs such as `http://CAM-IP/now.jpg`, `http://CAM-IP/shot.jpg`, or `http://CAM-IP/img/snapshot.cgi?size=2`.

Therefore, the compromised VSS component (e.g., CCTV camera, DVR, NVR) can exfiltrate the data employing steganography when generating the above-mentioned images/image streams. The attacker then only needs to capture digital snapshots of well-known URLs and recover the exfiltrated data. Whether an attacker has access to image streams and how he/she can access is not the purpose of this section. However, recent projects such as TRENDnet Exposed [106], Insecam [57],

[3] Almost always invisible, but it also depends on characteristics of the IR LEDs used. Here, we assume that it is difficult, if not impossible, for the human eye to easily distinguish between normal and abnormal use of IR LEDs.

Shodan images [97], corroborate studies such as [28], which demonstrate that it is feasible, even very easily in VSS and CCTV systems protected by current cyber security practices. To prevent exfiltration of data in steganography, as discussed above, automated methods could be used to detect steganography [12, 41].

4.3.3 Mechanical Movement and Position of the CTTV Camera

Many modern CCTV cameras have so-called Pan-Tilt-Zoom (PTZ) functionality. With PTZ, a CCTV camera can move or stay fixed in almost any direction in 3D (e.g., with pan and tilt functions) and also zoom in and out with multiple zoom factors (e.g., using a high-precision lens). Such functionality is usually implemented with stepper motors built into specific camera models and is generally controlled by PTZ data protocols. PTZ data protocols are byte sequences of commands sent over a communication channel to control pan, tilt, and zoom. PTZ commands are classically sent over RS-422 or RS-485 links, but can also be sent over classical Ethernet and WiFi channels. PTZ commands can be sent to PTZ-compatible cameras from custom PTZ-controllers (e.g., a special joystick keyboard for surveillance personnel) or from software (e.g., OS-specific heavyweight clients or browser-based lightweight clients).

In this context, a compromised CCTV camera can exfiltrate data to an external attacker by encoding data about its position or changes in movement. For example, it could change its normal fixed position to another specific fixed position that would encode a certain value. Assume that a compromised camera on the wall in its normal position "looks" *down-and-right*. To exfiltrate data, the compromised camera would then encode:

- bits `00`, moving itself to "look" *up-and-right*;
- bits `01`, moving itself to "look" *up-and-left*;
- bits `10`, moving itself to "look" *down-and-left*.

Adding bits to the data resolution (which increases exfiltration data rate) would increase the number of abnormal positions – just like in Phase-Shift-Keying (PSK) modulation – which would require an attacker to observe the compromised camera more closely from the outside.

Many VSS and CCTV systems are audio-capable, which allows them to record and process one or more audio channels coming from external microphones or microphones built into CCTV cameras. Therefore, a compromised VSS component (e.g., CCTV camera, DVR, NVR) can use the audio layer as a command-and-control channel, for example, by means of *hidden voice command* techniques [15].

4.4 Denial-of-Service and Jamming Attacks

We would like to emphasize the importance of Denial-of-Service (DoS) and jamming attacks on video surveillance systems. In this case, the emphasis is on the VSSs as

the *final target* of the attack. In cases where VSSs are infected and used in botnets to carry out DDoS attacks on other systems as final targets, VSS plays a role as the source of the attack [116].

In most cases, uninterrupted and untampered operation is critical to video surveillance systems, for example because they are used to monitor and record crimes or other important activities. Producing a DoS attack on a CCTV system for just a minute could cause it to miss an important event, such as an extremely fast bank robbery [6, 105] or a worse crime [63]. While a DoS attack on a home router could be a minor nuisance, DoS attacks on video surveillance systems have a critical impact that needs to be considered in design, evaluation, and testing. However, this in itself is non-trivial, as explained in detail in [42].

4.5 Online Network Attacks

The most useful and used feature of a modern video surveillance system is the *plug-and-play* feature for ease of installation and deployment, as well as for *remote access control* and video monitoring. As a result, many video surveillance systems are connected to the Internet [57]. Thus, they are directly exposed to the Internet, often even with default settings and credentials [28]. Therefore, we tried to estimate the number of video surveillance systems on the Internet in order to estimate the magnitude of potential exposure.

For this purpose, we compiled an extensive list of queries about video surveillance systems and then ran the queries in both online services and existing Internet scanning databases. Using the Shodan [98] online service, these queries revealed an incredible amount of over 2.2 million video surveillance systems produced by more than 20 vendors. Using the Internet Census 2012 database [58], these queries returned more than 400,000 video surveillance systems produced by more than 10 suppliers. At the same time, according to some reports [56], in 2014, there were nearly 245 million video surveillance cameras installed in the world. Unsurprisingly, finding, tracking, and publishing[4] online video surveillance systems that are vulnerable, compromised or poorly protect the privacy of their owners has always been an interesting topic of discussion. Projects such as TRENDnet Exposed [106], Insecam [57], Shodan images [97] and EFF ALPR [84] are examples of such initiatives. As a result, these projects received an incredible amount of media attention, public scrutiny, and outrage, again raising the issue of the lack of security and privacy in modern video surveillance systems.

Cui and Stolfo [28] reported that the inferior 39.72% of the cameras and surveillance systems they analyzed on the Internet in 2010 used default credentials. This basically means that they are completely vulnerable to all kinds of attacks, such as video feed eavesdropping[5], malicious firmware updates, and DNS hijackings. As a

[4] Many times along with their screen shots and video feeds.

[5] Practically extensively demonstrated in projects such as TRENDnet Exposed [106], Insecam [57], and Shodan images [97].

further example, we analyzed a set of firmware images from the DVR system and discovered a full admin back door. We then correlated the identification information extracted from the firmware images with the results of the above queries. The result was more than 130,000 affected devices using an online connection.

Even though some of these systems (i.e., their IP addresses) and vendors may overlap (or cannot be accurately calculated), these results give a lower limit on the vulnerability of video surveillance systems to cybersecurity threats. Running Internet queries and using vulnerability estimations of previous works [28] proved to be a very effective method for estimating the number of potentially exposed and vulnerable video surveillance systems.

4.6 Key Takeaways

Section 4 provides a systematic review of the security of video surveillance systems, detailing threats, vulnerabilities, attacks, and mitigation. The review is based on publicly available data as well as existing classifications and taxonomies. It provides comprehensive information on how video surveillance systems can be attacked and protected at different levels. This structured information can then be used to better understand and identify the security and privacy risks associated with the development, deployment, and use of these systems.

5 Conclusions

In this chapter, we looked in more detail at several CPS use-cases. In Sect. 2, we analyzed CPS related to the aviation transport sector and focused in particular on the next-generation ADS-B system used in radars, situational awareness, air-traffic control, and air-traffic management. We have demonstrated through real lab attacks that the wireless communication on which the entire ADS-B system is based is inherently insecure and vulnerable to most wireless attacks (e.g., jamming, eavesdropping, spoofing, impersonation).

In Sect. 3 we performed a cybersecurity analysis and attack implementations for a wireless firing system used in fireworks pyrotechnics. These types of CPS are particularly troublesome because they deal directly with explosives, thus threatening human lives and the physical world. We demonstrated that by starting with insecure firmware, we were able to quickly find cybersecurity issues in wireless communication related to the triggering the explosives. We also demonstrated that carrying out dangerous attacks is relatively easy and feasible even by incompetent attackers.

In Sect. 4 we surveyed the vulnerabilities in and attacks on CCTV and video surveillance systems. More than a billion CCTV cameras by 2021 [20] will represent perhaps the largest IoT and CPS attack surface in terms of number of devices. The Mirai botnet fully demonstrated the devastating power of just a relatively tiny fraction

of compromised CCTV/DVR/VSS systems [5]. As a CPS, the risks for and from CCTV comes from the direct interaction with the physical world in terms of privacy, face recognition, and (un)lawful surveillance. As discussed, mainly CCTV firmware vulnerabilities (but also wireless attacks against it) are the main cybersecurity risk factors for such systems.

In summary, we have found that CPSs in several critical sectors are prone to security vulnerabilities and attacks. All of the attacks presented can be executed with limited knowledge and affordable hardware/software setups. However, most of the vulnerabilities are in the firmware of the devices or in the wireless communications used in the system.

Last but not least, we invite the interested reader to take a deeper look at the following related works [115, 26, 27, 25, 40, 9, 14, 107, 99, 11].

Acknowledgements The author of this chapter would like to acknowledge the contributions of the author's collaborators, editors, editing assistants, and everyone involved in the production of this book. In particular, the author would like to acknowledge the contributions of Prof. Aurélien Francillon (EURECOM) as part of co-authoring the original papers related to Sects. 2 and 3.

References

1. About GNU Radio. GNU Radio, `https://gnuradio.org/about`.
2. K. Adjei-Darko. Pilots draw penis in sky over Russia: Investigation over flight path. Nationwide News Pty Limited, `https://www.news.com.au/travel/travel-updates/travel-stories/pilots-draw-penis-in-sky-over-russia-investigation-over-flight-path/news-story/4d436870952e06eec36ec70eb8d79298`, 2020.
3. ADS-B radar-like services: Preliminary hazard analysis. Capstone Safety Engineering Report #1 1, Federal Aviation Administration (FAA), 2000. `https://www.faa.gov/nextgen/programs/adsb/Archival/media/SERVOL1.PDF`.
4. Aeronautical telecommunications. Volume I: Radio navigational aids. Annex 10, International Civil Aviation Organization (ICAO), 2006.
5. M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J.A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou. Understanding the Mirai botnet. In *SEC'17: Proceedings of the 26th USENIX Conference on Security Symposium*, pages 1093–1110, Berkeley, CA, 2017. USENIX Association.
6. J. Aron. Want to rob a bank? Hack your way in. *NewScientist*, 220(2937), 2013.
7. B. Austin. Trendnet cameras: I always feel like somebody's watching me. Console Cowboys, `http://goo.gl/sYkUAF`, 2012.
8. B. Austin. Swann song: DVR insecurity. Console Cowboys, `http://goo.gl/oY3z3w`, 2013.
9. G. Avoine and J. Hernandez-Castro, editors. *Security of ubiquitous computing systems*. Springer, 2021.
10. AVR411: Secure rolling code algorithm for wireless link. Atmel Application Note 2600E-AVR-07/15, Atmel, 2015.
11. R.S. Baheti and H. Gill. Cyber-physical systems. In T. Samad and A. Annaswamy, editors, *The Impact of Control Technology: Overview, Success Stories, and Research Challenges*, pages 161–166. IEEE Control Systems Society, 2011.

12. G. Berg, I. Davidson, M.Y. Duan, and G. Paul. Searching for hidden messages: Automatic detection of steganography. In *Proceedings of the Fifteenth Conference on Innovative Applications of Artificial Intelligence*, pages 51–56. AAAI, 2003.
13. M. Brocker and S. Checkoway. iSeeYou: Disabling the MacBook webcam indicator LED. In *SEC'14: Proceedings of the 23rd USENIX Conference on Security Symposium*, pages 337–352, Berkeley, CA, 2014. USENIX Association.
14. A. Cardenas and S. Cruz. Cyber-physical systems security knowledge area. Issue 1.0, The Cyber Security Body Of Knowledge (CyBOK), 2019.
15. N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden voice commands. In *SEC'16: Proceedings of the 25th USENIX Conference on Security Symposium*, pages 513–530, Berkeley, CA, 2016. USENIX Association.
16. CCTV systems. CVE Details, `http://goo.gl/IB1Hk7`.
17. CCTV systems. InsecureOrg, `http://insecure.org/search.html?q=cctv`.
18. S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive experimental analyses of automotive attack surfaces. In *SEC'11: Proceedings of the 20th USENIX Conference on Security, Vol. 4*, pages 447–462, Berkeley, CA, 2011. USENIX Association.
19. J. Clark, S. Leblanc, and S. Knight. Hardware Trojan horse device based on unintended USB channels. In *2009 Third International Conference on Network and System Security*, pages 1–8. IEEE, 2009.
20. E. Cosgrove. One billion surveillance cameras will be watching around the world in 2021, a new study says. `https://www.cnbc.com/2019/12/06/one-billion-surveillance-cameras-will-be-watching-globally-in-2021.html`, 2019.
21. A. Costin. Poor man's panopticon: Mass CCTV surveillance for the masses. Presentation at POC 2013, 2013.
22. A. Costin. Security of CCTV and video surveillance systems: Threats, vulnerabilities, attacks, and mitigations. In *TrustED'16: Proceedings of the 6th International Workshop on Trustworthy Embedded Devices*, pages 45–54, New York, 2016. ACM.
23. A. Costin and A. Francillon. Ghost in the air(traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices. Presented at Black Hat USA 2012, `http://lib.21h.io/library/Y8STRIX5`, 2012.
24. A. Costin and A. Francillon. A dangerous 'pyrotechnic composition': Fireworks, embedded wireless and insecurity-by-design. In *WiSec'14: Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks*, pages 57–62, New York, 2014. ACM. Short Paper.
25. A. Costin and J. Zaddach. IoT malware: Comprehensive survey, analysis framework and case studies. Presentation at Black Hat USA 2018, 2018.
26. A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti. A large-scale analysis of the security of embedded firmwares. In *SEC'14: Proceedings of the 23rd USENIX Conference on Security Symposium*, pages 95–110, Berkeley, CA, 2014. USENIX Association.
27. A. Costin, A. Zarras, and A. Francillon. Automated dynamic firmware analysis at scale: A case study on embedded web interfaces. In *ASIA CCS '16: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 437–448, New York, 2016. ACM.
28. A. Cui and S.J. Stolfo. A quantitative analysis of the insecurity of embedded network devices: Results of a wide-area scan. In *ACSAC'10: Proceedings of the 26th Annual Computer Security Applications Conference*, pages 97–106, New York, 2010. ACM.
29. Dodgy callsigns from SkyWest flights. RadioReference.com, `https://forums.radioreference.com/threads/dodgy-callsigns-from-skywest-flights.216125/`, 2011.
30. K. Dolak and A. Shaw. Fireworks mishaps, parade fatalities Mar Fourth of July displays: July 4 celebrations across the country befell accidents and fatalities. ABC News, `https://abcnews.go.com/US/`

fourth-july-accidents-mar-parades-fireworks-displays/story?id=19583533, 2013.
31. Domain Awareness Center. Oakland Wiki, http://oaklandwiki.org/Domain_Awareness_Center.
32. Download the iSpy source code. iSpyConnect.com, https://www.ispyconnect.com/source.aspx. Accessed: July 26, 2016.
33. DVR systems. CVE Details, http://goo.gl/Xmv1jN.
34. DVR systems. InsecureOrg, http://insecure.org/search.html?q=dvr.
35. Eco-care reaches new (flight) levels. *SWISS Magazine*, May:104–106, 2012.
36. Fact sheet: Automatic Dependent Surveillance-Broadcast (ADS-B). Federal Aviation Administration (FAA), https://www.faa.gov/news/fact_sheets/news_story.cfm?newsKey=4172, 2006.
37. N. Falliere, L.O. Murchu, and E. Chien. W32.Stuxnet dossier. White paper, Symantec Corporation, 2010.
38. Fireworks electric firing systems. Skylighter Inc., https://www.skylighter.com/fireworks/how-to/setup-electric-firing-systems.asp, 2018.
39. N. Foster. gr-air-modes. GitHub, https://github.com/bistromath/gr-air-modes, 2012.
40. A. Francillon, S.L. Thomas, and A. Costin. Finding software bugs in embedded devices. In G. Avoine and J. Hernandez-Castro, editors, *Security of Ubiquitous Computing Systems*, pages 183–197. Springer, Cham, 2021.
41. J. Fridrich, M. Goljan, and R. Du. Reliable detection of LSB steganography in color and grayscale images. In *MM&Sec'01: Proceedings of the 2001 Workshop on Multimedia and Security: New Challenges*, pages 27–30, New York, 2001. ACM.
42. M. Gasser. *Building a secure computer system*. Van Nostrand Reinhold, 1988.
43. GoodFET. GitHub, https://github.com/travisgoodspeed/goodfet.
44. R.D. Grappel and R.T. Wiken. Guidance material for Mode S-specific Protocol application avionics. Project Report ATC-334, Massachusetts Institute of Technology, 2007.
45. A. Greenberg. Google Glass hacked with QR code photobombs. Forbes, https://www.forbes.com/sites/andygreenberg/2013/07/17/google-glass-hacked-with-qr-code-photobombs/?sh=62bc8ef57e49, 2013.
46. M. Guri, O. Hasson, G. Kedma, and Y. Elovici. VisiSploit: An optical covert-channel to leak data through an air-gap. arXiv:1607.03946, 2016.
47. M. Guri, A. Kachlon, O. Hasson, G. Kedma, Y. Mirsky, and Y. Elovici. GSMem: Data exfiltration from air-gapped computers over GSM frequencies. In *SEC'15: Proceedings of the 24th USENIX Conference on Security Symposium*, pages 849–864, Berkeley, CA, 2015. USENIX Association.
48. M. Guri, G. Kedma, A. Kachlon, and Y. Elovici. AirHopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies. In *2014 9th International Conference on Malicious and Unwanted Software: The Americas (MALWARE)*, pages 58–67. IEEE, 2014.
49. M. Guri, M. Monitz, Y. Mirski, and Y. Elovici. BitWhisper: Covert signaling channel between air-gapped computers using thermal manipulations. In *2015 IEEE 28th Computer Security Foundations Symposium*, pages 276–289. IEEE, 2015.
50. M. Guri, Y. Solewicz, A. Daidakulov, and Y. Elovici. Fansmitter: Acoustic data exfiltration from (speakerless) air-gapped computers. arXiv:1606.05915, 2016.
51. M. Hanspach and M. Goetz. On covert acoustical mesh networks in air. arXiv:1406.1213, 2014.
52. C. Heffner. Exploiting surveillance cameras: Like a Hollywood hacker. Presentation at Black Hat USA 2013, 2013.
53. K. Hill. 'baby monitor hack' could happen to 40,000 other Foscam users. Forbes, http://goo.gl/2cdYy0, 2013.
54. K. Hill. How a creep hacked a baby monitor to say lewd things to a 2-year-old. Forbes, http://goo.gl/92yg9G, 2013.

55. J. Hirsch and K. Bensinger. Toyota settles acceleration lawsuit after $3-million verdict. Los Angeles Times, https://www.latimes.com/business/la-xpm-2013-oct-25-la-fi-hy-toyota-damages-20131026-story.html, 2013.
56. IHS: 245 million surveillance cameras installed globally in 2014. SecurityInfoWatch.com, https://www.securityinfowatch.com/video-surveillance/news/12082966/245-million-surveillance-cameras-installed-globally-in-2014-ihs-says, 2015.
57. Insecam: Live cameras directory. http://insecam.org.
58. Internet Census 2012: Port scanning /0 using insecure embedded devices. http://census2012.sourceforge.net/paper.html, 2012.
59. IP cameras. CVE Details, http://goo.gl/ObpWCg.
60. IP cameras. InsecureOrg, http://insecure.org/search.html?q=IP%20camera.
61. Israeli road control system hacked, caused traffic jam on Haifa highway. The Hacker News, https://thehackernews.com/2013/10/israeli-road-control-system-hacked.html, 2013.
62. A. Kharraz, E. Kirda, W. Robertson, D. Balzarotti, and A. Francillon. Optical delusions: A study of malicious QR codes in the wild. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 192–203. IEEE, 2014.
63. A. Kidman. How a prison had its CCTV hacked. http://goo.gl/sKombD, 2012.
64. KillerBee. Google, http://code.google.com/p/killerbee/.
65. K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage. Experimental security analysis of a modern automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462. IEEE, 2010.
66. D. Kriesel. Xerox scanners/photocopiers randomly alter numbers in scanned documents. https://www.dkriesel.com/en/blog/2013/0802_xerox-workcentres_are_switching_written_numbers_when_scanning, 2014.
67. M.G. Kuhn and R.J. Anderson. Soft tempest: Hidden data transmission using electromagnetic emanations. In *International Workshop on Information Hiding*, pages 124–142, Berlin, 1998. Springer.
68. R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.
69. J. Loughry and D.A. Umphress. Information leakage from optical emanations. *ACM Transactions on Information and System Security*, 5(3):262–289, 2002.
70. J. Marpet. Physical security in a networked world: Video analytics, video surveillance, and you. Presentation at Black Hat DC 2010, 2010.
71. J. McMath. Automated Dependent Surveillance-broadcast Military (ADS-M). US Air Force Slides, 2007.
72. Y. Mirsky, M. Guri, and Y. Elovici. HVACKer: Bridging the air-gap by manipulating the environment temperature. DeepSec, https://deepsec.net/docs/Slides/2015/Bridging_the_Air-Gap_Data_Exfiltration_from_Air-Gap_%20Networks_-_Yisroel_Mirsky.pdf, 2015.
73. H.D. Moore. Ray sharp CCTV DVR password retrieval & remote root. Rapid7, http://goo.gl/Hnp3TO, 2013.
74. T. Morkel, J.H.P. Eloff, and M.S. Olivier. An overview of image steganography. In *Proceedings of the Fifth Annual Information Security South Africa Conference (ISSA2005)*, 2005.
75. K. Mowery, E. Wustrow, T. Wypych, C. Singleton, C. Comfort, E. Rescorla, S. Checkoway, J.A. Halderman, and H. Shacham. Security analysis of a full-body scanner. In *SEC'14: Proceedings of the 23rd USENIX Conference on Security Symposium*, pages 369–384, Berkeley, CA, 2014. USENIX Association.
76. NATO – STANAG 4193 PT I: Technical characteristics of the IFF Mk XIIA system. https://standards.globalspec.com/std/14346734/STANAG%204193%20PT%20I, 2016.
77. J. Obermaier and M. Hutle. Analyzing the security and privacy of cloud-based video surveillance systems. In *IoTPTS'16: Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, pages 22–28, New York, 2016. ACM.

78. M. Olson. Beware, even things on Amazon come with embedded malware... http://artfulhacker.com/post/142519805054/beware-even-things-on-amazon-come, April 2016. Accessed: July 25, 2016.
79. S. O'Malley and K.K.R. Choo. Bridging the air gap: Inaudible data exfiltration by insiders. In *Proceedings of the 20th Americas Conference on Information Systems (AMCIS 2014)*, 2014.
80. Owning a cop car. Digitalmunition, http://www.digitalmunition.com/OwningCopCar.pdf.
81. Owning big brother (or how to crack into Axis IP cameras). Purple paper, ProCheckUp, London. https://www.procheckup.com/media/1k0fv4mf/vulnerability_axis_2100_research.pdf.
82. N. Provos and P. Honeyman. Hide and seek: An introduction to steganography. *IEEE Security & Privacy*, 1(3):32–44, 2003.
83. V. Puri, S. Mahendru, R. Rana, and M. Deshpande. Firework injuries: A ten-year study. *Journal of Plastic, Reconstructive & Aesthetic Surgery*, 62(9):1103–1111, 2009.
84. C. Quintin and D. Maass. License plate readers exposed! How public safety agencies responded to major vulnerabilities in vehicle surveillance tech. Electronic Frontier Foundation (EFF), https://www.eff.org/deeplinks/2015/10/license-plate-readers-exposed-how-public-safety-agencies-responded-massive, 2015.
85. (Radar Gadgets) Planegadget Radar. RadioPics, http://www.radiopics.com/Flight%20(Air%20Band)/SS-Radar/Planegadget/Planegadget_Radar.htm.
86. J. Radcliffe. Hacking medical devices for fun and insulin: Breaking the human SCADA system. Presentation at Black Hat USA 2011, 2011.
87. T. Rondeau. Re: [Discuss-gnuradio] A chunks to symbols related question. Free Software Foundation (FSF), https://lists.gnu.org/archive/html/discuss-gnuradio/2012-01/msg00144.html.
88. E. Ronen and A. Shamir. Extended functionality attacks on IoT devices: The case of smart lights. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 3–12. IEEE, 2016.
89. RTCA DO-242A: Minimum aviation system performance standards for Automatic Dependent Surveillance Broadcast (ADS-B). RTCA, 2002.
90. RTCA DO-249: Development and implementation planning guide for Automatic Dependent Surveillance Broadcast (ADS-B) applications. RTCA, 1999.
91. RTCA DO-260A: Minimum operational performance standards for 1090 MHz Automatic Dependent Surveillance-Broadcast (ADS-B) and Traffic Information Services (TIS-B). RTCA, 2003.
92. RTCA DO-263: Application of airborne conflict management: Detection, prevention, & resolution. RTCA, 2000.
93. RTCA DO-282B: Minimum operational performance standards for Universal Access Transceiver (UAT) Automatic Dependent Surveillance-Broadcast (ADS-B). RTCA, 2009.
94. SBX 400-4400 MHz Rx/Tx (40 MHz). Ettus Research, https://www.ettus.com/all-products/SBX/.
95. V. Sepetnitsky, M. Guri, and Y. Elovici. Exfiltration of information from air-gapped machines using monitor's LED indicator. In *2014 IEEE Joint Intelligence and Security Informatics Conference*, pages 264–267. IEEE, 2014.
96. S. Shekyan and A. Harutyunyan. To watch or to be watched: Turning your surveillance camera against you. Presentation at HITB Security Conference 2013, Amsterdam, 2013.
97. Shodan images. https://images.shodan.io/.
98. Shodan search engine. http://www.shodan.io.
99. J. Sun. *The 1090 megahertz riddle: A guide to decoding Mode S and ADS-B signals*. TU Delft OPEN Publishing, 2nd edition, 2020.
100. Synapse module comparison chart. Solarbotics, https://solarbotics.com/wp-content/uploads/synapse_comparison_table.pdf.
101. Synapse Wireless. *Portal: Reference manual for version 2.6.6*. http://help.synapse-wireless.com/Portal/Portal-Reference-Manual.pdf.

102. Synapse Wireless. *SNAP network operating system: Reference manual for version 2.4*. https://cdn.sparkfun.com/datasheets/Wireless/General/SNAP%20Reference%20Manual.pdf.
103. Synapse Wireless. *SNAP stick user guide*, 2011. https://usermanual.wiki/Synapse-Wireless/SS200/html.
104. TelosB. Crossbow, https://www.willow.co.uk/TelosB_Datasheet.pdf.
105. The fastest robbery – 1 min in bank. YouTube, http://youtu.be/LFArxqcP4MI, 2012.
106. TRENDnet Exposed. https://twitter.com/trendnetexposed, This account doesn't exist!
107. H. Turtiainen, A. Costin, T. Hämäläinen, and T. Lahtinen. Towards large-scale, automated, accurate detection of CCTV camera objects using computer vision: Applications and implications for privacy, safety, and cybersecurity. arXiv:2006.03870, 2020.
108. Ukraine protests: Kiev fireworks 'rain on police'. BBC News, https://www.bbc.com/news/world-europe-25820899.
109. J.B. Ullrich. This is why your DVR attacked my synology disk station (and now with bitcoin miner!). SANS ISC InfoSec Forums, https://isc.sans.edu/forums/diary/More+Device+Malware+This+is+why+your+DVR+attacked+my+Synology+Disk+Station+and+now+with+Bitcoin+Miner/17879, 2014.
110. USRP1 (Universal Software Radio Peripheral). Ettus Research, https://www.ettus.com/products/.
111. M. van Berkum. ABUS TVIP 11550/21550 multiple vulnerabilities (and possibly other ABUS cams). SecurityFocus, http://www.securityfocus.com/archive/1/520045, 2011.
112. M. Vuagnoux and S. Pasini. Compromising electromagnetic emanations of wired and wireless keyboards. In *SSYM'09: Proceedings of the 18th Conference on USENIX Security Symposium*, pages 1–16, Berkeley, CA, 2009. USENIX Association.
113. C. Walsh. Police body cameras infected with conficker worm. https://www.carmelowalsh.com/tag/martel/, 2015.
114. C. Welch. FTC settles with Trendnet after 'hundreds' of home security cameras were hacked. http://goo.gl/94Ibmv, 2013.
115. J. Zaddach and A. Costin. Embedded devices security and firmware reverse engineering. Presentation at Black Hat USA 2013, 2013.
116. I. Zeifman, O. Gayer, and O. Wilder. CCTV DDoS botnet in our own back yard. Imperva, Inc., https://www.incapsula.com/blog/cctv-ddos-botnet-back-yard.html.
117. K. Zetter. Crooks spy on casino card games with hacked security cameras, win $33m. Wired, http://goo.gl/zmxVXe, 2013.
118. Zigduino r2. Logos Electromechanical, https://www.logos-electro.com/zigduino/.