Ready, Aim Fire: An Introduction to the Security of Fire Alarm Systems from the Perspective of a Penetration Tester

René Bisperink
Kiwa Fire Safety and Security Products, Apeldoorn, The Netherlands.

Abstract

This paper serves as an introduction to targeted attacks and the state of security around fire detection systems from the perspective of a penetration tester. Fire detection alarm systems are increasingly connected to the internet, which introduces new risks. Furthermore, the attack surface is shifting from physical access, or being in the surrounding area, to remotely connected devices.

Keywords: Fire alarm security, fire alarm vulnerabilities.

Introduction

One of the biggest problems with fire alarm systems and the industrial cyber security is that it spans two domains of specialized knowledge: Information Technology (IT) and Operational Technology (OT). There are two separate perspectives, two separate lifetimes of experience, that have to collaborate to secure these systems.

While the security focus with IT systems is mainly on the Integrity and availability of data, Industrial Control Systems, on the other hand, strive for the efficiency and reliability of a single, often fine-tuned system, while always addressing the safety of the personnel, plant, and environment in which they operate. These systems focus more on the availability of the system. Because of new technologies, the market trend is to increasingly connect the devices (mostly the control panel, often called Control and Indicating Equipment) to the internet. With the introduction of remote management, new risks are introduced to these systems.

The control panel is the main gateway from the sensors to the network of the asset owner and possibly to the internet. These control panels are often based upon embedded systems, running the manufacturers firmware. Devices that run firmware are known as embedded systems which often have limited hardware resources, such as storage capabilities as well as memory. Firmware is a kind of software that is written to a hardware device in order to control user applications and various system functions. The firmware contains low level programming code that enables software to access hardware functions.

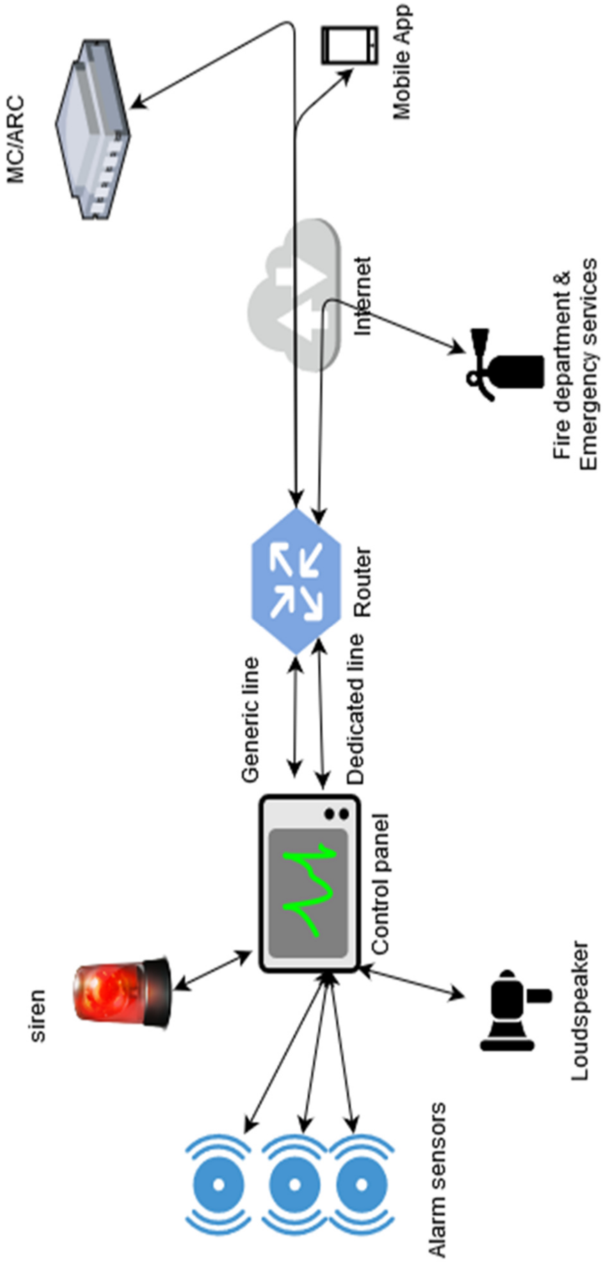

Fig. 1. Generic overview fire alarm system.

Fire alarm systems are designed to detect the presence of fire by monitoring environmental changes associated with combustion, with the help of sensors. Automated fire alarm systems are intended to notify the building occupants to evacuate in the event of a fire or other emergencies related to such an occurrence. Furthermore, automatic systems also report the event to an off-site location to summon emergency services, and to monitor and control the systems that are keeping the spread of fire and smoke in control. In the figure above, the Monitoring Center / Alarm Receiving center will receive a notification in case of an emergency.

These fire alarm systems are now also being integrated in Building Automation Systems (BAS). Along with advances in control systems, BAS have evolved into a gigantic system. It not only handles HVAC, lighting, air humidity, but also manages building security and safety subsystems by controlling and monitoring fire and flood safety, CCTV, elevator, power supply, and part of the process for room access authentication. Often, BAS provides a communication backbone which serves as an infrastructure that provides rules, policies, and integration medium for different subsystems. With the advancements in Internet of Things (IoT) devices, Fire Detection and Monitoring systems are in the process of becoming more intelligent by merging computing resources and network communication with physical control. Along with potential benefits, it also brings tremendous risks of security breaches and safety violations, especially when it comes to the logic controllers from (fire) alarm systems. (Wang, et al., 2015)

Intelligence Gathering

Fire alarm systems may be comprised of similar components: however, each system is unique to the terms of the exact composition, quantity, and criticality of these components. While these fire alarm systems are not the most common targets within industrial networks despite these similarities in components with other industrial devices, they are becoming more of a target in recent years.

Intelligence Gathering is performing reconnaissance against a target to gather as much information as possible to be utilized when testing the target during the vulnerability assessment and exploitation phases. The more information an attacker or tester is able to gather during this phase, the more vectors of attack they may be able to use in the future.

Open source intelligence (OSINT) is a form of intelligence collection management that involves finding, selecting, and acquiring information from publicly available sources and analysing it to produce actionable intelligence. This information can help with the threat model and the exploitation phase later on. Every test on a fire alarm system has an end goal in mind - a particular asset or process that the organization considers critical. Having the result in mind, the intelligence gathering phase should make sure to include all secondary and tertiary elements surrounding the end goal. Be it supporting technologies, 3rd parties, relevant personnel, etc... Making sure the focus is kept on the critical assets assures that lesser relevant intelligence elements are de-prioritized and categorized as such in order to not intervene with the analysis process. (pentest-standard.org; Various Authors, 2014).

Threat modelling

From the gathered intelligence, an attacker (or tester in collaboration with the organization) can set up a threat model. The threat modelling process provides clarity as far as the organization's risk appetite and prioritization (which assets are more important than others? what threat communities are more relevant than others?).

The threat model should be constructed in coordination with the organization being tested whenever possible, and even in a complete black-box situation where the tester does not have any prior information on the organization, the tester should create a threat model based on the attacker's view in combination with OSINT related to the target organization. The threat model and gathered information lead to possible attack vectors that someone could misuse. Examples of attack vectors on fire alarm systems are:

- Fire alarm / evaluation
- Fire suppressant system
- Building management network
- Software vendor support portal

Possible attack methods

- Exploitation of unpatched application (building management system)
- Installation of malware via unvalidated vendor software
- Network access through unprotected access points
- Network pivoting through unregulated network boundaries

Consequences of such an attack on a fire alarm system

- Delay, block, or alter the intended process or the information related to that process.
- Unauthorized changes to instructions or fire alarm thresholds that could damage, disable or shutdown mechanical equipment, such as sensors.
- Inaccurate information sent to operators could either be used to disguise unauthorized changes, or cause the operator to initiate inappropriate actions.
- Unauthorized release of suppressant.

Most industrial networks including those to which fire alarm systems are connected, employ automated safety systems to avoid catastrophic failures. However, many of these safety controls employ the same messaging and control protocols used by the industrial control network's operational processes, and in some cases, such as certain fieldbus or serial protocol implementations, the fire safety systems are supported directly within the same communications protocols as the operational controls.

Attacking versus exploiting a fire alarm system

It is important to understand at this point the difference between compromising or "owning" a target and attacking a target. There is no formal definition that defines either, but for the purposes of this paper a compromise can be thought of as the ability to exploit a target and perform an unknown action (such as running a malicious payload). An attack, on the other hand, can be thought of as causing a target to perform an undesirable action. In this case, the device may be performing as designed, yet the ability to attack the device and cause it to perform an action that is not desired by the engineer may lead to negative consequences. Many industrial devices can therefore be attacked via the exploitation of functionality versus the exploitation of vulnerabilities.

In other words, issuing a "shutdown" command to a control device does not represent any weakness in the device per se. However, if the lack of authentication enables a malicious user to inject a shutdown command (i.e. perform a replay attack), this is a major vulnerability.

Attacking (or Exploiting) fire alarm systems

There are many methods of attacking a target, once a target has been identified. This is often based on the highest risk from the threat model or a high chance of success based on a vulnerability assessment. Man-in-the-Middle (MitM), Denial-of-Service (DoS), Replay attacks, and countless more methods all remain very effective in industrial networks.

The primary reason for this is a combination of insecure communication protocols, little device-to-device authentication, and delicate communication stacks in embedded devices. If an industrial network can be penetrated and malware deposited (on disk or in memory) anywhere on the network, tools such as a Metasploit Meterpreter shell can be used to provide remote access to target systems, install keyloggers or keystroke injectors, enable local audio/video resources, manipulate control bits within industrial protocols, plus many other covert capabilities.

Man-in-the-middle

A man-in-the-middle attack refers to an attack where the attacker goes between communicating devices and snoops the traffic between them. The attacker is actually connecting to both devices, and then relaying traffic between them so that it appears that they are communicating directly, even though they are really communicating through a third device that is eavesdropping on the interaction.

To perform a MitM attack, the attacker must be able to intercept traffic between the two target systems and inject new traffic. If the connection lacks encryption and authentication—as is often the case with industrial protocol traffic—this is a very straightforward process. Where authentication or encryption are used, an MitM attack can still succeed by listening for key exchanges and passing the attacker's key in place of a legitimate key. This attack vector is somewhat complicated in industrial networks because devices can communicate via sessions that are established and remain intact for long periods of time. The attacker would have to first hijack an existing communication session.

The biggest challenge to a successful MitM attack is successfully inserting oneself into the message stream, which requires establishing trust. In other words, the attacker needs to convince both sides of the connection that it is the intended recipient. This impersonation can be thwarted with appropriate authentication controls. Many industrial protocols unfortunately authenticate in clear text (if at all), facilitating MitM attacks within the various industrial control systems. Man-in-the-middle attacks can be made more difficult by using renomated encryption and good authentication, authorization and session management.

Denial of Service

Denial-of-service attacks occur when some malicious event attempts to make a resource unavailable. It is a very broad category of attacks and can include anything from loss of communications with the device to inhibiting or crashing particular services within the device (storage, I/O, processing, continuous logic processing, etc.). While the illegitimate removal of a fire alarm system can also be described as a denial of service attack, most definitions only go into the network and software aspects of it. Denial of service attacks cannot be prevented completely, although having countermeasures in place which lower the risk help mitigating these attacks when they happen.

Replay

A big concern in industrial environments is the replay attack in which an attacker captures some type of data and resubmits it with the hopes of fooling the receiving device into thinking it is legitimate information. Many times, the data captured and resubmitted is authentication information and the attacker is trying to authenticate themselves as someone else to gain unauthorized access. Preventing such an attack is all about having the right method of encryption. To counter this possibility, both sender and receiver should establish a completely random session key, which is a type of code that is only valid for one transaction and can't be used again.

Another method to avoid becoming a victim is to have a password for each transaction that's only used once and discarded. That ensures that even if the message is recorded and resent by an attacker, the encryption code has expired and no longer works.

Closing thoughts on the vulnerability of these systems

A significant number of off-the-shelf products in the market are currently based on outdated technologies, which have limited security features, and unpredictable vulnerabilities due to their backward-compatible designs and focus on ease of maintenance. (Carey & Bathurst, 2013), (Tierney, 2017). Fire alarm systems widely use outdated low-level protocols, such as an obscured serial connection (Local Security Networks (LSN)), which sends data in plaintext and lacks proper authentication mechanisms. Works such as (Bolshev & icscorsair, 2014) and (Molina, 2014) show that attackers can easily sniff control packets, modify the Programmable Logic Controllers (PLC) arbitrarily, use carefully crafted low-level data gathered through PLCs to inject high-level control software. Secondly, if hardware flaws are found, it is very difficult, and sometimes impossible to patch them with software.

The system itself

It is important to understand what a firmware binary consists of as well as its associated properties. Firmware is comprised of a bootloader, kernel, filesystem, and various other resources. The bootloader is responsible for initializing the RAM for the volatile data storage, initializing the devices in the loop (e.g. the Local Security Network), serial ports, detecting the machine types and more. The kernel is has control over everything in the system. It facilitates interactions between the hardware and software components. There are different types of firmware, but the most common ones are built upon embedded Linux, embedded Windows, Windows IoT core, and various Real Time Operating Systems (RTOS).

Attacking the firmware

On the firmware side of the device, it becomes more and more of the reverse engineering of the firmware of such a fire detection system and checking it for possible bugs. This is a more tedious and manual process. Firmware can often be downloaded from the manufacturer’s website. More often than not, it is a protected file that could be cracked with the right tooling and a lot of time and expertise. Afterwards the file system can be researched, and the workings of the system will be known to the security researcher. Analysing and understanding a filesystem and its internal contents is all about the manual assessment skills of the tester. This is how vulnerabilities are identified.

To analyse firmware filesystem contents on a deeper level, techniques such as firmware diffing can be applied, with which you could compare one firmware with its previous version and look at the differences. This would enable you to understand the security fixes and modifications which have been made in the new version and identify even undisclosed security issues in the previous ones. This can be done based on researching outdated components and libraries or finding new previously unknown vulnerabilities in the software, that can be attacked or exploited. By applying obfuscation methods, and encrypting the firmware download, it will become much more difficult for an attacker to do research on the workings of the device, by reverse engineering the binary.

Lessons learned from previous (fire) alarm system attacks

In the past, certain intrusion and holdup alarm system brands gave you superuser access (highest permissions) when you connected directly to their serial interfaces. It was obfuscated by using voltage deviations compared to the normal interface standards, but still within the range of what the specific serial protocol allowed. (Babak & Howell, 2012).

Throughout the years, tamper detection has increased so the direct connection (with physical access) is more and more difficult to do as a penetration tester or a malicious party. This means that the focus will be more and more on the rest of the eco-system at the site of the asset owner or the data in transmission between the fire departments and or the remote connection. The end goal for many attackers is to get code execution on the device (or a root shell) in order to have the highest permissions possible, but other attacks are also certainly interesting for an attacker.

References and resources

[1] Babak, J., & Howell, K. (2012, September 27-30). *4140 Ways your alarm system can fai*. Retrieved from Derbycon: https://www.youtube.com/watch?v=g4-B7d3ZQUA

[2] Bolshev, G. A., & icscorsair. (2014, Autust). *How i will pwn your erp through 4-20 ma current loop*. Retrieved from Black Hat USA.

[3] Carey, M., & Bathurst, R. (2013, August` 1-4). *Doing Bad things to "Good" Security Appliances*. Retrieved from Defcon: https://www.youtube.com/watch?v=XyuwsJJzMDk

[4] Harris, S., & Maymi, F. (2018). *CISSP Exam Guide.* McGraw-Hill Education.

[5] Knapp, E. D., & Langill, J. T. (2015). *Industrial Network security: Securing Critical Infrastructure Networks for Smart Grid, SCADA and other Industrial Control Systems.* Waltham: Elsevier.

[6] Molina, J. (2014). *Learn how to control eery room at a luxury hotel remotely; The dangers of insecure home automation deployment*. Retrieved from Black Hat USA.

[7] pentest-standard.org; Various Authors. (2014, Augustus 16). *High Level Organization of the Standard*. Retrieved from pentest-standard.org: http://www.pentest-standard.org/index.php/Main_Page

[8] Tierney, A. (2017, 06 05). *Hacking Wireless house alarms*. Retrieved from pentestparners: https://www.pentestpartners.com/security-blog/hacking-wireless-house-alarms/

[9] Wang, X., Mizuno, M., Neilsen, M., Ou, X., Rajagopalan, S. R., Baldwin, W. G., & Philips, B. (2015). *Secure RTOS Architecture for Building Automation.*