Security of CCTV and Video Surveillance Systems: Threats, Vulnerabilities, Attacks, and Mitigations

Andrei Costin
andrei@firmware.re
Firmware.RE
Sophia-Antipolis, France

ABSTRACT

Video surveillance, closed-circuit TV and IP-camera systems became virtually omnipresent and indispensable for many organizations, businesses, and users. Their main purpose is to provide physical security, increase safety, and prevent crime. They also became increasingly complex, comprising many communication means, embedded hardware and non-trivial firmware. However, most research to date focused mainly on the privacy aspects of such systems, and did not fully address their issues related to cyber-security in general, and visual layer (i.e., imagery semantics) attacks in particular.

In this paper, we conduct a systematic review of existing and novel threats in video surveillance, closed-circuit TV and IP-camera systems based on publicly available data. The insights can then be used to better understand and identify the security and the privacy risks associated with the development, deployment and use of these systems. We study existing and novel threats, along with their existing or possible countermeasures, and summarize this knowledge into a comprehensive table that can be used in a practical way as a security checklist when assessing cyber-security level of existing or new CCTV designs and deployments. We also provide a set of recommendations and mitigations that can help improve the security and privacy levels provided by the hardware, the firmware, the network communications and the operation of video surveillance systems. We hope the findings in this paper will provide a valuable knowledge of the threat landscape that such systems are exposed to, as well as promote further research and widen the scope of this field beyond its current boundaries.

1. INTRODUCTION

Video surveillance, Closed-Circuit TV (CCTV), Digital or Network Video Recorder (DVR/NVR), and IP-camera (IPcam) systems [1] became extremely common all around the world. At present, VSSs are fundamental for most, if not all, life areas of the modern society. Their use is extremely wide, ranging from law enforcement and crime prevention, to transport safety and traffic monitoring, to industrial processes oversight and control of retail, to unauthorized [51, 64], illegal [115] and even criminal use [66]. And their number is incredibly large, some reports estimating it to be in the range of 245 million cameras/systems [63].

Generally, most of the concerns about video surveillance systems are related to privacy issues for obvious reasons. The privacy impact of VSSs is especially important in the light of revelations about global surveillance programs [4], and video surveillance scandals in particular [22]. However, besides privacy issues, an insecure or compromised VSS can raise a myriad of other non-privacy related issues. For example, their breach was shown to endanger the security and safety of a prison [66], pose theft risks to institutions operating with money such as banks [28] and casinos [115], emotionally affect other persons (especially children) [17], or interfere with police and law-enforcement [47]. At the same time, as more and more embedded devices are being analyzed at large scale for security vulnerabilities [40, 41], it is no surprise that VSSs have recently gained a dramatic increase of attention from security researchers [96, 77, 103, 59, 39, 114]. Those and similar studies led to more than a handful of vulnerabilities [2] with large-scale impact in real life [20, 12]. The variety of vendors and vulnerabilities disclosed in those studies and security advisories clearly indicates the unhealthy state of cyber-security of video surveillance systems.

In this paper, we conduct a systematic review of the existing threats and vulnerabilities in video surveillance, closed-circuit TV and IP-camera systems based on publicly available data. In addition to this, we review main threats and attacks taxonomies for video surveillance systems and for embedded devices. We also provide a set of recommendations and mitigations that can help improve the security and privacy levels provided by the hardware, the firmware, the network communications and the operation of video surveillance systems.

Our main contributions are:

- To the best of our knowledge, we present the first comprehensive and cross-disciplinary study of attacks and mitigations specific to VSS and CCTV systems.
- We discuss in-depth novel and specific attacks on VSS and CCTV systems.
- We present one novel covert channel specific to CCTV cameras (namely mechanical movement and position), and extend several existing covert channels to take advantage of specifics of VSS and CCTV systems.

[1] Throughout the rest of this paper we will refer to an instance of such a system as *video surveillance system* or *VSS*.

[2] While being long enough, this list is not meant to be exhaustive and is just a starting pointer for the interested readers [1, 2, 25, 23, 11, 9, 10, 14, 15, 13, 3, 85].

TrustED'16, October 28 2016, Vienna, Austria

ISBN 978-1-4503-4567-5/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2995289.2995290

2. RELATED WORK

On the one hand, researchers approached the security and the threat modeling of various parts of complex video surveillance systems. Kim and Han [67] developed a security model to ensure a safe and secure operation of an intelligent VSS. The model is represented by a set of particularly desired security functions which are grouped into several groups: video gathering group, video storage group, video control group, video application group. In their model, authors relate each identified security threat with particular security functions within defined groups. Lee and Wan [71] provide a high-level overview of security requirements for network CCTV in the context of *u-City* services. These requirements relate to the confidentiality, the integrity, the system protection and the content privacy. Park and Jun [94] summarize a subset of threats posed to networked and IP based CCTV systems. Subsequently, they propose two enhanced security protocols for user registration and authentication in order to increase the security of such networked systems. Coole et. al [38] describe a subset of the security issues related to networked, and especially Wi-Fi based, surveillance devices. The authors also discuss the significance of vulnerability exploitation of such devices in the context of confidentiality, integrity, availability. They conclude with a framework for implementing controls to reduce risk associated with Wi-Fi based CCTV systems. Recently, Obermaier and Hutle [85] provided a practical analysis of the security and privacy of four major cloud-based video surveillance systems. They reverse-engineered the security implementation and discovered several vulnerabilities in every of the tested systems. The authors considered two attacker models, namely local network attacker and remote network attacker. They demonstrated how these attackers can exploit VSS vulnerabilities to blackmail users and companies by DoS attacks, by injecting forged video streams, and by eavesdropping private video data, even without physical access to the systems. Their main findings, however, relate to *classical weaknesses* such as fallback to unsecured function, proprietary security protocols, weak passwords, and insecure authentication.

On the other hand, various taxonomies for threats, attacks and vulnerabilities for embedded systems exist. They range from generic taxonomies [93] to domain-specific ones [116, 46], however no comprehensive domain-specific taxonomy for video surveillance systems exist. Generic taxonomies are usually preferred because they provide fundamental understanding of the problems and solutions. However, domain-specific taxonomies can prove themselves very useful because they can capture details (e.g., visual layer attacks in Section 3.2.1 or video sensor attacks in Table 1) that are missed or cannot be captured by the generic taxonomies [3]. In this regard, our work leans towards a domain-specific classification system for video surveillance systems. To the best of our knowledge, it is the first comprehensive and cross-disciplinary study of attacks and mitigations specific to VSS and CCTV systems.

3. REVIEW OF THREATS, VULNERABILITIES, ATTACKS AND MITIGATIONS

In this section, we identify and describe the criteria we used for classification and review. For this purpose, we analyzed publicly reported vulnerabilities and published exploits that are related to video surveillance systems. Also, we checked whitepapers and presentation from computer security conferences (e.g., BlackHat, DefCon, HITB), including academic whitepapers with practical focus. We also checked the Internet for blog-posts, media reports, and mail-lists that discuss threats, vulnerabilities, attacks, and malware for video surveillance systems.

[3]Not at least without losing their practical or generic applicability.

3.1 Classification Criteria

To create the classification, we have chosen 7 criteria to describe the threats, vulnerabilities, attacks and mitigations for video surveillance systems: (1) attack surface, (2) attack type, (3) attacker type, (4) directly affected component(s), (5) exploitation complexity, (6) mitigation, and (7) mitigation complexity.

Table 1 presents our classification of threats, vulnerabilities, attacks, and mitigations for video surveillance systems. This table can help understand better the attack surface and the attacker's techniques, mitigate specific threats and focus on securing specific components of the VSSs. Alternatively, during the deployment or operation of a specific VSS, it can be used as a checklist to validate if it fulfills the security requirements for that particular deployment. For example, if DoS attacks on the WiFi and RF links are unacceptable, the checklist can help identify that and check if the proposed mitigations (or similar ones) are in implemented into the VSS. Or, for example, if the VSS is required to withstand dazzling, then appropriate checks on light filters and compensation techniques in hardware/software could be made according to this table.

Also, our classification could be easily mapped to the generic taxonomy of attacks and vulnerabilities for embedded devices from Papp et. al [93]. One way to perform the mapping could be as follows: (a) attack type $\rightarrow$ vulnerability; (b) attack surface + attacker type $\rightarrow$ precondition; (c) directly affected component $\rightarrow$ target; (d) attack type + mitigation $\rightarrow$ attack method; (e) attack type $\rightarrow$ effect. However, in contrast to existing methodologies, in particular [93], our classification also provides comprehensive information on mitigation. Additionally, it provides indicative (but non-authoritative) complexity levels for both executing the attack and for implementing the mitigation. For example, this can be useful to prioritize resources and tasks when performing risk/threat assessment or responding to an active attack on VSSs.

3.2 Discussion on Specific Attacks

3.2.1 Visual Layer Attacks

Compared to other embedded systems, video surveillance systems have an additional level of abstraction, i.e., the *visual layer*. Therefore, it is possible to (ab)use this layer to carry out novel attacks on the video surveillance systems that take advantage of the imagery semantics and image recognition.

Costin [39] presented first such an attack on CCTV cameras as the *visual layer backdoors*, which was also implemented into a full body scanner as the *secret knock image* by Mowery et al. [80]. This attack is a multi-stage one and works at the visual layer as follows. At the first stage, the VSS is infected with a malicious component (e.g., hardware, firmware). In some scenarios, this can be achieved locally via a malicious firmware update over the USB port, and remotely via a command injection or a malicious firmware upgrade over the web interface. In other scenarios, the VSS or CCTV system could be sold through legitimate sales channel with the malware already pre-installed [61, 86]. At the second stage, the malicious component is triggered and controlled via *malicious imagery inputs* when such imagery is "visualized" by the cameras and the video sensors. In the most general case, the trigger and command can be coded in any arbitrary data-to-image encoding scheme [4]. In one example, the malicious component could be designed to constantly blur pre-programmed faces or car plate reg-

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

istration numbers of attackers, or to disable certain functionality in the surveillance system (e.g., video recording functionality, or detection of a prohibited item such as a gun during a full body scan [80]). As a result, this type of malicious functionality could be used in theft and other criminal activities. In another example, it could read QR-like codes and interpret them as different commands. The *malicious imagery* [65] could be printed on t-shirts, cars or any accessory visible enough to the cameras. The commands could range from "stop recording" and "blur attacker face wearing malicious imagery or QR-code" to "contact command and control center" and "update malicious components". A variation of this attack was demonstrated in a Google Glass hack [16]. It used a specially crafted QR-code as malicious image input to control the Google Glass (in an unauthorized and unattended way) and force it to visit a malicious URL. To further complicate the detection by human operators and hide its payload, the visual layer attacks could use optical covert channel techniques. These attacks could use the camera sensitivity to infra-red and near-infra-red spectrum to send "invisible" information. Also, these attacks could use techniques similar to `VisiSploit` by Guri et al. [53], except that such a channel would be used to inject data and commands, rather than exfiltrate the data.

Finally, visual layer attacks are in fact not far-fetched. Since visual layer information of any kind will be processed at a certain point (e.g., image compression, face recognition, Optical Character Recognition (OCR)), this opens up opportunities for both intended and unintended errors. One example of such unintended error is the infamous example of Xerox scanners and photocopiers that were randomly altering numbers and data in documents [68]. Given the incredible processing complexity built into modern video surveillance systems (e.g., image compression, face recognition, Automatic License Plate Reading (ALPR)), it is reasonable to assume that similar problems (both intended and unintended) at the visual processing layer can affect or attack modern VSSs as well.

Solutions. A solution to *detect* such attacks could be tainting [82, 101] of video frames. Subsequent exploration of control and data flow graphs for both kernel and user space processes could detect "suspicious" code which tries to process video frames (e.g., blur, send them over standard or covert communication channels). At the same time, if such an attack would be implemented in hardware it could be orders of magnitude harder and costlier to defend against or detect as is generally the case with hardware-based backdoors [105]. Also, it could be very challenging to implement such a detection at runtime and most likely it would have to be performed during compliance tests and product certifications. Yet another solution to detect such attacks could be the use of performance counters [45, 112], since the malware performing a visual layer attack (i.e., image processing) would introduce an additional noticeable performance penalty.

A solution to *prevent* visual layer attacks by malware is to allow to VSS tamper with the image detection and recognition of the malware. One way to achieve such tampering is by introducing random pixel noise (simplest solution), or key-based pixel noise [5] (similar to direct-sequence spread-spectrum used in radio engineering). Another way is to use the recent advances in research of *adversarial images*, where a image is altered in particular ways to affect the detection and the classification of the image. In this context, image recognition systems have recently been shown to be vulnerable to simple attacks where slight modifications to only a handful of pixels can change the classification result dramatically [106, 52, 83, 75]. However, such a solution have drawbacks. First, the malware could use the same technique on the VSS and the raw video to prevent, for example, successful detection and recognition of human faces or vehicle plate numbers. Second, by tampering with the video feed without having a design that can offer *secured and private original feed*, the VSS could inadvertently tamper with important details in the video (e.g., criminal mugshot), and even render the whole video capture unauthentic and inadmissible as evidence [69]. We leave the implementation of these countermeasure solutions and their evaluation as future work.

Finally, to prevent both privacy and visual layer attacks by malware inside the VSS, a cryptographically-strong system could be used similar to the one by Castiglione et al. [36]. Such a system would guarantee lawfully secure and privacy preserving video feed by employing a hybrid cryptosystem based on a threshold multiparty key-sharing scheme.

3.2.2 Covert Channels Attacks

In the last several years, covert channels and data exfiltration (especially in air-gapped environments) was a subject of prolific research by (ab)using electromagnetic [70, 109, 55, 54], acoustic [92, 58, 57], thermal [56, 78], and optical [73, 102, 53, 100] channels. In the context of VSS and CCTV systems, we identify one novel covert channel and extend the application of several existing covert channels. Though the channels we present can be mainly used to exfiltrate data by means of a compromised VSS and CCTV component[61, 86], they can also be used for autonomous and distributed command-and-control purposes as explained below.

Normal LEDs.

Normal LEDs in the modern electronic equipment, such as those indicating various statuses of the equipment, have been repeatedly used in covert channels and data exfiltration [73, 37, 102]. Additionally, smart LED bulbs recently have been shown to pose similar threats [100]. Though sometimes the LEDs are physically linked to the hardware and cannot be controlled from software/firmware, recent attacks show that manipulating LEDs from software/firmware becomes increasingly practical and feasible [34]. The VSS and CCTV systems usually have plenty of status LEDs both on the core equipment as well as the CCTV cameras installed outside. Therefore the LEDs in VSS and CCTV systems could be used in data exfiltration attacks as well.

Infra-Red (IR) LEDs.

There is one major drawback of (ab)using normal LEDs in such attacks. If the LEDs are not manipulated in subtle ways (e.g., abnormal blinking frequencies, unusual luminosity levels) and are distinguishable to human eye, this can quickly and altogether compromise the covert channel. Therefore, we propose the use of the Infra-Red (IR) LEDs in optical covert channels. Arrays of IR LEDs are installed inside almost any modern CCTV camera. These IR LEDs are used for illumination and provide IR night-vision functionality to the cameras and VSSs. One important characteristic of IR LEDs is that when they operate, they are often invisible [6]. In order to see the operation of IR LEDs, one would have to use for example another camera which does not have IR cut-off filters (e.g., another CCTV IR-capable camera). Therefore, the IR-capable CCTV cameras can use the intensity of the IR LEDs (or their ON and OFF

[5]The derivation and agreement of the *key* that generates the noise patterns is beyond the scope of this paper.

[6]Almost always, but that depends on many characteristics of the IR LEDs used. For the purpose of this paper we assume it is hard, if not impossible, for a human eye to easily distinguish between normal and abnormal use of IR LEDs.

status) to modulate and exfiltrate data. Such exfiltration would be stealthy to human eye. One drawback of such an attack is that when the environment is dark and the cameras rely on IR LEDs to be ON, changing the intensity/status of IR LEDs would immediately be reflected in the camera capture and subsequently in the live displays of the operating personnel who might notice that something is wrong. When the environment is lighted, the changes in the IR LEDs would not be very visible in the camera capture and live displays, however the exfiltrated data can still be remotely captured by an attacker.

Finally, let us take this attack one step further and make the following assumptions:

- There is a *Collaborative Group (CG) of compromised CCTV cameras*, each of them having Line Of Sight (LOS) to at least one other CCTV camera in the CG. This is a reasonable assumption if we consider the present dense deployments of networks of cameras, and the rate and feasibility of compromising VSS and CCTV systems.
- All the cameras are IR-capable, i.e., have IR LEDs for illumination and besides the normal visible spectrum can also sense the IR and near-IR spectrum (i.e., do not have IR cut-off filters). This means such cameras could detect IR LEDs array patterns and intensity from other cameras or directly from the attacker. This is a reasonable assumption for today's CCTV camera standards.
- Some cameras in the CG could optionally have Pan-Tilt-Zoom (PTZ) functionality that would help them *optimally* focus on the IR LEDs and video sensors of other cameras in the CG. Having all or a part of CCTV cameras with PTZ functionality in such a CG is an optimistic assumption, but can become realistic very soon with the fast technological advance and the fast price drops within this extremely competitive market. However, though important, PTZ functionality is not essential to the essence of CG attack described.

Under these assumptions, the cameras in the CG could relay data and commands to each other using the IR spectrum. Therefore, this scenario could be used not just as data exfiltration, but also as an autonomous collaborative network of malicious CCTV cameras. Similar to *visual layer* attack, the attacker could then send command-and-control data to the CCTV cameras via the IR LEDs messaging (instead of coded visual images). Such a channel would constitute an addition to the classical (W)LAN and Internet channels used for communication and compromise.

VisiSploit Extension.

Recently, Guri et al. [53] presented `VisiSploit`, a new type of optical covert channel that exploits the limitations of human visual perception in order to unobtrusively leak data through a standard computer LCD display. Most of VSS and CCTV systems are connected to screens that are fully or partially visible to public. These screens present live image feeds from one or more cameras in the system. For example, this type of deployment is particularly popular in (super-)markets to deter shoplifting and to aid personnel in early detection of potential illegal or unethical activities. However, this type of deployment can also be seen in operational centers of large parkings, in reception lobbies of organizations (e.g., companies, hotels, elite residences), and many other places. Therefore, a compromised VSS and CCTV component could use screens in such deployments in combination with `VisiSploit` techniques to exfiltrate data.

Steganography.

Steganography is the art of hiding information within other information (e.g., images, documents, media streams or network protocols). Even though many different "carrier" media can be used for this purpose, digital images are the most popular because of their frequency on the Internet and their effectiveness in achieving steganography. A comprehensive overview of image steganography is presented in [97, 79].

A particular characteristic of VSS and CCTV systems is that virtually all systems provide both video and image streams [62]. The image streams can be either motion images (e.g., MJPEG) or still snapshots, and can be usually accessed at URLs such as http://CAM-IP/now.jpg, http://CAM-IP/shot.jpg or http://CAM-IP/img/snapshot.cgi?size=2 .

Therefore, a compromised VSS component (e.g., CCTV camera, DVR, NVR) can exfiltrate data by employing steganography when generating the image snapshots/streams mentioned above. Then, the attacker just would need to capture the digital image snapshots from well known URLs above and recover the exfiltrated data. Whether and how the attacker can access the image streams is beyond the purpose of this paper, but recent projects such as TRENDnet Exposed [27], Insecam [26], Shodan Images [104], corroborated with studies such as [43], demonstrate that it is very feasible and extremely easy to accomplish with the current cybersecurity practices within VSS and CCTV systems. To prevent data exfiltration involving steganography as presented above, automated methods for steganography detection could be applied [31, 49].

PTZ: Mechanical Movement and Position.

Many modern CCTV cameras have the so called Pan-Tilt-Zoom (PTZ) functionality. PTZ is a feature of CCTV cameras that allow them to move and remain fixed in almost any direction in 3D (e.g., using pan and tilt movements), and also zoom in and out by various zoom factors (e.g., using a high precission lense). Such functionality is usually enabled by stepper mottors built into specific camera models and is generally controlled by PTZ data protocols. The PTZ data protocols are sequences of bytes, composing commands and results, sent over a communication channel to control the pan, tilt and zoom. The PTZ commands and results are classically sent over RS-422 or RS-485 links, but can also be sent over classical Ethernet and WiFi channels. The PTZ commands can be sent to the PTZ-capable cameras from specialized PTZ-controls (e.g., special shortcuts keyboard with a joystick meant for surveillance room personnel) or from software (e.g., OS-speicifc thick clients or browser-based lightweight clients).

In this context, a compromised CCTV camera can exfiltrate data to the external attacker by encoding data into its position or movement changes. For example, it could change its normal fixed position to a specific fixed position that would encode a particular value. Let us assume that a compromised camera on a wall has the normal position "looking" *down-right*. To exfiltrate data, a compromised camera would then encode: bits `00` by moving itself to "look" *up-and-right*; bits `01` by moving itself to "look" *up-and-left*; bits `10` by moving itself to "look" *down-and-left*. To add more bits of data resolution (therefore increasing exfiltration data-rate), the number of such *abnormal* positions would be increased (pretty much like in Phase-Shift-Keying, or PSK, modulation) and would require an attacker a more precise observation of the compromised camera from the outside.

One drawback is that the attacker needs to identify in advance valid *abnormal* positions for each camera, as to not get confused with valid camera position during the data exfiltration. Another drawback is that suddenly changing camera position to a new one,

that could even not make any sense, could raise concerns of the surveillance personnel and compromise the covert channel. To compensate for this, the information to be exfiltrated could be encoded as *small deviations in PTZ movement* (e.g., slight changes in velocity of move, slight changes in the axis of move) compared to normal camera operation. By comarison, encoding data into movement of CCTV cameras (or parts of a system that allow movement) is comparable to *aircraft marshalling*, where well defined movements of the *marshaller* have well encoded messages to the *pilots* [7]. One drawback to this technique is that the attacker would need a quite sophisticated observation equipment to capture and measure the small and unobservable deviations to decode the exfiltrated data. Another drawback is that the attacker would need to have a baseline of normal move operation of each camera she plans to compromise, and this is untrivial.

Finally, let us assume a *Collaborative Group (CG) of compromised CCTV cameras* which are observable at the same time by the attacker. By combining and precisely coordinating the data they encode through PTZ, either via *abnormal* camera positions or via *small deviations in PTZ movement*, the data-rate of the exfiltration could be substantially increased. We leave the implementation and practical evaluation of such attacks as future work.

Audio Layer.

Many VSS and CCTV systems are audio-capable, meaning they can record and process one or more audio channels coming from external microphones or from microphones built into CCTV cameras. Therefore, a compromised VSS component (e.g., CCTV camera, DVR, NVR) can use the audio layer as a command-and-control channel, for example using *hidden voice commands* techniques [35].

3.2.3 Denial-of-Service and Jamming Attacks

We would like to emphasize on the importance of Denial-of-Service (DoS) and jamming attacks carried *onto* [8] video surveillance systems. In most cases, uninterrupted and untampered operation is critically important for video surveillance systems, for example because they are used to monitor and record crimes or other important activities. Producing a DoS attack on a CCTV systems even for 1 minute could make them miss an important event such as an extremely fast bank robbery [28, 113] or crimes with worse implications [66]. That is why, while a DoS attack on a home router could be a minor nuisance, the DoS attacks on video surveillance systems have critical impact and have to be taken into consideration during design, evaluation and testing [9].

4. ONLINE VIDEO SURVEILLANCE SYSTEMS

Some of the most useful and used features of a modern video surveillance system are the *plug-and-play* feature for easy installation and deployment, and the *remote access* features for management and video monitoring. As a result, many video surveillance systems end-up connected and exposed directly to the Internet [26], often having the default settings and credentials [43]. Therefore, we tried to estimate the number of Internet-facing video surveillance systems to be able to evaluate the magnitude of the potential exposure. For this purpose, we collected and compiled a large list of queries specific to video surveillance systems and then we ran them on both online services and existing Internet scanning databases. Using the Shodan [24] web-service, these queries revealed an incredible number of more than $2.2M$ video surveillance systems produced by more than 20 distinct vendors. Using the Internet Census 2012 database [19], these queries returned more than $400K$ VSS produced by more than 10 distinct vendors. At the same time, some reports [63] estimate there were nearly $245M$ video surveillance cameras installed globally in 2014. Unsurprisingly, the discovery, tracking and publication [10] of online video surveillance systems, that are vulnerable, compromised or lower the effective privacy of their owners, has always been a hot topic of interest and debate. Projects such as TRENDnet Exposed [27], Insecam [26], Shodan Images [104] and EFF ALPR [74] are several examples of such initiatives. As a result these projects received an incredible amount of media attention, public scrutiny and outrage, by raising once again the issue of lacking security and privacy in modern video surveillance systems.

Worse, according to Cui and Stolfo [43], 39.72% of cameras and surveillance systems they analyzed on the Internet in 2010 were running with default credentials. This basically means they are completely exposed to any kind of attacks such as video-feed eavesdropping [11], malicious firmware updates, DNS hijacking. As an additional example, we analyzed a set of firmware images for a DVR system and discovered a full admin backdoor. We then correlated identification information extracted from firmware images with the results of the queries we mentioned above. This resulted in more than $130K$ affected online accessible devices.

Even though some of these systems (i.e., their IP addresses) and vendors may overlap (or not be completely accurate accounted for), these results give a lower-bound estimate of the scale at which video surveillance systems are exposed and vulnerable to cybersecurity threats. Running Internet-wide queries and using vulnerability estimations from previous works [43], proved to be a very efficient method for estimating the number of potentially exposed and vulnerable video surveillance systems.

5. SOLUTIONS

Below we summarize a set of recommendations that we hope can help increase the security of the hardware, firmware and network communication of video surveillance systems. With increased security, we hope that a safer operation and an increased privacy of the entire VSS could be achieved.

Factory reset button.

Providing a *factory reset button* can help reset the system to a known *factory safe and secure* image and state from a non-volatile non-writable secure memory chip.

Secure scan chains.

Implementing *secure scan* techniques [60] may allow secure debugging, testing and restoring without the risk of unauthorized users to gain access to debug functionality.

[7] Example of information encoding schemes in aircraft marshalling are the North Atlantic Treaty Organization (NATO) Standardization Agreement 3117 and the Air Standardization Coordinating Committee Air Standard 44/42A

[8] In this case, the emphasis is on VSSs as the *final target* of the attack. In the cases when the VSSs are infected and used in botnets to carry out DDoS attacks *onto other systems* as final targets [51], those DDoS attacks are considered as attacks *from* video surveillance systems, where VSSs play the role of *originating source* of the attack.

[9] However, this in itself is non-trivial as thoroughly explained in [50].

[10] Many times along with their screenshots and video feeds.

[11] Practically demonstrated at large scale by projects such as TRENDnet Exposed [27], Insecam [26], and Shodan Images [104].

(Remote) attestation.

Implementing *(remote) software or device attestation* techniques would ensure, via static or dynamic root of trust, that critical code requiring safety and security is not tampered with. This can be achieved for example via minor firmware and hardware changes, as detailed in SMART [48].

Formal proof and verification.

Applying *formal proof and verification* techniques to hardware designs, firmware implementations, communication and security protocols. This can greatly improve safety and security of hardware, firmware and protocols.

Standards compliance.

Implementing *software and hardware security compliance* standards (similar to avionics field's *DO-254* and *DO-178B*) for hardware and software, respectively, can ensure stronger security for video surveillance systems; for example, at present very few VSSs implement standards such as *BS8418*, *BSEN50131-1* and *DD243:2004* (though these do not directly relate to cyber-security risks per-se as presented in Table 1).

Visual layer.

As discussed in Section 3.2.1, solutions for visual layer attacks are not trivial. However, implementing the above solutions correctly, such as (remote) attestation and secure firmware upgrade mechanisms, would eliminate the need for visual layer countermeasures as they are described in Section 3.2.1.

6. CONCLUSIONS

This paper provides a systematic review of security of video surveillance systems by describing in detail threats, vulnerabilities, attacks, and mitigations. Based on publicly available data and existing classifications and taxonomies, the review presented in this paper provides comprehensive information on how video surveillance systems can be attacked and protected at various levels. This structured knowledge can then be used to better understand and identify the security and privacy risks associated with the development, deployment and use of these systems. Moreover, this paper presented a set of recommendations and mitigations that can improve the security and privacy aspects of video surveillance systems.

Acknowledgements

The authors thank Prof. Aurélien Francillon for guidance and comments during early versions of this paper. The authors also thank and express their gratitude to Enno Rey and ERNW GmbH for their generous support that made it possible to present this paper and its results at *TrustED'16*.

7. REFERENCES

[1] ABUS TVIP 11550/21550 Multiple vulnerabilities. http://www.securityfocus.com/archive/1/520045.

[2] Anonymous authenticated access to MJPEG stream. http://goo.gl/sYkUAF.

[3] 'Baby Monitor Hack' Could Happen To 40,000 Other Foscam Users. http://goo.gl/2cdYy0.

[4] BuggedPlanet – Surveillance Industry and Country's Actings. http://buggedplanet.info/.

[5] CVE-2013-1391 – File disclosure in Hunt DVR and generic brands, discloses authentication information.

[6] CVE-2013-2560 – Directory traversal in the web interface on Foscam devices.

[7] CVE-2013-4981 – Denial-of-service in AVTECH AVN801 DVR.

[8] CVE-2013-6023 – Directory traversal in the TVT TD-2308SS-B DVR.

[9] CVE details – CCTV systems. http://goo.gl/IB1Hk7.

[10] CVE details – DVR systems. http://goo.gl/Xmv1jN.

[11] CVE details – IP cameras. http://goo.gl/ObpWCg.

[12] FTC settles with Trendnet after 'thousands' of home security cameras were hacked. http://goo.gl/94Ibmv.

[13] Full disclosure – CCTV systems. http://insecure.org/search.html?q=cctv.

[14] Full disclosure – DVR systems. http://insecure.org/search.html?q=dvr.

[15] Full disclosure – IP cameras. http://insecure.org/search.html?q=IP%20camera.

[16] Google Glass hacked by the image of a malicious QR code. http://goo.gl/Qqh72x.

[17] How A Creep Hacked A Baby Monitor To Say Lewd Things To A 2-Year-Old. http://goo.gl/92yg9G.

[18] How to ZAP a Camera: Using Lasers to Temporarily Neutralize Camera Sensors. http://www.naimark.net/projects/zap/howto.html.

[19] Internet Census 2012 – Port scanning /0 using insecure embedded devices. http://internetcensus2012.bitbucket.org.

[20] Israeli Road Control System hacked – malware to hit the security camera apparatus in the Carmel Tunnel toll. http://goo.gl/F5I0ou.

[21] Mal au Pixel # Festival – CCTV Sniffing Workshop. http://vimeo.com/57881594.

[22] Oakland Domain Awareness Center (DAC). http://oaklandwiki.org/Domain_Awareness_Center.

[23] Ray Sharp CCTV DVRs Password Retrieval. http://goo.gl/Hnp3TO.

[24] SHODAN – Computer Search Engine. http://www.shodan.io.

[25] Swann Song DVRs Insecurity. http://goo.gl/oY3z3w.

[26] Anonymous. Insecam Project – The world biggest directory of online (insecure) surveillance security cameras. http://insecam.org.

[27] Anonymous. TRENDnet Exposed. https://twitter.com/trendnetexposed.

[28] J. Aron. Want to rob a bank? Hack your way in. *New Scientist*, 220(2937):22, 2013.

[29] J. Bau, E. Bursztein, D. Gupta, and J. C. Mitchell. State of the Art: Automated Black-Box Web Application Vulnerability Testing. In *IEEE Symposium on Security and Privacy*, 2010.

[30] J. Bellardo and S. Savage. 802.11 denial-of-service attacks: Real vulnerabilities and practical solutions. In *Proceedings of the USENIX Security Symposium*, 2003.

[31] G. Berg, I. Davidson, M.-Y. Duan, and G. Paul. Searching for hidden messages: Automatic detection of steganography. In *IAAI*, pages 51–56, 2003.

[32] H. Bojinov, E. Bursztein, and D. Boneh. Xcs: Cross channel scripting and its impact on web applications. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pages 420–431, New York, NY, USA, 2009. ACM.

[33] H. Bojinov, E. Bursztein, E. Lovett, and D. Boneh. Embedded management interfaces: Emerging massive insecurity. *Blackhat USA*, July 2009.

[34] M. Brocker and S. Checkoway. iSeeYou: Disabling the MacBook webcam indicator LED. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 337–352, 2014.

[35] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou. Hidden Voice Commands. In *25th USENIX Security Symposium (USENIX Security 16), Austin, TX*, 2016.

[36] A. Castiglione, M. Cepparulo, A. De Santis, and F. Palmieri. Towards a lawfully secure and privacy preserving video surveillance system. In *International Conference on Electronic Commerce and Web Technologies*, pages 73–84. Springer, 2010.

[37] J. Clark, S. Leblanc, and S. Knight. Hardware trojan horse device based on unintended usb channels. In *Network and System Security, 2009. NSS'09. Third International Conference on*, pages 1–8. IEEE, 2009.

[38] M. Coole, A. Woodward, and C. Valli. Understanding the vulnerabilities in wi-fi and the impact on its use in cctv systems. 2012.

[39] A. Costin. Poor Man's Panopticon: Mass CCTV Surveillance for the masses. In *PowerOfCommunity*, November 2013.

[40] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti. A Large-Scale Analysis of the Security of Embedded Firmwares. In *USENIX Security Symposium*, 2014.

[41] A. Costin, A. Zarras, and A. Francillon. Automated Dynamic Firmware Analysis at Scale: A Case Study on Embedded Web Interfaces. In *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, 2016.

[42] A. Cui, M. Costello, and S. J. Stolfo. When firmware modifications attack: A case study of embedded exploitation. In *Proceedings of the Symposium on Network and Distributed System Security (NDSS)*, 2013.

[43] A. Cui and S. J. Stolfo. A quantitative analysis of the insecurity of embedded network devices: Results of a wide-area scan. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 97–106, New York, NY, USA, 2010. ACM.

[44] A. Dabrowski and M. Slunsky. Hacking CCTV – Watching the watchers, having fun with cctv cameras, making yourself invisible. In *22nd Chaos Communication Congress*, 2005.

[45] J. Demme, M. Maycock, J. Schmitz, A. Tang, A. Waksman, S. Sethumadhavan, and S. Stolfo. On the feasibility of online malware detection with performance counters. In *ACM SIGARCH Computer Architecture News*, volume 41, pages 559–570. ACM, 2013.

[46] A. Dessiatnikoff, Y. Deswarte, E. Alata, and V. Nicomette. Potential attacks on onboard aerospace systems. *IEEE Security & Privacy*, (4):71–74, 2012.

[47] DigitalMunition. Owning a Police Car and It's DVR. http://www.digitalmunition.com/OwningCopCar.pdf.

[48] K. El Defrawy, A. Francillon, D. Perito, and G. Tsudik. Smart: Secure and minimal architecture for (establishing a dynamic) root of trust. In *Proceedings of the Network & Distributed System Security Symposium (NDSS), San Diego, CA*, 2012.

[49] J. Fridrich, M. Goljan, and R. Du. Reliable detection of lsb steganography in color and grayscale images. In *Proceedings of the 2001 workshop on Multimedia and security: new challenges*, pages 27–30. ACM, 2001.

[50] M. Gasser. *Building a secure computer system*. 1988.

[51] O. Gayer, O. Wilder, and I. Zeifman. CCTV Botnet In Our Own Back Yard. https://www.incapsula.com/blog/cctv-ddos-botnet-back-yard.html.

[52] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[53] M. Guri, O. Hasson, G. Kedma, and Y. Elovici. Visisploit: An optical covert-channel. *arXiv preprint arXiv:1607.03946*, 2016.

[54] M. Guri, A. Kachlon, O. Hasson, G. Kedma, Y. Mirsky, and Y. Elovici. Gsmem: data exfiltration from air-gapped computers over gsm frequencies. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 849–864, 2015.

[55] M. Guri, G. Kedma, A. Kachlon, and Y. Elovici. Airhopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies. In *Malicious and Unwanted Software: The Americas (MALWARE), 2014 9th International Conference on*, pages 58–67. IEEE, 2014.

[56] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici. Bitwhisper: Covert signaling channel between air-gapped computers using thermal manipulations. In *2015 IEEE 28th Computer Security Foundations Symposium*, pages 276–289. IEEE, 2015.

[57] M. Guri, Y. Solewicz, A. Daidakulov, and Y. Elovici. Fansmitter: Acoustic data exfiltration from (speakerless) air-gapped computers. *arXiv preprint arXiv:1606.05915*, 2016.

[58] M. Hanspach and M. Goetz. On covert acoustical mesh networks in air. *arXiv preprint arXiv:1406.1213*, 2014.

[59] C. Heffner. Exploiting Surveillance Cameras. Like a Hollywood Hacker. In *BlackHat US*, 2013.

[60] D. Hely, F. Bancel, M.-L. Flottes, and B. Rouzeyre. Secure scan techniques: a comparison. In *IEEE International On-Line Testing Symposium (IOLTS)*, 2006.

[61] iPower Technologies. Hidden Virus Discovered in Martel Police Body Camera. http://www.goipower.com/?pageId=40, November 2015. Accessed: July 25, 2016.

[62] iSpy. iSpyConnect – the worldâĂŹs most popular open source video surveillance application. https://www.ispyconnect.com/sources.aspx, 2007. Accessed: July 26, 2016.

[63] N. Jenkins. 245 million video surveillance cameras installed globally in 2014. June 2015.

[64] U. Johannes. This is why your DVR attacked my Synology Disk Station (and now with Bitcoin Miner!), April 2014.

[65] A. Kharraz, E. Kirda, W. Robertson, D. Balzarotti, and A. Francillon. Optical delusions: A study of malicious QR codes in the wild. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 192–203. IEEE, 2014.

[66] A. Kidman. How A Prison Had Its CCTV Hacked. http://goo.gl/sKombD, September 2012.

[67] G.-W. Kim and J.-W. Han. Security model for video surveillance system. In *International Conference on ICT Convergence (ICTC)*. IEEE, 2012.

[68] D. Kriesel. Xerox scanners/photocopiers randomly alter numbers in scanned documents, 2014.

[69] J. Kuboviak. Legal admissibility of digital video recordings. *LAW AND ORDER-WILMETTE THEN DEERFIELD-*, 52(4):92–99, 2004.

[70] M. G. Kuhn and R. J. Anderson. Soft tempest: Hidden data transmission using electromagnetic emanations. In *International Workshop on Information Hiding*, pages 124–142. Springer, 1998.

[71] I.-S. Lee and S. Y. Wan. Security Requirements for Network CCTV. *World Academy of Science*, 70, 2010.

[72] Y. Liu, P. Ning, H. Dai, and A. Liu. Randomized differential dsss: Jamming-resistant wireless broadcast communication. In *INFOCOM*. IEEE, 2010.

[73] J. Loughry and D. A. Umphress. Information leakage from optical emanations. *ACM Transactions on Information and System Security (TISSEC)*, 5(3):262–289, 2002.

[74] D. Maass, C. Quintin, and EFF. License Plate Readers Exposed!, October 2015.

[75] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)*, pages 5188–5196. IEEE, 2015.

[76] MajorMalfuntion. Old Skewl Hacking – InfraRed updated. In *22nd Chaos Communication Congress*, 2005.

[77] J. Marpet. Physical Security in a Networked World: Video Analytics, Video Surveillance, and You. In *DefCon*, 2010.

[78] Y. Mirsky, M. Guri, and Y. Elovici. Hvacker: Bridging the air-gap by manipulating the environment temperature.

[79] T. Morkel, J. H. Eloff, and M. S. Olivier. An overview of image steganography. In *ISSA*, pages 1–11, 2005.

[80] K. Mowery, E. Wustrow, T. Wypych, C. Singleton, C. Comfort, E. Rescorla, J. A. Halderman, H. Shacham, and S. Checkoway. Security analysis of a full-body scanner. In *23rd USENIX Security Symposium (USENIX Security 14)*, pages 369–384, 2014.

[81] C. Mulliner and B. Michéle. Read it twice! a mass-storage-based TOCTTOU attack. In *Proceedings of the 6th USENIX conference on Offensive Technologies*, pages 11–11. USENIX Association, 2012.

[82] J. Newsome and D. Song. Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software. 2005.

[83] A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436. IEEE, 2015.

[84] R. K. Nichols and P. C. Lekkas. *Wireless security*. McGraw-Hill New York.

[85] J. Obermaier and M. Hutle. Analyzing the Security and Privacy of Cloud-based Video Surveillance Systems. In *Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, pages 22–28. ACM, 2016.

[86] M. Olson. Beware, even things on Amazon come with embedded malwareâĂę. http://artfulhacker.com/post/142519805054/beware-even-things-on-amazon-come, April 2016. Accessed: July 25, 2016.

[87] OWASP. Buffer Overflow. owasp.org/index.php/Buffer_overflow_attack.

[88] OWASP. Command Injection. owasp.org/index.php/Command_Injection.

[89] OWASP. Information Leakage. owasp.org/index.php/Information_Leakage.

[90] OWASP. Path Traversal. owasp.org/index.php/Path_Traversal.

[91] OWASP. Top 10 Vulnerabilities 2013. owasp.org/index.php/Top_10_2013-Top_10.

[92] S. J. OâĂŹMalley and K.-K. R. Choo. Bridging the air gap: Inaudible data exfiltration by insiders. In *20th Americas Conference on Information Systems (AMCIS 2014)*, pages 7–10, 2014.

[93] D. Papp, Z. Ma, and L. Buttyan. Embedded systems security: Threats, vulnerabilities, and attack taxonomy. In *Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2015.

[94] T.-S. Park and M.-S. Jun. User authentication protocol for blocking malicious user in Network CCTV environment. In *Computer Sciences and Convergence Information Technology (ICCIT), 2011 6th International Conference on*, pages 18–24. IEEE, 2011.

[95] C. Pöpper, M. Strasser, and S. Capkun. Jamming-resistant broadcast communication without shared keys. In *USENIX security Symposium*, pages 231–248, 2009.

[96] ProCheckup. Owning Big Brother: Multiple vulnerabilities on Axis 2100.

[97] N. Provos and P. Honeyman. Hide and seek: An introduction to steganography. *IEEE Security & Privacy*, 1(3):32–44, 2003.

[98] C. Pu and J. Wei. A methodical defense against tocttou attacks: The edgi approach. In *International Symposium on Secure Software Engineering (ISSSE)*, 2006.

[99] G. Ritt and B. Eberle. Sensor protection against laser dazzling. In *Security+ Defence*, pages 783404–783404. International Society for Optics and Photonics, 2010.

[100] E. Ronen and A. Shamir. Extended Functionality Attacks on IoT Devices: The Case of Smart Lights. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 3–12. IEEE, 2016.

[101] E. J. Schwartz, T. Avgerinos, and D. Brumley. All you ever wanted to know about dynamic taint analysis and forward symbolic execution (but might have been afraid to ask). In *IEEE Symposium on Security and Privacy*. IEEE, 2010.

[102] V. Sepetnitsky, M. Guri, and Y. Elovici. Exfiltration of information from air-gapped machines using monitor's led indicator. In *Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint*, pages 264–267. IEEE, 2014.

[103] S. Shekyan and A. Harutyunyan. To Watch Or To Be Watched. Turning your surveillance camera against you. In *HITB Amsterdam*, 2013.

[104] Shodan. Shodan Images – an easier way to browse the screenshots that Shodan collects. https://images.shodan.io/.

[105] S. Skorobogatov and C. Woods. Breakthrough silicon scanning discovers backdoor in military chip. In E. Prouff and P. Schaumont, editors, *Cryptographic Hardware and Embedded Systems âĂŞ CHES 2012*, volume 7428 of *Lecture Notes in Computer Science*, pages 23–40. Springer Berlin Heidelberg, 2012.

[106] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan,

I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[107] D. H. Titterton. A review of the development of optical countermeasures. In *European Symposium on Optics and Photonics for Defence and Security*, pages 1–15. International Society for Optics and Photonics, 2004.

[108] D. Tsafrir, T. Hertz, D. Wagner, and D. Da Silva. Portably Solving File TOCTTOU Races with Hardness Amplification. In *FAST*, volume 8, pages 1–18.

[109] M. Vuagnoux and S. Pasini. Compromising electromagnetic emanations of wired and wireless keyboards. In *USENIX security symposium*, pages 1–16, 2009.

[110] G. Wei. Evaluation method for jamming effectiveness on electro-optical imaging systems [j]. *Opto-Electronic Engineering*, 33(2):5–8, 2006.

[111] J. Wei and C. Pu. TOCTTOU Vulnerabilities in UNIX-Style File Systems: An Anatomical Study. In *FAST*, volume 5, pages 12–12, 2005.

[112] Y. Xia, Y. Liu, H. Chen, and B. Zang. Cfimon: Detecting violation of control flow integrity using performance counters. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012)*, pages 1–12. IEEE, 2012.

[113] YouTube. The fastest robbery – 1 min in bank. http://youtu.be/LFArxqcP4MI.

[114] J. Zaddach and A. Costin. Embedded devices security and firmware reverse engineering. *BlackHat USA*, 2013.

[115] K. Zetter. CCTV Hack Results In 33M USD Casino Theft. http://goo.gl/zmxVXe.

[116] B. Zhu, A. Joseph, and S. Sastry. A taxonomy of cyber attacks on SCADA systems. In *International conference on cyber, physical and social computing Internet of things (iThings/CPSCom)*. IEEE, 2011.

Table 1: Summary of threats, vulnerabilities, attacks, and mitigations classification for video surveillance systems.

Attack category	Attack surface	Attack type	Attacker type	Directly affected components	Exploitation complexity	Mitigation complexity	Additional comments on mitigation (if applicable)
Software	Web Interface Other Interfaces	Weak access control or weak authentication	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy	Easy [91]	- Do not use/disable default passwords - Remove hard-coded passwords/accounts - Implement and enforce strong password update policies
Software	Web Interface Other Interfaces	Insufficient Transport Layer Protection [91]	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy	Easy [91]	- Disable clear-text and non-mutually authenticated protocols - Enable and use only HTTPS-like secured protocols - Enable mutually-authenticated protocols
Software	Web Interface Other Interfaces	Denial-of-Service (DoS)	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy [7]	Complex (it is far easier to build a secure system than to build a correct (and robust) system [50])	- Limit resource allocation - Cache content - Reinforce error handlers - Check buffer overflows - Validate inputs
Software	Web Interface	XSS [91]	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy [33, 29, 32]	Easy to Complex [91]	- Properly escape all untrusted data - Positive or "whitelist" input validation - Use auto-sanitization libraries
Software	Web Interface	CSRF [91]	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy [33, 29, 32]	Easy to Complex [91]	- Use unpredictable tokens in each HTTP request - Generate and include the unique token in a hidden field - Reauthenticate and re-CAPTCHA users
Software	Web Interface	Path traversal [90]	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy [8, 6]	Easy [90]	- Validate and escape the inputs - Use chrooted jails and code access policies - Normalize the input
Software	Web Interface	Information leakage via file disclosure [89]	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy [5]	Easy [89]	- Reinforce error handlers - Validate inputs - Check request authorization - Disable verbose logging
Software	Web Interface	Command injection [88]	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy [25]	Easy to Medium [88]	- Validate and normalize inputs - Use APIs instead of raw system calls - Implement a positive or "whitelist" security model
Software	Web Interface	Buffer overflow [87]	Network-Remote Network-Local	- Firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy [25]	Easy to Complex [87]	- Validate inputs - Use safe APIs instead of outdated unsafe versions - Use static and dynamic checking tools for discovery - Use compiler-based canary mechanisms
Software	Firmware Update	Reverse engineering	Network-Remote Network-Local Physical-Local	- Bootloader, kernel, firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy to Complex	Easy to Complex	- Firmware encryption using crypto standards and PKI
Software	Firmware Update	Unsigned/unverified upgrade [42]	Network-Remote Network-Local Physical-Local	- Bootloader, kernel, firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy to Complex	Easy to Complex [42]	- Firmware signing and verification using PKI, secure hashing
Software/Hardware	Mechanical Pan-Tilt-Zoom (PTZ)	Data exfiltration	Network-Remote Network-Local Physical-Local	- Cameras with PTZ support - Data "within reach" of camera	Complex	Easy to Medium	
Hardware	Debug Port	- Debug protocols attacks - Bootloader attacks - Unsigned/unverified upgrade [42]	Physical-Local	- Bootloader, kernel, firmware of DVR, NVR, IP-camera	Easy to Complex [25]	Complex	- Implement "secure scan" techniques [60] - Securely sign and verify bootloaders and firmware images
Hardware	USB Port	- TOCTTOU [81, 111] - Unsigned/unverified upgrade [42]	Physical-Local	- Bootloader, kernel, firmware of DVR, NVR, IP-camera - Software of VMS, CMS, video server	Easy	Medium to Complex [108, 98]	- Copy the software or firmware files to internal storage and then execute the checks on the copy
Optical	Visual Layer Malicious Images (Imagery Semantics)	- Command and control - Data infiltration	Physical-Local Line of sight	- Cameras - Video sensors - NVR/DVR - Video/Image processing elements	Easy to Complex [39, 80]	Easy to Complex [105, 45, 112, 36]	
Optical	Visual Layer VisiSploit (Imagery Semantics)	Data exfiltration	Physical-Local Line of sight	- VSS, Cameras, DVR, NVR connected to LCD displays visible to attacker	Complex [53]	Complex [53]	
Optical	Visual Layer Steganography (Imagery Semantics, Metadata)	Data exfiltration	Network-Remote Network-Local Physical-Local	- VSS, Cameras, DVR, NVR providing image and video feeds	Easy to Medium [97, 79]	Easy to Complex [31, 49]	
Optical	- PHY LED (output) - PHY Infrared (output)	- Data exfiltration - Command and control	Physical-Local Line of sight	- Cameras with normal and/or IR LEDs - Data "within reach" of camera	Easy to Medium [73, 37, 102, 34]	Medium to Complex	
Optical	PHY Infrared	- Command and control - Denial-of-Service (DoS)	Physical-Local Line of sight	- NVR/DVR with IR remote control - Cameras with IR remote control	Easy to Complex [76]	Medium to Complex	
Optical	PHY Infrared	Camera blinding (dazzling)	Physical-Local Line of sight	- Cameras - Video sensors	Easy to Medium [18, 44]	Easy to Medium [107]	- Use infrared filters (in turn, that affects night-vision features)
Optical	PHY Laser	Camera blinding (dazzling)	Physical-Local Line of sight	- Cameras - Video sensors	Medium to Complex [18, 44]	Complex [107, 99, 110]	- Use wave-length agile filters - Spatial light modulator and wavelength multiplexing
RF/Wireless	Radio Frequency (RF)	Denial-of-Service (DoS) RF Jamming	Physical-Local Line of sight Physical-Remote	- Communication links	Easy	Medium to Complex	- Spread spectrum solutions as - DSSS [84], FHSS [84], UDSSS [95], RD-DSSS [72]
RF/Wireless	Radio Frequency (RF)	Eavesdropping	Physical-Local Line of sight Physical-Remote	- Communication links - Private data	Easy [21]	Medium to Complex	- Spread spectrum solutions as - DSSS [84], FHSS [84], UDSSS [95], RD-DSSS [72]
RF/Wireless	Wi-Fi 802.11	Denial-of-Service (DoS) RF Jamming	Physical-Local Line of sight Physical-Remote	- Communication links	Easy [30]	Medium to Complex [30]	
RF/Wireless	Wi-Fi 802.11	Eavesdropping	Physical-Local Line of sight Physical-Remote	- Communication links - Private data	Easy	Easy	- Do not use default or simple credentials - Use strong protocols (e.g., WPA2)