Video Surveillance Systems

Subjects: Others

Contributor: Preethi Vennam

Video surveillance systems are widely deployed with large systems for use in strategic places such as home security, public transportation, banks, ATM centers, city centers, airports, and public roads, and play a vital role in protecting critical infrastructures. As various attacks are possible in these systems, identifying attacks and considering suitable security measures are essential. In this paper, we present a detailed review of existing and possible threats in video surveillance, CCTV, and IP-camera systems.

Keywords: video surveillance system (VSS) ; closed-circuit television cameras (CCTV) ; Internet of Things (IoT) ; security attacks

1. Introduction

Government and private organizations, residential societies, and commercial and public spaces, are using these systems to keep a check on various activities for security and safety purposes. Surveillance means monitoring movements, activities and behavior in order to manage, control, and protect people. To view events as they occur and to monitor activities in any area at a later time, closed-circuit television systems (CCTV) technology is being used. Increasing thefts and criminal activities demand the usage of CCTV cameras in both commercial and residential sectors for security purposes.

The virtue of IoT is that it gives new look for the upcoming video surveillance systems. Instead of capturing footage and visualizing it later in order to detect theft, violence or vandalism, there is a need for cameras to self-detect the abnormal events and interpret the same to other systems for necessary actions. The smart cameras have exploited the benefits of computer vision, machine learning and automation. IoT helps to connect network-enabled cameras with other devices and systems and thus transforms secure surveillance into smart security surveillance systems.

Overview of VSS: Video surveillance systems are widely used in cyber-systems such as healthcare, traffic analysis, wildlife monitoring, environmental monitoring, weather forecasting and public safety. Each node performs video compression, data transmission and video capturing as the basic function. The data processing unit and data transmission unit at each wireless node process a large amount of video data without degrading information and security, which is a most challenging task in video surveillance applications [1].

The usage of VSS is ubiquitous in today's scenario. Attackers are continuously targeting these systems with new attacks and vulnerabilities. For example, when a simple search word such as "webcamXP" is given on Shodan.io [2], an IoT search engine, one can access random video footage of retail stores, city centers, boating docks, and domestic spaces. The large scale, restricted resources, outdated firmware, poorly secured IoT devices and inbuilt vulnerabilities have attracted bad actors to perform various attacks on the IoT ecosystem.

The motivation for an attacker could be blackmailing, the ability to observe live video feed, access to video footage, access to VSS network, disabling video feeds, violating privacy, remotely disabling the connection, and performing DoS attacks, etc. As VSSs are used in important places, only authorized agents should have the access to monitor and control it. Privacy and security are the foremost concerns while using such systems. Considering all this, this paper first identifies the possible attacks on such systems and then discusses the measures that can be incorporated to prevent security attacks.

After the launch of the Mirai attack and its consequences in the year 2016, there has been a dramatic increase in studies related to attacks and vulnerabilities in the VSS domain. Keywords used for the literature survey are as follows: video surveillance systems, attacks on VSS, security frameworks for VSS, privacy issues with IP camera and botnet. To understand the security loopholes and possible solutions to mitigate the threats in VSS, this paper follows the following steps (**Figure 1**) to articulate the security issues of VSS.

Figure 1. Roadmap of the paper.

The rest of the paper is organized as follows: various types of attacks in VSS are given in Section 2. The security measures for VSS are summarized in Section 3. In Section 4, a detailed review and analysis of the latest advances in VSS frameworks are presented and tabulated.

2. Attacks on VSS

In this section, we present all the possible types of active attacks at different layers of the video surveillance systems. The main issues are (a) privacy and security that concerns a surveillance system, (b) the uncertainty of not knowing what happens to your data when it is stored in the cloud and (c) how the user monitoring devices such as smartphones can also be a cause of the attack in the surveillance network [3][4][5][6][7][8][9][10][11][12].

In an attack scenario, the basic steps are: (a) information gathering, (b) assessing vulnerability, (c) launching attack, and (d) cleaning up. Some of the tools used by attackers at different steps are listed in **Figure 2**. [13] present more elaborate details of information gathering and attack launching tools that can be used by attackers. [14] presents details of different vulnerability databases available, attack surfaces and their details.

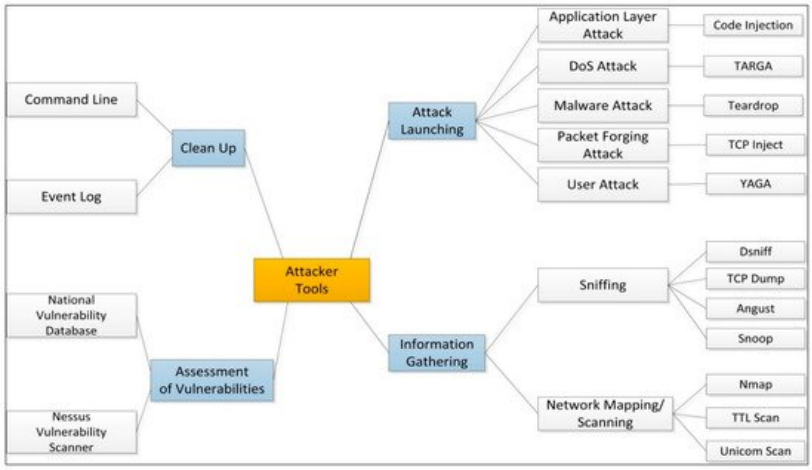

Figure 2. Common tools used by attackers.

VSSs are used by applications for the recognition of facial features, the automatic reading of license plates, scanning and reading QR codes and the compression of image data. VSS has an additional level of abstraction, i.e., the visual layer. This layer is prone to a few types of attacks as they involve imagery semantics and image recognition. The attacks are spread/injected in a multitude of ways, such as preinstalled malware in the system or through a firmware update or remote command insertion [15].

One of the most common attacks on a live feed from VSS is frame duplication attack. An attacker, once gaining access to a VSS system, can insert previously recorded “normal” looking frames in place of the live stream, to avoid the detection of ongoing suspicious activity. To detect these frame duplication attacks, spatial and temporal domain similarities between frames are extracted and analyzed using various correlation techniques. To achieve this, a massive database is required for storing a huge amount of data and an enormous amount of computation time is required to detect and prevent such attacks in real time.

In this type of attack, informational objects between processes can communicate which normally should be blocked as per the security policy. These attacks are different from legitimate channel exploitations which attack semi-secured systems using techniques such as steganography, to disguise prohibited objects inside actual informational objects. Based on criteria such as timing/storage, network/OS/hardware, and value/transition based, covert attacks can be classified into various types. Some examples of covert channel attacks [16][17] are:Manipulating CCTV/VSS infrared LEDs: by sending command/control data to the VSS cameras by using the infrared LED messages;A new type of optical covert channel named (VisisSploit)

Steganography involves a method to use the unused or less important information bits of the user content (such as images, videos, network traffic). Two types of common steganography attacks are hiding the malicious code in the genuine application and by a command and control (C & C) communications channel [18][19].

A common technique in many malware droppers is to append data to the end of the file or utilize unused portions of the file format [20]. In any method of steganography, it is hard to detect malicious code coming through user files in a network. Malicious payloads can be embedded into a set of PNG files. The PNG files can then be compiled into a legitimate application, along with a function that would extract and drop the malware onto the system.

In command and control protocol attacks, the “Domain Name Server (DNS) and Hyper Text Transfer Protocol (HTTP)”, can be used to embed the malicious code in response to a request from a client.

As an example, we consider User A (house or office owner) who approaches the IP camera. (named a cover image). The steganography technique is now used to send a “stego image” (combined image where the actual image is hidden in the cover image). This “stego image” is stored in the home server, which then will be processed using the reverse steganography technique to retrieve the original image with the face.

Another User B (attacker) intercepts data transmission between the IP camera and the home server and captures all the data which have “stego image” along with other captured images. At this point, the attacker can perform three categories of attack; namely, stego-only attack, known cover attack and known message attack. Any change on the LSB bits of the face image will not alter it significantly, whereas changes in the MSB bits will significantly degrade the quality of the face image. An attacker can use statistical analysis for the detection of changes in LSB bits or human visual perception to detect the changes in the MSB bits to detect the face image from the cover image.

Zoom (PTZ) is a functional characteristic of a surveillance camera that can zoom in and out, and change the view of the camera to horizontal (right, left) and vertical (up, down) angles. Camera models utilize stepper motors built into them and employ PTZ data protocols to achieve this functionality. When a user is using a mobile application to watch a live feed from the camera through a cloud server, then all the PTZ requests are routed through cloud servers to the camera. If this communication is carried out after an interval of every few seconds, an attacker who is intercepting this communication may not be able to decode the PTZ data but can precisely find the interval after which communication is happening.

When monitoring important activities such as real time crimes, in many video surveillance systems, it is critically important to have an un-tampered and uninterrupted operation. A denial-of-service attack on a home surveillance camera will not have a major impact when compared to denial-of-service attacks on commercial surveillance systems, which may have a greater impact. These kinds of attacks must be taken into consideration during the early phases of the setup and testing of the surveillance system. For example, “BrickerBot is a malware that attacks IoT devices that run a specific version of the DropBear SSH server and target Linux devices running Busy box (usually IP cameras)”.

DoS attacks can be classified into two types: flooding and logic attacks. Flooding attacks work by overwhelming the current network with a large volume of complex data packets to deplete their resources such as memory and bandwidth. Logic attacks exploit the known vulnerabilities in the system to attack the remote servers. Out of these two types of attacks, flooding attacks are more dangerous as it is difficult (resource-intensive, time-intensive and cost-intensive) to differentiate real data packets from the flooded data.

In a smartphone, users download mobile applications, and malicious code embedded into the application program can gain access to personal information which the attackers can then exploit for financial gain [21]. Nor does anyone want a picture or video of their device or application storage that went viral on social media due to their camera (which can be a surveillance camera or smartphone camera) being hacked. Due to this, the attacker can have a different way to invade by performing malicious code injection, data leaks and also performing privilege escalation. Access control entry vulnerabilities have been discovered on IP cameras, DVRs, and VPN routers which are publicly listed inhttps://cve.mitre.org.

In a multiple user architecture of any application or device network, access permissions to its users are restricted. Users at different levels have different permissions. In Android user applications or surveillance applications, components such as service, content provider, broadcast receiver and activity may be able to use privilege escalation to receive more permissions than required or desired. Two variants of privilege escalation are Vertical Privilege Escalation and Horizontal Privilege Escalation.

Vertical Privilege Escalation: bugs and design flaws can be applied to allow the smartphone user to execute higher level applications or functions. Even a process, for instance, may use a bug in the system kernel and run functions with system privileges. There must be at least one process running with system rights to enable another lower-level process to

escalate.

Horizontal Privilege Escalation: the user and applications are located at the same permission level. Privilege escalation takes place if a user or an application can access data or functions of another user or application.

One of the Android built-in security features is the Android application sandbox. It is a technique to manage and separate the user applications from the critical system resources and applications. Privilege escalation attack bypasses sandbox restriction by running malicious code at run time [22]. An application which is “non-privileged” can still access files of “privileged” system applications such as geo-location, user passcode, battery status, camera permission, etc.

Similarly, in a video surveillance system, an attacker can exploit the firmware default port and login information and access the device as a user with privileged rights [23]. In such a scenario, companies could do nothing but recommend their customers apply newer firmware and use stronger passwords.

The prevalent attacks on different parts of the VSS infrastructure are outlined in **Figure 3**.**Table 1** gives information on different types of attacks, their description and examples of how such attacks are conducted by the attackers.

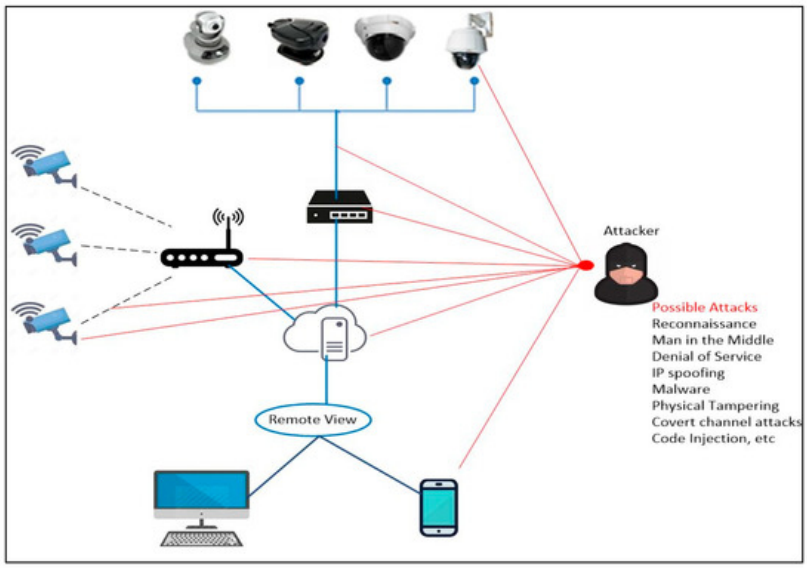

Figure 3. Pervasive attacks on VSS.

Table 1. VSS architecture layers, attacks and their examples.

Layer	Attacks	Threats	Description	Examples
Perception Layer	Device attack	Physical Attacks, impersonation, malicious code injection	Someone takes advantage of a bug or inherent vulnerability to gain access to the infrastructure.	Physical access to a security surveillance camera and modifying the design settings.
	IoT botnet	DoS attacks, routing attacks	Group of hacked computers, smart devices, and appliances connected to the Internet are known as an IoT botnet.	The Mirai malware is seen as a milestone in the threat landscape and exploits security holes in IoT devices and launches attacks.

Layer	Attacks	Threats	Description	Examples
Network Layer	Attacks on Wifi/Ethernet	Routing attacks, data transit attacks	Numerous malicious activities can be performed on devices if an attacker gains physical access to the local network wirelessly.	In the network level attacks, cybercriminals are able to redirect network traffic; for example, Address Resolution Protocol poisoning (ARP) or by changing the Domain Name System (DNS) settings.
	Reconnaissance	DoS attacks, routing attacks	The aim is to collect data about an infrastructure, including the network services and devices that are running.	This can be achieved by scanning network ports and packet sniffers.
	Man-in-the-middle attack	Data transit attacks	It is a type of eavesdropping attack. This attack could permit the attacker to secretly relay and possibly alter the communications between two IoT devices.	Attackers can use a network packet analyzer, i.e., Wireshark for analyzing network traffic. If communications are not encrypted or authenticated, an attacker can easily steal the data.
Application Layer	Cloud infrastructure	Data leakage, DoS attacks, malicious code injection	An IoT device interconnects with back-end cloud services. IoT cloud services might permit the client to select simple passwords.	A lot of cloud services have a logical weakness, which is actually the permission of cloud to a cybercriminal to obtain sensitive information of the customer and also access to the device without any authentication.
	Privilege escalation	Data leakage, malicious code injection	The attacker takes advantage of programming errors or software flaws to permit cybercriminals to elevate access to an IoT infrastructure.	Grant the cybercriminal elevated access to the IoT ecosystem and its associated data and applications.
	Server-side denial of service (DoS)	DoS attacks, Malicious Code Injection	Electronic devices and its connected devices are deactivated or changed by a cybercriminal, via physical or remote access to the IoT sensors.	An attacker can deny the sensors to send and receive communications. Another example could be battery abuse, device disabling, or device bricking.

3. Security Measures for VSS

The security of the hardware, firmware and network communications of video surveillance systems can be enhanced by following the guidelines summarized in this section. Vendors must adopt good practices for built-in security measures, such as secure remote access, basic encryption, and patching all known vulnerabilities [24][25][26][27][28][29][30][31][32]. Without proper safeguarding, IP-connected cameras are vulnerable to hacking, which can lead to the compromise of millions of security cameras and video recorders. To protect from security attacks, the security measures that are suitable at different layers (perception layer, network layer and application layer) are summarized in **Figure 4**.

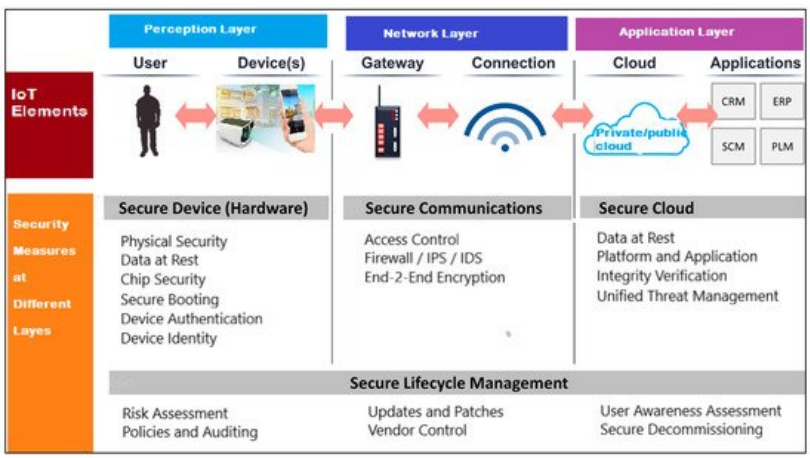

Figure 4. Summarizing the security measures at different layers.

The basic and necessary steps to avoid video surveillance camera attacks are as follows: Network topology and configuration of a system is critical in maintaining the security of IP-based cameras, as there are multiple entry gateways through which it can be attacked. In this type of locally connected system, rather than relying on a password to gain access to the firewall of a camera system, cloud-connected IP security cameras will communicate with a secured cloud-based server over an encrypted connection. Cloud-connected devices have the added advantage of continuous monitoring over locally connected systems.

References

1. Yan, W.Q.; Zhou, L.; Shu, Y.; Yu, J. CVSS: A Cloud-Based Visual Surveillance System. Int. J. Digit. Crime For. 2018, 10, 79–91.
2. Shodan. Available online: (accessed on 2 November 2020).
3. Becher, M.; Freiling, F.C.; Hoffmann, J.; Holz, T.; Uellenbeck, S.; Wolf, C. Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of Mobile Devices. In Proceedings of the 2011 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 22–25 May 2011; pp. 96–111.
4. McAfee Labs. McAfee Threats Report: First Quarter 2013; McAffe Press: San Jose, CA, USA, 2013; Available online: (accessed on 15 May 2021).
5. F-Secure Labs. Mobile Threat Report January-March 2013; F-Secure Labs: Helsinki, Finland, 2013; Available online: (accessed on 15 May 2021).
6. Stites, D.; Tadimla, A. A Survey of Mobile Device Security, Threats, Vulnerabilities and Defences. 2011. Available online: (accessed on 15 May 2021).
7. Enck, W.; Gilbert, P.; Chun, B.G.; Cox, L.P.; Jung, J.; McDaniel, P.; Sheth, A.P.; Droid, T. An Information on Tracking System for Real Time Privacy Monitoring on Smart-Phones. In Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, Vancouver, BC, Canada, 4–6 October 2010; USENIX Association: Berkeley, CA, USA, 2020; pp. 1–6.
8. Franklin, J.; Brown, C.; Dog, S.; McNab, N.; Voss-Northrop, S.; Peck, M.; Stidham, B. Assessing Threats to Mobile Devices & Infrastructure NISTIR 8144. Available online: (accessed on 15 May 2021).
9. Zheng, P.; Lionel, M.N. Spotlight: The rise of the smartphone. IEEE Distrib. Syst. Online 2006, 7, 3.
10. Liranzo, J.; Hayajneh, T. Security and Privacy Issues Affecting Cloud-Based IP camera. In Proceedings of the 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), New York, NY, USA, 19–21 October 2017; pp. 458–465.
11. Hoque, N.; Bhuyan, M.H.; Baishya, R.C.; Bhattacharyya, D.K.; Kalita, J.K. Network attacks: Taxonomy, tools and systems. J. Netw. Comput. Appl. 2014, 40, 307–324.
12. Rytel, M.; Felkner, A.; Janiszewski, M. Towards a Safer Internet of Things—A Survey of IoT Vulnerability Data Sources. Sensors 2020, 20, 5969.
13. Costin, A. Poor Man's Panopticon: Mass CCTV Surveillance for the Masses. Available online: (accessed on 15 May 2021).
14. Mowery, K.; Wustrow, E.; Wypych, T.; Singleton, C.; Comfort, C.; Rescorla, E.; Halderman, J.A.; Shacham, H.; Checkoway, S. Security analysis of a full-body scanner. In 23rd USENIX Security Symposium USENIX Security 14; USENIX Association: San Diego, CA, USA, 2014; pp. 369–384.
15. Jones, E.; Le Moigne, O.; Robert, J.-M. IP traceback solutions based on time to live covert channel. In Proceedings of the 2004 12th IEEE International Conference on Networks (ICON 2004) (IEEE Cat. No. 04EX955), Singapore, 19 November 2004; pp. 451–457.
16. Guri, M.; Hasson, O.; Kedma, G.; Elovici, Y. Visisploit: An optical covert-channel. arXiv 2016, arXiv:1607.03946.
17. Sloan, T.; Hernandez-Castro, J. Forensic analysis of video steganography tools. PeerJ Comput. Sci. 2015, 1, e7.
18. Senthil, M. CCTV Surveillance System, attacks and design goals. Int. J. Electr. Comput. Eng. 2018, 8, 2072–2082.
19. Maharjan, R.; Shrestha, A.K.; Basnet, R. Image Steganography: Protection of Digital Properties against Eavesdropping. arXiv 2019, arXiv:1909.04685.
20. Yin, J.; Fen, G.; Mughal, F.; Iranmanesh, V. Internet of Things: Securing Data using Image Steganography. In Proceedings of the 2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS), Kota Kinabalu, Malaysia, 2–4 December 2015.

21. Available online: (accessed on 11 March 2021).
22. Hur, J.B.; Shamsi, J.A. A survey on security issues, vulnerabilities and attacks in Android based smartphone. In Proceedings of the 2017 International Conference on Information and Communication Technologies (ICICT), Karachi, Pakistan, 30–31 December 2017; pp. 40–46.
23. Cai, Y.; Tang, Y.; Li, H.; Yu, L.; Zhou, H.; Luo, X.; He, L.; Su, P. Resource Race Attacks on Android. In Proceedings of the 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER), London, ON, Canada, 18–21 February 2020; pp. 47–58.
24. Raveendranath, R.; Rajamani, V.; Babu, A.J.; Datta, S.K. Android malware attacks and countermeasures: Cur-rent and future directions. In Proceedings of the 2014 International Conference on Control, Instrumentation, Communication and Computa-tional Technologies (ICCICCT), Kanyakumari, India, 10–11 July 2014; pp. 137–143.
25. Wetherall, D.; ChoDnes, D.; Greenstein, B.; Han, S.; Homyack, P.; Jung, J.; Schechter, S.; Wang, X. Privacy revela-tions for web and mobile apps. In 13th Workshop on Hot Topics in Operating Systems HotOS XIII; USENIX Association: Napa, CA, USA, 2011.
26. Jung, S.; Kwon, T. Automatic Smudge Attack Based on Machine Learning and Pattern Lock System Safety Analysis. J. Korea Inst. Inf. Secur. 2016, 26, 903–910.
27. Prema, S.; Pramod, T.C. Key Establishment Scheme for Intra and Inter Cluster Communication in WSN. In Proceedings of the 2018 Second. International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 15–16 February 2018; pp. 942–944.
28. T.C., P.; G.S., T.; Iyengar, S.S.; Sunitha, N.R. CKMI: Comprehensive key management infrastructure design for Industrial Automation and Control Systems. Future Internet 2019, 11, 126.
29. Wang, S.; Bie, R.; Zhao, F.; Zhang, N.; Cheng, X.; Choi, H. Security in wearable communications. IEEE Netw. 2016, 30, 61–67.
30. Pramod, T.; Sunitha, N. Key pre-distribution schemes to support various architectural deployment models in WSN. Int. J. Inf. Comput. Secur. 2016, 8, 139.
31. Pramod, T.C.; Sunitha, N.R. An approach to detect malicious activities in SCADA systems. In Proceedings of the 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, India, 4–6 July 2013; pp. 1–7.
32. Kalbo, N.; Mirsky, Y.; Shabtai, A.; Elovici, Y. The Security of IP-Based Video Surveillance Systems. Sensors 2020, 20, 4806.