Rise of the Machines: Transforming Cybersecurity Strategy for the Age of IoT

What are these “things”, what do they have to say, and are we listening?

A Technical Report from Forescout Research Labs

Contents

1. Introduction

In most organizations, Information Technology (IT) includes the technology used to manage and process information, while Operational Technology (OT) encompasses the devices and technology that interact with the physical world. IT and OT were, for a long time, regarded as two distinct areas of an organization. Nowadays, these two domains are converging with the rise of connected embedded devices in the Internet of Things (IoT). Consequently, IT security teams are increasingly responsible not only for protecting the processing of information but also for running secure business operations. According to Gartner, "by 2021, 70 percent of OT security will be managed directly by the CIO, CISO, or CSO department up from 35 percent today" [1].

> By 2021, 70 percent of OT security will be managed directly by the CIO, CISO, or CSO department up from 35 percent today.

The IoT revolution is happening all around us, but the definition of what these 'things' are can be fuzzy and often incomplete. Indeed, the definition of IoT is evolving and overlaps with previous concepts like computing and wireless sensor networks as well as, more recently, cyber-physical systems [2] [3]. For a historical overview of the IoT, see [4].

According to Gartner, "the Internet of Things (IoT) is the network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment" [5]. Every business vertical has its specific devices, but common examples of "things" that make up the IoT include smart thermostats, smart sensors and actuators, smart lights, smart TVs, smart cameras, and smart medical devices. Most of these things are IP-enabled, many are wireless, they are often mobile and have shared ownership, varying degrees of computational power, and are used in applications ranging from small home automation systems to large smart cities and very large smart grids.

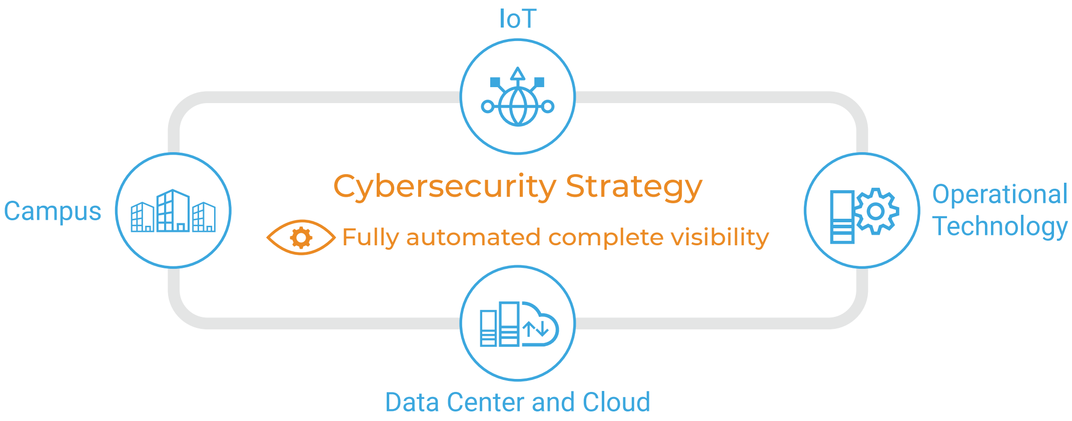
> The IoT has already experienced significant growth in the past decade and is expected to reach more than 30 billion connected devices by 2020.

The IoT has already experienced significant growth in the past decade and is expected to reach more than 30 billion connected devices by 2022 [6]. The age of IoT is rapidly transforming many business verticals such as manufacturing, energy, automotive, healthcare, and finance [7] by allowing, via smart sensors, the collection of large amounts of data, which, after being processed, can be used to drive actions in the physical world through smart actuators.

This business transformation brings along many opportunities for increased productivity, but it also presents challenges since the number of devices in a typical organization's network is rapidly increasing, these devices are mostly unmanaged, come from a multitude of vendors, use non-standard operating systems, support a diversity of, often insecure, protocols, and may dynamically connect to other devices inside or outside the organization's network [8]. Often, these devices enter an organization via "shadow IT" deployments or employees bringing their personal devices (BYOD) [9]. This heterogeneous and highly dynamic environment increases the attack surface of the organization and is further complicated by the fact that many of these devices are safety-critical and their disruption or tampering can have severe consequences in the physical world. Consider, for instance, the impact of a life-supporting medical device being accidently taken offline because of a networking issue or a cyberattack.

The challenges described above have led to a realization that cybersecurity management strategies have to change in this new era [8] and to a prediction that, by 2020, more than 25% of identified attacks in enterprises will involve the IoT [9]. Unsurprisingly, security has been identified as the main concern in IoT from a business perspective [10], as well as from a technical perspective [11].

In the age of IoT, legacy security solutions like endpoint agents, antivirus, and traditional IT intrusion detection systems are not enough because either they are unsupported by embedded devices or they are incapable of understanding the network traffic generated by these devices. Therefore, new solutions are required. Security teams must have complete visibility and enhanced control of all the assets in their network. Given the volume and diversity of devices, visibility must be fully automated. Given the range of applicable security solutions, such as device compliance, network segmentation, and incident response, control must be efficiently orchestrated.



An effective cybersecurity strategy must apply these solutions not only to the traditional campus network, but also to the IoT, data centers/cloud, and OT infrastructures of the business.

Smart buildings perfectly exemplify a cross-industry domain where IT and OT are converging and where IoT devices are proliferating [12]. Recently, there has been much talk about securing the Industrial IoT (IIoT) [13] [14], the Internet of Medical Things (IoMT) [15], and the IoT in home automation [16] [17]. On the other hand, the security implications of the IoT in smart buildings are often neglected [18] (with the notable exception of a recent NIST project [19]). Our recent research [20] has shown how these buildings can be vulnerable and how IoT devices can be leveraged as an entry point to the building's network. A real case of an IoT device serving as entry point to a corporate network is a data breach where a casino was hacked via the Internet-connected thermometer in an aquarium [22].

In this report, we use a smart building as a case study of a network where legacy OT assets (such as programmable logic controllers), IT systems (such as workstations) and IoT devices (such as IP cameras and smart lights) share the same network [21]. We leverage this case study to bring to the reader the following benefits:

- To shed light on the cybersecurity landscape and impacts of IoT in the networks of modern organizations, focusing on the interplay between modern IoT and legacy OT devices.
- To increase awareness about the cyber-risks to which these networks are exposed by evaluating their security posture.
- To demonstrate an easy-to-implement network monitoring solution that improves network resilience via device visibility and control.

2. Smart Buildings: A Case Study of IT-OT Convergence in the Age of IoT

In response to the need to reduce energy consumption and make buildings self-sustainable and more comfortable, a wide range of IoT devices is entering the smart building eco-system. We now have badges to access specific areas of a building, sensors to measure the air quality level in offices, solar panels to produce electricity, and smart meters to lower energy bills. A staggering range of new applications and services are enabled by these systems and devices, especially by their integration and communication.

> Smart buildings are much more “open” and interconnected than ICS, and while consumer-grade IoT devices will likely not get through the perimeter of ICS, they are entering (and reshaping) the building automation industry.

The benefits of the IoT in smart buildings are immeasurable, but unfortunately this evolution does not come without risks. One might think that smart buildings are just another incarnation of industrial control systems (ICS) and that their security should be handled like ICS security. This is a misunderstanding, since smart buildings are much more “open” and interconnected than ICS, and while consumer-grade IoT devices will likely not get through the perimeter of ICS, they are entering (and reshaping) the building automation industry.

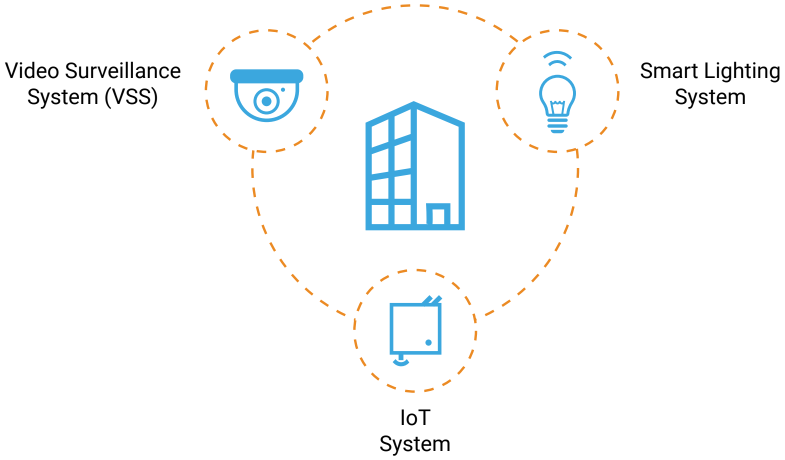
The new generation of smart buildings will most likely not replace existing legacy systems, but rather enhance them with new technologies. This means that we will witness the integration of old OT systems with the latest information technology IT and IoT devices, following the trend of IT-OT convergence.

This convergence is happening in many critical areas, such as medical facilities where IoT technology combined with legacy infrastructure can be used to improve the comfort of patients and make them more independent from medical staff. In fact, smart hospitals leverage smart buildings with the addition of healthcare-specific systems and devices that optimize patient comfort while increasing staff efficiency and effectiveness [22]. For example, patients who undergo long hospitalization periods can autonomously control HVAC and window blinds in their rooms, directly interact with entertainment systems like smart TVs and speakers, and have their vital signs continuously monitored and controlled thanks to patient monitoring devices.

Legacy buildings are managed by building automation systems (BAS) that include industry-specific sensors, actuators, and controllers that are expensive and can only be acquired through specific channels. With the advent of the IoT, sensors (e.g., for presence, humidity or temperature), basic dedicated controllers (e.g., thermostats), and many other devices (e.g., surveillance cameras) are available in consumer shops. They are much cheaper than industrial devices and far easier to install. In addition, they offer remote management via wireless connections such as Wi-Fi, Bluetooth or ZigBee. However, because of their fast time-to-market, they often lack security features [23] [16] and vulnerabilities are being discovered with increasing frequency [14] [24]. Often, the same device is sold under a variety of brands and names as a white label product [25], thus owners may not even be aware that their devices are vulnerable. In the worst case, some IoT vendors do not offer security patches even after vulnerabilities have been discovered because they lack the resources to do so [28]. In addition, bad security practices such as default credentials, simple passwords, unencrypted traffic and lack of network segregation remain common.

This situation should raise immediate concerns for facility managers, IT and OT security staff, as well as CIOs/CISOs for many reasons:

1. The increasing usage of consumer-grade components within a building's subsystems make it easier for an attacker to disrupt the building function (by 2020, 2.5 billion IoT sensors are expected to be deployed in smart buildings [21]).
2. A vulnerability in a connected IoT sensor might let the attacker perform lateral movement into a more critical (and far more fragile) network where great damage can be carried out [26] [20].
3. The complexity created by the interactions between simple IoT devices can be exploited by attackers to carry out actions that have unintended consequences [27] [17].
4. Emerging attack models such as "siegeware", when entire buildings are held for ransom [28], are facilitated by the increased attack surface provided by IoT devices.



A smart building is a treasure trove for attackers seeking to leverage IoT devices to cause physical disruptions or enable physical attacks. This is partly due to the number of subsystems that can be attacked and are interconnected via the IoT. To exemplify how attackers can achieve their malicious goals, in this report we focus on three subsystems commonly found in building automation networks:

1. A **Video Surveillance System (VSS)**, which helps to ensure the security of occupants by allowing a building asset owner to continuously monitor locations inside and near their facility. VSSs are highly exposed to external actors. This exposure is both physical, since many cameras are placed in external locations that make it easier for an attacker to tamper with them, and logical, since modern cameras and recording equipment support remote access for improved management and access to cloud services. The last few years have shown a surge of interest in IP cameras and network video recorders both from the security research community [29] [30] [31] [32] and from malicious actors [33] [34] [35].
2. A **Smart Lighting System**, which can automatically control the lights in a building based on factors like room occupancy and available daylight. As lights are integrated into building automation systems, they too become the sources and targets of attacks [36] [37] [38] [39]. Although smart lights are still not as widely deployed as surveillance cameras, and most attacks on smart lights are either academic or proof-of-concept examples, smart lighting is being rapidly adopted, with Gartner forecasting the technology to reach the "plateau of productivity" in its hype cycle in less than 2 years [40]. We believe that smart lighting in building automation is a trend that could soon be exploited by malicious actors.

3. An **IoT System**, which integrates components in different subsystems to offer services such as monitoring energy consumption and space utilization or predicting infrastructure maintenance needs. An IoT system is typically made up of several components [41]: IoT devices such as smart TVs and smart plugs; IoT gateways that allow the devices to communicate the data and measurements they collect; and an IoT platform (generally running on the cloud) that aggregates collected data and enables the provisioning of different services.

Notice that the "IoT system" could include both Video Surveillance and Smart Lighting, since IP cameras and smart lights are traditionally considered IoT devices. Nevertheless, we chose to classify those two systems separately because they have a well-defined scope and are well-understood within the building automation community. On the other hand, the "IoT system" in our categorization includes generic IoT devices, such as smart sensors and actuators. These devices act as linking elements with other subsystems or as standalone devices which do not fit in a pre-existing subsystem.

How attackers can exploit security vulnerabilities of other systems typically found in smart buildings (e.g., Access Control and HVAC) has been extensively discussed in our previous report [20].

3. Smart Building Reference Architecture

Devices in a smart building network need to communicate to share information about their status and to send commands to each other. For instance, a sensor reads the temperature of a room and provides it to a controller, which decides to switch a fan on or off, according to a setpoint configured by a management workstation.

These devices are typically grouped into subsystems according to their functionalities. For example, smoke detectors are part of the fire alarm system, whereas badge readers are part of the access control system. Ideally, these subsystem networks should be segregated from each other, and especially from the IT network, although that is rarely the case in practice, as confirmed by our daily experience with securing production networks. Sometimes different subsystems are configured in different VLANs for network segmentation, but misconfigurations allowing cross-VLAN communication (VLAN hopping) are common [45].

The architecture of a typical smart building network is shown in Figure 1, where systems including Video Surveillance, Access Control, IoT, HVAC, and Smart Lighting are connected. Besides residential and commercial buildings, the reference architecture shown in Figure 1 can also represent the networks found in critical or sensitive facilities such as hospitals, factories, airports, stadiums, schools, data centers, and many other types of buildings with a large number of occupants.

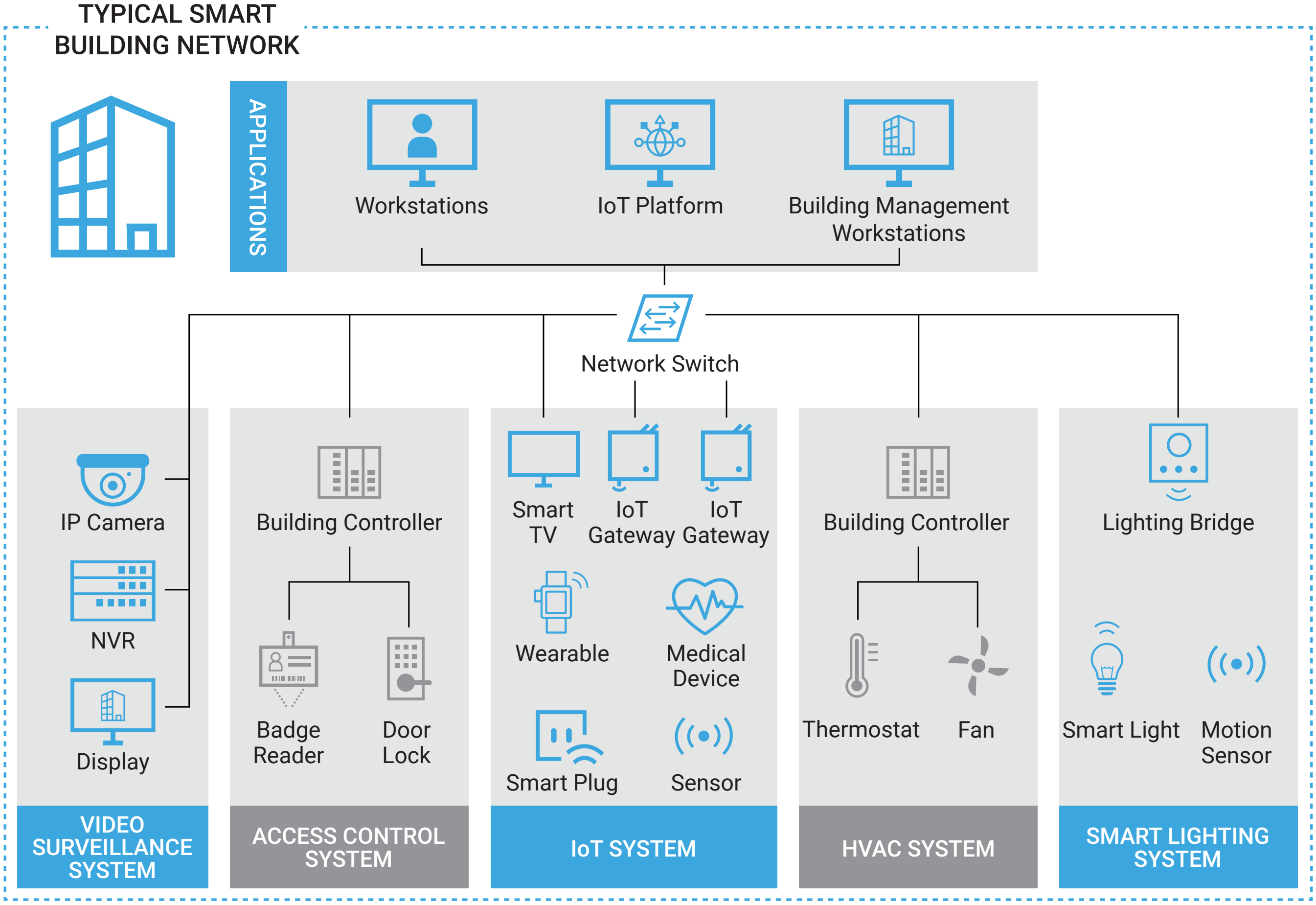


Figure 1 - Building automation network with IoT devices

The OT devices in the different subsystems use either proprietary or standard domain-specific protocols such as BACnet, KNX, and LonTalk [42] to communicate. More recently, IoT devices like smart lights, smart locks, smart electrical plugs, and other sensors and actuators started being deployed alongside building automation systems [43] using protocols such as Message Queue Telemetry Transport (MQTT) and the Constrained Application Protocol (CoAP) to achieve machine-to-machine (M2M) communication and establish a common message bus (see modern building automation controllers adopting MQTT for data exchange [44] [45]).

The details of each system are abstracted in Figure 1, but in the remainder of this section, we will detail the three systems of interest in this report (shown in blue in the Figure).

3.1 Surveillance System

The precursors of modern video surveillance are Closed-Circuit Television (CCTV) systems, which use analog signals and coax cables to communicate in a closed network. As technology advanced, digital cameras supporting IP communication came into existence and got integrated into VSSs. Nowadays, video surveillance with IP cameras is used not only in large corporations and highly secure locations, but also in most public buildings and increasingly in private home automation systems [46] [47].

Modern video surveillance systems are composed of the following main components:

1. *Cameras*, which provide video monitoring of physical locations. They can be grouped into CCTV (analog) and IP (digital) cameras, which, as opposed to their analog versions, can be directly connected to an Ethernet network. In this work, our focus is on IP cameras only.
2. *Recorders*, which store camera footage. Analog cameras use a Videocassette Recorder (VCR) or Digital Video Recorder (DVR), while IP cameras use a software or dedicated device that records and stores video in a digital format, called a Network Video Recorder (NVR). Some advanced IP camera models also integrate Video Management software (VMS) for local storage of recorder footage.
3. *Monitors*, which are used to watch real-time or recorded footage. Monitors can also be analog or digital, such as a computer, smartphone or almost anything with a screen that can display video.

More complex systems can also contain media servers, gateways, routers and switches. Based on the components present on a VSS network, we can differentiate three types of surveillance systems:

1. Analog systems contain devices that cannot communicate on the Ethernet network. They are much less prone to cyberattacks and are out of the scope of this report.
2. Digital systems comprise IP cameras, NVRs, switches, routers, and digital monitors, which all can send and receive Ethernet network traffic. Most of these devices also support remote access, maintenance, and alerting via HTTP, FTP, SSH, SMTP, and similar protocols, and in some cases, also the old and insecure Telnet protocol. Video streaming uses RTP, RTCP, and RTSP, as explained below.
3. Hybrid systems comprise both digital and analog devices. Besides the devices mentioned above, these systems can also contain video encoders or hybrid DVRs to connect analog cameras to the IP network and video decoders to view the digital data on analog monitors.

The architecture of a hybrid video surveillance system can be quite complex, containing a variety of legacy and new technologies. Figure 2 shows an example of such a system, where the direction of the arrows indicates the direction of communication.

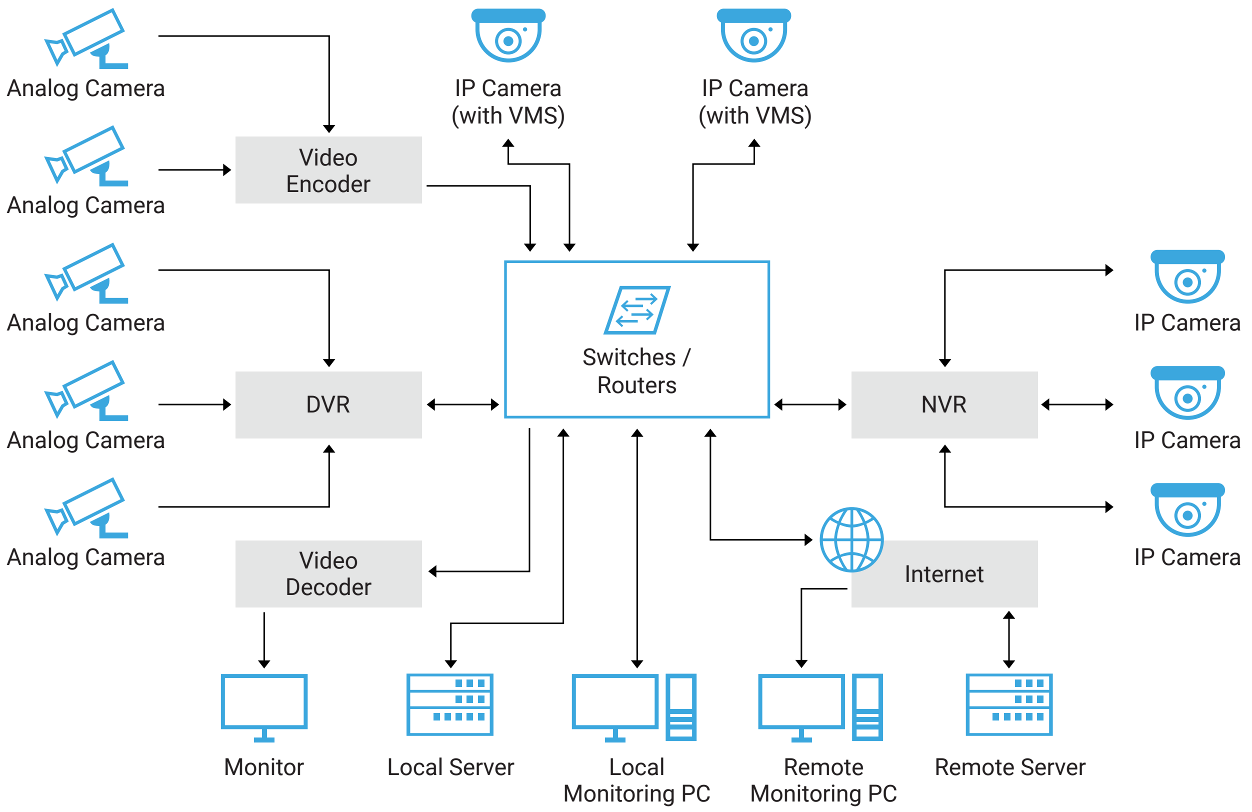


Figure 2 - Surveillance system architecture as found in a modern building

VSS devices, unlike others found in smart building networks, need to cope with real-time transfer of large amounts of data. For that reason, dedicated protocols are used, the most popular of which are RTP, RTCP, and RTSP.

RTP has two versions [48] [49] and is used for real-time transfer of streaming data, such as audio or video. The data transport is augmented by a control protocol (RTCP) to allow monitoring of data delivery and minimal control and identification functionalities. RTP and RTCP are designed to be independent of the underlying transport and network layers but are usually run on top of UDP to ensure a stable streaming even in the case of some packet loss. There are secure versions of RTP, called SRTP, and RTCP, called SRTCP [50], which provide confidentiality, authentication, integrity and replay attack protection. Our extensive experience dealing with the surveillance networks of large corporations shows that these secure variants are rarely used in real-world deployments.

RTSP also has two versions [51] [52], although the first is still the most widely used. RTSP is a text-based protocol, with a syntax that resembles HTTP, supporting commands such as PLAY, PAUSE, and TEARDOWN to establish and control media sessions between client and server endpoints, such as IP cameras and NVRs. RTSP typically uses TCP as the transport protocol and relies on RTP for delivering the media stream. Currently, RTSP does not natively support stream encryption. This means that the packets can be easily sniffed and tampered with by a malicious actor on the network. A viable workaround to this is to tunnel the RTSP traffic through an encrypted Transport Layer Security (TLS) stream. However, as mentioned above, this is rarely applied in practice.

In Table 1, the existing RTSP commands are grouped based on their allowed direction: C → S are commands from client to server; S → C from server to client; and S → C are commands that can be sent from both the client and the server.

C → S	S → C	S ↔ C
PLAY, PAUSE, DESCRIBE, RECORD, SETUP, TEARDOWN	REDIRECT	ANNOUNCE, GET_PARAMETER, OPTIONS, SET_PARAMETERS

Table 1 - Available RTSP commands

Most RTSP commands require authentication and, similarly to HTTP, RTSP supports two modes of authentication:

1. In basic mode, the username is concatenated with a colon and with the password and then encoded in base64, which can easily be decoded to get the original value back.
2. In digest mode, an authentication token is calculated by MD5-hashing the username and password, as well as the issued command, URL, and nonces, to prevent replay attacks.

3.2 Smart Lighting System

Smart lighting systems are lighting systems connected to a network, which allows them to be monitored and controlled from a central system or via the cloud [40]. These systems use automated control to switch on, dim, switch off or change the colors of lights based on conditions such as occupancy or daylight availability, thus increasing energy efficiency, improving working conditions and optimizing space utilization in a building. Energy savings with smart lighting can reach 70% compared to conventional lighting [40].

Network protocols used in lighting systems can be wired, using something like the popular Digital Addressable Lighting Interface (DALI), or wireless. Wireless protocols are gaining popularity due to easier installation and improved controls. The most common wireless technologies for lighting include ZigBee, Bluetooth, Wi-Fi, and EnOcean [53].

Currently, one of the most popular smart lighting systems is the Philips Hue, which provides easy installation, user-friendly interaction and many third-party applications [54]. Philips Hue was introduced in October 2012, as one of the first IoT devices that could be controlled with a smartphone. The Hue system is composed of at least a Smart Bridge and a set of light bulbs, but it can also contain other elements like motion sensors.

The architecture of a Hue system is depicted in Figure 3. The smart lights do not require a connection to the network for basic functions. Even when they are offline, they can be used as regular bulbs and controlled by a classic switch. For smart functions, monitoring and control, the lights and other devices communicate with a bridge using the ZigBee Light Link (ZLL) protocol. The bridge must be connected to a network router. Communication between a control device and the bridge is done via Ethernet (usually, Wi-Fi), while the bridge translates requests to ZLL commands.

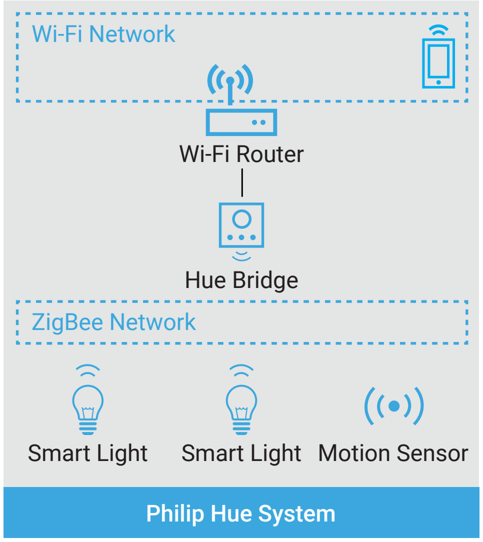


Figure 3 - Philips Hue network architecture

Alternative smart lightning systems such as LIFX [55] connect directly to the Wi-Fi router without the need for a bridge device. However, with the Philips Hue architecture, the smart bulbs are controlled in a centralized manner, which makes it a better choice for larger smart buildings.

> The IoT System in a smart building can connect edge devices in different subsystems.

3.3 IoT System

The IoT System in a smart building can connect edge devices in different subsystems, including the ones mentioned thus far, enterprise devices like VoIP phones and teleconference systems, and even personal devices such as wearables and smartphones.

Besides the edge devices, which can collect data and act on the environment, two important components of this system are:

1. IoT gateways, which can aggregate data and allow edge devices to communicate.
2. An IoT platform that enables the provisioning of different services by processing the collected data and controlling the edge devices.

The IoT system relies on many communication (Ethernet, WiFi, Bluetooth, Z-Wave) and messaging (MQTT, CoAP AMQP, DDS, XMPP) standards [11]. In this report we will focus on the messaging (i.e. application) layer and more specifically on MQTT, since it is the most used protocol to implement IoT systems [11].

MQTT is an M2M connectivity protocol, designed to be a lightweight, publish-subscribe messaging protocol working on top of TCP/IP [56]. MQTT defines a star topology managed by a central broker, which connects several client devices. A client can be either a publisher of information or a subscriber, with information organized in a hierarchy of topics following a URL-like structure (e.g., building/floor1/room2/temperature). When there is new information to be distributed, a publisher sends the topic and the data to the broker, which distributes the information to all nodes that have subscribed to that topic. MQTT is used not only to share telemetry information, but also for basic control of devices in some cases, like switching lights on or off and opening or closing doors. This topology of an MQTT network is shown in Figure 4.

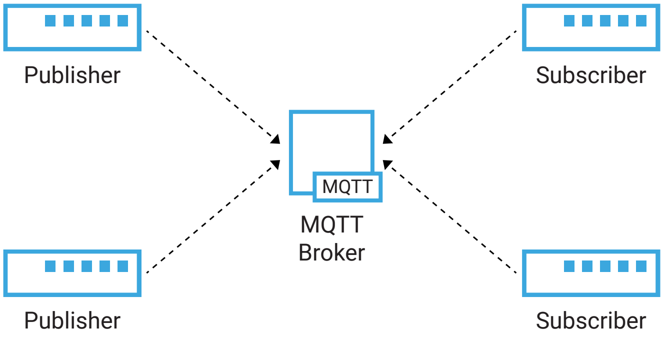


Figure 4 - MQTT topology

MQTT supports authentication via username and password pairs, but no encryption since it is meant to be lightweight. Therefore, it is recommended to use TLS on MQTT communications, since unencrypted traffic may disclose sensitive information, including topics, values of data points or even credentials. In practice, similarly to other IoT protocols, it is possible to find thousands of MQTT servers online with no authentication disclosing sensitive information, as well as allowing remote control, to any client who remotely subscribes to a topic [57] [58].

4. Security Challenges

The security challenges presented in the Introduction (growing number of devices, vendors, operating systems, and protocols leading to an increased attack surface) are exacerbated by the fact that IoT systems, including devices, gateways, and platforms, are notoriously vulnerable to cyberattacks [59] [60] [61] [62] [63]. The OWASP organization, for instance, maintains well-known lists of top vulnerabilities [64] and attack surface areas [65] in the Internet of Things.

Besides the infamous exploitation of default or insecure credentials [33], which affect devices as critical as networked refrigerators in medical facilities [66], attacks against IoT systems include:

1. **Web application and API attacks**, including database and command injections, directory traversal, and cross-site scripting. This category encompasses well-known attacks that have been used against web applications over the last couple of decades. They represent the low-hanging fruit for an attacker targeting an Internet-connected IoT device and can be performed in a semi-automatic fashion using available open source tools. Many of the vulnerabilities allowing these attacks can be found easily by using standard security scanners [67].
2. **Lower-level exploits** against device firmware, such as buffer overflows or memory corruption issues that can either disable the device or allow arbitrary code execution. Exploits for this category usually require relevant binary reverse engineering skills and low-level knowledge (assembly language, CPU instructions) to be conducted. Therefore, real-world attacks based on such exploits are more difficult to perform than those in the first category, although potentially more damaging.
3. **Protocol-based attacks** exploiting vulnerabilities like lack of authentication, encryption, and integrity validation. These attacks can be used by an attacker to easily sniff and exfiltrate or to tamper with sensitive data on the network.

In the last years, several vulnerabilities in these three categories have been discovered on devices from several popular vendors [29] [30] [31], but exploiting one or more vulnerabilities in the categories above is usually just the starting point for an attack campaign.

Malicious actors may leverage vulnerable IoT devices to penetrate the business network of a corporation and perpetrate criminal activities such as exfiltrating confidential data and dropping ransomware.

Malicious actors may leverage vulnerable IoT devices to penetrate the business network of a corporation and perpetrate criminal activities such as exfiltrating confidential data and dropping ransomware. Another common end goal is to use these devices as part of botnets that realize further attacks, such as a distributed denial-of-service (DDoS). This kind of attack became famous in 2016 with Mirai [33] [34] and is evolving to exploit vulnerabilities in IoT protocols to achieve amplification effects [68].

In this report, we want to draw attention to another aspect of IoT security: how an attacker, after establishing an initial foothold on a network, can disrupt the normal functioning of these devices, thus rendering systems such as video surveillance and smart lighting useless and allowing physical attacks to take place. Preventing these attacks is crucial to ensure effective protection and correct functioning of a smart building. A more detailed discussion is given in the next section, where several practical attacks are presented.

5. Three Simple Strategies to Tear Down a Building Network

To demonstrate in practice the exploitation of a smart building, we set up a lab (depicted in Figure 5) containing the three systems described in the previous sections: video surveillance, smart lighting, and IoT.

The VSS (left-hand side of Figure 5) contains three widely used IP cameras from top surveillance system vendors and a popular open-source NVR software. The Smart Lighting System (right-hand side) contains one Philips Hue Bridge, two smart lights, and a motion sensor. The IoT System (in the center) contains an IoT gateway implemented via an MQTT broker. We also used a Raspberry Pi to act as a local attacker.

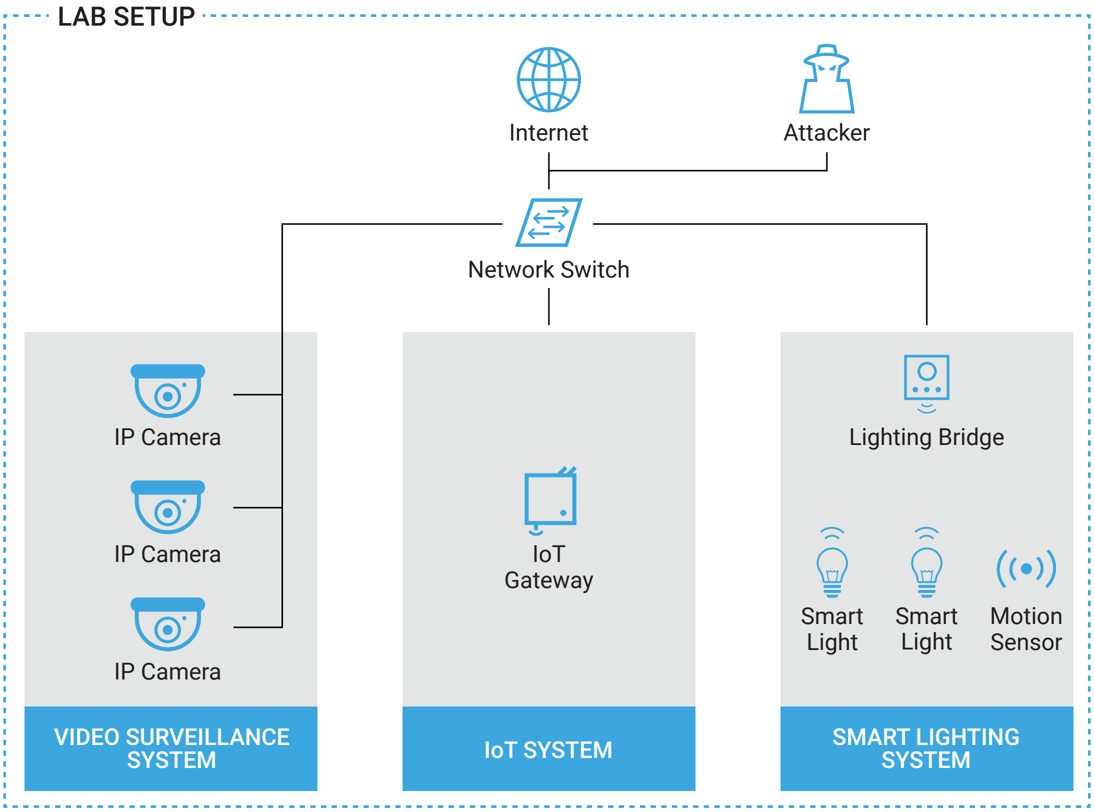


Figure 5 - Architecture of the lab

After setting up the lab, we then proceeded to analyze how an attacker could obtain initial access to this network and some of the attacks they could implement for each subsystem. We will present the results of this analysis in the remainder of this section.

5.1 Research and Reconnaissance

When we started analyzing the lab setup, the first thing we noticed was that some devices do not even support encrypted protocols for video streaming (SRTP), file transfer (SFTP) and web management (HTTPS), and those devices that support encrypted protocols do not suggest their use by default. The result is well known: many IoT devices are setup and managed with insecure protocols, allowing traffic sniffing and tampering, including sniffing credentials and sensitive information, including patient information in hospitals or video footage.

> Many IoT devices are setup and managed with insecure protocols, allowing traffic sniffing and tampering, including sniffing credentials and sensitive information.

Another issue that immediately caught our attention was the prevalence of severe bugs leading to remote code execution and complete takeover of the cameras. Two of the cameras we purchased for the lab, with the latest firmware versions installed, were found critically vulnerable (by another company) a few weeks after we started our research.

We then assumed that real deployments of IoT devices in the smart building networks of large corporations or critical facilities would be in a similar situation, containing devices that are either critically vulnerable, running insecure protocols, or both. Our assumption was confirmed by previous research on devices accessible online [69] [70], but especially by having access to the networks of customers with thousands of cameras. Among the many problems we found were:

- Unwanted communication links between the IT network and the VSS caused by firewall misconfiguration.
- Unwanted services and insecure protocols enabled (e.g., FTP and UPnP).
- Weak passwords to access IP cameras.
- Vulnerable cameras.

These findings are worrying because, besides the obvious attack paths created by cameras and IoT devices with remote code execution vulnerabilities or weak/default passwords, vulnerable UPnP implementations have been used to proxy malicious traffic [71] and expose vulnerable machines [72]. In fact, the security risks and many attacks leveraging UPnP have been known for more than a decade [73].

5.2 Obtaining Access

As in virtually every network, there are three main ways that an attacker can obtain access to a smart building such as the one simulated in our lab setup:

1. Via an externally reachable device that is vulnerable (e.g., remote code execution or weak/default credentials)
2. By tricking a user to give external access with a reverse shell (e.g., via phishing or an infected USB key)
3. By placing a rogue device (like a Raspberry Pi) in the network, which usually involves bypassing physical security

Option 1 was validated by our analysis above and, although we did not test options 2 and 3 in real scenarios, they have been known to work in environments as secure as hospitals and critical infrastructure facilities [74] [75] [76].

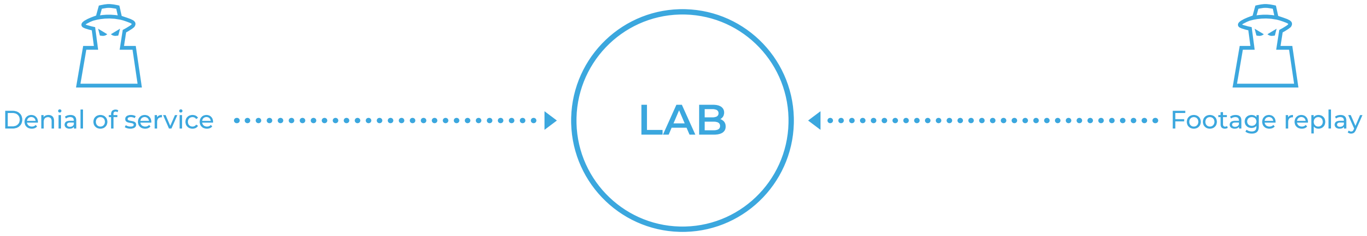
Having confirmed that obtaining initial access to a smart building network is not unrealistic and that the usage of insecure protocols is common in real-world systems, in our lab setup we adopted an attacker model that assumes an attacker already inside the network and that has the ability to sniff and, when necessary for the attack, modify packets in the network, essentially acting as a man-in-the-middle (MitM) [77].

The most popular way of achieving MitM in a network is via ARP poisoning (also known as ARP spoofing). The Address Resolution Protocol (ARP) is used by devices in a network to resolve IP addresses to physical MAC addresses. ARP poisoning exploits the lack of authentication in the protocol by sending spoofed messages to the network with the goal of associating the attacker's host MAC address with the IP address of a target host. This can be achieved automatically using tools such as ettercap [78].

ARP poisoning can be easily detected in the network if security monitoring tools are deployed (although nowadays it is known to be used by a few IoT devices, such as Disney Circle [79] and CUJO [80] to implement "security" features), but a much more silent approach to MitM can be achieved by exploiting vulnerabilities in routers [81] [82].

5.3 Abusing Camera Streams

For the video surveillance system, we wanted to demonstrate how easy it is for an attacker to exploit insecure streaming protocols to prevent the system from displaying the correct footage to an operator, a scenario that is commonly seen in heist movies. So, we devised and implemented two types of attacks against our lab: *denial of service* and *footage replay*. Notice that even though the attacks were carried out against specific products used in our lab, they only leverage weaknesses of the streaming protocols, which means they can be applied against many other similar setups.



We focus on attacks targeting the VSS via network protocols, instead of attacks that leverage the VSS as the source of further compromises or attacks that compromise specific camera models (e.g., code execution vulnerabilities) for the following reasons:

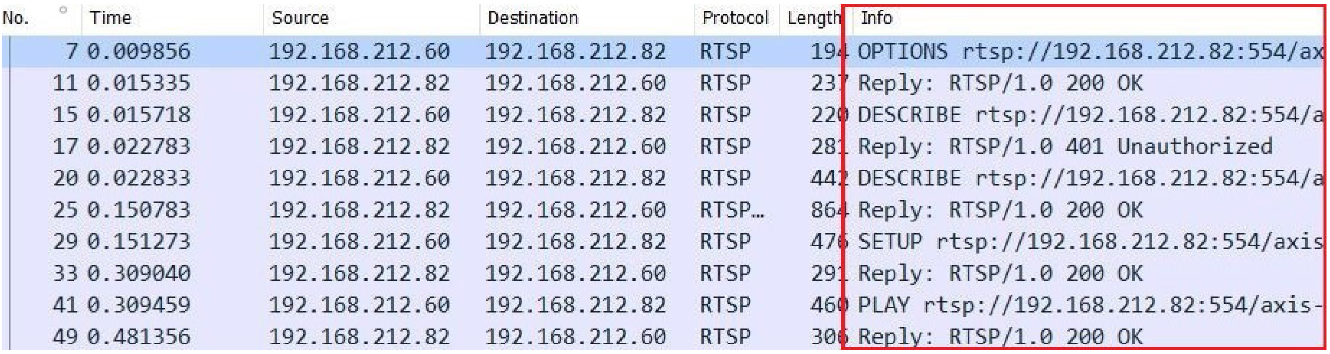
- We have another report which demonstrates how the exploitation of an IP camera can lead to a compromise of the whole building automation network [20].
- There are already a plethora of works describing vulnerabilities for specific IP camera and NVR models (see [29] [30] [31] [32] [83]). We want to demonstrate the concrete effects of an attack on the VSS, which are often neglected. Even if the attacker has a remote code execution exploit for a camera or NVR, simply taking that device offline or using it for further compromise may not be their goal.

The last point is crucial, especially for highly secured facilities and critical infrastructure buildings like airports, data centers, military facilities, etc. In these locations, a compromise of the VSS could be only the first step of a physical intrusion. The attacks described below are inspired by this scenario, where criminals hack the feed of a surveillance camera to stop recording or loop old footage to allow them to perform malicious actions without being recorded.

Denial of Service

The goal of these attacks is to prevent the VSS from displaying, recording, and storing camera footage by abusing either RTSP or RTP traffic.

When the NVR tries to establish a connection with a camera, it issues a sequence of RTSP commands: `OPTIONS, DESCRIBE, SETUP,` and **`PLAY`**. Figure 6 exemplifies this sequence in our lab setup. The **`DESCRIBE`** command occurs twice because it's the first in the sequence to require authentication. In other cameras, the **`OPTIONS`** command may require authentication, and therefore occur twice.



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 6 - RTSP setup sequence

Interfering with any of these messages prevents the NVR from successfully establishing a connection with a camera. Some examples of this interference that we implemented are:

1. Drop a command request – as the request does not reach the camera, it will not send a response. The NVR will keep reissuing the same request instead of proceeding with the sequence. Any command in the setup sequence can be dropped to achieve this result.
2. Tamper with a request – the attacker changes the requested port value in the `SETUP` request, thus the NVR will listen on a port different than the one where the camera is streaming, resulting in no footage being displayed.
3. Drop/tamper with a response – dropping any of the responses in the sequence or tampering with it by changing the success status `(200 OK)` to an unsuccessful one `(401 Unauthorized)` has a similar effect as the first attack.

The DoS attacks above target the setup sequence, which should only happen once, when the camera is first configured to work with the NVR. However, we can also terminate an ongoing session, thus forcing a new setup sequence. This can be done by exploiting the RTSP timeout defined in the response of the SETUP reply of the camera. The timeout parameter indicates how long 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 must send a periodical RTSP command (e.g., `GET_PARAMETER`) before the defined timeout. We implemented two attacks to terminate the session:

1. Drop the `GET_PARAMETER` request - this 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 to establish a new session to receive traffic again.
2. Replace the `GET_PARAMETER` command - replace the `GET_PARAMETER` command in a request with the `DESCRIBE` command, causing the camera to respond with status "`455 Method Not Valid in This State`", after which the NVR sends a `TEARDOWN` command to terminate the session and establish a new one. We can also replace the `GET_PARAMETER` command directly with `TEARDOWN`, so the camera will terminate the session and stop streaming immediately.

We can also attack RTP, instead of RTSP. Like the DoS attacks above, we can drop some packets to trick the NVR into terminating an ongoing session and initializing a new setup sequence. Instead of dropping packets, another attack is to inject RTP packets to flood the NVR, which leads to the unpredictable behavior described below:

- A frozen image from the original footage is seen on the NVR (shown in Figure 7).
- The streamed footage from the attacker machine is seen on the NVR (shown in Figure 8).
- A green image is shown because both streams interfere with each other (shown in Figure 9 and Figure 10).

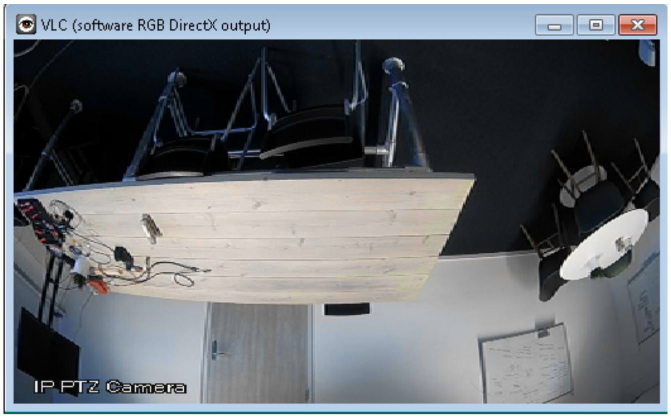


Figure 7 - Original footage

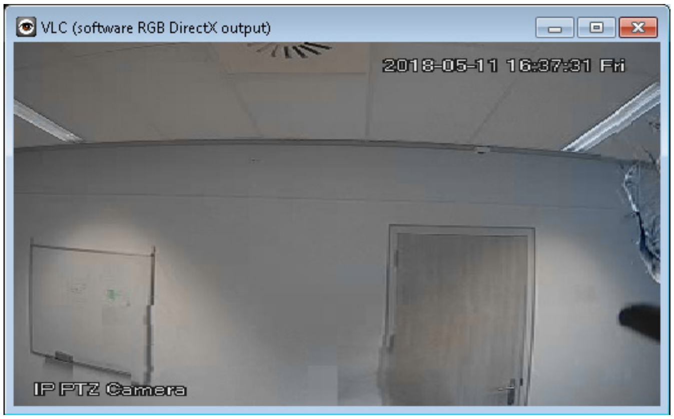


Figure 8 - Prerecorded footage

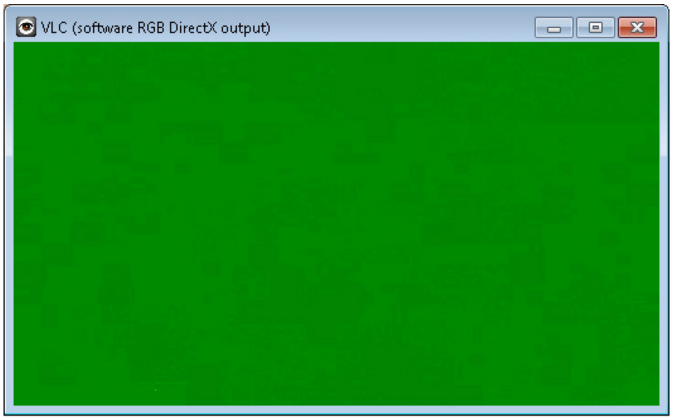


Figure 9 - Green footage 1

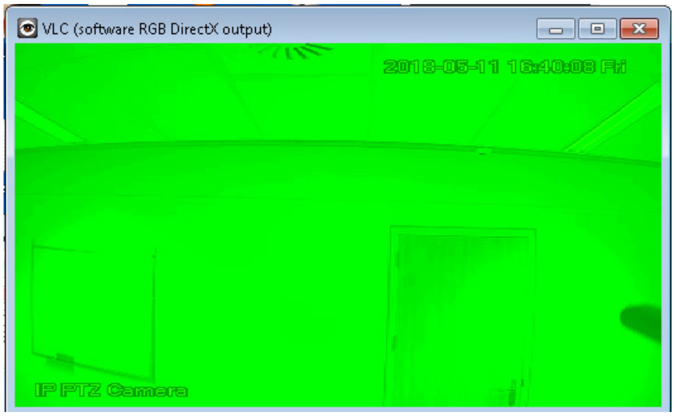


Figure 10 - Green footage 2

Footage Replay

The goal of this attack is to force the NVR to replay pre-recorded footage, instead of showing the real footage being streamed by a camera. This attack reuses some of the DoS attacks described above and is done in three steps:

1. Capture the network traffic containing camera footage and extract it for replay.
2. Force the camera to end a current session by changing a `GET_PARAMETER` to a `TEARDOWN` request, as described above.
3. When the NVR tries to establish a new session, capture the SETUP request and change the client port to a different one. This results in making the camera stream to the port specified by the attacker instead of the one initially requested by the NVR. After sending the `PLAY` command, the NVR will wait for traffic on the port which it specified in the `SETUP` request, but the camera will stream to a different port. Again, not receiving traffic will result into the NVR trying to set up a new connection, therefore there is only a limited time frame available to start streaming media to the correct port (the one initially specified in the `SETUP`) to show the pre-recorded footage.

The result of this attack can be seen on Figure 11 and Figure 12.

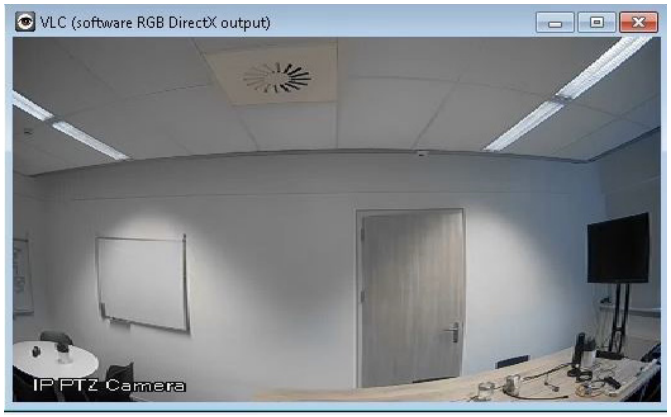


Figure 11 – Pre-recorded footage

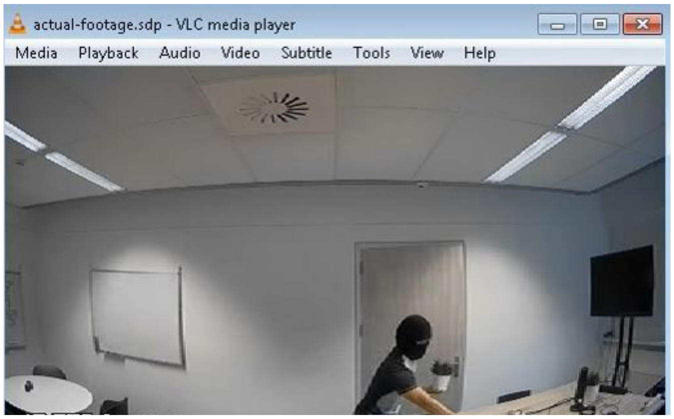
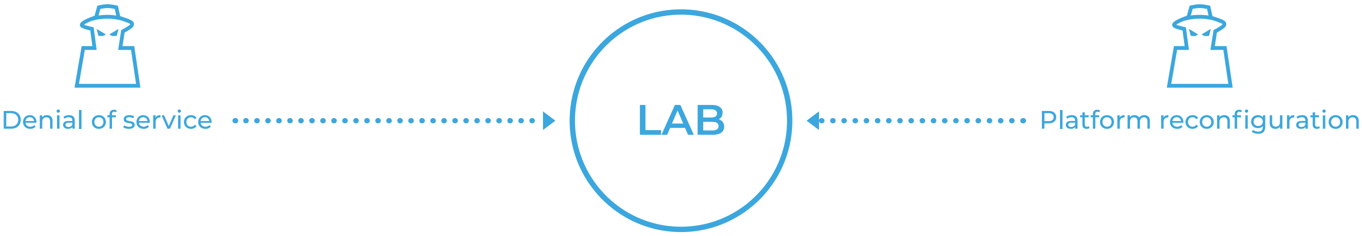


Figure 12 – Real footage

5.4 Exploiting the Philips Hue

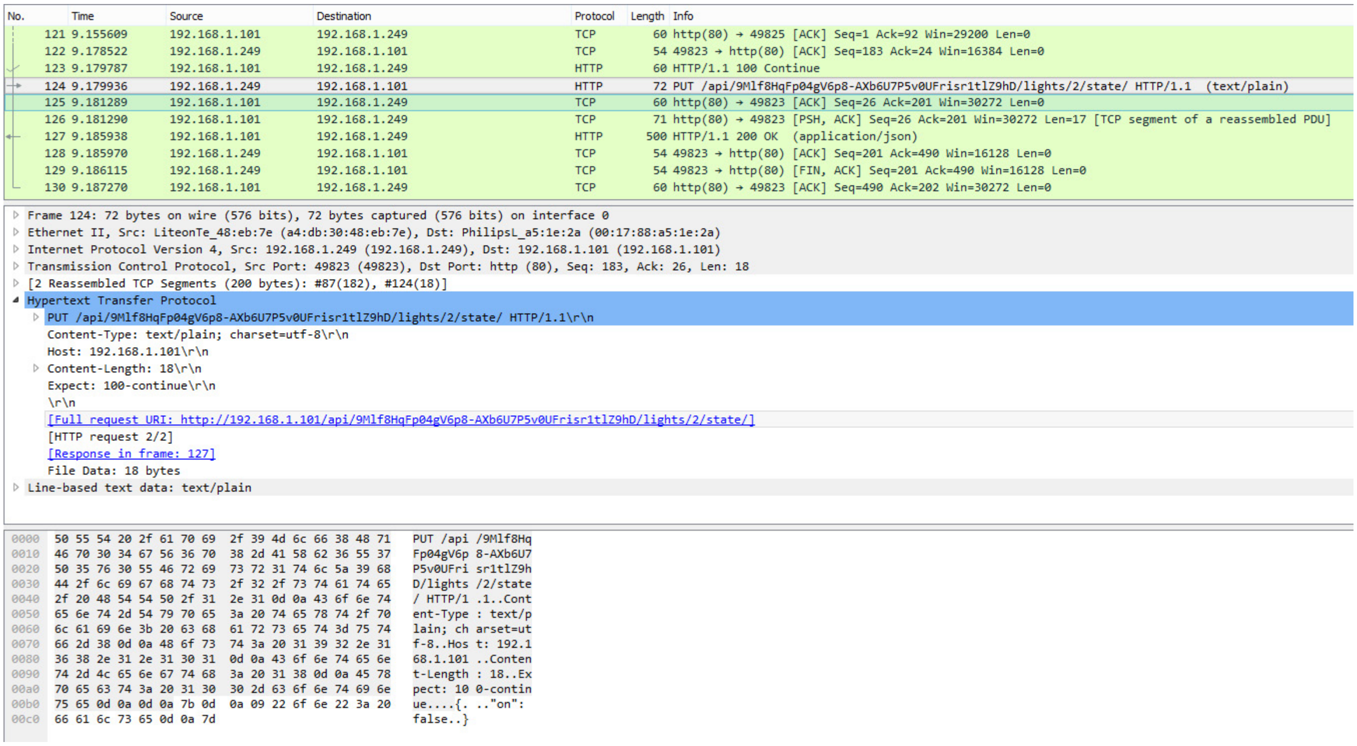
For the smart lighting system, we will focus on two kinds of attacks with a physical consequence: *denial of service* by switching off the lights and *platform reconfiguration*.



As described in section 3.2, the Hue system uses ZigBee communication between the bridge and the smart lights and Ethernet communication between a router and the bridge. We will focus on attacks leveraging the Ethernet network and ignore the ZigBee side, to be consistent with the attacker model we defined at the beginning of this section.

The Philips Hue supports an API that allows a user to interact with a bridge, and therefore the lights, using RESTful HTTP requests. The attacks we describe below are based on misusing this API for malicious purposes. Authentication in the API is handled by sending, with every request, a token that is generated when a user registers with the bridge. Malicious access can be achieved either by sniffing the network and capturing the token of an existing user or by registering a new user.

Hue authorization tokens are sent in cleartext with API requests, a vulnerability that has been known for a long time in the Hue system [84], so they can be copied by an attacker who has access to the network and can sniff traffic. Valid tokens can be seen in any authenticated request, which are of the form `http://<bridge_addr>/api/<token>/` where `<bridge_addr>` is the network address of the Hue bridge and `<token>` is the API token in cleartext. An example request with a valid token is shown in Figure 13, where the user token starts with `9Mlf`.



```
No.  Time      Source         Destination    Protocol Length Info
 121 9.155609  192.168.1.101  192.168.1.249  TCP       60 http(80) → 49825 [ACK] Seq=1 Ack=92 Win=29200 Len=0
 122 9.178522  192.168.1.249  192.168.1.101  TCP       54 49823 → http(80) [ACK] Seq=183 Ack=24 Win=16384 Len=0
 123 9.179787  192.168.1.101  192.168.1.249  HTTP      60 HTTP/1.1 100 Continue
 124 9.179936  192.168.1.249  192.168.1.101  HTTP      72 PUT /api/9Mlf8HqFp04gV6p8-AXb6U7P5v0UFrisr1tlZ9hD/lights/2/state/ HTTP/1.1  (text/plain)
 125 9.181289  192.168.1.101  192.168.1.249  TCP       60 http(80) → 49823 [ACK] Seq=26 Ack=201 Win=30272 Len=0
 126 9.181290  192.168.1.101  192.168.1.249  TCP       71 http(80) → 49823 [PSH, ACK] Seq=26 Ack=201 Win=30272 Len=17 [TCP segment of a reassembled PDU]
 127 9.185938  192.168.1.101  192.168.1.249  HTTP     500 HTTP/1.1 200 OK  (application/json)
 128 9.185970  192.168.1.249  192.168.1.101  TCP       54 49823 → http(80) [ACK] Seq=201 Ack=490 Win=16128 Len=0
 129 9.186115  192.168.1.249  192.168.1.101  TCP       54 49823 → http(80) [FIN, ACK] Seq=201 Ack=490 Win=16128 Len=0
 130 9.187270  192.168.1.101  192.168.1.249  TCP       60 http(80) → 49823 [ACK] Seq=490 Ack=202 Win=30272 Len=0

▷ Frame 124: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
▷ Ethernet II, Src: LiteonTe_48:eb:7e (a4:db:30:48:eb:7e), Dst: PhilipsL_a5:1e:2a (00:17:88:a5:1e:2a)
▷ Internet Protocol Version 4, Src: 192.168.1.249 (192.168.1.249), Dst: 192.168.1.101 (192.168.1.101)
▷ Transmission Control Protocol, Src Port: 49823 (49823), Dst Port: http (80), Seq: 183, Ack: 26, Len: 18
▷ [2 Reassembled TCP Segments (200 bytes): #87(182), #124(18)]
◢ Hypertext Transfer Protocol
  ▷ PUT /api/9Mlf8HqFp04gV6p8-AXb6U7P5v0UFrisr1tlZ9hD/lights/2/state/ HTTP/1.1\r\n
    Content-Type: text/plain; charset=utf-8\r\n
    Host: 192.168.1.101\r\n
  ▷ Content-Length: 18\r\n
    Expect: 100-continue\r\n
    \r\n
    [Full request URI: http://192.168.1.101/api/9Mlf8HqFp04gV6p8-AXb6U7P5v0UFrisr1tlZ9hD/lights/2/state/]
    [HTTP request 2/2]
    [Response in frame: 127]
    File Data: 18 bytes
▷ Line-based text data: text/plain

0000  50 55 54 20 2f 61 70 69  2f 39 4d 6c 66 38 48 71   PUT /api /9Mlf8Hq
0010  46 70 30 34 67 56 36 70  38 2d 41 58 62 36 55 37   Fp04gV6p 8-AXb6U7
0020  50 35 76 30 55 46 72 69  73 72 31 74 6c 5a 39 68   P5v0UFri sr1tlZ9h
0030  44 2f 6c 69 67 68 74 73  2f 32 2f 73 74 61 74 65   D/lights /2/state
0040  2f 20 48 54 54 50 2f 31  2e 31 0d 0a 43 6f 6e 74   / HTTP/1 .1..Cont
0050  65 6e 74 2d 54 79 70 65  3a 20 74 65 78 74 2f 70   ent-Type : text/p
0060  6c 61 69 6e 3b 20 63 68  61 72 73 65 74 3d 75 74   lain; ch arset=ut
0070  66 2d 38 0d 0a 48 6f 73  74 3a 20 31 39 32 2e 31   f-8..Hos t: 192.1
0080  36 38 2e 31 2e 31 30 31  0d 0a 43 6f 6e 74 65 6e   68.1.101 ..Conten
0090  74 2d 4c 65 6e 67 74 68  3a 20 31 38 0d 0a 45 78   t-Length : 18..Ex
00a0  70 65 63 74 3a 20 31 30  30 2d 63 6f 6e 74 69 6e   pect: 10 0-contin
00b0  75 65 0d 0a 0d 0a 7b 0d  0a 09 22 6f 6e 22 3a 20   ue....{. .."on":
00c0  66 61 6c 73 65 0d 0a 7d                            false..}
```

Figure 13 - Token sent in cleartext

To register a new user, the platform requires a physical button in the bridge to be pushed before a registration request is sent. Surprisingly, the button can be virtually "pressed" via the following HTTP request:

```
PUT http:/<bridge_addr>/<token>
{"linkbutton":true}
```

Although that request requires a valid token, which can be obtained via sniffing, as described above.

When the bridge authorizes a new application or user, it remains whitelisted until a factory reset is performed on the device. Assuming the attacker has obtained a valid token using one of the methods above, we describe below a few malicious actions that can be taken.

Denial of Service: Switching Off or Flickering the Lights

To switch off a specific light, a user can send the following HTTP request, which requires a valid token

```
PUT http://<bridge_addr>/api/<token>/lights/<number>/state
{"on":false}
```

Where <number> is an integer identifying the light bulb to be switched off.

The request above can be automated with a scripting language like Python, allowing an attacker to perform malicious actions on a loop, thus denying the user the possibility of using the lighting system. An example of this automated exploitation is the following code:

```
import json, requests, time
url = “http://<bridge_addr>/api/<token>/lights/<number>/state”
payload = {“on”:”false”}
headers = {“content-type”:”application/json”}

while True:
 r = requests.put(url, data=json.dumps(payload), headers=headers)
 time.sleep(2)
```

This switches off one specific light every two seconds. This example could also be extended to switch off every light by varying the `<number>` identifier using another loop on the number of lights available in the system.

Another possible attack that renders the system unusable is to blink the lights by abusing the “alert mode” functionality. In this case, the attacker changes the payload on the requests above from `{“on”:”false”}` to `{“alert”:”lselect”}`.

Platform Reconfiguration

The network configuration of the bridge can be changed with the following HTTP request, which requires a valid token:

```
PUT http://<bridge_addr>/api/<token>/config
{“ipaddress”:<ip_addr>, “dhcp”:false, “netmask”:<netmask>, “gateway”:<gtw>}
```

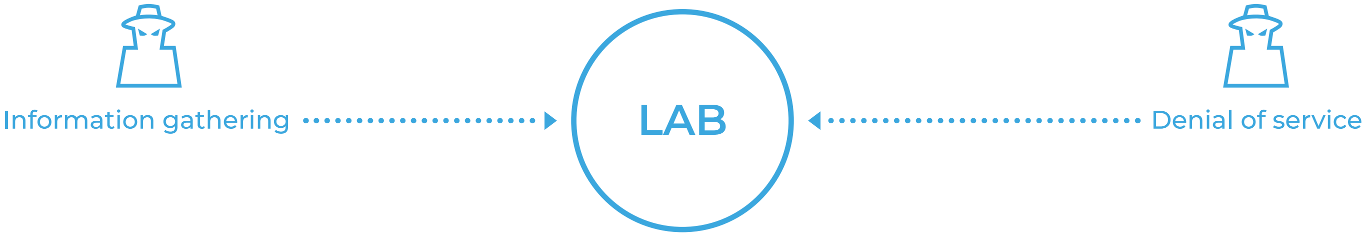
Where the attacker can set their desired values for `<ip_addr>`, `<netmask>` and `<gtw>`.

Depending on the network where the device is located, this may allow the attacker to set a public IP for the device, thus enabling remote access via the Internet and using the bridge as an entry or pivot point into the smart building network.

*According to Signify (former Philips Lighting), HTTPS support to the Hue bridge was added in September 2018 through a firmware update. However, for backward compatibility, HTTP support has not been dropped just yet. A sunset date for legacy applications will be likely announced next year. In addition, with a firmware update, the “virtual button press” was disabled in April 2019. Since then the button’s state can no longer be set using the rest API. For completeness, note that the Hue bridge itself uses HTTP for communication with the Philips device management cloud to provide support for a broad range of devices/appliances (some with very restricted resources). This means protection for device management is not offered on the link layer but has to be taken care of at application layer.

5.5 Attacking an IoT System

Like the attacks on the video surveillance system, for the case of the IoT system, we will describe attacks leveraging a protocol (MQTT), rather than specific devices. We describe below two kinds of attacks, *information gathering* and *denial of service*. Although we explain these attacks step-by-step, nowadays there are automated tools developed for penetration testing that can launch these and similar attacks on MQTT [85] and other protocols like CoAP [86].



Information Gathering

The goal of this attack is to gather information about an IoT network, which can include available assets and their location, configuration information or even sensitive information like credentials.

Besides passively sniffing traffic and sampling topics over time, MQTT allows any authorized client to subscribe to a topic or publish their own topic. In most networks, clients can also subscribe using wildcards that match existing topics in the broker. There are two types of wildcards on MQTT:

Multiple Level (#): Refers to all the topics under a level of the tree. For instance, a subscription to `/gfloor/#` will subscribe to `/gfloor/kitchen/temp`, `/gfloor/kitchen/humidity` and `/gfloor/livingroom/temp` but not to `/1floor/kitchen/temp`

Single Level (+): Refers to all the topics of a single level of the tree sharing the same termination. A subscription to `/gfloor/+/temp` will subscribe to `/gfloor/kitchen/temp` and `/gfloor/livingroom/temp` but not to `/gfloor/kitchen/humidity`, `/1floor/kitchen/humidity` and `/gfloor/kitchen/fridge/temp`.

Subscribing with wildcards allows an attacker to obtain information even without knowing the available topics beforehand.

Denial of Service

MQTT is usually deployed over TCP, which requires acknowledgment packets that can exhaust the resources of a device if enough simultaneous requests are sent (especially considering that some MQTT clients are very resource-constrained). An MQTT broker can be efficiently flooded by using CONNECT packets, which require more resources than typical message packets, since the broker must decide whether the client can establish the connection or not. Both clients and brokers can also be flooded by using heavy payloads, since MQTT supports payloads of up to 256MB.

DoS attacks can be enhanced by requiring higher Quality of Service (QoS) levels. MQTT supports QoS levels from 0 to 2. Level 0 allows the client to send an MQTT packet without requiring an acknowledgment (only TCP guarantees are assumed). Level 1 requires an acknowledgment for every request of a client. Level 2 requires that every packet is received only once by the other party, which means that received data is stored until it is guaranteed that the other party has received the message, then it is discarded to prevent duplicates.

The effects of a denial of service depend on the devices connected to the IoT system, but it can be used to prevent measurements from a sensor (e.g., temperature, humidity, motion, presence) from reaching a target device, thus rendering a desired action impossible (e.g., switching on/off a fan, light, alarm, window blind).

6. Detecting Vulnerabilities and Attacks

Network monitoring is an effective solution to identify and mitigate vulnerabilities in devices and the consequences of attacks like those presented in the previous section. Forescout's eyeInspect (formerly SilentDefense) is an advanced network monitoring and intelligence platform used by critical infrastructure and building automation operators worldwide to preserve the stability of their networks.

Network monitoring is an effective solution to identify and mitigate vulnerabilities in devices and the consequences of attacks like those presented in the previous section.

eyeInspect continuously monitors and analyzes network communications, compares them with a baseline of legitimate/desired operations and with the "known bad" defined in a collection of checks called the Industrial Threat Library, and reports problems and threats to the network in real time. Some examples include:

- Attempted and ongoing intrusions
- Misbehaving and misconfigured devices
- Undesired process operations
- Operational mistakes
- Known and *zero-day* attacks

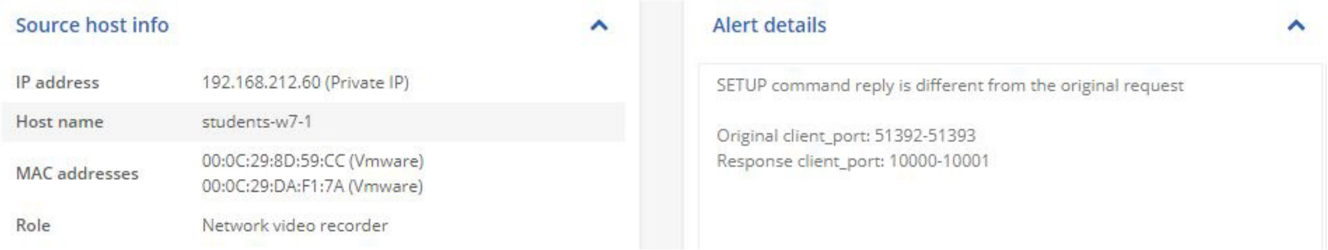
These threats are detected and presented to the operator in two main formats:

Visual analytics: The operator benefits from a visual representation of all aspects of the network with different types of graphs and charts. These graphs and charts are preconfigured to obtain at-a-glance insights into the most relevant aspects of current network activity and can be fully customized by the operator to obtain different views. In fact, the visual analytics platform is built on top of a full-fledged data warehouse, which means that the operator is able to query and represent areas of interest in the network at any moment in time. This lets them see what is currently happening, detect strange network behavior, and also analyze what happened in the past (to investigate a suspicious event).

Real-time alerts: As soon as something bad or unexpected occurs in the network, eyeInspect notifies the operator and provides them with all the intelligence required to respond to the event. This includes information about the source of the problem, the targeted device(s), the nature of the problem, and even a packet capture of the traffic related to the event. This traffic capture can become critical information to have if advanced threats, such as a zero-day attack, occur. This key data can then be forwarded to specialized security vendors and organizations for further analysis.

To detect the attacks presented in the previous section, eyeInspect uses the combined action of a powerful man-in-the-middle detection module and custom security checks. **These custom checks are capable of detecting attack signatures even without learning the normal behavior of the network, thus allowing for immediate response.**

Figure 14 and Figure 15 show examples of alerts raised by eyeInspect when an attempted exploitation of the footage replay attack is detected.

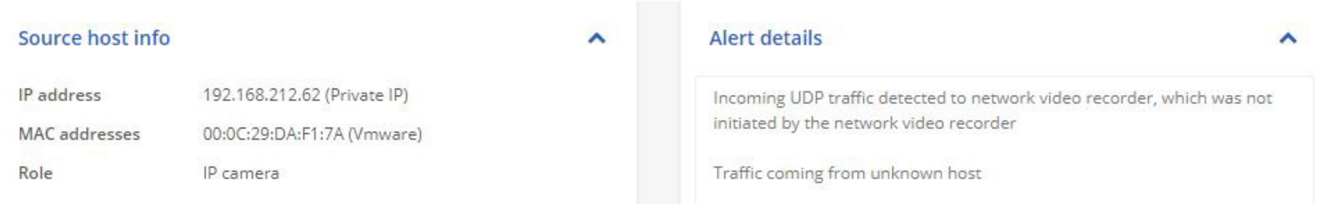


IP address	192.168.212.60 (Private IP)
Host name	students-w7-1
MAC addresses	00:0C:29:8D:59:CC (Vmware) 00:0C:29:DA:F1:7A (Vmware)
Role	Network video recorder

SETUP command reply is different from the original request

Original client_port: 51392-51393
Response client_port: 10000-10001

Figure 14 - Alert raised when RTSP change of ports is detected



IP address	192.168.212.62 (Private IP)
MAC addresses	00:0C:29:DA:F1:7A (Vmware)
Role	IP camera

Incoming UDP traffic detected to network video recorder, which was not initiated by the network video recorder

Traffic coming from unknown host

Figure 15 - Alert raised when an unknown host streams to an NVR

Additionally, by analyzing IoT network traffic, it is also possible to alert when dangerous operations or misconfigurations are detected. These actions, even though they can be ascribed to an expected maintenance window, define an anomalous behavior which might be a precursor for a subsequent attack. The list of dangerous operations/misconfigurations may include:

- Shutting down video streaming via RTSP TEARDOWN.
- Rebooting a device.
- Updating the firmware of a device.
- Using basic authentication mode to connect to a web interface.

> Network visibility and asset inventory are crucial to identify vulnerable network segments, ensure business continuity and improve incident response strategies for both industrial and building automation networks.

Finally, another important feature of eyeInspect's monitoring capabilities is improved network visibility by passively detecting and classifying hosts seen on the network. Figure 16 shows an example of some hosts used in our lab setup as they are detected and classified along with their network links by eyeInspect. Network visibility and asset inventory are crucial to identify vulnerable network segments, ensure business continuity and improve incident response strategies for both industrial and building automation networks. This holds especially true for smart buildings serving large corporations, since the network can contain several thousand assets distributed across multiple sites all over the world.

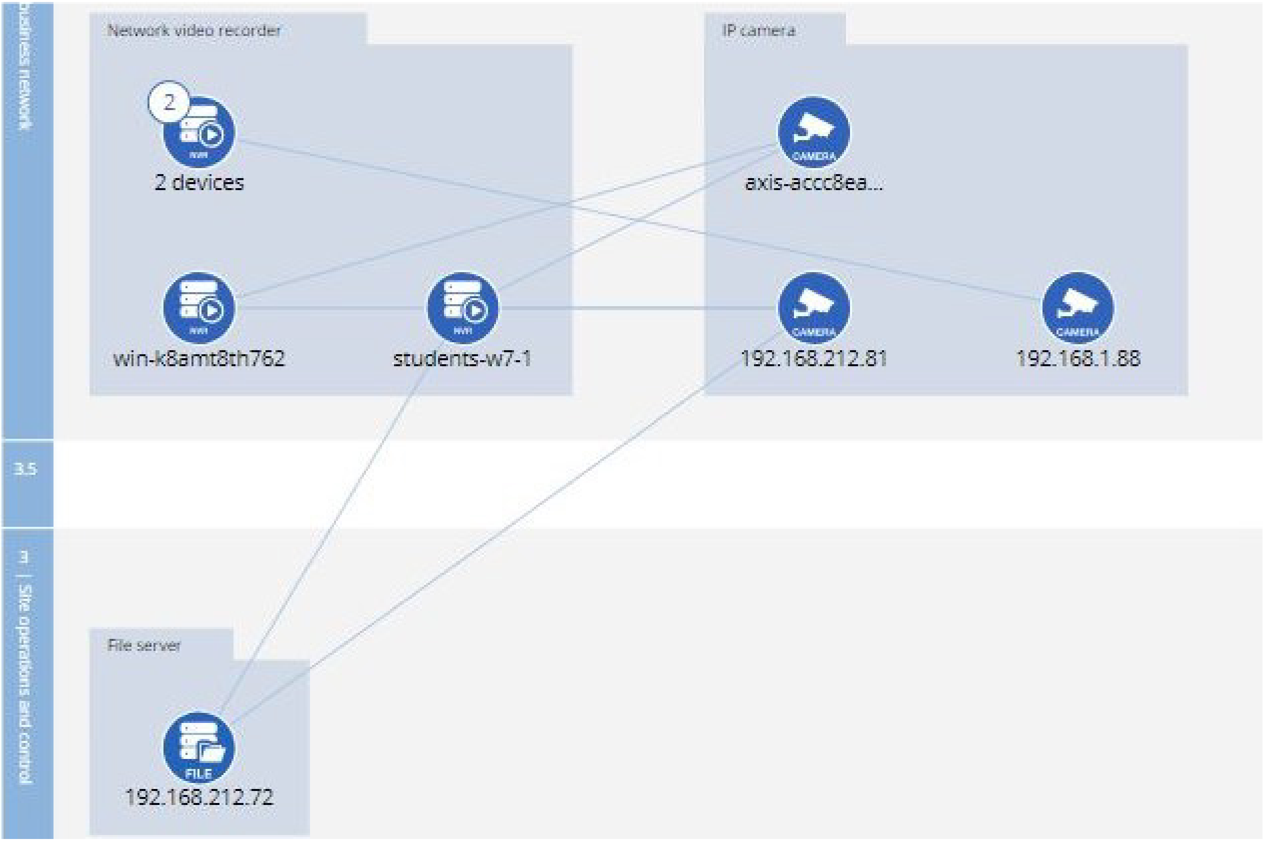


Figure 15 - Alert raised when an unknown host streams to an NVR

7. Discussion: Transforming Cybersecurity Strategy for the Age of IoT

This report analyzed the threat landscape for the IoT using smart buildings as a case study, demonstrated how easy it is to perform practical attacks on these systems and described how eyeInspect, a leading network monitoring and threat detection tool for critical infrastructure, can be used to detect these and other similar threats.

> The main takeaway message is the relative ease with which malicious actors could disrupt the normal functioning of a smart building with state-of-the-art components, but without proper security controls in place.

With this work, we hope to have highlighted the insecurity of current IoT deployments, how important it is to implement security countermeasures, and how legacy, agent-based, and IT-focused, cybersecurity solutions are inappropriate in this new landscape. The main takeaway message is the relative ease with which malicious actors could disrupt the normal functioning of a smart building with state-of-the-art components, but without proper security controls in place.

Although we used smart buildings as a case study, our conclusions could be extended to other domains. Healthcare is an immediate example, which has been suffering with legacy systems, insecure protocols, and unsegmented networks for a long time [87] [88] [89] [90]. Attacks similar to the ones we demonstrated (leveraging the lack of encryption, authentication, and integrity in specialized protocols to tamper with legitimate data) can potentially lead to wrong diagnostics and treatments, thus directly or indirectly harming patients. Another case is automotive [91], where injection attacks may be used to take control of a car and accelerate or brake at the wrong time, thus leading to potentially fatal accidents. Other interesting domains include smart manufacturing [97] and intelligent transportation systems [98].

As pointed out in the Introduction and demonstrated throughout this report, cybersecurity strategy must be transformed to cope with the rise of the IoT. There are many activities that should be considered in a cybersecurity strategy, such as threat modeling, threat intelligence, vulnerability management, risk management, security reviews, and supply chain risks. However, we believe that **the cornerstones of an appropriate cybersecurity strategy for the age of IoT are device visibility, control, and orchestration,** since they are crucial enablers for other cybersecurity-related activities.

Visibility: Security must begin by knowing what is on the network. Complete visibility into IoT devices is key to identifying the attacks described in this report. Adding enhanced security with network monitoring can give organizations a thorough understanding of their IoT environment and its connections with other systems and with the outside world. This makes it easier to design effective security architectures, identify attack vectors, and resolve operational security issues including vulnerabilities, misconfigurations, access policy violations, and weak security controls.

Control: Beyond knowing the assets in the network and identifying attacks, security teams must be able to automate and orchestrate appropriate responses, as well as be able to prevent further issues by strengthening their security posture. The situational awareness enabled by complete visibility and the integration of security solutions allow for timely prioritization and proper action in response to identified events. On the other hand, as organizations define their next-generation security architectures for IoT and OT, segmentation plays a leading role. Unlike traditional devices, IoT and OT devices cannot be regularly patched or secured through agents. Hence, segmenting these devices into logical security zones is an essential risk-mitigation strategy.

Orchestration: Many organizations have dozens of security solutions that operate independently. This approach prevents coordinated, enterprise-wide security response and results in manual, inefficient processes that can’t scale to address the growth of IoT devices. Effective and efficient Security Orchestration, Automation, and Response (SOAR) depends on sharing contextual insight into devices, automating security workflows, and enabling automated response actions.

About the Authors

Daniel dos Santos holds a PhD in computer science from the University of Trento and has experience in security consulting and research. He is a researcher at Forescout, focusing on vulnerability research and the development of innovative features for eyeInspect.

Mario Dagrada holds a PhD in computational physics from the University Pierre Marie Curie in Paris and has experience in high performance software development, security and research. He is a researcher at Forescout, focusing on medical device security and the development of innovative features for eyeInspect.

Michael Yeh holds a joint master's degree in cybersecurity from the Technical University of Eindhoven and the Radboud University. He worked as an intern at Forescout during the development of this research project.

Martín Pérez Rodríguez has studied Computer Science & Engineering at the Universidad Politécnica de Madrid and the Technical University of Eindhoven. After his internship, he started working as a DevOps Engineer at Forescout.

Elisa Costante holds a PhD in computer science from the Eindhoven University of Technology. She is an expert in IT and OT security and privacy. As head of industrial and OT research at Forescout, she manages the internal and external research activities. Her tasks include the management of national and international projects, the planning of the research strategy and the supervision of the activities related to the development of prototypes of innovative features to be added to eyeInspect.

References

[1] Gartner, “2018 Strategic Roadmap for Integrated IT Security,” [Online]. Available: https://www.gartner.com/en/documents/3873972.

[2] J. McCann, G. Picco, A. Gluhak, K. Johansson, M. Törngren and L. Gide, “Connected Things Connecting Europe,” 2019. [Online]. Available: https://cacm.acm.org/magazines/2019/4/235602-connected-things-connecting-europe/.

[3] C. Greer, M. Burns, D. Wollman and E. Griffor, “Cyber-Physical Systems and Internet of Things,” 2019. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1900-202.pdf.

[4] R. Minerva, A. Biru and D. Rotondi, “Towards a definition of the Internet of Things (IoT),” 2015. [Online]. Available: https://iot.ieee.org/definition.html.

[5] Gartner, “Internet of Things,” [Online]. Available: https://www.gartner.com/it-glossary/internet-of-things/.

[6] ABI Research, Internet of Everything Market Tracker, QTR 3, 2018.

[7] P. Middleton, “Forecast Analysis: Internet of Things - Endpoints, Worldwide, 2017 Update,” Gartner, 2017. [Online]. Available: https://www.gartner.com/doc/3841268/forecast-analysis-internet-things-.

[8] K. Boeckl, M. Fagan, W. Fisher, N. Lefkovitz, K. Megas, E. Nadeau, D. O’Rourke, B. Piccarreta and K. Scarfone, “Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks,” 2018. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8228-draft.pdf.

[9] M. Hung, “Leading the IoT: Gartner Insights on How to Lead in a Connected World,” Gartner, 2017. [Online]. Available: https://www.gartner.com/imagesrv/books/iot/iotEbook_digital.pdf.

[10] K. Thielemann, “Focus More on the Realities of Cyber-Physical Systems Security Than on the Concepts of IoT,” Gartner, 2018.

[11] Eclipse IoT Working Group, AGILE IoT, IEEE, and Open Mobile Alliance, “IoT Developer Survey 2018,” 2018. [Online]. Available: https://iot.eclipse.org/resources/iot-developer-survey/iot-developer-survey-2018.pdf.

[12] Memoori, “The Collision of IT & OT is Shaping the Future of Buildings in the IoT Age,” 2018. [Online]. Available: https://www.memoori.com/collision-ot-shaping-future-buildings-iot-age/.

[13] A. Sadeghi, C. Wachsmann and M. Waidner, “Security and privacy challenges in industrial Internet of Things,” in Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference, 2015.

[14] F. Maggi, R. Vosseler and D. Quarta, “The Fragility of Industrial IoT’s Data Backbone,” 2018. [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/mqtt-and-coap-security-and-privacy-issues-in-iot-and-iiot-communication-protocols.

[15] F. Alsubaei, A. Abuhussein and S. Shiva, “Security and Privacy in the Internet of Medical Things: Taxonomy and Risk Assessment,” in IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops), 2017.

[16] M. Barcena and C. Wueest, “Insecurity in the Internet of Things,” Symantec, 2015. [Online]. Available: https://www.symantec.com/content/dam/symantec/docs/security-center/white-papers/insecurity-in-the-internet-of-things-15-en.pdf.

[17] S. Hilt, N. Huq, M. Rösler and A. Urano, “Cybersecurity Risks in Complex IoT Environments: Threats to Smart Homes, Buildings and Other Structures,” TrendMicro, 2019. [Online]. Available: https://documents.trendmicro.com/assets/white_papers/wp-cybersecurity-risks-in-complex-iot-environments.pdf.

[18] P. Ciholas, A. Lennie, P. Sadigova and J. Such, “The Security of Smart Buildings: a Systematic Literature Review,” 2019. [Online]. Available: https://arxiv.org/abs/1901.05837.

[19] NIST, “Security for IoT Sensor Networks,” [Online]. Available: https://www.nccoe.nist.gov/projects/building-blocks/iot-sensor-security.

[20] D. dos Santos, C. Speybrouk and E. Costante, “Cybersecurity in Building Automation Systems,” 2019. [Online]. Available: https://www.sec-matters.com/whitepaper-cybersecurity-in-building-automation-systems.

[21] G. Tay, C. Rozwell and D. Freeman, “Use the Internet of Things in Smart Buildings to Achieve Work-Life Ambience,” Gartner, 2017.

[22] Memoori, “Smart Buildings at the Center of a Fundamental Shift in Healthcare,” 2018. [Online]. Available: https://www.memoori.com/smart-buildings-at-the-center-of-a-fundamental-shift-in-healthcare/.

[23] Wired, “Wi-Fi passwords can be stolen by hacking smart lightbulbs,” [Online]. Available: http://www.wired.co.uk/article/crypto-weakness-lightbulbs.

[24] Bitdefender, “The IoT Threat Landscape and Top Smart Home Vulnerabilities in 2018,” 2018. [Online]. Available: https://www.bitdefender.com/files/News/CaseStudies/study/229/Bitdefender-Whitepaper-The-IoT-Threat-Landscape-and-Top-Smart-Home-Vulnerabilities-in-2018.pdf.

[25] C. Cimpanu, “Over nine million cameras and DVRs open to APTs, botnet herders, and voyeurs,” 2018. [Online]. Available: https://www.zdnet.com/article/over-nine-million-cameras-and-dvrs-open-to-apts-botnet-herders-and-voyeurs/.

[26] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz and J. Lopez, “A Survey of IoT-Enabled Cyberattacks: Assessing Attack Paths to Critical Infrastructures and Services,” IEEE Communications Surveys & Tutorials, vol. 20, no. 4, 2018.

[27] W. Ding and H. Hu, “On the Safety of IoT Device Physical Interaction Control,” in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018.

[28] S. Cobb, “Siegeware: When criminals take over your smart building,” 2019. [Online]. Available: https://www.welivesecurity.com/2019/02/20/siegeware-when-criminals-take-over-your-smart-building/.

[29] VDOO, “VDOO discovers significant vulnerabilities in Axis cameras,” 2018. [Online]. Available: https://blog.vdoo.com/2018/06/18/vdoo-discovers-significant-vulnerabilities-in-axis-cameras/.

[30] VDOO, “Significant Vulnerability in Hikvision Cameras,” 2018. [Online]. Available: https://blog.vdoo.com/2018/11/13/significant-vulnerability-in-hikvision-cameras/.

[31] VDOO, “Major Vulnerabilities in Foscam Cameras,” 2018. [Online]. Available: https://blog.vdoo.com/2018/06/06/vdoo-has-found-major-vulnerabilities-in-foscam-cameras/.

[32] Tenable Research, “Peekaboo: Don’t Be Surprised by These Not So Candid Cameras,” [Online]. Available: https://www.tenable.com/blog/peekaboo.

[33] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas and Y. Zhou, “Understanding the Mirai Botnet,” in Proceedings of the 26th USENIX Security Symposium, 2017.

[34] A. Costin and J. Zaddach, “IoT Malware: Comprehensive Survey, Analysis Framework and Case Studies,” in Blackhat Briefings USA, 2018.

[35] Wired, “The Reaper IoT Botnet Has Already Infected A Million Networks,” [Online]. Available: https://www.wired.com/story/reaper-iot-botnet-infected-million-networks/.

[36] E. Ronen and A. Shamir, “Extended Functionality Attacks on IoT Devices: The Case of Smart Lights,” in Proceedings of the IEEE European Symposium on Security and Privacy, 2016.

[37] P. Morgner, S. Mattejat and Z. Benenson, “All Your Bulbs Are Belong to Us: Investigating the Current State of Security in Connected Lighting Systems,” in Proceedings of the 10th ACM Conference on Security & Privacy in Wireless and Mobile Networks, 2017.

[38] E. Ronen, C. O’Flynn, A. Shamir and A. Weingarten, “IoT Goes Nuclear: Creating a ZigBee Chain Reaction,” in Proceedings of the IEEE Symposium on Security and Privacy, 2017.

[39] Limited Results, “Pwn the LIFX Mini white,” 2019. [Online]. Available: https://limitedresults.com/2019/01/pwn-the-lifx-mini-white/.

[40] B. Tratz-Ryan and B. Finnerty, “Hype Cycle for Smart City Technologies and Solutions,” Gartner, 2018.

[41] i-SCOOP, “IoT technology stack – from IoT devices, sensors, actuators and gateways to IoT platforms,” [Online]. Available: https://www.i-scoop.eu/internet-of-things-guide/iot-technology-stack-devices-gateways-platforms/.

[42] O. Hersent, D. Boswarthick and O. Elloumi, The Internet of Things: Key Applications and Protocols, Wiley, 2012.

[43] K. Walker, “The impact of the internet of things on buildings,” 2018. [Online]. Available: http://www.smartbuildingsmagazine.com/features/the-impact-of-the-internet-of-things-on-buildings.

[44] Tridium, “Niagara MQTT Architecture,” [Online]. Available: https://www.tridium.com/~/media/tridium/library/documents/software/niagara%204/mqtt%20architecture.ashx.

[45] Systec Electronic, “Industrial control in special applications,” [Online]. Available: https://www.systec-electronic.com/en/solutions/smart-buildings-industrial-building-automation/.

[46] J. Bugeja, A. Jacobsson and P. Davidsson, “An Empirical Analysis of Smart Connected Home Data,” in International Conference on Internet of Things, 2018.

[47] H. Guo and J. Heidemann, “Detecting IoT Devices in the Internet (Extended),” USC/Information Sciences Institute, 2018.

[48] IETF, “RFC 1889 - RTP: A Transport Protocol for Real-Time Applications,” [Online]. Available: https://tools.ietf.org/html/rfc1889.

[49] IETF, “RFC 3350 - RTP: A Transport Protocol for Real-Time Applications,” [Online]. Available: https://tools.ietf.org/html/rfc3550.

[50] IETF, “RFC 3711 - The Secure Real-time Transport Protocol (SRTP),” [Online]. Available: https://tools.ietf.org/html/rfc3711.

[51] IETF, “RFC 2326 - Real Time Streaming Protocol (RTSP),” [Online]. Available: https://tools.ietf.org/html/rfc2326.

[52] IETF, “RFC 7826 - Real-Time Streaming Protocol Version 2.0,” [Online]. Available: https://tools.ietf.org/html/rfc7826.

[53] Arrow Intelligent Systems, “Connectivity Protocols for Smart Lighting Systems,” [Online]. Available: https://static4.arrow.com/-/media/arrow/files/pdf/c/connectivityprotocolsforsmartlighting_final.pdf.

[54] Philips, “Philips Hue,” [Online]. Available: https://www2.meethue.com/en-us.

[55] LIFX, [Online]. Available: https://www.lifx.com/.

[56] “MQTT,” [Online]. Available: http://mqtt.org/.

[57] V. Pasknel, “Hacking the IoT with MQTT,” 2017. [Online]. Available: https://morphuslabs.com/hacking-the-iot-with-mqtt-8edaf0d07b9b.

[58] M. Hron, “Are smart homes vulnerable to hacking?,” Avast, 2018. [Online]. Available: https://blog.avast.com/mqtt-vulnerabilities-hacking-smart-homes.

[59] H. Xu and F. Xu, “Internet Protocol Cameras with No Password Protection: An Empirical Investigation,” in International Conference on Passive and Active Network Measurement, 2018.

[60] P. Vervier and Y. Shen, “Before Toasters Rise Up: A View into the Emerging IoT Threat Landscape,” in Research in Attacks, Intrusions, and Defenses, 2018.

[61] A. Costin, “Security of CCTV and Video Surveillance Systems:,” in Proceedings of the 6th International Workshop on Trustworthy Embedded Devices, 2016.

[62] D. Nitesh, Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts, O’Reilly, 2015.

[63] C. Bozzato, “Vulnerability Spotlight: Multiple Vulnerabilities in Samsung SmartThings Hub,” 2018. [Online]. Available: https://blog.talosintelligence.com/2018/07/samsung-smartthings-vulns.html.

[64] OWASP, “Internet of Things (IoT) Top 10 2018,” 2018. [Online]. Available: https://www.owasp.org/images/1/1c/OWASP-IoT-Top-10-2018-final.pdf.

[65] OWASP, “IoT Attack Surface Areas,” [Online]. Available: https://www.owasp.org/index.php/IoT_Attack_Surface_Areas.

[66] SafetyDetective, “Major Security Breach Found in Hospital and Supermarket Refrigeration Systems,” 2019. [Online]. Available: https://www.safetydetective.com/blog/rdm-report/.

[67] OWASP, “Open Web Application Security Project,” [Online]. Available: https://www.owasp.org/index.php/Main_Page.

[68] C. Cimpanu, “The CoAP protocol is the next big thing for DDoS attacks,” 2018. [Online]. Available: https://www.zdnet.com/article/the-coap-protocol-is-the-next-big-thing-for-ddos-attacks/.

[69] M. Patton, E. Gross, R. Chinn, S. Forbis, L. Walker and H. Chen, “Uninvited Connections: A Study of Vulnerable Devices on the Internet of Things (IoT),” IEEE Joint Intelligence and Security Informatics Conference, pp. 232-235, 2014.

[70] J. Bugeja, D. Jönsson and A. Jacobsson, “An Investigation of Vulnerabilities in Smart Connected Cameras,” IEEE International Conference on Pervasive Computing and Communications Workshops, pp. 537-542, 2018.

[71] Akamai, “UPnProxy: Blackhat Proxies via NAT Injections,” 2018. [Online]. Available: https://www.akamai.com/uk/en/multimedia/documents/white-paper/upnproxy-blackhat-proxies-via-nat-injections-white-paper.pdf.

[72] C. Seaman, “UPnProxy: EternalSilence,” Akamai, 2018. [Online]. Available: https://blogs.akamai.com/sitr/2018/11/upnproxy-eternalsilence.

[73] A. Hemel, “UPnP Hacks,” 2006. [Online]. Available: http://www.upnp-hacks.org/.

[74] K. Zetter, “Hospital viruses: Fake cancerous nodes in CT scans, created by malware, trick radiologists,” The Washington Post, 2019. [Online]. Available: https://www.washingtonpost.com/technology/2019/04/03/hospital-viruses-fake-cancerous-nodes-ct-scans-created-by-malware-trick-radiologists/.

[75] Kaspersky Lab, “USB threats from malware to miners,” 2018. [Online]. Available: https://securelist.com/usb-threats-from-malware-to-miners/87989/.

[76] W. Turton, “Hackers Are Using Infected USB Drives to Attack Critical Infrastructure,” Gizmodo, 2016. [Online]. Available: https://gizmodo.com/hackers-are-using-infected-usb-drives-to-attack-critica-1780304775.

[77] Rapid7, “Man-in-the-Middle (MITM) Attacks,” [Online]. Available: https://www.rapid7.com/fundamentals/man-in-the-middle-attacks/.

[78] Ettercap, “Ettercap Project,” [Online]. Available: https://www.ettercap-project.org/.

[79] L. Wyatt, “IoT RCE, a Study With Disney,” 2018. [Online]. Available: https://www.youtube.com/watch?v=X-FQ9NySBX4.

[80] D. Reid, “CUJO Smart Internet Security Firewall Reviewed,” 2016. [Online]. Available: https://www.smallnetbuilder.com/lanwan/lanwan-reviews/33018-cujo-smart-internet-security-firewall-reviewed.

[81] J. Sanders, “Why router-based attacks could be the next big trend in cybersecurity,” 2018. [Online]. Available: https://www.techrepublic.com/article/why-router-based-attacks-could-be-the-next-big-trend-in-cybersecurity/.

[82] US-CERT, “Alert (TA18-106A): Russian State-Sponsored Cyber Actors Targeting Network Infrastructure Devices,” [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA18-106A.

[83] C. Heffner, “Exploiting Surveillance Cameras Like a Hollywood Hacker,” 2013. [Online]. Available: https://media.blackhat.com/us-13/US-13-Heffner-Exploiting-Network-Surveillance-Cameras-Like-A-Hollywood-Hacker-WP.pdf.

[84] PenTestPartners, “Hijacking Philips Hue,” [Online]. Available: https://www.pentestpartners.com/security-blog/hijacking-philips-hue/.

[85] Akamai Threat Research, “MQTT-PWN,” 2019. [Online]. Available: https://github.com/akamai-threat-research/mqtt-pwn.

[86] A. Jakhar, “Expliot - Internet of Things Exploitation framework,” 2019. [Online]. Available: https://gitlab.com/expliot_framework/expliot.

[87] A. Duggal, “Understanding HL7 2.X Standards, Pen Testing, and Defending HL7 2.X Messages,” 2016. [Online]. Available: https://www.blackhat.com/us-16/briefings/schedule/#understanding-hl7-2x-standards-pen-testing-and-defending-hl7-2x-messages-4063.

[88] J. Tully, C. Dameff and M. Bland, “Pestilential Protocol: How Unsecure HL7 Messages Threaten Patient Lives,” 2018. [Online]. Available: https://www.blackhat.com/us-18/briefings/schedule/index.html#pestilential-protocol-how-unsecure-hl-messages-threaten-patient-lives-11726.

[89] Cylera, “HIPAA-Protected Malware? Exploiting DICOM Flaw to Embed Malware in CT/MRI Imagery,” 2019. [Online]. Available: https://labs.cylera.com/2019/04/16/pe-dicom-medical-malware/.

[90] Vectra, “Healthcare’s legacy infrastructureof unmanaged devices exposesa vulnerable attack surface,” 2019. [Online]. Available: https://www.vectra.ai/download/spotlight-report-on-healthcare-2019.

[91] C. Miller and C. Valasek, “A Survey of Remote Automotive Attack Surfaces,” 2014. [Online]. Available: https://packetstormsecurity.com/files/142305/A-Survey-Of-Remote-Automotive-Attack-Surfaces.html.

FORESCOUT

Forescout Technologies, Inc.
190 W Tasman Dr.
San Jose, CA 95134 USA

Toll-Free (US) 1-866-377-8771
Tel (Intl) +1-408-213-3191
Support +1-708-237-6591

Learn more at Forescout.com