Information Exposure in Smart Home Devices in Nordic Countries

Jacob Wahlman
Linköping University
Linköping, Sweden
jacwa448@student.liu.se

Lukas Nee
Linköping University
Linköping, Sweden
lukne541@student.liu.se

Andrei Gurtov
Supervisor
Linköping University
Linköping, Sweden
andrei.gurtov@liu.se

***Abstract*—With an increasing number of Internet of Things related devices being connected to the public Internet a lot of these devices remain insecure due to vulnerabilities, weak credentials, lack of credentials and bad configuration. These types of security issues often result in these devices being accessible to unauthorized users and could lead to unintended information exposure possibly leaking sensitive information.**

This paper uses *Shodan*[1] to analyze what Internet of Things related smart home devices that are exposed to the public Internet and the security issues in form of vulnerabilities and misconfiguration that could lead to unintended information exposure. The analysis is entirely focused on the Nordic countries. The paper does not investigate weak credentials or try to exploit any vulnerabilities identified.

The paper shows that there are a lot of different smart home devices exposed to the Internet communicating over different protocols. In some cases, these devices were found to not use any sort of authentication at all and in some cases, vulnerabilities were present on the devices. These types of misconfigurations, vulnerabilities and other factors could lead to severe information exposure for the owners of the device.

We also found that some of these devices did implement security controls such as authentication mechanisms but failed to implement it correctly allowing otherwise secured data to be exposed to an attacker.

***Index Terms*—smart homes, smart devices, internet of things, iot, privacy, computer crime, information security, shodan**

I. Introduction

Internet of Things (IoT) devices are simple physical electronic devices, for example common household objects, cameras, and more. IoT-devices are often modified and modernized versions of traditional devices, for example doorbells[2], the modernized versions are often referred to as smart home devices. The smart home devices are often modified to have extra sensors, actuators and communication abilities using different network protocols. Together these IoT-devices form local networks, this network is often referred to as Internet of Things. [1]

Due to most of the smart home devices communicating using network protocols, these devices often allow for incoming/outgoing connections in order to retrieve/send instructions and data to and from other smart home devices or applications, this allows the devices to become smarter since they are able to make informed decisions. The implications of communicating using network protocols also implies that security mechanisms need to be imposed in order to authenticate users and applications that should have access to the device. These security mechanisms are not always properly implemented, non-existant or misconfigured allowing an attacker to bypass the security mechanism of the device, as shown by Geneiatakis et al. in *Security and privacy issues for an IoT based smart home* [2].

The number of IoT-devices is increasing rapidly [3] thus it is imperative to understand how many of these devices are insecure. Collecting this data, making a snapshot, makes it possible to track trends and affect change. Having a high fidelity of the data is also important; one could look at sales records instead, which is a better source of total number of IoT-devices, however this tells nothing about the security flaws and how they could be resolved. Nor does sales records entail which devices are responsible for these security flaws.

This paper seeks to understand the number and types of smart home devices in the Nordic countries that have security vulnerabilities resulting in unintended information exposure for the owner. Primarily this paper focus on devices that if compromised will lead to substantial amounts of information exposure, for example web cameras, home automation hubs and other common smart devices. Finally, this paper aggregates the amoiunt and the general types of vulnerability leading to the information exposures.

The analysis of the types of devices and vulnerabilities can then be used to determine the current state of the security mechanisms present in smart home devices.

The rest of this paper is organized as follows. Section II describes the methodologies of collecting information on smart home devices, finding exposed smart home devices, and identifying vulnerabilities in these devices. Section III presents the results of the data collection. Section IV discusses the implications of the findings presented in section III. Finally, in section V the conclusions from our findings are presented as well as any potential future work on the subject is presented.

II. Methodologies

All form of information on the devices analyzed in this paper is collected using the Shodan search engine. Shodan is

[1] https://www.shodan.io/
[2] https://ring.com/doorbell-cameras

a search engine for Internet connected devices and tries to list information related to the device by performing port scanning[3] and other methods of prediction analyzing the banners sent as response to their queries.

A. Collecting Smart Home Device Metadata

In order to identify smart home devices on Shodan we need to find metadata relating the traffic that the device sends back to the device. This metadata could include manufacturer name, device serial numbers, device model number, device names and specific ports used by the services on the device.

Identifying the metadata used by a specific set of smart home devices is done by utilizing existing categories and filters on Shodan[4].

The results are limited to IPv4 results because Shodan cannot scan the entire IPv6 space (would take circa 1078 billion years with 1 trillion searches per second). This limits the number of results obtained. But as we mainly focus on what types of information is leaked and currently known vulnerabilities in the found devices, these results could act as a representation of the complete picture. However certain devices could be over-/underrepresented in IPv4 vs IPv6 and to know for certain one would have to compare the findings with sales records, however this is out of scope for this paper.

We also use resellers local to the Nordic countries in order to identify popular smart home devices, we then try to find matching filters on Shodan for the devices, if no such filters exist for the device, then we try to identify log files and other related documentation on the device uploaded to the Internet and use the common patterns as filters on Shodan.

B. Identifying Exposed Smart Home Devices

Given a filter Shodan returns a list of matching hosts, depending on the filter these results might vary in accuracy. From the results generated by Shodan we filter out results that we can identify as being a smart home device, this is done by looking at the banners in combination with other information such as the port the device operates on.

Shodan has additional functionality in helping to identify these smart home devices by using filters such as *has_screenshot* Shodan will display a screenshot from the device, this helps in identifying devices like web cameras that lack authentication mechanisms.

The list of positively identified smart home devices is then aggregated into a list of devices for further investigation.

C. Identifying Vulnerabilities in Smart Home Devices

Given a device identified by Shodan as being of a certain device allows us to use vulnerability databases such as Rapid7 vulnerability database[5] and Exploit-DB exploit database[6] to identify vulnerabilities and exploits for a certain device type or platform. In some cases, Shodan is able to display vulnerabilities for a device if Shodan is able to identify it as a certain type of device, identify a certain service and connect them to known vulnerabilities.

In order to identify configuration issues, we are going to rely on Shodan filters such as *has_screenshot* for web cameras. For other types of smart home devices, we are going to use the default access method for the device in order to determine if a security mechanism is present. Any further security testing on the device such as using default credentials, guessing credentials, and actively exploiting vulnerabilities is deemed out of scope for this paper and is legally questionable.

This paper assumes that not using any form of authentication due to no credentials being used or being able to bypass the authentication without exploits is categorized as a security vulnerability through misconfiguration of the device.

III. Related Work

There are a lot of papers investigated the current state of security for smart home devices using either Shodan to gather information or similar tools together with vulnerability and exploit databases.

In the paper *An Investigation of Vulnerabilities in Smart Connected Cameras* authors B. Joseph, D. Jönsson and A. Jacobsson investigate IP cameras using Shodan and an exploit database in order to determine the global vulnerability state of such cameras. The authors find that a lot of cameras identified either lacked security control, were misconfigured or had exploits available for vulnerabilities leading to the attackers being able to gain access. Finally, the authors claim that the information gathered from the cameras lead to severe information exposure and a breach of the personal integrity for the owners. [4]

In the paper *Identifying Vulnerabilities of Consumer Internet of Things (IoT) Devices: A Scalable Approach* authors R. Williams et al. try to map the vulnerabilities present on IoT-devices using tools like Shodan and Nessus. Nessus is a vulnerability scanner able to scan hosts on the Internet for vulnerabilities by comparing patterns like software, versions and other metadata to a vulnerability database.[7] The authors found that the most vulnerable devices identified were printers followed by IP cameras and Smart TVs. [5]

In the paper *Determining Home Users' Vulnerability to Universal Plug and Play (UPnP) Attacks* authors Golam Kayas, Mahmud Hossain, Jamie Payton, and S. M. Riazul Islam analyzes the security vulnerabilities of IoT-device using Universal Plug and Play (UPnP). The authors identify attack vectors which exploit these security vulnerabilities. The authors identify multiple remotely exploitable vulnerabilities. [6]

Finally, in the paper *An exploration of the cybercrime ecosystem around Shodan* authors M. Bada and P. Ildiko the usage of Shodan among cybercriminal communities and to what extent Shodan is used in information gathering. The authors investigate this by analyzing thread discussions from

[3]https://csrc.nist.gov/glossary/term/Port_Scanning
[4]https://www.shodan.io/explore
[5]https://www.rapid7.com/db/
[6]https://www.exploit-db.com/
[7]https://www.tenable.com/

several forums and find that Shodan is actively used by cybercriminals in order to gather intelligence, building botnets used for further attacks. [7]

IV. Results

In this section the types of IoT-devices identified using resellers and Shodan are listed as tables.

For each device we identified the manufacturer of the device, what filters that were used to identify the devices of that manufacturer on Shodan, number of devices found matching the filter and the number of devices determined to be vulnerable in some form.

We also identified the types of vulnerabilities for each manufacturer that we identified as well as the number of occurrences for the identified types of vulnerabilities.

For all of the results below we only list the devices that we were successful in identifying using Shodan.

A. IP Cameras

IP cameras is a type of digital camera that communicates over the Internet protocol. This makes it possible to send/stream video and images from a stationary surveillance camera using common Internet protocols such as Real-Time Streaming Protocol (RTSP) and other similar protocols.

For each of the filters listed in *Table I* we also queried using the filter *has_screenshot:true* as well as without in order to determine if the device streams were directly accessible.

For some of the IP cameras it was impossible to determine what the manufacturer of it was without further individual inspection so those devices will be marked as unknown.

In *Table I* we see that D-Link is the most common manufacturer of IP cameras that we were able to identify using Shodan that has vulnerable devices.

Furthermore, we can see that the manufacturer that has the most vulnerable devices is D-Link and the manufacturer that has the most vulnerable IP cameras in comparison to the total IP cameras exposed is the YawCam.

The D-Link IP cameras that we identified required in almost all cases credentials in order to access the IP camera stream over HTTP. We saw that some devices lacked credentials which allows anyone to access the IP camera stream over HTTP without authentication.

The HipCam IP cameras that we identified required in almost all cases credentials in order to access the IP camera stream over HTTP, however, we noticed that HipCam uses RTSP to traffic the video stream and this RTSP stream was in a lot of cases not protected by credentials even though the web portal was.

By using a program that is able to stream RTSP traffic we were able to stream IP camera traffic from any HipCam that did not have credentials on the RTSP port.

The Blue Iris IP cameras that we identified required in almost all cases credentials in order to access the IP camera stream over HTTP. We were not able to identify any vulnerabilities that can be used to gain direct access to the Blue Iris IP camera stream.

The YawCam IP cameras that we identified did not require any credentials in order to access the IP camera stream over HTTP. We also identified at least one vulnerability that can be used to gain entry to the YawCam IP camera if it were protected by credentials.

The unknown IP cameras that we identified required in some cases credentials in order to access the IP camera stream over HTTP. We classified it as unknown since they gave no information on manufacturer in the returned headers and response on Shodan, however, they did all have the string H264DVR present.

We also identified using the versions and device models identified by Shodan from our queries a number of potential vulnerabilities that can be used to collect information from the IP camera manufacturers, this information is displayed in *Table II* for D-Link, *Table III* for HipCam, *Table IV* for Blue Iris, *Table V* and *Table VI* for unknown IP camera manufacturers.

TABLE I
IP Cameras By Manufacturer on *2021-03-30*

Manufacturer	***Shodan Filters***	***Devices***	***Vulnerable Devices***
D-Link[8]	dcs-lig-httpd Camera country:se,fi,dk,no,is, Server: alphapd country:se,fi,dk,no,is	25618	2346
HipCam[9]	Hipcam RealServer/V1.0 country:se,fi,dk,no,is	944	352
Blue Iris[10]	BlueIris country:se,fi,dk,no,is	949	4
YawCam[11]	yawcam country:se,fi,dk,no,is	36	36
Unknown	H264DVR country:se,fi,dk,no,is	198	78

In *Table II* we see the types of vulnerabilities identified for IP cameras by the manufacturer D-Link as well as the occurrences of each vulnerability type.

The most prevalent IP cameras manufactured by D-Link that we could identify include *DCS-8000LH*, *DCS-936L*, *DCS-942L*, *DCS-5222L*, *DCS-942LB1*, *DCS-2121*, *DCS-2132L* and *DCS-825L*. The *DCS-8000LH*. *DCS-936L*, *DCS-942L* and *DCS-942LB1* are all cheaper IP cameras (less than $60), and the rest are a bit more expensive (more than $100).

Of the devices identified we found that the most prevalent versions of the firmware the devices ran on include 1.02, 1.07, 1.27, 2.12, 1.14, 1.06, 1.09, 1.01 and 1.05, however there were a lot of other versions albeit not as prevalent.

We identified two vulnerabilities that could lead to information exposure relating to the devices we identified. The first vulnerability is *CVE-2018-18441* [8], which is a vulnerability that exposes sensitive data about the devices without authentication. We also found a man-in-the-middle (MiTM) attack on the *DCS-2132L* device discovered by ESET [9] researchers in 2019, the vulnerability allowed attackers to intercept audio and video streams as well as modify the firmware that the device ran on. The MiTM attack is an attack on the web extension *mydlink services* and not on the device necessarily.

The *CVE-2018-18441* vulnerability are confirmed to affect *DCS-936L*, *DCS-942L*, *DCS-8000LH*, *DCS-942LB1*, *DCS-5222L* and *DCS-2121*, however more devices are likely to be affected as well according to the CVE details. The information exposed include fields such as model, product, version, device name, IP address, gateway IP address and settings related to the speaker and sensors. [8]

The vulnerability on the *mydlink service* extension is confirmed by the authors of the article to affect *DCS-2132L*, however the authors do not speculate on if this might be exploitable in other D-Link IP cameras as well. The vulnerability allowed any attacker to perform an MiTM attack on the communication between the extension *mydlink service* and the cameras, from there the attacker would be able to view any sound and video streamed. The vulnerability also allows any attacker to modify or completely replace the firmware running on the IP camera. The authors further note that D-Link IP cameras utilize *Universal Plug and Play* (UPnP) to set up port forwarding to itself on the router that it is connected to, this effectively exposes the IP camera to the Internet if no other control mechanisms are in place, the authors find that this can occur without the user's consent. [9]

TABLE II
OCCURRENCES OF VULNERABILITY TYPES FOR D-LINK IP CAMERAS

Vulnerability	*Occurrences*
No Credentials	114
Sensitive Information Exposed	2232

In *Table III* we see the types of vulnerabilities identified for IP cameras by the manufacturer HipCam as well as the occurrences of each vulnerability type.

For HipCam we were not able to determine what models that were used from the Shodan response or by accessing the IP camera landing pages. So, we were not able to identify any specific vulnerabilities. However, we were able to access the IP cameras by accessing port 554 communicating over *RTSP* without providing any authentication effectively bypassing the authentication mechanism in place.

TABLE III
OCCURRENCES OF VULNERABILITY TYPES FOR HIPCAM IP CAMERAS

Vulnerability	*Occurrences*
No Credentials	352

In *Table IV* we see the types of vulnerabilities identified for IP cameras by the manufacturer Blue Iris as well as the occurrences of each vulnerability type.

For BlueIris we were not able to determine what models that were used from the Shodan response or by accessing the IP camera landing pages. We were only able to identify one exploit relating to a denial-of-service (DoS) attack which did not leak any sensitive information.

In *Table V* we see the types of vulnerabilities identified for IP cameras by the manufacturer YawCam as well as the occurrences of each vulnerability type.

TABLE IV
OCCURRENCES OF VULNERABILITY TYPES FOR BLUE IRIS IP CAMERAS

Vulnerability	*Occurrences*
No Credentials	4

Of the devices identified the following versions of the firmware the devices ran on include $0.7.0$, $0.3.6$, $0.4.1$, $0.6.2$ and $0.6.1$.

For YawCam we were not able to determine what models that were used from the Shodan response or by accessing the IP camera landing pages. We were able to identify two vulnerabilities for YawCam IP cameras relating to information exposure. The first vulnerability *CVE-2017-17662* which is a directory traversal vulnerability allowing an attacker to read arbitrary files on the host system. *CVE-2017-17662* is confirmed to affect YawCam IP cameras running firmware version $0.2.6$ up to $0.6.0$. [10] The other vulnerability *CVE-2005-1230* is also a directory traversal vulnerability affecting YawCam IP cameras running firmware version $0.2.5$. [11]

TABLE V
OCCURRENCES OF VULNERABILITY TYPES FOR YAWCAM IP CAMERAS

Vulnerability	*Occurrences*
No Credentials	36
Directory Traversal	17

In *Table VI* we see the types of vulnerabilities identified for any unknown manufacturer of IP cameras as well as the occurrences of each vulnerability type.

For the unkown IP camera manufacturers we were not able to determine any versions or specific models and therefore not any vulnerabilities other than lack of authentication.

TABLE VI
OCCURRENCES OF VULNERABILITY TYPES FOR UNKNOWN IP CAMERAS

Vulnerability	*Occurrences*
No Credentials	78
Exploit Available	-

B. Control Panels

Control panels or commonly called hubs is a IoT-device that acts as a central communication hub for other IoT-devices and devices that are able to communicate using Internet protocols or other communication protocols.

The devices are often used to control and observe the state of other IoT-devices in order to centralize and simplify the usage of multiple devices. The panels often use protocols like MQTT and MODBUS to communicate.

MQTT consists of a broker that clients connect to and from there they client can publish data from the client for example temperature or any other data. Publishing data to the broker creates a topic, other clients can then connect and subscribe to these topics and in turn get the latest published data. For some topics it is possible to publish data from any client.

In order to test the vulnerability of the identified devices we used the script VII in order to connect and subscribe to all topics on the devices. The script then displays all the latest information from the topics.

In *Table VII* we see that Home Assistant is the most common manufacturer of control panels that we were able to identify using Shodan that has vulnerable devices.

None of the control panels that was identified using Shodan implemented any sort of authentication, allowing for direct unauthenticated communication with the control panel MQTT broker.

For the Home Assistant platform, we noticed that Home Assistant is most likely not producing a lot of the control panels and is instead installed on devices like Raspberry Pi[12].

The information available on the MQTT control panels differed greatly between different users depending on what generally we were able to identify information regarding burglar alarm system, lights, electricity consumption, sensors, and a lot more similar metrics.

Generally the information available depended on the number of integrations that were installed on the control panel device.

We also identified using the versions of the MQTT brokers identified by Shodan from our queries a number of potential vulnerabilities that can be used to collect information from the control panels, this information is displayed in *Table VIII* for Homey, *Table IX* for Home Assistant and *Table X* for Z-Wave.

TABLE VII
CONTROL PANELS BY MANUFACTURER ON *2021-03-30*

Manufacturer	***Shodan Filters***	***Devices***	***Vulnerable Devices***
Homey[13]	”homey/homey country:se,fi,dk,no,is	7	7
Home Assistant[14]	”homeassistant” MQTT country:se,fi,dk,no,is	38	38
Z-Wave[15]	”zwave” MQTT country:se,fi,dk,no,is	7	7

In *Table VIII* we see the types of vulnerabilities identified for control panels by the manufacturer Athom Homey as well as the occurrences of each vulnerability type.

We were not able to identify any information regarding the specific firmware version that the Homey control panel ran on for the identified devices.

We were able to identify two vulnerabilities for the Homey control panel. The first vulnerability is *CVE-2020-28952* which is a hard coded encryption and decryption key in the control panel used for debugging, an attacker would knowing this key and within range of the Homey control panel be able to read and control the devices connected to the Homey control panel. [12] The second vulnerability is *CVE-2020-9462* in which if an attacker is within radio frequency range of the device they would be able to obtain information relating to the network configuration of the device. [13]

[12]https://www.home-assistant.io/installation/

TABLE VIII
OCCURRENCES OF VULNERABILITY TYPES FOR ATHOM HOMEY CONTROL PANELS

Vulnerability	***Occurrences***
No Credentials	7

In *Table IX* we see the types of vulnerabilities identified for control panels by the manufacturer Home Assistant as well as the occurrences of each vulnerability type.

We were not able to identify any information regarding the specific firmware version that the Home Assistant control panel ran on for the identified devices.

We were able to identify two vulnerabilities for the Home Assistant control panel. The first vulnerability is *CVE-2021-3152* which is an issue relating to Home Assistant control panels not having a protective layer against directory traversal attacks in custom plugins running on the control panel. *CVE-2021-3152* affects devices running firmware versions below 2021.1.3. [14] The second vulnerability is *CVE-2018-21019* which is a vulnerability disclosing sensitive application data by allowing an attacker to read the error log of the device whilst being unauthenticated. *CVE-2018-21019* affects devices running firmware versions below 0.67.0. [15]

TABLE IX
OCCURRENCES OF VULNERABILITY TYPES FOR HOME ASSISTANT CONTROL PANELS

Vulnerability	***Occurrences***
No Credentials	38

In *Table X* we see the types of vulnerabilities identified for control panels by the manufacturer Z-Wave as well as the occurrences of each vulnerability type.

For Z-Wave we were not able to determine what models that were used from the Shodan response. We were also not able to identify any vulnerabilities for the Z-Wave control panels.

TABLE X
OCCURRENCES OF VULNERABILITY TYPES FOR Z-WAVE CONTROL PANELS

Vulnerability	***Occurrences***
No Credentials	7

C. Media Servers

Media servers are servers used to store and stream media including video and audio.

These servers are often used in media streaming boxes and allow the user to stream their media on demand.

Oftentimes, these devices implement some sort of web server to allow for remote access to the device, for example, allowing other devices on the network to control the music being played through the speakers or control the playback of a movie.

For the manufacturers in *Table XI* we added *200 OK* to the query in order to determine if the servers were directly accessible.

In *Table XI* we see that Plex is the most common manufacturer of media servers that we were able to identify using Shodan that has vulnerable devices.

Furthermore, we can see that the manufacturer that has the most vulnerable devices is Plex and the manufacturer that has the most vulnerable IP cameras in comparison to the total number of media servers exposed in Universal Media Server.

We found that most of the media servers required authentication through credentials, however, those that did not require authentication often contained varying types of media for example, movies, music, and pictures.

We were only able to identify vulnerabilities for the Plex media server that can be used to collect information, this information is displayed in *Table XIII* for Plex.

We found no vulnerabilities in Samsung Allshare. However, all of these media servers use UPnP which is vulnerable. [6] Since this is a vulnerability with UPnP and not the actual media server (the media server can be used without the use of UPnP) it is not declared as a vulnerability in XI. Using UPnP is not unique here to the Samsung Allshare servers, but is widely used among all types of IoT-devices. [16]

TABLE XI
MEDIA SERVERS BY MANUFACTURER ON *2021-04-06*

Manufacturer	***Shodan Filters***	***Devices***	***Vulnerable Devices***
Logitech[16]	Logitech Media Server country:se,fi,dk,no,is	74	25
Plex[17]	X-Plex-Protocol country:se,fi,dk,no,is	17860	119
Universial Media Server[18]	New WebUI country:se,fi,dk,no,is	14	14
Samsung AllShare Server[19]	"SERVER: UPnP/1.1 Samsung AllShare Server/1.0" country:se,fi,no,dk,is	790	X

In *Table XII* we see the types of vulnerabilities identified for media servers by the manufacturer Logitech as well as the occurrences of each vulnerability type.

Of the devices identified the following versions of the firmware the devices ran on include 8.1.1, 8.2.0, 8.1.0, 8.0.0, 7.7.6, 7.7.5 and 7.7.4.

We were not able to identify any vulnerabilities that could directly lead sensitive information from the Logitech Media Servers.

TABLE XII
OCCURRENCES OF VULNERABILITY TYPES FOR LOGITECH MEDIA SERVERS

Vulnerability	***Occurrences***
No Credentials	25

In *Table XIII* we see the types of vulnerabilities identified for control panels by the manufacturer Plex as well as the occurrences of each vulnerability type.

Of the instances identified the most prevalent instance versions include 1.14.0, 1.19.6, 1.18.0, 2.4.25, 1.10.3, 1.14.1 and 1.16.0

We were able to identify a number of vulnerabilities that could lead to information exposure in Plex instances. *CVE-2020-5742* is a vulnerability allowing an attacker to make requests to the Plex instances from a different origin making it possible for an attacker to retrieve data from active user sessions and retrieve potentially sensitive information. [17] *CVE-2020-5741* is a vulnerability allowing an attacker to execute arbitrary *Python* code on the Plex instances through deserialization of untrusted data. [17] *CVE-2020-5740* is a vulnerability allowing an attacker to execute arbitrary *Python* code on the Plex instances due to improper input validation. [18] and several more vulnerabilities leading to remote code execution due to improper file upload validation, directory traversal and XML external entity processing (XXE) vulnerabilities.

All versions up to and including 1.18.2 is affected by one or more of these vulnerabilities.

TABLE XIII
OCCURRENCES OF VULNERABILITY TYPES FOR PLEX MEDIA SERVERS

Vulnerability	***Occurrences***
No Credentials	10
Remote Command Execution	71

In *Table XIV* we see the types of vulnerabilities identified for control panels by the manufacturer Universal Media Server as well as the occurrences of each vulnerability type.

For Universal Media Server we were not able to determine what models or versions that were used from the Shodan response or by accessing the landing pages.

We identified one vulnerability that could lead to information exposure in Universal Media Servers. *CVE-2018-13416* is a vulnerability allowing an attacker to access and read arbitrary files on the host system and remote command execution in Windows domains by allowing an attacker to initiate SMB connections. [19]

TABLE XIV
OCCURRENCES OF VULNERABILITY TYPES FOR UNIVERSAL MEDIA SERVER MEDIA SERVERS

Vulnerability	***Occurrences***
No Credentials	14

Svane et al. showed how ports used for Samsung smart TV's media server Samsung Allshare server could leak information. In XV we see that the same ports being open and thus exploitable. [20]

D. Smart TVs

Smart TVs are traditional television devices with integrated Internet capabilities and allow for Web 2.0 content.

TABLE XV
OCCURENCES OF SAMSUNG ALLSHARE SERVER OPENED PORTS

Port	*Occurrences*
9119	499
7676	283
9110	7
9295	1

In *Table XVI* we see that Samsung is the most common manufacturer of Smart TVs that we were able to identify using Shodan.

TABLE XVI
SMART TVS BY MANUFACTURER ON *2021-04-06*

Manufacturer	*Shodan Filters*	*Devices*	*Vulnerable Devices*
Philips	”Philips TV” country:se,fi,dk,no,is	6	0
Samsung	”Samsung Smart TV” country:se,fi,dk,no,is	2070	0

No vulnerabilities were identified by Shodan for Smart TVs by the manufacturer Philips or Samsung.

V. DISCUSSION

In this section the types of IoT-devices identified in the results and the security findings related to them are discussed as well as the possible consequences these findings might have in respect to information exposure. Any limitations on the findings are also discussed.

For some of the devices we were not able to collect information regarding the versions that the device utilizes in its firmware or software, this makes it impossible to determine if a device is vulnerable to a certain exploit without testing the exploit.

Generally, we were able to identify several vulnerabilities in the smart home devices that when exploited could lead to severe information exposure or device takeovers through various vulnerabilities or lack of authentication mechanisms.

A. IP Cameras

From our findings in *Table I* we can see that the lack of credentials were the most common type of vulnerability, this type of vulnerability can lead to severe information exposure since it requires little to no technical expertise in order to gain access to the IP cameras.

Some of the cameras identified in *Table I* were accessible both through a landing page using HTTP/S as well as through RTSP.

By using any program that supports RTSP traffic such as VLC[20] we were able to view IP cameras that otherwise required authentication through the HTTP/S landing page. This is especially severe since the user is not necessarily aware of this and assumes that their IP camera is secured by the presence of credentials in the HTTP/S landing page. For

[20]https://www.videolan.org/vlc/

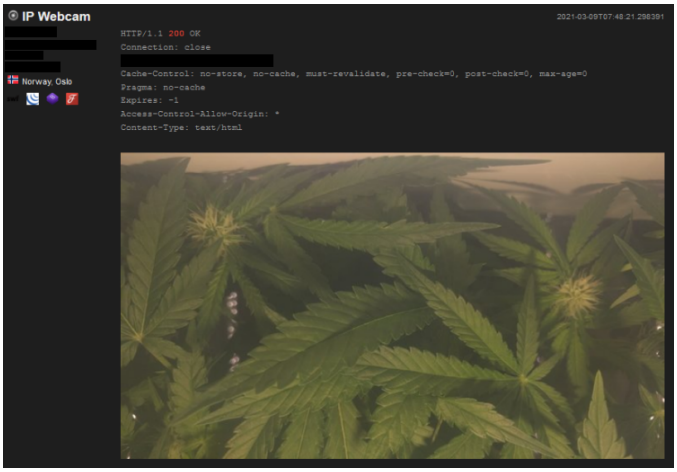

Fig. 1. Possible Narcotics Displayed on Vulnerable IP Camera

example, the IP cameras showed a lot of different subjects such as individuals sleeping in their bedrooms, recordings of living spaces and other arbitrary objects, for example see *Figure 1*

We also noted that there were relatively few discovered vulnerabilities on the IP cameras even though these types of devices leak a lot of sensitive information if the location of the camera is pointed in the right direction. Most of the vulnerabilities that we discovered on the IP cameras allowed attackers to collect information by exploiting directory traversal vulnerabilities in the camera firmware. However, as Čermák and Fránik discussed in their vulnerability disclosure regarding the MiTM D-Link vulnerability these types of devices oftentimes utilize UPnP to forward traffic to them on the owners home routers exposing the IP cameras to the public Internet.

In the paper *Identifying vulnerabilities of consumer Internet of Things (IoT) devices: A scalable approach* by Williams et al. the authors found that the majority of the vulnerable devices were of the manufacturer D-Link followed by Axis. We also found that most of the identified IP cameras were made by the manufacturer D-Link, however, we found that most of the IP cameras implemented some sort of security mechanism, and no vulnerabilities were identified to be usable to gain access to the IP camera.

Depending on the placement of the camera and the victim it is possible for an attacker to gain information on a victim by monitoring the IP cameras installed by the victim, the attacker might then be able to use the information gathered for blackmailing.

Due to the simplicity of gaining access to the IP cameras it would also be entirely possible for other types of malicious activities, for example gathering information in preparation for a robbery.

We also noted that some of the IP cameras identified are most likely exposed to the Internet by choice, for example, cameras that film nature reserves or a town square. However, since it is difficult to determine if the cameras were exposed by choice, it will be assumed that the number of cameras exposed

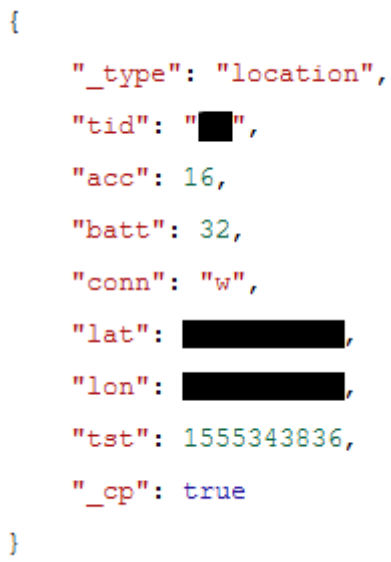

Fig. 2. MQTT Response of Owntracks

by choice is negligible.

B. Control Panels

From our findings in *Table VII* we can see that no credentials were the most common type of vulnerability, this type of vulnerability can lead to severe information exposure since it requires little technical expertise, it still requires the attacker to have knowledge on how to connect to the MQTT brokers and subscribe to the topics in order to gather information.

In order to determine if the MQTT devices exposed sensitive information we tried to subscribe to all topics and analyze the results returned using a simple script *Listing VII* to subscribe to all topics on the MQTT brokers for each vulnerable device.

The output from such a script depends on the number of devices connected to the MQTT broker, for example, we observed devices connected measuring temperatures of rooms, battery percentages as well as devices controlling objects in the vicinity for example turning on and off the television, lights and more.

We also identified some users using the Owntracks[21] application communicating using MQTT with the control panels, when subscribing to the topic it will return the latest known coordinates of the device using Owntracks making it possible to know the exact location of all the user devices that has this application installed. See *Figure 2*

The vulnerabilities that we discovered on the control panels were relatively few and often did not leak very sensitive data about the devices or the owner of the device. However, we believe that utilizing a solution like Home Assistant where the user is able to install any custom plugin onto their device is potentially dangerous to their information if the plugin lacks proper security controls, this however is oftentimes difficult to ensure for any non-technical user.

[21] https://owntracks.org/

An attacker having access to such a device would be able to gather a lot of information about devices connected to the control panel even though they are not necessarily exposed to the Internet themselves. For example, gathering location information on where a victim is, ability to control certain devices and gather information about devices connected to the control panel.

The topics we found on the identified control panels were mostly exposed controls to lights, electronics and other appliances as well as some measurement devices like thermometers. This information at first glance seems harmless, whether a lamp is on or off does not necessarily indicate whether anyone is home. But the information being available at all times means an assailant could find patterns. Perhaps even identify not just when anyone is home, but when someone in particular is home, it is not impossible that different people enjoy different combinations of active devices.

There has been previous research regarding the lack of security and the risk of using MQTT in IoT-devices with proposed new protocols and security mechanisms to ensure integrity and confidentiality for the data transmitted. For example, Singh et al. proposed in their paper *Secure MQTT for Internet of Things (IoT)* that a new protocol secure MQTT or SMQTT be used with encryption of published data making it harder for an attacker to acquire the data. The performance of the SMQTT protocol was shown to be of no concern for most IoT-devices and so more secure protocols instead of MQTT should be a possibility in order to ensure the confidentiality of future IoT-devices using MQTT as a communication protocol. [21]

Finally, we identified a lot of devices that are very sensitive and can have severe consequences if one can control them, for example the burglar alarms, since we do not want to alter or manipulate devices that we do not own we were not able to confirm that we could manipulate the state of these types of devices.

C. Media Servers

From our findings in *Table XI* we can see that no credentials was the most common type of vulnerability, for the media servers this can of course lead to some very personal information exposure depending on the information stored on the media server.

When accessed these devices often revealed information regarding the user's music and audio libraries, videos, movies, and other types of media content allowing an attacker to access and download any of the data from the server.

Whilst a majority of the media servers only contained movies and music albums, we also saw some of the media servers containing videos and pictures of a more personal nature.

Depending on the type of content being stored it varies on the severity of the personal information being leaked, for example an attacker stealing your music library might not be an as sensitive breach compared to an attacker stealing personal videos. For example, see *Figure 3*.

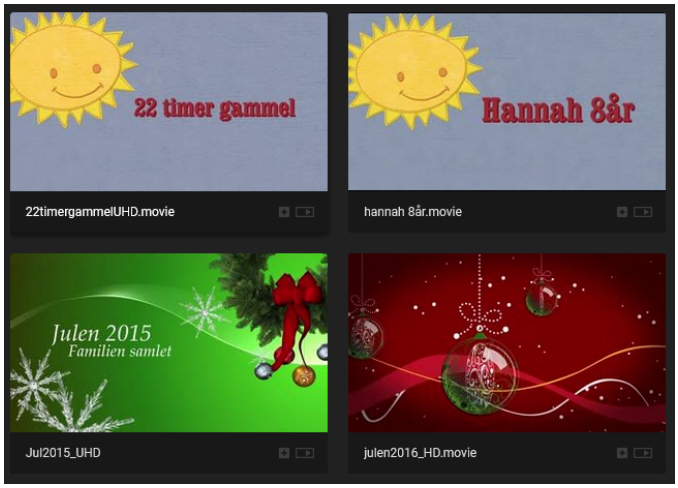

Fig. 3. Media Server Containing Personal Videos

We were able to identify a lot of vulnerabilities for the media servers, especially the Plex Media Server that has had several severe vulnerabilities disclosed over a relatively short span of time (2018-2021) and all of the vulnerabilities being very severe. These types of devices can seem ”dumb”, and a user might believe that even if an attacker gains access to it they won't be able to get any information. An attacker can use any of the vulnerabilities we presented collect information and spread through a host network and collect more information about a user relatively easy because of the implications of the vulnerabilities.

D. Smart TVs

There were not any obvious vulnerabilities with smart TVs among our findings. They are known to be used in botnets such as Mirai, however Mirai botnets primarily targeted IP cameras and home routers[22].

Modern smart TVs can be equipped with webcams, microphones, and can be connected to the LAN. [22] A lot of information can thus be disclosed if they are vulnerable. And as for all devices on a LAN, any unprotected device on that LAN is now unsafe.

On certain smart TVs you can login to services such as Facebook[23] and YouTube[24]. Most Smart TVs are also often equipped with a traditional web browser making common web attacks a possible attack vector.

Bachy et al. showed in 2013 how certain smart TVs were vulnerable to attack by using a malicious antenna. These TVs usually had more buses used to communicate, all of which opens up the possibility of a vulnerability. However, succeeding with an attack as severe as remote code execution would require deep knowledge of the installed firmware or a lot of experimentation work. [23] These vulnerabilities should be obsolete in modern smart TVs, but any device with multiple communication buses is open for similar vulnerabilities and old smart TVs are still in use.

[22]https://en.wikipedia.org/wiki/Mirai_(malware)

[23]https://www.lg.com/us/press-release/lg-smart-tvs-add-support-for-facebook-watch-tv-app

[24]https://support.google.com/youtube/answer/3015415?hl=en

VI. Conclusion

Today devices are becoming more and more connected in an effort to simplify life and connectivity for the owners of the devices, the technical expertise of those acquiring smart home devices vary wildly leading to an increase in devices being misconfigured or lacking updates to critical components leading to information leakage since the responsibility of the device security is shifted onto the owner.

Using Shodan we were able to identify a wide variety of devices used by Smart Home systems, for example home automation, surveillance, and other types of sensors in the Nordic geographical area.

These devices often lacked authentication allowing anyone to gain access and gather information both on the device content as well as in some cases personal information. In some cases there were vulnerabilities present or possibly present on the identified devices, these vulnerabilities could in cases where other security mechanisms were present still expose the owners of the devices to serious information exposure.

The information gained from accessing these devices could expose the owners to further risk, for example extortion, burglary or any other crime requiring general information gathering since these devices often leak information pertaining to location, availability to property as well as control of certain devices.

The study shows that these devices present a serious risk to the personal information of the user and it might not be obvious to the user that their data and devices are exposed and available on the public Internet when the devices oftentimes without instruction expose themselves to the Internet.

This paper shows that giving access to sensitive data to manufacturers and trusting them to safe-keep it is a dangerous presumption and that the users of the devices and services discussed in this paper should be careful and stay vigilant regarding vulnerabilities on their devices.

Acknowledgment

Shodan.io provided the possibility to gather the information used in this study in an excellent format. The National Vulnerability Database, Rapid-7 and ExploitDB for providing easy to use vulnerability and exploit databases.

References

[1] Internet Engineering Task Force, “The internet of things,” 2021. https://www.ietf.org/topics/iot/.

[2] D. Geneiatakis, I. Kounelis, R. Neisse, I. Nai-Fovino, G. Steri, and G. Baldini, “Security and privacy issues for an iot based smart home,” in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1292–1297, 2017.

[3] D. Bastos, M. Shackleton, and F. El-Moussa, “Internet of things: A survey of technologies and security risks in smart home and city environments,” in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, pp. 1–7, 2018.

[4] J. Bugeja, D. Jönsson, and A. Jacobsson, "An investigation of vulnerabilities in smart connected cameras.," 2018.
[5] R. Williams, E. McMahon, S. Samtani, M. Patton, and H. Chen, "Identifying vulnerabilities of consumer internet of things (iot) devices: A scalable approach," in *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 179–181, 2017.
[6] G. Kayas, M. Hossain, J. Payton, and S. M. R. Islam, "An overview of upnp-based iot security: Threats, vulnerabilities, and prospective solutions," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0452–0460, 2020.
[7] M. Bada and I. Pete, "An exploration of the cybercrime ecosystem around shodan," in *2020 7th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pp. 1–8, 2020.
[8] "Cve-2018-18441." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-18441, 2018.
[9] M. Čermák and M. Fránik, 2019.
[10] "Cve-2017-17662." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-17662, 2017.
[11] "Cve-2005-1230." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2005-1230, 2005.
[12] "Cve-2020-28952." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-28952, 2020.
[13] "Cve-2020-9462." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-9462, 2020.
[14] "Cve-2021-3152." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-3152, 2021.
[15] "Cve-2018-21019." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-21019, 2018.
[16] G. Kayas, M. Hossain, J. Payton, and S. M. R. Islam, "An overview of upnp-based iot security: Threats, vulnerabilities, and prospective solutions," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0452–0460, 2020.
[17] "Cve-2020-5741." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-5741, 2020.
[18] "Cve-2020-5740." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-5740, 2020.
[19] "Cve-2018-13416." https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-13416, 2018.
[20] T. Svane and S. Van Dessel, "Investigating consumer smart home vulnerability," 2021.
[21] M. Singh, M. Rajan, V. Shivraj, and P. Balamuralidhar, "Secure mqtt for internet of things (iot)," in *2015 Fifth International Conference on Communication Systems and Network Technologies*, pp. 746–751, 2015.
[22] M. Ghiglieri, "Smart tv privacy risks and protection measures.," 2017.
[23] Y. Bachy, F. Basse, V. Nicomette, E. Alata, M. Kaâniche, J.-C. Courrège, and P. Lukjanenko, "Smart-tv security analysis: Practical experiments," in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 497–504, 2015.

VII. Appendix

```
import paho.mqtt.client as mqtt
import sys

# Author: Victor Pasknel
# https://morphuslabs.com/hacking-the-iot-with-mqtt-8edaf0d07b9b

def on_connect(client, userdata, flags, rc):
    print("[+] Connection successful")
    client.subscribe('#', qos = 1) # Subscribe to all topics
    client.subscribe('$SYS/#') # Broker Status (Mosquitto)

def on_message(client, userdata, msg):
    print('[+] Topic: %s - Message: %s' % (msg.topic, msg.payload))

client = mqtt.Client(client_id = "MqttClient")
client.on_connect = on_connect
client.on_message = on_message
client.connect(sys.argv[1], int(sys.argv[2]), 60)
client.loop_forever()
```