Industrialising Blackmail: Privacy Invasion Based IoT Ransomware

Calvin Brierley[0000−0001−8766−822X], Budi Arief[0000−0002−1830−1587], David Barnes[0000−0001−6073−0951], and Julio Hernandez-Castro[0000−0002−6432−5328]

School of Computing, University of Kent, Canterbury, England
{C.R.Brierley, B.Arief, D.J.Barnes, jch27}@kent.ac.uk

Abstract. Ransomware (malware that threatens to lock or publish victims' assets unless a ransom is paid) has become a serious security threat, targeting individual users, companies and even governments, causing significant damage, disruption and cost. Instances of ransomware have also been observed stealing private data and blackmailing their victims. Worryingly, the prevalence of Internet of Things (IoT) devices and the massive amount of personal data that they collect have opened up another avenue of attack. The main aim of this paper is to determine whether *privacy invasion based* ransomware would be a viable vector for attackers to use on IoT devices. The secondary aim is to identify countermeasures that can be implemented to prevent such attacks from being used. To accomplish these aims, we examined how private data accessible via IoT devices could be obtained, processed and managed by a ransomware attacker. We identified a number of data sources on IoT devices that can be used to access private data, such as audio and video feeds. We then investigated methods to interpret such data in order to blackmail the device's owner. We then produced proof of concept malware for multiple IoT devices, including an external "collator" that manages the valuable data collected, demonstrating that an attack could be performed at scale. This research shows that attackers can use the functionality of an infected device to invade the privacy of the device's owner, as part of a ransomware attack. We have demonstrated that, given suitable infrastructure, attackers would be able to ransom users for values higher than the cost of the compromised device, as well as heavily damage the trust in the device itself, which would cause further negative impact on the device manufacturer. Finally, we highlight the need for proactive measures to deter this style of attack by applying the suggested countermeasures.

Keywords: Security · Privacy · IoT · Ransomware · Malware · Cloud Services · Cybercrime · Blackmail.

1 Introduction

The increasing popularity of the Internet of Things (IoT) has lead to a corresponding increase in attacks on IoT devices. While IoT devices themselves are used for many different purposes – such as light bulbs, digital video recorders,

and fridges – when infected, they are typically used to perform either Distributed Denial of Service (DDoS) attacks [2], or to mine cryptocurrency [48]. However, ransomware has also become increasingly prevalent [41, 5, 9], and its success has garnered significant interest in carrying out ransomware attacks on IoT devices.

The volume and the relative insecurity of IoT devices make them a potentially profitable target for ransomware authors. To evaluate the potential threat of IoT ransomware, researchers have developed proof of concepts investigating how IoT devices could be attacked [6, 26]. However, as IoT devices rarely store files that their user may consider essential, typical crypto ransomware may not be as effective as they would be on regular personal computers. Instead, early IoT ransomware strains typically "lock" infected devices, preventing them from working correctly unless a payment is made [6]. While this method of ransom may be effective, there are a number of limitations (discussed later in this paper), which may dissuade ransomware operators from using it. Attackers are likely to explore other methods of monetising IoT-based ransomware in the future. One such method involves extracting private data from and/or using the IoT device, which can then be used to extort the user under threat of public release.

In this paper, we aim *to determine the viability of ransomware attack leveraging privacy invasion techniques on IoT devices*, and *devise countermeasures that can be implemented to prevent such attacks from being used by cybercriminals.*

Contributions. The key contributions of our paper are: **(i)** a demonstration of how attackers may identify and extract private data accessible via IoT devices to facilitate ransomware; **(ii)** an overview of how such an attack might be structured and managed; **(iii)** an identification of possible weaknesses that may be introduced by attackers when performing such an attack; **(iv)** a list of countermeasures that could be used to hinder or prevent such an attack.

The rest of the paper is organised as follows. Section 2 covers previous privacy based ransomware attacks and IoT privacy research. Section 3 investigates data sources commonly found on IoT devices, and how they could be accessed by attackers. Section 4 describes how attackers could interpret exfiltrated data to identify private information. Section 5 shows how attackers could collate information extracted from IoT devices during a ransomware campaign. Section 6 demonstrates some of the privacy-invasion techniques on IoT devices with differing sensors and uses. Section 7 discusses countermeasures that could be used to prevent such attacks, the limitations of the current work, and further research that could be performed. Finally, Section 8 summarises our findings.

2 Background and Related Work

Ransomware is class of malware that uses a number of techniques to restrict access to assets owned by users, typically requiring a payment in cryptocurrency to be made for access to be returned [34, 30]. As ransomware continues to evolve, new methods have been used to ransom victims more effectively. One of the latest trends is for ransomware operators to steal sensitive data and to threaten

the owners with its release, unless a ransom demand is paid. This method is particularly effective if the stolen data is confidential or embarrassing in nature, as it could be severely damaging if made public.

Multiple companies have already been impacted by this method. In February 2021, CD Projekt Red, a games development company, was subjected to a ransomware attack. As part of the ransom note, the attackers claimed to have stolen source code, employee details and accounting information, which they threatened to release if payment was not made within 48 hours [8]. After CD Projekt Red refused to pay the ransom, the source code was put up for auction [38]. It was later revealed that portions of the data were potentially being leaked online [15]. In December 2020, the Scottish Environmental Protection Agency (SEPA) was also subject to a ransomware attack, with the attackers stealing approximately 1.2GB of files. After refusing to pay the ransom, the attackers publicly released over 4,000 documents on the dark web, including emails and databases used for contracts and commercial services [45, 39].

2.1 IoT Based Ransomware

As both IoT devices and ransomware have become more popular, it is not surprising to see an increased interest in IoT based ransomware – from both security researchers and attackers. Initial attempts to produce IoT based ransomware have implemented various "locking" methods to ransom users, i.e., preventing infected devices from functioning correctly until a payment is made [6, 51, 28]. More complex types of ransomware may require persistence, which while possible, may be difficult to achieve, depending on the design of the device [7].

While these techniques may work in certain circumstances, consumer IoT devices impose two obvious limitations for successful crypto- and locker-based ransomware: *replaceability* (most IoT devices are designed to be relatively "cheap" when compared to traditional desktop targets – as such, users may instead opt to simply replace the device rather than pay a ransom); and *lack of valuable files* (IoT devices rarely contain files that are essential to the user, so crypto-based ransomware is unlikely to be as effective). However, as IoT devices are often designed to have access to data associated with their user's personal environment, they thereby may provide a unique opportunity for attackers. In what follows, we describe how IoT devices may be used by attackers to invade the privacy of their users.

2.2 Privacy Invasion

IoT devices often have direct access to sensors within a user's home, which has lead to a significant amount of research into the privacy of data that they manage or create [42, 29, 43]. This is especially important as IoT devices are, by design, required to be connected to the Internet. Therefore, if a device is found to be exploitable, this information may be exposed to remote attackers.

Previous research has investigated how attacks on IoT devices may impact users, including case studies that demonstrate the possible methods attackers

could use to track user activity [3]. Various attacks have also been performed "in the wild"; for instance, there have been numerous instances of attackers accessing network cameras exposed to the Internet, allowing them to view video feeds inside homes and, in some cases, sell obtained "adult content" to others [47]. In one instance, an attacker used a camera's speaker to threaten victims and demand a ransom of 50 bitcoin [1].

It is therefore straightforward to see that the natural progression of ransomware attack strategy would be to threaten to leak data belonging to victims in order to encourage payment. It may be possible for attackers to exploit IoT devices' access to sensors – e.g., by monitoring or turning on a microphone or camera without the owner's knowledge – in order to capture personal or potentially embarrassing data. In the next section, we will discuss the possible sources of private information that could be exploited by an attacker.

3 Data Sources

Many IoT devices – such as wearables, smart toys, and medical devices – process or generate private data that their legitimate users may not want to be publicly exposed. Below, we discuss the data sources commonly found on such IoT devices, and how they could be used by a malicious attacker:

- *In-built Sensors.* An IoT device typically uses sensors to measure aspects of its environment in order to function. Some of the most commonly available sensors are cameras (which are often used in Internet-connected security systems), microphones (which are sometimes used for communication and control) and geolocation sensors (which can be used to determine the current location of the user).
- *Network Data.* IoT devices, by definition, must be able to connect to the Internet, allowing them to communicate with other devices and their users. However, if the device has permissions to send, receive or view any sensitive data, attackers who exploit the device will gain the same privileges. It can lead to security and privacy issues such as passive monitoring, where if the infected device acts as a gateway to the internet (e.g. a router), the attacker may be able to "sniff" the packets sent through it. The attacker may also be able to scan the internal network of the device's local network, which could lead to the discovery of additional sources of personal information such as network accessible file storage or other vulnerable IoT devices.
- *Local Configuration settings.* While IoT devices are less likely to contain significant amounts of user-created data, they may still store personal information that is of value. An IoT device may request information from their users during the device's set-up stage – such as their name or email address – which is often stored within the device's configuration settings. If the location of this information is known to the attacker, it could be extracted to facilitate communication with, or intimidate, the victim. The attacker could also scan the memory of local processes or storage for data with a recognisable structure, such as email addresses or dates, using regular expressions.

4 Identifying Private Data

For privacy based ransomware attacks to be successful, the attacker must first be able to extract data from IoT devices, but more importantly, identify data of value which could be used to extort their victims. For large ransomware campaigns, it is infeasible to manually search through large volumes of collected data to pick out relevant information. Instead, it would be necessary for attackers to develop methods to categorise and sift through the available data automatically and efficiently. Below, we discuss some the methods that could be used.

4.1 Malicious Use of Machine Learning

IoT devices typically have access to various types of structured data, such as configuration settings, which would be relatively easy for attackers to access and interpret. However, raw data collected from IoT devices' sensors will first need to be processed before its "value" can be determined. One approach is to use machine learning tools to automatically classify input data, drastically lowering the amount of manual intervention required by the attacker. This method could exploit two data sources commonly found on IoT devices, as shown below.

Identifying Private Images with Image Recognition. Cameras are often considered as a vector to invade a user's privacy, as if an attacker is able to gain access, they would also be able to extract images from within a victim's home without their knowledge. However, the attacker must be able to identify which images are likely to be "valuable". The process for selecting potentially ransomable images could be performed manually by the attacker, but it would be a time-consuming process that would not scale well. Therefore, automating this process would be desirable for the attacker. There are various different models that may assist in identifying ransom-able images, such as:

- *Theme/Object Recognition.* If certain themes or objects are detected – such as cars, buildings, or crowds – it could indicate that the infected device is stationed outside, and are likely to produce images of "low value". If people or objects likely to be inside, such as furniture, are detected, they will raise the potential value of the images extracted from the device.
- *Face Detection.* Face detection could be used to confirm the presence of human victims within obtained images. If a victim is confirmed to be within the image, it could be very valuable when used in a ransom note as proof of exploitation, especially if the victim was caught in a compromising position.
- *Explicit Content Detection.* Some online services offer explicit content detection for uploaded images/videos. A typical use case would be to prevent the upload/transmission of explicit content on "safe-for-work" platforms. An attacker could use this maliciously by scanning for explicit content taken without the victim's consent, which could then be used to ransom the victim.

Identifying and Transcribing Private Conversations. The possibility of eavesdropping via vulnerable IoT devices has been explored in previous research [50, 13] but not in the context of ransomware. For this method, the attacker aims to transcribe using speech-to-text engines private conversations held by the victim. Once the audio has been transcribed, the attacker can use automated methods to search for keywords, such as those related to potentially exploitable activity.

4.2 Network-Based Privacy Invasion

There are several techniques that attackers could use to extract private information by interacting with the local network using compromised IoT devices.

Intercepting Browsed Domains. If an attacker is able to intercept a user's Internet traffic via an infected device (such as a router), they may be able to extract sensitive information about the user's browsing habits. In this case, the attacker may intercept traffic passing through the device and extract domain names of any websites that the user visits from various protocols, such as DNS [33], HTTP [14] or HTTPS [10]. The websites can then be compared against a list of domains associated with illegal or compromising activities. If a match is found, details could then be logged to a Command and Control (C&C) server.

Intercepting Web Content. It may also be possible to intercept the content of visited web pages, and the content of websites with known structures could be read to extract important information, such as video titles, usernames or personal information. For HTTP traffic, this is relatively simple, as communication is typically performed in plaintext, allowing attackers to access any transferred content. Increasingly, web traffic is using HTTPS, which encrypts the communication between the client and server when transmitting web content [11, 18]. However, it could still be possible to gain access to encrypted content using "man in the middle" (MitM) attacks, such as `SSLStrip`, which allows attackers to intercept and modify victim's web requests to bypass HTTPS encryption [32]. This allows the attacker to catch inattentive users unaware and extract plaintext communication from typically encrypted traffic. A similar style of attack has been previously implemented by the IoT malware `VPNFilter` to extract usernames, passwords and logins [24].

Identifying Device Locations via WiFi Positioning. The location of the infected device could be used to determine the address of the user. However, in order to ascertain the location of the infected device, the attacker must make use of the available data sources. Some devices need to be aware of their current position in order to function correctly, such as fitness trackers, which may need to periodically acquire the current location of the device to track a user's running activity and route. Ideally, this type of information would be acquired using a Global Positioning System (GPS), however, most IoT devices are unlikely to implement GPS sensors, especially if they are not designed to be moved often.

Online WiFi Positioning systems allow users to triangulate their current position by comparing a scan of local WiFi signals to a list of known signal locations stored in an online database. The accuracy of this measurement is dependant on various factors, such as the number of detected signals, or matches found in the service providers' database.

If an infected device has wireless capabilities, attackers may be able to perform a scan to discover the SSIDs, MAC addresses and signal strengths of nearby routers, which can then be sent to the C&C server. The attacker could then upload it to an online service such as Mozilla Location Services or the Google Cloud platform to obtain an estimate of the device's location [22, 36].

Internal Network Structure. Infected devices could provide attackers with access to other devices on the local network which would be otherwise inaccessible from the Internet. The attackers would then be able to scan or attack previously inaccessible devices, potentially gaining access to further private data.

4.3 Data Processing

Once data has been successfully extracted from the device, it must then be processed to identify any potentially ransomable information. For network data, which is typically well structured, this is a computationally inexpensive process.

Less structured data, such as that which is collected from device sensors, can be much more difficult to interpret. While the use of machine learning can significantly reduce the amount of manual effort required to identify ransomable data, there are some logistical issues that attackers may need to overcome before it can be considered viable. Many IoT devices are unlikely to have the hardware to run the required machine learning models, and IoT devices' internal memory is often limited to only what is required to run the system, which may prevent collected data from being locally stored.

To circumvent these issues, attackers may instead process, classify, and store images collected by infected devices on remote systems. For example, attackers could choose to process collected data on their own server using publicly available models. However, this may not scale well, and a large ransomware campaign may cause immense network strain on the attacker's infrastructure, which could be quite costly to maintain. Therefore, it may become necessary to outsource processing to a third party, such as cloud services.

5 Data Collation

The privacy invasion methods we have discussed present possible avenues for ransomware authors to extract private information from IoT devices. However, using the extracted information to perform a ransomware attack in a large campaign presents multiple challenges, such as how to manage the collected data, how to generate an effective ransom note, and how the information could be published should the ransom not be paid. In this section, we will examine how these challenges may be approached by future attackers.

5.1 Data Management

As demonstrated in the previous section, there are various methods attackers may use to extract private data from victims. However, the collected data must be correctly managed for threats of publication to be effective. As part of this research, we created a basic proof of concept collator that allows the attacker to manage data collected from various compromised devices. An abstract view of the collator's operating structure can be found in Appendix A.1 (Figure 4).

The collator exposes an API for infected devices to interact with, allowing various types of private data to be uploaded, such as images, audio recordings or browsing history. Once data is received by the collator, it can be processed using the appropriate method, such as those described in Section 4.3. Each data point is associated with the infected device's MAC address, as it is an easily available unique identifier that is unlikely to change, even through reboots.

The attacker can then access the data processed by the collator via a web interface, shown in Appendix A.2 (Figure 5a). Additional features, such as highlighting particularly interesting collected information, such as valuble words in audio transcripts or private browsing activity, could also be implemented.

5.2 The Ransom

Once adequate personal information has been collected, a ransom note demanding payment can be generated and displayed to the victim. If any contact information has been extracted from the device, such as an email address, the ransom note could be sent directly to the user. Alternatively, the attacker could attempt to display the ransom note by hijacking communication methods native to the device, such as attached screens or network services [6].

Typically for ransomware attacks, the ransom note would likely contain a description as to what has occurred, a timer, and instructions for paying the ransom. However, unlike ransomware that prevents users from accessing their resources, privacy invasion ransomware threatens to release private information unless a ransom is paid before a certain time. Therefore, including select private information in the ransom note that has been obtained throughout the collection stage may provide sufficient evidence to force the victim into making a payment. By "personalising" ransom notes in this manner, it may lead less technically-aware victims to conclude that the attack was a manual effort made to target them specifically, which may further encourage payment.

5.3 Publishing Private Information

As part of a privacy-based ransomware attack, the victim is threatened with the release of their private information unless a payment is made. Private information could be publicised in a number of ways, varying in complexity.

Centralised Publication. One method attackers could use to publicise information is to create a centralised "leaking platform" available via a publicly accessible website. Any victims that do not make a payment would have their

information published to the website for anyone to view. As part of the ransom note, victims would be encouraged to visit the website for further information or to facilitate payment, acting as form of advertisement. Previous victims' private information would be visible to the "new users", which would serve as proof that the attacker will follow through with threats to publicise.

"Direct" Publication. Attackers could use information previously gathered about the victim to determine who would be most impacted by its release, such as friends, family or co-workers. For example, if the attacker identifies the victim's social media accounts during the information gathering stage, they may be able to enumerate people that the victim associates with. They could then attempt to use the same social media platforms to distribute the victim's private information, such as through the use of automated chat-bots. If this technique is used alongside the aforementioned leaking platform method of distribution, these messages could also serve to advertise it.

While this approach could drastically increase the impact of publicising information, it may also increase the complexity of the ransomware, as the attacker would need to automate account identification, enumeration and distribution for supported social media platforms.

5.4 Scale of Operation

Previously, such malware would require significant manual oversight. The automation steps outlined above, such as the use of machine learning and managing large volumes of data with a collator, would allow attacks to be performed without needing costly manual labour.

6 Proof of Concepts

To test the viability of privacy-based ransomware on IoT devices, we attempted to extract private information from a number of different device types, then collated it such that it could be used to ransom a user. For an attack to succeed, it is assumed that the attacker is able to access the vulnerable service such that they are able to exploit it remotely.

6.1 Netgear R6250 Router

As routers often act as the main gateway for Internet traffic in a network, we determined that they would be ideal for testing the network data extraction techniques discussed in Section 4.2. We chose to use a Netgear R6250 router for testing, which could be exploited using a previously discovered command injection vulnerability [31, 37].

Domain Extraction. To test extracting data from network activity, we created a program to sniff local packets using the `libpcap` library [25], which was cross-compiled to be compatible with the target router's architecture. The program

intercepts any packets destined for port 80 or 443 (the default ports for HTTP and HTTPS), extracts visited domain names and compares them against a hard-coded list. If a match is found, an API call is made to the collator, which records the visited domain, a timestamp of the visit, and the device's MAC address.

We created a network consisting of the R6250 router, a phone and a desktop computer. After exploiting the router, we uploaded and ran the application, then browsed various websites using the connected devices. The application successfully identified and reported domains visited using both HTTP and HTTPS to the collator, which the "attacker" was then able to view. For this proof of concept, we did not implement interpretation of any web content, but this could theoretically be implemented by a dedicated attacker in the future.

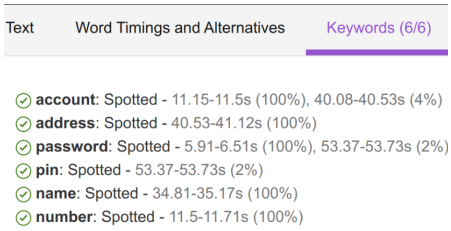
WiFi-Positioning. While the router did exhibit wireless capabilities, we were not able to scan for nearby SSIDs and MAC addresses. This may be due to limitations imposed by the expected usage of the device. However, we were able to view the local MAC address and SSID of the router, which could then be used to query a WiFi-Positioning service. While only one "signal" would be available for reference, which may reduce the result's accuracy, it should still allow attackers to make an approximate guess of the user's location, as WiFi signals have a limited range within which they can be detected.

Configuration Extraction. During the investigation of the device, we attempted to identify where user settings were being stored. We found that user settings were being saved to the second partition on the flash chip, which was accessible via the `/dev/mtdblock1` file. By using a simple `grep` command, we were able to view sensitive configuration data that was stored in plain text, as shown in Figure 1.

Ransom Note. Previous research has shown that it was possible to redirect DNS requests made to a compromised router [6]. Using this technique, an attacker could redirect users browsing the internet to a webpage containing a ransom note. In addition to traditional ransomware elements, such as a timer and a demand for payment, it could also include select personal information

```
# head /dev/mtdblock1 | grep 'wlg_passphrase\|http_username
\|http_passwd\|wla_passphrase\|pppoe_username\|pppoe_passwd
'
wlg_passphrase=adminlol
http_username=admin
ipv6_pppoe_username=
wla_passphrase_backup=
http_passwd=adminlol
wlg_passphrase_2=
wla_passphrase_2=
wla_passphrase_3=
wla_passphrase_4=
ipv6_pppoe_passwd=
pppoe_passwd=examplepw123
wlg_passphrase_backup=
pppoe_username=exampleemail@isp
wla_passphrase=adminlol
```

Fig. 1: Extracting configuration data



(a) IBM demo recognising keywords

(b) Ransom note hijacking the screen

Fig. 2: Attacking the Yealink SIP-T38g

collected by the malware to act as "proof of compromise". An example of how the ransom note could be presented is shown in Figure 5b in Appendix A.2.

6.2 Yealink SIP-T38g Phone

The SIP-T38g is an Internet connected IP phone with a built in LCD screen. As the device is designed for direct communication, we used it to test the audio extraction techniques described in Section 4.1.

Private Conversation Extraction. The first step for extracting private conversations is to obtain audio from the device when a call is made. While we could have potentially recorded audio directly from the device's microphone, we instead chose to extract call data from the device's network activity, as this would allow us to hear *both* sides of the conversation. To do this, we used VoIPong [12, 4], an open source tool that allows the interception and decoding of VoIP calls.

We modified, configured, and cross-compiled a custom version of VoIPong such that it would be able to run natively on the phone. We then exploited the device using a command injection vulnerability present in its web interface, allowing us to upload and run the application, which would then save calls to a pre-defined folder. Unfortunately, the phone had limited storage, with only a collective 60 megabytes of space across all the available partitions. To overcome this, we hosted a Network File System (NFS) share on the collator server, which the phone could then mount and modify as if it were a local directory. The collator then periodically checked for "file close" events within the share folder such that, when recordings were finished, conversations could be transcribed.

When the audio is ready to be processed, it is passed to a speech-to-text service for transcription. Initially, we attempted to use a local instance of Mozilla's "deepspeech" engine with a pre-trained model and scorer [35]. However, audio extracted from the intercepted calls were sampled at a rate of 8kHz, also known as "narrowband", while the Mozilla model expected a sample rate of 16kHz, which lead to unsatisfactory performance. While a new model could be trained to understand narrowband audio, it was considered to be out of scope for this paper. Instead, we tested various online services to transcribe the call accurately.

The Google Cloud Services API [23] successfully transcribed conversations with higher accuracy. We also tested using an "IBM Watson Speech to Text" demo [27] (which included support for narrowband audio), to successfully extract key components of the conversation. This demo also featured keyword identification, which could be used by attackers to listen for subjects of interest, as shown in Figure 2a. Finally, we were able to upload the call to YouTube after converting it to a video format. Approximately ten minutes after the initial upload, captions had been automatically added, and could be scraped from the source of the video's webpage. Given that YouTube provides this feature for free, it could potentially be used by attackers to avoid paying for the use of cloud services.

After the conversation has been transcribed, the text and audio file can be inserted into the collator. The attacker can then search for "valuable" words in the text, such as "password" or "address", as potential blackmail material. This entire process can be fully automated without giving the victim any indication that they are being monitored, until the ransom note is triggered.

Ransom Note. As with the R6250 router, the attacker could hijack the device's web server to display a ransom note, including "proof of compromise" such as recordings of the victim. However, as the web server is unlikely to be accessed in day to day usage, they could also hijack the connected screen [6], as shown in Figure 2b. It could be possible to expand to other communication media, such as using the speakers to play back recorded conversations, but this is unlikely to be unnecessary if the previous approaches are successful.

6.3 DCS-932L Camera

The DCS-932L is an Internet connected camera designed by D-Link. We selected this device to test WiFi-positioning based location extraction, and image based privacy invasion.

WiFi-Positioning. During our testing, we found that when the camera uses WiFi to connect to the Internet, it was possible to scan for nearby SSIDs and MAC addresses. We used a previously discovered buffer overflow exploit [44] to upload and run a WiFi scanning application, which returned information on three nearby access points. By uploading the access point information to Google Cloud Services we were able to determine our location within 15 meters.

Image extraction. As the camera is intended to be used for surveillance, this device was ideal for testing image based privacy invasion techniques. We found that during normal operation, the device would provide a snapshot from the camera to the user when they visited the web server. After infecting the device, we were able to make direct requests to this snapshot at `/image.jpg` on the local webserver. We uploaded an application that would save, encode and transfer images to the collator, which would then use Google Cloud services [16] to label recognised objects, locations and activities [21]. As shown in Figure 3, the platform was able to recognise and correctly label objects within the extracted images. If required, other services such as face detection [20] or explicit content detection [19] could also be applied with minimal changes.

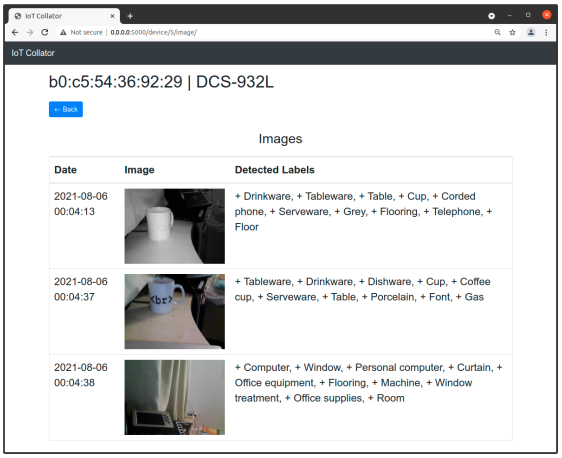


Fig. 3: Labelling images extracted from an infected DCS-932L Camera

Table 1: Privacy invasion methods used for each device

Device	Domain Extraction	Config Extraction	Audio Transcription	Image Recognition	Location Identification
Netgear R6250	✓	✓	-	-	Partial[1]
Yealink SIP-T38g	-	-	✓	-	-
D-Link DCS-932L	-	-	-	✓	✓

Ransom Note. The DCS-932L camera did not contain many methods to communicate with the user. As most interaction with the device was performed via the web service (which displays the current view from the camera), the attacker could use the same method as described in Section 6.1 to hijack the webserver to display a ransom note.

6.4 Summary

In this section, we demonstrated practical examples of how private information could be extracted from various IoT devices of differing types: router-based information, audio data and image data. We have also shown how the collected data could feasibly be analysed, organised, and used by an attacker to facilitate privacy invasion based IoT ransomware.

Table 1 provides a summary of the six privacy invasion methods that can be used, namely *Domain Extraction*, *Config Extraction*, *Audio Transcription*, *Image Recognition*, and *Location Identification*. Additionally, Table 1 also shows how these methods fare when applied against the three IoT devices we included in our proof of concepts.

[1] Unfortunately, we were unable to fully test the WiFi-positioning method for the R6250 router, as it was only powered when performing our analysis, preventing its MAC address from being detected or stored by any WiFi-positioning services.

While using IoT devices to invade the privacy of users has been theorised in the past, it has rarely been explored as a practical option for the average attacker. Here, we have shown several examples as to how such privacy invasions could potentially be monetised using ransomware, and how such attacks could be implemented at scale.

7 Discussion

Privacy-based IoT ransomware could have very negative impacts on users and their perception of IoT devices. Therefore, it is important to investigate potential countermeasures. Additionally, some limitations of our current work is discussed, along with several ideas for future research.

7.1 Countermeasures

There are a number of countermeasures that could be implemented by device developers, cloud providers, or IoT device users, as discussed below.

Domain Interception Protections. As shown in Section 6.1, it is possible for an attacker to extract the domains of websites that victims visit. While users can protect themselves by using privacy tools such as VPNs or Tor [46], it is unrealistic to suggest every user use such tools just in case one of their devices is infected with such malware. Alternative methods to secure communication between users and web services must instead be implemented by website hosts.

As HTTP traffic is designed to be unencrypted by default and requires the domain to be included within the headers, it is very simple to extract information from any traffic generated by the victim. By using HTTPS, the user can limit the information that an attacker can extract through the use of encryption. However, as mentioned in Sections 4.2 and 4.2, it is still possible to extract the visited domain or perform downgrade attacks. These attacks can be prevented through the use of:

- *Encrypted Server Name Indication (ESNI).* While the contents of HTTPS communication is encrypted, the domain can be extracted from the SNI portion of HTTPS handshake packets. Encrypting this portion of the header using a compatible DNS server will prevent the attacker from being able to discern the visited domain [10]. Encrypted Client Hello (ECH), a more recent protection mechanism, could also be used to prevent domain extraction in the future [40].
- *HTTP Strict Transport Security (HSTS).* In Section 4.2, HTTPS downgrade attacks were highlighted as a possible method for intercepting the contents of web service communications. HSTS allows web hosts to force clients to only use HTTPS when visiting their domain, preventing such downgrade attacks. Some of the most popular browsers even contain hard-coded lists of HTTPS-only websites by default [17].

Malicious Activity Detection in Cloud Services. Currently, attackers may find it difficult to natively implement software on infected IoT devices that can process data collected from its sensors, such as object recognition on captured images. While this may change in the future – either through more cost-efficient machine learning algorithms, or more resources being made available on the average IoT device – attackers are currently more likely to rely on outside processing, such as online cloud services. As such, attackers may need to use these cloud services at scale in order to adequately manage the throughput of infected devices. Cloud providers may be able to detect such malicious behaviour through the measuring of various metrics, such as:

- An account using multiple IP addresses to call the API, which may imply that functions are being called directly from infected IoT devices.
- "Privacy related" functions being called excessively or in certain sequences, such as facial or object recognition followed by nudity detection.
- Whether a trial account is being used, as it may imply that the attacker is aiming to reduce costs by using free processing without payment.

If the cloud service provider is able to identify a user as malicious, banning or shutting down the associated account may delay the operation of the malware campaign. A more extreme approach may be to prevent accounts from accessing certain functionality commonly associated with privacy based ransomware until the owner has provided sufficient proof of identity.

Data Devaluation. If a victim is threatened with the public release of their private data, there are very few steps that they can take to reduce the impact, as they do not have any method to remove the stolen data from the attacker's storage. However, it may be possible to reduce the trustworthiness of information attained by the attackers by providing false data to the C&C server, thus reducing the overall value of files that are released. This may also waste the attacker's time and resources, as they would need to receive, store and analyse any data sent by the fake "victim".

Updating. While this has often been mentioned, it is worth re-enforcing the principle that applying updates and patches, and changing default passwords, are important steps in securing IoT devices against possible compromise.

7.2 Limitations and Further Work

Countermeasure Creation. Due in part to the variety in the design of IoT devices, the creation of universal countermeasures is not a simple process. While the countermeasures discussed above can be effective, it could be argued that some are only applicable in certain scenarios. This work highlights the need for further research as to how IoT devices can be designed to limit the effectiveness of privacy-invasion based malware.

Native Malicious Machine Learning. Currently, the identification and management of data presents a significant hurdle that attackers must overcome in order to create effective privacy invasion based ransomware. The infrastructure required to transfer, store, and process collected data may dissuade malicious actors from attempting to perform these types of attacks. However, as the hardware present in IoT devices continues to improve, and machine learning techniques become increasingly efficient, it may eventually be possible to run machine learning tools natively on infected devices rather than outsourcing the data processing. It may be beneficial to investigate the viability of such native tools, as it may heavily reduce the costs when running a large malware campaign.

Psychological Effects. Unlike other malware, which typically targets the restriction of information, privacy based ransomware instead threatens to expose it, which has the potential of being very distressing for victims. A study of the psychological effects of this malware could reveal the non-monetary costs of infection, such as how public perception may change concerning IoT devices, should this affect a significant number of devices.

ARP Poisoning. In Section 4.2 we described techniques that intercept network traffic to extract private information. Typically, these require the infected device to be positioned such that it is a "man in the middle" (MitM), with the user's network activity passing through it. Routers are perfectly positioned for this type of attack. However, devices that do not hold this position, such as network cameras, will only be able to examine their own network activity.

A possible way that infected IoT devices could use is an Address Resolution Protocol (ARP) poisoning attack, which would allow attackers to insert themselves in-between the network gateway and another target [49]. If IoT devices are shown to be capable of performing such attacks, they may be able to use MitM attacks on other devices on the same network without acting as the gateway.

8 Conclusions

In this paper we investigated how IoT devices could be used to facilitate privacy-invasion based ransomware targeting consumers. To do this, we first examined various data sources commonly found on IoT devices and how they could be leveraged by attackers to extract data. We then proposed methods attackers could use to identify and process that data to extract sensitive user information for the purpose of performing a ransomware attack. We discussed how automated machine learning and data collation could be used to manage data collected from vulnerable IoT devices to perform ransomware attacks at a large scale.

We showed how some of the privacy-invasion techniques could be realised on three IoT devices with differing sensors and data sources. During the demonstration, we were able to extract various mock "private data" and send it to a remote data collation service, such that an attacker could easily track and process it.

We then discussed potential countermeasures that could be implemented by users or IoT developers to prevent or reduce the impact of such attacks, before finally identifying the work's limitations and opportunities for future research.

A Appendices

A.1 Data Collator Structure

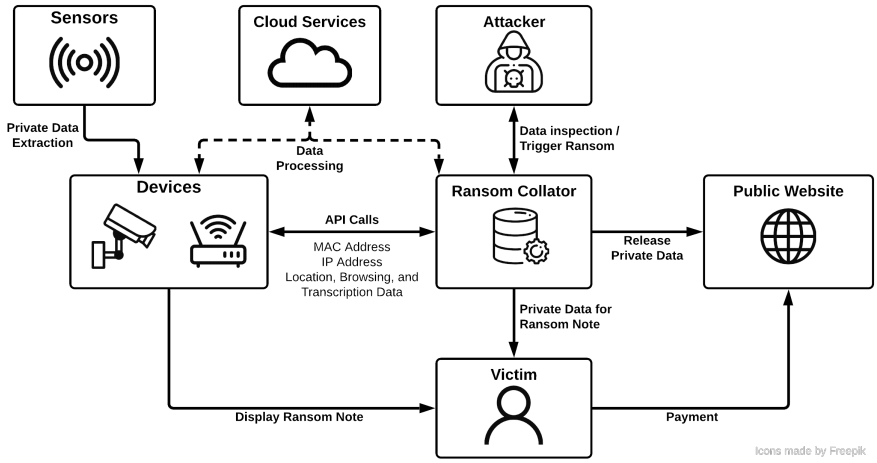
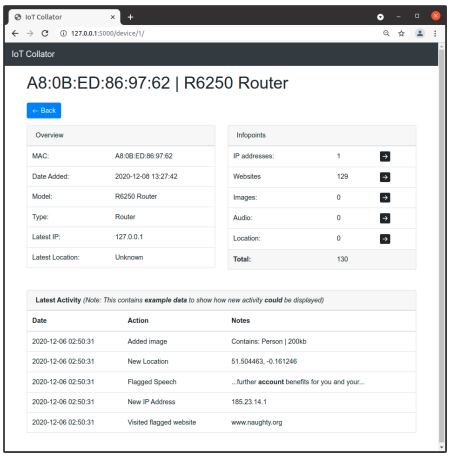
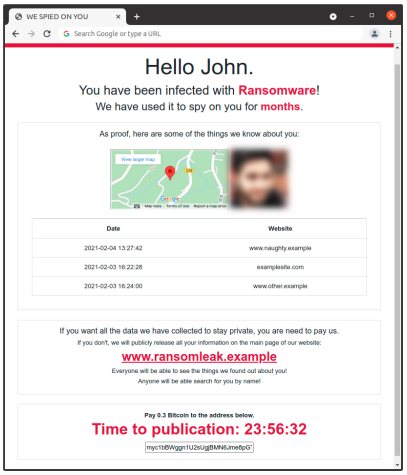


Fig. 4: Data Collator Structure

A.2 Collator and Ransom Note



(a) IoT Collator summarising information collected from a router



(b) An example ransom note, including proof of compromise

Fig. 5: Collator and example ransom note

References

1. abcNEWS: Terrifying video of family's hacked ring camera system (2019), https://abcnews.go.com/GMA/News/video/terrifying-video-familys-hacked-ring-camera-system-67704081/ [Accessed: June 2021]
2. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., et al.: Understanding the mirai botnet. In: 26th {USENIX} security symposium ({USENIX} Security 17). pp. 1093–1110 (2017)
3. Arias, O., Wurm, J., Hoang, K., Jin, Y.: Privacy and security in internet of things and wearable devices. IEEE Transactions on Multi-Scale Computing Systems **1**(2), 99–109 (2015)
4. Balaban, M.: Voipong user's manual (2005), http://www.enderunix.org/voipong/manual/ [Accessed: April 2021]
5. Bitdefender: Security 2020 consumer threat landscape report (2021), https://www.bitdefender.com/files/News/CaseStudies/study/395/Bitdefender-2020-Consumer-Threat-Landscape-Report.pdf [Accessed: July 2021]
6. Brierley, C., Pont, J., Arief, B., Barnes, D.J., Hernandez-Castro, J.: Paperw8: an iot bricking ransomware proof of concept. In: Proceedings of the 15th International Conference on Availability, Reliability and Security. pp. 1–10 (2020)
7. Brierley, C., Pont, J., Arief, B., Barnes, D.J., Hernandez-Castro, J.: Persistence in linux-based iot malware. In: Secure IT Systems: 25th Nordic Conference, NordSec 2020, Virtual Event, November 23–24, 2020, Proceedings 25. pp. 3–19. Springer (2021)
8. @CDPROJEKTRED: Important update (2021), https://twitter.com/CDPROJEKTRED/status/1359048125403590660 [Accessed: June 2021]
9. Center, I.C.C.: Internet crime report 2020 (2021), https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf [Accessed: July 2021]
10. Chai, Z., Ghafari, A., Houmansadr, A.: On the importance of encrypted-sni ({ESNI}) to censorship circumvention. In: 9th {USENIX} Workshop on Free and Open Communications on the Internet ({FOCI} 19) (2019)
11. Encrypt, L.: Let's encrypt stats, https://letsencrypt.org/stats/ [Accessed: July 2021]
12. EnderUNIX: Voipong (2011), https://github.com/EnderUNIX/VoIPong [Accessed: July 2021]
13. Fabian Bräunlein, L.F.: Smart spies: Alexa and google home expose users to vishing and eavesdropping (2019), https://www.srlabs.de/bites/smart-spies [Accessed: July 2021]
14. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: Rfc2616: Hypertext transfer protocol–http/1.1 (1999)
15. Goodin, D.: Cd projekt red does an about-face, says ransomware crooks are leaking data (2021), https://arstechnica.com/gadgets/2021/06/cd-projekt-red-says-its-data-is-likely-circulating-online-after-ransom-attack/ [Accessed: June 2021]
16. Google: Cloud computing services — google cloud, https://cloud.google.com/ [Accessed: July 2021]
17. Google: Http strict transport security, https://www.chromium.org/hsts/ [Accessed: July 2021]
18. Google: Https encryption on the web, https://transparencyreport.google.com/https/overview [Accessed: July 2021]

19. Google: Detect explicit content (safesearch) (2021), https://cloud.google.com/vision/docs/detecting-safe-search [Accessed: August 2021]
20. Google: Detect faces (2021), https://cloud.google.com/vision/docs/detecting-faces [Accessed: August 2021]
21. Google: Detect labels (2021), https://cloud.google.com/vision/docs/labels [Accessed: August 2021]
22. Google: Geolocation api (2021), https://developers.google.com/maps/documentation/geolocation/overview [Accessed: July 2021]
23. Google: Method: speech.recognize (2021), https://cloud.google.com/speech-to-text/docs/reference/rest/v1/speech/recognize [Accessed: July 2021]
24. Group, T.I.: Vpnfilter update - vpnfilter exploits endpoints, targets new devices (2018), https://blog.talosintelligence.com/2018/06/vpnfilter-update.html [Accessed: July 2021]
25. Group, T.T.: Tcpdump/libcap public repository (2021), https://www.tcpdump.org/ [Accessed: July 2021]
26. Hron, M.: The fresh smell of ransomed coffee (2020), https://decoded.avast.io/martinhron/the-fresh-smell-of-ransomed-coffee/ [Accessed: July 2021]
27. IBM: Speech to text demo, https://speech-to-text-demo.ng.bluemix.net/ [Accessed: July 2021]
28. Ilascu, I.: Hacker used ransomware to lock victims in their iot chastity belt (2021), https://www.bleepingcomputer.com/news/security/hacker-used-ransomware-to-lock-victims-in-their-iot-chastity-belt/ [Accessed: June 2021]
29. Kalbo, N., Mirsky, Y., Shabtai, A., Elovici, Y.: The security of ip-based video surveillance systems. Sensors **20**(17), 4806 (2020)
30. Kharraz, A., Robertson, W., Balzarotti, D., Bilge, L., Kirda, E.: Cutting the gordian knot: A look under the hood of ransomware attacks. In: International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. pp. 3–24. Springer (2015)
31. Land, J.: Multiple netgear routers are vulnerable to arbitrary command injection (2016), https://www.kb.cert.org/vuls/id/582384/ [Accessed: July 2021]
32. Marlinspike, M.: New tricks for defeating ssl in practice. Black Hat DC **2** (2009)
33. Mockapetris, P.: Domain names - concepts and facilities (1987), https://datatracker.ietf.org/doc/html/rfc1034\#section-5.3.2 [Accessed: July 2021]
34. Mohurle, S., Patil, M.: A brief study of wannacry threat: Ransomware attack 2017. International Journal of Advanced Research in Computer Science **8**(5), 1938–1940 (2017)
35. Morais, R.: Deepspeech 0.9.3 (2020), https://github.com/mozilla/DeepSpeech/releases/tag/v0.9.3 [Accessed: July 2021]
36. Mozilla: Geolocate (2020), https://ichnaea.readthedocs.io/en/latest/api/geolocate.html [Accessed: July 2021]
37. NIST: Cve-2016-6277 detail (2017), https://nvd.nist.gov/vuln/detail/CVE-2016-6277 [Accessed: July 2021]
38. Orland, K.: Cd projekt red source code reportedly sells for millions in dark web auction [updated] (2021), https://arstechnica.com/gaming/2021/02/cd-projekt-red-source-code-reportedly-sells-for-millions-in-dark-web-auction/ [Accessed: June 2021]
39. Palmer, D.: Hackers publish thousands of files after government agency refuses to pay ransom (2021), https://www.zdnet.com/article/hackers-publish-

thousands-of-files-after-government-agency-refuses-to-pay-ransom/ [Accessed: July 2021]
40. Patton, C.: Good-bye esni, hello ech! (2020), https://blog.cloudflare.com/encrypted-client-hello/ [Accessed: July 2021]
41. SonicWall: Sonicwall cyber threat report (2021), https://www.sonicwall.com/medialibrary/en/white-paper/2021-cyber-threat-report.pdf [Accessed: July 2021]
42. Sun, K., Chen, C., Zhang, X.: "alexa, stop spying on me!" speech privacy protection against voice assistants. In: Proceedings of the 18th Conference on Embedded Networked Sensor Systems. pp. 298–311 (2020)
43. Surbatovich, M., Aljuraidan, J., Bauer, L., Das, A., Jia, L.: Some recipes can do more than spoil your appetite: Analyzing the security and privacy risks of ifttt recipes. In: Proceedings of the 26th International Conference on World Wide Web. pp. 1501–1510 (2017)
44. tacnetsol: Cve-2019-10999 (2019), https://github.com/tacnetsol/CVE-2019-10999 [Accessed: July 2021]
45. Tidy, J.: Cyber criminals publish more than 4,000 stolen sepa files (2021), https://www.bbc.co.uk/news/uk-scotland-55757884 [Accessed: June 2021]
46. Tor: Tor project — anonymity online, www.torproject.org/ [Accessed: July 2021]
47. TrendMicro: Exposed video streams: How hackers abuse surveillance cameras (2018), https://www.trendmicro.com/vinfo/us/security/news/internet-of-things/exposed-video-streams-how-hackers-abuse-surveillance-cameras [Accessed: June 2021]
48. TrendMicro: Over 200,000 mikrotik routers compromised in cryptojacking campaign (2018), https://www.trendmicro.com/vinfo/nl/security/news/cybercrime-and-digital-threats/over-200-000-mikrotik-routers-compromised-in-cryptojacking-campaign [Accessed: July 2021]
49. Whalen, S., Engle, S., Romeo, D.: An introduction to arp spoofing. Node99 [Online Document] (2001), https://www.cavalcantetreinamentos.com.br/blog/material-sala-de-aula/Seguranca em Redes/Outros/arp_spoofing_slides.pdf
50. Zhang, N., Mi, X., Feng, X., Wang, X., Tian, Y., Qian, F.: Understanding and mitigating the security risks of voice-controlled third-party skills on amazon alexa and google home. arXiv preprint arXiv:1805.01525 (2018)
51. Zhang, Y., Sun, Z., Yang, L., Li, Z., Zeng, Q., He, Y., Zhang, X.: A11 your plcs belong to me: Ics ransomware is realistic. In: 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). pp. 502–509. IEEE (2020)