

Ransomware

October 18, 2022

Overview

Ransomware, a type of malware that is constantly evolving, is a major and pervasive threat to computer security. A ransomware attack encrypts a victim's data, rendering files and the systems that rely on them useless. Cybercriminals demand a ransom to decrypt the files and require the user to pay a fee, usually in cryptocurrency, to regain access to the contents. Ransomware attacks have grown at an alarming rate since they were first reported. In fact, ransomware affected 37% of all businesses and organizations in 2021, costing the world \$20 billion, a figure that is expected to rise to \$265 billion by 2031, (www.cloudwards.net/ransomware-statistics).

In this lab, we will learn and explore the structure of how Ransomware could work, including infection, encryption, remuneration, and decryption processes.

To do this, we will utilize several Docker containers and RansomWhale, a ransomware I created that targets all of the files in a user's home directory.

Contents

1	Ransomware Infection	2
1.1	Lab Prep (Docker)	2
1.2	Infection	2
2	File Encryption	3
2.1	Exploring the Victim's files	3
2.2	Exploring the Attacker's files	3
2.3	Getting RansomWhale'd	4
3	Remuneration and Decryption	5
3.1	Ransomware Aftermath	5
3.2	Payment	5
3.3	Key Exchange	6
3.4	File Decryption	6
4	Troubleshooting	7
5	General Questions	7

1 Ransomware Infection

A ransomware campaign can be launched by an attacker using a variety of techniques, such as phishing, social engineering, spam, and advertisements. One of the most popular methods an attacker may deploy ransomware is by discreetly and innocently sending out phishing emails with malicious links and/or attachments so the victim won't notice.

In this section, you will see an example of a malicious website built to persuade users into downloading the software.

1.1 Lab Prep (Docker)

Go to your local GitHub folder and pull the files using the git command below.

```
git clone https://github.com/imtaylorgriffin/RansomWhale.git
# Use "git pull" if you have cloned the project before
```

Next, have two separate terminal windows open and start the container instances:

```
docker-compose up -d
# Starts all containers, make sure you're in the Ransomwhale folder

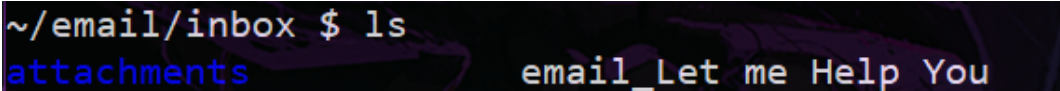
docker container exec -it victim /bin/sh
# Victim terminal - open in terminal window 1

docker container exec -it attacker /bin/sh
# Attacker terminal - open in terminal window 2
```

1.2 Infection

In this section, we will utilize Docker to look at how one may get infected with Ransomware.

Go into the victim terminal and `cd ~/email/` this is where your Inbox and Sent emails will show up. Go ahead and `cd` to your inbox and check the files with `ls`.



```
~/email/inbox $ ls
attachments      email_Let me Help You
```

It looks like you have a new email!. Go ahead and `cat` the email file and follow the directions.

After following the directions, you should be greeted by the HomeworkHeroes website. This is our Attacker's website, many websites will try to look as legitimate as possible in order to persuade you to download their software.

Click the Download button.

```
# It is very important you do not open the application yet.
# Move the downloaded HomeworkHeroes.py to Ransomware/victim/Downloads
```

2 File Encryption

Following the execution of the Ransomware, it begins encrypting all of the victim's non-boot dependent files, such as documents, videos, photos, and other personal data.

Normally after encryption, a pop-up window appears telling you that your files have been encrypted, along with payment procedures to follow if you wish to restore them.

When you try to open an encrypted file, it will fail to operate normally, and examining it in a text editor will disclose the contents as a jumbled up output of characters.

This is because Ransomware encrypts the file's bytes, resulting in strange-looking content and the inability to run the file normally.

It is also a popular practice to change the extension of the encrypted files, an example would be .locked, .encrypted, and .locky, to name a few. This shows a very apparent indication that the victim's files have been modified by an attacker.

2.1 Exploring the Victim's files

If you haven't already, go into your Victim and Attacker terminals with:

```
docker container exec -it victim /bin/sh
# Victim terminal
```


```
docker container exec -it attacker /bin/sh
# Attacker terminal
```

In the Victim terminal, `cd ~` to go to your home folder. You can go ahead and `cat` the example files so you can see what they look like.

2.1.1 (Optional) You can create a file with `vi example.txt` and type whatever you'd like.

2.2 Exploring the Attacker's files

Go into your Attacker's terminal, if you `ls` you should see the email directory and an application called `listener.py`



```
~ $ ls
email      listener.py
~ $
```

RansomWhale uses asymmetric encryption, so it works by first creating Public and Private RSA keys, after creation, the Private key will secretly be sent to the Attacker, so it is important the Attacker first uses `listener.py` to listen for, and store, the Private key information.

On the attacker terminal, start the listener application using *python listener.py* you should see the following screen:

```
~ $ ls
email      listener.py
~ $ python listener.py
< ☺ > Listening as 0.0.0.0:5001....
```

2.3 Getting RansomWhale'd

After starting the *listener.py* application on your Attacker's terminal, head over to your Victim's terminal and *cd ~/app/* Once here, *ls* to see any changes.

You may notice that there is a new file present!

```
~ $ cd ~/app/
~/app $ ls
HomeworkHeroes.py
```

When you're ready, and made sure *listener.py* is running, use *python HomeworkHeroes.py* to start RansomWhale!

2.3.1 Take a screenshot of your Victim's terminal

2.3.2 Take a screenshot of your Attacker's terminal

If successful, all of the Victim's home folder files should be encrypted, besides the ransomware and a few other things.

ls on your Attacker's terminal and you should see a *private_key.pem* file.

```
~ $ ls
email      listener.py  private_key.pem
```

3 Remuneration and Decryption

Oh no! All of your important files have been encrypted! What do you do?

After file encryption, the average user may have no idea how to resolve their issue. The attacker expects you to pay the ransom in order for them to restore your files.

When RansomWhale's Private key was generated, it was immediately sent to the attacker and wiped from the victim's computer. RansomWhale is designed to both encrypt and decrypt files, but it can only decrypt when its original Private Key is present. So, after paying the ransom, the attacker will send the necessary key to the victim, and all the victim has to do is place it in the same folder.

In this section, we are going to simulate payment of the attacker and the decryption of the victim's files.

3.1 Ransomware Aftermath

View the changes in your home folder by going on your Victim's terminal, `cd ~` to go back to your home folder and `ls`.

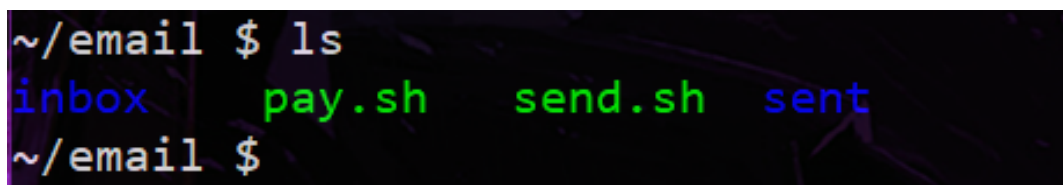
3.1.1 Take a screenshot of your Victim's home folder files.

`cat` a few of the files to view their outputs, You should notice that all of the outputs are messed up, this is an indication that the bytes of the files have been modified!

3.1.2 (Optional) `cp` a sampleFile over to your app directory and try to open it outside of docker (RansomWhale/victim/Downloads) and describe your result.

3.2 Payment

To simulate the payment of the ransom, I have included a small payment script. On the Victim's terminal, `cd ~/email/` and `ls` you should see both a `pay.sh` and `send` script:

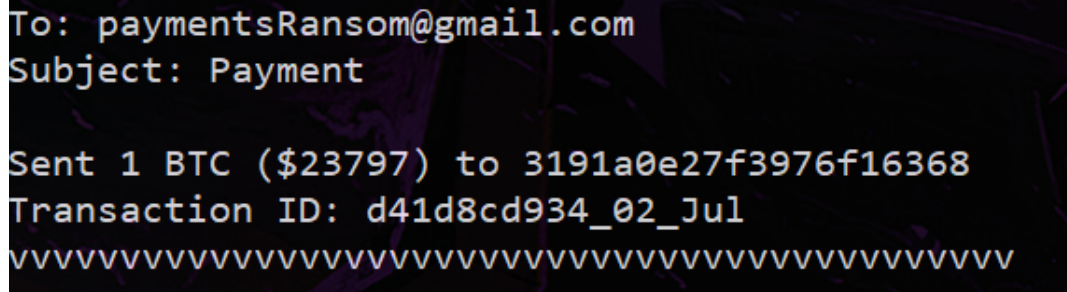


```
~/email $ ls
inbox      pay.sh     send.sh    sent
~/email $
```

If needed, reread the ransom note by `cat ~/app/ransom.note` to make sure you send the correct amount of BTC. To start the payment script, use `sh pay.sh` and it should ask you to input the value. After the payment is sent, highlight the Sent and Transaction ID lines and copy them.

Now to send your email! use `vi email_payment` and paste the lines you copied like so:

After that, save your file and go ahead and use `sh send.sh` to send your email!



3.2.1 Is paying the ransom a good idea? Why might an organization pay the ransom?

3.2.2 Is it possible to avoid paying the ransom? How else might you solve this issue?

3.3 Key Exchange

After receiving payment, it is the Attacker's turn to respond. Switch over to your Attacker's terminal

```
cd email/inbox/  
# Check for new emails!  
# If the victim's email successfully went through, it should show up.  
  
cat email_payment  
# To read it.
```

After reading the receipt, you should email the private key over by

```
mv ~/private_key.pem ~/email/ && sh send.sh
```

To verify if the email was successfully sent, `ls ~/email/sent/attachments/`

3.4 File Decryption

Finally, we are now able to restore our files back to normal, all we have to do is grab the key.

In the Victim's terminal,

```
ls ~/email/inbox/attachments  
# you should see the private_key.pem file!  
  
mv ~/email/inbox/attachments/private_key.pem ~/app/  
# to place it in the same folder as RansomWhale  
  
cd to the app directory.
```

To start the decryption process, make sure you have both the Public and Private key in the same directory with HomeworkHeroes.py, then use `python HomeworkHeroes.py`

3.4.1 Take a screenshot of the output

Lastly, `cd ~` to go to your home directory and view the files. `cat` a few of the files, and if successful, they should all have their original data!

4 Troubleshooting

Common issues and ways to fix them

4.0.1 Want to reset the containers and redo?

Make sure you are in the RansomWhale folder:

```
docker-compose up -d --build --remove-orphans --force-recreate  
# Forcefully recreates everything
```

4.0.2 HomeworkHeroes.py not running?

Make sure you started the listener application first in Attacker's terminal using:

```
python listener.py  
# Starts the application, will close after receiving key info
```

4.0.3 Not able to start the containers?

Make sure you are inside the cloned RansomWhale file and then use:

```
docker-compose up -d  
# Starts all the containers
```

5 General Questions

5.0.1 Any corrections or general comments about this lab?

5.0.2 How long did it take you to complete this lab?

5.0.3 Was it an appropriate length lab?