

# Ransomeware project

Team Members:

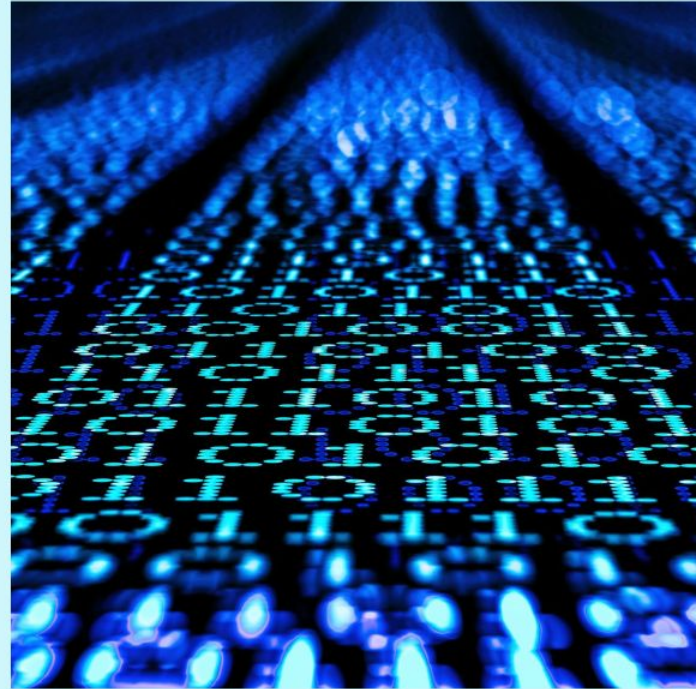
Muvva Hima sai -- 11695369

Tagore Hari Prasad Chintamaneni -- 11702057

Ajay Eedara -- 11702053

Venkata Naga sai hemanth varma kunaparaju --  
11701764

Jaideep Tripurani -- 11709159



# What is Ransomware

- Ransomware is a malicious software or we can it as code designed to block access to a computer system or files until ransom is paid.
- System may be Server or Client.
- They maintained a secure channel for communication and attackers mostly ask ransom through cryptocurrency.
- Its developed using cryptography.

# Cryptography

- Why Cryptography?

Because Ransomware is implemented using Cryptography techniques.

- Types of Cryptography

Symmetric and Asymmetric cryptography

# Symmetric and Asymmetric

- Symmetric is implemented with one key for Encryption and Decryption of data.
- Asymmetric is implemented with two keys that is public and private key for encryption and decryption.
- In this Project We have implemented the Symmetric Key Encryption.

# Our Implementation Plan

- We have developed a ransomware using python
- Converted the python to executable file
- Sended the exe file over link to victim
- Victim installed the exe because of phishing
- Implemented the Monitoring, Detection and Mitigation tools from stopping the Ransomware attack.

# Action

- Encryption

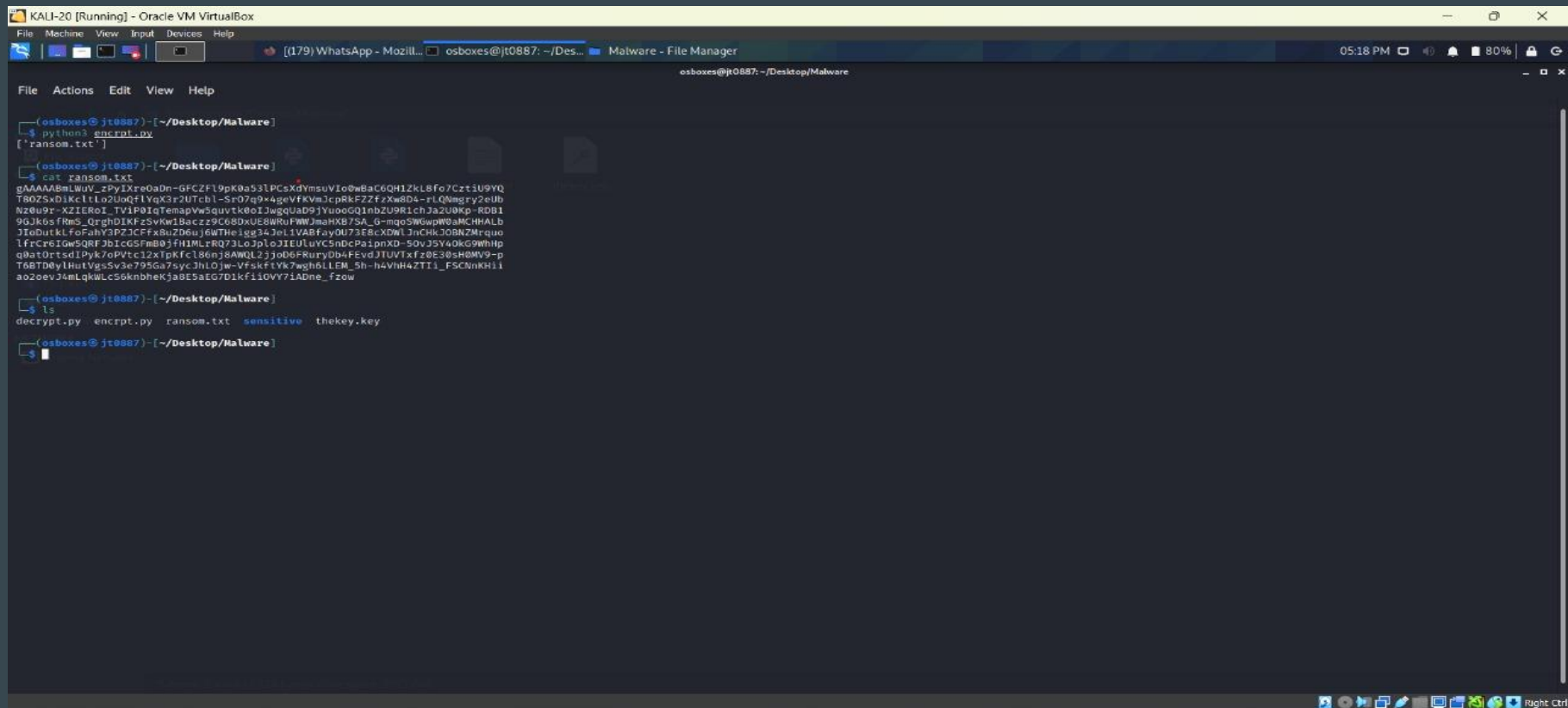
We have implemented the python code which works on symmetric method which encrypts the directory recursively.

- Decryption

We have implemented the python code which works on symmetric method and decrypts the directory recursively.

Next We have the image of implementation.

# Encryption Image



The screenshot shows a Kali Linux virtual machine running on Oracle VM VirtualBox. The terminal window is titled "KALI-20 [Running] - Oracle VM VirtualBox" and displays the following commands and output:

```
(osboxes@jt0887)-[~/Desktop/Malware]
$ python3 encrypt.py
['ransom.txt']

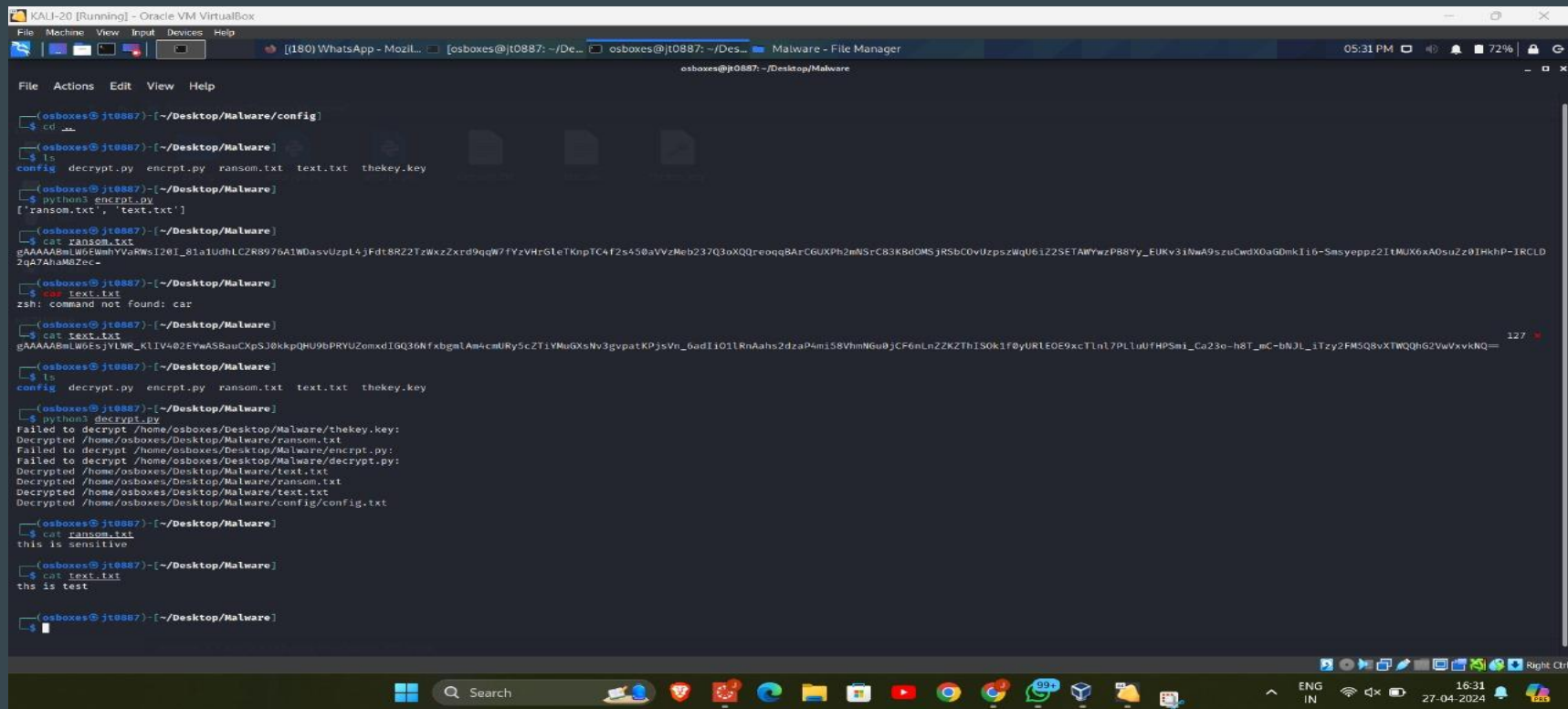
(osboxes@jt0887)-[~/Desktop/Malware]
$ cat ransom.txt
gAAAAABmLWuV_zPyIXre0aDn-GFCZF\9pK0a53\PCsXdYmsuVio@wBaC6QH12kL8fo7CztIu9YQ
T80Z5xdiKcltLo2UoQfIYqX3r2UTchl-Sr07q9+4geVfKvMjcpRkFZfzXw8D4-rlQWagry2eUb
Nzu09r-XZIERoI_TVIP8iqTemapVw5quvtk0eI3wgQaD9jYuo0GQ1nbZU9R1chJa2U0Kp-RDBI
90Jk6sFm6-QrghD1KfzSVKwI8aczz9C68DXUE8WRuFWWJmahXB7SA_g-mgo5N6wpW0sKCHALB
JI0utKtfoFahY3P2zCfz8uZ06uj6wThe8g343et1VABfay0U73f8X0wL3nChk308M7Mrquo
lfrcr6Igw5QRF3bIcG5Fm0jfhIIMLrRQ73Lo3ploJIEULuYC5nDcPaipnXD-50vJ5Y40K9Wnhp
q0at0rtsdIPyk7oPvtcl2xTpKfcl86n38AWQL2jjoD6FRuryDb4FevdJTUVTxfz0E30sH0MV9-p
T68TD0yIHutVgsSv3e795Ga7syc3hLOjw-VfskftYk7wgh6LLEM_5h-h4VhH4ZTTi_FSCNnKHii
a0z0evJ4mLqkWLc56knbheKja8E5aEGD1Kf110VY71ADne_fzow

(osboxes@jt0887)-[~/Desktop/Malware]
$ ls
decrypt.py  encrypt.py  ransom.txt  sensitive  thekey.key

(osboxes@jt0887)-[~/Desktop/Malware]
$
```

The terminal output shows a long string of alphanumeric characters, likely a ransomware key or a large file hash. The file manager window in the background shows the contents of the ~/Desktop/Malware directory, including the files listed in the terminal output.

# Decryption image



```
KAU-20 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
[(180) WhatsApp - Mozil... [osboxes@jt0887: ~/De... osboxes@jt0887: ~/Des... Malware - File Manager 05:31 PM 72%
osboxes@jt0887: ~/Desktop/Malware

File Actions Edit View Help

(osboxes@jt0887) [~/Desktop/Malware/config]
$ cd ..

(osboxes@jt0887) [~/Desktop/Malware]
$ ls
config decrypt.py encrypt.py ransom.txt text.txt thekey.key

(osboxes@jt0887) [~/Desktop/Malware]
$ python3 encrypt.py
['ransom.txt', 'text.txt']

(osboxes@jt0887) [~/Desktop/Malware]
$ cat ransom.txt
gAAAAA8mLW5FWmhYVaRwsI20I_81a1udhLC2R8976A1W0asvUzpL4jFdt8R22TzWkzZxrd9qqW7fYzVHRtKtKnPTC4f2s450aVvzMeB237Q3oxQQre0qg8ArCGUXPh2mN8rC83KBdDMSjRSbC0vUzpsrWqU6iZ2SETAWYwzP88Yy_EUKv31NwA9szuCdWx0aG0mkIi6-Smsyoppz21tMUX6xA0suZz0IHkHP-IRCLD
2qA7AhaM6Zec-

(osboxes@jt0887) [~/Desktop/Malware]
$ cat text.txt
zsh: command not found: cat

(osboxes@jt0887) [~/Desktop/Malware]
$ cat text.txt
gAAAAA8mLW5FWmhYVaRwsI20I_81a1udhLC2R8976A1W0asvUzpL4jFdt8R22TzWkzZxrd9qqW7fYzVHRtKtKnPTC4f2s450aVvzMeB237Q3oxQQre0qg8ArCGUXPh2mN8rC83KBdDMSjRSbC0vUzpsrWqU6iZ2SETAWYwzP88Yy_EUKv31NwA9szuCdWx0aG0mkIi6-Smsyoppz21tMUX6xA0suZz0IHkHP-IRCLD
2qA7AhaM6Zec-

(osboxes@jt0887) [~/Desktop/Malware]
$ ls
config decrypt.py encrypt.py ransom.txt text.txt thekey.key

(osboxes@jt0887) [~/Desktop/Malware]
$ python3 decrypt.py
failed to decrypt /home/osboxes/Desktop/Malware/thekey.key:
Decrypted /home/osboxes/Desktop/Malware/ransom.txt
Failed to decrypt /home/osboxes/Desktop/Malware/encrypt.py:
failed to decrypt /home/osboxes/Desktop/Malware/decrypt.py:
Decrypted /home/osboxes/Desktop/Malware/text.txt
Decrypted /home/osboxes/Desktop/Malware/ransom.txt
Decrypted /home/osboxes/Desktop/Malware/text.txt
Decrypted /home/osboxes/Desktop/Malware/config/config.txt

(osboxes@jt0887) [~/Desktop/Malware]
$ cat ransom.txt
this is sensitive

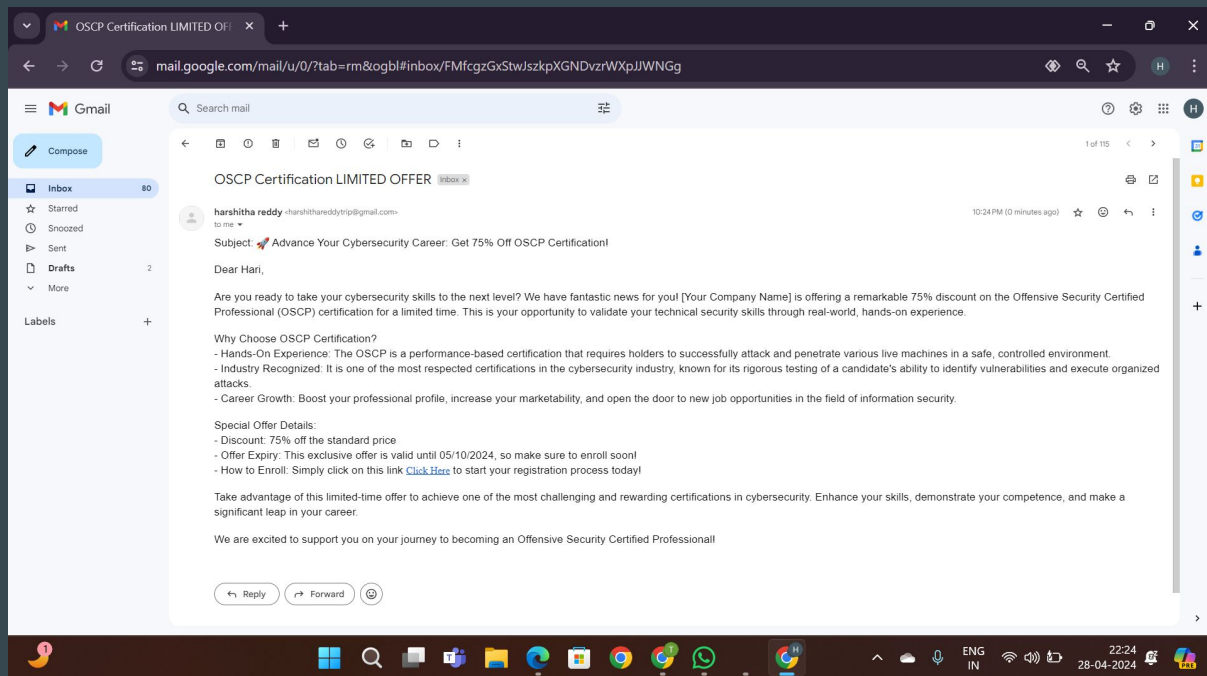
(osboxes@jt0887) [~/Desktop/Malware]
$ cat text.txt
this is test

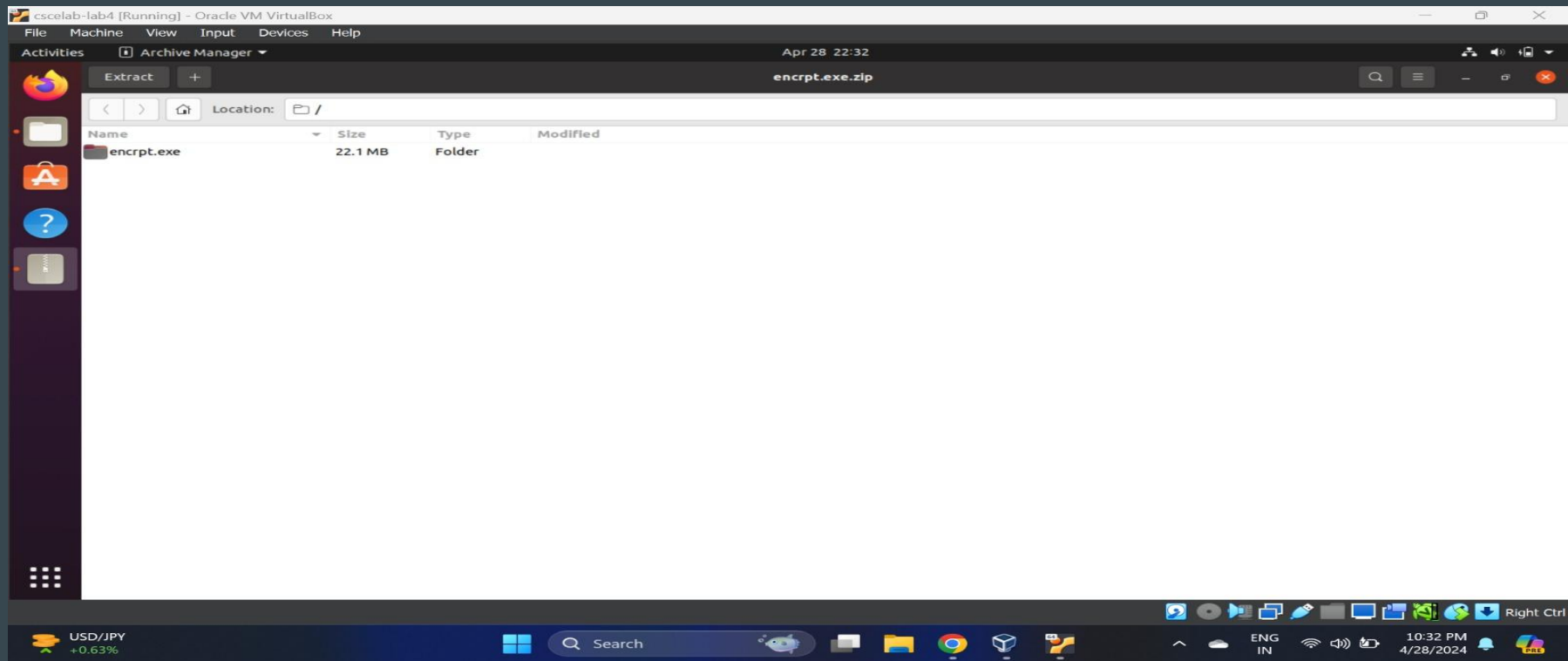
(osboxes@jt0887) [~/Desktop/Malware]
$
```



# Infection

- We have generated the exe file from python file
- Sended the exe file as a link to the victim which is a phishing mail.





As you can see the file has been downloaded into victim's system after victim click on the link which we have sent through mail.

# Monitoring

- We have used tools like apparmor for monitoring the system for file changes and configured the file permissions based on the policy.
- We have used the windows defender for monitoring the system for malware monitoring .

# Monitoring Policy Apparmor

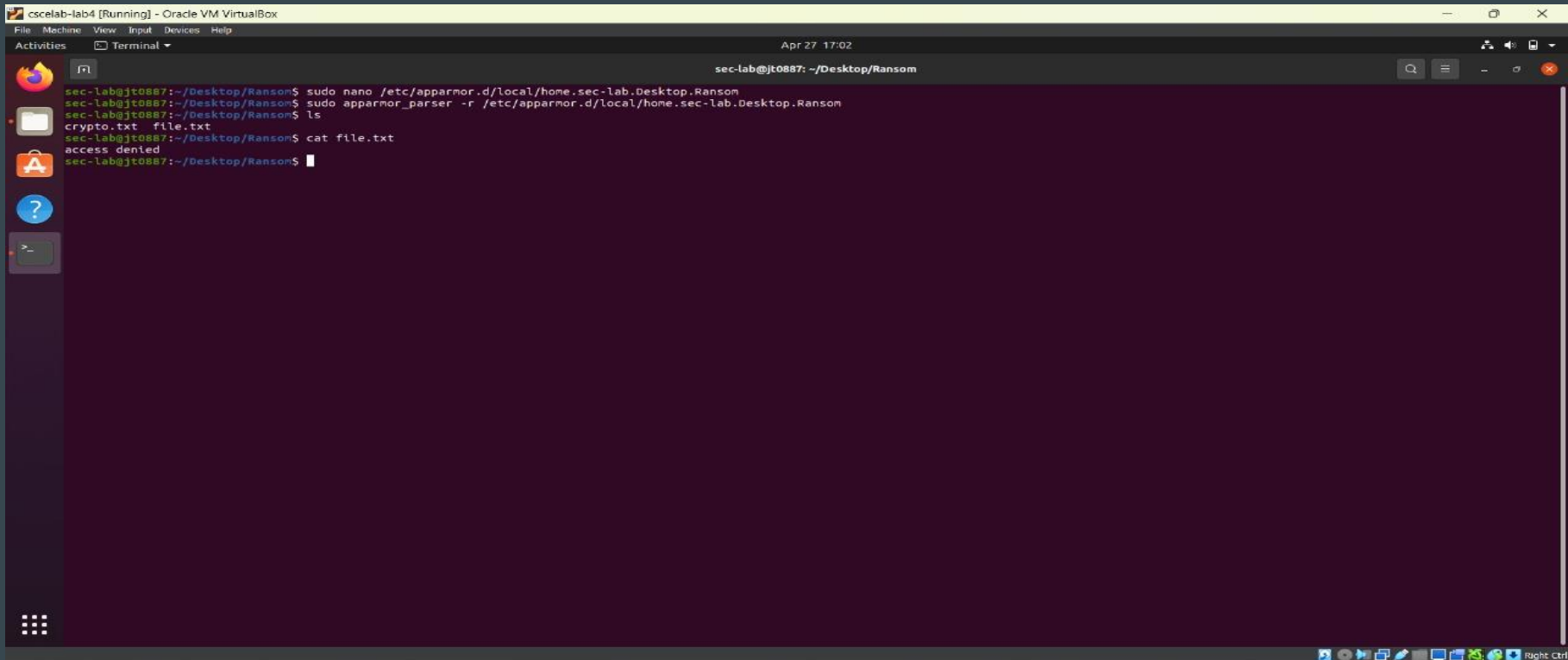


The screenshot shows a terminal window titled "cscelab-lab4 [Running] - Oracle VM VirtualBox". The terminal is running the nano text editor, editing the AppArmor policy file located at `/etc/apparmor.d/local/home.sec-lab.Desktop.Ransom`. The policy content is as follows:

```
GNU nano 4.8 /etc/apparmor.d/local/home.sec-lab.Desktop.Ransom
/home/sec-lab/Desktop/Ransom/** {
/** mrw,
}
```

The terminal window includes a menu bar with options: File, Machine, View, Input, Devices, Help. The status bar at the bottom displays various keyboard shortcuts for editing and navigation, such as "Get Help", "Write Out", "Where Is", "Cut Text", "Justify", "Cur Pos", "Go To Line", "Undo", "Redo", "Mark Text", "Copy Text", "To Bracket", "Where Was", "Previous", "Next", "Back", "Forward", "Prev Word", and "Next Word".

# Monitoring apparmor



The screenshot shows a terminal window titled "cscelab-lab4 [Running] - Oracle VM VirtualBox". The terminal output is as follows:

```
sec-lab@jt0887: ~/Desktop/Ransom$ sudo nano /etc/apparmor.d/local/home.sec-lab.Desktop.Ransom
sec-lab@jt0887:~/Desktop/Ransom$ sudo apparmor_parser -r /etc/apparmor.d/local/home.sec-lab.Desktop.Ransom
sec-lab@jt0887:~/Desktop/Ransom$ ls
crypto.txt  file.txt
sec-lab@jt0887:~/Desktop/Ransom$ cat file.txt
access denied
sec-lab@jt0887:~/Desktop/Ransom$
```

The terminal window includes a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". The status bar at the bottom shows the date and time as "Apr 27 17:02". The left sidebar contains icons for "Activities", "Terminal", and a dock with application icons (Firefox, Files, App Store, Help, and a terminal icon). The bottom of the window shows a taskbar with various system icons and a "Right Ctrl" label.

# Monitoring using File Integrity Check

```
fileintegriy.py x
1 import os
2 import hashlib
3 import json
4
5 def calculate_file_hash(file_path):
6     hasher = hashlib.sha256()
7     with open(file_path, 'rb') as file:
8         for chunk in iter(lambda: file.read(4096), b''):
9             hasher.update(chunk)
10    return hasher.hexdigest()
11
12 def create_or_load_hash_database(database_path):
13     if os.path.exists(database_path):
14         with open(database_path, 'r') as db_file:
15             return json.load(db_file)
16     else:
17         return {}
18
19 def update_hash_database(database_path, file_path, file_hash):
20     database = create_or_load_hash_database(database_path)
21     database[file_path] = file_hash
22     with open(database_path, 'w') as db_file:
23         json.dump(database, db_file, indent=2)
24
25 def check_integrity(directory_to_monitor, database_path):
26     # Implementation of check_integrity function
```

LightEdit mode. Access full IDE v

Autosave: off 1:1

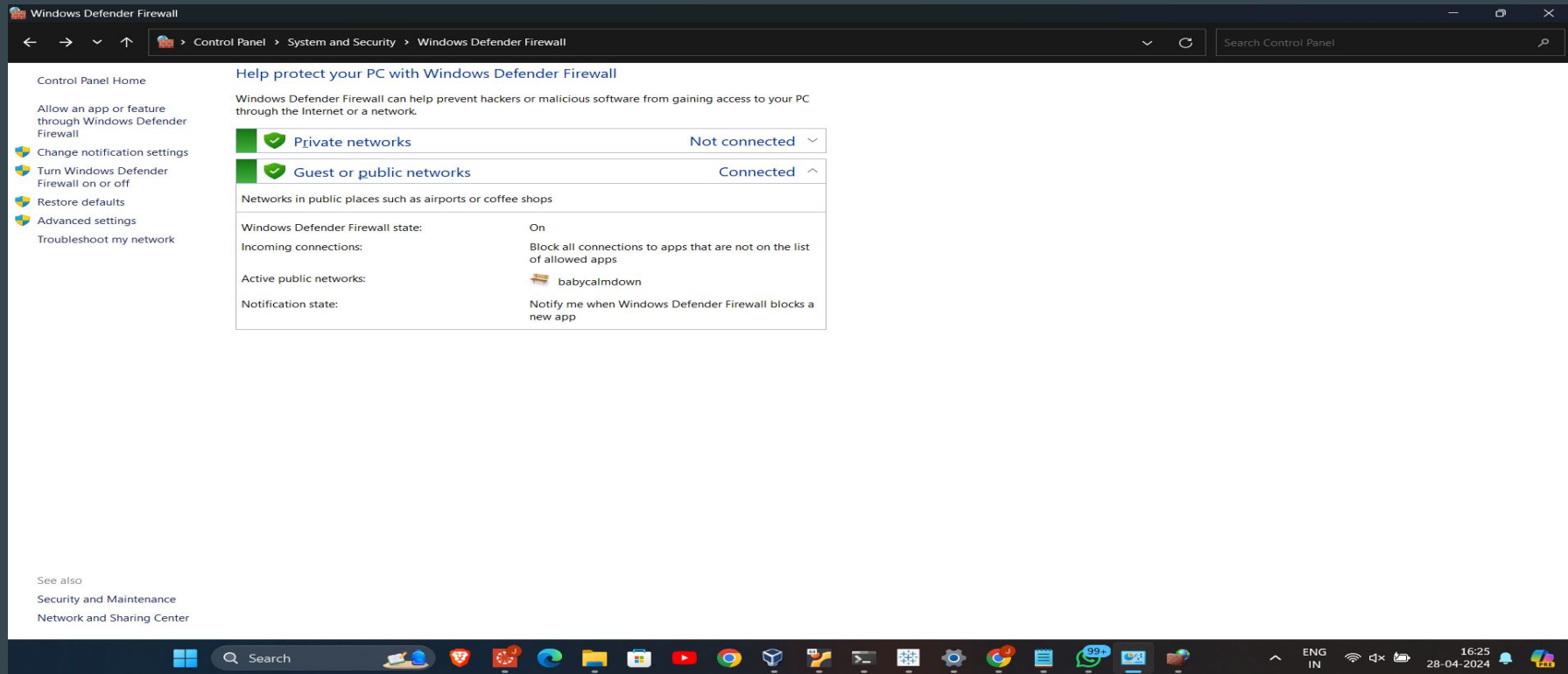
Items 1 item selected 1.59 KB

Windows taskbar: Search, File Explorer, Google Chrome, Microsoft Edge, PC icon, System tray (1:59 PM, 4/28/2024)

# Detection

- We have used the Windows Defender and Windows Firewall for detection of malware
- In windows we have the real time malware detection and its the best approach for the detection of malware

# Detection







Home



Virus & threat protection



Account protection



Firewall & network protection



App & browser control



Device security



Device performance & health



Family options



Protection history

## CrowdStrike Falcon Sensor

CrowdStrike Falcon Sensor is turned on.

### Current threats

✓ No actions needed.

### Protection settings

✓ No actions needed.

### Protection updates

✓ No actions needed.

[Open app](#)

### Microsoft Defender Antivirus options

You can keep using your current provider, and have Microsoft Defender Antivirus periodically check for threats.

Periodic scanning

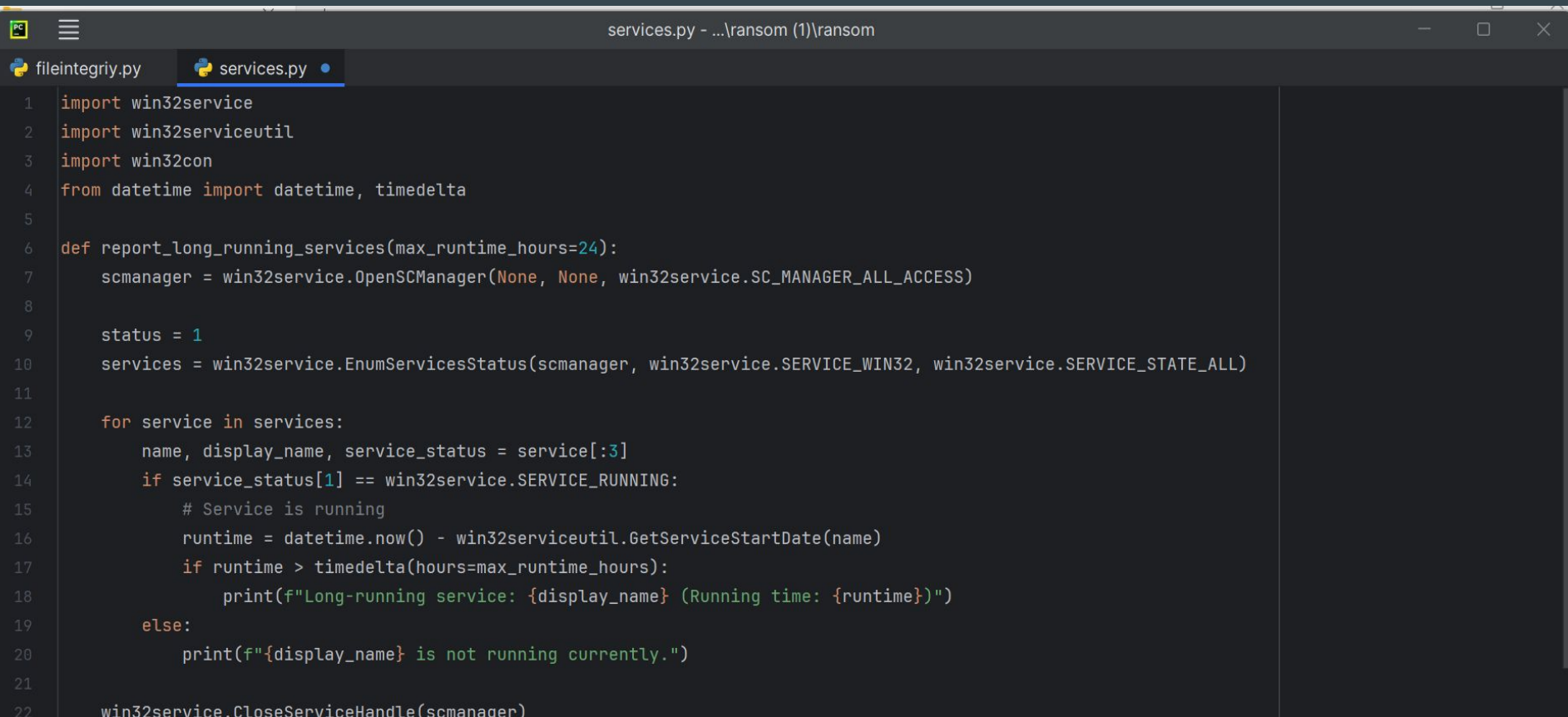


Off

# Code for services check

- We have a idea of python code which check the suspicious long running services check in the system.
- It gives the names of the services which is one of the way of detection.

# Image of services check

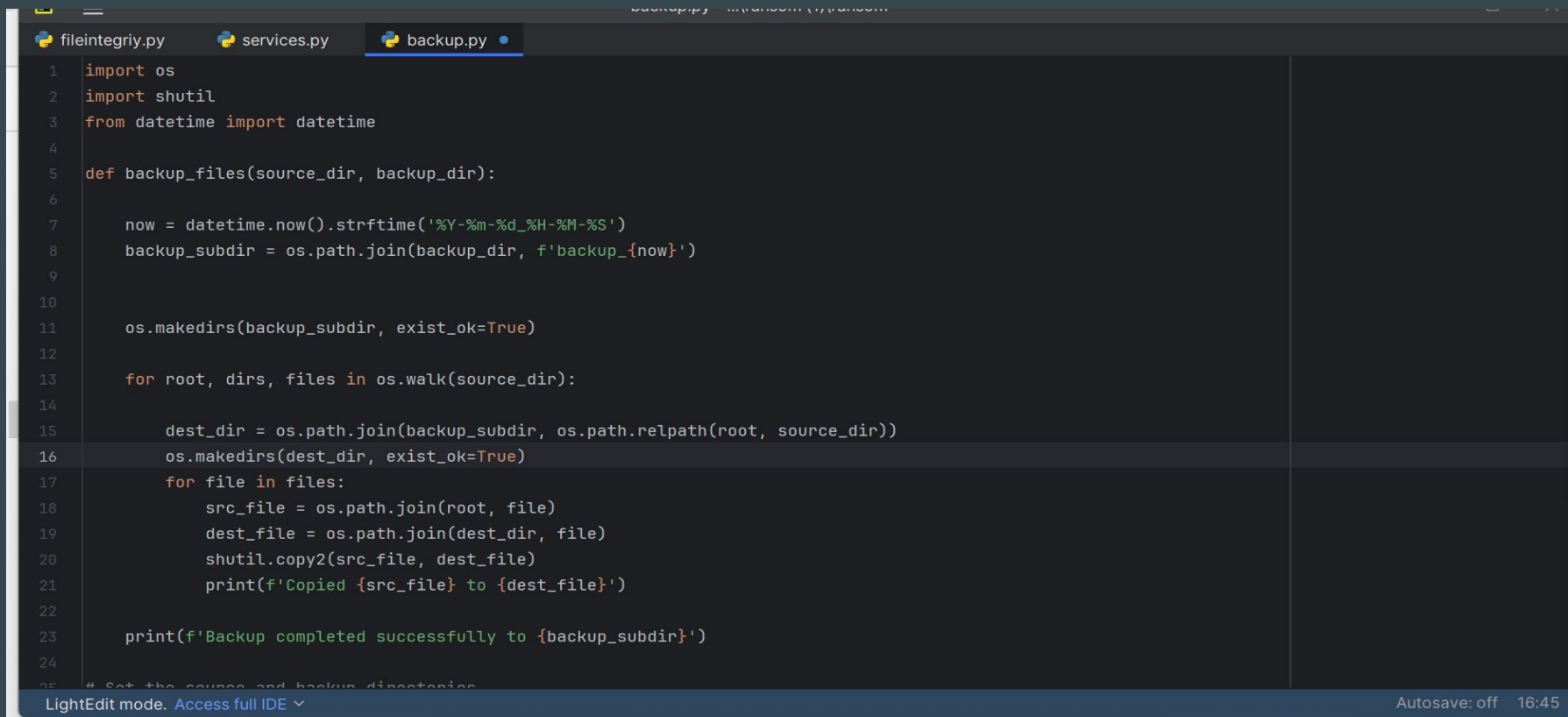


```
1 import win32service
2 import win32serviceutil
3 import win32con
4 from datetime import datetime, timedelta
5
6 def report_long_running_services(max_runtime_hours=24):
7     scmanager = win32service.OpenSCManager(None, None, win32service.SC_MANAGER_ALL_ACCESS)
8
9     status = 1
10    services = win32service.EnumServicesStatus(scmanager, win32service.SERVICE_WIN32, win32service.SERVICE_STATE_ALL)
11
12    for service in services:
13        name, display_name, service_status = service[:3]
14        if service_status[1] == win32service.SERVICE_RUNNING:
15            # Service is running
16            runtime = datetime.now() - win32serviceutil.GetServiceStartDate(name)
17            if runtime > timedelta(hours=max_runtime_hours):
18                print(f"Long-running service: {display_name} (Running time: {runtime})")
19        else:
20            print(f"{display_name} is not running currently.")
21
22    win32service.CloseServiceHandle(scmanager)
```

# mitigation

- Maintaining the backup is the best approach for mitigation.
- Maintaining the data at different availability zones and locations.
- We have a Python Script which automatically updates the data for every 4 hours and maintains the redundancy.
- Removing the services which are suspicious other way of mitigation.

# Image of backup



The image shows a code editor window with three tabs: 'fileintegrity.py', 'services.py', and 'backup.py'. The 'backup.py' tab is active, displaying a Python script for creating a backup. The script imports 'os', 'shutil', and 'datetime'. It defines a function 'backup\_files(source\_dir, backup\_dir)' which creates a subdirectory for the backup, walks through the source directory, and copies each file to the backup directory. The script also includes a main block to set source and backup directories.

```
1 import os
2 import shutil
3 from datetime import datetime
4
5 def backup_files(source_dir, backup_dir):
6
7     now = datetime.now().strftime('%Y-%m-%d_%H-%M-%S')
8     backup_subdir = os.path.join(backup_dir, f'backup_{now}')
9
10
11     os.makedirs(backup_subdir, exist_ok=True)
12
13     for root, dirs, files in os.walk(source_dir):
14
15         dest_dir = os.path.join(backup_subdir, os.path.relpath(root, source_dir))
16         os.makedirs(dest_dir, exist_ok=True)
17         for file in files:
18             src_file = os.path.join(root, file)
19             dest_file = os.path.join(dest_dir, file)
20             shutil.copy2(src_file, dest_file)
21             print(f'Copied {src_file} to {dest_file}')
22
23     print(f'Backup completed successfully to {backup_subdir}')
24
25 # Set the source and backup directories
```

LightEdit mode. Access full IDE ▾

Autosave: off 16:45

PS C:\hemanthvarma> ls

Directory: C:\hemanthvarma

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	4/28/2024 11:35 PM	1476	backup.py

PS C:\hemanthvarma> python3 backup.py

Copied C:\Users\dp1492\Desktop\data\3.py to C:\Users\dp1492\Desktop\crypto\backup\_2024-04-28\_23-37-32\.\3.py

Copied C:\Users\dp1492\Desktop\data\ciphertext.txt to C:\Users\dp1492\Desktop\crypto\backup\_2024-04-28\_23-37-32\.\ciphertext.txt

Backup completed successfully to C:\Users\dp1492\Desktop\crypto\backup\_2024-04-28\_23-37-32

PS C:\hemanthvarma> python3 backup.py

Copied C:\Users\dp1492\Desktop\data\3.py to C:\Users\dp1492\Desktop\crypto\backup\_2024-04-28\_23-37-36\.\3.py

Copied C:\Users\dp1492\Desktop\data\ciphertext.txt to C:\Users\dp1492\Desktop\crypto\backup\_2024-04-28\_23-37-36\.\ciphertext.txt

Backup completed successfully to C:\Users\dp1492\Desktop\crypto\backup\_2024-04-28\_23-37-36

PS C:\hemanthvarma>

code

×

+

←

→

↑

↻

🖥️

>

code

>

⊕ New ▾

✂

📄

📁

📄🔍

🔗

🗑

↕ Sort ▾

☰ View ▾

⋮

🏠 Home

🖼 Gallery

Name	Date modified	Type	Size
📁 hemanthvarma	4/28/2024 11:33 PM	File folder	

# Conclusion

In conclusion, our study on attacks using ransomware provides a thorough examination of the details of their deployment, the vulnerabilities they exploit, and the direct effects they cause. We demonstrated the ease with which these assaults may be carried out by creating a proof-of-concept ransomware utilizing the AES encryption standard and Python's Fernet library, notably through social engineering approaches like as phishing. Our research highlights the critical need for strong cybersecurity measures and user attention. The use of strong encryption practices and the simulation of ransomware attacks in a controlled setting highlight the vital role of security awareness and proactive defenses in reducing the disastrous impacts of such cyber threats. Furthermore, our findings provide insight into effective monitoring, detection, and mitigation options for protecting against ransomware.

In Final as a team we have implemented the Ransomware from Development, Infection, Monitoring, detection and till the mitigation of malware Ransomware.



# References

- 1) J. Hurtuk, M. Chovanec, M. Kičina and R. Billík, "Case Study of Ransomware Malware Hiding Using Obfuscation Methods," 2018 16th International Conference on Emerging eLearning Technologies and Applications (ICETA), Štrý Smokovec, Slovakia, 2018, pp. 215-220, doi: 10.1109/ICETA.2018.8572218. keywords: {Ransomware;Encryption;Computers;Conferences;Electronic learning}, <https://ieeexplore.ieee.org/document/8572218>
- 2) Ekta and U. Bansal, "A Review on Ransomware Attack," 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCC), Jalandhar, India, 2021, pp. 221-226, doi: 10.1109/ICSCCC51823.2021.9478148. keywords: {Ransomware;Cryptography;Surges;Cyberattack;Ransomware;Locky;Malware;Classification;DNS;Cyber Attack}, <https://ieeexplore.ieee.org/document/9478148>
- 3) R. Wilson and I. Iftimie, "Emerging ransomware threats: An anticipatory ethical analysis," 2021 IEEE International Symposium on Technology and Society (ISTAS), Waterloo, ON, Canada, 2021, pp. 1-1, doi: 10.1109/ISTAS52410.2021.9629211. keywords: {Ethics;Government;Medical services;Games;Market research;Software;Ransomware}, <https://ieeexplore.ieee.org/document/9629211>
- 4) P. K. B. Nataraj and P. Duraisamy, "An Investigation on Attacks in Application Layer Protocols and Ransomware Threats in Internet of Things," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 668-672, doi: 10.1109/ICACCS57279.2023.10112669. keywords: {Protocols;Taxonomy;Computer architecture;Agriculture;Internet of Things;Ransomware;Security;Internet of Things;Architecture layers;Application layer;Security threats;Ransomware Attacks;RFID Attacks;Application Layer Protocols}, <https://ieeexplore.ieee.org/document/10112669>
- 5) Lena Yuryna Connolly, David S Wall, Michael Lang, Bruce Oddson, An empirical study of ransomware attacks on organizations: an assessment of severity and salient factors affecting vulnerability, Journal of Cybersecurity, Volume 6, Issue 1, 2020, tyaa023, <https://doi.org/10.1093/cybsec/tyaa023>, <https://academic.oup.com/cybersecurity/article/6/1/tyaa023/6047253>
- 6) Kapoor, A., Gupta, A., Gupta, R., Tanwar, S., Sharma, G., & Davidson, I. E. (2021). Ransomware Detection, Avoidance, and Mitigation Scheme: A Review and Future Directions. \*Sustainability, 14\*(1), 8. <https://doi.org/10.3390/su14010008>, <https://www.mdpi.com/2071-1050/14/1/8>

Thank You