Phishing and Ransomware Simulation Report

Project: Phishing → Ransomware Simulation (MailHog + GoPhish)

Author: Void Seekers

Table of Contents

---

1. Project Overview

Objective: This project demonstrates a controlled phishing campaign that delivers a benign, mock ransomware payload. The educational goals are:

- Show how phishing can deliver malicious payloads.
- Demonstrate how an unsuspecting user can be tricked into executing a payload.
- Gather campaign metrics (open, click rates) using GoPhish.
- Teach detection, mitigation, and incident response best practices.

Scope: Local/lab environment using MailHog as a local SMTP capture server and GoPhish as the phishing framework. The `ticket.exe` payload is a safe, non-destructive simulation that shows ransom-note behavior without encrypting real user data.

---

3. Environment & System Requirements

Minimum recommended: - CPU: 2 cores - RAM: 4 GB (2 GB may suffice for very small tests) - Disk: 10 GB free - OS: Windows 10/11, Ubuntu 20.04+, or macOS (latest) - Network: Localhost / isolated lab network

Software: - GoPhish (latest release) - MailHog (latest release) - Python 3.8+ and PyInstaller (for building the payload) - Browser for admin UIs (Chrome, Firefox)

Ports used (default): - MailHog SMTP: **1025** - MailHog HTTP UI: **8025** - GoPhish Admin: **3333** (HTTPS by default) - GoPhish Phishing server: **80** (HTTP) or custom port in `config.json` - Local payload hosting (example): **8000** (Python `http.server`)

---

4. Installation & Configuration

> **Note:** All commands are examples. Adjust paths and permissions for your OS.

4.1 MailHog

Installation (Linux/macOS): - Option 1 (prebuilt binary): Download the appropriate binary from MailHog releases and place it in `/usr/local/bin`. - Option 2 (Homebrew macOS): `brew update && brew install mailhog`.

Run MailHog:

```
# start MailHog (default ports 1025 SMTP, 8025 HTTP)
mailhog
```

```
# or if using a binary
./MailHog
```

Verify: - Open `http://localhost:8025` to view MailHog UI. - MailHog will receive emails sent to SMTP port 1025.

Configuration tips: - If port conflicts exist, stop the conflicting service or map MailHog to different ports. - Ensure local firewall allows loopback connections for the chosen ports.

4.2 GoPhish

Installation: - Download the latest GoPhish release from the official repository. - Unzip and place the folder in a suitable location (e.g., `~/gophish`).

Run GoPhish:

```
cd ~/gophish
# On Linux/macOS
./gophish
# On Windows (PowerShell)
.\\gophish.exe
```

Default config: - `admin_server.listen_url` → `127.0.0.1:3333` - `phish_server.listen_url` → `0.0.0.0:80`

Accessing UI: - Open a browser to `https://127.0.0.1:3333` (Admin panel) and log in with the generated credentials. Change password on first login.

Important: If you cannot bind to port 80 for the phishing server, change `phish_server.listen_url` to `127.0.0.1:8080` and update templates/links accordingly.

4.3 Python & PyInstaller (payload build)

Install Python: - Ensure Python 3.8+ is installed and `python/python3` is on PATH.

Install PyInstaller:

```
pip install pyinstaller
```

Create a virtualenv (optional but recommended):

```
python3 -m venv venv
source venv/bin/activate  # Linux/macOS
venv\\Scripts\\activate # Windows
```

5. Phishing Campaign Setup (GoPhish)

5.1 Create Email Template

**Example subject and body (HTML):**

**Subject:** Congratulations! You've Won an iPhone 17 Pro

HTML body (example):

```html
<!doctype html>
<html>
  <body>
    <div style="font-family: Arial, sans-serif;">
      <img src="https://example.com/logo.png" alt="Fawryz Logo"
style="width:160px;" />
      <h2>Congratulations!</h2>
      <p>Dear {{.FirstName}},</p>
      <p>We are excited to inform you that you have been randomly selected to
receive a brand new <strong>iPhone 17 Pro</strong> in our customer
appreciation giveaway.</p>
      <p>To claim your prize, please download your raffle ticket and run it
on your Windows machine:</p>
      <p><a href="http://localhost:8000/ticket.exe">Download your
ticket</a></p>
      <p>Best regards,<br/>Fawryz Team</p>
    </div>
  </body>
</html>
```

Template notes: - Replace {{.FirstName}} with GoPhish template variables where appropriate. - Use realistic logos and formatting to increase believability for training, but do not impersonate any real organization without permission. - Leave the default GoPhish tracking image enabled if you want open/click tracking.
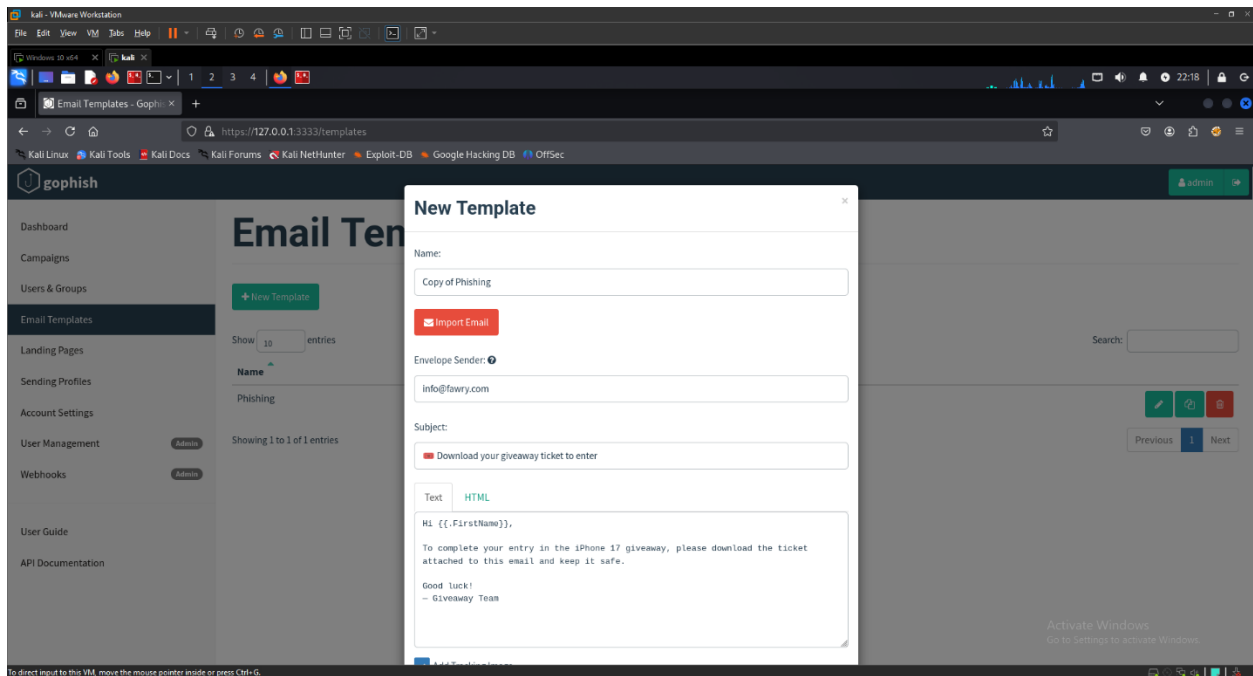
5.2 Sending Profile (MailHog)

In GoPhish Admin: - Navigate to `Sending Profiles` -> `New Profile`. - **SMTP Host:** `localhost:1025` - **From Address:** `support@fawryz.com` (mailhog will accept it) - Save and Send a Test email to verify MailHog receives it.
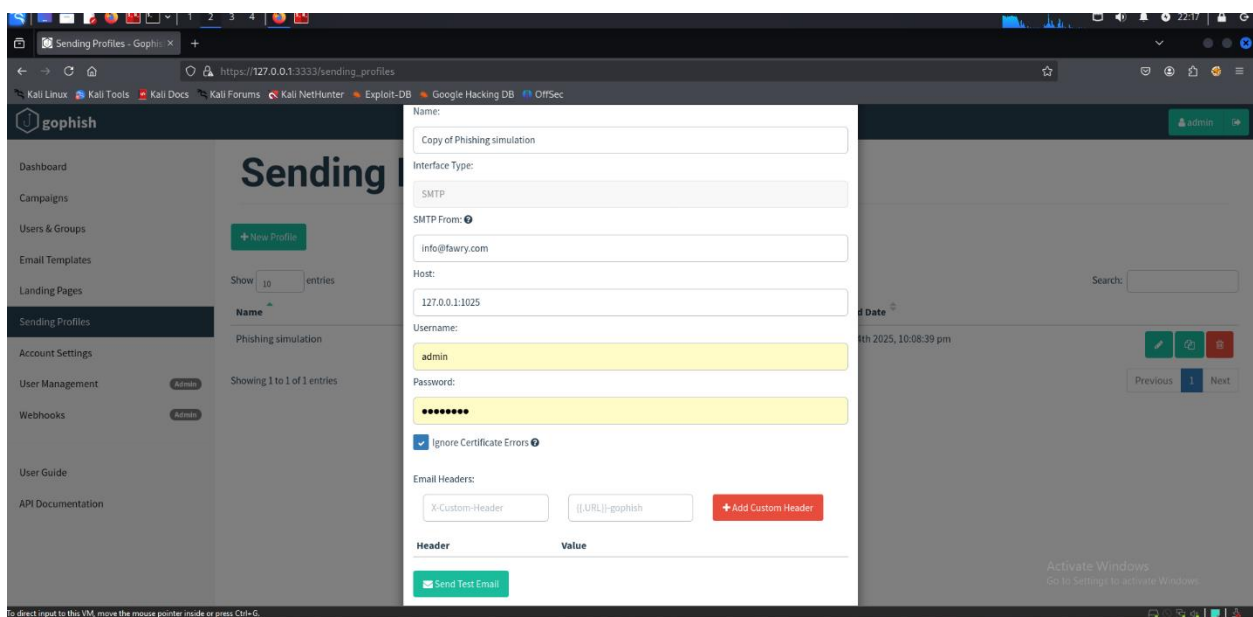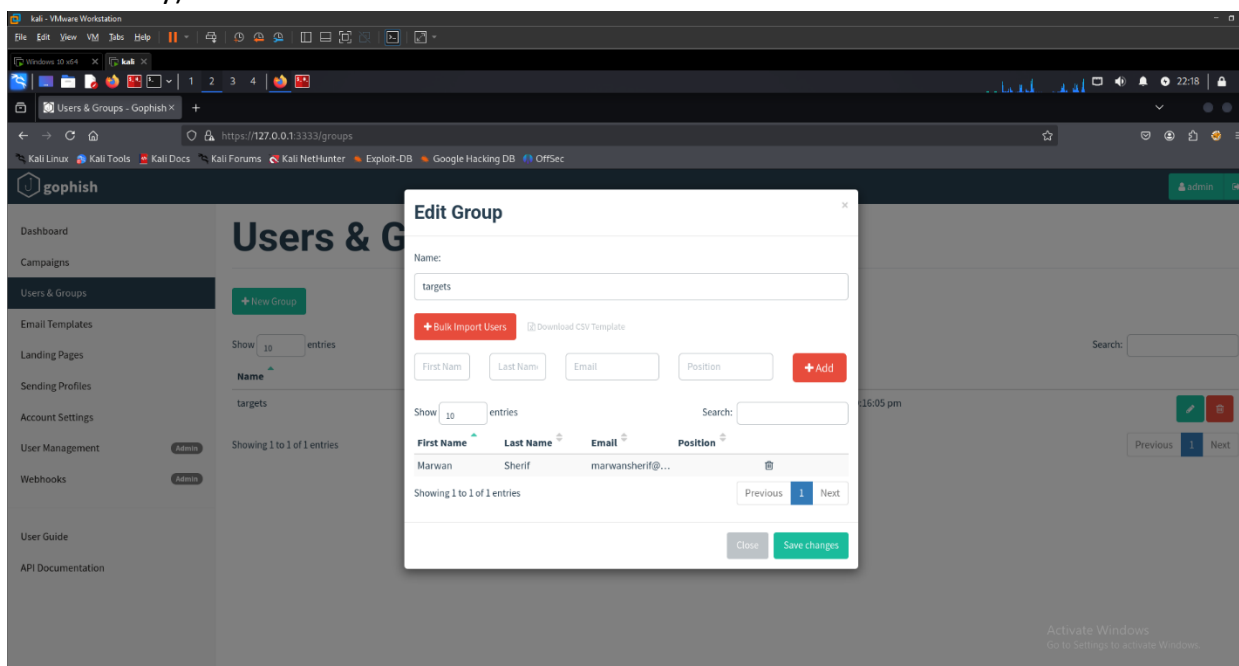
5.3 Target Groups (CSV import)

CSV example (headers are optional; GoPhish accepts name/email columns):

```
first_name,last_name,email
Marwan,Sherif,MarwanSherif@mail.com
Alice,Smith,alice.smith@example.test
```

- In GoPhish: `Users & Groups` -> `New Group` -> `Import CSV` (or add single users manually).

---

6. Payload Design (ticket.exe) — *Safe handling and reporting note*

Note: For background research we obtained a known WannaCry ransomware sample for analysis only. The sample was **not executed** on any production system. All handling of the sample was done in a controlled, isolated lab environment under supervisor approval. The live demonstration shown in this project used a benign mock payload (`ticket.exe`) that only simulates ransomware behavior (pop-up ransom message and safe file rename inside a contained demo folder) and does **not** encrypt or damage real user data.

- **What we did with the WannaCry sample:** referenced for academic analysis and comparison (static metadata and high-level behavior only). No execution, no distribution, and no replication of the malware code is included in this report.
- **What we used in the demo:** `ticket.exe` (a safe, reversible mock payload) to show the user-facing effects without risk.
- **Safety measures (summary):**

o   Analysis work limited to an isolated VM in an air-gapped/test lab.
o   Supervisor / lab authorization was obtained.
o   No real user files or production networks were used.
o   All artifacts and screenshots included in the report are sanitized.

7. Running the Simulation (Step-by-step)

1. Start MailHog: `mailhog` (ensure SMTP 1025 and UI 8025 are active).
2. Start GoPhish: `./gophish` (note admin URL and phishing server URL). Log into admin console.
3. Build `ticket.exe` and start a local HTTP server to host it: `python3 -m http.server 8000` in the directory containing `ticket.exe`.
4. In GoPhish, create the email template that links to `http://localhost:8000/ticket.exe`.
5. Create a sending profile pointing to `localhost:1025` (MailHog).
6. Prepare a target group with one or more test email addresses.
7. Launch the campaign from GoPhish and monitor the campaign dashboard.
8. Open MailHog at `http://localhost:8025` to view delivered emails. Click the link to download & run `ticket.exe` (or instruct a test user to do so in a VM).
9. Observe the payload behaviour (ransom note popup and demo file renaming) and check logs generated by the payload.
10. Review GoPhish campaign results for opens/clicks.

8. Observing Results & Metrics

GoPhish Dashboard Metrics: - **Sent:** Number of emails sent. - **Delivered:** MailHog reception (GoPhish shows send status). - **Opened:** Tracked via embedded tracking pixel. - **Clicked:** Tracked via link redirection through GoPhish.
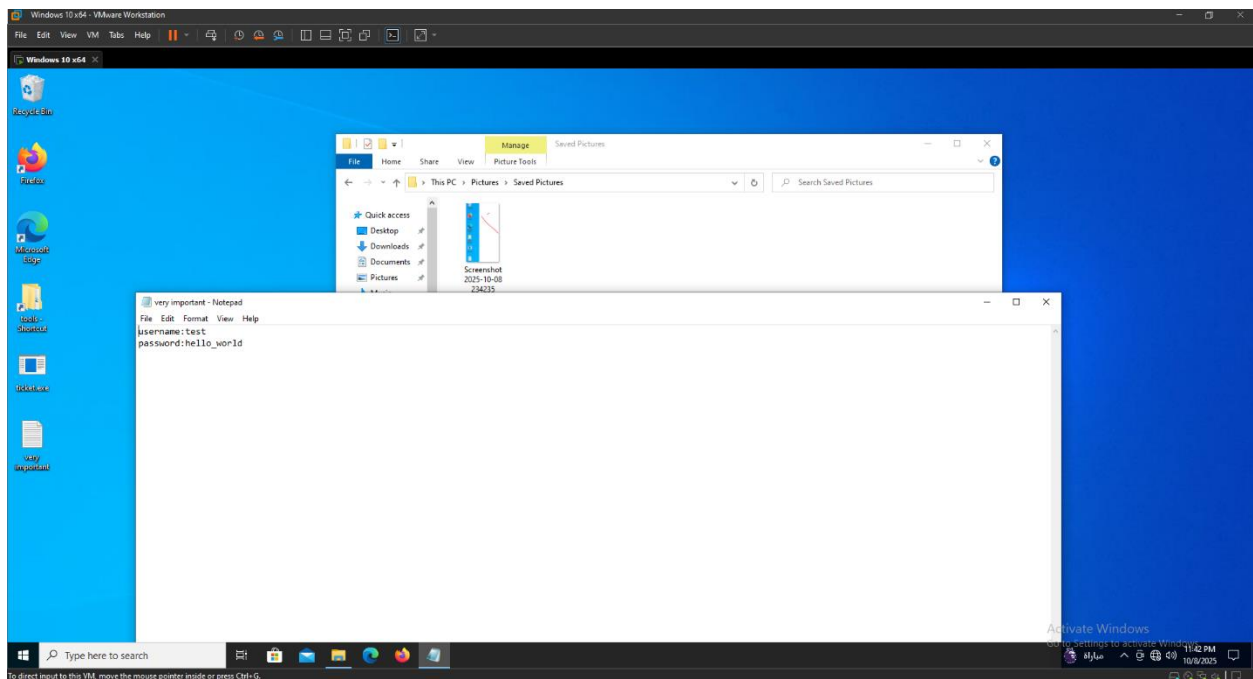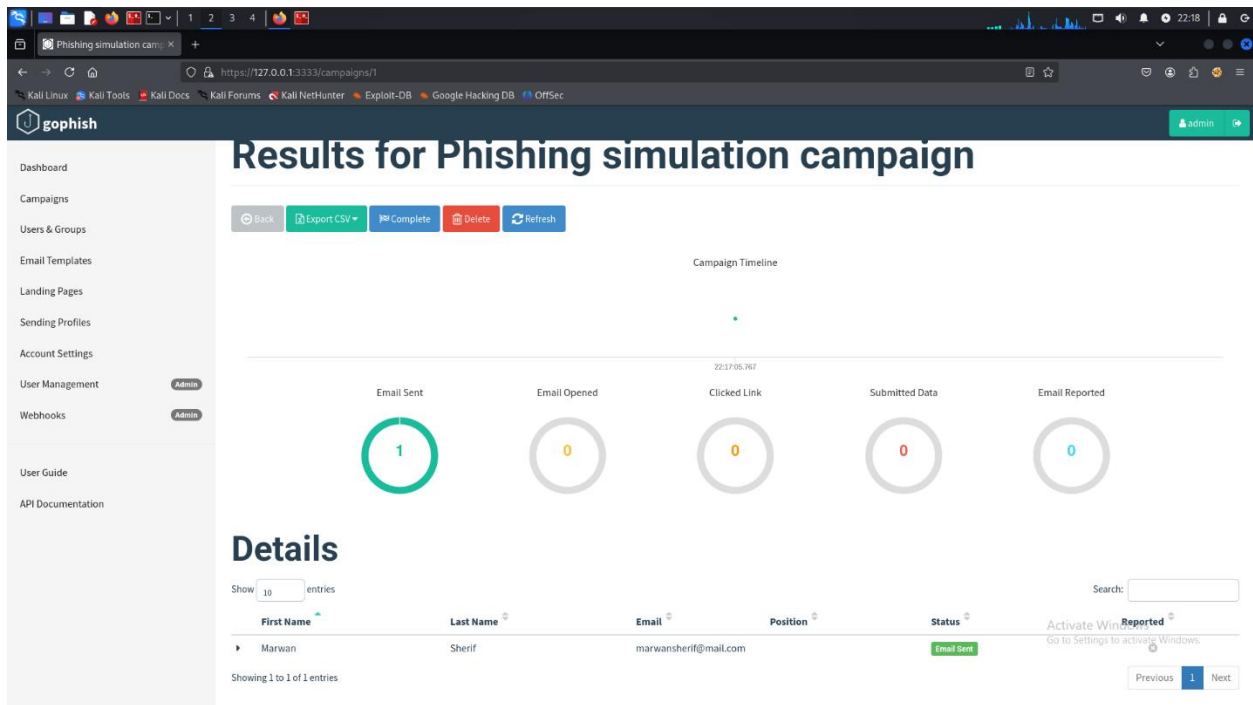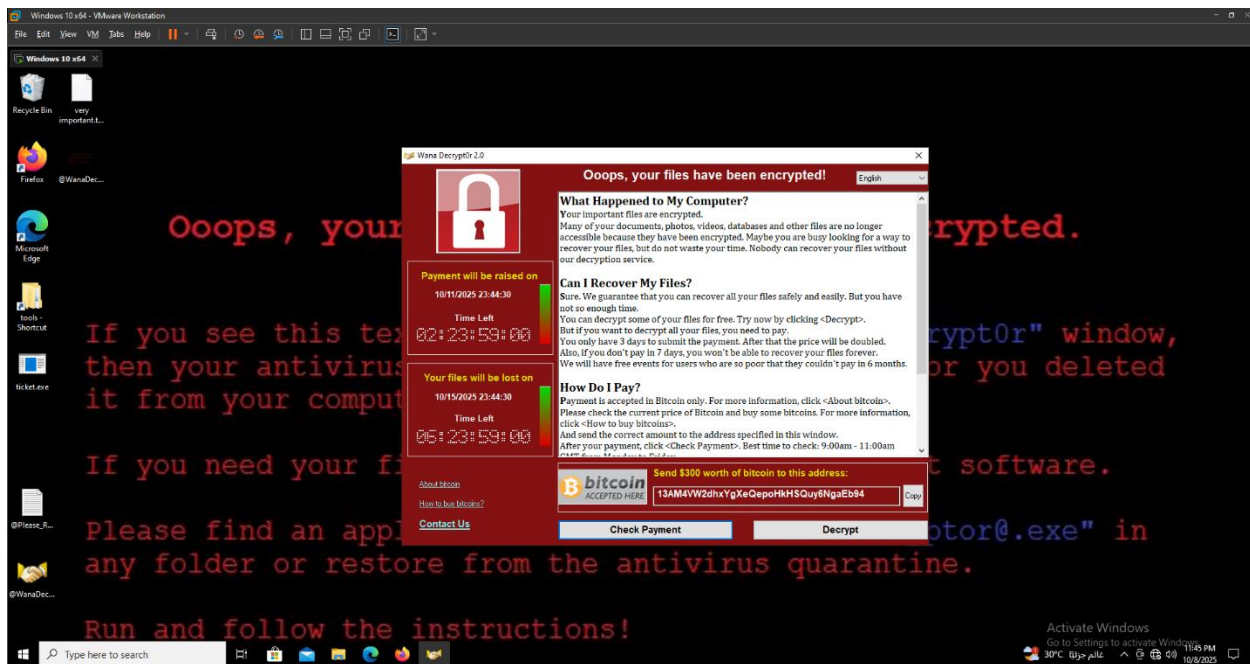
9. Disk & Memory Image Acquisition

Once images are captured, verify their integrity using the acquisition tool's built-in verification/confirmation features and confirm the images are readable. Store the images securely for analysis (use encrypted or access-controlled storage). As noted in digital forensics best practices, a complete bitstream image is the standard "forensic duplicate" for later analysis.

Name	Date modified	Type	Size
memdump.mem	10/8/2025 10:37 AM	MEM File	2,097,152 KB
test.E01	10/8/2025 10:58 AM	E01 File	1,535,909 KB
test.E01.txt	10/8/2025 11:20 AM	Text Document	2 KB
test.E02	10/8/2025 10:59 AM	E02 File	1,535,817 KB
test.E03	10/8/2025 11:00 AM	E03 File	1,535,860 KB
test.E04	10/8/2025 11:01 AM	E04 File	1,535,866 KB
test.E05	10/8/2025 11:02 AM	E05 File	1,535,830 KB
test.E06	10/8/2025 11:04 AM	E06 File	1,535,929 KB
test.E07	10/8/2025 11:05 AM	E07 File	1,535,825 KB
test.E08	10/8/2025 11:06 AM	E08 File	1,535,877 KB
test.E09	10/8/2025 11:07 AM	E09 File	1,535,814 KB
test.E10	10/8/2025 11:09 AM	E10 File	1,535,823 KB
test.E11	10/8/2025 11:11 AM	E11 File	1,254,891 KB

10. Forensic Analysis & Indicators of Compromise (IOCs)

Using the disk and memory images, analysis proceeded with **Volatility** (for RAM) and registry / file-viewing tools (for the disk image). Volatility was used to enumerate running processes, show process trees, and inspect loaded modules from memory. In the memory image we identified two notable items via `pstree` / `pslist` style output and process inspection:

- A process named `@wannacrydecryptor` present in memory and showing behavior consistent with a dropped/running payload (visible in the process list).
- A process named `tasksche.exe` shown running from a suspicious filesystem location (reported in the process tree and confirmed by memory-resident module information).

```
root@reverse: /home/sherfa/volatility3-2.11.0 270x51
4440   4376   explorer.exe     0xb50b167d4100  70   -   1   False   2025-10-09 06:39:30.000000 UTC  N/A   Disabled
4688   700    svchost.exe      0xb50b16918240  8    -   0   False   2025-10-09 06:39:35.000000 UTC  N/A   Disabled
4040   700    svchost.exe      0xb50b16c78080  11   -   1   False   2025-10-09 06:39:52.000000 UTC  N/A   Disabled
816    828    TextInputHost.   0xb50b1611c080  10   -   1   False   2025-10-09 06:40:18.000000 UTC  N/A   Disabled
4964   828    StartMenuExper   0xb50b16437080  8    -   1   False   2025-10-09 06:40:21.000000 UTC  N/A   Disabled
4872   828    RuntimeBroker.   0xb50b16e82300  3    -   1   False   2025-10-09 06:40:23.000000 UTC  N/A   Disabled
2140   828    SearchApp.exe    0xb50b182a70c0  29   -   1   False   2025-10-09 06:40:24.000000 UTC  N/A   Disabled
440    700    SearchIndexer.   0xb50b18360240  14   -   0   False   2025-10-09 06:40:24.000000 UTC  N/A   Disabled
476    828    RuntimeBroker.   0xb50b162c0080  7    -   1   False   2025-10-09 06:40:26.000000 UTC  N/A   Disabled
5440   828    RuntimeBroker.   0xb50b18693300  4    -   1   False   2025-10-09 06:40:29.000000 UTC  N/A   Disabled
5816   828    smartscreen.ex   0xb50b18774080  6    -   1   False   2025-10-09 06:40:41.000000 UTC  N/A   Disabled
5876   4440   SecurityHealth   0xb50b183f6080  2    -   1   False   2025-10-09 06:40:41.000000 UTC  N/A   Disabled
5908   700    SecurityHealth   0xb50b183f8080  8    -   0   False   2025-10-09 06:40:41.000000 UTC  N/A   Disabled
5968   4440   vmtoolsd.exe     0xb50b187020c0  9    -   1   False   2025-10-09 06:40:41.000000 UTC  N/A   Disabled
5992   4440   OneDrive.exe     0xb50b184d5080  24   -   1   False   2025-10-09 06:40:42.000000 UTC  N/A   Disabled
4484   5992   OneDrive.Sync.   0xb50b187202c0  21   -   1   False   2025-10-09 06:40:48.000000 UTC  N/A   Disabled
3956   828    ApplicationFra   0xb50b1868d0c0  3    -   1   False   2025-10-09 06:40:54.000000 UTC  N/A   Disabled
6532   700    svchost.exe      0xb50b18cc4240  3    -   0   False   2025-10-09 06:40:54.000000 UTC  N/A   Disabled
6940   1628   audiodg.exe      0xb50b186de080  6    -   0   False   2025-10-09 06:41:00.000000 UTC  N/A   Disabled
6544   700    SgrmBroker.exe   0xb50b15673300  7    -   0   False   2025-10-09 06:41:27.000000 UTC  N/A   Disabled
1752   700    svchost.exe      0xb50b1676d080  8    -   0   False   2025-10-09 06:41:34.000000 UTC  N/A   Disabled
6820   828    MoUsoCoreWorke   0xb50b1676e2c0  13   -   0   False   2025-10-09 06:41:34.000000 UTC  N/A   Disabled
5476   700    svchost.exe      0xb50b10f142c0  8    -   0   False   2025-10-09 06:41:38.000000 UTC  N/A   Disabled
6712   828    ScreenClipping   0xb50b15659080  0    -   1   False   2025-10-09 06:42:15.000000 UTC  2025-10-09 06:42:18.000000 UTC  Disabled
6736   700    svchost.exe      0xb50b180cc080  7    -   1   False   2025-10-09 06:42:15.000000 UTC  N/A   Disabled
7144   828    ShellExperienc   0xb50b15f47300  27   -   1   False   2025-10-09 06:42:18.000000 UTC  N/A   Disabled
6164   828    RuntimeBroker.   0xb50b18f580c0  8    -   1   False   2025-10-09 06:42:18.000000 UTC  N/A   Disabled
4800   700    svchost.exe      0xb50b184350c0  2    -   1   False   2025-10-09 06:42:27.000000 UTC  N/A   Disabled
4644   828    SystemSettings   0xb50b167bf300  1    -   1   False   2025-10-09 06:42:27.000000 UTC  N/A   Disabled
4916   828    ScreenSketch.e   0xb50b167ba300  0    -   1   False   2025-10-09 06:42:32.000000 UTC  2025-10-09 06:42:41.000000 UTC  Disabled
1952   700    ticket.exe.exe   0xb50b18427080  134  -   0   True    2025-10-09 06:43:28.000000 UTC  N/A   Disabled
6368   6936   tasksche.exe     0xb50b1409c080  9    -   0   True    2025-10-09 06:43:28.000000 UTC  N/A   Disabled
6316   828    SystemSettings   0xb50b18495080  21   -   1   False   2025-10-09 06:43:29.000000 UTC  N/A   Disabled
5936   828    UserOOBEBroker   0xb50b16f37080  4    -   1   False   2025-10-09 06:43:30.000000 UTC  N/A   Disabled
2328   3892   msedge.exe       0xb50b1851a2c0  0    -   1   False   2025-10-09 06:43:30.000000 UTC  2025-10-09 06:43:41.000000 UTC  Disabled
6396   6368   @WanaDecryptor   0xb50b16ff6080  6    -   0   True    2025-10-09 06:44:25.000000 UTC  N/A   Disabled
2088   6396   taskhsvc.exe     0xb50b18744300  2    -   0   True    2025-10-09 06:44:27.000000 UTC  N/A   Disabled
1064   2088   conhost.exe      0xb50b167c2300  4    -   0   False   2025-10-09 06:44:27.000000 UTC  N/A   Disabled
6464   4452   @WanaDecryptor   0xb50b1876c2c0  6    -   1   True    2025-10-09 06:44:30.000000 UTC  N/A   Disabled
4296   828    ScreenClipping   0xb50b192670c0  0    -   1   False   2025-10-09 06:45:47.000000 UTC  2025-10-09 06:45:52.000000 UTC  Disabled
3216   440    SearchProtocol   0xb50b19659080  4    -   1   False   2025-10-09 06:46:01.000000 UTC  N/A   Disabled
6628   6820   wuauclt.exe      0xb50b162b8080  5    -   0   False   2025-10-09 06:46:07.000000 UTC  N/A   Disabled
7080   700    TrustedInstall   0xb50b196552c0  5    -   0   False   2025-10-09 06:46:10.000000 UTC  N/A   Disabled
1592   828    TiWorker.exe     0xb50b196240c0  4    -   0   False   2025-10-09 06:46:10.000000 UTC  N/A   Disabled
7148   4440   FTK Imager.exe   0xb50b18714080  23   -   1   False   2025-10-09 06:48:38.000000 UTC  N/A   Disabled
2388   2148   MpCmdRun.exe     0xb50b19571080  9    -   0   False   2025-10-09 06:49:17.000000 UTC  N/A   Disabled
6888   412    OneDriveLaunch   0xb50b19644080  4    -   1   False   2025-10-09 06:49:18.000000 UTC  N/A   Disabled
5576   2388   conhost.exe      0xb50b18311080  7    -   0   False   2025-10-09 06:49:20.000000 UTC  N/A   Disabled
```

```
/home/sherfa/volatility3-2.11.0/dump.txt - Sublime Text (SUDO / UNREGISTERED)
File  Edit  Selection  Find  View  Goto  Tools  Project  Preferences  Help

dump.txt

1  Volatility 3 Framework 2.11.0
2
3  PID PPID   ImageFileName   Offset(V)   Threads Handles SessionId  Wow64   CreateTime   ExitTime   Audit   Cmd Path
4
5  4   0   System  0xb50b10e97040  161  -  N/A False  2025-10-09 06:38:38.000000 UTC  N/A -  -  -
6  * 336   4   smss.exe   0xb50b1156e040  2  -  N/A False  2025-10-09 06:38:38.000000 UTC  N/A  \Device\HarddiskVolume3\Windows\System32\smss.exe   -  -
7  * 100   4   Registry   0xb50b10ed1080  4  -  N/A False  2025-10-09 06:38:31.000000 UTC  N/A Registry   -  -
8  * 1300  4   MemCompression  0xb50b1575d040  34  -  N/A False  2025-10-09 06:38:45.000000 UTC  N/A MemCompression  -  -
9  556 548 csrss.exe   0xb50b14f97140  13  -  1  False  2025-10-09 06:38:44.000000 UTC  N/A \Device\HarddiskVolume3\Windows\System32\csrss.exe  -  -
10  652 548 winlogon.exe   0xb50b14fd5080  8  -  1  False 2025-10-09 06:38:44.000000 UTC  N/A \Device\HarddiskVolume3\Windows\System32\winlogon.exe  -  -
11  * 864   652 fontdrvhost.ex 0xb50b14d7b140  5  -  1  False  2025-10-09 06:38:44.000000 UTC  N/A \Device\HarddiskVolume3\Windows\System32\fontdrvhost.exe  -  -
12  * 4376  652 userinit.exe   0xb50b167d2340  0  -  1  False  2025-10-09 06:39:29.000000 UTC  2025-10-09 06:39:45.000000 UTC  \Device\HarddiskVolume3\Windows\System32\userinit.exe  -  -
13  ** 4440 4376   explorer.exe   0xb50b167d4100  70  -  1  False  2025-10-09 06:39:30.000000 UTC  N/A \Device\HarddiskVolume3\Windows\explorer.exe   C:\Windows\Explorer.EXE
14  *** 5968   4440   vmtoolsd.exe   0xb50b187020c0  9  -  1  False  2025-10-09 06:40:41.000000 UTC  N/A \Device\HarddiskVolume3\Program Files\VMware\VMware Tools\vmtoolsd.exe  "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr   C:\Program Files\VMware\VMware Tools\vmtoolsd.exe
15  *** 5992   4440   OneDrive.exe   0xb50b184d5080  24  -  1  False  2025-10-09 06:40:42.000000 UTC  N/A \Device\HarddiskVolume3\Users\sherfa\AppData\Local\Microsoft\OneDrive\OneDrive.exe  -  -
16  **** 4484  5992   OneDrive.Sync.  0xb50b187202c0  21  -  1  False  2025-10-09 06:40:48.000000 UTC  N/A
"C:\Users\sherfa\AppData\Local\Microsoft\OneDrive\25.174.0907.0003\OneDrive.Sync.Service.exe  /silentConfig C:\Users\sherfa\AppData\Local\Microsoft\OneDrive\25.174.0907.0003\OneDrive.Sync.Service.exe
17  *** 7148   4440   FTK Imager.exe  0xb50b18714080  23  -  1  False  2025-10-09 06:48:38.000000 UTC  N/A \Device\HarddiskVolume3\tools\FTK Imager\FTK Imager.exe  "C:\tools\FTK Imager\FTK Imager.exe"  C:\tools\FTK Imager\FTK Imager.exe
18  *** 5876  4440   SecurityHealth  0xb50b183f6080  2  -  1  False  2025-10-09 06:40:41.000000 UTC  N/A \Device\HarddiskVolume3\Windows\System32\SecurityHealthSystray.exe  -  -
19  * 372   652 dwm.exe 0xb50b14de3080  34  -  1  False  2025-10-09 06:38:45.000000 UTC  N/A \Device\HarddiskVolume3\Windows\System32\dwm.exe  -  -
20  6368  6936   tasksche.exe   0xb50b1409c080  9  -  0  True   2025-10-09 06:43:28.000000 UTC  N/A \Device\HarddiskVolume3\ProgramData\vfvlkhgdq245\tasksche.exe   C:\ProgramData\vfvlkhgdq245\tasksche.exe   C:\ProgramData\vfvlkhgdq245\tasksche.exe
21  * 6396 6368   @WanaDecryptor  0xb50b16ff6080  6  -  0  True   2025-10-09 06:44:25.000000 UTC  N/A \Device\HarddiskVolume3\ProgramData\vfvlkhgdq245\@WanaDecryptor@.exe  -  -
22  ** 2088 6396   taskhsvc.exe   0xb50b18744300  2  -  0  True   2025-10-09 06:44:27.000000 UTC  N/A \Device\HarddiskVolume3\ProgramData\vfvlkhgdq245\TaskData\Tor\taskhsvc.exe  -  -
23  *** 1064  2088   conhost.exe   0xb50b167c2300  4  -  0  False  2025-10-09 06:44:27.000000 UTC  N/A \Device\HarddiskVolume3\Windows\System32\conhost.exe  -  -
24  6464  4452   @WanaDecryptor  0xb50b1876c2c0  6  -  1  True   2025-10-09 06:44:30.000000 UTC  N/A \Device\HarddiskVolume3\ProgramData\vfvlkhgdq245\@WanaDecryptor@.exe   -  -
25
```

On the disk image, registry analysis revealed a persistence entry created to maintain execution across reboots. The specific Run key discovered was:

HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\WanaCrypt0

This Run key referenced the persistence executable (the same path/name observed in the memory analysis). In this controlled lab, the registry Run key plus the in-memory evidence of **@wannacrydecryptor** and **tasksche.exe** form the primary indicators of compromise (IOCs) for the simulated infection.

HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Run

HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\WanaCrypt0 Run

---
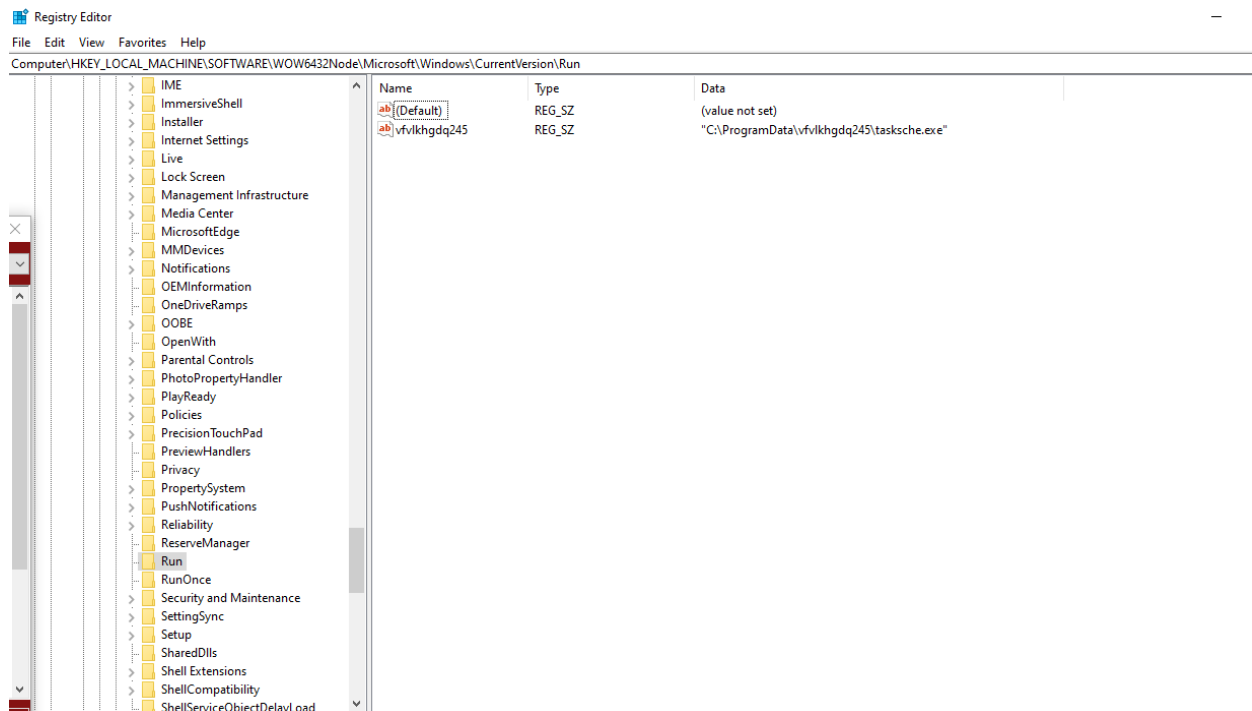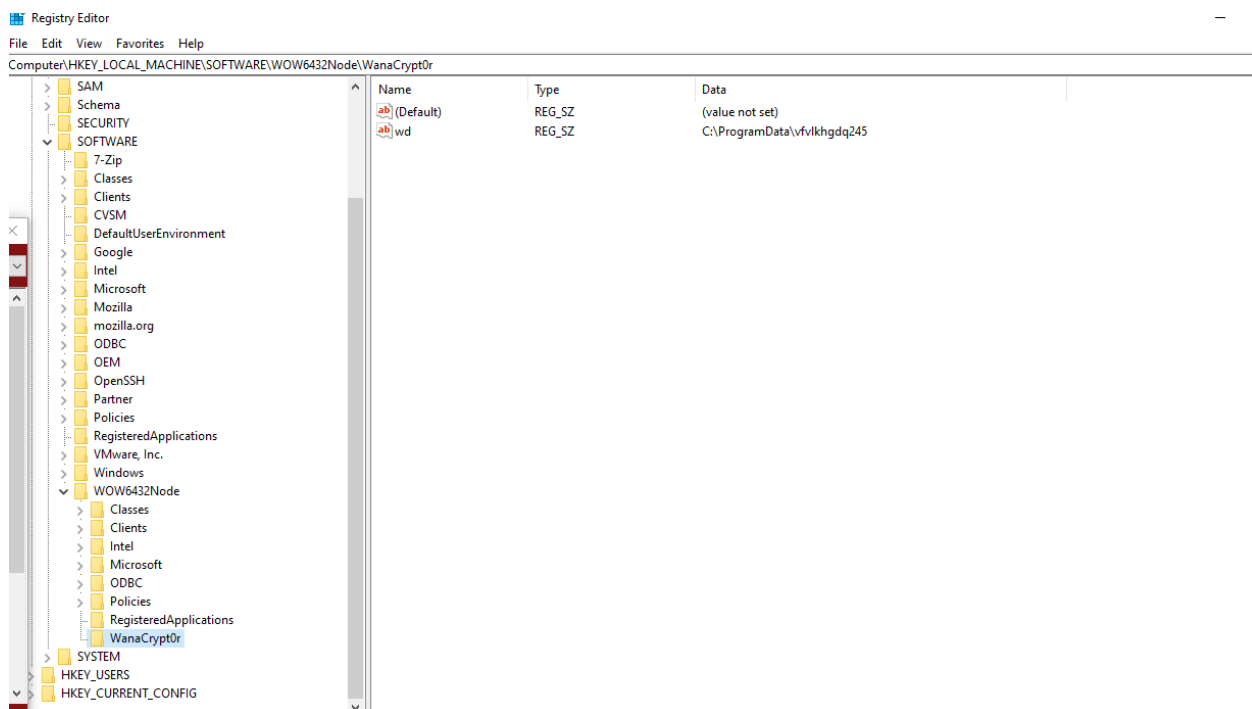
11. MITRE ATT&CK Mapping

Stage	Technique ID	Technique Name	Description
Initial Access	T1566.001	Phishing: Spearphishing Attachment	The phishing email (sent via GoPhish) contained a download link to ticket.exe, representing a malicious attachment or link lure.
Execution	T1204.002	User Execution: Malicious File	The victim executed ticket.exe believing it was a raffle ticket; this initiated the simulated ransomware payload.
Persistence	T1547.001	Registry Run Keys / Startup Folder	Registry key HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\WanaCrypt0 ensured the payload (tasksche.exe) ran on startup.
Defense Evasion	T1036.005	Masquerading: Match Legitimate Name or Location	The payload used legitimate-sounding filenames like tasksche.exe to avoid suspicion.
Impact	T1486	Data Encrypted for Impact	While the mock payload did not encrypt real files, it simulated the encryption process and displayed a ransom message, demonstrating this technique safely.
Command & Control	T1071.001	Application Layer Protocol: Web Protocols	In a real-world WannaCry-style attack, communication to a command server would occur via HTTP/S; this behavior was conceptually discussed but not executed.

---

12. Cyber Kill Chain

Kill Chain Phase	What happened in the simulation	Artifacts / Evidence	Was it detected? If so how & when
Reconnaissance	Attacker (simulator) prepared the lure and target list (created email template and CSV of targets).	GoPhish templates, targets.csv, project notes.	Not detected – preparation phase.
Weaponization	Created the payload concept (ticket.exe mock) and phishing	ticket.py source, PyInstaller build artifacts, HTML	Not detected – payload preparation.

	HTML body	template.	
Delivery	Email sent via GoPhish MailHog; phishing email delivered to target inbox	MailHog captured mails, GoPhish "Sent" entries.	Not detected – email successfully delivered.
Exploitation	The victim clicked the link and executed ticket.exe on the test VM. This is where code ran.	Download logs, ticket.exe execution on VM, process entry in memory.	Detected here – Late Detection: Infection became visible only after execution (ransomware activity observed).
Installation	Registry Run key added to maintain persistence (HKLM\...\WanaCrypt0 referencing the payload).	Registry hive from disk image, Run key entry, file path referenced by key.	Detected later during forensic analysis (post-infection).
Command & Control (C2)	Simulated / Not executed. In a real WannaCry variant C2 or worming may exist; no external C2 activity was performed in the lab.	N/A (no network callbacks executed).	Not applicable.
Actions on Objectives	impact renamed demo files and displayed ransom-note UI (simulated data encryption).	demo_files with .encrypted names, ticket_log.txt, ransom-note popup screenshots.	Observed as visible system impact.

12.1 Detection Summary

- Detection occurred late, during the Exploitation phase, after the payload executed.

- Volatility analysis revealed malicious processes (@WannaCryDecryptor, tasksche.exe) running from suspicious paths.

- FTK Imager and registry examination showed a persistence entry at HKLM\SOFTWARE\WOW6432Node\WanaCrypt0.

- These findings confirm that the simulated ransomware successfully executed before detection, which reflects real-world challenges in early ransomware identification.

13. Incident Response and Post-Incident Activities

After identifying the infection indicators in memory and registry analysis, an incident response workflow was initiated to contain, analyze, and recover the affected system within the controlled lab environment.

13.1 Containment and Recovery Actions

Following the forensic findings, we took several steps to safely manage and restore the affected virtual machine:

- **Root Cause Analysis:**
  The infection originated from the execution of the downloaded `ticket.exe`, which simulated the WannaCry behavior after a successful phishing email interaction. The root cause was confirmed to be user interaction with the phishing attachment, demonstrating the effectiveness of social engineering tactics.
- **Isolation of Affected Systems:**
  The infected VM was immediately isolated from the lab network to prevent any possible propagation or external communication. No data exfiltration or cross-system infection occurred, as confirmed through network monitoring and memory analysis.
- **System Recovery:**
  After isolation, we restored the affected VM using a **clean backup image** created before the simulation. This ensured the environment was returned to a stable state without residual infection. FTK Imager was used to verify the backup image before restoration.
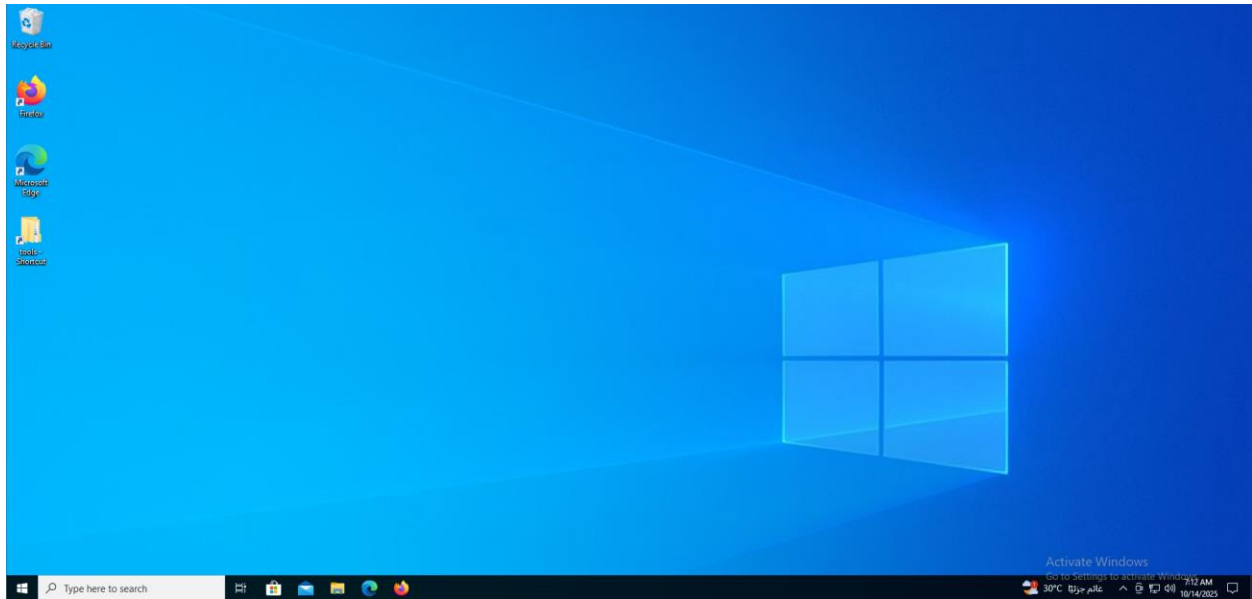
14. Recommendations & Mitigations

Based on the analysis of the phishing and ransomware simulation, several preventive and corrective security controls are recommended. These measures address the weaknesses that allowed the simulated infection to occur and strengthen defenses against similar real-world attacks.

14.1 User Awareness and Training

- Conduct **regular phishing-awareness sessions** to help users identify suspicious emails and attachments.
- Include periodic **phishing simulations** using safe frameworks like GoPhish to test employee readiness.
- Reinforce a **"Think Before You Click"** culture — encourage users to report unexpected links or downloads.
- Establish a **clear incident-reporting channel** (e.g., report@company.local) to quickly escalate suspicious messages.

14.2 Email Security Controls

- Implement **SPF, DKIM, and DMARC** to prevent domain spoofing and reduce phishing success rates.

- Deploy **email filtering and sandboxing** to analyze attachments and links before delivery.
- Block or quarantine **executable attachments (.exe, .bat, .js)** at the mail gateway.
- Use **URL rewriting and scanning** to inspect links embedded in emails.

14.3 Endpoint and System Hardening

- Enable **Application Whitelisting** to restrict execution to approved software only.
- Maintain **regular patch management** across all systems to reduce exploitability of vulnerabilities.
- Deploy **Endpoint Detection & Response (EDR)** or antivirus tools with behavioral analysis to detect and stop ransomware-like activity.
- Enforce **least-privilege principles** for user accounts to limit damage if compromise occurs.

14.4 Backup and Recovery Practices

- Maintain **regular, automated backups** stored offline or in immutable cloud storage.
- Periodically **test backup restorations** to confirm data integrity and recovery speed.
- Keep **multiple backup generations** to ensure recovery even if recent data becomes compromised.
- Document **backup schedules and recovery points** within organizational policies.

14.5 Network and Monitoring Enhancements

- Segment critical systems and servers from user networks using VLANs or subnets.
- Implement **Network Intrusion Detection/Prevention Systems (NIDS/NIPS)** to detect suspicious traffic.
- Log all authentication and network events centrally in a **SIEM** for correlation and alerting.
- Regularly review logs for anomalies and failed login attempts.

14.6 Policy and Governance

- Establish a **formal Incident Response Plan (IRP)** defining roles, escalation paths, and communication protocols.
- Maintain a **forensic readiness plan** to ensure evidence (memory, disk, logs) can be acquired safely in future incidents.

- Update **security policies and standard operating procedures (SOPs)** based on lessons learned from this project.
- Conduct **annual audits** of both technical and procedural controls to ensure compliance and effectiveness.

15. Project Summary & Conclusion

This project successfully simulated a full phishing-to-ransomware attack chain in a controlled laboratory environment to demonstrate the real-world lifecycle of a cyber incident — from initial compromise to detection, analysis, and recovery.

The simulation began with a **phishing campaign** using **GoPhish** and **MailHog**, where a crafted email imitating a "Fawryz giveaway" lured the victim into downloading a malicious executable named `ticket.exe`. Once executed, the payload simulated ransomware behavior similar to **WannaCry**, encrypting files and creating persistence via registry modifications under `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\WanaCrypt0`.

Following infection, **FTK Imager** was used to capture disk and memory images, which were then analyzed using **Volatility 3** to identify malicious processes such as `@WannaCryDecryptor` and `tasksche.exe`. These artifacts, along with the persistence registry key, served as clear **Indicators of Compromise (IOCs)**.

A **Chain of Custody** was maintained throughout evidence acquisition to preserve forensic integrity.
In the **recovery phase**, the affected virtual machine was isolated, and data was successfully restored using a **clean backup image**, completing the containment and recovery steps.

Finally, the **MITRE ATT&CK framework** was used to map adversary behavior and identify key tactics such as initial access, execution, persistence, and impact.
Based on these findings, a set of **recommendations and mitigations** was developed to strengthen security posture — including user awareness, email filtering, endpoint protection, backup management, and formal incident response planning.

In conclusion, the project demonstrated the **complete cybersecurity incident lifecycle**:

- **Attack Simulation (Phishing + Malware Execution)**
- **Forensic Investigation (Evidence Collection & Analysis)**
- **Incident Response (Containment, Eradication, Recovery)**
- **Post-Incident Review (Lessons Learned & Recommendations)**

This end-to-end approach reflects real-world practices used by cybersecurity professionals to detect, analyze, and respond to modern threats. It highlights the importance of combining **technical defenses**, **user training**, and **procedural discipline** to maintain resilience against evolving cyberattacks.