MORNING CATCH ENVIRONMENT

ADVANCED SOCIAL ENGINEERING CAMPAIGN

DEPI GRADUATION PROJECT

TARGET MACHINE

MORNING CATCH MACHINE
(192.168.10.129)

SCOPE OF WORK

IN-SCOPE
OUT-OF-SCOPE

01

WHAT IS SOCIAL ENGINEERING

Social engineering is a psychological manipulation technique used by attackers to trick people into giving away sensitive information, granting access, or performing actions that compromise security.

Instead of hacking systems, social engineering hacks people

HOW IT WORKS

- **Trust** (pretending to be IT, HR, manager...)
- **Urgency** (act fast or something bad happens)
- **Fear** (your account will be closed!)
- **Curiosity** (click here to view this document)
- **Authority** (CEO requests this immediately)

EXECUTIVE SUMMARY

- Red team simulated a targeted cyberattack on the Morning Catch environment.
- Engagement revealed major weaknesses: outdated systems, weak credentials, and lack of network segmentation.
- Reconnaissance identified exposed services and valid usernames via SMTP enumeration.
- A disguised reverse-shell payload was crafted and delivered through a phishing email to the CEO.
- Execution of the payload granted remote access and full control of the target machine.
- Persistence was established through scheduled tasks and later maintained via RDP access and privilege escalation to root.
- The operation demonstrated a full attack chain: reconnaissance → phishing → payload execution → persistence.
- Findings highlight the need for improved system hardening, stronger user awareness, and enhanced detection capabilities.

SCOPE

The engagement scope was strictly defined to simulate a realistic adversary while maintaining full safety and compliance, focusing solely on approved systems and services within the controlled Morning Catch environment.

In-Scope Activities

- Passive and active reconnaissance
- Service enumeration and vulnerability identification
- Social engineering (phishing) with pre-approved content •
- Payload generation and remote code execution
- Privilege escalation and persistence mechanisms

Out-of-Scope Assets and Activities

- Non-consensual attacks against third-party systems or external services
- • Denial-of-Service (DoS) testing or actions that may impact system availability
- • Data exfiltration involving real sensitive data (simulated only)

METHODOLOGY: CYBER KILL CHAIN

1. Reconnaissance (active and passive) Objective:
Gather intelligence to inform attack planning.

2. Weaponization Objective:
Develop a tailored payload for remote access.

3. Delivery Objective:
Deliver the payload through social engineering.

4. Exploitation Objective:
Execute the payload to gain a foothold.

5. Installation Objective:
Establish persistence on the target.

6. Obfuscation / Anti-forensics Objective:
covering the tracks, prevent forensic analysis and detection of their activities.

RECONNAISSANCE

Passive Reconnaissance:

- Passive reconnaissance was performed using OSINT without direct interaction with the target.
- Public information was collected from sources such as LinkedIn, the company website, and domain records.
- The goal was to identify internal technologies, staff roles, and likely email formats.

Active Reconnaissance:

- Active reconnaissance was performed after completing passive information gathering.
- The Morning Catch machine was directly scanned to identify exposed services.
- A full TCP scan using Nmap was conducted to enumerate open ports and service versions.
- The scan revealed accessible services, including SMTP (port 25) and HTTP (port 80), among others.

WEAPONIZATION

- A custom payload was created to establish remote access to the Morning Catch machine.
- The payload was generated using **msfvenom** tool to produce a Windows-compatible reverse shell executable.
- The file was disguised as a legitimate system update to increase the chance of execution.
- When run on the target, the payload initiated a remote connection back to the attacker, enabling system access.

The payload was disguised as a legitimate system update to avoid raising suspicion. It functioned as a backdoor that enabled ongoing, unauthorized access to the target system. Once executed, it created a hidden communication channel that allowed the attacker to remotely control the system without the user's knowledge or consent.

DELIVERY

Email-Based Social Engineering via SMTP (python3 -m http.server 8080)

The payload was delivered through a spoofed email sent via the open SMTP service, appearing to come from the internal admin and directed to the CEO.

Using a Telnet connection to the SMTP service on port 25, the email was manually crafted and sent with the intended spoofed details.

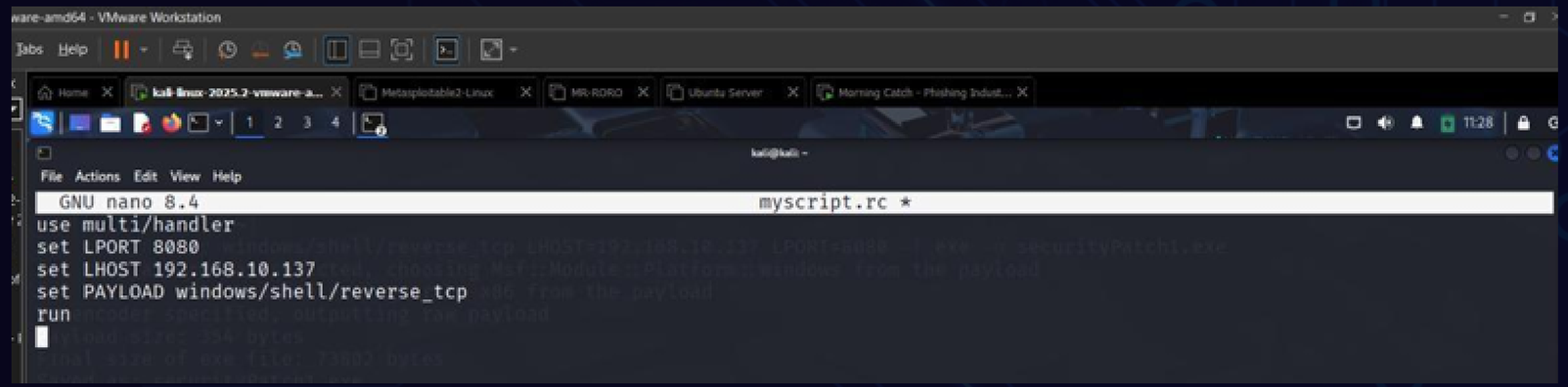
```
┌──(kali㉿kali)-[~]
└─$ telnet 192.168.10.129 25
Trying 192.168.10.129...
Connected to 192.168.10.129.
Escape character is '^]'.
220 morningcatch.ph ESMTP Sendmail 8.14.3/8.14.3/Debian-9.1ubuntu1; Wed, 12 Nov 2025 11:06:33 -0500; (No UCE/UBE)
rom: [192.168.10.137](FAIL)-[192.168.10.137]
HELO morningcatch.ph
250 morningcatch.ph Hello [192.168.10.137], pleased to meet you

500 5.5.1 Command unrecognized: ""
MAIL FROM:<bjenius@morningcatch.ph>
250 2.1.0 <bjenius@morningcatch.ph>... Sender ok
RCPT TO:<rbourne@morningcatch.ph>
250 2.1.5 <rbourne@morningcatch.ph>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
From:bjenius@morningcatch.ph
To:rbourne@morningcatch.ph

Subject: Urgent Security Patch

Hello Mr/Richard
I'd liketo inform you about a new security patch that will protect your system.
Please make sure to download it from this official link: http://192.168.10.137:8080/securitypatch1.exe

Abdelrahman Hamdy
Mustafa Muhamed

Best Wishes!
.
250 2.0.0 5ACG6XvI003976 Message accepted for delivery
```

The email convinced the CEO to download a fake patch containing the payload, and it was delivered successfully due to the vulnerable SMTP service.

EXPLOITATION

During exploitation, social engineering was used to persuade the victim to run the disguised payload **securityPatch1.exe**. The victim downloaded and executed it, believing it was a legitimate update, which triggered the remote connection back to the attacker.

A custom script was used to automate Metasploit tasks, including setting up the listener and managing the reverse shell.

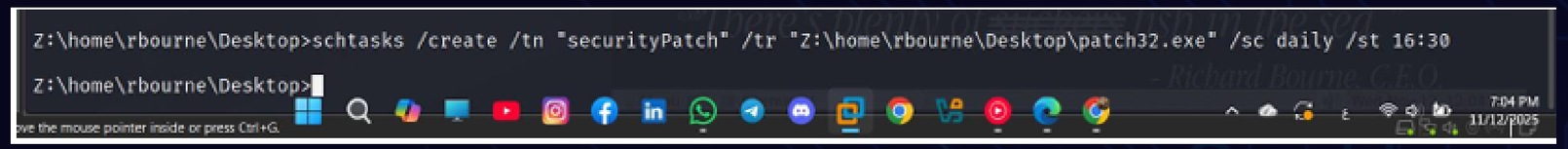


a remote desktop session was established to the target machine using **rdesktop -u rbourne -p password 192.168.10.129**

This provided full GUI access under the compromised user, allowing privilege escalation to root and resulting in complete administrative control of the system.

INSTALLATION

An initial persistence attempt was made by creating a scheduled task to run the payload at a set time, but this configuration did not execute successfully.



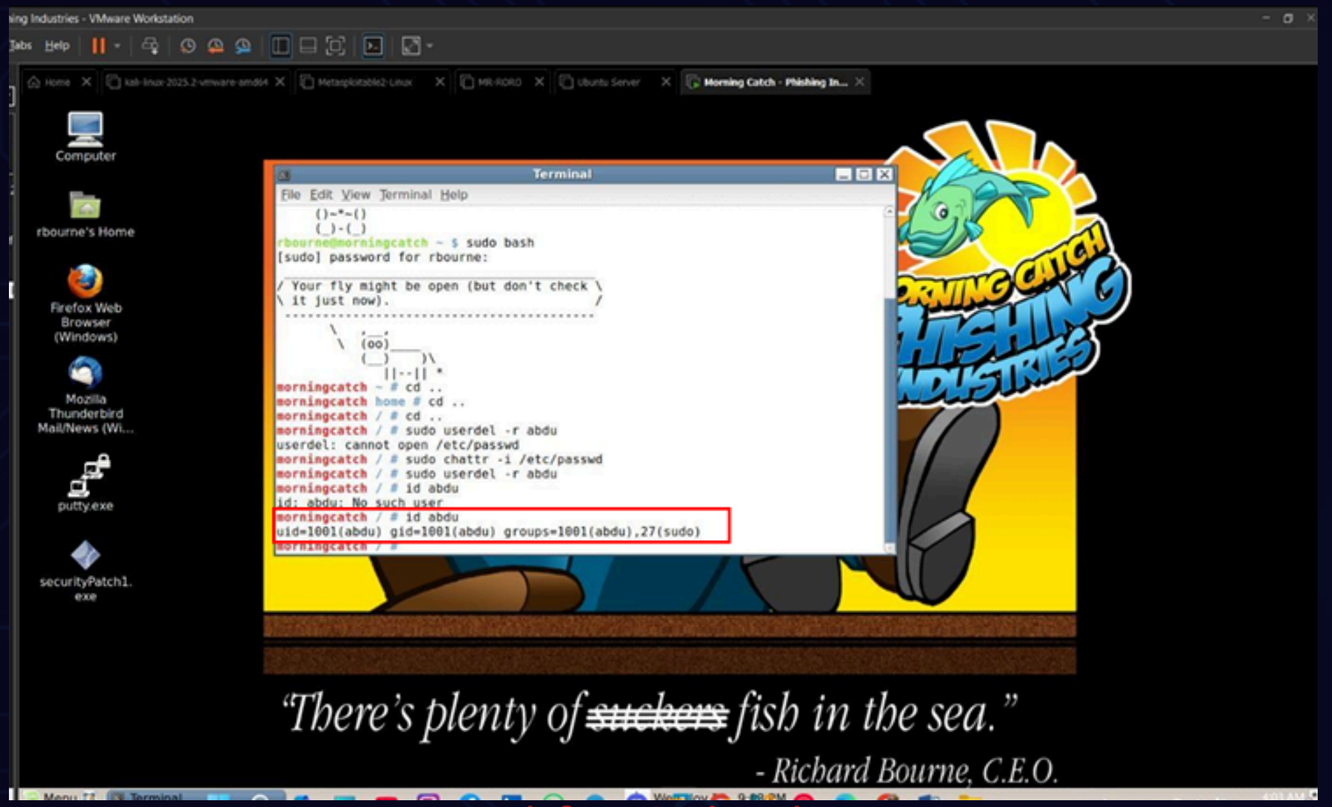
Task Name: securityPatch1.exe — named to appear as a legitimate system- related task
Executable: securityPatch1.exe (my payload).
Schedule: Daily at 16:30 (4:30 PM).

- RDP was used to gain full graphical access to the victim's desktop environment.
- After escalating to root, a new user account was created through system commands.
- The sudoers file was reviewed and modified to grant the new user full administrative privileges.
- This ensured persistent privileged access even if the original access was removed.

```
morningcatch etc # cd
morningcatch ~ # sudo EDITOR=nano visudo
```

INSTALLATION CONT.

To verify the persistence mechanism, the system was accessed again under a normal user account, and the backdoor user was manually deleted. After ten minutes, the user was automatically recreated with the same privileges, confirming that the cron job was functioning as intended. This demonstrated a self-healing persistence method capable of restoring access even after attempted removal.



(After 10 Minute)

COVERING ATTACK

To maintain privacy and ensure that sensitive system activity logs are not recoverable, I used the shred command to securely delete specific log files. Unlike a standard rm command, shred overwrites the file contents before deletion, making recovery significantly more difficult.

shred: A command-line utility used to overwrite a file to hide its contents and optionally delete it.

-u: Tells shred to truncate and remove the file after overwriting. This step helps prevent forensic recovery of logs that might reveal user activities or unauthorized access, which can be critical in penetration testing or controlled security assessments.

Commands Used

- **shred -u /var/log/syslog**

  Securely deletes the syslog file, which contains general system activity logs.

- **shred -u /var/log/auth.log**

  Securely deletes the auth.log file, which contains authentication and authorization-related entries (e.g., sudo access, user logins).

- **shred -u /var/log/xrdp.log**

  Securely deletes the xrdp.log file, which may include information about remote desktop sessions via XRDP.

KEY FINDINGS

The operation exposed major security weaknesses and showed how multiple attack steps could be combined to fully compromise the system.

Vulnerable Services

Outdated and unpatched services were found, creating potential entry points for attackers.

Weak Credential Practices

Valid username enumeration and a default password revealed weak credential management.

Lack of Network Segmentation

The ability to move laterally after initial access indicated weak network segmentation within the environment.

Insufficient Access Controls

The presence of publicly accessible user photos showed weak access controls and improper directory protections on the web server.

Persistence Mechanisms

Persistence was established, revealing long-term compromise risks and highlighting the need for stronger monitoring and detection.

Inadequate Logging and Monitoring

It showed that logging and monitoring were weak, making incident detection and analysis difficult.

Ineffective Social Engineering Countermeasures

The successful phishing attack on the CEO highlighted the organization’s vulnerability to social engineering and emphasized the need for stronger user awareness and email security.

PRIORITY-BASED RECOMMENDATIONS

Critical

Monitor & Restrict Cron/Schtasks Creation **(score: 10)**

Audit & Lock Down Scheduled Tasks **(score: 9)**

High

Centralize & Protect Log Storage; Monitor File Deletion **(score: 8)**

Monitor File Attribute Changes; Deploy FIM **(score: 7)**

Audit sudoers & Enforce Least Privilege **(score: 7)**

Medium

Monitor Access to /etc/passwd & /etc/shadow **(score: 5)**

Low

Disable Directory Listing & Restrict Web Directories **(score: 3)**

THANK YOU

Abdelrahman Hamdy Abdellatif Mahmoud

Mustafa Muhamed Elsyed

Farah Ayman yakout