# TryHackMe: Splunk 2 Walkthrough (splunk2gcd5)

Onur Alp Akin · Follow

12 min read · Apr 13, 2023

Listen Share

Check out Splunk 2 room on TryHackMe

Based on version 2 of the Boss of the SOC (BOTS) competition by Splunk.

Original Publish Date: Dec 31, 2022

## 100 Series Questions

The first objective is to find out what competitor website she visited. What is a good starting point?

When it comes to HTTP traffic, the source and destination IP addresses should be recorded in logs. You need Amber's IP address.

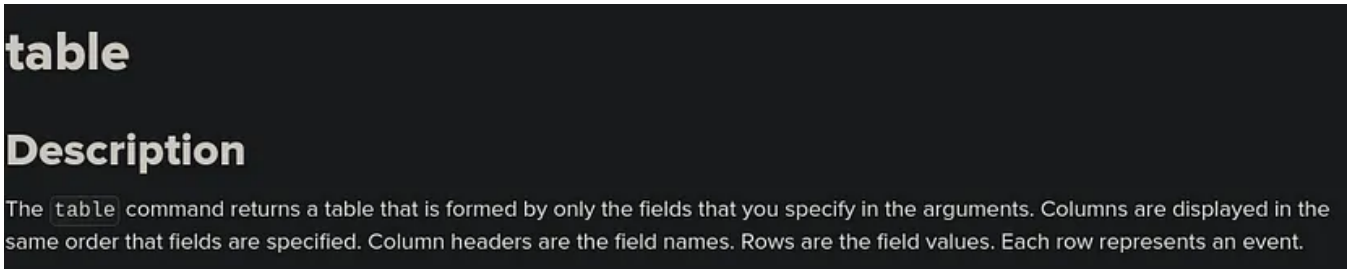
I start with a simple query to find Amber's IP address

```
index="botsv2" sourcetype="pan:traffic" amber
```


Found it

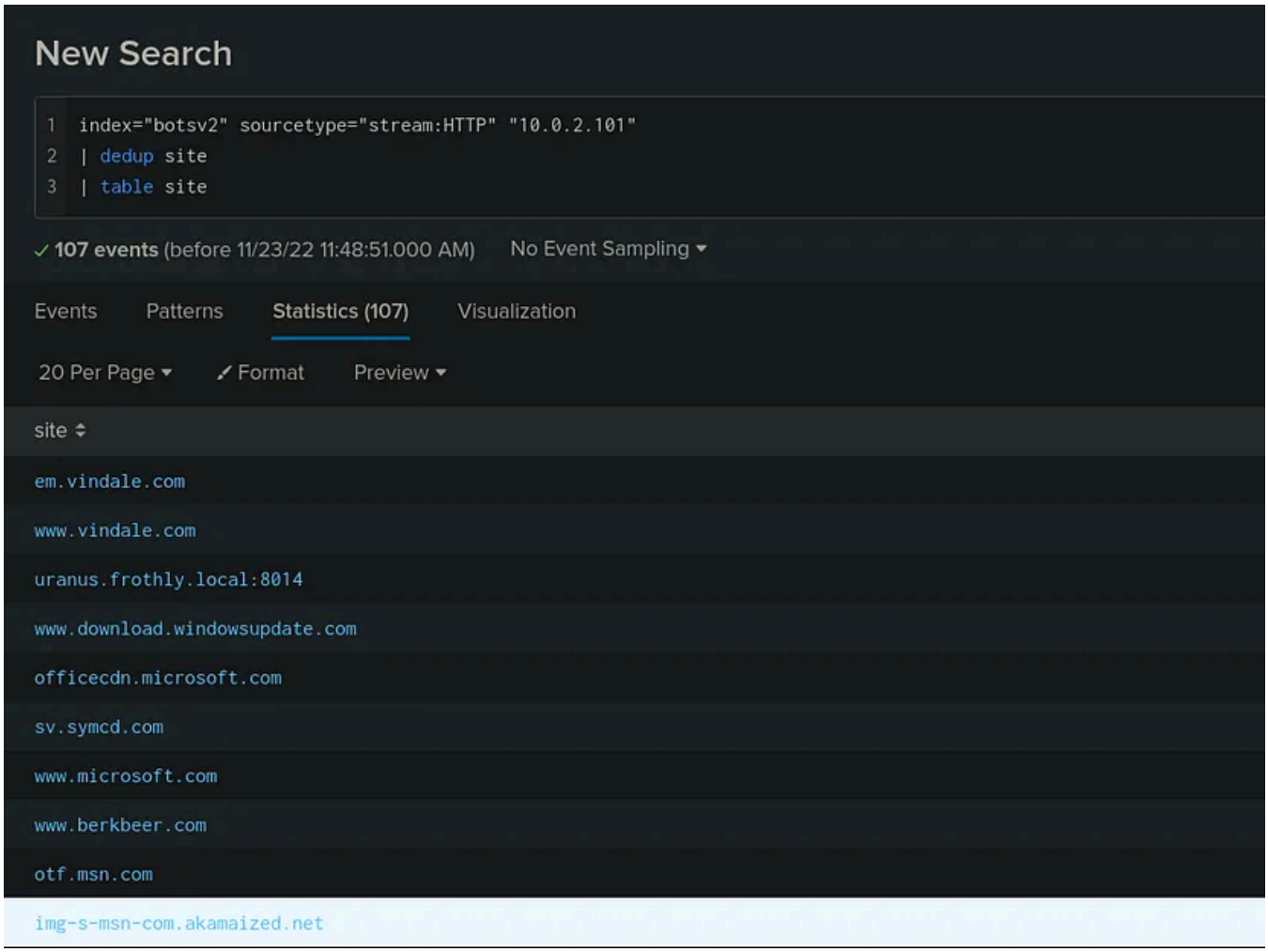
After adding it to the search and changing source type to HTTP, room wants us to add some keywords to our query. More specifically, one to remove duplicate entries and one to list as a table.

Looking at the reference and searching inside the page, we can easily find our related keywords


Final query would be

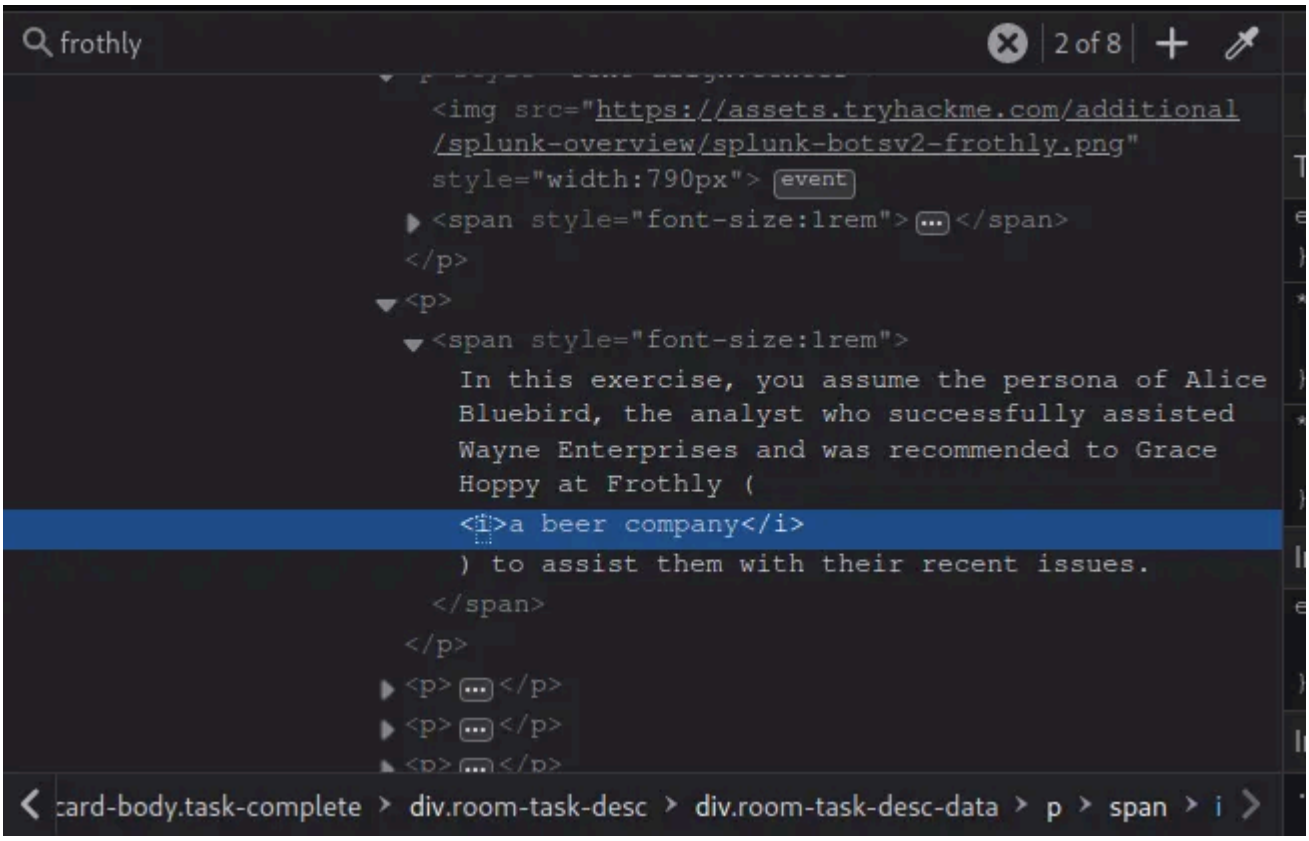
```
index="botsv2" sourcetype="stream:HTTP" "10.0.2.101"
| dedup site
| table site
```




To continue with the room, we have to find competitor website out of these.

And room says you can use industry, which Frothly is in. It's an imaginary company, thus you won't get anything by searching online :)

Tried to find in page, but in vain. Then I searched inside HTML with developer tools and found what we are looking for.




**1 — Amber Turing was hoping for Frothly to be acquired by a potential competitor, which fell through, but visited their website to find contact information for their executive team. What is the website domain that she visited?**

```
www.berkbeer.com
```

**2 — Amber found the executive contact information and emailed him. What image file displayed the executive's contact information? Answer example: /path/image.ext**

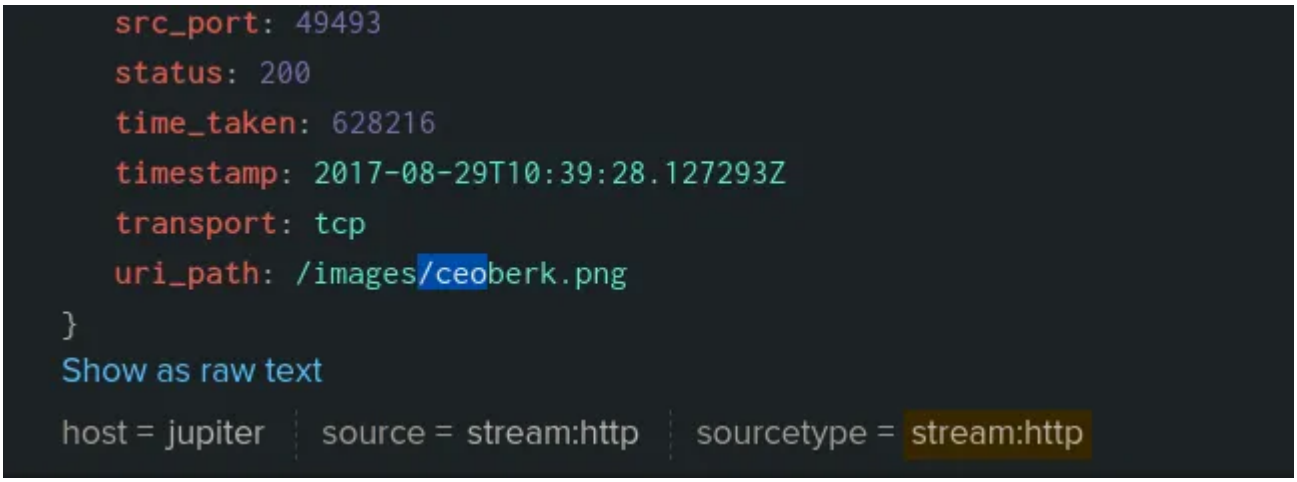
Query is now

```
index="botsv2" sourcetype="stream:HTTP" "10.0.2.101" berkbeer.com
```

Just guessed that filename would include the abbreviation CEO in it.

So our answers is

```
/images/ceoberk.png
```




Now to find email related answers, we need to change source type to SMTP.

```
index="botsv2" sourcetype="stream:smtp" berkbeer.com
```


We found Amber's email. Now we can add that to our search

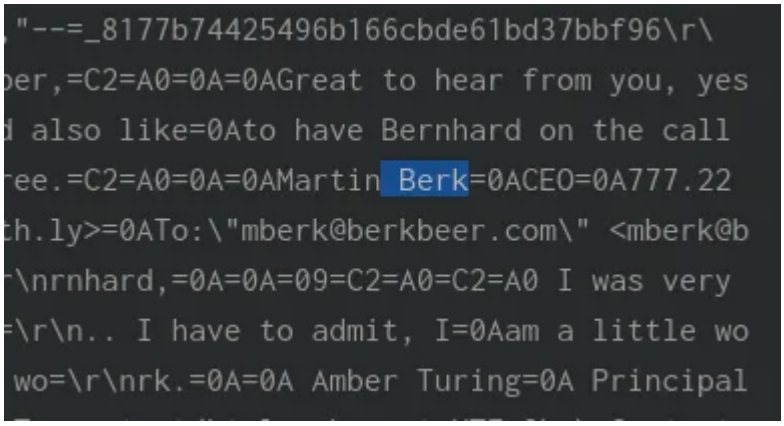
```
index="botsv2" sourcetype="stream:smtp" berkbeer.com "aturing@froth.ly"
```

We got 4 results. It's quite manageable.

**3 — What is the CEO's name? Provide the first and last name.**

In order to search every data column, I clicked all 4 "show as raw text" buttons and searched `[space]berk` in the page.

If it were more than a dozen, it would be a good idea to search with regex, however I didn't bother here.




We found it

Answer is:

```
Martin Berk
```

**4 — What is the CEO's email address?**

We can see an email right under the name which is

```
mberk@berkbeer.com
```

**5 — After the initial contact with the CEO, Amber contacted another employee at this competitor. What is that employee's email address?**

If we were to pay attention to the data column "receiver" under one of four packets, we can find the email in question.

```
hbernhard@berkbeer.com
```

```
received_date: [ [+]
]
receiver: [ [-]
  "'hbernhard@berkbeer.com'" <hbernhard@berkbeer.com>
]
receiver_alias: [ [-]
  'hbernhard@berkbeer.com'
]
receiver_email: [ [-]
  hbernhard@berkbeer.com
]
receiver_type: [ [+]
]
```

**6 — What is the name of the file attachment that Amber sent to a contact at the competitor?**

We can utilize Interesting Fields

```
Saccharomyces_cerevisiae_patent.docx
```

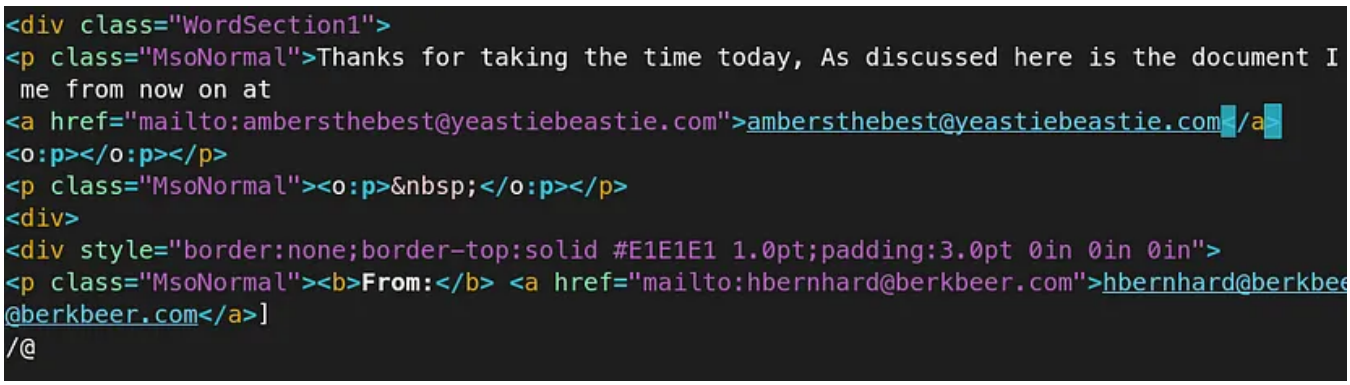

**7 — What is Amber's personal email address?**

After spending 10 to 15 minutes searching various email regexes, I couldn't find anything and decided to look at the hint which says look for encrypted data.

After the hint, I returned to the aforementioned 4 packets because one of them included lots of inconspicuous base64 data.


Using CyberChef to decode the longest looking base64 and searching for `@` character, we can find an email.

```
ambersthebest@yeastiebeastie.com
```



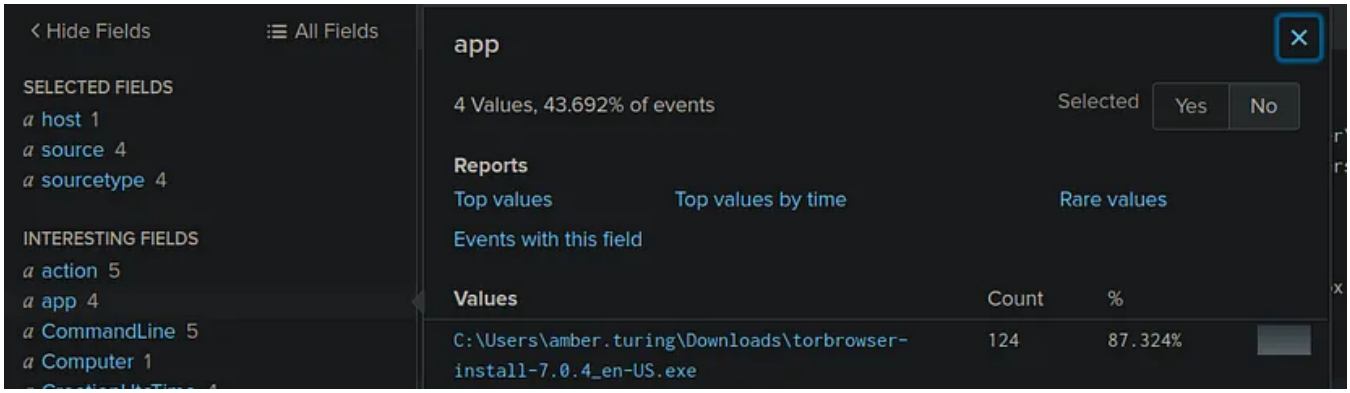

## 200 Series Questions

Starting with the query that we are given.

```
index="botsv2" amber tor
```

**1 — What version of TOR Browser did Amber install to obfuscate her web browsing? Answer guidance: Numeric with one or more delimiter.**

Because there are more than 300 results, I've decided to take a look at interesting fields one by one.




Et voilà! Answer is:

```
7.0.4
```

**2 — What is the public IPv4 address of the server running www.brewertalk.com?**

To find the IP, we can start simple.

```
index="botsv2" brewertalk.com
```

Because most of the destination port is 80 I'm including that in the search.

```
index="botsv2" brewertalk.com dest_port=80
```

By doing so, we now have only 2 IPs


The IP we are looking for would be the 2nd

```
52.42.208.228
```

**3 — Provide the IP address of the system used to run a web vulnerability scan against www.brewertalk.com.**

If we think about it simply, scan would send lots of packages, and we should look for the IP that has sent the most packages.

Searching

```
index="botsv2" www.brewertalk.com
```

Again, utilizing interesting fields, the answer is:

```
45.77.65.211
```

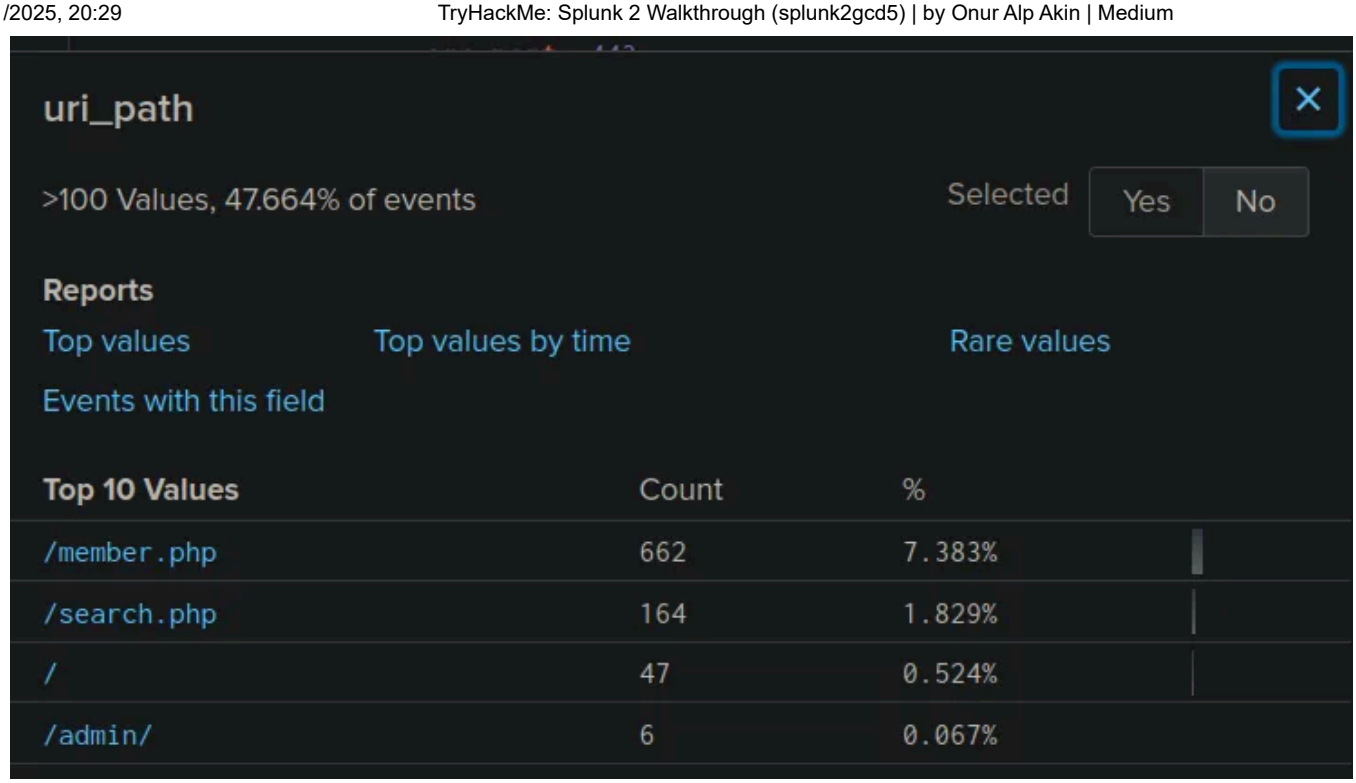

**4 — The IP address from Q#2 is also being used by a likely different piece of software to attack a URI path. What is the URI path? Answer guidance: Include the leading forward slash in your answer. Do not include the query string or other parts of the URI. Answer example: /phpinfo.php**

Base query is

```
index="botsv2" src_ip="45.77.65.211"
```

It returned lots of results. Once again, interesting fields come to our aid.

URI path field is most likely to reveal attacked path




You would expect one of the top results would be the answer. And because we know the answer format, anything but the first result seems unlikely because the second one is a search page.

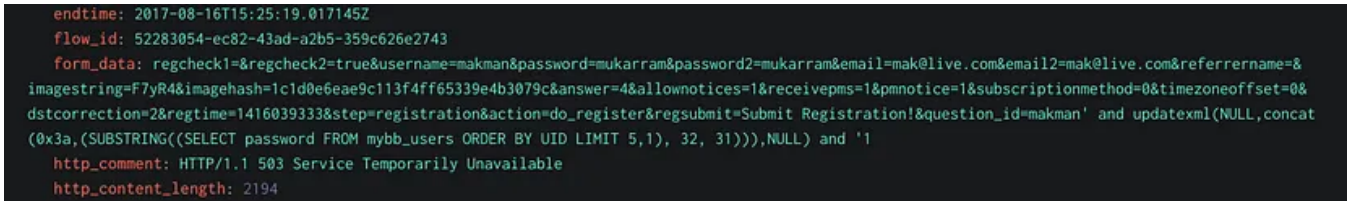
**5 — What SQL function is being abused on the URI path from the previous question?**

As per our recent findings now our query looks like this:

```
index="botsv2" src_ip="45.77.65.211" uri_path="/member.php"
```

Looking at the first item's 'form data' field, we can see the utilized function:

```
updatexml
```




Form data field

**Questions 6 & 7**

Awesome, thus far, you have identified Amber downloaded Tor Browser (you even know the exact version). You identified what URI path and the SQL function attacked on brewertalk.com.

Your task now is to identify the cookie value that was transmitted as part of an XSS attack. The user has been identified as Kevin.

Before diving right in, get some details on Kevin. This is the first time you hear of him.

Command:

```
index="botsv2" kevin
```

Ok, now you have Kevin's first and last name. Time to figure out the cookie value from the XSS attack.

As before, you can start with a simple keyword search.

You know that you're looking for events related to Kevin's HTTP traffic with an XSS payload, and you're focused on the cookie value.

Honestly, you should be able to tackle this one on your own as well. Use the previous search queries as your guide.

After you executed the search query that yields the events with the answer, you can identify the username used for the spear phishing attack.

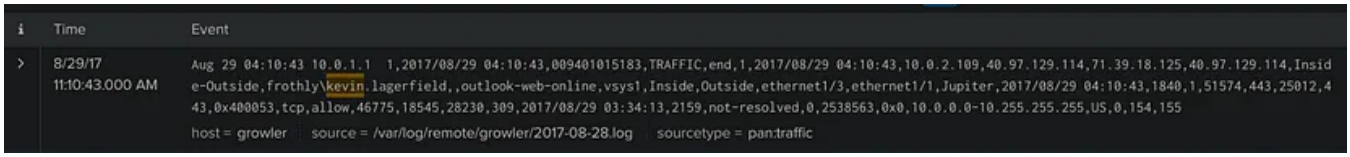
Based on the question hint, you can perform a keyword search query here as well.

**6 — What was the value of the cookie that Kevin's browser transmitted to the malicious URL as part of an XSS attack? Answer guidance: All digits. Not the cookie name or symbols like an equal sign.**

Suggested by question info we start with:

```
index="botsv2" kevin
```

And we learn his last name, which is `lagerfield`




Searching with the last name didn't give me much to work with, so I continue with just the name.

We know from the "story" that stream should be HTTP.

```
index="botsv2" kevin sourcetype="stream:http"
```

And using our knowledge from the previous queries, we know `brewertalk.com` uses PHP, so it would be a good assumption to say the attacked endpoint would include PHP in its URI path.

And to top it off, we can include script or document keywords.

So, final query would be:

```
index="botsv2" kevin sourcetype="stream:http" (script OR document) uri_path="*\
```

We must not forget we are looking for a cookie value :)

Peeking through the cookie field (all digits, first one)

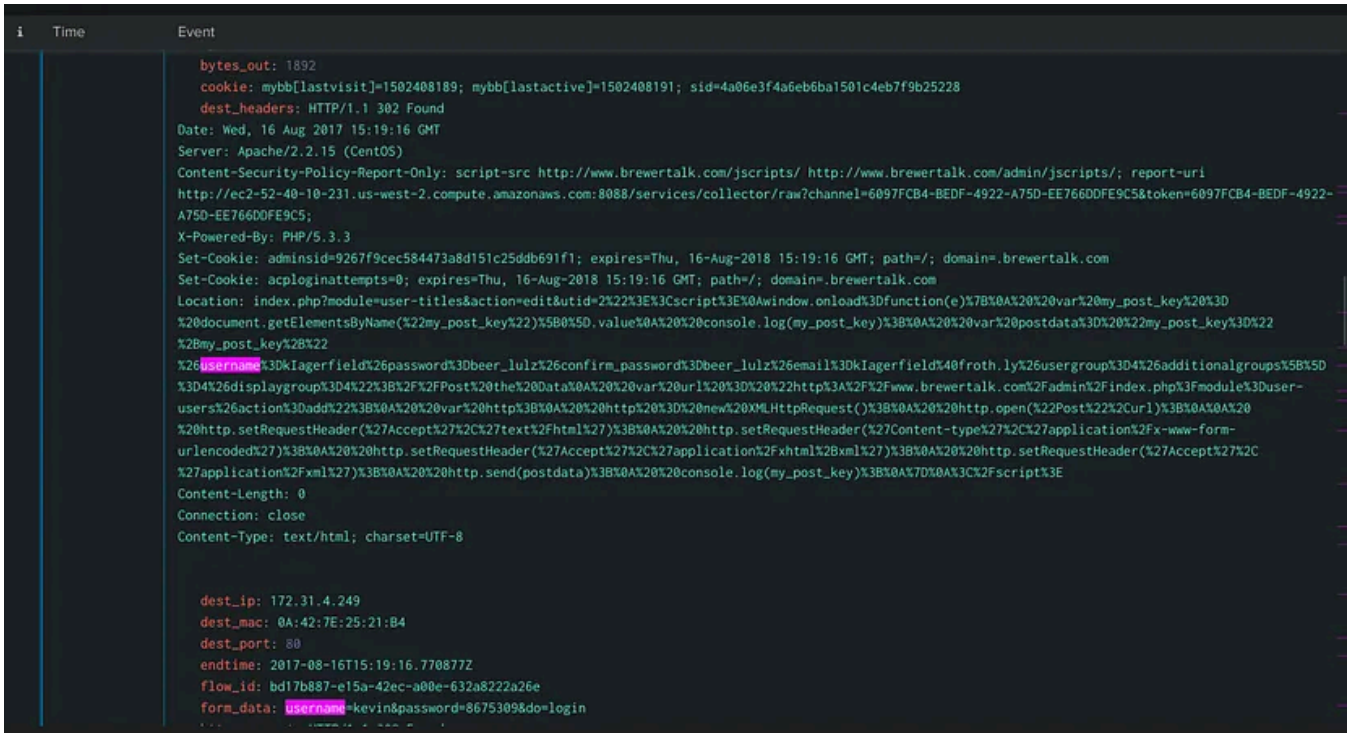
```
1502408189
```


**7 — What brewertalk.com username was maliciously created by a spear phishing attack?**

Our results from the previous query is manageable enough, so I'm just going to search in page for username :)

You can see clearly barring '`kevin`' only username is:

```
kIagerfield
```




## 300 Series Questions

Upward and onwards! Time to tackle some of the 300 series questions.

As with the 100 series questions, there are extra questions in this task that are not from the BOTS2 dataset.

**Questions 1 & 2**

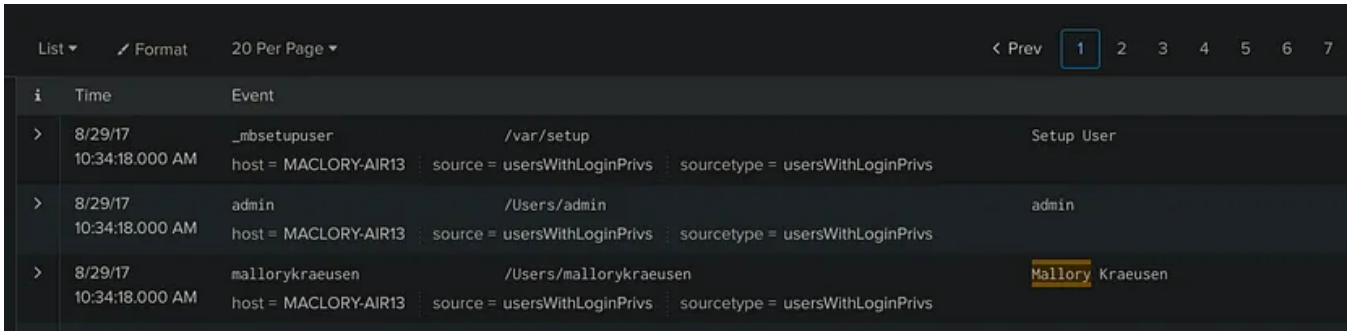
The questions start with an individual named Mallory, her MacBook, and some encrypted files.

As per the previous tasks, you can start with a keyword search to see what events are returned that are associated with Mallory.

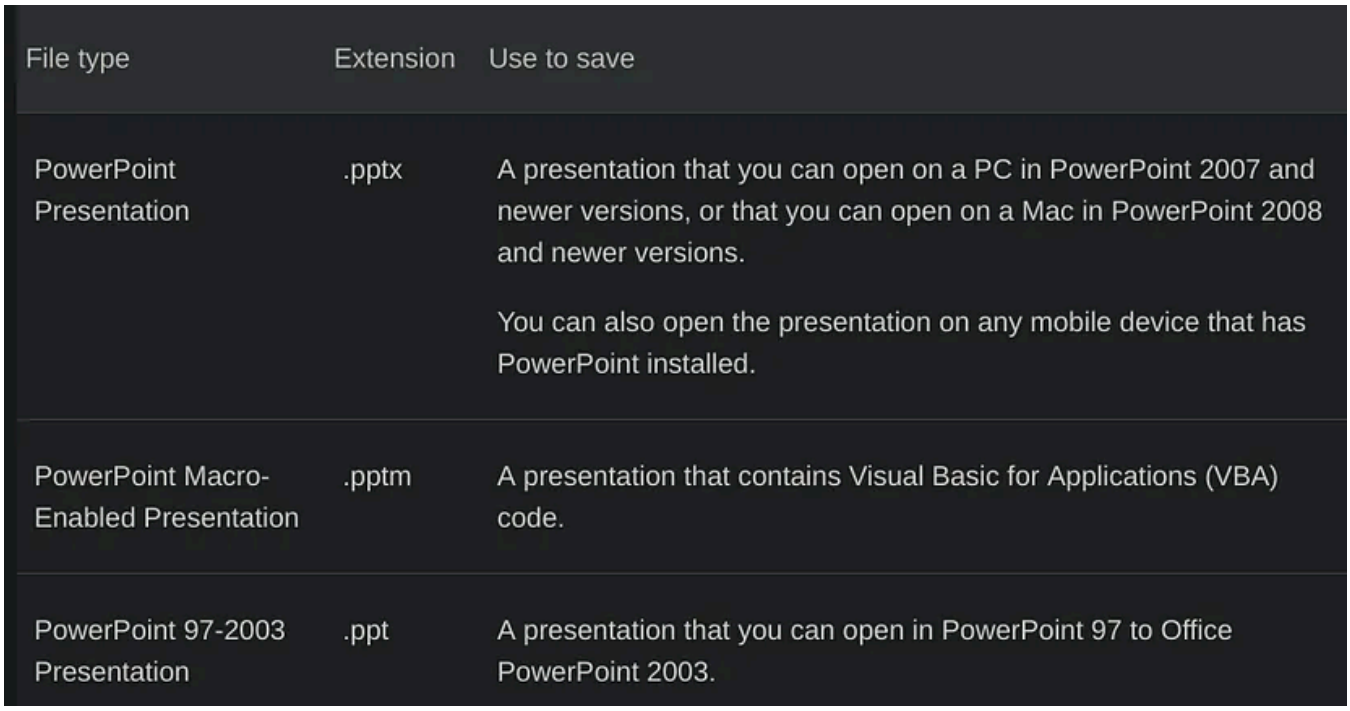
**1 — Mallory's critical PowerPoint presentation on her MacBook gets encrypted by ransomware on August 18. What is the name of this file after it was encrypted?**

Starting with a simple keyword:

```
index="botsv2" mallory
```




We found the hostname and or query should include file extension for PowerPoint.



| File type | Extension | Use to save |
| --- | --- | --- |
| PowerPoint Presentation | .pptx | A presentation that you can open on a PC in PowerPoint 2007 and newer versions, or that you can open on a Mac in PowerPoint 2008 and newer versions.<br><br>You can also open the presentation on any mobile device that has PowerPoint installed. |
| PowerPoint Macro-Enabled Presentation | .pptm | A presentation that contains Visual Basic for Applications (VBA) code. |
| PowerPoint 97-2003 Presentation | .ppt | A presentation that you can open in PowerPoint 97 to Office PowerPoint 2003. |

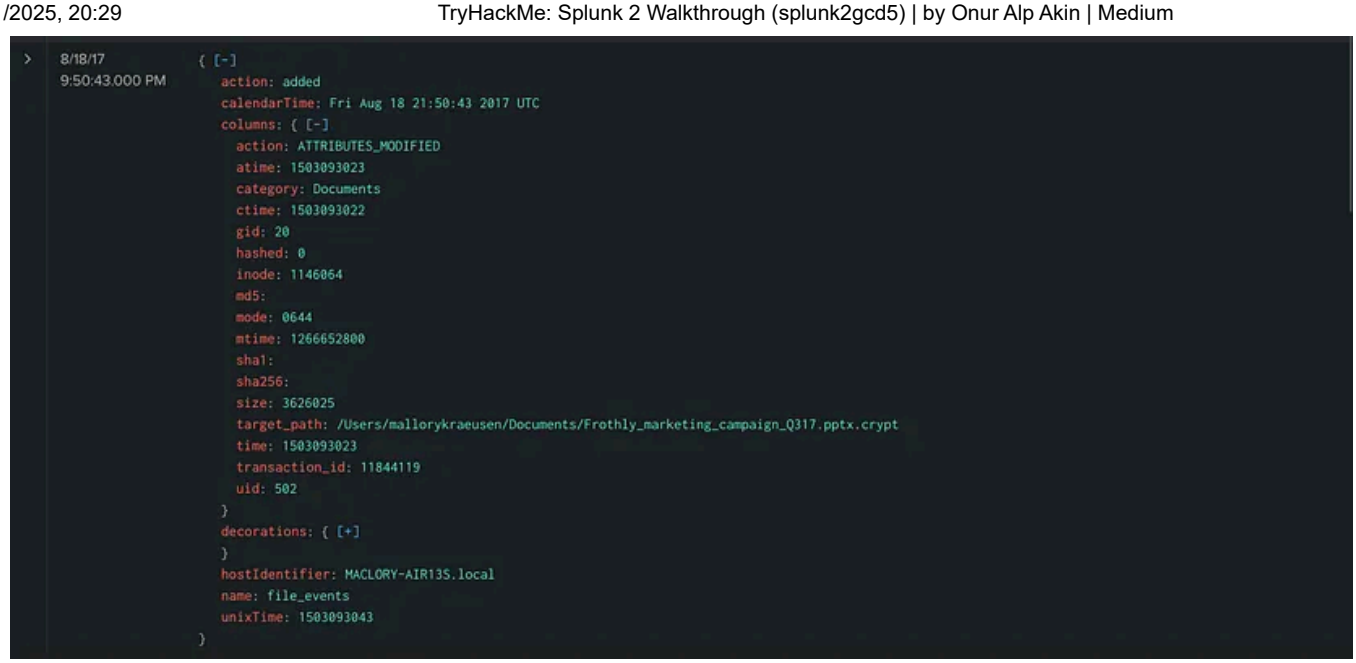
Microsoft documentation for PowerPoint extensions

Now our query looks like this:

```
index="botsv2" host="MACLORY-AIR13" (*.pptx OR *.pptm OR *.ppt)
```

So the answer is:

```
Frothly_marketing_campaign_Q317.pptx.crypt
```




### 2 — There is a ‘Games of Thrones’ movie file that was encrypted as well. What season and episode is it?

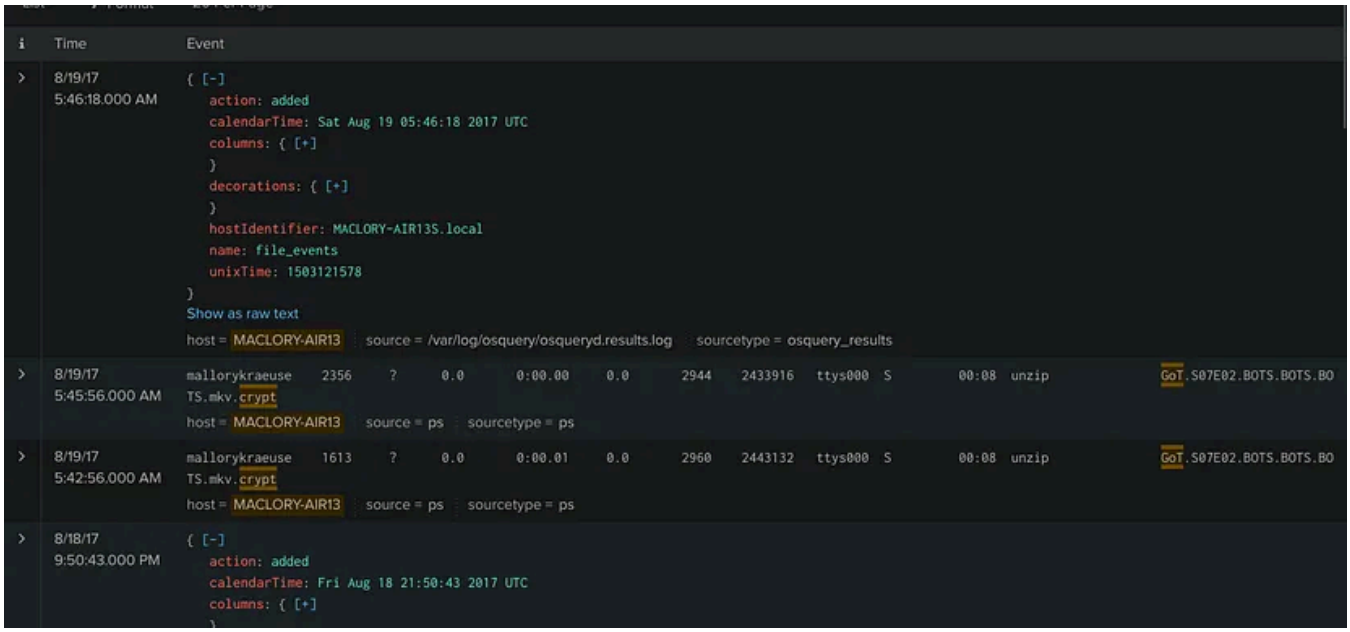
We know encrypted file extension which is `.crypt`

Now assuming the file would contain the series’ name, our query would be:

```
index="botsv2" host="MACLORY-AIR13" (got OR game OR thrones) crypt
```

And the answer is:

```
S07E02
```




**3 — Kevin Lagerfield used a USB drive to move malware onto `kutekitten`, Mallory's personal MacBook. She ran the malware, which obfuscates itself during execution. Provide the vendor name of the USB drive Kevin likely used. Answer Guidance: Use time correlation to identify the USB drive.**

Start query is:

```
index="botsv2" kutekitten
```

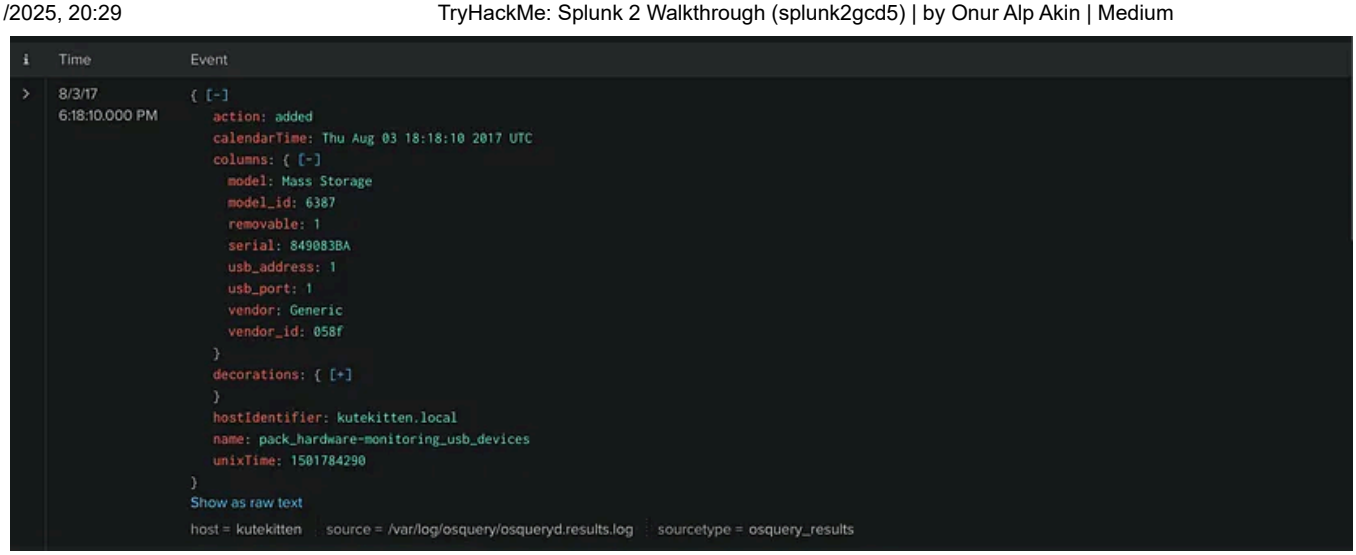
But it gives over 6k results. To reduce the number, I add USB keyword.

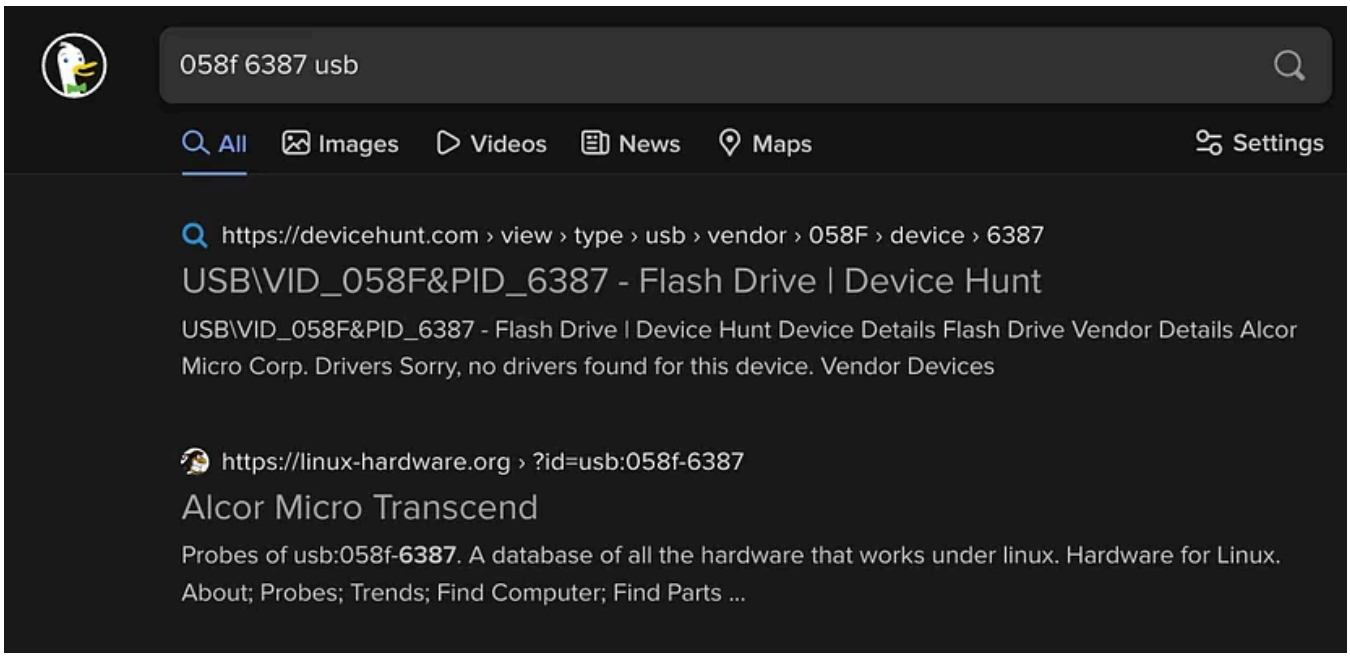
```
index="botsv2" kutekitten usb
```

While looking at the columns, I see tag column that has USB value too. So I select that. And I add vendor keyword too.

```
index="botsv2" kutekitten usb tag=usb vendor
```

Now that we have both vendor and device ID, we can look it up.




```
Alcor Micro Corp.
```

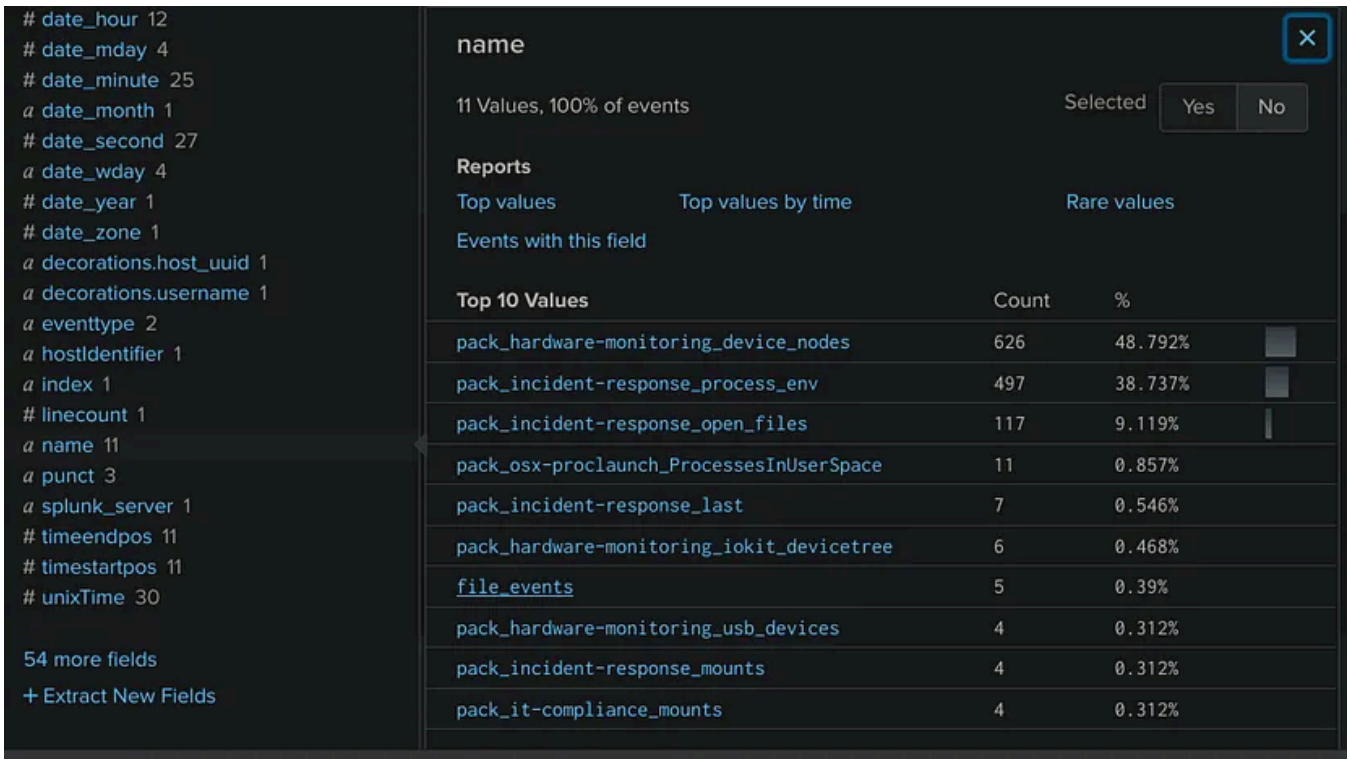
**4 — What programming language is at least part of the malware from the question above written in?**

First, we have got to get back to our first query, which is:

```
index="botsv2" kutekitten
```

Looking at the osquery results, I notice something suspicious under `/Users` which is `mkraeusen` user. And after adding the username as a keyword looking at interesting

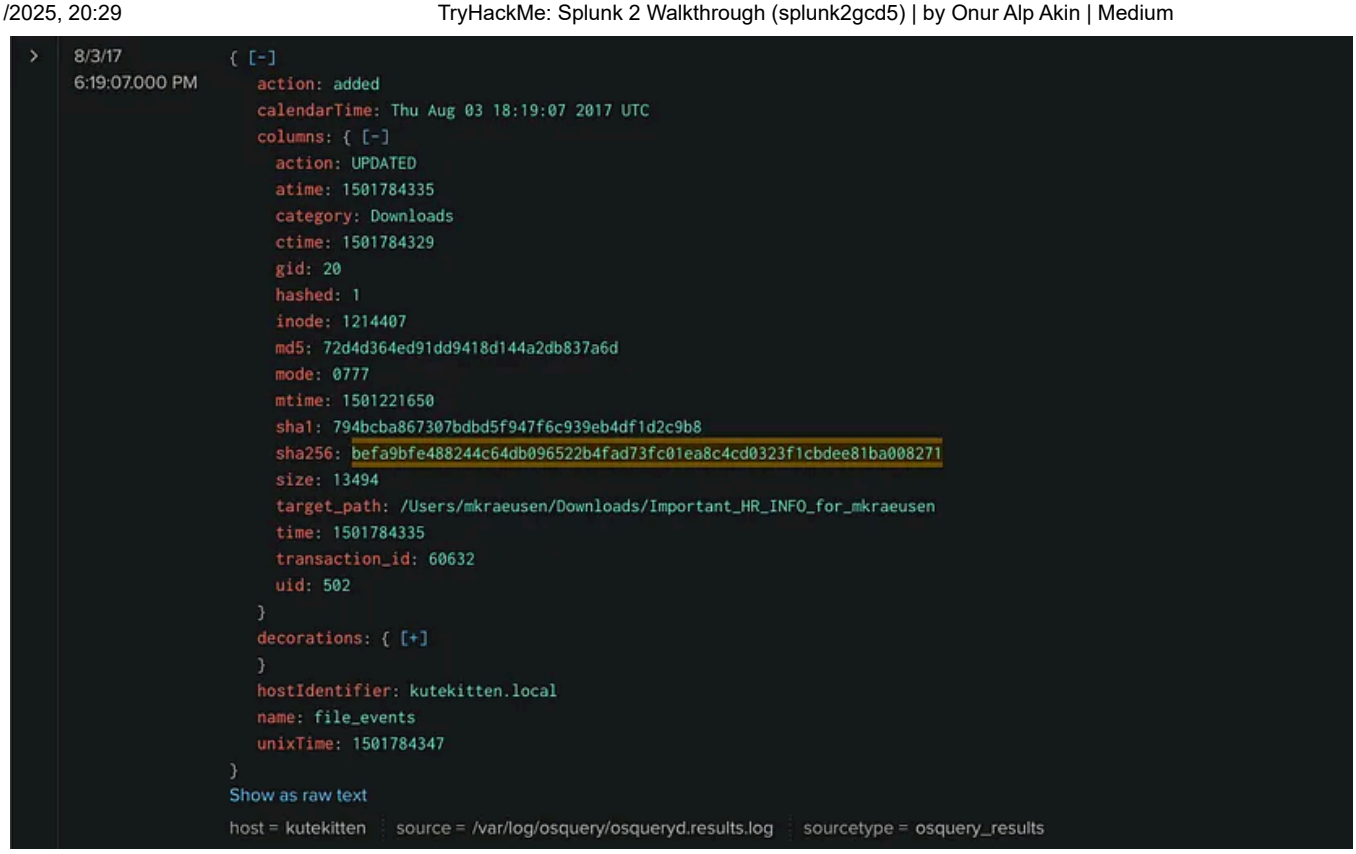
field names, something looks peculiar.




So now our query is:

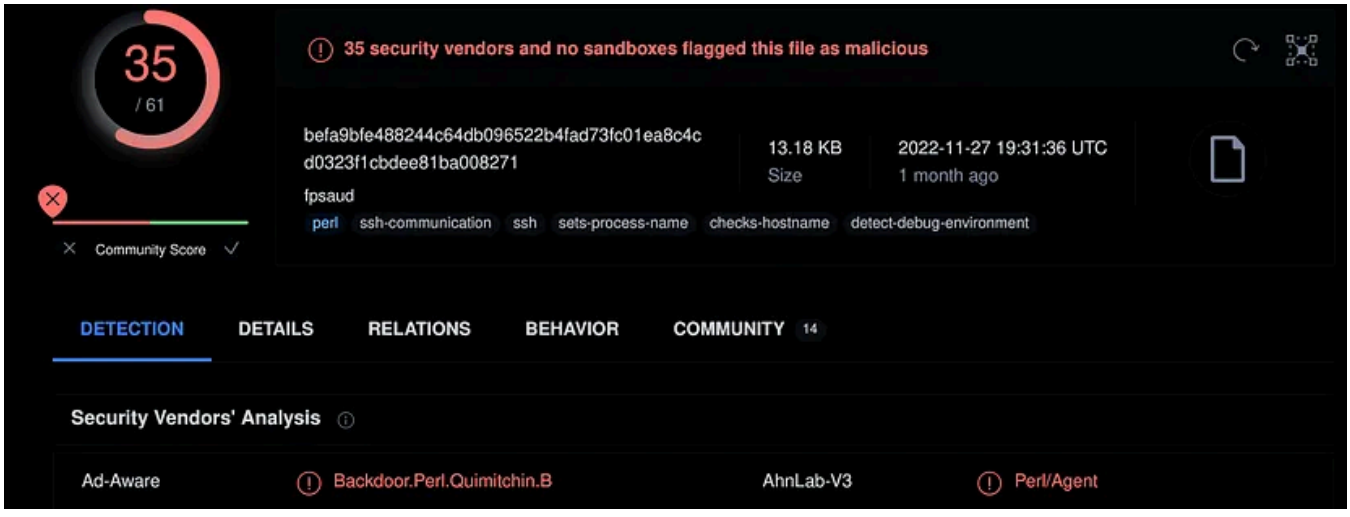
```
index="botsv2" kutekitten mkraeusen name=file_events
```

Et voilà! We have a file hash.




And searching for the hash on virustotal.com, we got our result:

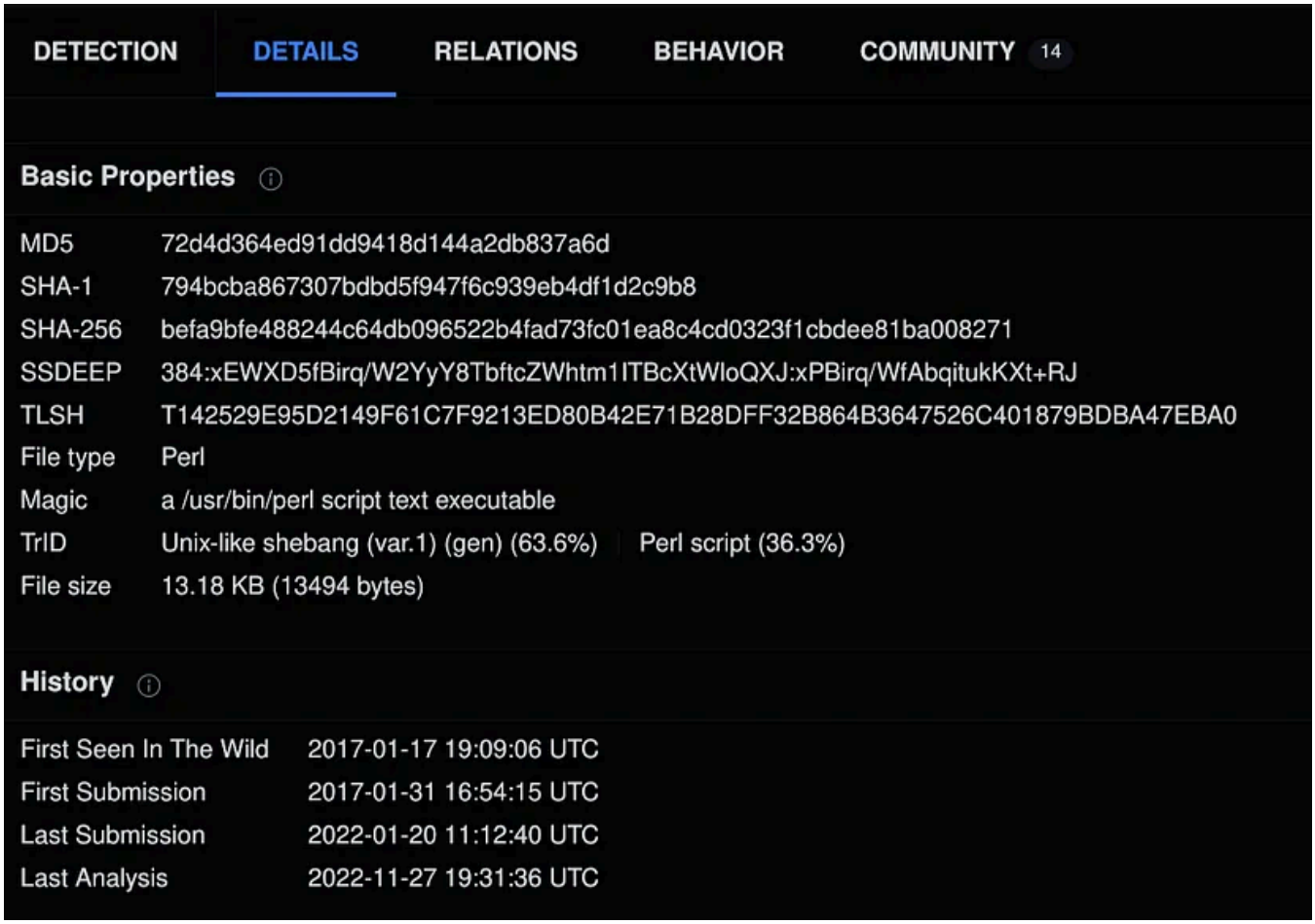
```
perl
```




**5 — When was this malware first seen in the wild? Answer Guidance:** `YYYY-MM-DD`

Still on `virustotal`, under details tab:

```
2017-01-17
```




**6 — The malware infecting `kutekitten` uses dynamic DNS destinations to communicate with two C&C servers shortly after installation. What is the fully-qualified domain name (FQDN) of the first (alphabetically) of these destinations?**

Relations tab:

```
eidk.duckdns.org
```


**7 — From the question above, what is the fully-qualified domain name (FQDN) of the second (alphabetically) contacted C&C server?**

```
eidk.hopto.org
```

## 400 Series Questions

**1 — A Federal law enforcement agency reports that Taedonggang often spear phishes its victims with zip files that have to be opened with a password. What is the name of the attachment sent to Frothly by a malicious Taedonggang actor?**

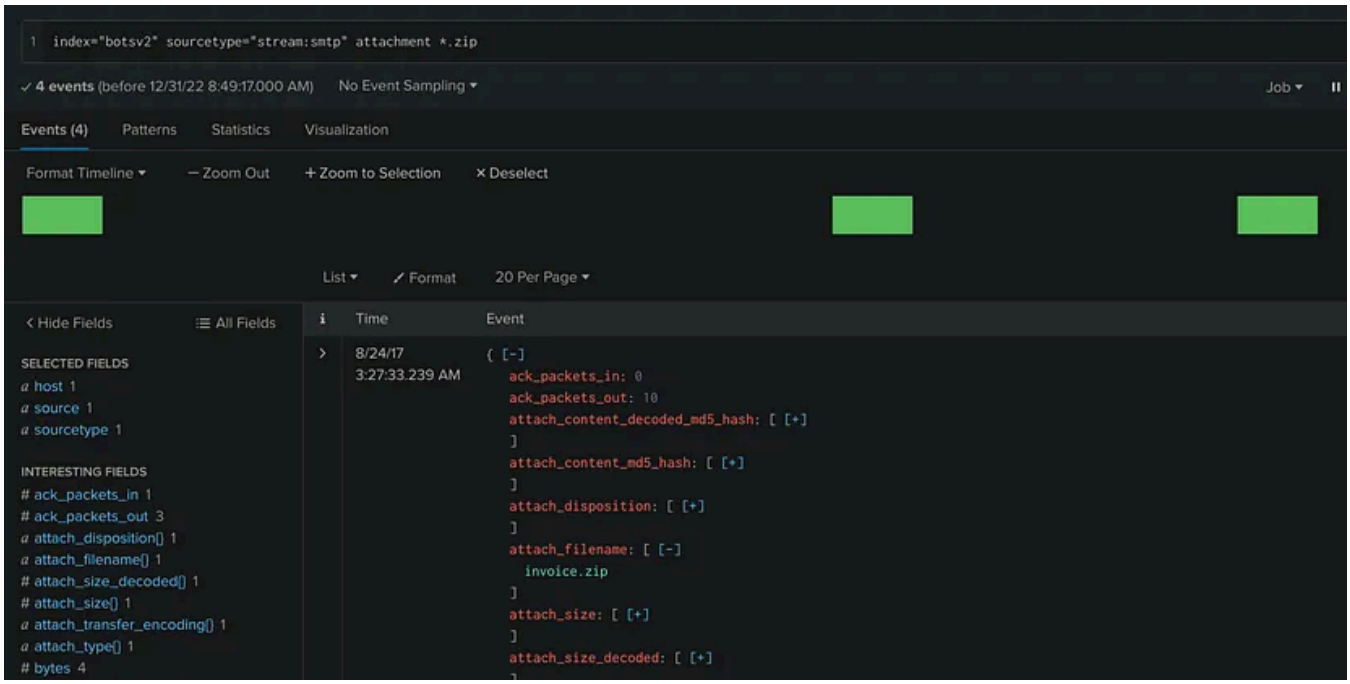
Now, because we are dealing with emails here, it makes sense to filter results to SMTP packets. After source filter, we still have lots to deal with. So adding 'attachment' keyword and zip extension also makes sense.

The final query would look like this:

```
index="botsv2" sourcetype="stream:smtp" attachment *.zip
```

And the answer is:

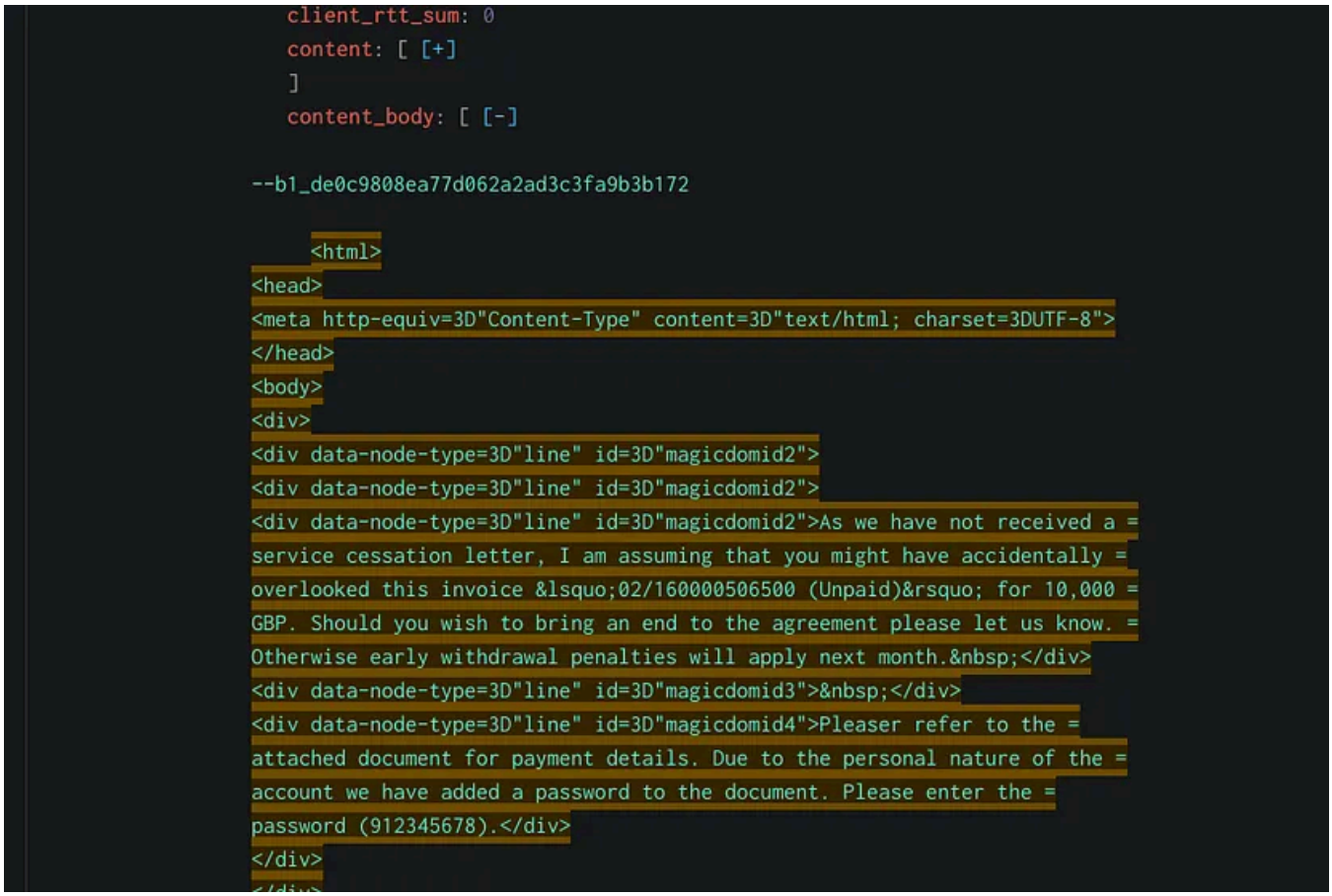
```
invoice.zip
```




### 2 — What is the password to open the zip file?

Same email, content body includes the password.

```
912345678
```



**3 — The Taedonggang APT group encrypts most of their traffic with SSL. What is the "SSL Issuer" that they use for the majority of their traffic? Answer guidance: Copy the field exactly, including spaces.**

For this question, you will need the attacker's IP. Remember, there was an IP address scanning brewertalk.com.

Starting with the query including IP address and SSL keyword

```
index="botsv2" SSL 45.77.65.211
```

Interesting field > `ssl_issuer` gives the answer:

```
C = US
```


**4 — What unusual file (for an American company) does winsys32.dll cause to be downloaded into the Frothly environment?**

Initial query:

```
index="botsv2" winsys32.dll
```

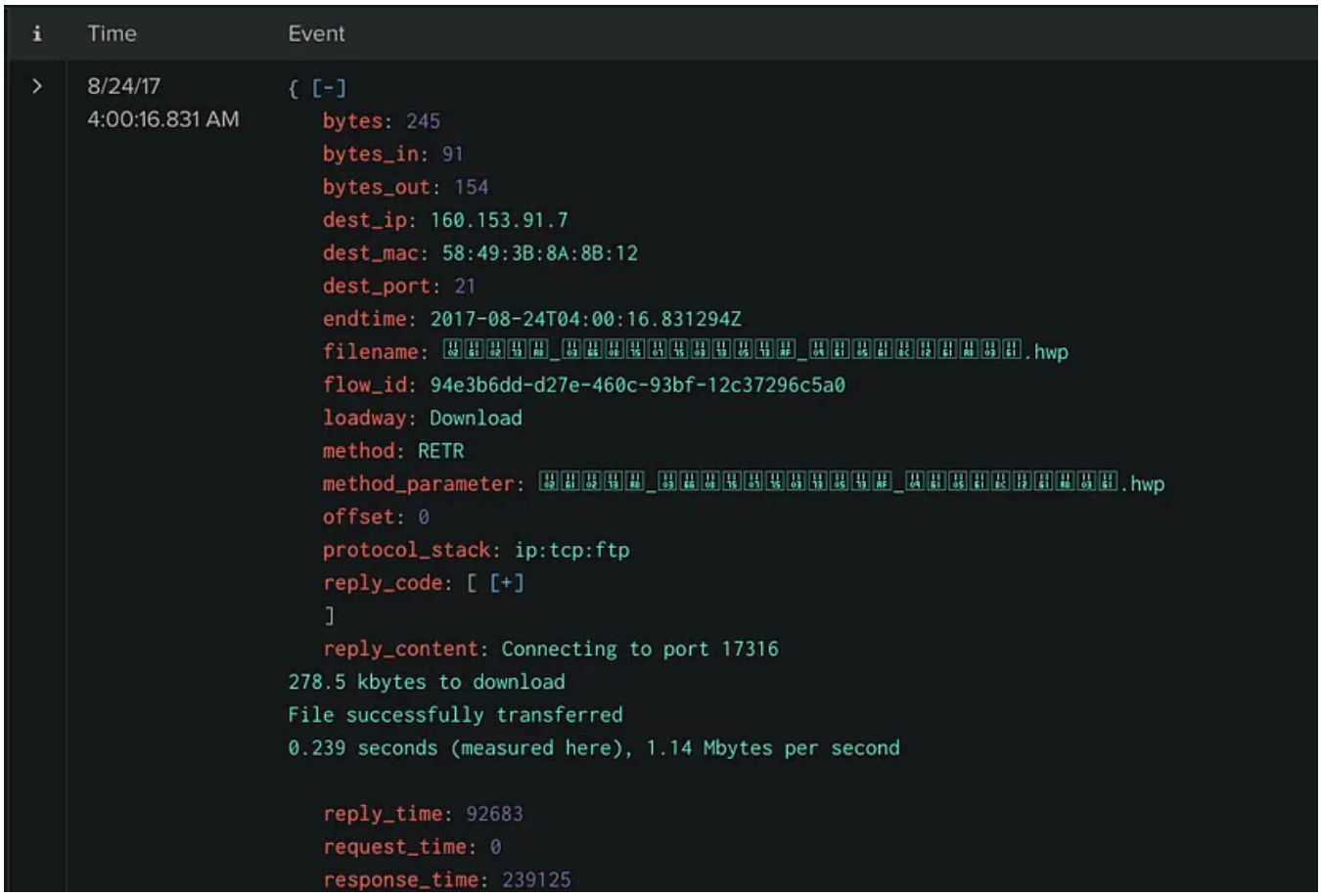
From there we can see ftp client running. We change our source to ftp stream.

```
index="botsv2" sourcetype="stream:ftp"
```

Now we can select loadway key from interesting fields. We are looking for downloads.

```
index="botsv2" sourcetype="stream:ftp" loadway=Download
```


```
나는_데이비드를_사랑한다.hwp
```



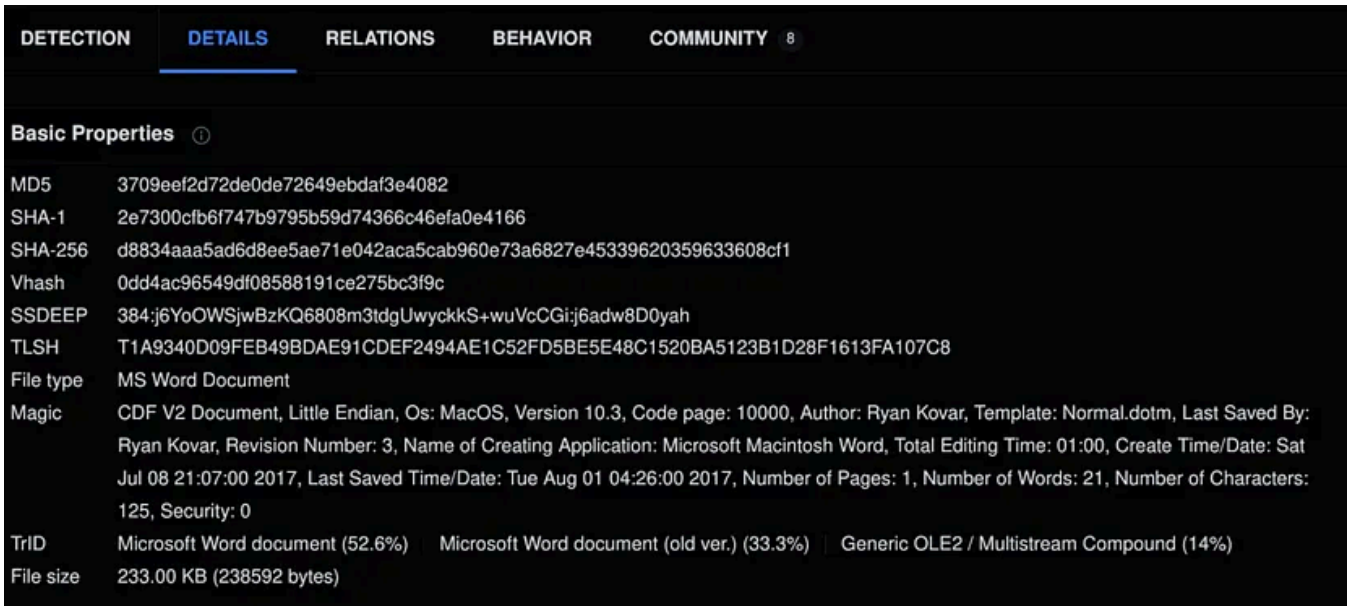

We found the filename

**5 — What is the first and last name of the poor innocent sap who was implicated in the metadata of the file that executed PowerShell Empire on the first victim's workstation? Answer example: John Smith**

Use the following links to examine the execution of the malware contained within the aforementioned zip file.

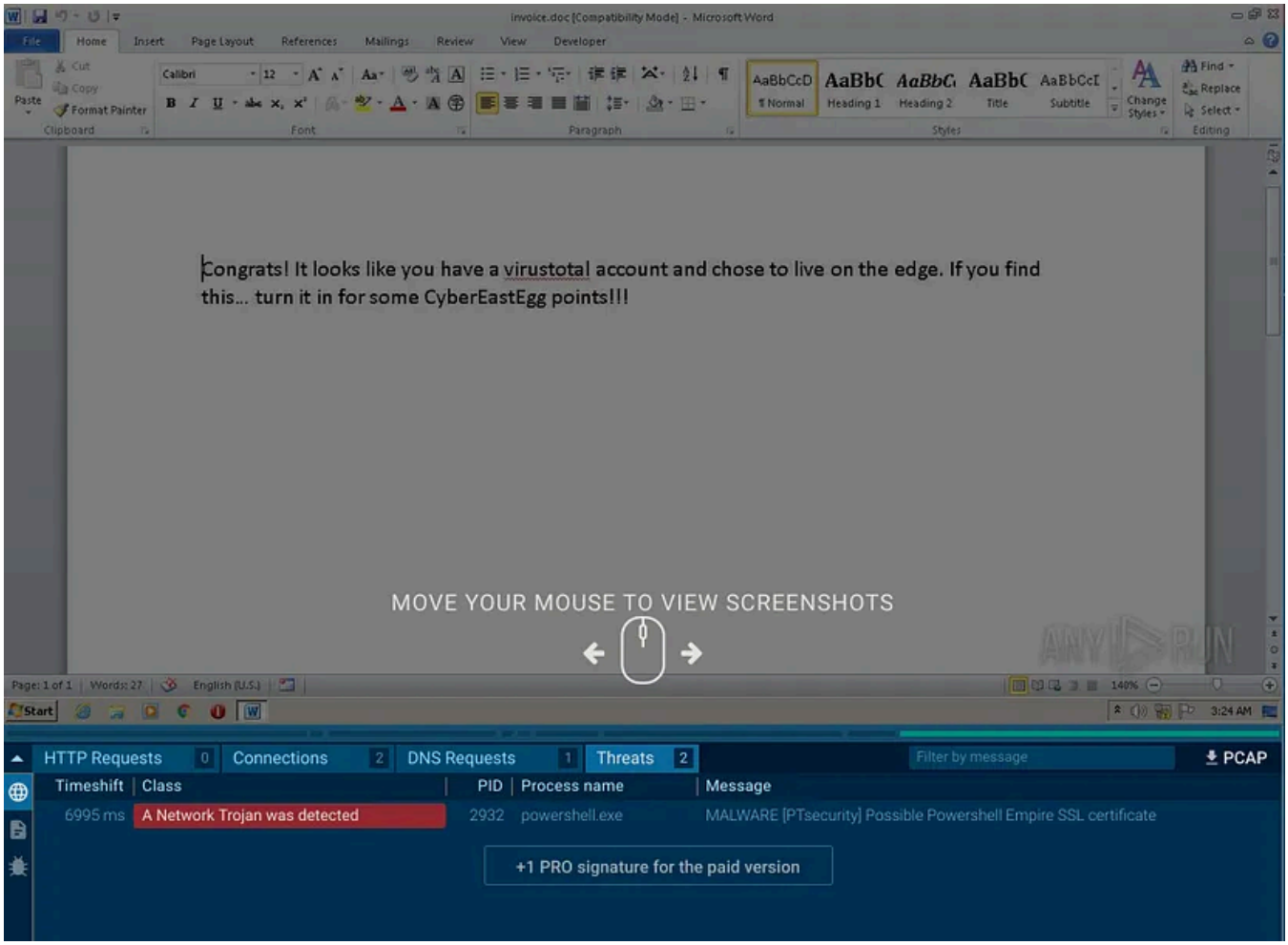
Hybrid Analysis
VirusTotal
Any.run

Ryan Kovar




## 6 — Within the document, what kind of points is mentioned if you found the text?

```
CyberEastEgg
```




## 7 — To maintain persistence in the Frothly network, Taedonggang APT configured several Scheduled Tasks to beacon back to their C2 server. What single webpage is most contacted

**by these Scheduled Tasks? Answer example: index.php or images.html**

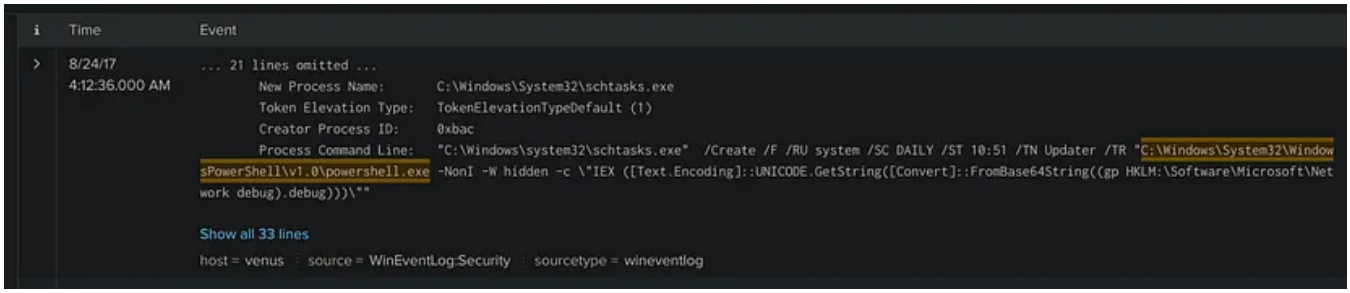
Starting with:

```
index="botsv2" schtasks.exe
```

We know the account domain should be Frothly.

```
index="botsv2" schtasks.exe Account_Domain=FROTHLY
```

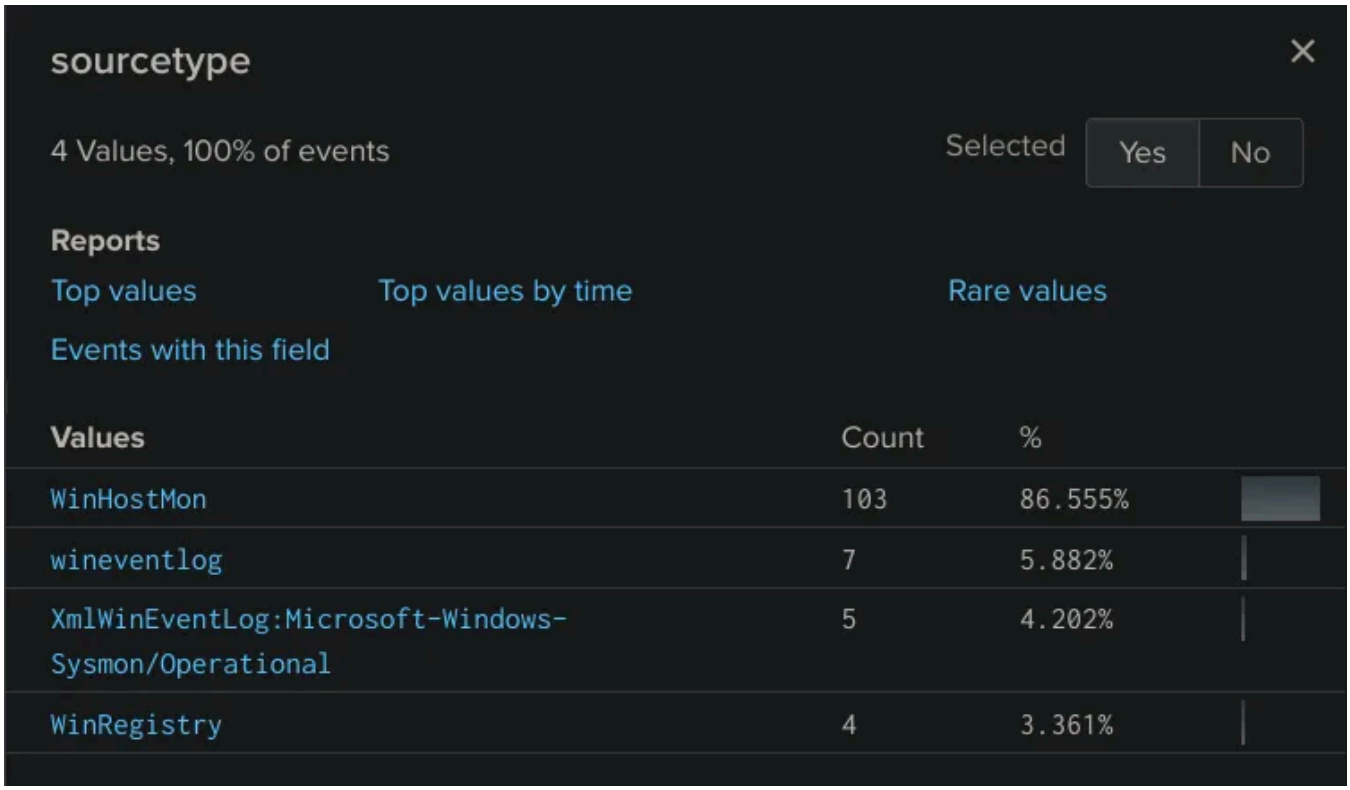
Looking at the results we can see couple of powershell executions that are conspicuous (we know that they utilize powershell but for the sake of the hunt we will discard that information and look for oddness manually.)




Now we know registry key.

Searching for

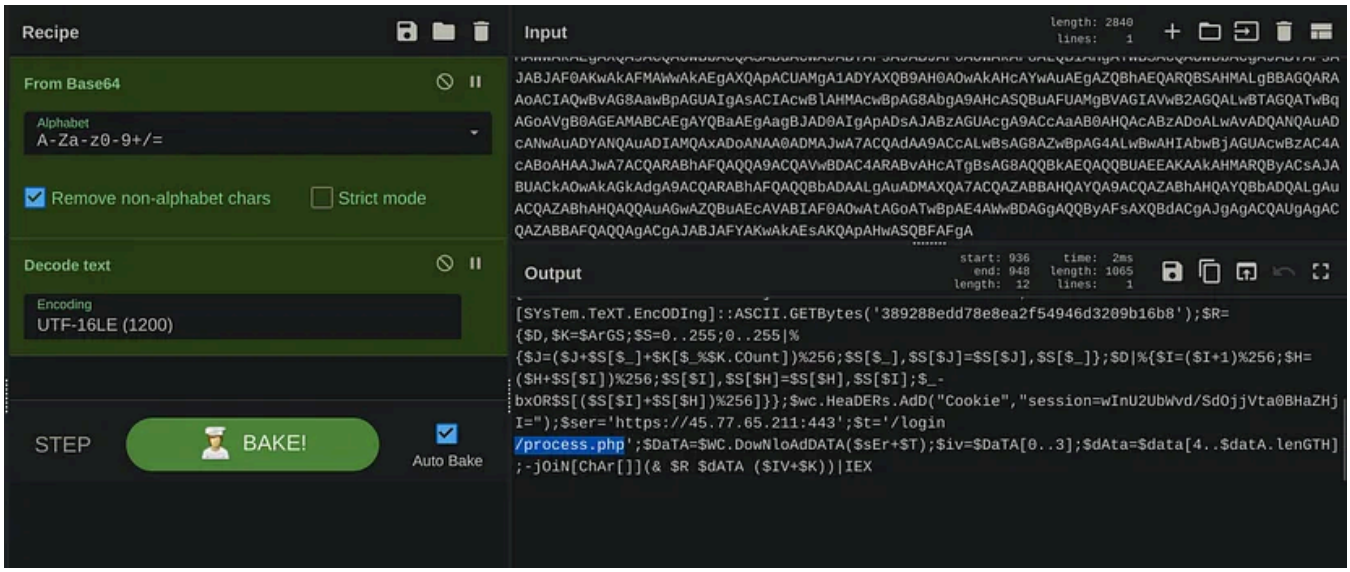
```
index="botsv2" \\Software\\Microsoft\\Network
```




WinRegistry looks worth while.

There are 4 items with base64 encoded text. Decoding them one by one we can deduct the answer which is:

```
process.php
```




We have completed **Splunk Boss of the Soc 2** (**BOTS2**) competition dataset to increase our capabilities using Splunk.

That was it! Thank you for reading. :)



Follow

## Written by Onur Alp Akin

11 Followers · 1 Following

## Responses (1)

What are your thoughts?

Respond

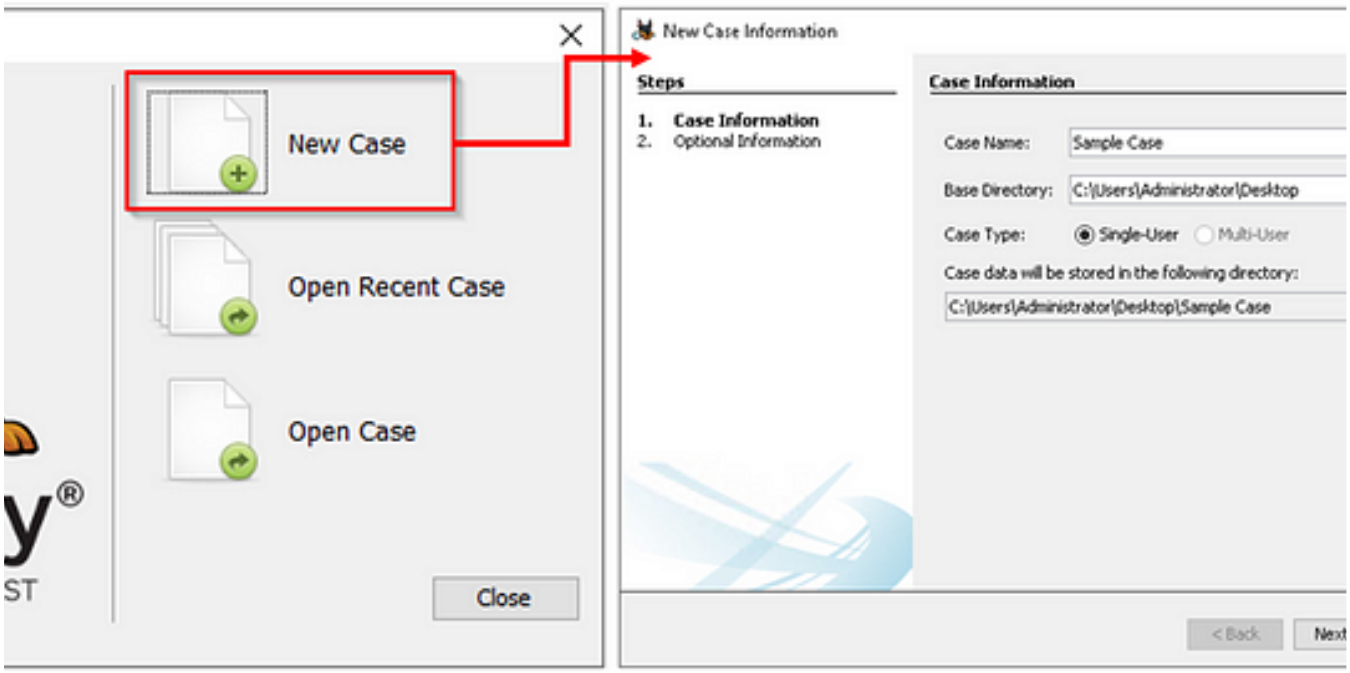
Shashank_B
3 months ago

Question: Using CyberChef to decode the longest looking base64 and searching for @ character, we can find an email.

-- Would there be a better logic as to why we selected the longest o[illegible]?

Reply

## More from Onur Alp Akin




Onur Alp Akin

### TryHackMe: Autopsy Walkthrough

Learn how to use Autopsy to investigate artifacts from a disk image. Use your knowledge to investigate an employee who is being accused of...

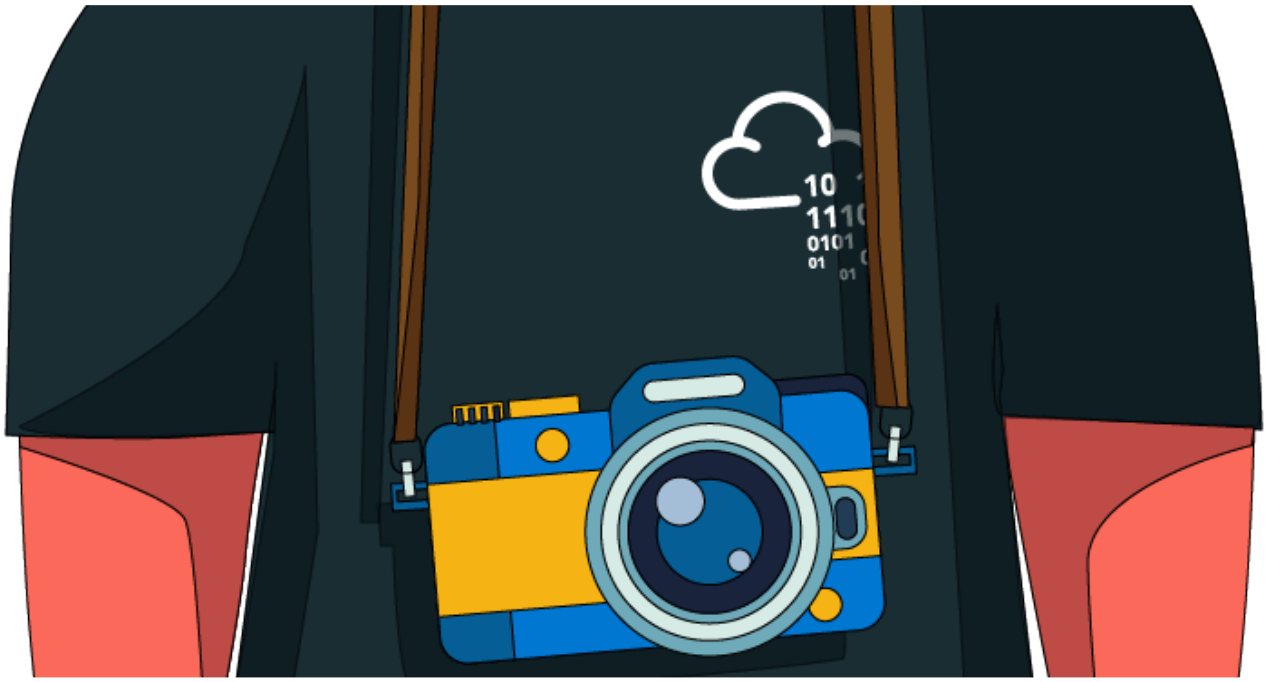
Apr 4, 2023 1

Onur Alp Akin

## TryHackMe: Attacktive Directory Walkthrough

Hi, in this room we will exploit a vulnerable Domain Controller.

Apr 13, 2023

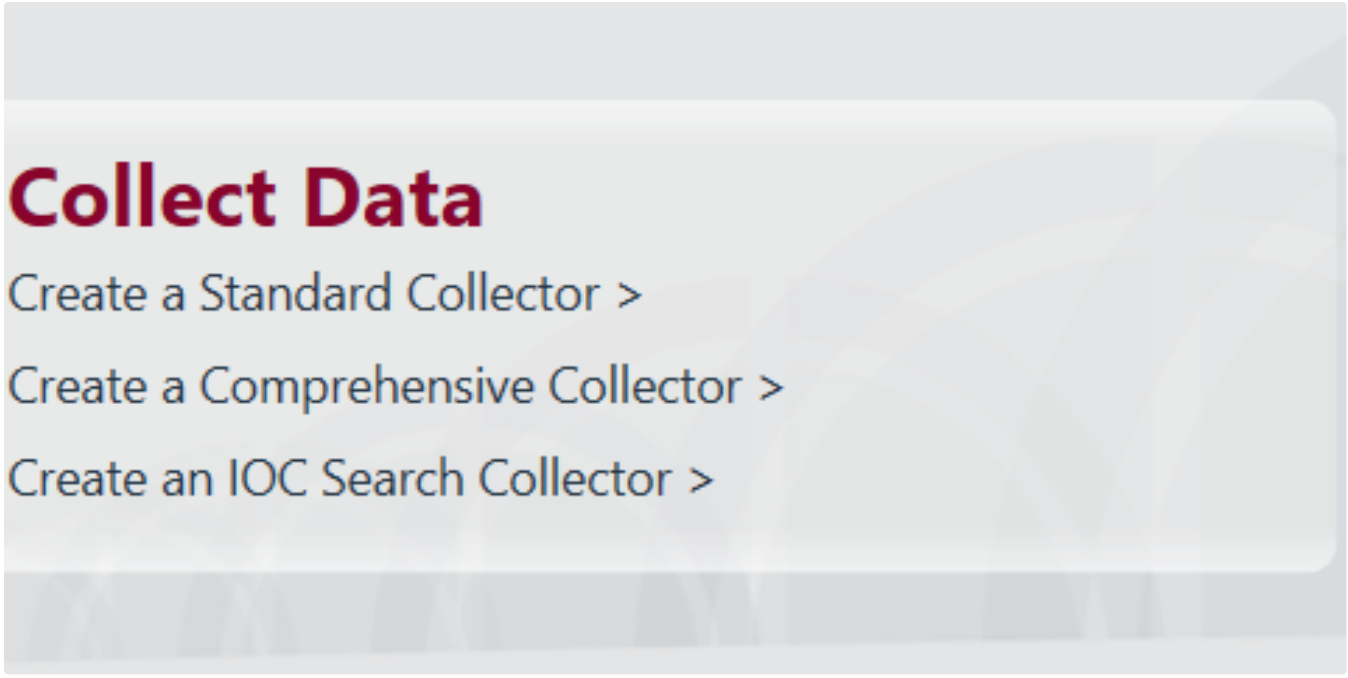


Onur Alp Akin

## TryHackMe: Windows Forensics 2 Walkthrough

Learn about common Windows file systems and forensic artifacts in the file systems

Apr 1, 2023




Onur Alp Akin

## TryHackMe: Redline Walkthrough

Learn how to use Redline to perform memory analysis and to scan for IOCs on an endpoint.

Apr 3, 2023 4 1

See all from Onur Alp Akin

## Recommended from Medium


In T3CH by Axoloth

## TryHackMe | Training Impact on Teams | WriteUp

Discover the impact of training on teams and organisations

Nov 5, 2024 60


Fritzadriano

## Retracted—TryHackMe WriteUp

IInvestigate the case of the missing ransomware. After learning about Wazuh previously, today's task is a bit different.

Sep 4, 2024 50

## Lists

### Staff picks

791 stories · 1538 saves

### Stories to Help You Level-Up at Work

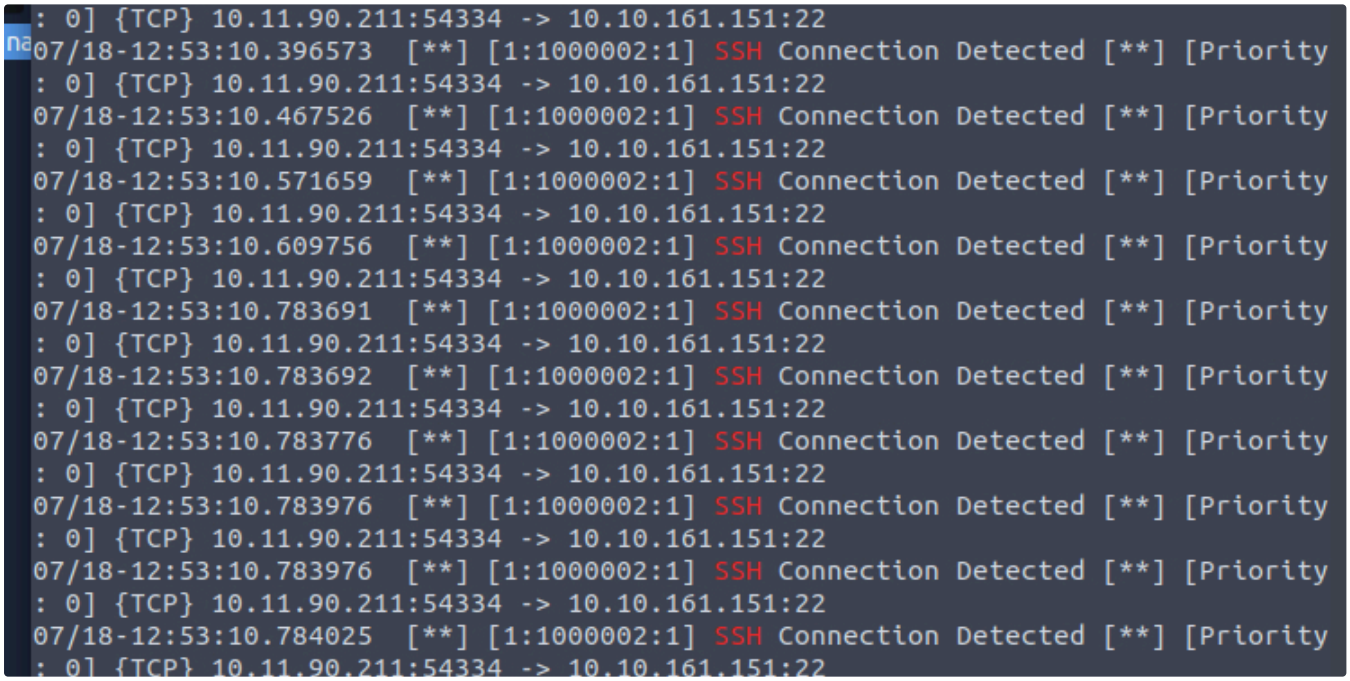
19 stories · 907 saves

### Self-Improvement 101

20 stories · 3175 saves

### Productivity 101

20 stories · 2690 saves




embossdotar

## TryHackMe—IDS Fundamentals—Writeup

Key points: Intrusion Detection System | IDS | Snort | Rules. IDS Fundamentals by awesome TryHackMe! 🎉

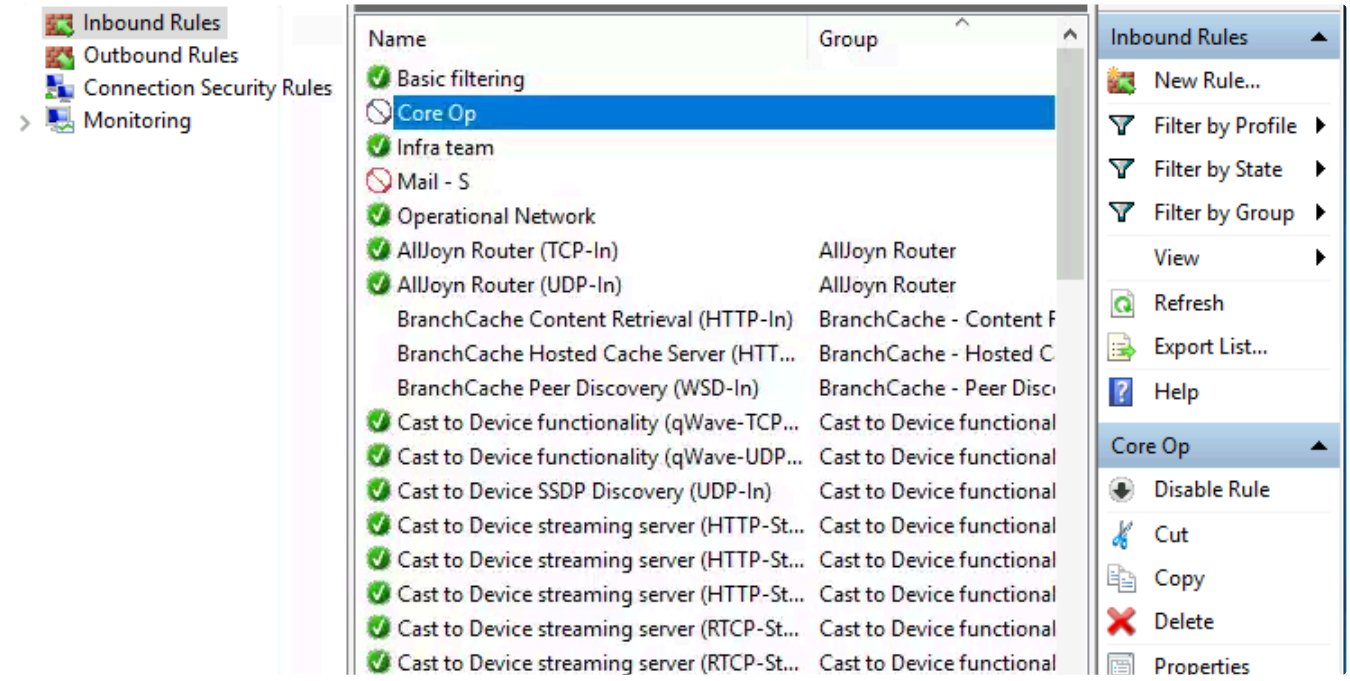
Oct 22, 2024 101

Zach Gillespie

## Conti: Ransomware Investigation with Splunk

In this Tryhackme challenge, an organization exchange server has been compromised with ransomware.

Aug 26, 2024 1

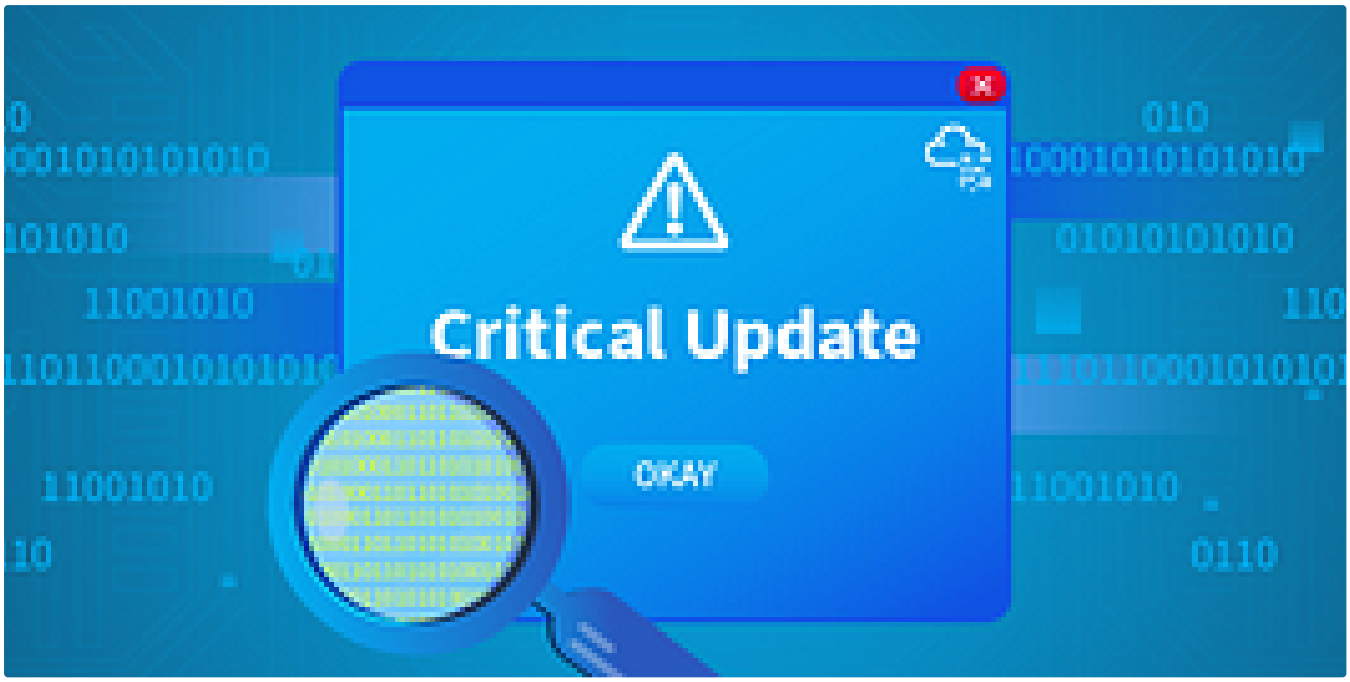



embossdotar

## TryHackMe—Firewall Fundamentals—Writeup

Key points: Firewall | FW | Types | Windows built-in firewall | Linux built-in firewall | Rules. Firewall Fundamentals by awesome...

Oct 22, 2024 35 2




In T3CH by Axoloth

## TryHackMe | Critical | WriteUp

Acquire the basic skills to analyze a memory dump in a practical scenario.

Jul 21, 2024 104

See more recommendations