

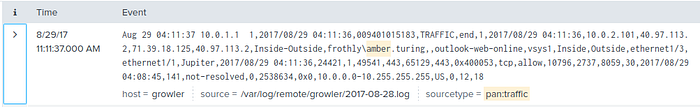
The Try Hack Me Splunk 2 room is a continuation of Splunk: Basics, which introduces Splunk, one of the leading SIEM solutions for collecting, analyzing, and correlating logs. The Splunk 2 room is based on the Boss of the SOC (BOTS) competition hosted by Splunk. In this article I’ll walk-through all questions on each of the tasks.

**100 Series Questions**

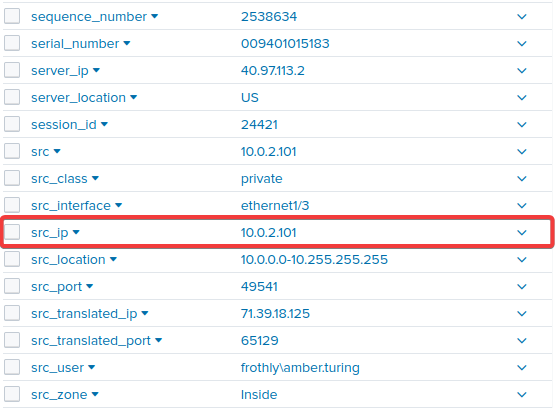
1). Amber Turing was hoping for Frothly to be acquired by a potential competitor which fell through, but visited their website to find contact information for their executive team. What is the website domain that she visited?

To find which website Amber visited, I first need to figure out her IP address. As mentioned in the introduction, “The environment included a Palo Alto Networks (PAN) next-generation firewall to capture traffic and provide web proxy services”. Knowing that web traffic was captured by the PAN firewall & proxy, I can search for her IP there:

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



Clicking the drop down next to the event displays the field names and values:

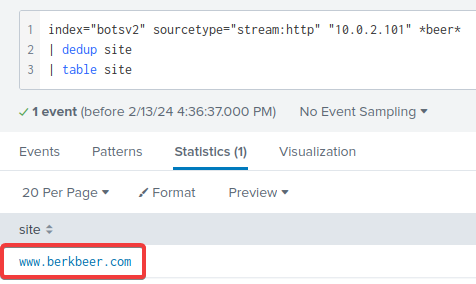


Now that I know Amber’s IP address is 10.0.2.101, I can add that to my search. Since I only want to see web traffic, I will edit my source type to HTTP stream as well:

index="botsv2" sourcetype="stream:http" "10.0.2.101"

While this command will display all websites that Amber visited, I only care about the competitor’s website she visited. Knowing they work in the Beer industry, I can add that to my search using wildcard operators. I do not want duplicates returned, so I’ll make sure to exclude those (ded up), while displaying my results in a table view (table):

index="botsv2" sourcetype="stream:http" "10.0.2.101" \*beer\*  
| dedup site  
| table site



2). Amber found the executive contact information and sent him an email. What image file displayed the executive’s contact information? Answer example: /path/image.ext

Now that I know the competitor’s website, I can add it to my search:

index="botsv2" sourcetype="stream:http" "10.0.2.101" [www.berkbeer.com](http://www.berkbeer.com)

There are only 12 events in this search and looking through the results I find that this URI path to the ‘ceoburk.png’ is likely what I am looking for:

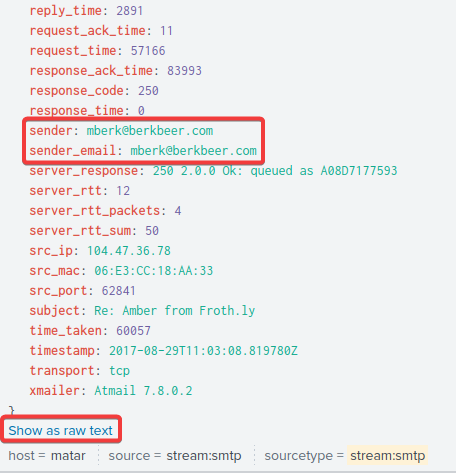


3). What is the CEO’s name? Provide the first and last name.

Changing the source type from HTTP to SMTP will display all captured email communications. I will remove Amber’s IP address and just search for any mail traffic going to or from berkbeer.com:

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

This returns 6 events and allows me to find an being sent from the CEO, [mberk@berkbeer.com](mailto:mberk@berkbeer.com):



Since SMTP is sent in plaintext over the network, I’ll be able to see the contents (body) of the email by clicking the ‘Show as raw text’:

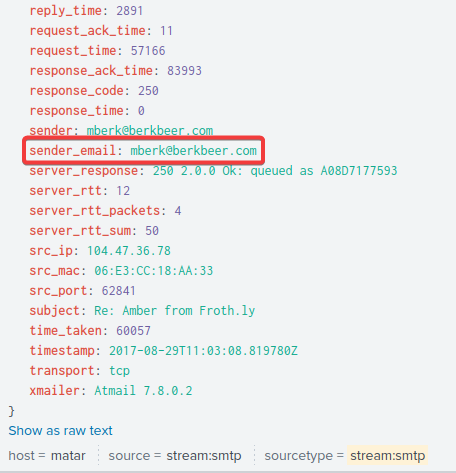


Reading through the contents of the email, I’m able to find the full name of the CEO, Martin Berk, likely as an email signature.

4). What is the CEO's email address?

As shown in the previous question the SMTP events expose Martin Berk’s email address:

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



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

Again, looking at the SMTP traffic, the first (most recent) event shows an email being sent from Amber:

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

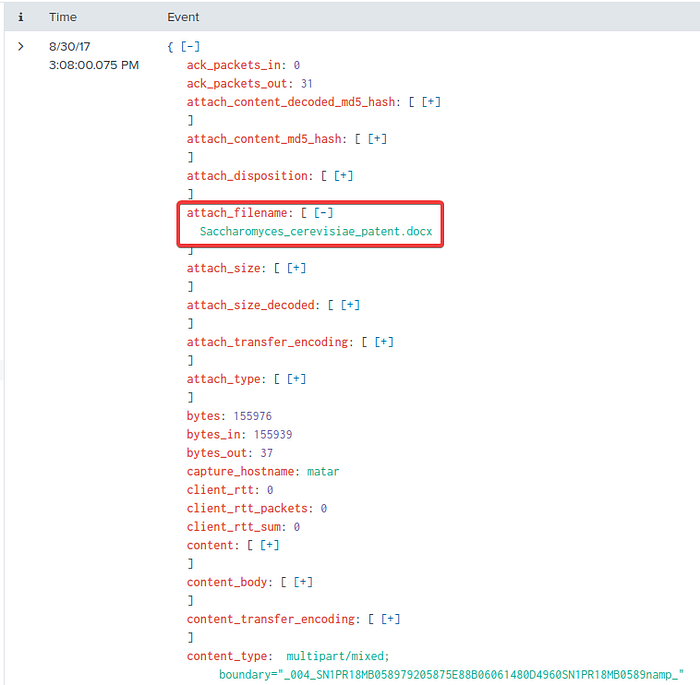


Looking into the email contents by clicking on ‘Show as raw text’, I can see that it is an email being sent from aturning@froth.ly, to [hbernhard@berkbeer.com](mailto:hbernhard@berkbeer.com):



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

Looking into the same event as the previous question, there is a field called ‘attach\_filename’ and when expanded, exposes the file that was attached to Amber’s email to Bernhard:

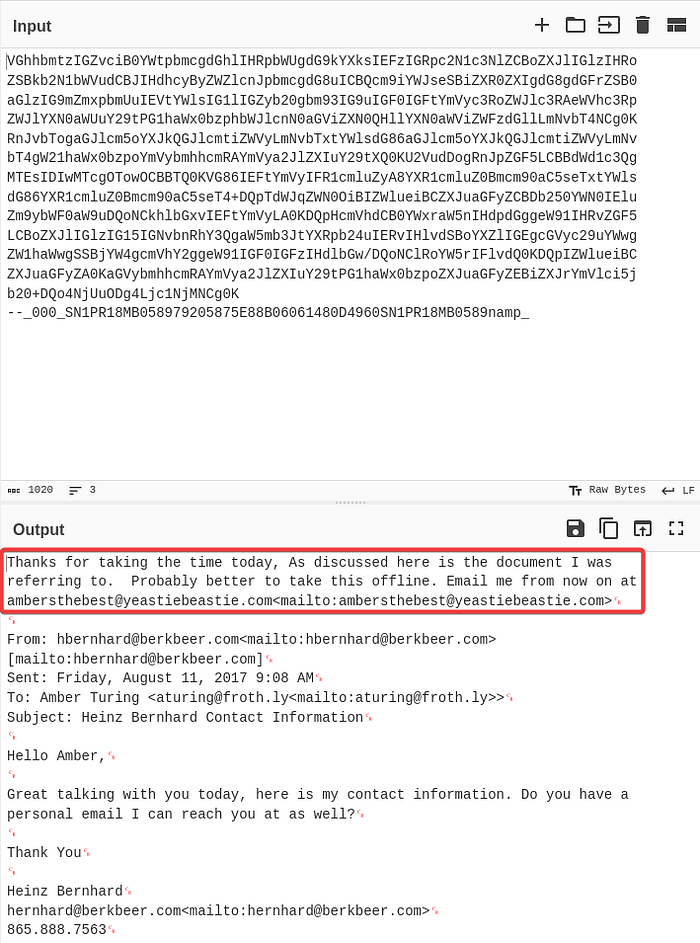


7). What is Amber's personal email address?

Referencing the same event as shown in the previous 2 questions, I can see that a part of the content of the email appears the be base64 encoded:



Decoding this text, using a tool like CyberChef, results in the following:

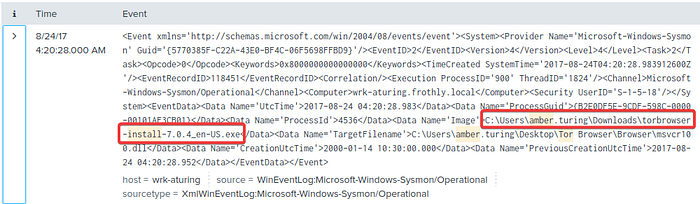


**200 Series Questions**

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

I want to search the botsv2 index for the initial installation of the TOR browser on Amber’s computer. To do so, I’ll include the keywords ‘amber’, ‘tor’, and ‘install’, sorting my results with the oldest at the top:

index="botsv2" "tor" "amber" "install"  
| sort -\_time desc

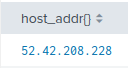


In the first event, I can see that the user amber.turing downloaded version 7.0.4 of the TOR browser to her downloads folder.

2). What is the public IPv4 address of the server running [www.brewertalk.com](http://www.brewertalk.com)?

To find the public IPv4 address of the server running www.brewertalk.com, I’ll need to look into the DNS logs, since the HTTP logs will only display private IP addresses. The specific field I’m looking for here is ‘host\_addr’:

index="botsv2" source="stream:dns" "www.brewertalk.com"  
| table host\_addr{}  
| dedup host\_addr{}



This returns a single result of 52.42.208.228. I can also find the private IPv4 address of the by searching for the destination IP address field:

index="botsv2" source="stream:http" "www.brewertalk.com"  
| table dest\_ip  
| dedup dest\_ip

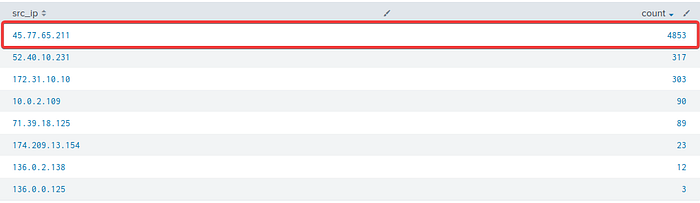


Here I find a likely private IP address of 172.31.4.249, as well as the public IP address I found earlier.

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

A web vulnerability scan will cause a lot of traffic to be sent from a single IP address to the web server. I can search the HTTP logs for a count of source IP addresses hitting www.brewertalk.com:

index="botsv2" source="stream:http" "www.brewertalk.com"  
| stats count by src\_ip



45.77.65.211 is the source IP address that has the most traffic hitting the server by a significant amount so it’s safe to assume that this is the IP address of the system that ran the vulnerability scan.

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

Since I know the attackers IP address, 45.77.65.211, and the IP address of the web server, 172.31.4.249, I can search the logs for the most common occurrence of the URI path that’s being hit:

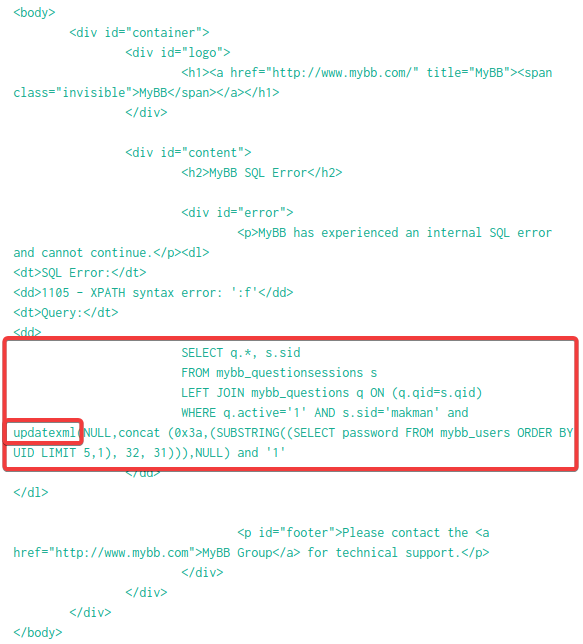
index="botsv2" src\_ip="45.77.65.211" dest\_ip="172.31.4.249"  
| stats count by uri\_path



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

I added the URI path I found from the previous question to my search, as well as the keyword ‘select’, since that’s a required statement in SQL commands:

index="botsv2" src\_ip="45.77.65.211" dest\_ip="172.31.4.249" uri\_path="/member.php" select



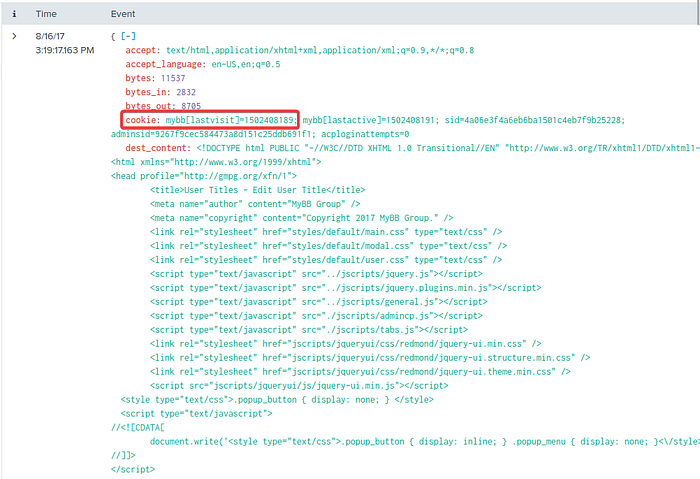


What was the value of the cookie that Kevin’s browser transmitted to the malicious URL as part of an XSS attack? Answer guidance: All digits. Not the cookie name or symbols like an

I know cross-site scripting is an attack against a web application, so I can set the source type as HTTP. Additionally, I’ll add ‘kevin’ to my search since it’s asking for Kevin’s browser cookie. Finally, I’ll add ‘<script>’ to my search since that’s a common tag used in XXS attacks:

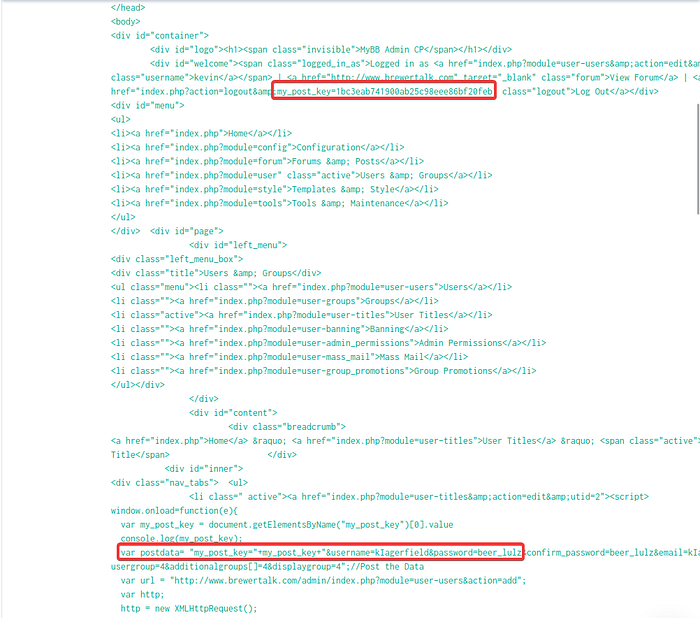
index="botsv2" sourcetype="stream:http" kevin "<script>"

This returns only 1 event, so I’ll grab that cookie value since it’s likely the one the attacker stole from Kevin’s browser:



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

In the same event from the previous question, I can see in the dest\_content field that Kevin’s CRSF token gets stored in a variable called ‘my\_post\_key’, which is then used to create a new account with the username ‘kIagerfield’ and a password of ‘beer\_lulz’:

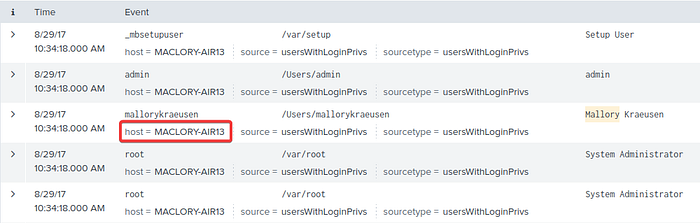


**300 Series Questions**

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?

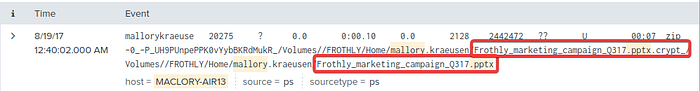
I first need to find the name of Mallory’s system

index="botsv2" Mallory



I’ll now add her MacBook’s host name to my search, including a wildcard for any of the three PowerPoint file extensions:

index="botsv2" mallory host="MACLORY-AIR13" \*.ppt\*

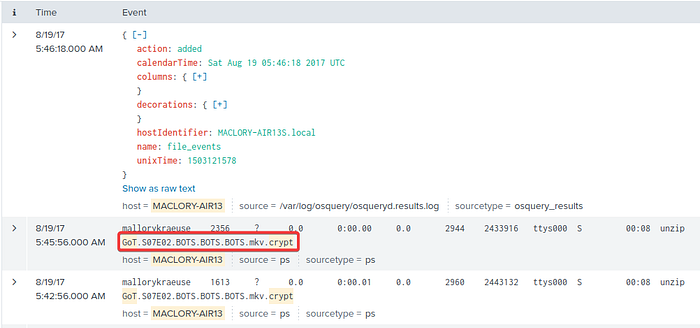


This allows me to find the name of the PowerPoint file before and after it got encrypted.

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

Knowing the the attacker’s ransomware encrypts files with the file extension ‘crypt’, I’ll add this to my search. Additionally, I’ll add any rendition of ‘Game of Thrones’ that I know of:

index="botsv2" host="maclory-air13" (got OR game OR thrones) \*crypt\*



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.

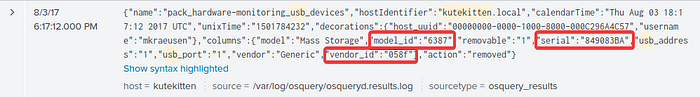
I’ll add the keyword ‘kutkitten’ to my search along with ‘usb’:

index="botsv2" kutekitten usb

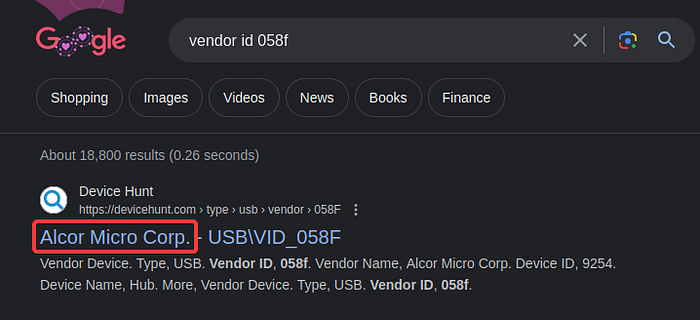


This returns 40 events, and looking through them I found an event where the name variable was set to ‘pack\_hardware-monitoring\_usb\_devices’ which sounds like what I’m looking for.

I clicked on ‘Show as raw text’ to get a better look and found the USB’s model ID, serial number, and vendor ID:

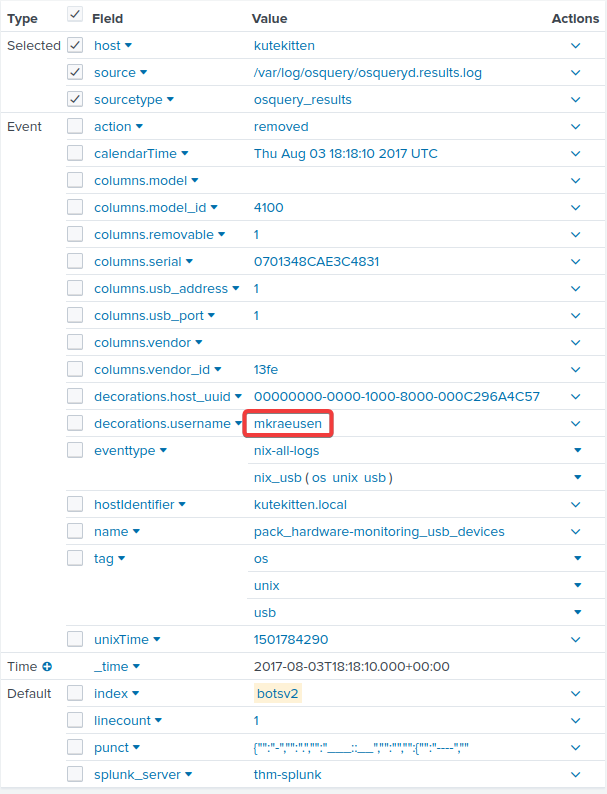


Simply Googling the vendor ID allows me to find the vendor name from a website called ‘Device Hunt’:



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

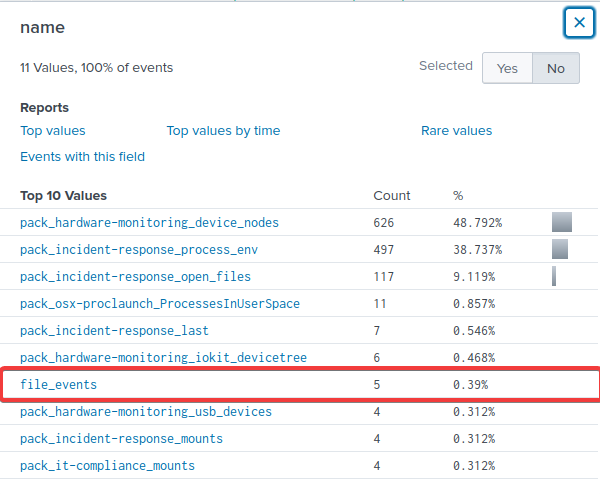
Looking at the fields from the event found in my last search, I find Mallory’s username one her personal computer:



Adding her username to my search, returns over 1,000 events:

index="botsv2" kutekitten mkraeusen

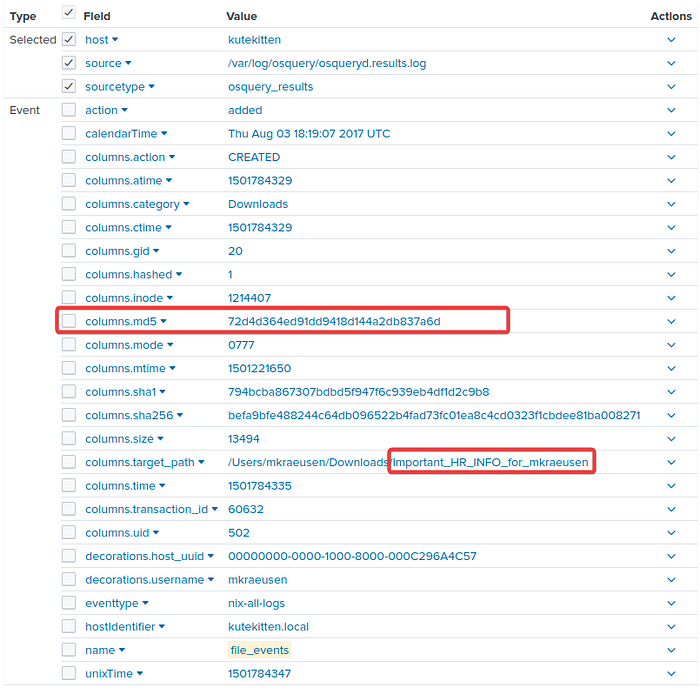
However, looking at the ‘name’ field, like I did to find the USB vendor, I find something that sticks out:



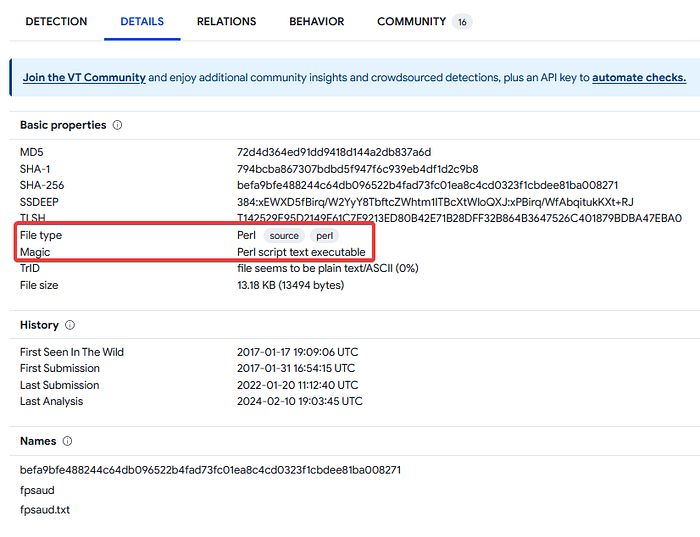
file\_events sounds like it could be related to the encryption of files from the ransomware, so I’ll add this to my search:

index="botsv2" kutekitten mkraeusen name=file\_events

This returns just 5 events, and looking into one of them reveals the likely ransomware file that Mallory downloaded, important\_HR\_INFO\_for\_mkraeusen, along with its MD5 hash:

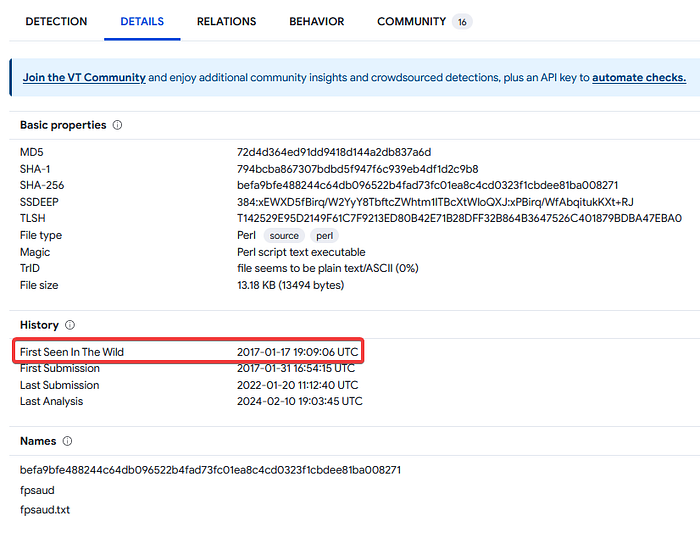


Searching for this MD5 hash on virus total reveals that it is a PERL executable:



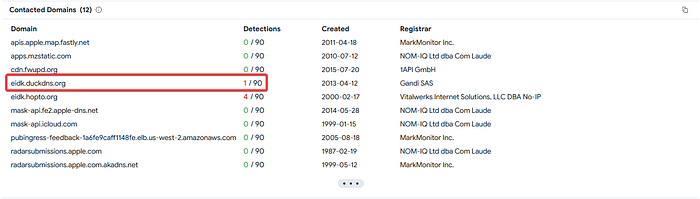
5). When was this malware first seen in the wild? Answer Guidance: YYYY-MM-DD

Referencing the details tab on VirusTotal, I find when this ransomware was first seen in the wild:

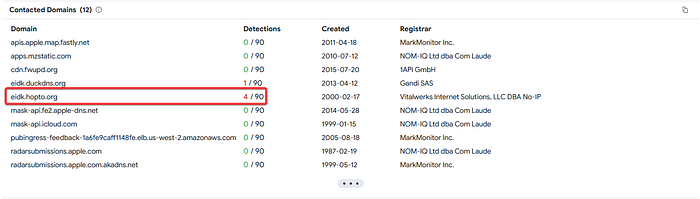


6). The malware infecting kutekitten uses dynamic DNS destinations to communicate with two C&C servers shortly after installation. What is the fully-qualified domain name (FQDN) of the first (alphabetically) of these destinations?

Switching to the Relations tab on VirusTotal, I find the 2 domain names of the C2 server that the ransomware connects back to:



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



**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?

Spearphishing is often done via email so I’ll set my source type to SMTP, searching for the file extension .zip:

index="botsv2" sourcetype="stream:smtp" .zip

This returns 6 events, and looking into the 2nd I find the file attachment file name that was sent:



2). What is the password to open the zip file?

It’s likely that the body of the email from the event I looked at in the previous question contains the password to the .zip file, since the threat actor wants the receiver to download and open the file. Looking into the email contents by clicking ‘Show as raw text’, I find the plaintext password:

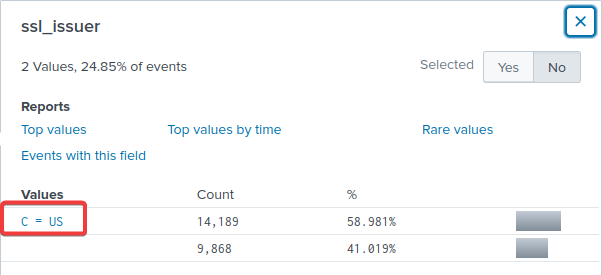


3). The Taedonggang APT group encrypts most of their traffic with SSL. What is the “SSL Issuer” that they use for the majority of their traffic? Answer guidance: Copy the field exactly, including spaces.

Knowing from previous questions that the attacker’s IP address is 45.77.65.211, I’ll add this to my search, in addition to ‘SSL’:

index="botsv2" dest\_ip="45.77.65.211" SSL

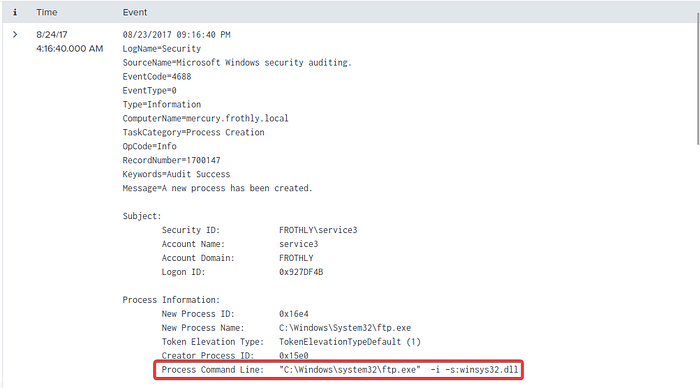
This returns over 90,000 events, but the ssl\_issuer field is consistent across all events, disregarding the blank values:



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

I first searched for any event containing the file ‘winsys32.dll’:

index="botsv2" winsys32.dll



This shows that FTP was used, so I’ll add include that in my search for source type. The ‘get’ and ‘retr’ commands are often used in FTP to download files from the FTP server, so I’ll include that as well:

index="botsv2" sourcetype="stream:ftp" ("get" OR "retr")

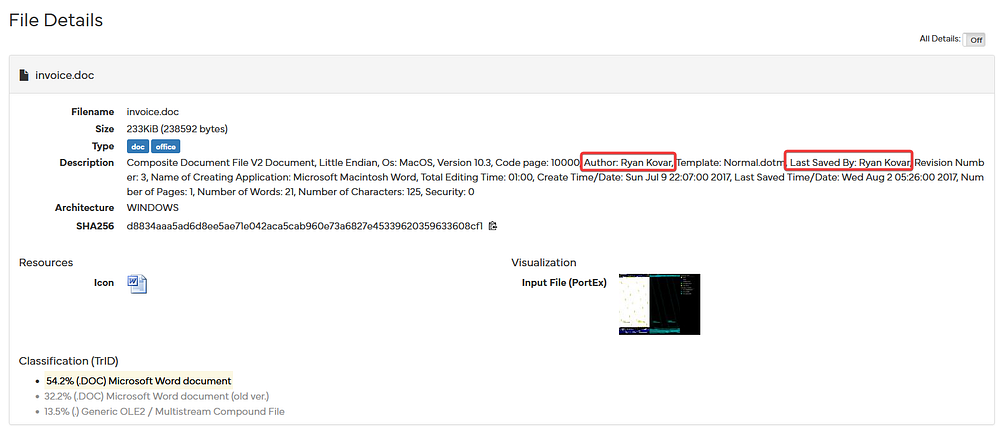
This returns just 14 events, and looking at the filename field shows me the likely answer (since the question states the file is unusual for an American company):



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

Clicking on the following link provided in the task, allowed me to find the answer to this question:

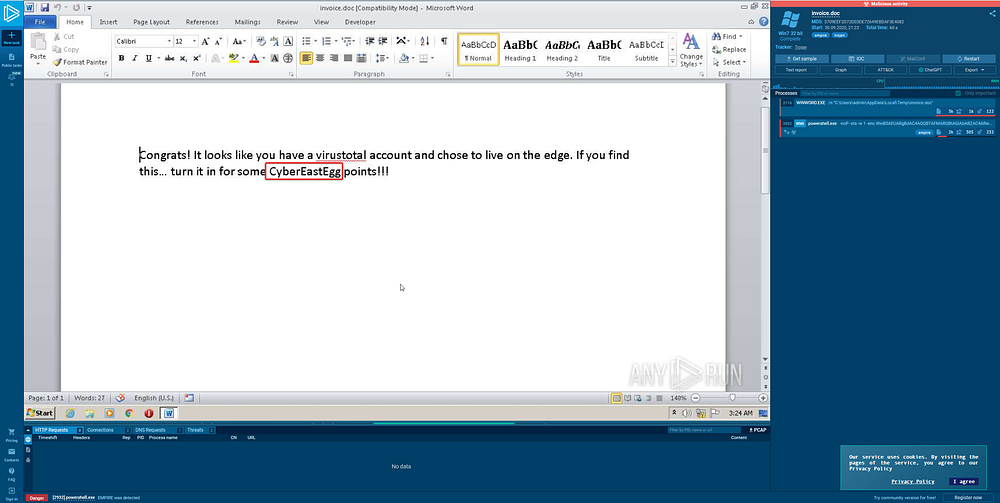
<https://www.hybrid-analysis.com/sample/d8834aaa5ad6d8ee5ae71e042aca5cab960e73a6827e45339620359633608cf1/598155a67ca3e1449f281ac4>



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

Clicking on the following link provided in the task, allowed me to find the answer to this question:

<https://app.any.run/tasks/15d17cd6-0eb6-4f52-968d-0f897fd6c3b3/>

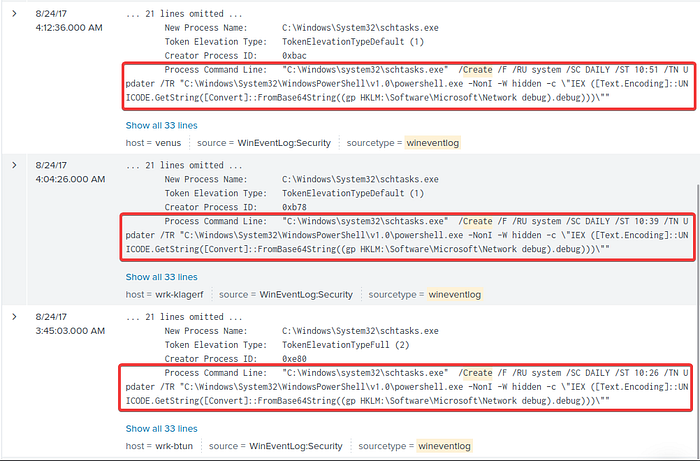


7). To maintain persistence in the Frothly network, Taedonggang APT configured several Scheduled Tasks to beacon back to their C2 server. What single webpage is most contacted by these Scheduled Tasks? Answer example: index.php or images.html

First I’ll search for newly created scheduled tasks. Since all newly created scheduled tasks are logged in the Windows Event Log, I’ll add that as my source type too:

index="botsv2" schtasks.exe sourcetype=wineventlog create

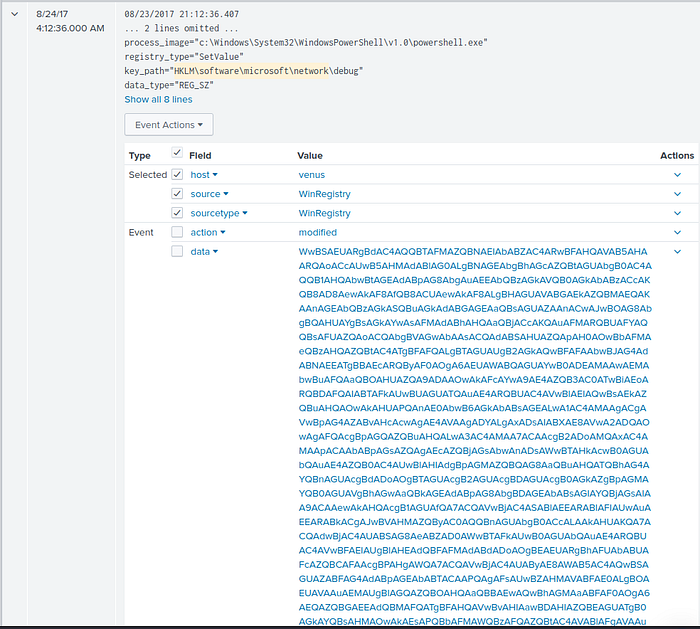
This returns 9 events, the first 6 being tasks to configure automatic updates and monitoring so I’ll ignore those. The last 3 events appear to be scheduled tasks created by the attacker:



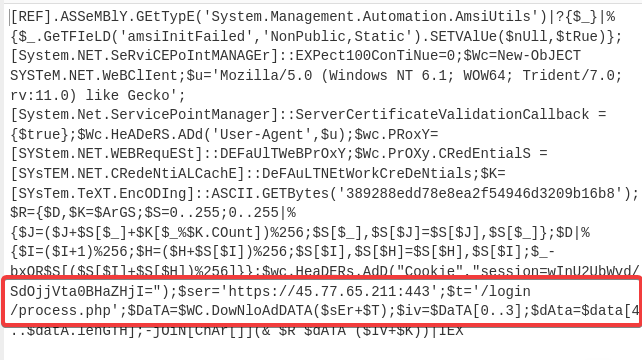
These PowerShell commands appear to be editing the ‘HKLM:\Software\Microsoft\Network’ registry hive to beacon back to the attacker’s C2 server. I’ll now search for this registry hive key value in my search to see if I can find the domain/IP address and URI the C2 server connects back to:

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

This returns just 4 events, and the data field of each appears to be base64 encoded:



Decoding the value in the data field of the 2nd event, then converting it to UTF 16 Little Endian text returns the following:



This shows that the C2 server is connection back to https://45.77.65.211:443/login/process.php.