

COMP3010 CW2

Contents

Introduction:.....	2
SOC Roles and Incident Handling Reflection:.....	2
SOC:.....	2
Prevention:.....	2
Detection:.....	2
Response:.....	3
Recovery:.....	3
Installation and Data Preparation:	3
Splunk Installation:.....	3
BOTSv3 Dataset Preparation:.....	8
Guided Questions:	13
Question 1:	14
Question 2:	14
Question 3:	15
Question 4:	16
Question 5:	17
Question 6:	17
Question 7:	18
Question 8:	19
Conclusion:.....	20
References:	21

Introduction:

In this report, I will be analysing the BOTSv3 dataset within Splunk. The BOTSv3 dataset is a public dataset made up of 2,083,056 events which simulates an incident to allow cyber security professionals to practice event analysis in a simulated environment. By utilising the 3 tiers of SOC – triage, investigation and hunting – I will answer a series of questions to determine how the incident occurred, including information about the user(s) and device(s) involved. This report will focus on 8 of those questions, explaining their relevance to SOC, as well as how to install Splunk and how to prepare the BOTSv3 dataset for analysis.

Screenshots can be found here: <https://github.com/Mark7567/COMP3010>

Video walkthrough: <https://www.youtube.com/watch?v=ijeZ6ZJcl>

SOC Roles and Incident Handling Reflection:

SOC:

The Security Operations Centre (SOC) consists of 3 tiers, each having unique roles and responsibilities surrounding cyber security prevention and analysis. Tier 1 - triage - involves monitoring dashboards and events to determine if an incident has occurred. Since attacks cannot be 100% prevented, tier 1 acts as a way to catch incidents early, as well as filter out false positives which prevention methods may have caught.

Once an event has been flagged as an incident, tier 2 - investigation - takes place. This involves checking logs and confirming that an incident has occurred and containing them to prevent any more damage from occurring. Tier 2 also involves determining how the incidents occurred by performing an in-depth analysis.

Lastly, tier 3 - hunting - involves performing an advanced analysis on the incident and improving overall security following it. The BOTSv3 dataset focusses predominantly on tier 2 of SOC, since the dataset contains incidents which have already occurred, with the purpose of the dataset being to confirm this and investigate how they happened.

Prevention:

Prevention is an incident handling methodology which involves mitigating the risk of an incident occurring rather than directly stopping it, since it is not possible to 100% prevent an attack. Rules and proper system configurations are implemented to make a system as secure as it can be in an attempt to reduce the risk of successful attacks. Prevention relates to the BOTSv3 exercise since the events in the dataset can help to determine normal user behaviour which may make any outliers more obvious as potentially malicious. Furthermore, it links with tier 1 of SOC as there are links with monitoring dashboards and analysing behaviour.

Detection:

Detection is an incident handling methodology that involves identifying incidents which could be considered suspicious and may indicate an incoming incident. Typical detection methods include analysing network traffic and timestamping user actions or system events to find concerning behaviour. The BOTSv3 dataset involves using filters and searching within Splunk to

analyse millions of events and identify information about them, such as which users were involved and which operating system their device was running. This links to tiers 1 and 2 of SOC since detection involves flagging potentially malicious events and analysing them to determine their causes.

Response:

Response is an incident handling methodology which involves containing an incident to prevent it from causing any more damage to a system, as well as investigating how the incident occurred. The BOTSv3 dataset contains a variety of incidents which need to be confirmed as valid, as well as investigating into how they occurred by gathering information about the events using Splunk filters. Since it is a premade dataset, Splunk cannot contain the incidents in any way, but it can be used to investigate them. Response has strong links with tier 2 of SOC as the purpose of this methodology is to investigate how the incident occurred, which is also the main idea behind SOC tier 2.

Recovery:

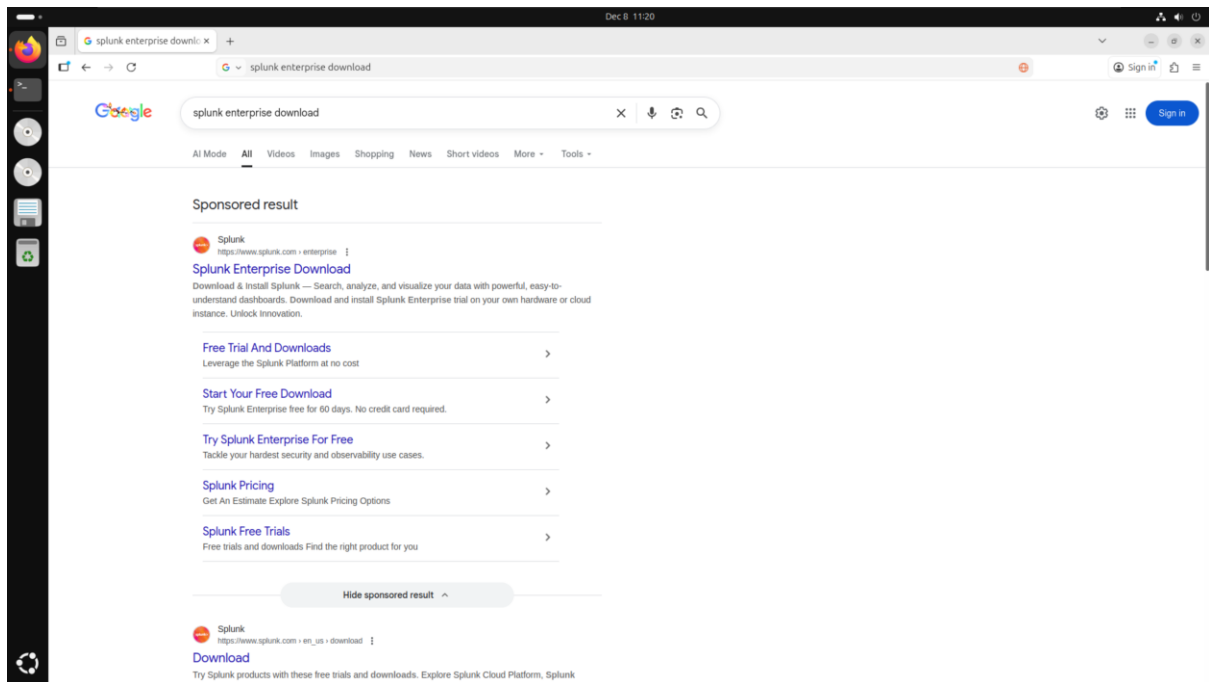
Recovery is an incident handling methodology which involves restoring damaged systems after an incident has been resolved. It also includes improving security to reduce the risk of a similar incident happening in future. While there is no active recovery within the BOTSv3 dataset, it helps with this methodology as the dataset is premade to contain events for analysis which can then be reviewed to prevent further incidents.

Installation and Data Preparation:

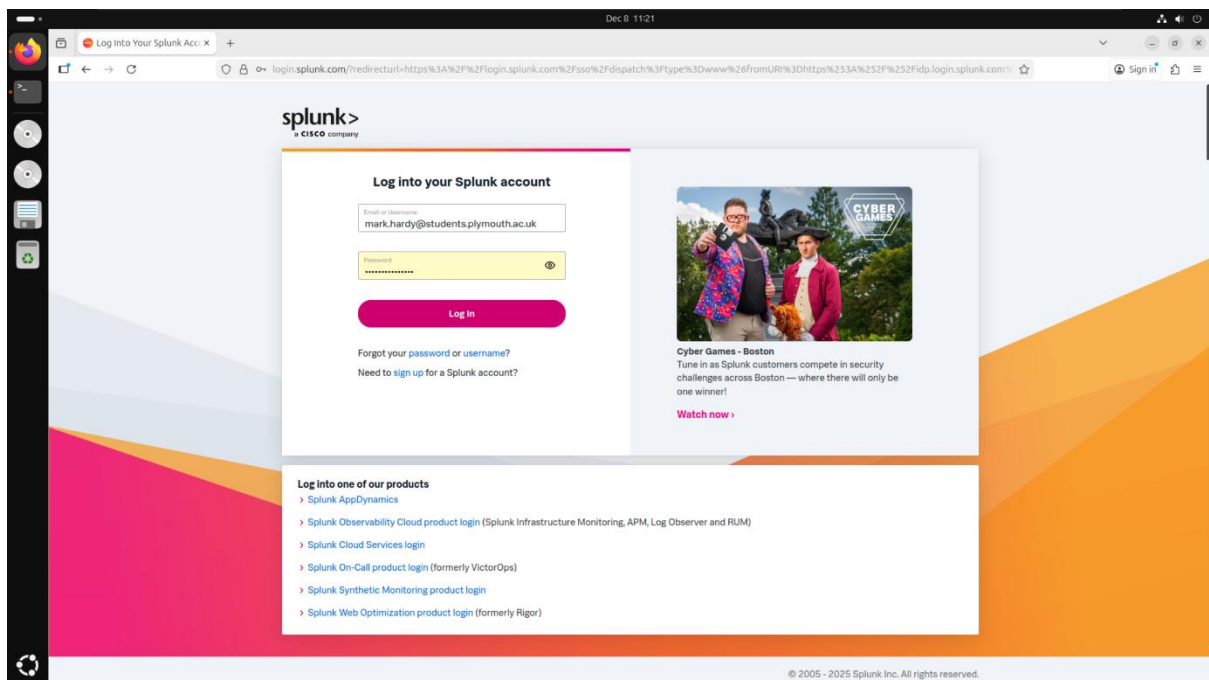
This section will explain how Splunk was installed within an Ubuntu Linux virtual machine hosted by VMWare, and how the BOTSv3 dataset was loaded. As I performed the installation before taking screenshots for this section, the screenshots display later timestamps compared to those for the guided questions. To effectively demonstrate the installation of Splunk, I reinstalled it on the same virtual machine ensuring the configuration of the machine remained the same throughout this report. The reason for using a Linux virtual machine is to mimic real world SOC environments which would typically use Linux-based systems. By installing Splunk locally on my device, this allowed me to have full control over the configuration, and which data was ingested, keeping the integrity and accuracy of the analysis.

Splunk Installation:

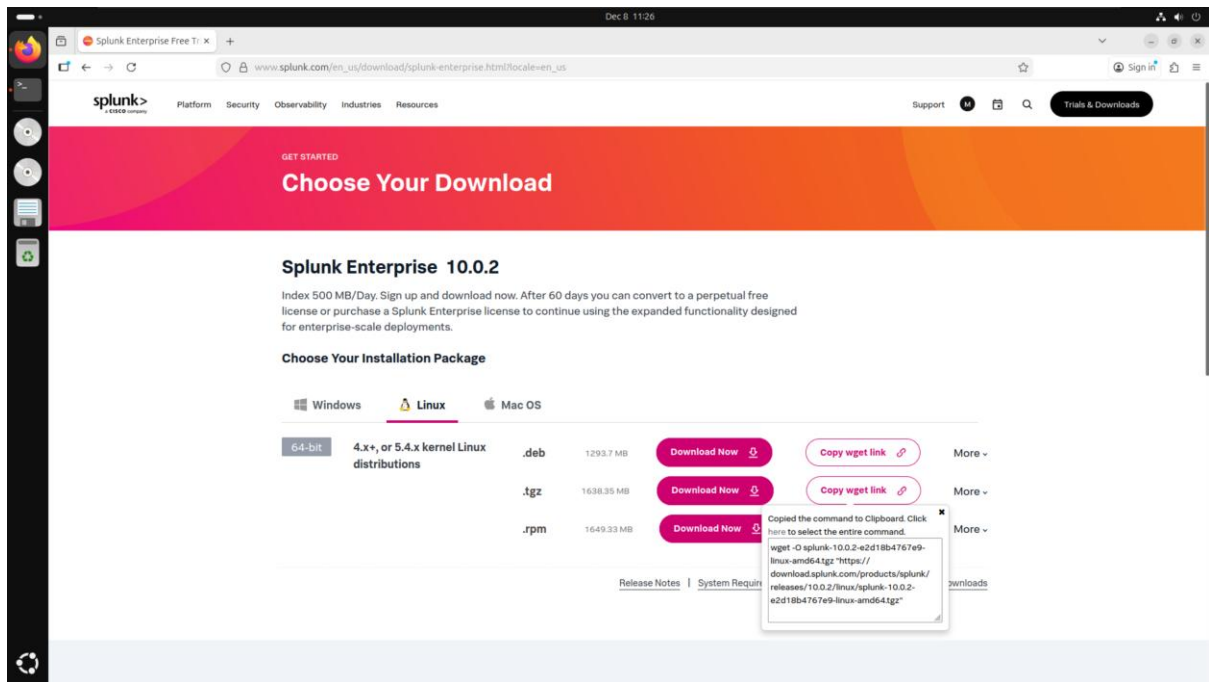
I opened Firefox inside an Ubuntu Linux virtual machine and searched for “splunk enterprise download”, clicking on the first link in the search results.



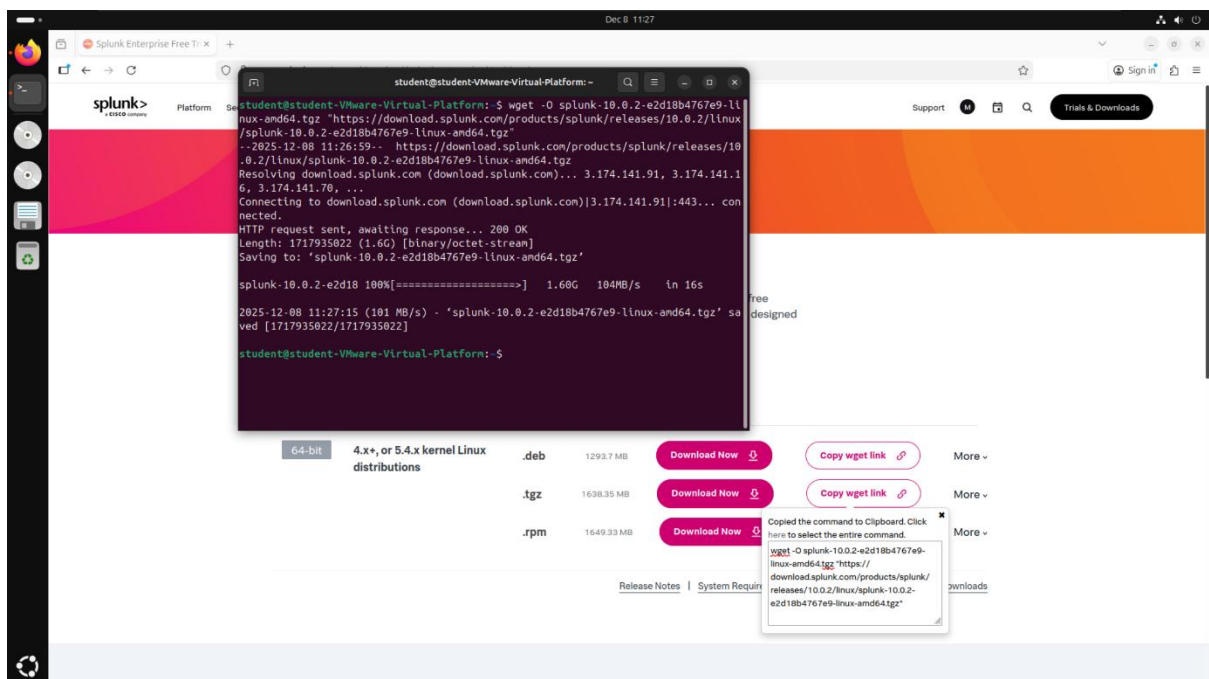
This link took me to the official Splunk website, where I logged into an account which I had already created.



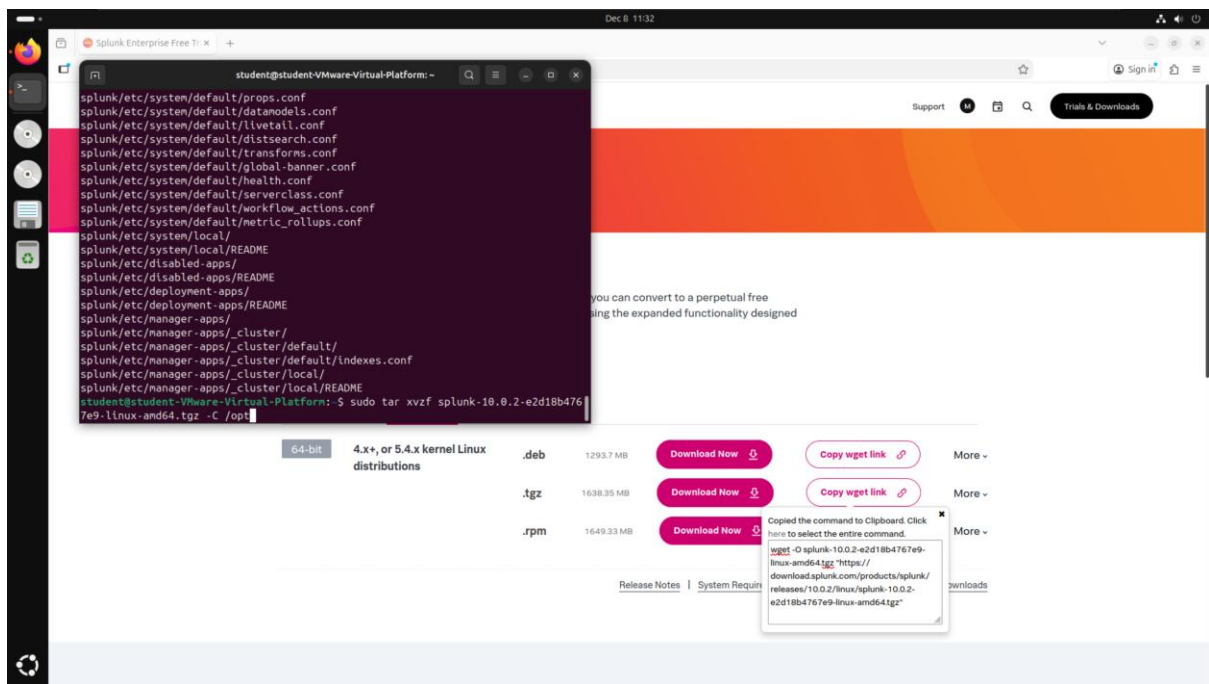
I selected "Copy wget link" for the .tgz version of Splunk, to ensure that all packages were downloaded alongside the main Splunk software. This meant I did not have to individually download all relevant packages to complete this coursework.



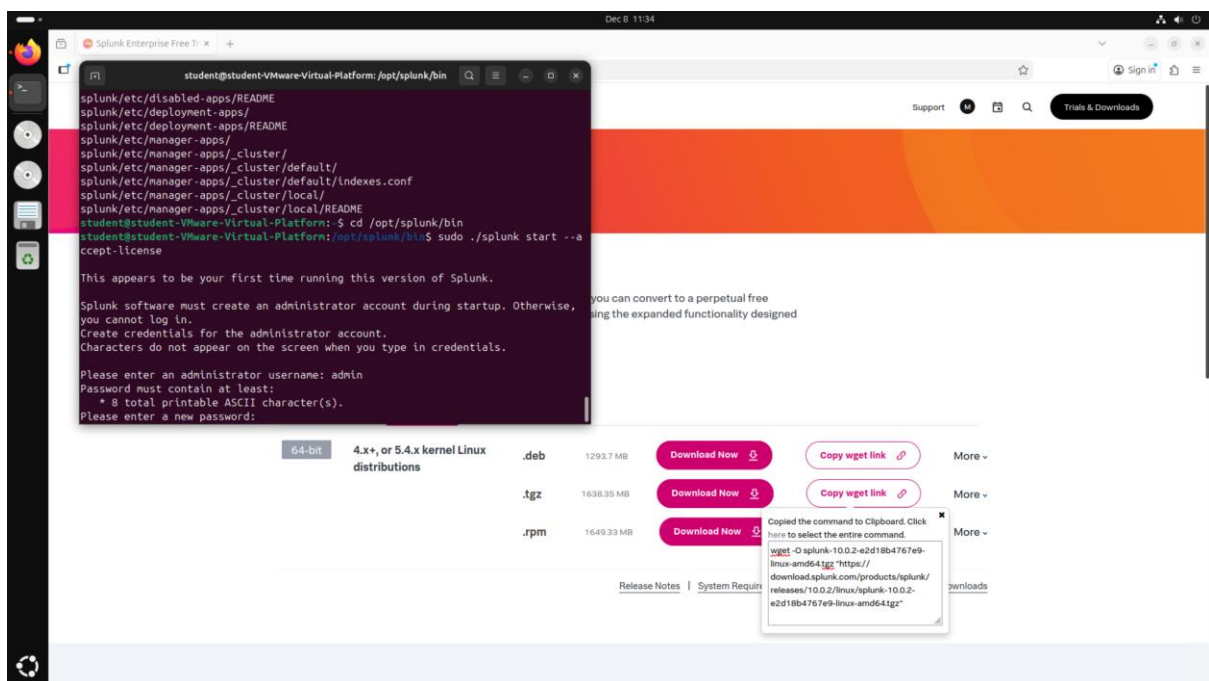
I downloaded Splunk directly from the Linux terminal by pasting the wget link directly into the terminal and running it. I received a confirmation message in the terminal to state that Splunk had been successfully downloaded.



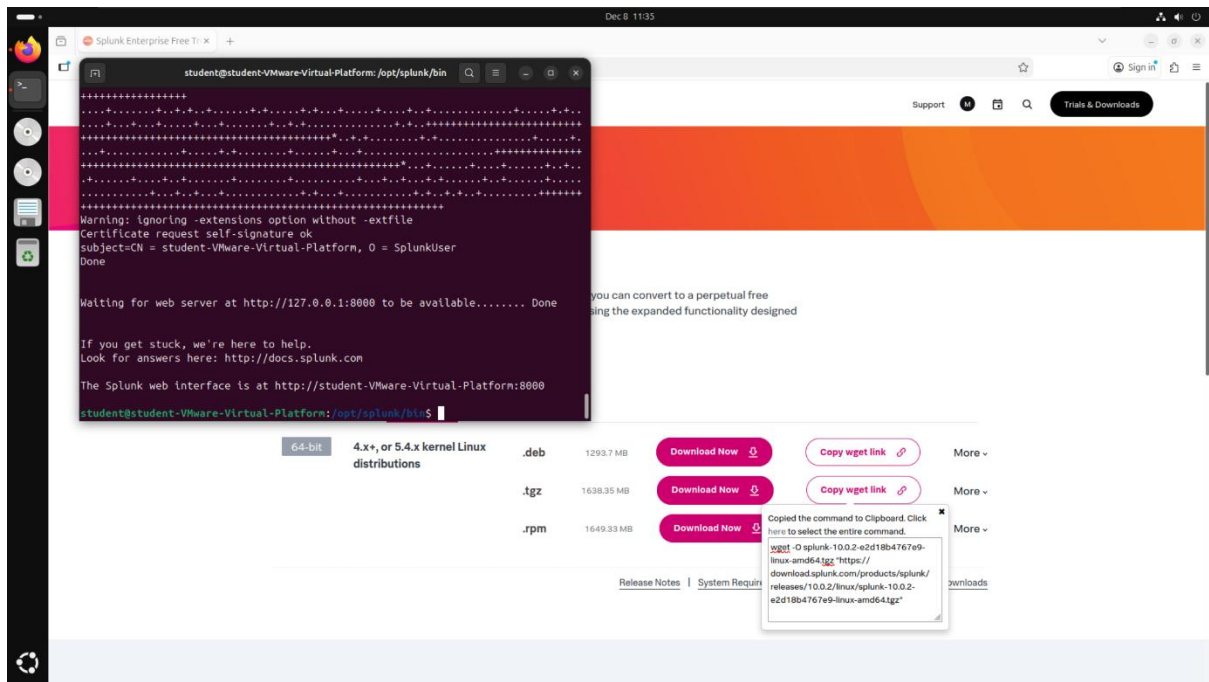
I ran the command “sudo tar xvf splunk-10.0.2-e2d18b4767e9-linux-amd64.tgz -C /opt” which installed Splunk and all its dependencies into the /opt folder.



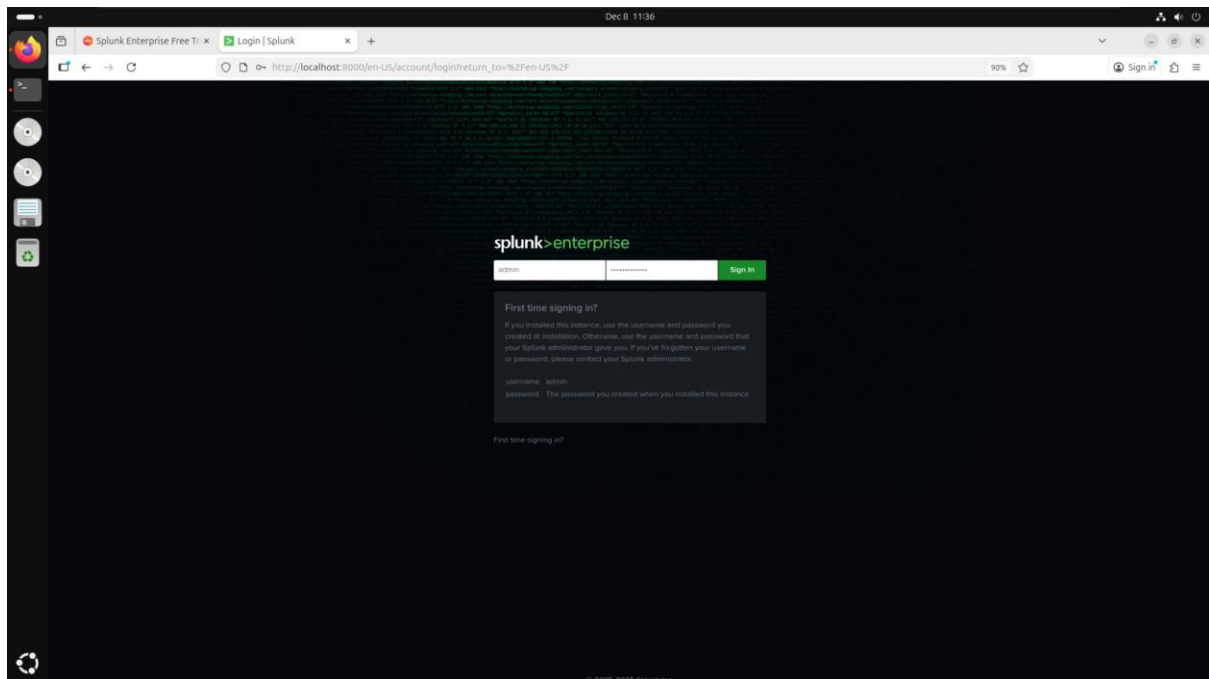
To start Splunk, I ran the command “sudo ./splunk start” from within the “/opt/splunk/bin” folder. Since this was my first time running Splunk on this installation, I had to include the command “--accept-license” which accepted the user license agreement for Splunk. I had to set a username and password for the administrator account which I could then use to sign in once on localhost:8000.



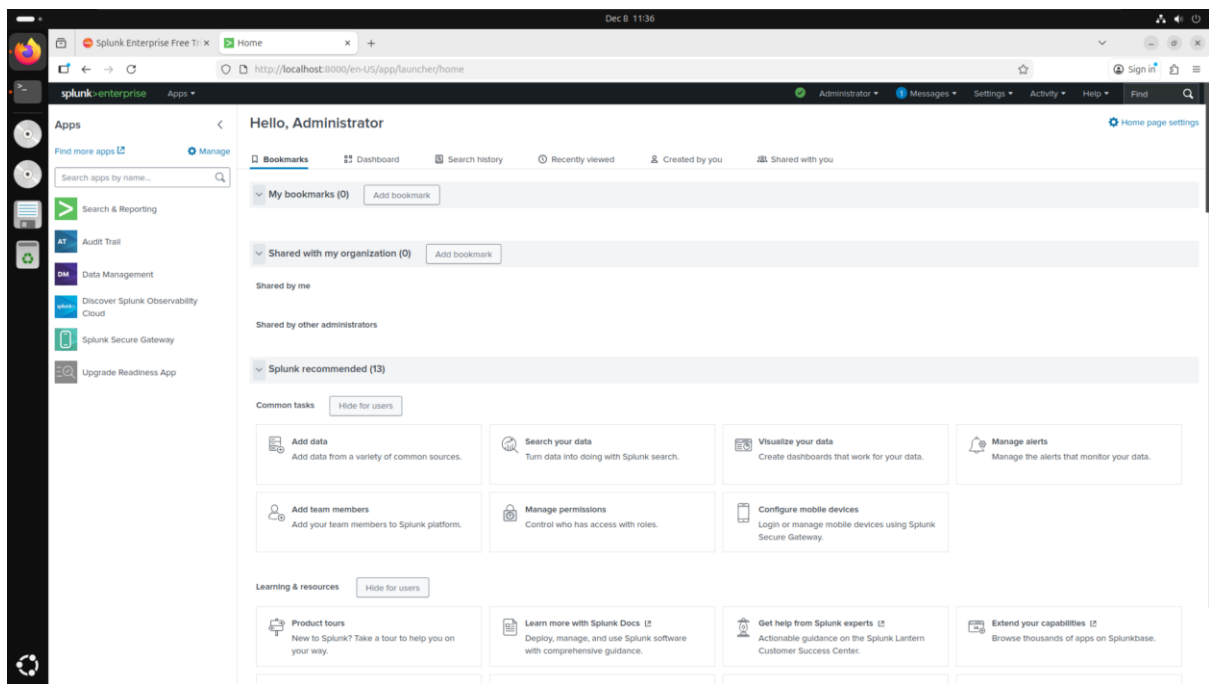
After setting up the initial login details, Splunk generated the web server on localhost:8000, providing a clickable link to the website to allow access into Splunk.



Searching for localhost:8000 on Firefox took me to the Splunk enterprise login page. I entered the username and password I created previously.

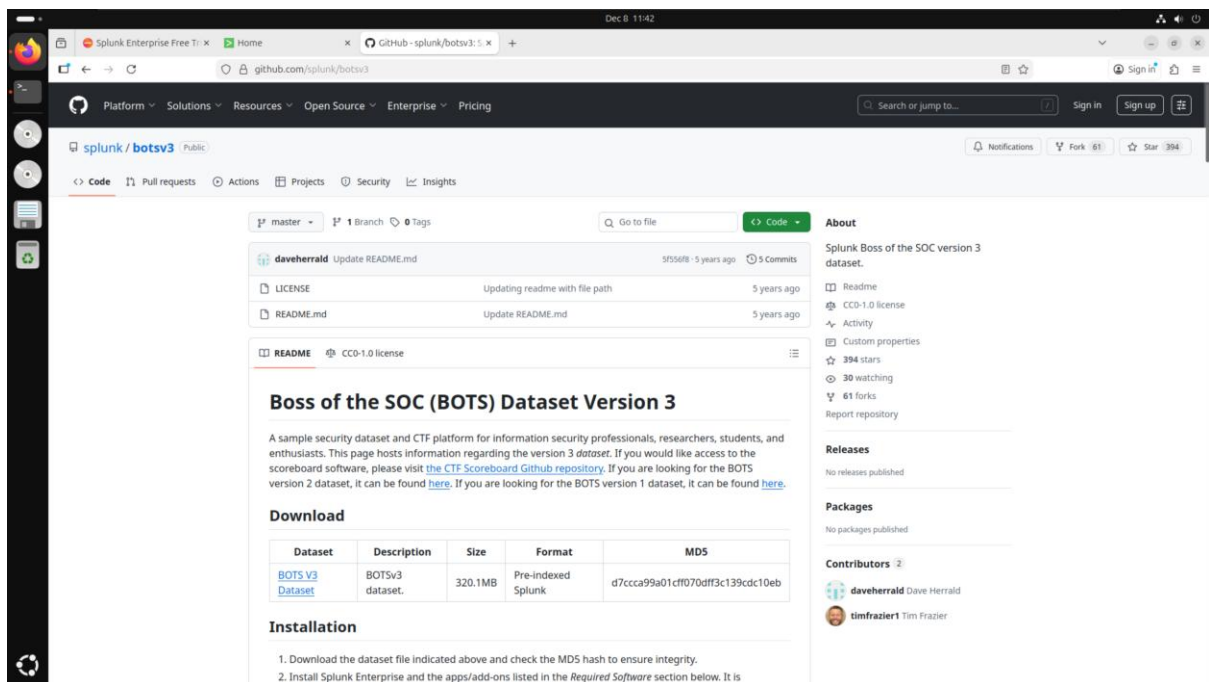


I was able to see the dashboard which houses all of the features I used to answer the guided questions, most notably the “Search & Reporting” tab, which will be explained below.

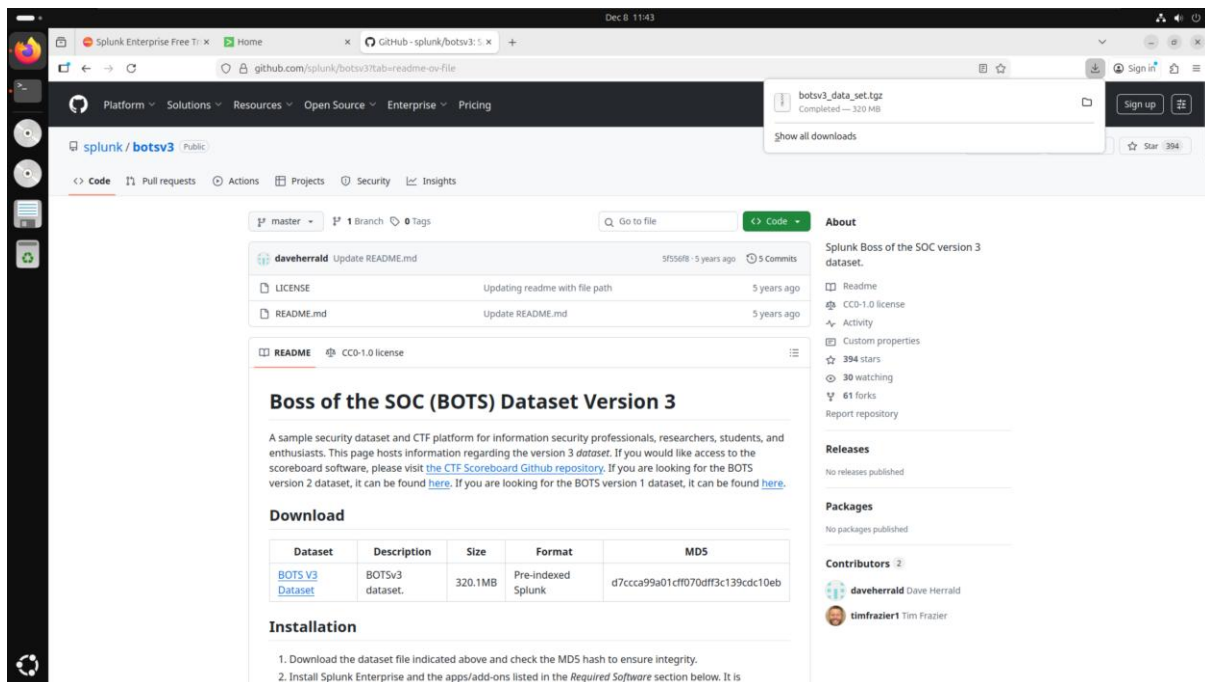


BOTSV3 Dataset Preparation:

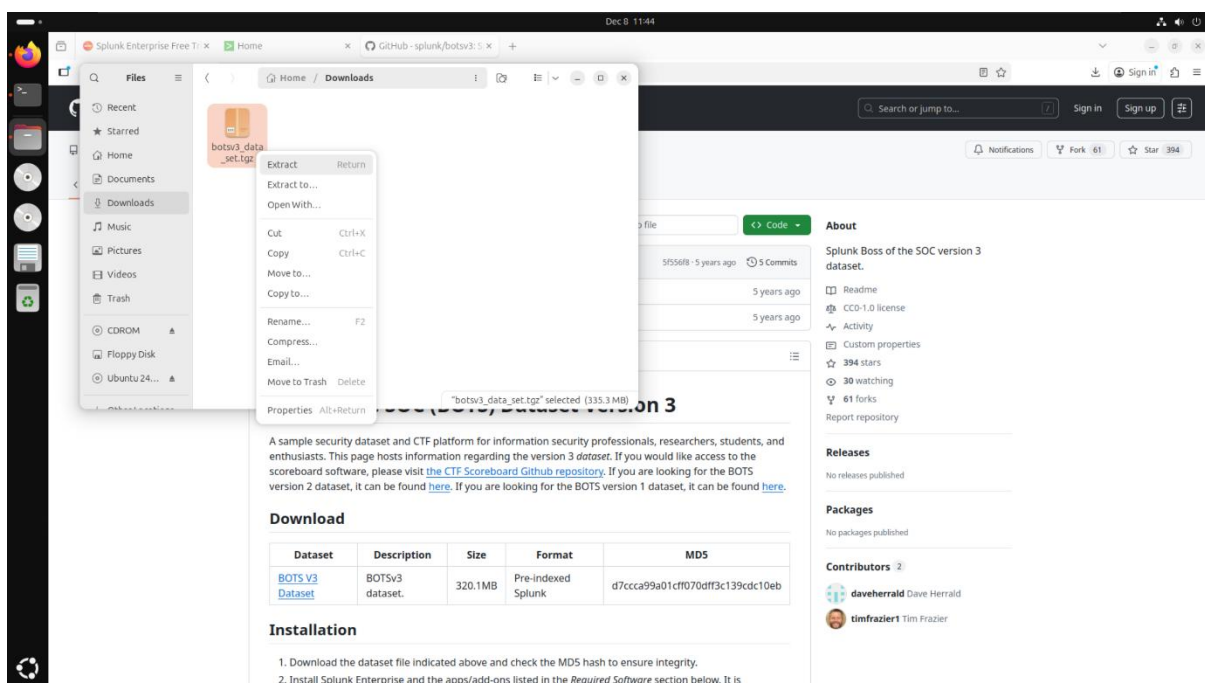
I visited the link <https://github.com/splunk/botsv3> and clicked on the link under dataset in the table to download BOTSV3 to my virtual machine.

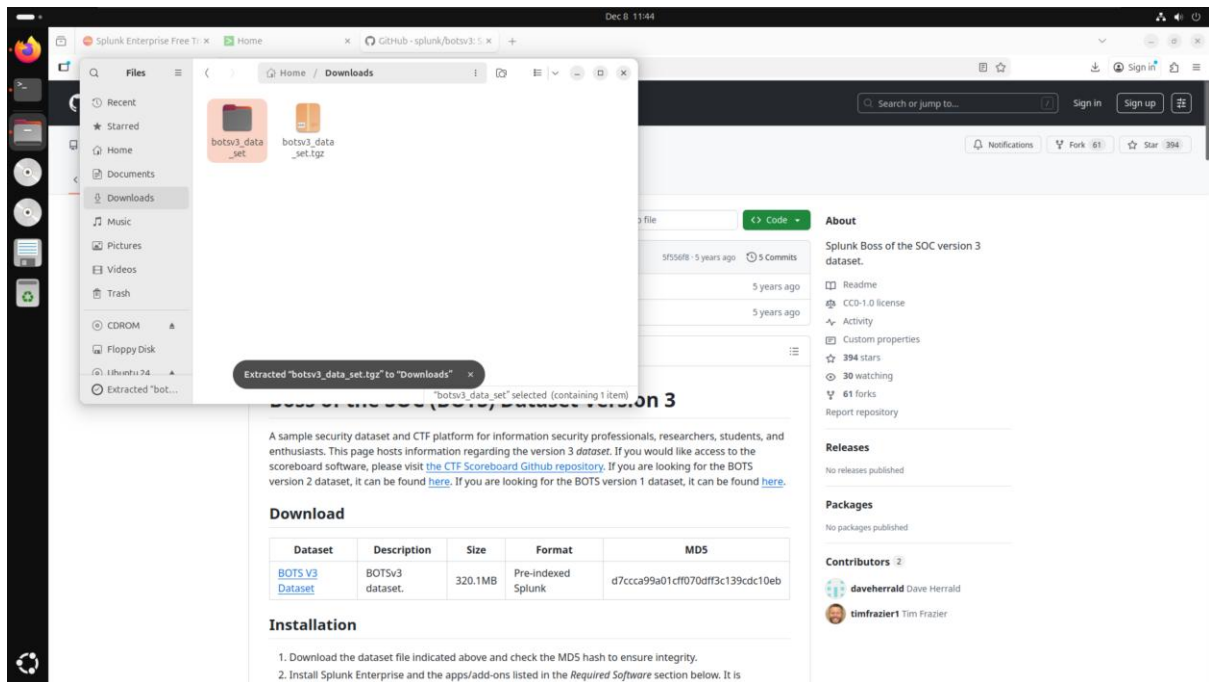


Clicking on the link downloaded the file botsv3_data_set.tgz onto my virtual machine as shown in the downloads tab on Firefox.

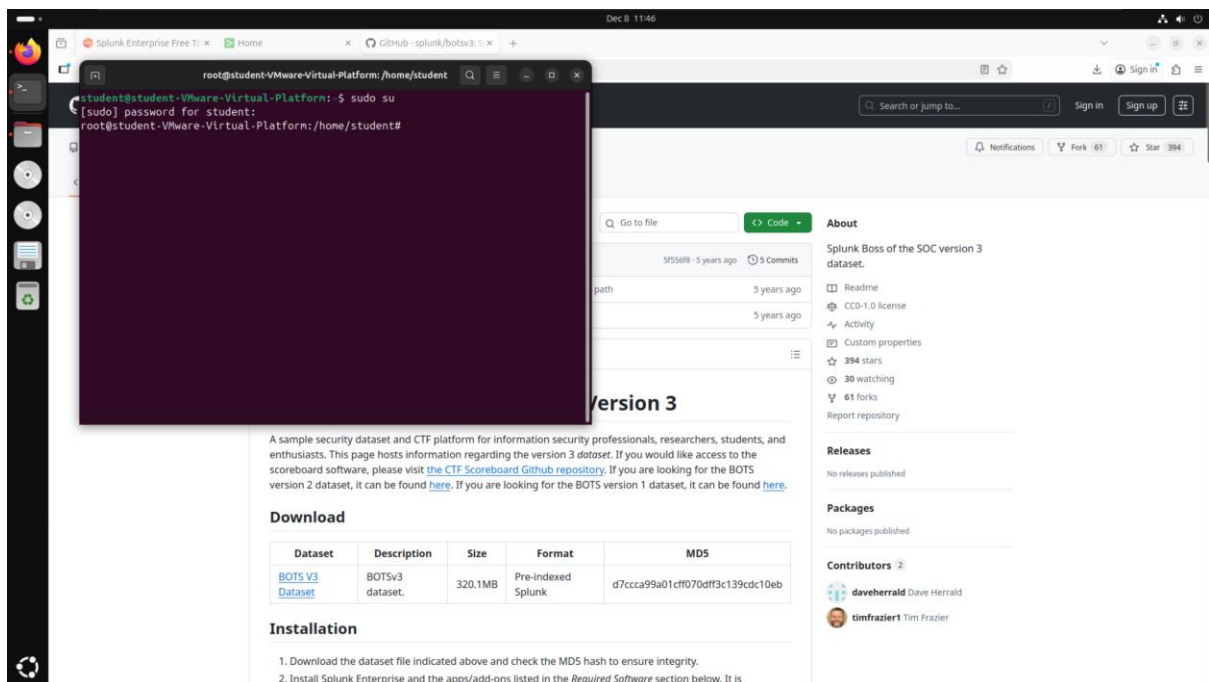


I opened up my file explorer and found the file in the downloads folder where I then extracted it.





To copy the folder to the correct path for Splunk, I entered root mode in the Linux terminal by running the command “sudo su” and entering the password for my virtual machine.



Once in root mode, I ran the command “cp -r botsv3_data_set /opt/splunk/etc/apps/” to copy the file (botsv3_data_set) into the correct path (/opt/splunk/etc/apps/) to allow it to load into Splunk correctly.

The screenshot shows a terminal window on the left and a web browser on the right. The terminal window displays the following commands and output:

```
root@student-Virtual-Platform: /home/student/Downloads
root@student-Virtual-Platform:~# sudo su
[sudo] password for student:
root@student-Virtual-Platform:/home/student# cd Downloads
root@student-Virtual-Platform:/home/student/Downloads# ls
botsv3_data_set  botsv3_data_set.tar
root@student-Virtual-Platform:/home/student/Downloads# cp -r botsv3_data_set /opt/splunk/etc/apps/
root@student-Virtual-Platform:/home/student/Downloads#
```

The web browser shows the GitHub page for the "Splunk Boss of the SOC version 3 dataset". The page includes a "Download" section with a table of dataset information and an "Installation" section with instructions.

Dataset	Description	Size	Format	MD5
BOTS V3 Dataset	BOTSv3 dataset.	320.1MB	Pre-indexed Splunk	d7ccca99a01cff070dff3c139cdc10eb

Installation

1. Download the dataset file indicated above and check the MD5 hash to ensure integrity.
2. Install Splunk Enterprise and the apps/add-ons listed in the *Required Software* section below. It is

I navigated to the correct path to confirm that the file had been correctly copied.

The screenshot shows a terminal window on the left and a web browser on the right. The terminal window displays the following commands and output:

```
root@student-Virtual-Platform: /opt/splunk/etc/apps
root@student-Virtual-Platform:/opt/splunk/etc/apps# ls
alert_logevent      splunk_data_management
alert_webhook       SplunkDeploymentServerConfig
appsbrowser         SplunkForwarder
audit_trail         splunk_gd
botsv3_data_set     splunk_hisinput
introspection_generator_addon  splunk_instrumentation
journald_input      splunk_internal_metrics
launcher            SplunkLightForwarder
learned             splunk_metrics_workspace
legacy             splunk_monitoring_console
python_upgrade_readiness_app  splunk_pipeline_builders
sample_app          splunk_rapiddiag
search             splunk_rolling_upgrade
splunk_app_for_splunk_o11y_cloud  splunk_secure_gateway
splunk_archiver     splunk-visual-exporter
splunk-dashboard-studio  user-profs
root@student-Virtual-Platform:/opt/splunk/etc/apps#
```

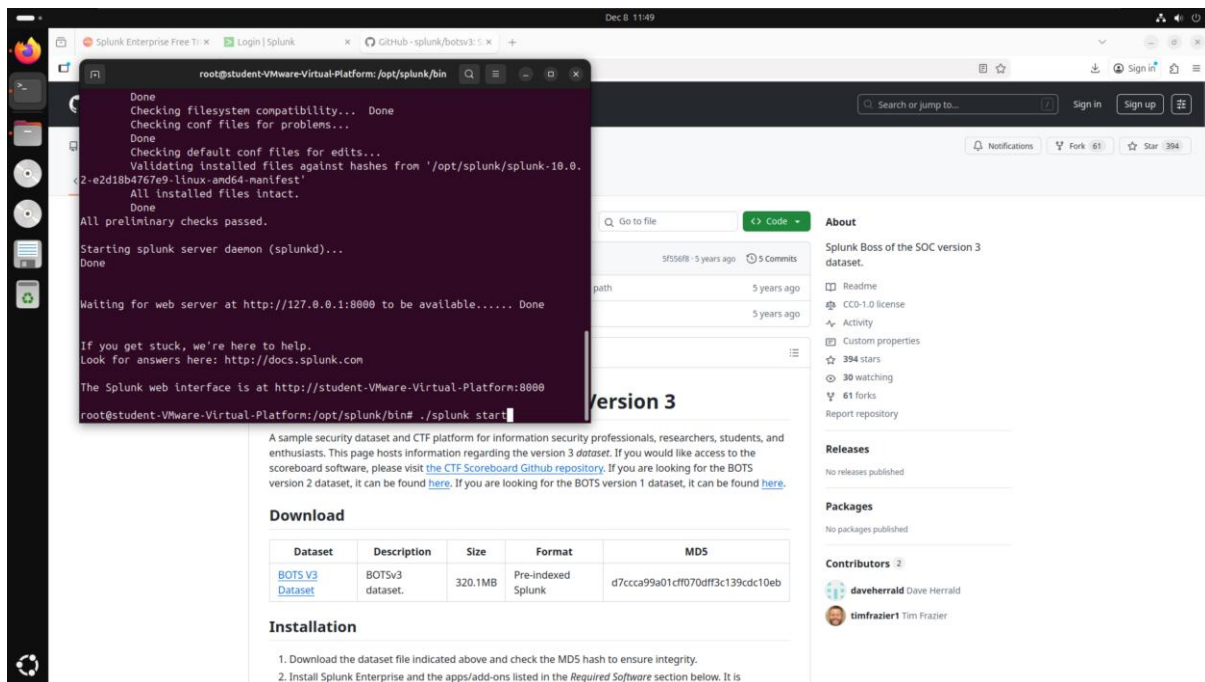
The web browser shows the GitHub page for the "Splunk Boss of the SOC version 3 dataset". The page includes a "Download" section with a table of dataset information and an "Installation" section with instructions.

Dataset	Description	Size	Format	MD5
BOTS V3 Dataset	BOTSv3 dataset.	320.1MB	Pre-indexed Splunk	d7ccca99a01cff070dff3c139cdc10eb

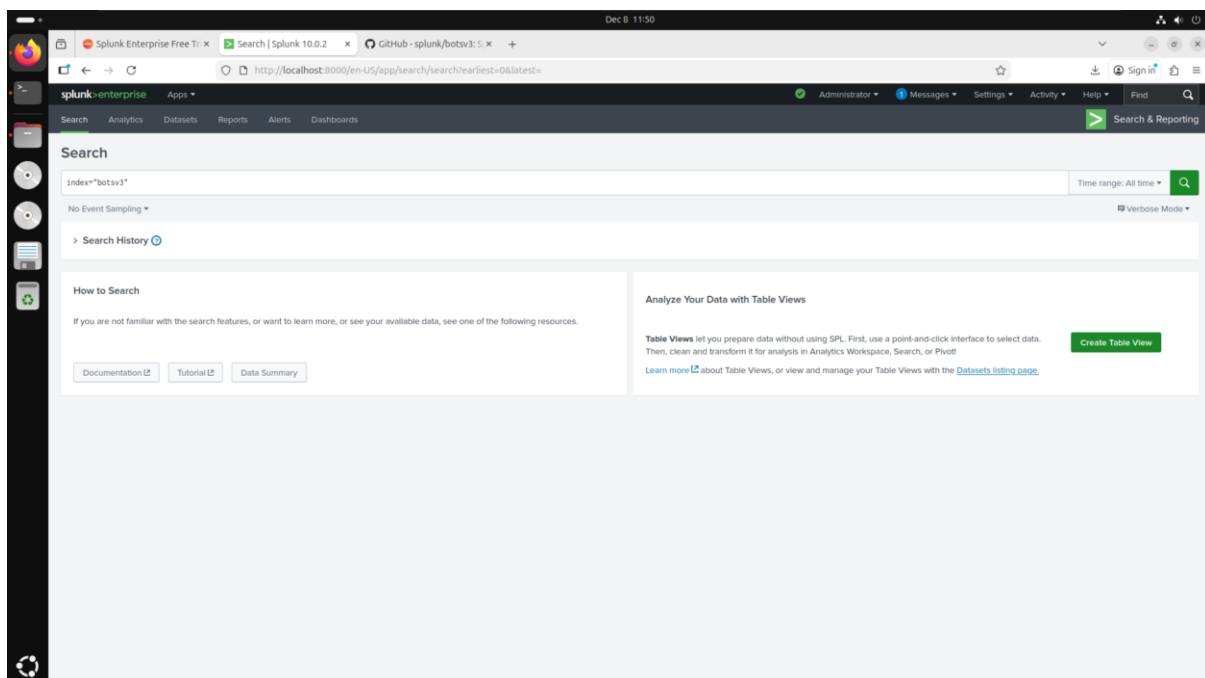
Installation

1. Download the dataset file indicated above and check the MD5 hash to ensure integrity.
2. Install Splunk Enterprise and the apps/add-ons listed in the *Required Software* section below. It is

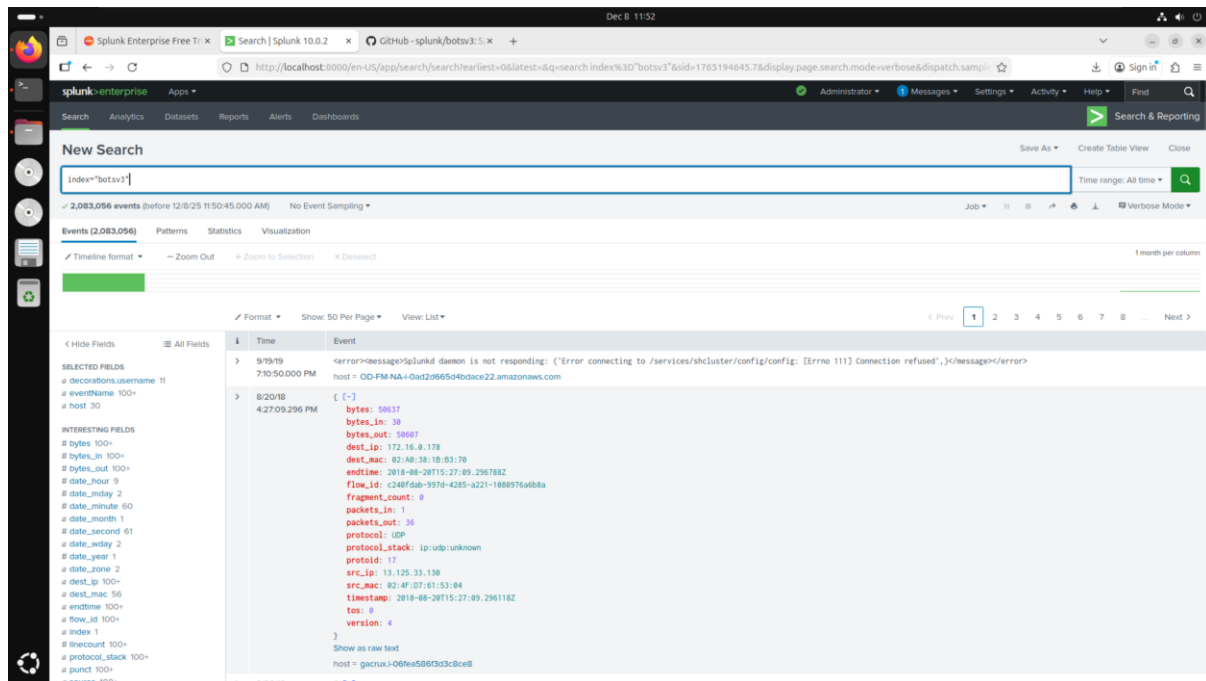
I started Splunk using the command “./splunk start”. I did not have to include the “sudo” command at the beginning of the line as I was still in root mode.



I opened Splunk by searching for localhost:8000 on Firefox. I navigated to the Search & Reporting page then applied the filter index="botsv3".



Running the search with the filter index="botsv3" and setting the time range to filter for all time, I confirmed that the dataset had correctly installed as I had the correct number of 2,083,056 search results. This meant my data was correctly prepared and ready for me to answer the questions. Validating the correct number of events was important to keep data integrity, which is critical in SOC environments since incorrect data could lead to incorrect results.



Guided Questions:

This section answers some of the 200-level questions relating to BOTSv3, showing a typical real-world SOC workflow, determining an attacker and identifying information surrounding an incident. There are 8 questions total that I will answer, outlining which Splunk filters I used and what my results were.

Question 1:

The screenshot shows the Splunk Enterprise web interface. The search bar contains the query `index=botsv3 sourcetype=aws::cloudtrail`. The search results show 6,571 events. A timeline view is displayed with a highlighted event. A modal window is open showing the event details for `userIdentity.userName`. The modal includes a table of values and a list of reports.

Search Query: `index=botsv3 sourcetype=aws::cloudtrail`

Results: 6,571 events (before 12/6/25 3:29:51.000 PM) No Event Sampling

Timeline View: 1 hour per column

Event Details Modal:

Field: `userIdentity.userName`

Values:

Values	Count	%
<code>splunk_access</code>	4,891	75.41%
<code>web_admin</code>	646	11.968%
<code>bstoll</code>	615	11.336%
<code>btun</code>	73	1.346%

Reports:

- Top values
- Top values by time
- Rare values
- Events with this field

Event Log:

```

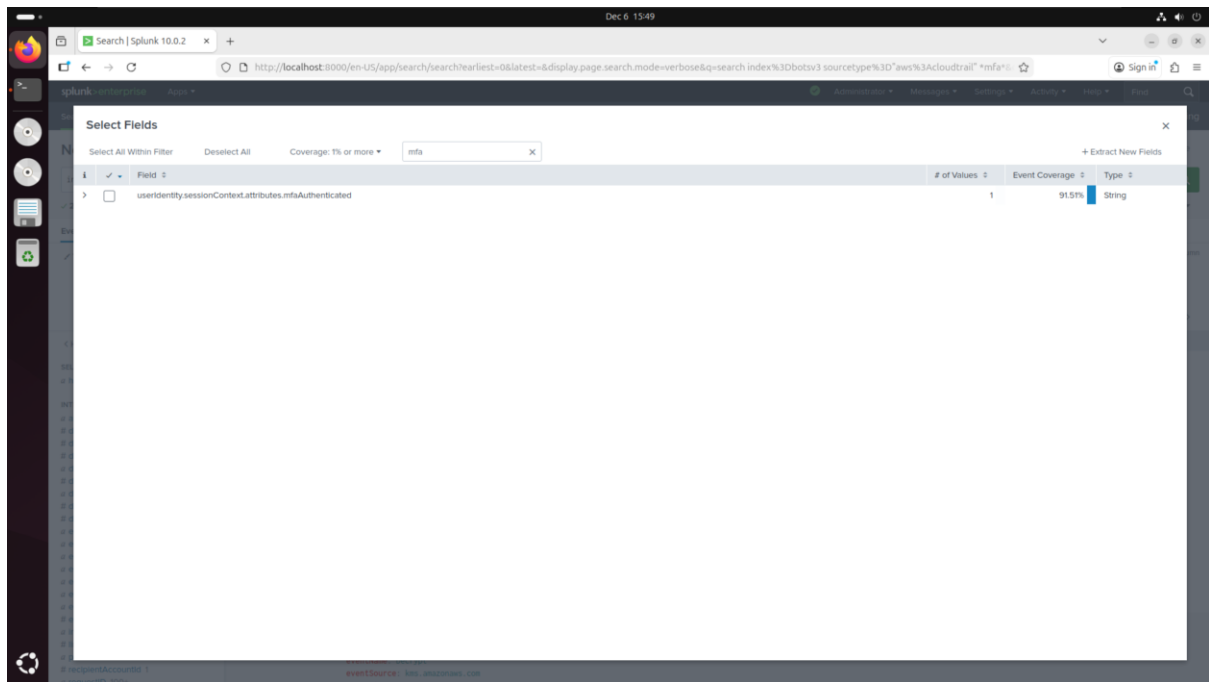
8/20/18 4:15:13.000 PM [ ]
  responseElements: null
  sourceIPAddress: autoscaling.amazonaws.com
  userAgent: autoscaling.amazonaws.com
  userIdentity: [ ]
}
Show as raw text
host = splunk.rothly

> 8/20/18 4:15:13.000 PM [ ]
  awsRegion: us-west-1
  eventId: if6881b-1b87-4320-b887-42219345e208
  eventName: Decrypt
  eventSource: kms.amazonaws.com
  
```

I used the provided hint to filter Splunk by sourcetype="aws:cloudtrail". I then searched through the available filters to find anything which could be related to usernames. Here I found the filter for "useridentity.username", which I then applied to show me the 4 usernames; bstoll, btun, splunk_access and web_admin. This links to tier 1 of SOC as I am monitoring the dashboards to find if an incident has occurred by checking which users have access.

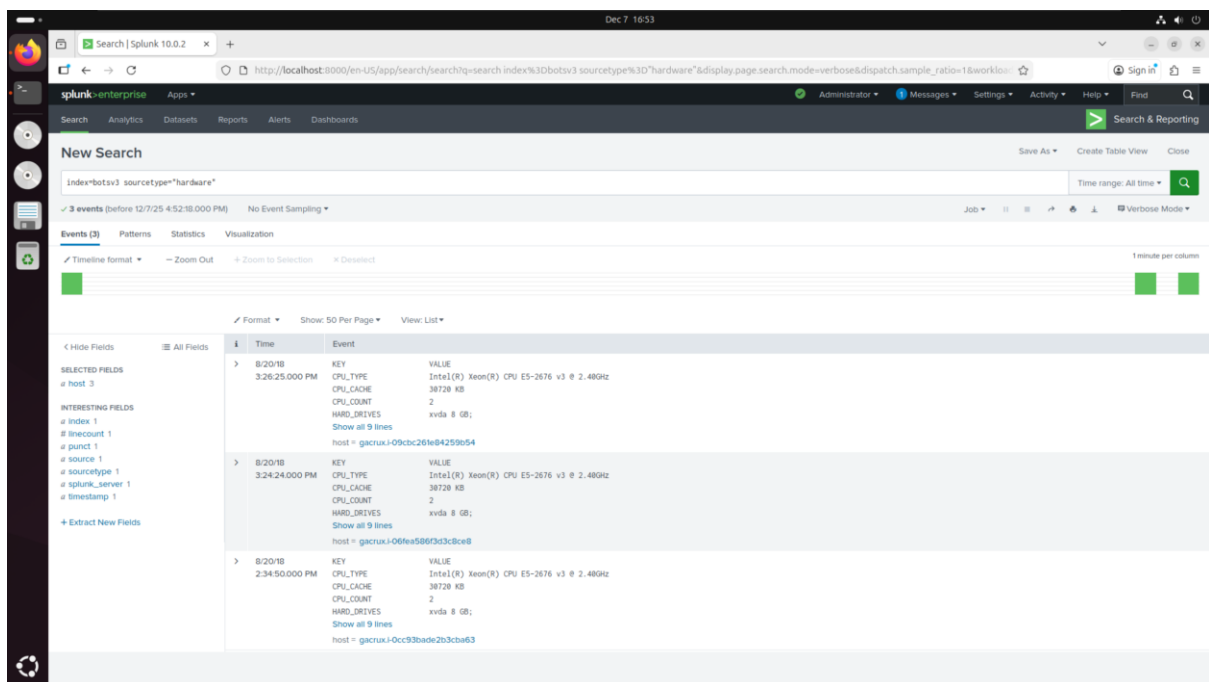
Question 2:

The screenshot shows the Splunk Enterprise search interface. The search bar contains the query: `index=botsv3 sourcetype="aws:cloudtrail" *mfaf`. The search results are displayed in a table format, showing event details like eventID, eventName, and eventSource. The interface includes a sidebar with navigation options like Search, Analytics, Datasets, Reports, Alerts, and Dashboards. The main content area shows a search bar, a time range selector, and a table of search results. The table has columns for Time and Event. The first result is for an eventID of 97d8f8b-c8cf-437c-8b85-4843635ce386, occurring on 8/20/18 at 4:15:20.000 PM. The event details show it was triggered by the awsRegion: us-west-1, eventSource: ec2.amazonaws.com, and eventTime: 2018-08-20T15:15:20Z. The event type is Aspicall, and the recipientAccountID is 622678721278. The requestID is fbd4e9b-e27c-4a52-93fa-fab7a7663639, and the requestParameters are listed. The second result is for an eventID of 6f68813b-5b67-432c-bd87-d2215345e928, occurring on 8/20/18 at 4:15:13.000 PM. The event details show it was triggered by the awsRegion: us-west-1, eventSource: kms.amazonaws.com, and eventTime: 2018-08-20T15:13:00Z. The event type is Decrypt, and the recipientAccountID is kms.amazonaws.com.



I kept the same filter as question 1 and searched additionally for *MFA* using the provided hint to find anything relating to multi-factor authentication. I then searched for “mfa” in the additional filters which led me to find the filter `userIdentity.sessionContext.attributes.mfaAuthenticated`.

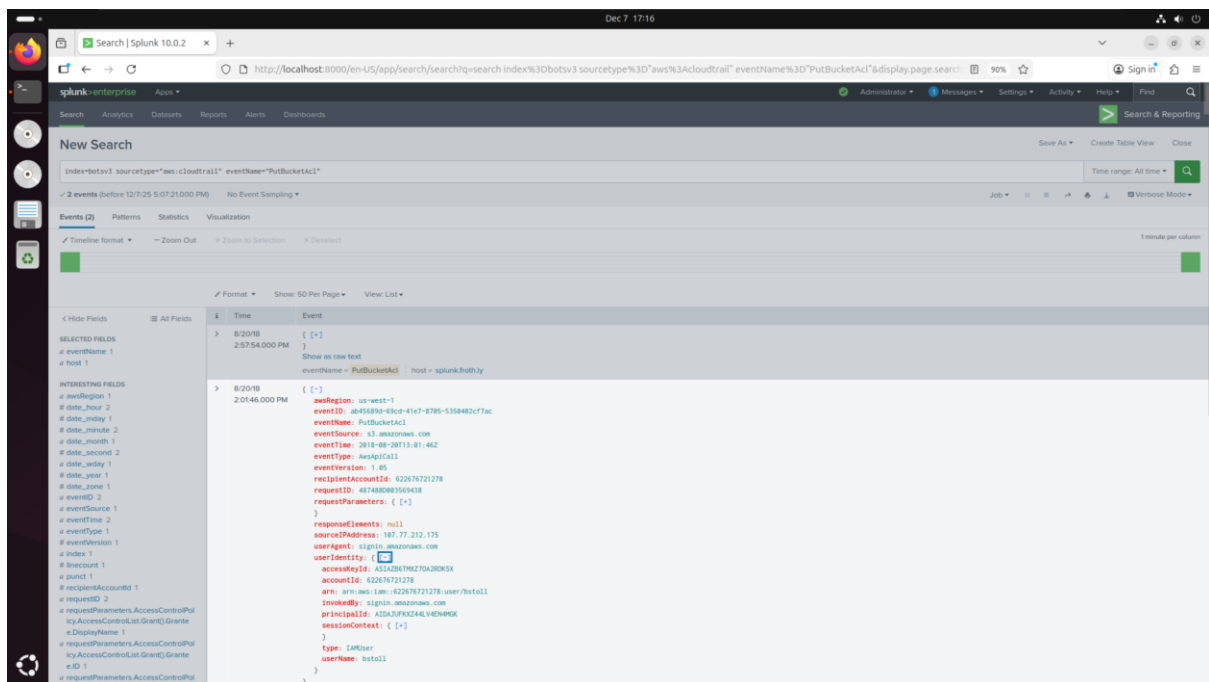
Question 3:



I changed the filter to be `sourcetype="hardware"` using the provided hint, which then showed me the list of hardware being used to connect to the server. Looking into these values, I found the value `CPU_TYPE` which was listed as the model number E5-2676.

The screenshot displays the Splunk Enterprise web interface. At the top, the browser address bar shows the URL: `http://localhost:8000/en-US/app/search/search?q=search index%3Dbotb3v3 sourcetype%3D%3Daws%3Acloudtrail eventName%3D%3DPutBucketAct&display.page.search.mode=`. The interface includes a navigation bar with tabs for Search, Analytics, Datasets, Reports, Alerts, and Dashboards. The main content area is titled "New Search" and shows a search query: `index=botb3v3 sourcetype=aws:cloudtrail eventName=PutBucketAct`. Below the query, it indicates "2 events (before 12/25 5:07:21.000 PM)" and "No Event Sampling". The search results are displayed in a table with columns for Time and Event. The first event is from 8/20/18 at 2:57:54.000 PM, showing a raw text view of an event where `eventName = PutBucketAct` and `host = splunk.broth.ly`. The second event is from 8/20/18 at 2:01:46.000 PM, showing a raw text view of a detailed event log entry, including fields like `awsRegion`, `eventID`, `eventName`, `eventSource`, `eventTime`, `eventType`, `eventVersion`, `eventSize`, `recipientAccountID`, `requestID`, `requestParameters`, `responseElements`, `sourceIPAddress`, `userAgent`, and `userIdentity`.

Question 5:

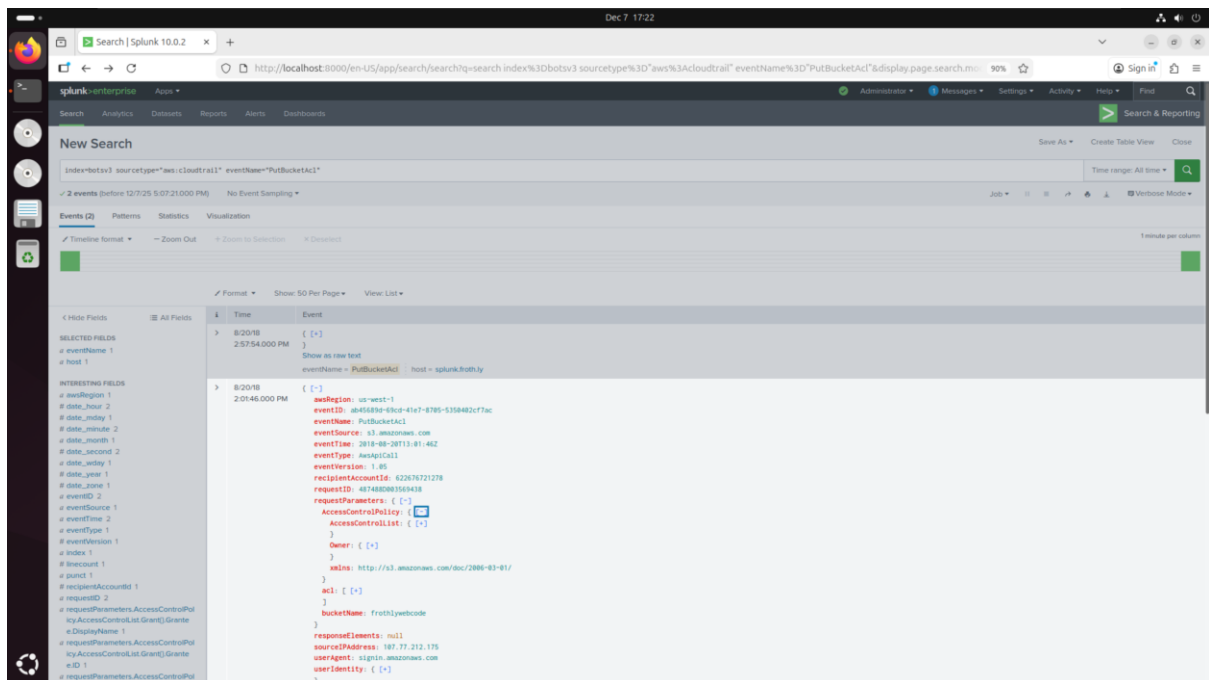


The screenshot shows the Splunk Enterprise interface with a search query: `index=botv3 sourcetype="aws:cloudtrail" eventName="PutBucketAcl"`. The search results show two events. The first event is expanded, showing the `userIdentity` field with the following details:

```
{
  "awsRegion": "us-west-1",
  "eventID": "ab4589b-6bc-4e7-8785-538482c7fac",
  "eventName": "PutBucketAcl",
  "eventSource": "s3.amazonaws.com",
  "eventTime": "2018-08-28T13:01:46Z",
  "eventType": "AwsApiCall",
  "eventVersion": "1.05",
  "recipientAccountID": "622676721278",
  "requestID": "4b7480083569438",
  "requestParameters": {
    "acl": "private"
  },
  "responseElements": null,
  "sourceIPAddress": "187.77.212.175",
  "userAgent": "signin.amazonaws.com",
  "userIdentity": {
    "accessKeyId": "ASIA298T9M270A2R0K5X",
    "accountId": "622676721278",
    "arn": "arn:aws:iam::622676721278:user/bstoll",
    "invokedBy": "signin.amazonaws.com",
    "principalId": "AIDAJUKZ44V4ENWPK",
    "sessionContext": {
      "type": "IAMUser",
      "userName": "bstoll"
    }
  }
}
```

I applied the same filters as question 4. From there, I opened the `userIdentity` field within the relevant event and found that the username was `bstoll`.

Question 6:



The screenshot shows the Splunk Enterprise interface with the same search query as in Question 5. The search results show two events. The second event is expanded, showing the `requestParameters` field with the following details:

```
{
  "AccessControlPolicy": {
    "AccessControlList": [
      {
        "Owner": {
          "name": "http://s3.amazonaws.com/doc/2006-03-01/"
        },
        "acl": [
          {
            "bucketName": "frothlywebcode"
          }
        ]
      }
    ]
  },
  "responseElements": null,
  "sourceIPAddress": "187.77.212.175",
  "userAgent": "signin.amazonaws.com",
  "userIdentity": {
    "accessKeyId": "ASIA298T9M270A2R0K5X",
    "accountId": "622676721278",
    "arn": "arn:aws:iam::622676721278:user/bstoll",
    "invokedBy": "signin.amazonaws.com",
    "principalId": "AIDAJUKZ44V4ENWPK",
    "sessionContext": {
      "type": "IAMUser",
      "userName": "bstoll"
    }
  }
}
```

I applied the same filters as question 4. From there, I opened the `requestParameters` field within the relevant event and found that the `bucketName` was `frothlywebcode`.

Question 7:

The screenshot shows the Splunk Enterprise search interface. The search bar contains the query: `index=botsv sourcetype="aws:s3:accesslogs" date_hour=14 PUT *txt*`. The search results show 1 event from 8/20/18 12:00:00.000 AM to 8/21/18 12:00:00.000 AM. The event details are as follows:

Time	Event
8/20/18 2:02:44.000 PM	<code>Ac818053e748f45b4e43f68c8f5eff6347745488ae548138432c3f64fa318d frothywebcode [28/Aug/2018:13:02:44 +0000] 52.66.146.128 - DF18A3809E2369B4 REST PUT OBJECT OPEN_BUCKET_PLEASE_FIX.txt /OPEN_BUCKET_PLEASE_FIX.txt HTTP/1.1" 200 - - 377 268 9 "-" "botso/1.7.62 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 Botocore/1.8.12" -</code> <code>host = splunkfrothy source = s3:/frothyweblogs/s32018-07-26-01-20-56-19073C05AA29AED8 sourcetype = aws:s3:accesslogs</code>

The interface also shows a list of fields on the left, including `date_hour`, `date_month`, `date_year`, `index`, `linecount`, `point`, `splunk_server`, `timeendpos`, and `timestartpos`.

I changed the filter to `sourcetype="aws:s3:accesslogs"` based on the provided hint. Then, knowing that the events in questions 4 to 6 occurred between 2pm and 3pm, I added a filter for `date_hour=14` to check specifically between these times. Next, I added another filter for HTTP PUT requests, since the question involved uploading a file to the S3 bucket. I lastly added a filter for `*txt*` since the question mentioned a text file being uploaded which hinted to me that it would be a .txt file. This led me to find the uploaded text file "OPEN_BUCKET_PLEASE_FIX.txt".

Question 8:

The screenshot displays the Splunk Enterprise web interface. At the top, the search bar contains the query: `index=botsv3 sourcetype=winhostmon source=operatingsystem OS=Microsoft Windows 10 Enterprise`. Below the search bar, the results are shown in a table format. The table has two main columns: **Time** and **Event**. There are four events listed, all from 8/20/18. Each event entry includes a timestamp, a system architecture string, a version number, and a build number. The interface also shows a sidebar with navigation options like Search, Analytics, Datasets, Reports, Alerts, and Dashboards. The top navigation bar includes links for Administrator, Messages, Settings, Activity, and Help. The bottom of the interface shows a status bar with the text 'Dec 8 02:35'.

Time	Event
8/20/18 4:44:22.000 PM	Type=operatingsystem OS=Microsoft Windows 10 Enterprise Architecture="64-bit" Version="18.0.17134" BuildNumber="17134" Show all 22 lines host = BSTOLL-L source = operatingsystem sourcetype = WinHostMon
8/20/18 4:04:21.000 PM	Type=operatingsystem OS=Microsoft Windows 10 Enterprise Architecture="64-bit" Version="18.0.17134" BuildNumber="17134" Show all 22 lines host = BSTOLL-L source = operatingsystem sourcetype = WinHostMon
8/20/18 3:54:26.000 PM	Type=operatingsystem OS=Microsoft Windows 10 Enterprise Architecture="64-bit" Version="18.0.17134" BuildNumber="17134" Show all 22 lines host = BSTOLL-L source = operatingsystem sourcetype = WinHostMon
8/20/18 3:25:27.000 PM	Type=operatingsystem OS=Microsoft Windows 10 Enterprise

Dec 8 02:36

Search | Splunk 10.0.2

http://127.0.0.1:8000/en-US/app/search/search?q=search index%3Dbotsv3 host%3DBTSTOLL-L&display.page.search.mode=verbose&dispatch.sample_ratio=1&workload_pool=

Administration Messages Settings Activity Help Find Search & Reporting

New Search

index=botsv3 host=BTSTOLL-L

240,882 events (before 12/8/25 2:35:48:00 AM) No Event Sampling

Events (240,882) Patterns Statistics Visualization

Timeline format Zoom Out Zoom to Selection Deselect

1 hour per column

Format Show: 50 Per Page View: List

Hide Fields All Fields

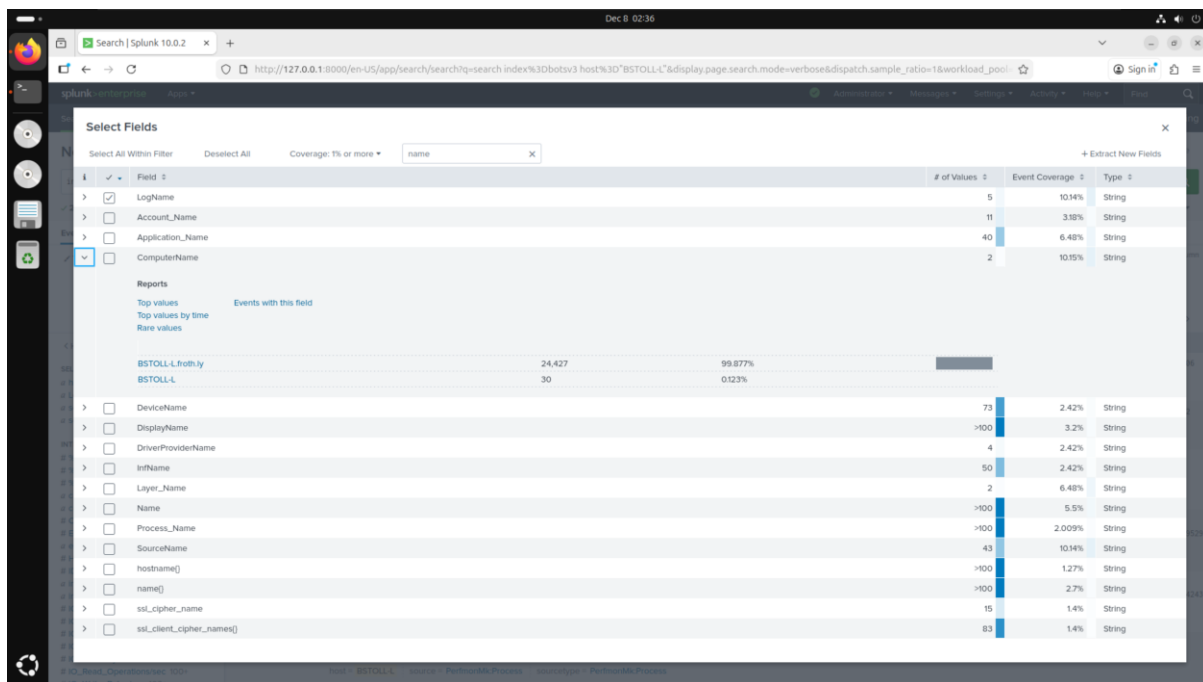
SELECTED FIELDS

- host 1
- LogName 5
- source 43
- sourcetype 21

INTERESTING FIELDS

- %_Privileged_Time 100+
- %_Processor_Time 100+
- %_User_Time 100+
- category 1
- collection 1
- Creating_Process_ID 100+
- Elapsed_Time 100+
- endtime 100+
- Handle_Count 100+
- ID_Process 100+
- index 1
- instance 100+
- IO_Data_Bytes/sec 100+
- IO_Operations/sec 100+
- IO_Other_Bytes/sec 100+
- IO_Other_Operations/sec 100+
- IO_Read_Bytes/sec 100+
- IO_Read_Operations/sec 100+

Time	Event	host	source	sourcetype	Count
8/20/18 4:17:59:00 PM	ApplicationFrameHost	7336 776 451480 31136 527	2283614883840 2283607338816 51679232 37842944 33497088 20021248 20021248 5 8 21824 4694806	BTSTOLL-L PerfMonMxProcess PerfMonMxProcess	21824 4694806
8/20/18 4:17:59:00 PM	Calculator	4621877248 4618731520 54685696 36228928 20492288 16691200 16691200 22 8 19887 784263 5212	4621877248 4618731520 54685696 36228928 20492288 16691200 16691200 22 8 19887 784263 5212	BTSTOLL-L PerfMonMxProcess PerfMonMxProcess	19887 784263 5212
8/20/18 4:17:59:00 PM	CrashPlanDesktop	0.3398138653191165 0.31380435643918553 0.6260887128794111 335644416 309473280 2.884475918851634 59437956 48525312 38785024	0.3398138653191165 0.31380435643918553 0.6260887128794111 335644416 309473280 2.884475918851634 59437956 48525312 38785024	BTSTOLL-L PerfMonMxProcess PerfMonMxProcess	38785024
8/20/18 4:17:59:00 PM	CrashPlanService	0.6260887128794111 0.15658217821985277 0.469586346595883 1532659848 1570185736 78592520 125364272 65955404 35579948	0.6260887128794111 0.15658217821985277 0.469586346595883 1532659848 1570185736 78592520 125364272 65955404 35579948	BTSTOLL-L PerfMonMxProcess PerfMonMxProcess	35579948
8/20/18 4:17:59:00 PM	GoogleCrashHandler	69746688 64563888 6803456 1314552 1953792 1667872 1667872 3 4 20981 8776684 1724 6148 124176 16280 168	69746688 64563888 6803456 1314552 1953792 1667872 1667872 3 4 20981 8776684 1724 6148 124176 16280 168	BTSTOLL-L PerfMonMxProcess PerfMonMxProcess	16280 168



I started by filtering for sourcetype="winhostmon" using the provided hint. I then searched through the filters to find a filter for "operatingsystem" and an additional filter for OS="Microsoft Windows 10 Enterprise". This operating system only had 30 uses, which all came from one user – BSTOLL-L. Once I had this user, I then removed my operatingsystem and OS filters, and instead added a filter for host="BSTOLL-L". I searched the filters list again for anything relating to FQDN, however I could not find anything. I instead searched the additional filters for anything relating to "name", where I found the filter ComputerName which when inspected had the value of BSTOLL-L.froth.ly.

Conclusion:

This report has demonstrated the real-world application of SOC tiers when analysing the BOTSv3 dataset within Splunk. I have mainly applied tier 2 of SOC, however, have also applied other incident handling methodologies to investigate and identify security incidents within the dataset. Overall, this report has highlighted the importance of real-world SOC operations in identifying and analysing security incidents.

References:

BOTSv3 GitHub download: <https://github.com/splunk/botsv3>

ChatGPT: <https://chatgpt.com/>

Splunk Enterprise download: https://www.splunk.com/en_us/download/splunk-enterprise.html

SOC roles explanation: <https://www.paloaltonetworks.com/cyberpedia/soc-roles-and-responsibilities#:~:text=A%20security%20operations%20center%2C%20or,security%20solutions%2C%20tools%20and%20products.>

NIST incident handling methodology: <https://www.nist.gov/cyberframework/getting-started/online-learning/five-functions>

AI Declaration:

Solo Work	S1 - Generative AI tools have not been used for this assessment.	<input type="checkbox"/>
Assisted Work	A1 – Idea Generation and Problem Exploration Used to generate project ideas, explore different approaches to solving a problem, or suggest features for software or systems. Students must critically assess AI-generated suggestions and ensure their own intellectual contributions are central.	<input type="checkbox"/>
	A2 - Planning & Structuring Projects AI may help outline the structure of reports, documentation and projects. The final structure and implementation must be the student's own work.	<input type="checkbox"/>
	A3 – Code Architecture AI tools maybe used to help outline code architecture (e.g. suggesting class hierarchies or module breakdowns). The final code structure must be the student's own work.	<input type="checkbox"/>
	A4 – Research Assistance Used to locate and summarise relevant articles, academic papers, technical documentation, or online resources (e.g. Stack Overflow, GitHub discussions). The interpretation and integration of research into the assignment remain the student's responsibility.	<input type="checkbox"/>
	A5 - Language Refinement Used to check grammar, refine language, improve sentence structure in documentation not code. AI should be used only to provide suggestions for improvement. Students must ensure that the documentation accurately reflects the code and is technically correct.	<input type="checkbox"/>
	A6 – Code Review AI tools can be used to check comments within the code and to suggest improvements to code readability, structure or syntax. AI should be used only to provide suggestions for improvement. Students must ensure that the code accurately reflects their knowledge and is technically correct.	<input type="checkbox"/>
	A7 - Code Generation for Learning Purposes Used to generate example code snippets to understand syntax, explore alternative implementations, or learn new programming paradigms. Students must not submit AI-generated code as their own and must be able to explain how it works.	<input type="checkbox"/>
	A8 - Technical Guidance & Debugging Support AI tools can be used to explain algorithms, programming concepts, or debugging strategies. Students may also help interpret error messages or suggest possible fixes. However, students must write, test, and debug their own code independently and understand all solutions submitted.	<input type="checkbox"/>
	A9 - Testing and Validation Support AI may assist in generating test cases, validating outputs, or suggesting edge cases for software testing. Students are responsible for designing comprehensive test plans and interpreting test results.	<input type="checkbox"/>
	A10 - Data Analysis and Visualization Guidance AI tools can help suggest ways to analyse datasets or visualize results (e.g.	<input type="checkbox"/>

	recommending chart types or statistical methods). Students must perform the analysis themselves and understand the implications of the results.	
	A11 - Other uses not listed above Please specify:	<input type="checkbox"/>
Partnered Work	P1 - Generative AI tool usage has been used integrally for this assessment Students can adopt approaches that are compliant with instructions in the assessment brief. Please Specify: <ul style="list-style-type: none"> - Locating reports - Summarising reports - Fixing grammar - More precise language 	<input checked="" type="checkbox"/>

Please provide details of AI usage and which elements of the coursework this relates to:

Partnered Work – Used to locate and summarise reports for SOC and incident handling, Used to make language more concise and in proper English, including fixing grammar

I understand that the ownership and responsibility for the academic integrity of this submitted assessment falls with me, the student.	<input checked="" type="checkbox"/>
I confirm that all details provide above are an accurate description of how AI was used for this assessment.	<input checked="" type="checkbox"/>