

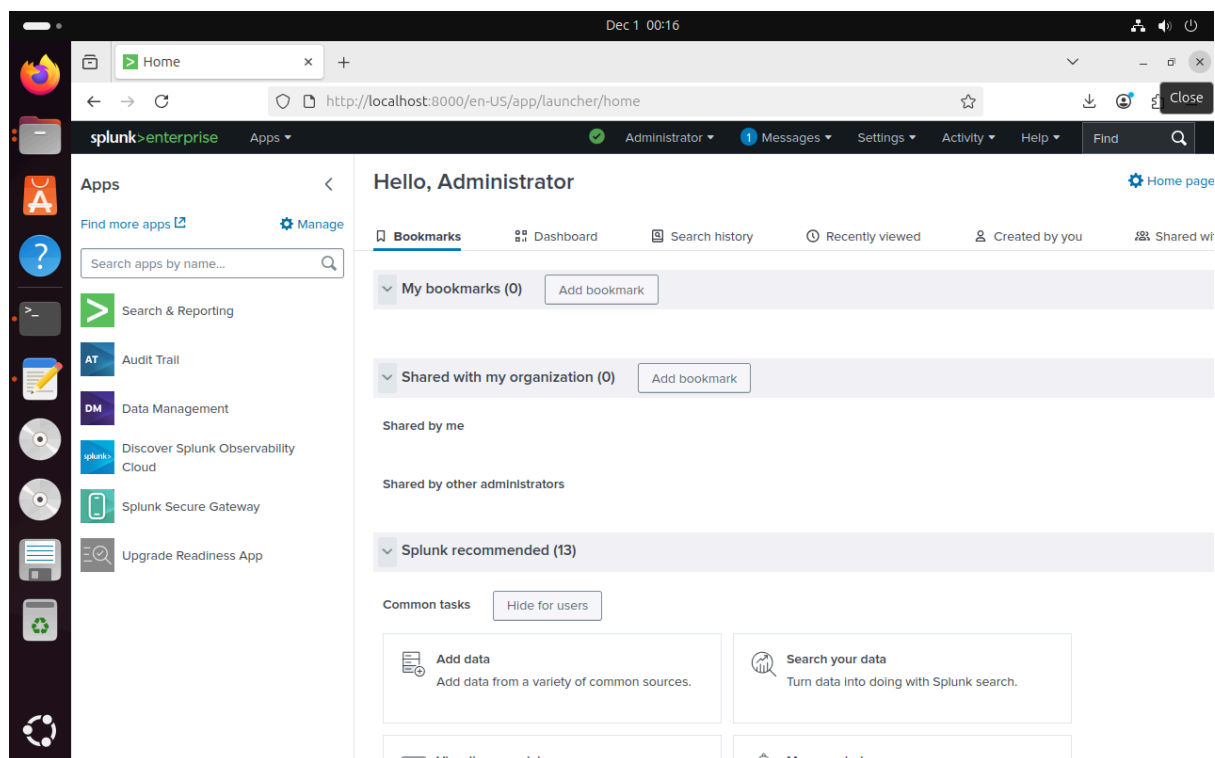
# Introduction

This report presents the analysis of the Splunk Boss of the SOC version 3 (BOTSv3) dataset, with an investigation focused on a simulated compromise at Frothly, an organisation with AWS cloud infrastructure and Windows endpoints. Using Splunk Enterprise, I ingested and analysed logs including AWS CloudTrail, S3 access logs, Windows host monitoring, and hardware inventory to answer the set of guided questions and produce this report.

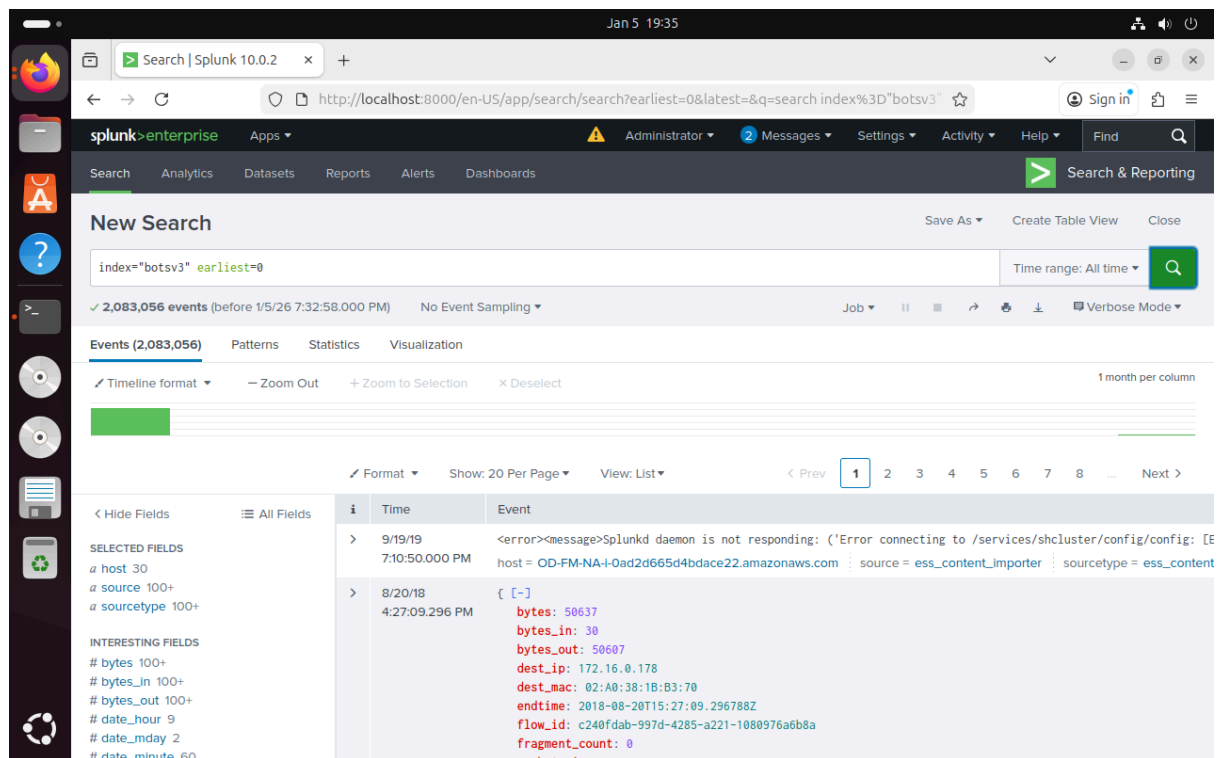
The following will detail the installation of software as well as the loading of the BOTSv3 dataset into Splunk before going over the investigation questions provided on the DLE. This information will be used to surmise an incident narrative, leading to conclusions and recommendations based on my findings.

## Installation and Data Preparation

Splunk Enterprise was installed on an Ubuntu virtual machine running in VMware Workstation Pro. The VM was configured with 4 GB RAM and a 70 GB disk to accommodate the large indexed dataset and prevent performance issues during long-running searches.



The official BOTSv3 dataset was downloaded from the Splunk GitHub repository and ingested using the provided *ingest\_botsv3.py script*. This created the index botsv3 containing AWS CloudTrail, S3 access logs, Windows host monitoring (winhostmon), hardware inventory, and other relevant sourcetypes.



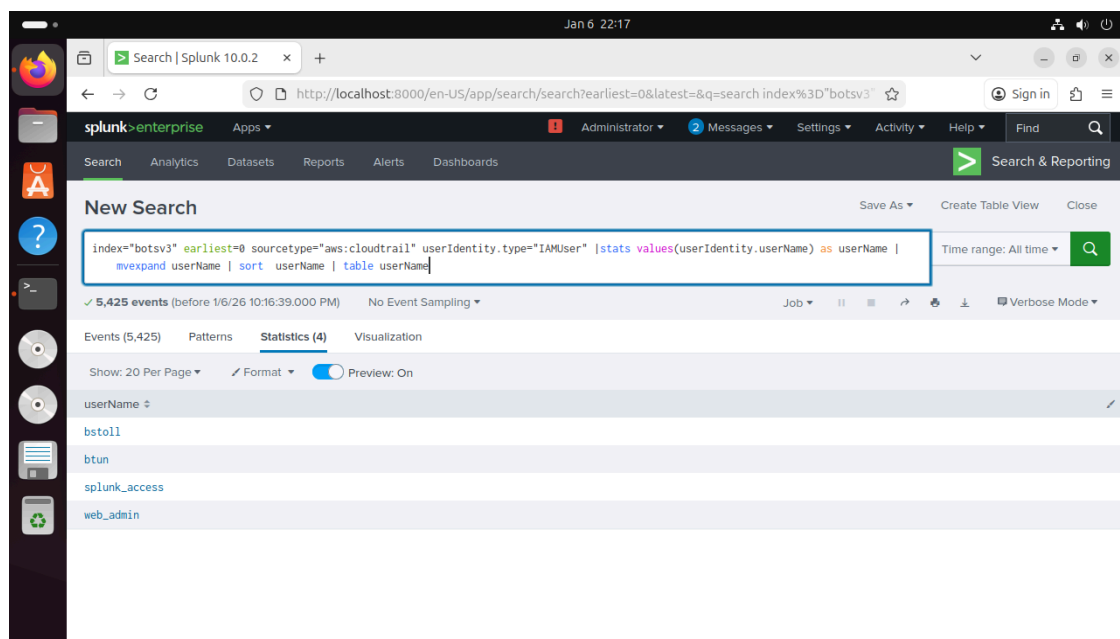
Post-ingestion verification confirmed that the data set's events were searchable so that queries could be done without sampling limitations.

## Guided Questions

The following eight questions were answered using targeted Splunk searches.

1. List out the IAM users that accessed an AWS service (successfully or unsuccessfully) in Frothly's AWS environment?

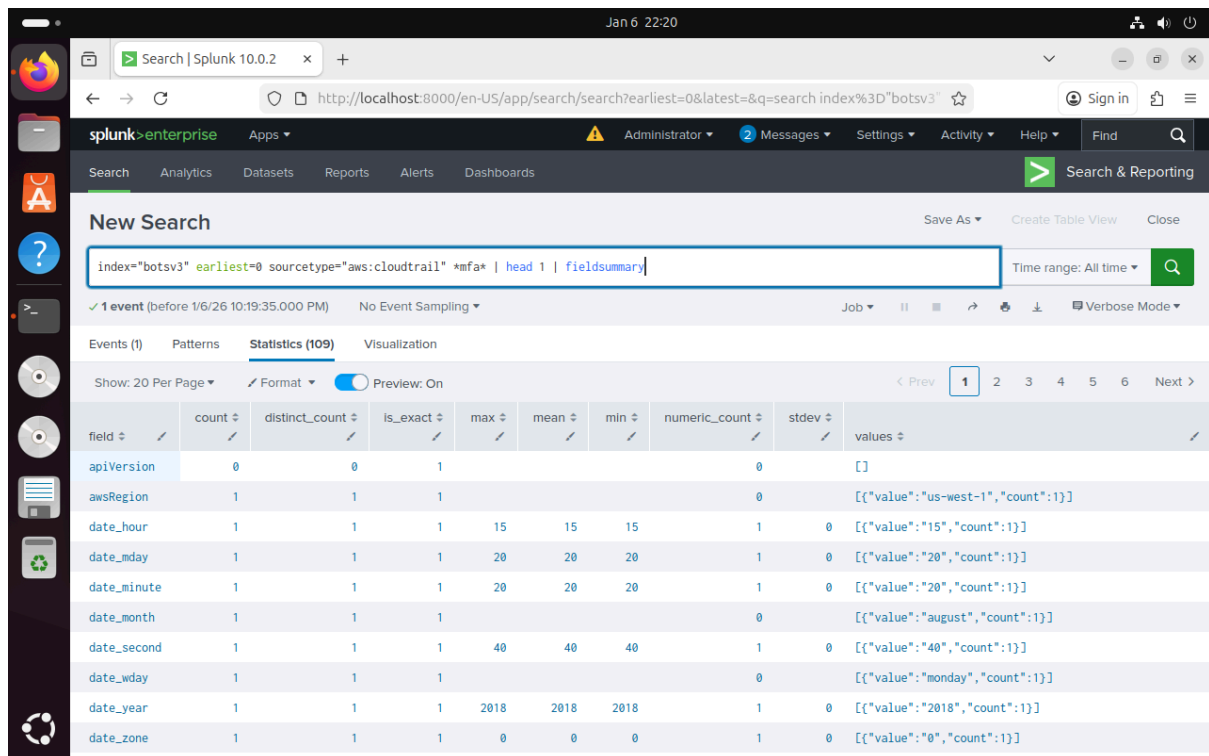
Answer: bstoll, btun, splunk\_access, web\_admin



2. What field would you use to alert that AWS API activity has occurred without MFA (multi-factor authentication)?

Answer: `userIdentity.sessionContext.attributes.mfaAuthenticated`

Any CloudTrail event can be inspected to reveal this field. When set to "false", it indicates non-MFA-authenticated activity.



Jan 6 22:20

Search | Splunk 10.0.2

http://localhost:8000/en-US/app/search/search?earliest=0&latest=&q=search index%3D"botsv3"

Search > enterprise Apps Administrator 2 Messages Settings Activity Help Find

Search Analytics Datasets Reports Alerts Dashboards Search & Reporting

New Search Save As Create Table View Close

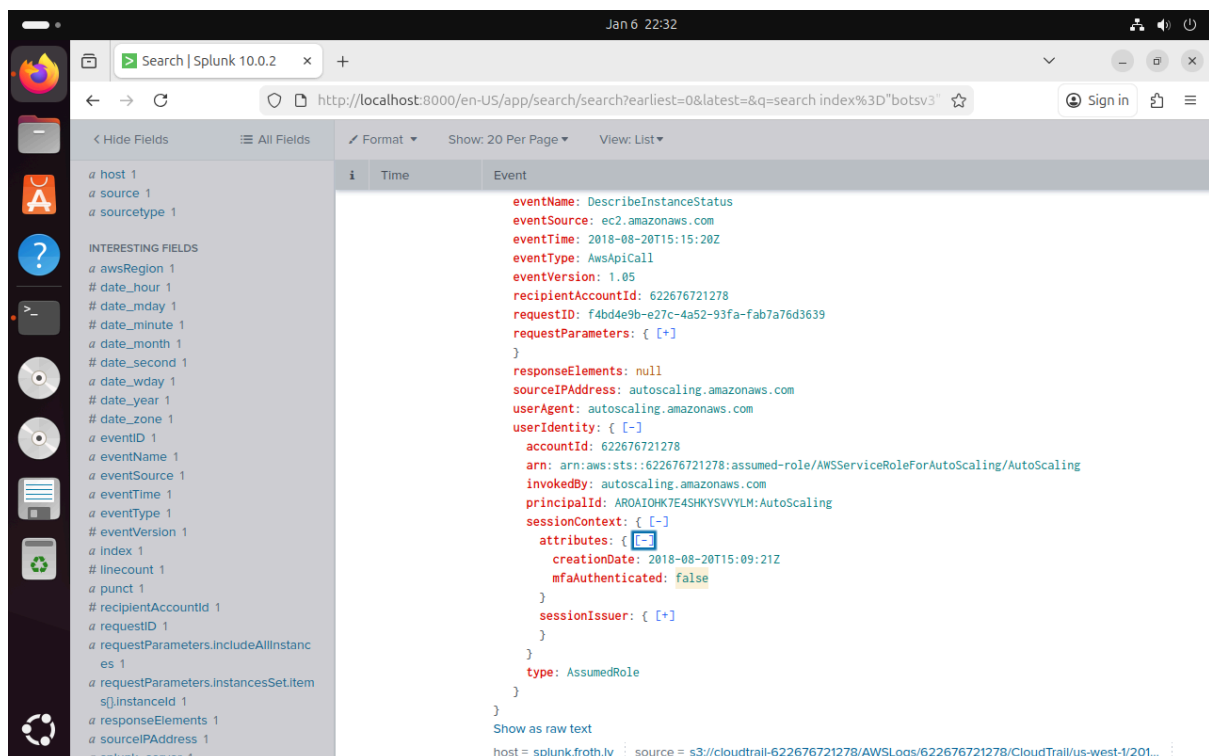
index="botsv3" earliest=0 sourcetype="aws:cloudtrail" \*mfa\* | head 1 | fieldsummary Time range: All time

1 event (before 1/6/26 10:19:35.000 PM) No Event Sampling Job

Events (1) Patterns Statistics (109) Visualization

Show: 20 Per Page Format Preview: On

field	count	distinct_count	is_exact	max	mean	min	numeric_count	stdev	values
apiVersion	0	0	1				0		["value": "us-west-1", "count": 1]
awsRegion	1	1	1				0		["value": "us-west-1", "count": 1]
date_hour	1	1	1	15	15	15	1	0	["value": "15", "count": 1]
date_mday	1	1	1	20	20	20	1	0	["value": "20", "count": 1]
date_minute	1	1	1	20	20	20	1	0	["value": "20", "count": 1]
date_month	1	1	1				0		["value": "august", "count": 1]
date_second	1	1	1	40	40	40	1	0	["value": "40", "count": 1]
date_wday	1	1	1				0		["value": "monday", "count": 1]
date_year	1	1	1	2018	2018	2018	1	0	["value": "2018", "count": 1]
date_zone	1	1	1	0	0	0	1	0	["value": "0", "count": 1]



Jan 6 22:32

Search | Splunk 10.0.2

http://localhost:8000/en-US/app/search/search?earliest=0&latest=&q=search index%3D"botsv3"

Search > enterprise Apps Administrator 2 Messages Settings Activity Help Find

Search Analytics Datasets Reports Alerts Dashboards Search & Reporting

New Search Save As Create Table View Close

index="botsv3" earliest=0 sourcetype="aws:cloudtrail" \*mfa\* | head 1 | fieldsummary Time range: All time

1 event (before 1/6/26 10:19:35.000 PM) No Event Sampling Job

Events (1) Patterns Statistics (109) Visualization

Show: 20 Per Page Format Preview: On

i	Time	Event
1	2018-08-20T15:15:20Z	<pre>eventSource: DescribeInstanceStatus eventSource: ec2.amazonaws.com eventTime: 2018-08-20T15:15:20Z eventType: AwsApiCall eventVersion: 1.05 recipientAccountID: 622676721278 requestID: f4bd4e9b-e27c-4a52-93fa-fab7a76d3639 requestParameters: { [+]} responseElements: null sourceIPAddress: autoscaling.amazonaws.com userAgent: autoscaling.amazonaws.com userIdentity: { [-]}   accountId: 622676721278   arn: arn:aws:sts::622676721278:assumed-role/AWSLambdaRole/AutoScaling   invokedBy: autoscaling.amazonaws.com   principalId: AROAI0HK7E4SHKYSVYLH:AutoScaling   sessionContext: { [-]}     attributes: { [-]}       creationDate: 2018-08-20T15:09:21Z       mfaAuthenticated: false     sessionIssuer: { [+]}   type: AssumedRole }</pre>

Show as raw text

host = splunk.froth.ly source = s3://cloudtrail-622676721278/AWSLogs/622676721278/CloudTrail/us-west-1/201...

3. What is the processor number used on the web servers?

Answer: E5-2676

The screenshot shows the Splunk search interface with the query `index="botsv3" earliest=0 sourcetype="hardware"`. The search results are displayed in a table view. The first event is from 8/20/18 at 3:26:25.000 PM, showing hardware details for a host. The processor number is highlighted as E5-2676.

Time	Event										
8/20/18 3:26:25.000 PM	<table border="1"><thead><tr><th>KEY</th><th>VALUE</th></tr></thead><tbody><tr><td>CPU_TYPE</td><td>Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz</td></tr><tr><td>CPU_CACHE</td><td>30720 KB</td></tr><tr><td>CPU_COUNT</td><td>2</td></tr><tr><td>HARD_DRIVES</td><td>xvda 8 GB;</td></tr></tbody></table>	KEY	VALUE	CPU_TYPE	Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz	CPU_CACHE	30720 KB	CPU_COUNT	2	HARD_DRIVES	xvda 8 GB;
KEY	VALUE										
CPU_TYPE	Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz										
CPU_CACHE	30720 KB										
CPU_COUNT	2										
HARD_DRIVES	xvda 8 GB;										

4. Bud accidentally makes an S3 bucket publicly accessible. What is the event ID of the API call that enabled public access?

Answer: ab45689d-69cd-41e7-8705-5350402cf7ac

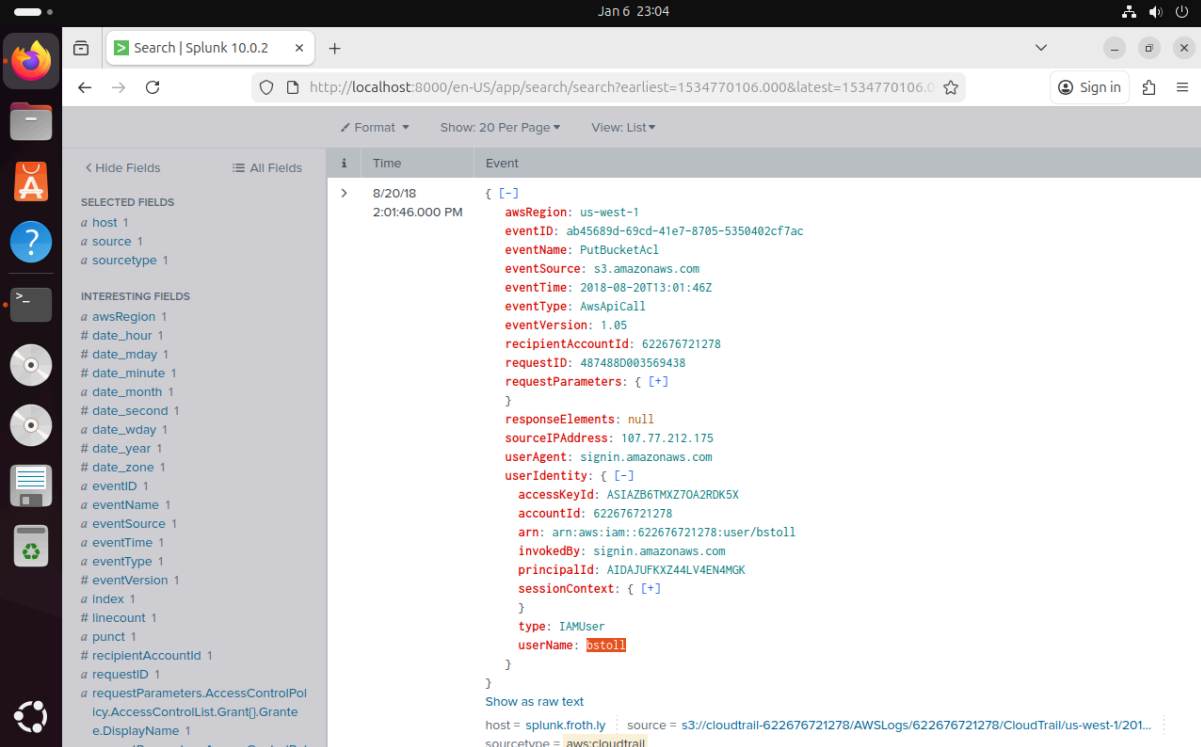
The screenshot shows the Splunk search interface with the query `index="botsv3" earliest=0 sourcetype="aws:cloudtrail" eventName="PutBucketACL" "AllUsers" | sort + _time | table _time eventID userIdentity .username requestParameters.bucketName`. The search results are displayed in a table view. The first event is from 2018-08-20 at 14:01:46, showing the event ID and the bucket name.

_time	eventID	useridentity.username	requestParameters.bucketName
2018-08-20 14:01:46	ab45689d-69cd-41e7-8705-5350402cf7ac		frothywebcode

The earliest event shows the PutBucketAcl call.

## 5. What is Bud's username

Answer: bstoll

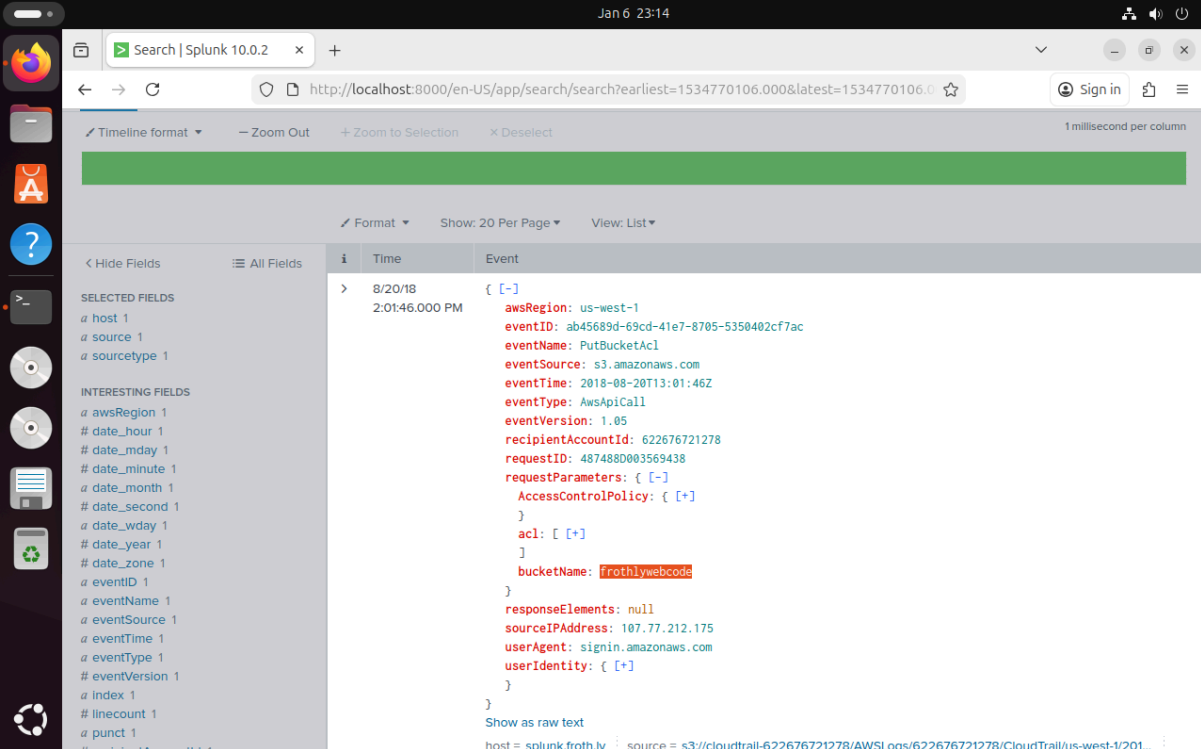


The screenshot shows the Splunk search interface with a single result. The event is an AWS CloudTrail log entry for a PutBucketAcl action. The user performing the action is identified as 'bstoll'.

Time	Event
8/20/18 2:01:46.000 PM	<pre>{   "awsRegion": "us-west-1",   "eventID": "ab45689d-69cd-41e7-8705-5350402cf7ac",   "eventName": "PutBucketAcl",   "eventSource": "s3.amazonaws.com",   "eventTime": "2018-08-20T13:01:46Z",   "eventType": "AwsApiCall",   "eventVersion": "1.05",   "recipientAccountId": "622676721278",   "requestID": "487488D003569438",   "requestParameters": {     "AccessControlPolicy": {       "AccessControlList": {         "Grantee": {           "displayName": "bstoll"         }       }     }   },   "responseElements": null,   "sourceIPAddress": "107.77.212.175",   "userAgent": "signin.amazonaws.com",   "userIdentity": {     "accessKeyId": "ASIAZB6TMXZ70A2RDK5X",     "accountId": "622676721278",     "arn": "arn:aws:iam::622676721278:user/bstoll",     "invokedBy": "signin.amazonaws.com",     "principalId": "AIDAJUFKXZ44LV4EN4MGK",     "sessionContext": {       "type": "IAMUser",       "userName": "bstoll"     }   } }</pre>

## 6. What is the name of the S3 bucket that was made publicly accessible?

Answer: frothywebcode

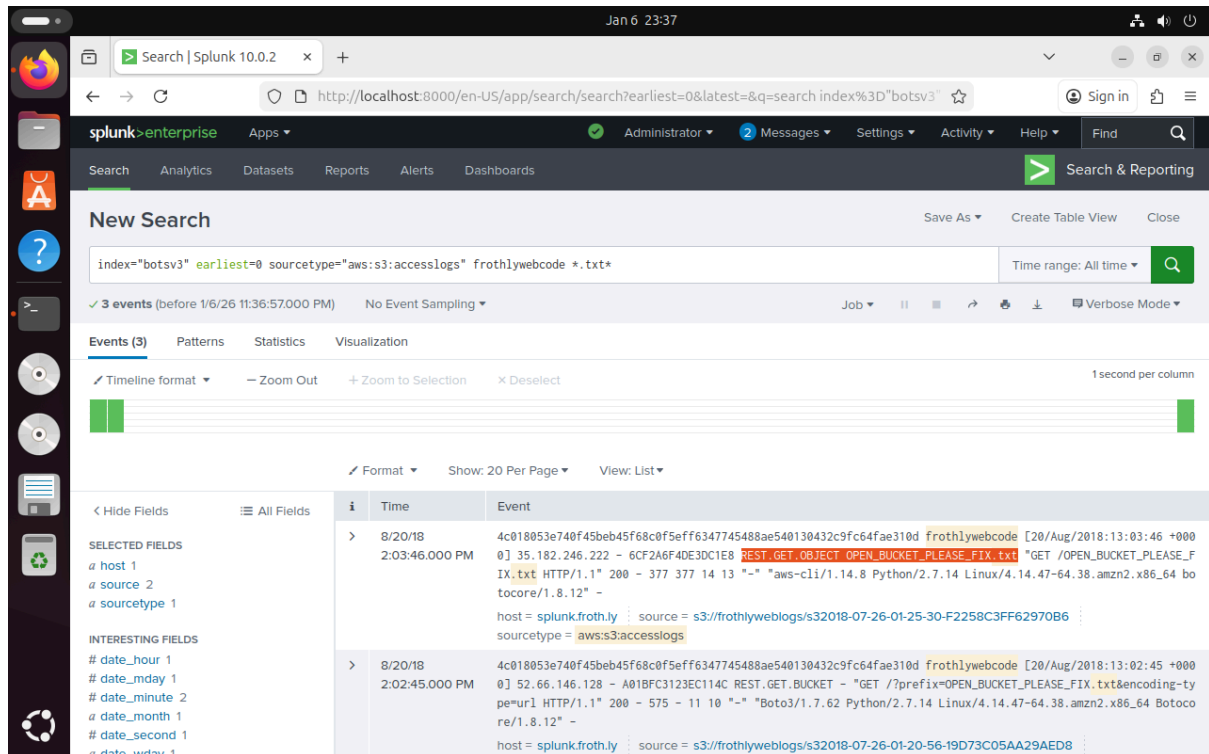


The screenshot shows the Splunk search interface with a single result. The event is an AWS CloudTrail log entry for a PutBucketAcl action. The bucket name is 'frothywebcode'.

Time	Event
8/20/18 2:01:46.000 PM	<pre>{   "awsRegion": "us-west-1",   "eventID": "ab45689d-69cd-41e7-8705-5350402cf7ac",   "eventName": "PutBucketAcl",   "eventSource": "s3.amazonaws.com",   "eventTime": "2018-08-20T13:01:46Z",   "eventType": "AwsApiCall",   "eventVersion": "1.05",   "recipientAccountId": "622676721278",   "requestID": "487488D003569438",   "requestParameters": {     "AccessControlPolicy": {       "AccessControlList": {         "Grantee": {           "displayName": "bstoll"         }       }     }   },   "responseElements": null,   "sourceIPAddress": "107.77.212.175",   "userAgent": "signin.amazonaws.com",   "userIdentity": {     "accessKeyId": "ASIAZB6TMXZ70A2RDK5X",     "accountId": "622676721278",     "arn": "arn:aws:iam::622676721278:user/bstoll",     "invokedBy": "signin.amazonaws.com",     "principalId": "AIDAJUFKXZ44LV4EN4MGK",     "sessionContext": {       "type": "IAMUser",       "userName": "bstoll"     }   } }</pre>

7. What is the name of the text file that was successfully uploaded into the S3 bucket while it was publicly accessible?

Answer: OPEN\_BUCKET\_PLEASE\_FIX.txt

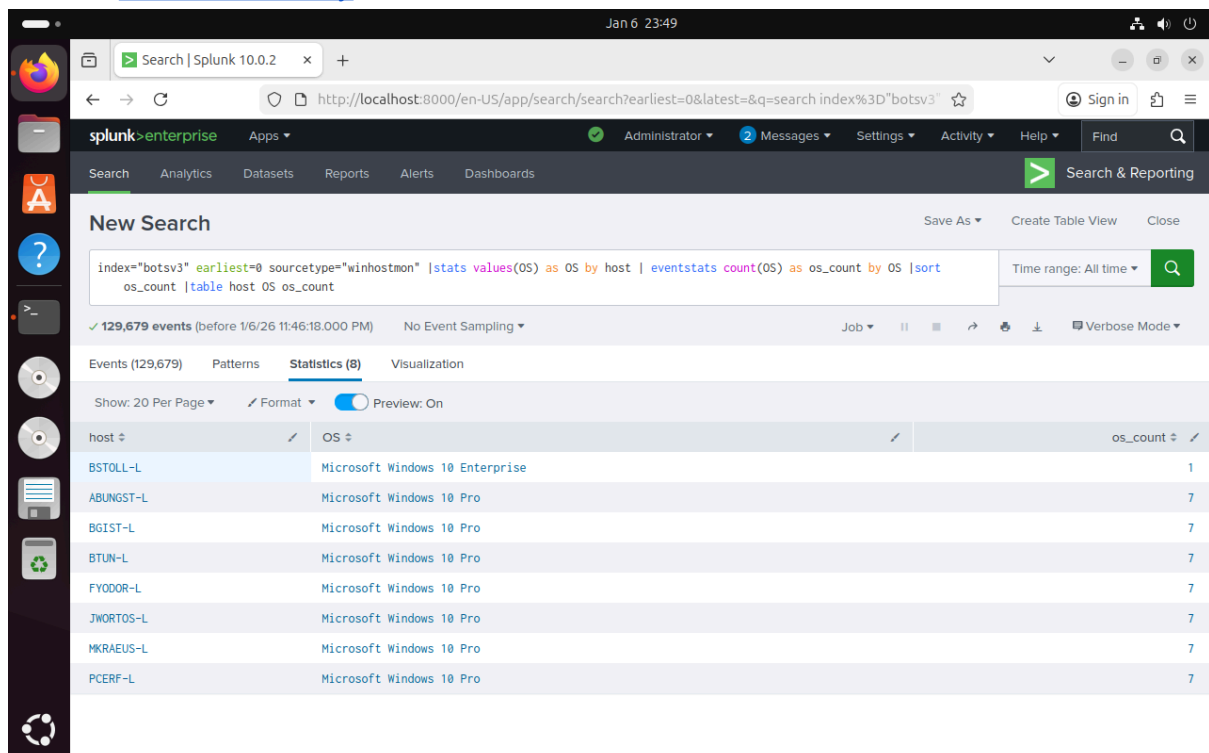


The screenshot shows a Splunk search interface with the following search query: `index="botsv3" earliest=0 sourcetype="aws:s3:accesslogs" frothywebcode *.txt*`. The results show two events from 8/20/18. The first event, at 2:03:46.000 PM, shows a successful upload of a file named `OPEN_BUCKET_PLEASE_FIX.txt` to an S3 bucket. The second event, at 2:02:45.000 PM, shows a GET request for the same file. The source of both events is `splunk.froth.ly`.

This confirms external exfiltration-style activity while the bucket was public.

8. What is the FQDN of the endpoint that is running a different Windows operating system edition than the others?

Answer: [BSTOLL-L.froth.ly](http://BSTOLL-L.froth.ly)



The screenshot shows a Splunk search interface with the following search query: `index="botsv3" earliest=0 sourcetype="winhostmon" | stats values(OS) as OS by host | eventstats count(OS) as os_count by OS | sort os_count | table host OS os_count`. The results show a table with 8 rows, each representing a different host. The hosts are: `BSTOLL-L`, `ABUNGST-L`, `BGIST-L`, `BTUN-L`, `FYODOR-L`, `JWORTOS-L`, `MKRAEUS-L`, and `PCERF-L`. All hosts are running Microsoft Windows 10 Pro, except for `BSTOLL-L`, which is running Microsoft Windows 10 Enterprise.

The top screenshot shows the Splunk Search interface with the search query: `index="botsv3" earliest=0 host="BSTOLL-L" sourcetype="wineventlog:"`. The search results are displayed in a table format, showing the event details for the host `BSTOLL-L` on 8/20/2018 at 03:17:58 AM. The event details include `LogName=Security`, `SourceName=Microsoft Windows security auditing`, `EventCode=4689`, and `EventType=0`.

The bottom screenshot shows the 'Event Actions' table, which lists the fields and values for the selected event. The table has columns for Type, Field, Value, and Actions. The selected fields are `host` (value: `BSTOLL-L`), `source` (value: `WinEventLog:Security`), and `sourcetype` (value: `WinEventLog:Security`). The event details are also listed, including `Account_Domain` (value: `AzureAD`), `Account_Name` (value: `BudStoll`), `ComputerName` (value: `BSTOLL-L.froth.ly`), `EventCode` (value: `4689`), `EventType` (value: `0`), `Exit_Status` (value: `0x0`), `Keywords` (value: `Audit Success`), `LogName` (value: `Security`), `Logon_ID` (value: `0x4B339`), and `Message` (value: `A process has exited. Subject: Security ID: AzureAD\BudSt`).

## Incident Narrative

Questions 4–7 form a timeline of an insider mistake leading to data exposure. User bstoll (Bud) executed a PutBucketAcl API call granting "AllUsers" read/write access to the frothlywebcode bucket. Shortly after, an external entity uploaded OPEN\_BUCKET\_PLEASE\_FIX.txt, demonstrating successful exploitation. The fault lies in a lack of least-privilege bucket policies and non-continuous monitoring of CloudTrail and S3 access logs.

# SOC and Incident Handling Reflection

In a Security Operations Centre (SOC), incident detection and response typically follow a tiered analyst structure to ensure efficient handling of alerts at scale.

Tier 1 analysts serve as the first line of defence, monitoring dashboards and SIEM tools like Splunk for initial alerts. In the Frothly incident, they would detect anomalous CloudTrail events such as the PutBucketAcl API call granting public access, or unusual S3 access log entries indicating external uploads. Their role involves: validating the alert, picking out contextual data (e.g., user details from IAM), and escalating confirmed incidents to higher tiers while documenting initial findings.

Tier 2 analysts would conduct deeper investigation, performing the type of correlation demonstrated in this analysis, linking the configuration change by user bstoll to the subsequent unauthorised upload of OPEN\_BUCKET\_PLEASE\_FIX.txt. Then advanced Splunk searches would be carried out across sourcetypes (CloudTrail, S3 access logs, endpoint logs) and reconstruct timelines.

Tier 3 analysts would focus on root cause analysis, identifying whether the misconfiguration resulted from human error, lack of training or a potential insider threat and recommend long-term remediation. This tiered approach aims to meet common practices of a Computer Security Incident Response Team (CSIRT). Structured escalation ensures rapid containment while preserving evidence for forensic review. The use of Splunk as a unified platform for log ingestion, search, and correlation are key components of a SOC monitoring infrastructure.

## Conclusion

This analysis of the BOTSV3 dataset has successfully uncovered a security incident involving the unintentional exposure of an S3 bucket (frothlywebcode) to public access, triggered by a PutBucketAcl API call from the IAM user bstoll. This misconfiguration enabled an external actor to upload the file OPEN\_BUCKET\_PLEASE\_FIX.txt shortly afterward, demonstrating clear exploitation and a potential data exfiltration risk.

Key lessons from this investigation include the dangers of insufficient access controls in AWS services and the importance of proactive monitoring of API calls. When not properly managed, even seemingly minor actions by authorised users can lead to significant exposure. Strict policies and automated safeguards can help alleviate this risk. Centralised logging would have given the ability to correlate events across CloudTrail and S3 access logs, improving situational awareness.

For Frothly's Security Operations Centre (SOC) strategy, this case should draw attention to the importance of layered detection mechanisms and rapid response capabilities. Implementing real-time alerting on high-risk API actions would enable Tier 1 analysts to flag anomalies early, while tools like Splunk support Tier 2/3 investigations. For response, adopting automated playbooks would reduce mean time to remediate (MTTR), such as immediate notifications to administrators upon detection of public bucket changes.

To prevent recurrence and strengthen overall resilience, Frothly should adopt the following recommendations:

- Enforce multi-factor authentication (MFA) for all IAM users to mitigate risks from compromised credentials.
- Enable S3 Block Public Access at the account level to prevent buckets from being made publicly accessible, regardless of individual settings.
- Configure CloudTrail-based alerts for sensitive events such as PutBucketAcl and PutBucketPolicy to ensure immediate visibility.
- Conduct regular automated audits of bucket permissions using tools like AWS Config or third-party scanners.
- Deploy automated solutions, such as AWS Lambda functions triggered by CloudWatch Events, to revert unauthorised public access configurations instantly.
- Conduct employee training in order to lower the risks of breaches that come as a result of human error.

By addressing these areas, Frothly can significantly enhance its incident detection and response abilities, strengthening their SOC operation.