COMP3010HK Coursework 2 - BOTSv3 Incident Analysis Report

1. Introduction

The BOTSv3 (Boss of the SOC Version 3) dataset simulates a multi-stage security incident at Frothly, a fictitious craft beer company, reflecting operational scenarios commonly encountered in real-world Security Operations Centers (SOCs). The dataset integrates diverse log sources, including network traffic, endpoint telemetry, cloud service logs from AWS and Azure, and email communications. This multi-source integration enables comprehensive security incident analysis through the Splunk SIEM platform.

This investigation analyzes security incidents within the Frothly environment, with particular emphasis on identifying AWS misconfigurations, anomalous activities, and potential intrusion behaviors. The analysis focuses on AWS Identity and Access Management (IAM), S3 bucket security, and Windows endpoint configurations. Through structured Splunk queries and evidence-based analytical processes, this report demonstrates practical applications of SOC methodologies across detection, analysis, and incident response activities.

The scope of this analysis addresses eight AWS and endpoint-related questions within BOTSv3's 200-level investigation set. Rather than providing exhaustive dataset coverage, the investigation prioritizes authentic SOC workflows, through the effective use of Splunk Search Processing Language (SPL) to support incident analysis. This report systematically documents the investigation process and interprets findings from a SOC operational perspective, presenting actionable insights to inform improvement in cloud security monitoring, detection strategies, and incident response capabilities.

GitHub: https://github.com/wongoining/COMP3010HK_2_oi

2. SOC Roles & Incident Handling Reflection

Security Operations Centres commonly operate under a tiered model to manage incidents through structured escalation and role separation, and this operational logic is reflected in the BOTSv3 investigation. During the initial stages of analysis, the identification of anomalous AWS and endpoint activities through Splunk queries aligns with monitoring and triage functions typically associated with lower-tier SOC operations, where the primary objective is to recognise deviations from normal behaviour rather than to fully assess impact.

As the investigation progressed, the identification of S3 bucket misconfigurations and abnormal IAM activity required deeper analysis and correlation across multiple data sources, including CloudTrail and endpoint logs. This escalation reflects higher-tier SOC responsibilities, where analysts focus on determining incident scope, assessing potential impact, and supporting informed decision-making. The subsequent development of remediation and preventive recommendations, such as access control adjustments and credential review, represents a strategic response planning function rather than direct technical containment.

From an incident handling perspective, the BOTSv3 exercise primarily emphasises detection and analysis stages, with response activities limited to planning and recommendation rather than live remediation. This approach mirrors real-world SOC practices, where analytical validation and structured decision-making precede operational response. Overall, the investigation demonstrates how SOC roles and incident handling methodologies are practically applied within a realistic analytical context, highlighting the value of escalation, coordination, and reflective analysis in strengthening organisational security posture.

3. Installation & Data Preparation

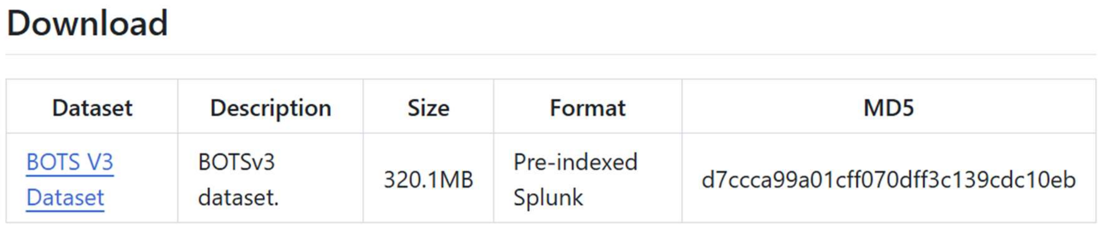
The investigation environment was established using Splunk Enterprise 10.0.2 deployed on an Ubuntu 24.04.3 LTS platform, providing a controlled and realistic setting that reflects a typical Security Operations Centre (SOC) analytics infrastructure. This configuration supports centralized log aggregation, query-driven investigation, and evidence preservation, which are core capabilities required for professional security analysis.

3.1 Environment Setup

Splunk Enterprise was installed alongside the required applications and add-ons specified by the BOTSv3 documentation, ensuring compatibility with the dataset and consistency with the intended analytical environment. Default index configurations were retained to align with the assumptions embedded within the BOTSv3 design.

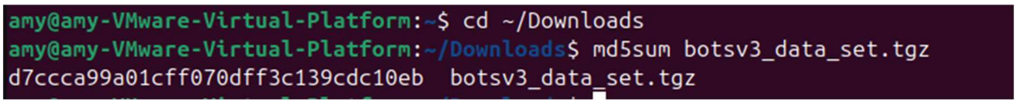
3.2 Dataset Acquisition and Validation

The BOTSv3 dataset was obtained directly from the official Splunk GitHub repository (https://github.com/splunk/botsv3). Prior to deployment, dataset integrity was verified using the published MD5 hash to confirm that the downloaded archive was complete and unaltered. This verification step is critical in SOC operations to ensure evidence integrity and maintain chain of custody throughout the investigation process.



Download

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



```
amy@amy-VMware-Virtual-Platform:~$ cd ~/Downloads
amy@amy-VMware-Virtual-Platform:~/Downloads$ md5sum botsv3_data_set.tgz
d7ccca99a01cff070dff3c139cdc10eb  botsv3_data_set.tgz
```

3.3 Data Ingestion Process

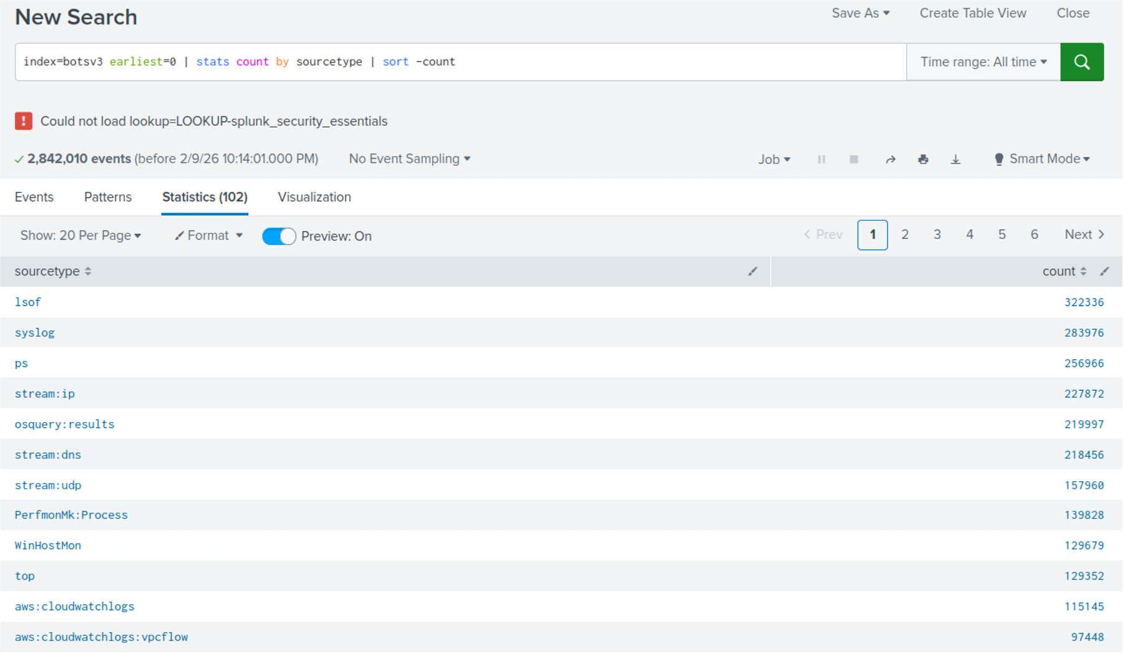
As the BOTSv3 dataset is distributed in a pre-indexed format, no custom indexing, parsing, or data normalisation was required. Following extraction of the archive, an initial deployment issue was identified in which the physical location of the pre-indexed buckets was not fully aligned with Splunk’s standard index storage paths. This resulted in the dataset not being immediately queryable despite successful installation.

To address this, a deployment-level correction was applied to align the BOTSv3 index configuration with Splunk’s standard storage layout. The index paths were updated to reference the default Splunk database location, and the pre-indexed buckets were relocated accordingly. This adjustment ensured that Splunk could correctly recognise and load the existing indexed data without altering the dataset structure or performing re-indexing.

Once the alignment was completed and Splunk restarted, the BOTSv3 data became fully accessible through the predefined index. This approach preserves the integrity and consistency of the official pre-indexed dataset while ensuring reliable ingestion and queryability within the investigation environment.

3.4 Post-Ingestion Validation

Comprehensive validation was conducted to confirm successful data ingestion and environment readiness.



```
index=botsv3 earliest=0 | stats count by sourcetype | sort -count
```

Could not load lookup=LOOKUP-splunk_security_essentials

✓ 2,842,010 events (before 2/9/26 10:14:01.000 PM) No Event Sampling ▾

sourcetype	count
lsof	322336
syslog	283976
ps	256966
stream:ip	227872
osquery:results	219997
stream:dns	218456
stream:udp	157960
PerfmonMk:Process	139828
WinHostMon	129679
top	129352
aws:cloudwatchlogs	115145
aws:cloudwatchlogs:vpcflow	97448

Validation Results:

- Total Events: 2,842,010
- Temporal Coverage: 2018-08-19 to 2018-08-21 (consistent with simulated incident period)
- Primary Sourcetypes:
    - WinHostMon: 129,679 events
    - aws:cloudwatchlogs: 115,145 events
    - aws:cloudwatchlogs:vpcflow: 97,448 events
    - WinEventLog: 48,101 events
    - aws:rds:audit: 35,192 events
    - aws:cloudtrail: 6,571 events
    - Additional supporting sourcetypes confirmed present

These sourcetypes were prioritized based on their direct relevance to the AWS security and Windows endpoint investigation focus. The dataset also includes extensive system monitoring data (steam:*, lsof, syslog, ps) providing comprehensive coverage across network, endpoint, and cloud infrastructure domains.

All expected data sources were verified as present and temporally consistent with, confirming the environment was ready for analytical operations.

3.5 SOC Infrastructure Alignment

From a SOC infrastructure perspective, this deployment mirrors operational practices in which analysts work with trusted, standardised data sources within a centralised SIEM platform. The validated environment supports query-driven investigation, cross-source correlation, and evidence preservation, providing a robust foundation for the subsequent BOTSv3 incident analysis.

4. Analysis

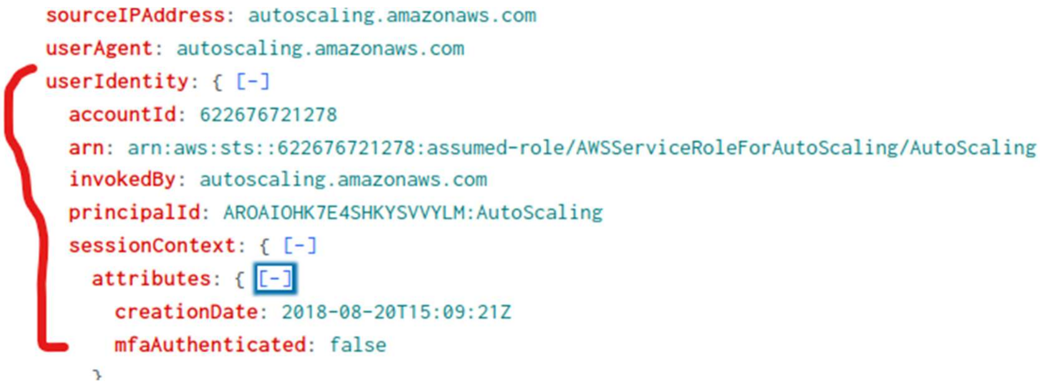
This section presents a structured technical analysis of the incident identified within the BOTSv3 environment. The investigation follows a Security Operations Centre (SOC) analytical workflow, progressing from identity visibility and authentication context, to cloud configuration analysis, impact validation, and endpoint attribution. All findings are derived from log-based evidence and corroborated through cross-source correlation.

The primary data sources used in this analysis include AWS CloudTrail logs, S3 access logs, and Windows endpoint monitoring data. By correlating identity activity, access control changes, and endpoint telemetry, the investigation establishes a coherent and technically supported incident narrative.

4.1 Identity and Authentication Context

The investigation began with an examination of AWS CloudTrail logs to establish visibility into identity activity within the environment. Enumerating IAM users based on recorded API interactions provided an initial behavioural baseline. Four IAM users were observed interacting with AWS services during the relevant timeframe.

Subsequent analysis focused on authentication context, specifically the presence or absence of multi-factor authentication (MFA) during API execution. CloudTrail session attributes indicated that certain API calls were performed without MFA enforcement. While no evidence of credential compromise was identified, the absence of MFA reduces the security assurance level associated with privileged operations. In cloud environments, insufficient authentication controls can amplify the impact of configuration errors or misuse of legitimate credentials.



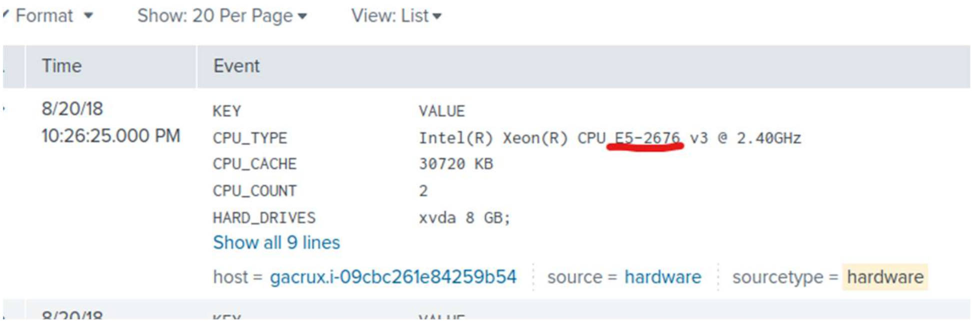
```
sourceIPAddress: autoscaling.amazonaws.com
userAgent: autoscaling.amazonaws.com
userIdentity: { [-]
  accountId: 622676721278
  arn: arn:aws:sts::622676721278:assumed-role/AWSServiceRoleForAutoScaling/AutoScaling
  invokedBy: autoscaling.amazonaws.com
  principalId: AROAIOHK7E4SHKYSVVYLM:AutoScaling
  sessionContext: { [-]
    attributes: { [-]
      creationDate: 2018-08-20T15:09:21Z
      mfaAuthenticated: false
    }
```

Establishing this identity and authentication baseline provided contextual grounding for evaluating later security-sensitive actions.

4.2 Infrastructure Context

To support environmental validation, system telemetry data was reviewed to understand the operational characteristics of the environment. Hardware information indicated that relevant systems were operating on Intel Xeon E5-series processors, consistent with enterprise or cloud-hosted infrastructure.

Although this information does not indicate malicious activity, it confirms that observed behaviours occurred within a realistic enterprise-grade operating context. This contextual understanding supports the credibility of subsequent cross-domain correlations.

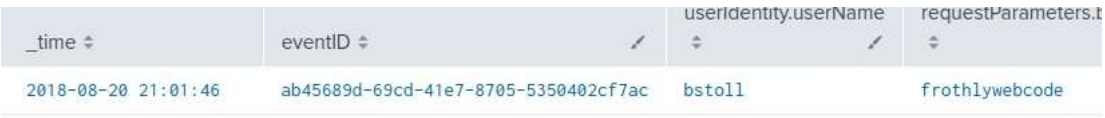


Time	Event
8/20/18 10:26:25.000 PM	KEY VALUE CPU_TYPE Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz CPU_CACHE 30720 KB CPU_COUNT 2 HARD_DRIVES xvda 8 GB; Show all 9 lines host = gacrux.i-09cbc261e84259b54 source = hardware sourcetype = hardware
8/20/18	KEY VALUE

4.3 S3 Access Control Modification and Exposure

The core security event identified in this investigation involved a modification to an S3 bucket's access control configuration. CloudTrail logs were analysed for the PutBucketAcl API action, which is used to modify a bucket's Access Control List (ACL). ACLs define which principals may perform read, write, or administrative operations on an S3 bucket (Amazon Web Services, n.d.). If misconfigured, ACL changes can expose resources to unintended entities, including the global "AllUsers" group.

The analysis revealed that the IAM user bstoll executed a PutBucketAcl operation affecting the S3 bucket frothlywebcode. Inspection of the request parameters confirmed that the ACL modification granted access permissions in a manner that resulted in public accessibility. Specifically, the configuration permitted access beyond intended administrative boundaries, effectively exposing the bucket.



_time ⬍	eventID ⬍	userIdentity.userName ⬍	requestParameters.b ⬍
2018-08-20 21:01:46	ab45689d-69cd-41e7-8705-5350402cf7ac	bstoll	frothlywebcode

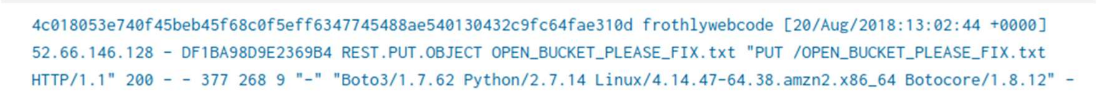
It is important to note that PutBucketAcl modifies bucket-level ACL settings and does not alter bucket policies. The exposure identified in this case was attributable to the ACL configuration itself.

The event originated from an authenticated user context, indicating that the exposure resulted from a configuration error rather than an external unauthorised intrusion. At this stage, the investigation transitioned from identifying a configuration change to assessing its operational impact.

4.4 Impact Verification Through S3 Access Logs

To determine whether the exposed bucket was actively accessed, S3 access logs were analysed for object-level operations following the ACL modification. The review identified successful object upload activity associated with the affected bucket during the exposure window.

A file named OPEN_BUCKET_PLEASE_FIX.txt was written to the bucket. This upload confirms that the public accessibility was not merely theoretical but was actively exploited. The presence of this file demonstrates that the misconfiguration resulted in unauthorised interaction with the resource.



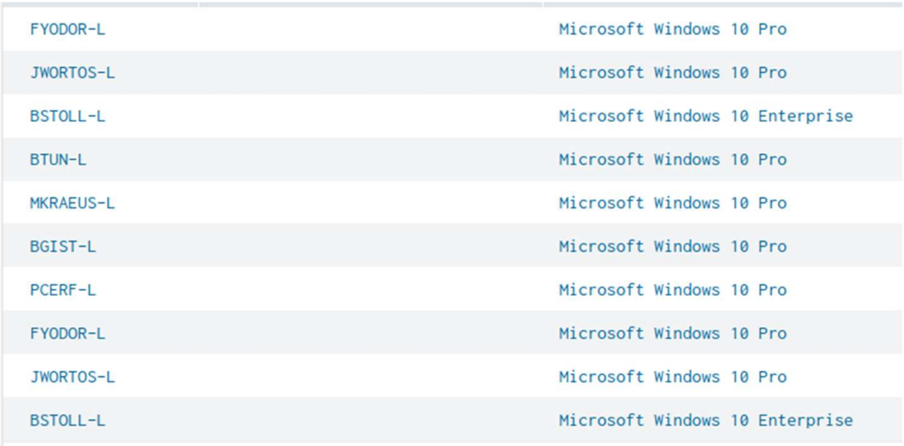
```
4c018053e740f45beb45f68c0f5eff6347745488ae540130432c9fc64fae310d frothlywebcode [20/Aug/2018:13:02:44 +0000]
52.66.146.128 - DF1BA98D9E2369B4 REST.PUT.OBJECT OPEN_BUCKET_PLEASE_FIX.txt "PUT /OPEN_BUCKET_PLEASE_FIX.txt
HTTP/1.1" 200 - - 377 268 9 "-" "Boto3/1.7.62 Python/2.7.14 Linux/4.14.47-64.38.amzn2.x86_64 Botocore/1.8.12" -
```

The analysis does not attribute the upload to a specific external actor; however, it establishes that the exposure materially increased risk and allowed unintended object-level operations. This progression from configuration change to verified impact demonstrates the practical consequences of the misconfiguration.

4.5 Endpoint Attribution and Correlation

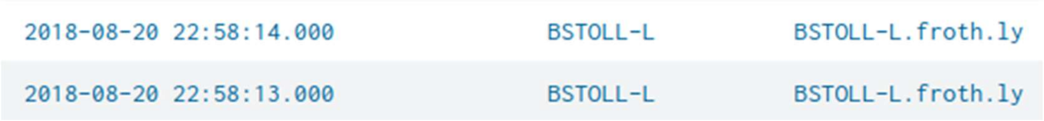
The final stage of the investigation focused on identifying the endpoint associated with the configuration change. Windows endpoint monitoring data (winhostmon) was analysed to identify systems operating under differing configurations.

The endpoint BSTOLL-L was identified as the only system running Windows 10 Enterprise, while other endpoints in the environment were running Windows 10 Pro, making it an environmental outlier.



FYODOR-L	Microsoft Windows 10 Pro
JWORTOS-L	Microsoft Windows 10 Pro
BSTOLL-L	Microsoft Windows 10 Enterprise
BTUN-L	Microsoft Windows 10 Pro
MKRAEUS-L	Microsoft Windows 10 Pro
BGIST-L	Microsoft Windows 10 Pro
PCERF-L	Microsoft Windows 10 Pro
FYODOR-L	Microsoft Windows 10 Pro
JWORTOS-L	Microsoft Windows 10 Pro
BSTOLL-L	Microsoft Windows 10 Enterprise

Further correlation with Windows event logs confirmed that the fully qualified domain name (FQDN) of this endpoint was BSTOLL-L.froth.ly. By aligning endpoint data with CloudTrail activity attributed to user bstoll, the investigation established a technically supported association between the administrative workstation and the PutBucketAcl operation affecting the S3 bucket.



2018-08-20 22:58:14.000	BSTOLL-L	BSTOLL-L.froth.ly
2018-08-20 22:58:13.000	BSTOLL-L	BSTOLL-L.froth.ly

Although the dataset does not contain host-level forensic artefacts (such as process execution records explicitly invoking AWS CLI commands), the correlation across identity logs, access control modification events, and endpoint characteristics provides sufficient evidential support to associate the administrative workstation with the S3 ACL misconfiguration.

5. Conclusion

This investigation identified a cloud security incident arising from an S3 access control misconfiguration. An authenticated IAM user executed a PutBucketAcl operation that rendered the bucket frothlywebcode publicly accessible. Subsequent analysis of S3 access logs confirmed unauthorised file upload activity, demonstrating that the exposure was operationally exploitable rather than theoretical. Correlation of identity logs, API activity, and endpoint telemetry provided a technically supported association between the administrative workstation BSTOLL-L.froth.ly and the S3 ACL modification.

The root cause was a configuration error performed under legitimate credentials, highlighting the security risks posed by insufficient governance over privileged operations. The absence of enforced multi-factor authentication (MFA) and real-time monitoring increased the potential impact of the misconfiguration.

From a SOC perspective, this case underscores the importance of cross-source log correlation and proactive monitoring of high-risk API actions. Strengthening access governance, enforcing MFA, and implementing timely detection controls are essential to reducing the likelihood and impact of similar cloud misconfiguration incidents.

Reference:

Amazon Web Services. (n.d.-a). *Access control list (ACL) overview*. Amazon S3 User Guide. https://docs.aws.amazon.com/AmazonS3/latest/userguide/acl-overview.html

Amazon Web Services. (n.d.-b). *PutBucketAcl – Amazon Simple Storage Service API reference*. https://docs.aws.amazon.com/AmazonS3/latest/API/API_PutBucketAcl.html

Amazon Web Services. (n.d.-c). *AWS CloudTrail user guide*. https://docs.aws.amazon.com/pdfs/awscloudtrail/latest/userguide/awscloudtrail-ug.pdf

Splunk. (n.d.). *Boss of the SOC (BOTS) version 3 dataset*. GitHub. https://github.com/splunk/botsv3

Appendix: Generative AI Declaration

Student Declaration of AI Tool use in this Assessment

Please indicate your level of usage of generative AI for this assessment - please tick the appropriate category(s).

If the "Assisted Work" or "Partnered Work" category is selected, please expand on the usage and in which elements of the assignment the usage refers to.

Solo Work	**S1 - Generative AI tools have not been used for this assessment.**	☐
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.	☐
	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.	☒
	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.	☐
	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.	☐
	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.	☒
	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.	☐
	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.	
	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.	□
	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.	□
	A10 - Data Analysis and Visualization Guidance AI tools can help suggest ways to analyse datasets or visualize results (e.g. 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:	□

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: Generative AI was used to support sentence refinement, structural organisation, grammar consistency checks, and to provide guidance on improving SPL query structure. All investigative queries were independently executed, validated, and interpreted by the student, and all findings reflect the student’s own analytical work.	☒

Please provide details of AI usage and which elements of the coursework this relates to: Generative AI tools were used to assist with planning the structure of the report and refining language for clarity and conciseness. All technical analysis, packet capture interpretation, identification of indicators of compromise, and conclusions were conducted independently by the student. No generative AI tools were used to generate technical findings, analyse packet capture data, or fabricate evidence.

I understand that the ownership and responsibility for the academic integrity of this submitted assessment falls with me, the student.	☒
I confirm that all details provide above are an accurate description of how AI was used for this assessment.	☒