National College of Ireland

THRIVE: A Structured, CTI-Prioritised Threat Hunting Methodology for Security Operations Centres (SOCs)

MSc Research Project
MSc Cybersecurity

Amruta Pandhare
Student ID: x23269227

School of Computing
National College of Ireland

Supervisor: Kamil Mahajan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Amruta Anant Pandhare
Student ID:	x23269227
Programme:	MSc Cybersecurity Year: 2024-2025
Module:	MSc Research Project
Supervisor:	Kamil Mahajan
Submission Due Date:	01/09/2025
Project Title:	THRIVE: A Structured, CTI-Prioritised Threat Hunting Methodology for Security Operations Centres (SOCs)
Word Count:	8458 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Amruta Anant Pandhare ……

Date: 01/09/2025 ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

THRIVE: A Structured, CTI-Prioritised Threat Hunting Methodology for Security Operations centres (SOCs)

Amruta Pandhare
x23269227

Abstract

Proactive threat hunting remains essential for identifying sophisticated adversaries and minimizing detection gaps. While existing frameworks such as Sqrrl's Hunting Maturity Model, TaHiTI, and PEAK offer structured approaches, they encounter limitations in scalability, intelligence integration, and measurable assessment. This thesis introduces THRIVE; a nine-phase methodology aimed at operationalizing cyber threat intelligence (CTI) and standardizing detection engineering within Security Operations centres (SOCs). THRIVE incorporates a CTI scoring model, hypothesis development aligned with MITRE ATT&CK techniques, collaborative peer review, validation through red-blue team simulations, and the transformation of hunts into automated detections. Dashboards enhance visibility into coverage, validated detections, and resolved visibility gaps. Developed using the Design Science Research Methodology (DSRM), THRIVE was applied during a 12-week SOC internship and evaluated through case studies. Evaluation highlights THRIVE's ability to translate CTI into validated detection rules, enforce accountability through peer review, and provide metrics for leadership. Comparative analysis confirms its originality in combining intelligence-led prioritization with measurable operational outcomes, advancing the maturity of SOC hunting.

Keywords: Threat Hunting, Cyber Threat Intelligence, Security Operation Centres, MITRE ATT&CK, Security Information and Event Management (SIEM), etc.

1 Introduction

Cybersecurity has experienced a significant evolution in the last few years, driven by the increased level of sophistication of attackers, the commercialization of offensive technologies, and the mainstreaming of hybrid and cloud-native environments. Although many organizations have invested huge efforts into security monitoring tools, including Security Information and Event Management (SIEM), Endpoint Detection and Response (EDR), and Intrusion Detection Systems (IDS), a substantial percentage of attacks go unnoticed over considerable periods of time. This time period, often referred to as attacker dwell time, is a critical point of weakness in the current defence systems (Alevizos and Dekker, 2024). The 2025 M-Trends report by (Mandiant, 2025), for example, says that the global median dwell time rose to 11 days from 10 days in 2023. Global median dwell time was 26 days when external entities notified, 5 days when adversaries notified (notably in ransomware cases), and 10 days when organizations discovered malicious activity internally, highlighting that reactive detection, alone, is not enough to prevent breaches. To address this, the security community has increasingly embraced the practice of proactive threat hunting which is a human-led process of hypothesizing,

investigating, and uncovering threats that bypass automated detections. To reduce the time needed to detect as well as respond to emerging threats, threat hunting seeks to identify novel indicators of compromise via closing visibility gaps. According to the SANS 2025 Threat Hunting Survey (Lemon, 2025), over 70% of organizations do still rely on unstructured or internally developed approaches. However, this reliance remains still high. The lack of standardized, repeatable, and adaptable methodologies results in inefficiencies, inconsistent outcomes, and limited long-term value from hunting efforts. Several frameworks have emerged to bring structure to the threat hunting lifecycle along with MITRE's TTP-Based Hunting, Sqrrl's Hunting Maturity Model, TaHiTI, and PEAK. These methodologies often make assumptions regarding an organization's security maturity, data availability or technical capacity. They either focus narrowly on specific phases (e.g., hypothesis development or CTI integration) or lack clear operational guidance for peer review, validation, documentation, and hunt metrics. As a result, many security teams, particularly those in resource-constrained environments, struggle to adopt these models meaningfully.

This project aims to address this gap by proposing and evaluating a new, lightweight, and operationally complete methodology THRIVE. THRIVE (Threat Hunting through Repetition, Intelligence, Validation, and Engineering) is a systematic, step-by-step methodology developed and reflectively tested in a real Security Operation Centre (SOC) setting during a 12-week internship and aimed to enable more structured hunts at every stage, starting with CTI prioritization and developing hypotheses to operationalization and feedback metrics. In contrast to other models, THRIVE includes multiple aspects of collaboration, simple tooling, and consistent maturity evaluation, which is why it is approachable to both new and mature SOC teams.

This research is driven by a key question: **How can organizations improve their threat hunting approach by effectively integrating Cyber Threat Intelligence (CTI), managing limited resources, and systematically documenting and measuring threat hunting effectiveness?**

- How can cyber threat intelligence (CTI) from different sources, such as vendor blogs and OSINT providers, be organized and simplified to make it more useful for threat hunting?
- How can organizations with limited resources create a simple but efficient threat-hunting process by making use of existing security tools and concentrating on clearly defined threat scenarios?
- How can the results of individual threat hunts be recorded, shared, and systematically used to strengthen an organization's overall security (e.g., updating rules, improving processes), while also measuring the effectiveness of threat hunting and exploring relevant metrics?

To answer these questions, the project states the following research objectives:

- To analyse the limitations of existing threat hunting methodologies and outline any critical gaps in operational application.
- To design and implement a modular, scalable threat hunting process that is compatible with real-world SOC operational workflows.
- To validate the implemented methodology through case studies using real and simulated log data, peer-reviewed queries, and CTI-driven prioritization.

- To evaluate the methodology's effectiveness against both operational metrics and existing frameworks in the field.

This research makes a significant contribution by bridging the gap between conceptual threat hunting frameworks and the real-world SOC operational workflows. It operationalizes threat hunting into a structured and repeatable lifecycle tailored for the operational needs of modern SOCs by including maturity evaluation, CTI scoring, red/blue team testing, collaborative review, and hunt outcome visualization. Its lightweight and adaptable nature makes it particularly suited for adoption in diverse operational settings, including those with limited resources.

The structure of the document is as follows: **Section 2: Related Work** covers the related literature addressing the current challenges and existing threat hunting methodologies. **Section 3: Research Methodology** presents the research design methodology and evaluation criteria. **Section 4: Design Specification** details the proposed THRIVE methodology and its system architecture. **Section 5: Implementation** describes the implementation of the methodology within an organizational SOC and simulated lab environment. **Section 6: Evaluation** presents the evaluation, including three detailed case studies, dashboard metrics, comparative analysis, and discussion. **Section 7: Conclusion & Future Work** concludes the work, summarizing key findings, limitations, and meaningful directions for future research and operational deployment. The final **Section 8: References** contains all research papers and industry references used in the document.

2 Related Work

Cyber threat hunting has emerged as a crucial and proactive cybersecurity strategy in response to the escalating sophistication and frequency of cyber threats. Unlike traditional, often reactive, security measures that primarily focus on preventing known attacks or responding to detected incidents, threat hunting involves security analysts actively and iteratively searching for potential threats that may have bypassed conventional security controls. This proactive approach is essential for identifying unknown or previously undetected threats, such as Advanced Persistent Threats (APTs) and zero-day exploits, before they can cause significant damage (Abzakh *et al.*, 2023). As (Sindiramutty, 2023) highlights, adversaries can often remain undetected within systems for extended periods, leading to severe consequences like data breaches and financial losses. The concept of cyber threat hunting, formally introduced around 2016, has since seen a rising trend in research and adoption. (Wang, 2022)'s systematic review, for instance, confirms a steady increase in publications on cyber threat hunting over the years, with a notable surge between 2021 and 2022. Threat hunting is generally recognized as a cyclical and iterative process that aims to uncover and neutralize sophisticated threats within an organization's network. This process often begins with a hypothesis about potential malicious activity, which is then investigated through various data sources and analysis techniques. Threat hunting approaches can typically be categorized into:

- **Data-driven hunting** using patterns or anomalies in observable data. (Mavroeidis and Jøsang, 2018), for example, demonstrated data-driven threat hunting using Sysmon logs to classify software into threat levels.
- **Intel-driven hunting** that leverages cyber threat intelligence (CTI) reports and feeds on emergent threats, malware, or vulnerabilities to identify potential breaches. (Gao *et al.*,

2021) suggested a system, THREATRAPTOR, which supports intel-driven hunting by processing structured threat behaviours from unstructured Open-Source Cyber Threat Intelligence (OSCTI) text and generating queries.

- **Techniques, Tactics, and Procedures (TTP)-driven hunting** where the emphasis is on the discovery of specific adversarial behaviour and methods. (Bhardwaj *et al.*, 2022) emphasize the effectiveness of TTPs, noting that consistent adversarial behaviour helps defend against attacks better than relying on standard operating procedures or low-level Indicators of Compromise (IoCs). (Chetwyn, Eian and Jøsang, 2024) also emphasized that TTPs are more difficult for attackers to manipulate compared to lower-level indicators such as hash values or IP addresses, thereby making TTP-based hunting extremely beneficial to defenders.

Recent research shows an increasing trend in incorporating Machine Learning (ML) and Deep Learning (DL) techniques into threat hunting. These AI-driven capabilities aim to enhance proactive threat detection, automate tasks, and improve predictive analysis, thereby significantly improving cybersecurity practices enormously. (Mahboubi *et al.*, 2024)'s recent review highlights that ML significantly enhances cybersecurity by facilitating a more analytical and efficient defence mechanism with less reliance on time and human effort. Several approaches and frameworks have been proposed to understand and structure threat hunting practices. However, the recent SANS 2025 Threat Hunting Survey highlights that organizations are still facing difficulties in optimizing their threat hunting strategies due to lack of standardized methodologies, shortages of skilled personnel, limited resources, cloud security complexities, and difficulties in measuring effectiveness (Lemon, 2025).

2.1 Current Challenges and Gaps in Threat Hunting

Despite the advancements and increasing adoption of threat hunting, organizations continue to face significant challenges in establishing and maintaining effective threat hunting capabilities. These challenges stem from a combination of business, technical, and operational factors.
(Mahboubi *et al.*, 2024) evaluated 117 research papers in a systematic review which highlighted several key challenges in current threat hunting models mentioned in the recent SANS 2025 Survey such as limited resources and budget constraints, shortage of skilled experts, lack of standardized methodologies, difficulty in assessing effectiveness and poor documentation. Many organizations, particularly Small and Medium-sized Enterprises (SMEs), frequently experience cyberattacks but often lack the necessary financial means and dedicated resources to implement robust cybersecurity defences and conduct proactive threat hunting. (Saeed *et al.*, 2023) also highlights that building and maintaining a robust CTI capability can be expensive, especially for smaller organizations with limited resources. Effective threat hunting requires specialized skills and expertise. (Bhardwaj *et al.*, 2022) note that persistent shortages of trained experts are a key challenge for security operations teams. (Sindiramutty, 2023) and (Mahboubi *et al.*, 2024) further explain that the scarcity of skilled analysts typically limits the ability of organizations to analyse and interpret the vast amounts of data generated by security tools, leading to potential oversight of critical threats. Despite automation, human experience and intuition are considered invaluable for subtle threat assessment.

Threat hunting remains relatively underdeveloped and vaguely defined in terms of its procedures and integration within organizational structures. Although a handful of organizations employ formally defined methods such as MITRE ATT&CK, PEAK, TaHiTI, and the Pyramid of Pain to define hunts, the percentage of organizations with formally defined methodologies has decreased significantly in 2025 (Lemon, 2025). This disparity makes it difficult to measure on a consistent basis and compare over different organizations or even the same organization over different points in time. One of the biggest challenges is the lack of standardized methods for measuring or tracking threat hunting effectiveness. organizations often struggle to document what they have learned about the threat landscape to aid future threat hunting. Organizations are being overwhelmed with volumes of CTI data from various sources, making it challenging to filter and extract actionable insights. The core problem is the conversion of raw shared intelligence from CTI into actionable information specific to a given organization's context and requirements. Even formats such as STIX (Structured Threat Information eXpression), which are dynamic and hold a lot of Indicators of Compromise (IoCs), are high in complexity, leaving it to analysts who might not have the resources, particularly for SMEs. (Van Haastrecht *et al.*, 2021) argue that collaborative CTI systems such as MISP are only valuable if shared intelligence can be converted into actionable information.

Although AI/ML offers significant potential, the use of it for threat hunting faces several challenges. (Mahboubi *et al.*, 2024) point out that insufficient labelled data and unbalanced data as a major problem, where the malicious samples are sparse, causing ML model biases and higher possibilities of false negatives. Using AI-powered models in autonomous threat hunting brings issues related to scalability (managing huge amounts of data), interpretability (comprehending why AI takes specific decisions), ethical concerns, and algorithmic bias. AI systems, according to tend to heavily depend on available training data, and it can restrict them from acquiring knowledge of new or changing patterns of attack mechanisms not identified in training data sets (Vegesna and Adepu, 2024). These limitations suggest the demands for affordable, reliable and practical solutions in the threat hunting process especially for the organizations with limited resources.

2.2 Existing Popular Threat Hunting Methodologies

The Sqrrl white paper (Sqrrl Data, Inc, 2018) provides a Sqrrl framework which is among the first systematic approaches to developing threat hunting maturity in organisations. It uses a layered maturity model focusing on the transition from reactive detection to intelligence driven proactive hunting. The ability of Sqrrl to offer more of a business structure as opposed to a technical one is far more flexible. Nevertheless, the factor that it is based on qualitative maturity levels as opposed to definitive measurements restricts its empirical analysis. Although it is conceptually effective, the framework lacks operationalisation of hypothesis testing and, more generally, how to prioritise hunts in diverse threat landscapes. This empirical deficiency has caused several users to layer Sqrrl with external threat intelligence sources and structured models, including MITRE ATT&CK.

TaHiTI (Targeted Hunting integrating Threat Intelligence) was developed as a collaboration in the Dutch financial sector, and presents a hypothesis-driven, structured process that has clear phases of initiation, execution and finalisation. Its main advantage is that it is highly integrated with threat intelligence as an input and as a means of enriching hunts. TaHiTI closes the gaps of detection effectively, shortens the dwell time, and uses MITRE ATT&CK and Paul Nichols Pyramid of Pain. There are however significant limitations of TaHiTI. The fact that it relies on high-quality CTI means that organisations should have mature intelligence functions, but most small to mid-size enterprises lack that. In addition, the approach has been criticized as requiring potential resources, and as having to involve knowledgeable analysts who can refine hypotheses and engage in recursive data analysis. These dependencies limit how well it can scale outside of well-resourced environments (Os and Bakker, 2018).

The pragmatic framework provided by (Splunk, 2023) uses the PEAK (Prepare-Execute-Act with Knowledge) model which divides hunts into three categories- hypothesis-driven hunts, baseline hunts and model-assisted hunts and has a simple three-phased cycle. The key strength in the tool is this integration of hunting into security operation centre processes and attention to documentation, metrics, and automation of hunts into automated detections. PEAK is vendor-authored and presumes access to Splunk's ecosystem, but it does not offer much advice on how to test and validate hypotheses, or to peer review. In addition, it encourages measurement but there is limited empirical evidence of effectiveness in a variety of organisations.

While the reviewed methodologies offer valuable frameworks and innovations from hypothesis-driven models and CTI integration to automation and optimization techniques significant challenges persist in the field of cyber threat hunting. These include high resource demands, reliance on expert personnel, difficulties in maintaining up-to-date threat intelligence, and limited real-world validation of effectiveness. Many existing approaches remain either too theoretical or overly dependent on specialized infrastructure, making them difficult to adopt for organizations with constrained resources. In response to these gaps, this research proposes THRIVE - a practical, detailed, and adaptable threat hunting methodology that balances structure with accessibility. Unlike prior models, it is designed to be implementable by a wide range of organizations, regardless of size or maturity, without compromising on effectiveness or scalability.

3 Research Methodology

3.1 Research Approach

This research uses a Design Science Research Methodology (DSRM) to develop a practical, repeatable, and light-weight threat hunting methodology for operational environments like enterprise SOCs. Design Science is the right approach here since the purpose of the research is to create an artefact in real-world context (Vom Brocke, Hevner and Maedche, 2020). The solution is created, deployed, and validated in a live organizational environment with iterative testing and validation. The research starts with gap analysis of existing frameworks (Section

2), stating the current challenges in threat hunting domain and that the majority organizations are based on reactionary alert triage and do not have a formal, proactive threat hunting approach. This research undertakes the six subsequent phases of the DSRM:

Problem Identification: Based on first-hand internship observation and sector research, the research identified serious challenges and limitations in threat hunting i.e., poor use of CTI, absence of standard practice, and absence of measurement of hunting effectiveness.

Solution Objectives: The objective was to develop a formal, actionable, and resource-efficient way of dealing with CTI, conducting hypothesis-based threat hunting, and feeding back findings into existing detection and response procedures.

Design & Development: The THRIVE methodology was created taking inspiration from industry standards like MITRE ATT&CK, Sqrrl, and TAHITI but with a light design and more detailed that is simple to apply without custom tooling or AI/ML platforms.

Demonstration: The methodology was applied within a real SOC environment. AWS Workspace and different attack techniques were leveraged to run simulated attacks and generate realistic log data.

Evaluation: The comparative analysis of existing threat hunting methodologies along with results of each successful hunt was judged by how it exposed behaviour, matched predicted TTPs, assisted in detection rule construction, and enhanced SOC knowledge.

Communication: The entire methodology, implementation, and findings were documented through structured reporting, configuration manuals, and demonstration videos.

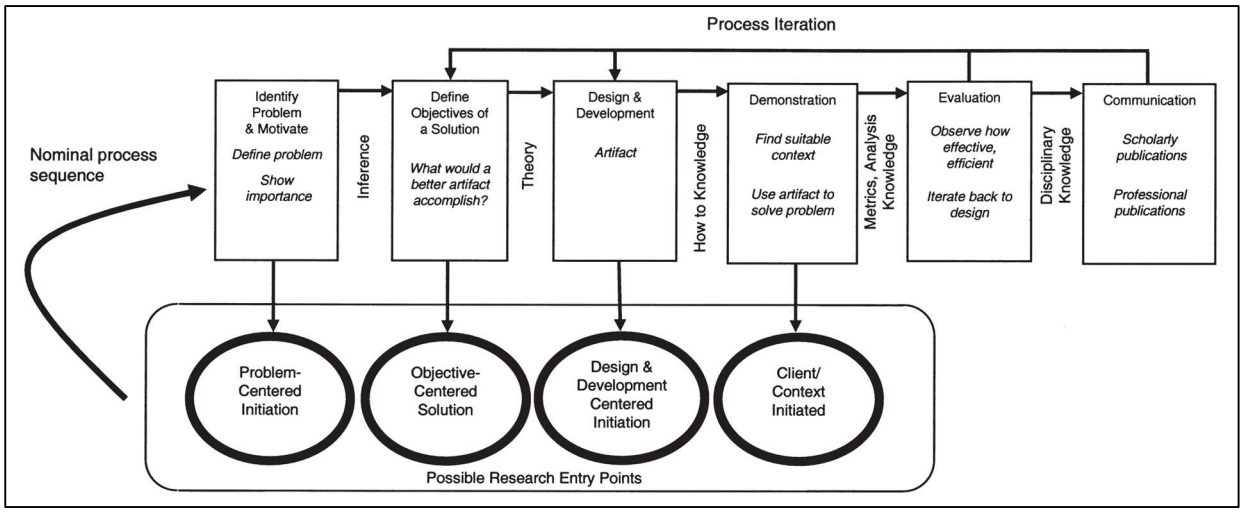

Figure 1. DSRM Process Model (Vom Brocke, Hevner and Maedche, 2020)

3.2 Data Collection and Experimental Setup

The data collection for this research was implemented in two parallel streams: real enterprise telemetry and simulated attack scenarios. An AWS-hosted Windows 10 environment was used to simulate known adversary behaviours. Cyber Threat Intel was collected from automated and manual means. Public RSS feeds were used from trustworthy sources such as CISA Alerts, The Hacker News, BleepingComputer, and indicators of compromise (IOCs) from open-source ThreatFox platform ingested using Microsoft Logic Apps in combination with Microsoft Teams channels.

3.3 Tooling and Environment

All experiments and data collection were carried out in a professional Security Operations Centre (SOC). The core technical tools were:

- **Microsoft Sentinel** for log aggregation, hunting queries, and detection engineering.
- **Microsoft Defender for Endpoint** for adversary behaviour telemetry.
- **AWS Workspace** for simulated attack activity and controlled experimentation.
- **Logic Apps** for automated ingestion of CTI feeds.
- **Amazon QuickSight** for dashboard-based visualization of metrics.

Supporting tools such as Excel (for CTI scoring), GitHub (for peer review simulation), and SharePoint (for documentation) were employed during implementation and are described in Section 5. Figure 3 (Threat Hunting Architecture) illustrates how these components interact to support the THRIVE methodology end to end.

3.4 Hypothesis Creation and Testing

The main idea behind the THRIVE methodology is turning real world threat intelligence into actionable hunting hypotheses and then testing those hypotheses in a safe controlled environment. Drawing from CTI gathered from recent threat intelligence feeds and past organization incidents; three important example hypotheses mentioned in section 6 were selected. Each one matched a known attack technique from the MITRE ATT&CK framework and tested against live or simulated telemetry. The results were classified based on multiple evaluation metrics. Detections based on validated hunts were then made operational

3.5 Evaluation Criteria and Metrics

The effectiveness of the proposed THRIVE threat hunting methodology was evaluated through a combination of real-world application, case-based experimentation, and structured measurement. Evaluation was carried out during a 12-week internship in an active Security Operations Centre (SOC), where three end-to-end hunts were executed using live organizational telemetry and, where required, simulated data in a controlled lab setup.

- **Case studies** to demonstrate the methodology in practice.
- **Operational metrics** including hunt execution, detection accuracy, detection conversion rate, coverage of ATT&CK techniques, etc.
- **Comparative analysis** against Sqrrl, TaHiTI, and PEAK to assess strengths and limitations.

This strategy provided both internal validity through demonstration and external relevance through benchmarking.

4 Design Specification

This section defines the architecture, methodology, and operational components of the proposed THRIVE methodology in detail. We begin by describing the two foundational elements of the THRIVE artifact; first a detailed overview of the nine steps THRIVE methodology defining what processes analysts will execute and second, the system architecture that helps carry out those steps smoothly.

4.1 Overview of the THRIVE Methodology

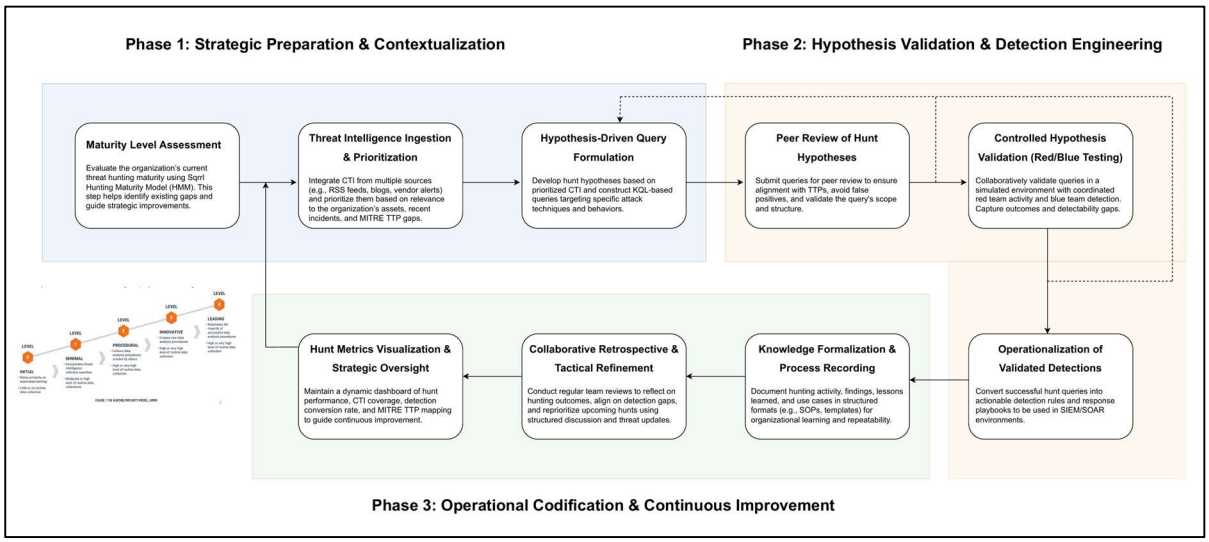

Figure 2. THRIVE Methodology Overview.

The THRIVE methodology fills important gaps in current methods by providing a lightweight, easy-to-follow process that relies on real threat intelligence. This methodology breaks threat hunting into a nine-step cycle, grouped into three main phases. Each phase designed to overcome specific operational challenges.

Phase 1: Strategic Preparation & Contextualization

- ***Maturity Level Assessment****:* Organizations often lack visibility into their own hunting capabilities and as a prerequisite to adoption, organizations are required to establish their threat hunting maturity currently through the Sqrrl Hunting Maturity Model (HMM). The scale assesses five functional domains such as data collection, tooling, hypothesis generation, TTP detection, and detection engineering and will score from Level 0 (Initial) to Level 4 (Leading) (sqrrl, 2018). The result determines the company's place on the framework and adjusts expectations of what can realistically be implemented. A standardized self-assessment questionnaire, based on the Sqrrl Hunting Maturity Model, is included in the configuration guide to help organizations find out their current level of hunting. Refer to configuration manual section 2 for HMM levels and questions summary.
- ***Threat Intelligence Ingestion & Prioritization****:* Threat intelligence (CTI) is the main input used to drive hunting hypotheses but too many SOCs buried in unfiltered threat feed don't turn intelligence into hunts. Intelligence is being consumed from open-source feeds (e.g., ThreatFox, BleepingComputer, The Hacker News RSS), past events, and vendor blogs. To prevent CTI flooding and concentrate hunting activities, a structured CTI scoring model is used. It ranks threat items according to TTP relevance, recent past active, industry targeted, confidence level, evasion capability, and detection feasibility. High-priority CTI records are chosen as inputs for hypothesis creation. The CTI scoring model assigns weights to threat intelligence inputs based on six criteria, TTP Alignment, Industry Targeted, Recency, Confidence Score (IOC/Feed), Evasion Capability, Detection Readiness. A

reusable Excel version and scoring guide are provided in the configuration manual for operational use. See configuration manual section 3.3 for a full scoring matrix.

- ***Hypothesis-Driven Query Formulation****:* Once a CTI artifact has been prioritized, a formal hunt hypothesis is developed. Each hypothesis contains a mapped MITRE ATT&CK technique, suspected adversary activity to be investigated, required data sources to detect, and context or triggering events, if relevant. This type of hypothesis is what the queries are based on.

Phase 2: Hypothesis Validation & Detection Engineering

- ***Peer Review of Hunt Hypotheses****:* Individual threat hunters may miss certain details or accidentally create rules that trigger too many false positives. To avoid this, queries should be submitted via structured code management systems for peer review. This enables collaborative review, and reviewers can check whether the queries match known attacker tactics, if they run efficiently, and whether they create too much noise. The peer review process enhances quality assurance and promotes shared understanding across the threat hunting team. Only validated queries proceed to testing.
- ***Controlled Hypothesis Validation (Red/Blue Team Testing)****:* Conceptual validation alone is insufficient for establishing detection reliability. Adversarial behaviours in the real world often vary from documented tactics. Therefore, validated detection queries are deployed within an isolated test environment. A coordinated red team simulates attacker activity while the blue team evaluates the efficacy of the query in identifying malicious behaviour. This process highlights limitations in telemetry coverage, log fidelity, and detection logic, ensuring the proposed solutions perform effectively under realistic conditions.
- ***Operationalization of Validated Detections****:* Where applicable, successful hunt queries are converted into scheduled analytics rules or detection playbooks. Due to tooling restrictions (e.g., Defender's rule creation limitations), the methodology supports documenting detection logic and escalation paths in structured SOPs, which can later be implemented in SIEM/SOAR platforms. This modularity ensures organizations can progress incrementally even without automation.

Phase 3: Operational Codification & Continuous Improvement

- ***Knowledge Formalization & Process Recording:*** Threat hunting activities produce critical insights that should not remain isolated within analyst workflows. To promote institutional knowledge retention and enable repeatability, each hunt is documented using structured SOPs and hunt templates. These templates capture hypothesis rationale, CTI references, query logic, MITRE technique mappings, test outcomes, and false positive observations. The documentation is version-controlled using platforms like Microsoft SharePoint, allowing teams to reuse, refine, or expand upon previous hunts. Over time, this library of hunting SOPs becomes a valuable knowledge base that informs detection engineering, analyst onboarding, and SOC training.
- ***Collaborative Retrospective & Tactical Refinement:*** There are also regular retrospectives, where hunting operations are critically reviewed, and the threat hunting program is

readjusted to adapt to changing threats and operational needs. These meetings entail the examination of significant finds such as which hunts created new detection, which had too many false positives, and which TTPs remain unexposed. Retrospectives assist in coming up with telemetry gaps, tuning CTI scoring model upon analyst experience, and renewing team focus to overlooked TTPs or attack surfaces. Such collective reflection promotes responsibility, strengthens team learning, and enables threat hunting to develop dynamically instead of becoming a static routine.

- ***Hunt Metrics Visualization & Strategic Oversight:*** The methodology integrates a metrics-driven evaluation approach. Core metrics such as number of hunts performed, detection conversion rates, TTP coverage, and false positive ratios are aggregated and visualized via Sentinel Workbooks or external dashboards. A dedicated MITRE ATT&CK coverage map is also maintained to highlight detection blind spots and prioritize future hunts. These visual dashboards enable SOC leaders to assess the operational value of threat hunting and realign detection objectives with threat intelligence trends, audit requirements, or executive directives. This strategic oversight ensures that the hunting function remains focused, data-driven, and continuously improving.

4.2 Overview of Architecture Diagram

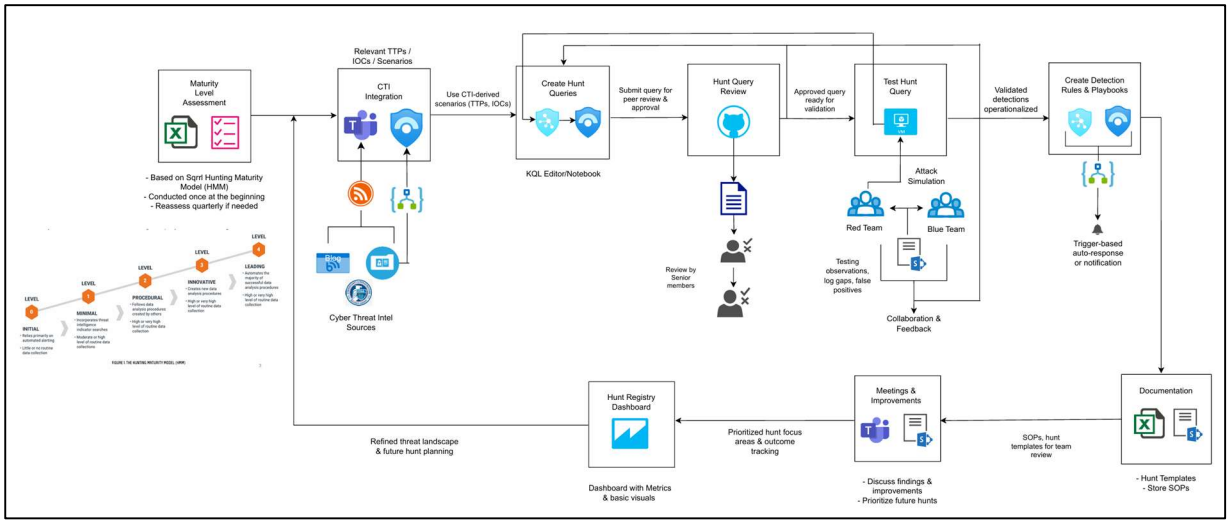

Figure 3. THRIVE Architecture Diagram.

The THRIVE architecture was designed to support each phase of the proposed threat hunting methodology in a tool-independent, modular and scalable manner. It enables organizations of different maturity levels to benefit from organized threat hunting with available processes and tools without vendor lock-in or heavy custom development. As shown in Figure 3, the architecture consists of nine interconnected components, each corresponding to a specific phase in the THRIVE methodology. Each component interacts with others by employing standard documentation, peer review, and dashboarding mechanisms, thereby enabling collaboration, accountability, and continuous improvement.

Inputs include CTI feeds (ThreatFox, RSS advisories), maturity assessment results, and telemetry from Sentinel and Defender, forming the basis for hypothesis-driven hunts.

Processes cover the execution of THRIVE's core phases: hypothesis generation, peer review, controlled validation, and the operationalisation of validated queries.
Outputs consist of validated detections (implemented as rules), QuickSight dashboards visualising hunt performance, and documented SOPs/templates to ensure repeatability.
Unlike earlier frameworks, the architecture formalises peer review and metrics dashboards as integral design elements rather than optional extensions. By linking CTI prioritisation and maturity assessment directly to hunt selection, THRIVE remains adaptable across different SOC maturity levels.

5 Implementation

5.1 Implementation Environment

The THRIVE threat hunting methodology was fully implemented and evaluated within a live Security Operations Centre (SOC) during a 12-week internship. The SOC environment had Microsoft Defender for Endpoint and Microsoft Sentinel already configured with continuous log ingestion from monitored organizational endpoints. As a result, the full hunting lifecycle including CTI-driven hypothesis development, query execution, peer review, and validation was carried out directly on real enterprise telemetry without requiring additional platform setup. To support the research component of this thesis and facilitate hands-on understanding, a parallel demonstration environment was configured using an Azure student subscription. While no hunts were conducted in this lab, it was designed to emulate SOC setup tasks, including virtual machine provisioning, Log Analytics workspace configuration, Microsoft Sentinel onboarding, and telemetry flow validation. This demo environment was primarily used to generate configuration screenshots and video demonstrations for the accompanying configuration manual. Following a formal access request, AWS WorkSpaces were provisioned within the internship environment to simulate attacker behaviours and generate synthetic telemetry data. This enabled controlled red team and blue team interaction, allowing for the practical validation of detection hypotheses in a realistic and monitored test scenario. The detailed implementation steps are mentioned in the attached configuration manual with screenshots.

5.2 Tools and Platforms Used

The following tools and platforms were used to execute the THRIVE methodology phases during the internship implementation:

Phase	Tools Used
Maturity Level Assessment	Excel-based Sqrrl HMM self-assessment
CTI Ingestion and Prioritization	ThreatFox IOCs using Logic Apps, RSS feeds (BleepingComputer, The Hacker News, etc), Excel-based CTI scoring matrix
Hypothesis & Query Generation	Microsoft Sentinel, Microsoft Defender for Endpoint (KQL Editor)

Peer Review Process	GitHub pull request workflow using student and personal accounts for demonstration
Test Execution & Validation	Live telemetry and activity simulation via AWS Workspace (Windows 10)
Detection Rule & Playbook Design	Create Sentinel Analytics Rules and playbook
Hunt Documentation	Microsoft SharePoint & Microsoft Excel
Meetings & Improvements	Microsoft Teams Meetings
Dashboarding and Metrics	Amazon QuickSight

Table 1. Tools used in implementing THRIVE methodology across phases.

5.3 Outputs Produced

The following tangible outputs were generated as a result of the implementation:

- **HMM Maturity Assessment Tool**: Excel-based self-assessment tool for checking organizational readiness
- **CTI Prioritization Matrix**: Weighted scoring matrix applied to real-world and open-source CTI
- **Hunt SOPs**: Three fully documented hunts including hypothesis, query logic, findings, false positives, and lessons learned
- **Query Peer Review Workflow**: GitHub repository showcasing PR-based collaborative validation
- **Detection Rules and Playbooks**: Three drafted analytics rules and detection playbook derived from validated hunt queries
- **Dashboard Visualization**: AWS QuickSight dashboard for real-time metrics on hunt activity and MITRE technique coverage
- **Configuration Manual**: Detailed step-by-step setup guide with screenshots of key components and flow of methodology
- **Demo Video**: Narrated recording showing lab setup, hunt flow, and detection rule creation.

6 Evaluation

This section evaluates the practical effectiveness, maturity alignment, and effectiveness of research of the suggested THRIVE threat hunting methodology. This is done by the assessment of THRIVE in a typical Security Operations Centre (SOC) environment on three different case studies and comparing its functionality against other established threat hunting practices like Sqrrl, TaHiTI, and PEAK as shown in Table 2. Each case study is based on a real or emerging threat scenario and shows the complete THRIVE process from CTI triage and hypothesis development to peer-reviewed query formulation, testing, and operational reporting. The hunts led to real detections and validated simulated outputs and triggered meaningful organizational responses. The evaluation also includes a structured comparison of THRIVE against existing frameworks, identifying specific research and operational gaps that it fills. Finally, the section reflects critically on challenges faced during the implementation.

6.1 Case Study 1 - FileFix-Style Browser-Launched Command Execution

To identify recent social engineering attack patterns specifically the FileFix attack, which abuses browsers to launch obfuscated command execution via the address bar (mr.d0x, 2025). The technique is relatively new, with proof-of-concept demonstrations in a simulation environment. This was aligned to MITRE technique T1059.001 (Command and Scripting Interpreter: PowerShell).

- **Methodology Execution:**

CTI Trigger: Hypothesis developed from technical blogs and researcher posts (e.g., mrd0x, Imran Elalami, LinkedIn write-ups) that exposed new abuse paths via browser interfaces.
Hypothesis: Adversaries may trick users into copying payloads into the File Explorer or browser address bar, which may result in encoded or weaponized command execution on the system.
Query Design: KQL logic correlated suspicious process launches from browser parents with obfuscation flags in command-line arguments.
Data Source: DeviceProcessEvents (Microsoft Defender for Endpoint).
Peer Review: Query reviewed in GitHub pull request, focusing on minimizing benign PowerShell triggers from legitimate admin activity.
Validation: No hits found in production logs. Hypothesis tested via simulated execution in AWS Workspaces, where the command chain was successfully observed and captured in Defender logs.

- **Findings:**

True Positives: No matches were found in live SOC telemetry (90 days window) but simulation of attack behaviour confirmed logs and logic validity as shows in figure 4 and figure 5.
Outcome: Hunt logic retained as a hunting-only rule and documented in the SOP repository. Future tuning will be based on monitoring real-world exploitation and updating command-line patterns accordingly.

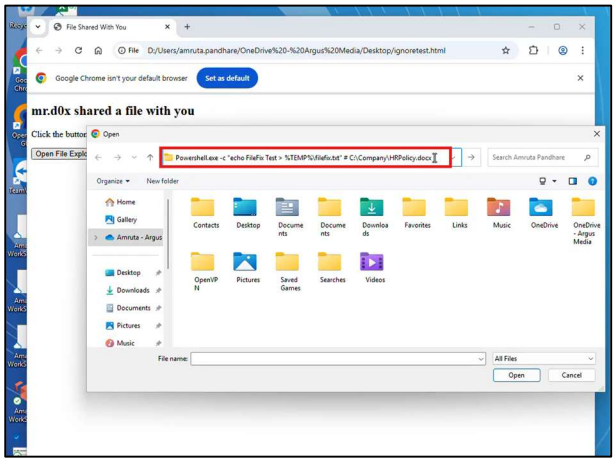

Figure 4. Filefix POC demo

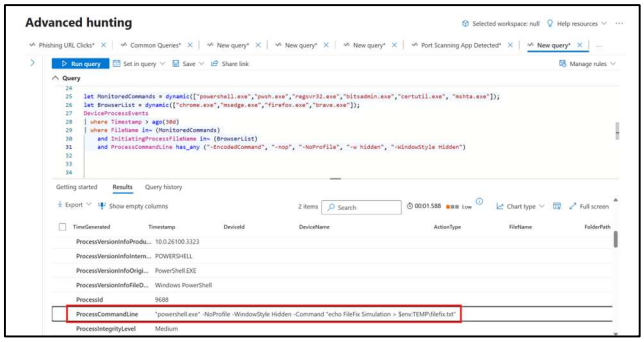

Figure 5. Filefix implementation logs

6.2 Case Study 2 - Potential Data Exfiltration (Insider Threat)

To detect potential data exfiltration to personal email domains by employees shortly before leaving the organization or account disablement indicating potential intellectual property theft by leavers. This was aligned to MITRE technique T1537 (Exfiltration Over Alternative Protocol).

- **Methodology Execution:**

CTI Trigger: Internal incident reports and HR audits flagged this as a recurring risk.
Hypothesis*:* Users being offboarded may send unauthorized attachments to personal email domains shortly before termination.
Query Design**:** KQL query correlated outbound emails with known personal domains (e.g., Gmail, Yahoo) and account disablement logs within a 30-day window.
Data Source**:** EmailEvents, AuditLogs, SignInLogs (Microsoft Sentinel).
Peer Review: Conducted through GitHub pull request with reviewer feedback focused on domain filtering and time range optimization.
Validation: Executed in the internship SOC environment against 30 days of production telemetry.

- **Findings:**

True Positives: 4 users sent multiple attachments to their personal emails. As shown in figure 6, top offender sent 36 emails with attachments to a Gmail account 10 days before their account was disabled and this was validated with the Mimecast's logs as well as shown in figure 7.
False Positives: 6 were false positives containing some legitimate resumes, payslips, expense reimbursement receipts or approved content.
Organizational Response: HR initiated an intellectual property review based on the findings. Provided list of leavers to monitor their activities after resigning.
Output: A detection rule and SOP was created and added to the threat hunting repository for future use.

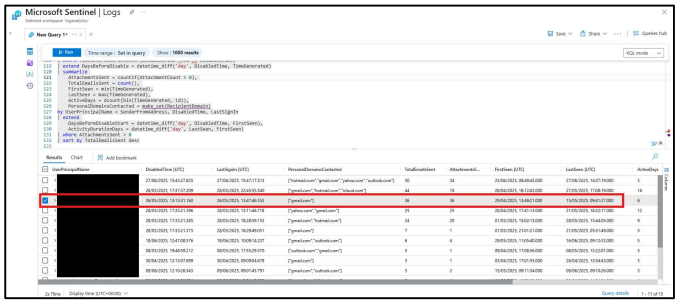

Figure 6. Sentinel email logs

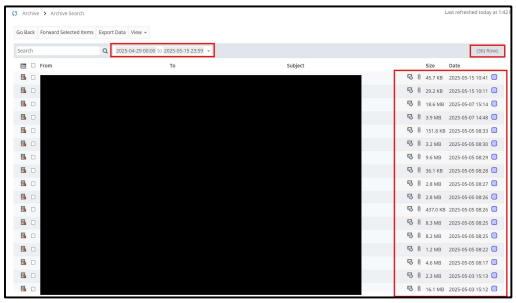

Figure 7. Mimecast validated logs

6.3 Case Study 3 - Potentially Malicious Browser Extensions

To identify the installation of known-malicious Chrome and Edge extensions (distributed via official stores) which were recently removed from Chrome and Edge Web Stores after being

exposed in multiple security publications due to spyware like behaviours (Dardikman, 2025). This was aligned to MITRE technique T1176 (Software Extensions: Browser Extensions).

- **Methodology Execution:**

***CTI Trigger*:** Multiple threat intel RSS reports from Forbes, Koi Security, and Dark Reading revealed 18 malicious extensions affecting over 2 million users.
Hypothesis: End users may have installed spyware or data-leaking extensions that were previously trusted due to being available in official web stores.
Query Design: A KQL query matched .crx extension installation logs with a static list of known-malicious extension IDs and names.
***Data Source*:** DeviceFileEvents (Microsoft Defender for Endpoint).
Peer Review: Query reviewed and confirmed as low risk (no false positives due to exact match on extension IDs).
Validation: Hunt executed over 30 days of telemetry in production environment.

- **Findings:**

True Positives: 1 user had installed the Color Picker Eyedropper extension, which was among the known-malicious entries as shown in figure 9.
Action Taken: The user was contacted; the extension was voluntarily removed. Chrome Web Store delisted it shortly thereafter.
Organizational Response: Sparked internal discussion around lack of visibility into browser extensions. This resulted in a trial enablement of Defender's browser extension monitoring for visibility into usage patterns.
Outcome: Documented in SOP and added as a recurring hunt in Sentinel. Defender browser visibility add-on enabled on a 30-day trial basis.

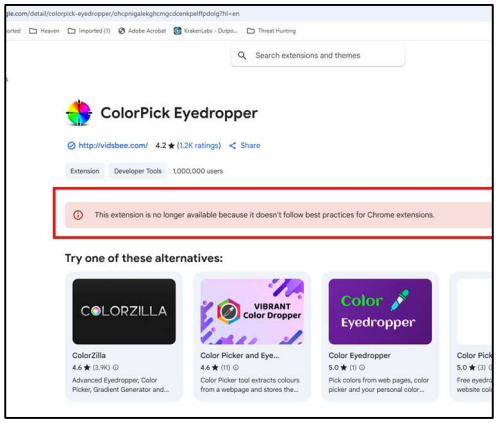

Figure 8. Malicious Chrome extension

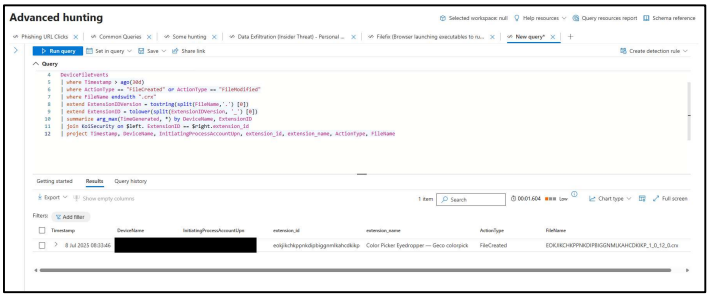

Figure 9. SOC device file events logs

Figure 10. QuickSight THRIVE Threat Hunting Dashboard

6.4 Discussion

- **Comparative Analysis of Existing Threat Hunting Models**

Parameter	Sqrrl Framework	TaHiTI Methodology	PEAK Framework	THRIVE Methodology
Methodology Structure	Structured around a continuous loop of four stages and a separate maturity model. It lacks an explicit, end-to-end procedural flow.	A three-phase process (Initiate, Hunt, Finalize) with six steps. It is a structured process but lacks specific steps for validation and continuous feedback loops.	A practical and customizable approach with three different hunt types, designed to refine and mature a hunting program. Each hunt type follows a three-stage process: Prepare, Execute, and Act.	A structured, nine-step cycle across three distinct phases (Strategic Preparation, Hypothesis Validation, and Operational Codification). This provides a more detailed, end-to-end workflow from initial planning to continuous improvement, including specific steps for validation and feedback.
Maturity Assessment	Uses the Hunting Maturity Model (HMM) to quantify an organization's hunting ability based on data	It does not mention the use of HMM or propose a formal maturity assessment model.	Incorporates the Hunting Maturity Model (HMM) to help leaders assess their program's current state and figure out	It mandates a maturity level assessment as a prerequisite to adoption using the Sqrrl HMM. This ensures that the methodology is adapted to the organization's current capabilities, setting realistic

	collection, visualization, and automation. The model serves as a roadmap for improvement.		how to get where they want to be.	expectations and a clear path for incremental improvement from the start.
Threat Intelligence (TI) Integration	Integrates threat intel feeds into automated alerting for basic matching at the HM1 level. The framework uses threat reports and open/closed sources to develop new hypotheses.	Emphasizes that TI is a major source for hunting hypotheses and can also be used to contextualize findings and generate new intelligence. It focuses on the top three layers of the Pyramid of Pain.	Across all three phases "Knowledge" is integrated. This Knowledge includes threat intelligence, alongside organizational expertise, and findings from previous hunts.	It addresses the problem of CTI overload by proposing a structured, six-criteria scoring model to prioritize threat intelligence. This ensures that hunting efforts are focused on the most relevant and impactful threats (e.g., based on TTP relevance, recency, and detection readiness), a detail lacking in the other methodologies. This moves beyond simply using TI to prioritize it.
Hypothesis Development	Hypotheses can be created manually or automatically by risk algorithms and are investigated using tools and techniques.	Hypotheses are created in the "Define / refine" phase. A poorly defined hypothesis can lead to incorrect conclusions.	The "ABLE method" is used to scope a hypothesis, breaking it down by Actor, Behaviour, Location, and Evidence.	It ties hypothesis formulation directly to prioritized CTI artifacts and maps each hypothesis to a specific MITRE ATT&CK technique. This provides a structured, verifiable, and threat-informed basis for each hunt.
Validation and Testing	The framework describes an iterative process where hunts uncover new patterns and TTPs, which in turn inform automated analytics. It does not explicitly mention red/blue team testing.	Hypotheses are validated by analysing data, and if the hypothesis is inconclusive, the hunter can cycle back to refine the parameters. It does not describe a formal testing or peer review process.	The Execute phase involves gathering and analysing data, and hunters can refine their hypothesis as they go. If critical findings are discovered, they are escalated to the incident response team. There is no explicit mention of red/blue team testing or peer review in the framework.	It has Peer Review of Hunt Hypotheses and Controlled Hypothesis Validation (Red/Blue Team Testing). Peer review enhances quality assurance and team collaboration, while red/blue team testing in a simulated environment ensures queries are effective under realistic conditions before being operationalized. This is a rigorous, multi-stage validation process that is absent from the other frameworks.
Operationalization & Improvement	Successful hunting processes are operationalized and turned into automated detections, freeing up analysts to create new ones. The goal is continuous	The final phase involves documenting findings and handing them over to other processes like security monitoring to create or update use cases. It has a	A key deliverable is turning hunts into automated detections to improve security posture over time. The "Act" phase focuses on documentation, automation, and communication of findings.	It provides a detailed and structured approach for operationalization and continuous improvement. This includes formalizing knowledge in version-controlled SOPs and hunt templates, conducting regular "retrospectives" to critically review operations and adapt to changes, and using a metrics-

	improvement of the detection program.	"lessons learned" section to help hunters improve.		driven dashboard for strategic oversight. These specific, codified steps ensure that lessons are not lost and that the program dynamically evolves.
Metrics and Oversight	Discusses the need to measure the effect of hunting, not just the effort, such as measuring new and updated detections. It suggests matching hunts against models like MITRE ATT&CK or the Pyramid of Pain.	Suggests a short list of metrics that are indicators of value, such as dwell time, incidents triggered, and the number of new use cases or intelligence created.	PEAK defines five key metrics to track, including the number of detections created or updated, incidents opened as a result of a hunt, and gaps identified/closed. It also incorporates the HMM to help leaders assess maturity.	It provides a comprehensive, metrics-driven evaluation approach with a dynamic dashboard. It tracks core metrics like detection conversion rates, TTP coverage via a MITRE ATT&CK map, and false positive ratios. This high-level visualization enables SOC leaders to not only justify the program's value but also to realign objectives with intelligence trends and strategic directives, providing a clear link between tactical hunts and strategic goals.

Table 2. Comparative analysis of Sqrrl, TaHiTI, PEAK, and THRIVE methodology.

- ## Challenges and limitations

While the THRIVE methodology demonstrated promising results during its implementation and evaluation, below are some of the challenges and limitations that were encountered.

CTI Overlap: While the CTI prioritization matrix was effective, over-reliance on multiple RSS sources led to information duplication. Some feeds repeated the same IOCs or incident references. This highlighted the need for source curation and noise reduction in CTI ingestion pipelines which was achieved by incorporating only reliable and limited RSS sources.

Telemetry and Logging Constraints: Although hypothesis generation was smooth, there were undocumented log gaps that emerged as certain behaviours could not be validated due to telemetry limitations. This is a common challenge in live SOCs and underscores the importance of continuous feedback from hunting to telemetry engineering. THRIVE supports this, but in this implementation, it was limited by project scope and timeframe.

Peer Review Constraints: While hunt queries and SOPs were discussed with team members, the GitHub-based peer review process was simulated in a personal repo due to organizational confidentiality. This workaround was functional but lacks the security and access controls of an enterprise code review system.

Testing Limitations: Full red team engagement was not possible due to organizational structure. As a result, controlled testing was conducted individually in AWS WorkSpaces, which limited the variety and depth of adversarial behaviours that could be simulated. While this did not affect validation of hunt logic, it constrained the diversity of telemetry for stress-testing the detection queries.

Dashboard Scalability: The QuickSight dashboard was created after hunting activities concluded, using manually populated datasets. Although visually informative, it lacks real-

time integration and required manual updates. With further development, automation of hunt telemetry ingestion could greatly enhance its strategic value.

7 Conclusion and Future Work

7.1 Conclusion

This research set out to address a fundamental question in cybersecurity operations: How can organizations improve their threat hunting approaches by effectively integrating Cyber Threat Intelligence (CTI), managing limited resources, and systematically documenting and measuring threat hunting effectiveness? To answer this, this research developed and evaluated THRIVE, a structured, lightweight, and accessible threat hunting methodology designed for practical use in real-world Security Operations Centres (SOCs), especially those constrained by limited budgets, tooling, or expertise. Drawing from industry practices, academic literature, and operational gaps identified through the SANS 2025 Threat Hunting Survey, this research pursued a Design Science Research Methodology (DSRM), involving iterative development, demonstration, and real-world evaluation of the proposed solution. The THRIVE methodology introduced a nine-step workflow across three core phases, addressing current challenges in existing frameworks such as lack of maturity alignment, unclear CTI prioritization, insufficient validation, and poor documentation. Its structured approach from maturity assessment to operational codification proved to be both repeatable and adaptable during the case studies. Notably, each case study led to real or simulated detections, highlighting the practical utility of the approach.

The research successfully met its objectives by designing a lightweight yet detailed methodology accessible even to teams with limited resources, integrating CTI scoring and hypothesis prioritization to enhance the signal-to-noise ratio, and developing structured templates and documentation workflows that promote retention, reuse, and continuous learning. It was practically implemented in a real Security Operations Centre (SOC), validated through red/blue team testing and peer-reviewed queries, thereby contributing a tangible, operational artifact to the cyber threat hunting domain. This work offers a structured, pragmatic approach that strengthens detection capabilities and aligns tactical efforts with broader strategic security goals. However, as discussed in Section 6.4 (Challenges & Limitations), the methodology faced constraints such as CTI Overlap, limited telemetry coverage, simulation boundaries, and incomplete peer-review due to organizational confidentiality, highlighting areas for future refinement and expanded validation.

7.2 Future Work

Although the THRIVE methodology has shown strong performance within its current scope, there are several ways it could be improved and expanded in the future. One area is the integration of automated cyber threat intelligence (CTI) feeds and real-time scoring, which would make the CTI prioritization process more dynamic and less manual, using platforms like MISP or STIX/TAXII to reflect threat trends and confidence levels. THRIVE could also evolve

by fully integrating with SOAR platforms to turn validated hunts into automated response playbooks, helping with enrichment, alert tuning, and containment to boost real-time responsiveness and reduce dwell time. Expanding the use of red team simulations with tools like Atomic Red Team or Caldera would allow for broader testing of attack techniques and strengthen detection capabilities. Another important direction is evaluating how well THRIVE works for junior analysts or new SOC teams, using usability studies and onboarding experiences to improve training and accessibility. Although THRIVE is designed to be vendor-neutral, it was implemented in Microsoft Sentinel, so future studies could test its compatibility with other platforms like Elastic SIEM, Splunk, or Wazuh to ensure its detection logic and dashboards are portable. Lastly, automating dashboard metrics and visualization could help track performance more effectively, with features like adaptive MITRE heatmaps, dwell time trends, false positive rates, and mean time to detection offering deeper insights into hunt effectiveness.

8 References

Abzakh, A. *et al.* (2023) "A Survey: Threat Hunting for the OT Systems," in *2023 International Conference on Information Technology (ICIT)*. *2023 International Conference on Information Technology (ICIT)*, Amman, Jordan: IEEE, pp. 130–134. Available at: https://doi.org/10.1109/ICIT58056.2023.10225758.

Alevizos, L. and Dekker, M. (2024) "Towards an AI-Enhanced Cyber Threat Intelligence Processing Pipeline," *Electronics*, 13(11), p. 2021. Available at: https://doi.org/10.3390/electronics13112021.

Bhardwaj, A. *et al.* (2022) "BTH: Behavior-Based Structured Threat Hunting Framework to Analyze and Detect Advanced Adversaries," *Electronics*, 11(19), p. 2992. Available at: https://doi.org/10.3390/electronics11192992.

Chetwyn, R.A., Eian, M. and Jøsang, A. (2024) "Modelling Indicators of Behaviour for Cyber Threat Hunting via Sysmon," in *European Interdisciplinary Cybersecurity Conference*. *EICC 2024: European Interdisciplinary Cybersecurity Conference*, Xanthi Greece: ACM, pp. 95–104. Available at: https://doi.org/10.1145/3655693.3655722.

Dardikman, I. (2025) *Google and Microsoft Trusted Them. 2.3 Million Users Installed Them. They Were Malware.*, *Medium*. Available at: https://blog.koi.security/google-and-microsoft-trusted-them-2-3-million-users-installed-them-they-were-malware-fb4ed4f40ff5 (Accessed: July 18, 2025).

Gao, P. *et al.* (2021) "A System for Efficiently Hunting for Cyber Threats in Computer Systems Using Threat Intelligence," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 2705–2708. Available at: https://doi.org/10.1109/ICDE51399.2021.00309.

Lemon, J. (2025) “SANS 2025 Threat Hunting Survey: Advancements in Threat Hunting Amid AI and Cloud Challenges.”

Mahboubi, A. *et al.* (2024) “Evolving techniques in cyber threat hunting: A systematic review,” *Journal of Network and Computer Applications*, 232, p. 104004. Available at: https://doi.org/10.1016/j.jnca.2024.104004.

Mandiant (2025) *M-Trends 2025: Data, Insights, and Recommendations From the Frontlines*, *Google Cloud Blog*. Available at: https://cloud.google.com/blog/topics/threat-intelligence/m-trends-2025 (Accessed: August 7, 2025).

Mavroeidis, V. and Jøsang, A. (2018) “Data-Driven Threat Hunting Using Sysmon,” in *Proceedings of the 2nd International Conference on Cryptography, Security and Privacy*, pp. 82–88. Available at: https://doi.org/10.1145/3199478.3199490.

mr.d0x (2025) *FileFix - A ClickFix Alternative | mr.d0x*. Available at: https://mrd0x.com/filefix-clickfix-alternative/ (Accessed: July 18, 2025).

Os, R. van and Bakker, M. (2018) “TaHiTI Threat Hunting Methodology.” Available at: https://www.betaalvereniging.nl/en/safety/tahiti/.

Saeed, S. *et al.* (2023) “A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience,” *19 August 2023* [Preprint].

Sindiramutty, S.R. (2023) “Autonomous Threat Hunting: A Future Paradigm for AI-Driven Threat Intelligence.”

Splunk. (2025). Introducing the PEAK Threat Hunting Framework | Splunk. [online] Available at: https://www.splunk.com/en_us/blog/security/peak-threat-hunting-framework.html [Accessed 20 July. 2025].

Sqrrl Data, Inc (2018) “A Framework for Cyber Threat Hunting.” Available at: https://www.threathunting.net/files/framework-for-threat-hunting-whitepaper.pdf.

Van Haastrecht, M. *et al.* (2021) “A Shared Cyber Threat Intelligence Solution for SMEs,” *Electronics*, 10(23), p. 2913. Available at: https://doi.org/10.3390/electronics10232913.

Vegesna, Dr.V. and Adepu, A. (2024) “Leveraging Artificial Intelligence for Predictive Cyber Threat Intelligence.”

Vom Brocke, J., Hevner, A. and Maedche, A. (2020) “Introduction to Design Science Research,” in J. Vom Brocke, A. Hevner, and A. Maedche (eds.) *Design Science Research. Cases*. Cham: Springer International Publishing (Progress in IS), pp. 1–13. Available at: https://doi.org/10.1007/978-3-030-46781-4_1.

Wang, Z. (2022) “A systematic literature review on cyber threat hunting.” arXiv. Available at: https://doi.org/10.48550/arXiv.2212.05310.