UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**

Securing the Digital Frontier

*Identifying and Closing Security Gaps
with MITRE ATT&CK and
Open-Source Detection Controls*

Aditya Kumar Sharma

**M.Sc. Thesis
November 2023**

Supervisors:
Dr. José Jair Cardoso de Santanna (UT)
Dr. Florian W. Hahn (UT)
Dennis Snijders (Computest)

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Preface

The inspiration behind this thesis titled ”Securing the Digital Frontier: Identifying and Closing Security Gaps with MITRE ATT&CK and Open-Source Detection Controls” came to the author’s mind during working as a Security Operation Center (SOC) Analyst before starting his master’s degree. He was curious to find and implement improvements in the modern detection capabilities of organizations. With this work he aims to contribute to the security community, a way to further advance their detection of actions performed by attackers in the real-world.

To achieve a more academic tone this thesis make use of ChatGPT from OpenAI to re-write selected parts of the thesis. For this the prompt provided to ChatGPT was ”Rewrite these sentences with proper grammar and an academic tone:”, following with the manually written part. The response from ChatGPT was also modified to eliminate the use of obscure words that are not used commonly or are difficult to understand.

Aditya Kumar Sharma
November 2023

Contents

List of acronyms

ATT&CK	Adversarial Tactics, Techniques, and Common Knowledge
DeTT&CT	Detect Tactics, Techniques and Combat Threats
EDR	Endpoint Detection and Response
ICS	Industrial Control Systems
IT	Information Technology
IDS	Intrusion Detection Systems
IPS	Intrusion Prevention Systems
KQL	Kusto Query Language
MiTM	Man-in-The-Middle
OT	Operational Technology
SIEM	Security Information and Event Management
SOC	Security Operations Center
TTP	Tactics, Techniques, and Procedures
VPN	Virtual Private Network

Abstract

The COVID-19 pandemic has transformed how organizations operate, making remote work an integral part of most organizations' work culture. Unfortunately, this shift has increased the attack surface, opening more vulnerabilities and creating more opportunities for cyber-criminals to exploit. Globally, the average hourly loss in 2021 was estimated to be $787,671, due to a 125% increase in cyber-attacks compared to the previous year - a trend that is expected to continue, as per AAG Cyber Crime Statistics. According to the Mandiant Security Effectiveness Report, 2020, 53% of successful cyber-attacks go undetected within enterprises, and 91% of incidents don't trigger alerts.

This portrays a security gap within the detection capabilities of organizations. There may be two hypotheses for this gap to exist. The first one is that organizations struggle to identify ”what to detect?” and the second one is ”how to detect?”. This master's thesis aims to identify and bridge detection gaps by answering these two questions. It compares cyber-security frameworks, like MITRE ATT&CK and NIST, to suggest which framework would provide detection coverage that could be implemented and measured in real-world attacks. Additionally, it also identifies the best-in-market tool(s) which this coverage can be implemented within. The goal here is to find and pinpoint deficiencies in security tools or solutions utilized by organizations while facilitating remediation efforts with the help of open-source detection controls.

In conclusion, this document serves as a comprehensive resource for organizations seeking to enhance their detection capabilities. It highlights best-in-class tools, frameworks, and open-source detection controls while shedding light on critical gaps in current detection strategies.

Chapter 1

Introduction

Due to COVID-19, employees could not commute to offices for work, and organizations were forced to use remote work options. Organizations had no insight into or control of employee's remote/home networks. Also, employees who are not in any technical field tend to keep the bare minimum efforts and configuration to connect to the internet, which makes their home network insecure. This increased the attack surface of organizations. With something as easily attainable as default admin credentials for a home router, an attacker can access an employee's home network. From there, attackers can monitor the network traffic. They can perform various attacks, such as Man-in-the-Middle (MiTM) attacks to extract credentials of the organization's virtual private network (VPN) and get access to an employee's organization network. Once inside the network, attackers can try to execute various scripts and security tools to evade detection and perform lateral movement in the environment.

To get an understanding of the current state of detections, we look at Mandiant Security Effectiveness Report [1] confirms that 91% of attacks are not detected. We consider a hypothesis that there are four types of explanations for this. One can be that there is no detection control in place for the technique used by the attacker on the monitoring platform used by the organization. The second can be that the detection control was not configured correctly to detect the technique. An example of this would be a detection rule that identifies password-guessing attacks that might only look at attempts which are done within a certain time frame like one hour. However, if an attacker were to delay the attempt to 1 guess per hour they could evade detection from such a rule. The third case would be that the organization might not have the tool in place to detect the used technique. An example of this includes small to mid-size organizations that tend to have limited budgets. Such organizations often struggle to choose a security tool that not only fits within their budget but also effectively detects any cyber-attacks on their infrastructure. The fourth and final would be if an attacker is exploiting a zero-day vulnerability, a vulnerability in a system or device that is unknown to the parties responsible for patching or fixing the flaw, such as the vendor or developer. An exploit that attacks a zero-day vulnerability is called a zero-day exploit. These types of exploits usually don't have a specific detection control to identify such attacks. In any of the above cases, the attack from an attacker will go undetected leading to a serious problem for organizations.

Now since we've established four cases where an attack from a cyber criminal will not get detected, our next agenda would be to come up with a solution to detect these kinds of cyber-attacks. The first step in coming up with a solution has to be to identify what is the scope of techniques used by attackers in real-world cyber-attacks. Once the scope of detection is identified, we would require a security tool to which the identified scope of techniques can be applied. After identifying a security tool, we would need to figure out a way to find and configure detection controls on the tool which can detect the techniques. To achieve reproducible results that can be implemented in multiple organizations, this thesis will aim to find open-source available controls. Finally, we would compare the state of an organization's detection before and after applying the identified solutions to the detection problems. This will inform us about the added value an organization would get if they applied the solution discussed in this thesis. To answer these, we have created a set of research questions which can help organizations increase their current detection capabilities by detecting such cyber-attacks.

1.1 Research questions

To pursue our goal of bridging the security gaps, we have defined the following set of research questions:

RQ1. Which cyber-security framework is most effective in detecting an adversary's techniques in real-world cyber-attacks and why?

RQ2. Which type of tools can be best suited to monitoring an adversary's actions to provide a full view of their attack?

RQ3. What are available open-source detection controls for SIEM tools and how to map them to techniques mentioned in MITRE ATT&CK framework?

RQ4. In practice, what are the benefits of implementing MITRE ATT&CK-based detection controls on SIEM?

In related work, the discussion revolves around four pieces of research which closely relate to what this thesis is focused on achieving. Kinnunen [2] and Georgiadou et al. [3] provide a base for performing gap analysis assessments in organizations. Rabobank [4] provides information about log sources which are to be monitored for a full view of an adversary's actions. Rabobank [4] and Xiong et al. [5] both provide an idea for the enhancement of security measures/detection controls within organizations. More details about them will covered in the next chapter 2.

1.2 Contribution to Cybersecurity Research

This thesis document aims to make a significant contribution to the cybersecurity research community by providing useful insights, best-practice guidelines, and practical recommendations for the enhancement of threat detection capabilities, the proficient monitoring of adversarial activities, the considerate use of open-source assets, and the adept incorporation of detection controls based on real-world attacks. The transmission of such knowledge has the potential to significantly improve the efficacy of cybersecurity measures in practical contexts, benefiting both individual enterprises and the broader cybersecurity community.

Chapter 2

Related Work

In this chapter, the aim is to identify and discuss similar solutions or frameworks that utilize MITRE ATT&CK to find and remediate security gaps within organizations. First, a discussion on the discovery of similar solutions or frameworks is presented. This is followed by detailed descriptions of each identified research. Finally, a comparison between each research and this thesis is provided.

2.1 Identifying Similar Research

To find similar research, the search began with the Google search engine by typing the keyword ”Frameworks for Detecting Security Gaps with MITRE ATT&CK,” which yielded 291,000 results as of 14 August 2023. Within those results, a similar research published in February 2022 from Kinnunen [2] titled ”Threat Detection Gap Analysis Using MITRE ATT&CK Framework” was identified, it explores the enhancement of organizational knowledge regarding threat detection capabilities. Similar to the idea in this thesis document, it utilized the MITRE ATT&CK framework to address the challenge. Employing Design Science Research, it maps threat detection features from selected security products to this framework and performs a gap analysis. The evaluation, based on a questionnaire within the assigning organization, revealed that utilizing the MITRE ATT&CK framework helps in identifying threat detection gaps, which can be useful in improving the organization's defence capabilities. However, this thesis broadens the scope by comparing different cybersecurity frameworks like MITRE ATT&CK and NIST. It further identifies best-in-market tools for implementing detection coverage by comparing various detection tools. It also provides a method to find and implement the coverage of identified gaps.

Within the results of the Google search engine, we also identified a similar framework to our research known as DeTT&CT [4], standing for Detect Tactics, Techniques & Combat Threats, that aims to assist blue teams in using ATT&CK to score and compare data log source quality, visibility coverage, detection coverage and threat actor behaviours. The DeTT&CT framework was developed to assist blue teams in utilizing the MITRE ATT&CK framework to evaluate and enhance their visibility and detection coverage. It provides various mappings including data source mapping, visibility coverage mapping based on techniques and data sources, and detection coverage mapping based on techniques. With components like a Python tool, YAML administration files, and a dedicated editor, it facilitates swift identification and remediation of coverage gaps. Additionally, it supports threat actor group mapping and offers statistical insights on ATT&CK data source updates, helping to strengthen the protection of the organization against cyber threats. It does provide mapping data sources to MITRE ATT&CK. However, it doesn't talk about what to do after identifying those gaps in detection coverage. This thesis expands on how to use the identified gaps and implement solutions to extend detection coverage.

From the remaining results of a Google search, most were blogs discussing the utility of MITRE ATT&CK. Hence, the focus was shifted to a more research-centric search using Google Scholar, a freely accessible web search engine indexing the full text or metadata of scholarly literature across the web. This engine encompasses peer-reviewed online journals, theses, conference papers, patents, technical reports, and books. It allows users to search for publications, view citations, related articles, and referenced books, proving to be a useful tool for academic and professional research. The search began with the keywords ”Security Gaps MITRE ATT&CK,” yielding 2710 results as of 15 August 2023. The initial two pages (20 results) were downloaded, and the abstract of each was read. Among these, another two related works by Georgiadou et al. [3] and Xiong et al. [5] were discovered.

The research paper ”Assessing MITRE ATT&CK Risk Using a Cyber-Security Culture Framework” by Georgiadou et al. [3] delves into a novel exploration of combining organizational and individual culture factors with security vulnerabilities through the lens of the MITRE ATT&CK framework. The paper endeavours to fill a noticeable gap in the existing literature by proffering a cyber-security culture framework that thoroughly associates these culture factors with security vulnerabilities, mapped to adversary behaviours and patterns within the MITRE ATT&CK framework. This framework is crafted with a focus on critical infrastructures, particularly the energy sector, which shows a clear interaction between Information Technology (IT) and Operational Technology (OT) networks. By emphasizing a hybrid approach including both the enterprise and Industrial Control Systems (ICS) within the MITRE ATT&CK framework, the paper presents a more holistic methodology. The paper stands out by exploring a new area of science that hasn't been explored before (as per the research) in exploiting the MITRE ATT&CK framework for security assessment and defensive design, thus laying down a robust foundation for future endeavours

Aspect	Kinnunen, Jarkko (2022)	DETT&CT (Rabobank)	This Thesis
Core Framework	MITRE ATT&CK	MITRE ATT&CK + DETT&CT	MITRE ATT&CK
Focus	Gap Analysis Based on Mapping Specific Products to MITRE ATT&CK	Detection and Visibility Based on Log Source	Gap Analysis Based on Mapping Detection Controls to MITRE ATT&CK
Evaluation Method	Analysis based	Structured Evaluation	Analysis based
Improvement Aspect	Enhancing Threat Detection	Enhancing Detection Coverage	Enhancing Detection Coverage

Aspect	Anna Georgiadou et. al. (2021)	Xiong et. al (2022)	This Thesis
Core Framework	MITRE ATT&CK	MITRE ATT&CK	MITRE ATT&CK
Focus	Risk Assessment	Threat Modelling Language	Gap Analysis Based on Mapping Detection Controls to MITRE ATT&CK
Evaluation Method	Culture-Based Evaluation	Meta Attack Language Framework	Analysis based
Improvement Aspect	Enhancing Risk Assessment	Enhancing Security Measures	Enhancing Detection Coverage

Table 2.1: Comparison between related work and this thesis

in enhancing cybersecurity measures in critical infrastructures. The research by Georgiadou et al. [3], similar to this thesis, utilizes MITRE ATT&CK to identify security gaps in organizations. However, it diverges from this thesis since the outcome of this thesis focuses on helping organizations broaden their current detection coverage, whereas the research by Georgiadou et al. [3] aims to provide a security assessment concerning the aspect of human behaviour.

The research paper titled ”Cyber security threat modelling based on the MITRE Enterprise ATT&CK Matrix” by Xiong et al. [5] proposes a threat modelling language aimed at proactively addressing security issues within organizations. This threat modelling language is based on the MITRE ATT&CK Enterprise Matrix same as this thesis, however, is created using the Meta Attack Language framework which is different from this work. The primary focus of this threat modelling language is to outline system assets, explain attack steps, outline defences, and establish asset associations. The paper emphasizes the increasing complexity of enterprise systems, especially with the adoption of cloud and mobile services, which significantly expands the attack surface. By leveraging available tools, the proposed threat modelling language facilitates attack simulations on enterprise systems to enhance security measures.

Among the rest of the relevant results from Google Scholar, most of the research papers were using MITRE ATT&CK for risk assessment same as in research by Georgiadou et al. [3] with only a difference in methods. Therefore, the search for related work was limited to these four research. An overview of similarities and differences can be observed in table 2.1. Table 2.1 explains the core framework, focus, evaluation method, and improvement aspect used in each related work and compares it to work in this thesis. The next chapter will initiate this thesis by answering RQ1.

Chapter 3

Identifying Framework

The goal of this chapter is to answer RQ1, it discusses identifying a framework that will be used to provide a scope for detection. We require an industry-approved framework that can be also used to reference/map for the current detection capabilities.

3.1 Cybersecurity frameworks

A cybersecurity framework is a structured set of guidelines that help an organization develop a strong foundation for managing cybersecurity-related risks more effectively. Such frameworks offer organizations a way to manage and mitigate risks, protect valuable assets, and ensure that they are aligned with the overall business objectives and regulatory requirements.

To find the available frameworks, we started looking online by Googling ”Cyber Security Frameworks”. After going through the results of said search we found the following relevant types of cybersecurity frameworks which are used by organizations:

- **MITRE ATT&CK**:

  MITRE ATT&CK (Adversarial Tactics, Techniques, & Common Knowledge) is a global database consisting of adversary tactics, techniques, & procedures that are taken from real-world observations. ”Tactics” represent the adversaries’ objectives like Initial Access, Lateral Movement., etc. ”Techniques” describe how adversaries achieve those objectives like Phishing, BruteForce., etc.”Procedures” are the specific steps adversaries take to execute techniques. From this database, an organization can look for specific techniques used by adversaries in the wild and build a monitoring use case around them. In 2023, MITRE ATT&CK contains three different matrices:

  - Enterprise: This matrix contains tactics & techniques for Windows, macOS, Linux, PRE, Azure AD, Office 365, Google Workspace, SaaS, IaaS, Network, and Containers platforms. The current model version, released on September 01, 2023, incorporates 14 enterprise tactics analyzed into 196 techniques and 411 sub-techniques, provisioning 43 mitigations. [6]

  - Mobile: This matrix contains tactics & techniques for Android, iOS platforms. The current version, released on September 01, 2023, consists of 12 tactics analyzed into 66 techniques and 41 sub-techniques addressed by 11 mitigations. [6]
  - Industrial Control System (ICS): This matrix contains tactics & techniques for devices used in an industrial plant like Programmable Logic Controller (PLC) and others. Its current version (updated on September 01, 2023) consists of 12 tactics, 81 techniques, and 52 mitigations. [6]

- **NIST (National Institute of Standards and Technology)**:

  The NIST Cybersecurity Framework, created by the U.S. National Institute of Standards and Technology, assists organizations in beginning or enhancing their cybersecurity programs. It's a voluntary guide comprising standards, guidelines, and practices aimed at better managing cybersecurity risks. Initially aimed at critical infrastructure operators, it's now utilized by various organizations to assess their risks. The framework encapsulates five key areas: Identify, Protect, Detect, Respond, and Recover, aiding organizations in prioritizing their cybersecurity efforts to improve their posture. [7]

- **ISO/IEC 27001**[8]:

  ISO/IEC 27001 is a globally recognized standard for managing information security, formulated jointly by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC). Initially published in 2005, it underwent revisions in 2013 and 2022 to stay updated with the evolving cybersecurity landscape. The standard aims at aiding organizations of any size across any industry in safeguarding their information in a structured and cost-effective manner by adopting an Information Security Management System (ISMS) [9].

The next section will compare the identified frameworks which will lead to selecting a desired framework for this thesis.

Aspect	MITRE ATT&CK	NIST	ISO/IEC 27001
Focus Area	Tactical/Operational	Strategic	Strategic
Scope	Threat Intelligence	Organizational Cybersecurity	Organizational Cybersecurity
Use Cases	Red/Blue Teaming, Threat Intelligence	Risk Management, Compliance	Risk Management, Compliance
Flexibility	High (Dynamic)	Moderate (Stable)	Moderate (Stable)
Geographical Applicability	Global	Primarily US (Global applicability)	Global

Table 3.1: Comparison of Cybersecurity Frameworks

3.2 MITRE ATT&CK vs NIST vs ISO/IEC 27001

After identifying the available frameworks, we need to find which one of the frameworks can used for the purpose of this research which is to measure security gaps in detection controls based on the attacks that are seen in the real world. For this purpose, we classified the frameworks in the following sections:

- **Focus Area:** The primary objective or purpose of the framework.
- **Scope:** The range of cybersecurity aspects covered by the framework.
- **Use Cases:** Typical scenarios or situations where the framework is applied.
- **Flexibility:** The adaptability of the framework to new threats and organizational changes.
- **Geographical Applicability:** The regions where the framework is commonly used or recognized.

From the table 3.1, for the purpose of this thesis we choose MITRE ATT&CK as the framework we will use going forward. The details of why we chose this are as follows:

- MITRE ATT&CK is an operational framework that suggests that it can be easily applied to the current set of detection controls and focuses on day-to-day security activities, technical controls, and immediate threat responses. Whereas, NIST and ISO/IEC 27001 are strategic frameworks, their main focus is long-term security vision, policy development, and organizational objectives in cybersecurity, ensuring alignment with business goals and risk management.

- MITRE ATT&CK is scoped by threat intelligence and updated regularly with techniques used in real-world cyber-attacks.
- MITRE ATT&CK can be used by red (offensive)/ blue (defensive) for gathering information on how attackers operate. Whereas, NIST and ISO/IEC 27001 are mostly used in risk management and compliance area.
- MITRE ATT&CK is updated frequently whereas NIST and ISO/IEC 27001 are updated very rarely.
- MITRE ATT&CK and ISO/IEC 27001 are frameworks that are used and maintained by global researchers, whereas NIST is primarily based in the US.

3.3 Conclusion

The goal of this chapter was to identify a cybersecurity framework that will be most effective in detecting adversary's techniques in real-world attacks [RQ1], from the information provided in this chapter we were able to conclude that MITRE ATT&CK can be utilized to detect and measure security gaps in the current set of detection controls. The fact that MITRE ATT&CK has a matrix, can help in creating heat maps with the current set of techniques used by attackers. This will make it easy to identify the techniques which are used by attackers but are not monitored by organizations. In the following chapter, we will discuss which security tool will be most beneficial for organizations to identify the majority of techniques mentioned in MITRE ATT&CK.

Chapter 4

Identifying Detection Tool

The goal of this chapter is to answer RQ2. From the last chapter, we were able to identify MITRE ATT&CK as a reference framework which we are going to use forward. In this chapter, we want to identify a detection tool that can be utilized by organizations to detect most, if not all techniques mentioned in the MITRE ATT&CK Enterprise matrix.

4.1 Cybersecurity Detection Tool

A detection tool in cybersecurity is specialized software designed to identify, monitor, and analyze malicious activities or vulnerabilities within an organization's network or systems. By continuously observing system behavior and traffic, these tools help in recognizing unusual patterns, potentially harmful anomalies, and known threats, enabling timely defensive actions to protect sensitive data and maintain system integrity.

We used the same method to find the tools in cybersecurity that are used for detection. However, this time instead of using Google we switched to Bing as with the help of GPT-4 it has the ability to go through the results of the searched query to collect and answer the exact information that is needed. After entering the following keywords "What are types of cybersecurity detection tools". We got a list of the following 8 types of detection tools which were collected from multiple sources:

1. **Penetration testing tools**: These tools are used to simulate cyber attacks on a system to identify vulnerabilities and weaknesses. Examples include Kali Linux and Metasploit [10].

2. **Intrusion detection and prevention systems**: These tools monitor network traffic for signs of malicious activity and can either alert security personnel or block the traffic altogether. Examples include Snort and Suricata [11].

3. **Security information and event management (SIEM) systems**: SIEM is a security management system that combines security event management (SEM) with security information management (SIM). It gathers event log data from many sources, analyses it in real-time to spot activity that differs from the usual, and then takes the necessary action.[12]. An example of a SIEM solution is Microsoft Sentinel. Examples include Splunk and IBM QRadar [10].

4. **Endpoint detection and response (EDR) systems**: These tools monitor endpoints such as laptops, desktops, and servers for signs of malicious activity. Examples include Carbon Black and CrowdStrike [10].

5. **Threat intelligence platforms**: These tools provide real-time information about emerging threats, allowing organizations to take proactive measures to protect their systems. Examples include Recorded Future and Anomali [13].

6. **Vulnerability scanners**: These tools scan networks and systems for known vulnerabilities that can be exploited by attackers. Examples include Nessus and OpenVAS [10].

7. **Firewalls**: These tools monitor incoming and outgoing network traffic to block unauthorized access to a system or network. Examples include Cisco ASA and Fortinet FortiGate [10].

8. **Antivirus software**: This software is used to detect, prevent, and remove malware from a system or network. Examples include Norton Antivirus and McAfee Antivirus [10].

From the types of detection tools identified above, a selection will be made for a tool that aids this thesis. The selection will be based on coverage of MITRE ATT&CK for these tools that will be explored in the following section.

4.2 ATT&CK Coverage Of Detection Tools

Out of the above-mentioned detection tools, we would eliminate penetration testing tools as those are used by red team operators and they cannot be used to detect threats in a live environment rather they only inform the weaknesses in the current state of the environment. We would also eliminate threat intelligence platforms as those are a supplement to an existing detection tool. They only provide information that can be detected instead of detecting threats themselves. Lastly, we eliminate the Vulnerability Scanner as it detects vulnerabilities in the current system and does not monitor for an adversary's behavior. For the other 5 types of tools, in order to

Detection Tool Type	Supported ATT&CK Data Sources	# of Supported Data Sources
IDS/IPS	Network Traffic	1
SIEM	All mentioned except Persona, Internet Scan and Domain Name	34
EDR	All mentioned except Persona, Sensor Health, Network Traffic, Network Share, Internet Scan and Domain Name	31
Firewalls	Firewall, Network Traffic, Network Share	3
Antivirus	File, Driver, Malware Repository, Script	4

Table 4.1: Detection Tools Mapping To Supported Data Sources

find a detection tool that can cover most techniques, we will need to identify how many techniques can each type of tool detect. There were two ways identified for this purpose:

1. Compare each technique with the detection tool to see if it can detect that technique or not.

2. Group techniques into data sources and they check if that data source can be ingested by the detection tool or not.

Comparison with each technique would be time-consuming as there are a total of 196 techniques in the Enterprise matrix. Grouping techniques into data sources is more efficient and will reduce this time as MITRE already has grouped techniques into a total of 41 data sources. However, this number is for all 3 matrices. In this thesis, we are only analyzing the enterprise matrix which reduces the total number of data sources to 37. In the table 4.1, we are able to observe that among all the identified detection tools, the SIEM tool is able to ingest/support the most number of data sources mentioned in the list of MITRE ATT&CK data sources [14]. Hence, we will have identified SIEM as our detection tool. Next, we wanted to identify which SIEM solutions are widely used in the market.

4.3 Which SIEM to choose?

For finding an SIEM solution that is widely accepted, this thesis refers to Gartner, an IT consultancy and advisory firm based in the US. It works closely with organizations to develop technology strategies, plans, and budgets, assisting them in selecting the right technologies for their operations. The firm employs a robust research methodology that involves engagement with industry experts, primary data collection,

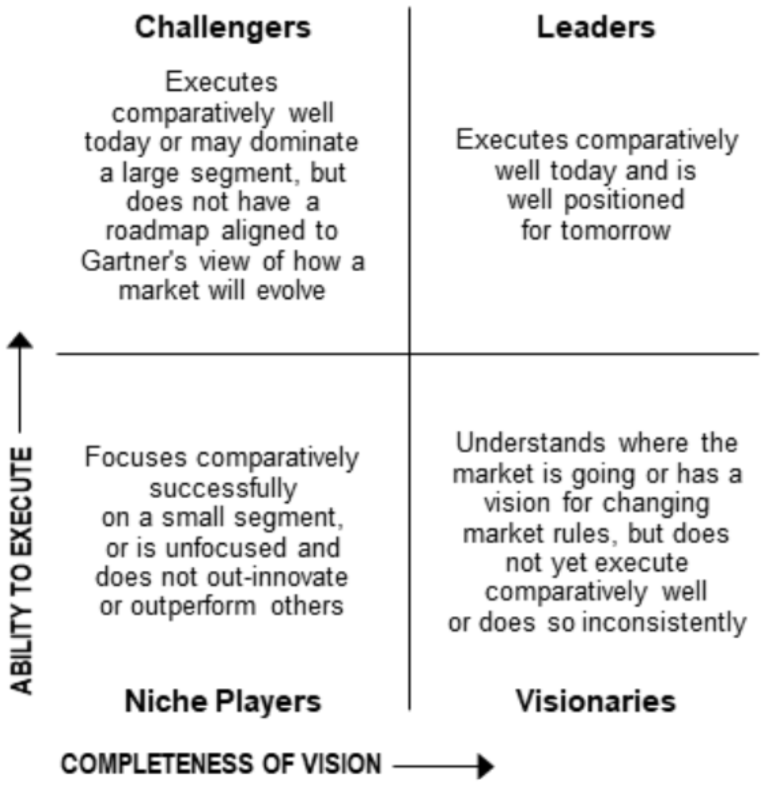
Challengers
Executes comparatively well today or may dominate a large segment, but does not have a roadmap aligned to Gartner's view of how a market will evolve
Leaders
Executes comparatively well today and is well positioned for tomorrow
Focuses comparatively successfully on a small segment, or is unfocused and does not out-innovate or outperform others
Understands where the market is going or has a vision for changing market rules, but does not yet execute comparatively well or does so inconsistently
Niche Players
Visionaries
ABILITY TO EXECUTE
COMPLETENESS OF VISION

Figure 4.1: Different Categories in Gartner and their meaning [16]

and meticulous analysis. This approach yields insights that are highly regarded and utilized worldwide. Gartner's reports, which are frequently updated to maintain relevance and accuracy, have become indispensable resources for business leaders and industry professionals. Among these reports, the renowned Gartner Magic Quadrant is an industry standard for evaluating and comparing technology products and services, thereby reinforcing Gartner's position as a trusted advisor in the IT sector. Gartner's 2022 report [15] provides expert guidance on such tools and divides them into four categories as mentioned below:

- Leaders: Microsoft Sentinel, IBM QRadar, Splunk, Exabeam & Securonix.
- Challengers: LogRhythm, Rapid7, Fortinet & Devo
- Visionaries: Gurucul, Sumo Logic, Elastic & Micro Focus.
- Niche Players: Logpoint, ManageEngine & Huawei

To understand the meaning of different categories, Gartner provides the following chart provided in figure 4.1. From the information provided by Gartner's Magic Quadrant, we can observe that the following SIEM solutions are the leaders:

- **Microsoft Sentinel:** It is a Security Information and Event Management (SIEM) product that runs in the cloud. It integrates SIEM and SOAR features to provide a single platform for threat detection, investigation, and response. As of October 15, 2023, it has 241 data connectors available that can be found on the content hub page. The query language used by Microsoft Sentinel is called Kusto Query Language (KQL) which helps in defining analytic and hunting rules to co-relate data from multiple data sources. [17]

- **IBM QRadar**: It is a Security Information and Event Management (SIEM) product that runs in the cloud as well as on-premise. As of October 15, 2023, it has 111 data connectors available that can be found on the IBM App Exchange [18]. QRadar employs Ariel Query Language (AQL) and QRadar Network Packet Capture Query Language (NTQL) for data retrieval and analysis.

- **Splunk**: It is a platform specializing in log management and data analytics for security purposes. It aggregates log data, security alerts, and events into a centralized platform, enabling real-time analysis for security monitoring. It operates by collecting, analyzing, and correlating network and machine data in real-time. It offers deployment options either on-premises or in the cloud. [19]

- **Exabeam**: It is designed to operate at cloud-scale, capable of ingesting, parsing, storing, searching, and reporting on petabytes of data from various sources. It provides integrations from 549 security as of October 15, 2023. This design delivers processing at over one million events per second [20]

- **Securonix**: It is a Security Information and Event Management (SIEM) product that runs in the only in cloud as a SaaS (Software-as-a-Service). It employs a Hive Query Language (HQL) which is SQL-like for querying its data. [21]

This means these SIEM solutions are widely accepted as go-to SIEM solutions when an organization plans to invest in SIEM. However, when we look at the score obtained by each of the SIEM tools in Gartner's report, we get to see the graph in figure 4.2. Through this scoring by Gartner, Microsoft Sentinel was the best SIEM solution for the year 2022. Hence, based on this industry-accepted report we chose Microsoft Sentinel as the SIEM solution that we are going to use in this thesis.

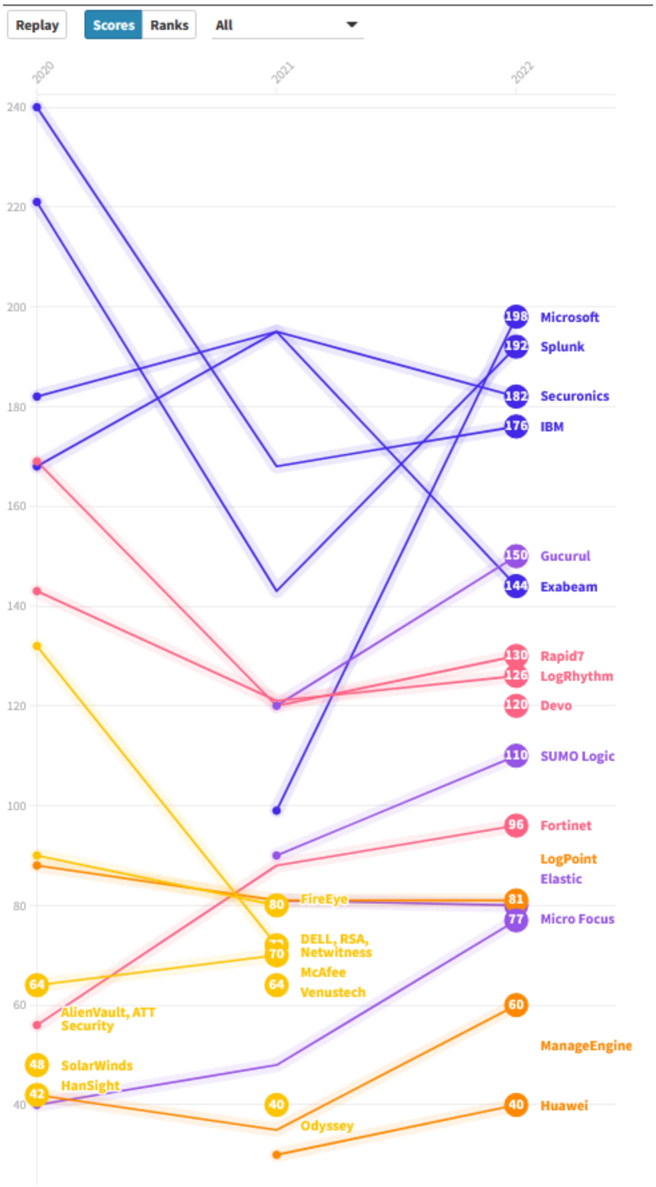

Figure 4.2: Gartner Scores for each SIEM provider for the year 2020-22 [15]

4.4 Conclusion

The goal of this chapter was to answer RQ2, for that it discussed multiple types of detection tools and identified SIEM as the best detection tool to monitor an adversary's actions. SIEM is able to do this due to having the capability to ingest 34 ATT&CK-mentioned log sources. We also identified Microsoft Sentinel as a widely accepted SIEM solution as per Gartner. Hence, we chose Microsoft Sentinel to perform our further analysis in this thesis. In the next chapter, we will look at detection controls for Microsoft Sentinel.

Chapter 5

Open-Source Detection Controls

The goal of this chapter is to answer RQ3. In the last chapter, we chose the Microsoft Sentinel SIEM tool as our detection tool that can detect and cover maximum techniques in the MITRE ATT&CK framework as compared to any other detection tool. In this chapter, we discuss what are available open-source detection controls for Microsoft Sentinel and how to find them. Once we find the detection controls from various sources, we select a trusted source that is also utilized widely by organizations to implement detection controls on their Microsoft Sentinel SIEM. Therefore, we look for sources of available KQL rules over the internet.

5.1 Finding KQL Rules

In this section, we discuss how we found open-source KQL rules that can be used to detect cyber threats. The initial search utilized the GitHub platform as it is the most prominent openly available code repository. We searched the keywords ”Azure Sentinel” which resulted in 642 results (as of 15 August 2023). The first repository was the official repository of Microsoft itself for Sentinel. Out of the files and folders that were present in the repository, we identified two folders called ”Detections” and ”Hunting Queries” which had in total of 969 KQL queries/rules in YAML Format. We pulled these KQL rules and put them all in one folder for data analysis. A Python script (”Identifying_mapped_rules.py” located at [22]) was created to analyze the data to check if all the KQL rules were mapped to the MITRE ATT&CK framework or not. We found the following data as per the table 5.1.

Mapping Status	Count of KQL rules
Mapped to MITRE ATT&CK	353
Not-mapped to MITRE ATT&CK	616

Table 5.1: Count of mapped/not-mapped KQL rules to MITRE ATT&CK

In the aforementioned search, a greater number of GitHub repositories were discovered containing KQL rules. However, to prevent the inclusion of duplicate KQL rules and under the assumption that organizations would favour official data sources over lesser-known ones, we opted to solely consider the official Microsoft repository for Sentinel as our source for open-source detection controls. However, some of the GitHub repositories were starred (considered good) by multiple people are mentioned in the table 5.2. This can be utilized by future researchers in this area.

Source	Number of Detection Controls
Azure [23] GitHub	969
Zorich [24] GitHub	443
Trent [25] GitHub	321
Canos [26] GitHub	180
Pals [27] GitHub	169
Koc [28] GitHub	59
Total	2141

Table 5.2: Number of Detection Controls with Source

From the table 5.1, it was observed that 616 KQL rules are not mapped to the MITRE ATT&CK framework. The next section discusses how to map these unmapped KQL rules to the MITRE ATT&CK framework.

5.2 Mapping KQL Rules to MITRE ATT&CK Framework

This section discusses how to map the 616 KQL rules that are not mapped to MITRE ATT&CK. We started by looking on the internet for an available solution to map a KQL query to a relevant MITRE ATT&CK technique. For this, we searched on Google with the keywords ”how to map kql query to technique id in mitre att&ck”, however upon going through the results of the search most results were related to already mapped KQL queries. There was one tool identified by MITRE Engenuity known as Threat Report ATT&CK MAPPER (TRAM) [29] which uses pre-trained data to map threat intelligence reports to its relevant technique ID in MITRE ATT&CK framework. However, after testing the tool on KQL query files gathered from the GitHub repository of Sentinel. The tool always seems to show an error, even when we change the format to .pdf or .txt from YAML format.

Finally, two ways were identified for mapping not-mapped KQL rules to an MITRE ATT&CK technique ID:

1. Manually going through the KQL query identifying what actions it monitors and searching a relevant MITRE ATT&CK technique ID in the official MITRE ATT&CK website [6]. This is time-consuming but would be more precise.

2. Utilizing the Bing AI ChatBot, a feature provided by Microsoft, allows users to interact with an AI-powered assistant capable of assisting with various tasks such as answering questions and performing searches [30]. The process involves searching the web using keywords specified in the KQL query, coupled with a pre-defined string instructing the ChatBot to map them to a relevant technique ID in MITRE ATT&CK. While this method is relatively fast, it tends to be less accurate 5.3.

This thesis didn't use any other AI bot such as ChatGPT for mapping, as it only contains information till September 2021 and MITRE updates its ATT&CK framework on a regular basis. Before using Bing AI ChatBot, the aim was to find with what accuracy can Bing AI ChatBot map these KQL rules. To answer this next section discusses how this thesis tests for the accuracy of said method of mapping.

5.3 Bing AI Accuracy

This section discusses the test for accuracy of the Bing AI ChatBot. Initially, a ground truth set of KQL rules, which were already mapped to the MITRE ATT&CK framework, was created. A Python script, ”Ground-truth-accuracy.py” (available at GitHub [22]), was then written using the `sydney.py` library and adhering to algorithm 5.1.

The algorithm 5.1 iterates through every file ending with .yaml or .yml in a directory, opens each file, and checks for a 'query' value. This query value is assigned to a string variable, and a custom string is appended to this variable to request Bing AI to map the query to its relevant MITRE ATT&CK technique. Subsequently, the string variable is passed to the `SydneyClient` function, which sends the entire string to an API handling Bing AI's chat feature through a POST request. However, due to its unofficial status, the API experienced multiple crashes during the research since Microsoft appears to classify it as bot behaviour, applying CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) checks which prevent the script from sending any data to API Client.

Examining the data that was able to be mapped from table 5.3, it can be confirmed that only 10/25 KQL rules were correctly mapped using this method, yielding an accuracy percentage of 40%. Given that this percentage is too low to be reliable, the decision was made to map the techniques through a manual process, which may provide greater accuracy compared to the Bing AI ChatBot.

KQL Query	Mentioned Technique ID	Bing (GPT-4) Technique ID
AADHealthMonAgentRegKeyAccess	T1005	T1202
AbnormallyLargeJPEGFiledDownloadedfromNewSource	T1001	T1078
AccessibilityFeaturesModification	T1546	T1546
AccountAddedtoPrivilegedPIMGroup	T1098	T1098
AccountElevatedtoNewRole	T1078	T1098
AdditionofaTemporaryAccessPasstoaPrivilegedAccount	T1078	T1550
ADFSDBLocalSqlStatements	T1005	T1200
AnomalousUserAppSigninLocationIncrease	T1078	T1078
AnomalousUserAppSigninLocationIncreaseDetail	T1078	T1018
AnomolousSignInsBasedonTime	T1078	T1078
AppGwWAF-SQLiDetection	T1211	T1505
ApplicationIDURIChanged	T1078	T1136
ASR–Rule-Ransomware-triggered	T1486	T1486
AuditPolicyManipulation_using_auditpol	T1204	T1484
AuthenticationAttemptfromNewCountry	T1078	T1078
AuthenticationMethodChangedforPrivilegedAccount	T1098	T1550
Azure-CloudShell-Usage	T1059	T1530
AzureStorageFileCreatedQuicklyDeleted	T1020	T1564
AzureStorageFileOnEndpoint	T1570	T1567
B64IPInURLFromMDE	T1071	T1071
ChangestoApplicationLogoutURL	T1078	T1098
ChangestoApplicationOwnership	T1078	T1098
ClientIPwithManyUserAgents	T1190	T1190
ConditionalAccessPolicyModifiedbyNewUser	T1078	T1078
Crashdumpdisabledonhost(ASIMVersion)	T1070	T1070

Table 5.3: KQL mapping accuracy results with Bing AI (GPT-4)
Red: Wrongly-mapped, Green: Correctly-mapped

It is required to know since Bing results are based on the web search results for the KQL query it changes when prompted again after a certain number of days. For instance, the technique ID for the ”Crash dump disabled on host (ASIM Version)” rule was T1112 on September 12, 2023. However, the same rule was classified as T1070 on October 10, 2023, which accurately reflects the ground truth value. Consequently, it is plausible that in the future, this accuracy may improve to a percentage more acceptable than the current rate.

Algorithm 5.1 Mapping KQL Query to MITRE Technique ID

```
Input: Path to the directory containing YAML files
Output: Technique ID from Bing AI mapped query
Set Bing U Cookie in environment variables
Define query_string and technique_id as global empty strings
procedure SYD
    Start asynchronous SydneyClient session named sydney
    Define prompt as query_string
    if prompt equals ”!reset” then
        Reset sydney conversation asynchronously
    end if
    Send asynchronous request: sydney.ask_stream(prompt)
    Await and print responses continuously until request completion    ▷ Pattern matching logic commented in code is skipped here
end procedure
for each filename in specified directory do
    if filename ends with ’.yaml’ or ’.yml’ then
        Read data from file using SafeLoader
        if key ’query’ is present in data then
            if data[”query”] is non-empty and non-None then
                for each line query_data in data[”query”] do
                    if query_data does not contain ”//” then
                        Append query_data to query_string
                    end if
                end for
                Append technique ID retrieval request to query_string
                Run SYD
            end if
        end if
    end if
end for
```

5.4 Conclusion

The goal of this chapter was to answer RQ3 that is related to identifying available open-source detection controls and mapping them to relevant MITRE ATT&CK technique. In this effort, 2141 detection controls in total, also known as KQL rules, for Microsoft Sentinel were identified, out of which 969 were chosen for this thesis as they were from official Microsoft sources. From the 969 KQL rules, 616 were not mapped to the MITRE ATT&CK framework by default. Consequently, two methods for mapping the KQL rules to the MITRE ATT&CK framework were explored. The first one was manual mapping which was less efficient due to its time consumption but provided greater accuracy. The second one was the use of artificial intelligence (AI) chat-bots like Bing AI Chat which was more efficient, however, the accuracy was at 40% based on the results from table 5.3. With this accuracy, the mapping of KQL analytic rules to the MITRE ATT&CK framework would not be beneficial as more than half would be wrongly mapped. Therefore, the choice was made to proceed with the manual mapping of KQL analytic rules to the MITRE ATT&CK framework. These manually mapped KQL are a contribution to the cybersecurity research community which can be used in future research in the area. In subsequent chapters, this data will be employed to identify gaps in detection by comparing it with real-world data from organizations.

Chapter 6

Identifying & Bridging Security Gaps

This chapter aims to address the RQ4 by identifying gaps in the detection controls of four pseudonymized organizations, referred to as Company1, Company2, Company3, and Company4, to maintain confidentiality. Real-world data, kindly provided by Computest Security [31], has been utilized to map the existing KQL analytic rules in each organization's SIEM tool against the MITRE ATT&CK framework. An overview of what this thesis discussed and will discuss further can be observed in figure 6.1.

The analysis involves creating heatmaps of the MITRE ATT&CK framework for each organization, highlighting the techniques that are currently not monitored. These heatmaps will then be compared and discussed with another, created using open-source data collected in a previous chapter 5. Such a comparison aims to reveal techniques that, while having available open-source detection controls, are still not monitored by the organizations.

The research analysis extends to the application of this newfound knowledge within the organizations' current environments. It involves the potential implementation of open-source KQL rules, depending on the availability of relevant data sources within each organization. Furthermore, in instances where specific techniques lack open-source KQL rules but have applicable data sources present, there is an opportunity to develop customized KQL rules. The next section will introduce information about the organizations in this research and which industry they belong to.

6.1 Selected Organizations

As previously mentioned, this research employs real-world data from four organizations to unveil security gaps within their respective environments. These organizations hail from diverse industries, thereby offering a glimpse into the state of detection controls across a spectrum of sectors.

Table 6.1 represents the currently active KQL rules employed by these orga-

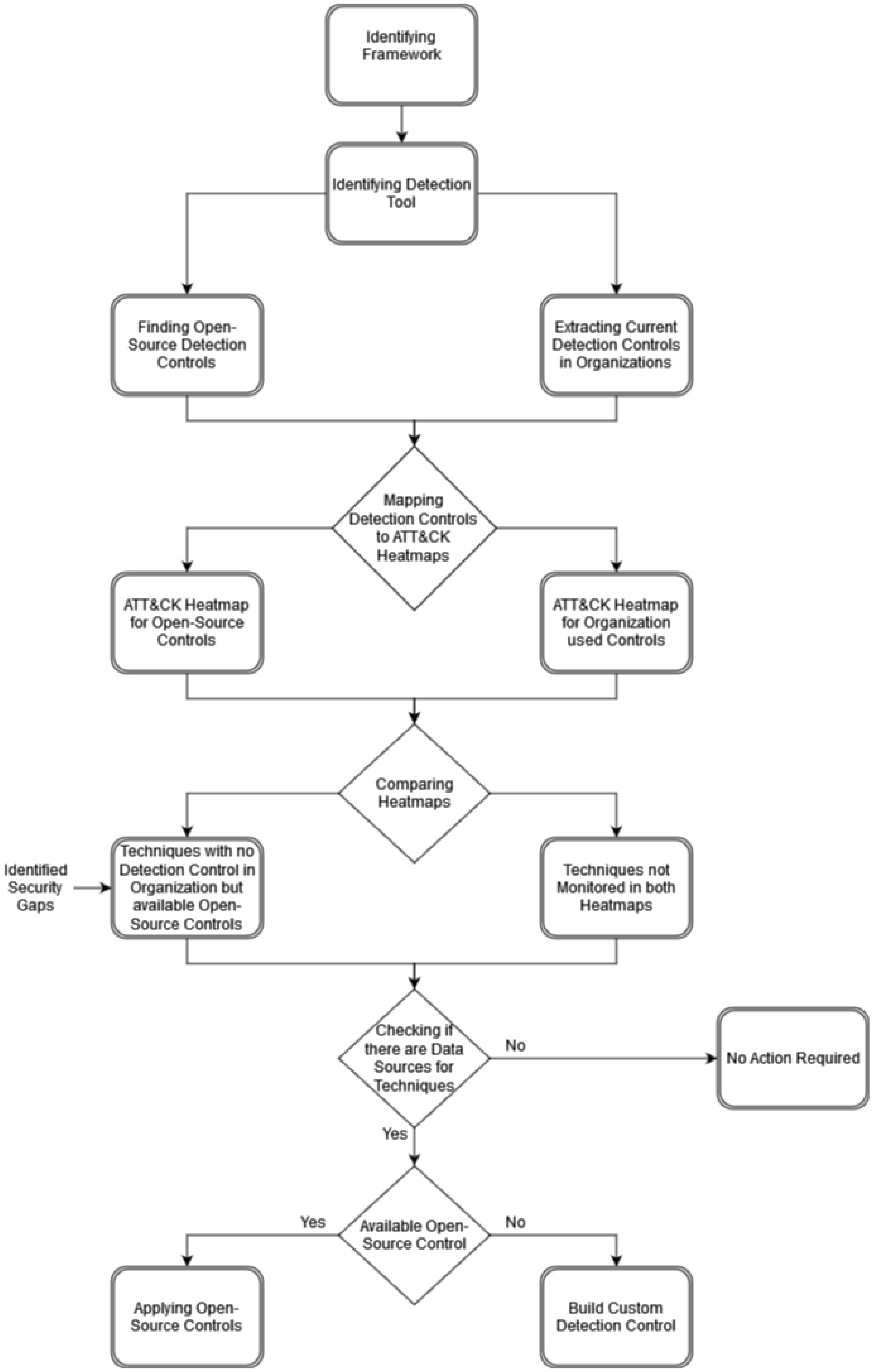
Identifying Framework
Identifying Detection Tool
Finding Open-Source Detection Controls
Extracting Current Detection Controls in Organizations
Mapping Detection Controls to ATT&CK Heatmaps
ATT&CK Heatmap for Open-Source Controls
ATT&CK Heatmap for Organization used Controls
Comparing Heatmaps
Identified Security Gaps
Techniques with no Detection Control in Organization but available Open-Source Controls
Techniques not Monitored in both Heatmaps
Checking if there are Data Sources for Techniques
No
No Action Required
Yes
Yes
Available Open-Source Control
No
Applying Open-Source Controls
Build Custom Detection Control

Figure 6.1: Overview of work performed in this thesis

nizations for cyber-attack monitoring. It should be noted that when the Microsoft Sentinel deployment team of each of these organizations were inquired about the source of these KQL rules, they disclosed that these were obtained from the Microsoft Sentinel's Official GitHub repository [23]. This repository is also the source of the open-source KQL rules utilized in this thesis. Also, some amount of KQL rules are custom-made, and this amount differs in each organization. Furthermore, the table 6.1 specifies the industry to which each organization belongs. Hence, the industries examined in this study encompass Management Consulting, Software Development, IT Consulting, and Law. The next section will discuss the methodology that this research utilized for generating heatmaps.

Organization	Count of KQL Rules	Industry
Company1	238	Management Consulting
Company2	384	Software Development
Company3	265	IT Consulting
Company4	241	Law

Table 6.1: Count of Detection Controls (KQL Rules) in Organizations

6.2 Heatmap Generation

Before starting the creation of heatmaps, it is necessary to download the current set of analytic rules from the Microsoft Sentinel environment of each organization. This can be accomplished using one of two methods:

- The steps for exporting using the Sentinel platform can be found at the official Microsoft link [32].
- If Sentinel is integrated with Azure DevOps, the files can be downloaded by visiting `dev.azure.com` and navigating to the `AnalyticRules` and `AnalyticRules-ARM` folders.

Once the rules folder is downloaded, the GitHub repository [22] for this thesis should be cloned. The folder location for the rules folder is to be pasted into the `directory_path` variable before running the Python file ("Heatmap_creation.py"). More details about the Python file will be discussed in a sub-section below 6.2.1. Upon successful execution of the code, the `heatmap.json` file will populate with techniques and a color scheme, whereby a higher rule count for a technique assigns it a darker color. This JSON file can be uploaded to MITRE ATT&CK Navigator [33] which is a web-based tool from MITRE that allows users to visualize and explore the

MITRE ATT&CK framework. The tool is designed to help users better understand the framework and how it can be used to improve cybersecurity. An example of this is that the MITRE ATT&CK Navigator displays techniques used by various advanced threat persistent (APT) groups in real-world attacks conducted by them.

Colors are assigned to a technique based on the count of analytic rules as follows:

Count of KQL Analytic Rules	Assigned Colour
1	light grayish blue
2	moderate blue
3	soft blue
>3	cyan-blue

Table 6.2: Colour Coding Information for Heatmaps

The next sub-section provides the details about the Python file that is used for the generation of heatmaps.

6.2.1 Heatmap_Creation.py

To facilitate the mapping of the current set of KQL analytic rules used for detection with the MITRE ATT&CK framework, a script has been written in Python, named `Heatmap_creation.py`. This script is accessible on GitHub [22]. The script takes a directory containing all analytic rules as its input. Following this, it populates a pre-existing JSON file, located within the same directory and named `heatmap.json`, with various techniques and their associated color mappings. This resulting JSON file can be uploaded to the MITRE Navigator [33] for visualization purposes and can also be downloaded as a heatmap. The mechanism behind this can be elaborated upon through the algorithm 6.1, explained in the following steps:

1. **Data Extraction:**
   (a) **Iterate through Files:** The algorithm iterates through every file in a specified directory, specifically targeting files with a ".json" extension. This approach is adopted because when KQL rules are exported from Microsoft Sentinel, they are provided in a .json format.
   (b) **Extract Techniques:** For each JSON file, it extracts technique IDs mentioned under the 'relevantTechniques' or 'techniques' key, specifically taking the first 5 characters of each ID as the focus is only on technique and not sub-technique.
2. **Data Analysis:**

(a) **Technique Frequency Count:** Utilizing the Counter class, it calculates the frequency of each extracted technique ID.

(b) **Technique Mapping:** Calls the map_attack_technique function to map each technique ID to a human-readable name and tactic name by referring to another JSON file (mitre.json). If no match is found, a warning is output.

3. **Data Visualization:**

   (a) **Heatmap Data Preparation:** Invokes the heatmap function to update a JSON file (heatmap.json) intended for visualization purposes (specifically, a heatmap). The heatmap encodes frequency information via color coding. More information about how color coding can be interpreted is available in table 6.2.

Through the process in this section, heatmaps for open-source data as well as four organizations were generated. The next section will discuss details about the generated heatmaps.

6.3 Heatmaps Discussion

In this section, the heatmaps will be discussed and compared to each other. Before talking about specific heatmaps one clearly visible thing from all heatmaps is that none of them cover any technique in the reconnaissance tactic. The reason behind this can be hypothesized by understanding the techniques used in reconnaissance like gathering various types of information and searching various sources all of which are outside of the scope of enterprise defenses and controls. Hence, there is no available detection control for them. However, if an organization wants to set a detection around techniques in reconnaissance tactic they can integrate external cybersecurity platforms like ZeroFox [34] to their Microsoft Sentinel workspace which can alert when their organization's details are mentioned on any public forum. While this will not eliminate the use of techniques in reconnaissance but will provide more control over what information is publicly available about their organization which can be used by an attacker. Before the comparison between heatmaps let's go over which techniques are covered and not covered in each heatmap.

The first heatmap generated was for the open-source KQL rules available in the official GitHub repository of Microsoft Sentinel. The heatmap was generated by mapping 969 KQL rules to the MITRE ATT&CK framework. The techniques identified as **not** monitored were the following:

- **Resource Development**: All except Compromise Accounts and Infrastructure.

Algorithm 6.1 Generating Heatmap for Monitored Detection Controls

```
Input: Path to the directory containing JSON files
Output: A heatmap visualization of techniques and statistics
Initialize empty list techniques
for each file in specified directory do
    if file ends with '.json' then
        Read JSON data from file
        if relevantTechniques key is present in data then
            for each technique_id in relevantTechniques do
                Append first 5 characters of technique_id to techniques list
            end for
        end if
    end if
end for
Compute a frequency dictionary count_of_tech of techniques
for each technique_id in techniques do
    technique_name, tactic_name ← map_attack_technique(technique_id)
    if technique_name is not None then
        Append technique_name to technique_names list
        heatmap(technique_id, tactic_name, count_of_tech[technique_id])
    else
        Output warning: "No technique found for technique_id."
    end if
end for
Compute and output a frequency dictionary of technique_names
Output the number of detected techniques
```

Algorithm 6.2 Function: map_attack_technique

```
Input: technique_id
Output: technique_name, tactic_name
Read JSON data from MITRE file
for each technique in data["technique"] do
    if technique["id"] equals technique_id then
        return technique["name"], technique["tactic_name"]
    end if
end for
return None, None
```

Algorithm 6.3 Function: heatmap

```
Input: new_technique, tac_name, count
Read JSON data from 'heatmap.json' file
Modify *tac_name* replacing spaces with hyphens and converting to lowercase
Select a color based on the value of count
Create a JSON object with *new_technique*, *tac_name*, and selected color
Append the created object to data["techniques"]
Write the updated JSON data back to 'heatmap.json' file
```

- **Initial Access**: Hardware Additions and Replication Through Removable Media.
- **Execution**: Cloud Administration Command, Container Administration Command, Deploy Container, Serverless Execution, and Shared Modules.
- **Persistence**: BITS Jobs, Boot or Logon Initialization Scripts, Browser Extensions, and Implant Internal Image.
- **Privilege Escalation**: Boot or Logon Initialization and Escape to Host.
- **Defense Evasion**: BITS Jobs, Build Image on Host, Debugger Evasion, Deploy Container, Direct Volume Access, Execution Guardrails, File and Directory Permissions Modification, Indirect Command Execution, Modify Registry, Modify System Image, Network Boundary Bridging, Plist File Modification, Pre-OS Boot, Reflective Code Loading, Rogue Domain Controller, Rootkit, Subvert Trust Controls, System Script Proxy Execution, Trusted Developer Utilities, Unused/Unsupported Cloud Regions, Sandbox Evasion, and XSL Script Processing.
- **Credential Access**: Forced Authentication, Forge Web Credentials, Multi-Factor Authentication Interception, Multi-Factor Authentication Request Generation, Steal or Forge Authentication Certificates, and Steal Web Session Cookie.
- **Discovery**: Application Window Discovery, Container and Resource Discovery, Cloud Infrastructure Discovery, Debugger Evasion, Device Driver Discovery, System Location Discovery, Peripheral Device Discovery, System Network Connections Discovery, System Owner/User Discovery, System Service Discovery, System Time Discovery, and Virtualization/Sandbox Evasion
- **Lateral Movement**: Internal Spearphishing, Remote Service Session Hijacking, Replication through Removable Media, Software Deployment Tools, and Taint Shared Content.

- **Collection**: Archive Collected Data, Audio Capture, Browser Session Hijacking, Clipboard Data, Data from Configuration Repository, Data from Network Shared Drive, Data from Removable Drive, Screen Capture, and Video Capture.

- **Command and Control**: Communication through Removable Media, Data Encoding, Encrypted Channel, Multi-Stage Channels, Protocol Tunnelling, Proxy, and Remote Access Software.

- **Exfiltration**: None

- **Impact**: Data Manipulation, Defacement, Firmware Corruption, and Network Denial of Service.

It can be said that the severity of tactics increases as one goes from left to right as this represents that the attacker has successfully exploited the previous tactic and is now on the next step of attack (e.g. once the attacker gets initial access through phishing, they'll move on to execution which gets more critical). This behavior can explain why the last seven tactics have more detection rules than the initial seven. Particularly, in the case of the Exfiltration tactic, a 100% coverage can be observed. This can be due to the fact that ex-filtrating data from an environment is the ultimate goal of an adversary. As with the ex-filtrated data the adversary can demand ransom from the organization in return for not publicly disclosing the sensitive internal data. Next, a comparison between the open-source and the company-generated heatmaps can be observed specifically mentioning the gaps in the four companies' detection controls.

The second heatmap generated was for the Company1 KQL rules extracted from the dedicated Microsoft Sentinel environment. The heatmap was generated by mapping 238 KQL rules to the MITRE ATT&CK framework. The techniques identified as **not** monitored but have open-source detection controls were the following:

- **Resource Development**: Compromise Accounts

- **Initial Access**: None

- **Execution**: Inter-Process Communication, Native API, Software Deployment Tools, and Windows Management Instrumentation.

- **Persistence**: Boot or Logon Autostart Execution, Pre-OS Boot, and Traffic Signaling.

- **Privilege Escalation**: Access Token Manipulation, Boot or Logon Autostart Execution, Domain Policy Modification, and Process Injection.

- **Defense Evasion**: Access Token Manipulation, Deobfuscate/Decode Files or Information, Domain Policy Modification, Exploitation for Defense Evasion, Hide Artifacts, Indicator Removal, Masquerading, Process Injection, System Binary Proxy Execution, Template Injection, Traffic Signaling and Weaken Encryption.
- **Credential Access**: Adversary-in-the-Middle, Credentials from Password Stores, Input Capture, Network Sniffing, Steal or Forge Kerberos Tickets, and Unsecured Credentials.
- **Discovery**: Browser Information Discovery, Cloud Service Dashboard, Cloud Service Discovery, File and Directory Discovery, Group Policy Discovery, Network Share Discovery, Network Sniffing, Password Policy Discovery, Process Discovery, Query Registry, Software Discovery, and System Information Discovery.
- **Lateral Movement**: Exploitation of Remote Services, and Remote Services.
- **Collection**: Adversary-in-the-Middle, Automated Collection, Data Staged and Input Capture.
- **Command and Control**: Data obfuscation, Dynamic Resolution, Fallback Channels, Non-Application Layer Protocol, and Traffic Signaling.
- **Exfiltration**: Exfiltration Over Other Network Medium, Scheduled Transfer, and Transfer Data to Cloud Account.
- **Impact**: Inhibit System Recovery, Resource Hijacking, Service Stop, and System Shutdown/Reboot.

The third heatmap generated was for the Company2 KQL rules extracted from the dedicated Microsoft Sentinel environment. The heatmap was generated by mapping 384 KQL rules to the MITRE ATT&CK framework. The techniques identified as **not** monitored but have open-source detection controls were the following:

- **Resource Development**: Compromise Accounts
- **Initial Access**: None
- **Execution**: None
- **Persistence**: Pre-OS Boot
- **Privilege Escalation**: None
- **Defense Evasion**: System Binary Proxy Execution, Template Injection, and Weaken Encryption.

- **Credential Access**: None
- **Discovery**: Password Policy Discovery, however, the company also covers System Network Connections Discovery that is not present in open-source controls.
- **Lateral Movement**: None, However, custom KQL(s) for Software Deployment Tools technique are present that are not available in open-source controls.
- **Collection**: Automated Collection. However, the company has extra coverage than open-source by covering Archive Collected Data, Clipboard Data, Screen Capture, and Video Capture.
- **Command and Control**: None, The company has more coverage than open-source controls due to the presence of custom KQLs.
- **Exfiltration**: None
- **Impact**: Inhibit System Recovery and System Shutdown/Reboot.

The fourth heatmap generated was for the Company3 KQL rules extracted from the dedicated Microsoft Sentinel environment. The heatmap was generated by mapping 265 KQL rules to the MITRE ATT&CK framework. The techniques identified as **not** monitored but have open-source detection controls were the following:

- **Resource Development**: Compromise Accounts
- **Initial Access**: None
- **Execution**: Native API and Windows Management Instrumentation.
- **Persistence**: Boot or Logon Autostart Execution, and Pre-OS Boot.
- **Privilege Escalation**: Access Token Manipulation, Boot or Logon Autostart Execution, and Process Injection.
- **Defense Evasion**: Access Token Manipulation, Deobfuscate/Decode Files or Information, Exploitation for Defense Evasion, Indicator Removal, Masquerading, Process Injection, System Binary Proxy Execution, Template Injection, and Weaken Encryption.
- **Credential Access**: Adversary-in-the-Middle, Credentials from Password Stores, Input Capture, Network Sniffing, and Steal or Forge Kerberos Tickets.
- **Discovery**: Browser Information Discovery, File and Directory Discovery, Group Policy Discovery, Network Sniffing, Password Policy Discovery, Process Discovery, Query Registry, and System Information Discovery.

- **Lateral Movement**: Exploitation of Remote Services, and Remote Services. However, custom KQL(s) for Software Deployment Tools technique are present that are not available in open-source controls.
- **Collection**: Adversary-in-the-Middle, Automated Collection, and Input Capture.
- **Command and Control**: None, in fact Company has more coverage than open-source controls due to the presence of custom KQLs.
- **Exfiltration**: None
- **Impact**: Inhibit System Recovery and System Shutdown/Reboot.

The fifth heatmap generated was for the Company4 KQL rules extracted from the dedicated Microsoft Sentinel environment. The heatmap was generated by mapping 241 KQL rules to the MITRE ATT&CK framework. The techniques identified as not monitored but have open-source detection controls were the following:

- **Resource Development**: Compromise Accounts
- **Initial Access**: None
- **Execution**: Inter-Process Communication, Native API, Software Deployment Tools, and Windows Management Instrumentation.
- **Persistence**: Boot or Logon Autostart Execution, and Pre-OS Boot.
- **Privilege Escalation**: Access Token Manipulation, Boot or Logon Autostart Execution, Domain Policy Modification, and Process Injection.
- **Defense Evasion**: Access Token Manipulation, Deobfuscate/Decode Files or Information, Domain Policy Modification, Exploitation for Defense Evasion, Hide Artifacts, Indicator Removal, Masquerading, Process Injection, System Binary Proxy Execution, Template Injection, and Weaken Encryption.
- **Credential Access**: Adversary-in-the-Middle, Credentials from Password Stores, Input Capture, Network Sniffing, Steal or Forge Kerberos Tickets, and Unsecured Credentials.
- **Discovery**: Browser Information Discovery, Cloud Service Dashboard, Cloud Service Discovery, File and Directory Discovery, Group Policy Discovery, Network Sniffing, Password Policy Discovery, Process Discovery, Query Registry, Software Discovery, and System Information Discovery.
- **Lateral Movement**: Exploitation of Remote Services, and Remote Services.

- **Collection**: Adversary-in-the-Middle, Automated Collection, and Input Capture.

- **Command and Control**: None, in fact Company has more coverage than open-source controls due to the presence of custom KQLs.

- **Exfiltration**: None

- **Impact**: Inhibit System Recovery and System Shutdown/Reboot.

As it can be observed except for Company1 all the other companies are monitoring the exfiltration tactic 100% due to it being the last line of defense. This gap was disclosed and bridged after the findings. For the Command and Control tactic from the table 6.3, it was observed that Company2, Company3, and Company4 have more detection than available open-source detection controls. The reason here is custom detection rules and rules from other sources (e.g. some companies have their own security team whereas some companies used to have different monitoring providers that provided custom rules). Among all companies in this study, if there were a rank of which company has the most coverage and which has the least coverage. Company2 which belongs to the Software development sector, was found to be most covered in terms of MITRE ATT&CK coverage. Then, Company3 which belongs to the IT Consulting sector followed by Company4 (Law sector), and with the least coverage was Company1 which belongs to Management Consulting. This trend indicates that the companies with more technology focus tend to be more cautious to have fewer gaps in their cyber-security infrastructure whereas other sectors whose main product is not technology-focused tend to have more security gaps. While there are more in-detail comparisons possible with the heatmaps. Due to time limitations, this thesis mentions a few of them.

The goal of identifying techniques not monitored by the organizations but having available open-source detection can be observed by generated heatmaps in appendix A and also discussed in the section 6.3. These are termed as security gaps. It is required to know that these gaps from open-source controls are from the set of KQLs that were downloaded from the official GitHub repository of Microsoft Sentinel while performing this thesis. In the future, there may be more KQLs added to the repository which will be able to identify more security gaps. The next section will discuss how organizations can bridge the identified gaps.

6.4 Increasing Detection Coverage with Open-Source Controls

In this section, the method for enhancing the detection coverage of the four organizations is discussed. The MITRE ATT&CK heatmap, generated by each company and depicted in Appendix A, showcases techniques that have at least one detection control (KQL Rule) via assigned colors. Table 6.3 outlines the number of monitored techniques by tactics for each organization along with open-source data. This data is in the form of the number of techniques monitored in that tactic/total number of techniques in that tactic. From this data, it is evident that open-source detection controls provide superior coverage of techniques, especially in the tactics of defense evasion and discovery, compared to the currently monitored detection controls within each organization. This insight implies the availability of open-source detection controls to bridge security gaps in organizations.

To identify the available open-source detection controls for these security gaps, a Python script named `Technique_Finder.py` (available on GitHub [22]) was developed. This script discovers the KQL rules based on the input of an unmonitored technique ID within a folder comprising all open-source KQL rules. Algorithm 6.4 outlines the procedure of this code: initially, a technique ID (e.g., T1078) is inputted. Following this, the script traverses through each file within the folder to identify files with the same technique ID. The resultant output comprises filenames containing KQL rules corresponding to the inputted technique ID. Organizations can subsequently implement these KQL rules in Microsoft Sentinel to broaden their detection coverage.

6.5 Conclusion

The goal of this chapter was to answer RQ4, which was done by identifying security gaps in organizations by utilizing MITRE ATT&CK heatmaps. Utilizing the method described in Section 6.2, heatmaps were constructed using open-source KQL rules obtained from Microsoft Sentinel's official GitHub repository [23], as well as from four pseudonymized organizations: Company1, Company2, Company3, and Company4. These heatmaps can be found in Appendix A.

The created heatmaps display the unmonitored techniques in the MITRE ATT&CK framework, pointing towards possible vulnerabilities that might be exploited by attackers to evade detection. An in-depth discussion on the security gaps of each organization is presented, emphasizing the unmonitored techniques. A comparative analysis is also conducted among the heatmaps of the four organizations, uncovering unmonitored techniques for which open-source detection controls exist.

Data Source	Reconnaissance	Resource Development	Initial Access
Company1	0/10	1/8	7/9
Company2	0/10	1/8	7/9
Company3	0/10	1/8	7/9
Company4	0/10	1/8	7/9
Open-Source	0/10	2/8	7/9

Data Source	Execution	Persistance	Privilege Escalation	Defense Evasion
Company1	5/14	12/19	7/13	8/42
Company2	9/14	14/19	11/13	18/42
Company3	7/14	13/19	8/13	11/42
Company4	5/14	13/19	7/13	9/42
Open-Source	9/14	15/19	11/13	20/42

Data Source	Credential Access	Discovery	Lateral Movement
Company1	5/17	6/31	2/9
Company2	11/17	18/31	5/9
Company3	6/17	10/31	3/9
Company4	5/17	7/31	2/9
Open-Source	11/17	18/31	4/9

Data Source	Collection	Command and Control	Exfiltration	Impact
Company1	4/17	4/16	6/9	5/13
Company2	11/17	14/16	9/9	9/13
Company3	5/17	14/16	9/9	9/13
Company4	5/17	14/16	9/9	9/13
Open-Source	8/17	9/16	9/9	9/13

Table 6.3: Monitored techniques from each data source by tactics

Algorithm 6.4 Technique Finder in Open-Source data

```
Input: Directory path containing JSON files, directory_path
Output: File names containing the specified technique ID
tid ← UserInput(Enter the technique ID which you want to find: )
for each filename in ListFiles(directory_path) do
    if FileExtension(filename) == .json then
        file_path ← JoinPath(directory_path, filename)
        data ← ReadJson(file_path)
        if 'relevantTechniques' in data and data['relevantTechniques'] then
            for each technique_id in data['relevantTechniques'] do
                if technique_id == tid then
                    Print(The KQL use case for technique is: )
                    Print(filename)
                end if
            end for
        end if
    end if
end for
```

Additionally, a method is introduced to identify these detection controls using the technique ID coupled with a Python script, named `Technique_Finder.py`. This allows for the extraction of relevant KQL rules from a collection of 969 open-source KQL rules, enabling organizations to incorporate them into their Microsoft Sentinel environments. As a result, organizations can enhance their coverage of the MITRE ATT&CK framework, improving their security postures.

Next chapter talks about the conclusion and the future work related to the work in this thesis.

Chapter 7

Conclusion and Future Work

The goal of this chapter is to conclude this thesis, the discussion begins with the conclusion of all previous chapters. It then provides information about the limitations encountered during the research conducted for this thesis, as well as outlines future work within this area, offering strategies to address the limitations faced during the current research endeavour.

7.1 Conclusion

In this thesis, the rise of cyber attacks is explored, highlighting that over half of these attacks evade detection [1], thereby indicating a security gap within organizational detection capabilities. To address this gap, this thesis proposes four research questions. The search begins with RQ1 which finds a framework that could aid organizations in identifying the techniques employed by cyber attackers. Upon evaluating several frameworks including NIST[7], ISO/IEC 27001[8], and MITRE ATT&CK[6], it was concluded that the MITRE ATT&CK framework encapsulates the techniques leveraged in real-world attacks most comprehensively, showcasing a more technical focus. This framework can serve as a reference for implementing and measuring the effectiveness of organizational detection controls.

Subsequent to this, the research shifted towards RQ2 pinpointing a detection tool resonating with the majority of techniques mentioned in the MITRE ATT&CK framework. Post a comparative analysis of various detection tool types based on the coverage of ATT&CK data sources, it emerged that the SIEM tool had the most coverage of MITRE ATT&CK, outperforming other detection tools discussed in this thesis. A further examination of multiple SIEM providers led to the selection of Microsoft Sentinel as the preferred SIEM for this research, credited to its superior rating by Gartner[15] in comparison to other SIEM solutions.

With the SIEM determined, the focus transitioned to selecting KQL rules as the detection controls owing to their applicability in detecting incidents on Microsoft Sentinel. Then, to answer RQ3 the search began for open-source detection controls (KQL rules) for Sentinel, which resulted in the discovery of an official GitHub repository[23] of Microsoft Sentinel encompassing a total of 969 KQL rules. Analysis revealed that 616 KQL rules lacked mapping to MITRE ATT&CK, posing a challenge for mapping. An experiment with Bing AI [30] yielded a mapping accuracy of 40%, which which not acceptable as that would indicate more than half of KQL analytic rules are wrongly mapped. This prompted a shift towards manual mapping of KQL rules to MITRE ATT&CK. The manual mapping involved first understanding what each KQL rule monitors, then looking for matching techniques on the official MITRE ATT&CK website.

Post mapping, a Python script was devised to generate a heatmap in JSON format, which could be uploaded to the MITRE ATT&CK Navigator[33] for a visual representation of monitored techniques. This script was employed to answer RQ4 by producing heatmaps for open-source data alongside data from four distinct organizational sectors. Following this, a comparison of these heatmaps revealed certain techniques lacking detection controls within the organizations, yet having detection controls in the open-source data. This benefits an organization by identifying which security gaps they can fill. A Python script 6.4 was created to locate the KQL analytic rules for the not-monitored techniques that have an open-source detection available, based on the technique ID. These identified KQL analytic rules can be implemented within the Microsoft Sentinel workspace of these organizations, thereby helping improve their ability to detect threats by fixing the identified security gaps.

The next section discusses future work in the area based on the limitations faced in this thesis.

7.2 Future Work

In this section, the discussion extends to potential future work aimed at enhancing the framework outlined in this thesis. The starting point for identifying improvements lies in examining the limitations encountered throughout this research. A significant hurdle was the mapping of KQL rules to their corresponding MITRE ATT&CK techniques. Several approaches can be adopted to overcome this challenge in the future.

Firstly, the development of an AI model capable of mapping a KQL rule to its relevant MITRE ATT&CK technique with greater accuracy than what was achieved in the Bing AI test could prove beneficial. The data generated in this thesis can contribute to training such an AI model by providing a larger set of pre-mapped KQL rules to MITRE ATT&CK techniques.

Secondly, this thesis relied on a single individual (the author) to map the dataset of unmapped KQL rules. Future explorations in this area could employ a diverse group of individuals to map KQL rules to MITRE ATT&CK techniques. By comparing the outcomes, this collective approach may mitigate, or at least diminish, the possibility of human error.

Another area for improvement in future work includes the incorporation of a broader range of open-source data to ascertain whether a more extensive set of techniques can be covered solely through the use of open-source KQL rules. Some additional repositories that could be utilized are referenced in this thesis at table 5.2.

References

[1] Mandiant. Security effectiveness 2020, 2020. URL https://www.mandiant.com/resources/security-effectiveness-2020-deep-dive-into-cyber-security-reality.

[2] Jarkko Kinnunen. What is iso 27001? a quick and easy explanation, 2023. URL https://www.theseus.fi/handle/10024/745250.

[3] Anna Georgiadou, Spiros Mouzakitis, and Dimitris Askounis. Assessing mitre att&amp;ck risk using a cyber-security culture framework. *Sensors*, 21(9), 2021. ISSN 1424-8220. doi: 10.3390/s21093267. URL https://www.mdpi.com/1424-8220/21/9/3267.

[4] Rabobank. Dettect v1.9, 2023. URL https://github.com/rabobank-cdc/DeTTECT.

[5] Wenjun Xiong, Emeline Legrand, Oscar Åberg, and Robert Lagerström. Cyber security threat modelling based on the mitre enterprise att&ck matrix. *Software and Systems Modeling*, 2022. URL https://doi.org/10.1007/s10270-021-00898-7.

[6] MITRE. Mitre att&ck framework, 2023. URL https://attack.mitre.org/.

[7] NIST. Cybersecurity framework, 2023. URL https://www.nist.gov/cyberframework.

[8] ISO. Iso/iec 27001, 2023. URL https://www.iso.org/standard/27001.

[9] Dejan Kosutic. Soc detection capabilities testing environment, 2021. URL https://advisera.com/27001academy/what-is-iso-27001/.

[10] George Mutune. 27 top cybersecurity tools for 2023, 2023. URL https://cyberexperts.com/cybersecurity-tools/.

[11] Vikki Davies. Top 10 threat detection tools for cybersecurity, 2023. URL https://cybermagazine.com/articles/top-10-threat-detection-tools.

[12] Microsoft. What is siem?, 2023. URL https://www.microsoft.com/en-us/security/business/security-101/what-is-siem.

[13] Snow Flake. Threat detection methods and best practices, 2023. URL https://www.snowflake.com/guides/threat-detection-methods.

[14] MITRE. Data sources, 2023. URL https://attack.mitre.org/datasources.

[15] Gartner. Siem magic quadrant, 2022. URL https://blog.a3sec.com/en/siem-2022-gartner-magic-quadrant.

[16] Magic Quadrant Research Methodology | Gartner, 2023. URL https://www.gartner.com/en/research/methodologies/magic-quadrants-research. [Online; accessed 22. Oct. 2023].

[17] yelevin. What is Microsoft Sentinel?, 2023. URL https://learn.microsoft.com/en-us/azure/sentinel/overview. [Online; accessed 22. Oct. 2023].

[18] IBM. Ibm app exchange, 2023. URL https://exchange.xforce.ibmcloud.com/hub?ippr=All&br=QRadar&con=C029.

[19] What is Splunk SIEM used for?, 2023. URL https://www.comodo.com/is-splunk-a-siem.php. [Online; accessed 22. Oct. 2023].

[20] Exabeam. Exabeam log management, 2023. URL https://www.exabeam.com/product/security-log-management/.

[21] Securonix: Security Analytics at Cloud Scale., 2023. URL https://www.securonix.com. [Online; accessed 22. Oct. 2023].

[22] Aditya Sharma. Sentinel_gap_analysis, 2023. URL https://github.com/Wr417h/Sentinel_Gap_Analysis.

[23] Azure. Azure sentinel analytics usecases, 2021. URL https://github.com/Azure/Azure-Sentinel/blob/master/Tools/RuleMigration/AnalyticsUseCases.md.

[24] Matt Zorich. Collection of kql queries, 2022. URL https://github.com/reprise99/Sentinel-Queries.

[25] Rod Trent. Azure sentinel kql, 2020. URL https://github.com/rod-trent/SentinelKQL.

[26] Jose Sebastian Canos. In this repository you may find kql (kusto query language) queries and watchlist schemes for data sources related to microsoft sentinel (a siem tool)., 2022. URL https://github.com/ep3p/Sentinel_KQL.

[27] Bert-Jan Pals. Kql queries. defender for endpoint and azure sentinel hunting and detection queries in kql. out of the box kql queries for: Advanced hunting, custom detection, analytics rules & hunting rules., 2022. URL https://github.com/Bert-JanP/Hunting-Queries-Detection-Rules.

[28] Ugur Koc. Kql_intune, 2022. URL https://github.com/ugurkocde/KQL_Intune.

[29] mitre attack. tram, 2023. URL https://github.com/mitre-attack/tram. [Online; accessed 23. Oct. 2023].

[30] Microsoft. Bing ai, 2023. URL https://www.microsoft.com/en-us/edge/features/bing-chat.

[31] Computest. Computest consultancy, 2023. URL https://www.computest.nl/en/.

[32] Microsoft. Export and import analytics rules to and from arm templates, 2023. URL https://learn.microsoft.com/en-us/azure/sentinel/import-export-analytics-rules.

[33] MITRE. Mitre att&ck navigator, 2023. URL https://mitre-attack.github.io/attack-navigator/.

[34] Partners & Integrations, November 2023. URL https://www.zerofox.com/partners-and-integrations/?partner=integration&type=siem#. [Online; accessed 16. Nov. 2023].

Appendix A

Generated Heatmaps

This appendix complements the thesis document by including MITRE ATT&CK heatmaps generated by four distinct organizations, alongside the open-source data retrieved from the official Microsoft Sentinel GitHub repository.

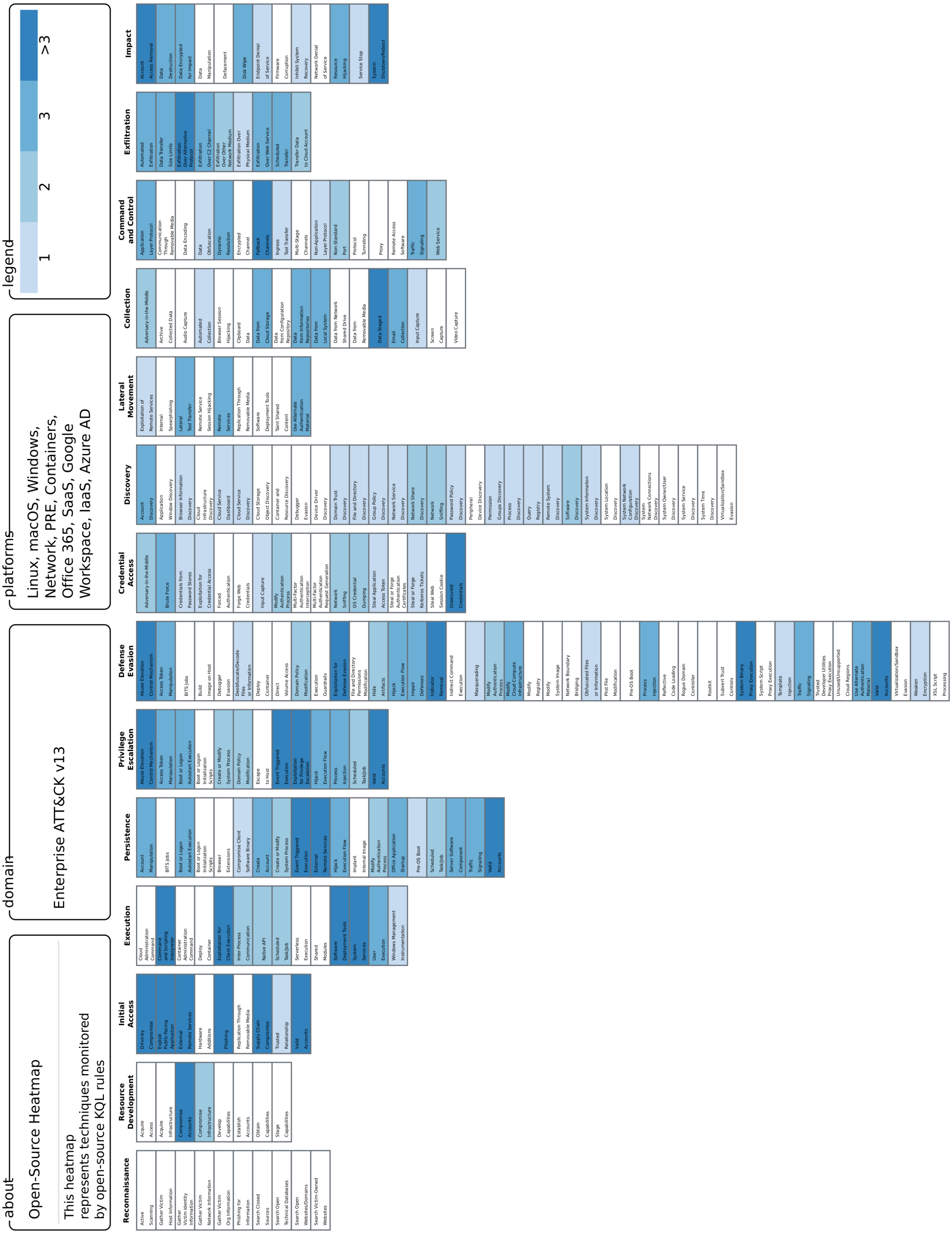

Figure A.1: Heatmap Generated by Open-Source detection controls

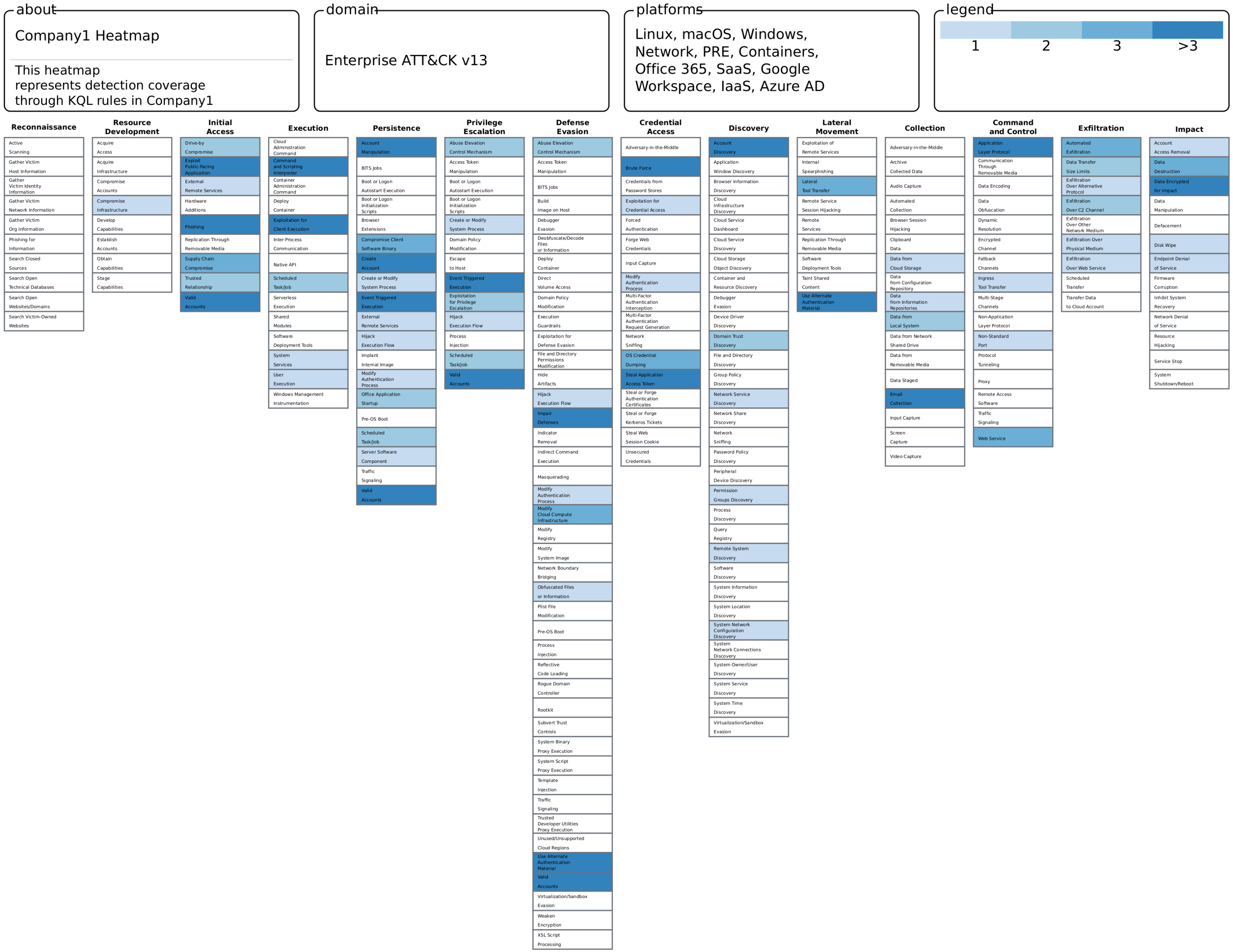

Figure A.2: Heatmap Generated by Company1 detection controls

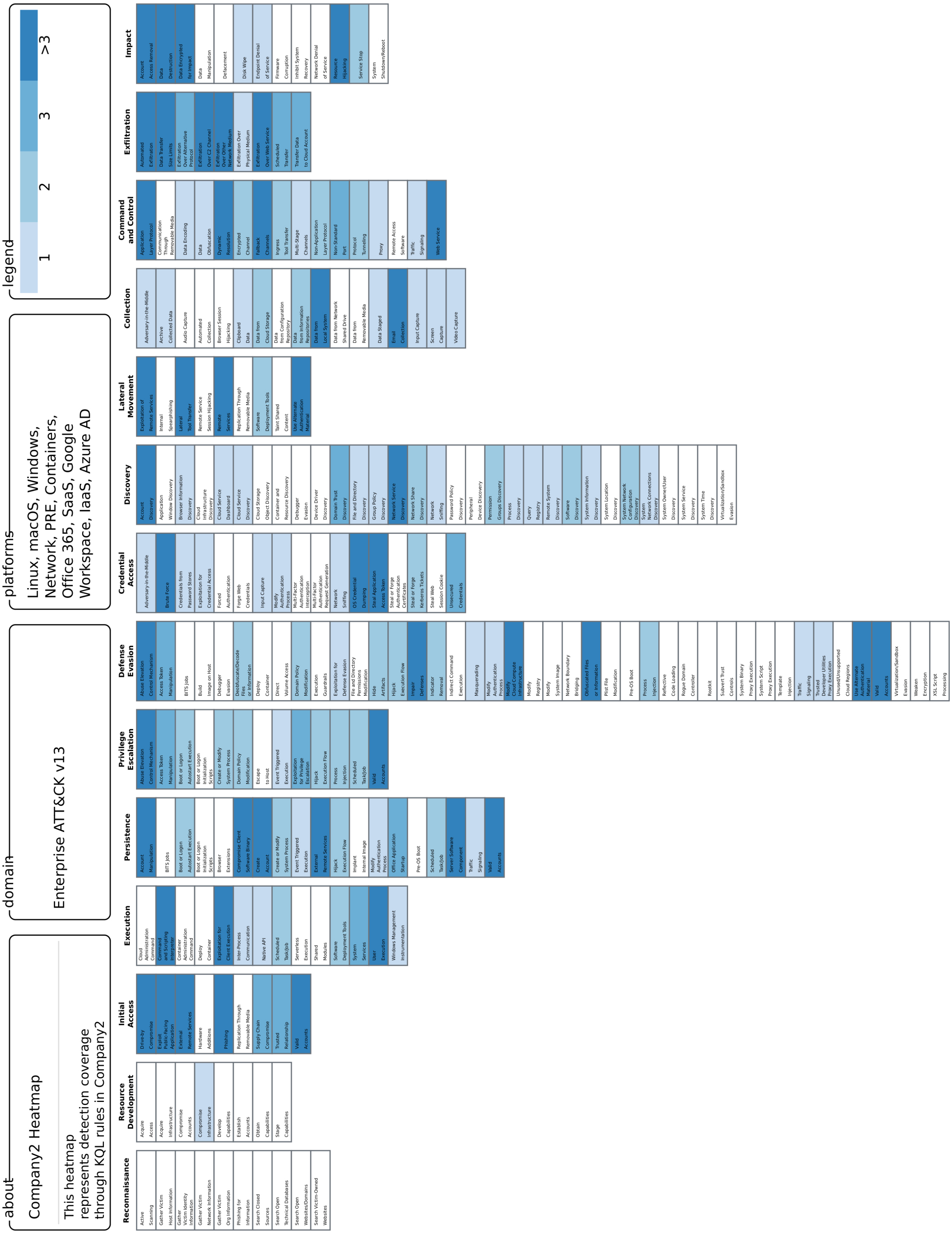

Figure A.3: Heatmap Generated by Company2 detection controls

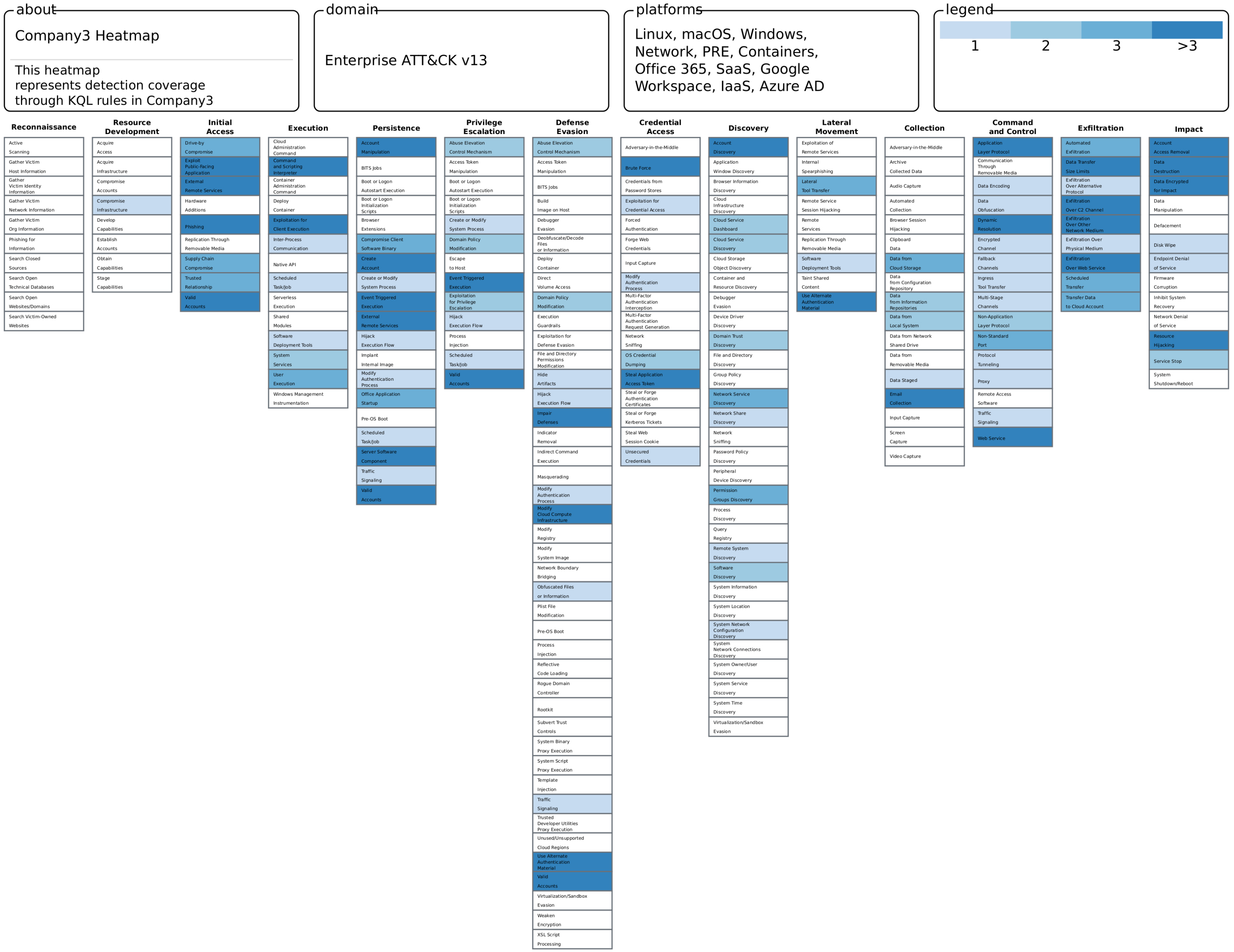

Figure A.4: Heatmap Generated by Company3 detection controls

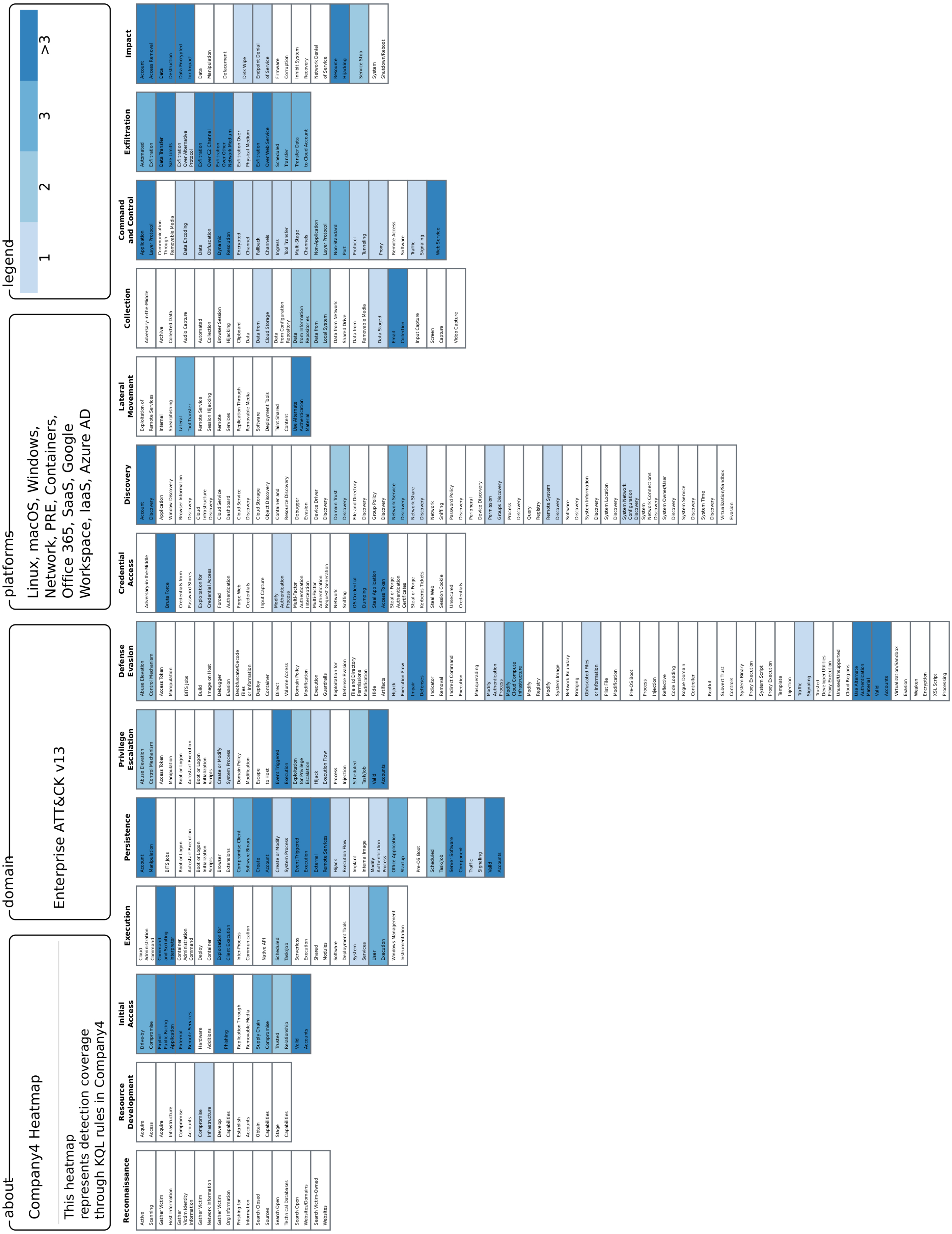

Figure A.5: Heatmap Generated by Company4 detection controls