University of Gloucestershire
School of Computing and Engineering
BSc in Computer and Cyber Security

# How can the implementation of SOCaaS reduce the risk of successful zero-day attacks against an organisations system?

Dan Mounter

May 25, 2021

## Abstract

This dissertation explores the idea of embracing a new trend in technology with existing mitigation techniques to help combat against one of the most critical cyber-attacks. Zero-day attacks are rare; yet their nature makes them considerably difficult to mitigate. Malicious threat actors exploit the absence of vulnerability existence to achieve their objectives. Zero-day defence techniques exist and often have a high degree of effectiveness, many involve machine-learning and exercise signature, anomaly, or behaviour-based algorithms to reduce risk. Despite this, technical frameworks are often restricted to a particular scope; for example, advanced port-monitoring is only effective against network related zero-day attacks. The dissertation aims to encourage Security Operations Centre-as-a-Service (SOCaaS) as a framework which can address the wider risk landscape of zero-day attacks. It is understood that the benefits of SOCaaS relate to some aspect of zero-day mitigation, including support for continuous monitoring, more efficient patch rollouts, and ability to adapt mitigation strategies. To further understand this concept, predictive analysis will be performed on existing zero-day vulnerabilities; to help justify the need for protective defence solutions. Results of this analysis will provide explanation into discovering the causal-relationship between SOCaaS implementation and the quality of zero-day attack mitigation.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Overview

Zero-day vulnerabilities are often considered the most serious security risk, absence of existence makes prevention considerably challenging. Risks associated with zero-day attacks are greater when organisations are targeted; depending on the system affected, initial exploitation often accelerates further attacks (Joshi, Singh, and Kanellopoulos, 2018). From a business-view the risk of zero-day attack is usually overlooked when implementing security models, Woody (2013) describes how existing security models are unsuitable for advanced attack vector mitigation. The advancement in obfuscation techniques and frequent lack of ability to adapt to technological evolution are just a few reasons behind the exponential rise in zero-day related breaches.

Various mitigation techniques exist; however, many are not yet implemented in business environments. Bedell and Bouchard (2018) understands that cloud-based technology supports many key features which can reduce the risk of zero-day attack if integrated with existing mitigation techniques. In recent years, the wide-spread introduction of cloud technology led to conceptual advancements in software-as-a-service, significant features include improved accessibility, scalability, affordability, and compatibility. The further development and understanding of Software as a Service (SaaS) led to the introduction of Security Operations Centre as a Service (SOCaaS); a concept which collates the benefits of SaaS in the form of an operational security model. Figure 1.1 shows a Google trend search visualizing that the topic of SOCaaS has increased in popularity over the last 5 years. This relatively new approach possesses advanced improvements from both security and enterprise perspectives, however current research suggests this is yet to be influential for businesses.


Figure 1.1: Google trend search from 2004-2020 for SOC-as-a-Service, from (Google, 2020)

## 1.2 Problem Statement

How can the implementation of SOCaaS reduce the risk of successful zero-day attacks against an organisations system?

## 1.3 Research Questions

This dissertation aims to answer the following questions:

1. What are the risks associated with zero-days vulnerabilities and what are current mitigation techniques?
2. How to understand why a proactive mitigation framework will improve mitigation quality?
3. How can the implementation of SOCaaS mitigate the risk of successful zero-day attacks against an organisations system?

## 1.4 Research Objectives

1. To investigate zero-day attack detection and how mitigation quality is measured in SOC environments.
2. To perform predictive data analysis on zero-day vulnerabilities, to identify future trends and solutions, with the aim of emphasising SOCaaS importance.
3. To evaluate the causal-relationship between SOCaaS implementation and the quality of zero-day attack mitigation.

## 1.5 Scope

The research aims to discover a relationship between SOCaaS implementation and the quality of zero-day attack mitigation. Risk management involving zero-day attacks is a diverse subject, there are many factors which can affect overall mitigation quality. For the purpose of evaluating this relationship, the focus will consist of measuring risk of zero-day vulnerabilities. The research will only explore methods to improve risk mitigation, and not complete prevention. This is due to the unavoidable nature of exploitable critical system vulnerabilities. This research will explore the properties and characteristics of recorded zero-day vulnerabilities between January 2020 and April 2021. The lack of openness from victims surrounding these types of attacks limits the accuracy of data analysis, thus only published zero-day CVEs will be measured. The chosen time range aims to highlight current and future trends in the zero-day landscape.

## 1.6 Declarations

I declare that this dissertation has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree or anywhere else. Except where states otherwise by citation and reference or acknowledgement, the work presented is entirely my own.

I confirm that all tables and figures in this dissertation/proposal are my works or a regeneration from other people's work with citation.

I confirm that all citations in the dissertation/proposal have been provided in the bibliography and they are all accessible. In case that university wants to cross examine the citation, I can provide the consent for the references which are not accessible.

I confirm that all the tools, software and datasets have been used in this dissertation/proposal followed the terms and conditions in their license agreement and university REC code of conduct.

## 1.7 Conclusion

After researching existing literature in the field of risk management, a stronger focus has been made on improving security risk models in recent years. Despite this, the increasing frequency of zero-day related breaches is evident, emphasising the need for more proactive approaches to cyber-security. Successful implementation of SOCaaS directly tackles the problem of sophisticated threats, research implies that benefits of SOCaaS relate to some aspect of zero-day mitigation.

# Chapter 2

# Literature Review

## 2.1 Introduction

This chapter explores existing literature to develop a deeper understanding of current perceptions, concepts and techniques.

## 2.2 Defining Advanced Persistent Threats

Specific attacks which are elaborate in nature became recognised as an Advanced Persistent Threat (APT). The National Institute of Standards and Technology (NIST) describes APTs as:

> *"an adversary that possesses sophisticated levels of expertise and significant resources which allow it to create opportunities to achieve its objectives by using multiple attack vectors..."* Source: (Standards and Technology, 2012)

Research into APTs is well-established, Wrightson (2015) details the attributes and delivery vectors associated with these attacks. Further to the levels of complication seen in the techniques and tools used to attack, the author describes APTs as long-term, meaning a high-level of planning is required before the aim is achieved. A publication by Tecuci, Marcu, Meckl, and Boicu (2018) explores the idea of integrating machine-learning with manual SOC techniques to provide automated APT detection. The research also supports Brewer (2014)'s interpretation of APT features and qualities, the researcher claims that threat actors use sophisticated attack techniques because they are more adaptable to defensive efforts.

The author goes further to agree that zero-day vulnerabilities are often the exploit of choice when targets implement particularly effective defence systems. Evidence of this can be explained through the published data of past known high-profile security breaches; an annual APT trend review by Emm (2019) shows that many attacks involved zero-day vulnerability exploitation.

## 2.3 Zero-Day Attack Characteristics

By definition, a zero-day attack is a highly sophisticated threat which involves custom exploits targeting undisclosed vulnerabilities. The term itself is a reference to the number of days developers and vendors have to address and patch the vulnerability.

As described by Aleroud and Karabatis (2013), this exploitation technique embraces the absence of vulnerability awareness to bypass traditional Intrusion Detection System (IDS), making this an incredibly effective method of obtaining unauthorised access. Known associated risks include

privilege escalation, business damage, and as evident from the infamous Stuxnet virus, physical damage. Stuxnet was an APT which exploited four Windows zero-day vulnerabilities in a targeted attack on Iran's uranium enrichment facilities. Novel analysis by Al-Rabiaah (2018) centres around the attacks sophistication by describing the exploits' ability to damage physical infrastructure by overloading Programmable Logic Controller (PLC). Research by Brewer (2014) reminds us that zero-day vulnerabilities are notorious for lying dormant, a key characteristic which Stuxnet relied on to hide activity.

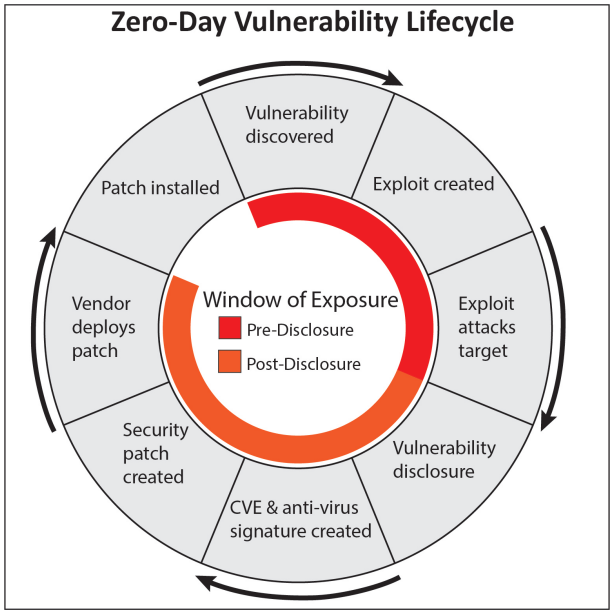



Figure 2.1: Zero-Day Vulnerability Life Cycle, adopted from Vaisla and Saini (2014)

Figure 2.1 illustrates the lifecycle of zero-day vulnerabilities and emphasises the hostile and unpredictable nature of this attack by visualising the risk of continuous exposure. Zero-day attacks are often the vector of choice due to the high probability of success, with Brewer (2014) further highlighting the effectiveness of combining spear-phishing with zero-day vulnerabilities. Vaisla and Saini (2014) acknowledges this concept and explains that zero-day attacks should be considered inevitable in the modern age of information security.

The strength of this claim is supported by Singh and Joshi (2018) with research focusing on the sophisticated characteristics and attributes associated with zero-day exploits. The researcher explains that threat actors recognise the act of patience when discovering zero-day vulnerabilities, a concept Wrightson (2015) associates with the APT lifecycle. Coincidentally, the severity of this attack vector has encouraged extensive research into mitigation frameworks.

## 2.4 Zero-Day Vulnerability Detection Techniques

Due to the unavoidable nature, current mitigation frameworks approach improving the accuracy of various mitigation goals. Extensive literature analysis shows that many existing frameworks target different areas to ultimately reduce risk relating to zero-days attacks. Different areas of zero-day mitigation include vulnerability detection, exploit detection, attack path identification, and reducing Zero-Day Attack Period (ZDAP).

A variety of techniques and frameworks have been developed to mitigate risks, and many involve some form of machine-learning, anomaly detection, and predictive defence. Hammarberg (2014)

describes how techniques can be classified into four main categories: behavioural, statistical, signature and hybrid-based detection.

Basing a framework on an individual technique is highly disadvantageous because it limits accuracy against a wider range of zero-day threats. Hammarberg (2014) acknowledges this theory and describes signature and statistical-based techniques as static, as they lack the ability to support changing zero-day attack vector trends. Kaur and Singh (2014) agrees and explains that statistical-based detection techniques, like the Semantics Aware Statistical (SAS) algorithm by Kong et al. (2011), can be easily evaded by injecting obfuscated packets into normal traffic.

Behaviour-based techniques focus on predicting future interactions by examining captured network traffic. These methods often involve anomaly detection techniques like honeypots to capture, analyse, and understand interactions outside subjectively defined behaviour groups. Often used in combination with signature-based techniques, seen in a successful hybrid mitigation technique by (Joshi, Singh, and Kanellopoulos, 2018).

The hybrid approach to zero-day mitigation addresses the limitations of single technique frameworks by collating benefits, leading to higher mitigation rate accuracy and a lower risk of false positives. Table 2.1 collates important mitigation frameworks from various security researchers, along with brief classification into the techniques implemented. A comprehensive technical description of each framework can be found in Appendix C.

| Author | B | A | Si | Sa | H |
|---|---|---|---|---|---|
| Joshi, Singh, and Kanellopoulos (2018) | ✓ | × | ✓ | × | ✓ |
| Kaur and Singh (2014) | × | ✓ | ✓ | × | ✓ |
| Aleroud and Karabatis (2013) | ✓ | ✓ | ✓ | × | × |
| Kong et al. (2011) | × | × | ✓ | ✓ | × |
| Sun, Dai, Liu, Singhal, and Yen (2016) | ✓ | ✓ | ✓ | × | × |
| Blaise, Bouet, Conan, and Secci (2020) | ✓ | ✓ | ✓ | × | × |
| Kao et al. (2015) | ✓ | ✓ | × | × | × |

Table 2.1: Detection Methods Review Table

Key: B = Behaviour-based, A = Anomaly-based, Si = Signature-based, Sa = Statistical-based, H = Hybrid-based

## 2.5 Evolution of Advanced Persistent Threats

Adapting to technological evolution is an essential method in ensuring effective threat mitigation. The following sections describes concepts which have increased the risk of APTs. Addressing the following concepts is vital when developing a successful mitigation framework.

### 2.5.1 Vulnerability of Complexity

Cyber-criminals adapt to the changing landscape by increasing the sophistication of their exploits. A publication by Markakis (2019) acknowledges positive correlation between technology dependency and threat complexity, and suggests reasoning lies with increasing numbers of connected devices. To complement this research, Castelluccio (2015) claims this new landscape is a result of the upward trend in IoT devices combined with the absence of security prioritization.

Research by Wrightson (2015) provides further depth to the idea by explaining that this issue evolved from general lack of user awareness. Interpreted as the Vulnerability of Complexity; it is understood that complication in software, hardware, and organisational infrastructure limits

understanding, igniting a reaction which ultimately leads to increased number of vulnerabilities. As an example, a software experiment by Javed, Alenezi, Akour, and Alzyod (2018) supports this idea by revealing how file and code quantity increase probability of vulnerabilities. Connecting this idea with the probabilistic nature of zero-day threats is a clear justification for accelerating threat complexity.

### 2.5.2 Objective-Value-Proportion

Another approach to understand the reasoning behind the development of APTs involves the concept of objective-value-proportion. The theory refers to the level of subjective importance an asset or entity holds within an organisation. Figure 2.2 visualises this concept and describes the associated risk measurement matrix.

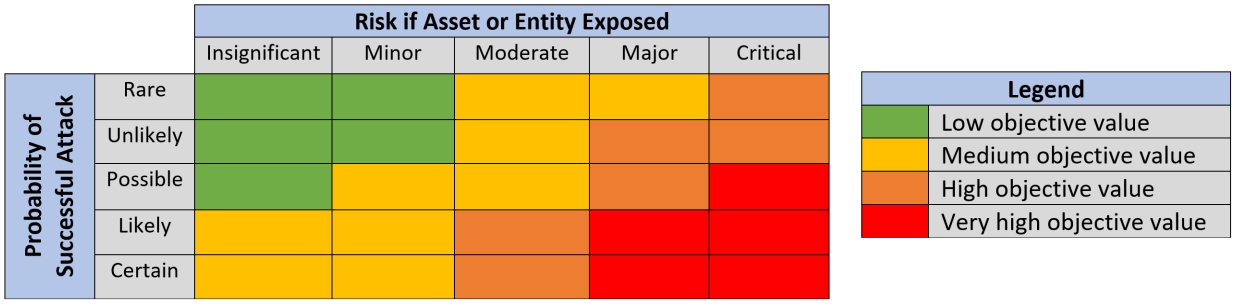



Figure 2.2: Objective Value Measurement Matrix

Research by Lake (2020) explains how the value of an objective should correlate to defence complexity. As an example, a government server stores high-risk data and thus requires stronger cyber defence solutions when compared to personal workstation security. The author also addresses the idea that the increasing value of assets encouraged the growth in threat sophistication, with the study explaining that a higher defence perimeter requires more sophisticated threats to guarantee objective success. Further to this, Woody (2013) presents another example by describing how the advancement of technology in business environments have increased risk. The author explains that quantity and quality of data stored has a positive correlation on risk if exposed or leaked.

While this concept is regularly addressed in security frameworks; the changing landscape of IT has often caused objective-value to become misunderstood. Wrightson (2015) supports this theory and argues that security frameworks are rarely in proportion with the value of identified objectives. Malicious intentions have evolved to cover a wider area of IT than previously assumed; with the author explaining that even computer resources become an objective to a willing APT hacker, referring to a discovered technique of obtaining botnets for further DDoS attacks. Known APT objectives now include computer resources, intellectual property, essential infrastructure systems, and government systems.

The advancement in objective-value is a likely reason behind the establishment of zero-day vulnerability markets. Bradbury (2018) implies that the effectiveness of APTs caused an increase in supply and demand for undisclosed vulnerabilities and bugs, eventually developing into an ethically unstable marketplace. This ecosystem is an example of malicious threat actors adapting to technological advancement by identifying the importance of objective-value.

### 2.5.3 Threat Misinterpretation

Failure to adapt to evolution also results in threat misinterpretation, an important topic which affects mitigation framework effectiveness. Previous threat documentation relied on data analysis of past security breaches; however, threat sophistication has reduced the effectiveness of this approach.. It is understood by Woody (2013) that threat classification is a subjective topic and involves target perspective. For definition, attacks which are subjectively more complex than the targets security measures are recognised as advanced threats.

Elaboration of Cyber Threat Intelligence (CTI) can focus the scope of a threat to ensure detailed classification; ultimately improving mitigation strategies. A study by Jouini, Rabai, and Aissa (2014) recognises the importance and presents a solution to improve the relationship between quality of CTI and threat assessment accuracy. Despite this, unknown sources of threats are not considered in this classification architecture, limiting effectiveness to classify more advanced threats.

In contrast, a classification model by Kolokotronis and Shiaeles (2019) supports advanced threat identification. The focus centres around Machine-Learning based Graphical-Cyber-Security-Models (ML-GCSM) and how merging these technologies can provide a dynamic approach to identify unknown threats. Threat attribute identification is a key mitigation method when involving advanced threats; Wrightson (2015) implies that growth of advanced threats is associated with the lack of proactive threat interpretation.

## 2.6 Existing Approaches to Security Models

As indicated by Bedell and Bouchard (2018), the rising sophistication of threats corresponds to a surge in more available defence tools. While this is true to some extent, Bradley (2019) argues that current security models are outdated and generally lack aspects of the NIST Framework. This claim is further backed by Woody (2013), who explains that current architecture is static and inflexible, and that a reactive security approach is no longer suitable for the increasing risks associated with organisations.

An investigation by Suby (2018) details another explanation from a business-focused perspective. While the researcher reinforces the belief that current security models are often outdated, the study claims that the problem lies with the organisations overall attitude to enforcing security. The focus implies organisations lack expertise, effort, resources, and personnel required to maintain an effective security model. These particular conclusions can be used further in this dissertation to encourage the use of out-sourced SOC environments.

Yet again, recent security trends have driven organisations to improve security tools, notably advancements in firewalls and anti-virus software. Whilst these are effective in blocking known malware, Bedell and Bouchard (2018) claims that such tools are ineffective against APTs. This construct is supported by Sukwong, Kim, and Hoe (2011), who studies the effectiveness of antivirus against various malware infections. Analysis of signature and behaviour-based detection techniques confirms that security tools will never provide 100 percent protection against advanced threats; due to the complex understanding required. These threats are frequent in today's age and the exponential growth of zero-days can cause severe problems if targeted systems depend solely on prevention-based tools.

Many publications recognise that businesses often fail to adapt security approaches in correspondence to cyber threat evolution. This belief is explored by Bedell and Bouchard (2018), who present a solution after a comparative analysis of alternate SOC options. They conclude that cloud-hosted SOC overcomes the problems faced by current security models. As a result,

the ability to recognise technological evolution in enterprise security is key to the reasoning behind establishing more dynamic defence solutions.

## 2.7 Introducing the SOCaaS Framework

Across many articles, the importance of SOCaaS is explained through the advanced features it provides with regards to threat detection and response capabilities. An early study by Alruwaili and Gulliver (2014) introduces the concept of cloud-based SOC, along with a detailed breakdown of framework architecture. The focus was to justify how a SOCaaS framework would ultimately improve threat mitigation, and in turn reduce system vulnerabilities. Discussed benefits include scalability, enhanced customer visibility and improved operational processes; all of which are viewed as proactive security measures. Although providing a high-level analysis, this approach to introducing SOCaaS is limited to cloud-based computing environments, and does not explore implementation in other business infrastructure. Moreover, there is a lack of discussion relating to financial requirements, a key factor in promoting businesses awareness.

Further analysis into associated economic advantages is supported by Suby (2018), who presents the cost-efficiency of SOCaaS using graphical illustrations. Statistical cost-comparison is based on interview responses from legitimate SOCaaS customers, ensuring dependable results. This focus is critical in developing a deeper understanding into the economic factors which strengthen views on SOCaaS. An article by Bedell and Bouchard (2018) summarises that SOCaaS provides solutions to many limitations of alternative SOC frameworks, such as SIEM, MSS and MDR models. Detailing these security models is a key theorem which emphasizes the benefits of SOCaaS considerably.

The theory that SOCaaS is a suitable security model is constituted, yet it could be argued that some studies have a subjective approach in their methods of underlining the value of SOCaaS. Research into the requirement for more widespread adoption of cloud-based security models is robust, yet it is evident businesses rarely take advantage. The comparison between managed SOC and SOCaaS is well-established, Richmond (2019) clarifies that a cloud-based approach improves on detection and response services; capabilities which are evidently lacking in current security models. Nonetheless, there is a lack of robust research on how SOCaaS can effectively counter more serious attack vectors. Considering the growth in threat sophistication, further investigation into this area could lead to wider enterprise adoption.

# Chapter 3

# Research Methodology

The following points provide a method and background for how the objectives will be achieved.

## 3.1 Objective 1

In order to meet the first objective, existing literature from articles, journals, books, and conference papers will be read and reviewed to achieve a clearer understanding. Key features to be looked at include specific areas of zero-day risk mitigation, such as exploit detection, vulnerability detection and current methods to reduce ZDAP. Detailing APT evolution can also justify how advantages of introducing SOCaaS relate to aspects of zero-day mitigation, and how such a framework can protect organisations. Papers and blogs will also be used to provide a background to cloud-based SOC environments, with the aim of highlighting the advantages.

## 3.2 Objective 2

The aim is to analyse previous zero-day attacks to identify current/future trends and provide mitigation procedures based on attribute classification. This function will be achieved by measuring the following variables: target system, current status, access vector , identification date, and CVE threat level.
The software used will be Python 3.5 because it offers many libraries dedicated to data analytics, including Matplotlib, Pandas, Numpy, and Sckit-learn. The dataset is to be obtained from an open-source zero-day vulnerability tracking website, developed by Marchuk (2021). Once downloaded, the dataset will be edited manually to suit the research scope. Further to this, scripts to perform analysis will be developed using Spyder, an open-source IDE based on the Anaconda3 package manager platform. Scripts will perform specific techniques to identify hypothesis correlation, including descriptive analytics, linear-regression and clustering. This analysis aims to harvest quantitative information to support further qualitative result interpretation for SOCaaS teams. With this research, more information can be gathered regarding how proactive techniques can reduce risk of zero-day attack.

## 3.3 Objective 3

This objective will be achieved by using qualitative data from analysis to explore how zero-day attacks can be mitigated with higher efficiency if a SOCaaS environment is involved. Examining the relationship between these two constructs can provide value justification, with the intention to encourage the use of SOCaaS in business environments. The hostile nature of zero-day attacks

means total mitigation is often non-existent; to first recognise a success matrix, principles of measuring system intrusion are to be addressed. Identifying quality of mitigation will be in the form of reviewing existing literature on the topic, correlation assessment will consider these measurements factors of success. Interpretation of predictive trends can discover if the qualities of SOCaaS can benefit zero-days risk mitigation.

## 3.4 Who will benefit from this research

This research can benefit any organisation which uses a form of information technology, the basis of a zero-day attack surrounds a weakness, bug or flaw discovered in a target system. Because these vulnerabilities are so difficult to prevent, researchers like Vaisla and Saini (2014) consider subsequent attacks to be unavoidable. Past evidence suggests zero-day attacks are often state-sponsored, and generally target high value systems, like government systems or essential infrastructure services. However, because zero-day attacks are a vector associated with APT hackers, targets can be considerably diverse. Wrightson (2015) reports the wider range of objectives available for malicious actors to leverage have caused the threat of zero-day attack to spread into every form of computer technology. Essentially, any organisation which processes and stores data should be considered vulnerable, however the subjective value of an asset has a direct correlation to risk. Research into promoting the use of cloud-hosted SOC environments will also benefit Small and Medium Sized Enterprises (SME), a study by Suby (2018) suggests that current security models adopted by these types of companies lack proactive solutions to reduce risk of zero-day attacks. To achieve this, the research will also explore economic advantages to encourage use of this theoretical framework.

## 3.5 Research Methodology Flowchart

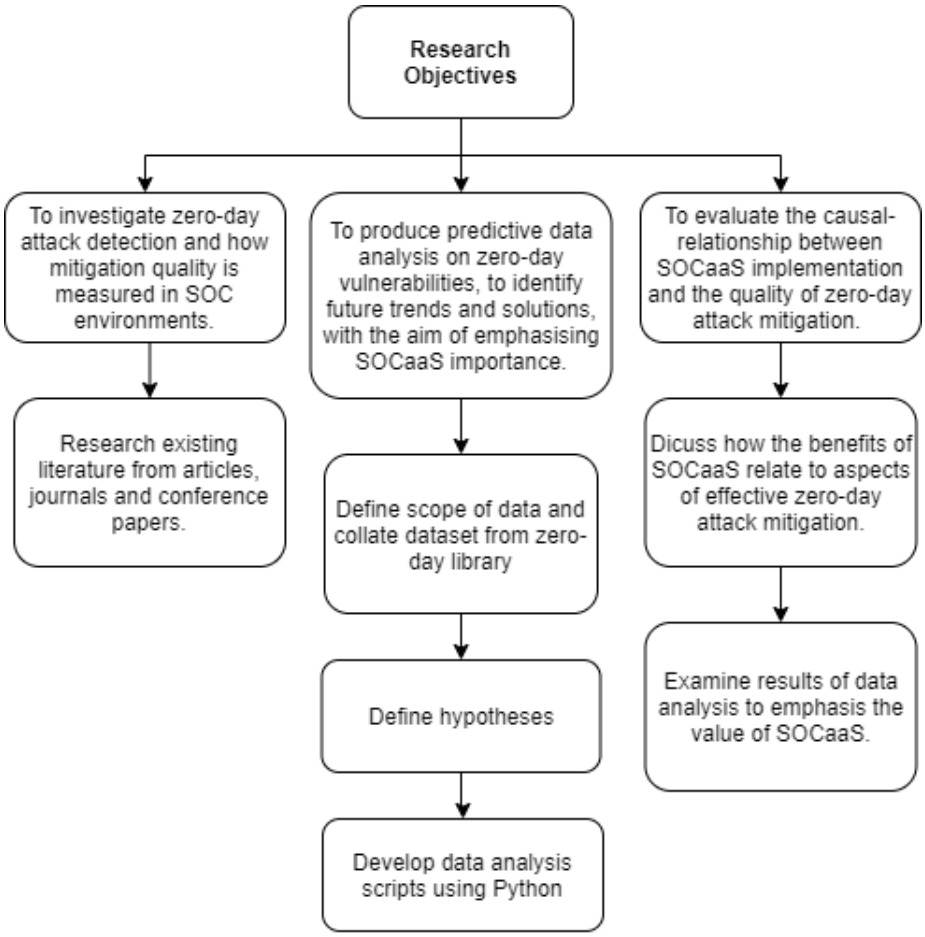



Figure 3.1: Research Methodology Flowchart

# Chapter 4

# Analysis

## 4.1 Introduction

This section will detail the process required to apply a descriptive and predictive analytic interpretation of data. Data was collected from an open-source zero-day vulnerability tracking website, developed by Marchuk (2021). As per the scope of this research, zero-day vulnerabilities between the months of January 2020 to April 2021 were documented. During this 16-month period, a total of 49 vulnerabilities were published with complete CVE reports. Each script output presents a graphical figure to support further qualitative interpretation. Python code for each figure is found in Appendix E. The following measurement variables were recorded for each published CVE.

- Target System
- Access Vector
- Time Until Patched
- Access Complexity
- CVSS 3.0 Threat Score
- Level of CIA Compromised

### 4.1.1 Hypotheses

These variables can be used to quantify important statistics, including threat severity and future target vectors; further identifying the need for cloud-hosted SOC. The following table visualises hypotheses which supports these ideas.

| Hypothesis type | Abbreviation | Hypothesis |
|---|---|---|
| Alternate | A1 | Positive correlation between time and number of zero-days |
| Null | A2 | Negative correlation between time and number of zero-days |
| Alternate | B1 | Positive correlation between time and severity of zero-days |
| Null | B1 | Negative correlation between time and severity of zero-days |

Table 4.1: Table of Hypotheses

## 4.2 Number of Zero-Day Vulnerabilities

Figure 4.1 pictures the first identifiable trend; the monthly amount of recorded zero-day vulnerabilities has increased over the 16-month period. From the bar chart, we can see that the rise has been somewhat steady and gradual. The noticeable absence of new vulnerabilities between the months of May to July could indicate a period of dormancy, a common characteristic associated with zero-days.

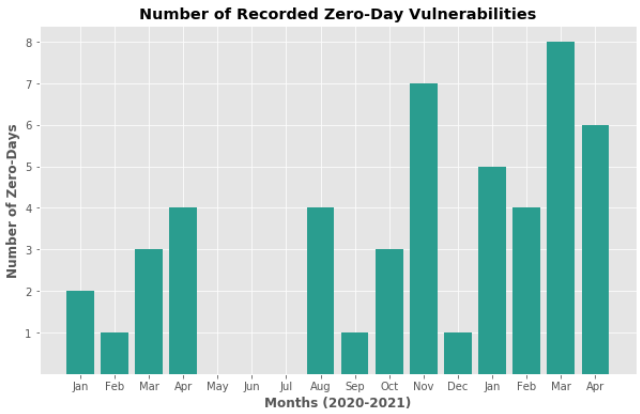



Figure 4.1: Number of Recorded Zero-Day Vulnerabilities

Figure 4.2 shows a simple linear regression model, also comparing number of recorded zero-day vulnerabilities per month. Linear regression is the first stage in building an effective prediction model. A regression line can be used to make accurate predictions of future trends, directions, and relationships between variables. Linear regression exploits the linear relationship between two or more variables, making this script useful for predicting trends. The output of this script visualises a gradual-positive regression line, further reinforcing hypothesis A1.

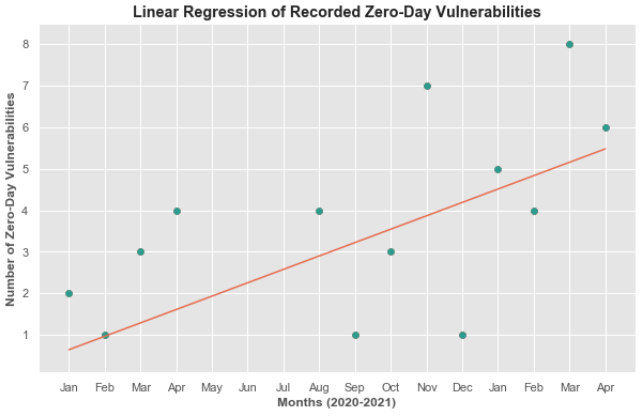



Figure 4.2: Linear Regression of Recorded Zero-Day Vulnerabilities

## 4.3 Severity of Zero-Day Vulnerabilities

To fully understand hypothesis B, severity metrics first need to be understood and identified. The following sections explore techniques to quantify mitigation importance for vulnerabilities recorded in the dataset.

### 4.3.1 CVSS 3.0 Threat Score

A recognisable method to quantify vulnerability severity is provided through the Common Vulnerability Scoring System (CVSS). This published standard offers a framework to produce numerical scores reflecting a vulnerabilities severity. When a CVE is disclosed, analysts use version 3.0 of this framework to assign a numerical value ranging from 0 to 10. As noted from the specification document (FIRST.org, 2019), three metric groups are followed: base, temporal and environmental. Base and temporal metrics are usually specified by target vector vendors and environmental metrics are relative to the organisation.

Figure 4.3 measures CVSS 3.0 threat score across all zero-day vulnerabilities recorded over 16 months. Numerical scores are grouped to better highlight the spread of severity, very few recorded zero-days have a threat score below 5. Most CVEs had a threat score between 7 and 7.9, representing the average. The dangerous nature of zero-day vulnerabilities is emphasised through the evidently higher quantity of CVEs with a score between 9 and 10, the highest possible metric.

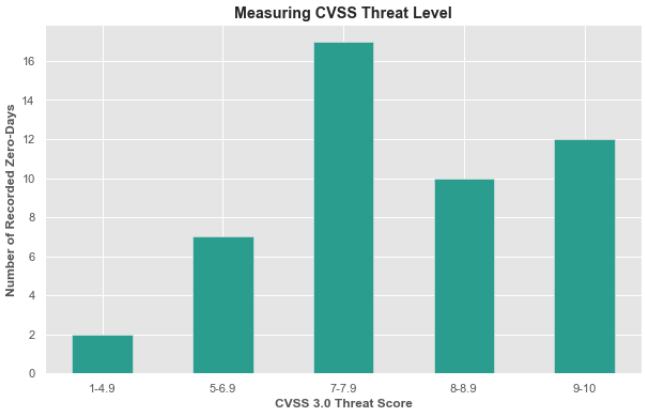



Figure 4.3: Measuring CVSS Threat Level

Figure 4.4 confirms that most recorded CVEs are measured with a high threat level. However, the scatter graph shows little correlation between time and threat CVSS score, with the average linear regression line showing a very gradual upwards slope. Over the 16-month analysis period, the average CVE is measured at 8. Measuring CVSS threat score alone does not provide enough data to answer hypothesis B, meaning different metrics must be used for justification.

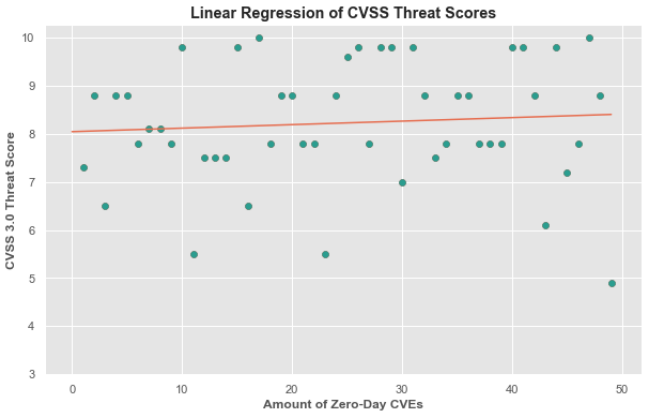



Figure 4.4: Comparing CVSS Threat Level to Target Vector

### 4.3.2 Attack Complication

Figure 4.8 visualises CVE attack complexity against the monthly number of recorded vulnerabilities. A vulnerability with a low level of attack complexity holds much more risk, due to the higher likelihood of exploitation, thus justifying a higher severity. Low attack complexity allows a vulnerability to be exploited by a larger range of individuals, a result of lower level of technical knowledge required. The bar chart shows that very few recorded CVEs require high difficulty to exploit, reinforcing hypothesis B1 by visualizing an increase in zero-day vulnerabilities requiring low levels of attack complication.

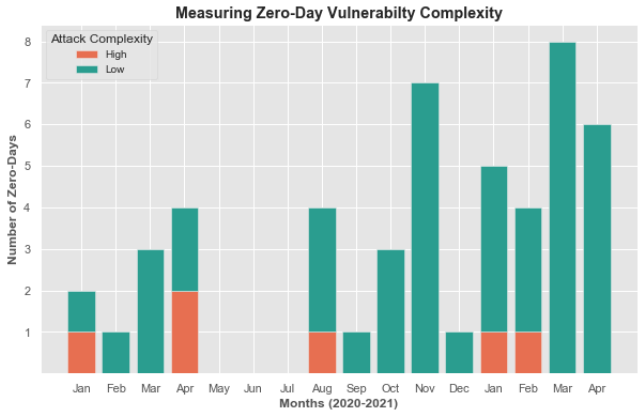



Figure 4.5: Measuring Zero-Day Vulnerability Complexity

### 4.3.3 Confidentiality, Integrity and Availability Triad

Figure 4.8 represents severity by measuring how zero-day vulnerabilities individually affect target confidentiality, integrity and availability. Defied by the CVSS framework, CIA refers to the impact metric of a successfully exploited vulnerability. This metric reflects the worst possible outcome for the impacted target or component. Vulnerabilities with a high impact are much more severe if exploited, compared to CVEs with low impact. The bar chart shows an increase in the amount of zero-days which recorded having a potentially high impact on an aspect of CIA, further providing another metric to support hypothesis B1.

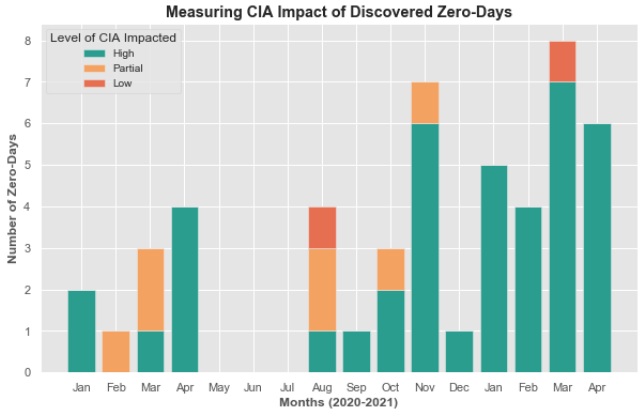



Figure 4.6: Measuring CIA Impact of Discovered Zero-Days

### 4.3.4 Zero-Day Attack Period

Zero-Day Attack Period (ZDAP) refers to the length of time a vulnerability is exploitable. As a result, the longer time it takes to patch a zero-day reflects a CVEs severity. This metric is not included in the CVSS framework, meaning ZDAP is subjective. Figure 4.7 shows that 73% of zero-day vulnerabilities between Jan 2020 and Apr 2021 received patches within one day, significantly reducing ZDAP and thus severity. Despite this, a considerable number of vendors took longer than a week to patch, and 4 zero-days remain unpatched as of April 2021. This could be a result of several factors, including cost, lack of expertise and even the organisational impacts of patching systems.

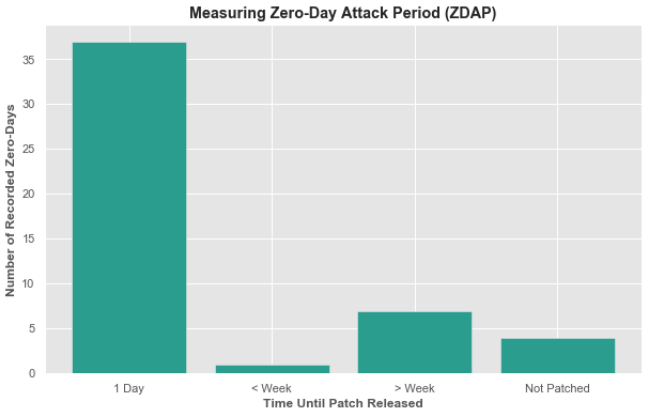



Figure 4.7: Measuring Zero-Day Attack Period (ZDAP)

## 4.4 Classification of Zero-Day Vulnerabilities

Zero-day vulnerabilities can be classified using various variables, for this research we explore target vectors and access vectors. Target vector can be grouped into three categories, browser, operating system and software. These targets reflect which area of technology which the zero-day could potentially target and successfully exploit. Figure 4.8 classifies the target vectors associated with each recorded zero-day CVE. In recent months, software applications have been the target vector for an increasing number of zero-day vulnerabilities. Figure 4.9 details further classification of zero-day vulnerabilities in a pie chart format, including access vectors and a deeper view into specific browser and OS types.

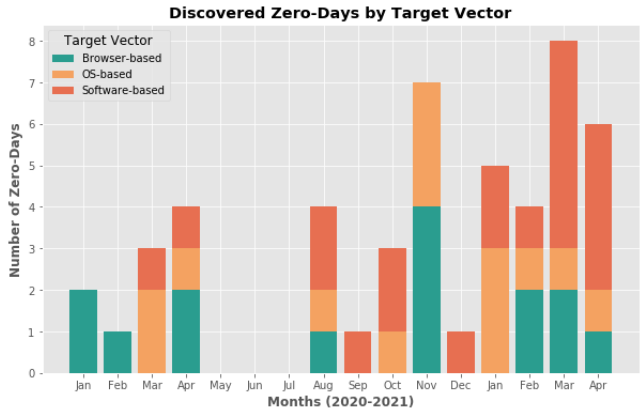
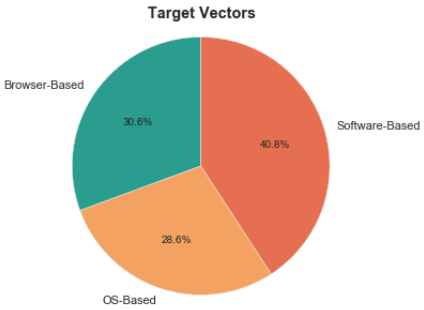
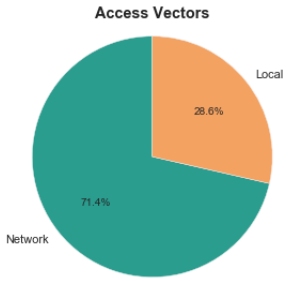


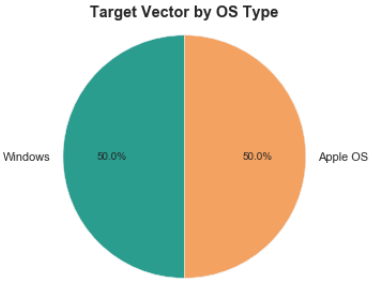

Figure 4.8: Measuring Zero-Day Vulnerabilities by Target Vector



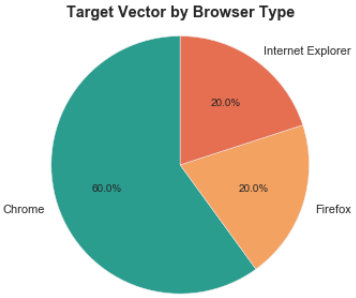

(a) Target Vectors




(b) Access Vectors




(c) Target Vector by OS Type




(d) Target Vector by Browser Type

Figure 4.9: Zero-Day Vulnerability Classification Pie Charts

## 4.5 Conclusion

After visualising data, it is clear that zero-day vulnerabilities are increasing in number and severity. Factors of severity include ZDAP, level of CIA compromised, attack complexity and CVSS threat score. An increasing number of vulnerabilities are software-based, with 20 CVEs falling under this category. This data has the potential to aid a SOCaaS team by encouraging adaptability to recommend mitigation for future vulnerabilities. The positive correlation between variables presents a difficult situation for organisations who lack the resources to setup and maintain a SOC capable of mitigating zero-days threats.

# Chapter 5

# Discussion

## 5.1 Introduction

This chapter explores the properties of zero-day vulnerabilities how these relate to characteristics of SOCaaS. The benefits of SOCaaS will be explored to help identify any correlation between implementing this mitigation framework and zero-day threat management.

## 5.2 Future Zero-Day Trends and Predictions

Data analysis from chapter 4 offers insight into future trends of zero-day vulnerabilities. Both alternate hypothesis were met, with a positive correlation being discovered between time and severity and number of zero-day vulnerabilities. The theory that zero-day vulnerabilities are increasing in number and severity presents a considerable amount of risk towards organisations. As mentioned previously, any organisation which processes and stores data should be considered vulnerable to zero-day attack.

Between the period of analysis, a growing number of zero-day vulnerabilities have been discovered. The simple linear regression graph shows an upward trend between time and number of vulnerabilities publicly reported. This relationship has the potential to increase in future months, with evidence suggesting zero-day vulnerabilities could become more common in 2021 onwards. This upward trend is also apparent with many variables of measuring severity, including CVSS threat score and attack complexity. Analysis shows that over January 2021 and April 2021, the average CVSS threat score is around 8. 24.4% of the total measured CVEs scored between 9.5 and 10, the highest possible metric. This reinforces the common theory that zero-day vulnerabilities are considered the most dangerous cyber threats in the modern IT landscape.

## 5.3 Characteristics and Qualities of SOCaaS

The collaboration between SaaS and SOC introduced a new environment for virtual incident management. SOCaaS is provided as a third-party framework to offer a more proactive approach to security analysis, by involving human-augmented machine learning. Many well-known security companies have now developed their own SOCaaS frameworks, with companies like AWN and AT&T now offering outsourced security management. By design, SOCaaS brings several benefits when compared with alternative security solutions, including notable improvements to scalability, availability and cost. These properties all align with effective security management

procedures, allowing this framework to provide a high level of defence against new and existing cyber threats; along with supporting general business fundamentals. Appendix D explores operational processes which form the SOCaaS framework, as defined by Alruwaili and Gulliver (2014).

### 5.3.1 Cost Efficiency

One of this security model's greatest strength is the low-cost relative to alternative SOC solutions. SIEM, MSS and MDR models all require time and resources to establish and maintain, with this being a significant limiting factor for many organisations. For example, an effective SIEM system requires hiring security analysts, introducing new technologies, policies and procedures, and training and expertise. Many organisations lack the budget for an internal SOC, which may explain the lack of effective incident response capabilities among SME businesses. SOCaaS offers a solution to this problem by including all core SOC requirements under a subscription-based service. This simplifies security management by reducing the necessary resources required to manually introduce a SOC, also known as DIY SOC.

Suby (2018) presents a three-year Total Cost of Ownership (TCO) comparison between internal SOC and SOCaaS provided by Arctic Wolf. The study offers an estimated breakdown of cost components, noting that staffing related expenses represent 96% of the total cost to manually setup a SOC. Table 5.1 presents a cost-comparison between DIY SOC and AWN SOCaaS for three years, between small, medium and large businesses.

| Size | End-users | AWN SOCaaS Average Cost | Estimated DIY SOC Cost |
|---|---|---|---|
| Small | 500 | $279,000 - $346,000 | $2,410,000 |
| Medium | 1000 | $502,000 - $591,000 | $3,000,000 |
| Large | 3000+ | $1,304,000 - $1,563,000 | $5,145,000 |

Table 5.1: AWN SOCaaS vs DIY SOC Costs - Retrieved from Suby (2018)

Principle observations highlight the affordability of outsourcing SOC services, with estimates discovering the cost for a small organisation to maintain a DIY SOC to be 8.8 times higher than a SOCaaS solution. DIY SOC pricing is highly volatile with many variables, including geographical location and incremental staffing costs. In contrast, SOCaaS pricing is much more structured and predictable, with AWN costs ranging between defined business sizes.

### 5.3.2 Improved Availability

The approach to outsource SOC as a cloud-based service also improves availability of security analysis and risk assessments. Basing this framework on a cloud-based platform allows SOCaaS to support 24/7 monitoring and response, a critical requirement for effective mitigation in today's threat landscape. Continuous environment monitoring is a key step in discovering malicious activity, SOCaaS provides 360-degree visibility by integrating System Agents to log activity across an entire business network. These sensors are deployed through cloud-applications and site premises to capture traffic and logs in real-time. The presence of Indicators of Compromise (IoC)) are much clearer when processed by SOCaaS, allowing for significantly reduced Mean Time To Detect (MTTD) and Mean Time To Respond (MTTR) when compared to manually integrated SOC. Many SOCaaS solutions offers simplified portals to further increase visibility; subscribing customers have the ability to learn about their security posture from detailed reports. Bedell and Bouchard (2018) also note the SOCaaS staffing model allows for 27/4

support from a dedicated team of security professionals, providing recommended remediation actions when a threat is discovered.

## 5.4 Correlation Between SOCaaS and Zero-Day Mitigation

Not only do these competitive advantages benefit customers from a business perspective, introducing SOCaaS allows for stronger mitigation for APT related threats. This section will explore how attributes of SOCaaS relate to zero-day attack mitigation.

### 5.4.1 Support for Technological Evolution

Adapting to technological evolution is an essential method in ensuring effective threat mitigation. Zero-day threats fall under the APT category, meaning they are highly sophisticated by nature and constantly evolving. Data analysis shows an increase in measured severity for zero-days discovered between 2020 and 2021. To compete with the evolution of zero-day vulnerabilities, mitigation frameworks must be scalable enough to support rapid detection and response. Existing SOC models often lack scalability, with many frameworks unable to mitigate more advanced zero-day threats. Woody (2013) suggests this is due to static and reactive architecture causing further organisational constraints when reacting to threats with speed and efficiency.

It is widely known that scalability is a significant attribute of cloud-hosted services, an aspect which SOCaaS subsequently capitalises on to better support technological evolution. The SOCaaS framework provides a virtual, scalable environment for security professionals to analyse security events with higher efficiency when compared to alternative solutions. This is an indirect improvement to SOC adaptability, with the pro-activeness of cloud-hosted operations allowing for a more direct line of response.

### 5.4.2 Advanced Monitoring

Many research papers discuss how zero-day vulnerabilities can be mitigated with higher efficiency when continuous monitoring is present. SOCaaS supports 24/7 monitoring and activity analysis, making this framework more effective in mitigating severe threats which often evade traditional SOCs. Zero-day vulnerabilities have also been seen to bypass ML-based mitigation frameworks; SOCaaS counters this by involving human-augmented machine learning, combining technology with human intelligence to apply risk management for the newest emerging threats. This involvement has allowed SOCaaS providers to decrease MTTD and MTTR, along with reducing false positives.

# Chapter 6

# Conclusion

To conclude, this research dissertation aimed to investigate the causal-relationship between SOCaaS implementation and the quality of zero-day attack mitigation. The literature review covers characteristics and evolution of zero-days to explain their increasing severity in today's technological landscape, noting a severe lack of association between APT risk mitigation and current security model frameworks. When carrying out data analysis against discovered zero-days between 2020 and 2021, we find that vulnerabilities are increasing in number and severity. This presents a difficult situation for organisations who lack the resources to setup and maintain an internal SOC, whilst requiring defence against these advanced vectors of attack. Existing security architecture is found to be outdated, static and not flexible enough to successfully mitigate more advanced threats. This research justifies how a cloud-hosted SOC framework overcomes existing limitations, along with explaining the how attributes of SOCaaS enables potential to support more advanced threat mitigation. Due to the unavoidable nature of zero-days, mitigation requires consistent activity analysis and monitoring. Frameworks also need to be proactive and scalable enough to support efficient patch response. These are both qualities a SOCaaS solution provides due to its virtual approach to 24/7 security management, and scalable framework offered through cloud environments. A business adopting this framework will see notable improvements to patch deployment efficiency, increased activity analysis and better support for technological evolution; hence protecting against the continuous growth in zero-day related threats.

# Bibliography


Al-Rabiaah, S. (2018). 'The "Stuxnet" Virus of 2010 As an Example of A "APT" and Its "Recent" Variances'. ISBN: 9781538641101.

Aleroud, A. and Karabatis, G. (2013). 'Toward zero-day attack identification using linear data transformation techniques', pp. 159–168. doi: 10.1109/sere.2013.16.

Alruwaili, F. F. and Gulliver, T. A. (2014). 'SOCaaS: Security Operations Center as a Service for Cloud Computing Environments'. *International Journal of Cloud Computing and Services Science (IJ-CLOSER*, 3 (2), pp. 87–96. ISSN: 2089-3337. Available at: http://iaesjournal.com/online/index.php/.

Bedell, C. and Bouchard, M. (2018). *Definitive Guide to SOC-as-a-Service*. Available at: https://www.arcticwolf.com..

Blaise, A., Bouet, M., Conan, V., and Secci, S. (2020). 'Detection of zero-day attacks: An unsupervised port-based approach'. doi: 10.1016/j.comnet.2020.107391. Available at: https://hal.archives-ouvertes.fr/hal-02889708.

Bradbury, D. (2018). *The Murky Market for Zero-Day Bugs*. Available at: https://www.infosecurity-magazine.com/magazine-features/the-murky-market-for-zeroday-bugs/.

Bradley, T. (Oct. 2019). *The Standard Cybersecurity Model Is Fundamentally Broken*. Available at: https://www.forbes.com/sites/tonybradley/2019/10/07/the-standard-cybersecurity-model-is-fundamentally-broken/?sh=6e47f8f91189.

Brewer, R. (2014). *Advanced persistent threats: minimising the damage*.

Castelluccio, M. (2015). *EMERGING CYBER THREATS*. Available at: https://search-proquest-com.glos.idm.oclc.org/docview/1672624862/fulltextPDF/C387CE6D65F542C7PQ/1?accountid=27114.

Emm, D. (Dec. 2019). *APT review: what the world's threat actors got up to in 2019*. Available at: https://securelist.com/ksb-2019-review-of-the-year/95394/.

FIRST.org (2019). *Common Vulnerability Scoring System version 3.1 Specification Document Revision 1*. Available at: https://www.first.org/cvss/specification-document.

Google (2020). *Google Trends*. Available at: https://trends.google.com/trends/explore?date=all&q=SOC%20as%20a%20service.

Hammarberg, D. (2014). *The Best Defenses Against Zero-day Exploits for Various-sized Organizations*.

Javed, Y., Alenezi, M., Akour, M., and Alzyod, A. (2018). 'Discovering the Relationship Between Software Complexity and Software Vulnerabilities'. *Article in Journal of Theoretical and*

*Applied Information Technology*, 31 (14). ISSN: 1817-3195. Available at: https://www.researchgate.net/publication/327436392.

Joshi, C., Singh, U. K., and Kanellopoulos, D. (2018). *An Enhanced Framework for Identification and Risks Assessment of Zero-Day Vulnerabilities*, pp. 10861–10870. Available at: http://www.ripublication.com.

Jouini, M., Rabai, L. B. A., and Aissa, A. B. (2014). 'Classification of security threats in information systems'. Vol. 32. Elsevier B.V., pp. 489–496. doi: 10.1016/j.procs.2014.05.452.

Kao, C. N. et al. (Dec. 2015). 'A predictive zero-day network defense using long-term port-scan recording'. Institute of Electrical and Electronics Engineers Inc., pp. 695–696. ISBN: 9781467378765. doi: 10.1109/cns.2015.7346890.

Kaur, R. and Singh, M. (2014). 'Efficient Hybrid Technique for Detecting Zero-Day Polymorphic Worms'. *ITM University. Department of Computer Science Engineering and Information Technology.*,

Kolokotronis, N. and Shiaeles, S. (2019). 'Effective response and mitigation of advanced cyber-attacks via an intelligent cyber-defence framework'. Available at: https://www.openaccessgovernment.org/advanced-cyber-attacks/73967/.

Kong, D., Jhi, Y.-C., Gong, T., Zhu, S., Liu, P., and Xi, H. (2011). *SAS: Semantics Aware Signature Generation for Polymorphic Worm Detection.*

Lake, J. (2020). *What is an advanced persistent threat (APT), with examples.* Available at: https://www.comparitech.com/blog/information-security/advanced-persistent-threat/#How_to_defend_against_advanced_persistent_threats_APTs.

Marchuk, V. (2021). *Zero-Day Tracking Project.* Available at: https://www.zero-day.cz/.

Markakis, E (2019). *A Universal Cyber Security Toolkit for Health-Care Industry.* Available at: http://sphinx-project.eu/.

Richmond, C. (2019). *SOCaaS versus Managed SOC (with video).* Available at: https://www.esg-global.com/blog/socaas-versus-managed-soc-with-video.

Singh, U. K. and Joshi, C. (2018). 'Scalable Approach Towards Discovery of Unknown Vulnerabilities'. *International Journal of Network Security*, 20 (5), pp. 827–835. doi: 10.6633/ijns.201809.20(5).03. Available at: https://www.researchgate.net/publication/325313064.

Standards, N. I. of and Technology (2012). *NIST Special Publication 800-30 Revision 1: Guide for conducting risk assessments.* National Institute of Standards and Technology, B–1–B–1. doi: 10.6028/nist.sp.800-30r1. Available at: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-30r1.pdf.

Suby, M. (2018). *SOC-as-a-Service versus DIY SOC The Business Case for AWN CyberSOC is Compelling.*

Sukwong, O., Kim, S. H., and Hoe, C. J. (2011). *Commercial Antivirus Software Effectiveness: An Empirical Study.* Available at: www.symantec.com/.

Sun, X., Dai, J., Liu, P., Singhal, A., and Yen, J. (2016). *Towards Probabilistic Identification of Zero-day Attack Paths.*

Tecuci, G., Marcu, D., Meckl, S., and Boicu, M. (2018). *Evidence-Based Detection of Advanced Persistent Threats.* Available at: www.computer.org/cise.

Vaisla, K. S. and Saini, R. (2014). *Analyzing of Zero Day Attack and its Identification Techniques*. Available at: https://www.researchgate.net/publication/260489192.

Woody, A. (2013). *Enterprise Security: A Data-Centric Approach to Securing the Enterprise*.

Wrightson, T. (2015). *Advanced Persistent Threat Hacking*. McGraw-Hill. Available at: https://r1.vlereader.com/Reader?ean=9780071828376#.

# Appendices

# Appendix A

# Glossary of Terms

**APT** Advanced Persistent Threat.

**CTI** Cyber Threat Intelligence.

**CVE** Common Vulnerability and Exposures.

**CVSS** Common Vulnerability Scoring System.

**IDE** Integrated Development Environment.

**IDS** Intrusion Detection System.

**IoC** Indicators of Compromise.

**IoT** Internet of Things.

**MDR** Managed Detection and Response.

**ML-GCSM** Machine-Learning based Graphical-Cyber-Security-Models.

**MSS** Managed Security Services.

**MTTD** Mean Time To Detect.

**MTTR** Mean Time To Respond.

**NIST** National Institute of Standards and Technology.

**PLC** Programmable Logic Controller.

**SaaS** Software as a Service.

**SIEM** Security Information and Event Management.

**SME** Small and Medium Sized Enterprises.

**SOC** Security Operations Centre.

**SOCaaS** Security Operations Centre as a Service.

**SOIDG** System Object Instance Dependency Graph.

**STF** Suspicious Traffic Filter.

**STG** State-Transition-Graph.

**TCO** Total Cost of Ownership.

**ZAE** Zero-day Attack Evaluation.

**ZDAP** Zero-Day Attack Period.

# Appendix B

# Gantt Chart

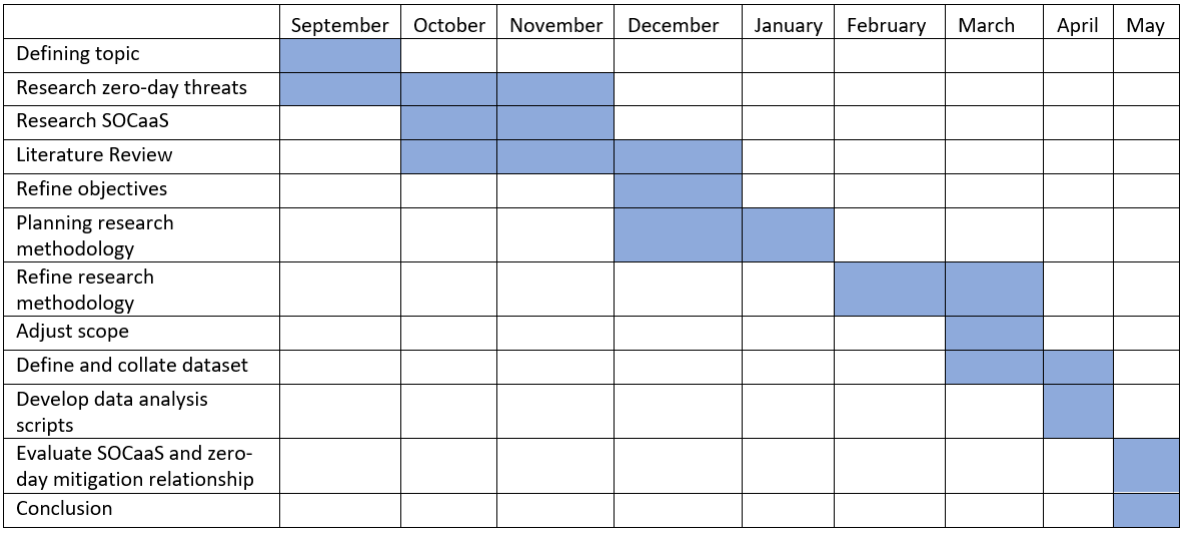



Figure B.1: Gantt Chart

# Appendix C

# Review of Zero-Day Mitigation Frameworks

| Author | Proposed Framework | Description | Outcome |
|---|---|---|---|
| Joshi, Singh, and Kanellopoulos (2018) | Probabilistic-based Detection and Prediction | The solution uses probabilistic-based detection to predict the likelihood of successful zero-day attack paths, along with classifying vulnerability severity. The authors propose three layers of framework architecture, involving a combination of behavioural and signature detection techniques to identify network related zero-day threats. | 96 percent average detection rate 0.3 percent false positive rate |
| Kong et al. (2011) | The Semantics Aware Statistical (SAS) algorithm | Performs statistical analysis on signature generation processes. This technique aims to automate polymorphic worm discovery by introducing a new method of signature matching. Involves two modules, semantic-aware signature extraction and semantic-aware signature matching. Data is then refined by the applied Hidden Markov Model to generate State-Transition-Graph (STG) based signatures. Although the framework can accurately detect worm signatures, the authors admit the algorithm does not support high level-obfuscation found in more modern zero-day worms. | Although the framework can accurately detect worm signatures, the authors admit the algorithm does not support high level-obfuscation found in more modern zero-day worms. |

| | | | |
|---|---|---|---|
| Sun, Dai, Liu, Singhal, and Yen (2016) | Probabilistic Identification of Zero-day Attack Paths | Proposed method aims to discover zero-day attack paths by introducing a Bayesian-Network (BN) technique to compute probabilities against intrusion evidence. System Object Instance Dependency Graph (SOIDG) is crafted to form a basis of BN analysis. Computing object instances with high infection probabilities with the Pr0bA system results in an identified zero-day attack path. | If attack time span is longer than analysed time period, the accuracy of generated SOIDG will be affected. |
| Aleroud and Karabatis (2013) | Linear Data Transformation | Three ML-based modules to approach zero-day attack detection. Linear data transformation calculates attack probability by analysing deviation between identified contextual network activities. Probability accuracy and anomaly detection is measured through the applied 1-class NN algorithm. | Low rate of false positives Good rate of detecting attacks |
| Kaur and Singh (2014) | Hybrid Polymorphic Worm Detection | Framework involves a combination of signature and anomaly-based techniques to detect and quarantine zero-day worms. Suspicious Traffic Filter (STF) passes traffic through Sebek-enabled honeypots to identify activity against signatures. The Zero-day Attack Evaluation (ZAE) function specialises in packet polymorphism and provides effective false-positive reduction. Content-based signatures are then generated for identified zero-day worms. | 96 percent average detection rate Almost 0 false positives |
| Kao et al. (2015) | Predictive Network Defence using Long-term Port-scan Recording | Proposed Prophetic Defender (PD) technique aims to reduce Zero-Day Attack Period (ZDAP). Focuses securing hosts by monitoring malicious port activity, using honeypot-based servers to detect evidence of port scans. A port scan attempt on the honeypot triggers temporary IP block using an open-flow based SDN switch. This block is made permanent when multiple attempts are discovered using the same scan. | Framework evaluation operated over 6 years shows that this method is 98 percent effective when detecting zero-day port scans. |

| | | | |
|---|---|---|---|
| Blaise, Bouet, Conan, and Secci (2020) | Unsupervised Port-based Network Anomaly Detection | The proposed Split-and Merge detection technique involves advanced port monitoring using the CIDS architecture. Focuses on zero-day botnet detection, using a statistical algorithm to spot anomalies when observing traffic. False positives are prevented by geographical modules sending anomalies to a central controller. Positive anomalies are considered based on their preceding location. | Detection rate of up to 100 percent when tested against 5 common botnets |

# Appendix D

# SOCaaS Framework Architecture

| Stage | Module | Description |
|---|---|---|
| 1 | System Agent (SA) | Virtual application or hardware appliance responsible for capturing and monitoring logs. Including physical component logs, application logs, user profile and activities, security logs, and directory service logs. Data is captured and sent in real time to the SEM module. |
| 2 | System Agent Management (SAM) | Responsible for deployment and maintenance of SA agents among infrastructure. Monitors and maintains SA to SEM connectivity. |
| 3 | System Event Management | Collects, correlates and analyses events and logs from System Agents. Consists of four components, Events and Logs Database (ELD), Event Correlation (EC), Event Knowledge Base (EKB), and Event Analysis (EA). |
| 4 | Event Response | Offers detailed responses consisting of recommended countermeasures according to predefined rules. Security analysts available to perform further analysis if required. |
| 5 | Integration Agent | Security policy agent which ensures operational compatibility of cloud security devices within the SOCaaS system. Ensures that only SOCaaS integrated devices, services or applications are able to perform event generation, detection and analysis. |
| 6 | Compliance and Audit Checking | Checks for compliance against security policies to ensure enforcement. Continuously scans and checks events with regulatory requirements and customer SLA agreements. |
| 7 | Security Assessment | Risk management module which determines vulnerability against malicious events. |
| 8 | Physical Security Monitoring | Allows for facility monitoring, and supports readiness in case of a physical emergency (fire, floods, electrical failure, etc). Connected to Personal Access Controls to monitor and log access. |

| | | |
|---|---|---|
| 9 | Reporting | Reporting is a component required for every module. Summaries all phases of events, along with forecasts. Aggregated to a central reporting module. Access is available if required by law enforcement authorities. |

# Appendix E

# Python Code

## E.1 Figure 4.2

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
#import libraries

A = np.array([2,1,3,4,0,0,0,4,1,3,7,1,5,4,8,6])
X = np.arange(16)
plt.figure(figsize=(10, 6))
plt.xticks(np.arange(min(X), max(X)+1, 1.0),
           ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
            'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec',
            'Jan', 'Feb', 'Mar', 'Apr'])
plt.yticks([1,2,3,4,5,6,7,8,9,10,11,12])
plt.bar(X, A, color="#2a9d8f")
plt.title('Number of Recorded Zero-Day Vulnerabilities', fontweight='bold')
plt.ylabel('Number of Zero-Days', fontweight='bold')
plt.xlabel('Months (2020-2021)', fontweight='bold')
plt.show()
```

## E.2 Figure 4.3

```
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import numpy as np
from sklearn.linear_model import LinearRegression

plt.style.use('ggplot')

plt.figure(figsize=(10, 6))
rng = np.random.RandomState(1)
x = np.array([1, 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16])
y = np.array([2,1,3,4,0,0,0,4,1,3,7,1,5,4,8,6])
plt.scatter(x, y);
```

```
model = LinearRegression(fit_intercept=True)
model.fit(x[:, np.newaxis], y)
xfit = np.linspace(1, 16)
yfit = model.predict(xfit[:, np.newaxis])
plt.title("Linear Regression of Recorded Zero-Day Vulnerabilities",
          fontweight='bold')

plt.scatter(x, y, color ='#2a9d8f')
plt.plot(xfit, yfit, color ='#e76f51');
plt.ylim(0.1)
plt.xticks([1, 2,3,4,5,6,7,8,9,10,11,12,13,14,15,16],
           ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
            'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec',
            'Jan', 'Feb', 'Mar', 'Apr'])
plt.ylabel("Number of Zero-Day Vulnerabilities", fontweight='bold')
plt.xlabel("Months (2020-2021)", fontweight='bold')

print("Model slope:    ", model.coef_[0])
print("Model intercept:", model.intercept_)
```

## E.3 Figure 4.4

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
#import libraries

# set width of bars
barWidth = 0.25
plt.figure(figsize=(10, 6))
# set heights of bars
browser = [1, 3, 3, 7, 1]
os = [0, 3, 7, 1, 2]
software = [1, 1, 7, 2, 9]

# Set position of bar on X axis
r1 = np.arange(len(browser))
r2 = [x + barWidth for x in r1]
r3 = [x + barWidth for x in r2]

# Make the plot
plt.bar(r1, browser, color='#2a9d8f', width=barWidth,
        edgecolor='#2a9d8f', label='Browser-Based')
plt.bar(r2, os, color='#f4a261', width=barWidth,
        edgecolor='#f4a261', label='OS-Based')
plt.bar(r3, software, color='#e76f51', width=barWidth,
        edgecolor='#e76f51', label='Software-Based')

# Add xticks on the middle of the group bars
```

```
plt.ylabel('Number of Zero-Days', fontweight='bold')
plt.xlabel('CVSS 3.0 Threat Score', fontweight='bold')
plt.xticks([r + barWidth for r in range(len(browser))],
            ['1-4.9', '5-6.9', '7-7.9', '8-8.9','9-10'])
plt.title('Comparing CVSS Threat Level to Target Vector',
          fontweight='bold')

# Create legend & Show graphic
plt.legend(title="Target Vector",loc=2, fontsize='medium',
           title_fontsize='large', fancybox=True)
plt.show()
```

## E.4 Figure 4.5

```
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
import numpy as np
from sklearn.linear_model import LinearRegression

plt.style.use('ggplot')
plt.figure(figsize=(10, 6))
x = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
              21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,
              38,39,40,41,42,43,44,45,46,47,48,49])
y = np.array([7.3,8.8,6.5,8.8,8.8,7.8,8.1,8.1,7.8,9.8,5.5,7.5,7.5,
              7.5,9.8,6.5,10,7.8,8.8,8.8,7.8,7.8,5.5,8.8,9.6,9.8,
              7.8,9.8,9.8,7,9.8,8.8,7.5,7.8,8.8,8.8,7.8,7.8,7.8,
              9.8,9.8,8.8,6.1,9.8,7.2,7.8,10,8.8,4.9])
plt.scatter(x, y);

model = LinearRegression(fit_intercept=True)
plt.ylim(3)

model.fit(x[:, np.newaxis], y)
xfit = np.linspace(0, 49)
yfit = model.predict(xfit[:, np.newaxis])
plt.title("Linear Regression of CVSS Threat Scores", fontweight='bold')

plt.scatter(x, y, color ='#2a9d8f')
plt.plot(xfit, yfit, color ='#e76f51')
plt.yticks([3,4,5,6,7,8,9,10])
plt.ylabel("CVSS 3.0 Threat Score", fontweight='bold')
plt.xlabel("Amount of Zero-Day CVEs", fontweight='bold')
```

## E.5 Figure 4.6

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
#import libraries

plt.style.use('ggplot')
A = np.array([2,0,1,4,0,0,0,1,1,2,6,1,5,4,7,6])
B = np.array([0,1,2,0,0,0,0,2,0,1,1,0,0,0,0,0])
C = np.array([0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0])
X = np.arange(16)

plt.figure(figsize=(10, 6))
plt.xticks(np.arange(min(X), max(X)+1, 1.0),
           ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
            'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec',
            'Jan', 'Feb', 'Mar', 'Apr'])
plt.bar(X, A, color="#2a9d8f", label='High')
plt.bar(X, B, color = '#f4a261', bottom = A, label ='Partial')
plt.bar(X, C, color = '#e76f51', bottom = A + B, label = 'Low')
plt.legend(title="Level of CIA Impacted",loc=2, fontsize='medium', title_fontsize='large', f
plt.title('Measuring CIA Impact of Discovered Zero-Days', fontweight='bold')
plt.ylabel('Number of Zero-Days', fontweight='bold')
plt.xlabel('Months (2020-2021)', fontweight='bold')
plt.show()
```

## E.6 Figure 4.7

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
#import libraries

plt.style.use('ggplot')

#=======Measuring Zero-Day Attack Period (ZDAP)============

A = np.array([37, 1, 7, 4])
X = np.arange(4)
plt.figure(figsize=(10, 6))
plt.xticks(np.arange(min(X), max(X)+1, 1.0),
           ['1 Day', '< Week', '> Week', 'Not Patched'])
plt.bar(X, A, color="#2a9d8f")
plt.title('Measuring Zero-Day Attack Period (ZDAP)', fontweight='bold')
plt.ylabel('Number of Recorded Zero-Days', fontweight='bold')
plt.xlabel('Time Until Patch Released', fontweight='bold')
plt.show()
```

## E.7 Figure 4.8

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
#import libraries
#=========Number of Discovered Zero-days by Target Vector=============

plt.style.use('ggplot')
A = np.array([2,1,0,2,0,0,0,1,0,0,4,0,0,2,2,1])
B = np.array([0,0,2,1,0,0,0,1,0,1,3,0,3,1,1,1])
C = np.array([0,0,1,1,0,0,0,2,1,2,0,1,2,1,5,4])
X = np.arange(16)

plt.figure(figsize=(10, 6))
plt.xticks(np.arange(min(X), max(X)+1, 1.0),
           ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
            'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec',
            'Jan', 'Feb', 'Mar', 'Apr'])
plt.bar(X, A, color="#2a9d8f", label='Browser-based')
plt.bar(X, B, color = '#f4a261', bottom = A, label ='OS-based')
plt.bar(X, C, color = '#e76f51', bottom = A + B, label = 'Software-based')
plt.legend(title="Target Vector",loc=2, fontsize='medium', title_fontsize='large', fancybox=
plt.title('Discovered Zero-Days by Target Vector', fontweight='bold')
plt.ylabel('Number of Zero-Days', fontweight='bold')
plt.xlabel('Months (2020-2021)', fontweight='bold')
plt.show()
```

## E.8 Figure 4.9

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
#import libraries

#declare global variables
colors = ['#2a9d8f','#f4a261','#e76f51']


#-------------------Target Vectors--------------------------
OS = ['Browser-Based', 'OS-Based', 'Software-Based']
sizes = [15, 14, 20]
fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=OS, colors=colors, autopct='%1.1f%%',
        shadow=False, startangle=90)
# Equal aspect ratio ensures that pie is drawn as a circle
ax1.axis('equal')
plt.tight_layout()
ax1.set_title("Target Vectors", fontweight='bold')
plt.show()

#------------Target Vector by browser---------------------------------

browsers = ['Chrome', 'Firefox', 'Internet Explorer']
```

```
sizes = [9, 3, 3]
explode = (0, 0, 0)
fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=browsers, colors=colors, autopct='%1.1f%%',
        shadow=False, startangle=90)
ax1.axis('equal')
plt.tight_layout()
ax1.set_title("Target Vector by Browser Type", fontweight='bold')
plt.show()

#---------------Target Vector by OS-----------------------

OS = ['Windows', 'Apple OS']
sizes = [7, 7]
fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=OS, colors=colors, autopct='%1.1f%%',
        shadow=False, startangle=90)
ax1.axis('equal')
ax1.set_title("Target Vector by OS Type", fontweight='bold')
plt.tight_layout()
plt.show()

#----------ACCESS VECTORS PIE ----------------------------------


OS = ['Network', 'Local',]
sizes = [35, 14,]
fig1, ax1 = plt.subplots()
ax1.pie(sizes, labels=OS, colors=colors, autopct='%1.1f%%',
        shadow=False, startangle=90)
ax1.axis('equal')
ax1.set_title("Access Vectors", fontweight='bold')
plt.tight_layout()
plt.show()
```