jamk.fi

SOAR Playbook Implementation - Incident Deduplication and Its Effects

Jani Purujoki

Bachelor's Thesis
November 2020
Technology
Information and communication technologies
Bachelor's Degree Pogramme in Information and Communications Technology

Jyväskylän ammattikorkeakoulu
JAMK University of Applied Sciences

jamk.fi

Description

Author(s)	Type of publication	Date
Purujoki, Jani	Bachelor's thesis	December 2020
		Language of publication: English
	Number of pages: 45	Permission for web publication: x

Title of publication

SOAR Playbook Implementation - Incident Deduplication and Its Effects

Degree programme

Information and Communications Technology

Supervisor(s)

Rantonen Mika, Nevala Jarmo

Assigned by

Nixu Oyj

Abstract

The thesis was assigned by Nixu Oyj. The objective was to implement a playbook for Security Orchestration, Automation, and Response (SOAR) platform. The playbook aimed to identify and process security incidents, i.e. cases that were potentially duplicates. The effects of the playbook on alert volumes generated within Security Operations Center (SOC) was observed by analyzing how the potentially duplicate cases were processed after they were identified. The goal was to reduce alert volumes within SOC in the future by implementing a SOAR playbook to identify duplicate cases and process them through the deduplication workflow.

The theory section aims to introduce SOC, its operations and tools in which SOAR is included. The implementation section covers how the workflow for deduplication process was designed and implemented to the SOAR playbook and how potentially duplicate cases were detected with the SOAR automation script.

The results showed that the ratio of potentially duplicate cases detected by the SOAR automation script to overall number of cases was significantly high. Only very small portion of potentially duplicate cases turned out to be real duplicates. The SOAR automation script that was working very poorly led to an alternative way to process the cases, which corrupted the data used to analyze the effects. However, it was clear that by looking at the data that the positive effects were nonexistent.

It was concluded that the SOAR playbook processed potentially duplicate cases well but the detection logic needed future development in order to enable fully automated deduplication process. The first version of the implementation also resulted in modifications in the SOAR playbook to minimize the manual work required.

Keywords/tags (subjects)

SOC, SOAR, security automation, incident deduplication

Miscellaneous (Confidential information)

jamk.fi

Kuvailulehti

Tekijä(t)	Julkaisun laji	Päivämäärä
Purujoki, Jani	Opinnäytetyö, AMK	Joulukuu 2020
	Sivumäärä 45	Julkaisun kieli Englanti
		Verkkojulkaisulupa myönnetty: x

Työn nimi

SOAR pelikirjan toteutus – tapahtumien deduplikointi ja sen vaikutukset

Tutkinto-ohjelma

Insinööri (AMK), tieto- ja viestintätekniikka

Työn ohjaaja(t)

Mika Rantonen, Jarmo Nevala

Toimeksiantaja(t)

Nixu Oyj

Tiivistelmä

Opinnäytetyön toimeksiantaja toimi Nixu Oyj. Opinnäytetyön tavoitteena oli suunnitella ja toteuttaa pelikirja Security Orchestration, Automation and Response (SOAR) -alustalle, joka tunnistaisi ja käsittelisi mahdollisia duplikaatteja tietoturvatapahtumia. SOAR pelikirjan vaikutusta Security Operations Center:n (SOC) hälytysmääriin havainnoitiin tutkimalla, kuinka mahdollisia duplikaatteja tietoturvatapahtumia käsiteltiin tunnistamisen jälkeen. SOC:n hälytysmääriä haluttiin vähentää tunnistamalla duplikaatteja tietoturvatapahtumia ja luomalla prosessi niiden käsittelyyn käyttäen SOAR -alustaa.

Teoriaosuudessa pyrittiin kuvaamaan itse SOC ja sen toimintaa sekä työkaluja, joihin myös SOAR kuuluu. Toteutusvaiheessa käytiin läpi, miten SOAR pelikirjan työnkulku suunniteltiin, kuinka tämä toteutettiin käytännössä pelikirjaan sekä käytiin läpi, miten mahdollisia duplikaatteja tunnistettiin käyttäen SOAR automaatioskriptiä.

Tuloksissa havaittiin, että SOAR automaatioskriptin tunnistamia mahdollisia duplikaatteja oli merkittävä osuus suhteessa kaikkeen tietoturvatapahtumiin. Mahdollisista duplikaateista kuitenkin paljastui vain hyvin pienen osuuden olevan oikeita duplikaatteja. Huonosti toimiva SOAR automaatioskripti puolestaan johti tietoturvatapahtumien käsittelyyn tavalla, joka vääristi tuloksiin tarkoitettua dataa. Datasta voitiin kuitenkin päätellä, että toteutuksen positiiviset vaikutukset olivat lähes olemattomat.

Lopputuloksena todettiin että pelikirja käsitteli mahdollisia duplikaatteja hyvin, mutta itse duplikaattejen tunnistaminen vaatii jatkokehitystä, jotta täysin automatisoidun deduplikointiprosessin toteuttaminen olisi mahdollista. Toteutuksen ensimmäinen versio johti myös korjaustoimenpiteisiin SOAR pelikirjassa, jotta manuaalisesti vaaditut työtehtävät olivisat minimaaliset.

Avainsanat (asiasanat)

SOC, SOAR, automatisointi, hälytysten deduplikointi

Muut tiedot (Salassa pidettävät liitteet)

Contents

Figures

Tables

Abbrevations

API	Application Programming Interface
CISO	Chief Information Security Office
CND	Computer Network Defense
DNS	Domain Name System
EDR	Endpoint Detection and Response
Email	Electronic mail
IDS	Intrusion Detection System
IP	Internet Protocol
IPFIX	Internet Protocol Flow Information Export
IPS	Intrusion Prevention System
IT	Information Technology
IoT	Internet of Things
JSON	JavaScript Object Notation
MSSP	Managed Security Services Provider
NIST	National Institute of Standards and Technology
QA	Quality Assurance
SIEM	Security Information and Event Management
SNMP	Simple Network Management Protocol
SOAR	Security Orchestration, Automation and Response
SOC	Security Operations Center
Syslog	System Logging Protocol
URL	Uniform Resource Locator

1 Introduction

1.1 Assigner

This thesis was assigned to the author by Nixu Oyj which was found in 1988. Nixu is a company that focuses on cybersecurity services and their shares are listed on Nasdaq Helsinki stock exchange. Nixu offers practical solutions for ensuring business continuity, and easy access to digital services and data protection for their customers. There are currently over 400 employees working for Nixu. Following cybersecurity related services are offered by Nixu: cloud transformation, security engineering, digital identity, cyber defense, cybersecurity exercises and training, Internet of Things (IoT), safety and reliability, compliance and certification, cybersecurity outsourcing and many more. (Nixu Corporation n.d.)

Nixu has been listed on the Nasdaq Helsinki stock exchange for past five years. Annual average growth rate during these years has been nearly 30% making Nixu the largest company specialized in cybersecurity services within the Nordic region. Rapid growth in 2019 resulted to revenue exceeding 50 million euros. Organic growth of 15% to 25% per annum, annual revenue of 100 million euros, 1000 employees, growth of managed continuous services to cover over 50% of revenue and expanding operations in existing and new markets to reach at least 25 million euro revenue are growth ambitions that Nixu has set for years 2020 to 2024. (Annual Review 2019.)

1.2 Purpose and objectives

Alert fatigue is a common problem for enterprises that practice security monitoring within the information technology (IT) environment. Large enterprises can generate thousands of alerts each day. The assumption is to catch any suspicious behavior from the environment by reviewing these alerts. Each time alert is generated it most likely requires a human analyst to verify whether there is a real threat or not. If the alert is not a real threat it is called false positive and on the opposite if the threat is real it is called true positive. As the alert counts raise and the false positive ratio is

commonly very high, this leads to the hard fact that actual threats get missed because alerts get ignored and analysts waste time on chasing the false positives leads. Checklist to reduce alert fatigue can contain multiple different concepts depending on the enterprise. This thesis focuses on implementation on how to detect and process duplicate alerts within the context of the client, and how the client was affected by the implementation.

Main objective of the research was to implement initial Security Orchestration, Automation and Response (SOAR) playbook designed to detect and process duplicate security incidents and study the effects of the playbook by different measures. Implementation involved defining of the process workflow for the duplicate detections and development of initial SOAR playbook for it, defining what is a duplicate security incident in the context of the assigner and modifiying of the automation script skeleton offered by the SOAR platform to detect duplicate security incidents based on the definition and the company needs. Two different workflows were defined for the process: one for quality assurance purposes and one for full automation that could be used in the future. The goal was to reduce the alert volumes in the future that the SOC team handles on daily basis by automating deduplication process.

1.3 Research questions and methods

This thesis reviews workflow and SOAR playbook development of the duplicate security incident detections step by step, what defines a duplicate and how the detection automation was molded for the company needs. The effects of the implementation are reviewed by analyzing how the potentially duplicate cases were processed with the playbook. The main questions that this thesis aims to answer is*: how to identify and process duplicate security incidents with a SOAR playbook and automation?*

The research method chosen for this thesis is quantitative. Quantitative research method is used to when the research is based on classifications, causal connections, comparisons and phenomenon's that are presented with numeric values. Numeric values are obtained by different measures that are meaningful for the research. These values are then analyzed with statistical methods. Statistical methods aim to

compress and explain the numeric values by using different variables. Finding and explaining the correlation of different values may be the result. In order to have trustworthy results, research material needs to be large enough. (Heikkilä 2014.)

The author chose quantitative research method to evaluate the effects of the implementation that was done as a part of this thesis. Quantitative research method seems appropriate since the effects of the implementation were tracked by collecting data on how each of the individual cases were processed with the implemented deduplication playbook. Effects could be categorized to smaller variables that were then used to analyze whether the implementation was effective or not.

2 Security Operations Center

2.1 What is a SOC

In the modern Computer Age, there are various cyber threats that target organizations of all sizes. In the worst-case scenario malicious actors breach into the organization and may cause huge losses of data and money. Cyberattacks, data breaches and malware infections have become so common that the most IT departments need to detect and mitigate these threats daily before they can cause any hazardous effects. Security Operations Center (SOC) provides continuous monitoring for the organizations with intention to enhance the security posture of the organization by analyzing and responding to threats that are detected as cybersecurity incidents. (Aher 2018.)

SOC is a facility where IT security team mainly works at. The security team focuses on organizations security posture by monitoring and analyzing the IT environment. IT systems are analyzed for flaws and threats using processes and tools mainly developed IT security in mind. Incident response teams are also working very closely with the SOC to take quick actions if needed. (Aher 2018.)

SOC team consists of multiple roles with different responsibilities. According to Kaspersky, SOC analysts that work closely with security incidents are divided to three different tiers:

- Tier 1 - Triage specialist
  - Incident registration and assignment.
  - Classification, verification, prioritization of security incidents.
  - Security sensors health monitoring (if applicable).
  - Collection data needed for Tier 2 analyst work.
- Tier 2 - Incident handler
  - Incident analysis and response.
  - Advice on containment and remediation actions.
  - Incident response coordination and support.
  - Tier 1 analyst work periodical review.
- Tier 3 - Security expert
  - Threat hunting.
  - Incident analysis and response (Tier 3).
  - Detection logic development and tuning.
  - Security monitoring system development.
  - Tier 2 analyst work review.

Other common core roles that SOC contains are Malware Analyst, Digital Forensics Analyst, Threat Intelligence Analyst, SOC System Admin, SOC Manager. Depending on the size of the SOC one person may be responsible of several roles. (Kaspersky for Security Operations Center 2019.)

IT departments can deal with the cyber security problems by managing their own SOC or by resorting to a Managed Security Services Provider (MSSP). Outsourcing SOC to an MSSP comes with multiple benefits. Paying for a service is more cost-effective instead of employing a whole department, investing in new hardware and software. MSSP focuses on the security aspect itself resulting in less downtime. (The Importance of Building a Security Operations Center n.d.)

2.2 How does a SOC work

Importance of human impact on preventing incidents has started to grow which can be seen when IT leaders are focusing on human impact instead of impacts of differ-

ent technologies. Existing and emerging threats are continuously studied and monitored by the members of a SOC team. Different kind of technologies can prevent basic attacks, but human analysis is almost always needed specifically on major incidents. (Aher 2018.)

SOC collects information with threat intelligence systems from external sources, relevant news feeds, incident reports, threat briefs and vulnerability reports. Collected information is correlated with the data received from organization to detect potentially malicious activities. SOC team is responsible for updating the threat intelligence data to the tools that perform the correlation with organizations data. (ibid.)

Security automations are leveraged by high-end SOCs making them more effective and efficient. Automations combined with security experts the ability to increase security measures, defend against security breaches and cyber-attacks are enhanced. SOAR platforms are one example of technologies that are used to implement security automations. (ibid.)

2.3 Tools

On daily basis SOC operation is heavily structured around various technologies that generate, collect, analyze, store or present huge amounts of data that is collected with Computer Network Defense (CND) in mind. Monitoring tools are placed around the network and systems that are under the scope assigned to the SOC. These tools collect raw contextual data to provide evidence of malicious or anomalous activity within the scope. Host systems and choke points in the network are common locations used to collect crucial data. By correlating collected data SOC is capable to find and analyze possible security incidents. (Zimmerman 2014.)

> *"The CND mission succeeds or fails by the SOC analysts' ability to collect and understand the right data at the right time in the right context (ibid., 32)."*

2.3.1 SIEM

Security Information and Event Management (SIEM) helps to aggregate data from across IT environment into centralized repository for further analysis and analytics. Collected data includes security information, logs, endpoint data and network data. This data can be correlated historically and in real time to identify anomalies, vulnerabilities and incidents. Mainly the focus lays on security related data for example login information, malware detections and escalation of privileges. SIEM also offers visualization and dashboarding for easier analyzing. Thus, making the tool very effective way for SOC's to efficiently respond to potential threats. (Nathans 2015.)

Architecture

Figure 1 illustrates how the SIEM architecture can be divided to different components that works as a pipeline and together form the SIEM tool (SIEM Architecture: Technology, Process and Data n.d.). Following components within the pipeline are defined in this thesis:

1. Event generation.
2. Event collection.
3. Normalization and enrichment.
4. Indexing, alerting and retention.

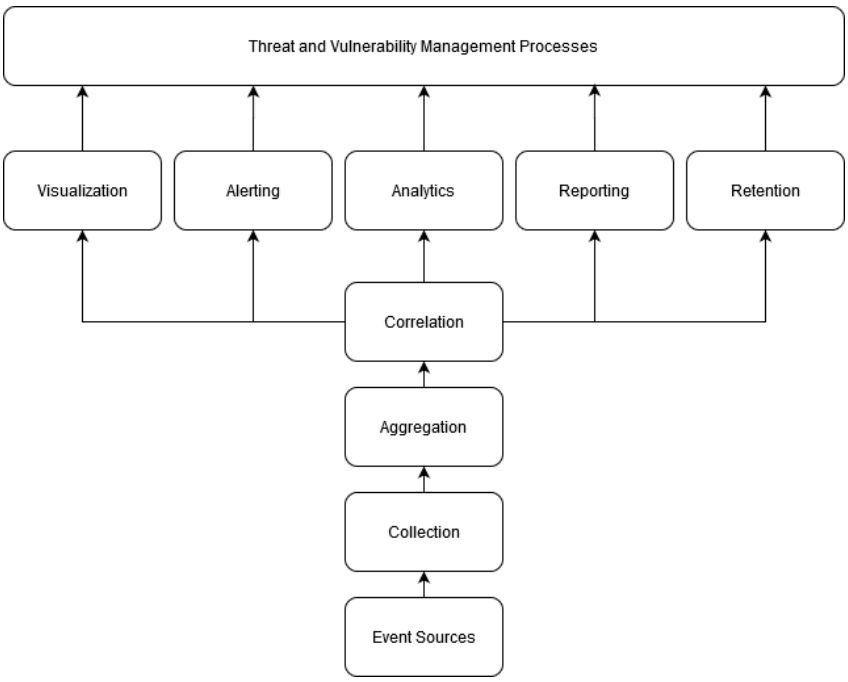

Figure 1. Common in-house SIEM architecture (SIEM Architecture: Technology, Process and Data 2020, adapted)

Event generation

Logs are the first part of SIEM pipeline. Events are collected and aggregated from large pool of data sources. Data can be collected from pretty much every device that has logging capabilities. Data sources can be allocated into high level categories: applications and devices, IT infrastructure, network logs and security events. (SIEM Architecture: Technology, Process and Data 2020.)

Event collection

Events are generated each time something happens in a data source. Collectors are used to connect to data sources and collect the events for SIEM. There are four types of collectors used to collect the data:

1. Agent data collectors that are placed to the data source.
2. Remote code connections to the device for example Application Programming Interface (API) calls.
3. Agent data collectors that directly access data source's log files in storage for example System Logging Protocol (Syslog).

4. Receivers that accepts events from data sources as event streams. Protocols like Simple Network Management Protocol (SNMP), NetFlow and Internet Protocol Flow Information Export (IPFIX) are commonly used. (Lane 2010.)

Normalization and enrichment

When events are collected from multiple data sources there are multiple types of events that contain even more different attributes. Commonly there are attributes like: time, user, operation, Internet Protocol (IP) address. Syslog for example groups the common attributes and provides extra information that does not fit the generic attribute template. Normalization collects only the attributes that are defined in the normalization process and rest is dropped from the normalized event log as illustrated in Figure 2. The original log events are kept because they may hold valuable information. Also, legal cases required full require full set of original records. (ibid.)

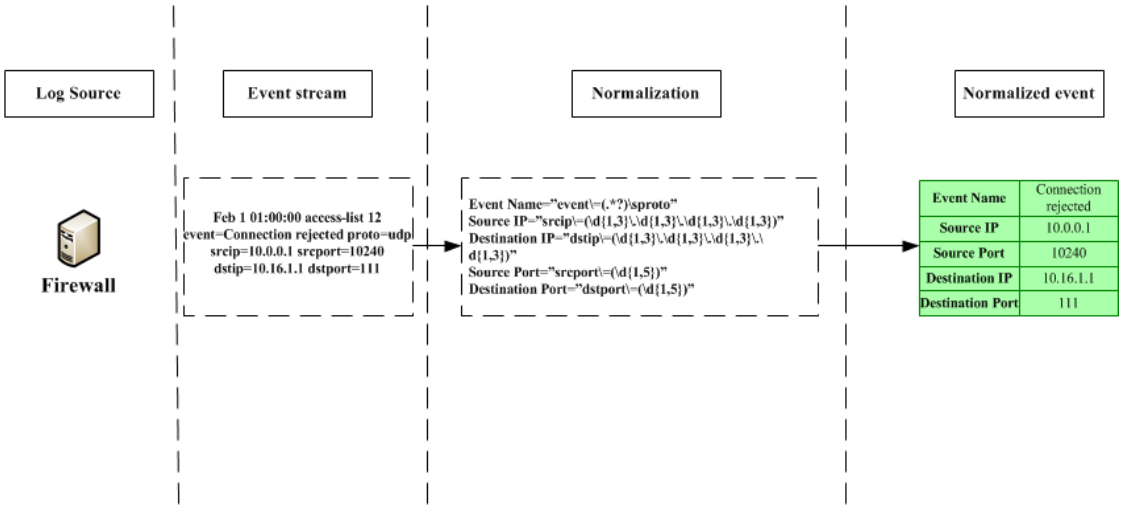

Figure 2. Event Normalization (Potapov n.d.)

Data enrichment adds more information for the attribute that is selected. Common example of data enrichment is to tie geo-location with a public IP address. This process adds additional and meaningful data thus making the original raw event more useful. (ibid.)

Indexing, retention and alerting

Purpose of indexing is to make searching of data more efficient in SIEM. Indexes are built for specific event attribute. When user initiates a search by using event attribute that is indexed, the search excludes data that is irrelevant for the search thus making the overall data volume smaller that needs to be processed. (Index Management n.d.)

Storing the vast number of logs is a vital part of the SIEM. Historical logs can be used for compliance, forensics and deep behavioral analysis by making historical queries. For example, User and Entity Behavior Analytics that is a process used to record normal behavior and alert on anomalous behavior would not be possible without retention. Figure 3 illustrates how an administrator login rule could be implemented. (Miller 2011.)

To trigger alerts the normalized events need to be processed through the rule engine. Rules may be fairly simple and straight forward or extremely complex. Common way to create rules is to determine if specific conditions are met based on alert type by using Boolean logic with normalized event attributes. (ibid.)

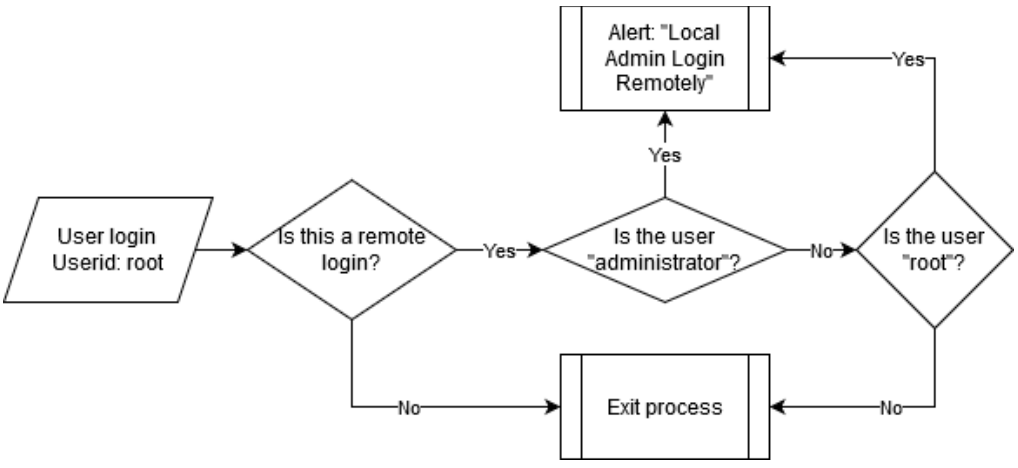

Figure 3. Administrator login rule example (Miller 2011, adapted)

Benefits

Before the age of SIEM tools monitoring IT environments were very tedious and time-consuming job. Primitive Instrusion Detection System (IDS) or file integrity checkers would generate alerts that had no context around them, and monitoring was performed by watching the computer screen with bare eyes. Electronic mail (email) generating systems were developed to make the job more manageable but all in vain. The vast amount of information was not manageable by email-based systems. Possibility to manage data from multiple types of devices and systems combined with rules around the data SIEM has made security monitoring a lot more efficient. (Miller 2011.)

2.3.2 IDS & IPS

IDS is designed to detect malicious behavior against all kinds of information systems. Network packets are analyzed passively for malicious behavior by using a network-based IDS (NIDS). Detected behavior may be attacks or unauthorized activity. Alerts are raised based on predefined activity or patterns that can be analyzed from network traffic. Unlike IDS, an Intrusion Prevention System (IPS) is designed to prevent detected malicious activity based on detected behavior in real time. (Miller 2011.)

2.3.3 Endpoint protection

Computer hardware devices, also known as endpoints, are protected with Endpoint Detection and Response (EDR) tools. EDR tools provide continuous monitoring for endpoints and focus primarily on detecting potentially malicious activity. EDR tools provide a platform that is used to monitor individual endpoints and servers for suspicious activities. Information like network events, configuration changes, process actions or file accesses are used determine whether the activity is suspicious or not. Meaning that the EDR tools are used to protect against malicious entities after the endpoint is already compromised. (Miller 2011.)

2.4 How do the false positives effect?

Afterall, tools do not provide way to the glory and richess when working with security incidents. Even if an alert is raised that does not neccesarily mean that something has been hacked or that the host raising the alert should be isolated. According to Zimmerman wheter or not something bad happened can be categorized as follows, based on what has actually happened:

1. True positives. System alerts on a threat that was an actual threat.
2. True negatives. System does not alert for activity that was not harmful.
3. False positives. System alerts when the threat is not real.
4. False negatives. System does not alert when something malicious happens. (Zimmerman 2014, 36.)

IDS systems often face the challenge to achieve a high true positive rate, even the systems are marketed to catch all the hacks. Security analyst tend to spend way too much time on analyzing the data regarding the alert because of too many false positives. In the worst case scenarios true positive alerts may be missed because of the noise generated by the false positives. Huge amounts of false positive data can lead to numbness towards true positive alerts which could lead to hazardous results when ignored. Figure 4 illustrates the activity categories and how the true positive alerts are minority mixed within the rest. (Zimmerman 2014, 37.)

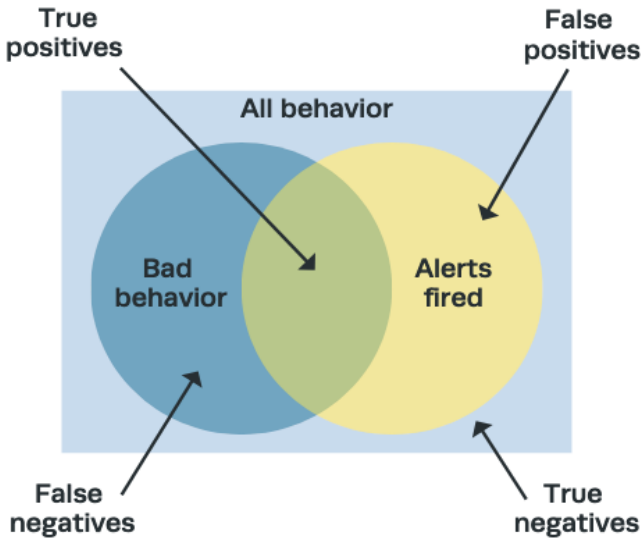

Figure 4. The Four Categories of Activity (Zimmerman 2014, 37).

IDS signatures may be also used to offer contextual data as false positive events instead of actual alerts. From millions of events collected by IDS sensor maybe few thousand may be valid for generating an alert. The rest of the events should not be considerated as false positivies. IDS may be configured to collect specifc log data when the SOC has no other way of getting the required data. For example missing Web proxy logs could be compensated by collecting Web requests with the IDS. One could argue that the term “false positive” is used only by incompetent security analyst who does not know how to read the data. As IDS systems do integrate with event priority but are lacking with concept of confidence, precision of the signature is a challenge. This challenge leads to continuous battle of balancing the data volumes with the value that it provides as illustrated in Figure 5. (ibid., 38.)

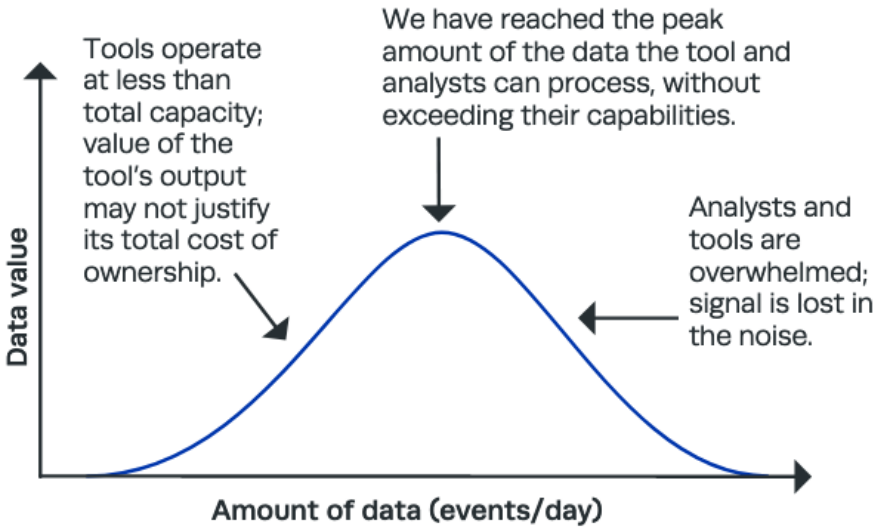

Figure 5. Balancing Data Volume with Value (Zimmerman 2014, 38).

Security analyst may use majority of their days analyzing security incidents that are false positive. Chasing red herrings can become mind-numbing and demoralising. These effect eventually lead to losing focus when dealing large amount of alerts (Heath n.d.). According to Ryan Tost (Tost n.d.) there are three different types of alert fatigues:

1. Too many alerts are overwhelming security analysts.
2. Same alerts are faced over and over again and are blindly closed.
3. When query for alert related data from log repository takes too long, alert may be disregarded.

Automated tools like SIEM and IDS can generate even if something suspicious happens. Each of the even slightly suspicious alerts are manually analyzed by the security analysts. However, automated tools tend to generated false positive alerts in great numbers. According to FireEye, most organizations generate up to 17 000 alerts in a week from which over half are false positives. Such remarkably high alert volumes generate the alert fatigue among security analysts. Alert fatigue may lead in the worst scenarios to situations where significant alerts drown in the mass of false positives. Ignoring even one significant alert may lead to organization wide security

breach. IDS can generate an alert based if singular event matches with specific pattern. For example, ransomware and and ZIP program both act similarly by read and writing multiple files in short period of time. Because of situations like these alerts are very likely to require more context around them. (Bates n.d.; The Numbers Game: How Many Alerts are too Many to Handle? N.d.)

3 Incident Management

3.1 Introduction

Event that violates organization's security policies and puts sensitive data at risk of exposure is known as a security incident. The term security incident itself holds various such as a data breach. Various events like malware infection, distributed denial of service attacks, unauthorized access, insider breaches, destructive attacks, unauthorized pivilegee escalation and loss or theft of equipment are considered as security incidents. Example of phishing and unauthorized access is when an employee within an organization receives a malicious email which leads to unauthorized access by adversary to the company network with compromised credentials. (Rich 2019.)

Companies implement a security incident management plan in order enchance their security posture by detecting security related events. Incident management plan helps to clarify all viable cybersecurity functions. When a security incident occurs and incidents and remediation needs to be rapid to prevent downtimes, security incident management kicks in. Threats or incidents are worked out in the real time by indentifying, managing, recording and analyzing. Incident response team is responsible of the first step when incident management kicks in. The incident response team investigates the potential threat by analyzing it to determine scope, evalute the damages and come out with a plan for mitigation. Security incident management plan guides the incident response team to act on threats in quick fashion by taking in account oher departments that may be working together with the technical teams. (RSI Security 2020.)

Security incident management plans can be considered more like as general guidelines on how to act in the different situations instead of rules that are written in the stone that would determine how the incident reponse process should be executed. After the threat is indentified, stakeholders are gathered together to tackle the task. The incident response team first aims to pinpoint the asset that the threat is based after initial investigation of the incident which includes analyzing how the incident is affecting the systems, data or user behavior. The issues may turn out to be a false positive detections in cases where software or hardware acts unexpectedly. If the issue turns out be a true positive generated by a cyber threat, all the data regarding the incident is collected and documented for further investigations so that the scope of the incident can be determined. After the scope is determined, mitigative actions are prepaired to resolve the threat. (ibid.)

Alone a plan on a paper does not resolve the cyber security threats that the organizations are facing. Plans are implemented so that actions could be performed rapidly and consistently from all corners of the organization. According to RSI security, there are five key elements that organizations should focus in order to effectively respond to cyber security threats with incident management process:

1. Incident identification.
2. Incident logging.
3. Investigation and diagnosis.
4. Assignment and escalation.
5. Resolution and closure. (ibid.)

The five key elements listed by RSI security (ibid.) are also present within the Digital Guardians blog post by Nate Lord regarding best practices for security incident management (Lord 2020a). According to Lord and various cyber security experts organizations should implement variety of best practices to develop a comprehensive security incident management plan in order to reduce recovery costs, potential liabilities and damage (Lord 2020b):

- Procedures like: how incidents are detected, reported, assessed and responded to should be included not just in the security incident management pland but also in the supporting policies.

- Incident response team should be established with clearly defined roles and responsibilities for each person. The team should not only be established from IT professionals. Other departments like legal, communications, finance and business management or operations should be also included at some degree to the team.
- Security incident management process should be tested consistently. This could be achieved by developing a comprehensive training program with test scenarios that reflects activities within security incident management procedures. Also refinements for the processes should be done based on the training.
- Adjustments should be made to the security program and incident management process. These adjustments can be defined by performing post-incident analysis from both successes and failures.
- Organizations should also be prepaired for situations when collecting data and analyzing forensics is necessary. Procedure to collect valid evidence should be implemented, especially for situations when the data needs to be accepted in the court of low. Some of the team members should be trained and experienced in forensics and functional techniques to analyze, report and investigate incidents with mindset towards the forensics. (ibid.)

3.2 Incident Response

Incident response process or its life cycle can be chopped down into multiple phases. Prepartion, detectiong and analysis, containment eradication and recovery and post-incident activity are the phases of incident response phases according to National Institute of Standards and Technology (NIST). Incident response process is a continuous process that cycles between the phases as illustrated in Figure 6. Example of continuous process cycle is when a threat is detected in singular host within an organization and the mitigation process is initiated. The detection process can be enchanced based on the analysis of the threat and more analysis can be applied by correlating the findings within the organization in order to make sure that the threat is not present on other hosts. After the incident is handled, the cause and costs of the incident should be noted and taken in consideration within the preparation phase. (Cichonski 2012, 12.)

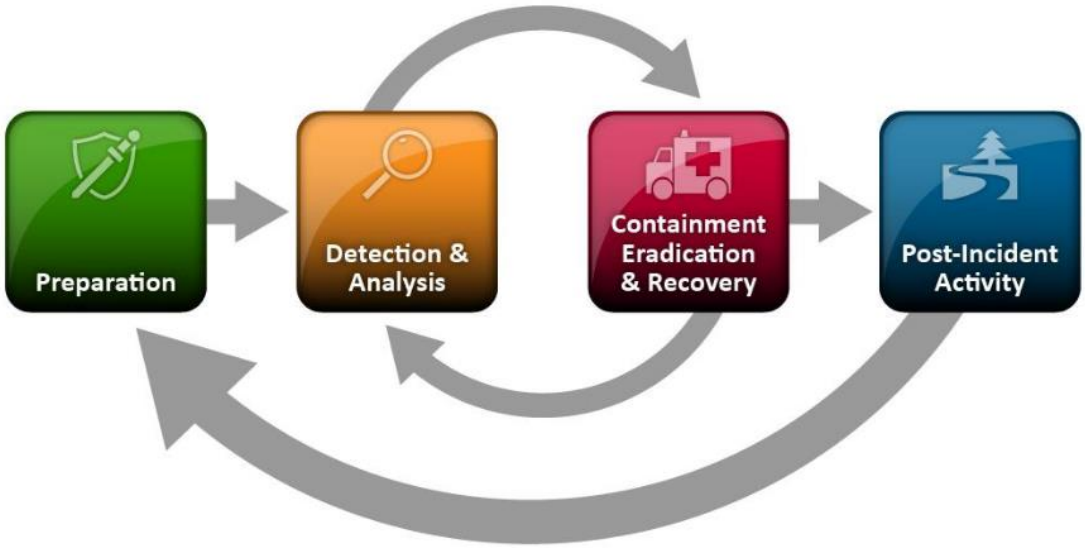

Figure 6. Incident Response Life Cycle

3.2.1 Preparation

Initial phase of incident response is preparation. It is important that the organization ensures that the incident response capabilities does not only consider response to the incidents but also preventing of the incidents is enabled in order to minimize incoming incidents. Commonly incident response team is not in charge of the prevention process but hardening of the IT environment is a fundamental part of successful incident response. (ibid., 21.)

Different kind of tools an resources may bring added value for incident handling process. Organizations should evalute what kind of tools and resources would bring added value for incident handlers. Tools and resources can be categorized to incident handler communications and facilities, incident analysis hardware and software, incident analysis resources, and incident mitigation software as follows:

- *Incident handler communications and facilities*: contact information, on-call information, incident reporting mechanisms, issue tracking system, smartphones, encryption software, war room and secure storage facility.
- *Incident analysis hardware and software*: separate digital forensics workstations and/or back up devices, laptops, spare workstations, severs and networking equipment, blank removeable media, portable printer, packet sniffers and protocol analyzers, digital forensics software, removable media, evidence gathering accessories.

- *Incident analysis resources*: port lists, documentation, network diagrams and lists of critical assests, current baselines, and cryptographic hashes.
- *Incident mitigation software*: access to images. (ibid., 21-23.)

3.2.2 Detection & Analysis

In order to steal information, data and money, adversaries use different kind of methods or pathways to accomplish the goal. These methods and pathways are called attack vectors in the cyber security field. Adversaries can use wide range of attack vectors which may lead to whole spectrum of different kind of incidents. It is not possible to develop runbook on how to handle all of the different kind of incidents. Instead, the most common attack vectors should taken in consideration within the organization. Some of the common attack vectors are for example removable medias, brute force and denial of services, website and web-based applications, emails, different kind of impersonations, improper usage of an authorized user, loss or theft of equipment. (DevOps Glossary n.d.; Cichonski 2012, 25-26.)

The hardest part of the incident response process is to detect and assess security incidents. Challenges originate from different kind of reasons. Incidents are generated from different kind of events which are collected either by automation or manually and they come in various levels of detail and fidelity. IDS and IPS systems, antivirus software and log analyzers generate incidents by automation. It is very common that automation based incident volumes are high. Example of manually generated incident can be when a user reports that their coworker has sent them suspicious email without their knowledge. In order to analyze the incidents, deeper technical knowledge is required with specialized understanding in cyber security standpoint. (Cichonski 2012, 26.)

If an event indicates that an incident could occur in the future it is called as precursor and on the opposite, if an event indicates that an incident has already occurred it is called as indicator. Precursors and indicators can be detected with multiple different ways:

- From computer security software generated alerts like IDS and IPS products, antivirus and spam softwares, file integrity checking softwares and third party monitoring services.
- Logs from operating systems, services, applications, network devices and network flows.
- Publicly available information on new vulnerabilities and exploits.
- People from within or other organizations. (ibid., 27-28.)

Only in the dream world all of the precursors and indicators are guaranteed to be accurate. Legitimacy of all of the indicators should be evaluated for this reason, making the detection and analysis process difficult. Cases where false positive indicators may occur are for example when the connection to a server is not working and user reports the issue or when an IDS generates a false positive alert. However it is important to consider all of the indicators even if they were not very accurate because you can not be sure wheter an incident has occurred or not. For example, when server crashes or ciritical files are modified could originate from various reasons but they could also be IOC's. In situations like these the legitimacy of detection should be determined. (ibid., 28.)

Incidents should be processed according to their priority instead of the order that they have occurred. This makes priorization one of the most critical decision points that incident handling process includes. In order to determine the priority of an incident, factories like functional impact, information impact, and recoverability from the incidents could be used. Incident that leads to data exfiltration of sensitive data is good example of an incident that could have serious impact on the organizations reputation. In situations like these recoverability is technically close to nonexistent if the data is posted publicly. In this case the incident should be approached by focusing on preventing similar incident in the future. (ibid., 33.)

3.2.3 Containment, Eradication & Recovery

As there are various types of incidents, containment strategies should be developed accordingly. Containment should be considered in the beginning of the incdent response process. Adversaries could spread within the organization by compromising systems from the initial foothold and increase the damage. Containment could be

carried out by shutting down a system, disconnecting it from a network or by disabeling certain functions. Predefined strategies and procedures help with the containment process and should be implemented preemptively by defining acceptable risks. (ibid., 35-36.)

Eradication could be performed by deleting malware, disabeling user accounts, and by identifying and mitigaiting vulnerabilities. Cycling back to the analysis phase is important during eradication in order to indentify all of the hosts related to the incident so that the threat can be remediated from entire organization. Recovery from the incident could be done by restorting systems from clean backups, rebuilding systems from scratch, replacing compromised files with clean versions, installing patches, changing passwords, and tightening network perimeter security. Because recovery phase can take long time, eradiction and recovery steps should be prioritized and performed in phases in order to remediate the most impactful threats that could lead to new incidents in the future. (ibid., 36-37.)

3.2.4 Post-Incident Activity

Learning and improving is also an important part of incident response process which can be skipped very easily. In order to enhance security measures and incident response process, “lessons learned” meetings should be organized with involved parties. Outcome of multiple incidents can be reviewed by reviewing what occurred, what was done, and how it worked out. Agenda of these meetings should be thought over before the meeting by collecting expectations and needs from all involved parties to fullfill the needs of everyone as best as possible. These meetings can also provide other benefits like training material, information on what should be updated regarding security policies and procedures, and handling similar incidents can be easier when follow-up reports are done properly. (ibid., 38-39.)

4 Security Orchestration, Automation and Response

4.1 Introduction

Main objective of SOAR is to combine three different security focused software capabilities together. The three capabilities: threat and vulnerability management (Orchestration), security operations automation (Automation) and security incident response (Response) were defined by Gartner who is also behind the term SOAR. Organizations often adopt SOAR for its capabilities to improve efficiency by determining the issues, defining the solutions and finally by automating the response for cyber security incidents. SOAR brings down the response times of threats and vulnerabilities by removing tasks previously handled by humans. (What is SOAR? Definition and Benefits n.d.)

At first glance SIEM and SOAR may seem like very similar products because of the capability to aggregate data from multiple sources. But SOAR platforms cover a lot more integration possibilities of internal and external applications. It is expected that SIEM vendors will be developing SOAR capabilities to their products in the future but for the time being SOAR is used to augment the SIEM software. (Rouse 2019.)

4.2 Orchestration

While the different security solutions are great arsenal for SOC, each of the solutions use different technologies and paradigms to develop, deploy and operate. These differences make it hard for SOC to integrate and work in conjunction. Security orchestration comes in play when technical and socio-technical security tool solutions of different vendors are merged in order to support SOC. Figure 7 illustrates how security orchestration improves security operations and management capabilities of an organization by enabling co-operation of people, practices and technologies. The best security orchestration platforms can automate different security tools, use playbooks that can contain complicated logic and track and orchestrate various tasks that are part of security analysts work. (Islam, Babar & Nepal 2020.)

Complexity of the overall incident response process is brought down by unifying different security solutions and processes, integrating to the security architecture of a company, connecting detection, network and endpoint security systems and by performing coordination of the security tools used by the company. Security analysts work becomes much more efficient and effective when activities can be merged from different security solutions. These combined activities are presented within a single console or platform which also removes operational silos. Insights of several security controls can be used to inform and educate the security analyst regarding threat behaviors and related support policies when actual human insight is needed. (ibid.)

Singular security solutions may be blind for a specific type of threat types. With orchestration, threat intelligence data is gathered from multiple sources to singular database. This data can be also gathered for example from blogposts which can be used to offer more precise context to the alerts related to them. Context may offer information for the security analysts that makes it a lot easier and faster to react to the threats. When an alert occurs, analyst needs to collect data manually through complex processes, investigate and plan for mitigation. Orchestration offers possibilities for streamlined workflow through automation for the alerts. Workflows require standardized process that contains planning of incident response, policy execution, investigation steps, response action and remediation process. It may also help if more data should be collected manually from the environment to analyze regarding the possible threat. Context from previous investigations can also be used as training material for security analysts. (ibid.)

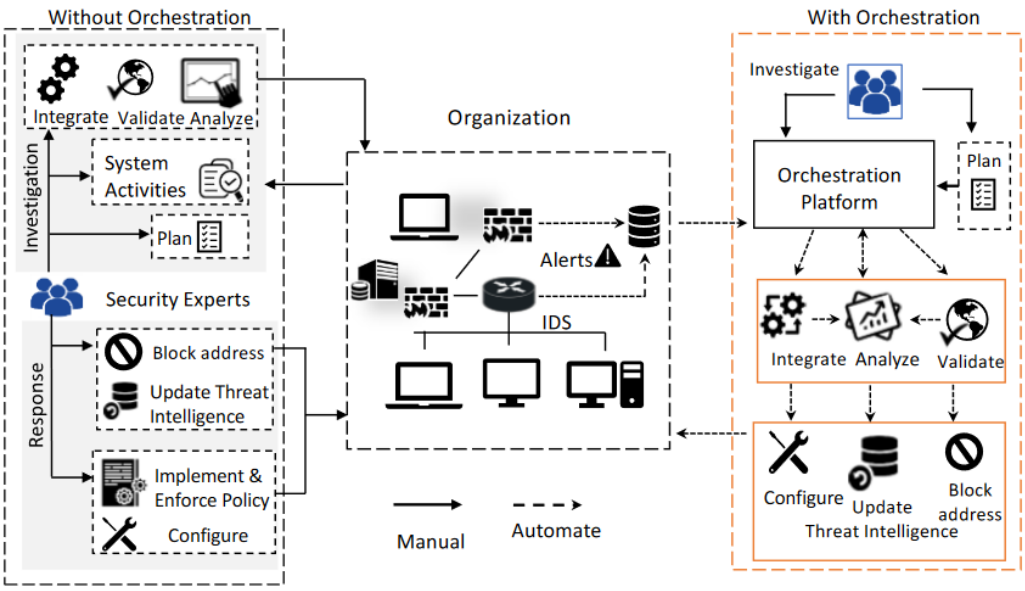

Figure 7. Overview of an organization decision against alerts without security orchestration and with security orchestration (Islam, Babar & Nepal 2020)

For example, a use case where potentially malicious email is received for analyzing. Security analysts go through several steps manually in order to determine if the email is malicious or not. Reputation and validity of the sender is correlated to threat intelligence data and origin of the email is checked using a Domain Name System (DNS) tool. Each of the Uniform Resource Locator's (URL) needs to be extracted from the email and analyzed for their validity. Also email attachments needs to be analyzed on secure environment. SOC may receive hundreds of emails daily to analyze and the manual investigation takes a lot of time. Security analysts make the decision based on the data collected from each investigation step. Orchestration can be used to perform these data collection steps using automation so that security analyst can make decision whether remediation is required. (Imam 2019.)

4.3 Automation

Sole purpose of security automation is to perform security related tasks without any human interaction. Automation can be applied on either side of computer security. Blue teams can leverage automation to prevent, detect and remediate threats. On

the other side red teams can apply automation to vulnerability assessments and to perform different kind of attack processes. Time of the security analysts can be used a lot of efficiently with automation so that they can focus on deeper analysis and develop proactive security measures. Core benefit of security automations is to release security analysts from time consuming tasks so that they become much more efficient in their work and they can focus on more interesting tasks. (Nanopoulos 2017.)

4.4 Response

When a security incident has occurred, the response functionality is applied in order to help security analysts to manage, collaborate and share data. SOAR performs alert triage and processing by collecting data related to the possible threat. Role of the security analyst is to perform analysis based on the data. If a threat is verified, deeper analysis is performed in case of other possible threats so that further attacks could be stopped. Security incident is resolved after executing the remediation process. Different modules are used with the security incidents so that communication and task management can be done within the SOC or outside of the SOC. Data that is related to the security incident can be collected and processed to threat intelligence so that proactive measures can be applied in the future. Different stakeholders like security analysts, the Chief Information Security Office (CISO), SOC managers and other security experts can take advantage of dashboard and reporting capabilities that SOAR offers. Well-designed reports enable further improvements. (Imam 2019.)

5 Implementation

5.1 Playbook

Workflow

First step of the SOAR playbook development for the deduplication process for security incidents also known as cases that were potentially duplicates was to sketch two

separate models for the workflow. Figure 8 illustrates the first model that was designed so that the final decision whether the case was a duplicate was done by security analyst manually. This enabled continuous Quality Assurance (QA) for the automation used to detect possible duplicate security incidents while used in the production environment. The second model was simplified version of the first model. All the manual actions required from the security analyst were stripped from the workflow in order to provide fully automated workflow for detected duplicate security incidents that could be used in the future.

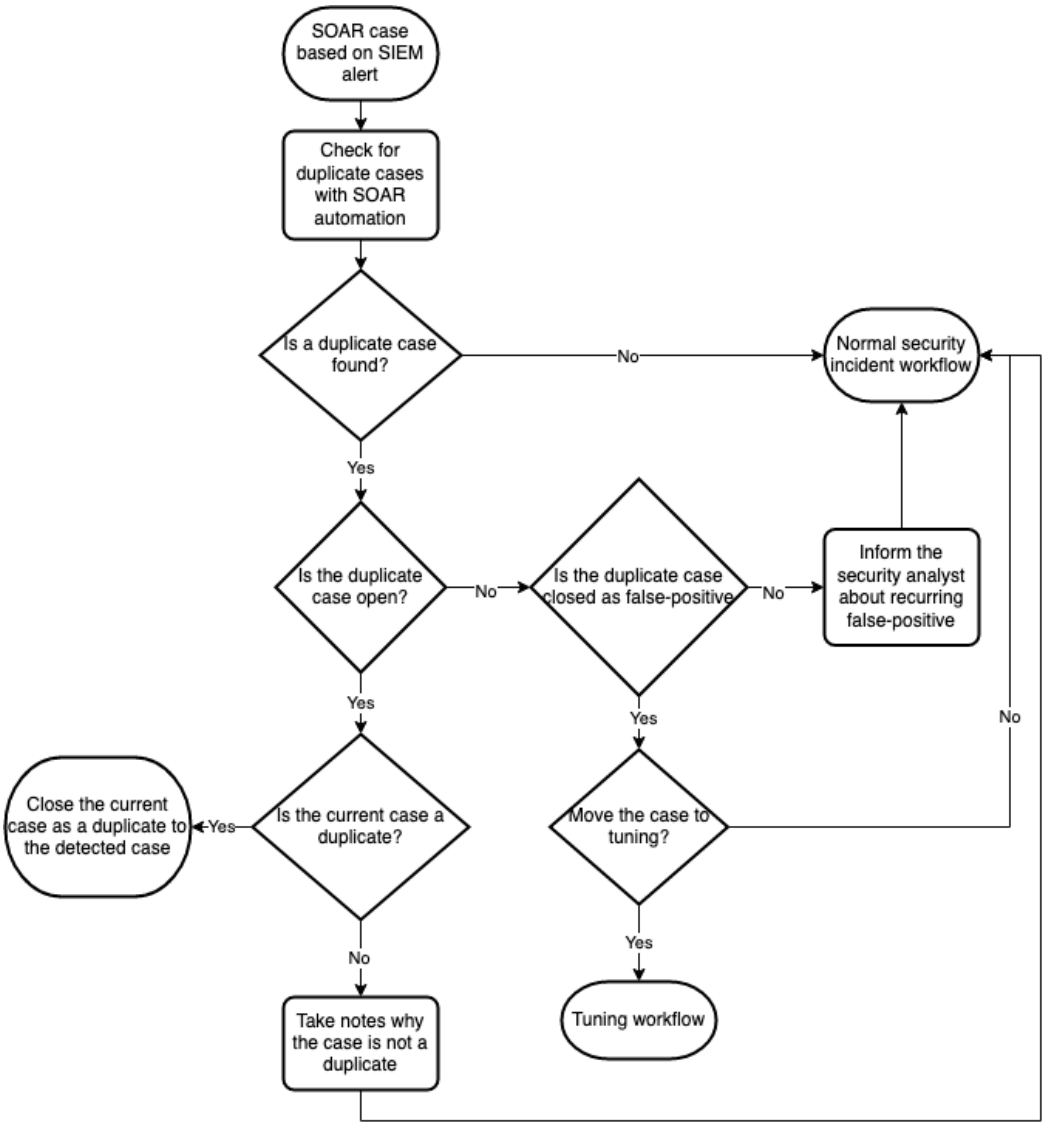

Figure 8. SOAR playbook process workflow during QA period

Technical implementation

Each alert generated by SIEM is pulled to the SOAR and generated to a case. Each case contains JavaScript Object Notation (JSON) object also known as context that is used to store data from integrations and automation scripts. The context is mainly used between different playbook tasks by outputting results to the context and reading inputs from it. The deduplication playbook requires specific key values as input from the context in order to process each case through the workflow. When a case enters the workflow, it is analyzed by using different key value pairs whether it is a duplicate with SOAR automation reviewed in chapter 5.2.

Based on the outcome of the SOAR automation, the current case will continue to general case workflow or duplicate handling is continued, illustrated in Figure 9.The deduplication playbook task validates from the SOAR automation output how to continue. There are three possible conditions that the validation task uses to determine how to continue: similar cases found, no similar cases found and error in automation. When there are no similar cases detected or any kind of error is occurred while running the automation script, the playbook is skipped, and the case is handled as general case. This design ensures that the case will not get stuck within the deduplication playbook because of any errors and the case will be presented to a security analyst so that the potential security incident is analyzed manually.

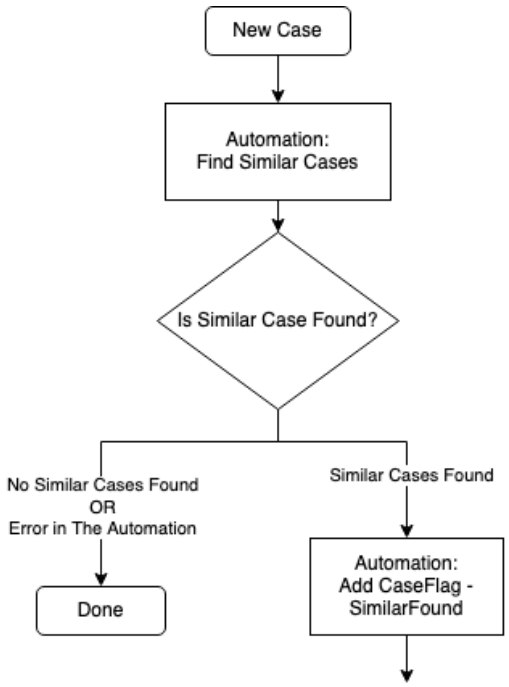

Figure 9. SOAR playbook task: Is Similar Case Found?

When one or more similar cases are found by the SOAR automation, the next task in the deduplication playbook is to validate status of the oldest similar case that has been detected. If any of the detected similar cases are active, security analyst will be prompted with information about the detection and asked to confirm if the current case is a duplicate. Confirmed duplicate case will be closed as a duplicate to the oldest detected active case. Closing the current case as a duplicate at this fashion seems a bit funny because by default security analysts are analyzing the cases from oldest to newest. But if the deduplication playbook would be running with full automation with each case, new duplicate cases would be closed as a duplicate to the same case thus providing more information for the security analyst, when analyzing the case. On the other hand, if security analyst decides that the current case is not a duplicate, a small comment is required in order to provide data before continuing with general case workflow so that tweaking of the SOAR automation will be easier in the future. Figure 10 illustrates the validation task presented to the security analyst when any of the detected duplicate cases are currently active.

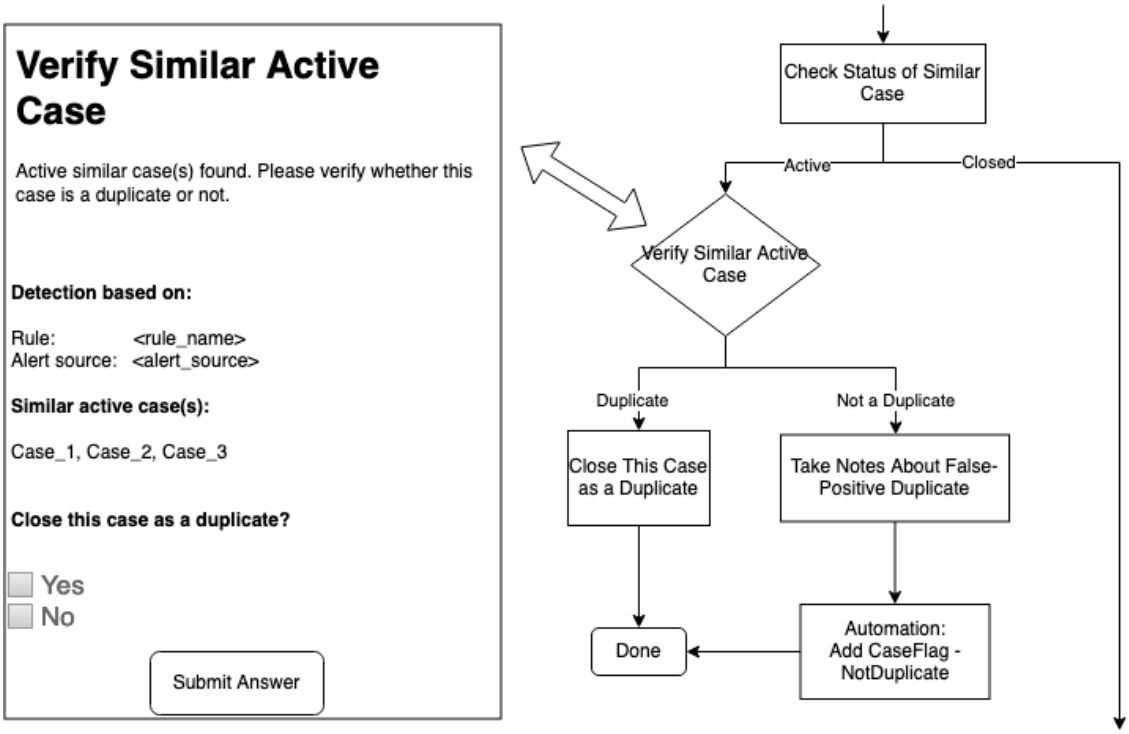

Figure 10. SOAR playbook task mockup: Verify Similar Active Case

When all the detected similar cases are closed, the deduplication playbook task fetches the closing reason of each closed case. If any of the detected similar cases are closed as security incident, the security analyst will be informed about the case by requiring them to submit “Acknowledged” before the general case workflow can be continued. If none the detected similar cases are closed as security incident, the security analyst will be prompted in the similar fashion as with active cases with information about the detection and asked to confirm whether the current case is a duplicate or not. Again, as with detected active similar cases, if the current case is not confirmed as a duplicate to the detected closed case, a small comment is required from the security analyst. But if the current case is confirmed as a duplicate to the closed case, the security analyst will be asked if the current case should be moved to the tuning queue or not. When the security analyst decides to move the current case to the tuning queue, the answer is added to the context and used in another playbooks validation task so that it is placed to the tuning queue. Figure 11 illustrates the validation task presented to the security analyst when any of the detected duplicate cases have been closed as security incident.

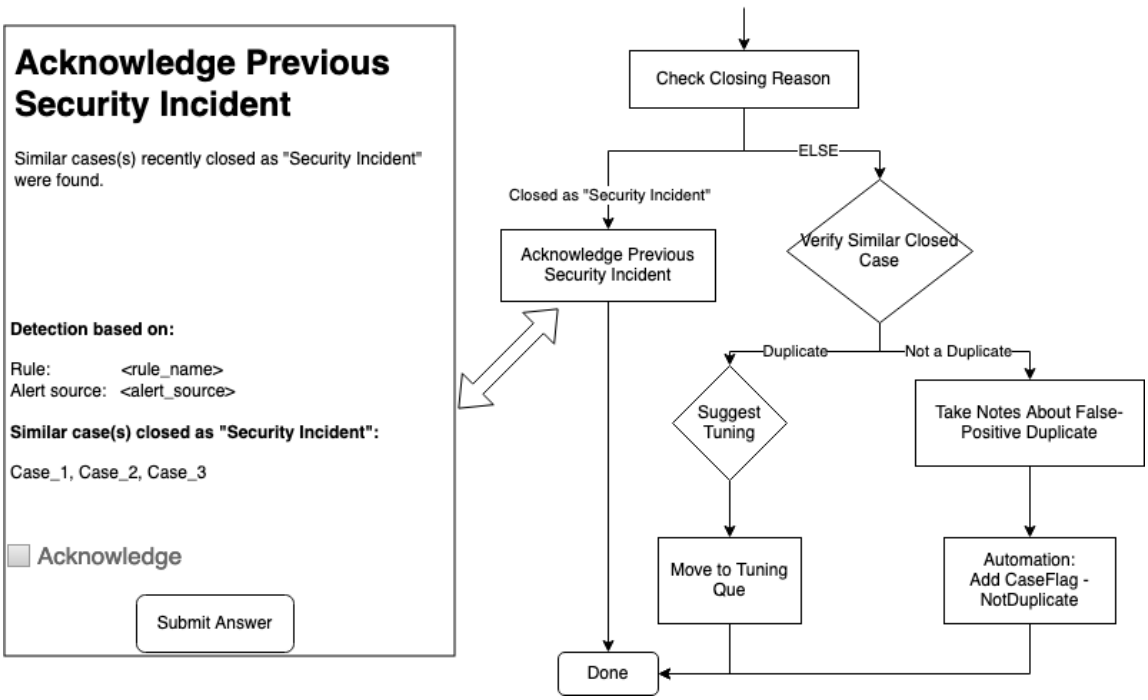

Figure 11. SOAR playbook task mockup: Acknowledge Previous Security Incident

Data collection

Separate SOAR automation script was implemented (not covered in this research) in order to add different case flags to track the cases that were processed through different playbooks. Case flags were appended to the context using automatic tasks within the playbooks. Data from potentially duplicate cases was collected by inserting descriptive case flags within the deduplication playbook. When the case was processed through workflow different case flags were added to the context of the current case. Case flags were appended to the context when a duplicate case was found, when the analyst decided that the case was not a duplicate and when the current case was moved to the tuning queue.

5.2 Automation

Sole purpose of the automations within the SOAR platform is to manipulate data in the system by combining multiple integrations together or by implementing tools

aimed to automate a singular task. Automations are scripts that contain common code that can be for instance written in Python or JavaScript. These scripts can be used to complete different kind of actions based on what the automation is aimed to do. The scripts can also contain commands associated with different integrations. All the APIs of the SOAR platform are accessible by the scripts.

The SOAR manufacturer had included a skeleton of an automation script that could be potentially used to detect possible duplicate cases from database that contained all the recorded cases. This was a rule-based script that used different kind of key-value pairs from the context to match exactly the same key-value pairs. Context of each case was built in form of a nested dictionary. Each of the nested dictionaries contained specific data related to different automation scripts and case related data that was fetched from SIEM. The most important data to look for duplicate incidents was within a dictionary key that was named as "incident". The "incident" key included data related to each separate case that was generated from SIEM alerts. Values from these keys combined with couple of different search related arguments were used as input values for arguments that the automation script required to work.

Duplicate handling was already considered within the SIEM. Rules for each alert were designed so that the rules would not raise multiple alerts based on the alert source but instead the offense related events would be collected under the same alert. Logic of some rules was not capable of this kind processing and they would occasionally raise multiple cases based on the same alert source even if there was an open case. Also alerts that were closed in the past could also trigger again if no tuning were performed or a case that was actual security incident would reoccur.

Based on the previously used logic with SIEM to index events based on "rule" and "alert source", same values were also used to detect potentially duplicate cases with SOAR automation. Automation needed to be run through the whole database to find potential duplicates from active and closed cases. Search time was adjusted to check cases from current day back to 744 hours which is equal to one month. It was also crucial to specify for the automation to use AND condition with the "rule" and "alert

source" values because the same alert source could be generating alert from a different rule. Final argument for the automation script was to skip the cases that were missing any values. This was turned on because when the automation would encounter a case that was missing any of the required values, an error would occur and break the automation for unknown reason. Methodology used to query for potentially duplicate cases with the SOAR automation was as follows:

- Cases that were created 744 hours before the current case were processed.
- Include all closed and active cases within the time range.
- Filter the cases with logic where both values for keys "rule" and "alert source" are identical.
- If values for keys "rule" or "alert source" are missing, ignore the case.

The original automation script collected various values from the detected similar cases and appended these to the context. How ever there was no categorization included for the detected cases, which was needed in order to act upon the detection as it was intended. Active, closed and cases closed as "Security Incident" needed to be sorted out from detected cases and the context updated accordingly. Python based automation script was modified in order to use the data with the deduplication playbook to process the potentially duplicate cases. Some minor tweaking was also done for the data that was stored to present relevant information for the security analyst when working on the case that is potentially a duplicate.

6 Results

The goal of the results part of the thesis was to determine what kind of impact the implemented SOAR playbook had in respect to alert volumes of assigner’s SOC. On higher level, points of interest were how many cases were detected as potentially duplicates and if they were duplicates or not. These parameters were used to dig deeper if there were any correlation, especially on potentially duplicate cases that were determined to be false positive duplicates. Finally, different outcomes were

categorized as “positive” or “negative” to reflect what kind of impact the SOAR play-book had on the total alert volumes within the 8 week time period that it was used in the production environment. Table 1 shows collage of how the potentially duplicate cases were processed during the time period.

Table 1. Ratio of total actions taken to process potentially duplicate cases to total case volume

Action	Impact	Percentage
Playbook skipped	Negative	14,02 %
Not a duplicate	Negative	10,41 %
Security incident	Negative	0,40 %
Moved to tuning	Positive	1,60 %
Closed as duplicate	Positive	0,53 %

Every time that the automation script detected potentially duplicate case, this was recorded. These detections did not consider anything else than whether there were similar cases or not. Overall, from total number of cases that were generated to SOAR, a little bit over on quarter were detected as potentially duplicates as shown in Figure 12.

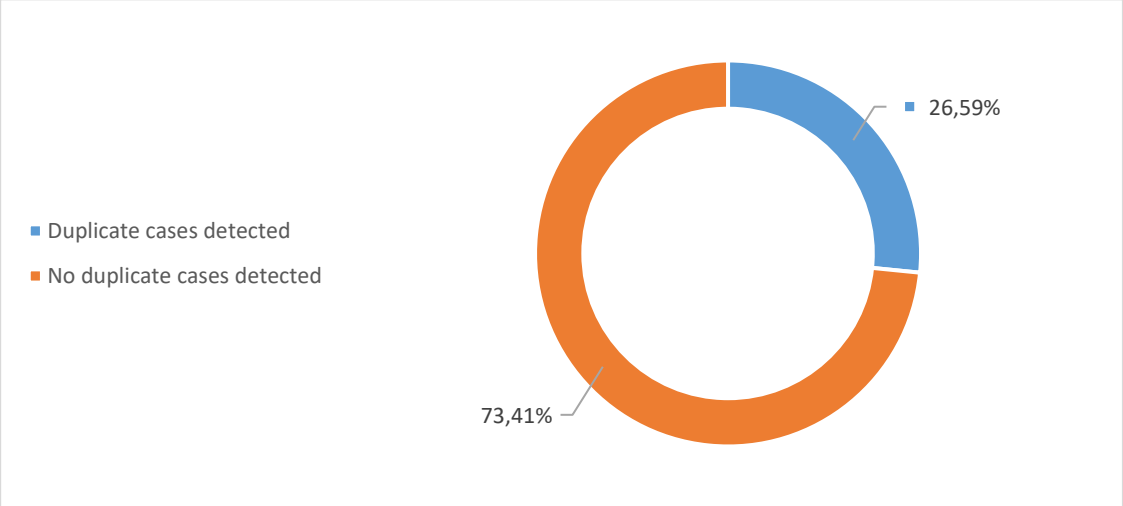

Figure 12. Total detections by the automation script

Potentially duplicate cases were divided into different significant groups based on how they were processed with the deduplication playbook. These groups were set to categorize potentially duplicate cases on high level in order to divide them to “positive” and “negative” groups. Figure 13 illustrates distribution of potentially duplicate cases.

Cases that were closed as duplicate or moved to tuning were counted towards positive impact. These cases theoretically either reduced the time required to investigate the individual case or by eliminating false positive case from generating recurring cases after moving it to the tuning queue. On the opposite, when the deduplication playbook was skipped or the case was not a duplicate, they were counted towards negative impact. These cases required additional work from security analysts on top of the regular analyst work for each case. Also, if the implementation would have been fully automated these cases could have led to hazardous consequences.

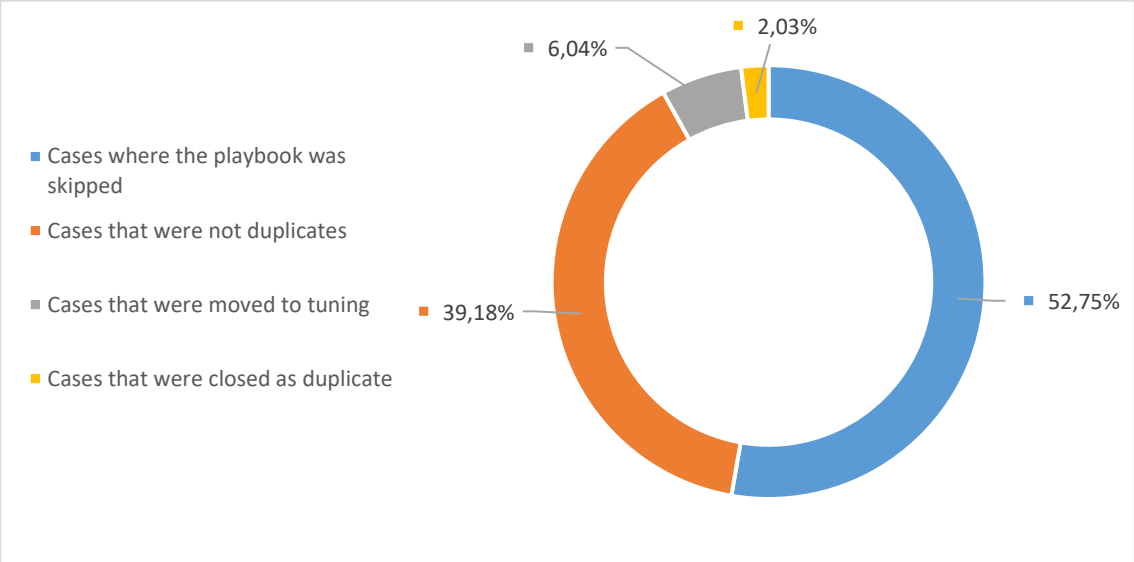

Figure 13. Potentially duplicate cases detected by the automation script

Positive impact was measured when the potentially duplicate cases were determined to be duplicate cases by closing the case as duplicate or by moving the case to the tuning queue. Figure 14 illustrates the positive impacts of the SOAR playbook compared to the total amount of unique cases.

When the case was closed as duplicate, security analyst who was making the decision did not need to spent time on analyzing the threat. Instead they were only required to spent small amount of time to determine if the threat was exactly the same as with the detected similar case. These cases impacted positively by bringing down the time needed to use on analyzing the case and by preventing situations where one security incident would be reported multiple times to the customer.

Cases that were moved to the tuning queue were detected as similar cases to previously closed cases. These closed cases were determined to be false positive cases that were recurring. Situation were the security analyst working on the case determines that the case is false positive could result to recurring cases when proper tuning is not done. These cases impacted positively by bringing down the time needed to use on analyzing the case and by preventing false positive cases from recurring.

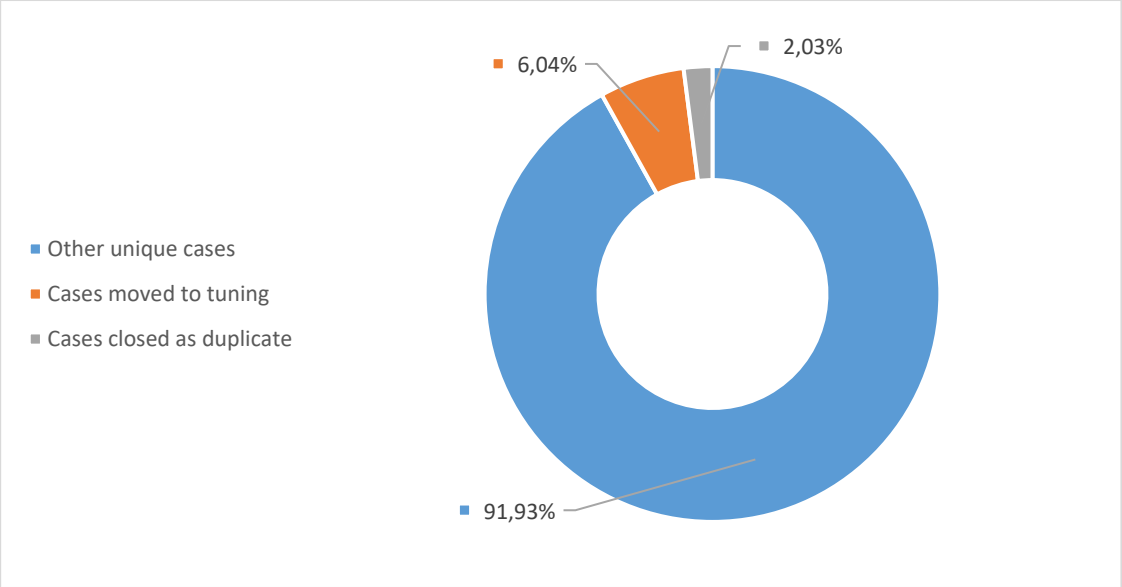

Figure 14. Positive impact of the implementation

Negative impact was measured when the security analyst working on the cases concluded that the potentially duplicate cases were not duplicates. These cases were divided to more specific groups to give better understanding of the negative impacts.

Figure 15 illustrates the negative impacts of the SOAR playbook in respect to total amount of unique cases.

Cases that were not duplicates and were processed through the playbook were analyzed by using two different groups. These groups aimed to differentiate wheter the similar cases were actual security incidents instead of duplicates. These cases had negative impact by requiring additional work from security analysts before they could start to analyze the current cases. Also, these cases were most important in respect to the possibility to fully automate the deduplication process. Especially cases that would be actual security incidents instead of duplicates would be hazardous for the security posture of the company in question.

As the SOAR platform allowed different ways for security analysts to process the cases, it was necessary to track if the cases were not processed with the deduplication playbook. In order to skip the deduplication playbook security analysts could only close the case manually or by executing a command thus indicating that the cases were not duplicates and that they were false positives.

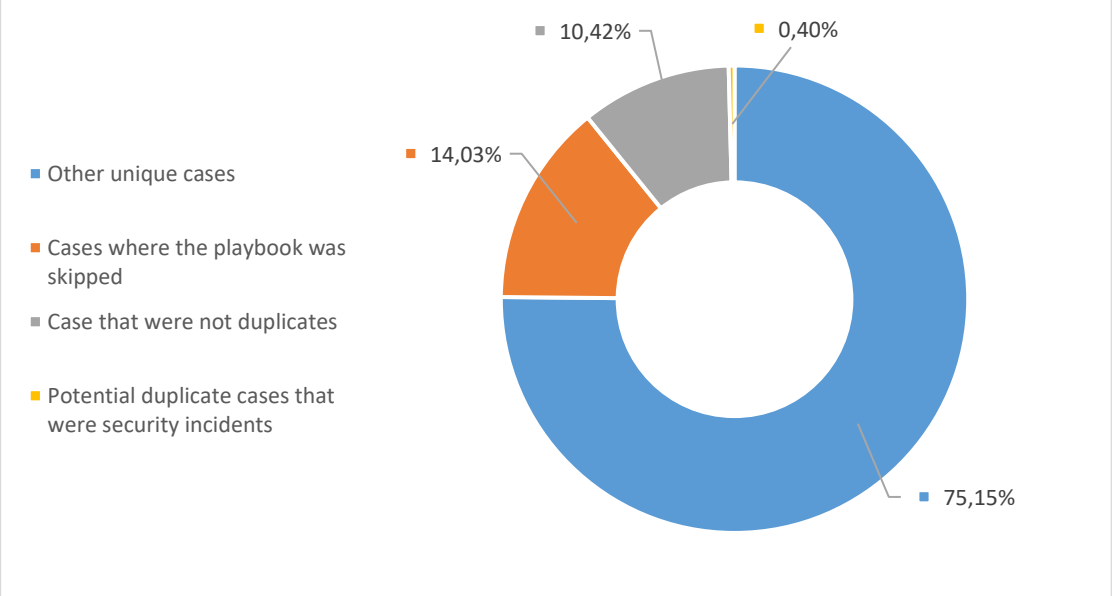

Figure 15. Negative impact of the implementation

7 Discussions

This thesis aimed to implement an initial SOAR playbook for deduplication process workflow, an automation script to detect potentially duplicate cases and study its effects. The goal of the implementation was to reduce alert volumes that the SOC team handles in future by detecting and processing duplicate cases automatically. The implementation included development of workflow for the deduplication process and applying it to the SOAR playbook and modifying a skeleton automation script to detect potentially duplicate cases. The goal of working implementation to reduced alert volumes was not met in any means. Alert volumes were not reduced, and security analysts faced even more work than previously. The main reason why the goal was not met was because the logic used with the automation script to detect similar cases was creating huge amounts of false positive detections. Increased number of tasks required from security analysts also led to frustration and skipping of the playbook thus making the collected data somewhat unreliable. However, it was clear that the implementation did not provide positive impact and the automation script needs to be enhanced in the future.

End result of the implementation was a functioning SOAR playbook that was embedded to the general workflow in production environment for 8 weeks. The deduplication playbook leveraged an automation script that checked whether the current case was potentially duplicate or not. The cases that were detected as potentially duplicates were processed through the workflow that was developed for deduplication use case which required decisions applied manually from security analysts. Manual steps were put in place so that the playbook would be going through continuous QA process. Initial workflow revealed to be somewhat irritating to work with because of the number of manual steps required from the security analyst in order to handle the case. This was noticed by looking at the ratio of how many times the playbook was skipped. The workflow was later reduced to a single manual step so that the false positive detections would not be so irritating to handle. Overall, the SOAR playbook worked as intended in terms of the process workflow. The manual steps that were required from the security analysts with QA in mind were effective since the amount ratio of negative effects was so high.

Detecting duplicate cases turned out to be a lot more challenging than was expected before starting this thesis. Rules that generated the cases to SOAR from SIEM were already designed to generate as little as possible ongoing duplicate offenses by proper indexing. The same logic was applied to the automation script so that the re-curring cases would be picked up and processed accordingly to prevent recurring cases. This logic did not work in practice with majority of the rules. The logic problem was solved in the second version of the implementation by specifying which rules are approved to be processed with the deduplication playbook. Effectiveness of the sec-ond implementation was not included in this thesis because of the time limitations.

Data that was gathered for results of this thesis, aimed to study effectiveness of the implementation was corrupt because the SOAR playbook was skipped significantly by the security analysts. However, from remaining data it was very obvious that the alert volumes and workload would not be decreased by the implementation. Skip-ping of the playbook and irritation of the manual steps required from the security an-alysts was revealed only after looking at the gathered data. Because of this the man-ual steps that were developed for continuous QA purposes turned out be both good and bad. False positive duplicates were not closed as duplicates because of decisions made by security analysts but making these decisions meant spending extra work to analyze each case which in the end lead to frustration and skipping of the playbook.

The implementation established a solid foundation for deduplication process within the SOAR platform for the assigner. Future development needs to address the prob-lem regarding the automation script that is used to identify potentially duplicate cases. This could be approached by various strategies:

- Going back to the SIEM rules and reconstruct the rules accordingly.
- Looking into more complex ways to identify duplicate cases from event related data.
- Digging into machine learning capabilities of the SOAR platform.

References

Aher, B. 2018. Importance of a Security Operations Center. Accessed 17.7.2020. https://dzone.com/articles/importance-of-security-operations-center

Annual Review. 2019. Nixu's Annual Report 2019. Accessed 19.11.2020. https://www.nixu.com/sites/default/files/NIXU_Annual_Review_2019.pdf

Bates, A., Chen, Z., Guo, S., Hassan, W., Jee, K. Li, Z. N.d. NoDoze: Combatting Threat Alert Fatigue with Automated Provenance Triage. Accessed 29.11.2020. https://par.nsf.gov/servlets/purl/1008566

Cichonski, P., Millar, T., Grance, T. & Scarfone, K. 2012. Computer SecurityIncident Handling Guide. National Institute of Standards and Technology (NIST). Accessed 25.11.2020. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-61r2.pdf

DevOps Glossary. N.d. Attack Vector Definition. Accessed 27.11.2020. https://www.sumologic.com/glossary/attack-vector/

Heath, M., Sayer, M. N.d. Is "alert fatigue" your biggest cyber threat?. Accessed 29.11.2020. https://www.accenture.com/au-en/blogs/anztrends/is-alert-fatigue-your-biggest-cyber-threat

Heikkilä, T. 2014. Kvantitatiivinen Tutkimus. Accessed 1.11.2020. http://www.tilastollinentutkimus.fi/1.TUTKIMUSTUKI/KvantitatiivinenTutkimus.pdf

Imam, F. 2019. Security Orchestration, Automation and Response (SOAR). Accessed 14.10.2020. https://resources.infosecinstitute.com/topic/security-orchestration-automation-and-response-soar/

Index Management. N.d. IBM QRadar SIEM administration document. Accessed 15.8.2020. https://www.ibm.com/support/knowledgecenter/SS42VS_7.3.2/com.ibm.qradar.doc/c_qradar_adm_index_mgmt.html

Islam, C., Babar, M. and Nepal, S. 2020. A Multi-Vocal Review of Security Orchestration. Accessed 10.10.2020. https://arxiv.org/ftp/arxiv/papers/2002/2002.09190.pdf

Kaspersky For Security Operations Center. 2019. Kaspersky introductory document. Accessed 20.7.2020. https://media.kaspersky.com/en/business-security/enterprise/brochure-soc-powered-by-kl-eng.pdf

Lane, A. 2010. Understanding and Selecting SIEM/LM: Aggregation, Normalization, And Enrichment. Accessed 11.8.2020. https://securosis.com/blog/understanding-and-selecting-siem-lm-aggregation-normalization-and-enrichmen

Lane, A. 2010. Understanding and Selecting SIEM/LM: Data Collection. Accessed 10.8.2020. https://securosis.com/blog/understanding-and-selecting-siem-lm-data-collection

Lord, N. 2020. Cyber Security Incident Response Planning: Expert Tips, Steps, Testing & More. Accessed 24.11.2020. https://digitalguardian.com/blog/incident-response-plan

Lord, N. 2020. What is Security Incident Management? The Cybersecurity Incident Management Process, Examples, Best Practices, and More. Accessed 24.11.2020. https://digitalguardian.com/blog/what-security-incident-management-cybersecurity-incident-management-process

Miller, D., Harris, S., Harper, A., Vandyke, S. & Blask, C. 2011. Security information and event management (SIEM) implementation. Accessed 15.9.2020. https://library.books24x7.com/

Nanopoulos, R. 2017. What Is Security Automation? Accessed 15.10.2020. https://www.rapid7.com/resources/wbw-security-automation/.

Nathans, D. 2015. Designing and Building a Security Operations Center. Accessed 7.10.2020. https://library.books24x7.com/

Nixu Corporation. N.d. About page on Nixu's website. Accessed 19.11.2020. https://www.nixu.com/about

Potapov, V. n.d. Event normalization in SIEM. Accessed 29.11.2020. https://vpotapov.wordpress.com/2017/02/13/event-normalization/

RSI Security. 2020. What Is Security Incident Management?. Accessed 24.11.2020. https://blog.rsisecurity.com/what-is-security-incident-management/

Rich, M. 2019. What Is the Difference Between a Security Incident and a Security Breach?. Accessed 25.11.2020. https://www.blackstratus.com/what-is-the-difference-between-a-security-incident-and-a-security-breach/

Rouse, M., 2019. SOAR (Security Orchestration, Automation and Response). Accessed 9.10.2020. https://searchsecurity.techtarget.com/definition/SOAR

SIEM Architecture: Technology, Process and Data. N.d. Exabeam SIEM guide. Accessed 10.8.2020. https://www.exabeam.com/siem-guide/siem-architecture/

The Importance of Building a Security Operations Center. N.d. Article by McAfee. Accessed 2.8.2020. https://www.mcafee.com/enterprise/en-us/security-awareness/operations/building-a-soc.html

The Numbers Game: How Many Alerts are too Many to Handle?. N.d. Report by FireEye. Accessed 29.11.2020. https://www.fireeye.com/offers/rpt-idc-the-numbers-game.html

What Is SOAR? Definition and Benefits. N.d. Article by FireEye. Accessed 8.10.2020. https://www.fireeye.com/products/helix/what-is-soar.html

Zimmerman, C. 2014. Ten Strategies of a World-Class Cybersecurity Operations Center. MITRE Corporation. Accessed 5.8.2020. https://www.mitre.org/sites/default/files/publications/pr-13-1028-mitre-10-strategies-cyber-ops-center.pdf