



SentinelOne Automation with FortiSOAR playbook

Martin Kaasik

Thesis, AMK

05/2022

JAMK University of Applied Sciences

ICT-engineer, Cyber Security

Kaasik, Martin

SentinelOne automatisointi FortiSOAR playbook avulla

Jyväskylä: Jyväskylän ammattikorkeakoulu. Toukokuu 2022, 27 sivua.

Tietojenkäsittely ja tietoliikenne. Tieto- ja viestintäteknikan tutkinto-ohjelma. Opinnäytetyö AMK

Julkaisun kieli: Englanti

Verkkojulkaisulupa myönnetty: kyllä

Tiivistelmä

Tämä opinnäytetyö tehtiin Loihde Trustin toimeksiannosta, ja sen tavoitteena oli luoda Security, Orchestration, Automation, Response (SOAR) -pelikirja avulla automaatio SentinelOne agenttien päivityksiä varten. Ennen automatisointia ratkaistavana ongelmana oli luoda prosessi, joka ei edellyttäisi kommunikointia asiakkaan kanssa että olisi lupa päivittää, ellei ongelmaa ilmene.

Tämän opinnäytetyön teoriaosuudessa esitellään, mikä on Security Operations Center (SOC), sen rakenne, kuvaus sen toiminnasta ja sen käyttämistä työkaluista. Työkalut, joihin keskitytään tarkemmin, ovat SOAR, SOAR playbook, SOAR connector ja SentinelOne, koska näitä työkaluja käytetään tämän opinnäytetyön tehtävien toteuttamiseen.

Toteutus käsittelee SOAR-pelikirjan luomista, joka päivittää SentinelOne-agentteja syötetyn tiedon perusteella. Käytettävät muuttujat luetellaan ja pelikirjan luominen selitetään vaihe vaiheelta. Luvussa 6 tarkistetaan pelikirjan toimivuus testiympäristössä olevilla väliaikaisilla työasemilla.

Johtopäätöksissä tarkastellaan automaation etuja ja haittoja sekä syitä, miksi automaatio on tärkeä tietoteknikassa. Johtopäätöksien jälkeen seuraa kommentti osuus jossa käsitellään henkilökohtaisia ajatuksia tämän opinnäytetyön tuloksista ja ilmenneistä ongelmista.

Avainsanat (asiasanat)

Automatisointi, SOAR, SOC

Muut tiedot (salassa pidettävät liitteet)

Kaasik Martin

SentinelOne automation with FortiSOAR playbook

Jyväskylä: JAMK University of Applied Sciences, May 2022, 27 pages.

Information and Communications. Degree program in Information and Communication Technology. Bachelor's thesis

Permission for web publication: yes

Language of publication: English

Abstract

This thesis was commissioned by Loihde Trust and the goal was to create an automation script for SentinelOne agent updates through Security, Orchestration, Automation, Response (SOAR) playbook. Problems to solve before automation was to create a process that would not require communication and get permission to update from the customer unless a problem arises.

In the theory section of this thesis, there will be an introduction to what is Security Operations Center (SOC), its structure, a description of what they do and about tools they use. The tools that are more focused on will be SOAR, SOAR playbook, SOAR connector and SentinelOne as these are the tools used to implement the task of this thesis.

Implementation Chapter 5 is about creating a SOAR playbook which will update SentinelOne agents according to variables that are used. Variables used are listed and the creation of the playbook is explained step by step. In Chapter 6 the playbook's functionality will be checked on temporary workstations in the test environment.

The conclusion section examines the advantages and disadvantages of automation, as well as the reasons why automation is important in Informational Technology (IT). The section is followed by a Discussion which is about personal thoughts about the results of this thesis and about the problems that occurred.

Keywords/tags (subjects)

Automation, SOAR, SOC

Miscellaneous (Confidential information)

Table of contents

Terminology.....	3
1 Introduction	4
1.1 Assignment.....	4
1.2 Research question.....	5
2 Security Operations Center	6
2.1 What does SOC do?.....	8
3 Tools and features	8
3.1 Security, Orchestration, Automation and Response	9
3.1.1 FortiSOAR Playbook	10
3.1.2 SOAR connector	11
3.2 SentinelOne.....	11
3.2.1 S1 agent	12
4 Process for updating S1 agents for customers	12
5 Playbook Creation and features	15
5.1 Customer process changes.....	20
6 Functionality testing	21
7 Conclusion	24
8 Discussion.....	25
Sources	27

Figures

Figure 1. SOC.....	7
Figure 2. SOAR.....	9
Figure 3. Visual mapping of process	12
Figure 4. Current S1 agent update process (Loihde sisäinen)	13
Figure 5. Automated version of S1 agent process	14
Figure 6. S1 timetables.....	14
Figure 7. Tenant conversion.....	17
Figure 8. Playbook standby step after Sending email.....	18
Figure 9. if/else step deciding if the playbook will continue	19
Figure 10. Playbook's finalized form.....	20
Figure 11. New update process	21
Figure 12. Playbook run prompt	22

Figure 13. Removing step progress in the playbook..... 22

Figure 14. Email notification list..... 23

Figure 15. No operation step 23

Figure 16. update confirmation 24

Tables

Table1. Comparing automation benefits and disadvantages 25

Terminology

AI	Artificial intelligence
DNS	Domain Name System
DoS	Denial of Service
EDR	Endpoint Detection and Response
EPP	Endpoint Protection Platform
HTML	HyperText Markup Language
IoT	Internet of Things
IP	Internet Protocol
IT	Informational Technology
PoC	Proof of Concept
S1	SentinelOne
SIEM	Security information and event management
SOAR	Security, Orchestration, Automation, Response
SOC	Security Operations Center

1 Introduction

As cyber threats are becoming more common and more vulnerabilities are found as time goes by, organizations' need for tools to protect against cyber threats is also growing. There is too much information coming in for humans to monitor without filtering unnecessary data. Automation and Artificial Intelligence (AI) is an assisting solution when defending from constant cyber threats. According to VIB (2018, 4), hiring, training and retaining new security analysts takes an average of 8 months. Despite this, a quarter of employees are likely to leave within 2 years. To combat that, Security, Orchestration, Automation, Response tools (SOAR) (more in depth explanation in chapter 3.1) is used to fill personnel gaps and make existing analysts' jobs easier and more fruitful.

Automation tools in cyber security assist with protection from known threats, attack types, malware and vulnerabilities while also helps with mitigation. When known methods of attacks are automatically mitigated, then there is more time for security specialists to investigate events that need more in-depth investigation. Because none of the systems are perfect, there are bound to be lots of false positive alerts which can be filtered and rule tuned with the help of automation tools.

In this thesis, there is a demonstration of how to automate service functions through SOAR playbooks which will be showcased by creating a SentinelOne (S1) agent update playbook with FortiSOAR playbook feature. The goal is to have a Proof of Concept (PoC) automated update for endpoints in the environment which could be later improved and implemented into production. As this is PoC, there will be much to improve and the variety of conditions that different customers might have a need to be taken into consideration in their environment and make exclusion rules accordingly. As this is done for Loihde Trust as part of the Security Operations Center (SOC) team, there will be an explanation of general information about SOC, its structure and the purpose of SOC as well as the tools used that will be used in this PoC build.

1.1 Assignment

This thesis is created for the company Loihde Trust. Loihde Trust, previously known as Viria Security, is a branch of Loihde group based in Finland which is a combination of both physical and digital security. According to Loihde Group, the company's statistics of 2020 were 106.8M in revenue, 6.8M profit and 714 personnel" (Loihde Group N.d).

Loihde Trust described in Loihde news article: “Loihde Trust is a corporate network that consists of Loihde Trust that focuses on protecting the physical and digital world, Loihde Trust Spellpoint that focuses on identity and access management and Loihde Trust Tansec that focuses on data transfer.” Loihde group consists of 4 subsidiaries Loihde Advisory, Loihde Analytics, Loihde Factor and Loihde Trust. (Loihde Trust 2021).

The current process for updating S1 agents is inefficient and requires a lot of time from the person in charge of S1. The problem lies in slow customer approval for updates and inconsistent update process between different customers. The current process relies heavily on the customer being in active communication with SOC, unfortunately that is not always the case.

The goal is to try and find a solution for the update process to reduce workload from human resources. The idea is to create an automated process of updating the S1 agents which requires minimal intervention from SOC and removes the need to manually contact customers. The solution for that would be an automated email being sent to customers saying that their environment’s S1 agents will be updated in 3 days unless they reply saying that we should not update the agents and then SOC would contact them to discuss the problem.

1.2 Research question

The research question for this thesis is about comparing the pros and cons of having tasks automated that could reduce the use of human resources. The goal is to see the benefits of automating tasks that should not require as much resource as it does manually, such as updating S1 agents which is used in this thesis. The biggest problem with the current process is that it’s heavily dependent on active communication between customers and SOC to schedule agent updates, which in itself is not a problem but often there is a lack of response from the customers which then uses up needless time from SOC trying to reach them. The goal of automation is to remove the need for action from SOC other than problems that occur with the update or if the customer wants to propose their own update schedule.

2 Security Operations Center

SOC is an organization that can range from small five person operations to large to national coordination centers (Zimmerman 2014, 21). SOC is an extra layer of security from cyber threats to companies by monitoring, analyzing, responding and preventing cybersecurity incidents. SOC uses various tools to monitor and protect customers' organization's environment from different types of cyber threats.

There are two types of SOC, one is internal SOC which means that the company deploys their own team to protect from cyber security attacks and the second is external SOC where a company buys SOC services from a third party to give them an extra layer of security from cyber threats where they pay SOC for service + monitoring tools (Nelson 2018, 4). SOC usually has 2 or 3 teams that are categorized as tier 1-3. Tier 1 is SOC analysts whose job is to analyze incoming alerts and escalate when suspicion of real a threat is found. Tier 2 is the management team whose job is to manage servers, Security information and event management (SIEM), services, configure compatibility of everything to work in the SOC environment and have meetings with customers to improve security in their environment. Tier 3 (in some SOC Tier 2) is the incident response team which handles security matters and in case of a breach, they investigate infected endpoints/servers, find out what has happened, assess the damage, report on their findings and find solutions for remediation. In figure 1 it is shown how SOC combines all various sources of information to protect against cyber threats.

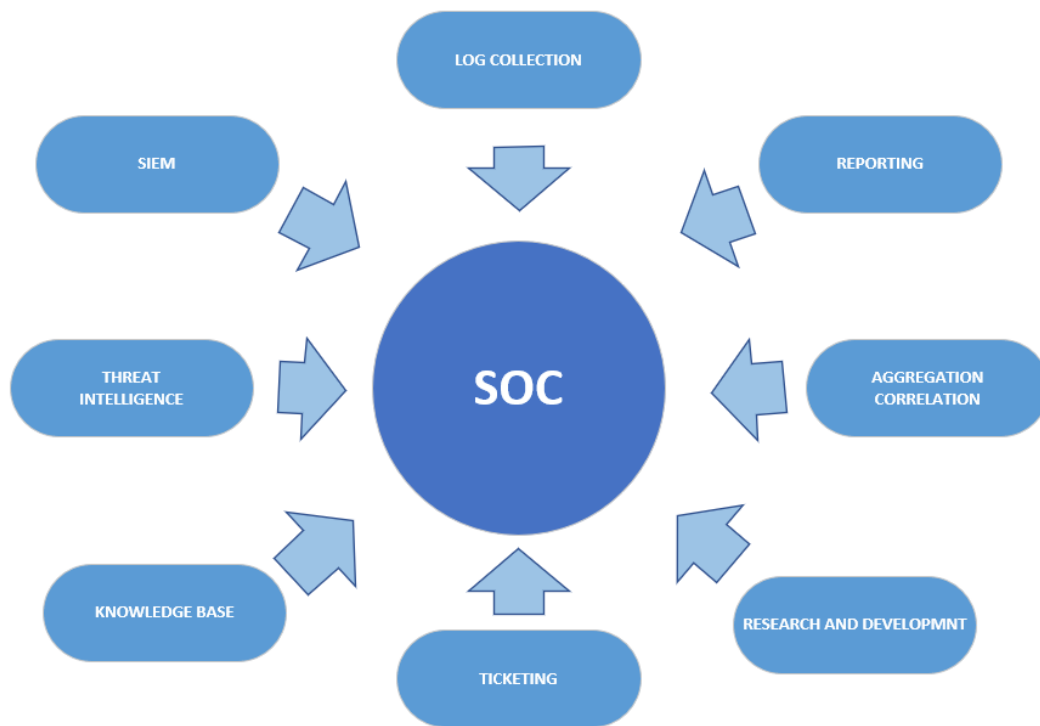


Figure 1. SOC

The most common tool for investigating alerts for SOC analysts is SIEM which either gets log and event data from the source or the source sends logs to SIEM (David, Shon, Allen, Stephen & Chris 2011, 254). SIEM then translates raw logs from different sources into a format that analysts can understand and generates alert events when certain conditions are met from log data. Data then is cross referenced with known attack types, malware and malicious IP addresses to give more information about the event and formats it into data that helps analysts in analyzing it.

It is more popular now to have SIEM and other protection tools send their data to SOAR as SOAR enriches the data from different protection tools into one place and can make SOC analyst's job easier through automation and response built in SOAR. All the incoming alerts from different sources are shown through the SOAR alert feed and when investigating an alert, all the important data is parsed into readable format.

2.1 What does SOC do?

SOC's job is to detect, analyze, respond, report and prevent cybersecurity events. Essentially it is SOC's responsibility to monitor and detect malicious or abnormal activity within the environment with visibility permitted to them and prevent harm to ensure that the company can focus on their day to day activities. When malicious activity is detected or more clarification is needed from the suspicious event, it is informed to the customer. If the event is true positive, then the customer either solves it themselves or asks SOC's help in investigating and mitigating the threat. After the threat is mitigated the discussion about how to prevent such events from happening in the future is conducted. The case of having the company's infrastructure breached will be more costly to the company than paying the SOC team for extra vigilance in their environment. SOC is an additional layer of security for companies so that the company can run without having to worry about being attacked by malicious hackers.

Common tasks for SOC teams according to (Nelson 2018, 4) are the following:

- Real-time monitoring and triage
- Cyber Intel collection and analysis
- Distribution, creation and fusion of services
- Trending, the long-term analysis of event feeds, collected malware, and incident data for evidence of malicious or anomalous activity
- Threat assessment, incident analysis/response coordination
- Countermeasure implementation including firewall blocks, DNS (Domain Name System) black holes, IP (Internet Protocol) blocks, path deployment and account deactivation
- Forensic artifact handling and analysis malware and implant analysis also known as malware reverse engineering

These tasks are separated into 3 different tiers in the SOC team. Tier 1 handles real time monitoring and analysis. Tier 2 is in charge of distribution, creation and fusion of services and tier 3 does threat assessment, incident analysis and forensic artifact handling.

3 Tools and features

In SOC there are various tools used to monitor and protect environments from external threats such as asset discovery, vulnerability assessment, behavioural monitoring, intrusion detection, and SIEM (Security operations center (SOC) tools, 2022). Because there is an abundance of different

monitoring tools, it is convenient to integrate them into one place. A common tool for that is SOAR.

3.1 Security, Orchestration, Automation and Response

SOAR is a tool to manage incoming security data from other analysis tools that you can integrate into SOAR, automate tasks and respond to threats showcased in figure 2. SOAR is used to automate the process of translating the contents of an alert, classifying the alert name, enriching data, classifying threat indicators, investigating and closing false positive alerts automatically, without any human interaction and escalation process.

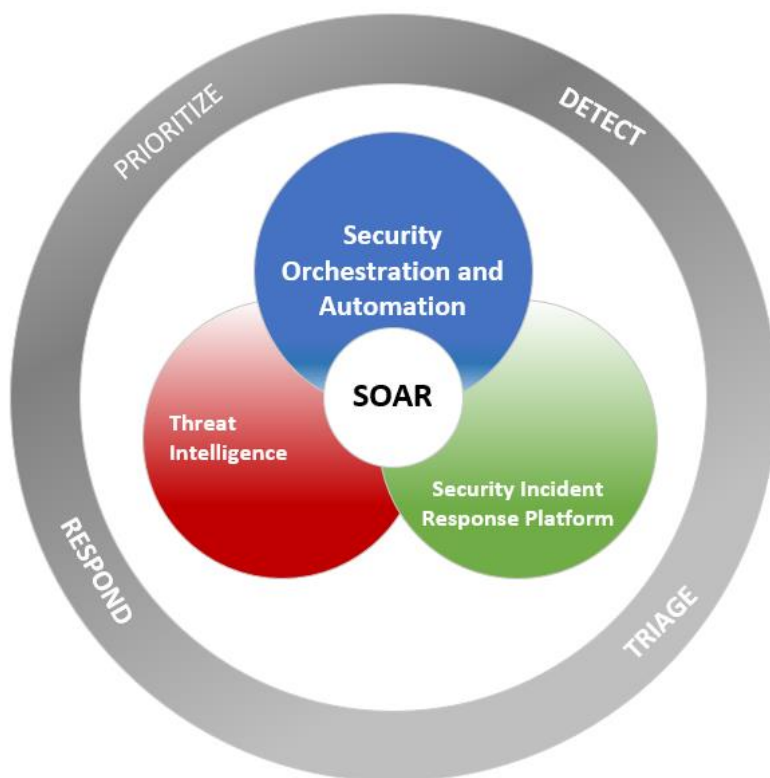


Figure 2. SOAR

Having data from different sources shown in one client (SOAR) makes it easier for the SOC analyst to analyze incoming alerts and saves them from swapping between different services checking for alerts. Alerts that come to SOAR can be filtered through playbooks to close events automatically that meet certain criteria, have been investigated and proven false positive. This lessens the

amount of confirmed false positive alerts popping up to the alerts feed and saves SOC analyst from spending unnecessary time analyzing same events which reduces alert fatigue.

Orchestration is the automated configuration, management, and coordination of computer systems, applications, and services. Orchestration helps Informational Technology (IT) to more easily manage complex tasks and workflows.

IT teams must manage many servers and applications but doing so manually isn't a scalable strategy. The more complex an IT system, the more complex managing all the moving parts can become. The need to combine multiple automated tasks and their configurations across groups of systems or machines increases. That's where orchestration can help. **(RedHat,2019)**

Automation in SOAR is used to automate tasks that would reduce human interaction needed and assist SOC with alert handling. In any industry it is preferred to automate repetitive tasks so that there is more time to deal with more meaningful tasks that require human intervention. In IT related industries automation is common and required for the most part. **(RedHat, 2018)**

In SOC environment SOAR Automation is done to lessen the manual process of identifying, investigating and remediating potential threats. "Repetitive, time-consuming tasks are filtered out for the security analysts when their systems are automated so that they can focus on greater value-adding work. It also eradicates human error, including inexperience, fatigue, and carelessness."(Mohammad & Surya, 2018, p.1). They are mentioning alert fatigue which loosely means that analysts are getting overwhelmed by a massive amount of repetitive false positive alerts which then makes them desensitized to those types of alerts and results in lacking investigation. Alert fatigue is something that every SOC analyst has to deal with as the amount of alerts incoming per day to be investigated is ever growing alongside the alerts that are being filtered as new vulnerabilities are being found.

3.1.1 FortiSOAR Playbook

Playbooks in FortiSOAR allow you to automate your security processes across external systems while respecting the business process required for your organization to function. Playbook templates can be customized to follow an organization's current procedures while leveraging the automation capabilities of FortiSOAR. Playbooks can leverage a number of different FortiSOAR capa-

bilities, such as inserting new data records, sending email notifications, and even referencing specified conditions to determine what path(s) to continue executing. Playbooks are highly configurable and provide consistent and thorough execution of IR response plans, enabling swift triage and containment of any potential cybersecurity threats.

The Playbook Engine runs asynchronously, meaning as an independent service, within the FortiSOAR application. This allows for better scalability and also frees the Application Engine to focus on request execution for better responsiveness to human users.

(Fortinet document library)

3.1.2 SOAR connector

FortiSOAR connector is a tool that allows SOAR to parse data variables from other services which are integrated into the SOAR environment so that automation is possible. SOAR connector is making it possible for services to communicate through SOAR. FortiSOAR has general connectors for services available to install from the FortiSOAR Connector Store that gets data and variables of a connector's service for automation use, in case there is not one, one can be created manually.

3.2 SentinelOne

S1 is an advanced Endpoint detection and response (EDR) tool that uses AI-powered threat detection and response. It combines EDR and endpoint protection platform (EPP) capabilities and operates across all aspects of a network, including endpoints, containers, cloud workloads and internet of things (IoT) devices. Its patented behavioral and static AI models provide powerful automation for identifying and blocking threats. S1 offers protection against executables, memory-only malware, exploits in documents, spear phishing emails, macros, drive-by downloads and other browser exploits, scripts such as powershell, and credential encroachments. **(Guercio, 2021)**

For the SOC, S1 is a useful tool that offers great protection to customer's endpoints through its agents. It has various features that make it easy for analysts to react, investigate and remediate incoming alerts. Opening an alert shows the visual path of a process from which the alert was generated, as shown in figure 3.



Figure 3. Visual mapping of process

If a serious threat is detected, it is possible to kill and quarantine processes through the S1 panel remotely and in known cases, S1 does it automatically. It's also possible to disconnect the endpoint from the network to stop malware from spreading into other endpoints or shut down the endpoint remotely in critical situations. There is also a feature to download a threat file which you can then analyze in the sandbox or use deep visibility option which can be used to check what actions have been performed in the endpoint.

3.2.1 S1 agent

S1 agent is a tool that will protect the endpoint once installed and will offer full visibility of the endpoint. S1 agent automatically remediates known threats, when an event is not remediated and alert gets generated, then SOC analyst can easily investigate the processes created by the suspicious source. S1 agents are being updated against up to date known malicious attack vectors and malware.

4 Process for updating S1 agents for customers

The biggest problem right now in the process of updating S1 agents for customers is that it relies heavily on active communication with the customers. Many customers are not actively working with SOC to get the best possible protection for their environment and in some cases, the responses are delayed for a long time. This makes it hard for SOC to work on the same customer and get their environment up to date smoothly. Because the communication is not active, SOC can't keep other customers waiting to improve their environments and need to handle multiple environments at the same time which complicates keeping track of old tickets.

As seen in figure 4, the constant need for permission is required in the process for SOC to continue onto the next step. The solution used in this PoC for this problem is an automated email being sent to the customer which will notify them that the update process will start in 3 days and if the customer doesn't respond to that notification then the update process will start.

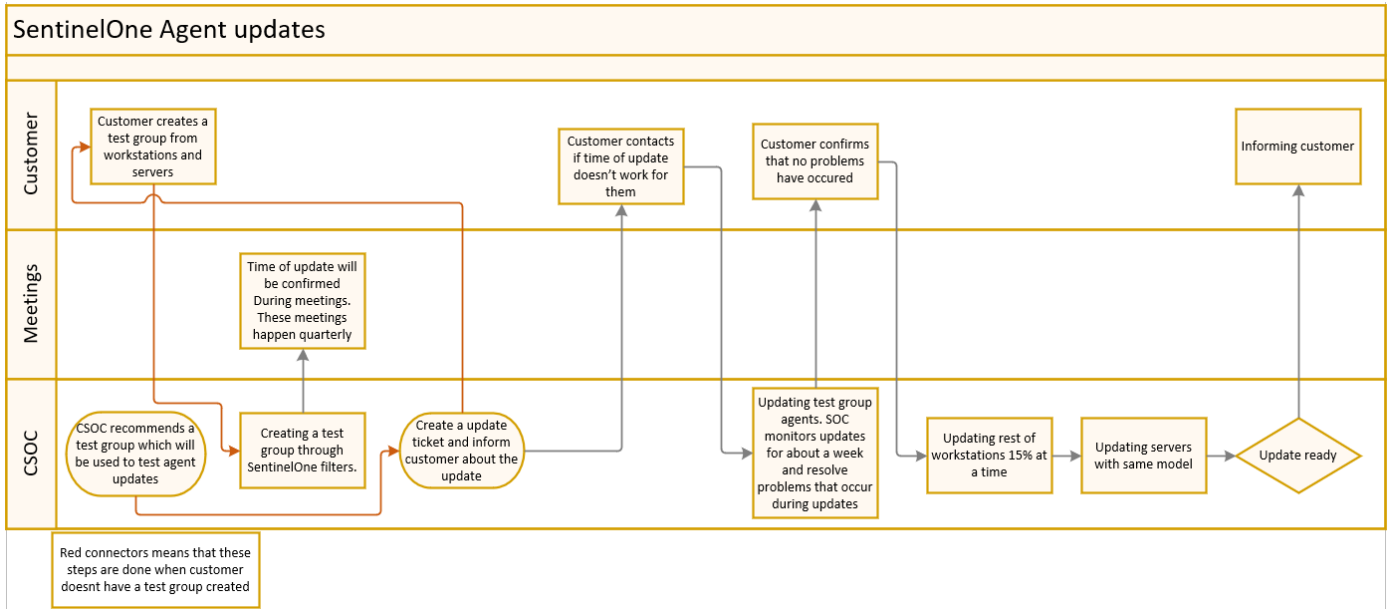


Figure 4. Current S1 agent update process (Loihde sisäinen)

In figure 5 you can see that an automated process would require much less human interaction and won't get hindered by lack of communication. The customer then would be responsible to contact us if they have any reason to not want updates to happen. This would save lots of time from both SOC and the customer while getting up to date protection to their environment.

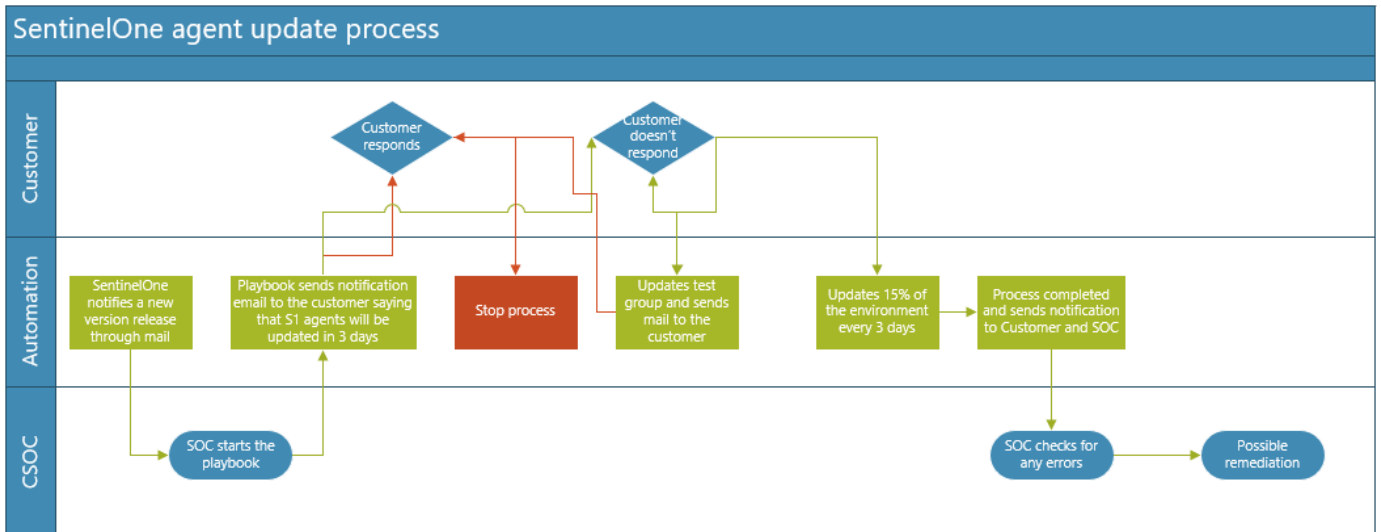


Figure 5. Automated version of S1 agent process

As a possible solution for having acceptable timeframes for the customer, S1 has a maintenance timeframe feature which allows to configure the time of the agent update seen in Figure 6.

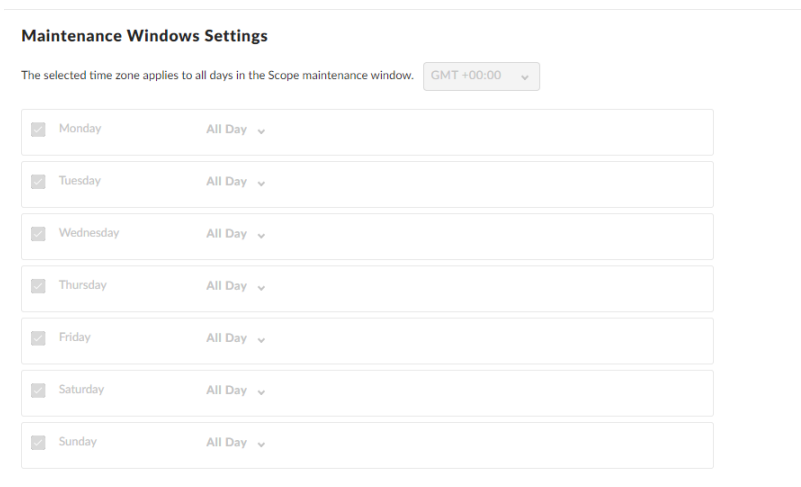


Figure 6. S1 timetables

This is intended for when updating agents on servers and databases since they are mostly online 24/7. This would not work for workstations if the update would be needed in the middle of the night, since the device needs to be online for the S1 agent to update. S1 agent update is not heavy on the system overall so this feature isn't used often but it could be useful when updating servers

or databases since conflicts might appear during updates and it would be safe to do it during the timeframe where the impact would be the lowest.

5 Playbook Creation and features

The PoC for this thesis will be created with the SOAR playbook through development SOAR environment which is a test environment that allows to create, modify and run test playbooks without any consequences. Playbook creation through FortiSOAR is visual which is easier to use than creating through pure coding and doesn't require as much technical knowledge. You can use connectors to get required variables and data from 3rd party sources. In this PoC custom modified S1 v3.0.0 connector is used for the playbook.

This project is going to be a first build that will potentially be the alpha version in the future. To start the playbook, there is a need for a start condition that will decide how the playbook will be executed and its trigger location. For this project, it is set to the "Tenant" page and can be executed through actions. This is done by naming the trigger button which in the playbook is "Update S1 Agents" and does not require a record input to run. There are also start parameters specified which will be asked for necessary info when ran, which playbook will use to update S1 agents with the following values (these will be used in later steps in this chapter):

- The first value is a required option "New Agent Version" which has a custom text field specified and requires the agent version to be inputted manually.
- The second value is "OS Type" which is created as a custom picklist and lists options for different OS types.
- The third value is "Tenant" which is a custom picklist and will have an option of choosing the Tenant name from which environment the endpoints will be found
- The fourth value is "Installer type" which is a custom picklist and includes an option to choose agent update package type.
- The fifth value is "S1 update package name" which is a custom text field and requires the S1 agent update package's name inputted manually.

The next step in the playbook is "Get old agents from tenant" to get the list of endpoints which are getting their S1 agent updated. For this step, there are custom S1 connector variables in use.

These are "Site ID" which is Tenant ID on the S1 side and will get its input from the Start step's start parameters "Tenant", "Agent Version NOT in" which will have a function of choosing all end-

points that have agent version other than specified and will get its value from start parameter “New Agent Version”, “Installer type” which defines the agent installer type (“.deb”, “.exe”, “.gz”, “.msi”, “.pkg” or “.rpm”) and will get its value from start parameter “Installer Type”, and OS type which defines if the endpoint is on Windows, Linux or MAC and will get its value from parameter “OS Type. The agent version field is already there from the default S1 connector.

Before using these values in steps, there are prerequisites needed for translation in the “Tenant” value. Because Tenants are defined with their Site ID in S1, it is hard to know what ID belongs to which Tenant. This problem is remediated it with creating a Set Variable step and create variables “tenant_name” which has value `{{vars.input.params.tenant.itemValue}}` this takes the value of Tenant ID from Start step and variable “tenant_convert_name” which translates the tenant ID into tenant name as follows: 'Customer1': 'xxxxxxxxxx','Customer2': 'xxxxxxxxxx'... (xxxxxx is filler for ID as there will be no real information disclosed). Then there is another step for translation which will be done with Code Snippet which is a custom python script step in the playbook. The python code will take the “Tenant_convert” step’s value and prints the list:

```
listtable = {{vars.tenant_convert_name}}
print(listtable[{{vars.tenant_name}}'
])
```

Now to the contents of step “Get old agents from tenant”. There is Site ID with value `{{vars.steps.convert.data['code_output']}}`, “Agent Version NOT in” with value `{{vars.input.params.newAgentVersion}}` and “OS Type” with value `{{vars.input.params.oSType.itemValue}}`. These are the values that are determined from when you execute the playbook and previous translation steps shown in figure 7.

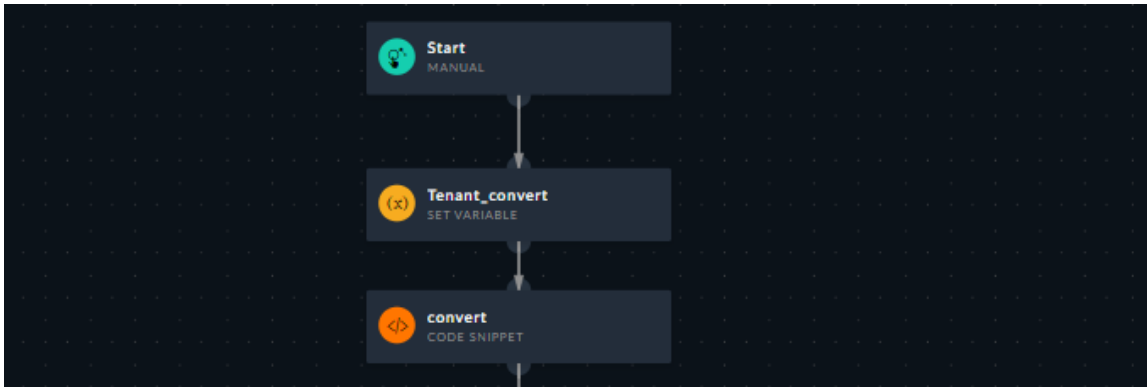


Figure 7. Tenant conversion

To use the output of the “Get old agents from tenant” there are 2 code snippet steps created with python code. The first one is storing the output into a list so it can be used in later stages of the playbook which is done with the following output in step “Agent UUID”:

```


    fetched_agents = {{vars.steps.Get_old_agents_from_tenant.data.data}}
    agents = []
    for i in range(len(fetched_agents)):
        agents.append(fetched_agents[i]["uuid"])
    print(agents)
  

```

Second code snippet will be used for sending informational email to the customers who’s S1 agents are getting updated. This will be done with following python scrip in step “Endpoints needing update”:

```


    fetched_agents = {{vars.steps.Get_old_agents_from_tenant.data.data}}
    for i in range(len(fetched_agents)):
        print(fetched_agents[i]["computerName"], end="<br>")
  

```

The result is printed with **end=
** because email uses HTML (HyperText Markup Language) and this will make it visually more appealing to read.

Now to send an email to the customer, there is an exchange connector which in this playbook is custom but works with also default connector. in this Exchange connector step “send email”, the fields defined are *subject, to recipients and body*. The following is in the body section:

{{vars.steps.Endpoints_needing_update.data.code_output}} which lists endpoints that are getting updated.

Because the playbook runs instantly there is a required step that will pause the script and wait for the customer to get a notification and a chance to respond. This is done with a wait step which I named “response time”. As seen in the S1 agent update process figure 8. there is a 3-day response time window for the customer and can simply be created with an input value of 3 to “days” field in the wait step.

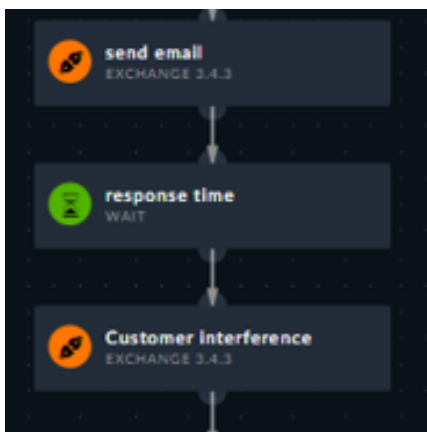


Figure 8. Playbook standby step after Sending email

For this alpha version, the playbook process will end when the email is replied to by the customer with any message. This is implemented with the Exchange connector step “Customer interference” by using the action field in the connector as Get Unread Emails have Folder Name as “test response” which is a custom folder in the email that sends update notifications.

Next, to make it possible for the playbook to have 2 possible outcomes: stop the playbook or continue with updating the agents. This is done with a connector Decision “Checkpoint for next step”. This follows the principle of programming’s if/else statement. Condition 1 value is **vars.steps.Customer_interference.data[0] != null** which means that if the customer responded to

the email then the next step will be “No Op” which is no operation step and stops the playbook. The default step is “S1 Update” which will continue with the update of S1 agents. In essence, it is “if customer interfered → no operation, else continue with update”, shown in figure 9.

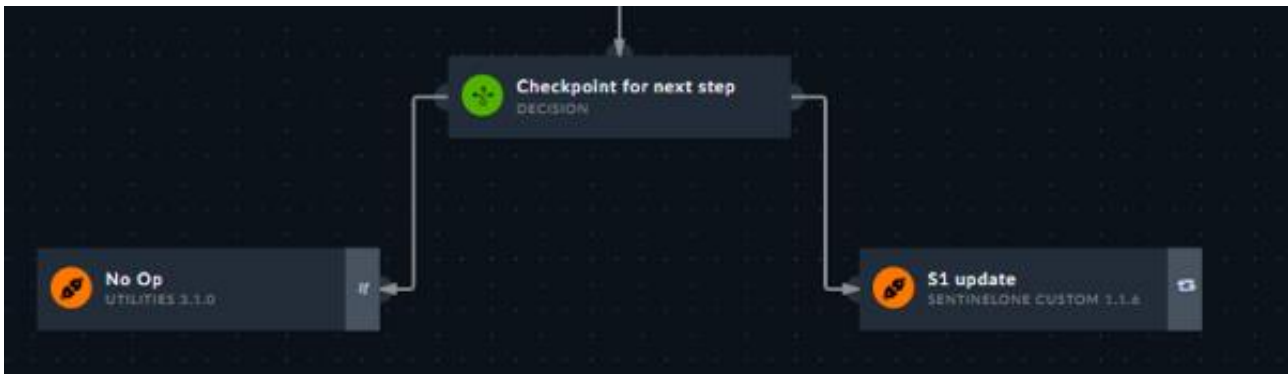


Figure 9. if/else step deciding if the playbook will continue

The last step for this playbook is “S1 update” which uses an S1 connector and will update the endpoints that the playbook has found in previous steps. This will be done with a custom action that does not exist in the default S1 connector. In the Action field, there is “Update agent” which is the S1 action “Upgrade agent”. To update agents, this connector needs agent UUID which will get its value from the “Agent UUID” step with `{{vars.item}}`. `vars.item` is used for the playbook’s own loop function. Installer File Name which gets its value from start parameter “S1 update package name” `{{vars.input.params.s1UpdatePacketName}}` and OS Type which gets its value from start parameter “OS Type” `{{vars.input.params.oSType.itemValue}}`. This time playbook’s loop function can be used which will loop **for each** value `{{vars.steps.Agent_UUID.data['code_output']}}`. This will go through the list of agents one by one from the “Get old agents from tenant” step and updates the agents to package version from value “S1 update package name”.

This is the alpha version of this automatization playbook and the completed playbook is seen below in figure 10.



Figure 10. Playbook's finalized form

5.1 Customer process changes

After working on the playbook until the “No operation step”, shown in figure 9 and consulting a few SOC specialists, it was decided that the customer process that was planned in figure 2 is not possible. Automation will not be possible with the test groups as every customer's update process currently is done differently and it is not possible to create full automation that works with every customer. Not every customer has a test group in their environment and test groups at the moment are done by filters which makes it not possible to fetch from different environments unless they are all named the same name. It is possible but was decided that this will be tweaked later in the future and the new process focuses on creating automation for large scale implementation.

The new update process for this thesis will be the mix of the currently used process and the suggested process in figure 5. The new process is seen in figure 11.

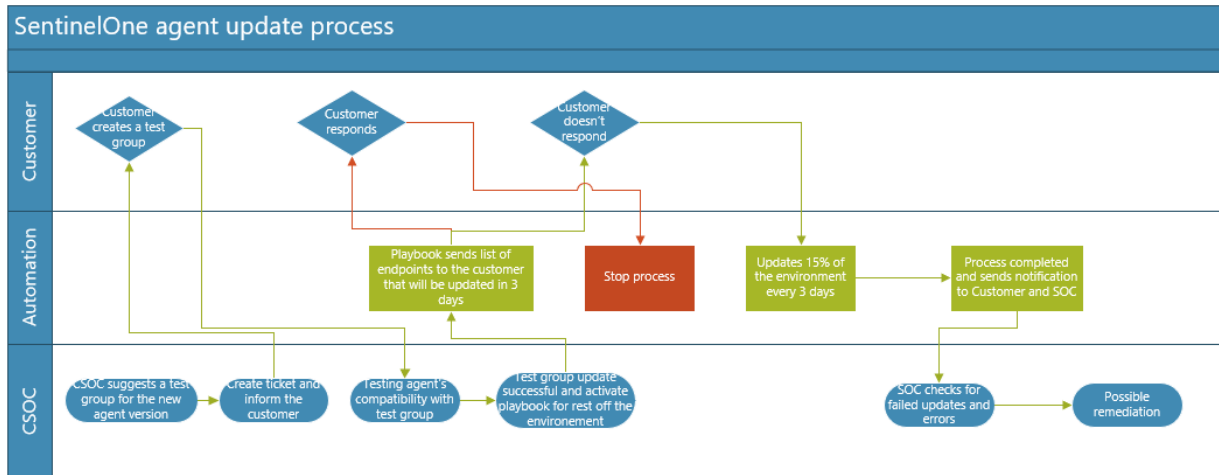
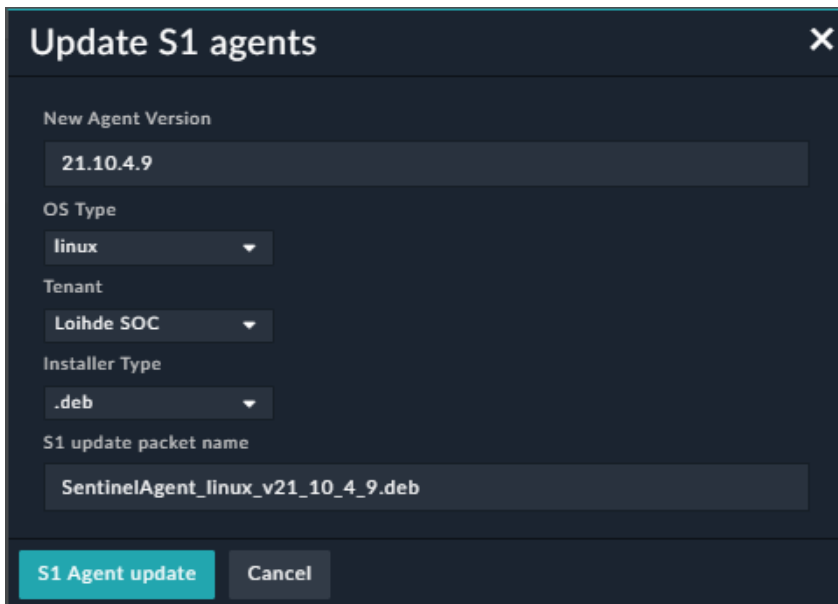


Figure 11. New update process

6 Functionality testing

Now that the playbook is ready it's time to do a test run. There are 3 Debian OS virtual machines with S1 agents installed on them and reside inside a SOC site(tenant) in S1 which is used for testing. When the playbook is run, it will request the user to fill required information for an update where the user inputs the new agent version, OS type, tenant and S1 agent update install packet name from S1. For this, test fields will be filled by test virtual machine information and S1 agent version 21.10.4.9 as seen in figure 12.



Update S1 agents

New Agent Version
21.10.4.9

OS Type
linux

Tenant
Loihde SOC

Installer Type
.deb

S1 update packet name
SentinelAgent_linux_v21_10_4_9.deb

S1 Agent update Cancel

Figure 12. Playbook run prompt

The first thing to test is if the Email notification function works as intended. The continuation of the playbook run process can be cut short by simply removing the line between steps as seen in figure 13.

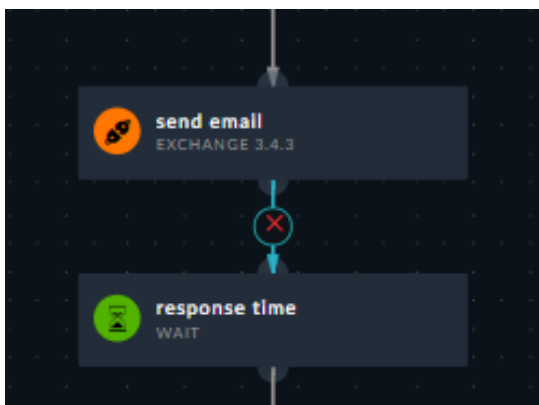


Figure 13. Removing step progress in the playbook

For testing this function, a test e-mail account is created that will be sending the notification which will be received on a personal e-mail that acts as a stand-in customer. As seen in figure 14 the

notification is sent to personal e-mail and lists the test endpoints that are deployed in the environment for the update.

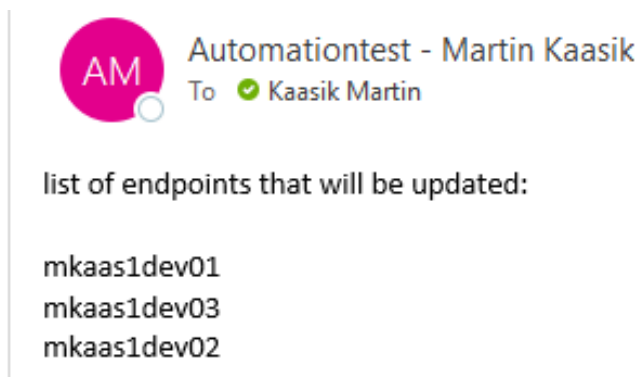


Figure 14. Email notification list

Next to test is the “No Operation” function and see if it stops the playbook process completely once a reply email is sent. To test this, the response time is going to be changed into 1 minute and a reply will be sent to the notification email that the playbook sends. As seen in figure 15 the playbook has gone to No Op step after receiving the email and has stopped all operations.

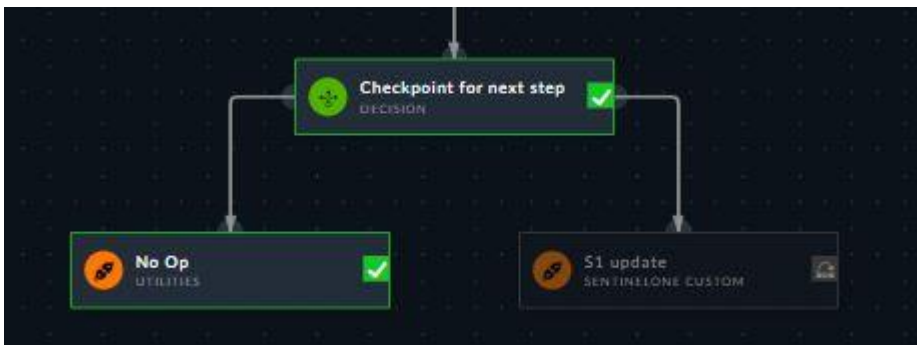


Figure 15. No operation step

The last test is to see if agents on the test endpoints will get the specified S1 agent version installed. The version for this test will be the 21.10.4.9 version from S1 for Debian based systems as test endpoints are Debian. As seen in figure 16 the test endpoints got updated to the newest agent version successfully.

Task Name	Description	Status	Initiated by	Initiated time
Agent Upgra...	SentinelAgent_linux...	Completed	CSOC Servic...	Jan 27, 2022

Figure 16. update confirmation

This is the end of PoC, there is much to improve and won't be used in actual production. The biggest problem with implementing this playbook is that agent version updates are not 100% successful, whether you update them manually or using the playbook, and in many cases need multiple attempts or troubleshooting to see what is preventing the update. The second problem is that in most environments there are legacy endpoints where agents can't be updated or high risk endpoints that need confirmation before the update is run to make sure it won't cause any possible disruption in production.

7 Conclusion

Automation is a great tool that assists in saving time from menial tasks that can be automated so that human interference is minimal. Automation in companies reduces repetitive work so that humans can work on more meaningful tasks that need human interaction. But nothing is perfect, even though in most cases automation is technically possible, there are often other factors that require constant human interaction thus losing the purpose of automating a process. Some parts of the process could still be automated to alleviate the workload for workers and assist in faster resolution from human interaction. The PoC that was created in Chapter 5 is a great example that started with a simple idea and seemed like the whole process could be automated easily until roadblocks kept popping up. For example, every environment's S1 agents are updated with different rules, updates not going through for unknown reasons, no real consistent update process that would apply to all tenants/environments etc. Even though it would be of no use in production currently, it is a working alpha version that can be improved so it could be used in actual production.

Currently, it is missing features to narrow down endpoint selection further and check whether the S1 agent update succeeded on endpoints.

Automation in production is beneficial and often a must. Not everything can be automated it's important to know in what situation automation should be used. Automation shines best in repetitive tasks that are straightforward and logically consistent. Table 1 will go over about benefits and disadvantages that might occur in automation.

Table 1. Comparing automation benefits and disadvantages

The pros of automation	The cons of automation
Doesn't need human interaction once it's running	Does what it is programmed to do and cannot handle exceptions that it's not coded for
Makes production run smoother	May Denial of Service (DoS) own services with bad configuration
More time to work on tasks that need human interaction / less repetitive tasks	Susceptible for errors
Saves money in the long term	May break with updates
Runs complex actions with simple inputs	May have conflict with other services

It is important to understand the pros and cons of automation in order to recognize when and where automation may be implemented. That way you can avoid counterproductive work and have an efficient workflow within the company.

8 Discussion

I did not reach the original goal I had in mind as the automation process for the S1 agent update was of a bigger scale than what I had initially thought. The goal of the thesis was to automate the S1 agent update process which seemed like it would be a simple automation task, but along the way, more problems started popping up and I had to decrease the scope of this project significantly. The first problem that I came across is that there are too many customers whose update process hasn't been finalized and have special requirements which makes it hard to implement in a single playbook. In the end, I had to reduce the scope of my project to mass update which would be done after all the testing steps in the customer environment. The main problem that cannot be

fixed is that S1 agent updates are often not going through for unknown reasons (also manually, not just with playbook) and require restarting the update again on endpoints. Every time an update is executed, it would require a check-up that the update has succeeded thus rendering the automation useless as it needs human investigation if the update failed.

Even though the playbook could not be used in actual production, it does the job that it is made to do. Automation is a delicate process where in some cases it will only work if everything else is robust. The more external factors there are that can affect the automation the more likely problems might occur which will hinder the automation. The problems that arose during this project were external problems which turned out to be a great example of why automation is sometimes not possible. The core problem is that S1 agent updates fail due to unknown errors which S1 support couldn't find the reason for either. This complicates the reliability of the update process if it were to be automated. For that reason, the project ended in a dead end, but overall, it was a decent attempt at creating automation with the SOAR playbook. This proved that things outside SOAR can be automated with a SOAR playbook which was the goal of this project.

I learned a lot about playbook creation but had some challenges trying to find workarounds for missing functions that the playbook was lacking. I was surprised by how easy it was to create a playbook and how versatile it is for how little input it needs. I was using a custom connector for S1 which is a modified version of the stock S1 connector. Because I didn't have to modify the connector myself, it was easier for me to create a playbook as it had the actions and variables that I needed. For actions that were not available in the connector were fixed by creating a simple python script which ran the action with the code snippet tool in the playbook.

This playbook can still be improved and possibly be used in production after finding a solution for core problems. This would make the automation possible in the future, unless S1 launches their own automated update function.

Sources

David, M & Shon, H & Allen, H & Stephen, V & Chris, B. Published 2011 by McGraw-Hill Companies. Security Information and Event Management (SIEM) Implementation

Fortinet Document Library. Introduction to Playbooks v7.0.0. Referenced 2021, December 19 <https://docs.fortinet.com/document/fortisoar/7.0.0/playbooks-guide/331279/introduction-toplaybooks>

Loihde Trust. Published 2021, September 2. Loihde's subsidiaries Viria Security, Tansec and Spellpoint operate together under the Loihde Trust brand as of 1 September 2021. Referenced 19.11.2021. <https://www.loihdetrust.com/en/news/loihdes-subsiadiaries-viria-security-tansec-and-spellpoint-operate-together-under-the-loihde-trust-brand-as-of-1-september-2021-2/>.

Loihde Group.N.d. Referenced 19.11.2021. <https://www.loihde.com/en/frontpage/>

Mohammad, S & Surya, L. 2018, June 1. Security Automation in Information Technology. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3652597

Nelson, H. 2018. NOC/SOC Integration: Opportunities for Increased Efficiency in Incident Response within Cyber-Security. <https://www.sans.org/white-papers/38290/>

RedHat. Published 2018, March 21. Understanding automation. Referenced 13.01.2022. <https://www.redhat.com/en/topics/automation>.

RedHat. Published 2019, October 15. What is orchestration?. Referenced 13.01.2022. <https://www.redhat.com/en/topics/automation/what-is-orchestration>.

Security operations center (SOC) tools. Referenced 08.05.2022. <https://cybersecurity.att.com/solutions/security-operations-center/building-a-soc/soc-tools>.

VIB. Published 2018. The State of SOAR 2018

Zimmerman, C. 2014. Ten Strategies of a World-Class Cybersecurity Operations Center, MITRE