5. melléklet

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

NEUMANN JÁNOS INFORMATIKAI KAR

THESIS

OE-NIK Hallgató neve: **Boros Alex**
2022 Hallgató törzskönyvi száma: **T/006060/FI12904/N**

Óbudai University
John von Neumann Faculty of Informatics
Institute for Cyber-physical Systems

THESIS WORK SPECIFICATION

Name of the student:	**Alex Boros**
Identification number of the student:	T/006060/FI12904/N
Neptun's code:	[redacted]

Title of the thesis:

Modern Automated Threat Detection and Prevention utilizing Cisco SecureX
Veszélyforrások automatizált észlelése és támadások megelőzése modern módszerekkel a Cisco SecureX rendszer segítségével

Internal supervisor:	Dr. Eszter Balázsné Kail
External supervisor:	Ács György
Deadline:	15 December 2021
Subjects of the final exam:	Computer Architectures Cloud service technologies and IT security specialization

The task:

The modern concern for cyber threats does not only cover the initial access. We have to consider attacks from the inside of the network with a well-guarded boundary. The most prevalent types of attacks are well defined in MITRE ATT&CK, a public database of adversary tactics and techniques. Many solutions have started combining their existing technologies into a single coherent system. To utilize the resources correctly and effectively, there needs to be a form of automation such as Cisco SecureX Orchestration.

The goal of this thesis is to establish a system with the ability to map the occurrences, their frequency, and severity to the ATT&CK matrix of threats. Once mapped, we should construct new workflows to prevent known sources of possible damage using Orchestration.

The thesis must contain:

- reviewing the ATT&CK model and its limitations,
- exploration of existing solutions benefitting from and contributing to ATT&CK,
- designing a network of workflows to detect or redirect already deployed detections to contribute to the statistics,
- evaluating the results of the detection map and selecting the dominant threats,
- implementing intrusion prevention and termination of previously entered infections,
- testing of the prevention system using artificial related and non-related attacks,
- future development plans.

Place of stamp

.....
Dr. Rita Fleiner
head of institute

This specification is valid until: **15 December 2023**
(According to OE TVSz 55.§)

I consider the thesis suitable for submission:

.....
external supervisor

.....
internal supervisor

7. melléklet

HALLGATÓI NYILATKOZAT

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar

Alulírott hallgató kijelentem, hogy a szakdolgozat / diplomamunka saját munkám eredménye, a felhasznált szakirodalmat és eszközöket azonosíthatóan közöltem. Az elkészült szakdolgozatomban / diplomamunkámban található eredményeket az egyetem és a feladatot kiíró intézmény saját céljára térítés nélkül felhasználhatja.

Budapest, 2022. MÁJUS 14.

Boros Alex

hallgató aláírása

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve: BOKOS ALEX

Neptun kód: [redacted]

Tagozat: ANGOL NÉPPINKINFORMATIKUS BSC

Telefon: [redacted]

Levelezési cím (pl: lakcím): [redacted]

Szakdolgozat / Diplomamunka[1] címe magyarul:
VESZÉLYFORRÁSOK AUTOMATIZÁLT ÉSZLELÉSE ÉS TÁMADÁSOK MEGELŐZÉSE MODERN MÓDSZEREKKEL A CISCO SECUREX RENDSZER SEGÍTSÉGÉVEL

Szakdolgozat / Diplomamunka[2] címe angolul:
MODERN AUTOMATED THREAT DETECTION AND PREVENTION UTILIZING CISCO SECUREX

Intézményi konzulens: BALÁZSNÉ DR. KAIL ESZTER

Külső konzulens: KÁS GYÖRGY

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.02.26	CÍMVÁLASZTÁS	Kail
2.	2021.03.05	FELADATKIÍRÁS TÁJÉKOZTATÁS	Kail
3.	2021.03.21	PONTOSABB FELADATKIÍRÁS	Kail
4.	2021.05.10	TARTALOM ELLENŐRZÉS	Kail

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplo-mamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2021. 05. 10.

Kail
.....
intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

ÓBUDAI EGYETEM
ÓBUDA UNIVERSITY

Neumann János Informatikai Kar
9. sz. melléklet

KONZULTÁCIÓS NAPLÓ

Hallgató neve: Boros Alex

Neptun kód: [redacted]

Tagozat: Angol Mérnökinformatikus Bsc

Telefon: [redacted]

Levelezési cím (pl: lakcím): [redacted]

Szakdolgozat / Diplomamunka[1] címe magyarul:

Veszélyforrások automatizált észlelése és támadások megelőzése modern módszerekkel a Cisco SecureX rendszer segítségével

Szakdolgozat / Diplomamunka[2] címe angolul:

Modern Automated Threat Detection and Prevention utilizing Cisco SecureX

Intézményi konzulens: Balázsné Dr. Kail Eszter

Külső konzulens: Ács György

Kérjük, hogy az adatokat nyomtatott nagybetűkkel írja!

Alk.	Dátum	Tartalom	Aláírás
1.	2021.10.18.	A megvalósítás elméleti megtervezése	Kail
2.	2021.10.21.	A feladatkiírás pontjainak átbeszélése	Kail
3.	2021.11.09.	A hiányzó tartalom megfogalmazása	Kail
4.	2021.11.16.	Javítandó pontok feljegyzése, ellenőrzés	Kail

A Konzultációs naplót összesen 4 alkalommal, az egyes konzultációk alkalmával kell láttamoztatni bármelyik konzulenssel.

A hallgató a Szakdolgozat I. / Szakdolgozat II. (BSc) vagy Diplomamunka 1 / Diplo-mamunka 2 / Diplomamunka 3 / Diplomamunka 4[3] tantárgy követelményét teljesítette, beszámolóra / védésre[4] bocsátható.

Budapest, 2021.12.09.

Kail

intézményi konzulens

[1] Megfelelő aláhúzandó!
[2] Megfelelő aláhúzandó!
[3] Megfelelő aláhúzandó!
[4] Megfelelő aláhúzandó!

TABLE OF CONTENTS

ABSTRACT

Malicious software (malware) has evolved to do more in different ways, so much so that organizations like MITRE started arranging real-life data about adversary tactics and techniques into table-like structures. What the dominant if not single form of malware was decades ago is just a single column labeled Initial Access in the known tactics today. Most sources are contributing to the unified techniques often through generalization. This thesis deals with the inverse: using generalized knowledge to build specific defense.

It is advised not to use any generalized idea as a test of coverage. Is it also a misstep to base the protection on a collection of generalized threats? If we had a ready-made drop-in solution for known adversary activities, we would have more time to discover new, previously hidden malicious acts.

MITRE ATT&CK is well known to be a collection of Adversarial Tactics, Techniques, and Common Knowledge, without any specific implementations, giving a common language to corporate entities of different cultures or even individuals that can help in communicating attacks or identifying behavior of well-known adversary groups.

Cybersecurity software are commonly categorized in specialized tactics and rated against different techniques but deploying active defense against uncontacted menaces is unheard of. This new incentive might help prevent catastrophes in the long term. Passive defense is already known to work well but some areas require active protection that has little to no specific documentation.

This thesis deals with realizing active countermeasures against the tactic TA0008 Lateral Movement of ATT&CK using Cisco SecureX. The solutions were planned to be as native to SecureX as possible, but they could not be implemented in a perfect way. The work even includes a counterexample, where sticking to passive solutions would be more logical in a real scenario.

1. INTRODUCTION

Modern cybersecurity requires up to date definitions and substantially dynamic approaches to combat offensive behavior originating from outside of a network, in addition to guard against advancement of undetected infections and attacks emerging from the “trusted” part of the network.

1.1. PERSONAL MOTIVATION

I have been given the opportunity to use Cisco SecureX[1] in form of a Proof of Value license. I am always fascinated by programming languages, or rather block-based visual languages. I believe it is the perfect balance between fundamental realization and ease-of-use, especially if the list of available building blocks can be extended using other languages, which is Python[2] in SecureX’s case. Having the ability to link independent programs, systems, or networks together via a standard interface is akin to the microservice architecture, both having the same benefits and drawbacks.

As a Computer Science Engineering student, I have great interest in solving manual tasks with code, more so if the original problem involves networking. The programs I am familiar with have a strong dependence on operational and clean networks, which is how I imagine the future development standards will shift, and I am an advocate of this move.

In this thesis work, I wish to start an incentive to fully adopt MITRE’s community-driven public library of known tactics and techniques defined in their product, ATT&CK[3]. While it is recommended for testing against, rather than basing on, I believe that the most straightforward and fastest way of helping ATT&CK and similar knowledge bases grow is to first develop a defense sensibly against their definitions of techniques, or to focus on the gaps pointed out in existing lines of defense. Personally, I think this method would make common threats stop working on a large scale, bringing light to the gaps of the knowledge base through contributions about newly discovered adversary techniques.

1.2. GOAL

The goal of this work is to implement countermeasures for all Windows targeted techniques of ATT&CK’s Lateral Movement tactic. The reasoning behind limiting the implementation to Windows only is that Cisco only supported Orbital[4] (their end-device operating system query service) for Windows 10 build 1709 (or server equivalent) or later when I began working on the implementation. The tactic contains techniques along the lines of Lateral Tool Transfer and Replication Through Removable Media. I believe that stopping Lateral Movement via a central system like SecureX is an adequate representation of both SecureX’s capabilities and the value of the ATT&CK Matrix.

2. OVERVIEW OF THE REFERENCED SYSTEMS

This paper is going to deal with a partial implementation of MITRE ATT&CK into Cisco's SecureX cloud-native automation solution. Therefore, an adequate introduction of the mentioned systems is necessary. Cisco has decided to make Cisco SecureX Orchestration a distinct yet homogeneous product, which is the reason for the separated introduction of these products.

2.1. MITRE ATT&CK

MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) is a knowledge base of adversary tactics and techniques that is built to be publicly accessible. The data is based on real-life observations, and it is used to develop threat models and procedures in common circumstances.

The ATT&CK knowledge base is merely a guide for the right mindset when dealing with threat activity, as it is impossible to provide 100% coverage of every adversary tactic and technique. Also, using ATT&CK as a "checklist" when implementing cybersecurity and adversary defense is discouraged[5]. Not all adversary behaviors should be addressed, because it will provide an unreasonable amount of information for any current human or system to analyze.

As of March 2020, three different Matrices are published, each regarding a different category of end-devices: Enterprise, Mobile, and Industrial Control Systems[6]. This thesis work will focus on the Enterprise Matrix that contains platforms like Windows, Linux, or containers.

Several use cases can benefit from ATT&CK. Highlighting the main use cases from the ATT&CK Design and Philosophy[7] white paper, these are:

Red Teaming – Performing undetected penetration and spread using the Matrix, also known as "ethical hacking."

Behavioral Analytics Development – The initiative to detect potentially malicious behavior often without information about the used adversary means or tools. In other words, a dynamic system for identifying indicators of compromise.

Cyber Threat Intelligence Enrichment – Linking a specific combination of tactics and techniques to adversary groups to aid the categorization of behavior regardless of the tools used.

The other use cases mentioned in the paper are Defense Gap Assessment that helps in finding the "blind spots" of a network or system; and Security Operations Center Maturity Assessment that evaluates the effectiveness of the deployed defense against acknowledged threats. The use cases are further extended by Failure Scenario

Development and Educational Resource in the ATT&CK for ICS Philosophy[8] publication which only lightly applies for the Enterprise Matrix.

2.1. figure: Five arbitrarily chosen tactics with custom highlighting to help with the explanation. Made using the ATT&CK Navigator[6].

The model of ATT&CK consists of sets of techniques and sub-techniques that adversaries can use to reach their goal. These sets are categorized under one or more tactics (the columns of figure 2.1.) that identify the aforementioned goal of an adversary. The relationship between tactics, techniques, and sub-techniques can be conveniently visualized via the ATT&CK Matrix. For example, the techniques highlighted in yellow in figure 2.1. are all techniques categorized under the Impact tactic. The green colored techniques are sub-techniques of Data Obfuscation that belongs to the Command-and-Control tactic.

Techniques can have Mitigations attached. These are vendor indifferent descriptions that help with finding a real solution by categorizing and classifying the possible set of solutions without promoting a single product.

2.2. CISCO SECUREX

Cisco SecureX[1] is a cloud hosted platform that links together Cisco products and third-party implementations for the objective of unifying security related instruments. SecureX is natively integrated with domestic services, for instance Cisco Secure Endpoint and Cisco Umbrella, as well as foreign services including VirusTotal[9]. It also provides easy integration with any Application Programming Interface (API) over the network. It has six key features that make up SecureX.

2.2.1. RIBBON

The ribbon is an ease-of-access tool that appears on ribbon-enabled applications, such as AMP for Endpoints and Threat Response. It provides a unified experience with the addition of ribbon applications that maintain context when navigating across Cisco Secure products.

2.2.2. THREAT RESPONSE

SecureX Threat Response[10] is a single console that gathers information about incidents from Cisco security products and third-party APIs. It also offers global threat intelligence from Cisco Talos[11]. The point of Threat Response is not just collecting and displaying information but being able to react directly from the interface without leaving the module or opening up another console. It is capable of blocking files, URLs, or domains, isolating hosts, and launching Orchestration workflows to help with automatization.

The most visually prominent feature of Threat Response is that it can give information about and specify the exact relation of observables which might be domain names, IP addresses, file hashes, devices, or even users and context-specific entities. These observables can be either clean, malicious, suspicious, common, or unknown; each easily distinguishable by the color coded labels and icons[12] as seen in figure 2.2.2.

To present an example, I have entered 52 observables using the official ribbon browser extension[13] from a Cisco Talos report on Lemon Duck[14], a malware that spreads across the network to convert resources to cryptocurrency mining slaves. Figure 2.2.2. clearly shows that observables in relation with Lemon Duck are indeed deemed malicious (red-colored). There are also unknown observables depicted with gray symbols. Conveniently, Threat Response can identify relationship clusters like in this case. The left side of the graph has been rearranged to be visible, but the right side is untouched.

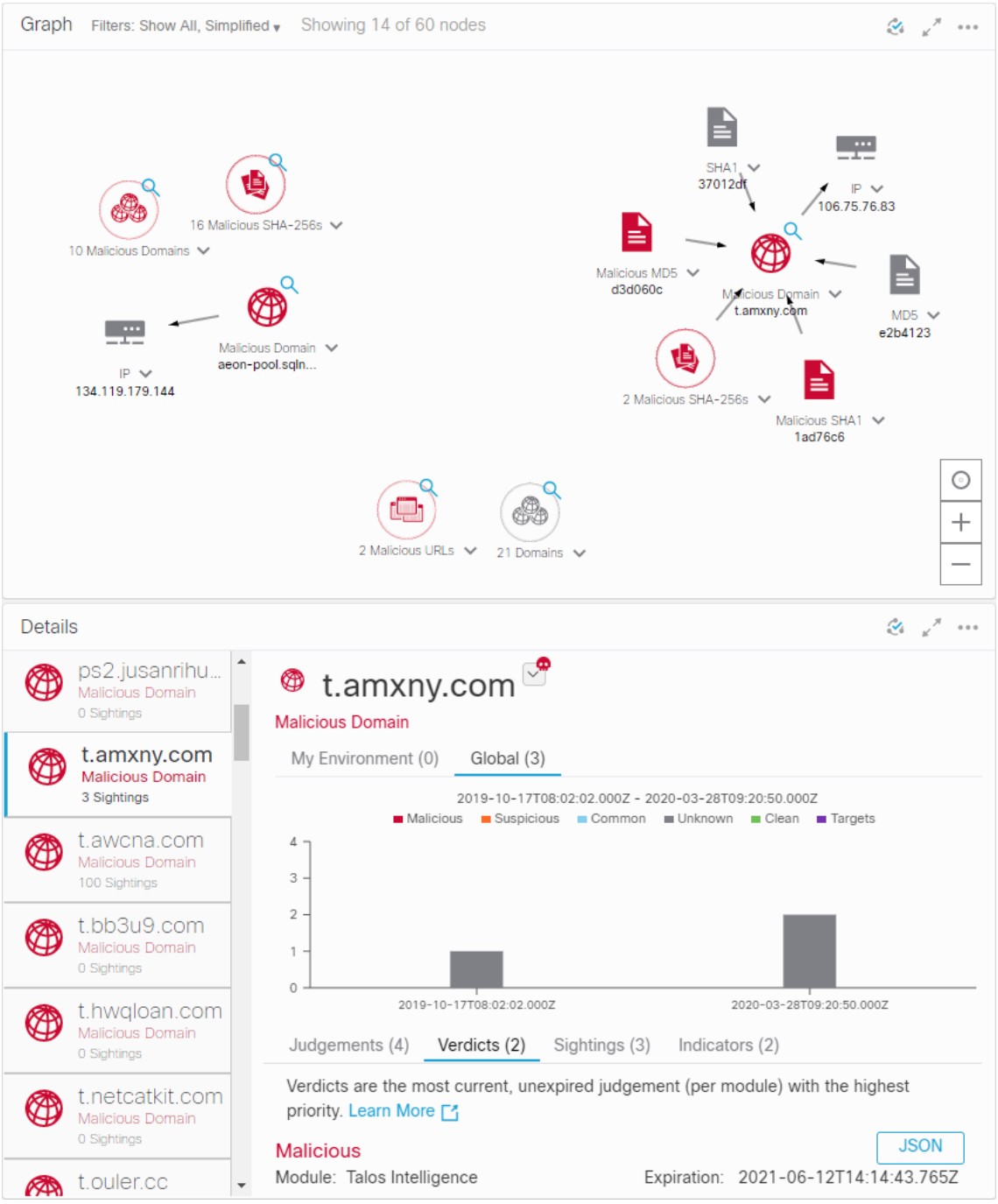

Figure 2.2.2. Enlarged, rearranged Threat Response dashboard (cropped) to show the original bottom half (without the ribbon) of Threat Response that contains the "Graph" and "Details" tiles.

On the bottom half of figure 2.2.2. (Originally the lower right tile of Threat Response), an arbitrarily selected malicious domain is visible. The Details tile shows that two global sightings occurred, both in 2018. Jumping to the Verdicts tab, we can see the unexpired Judgements (one visible) that are sourced from local decisions and Threat Intelligence Teams, in this case Cisco Talos.

2.2.3. ORCHESTRATION

Cisco SecureX Orchestration[15] is a key module of Cisco SecureX that allows automating jobs and functions using workflows. Cisco provides a prebuilt collection of workflows that can solve common problems; or new workflows can be added to the system with a simplified graphical interface that often requires no programming code to be written.

This module is responsible for Security, Orchestration, Automation, Response (SOAR) in SecureX. It pursues consistency, efficiency, and accuracy by partly removing the human factor from repeated tasks and offering automatic triaging, containment, and eradication of confirmed threats[16].

Orchestration is going to be the main module in this thesis work because it provides access to all integrations on top of the ability to create non-core activities using Python, its input variables, and the capability to export any variable from runtime to use in other activities inside the workflows. For added clarity, let us explore the Orchestration components[17].

Activities – Activities in Orchestration are the building blocks like functions in traditional languages. Cisco provides a variety of built-in "core" activities including running a custom Python script.

Workflows – Similar to scripts in other programming languages, workflows provide a canvas to connect activities in logical order that may or may not depend on conditionals or loops.

Atomic Actions – Atomics are similar to defined functions in other languages. They can be defined as workflows inside workflows. An Atomic takes up one Activity space in a workflow. It is recommended to make Atomic Actions reusable.

Calendars – A collection of days is called a calendar. For example: weekdays; weekends; all Hungarian workdays.

Schedules – Schedules derive from calendars. They can be used to trigger workflows at a specific time in a freely chosen time zone. Additionally, the number of repetitions and the delay between each repetition can be set.

Events – Events are triggers that activate when a condition is met. A common example is an arriving mail that triggers a workflow.

Targets – Communication endpoints are called targets in Orchestration. Some targets will require authentication, that is why Account Keys can be defined.

Account Keys – Account Keys are essentially credentials. Orchestration supports a wide variety of default credential types, such as AWS Credentials, Git Token-Based

Credentials, et cetera. Naturally, custom HTTP Basic Authentication is the preferred method when setting up an account with an unknown third-party service.

Tasks – As workflows run "in the background," without a visible terminal, dialog with the workflow is impossible without a medium. Tasks provide a way for workflows to ask questions and pause until they are answered. The question can be as simple as a yes/no question, but it can even be a multiple field text input.

Variables – Variables are convenient storage labels for arbitrary data types of data. For complex data, Cisco recommends* the Table data type that is similar to an Excel spreadsheet. Variables exist in the context of a workflow, but also global variables can be defined to share information between multiple workflows, or to have a single reference, thus making future changes easier and more consistent. Workflows can have input variables. If a workflow has an input variable, each execution requires an input variable to run. Commonly context from Threat Response, or manual user input is used.

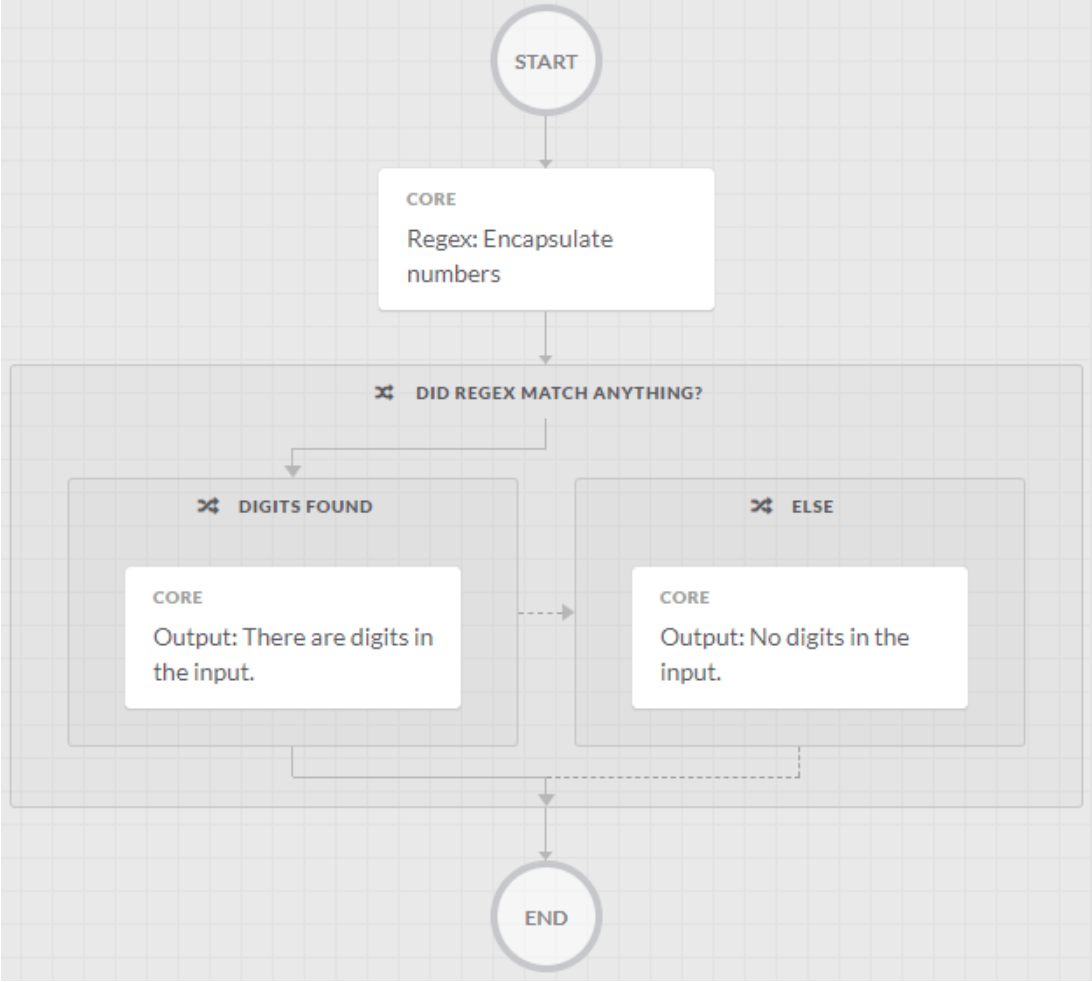

Figure 2.2.3. Screenshot of a sample workflow constructed to make this example.

* Recommended in the Cisco Partner Training Program.

Figure 2.2.3. shows a sample workflow that detects numerals in the input. The workflow is configured to have an input (with the "required" flag) and an output variable, making it suitable to convert it to an Atomic. In this example, I used regex to match "/\d+/g" which is fundamentally checking for numbers. A condition block follows the first Activity that has "Regex Activity → Matched (Boolean) equals true" in the left-hand tree and "[…] not equals true" in the other tree. Both trees have a Set Variable (Core) Activity that set the output variable of the workflow to the values seen in the figure. Without any further setup, this workflow can be manually run to evaluate if a string (entered when executing the workflow) has number characters in it. Another possibility is to convert the workflow into an Atomic and use it as a simple "numeral detector" in another workflow.

2.2.4. DASHBOARD

The dashboard gives unified visibility over the security environment. The dashboard is empty out of the box and can be populated with modules from integrations. Each user has a personal dashboard and has the ability to create more, but shared dashboards can be set up for a unified team experience. The default page of SecureX is the dashboard.

2.2.5. SIGN-ON

SecureX supports many means of authentication, with Cisco Secure Access by Duo multi-factor authentication (MFA) being the standard tool for two or more factor authentication. Duo is integrated into SecureX by default, and it has the added feature to quickly address Orchestration tasks. SecureX comes with all Cisco Security solutions, for this reason SecureX has a special Sign-on process that accepts many forms of credentials.

2.2.6. INTEGRATIONS

SecureX is integrated with Cisco Secure products by default and has pre-packaged integrations for select third parties. Since SecureX is an open platform to integrate with any third-party infrastructure, customers are encouraged to migrate their deployed solutions into SecureX. The requirement for integration is an API that is linked to SecureX via an API key. An extensive documentation is available to make tools SecureX-capable[18]. An alternative way of integrating is to use Threat Response[19].

2.3. CISCO SECURE ENDPOINT

Cisco Secure Endpoint[20] (formerly known as AMP for Endpoints, or AMP for short) is a product of Cisco that takes on prevention, detection, and response, with appended threat hunting capabilities in the way that it comes closely integrated and bundled with SecureX. Cisco SecureX comes with all Cisco Secure products, and AMP is one of them. Even without reinforcing AMP with SecureX, it has connections to other Cisco products

like Talos[11], Cisco's Threat Intelligence Group. These connections enable Cisco Secure Endpoint to function as a basic but well-informed anti-virus software as long as the platforms wished to be protected are any of the supported versions. These compatible platforms include desktop and server operating systems like Windows versions that have not reached End of Support, Red Hat Linux, and macOS, together with mobile platforms like Android or iOS.

2.3.1. CISCO ORBITAL

This thesis work will attempt to adopt the generic detection of one of ATT&CK's tactics. For this reason, I will need AMP to provide its threat hunting spotlight: Cisco Orbital[4].

Cisco Orbital is a service that uses Osquery that lets other interfaces access the operating system as if it were a relational database, meaning that Structured Query Language (SQL) queries can be issued to collect information about a machine running AMP with Orbital support. Orbital is not only a tool for SecureX users. Cisco's integrations allow other products to automatically communicate with Orbital-enabled endpoints automatically.

Authentication is simplified: instead of a separate set of credentials, users or other applications can authenticate themselves using SecureX or AMP logins or users when accessing the Orbital API. It was only available for Windows endpoints using Cisco Secure Endpoint at the start of my work, which is why I will be mentioning Windows specific features, such as Windows registry.

My chosen tactic of realization from the ATT&CK Matrix is Lateral Movement that will require fetching a filtered version of the Windows registry values, which can be natively done using Orbital queries. For starters and advanced users performing common tasks, Cisco provides a query catalog that offers a library of queries categorized by use case e.g., threat hunting.

Once a suitable query from the catalog is selected or it is written by hand, the query can be run on selected endpoints that immediately show the results upon completion. An alternative is to create a job that can be run on a scheduled basis. Integrated products from SecureX can also access jobs, start them, and view their results, which is key for unified automation.

3. EXISTING SOLUTIONS

Developers of security related products often reference the MITRE ATT&CK Matrices in their reports, as a way to represent coverage of their product. Also, ATT&CK provides a way to share intel about Advanced Persistent Threat (APT) Groups: “Leveraging MITRE ATT&CK translates Kaspersky’s unique APT expertise and knowledge into common language.” [21]

In this work, I will attempt mapping the ATT&CK Enterprise Matrix’s Lateral Movement tactic to Cisco SecureX, using the Matrix as the roadmap as opposed to using it as a tool for Defense Gap Assessment.

3.1. OSQUERY-ATT&CK BY TEOSELLER

Osquery-ATT&CK[22] is an open-source effort made by Filippo Mottini[23] to extract information from Osquery that can be used for threat hunting. It successfully maps 94 techniques in total, of which 65 for Windows and 50 for Linux machines. The project is provided in Osquery Pack format. The code is meant to be used as the source for a critical information logger that gathers information based on ATT&CK tactics labels.

For example, “windows-incorrect_path_process.conf” tries to solve tactic T1034: “Detect processes masquerading as legitimate Windows processes” by looking for well-known Windows process names that do not run from the official directory path. I reckon this file can be deployed to production use because it is safe to use. Core Windows components should never run from a different location.

The main disadvantage of this Osquery Pack collection is that most queries are under filtered. Take “windows_scheduled_tasks.conf” as an example: it lists all tasks in the Task Scheduler. Understandably, no real generic solution is possible, this might be the reason for leaving a query of this size in the configuration.

The repository accepts contributions, which is why I am going to suggest two improvements.

- First, the Osquery Packs were made in 2018, which is considered extremely outdated by today’s standards. All configuration files should be re-visited and updated to match the current definitions of MITRE ATT&CK.
- As an added feature, to spot the “suspicious” activity, the query should only present, or present a list separately that only contains uncommon records from the enormous queries.

3.2. SENTINEL ATT&CK BY BLUETEAMLABS

BlueTeamLabs[24] decided to create an open-source GitHub project that aims to provide a simple and quickly deployable Microsoft Azure hosted software, the Sentinel ATT&CK[25]. It communicates with the SYSMON[26] (Microsoft System Monitor API) to collect data for threat hunting. Its key feature is its integration with Azure Sentinel[27], Microsoft's cloud-native Security Information and Event Manager (SIEM).

The project is realized using an automated Azure Resource Manager template for fast deployment, a configuration file for SYSMON, and 177 queries written in Kusto[28], the Azure Data Explorer's query language. In combination, the tool covers the detection of 156 ATT&CK techniques.

The official GitHub page provides an extensive documentation of use cases and real usage, accompanied by examples. Without any configuration, Sentinel ATT&CK can be used in test environments, and for purple teaming. For production use, it requires tuning and testing to be effective due to the amount of information being presented.

Although the configuration file for SYSMON is meaningfully constructed, it may leave blind spots. For example, the DNS report excludes common trusted sites, two of which are Skype's domains. Skype is a peer-to-peer messaging and calling service that enables users to mask their public IP addresses if requested under the Skype domain. While it is unlikely for adversaries to attack using Skype as a proxy, considering that there may be countermeasures implemented on Skype's side, a targeted attack could easily exploit this gap in the defense.

3.3. SIGMA2ATTACKNET BY 3CORESEC

It is worth discussing whether mappers are considered as ATT&CK based implementations. They are essentially taking the output of an already deployed, operational solution, like a SIEM that uses the open-source Sigma format in this case, and converting it to MITRE's predefined format[29] that can be loaded to the ATT&CK Navigator[6].

Sigma2AttackNet[30] is made to scan a folder that contains Sigma rules and provide ATT&CK compatible output. The feature that made me introduce this work is that this program is capable of classifying Sigma rules as ATT&CK techniques, and the categorization of rules as ATT&CK tactics, wherefrom it can spot mismatches between tactics and techniques. This is a useful feature for discovering improper categorization that might have held back an investigation.

Due to the lack of substantial documentation, I could not find any weaknesses in the system, I recommend browsing the source code, as it is an open-source project. The code

is written in .NET Core without Python dependencies, which are often required by other programs that interact with MITRE ATT&CK in general.

3.4. DETT&CT

DeTT&CT[31] is another mapper for ATT&CK that is able to score visibility and detection using analysis of data sources. It relies on a SIEM or a similar security information aggregator that can output custom YAML[32] files. DeTT&CT has its own specification of data source administration (input) file that has the option to be monitored for changes and can operate real-time with an ATT&CK Matrix heatmap output that reflects the live visibility and detection. Depending on the size of the data source pool, it can take a long time to set up automatic analyzation of every given data source and dynamically rate them in five quality dimensions: device completeness, data field completeness, timeliness, consistency, retention. DeTT&CT provides a standardized scoring table to help in implementing the functions or to assist manual evaluation.

In contrast with Sigma2AttackNet, this project follows the customs built by MITRE and uses Python as its core language. It lists attackcti among other required Python libraries that is able to retrieve ATT&CK techniques from data source definitions. It is available as a local tool or as a Docker[33] image.

DeTT&CT offers an editor[34] program for their YAML file to help out beginners that are overwhelmed by the usage of DeTT&CT, as well as for experienced users who either wish to estimate by hand or would like to create a custom template that is different from the two examples available on the official GitHub page[31].

MITRE recommends[35] DeTT&CT as the next step after becoming familiar with ATT&CK.

4. REALIZING DETECTION

This section of the thesis mainly deals with mapping the Lateral Movement tactic of the MITRE ATT&CK Enterprise Matrix to SecureX. I will be creating workflows with the primary goal of detection of all techniques (in the selected tactic) defined in the Matrix. If the detection results yield promising accuracy and not many false positives, I will continue to add prevention.

The main reasoning behind my choice of techniques is that Lateral Tool Transfer was the first technique that stood out to me when glancing over the ATT&CK matrix. I instantly had a plan in my mind for it and it matured well enough to make it into this thesis work. The other techniques were chosen with one goal in mind: "Do more with less." They have common elements with other techniques of the same tactic.

The chosen tactic, Lateral Movement[36] (code TA0008) is characterized by generally spreading, relocating, or moving in a targeted direction after the initial access. The objective of Lateral Movement can be of various natures. For example, the main goal might be to create a network map for future use, but it is also possible, that a specific host is targeted that is inaccessible directly from the outside network. In a greater impact scenario, the adversary might want to take over the whole network using tactics from Command and Control or Impact, for which Lateral Movement is a prerequisite.

I am limiting the development for Windows end devices only. Cisco Orbital[4] is a service built into AMP for Endpoints that lets SecureX Orchestration query the operating system as if it was a database. At the time of writing, Orbital only supported Windows systems running Windows 10 build 1709 (also known as "Redstone 3" or "2017 Fall Creators Update") and later or Windows Server 2016 or later. There is the possibility that macOS support will be released while realizing the goal but utilizing early releases for research is discouraged as they are not stable enough for consistent results.

To fully utilize SecureX, I am planning to capture natively supported detections in the Lateral Movement category and using them as triggers for workflows that categorize them in their ATT&CK Tactics, Techniques, and Procedures (TTP).T1570 Lateral Tool Transfer

Lateral Tool Transfer, or technique code T1570 is the copying of files from adversaries often in a silent manner. Server Message Block (SMB) protocol connected networks are supported by Windows and the SMB feature is enabled by default in a standard installation[37]. Exploiting file sharing protocols like SMB is the core aspect of Lateral Tool Transfer.

My solution to T1570 would be to introduce a new policy, in which file and folder sharing is monitored by Cisco Secure Endpoint's built-in Cisco Orbital service which is Cisco's implementation of Osquery.

The next step is creating a list of exceptions, which are in the form of file hashes. Officially approved documents and trusted executables should be included in the list. In SecureX, a global variable of type table[38] would be fitting for this task. I suggest two columns: file hash and thorough description. Once we have the list of exceptions, we can apply a new alerting behavior using workflows: files that are not present in the exception list but are present on more than a set number (threshold) of computers should raise an alert for examination. This threshold could be adjusted based on the deploying company's customs. In this work I will use "more than two copies" as the threshold because my testing environment is minute.

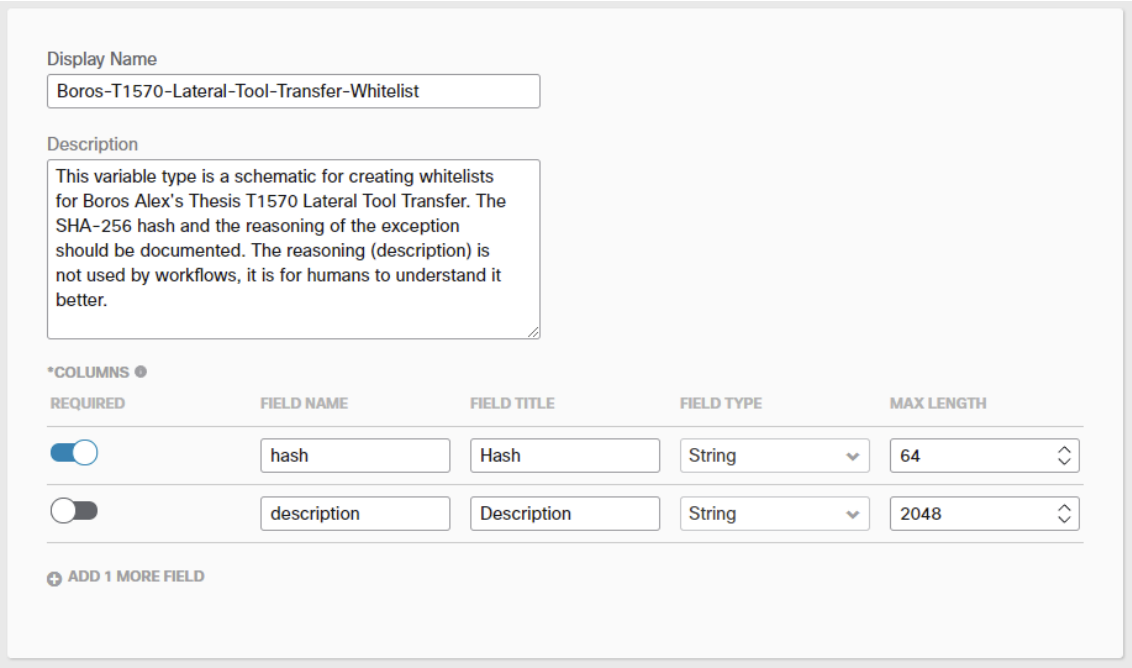

Figure 4.1.1. Table variable type: non-monitored files that are manually allowed.

Figure 4.1.1. represents the very first step to ensure user-friendliness. In SecureX Orchestration, For Each Blocks do not work without an iterable data type. The only iterable data type in Orchestration at the point of writing is tables. Persistent tables cannot be created on-the-fly, they need a template, which can be created separately in the "Variables" → "Variable Types" menu of Orchestration. I have chosen one field to be required and one to be optional:

- File Hash: an arbitrary hashing function's digest of a file's contents. (It needs to be consistent to work. Orbital will use SHA-256 by default.)
- File Description: *(optional)* A description of the file to be excluded. File hashes are not human-readable so there needs to be good documentation of the exclusions.

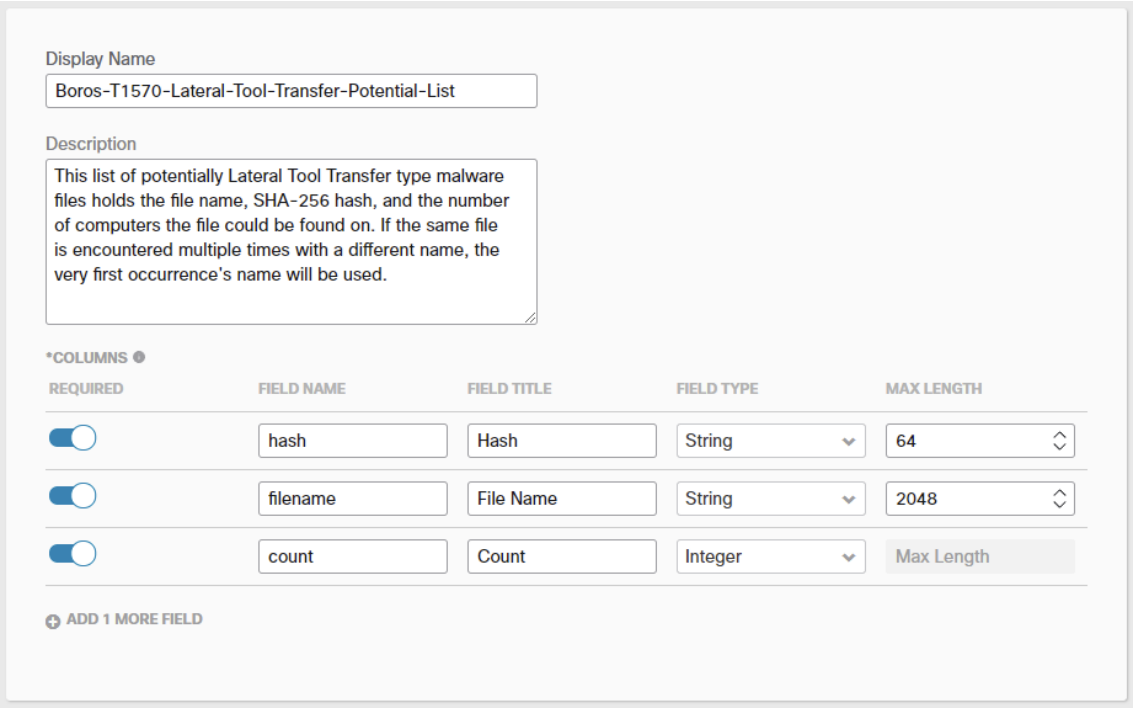

Figure 4.1.2. Table variable type: files found in shared folders.

A useful addition can be another variable type. Although files found in shared folders can be stored in temporary variables, it is better if suspicious files are logged in persistent table types. Persistent tables of this kind are especially useful for case-by-case investigation for Cisco Threat Response that can organize data from different sources. Figure 4.1.2. shows my idea of a table type suitable for persistent threat logging. It also features a non-string variable: the count. This count is only incremented per computer where the file can be found. Repeated discoveries of the same file on the same computer do not progress the count further. The file name might not be uniform across devices, so the very first occurrence will be inspected for file name.

4.1.1. CISCO ORBITAL QUERY

My original design would have been too complex for one Cisco Orchestration workflow therefore I broke it up into two parts. First, we need a workflow that gathers information about a single endpoint.

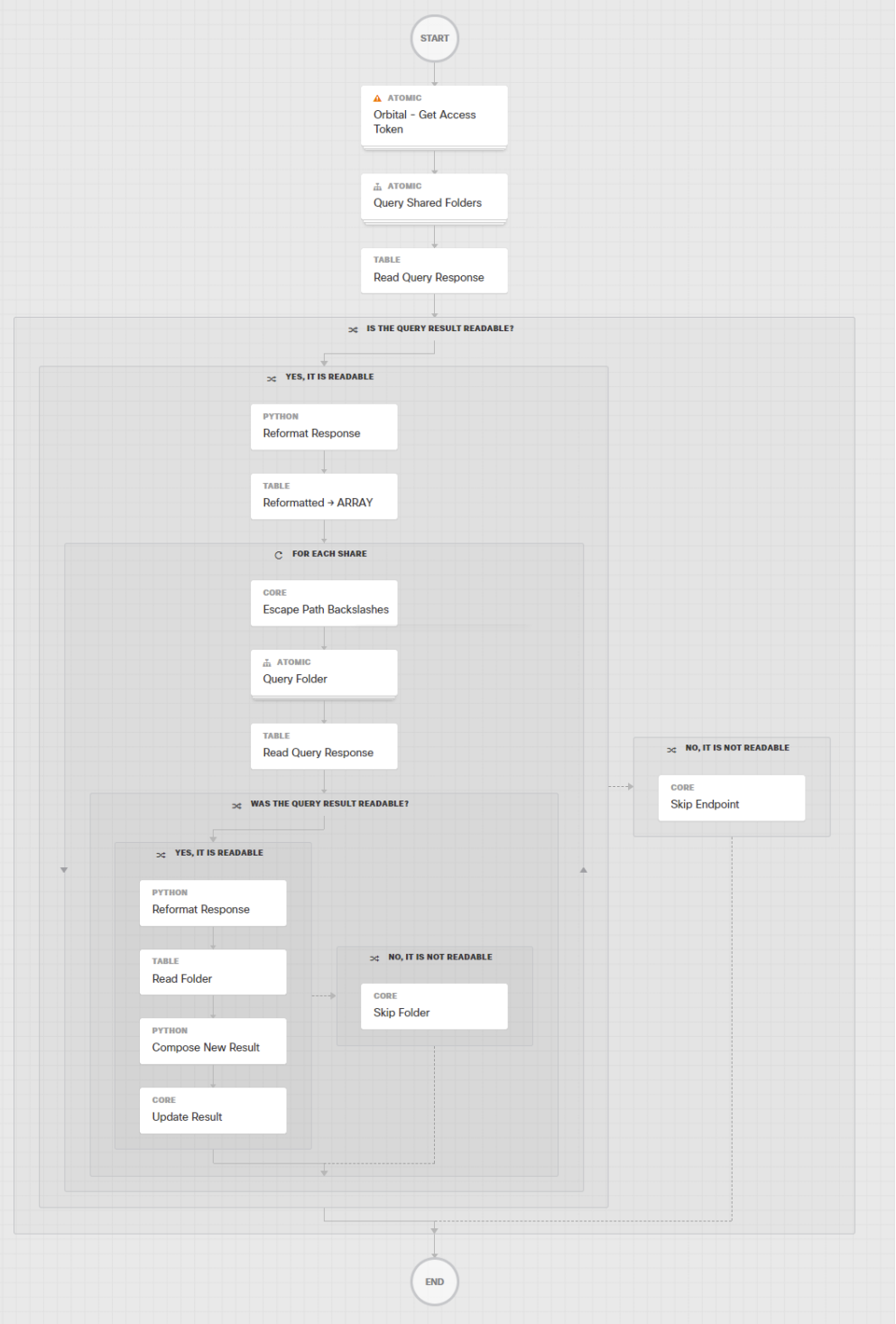

Figure 4.1.3. Cisco Orchestration Workflow: Query Endpoint

The workflow starts with requesting an access token for Cisco Orbital. This is needed for all Orbital interactions. Take note that this workflow is going to be used multiple times inside another workflow. I could have moved the access token outside of this workflow, however that would have impacted unit testing which I value highly. According to a number of test runs, the retrieval of an access token is fast, it does not impact performance enough to sacrifice the workflow's ability to be run independently. If speed is very important, a condition block could check for an existing (passed) access token from another workflow and get a new one if no token is present.

The main reason why this workflow is not an Atomic Action is because an Atomic Action cannot contain another Atomic Action. Regardless, the performance and behavior is unchanged, it would only matter in the sense of categorization.

According to the Cisco Orbital documentation[4], the `shared_resources` Osquery table contains information about Windows hosts' shared folders. In Windows, there are three default shares that I have excluded. These shares can pose serious vulnerabilities, but they should be monitored for other exploits. This workflow detects Lateral Tool Transfer, which is why it only requests shared folders. With these conditions, the Osquery SQL code becomes the following:

```
SELECT path
FROM shared_resources
WHERE name!="ADMIN$"
    AND name!="C$"
    AND name!="IPC$";
```

Cisco Orbital responds in JSON in a very detailed way so we need to discard unneeded information. The paths of shared folders is located at the JSON Path[39] `$[0].osQueryResult[0].values` of the full response body. Unfortunately, Orchestration Core components cannot deal with arrays of strings by default. As a workaround, I have used a Core component ("Read Table from JSON") to pre-sort the data for a Python script.

Before the Python script restructures the JSON, there is a condition block. If the pre-sorting activity raised any errors (which occurs when the data is `null`) the condition block does not execute further activities on the data. There is an important difference in the location of the error. By separating the table creation into a pre-sorting activity and a Python script, we can catch very specific errors:

- Unsuccessful querying activity: there was a problem while contacting Cisco Orbital. The workflow is marked as Failed and is stopped.
- The pre-sorting resulted in an error: the received data was `null` or equivalent. The workflow continues, skipping this endpoint.
- The Python script could not run: the data contained illegal characters or it was corrupted.

This is why we can use the “success state” of the pre-sorting activity as the condition block’s truth check value. If it is successful, the Python script begins processing the JSON. It repacks a simple array of single string values into an array of keyed (named) JSON objects that is compatible with Orchestration’s native components. The Python Script Module does not support exporting arrays to workflow variables yet, so the JSON string needs to be processed by another “Read Table from JSON” activity.

If an endpoint only has a single shared folder, the workflow could continue as-is, but to add support for devices with multiple shared folders, a For Each block is necessary. It takes a native Orchestration workflow array and runs the activities inside itself using the values of the array one-by-one.

With the values separated, we can start querying the contents of the folders using Cisco Orbital again. The access token retrieved at the start can be reused here. Unlike the first Orbital query, this time the SQL code is not static. We need to specify the path exactly. For this, the query is:

```
SELECT f.filename, h.sha256
FROM file f
    LEFT JOIN hash h ON f.path=h.path
WHERE f.path LIKE "${PATH}\\%%"
    AND f.filename!=".";
```

Where `${PATH}` is the value of the current For Each iteration. The Query activity accepts escaped special characters, and the escape character is backslash. This adds difficulty because Windows uses backslashes as separators so the correct escaping of Windows paths is done by doubling the backslashes. For this, there is a Core function: Replace String that does this.

After the query, the same procedure is used again, but this time the query returns two values (file name and hash) per file, which means every second value starts a new row. This is realized with a simple modulo operation, inspecting the remainder of the index of the data divided by the number of columns in the table, which is two in this case.

The newly formed JSON result string contains the queried folder’s files’ name and SHA-256 hash. This would be enough, but we are still inside a For Each loop. To my knowledge, Orchestration provides no tool for merging temporary tables. For this reason, I involved Python again to continuously update a “result string” that is an output of the workflow which was added exclusively for this reason.

Since the newly updated variable that should contain information about all shared files of a system is already an output variable, only the inputs remain unconfigured. For querying endpoints, Cisco Orbital requires the querying process to define the targets by either hostname, Secure Endpoint GUID, IP address, or MAC address.

I suspect there is a bug in the Query Endpoint atomic activity. It did not accept identification by hostnames without also specifying a MAC address. Although first, I believed any MAC address would work, so I used the address 00:00:00:00:00:00 because it seemed it was not influencing the outcome; later in my testing, I came across inexplicable errors that all resolved by themselves after specifying the correct addresses. For this reason, I had to make two input values for this workflow:

- Hostname: the hostname of the target endpoint.
- MAC address: (optional) the MAC address of the target. If unspecified, it defaults to a valid MAC address. Should be removed when the Query Endpoint atomic activity works as intended as a result of an update.

With these parameters, the workflow is unit testable (can be used independently of other workflows) and it can be used as an atomic action in another workflow. This workflow only queries a single endpoint, but Lateral Tool Transfer is recognized when multiple endpoints have the same file in their shared folders. This workflow will be used to elevate abstraction to reduce other workflows' complexity.

4.1.2. AGGREGATING QUERIES

A workflow that aggregates output from shared folder queries is a relatively error-free idea. All errors are handled inside the sub-workflow that carry out shared folders queries. Neither Cisco Secure Endpoint nor Cisco Orbital supports fetching computer details by Secure Endpoint groups or policies. To get the computer details, we need to request all known computers, then filter by group GUID. This will be one of the input values for this workflow. The group GUID will let the workflow know which groups of computers need to be checked.

Cisco Secure Endpoint (formerly called AMP) API lets us get all known computers. It is done through basic HTTP authorization that is natively supported in Orchestration. To filter out unneeded computers, a Core "Read Table from JSON" is used with a special JSON Path query:

```
$.data[?(@.group_guid =~ /${GUID}/i)]
```

This JSON Path query only selects the computers with the appropriate Group GUID if `${GUID}` is substituted correctly. The AMP API's response is structured in a way that makes fetching the MAC addresses impossible in one "Read Table from JSON" action. The MAC address is on a different level (nesting) of the JSON objects than the hostname, moreover it is inside an array that makes selection even harder. For this purpose, I involved Python again. The Python script does not add the MAC addresses to the existing array of hostnames because any interaction in Python would make the array lose its array properties inside Orchestration. Instead, the script compiles a dictionary of hostnames to MAC addresses for later use.

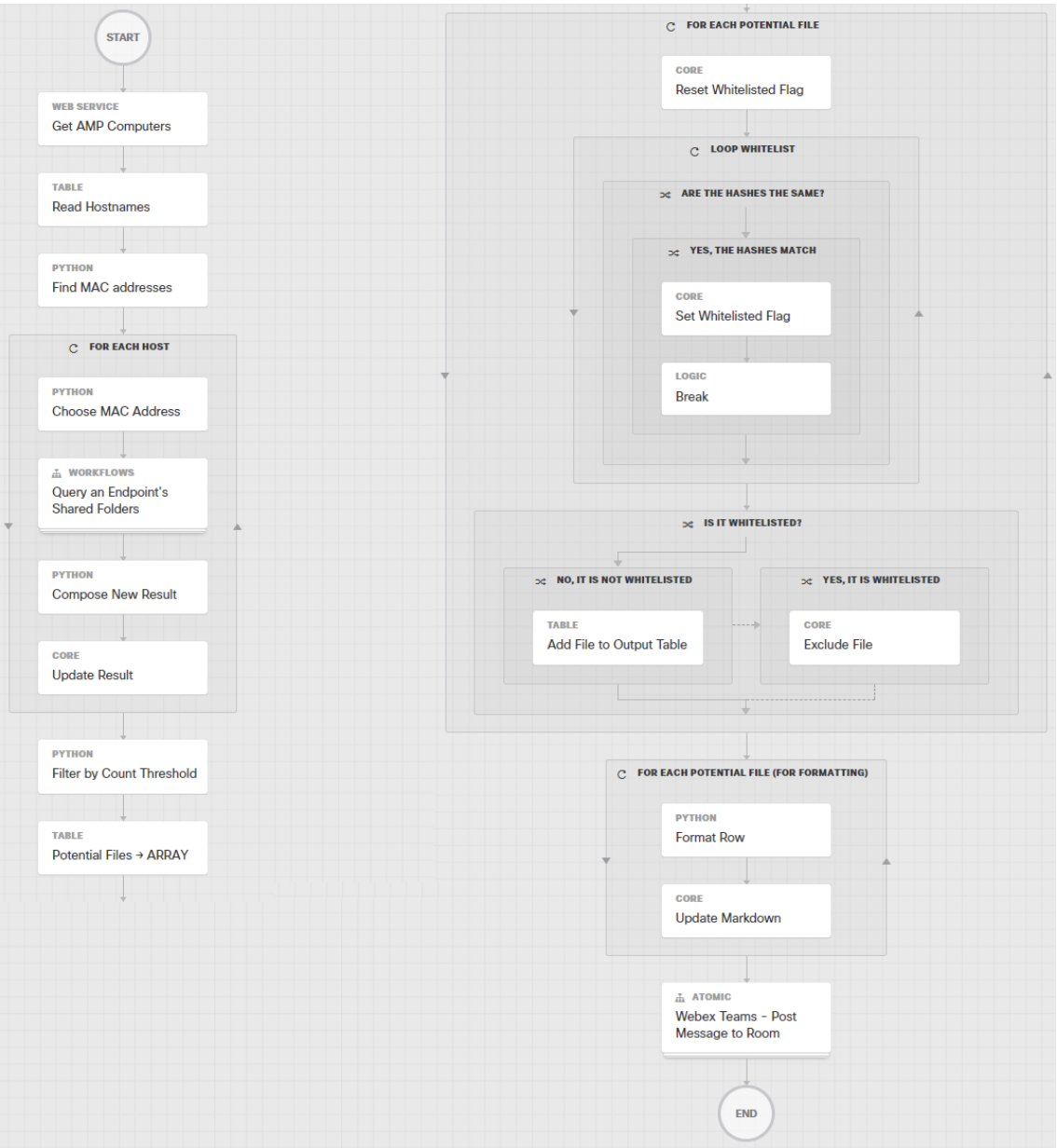

Figure 4.1.4. Cisco Orchestration Workflow: T1570 Aggregator (split)

Once everything is set up, a For Each block begins with the gathered hostnames. The very first action is a Python script that looks up the needed MAC address by the current iteration's data which is the hostname. With this simple trick, we have both the hostname and MAC address of the endpoint we want to query, which is all requirements of the Cisco Orbital Query workflow detailed in the previous section. To merge the results, the same process of "result strings" is used. A small modification is that a third value is added to the temporary results table, which is "count." This new column keeps track of the number of endpoints this file has been found on.

At this point, the workflow functions as a map of all files on the union of shared network space. To make it a Lateral Tool Transfer detection, a threshold of maximum file duplicates is essential. This is the other input value for this workflow. The operation is done with a simple Python script checking if the “count” column of each row is high enough to be kept. If it does not meet the criterion, the row is simply dropped.

If we remove items that are also in a globally defined whitelist, we only have potentially harmful files left. The easiest method I found to exclude files present on the whitelist was a Boolean flag system in a double-loop: at the start of each loop, the flag is set to “false” then the whitelist is looped. If the inner loop has the same file hash as the outer loop, the flag is set to “true.”

After the inner loop ends, a logical condition checks whether the flag is “true” or “false.” If it is “true,” the next item in the outer loop follows. If it is “false,” we add the file to a persistent table created at the beginning of each run of this workflow. All three columns are transferred to the new table: the file hash, name, and count across devices.

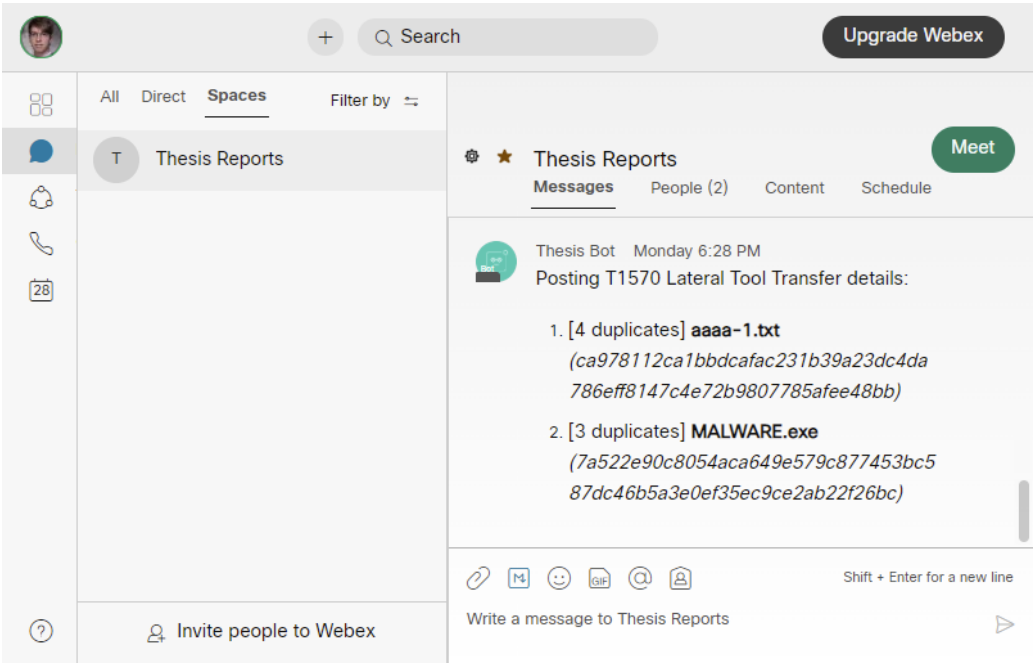

Figure 4.1.5. The Webex Teams chat bot reporting non-whitelisted file duplicates.

Keeping the table in the system is not verbose enough. For this reason, I added a chat bot interaction at the end of this workflow that messages a Security Operator the potential hazard. The message is formatted using a For Each block and Python scripts that convert the rows to the Markdown syntax to be easily readable. The final activity launches the message with the formatted content and the appropriate parameters. Figure 4.1.5. shows the chat bot in action. In the example, three files exceeded the maximum number of allowed duplicates, but one was already whitelisted. The message tells the operator some details about the new excessively duplicated files.

4.2. T1091 REPLICATION THROUGH REMOVABLE MEDIA

Technique code T1091 is Replication Through Removable Media. This technique is a Windows-only procedure with no sub-techniques. Although the current version of T1091 in the MITRE ATT&CK Matrix is specified to only work on Windows, any platforms are vulnerable to this exact threat type, for example some GNU/Linux Desktop Environments can offer Windows-like automatic program execution on device mounts.

Although AutoRun, the feature that automatically executes a program when a new device is connected, is still present in modern versions of Windows, it has been replaced with AutoPlay on non-optical drives starting from Windows 7. The main difference is that when a device is connected for the first time, AutoPlay displays a prompt of actions that can be performed every time the same device (or same device type) is connected. AutoRun is still an option in the AutoPlay menu, if a file named autorun.inf is present on the connected device. Once an option is chosen, it is saved for future connections of the same device type. If AutoRun is the default option for USB storage sticks with AutoRun capability, there is a possibility of malicious code execution without any user prompts.

The solution I am proposing uses the built-in Windows Task Scheduler and Python to contact SecureX for threat intel. The workflow responsible for handling the requests in SecureX Orchestration then automatically isolates the host if a known malware entered the system or creates a task for the manual reviewal of an unknown automatically run program.

4.2.1. SETTING UP THE END DEVICES

The solution would have been possible only using Cisco SecureX and Cisco Orbital queries, but I value speed in detection, so Windows's native Task Scheduler was used as the trigger. Task Scheduler supports custom user-defined tasks, but its functionality to run a task based on a system event was the main reason of choice.

In the Windows Event Viewer, the event log under "`Applications and Services Logs\Microsoft\Windows\DriverFrameworks-UserMode\Operational`" records the USB connections among many other initializations, but it is disabled by default. To attach a task to an Event Viewer record type, we need to enable logging. If space is crucial, it can be kept to just a few kilobytes, but this is not a concern during this work.

I chose Python as the script language for the clients (end devices) because it is used in the SecureX workflows as well. Using other languages would not have been impossible, in fact, all communication is handled using JSON, therefore any language supporting operating system drive discovery would suffice.

SecureX supports running its Orchestration workflows by remote requests. There are two main ways to launch workflows externally: via the Threat Response API or using

webhooks. I chose the Threat Response API simply because I managed to get it working before webhooks. Traditionally, Threat Response API workflow triggers are made for simple response actions, like isolating hosts, reporting malicious observables, or logging. In every case, it only supports one observable per triggered workflow. In my case, I wanted to make it a one-step procedure because I value time in threat detection so it can become prevention. For this reason, I had to create a workaround solution that allows sending JSON data as "a single observable," except it contains multiple observables.

The task I set up in the Task Scheduler is bound to trigger when an "Event ID 2004" is logged in the aforementioned Operational event log. It is logged once every time a new device is connected. This includes hot swappable SATA/SAS drives as well as USB devices. The event is so broadly called that even connecting through Remote Desktop Protocol leaves an event with the identifier 2004. I consider this a plus because the deployed Replication Through Removable Media script will run every time a new potentially malicious device is connected, physically or even virtually.

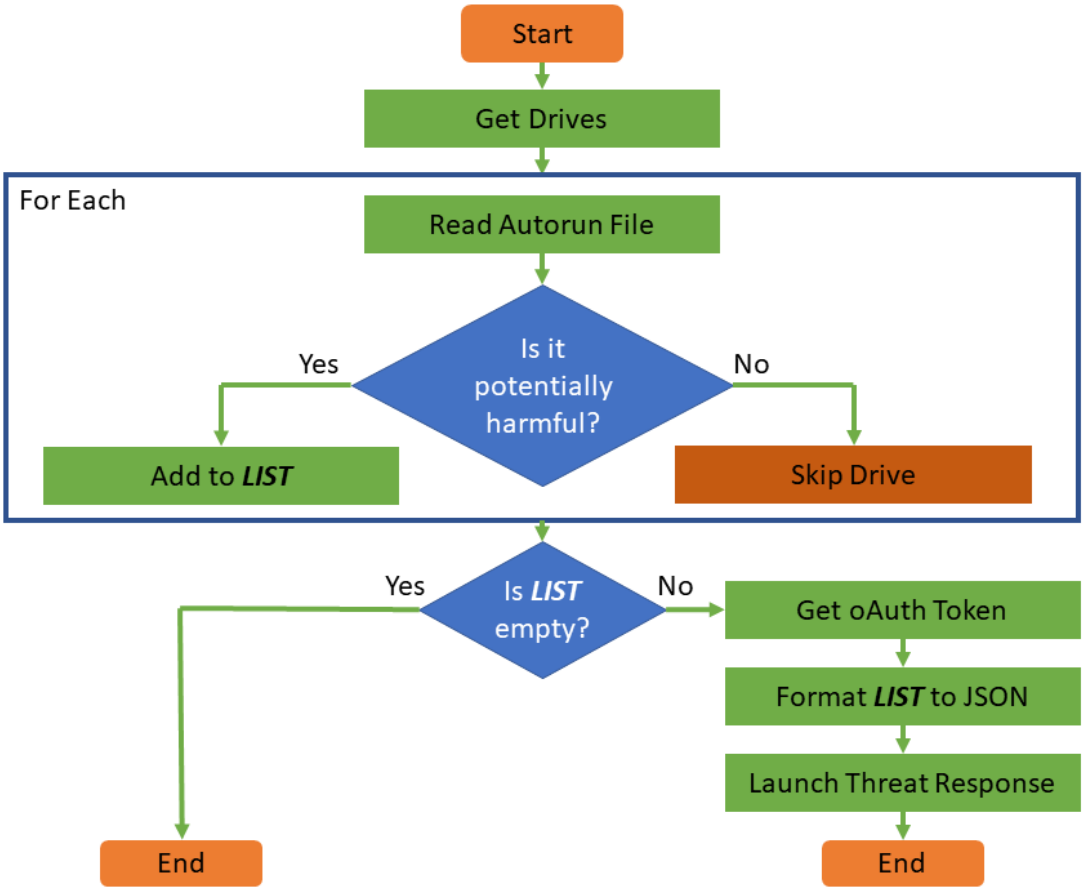

Figure 4.2.1. Simplified Flowchart for the End Devices' Python Script

There has to be an action when the task is triggered. The action is set to “Start a program” and it points to a Python script file. The reason why the Python script runs correctly when it is “started” is because Python 3.10.0 is installed on the system and files ending in .py are associated to be interpreted using Python’s interpreter executable.

The script uses the `kernel32` module of `windll` from the `ctypes` library to request all logical system drives, then for every drive found, it checks for a file named autorun.inf located at the root of the drive. If a drive has the AutoRun file, it reads the file and looks for options “open” or “shellexecute” because these two options are capable of starting executables. Any AutoRun file with a statement to execute a file then will be further investigated by looking up the file that the statement points to. If Python finds the file, it will hash the raw contents with SHA-256, producing a valid digest, also known as checksum or simply (file) hash.

Once the script has finished searching for files, a bundled report is created in JSON format (using Python’s dictionary structures) and if the report contains at least one digest, it connects to Cisco Threat Grid first to get an OAuth 2.0 access token, then using the received token it invokes the workflow via a second request.

To specify which workflow should be run, there is a dedicated workflow in Orchestration. It lists all workflows’ Action URLs that are tagged with the “response” category. Response workflows can only have two inputs: `observable_type` and `observable_value`. The type must be one of the pre-defined observable types that Cisco’s systems synonymously understand. My choice was the URL type observable because its checks are lenient enough to allow a Base64 encoded JSON document, and even if stricter checks would be enforced in the future, an URL’s query string will genuinely allow Base64 content as it is widely used on the World Wide Web.

The Anatomy of the JSON structure is as follows:

```
{
    "guid": <GUID of the AMP Connector>,
    "host": <Hostname of the Computer>,
    "drives": [
        {
            "letter": <Drive Letter>,
            "drive": <Drive Path>,
            "contents": <Contents of the AutoRun File>,
            "file": <Name of the File>,
            "hash": <SHA-256 Digest of the File>,
        },
        ...
    ]
}
```

The difference between the drive letter data value and the drive path value is that the letter is only the alphabetical “logical” drive letter (can be empty) while the drive path is the fully qualified path to the new drive. A common situation when they differ is when removable devices are set up to mount to an NTFS folder such as “C:\mount” instead of the default Windows alphabetically increasing scheme.

4.2.2. ATOMIC ACTIONS FOR HASH CHECKING

We need some ways to verify whether the file that is going to be AutoRun is malicious or not. Sadly, there is no way to know if a file is perfectly safe, but I chose two services that offer file details by submitting file hashes. First, a module of SecureX, Cisco Secure Malware Analytics (formerly known as Threat Grid) is readily available from the SecureX dashboard, and it offers API interconnection, which is how Orchestration can use it. The second service I chose is VirusTotal[40]. It offers a “community” option when registering that is free of charge and it grants access to the very basics of their threat intelligence. I could have fit everything into a single workflow but utilizing the Atomic Action feature of Orchestration can make querying APIs like the two services I chose really modular. With modularity, quick addition of new APIs to check against should be easier.

Starting with the Atomic workflow “Get Highest Threat Grid Threat Score,” it asks Cisco Secure Malware Analytics to return every known file that has the SHA-256 digest specified in the input of the workflow. The information we get about the returned files is mainly about behavior analytics, but without complex analyses, we are also presented with a “Threat Score.” This score shows how likely it is that the file is malicious from a scale to 0 to 100.

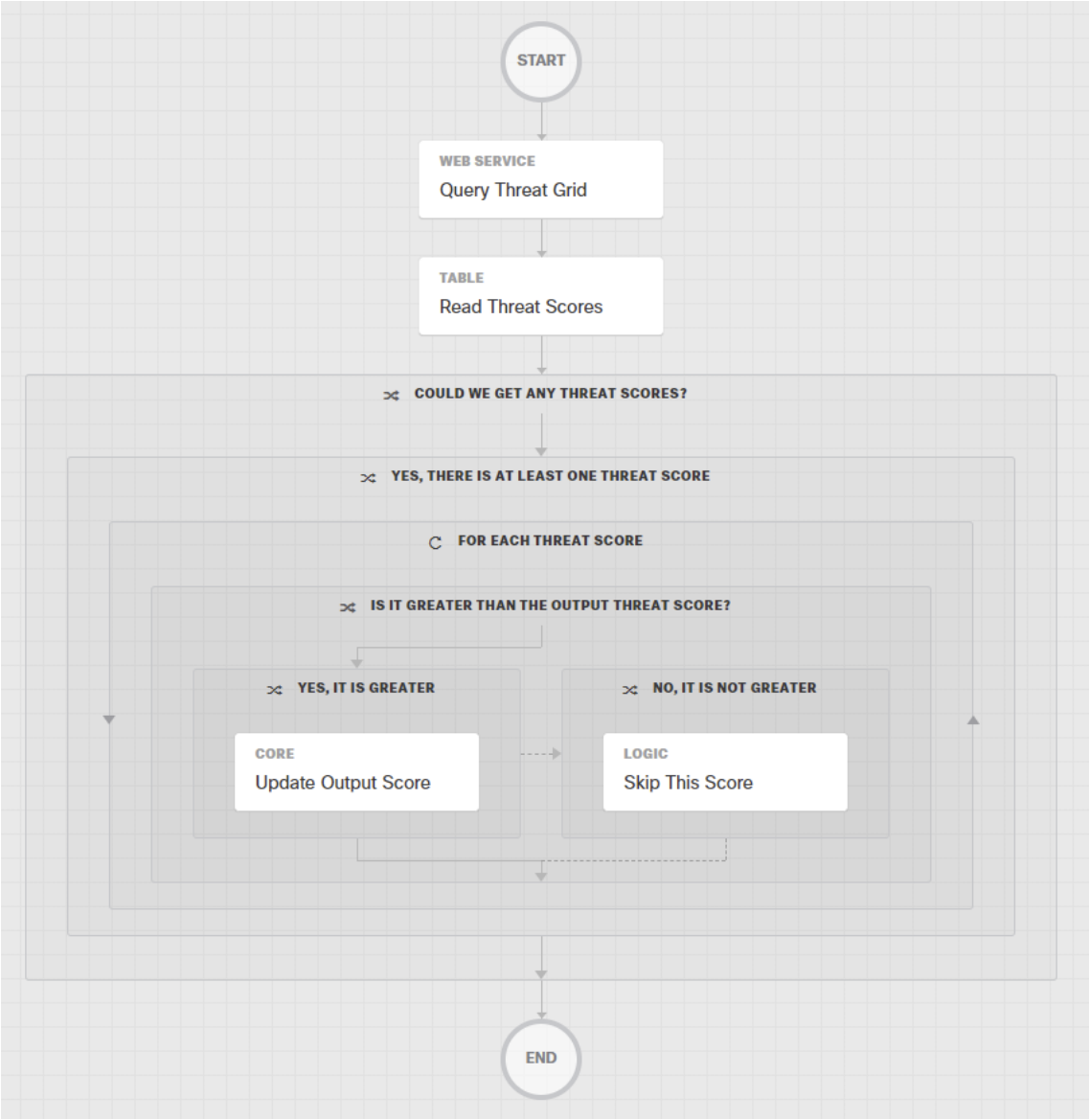

Figure 4.2.1. Cisco Orchestration Workflow: Get Highest Threat Grid Threat Score

As it is shown in Figure 4.2.1., there is a For Each block to go over each file that had the same SHA-256 checksum. The output is zero by default, then it is updated by every higher Threat Score found in the matched files. At the end, the highest will be the output.

The workflow does not stop on errors, it will return the output value zero if the hash was not found in the system or if there was an error of unknown origin. One of the drawbacks of this design is that we do not know whether it succeeded or not. A second Atomic Action will help ruling out occasional downtimes of this service, but it will also broaden the search space. There is no single threat intelligence database that has encountered each and every iteration of all malwares in the world and analyzed it in their ecosystem.

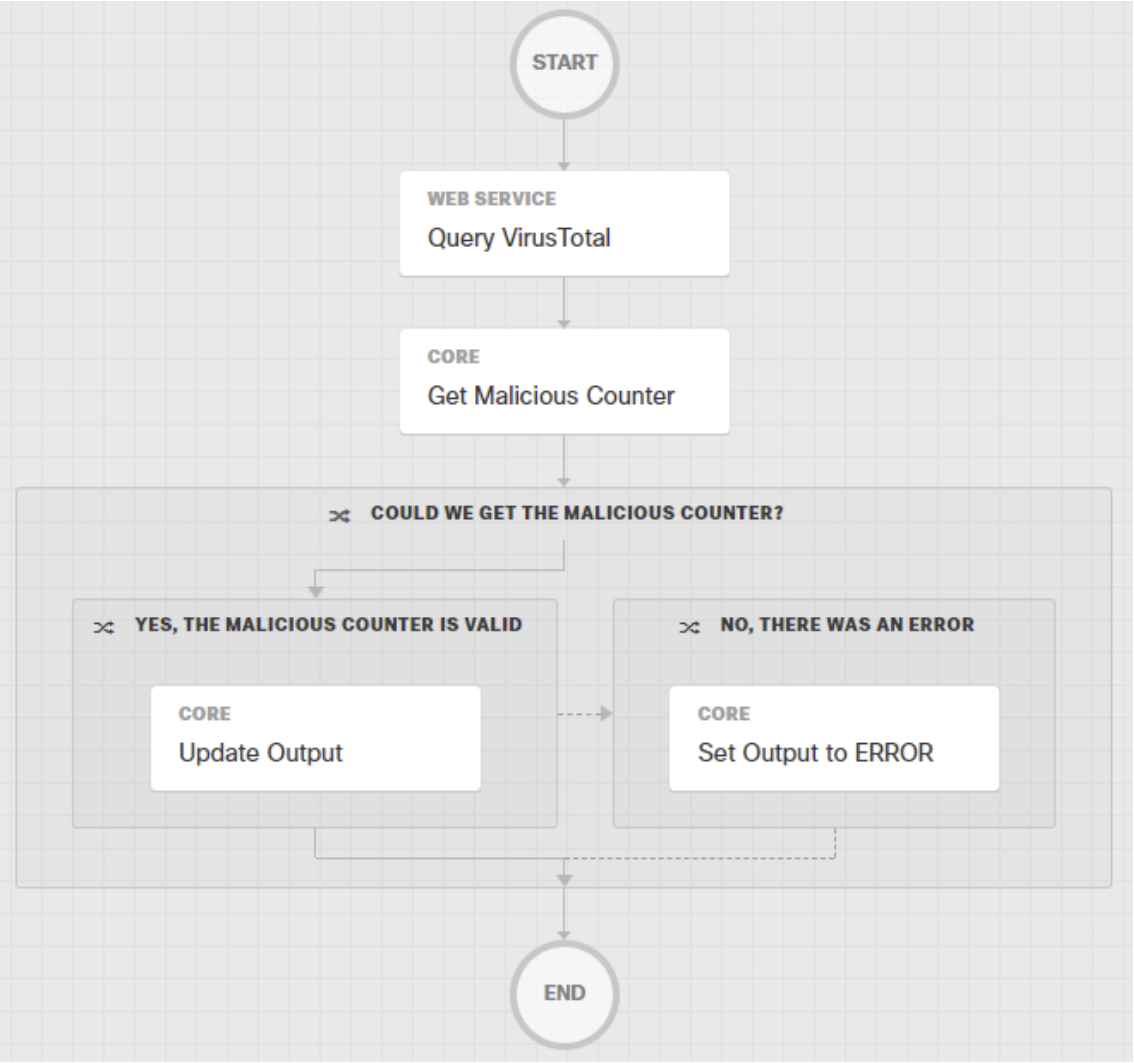

Figure 4.2.2. Cisco Orchestration Workflow: Get VirusTotal Malicious Counter

In the second workflow, we have the same input and output variables. The output variable is called “Malicious Counter” this time, because VirusTotal has a different approach. When a file is identified (by uploading it, specifying hashes, or any other supported means) it presents results of partnered antivirus services, it does not produce its own verdict.

The first Action asks VirusTotal via their HTTPS API, however SecureX does not support VirusTotal’s API authentication in the Account Keys component. By specification, a custom HTTP header must be sent with the key “x-apikey” and the header value being the API key tied to the VirusTotal account. This does not mean that SecureX cannot be used to access this API but rather we have to use an extra element with a bit more configuration than Account Keys. SecureX’s “secure string” data type allows us to store the VirusTotal API key securely and in the global scope of SecureX. This way, if another VirusTotal workflow is made, the same API key can be used and then it can be

simultaneously updated if needed. There is a Core Component called HTTP Request that can accept arbitrary headers like VirusTotal's "x-apikey" condition. An HTTP Request combined with a Secure String can perfectly access the API without any problems.

The response will be a JSON document filled with results from different modules. Luckily, it includes a combined statistics list at the JSON Path:

```
$.data.attributes.last_analysis_stats
```

The list at the JSON Path above provides an overview of verdicts categorized harmless, invalid, malicious, etc. In our case, we need the malicious data node, so the Path query needs to be extended with: `.malicious`

In contrast with Threat Grid's Threat Score, this Malicious Counter shows how many analyzers deemed the file malicious as opposed to giving a score of likeliness.

4.2.3. THE MAIN WORKFLOW

In the main workflow, the input values are restricted to `observable_type` and `observable_value` because of the "response" category.

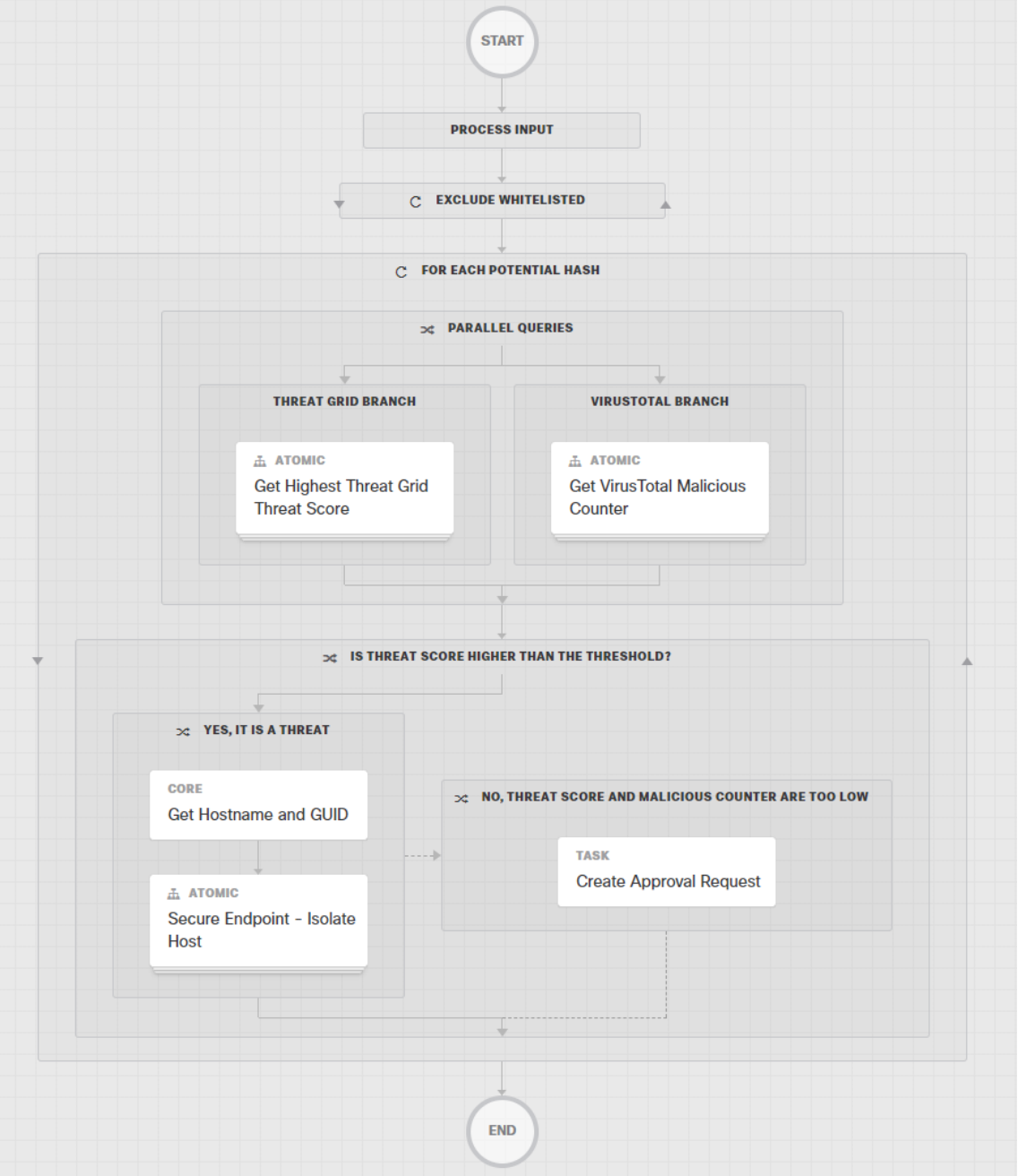

Figure 4.2.3. Cisco Orchestration Workflow: Replication Through Removable Media

The first few Actions are hidden in a collapsed group in Figure 4.2.3. because the figure would have been too tall. The Actions inside it are the reverse of the end devices' Python script's encoding. It reads the received data into a table recognized by SecureX. The next group, a For Each block is collapsed in the Figure as well, because it is the exact copy of

the block that excludes the whitelisted values from the T1570 Lateral Tool Transfer Aggregator workflow.

The remaining hashes are looped in a For Each block. In each iteration, the Parallel block is used to make the response faster, because as I have stated, I value time in detection. Also, this Parallel block serves as an easy-to-upgrade point because a new branch can be added with another Atomic Action that consults another API or integrated module.

As soon as each branch in the Parallel block return, the evaluation begins: an adjustable (local) threshold is set in the workflow for both results. I believe everything above the Threat Score of 90 or the Malicious Counter of 2 should be dealt with as soon as possible. The Condition block is set to run the left “Yes, it is a threat” branch if either value is greater or equal to the threshold specified. The branch first gets the Hostname and GUID of the AMP Connector from the original response, then makes an isolation request to Cisco Secure Endpoint to isolate the machine to prevent any further complications.

If neither the Threat Score nor the Malicious Counter are high enough, the workflow creates an approval request. This feature might not be needed, but detection based on file hashes alone cannot be perfect, if an unknown AutoRun file was found on a device, it should be manually assessed. The created approval request offers a quick way to add the hash into the whitelist, which brings us to the next point.

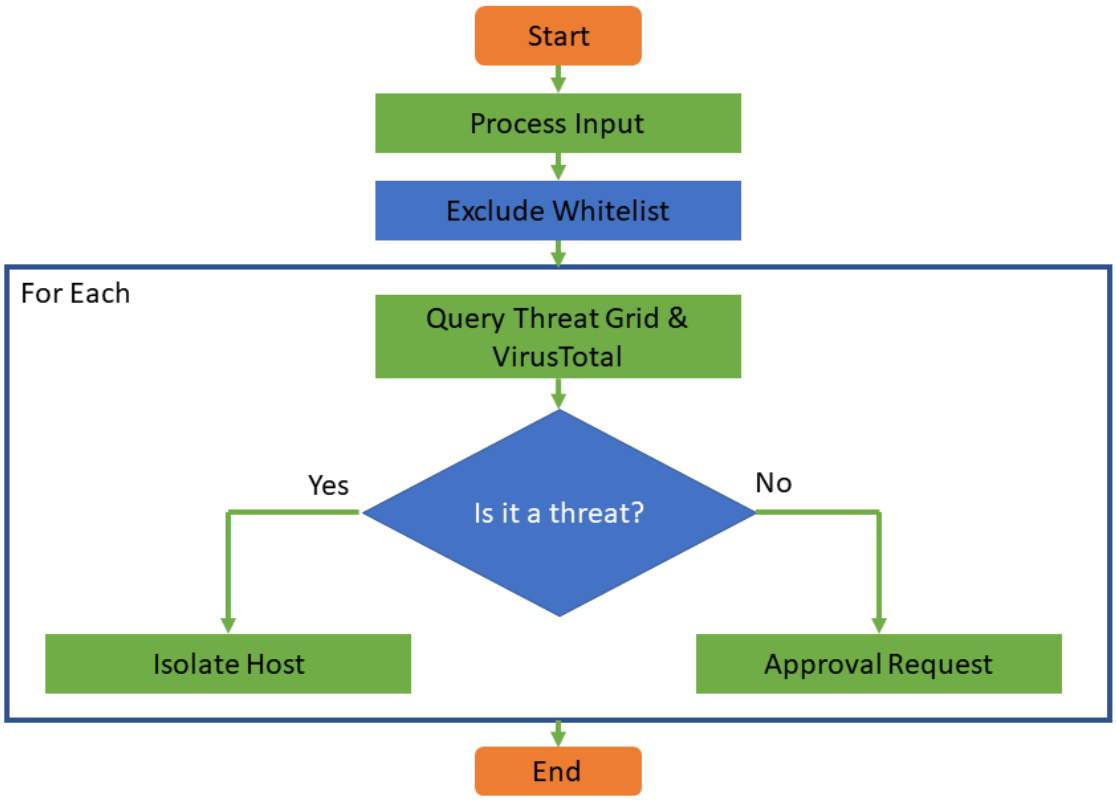

Figure 4.2.4. Simplified flowchart for Replication Through Removable Media

4.2.4. WHITELISTING ALLOWED FILES

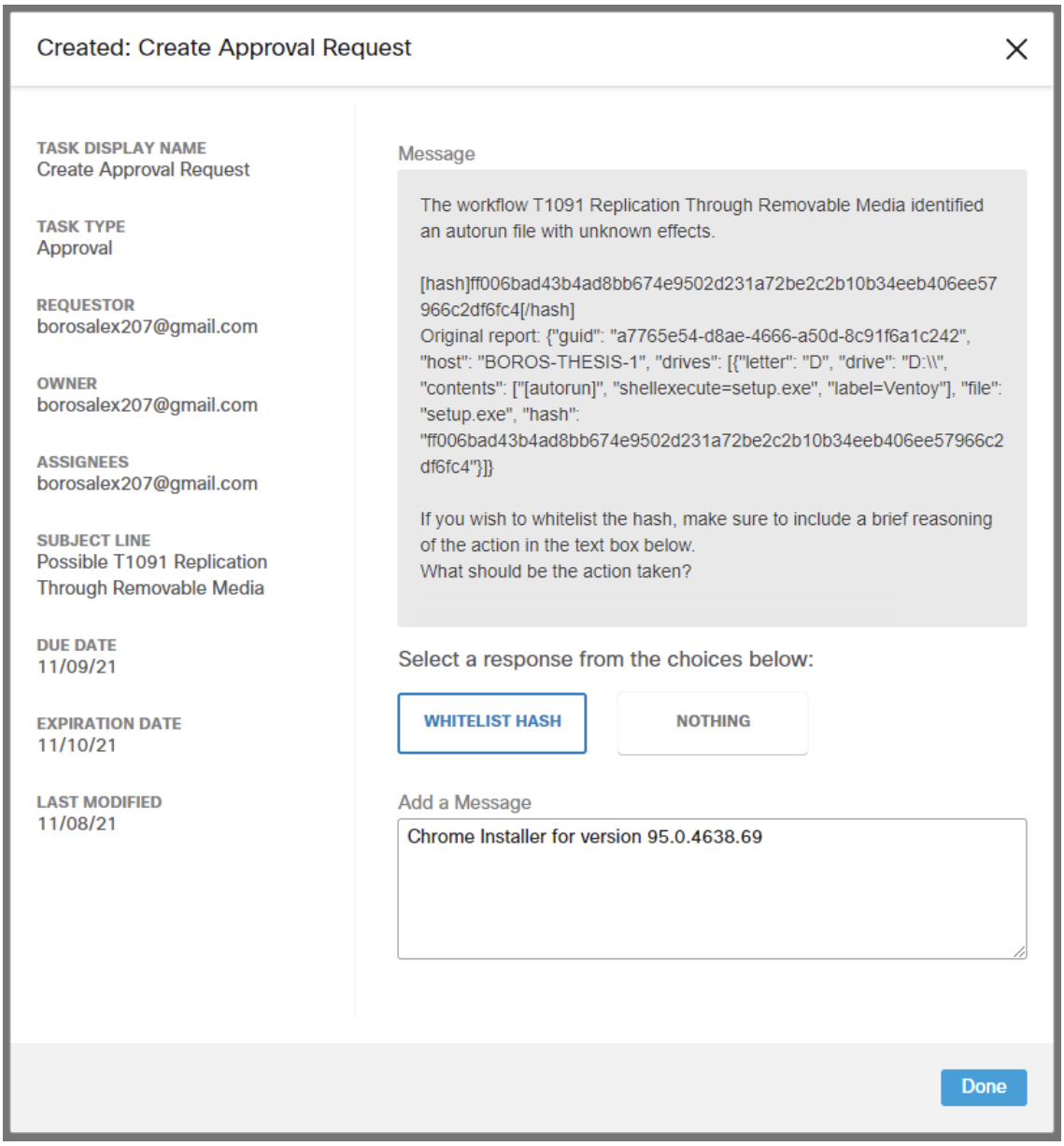

Figure 4.2.5. Cisco Orchestration: Approval Request Task for T1091

The trigger of the last workflow in this technique runs when a task is completed with a "Whitelist Hash" choice and an added note. It only runs if the Subject Line (the 6th block on the left-hand side of Figure 4.2.4.) matches "Possible T1091 Replication Through Removable Media."

The workflow is so short that it can be explained easily without a figure. It first gets the original "Message" (description) of the task and finds the hash between the [hash][/hash] tag. Then it adds the user defined "Add a Message" field's contents along with the found hash to the whitelist table.

4.3. T1021 REMOTE SERVICES

Adversaries may use remote services such as Remote Desktop Protocol (RDP) or Server Message Block (SMB) to move laterally once they gained access of a device from inside the network. This technique closely resembles two other techniques with minor differences. T1021 Remote Services (this technique) builds on initiating new connections through remote services by completely ordinary means, as if a user did so. T1563 Remote Service Session Hijacking works by keeping open already connected sessions, then spreading or executing malicious code over valid sessions (but no longer in control of the original owner) after they are thought to be closed and fully terminated. T1210 Exploitation of Remote Services has the same effect of this technique, but it relies on exploitable entrances and bugs to gain unauthorized full access without being officially privileged.

My method of detection can point to all three of these techniques. While early signs of an upcoming attack might be port scanning as indicated in technique T1210[41], leaving an RDP or SMB connection open for more than twice an employee's shift time is unreasonable. Even if the person works overtime, there is no reason to keep a remote controlling connection alive if no one uses it.

4.3.1. ORCHESTRATION WORKFLOW

I could organize everything into a single workflow, but there is a tradeoff. The workflow relies on a chain of Python script, which is not encouraged. If the SecureX core components change, the Python scripts might break, and the other core components would be automatically updated to stay compatible. For this reason, I avoided using too long Python code actions, but the built-in actions approach would have been harder to explain and too large to show on even two pages in this thesis.

This workflow reuses elements from the previous two techniques' workflows, which I have organized into groups and collapsed them to be simpler in Figure 4.3.1. It starts with the familiar "Get AMP Computers" that is an exact copy from the Lateral Tool Transfer workflow. I could not make this an Atomic Action because outputting arrays that can be used by the following For Each block is not supported. Instead of making an Orbital sub-workflow or Atomic Action, I put an action to the beginning of the workflow to get an Access Token, then I used a "Query Endpoint" Action in a For Each block to query the computers one by one. I felt like refactoring this to a separate Atomic Action would have complicated and slowed the process, because the Access Token would have been requested every time an endpoint is queried. On the other hand, passing the Access Token via an input field would have made no sense, because factoring out a single block is nothing but an extra layer of complexity with no added benefit. Simply put, the very same behavior is achieved without using extra Atomic Actions.

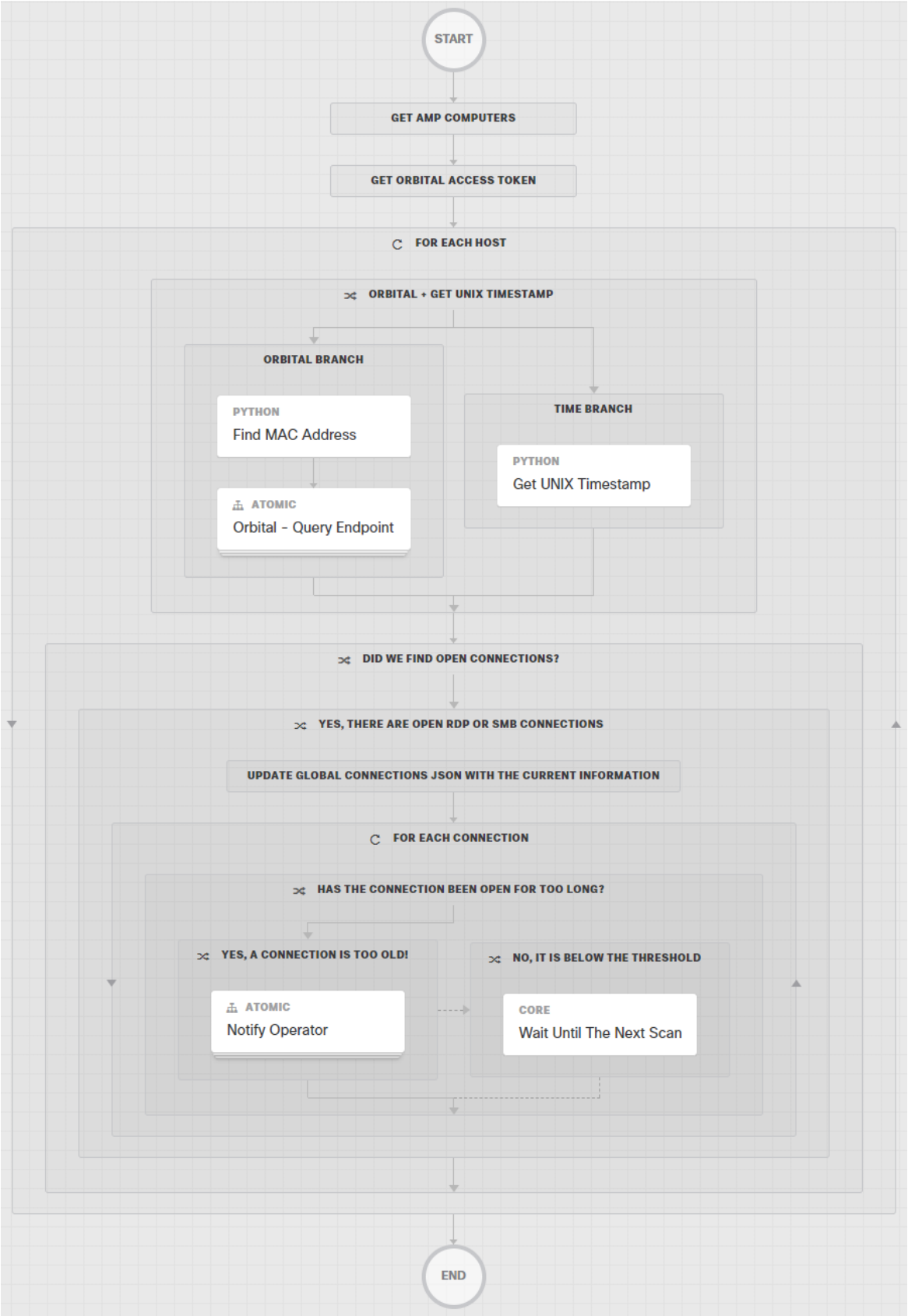

Figure 4.3.1. Cisco Orchestration Workflow: T1021 Remote Services

I decided to readjust the UNIX Timestamp every time a Cisco Orbital query is started. This is to ensure that if hundreds of computers are queried one at a time, the hundredth computer's connection times are still accurate to the minute. This is realized with a Parallel block to save time while the inherently longer query action is running. The locally calculated timestamp should always finish before an external API call concludes.

The query uses the `process_open_sockets` Osquery table and its `remote_address` and `local_port` columns are selected where only the SMB and RDP Transmission Control Protocol connections are shown. The next string of Python scripts are grouped together because they form a single logical unit, but it is made to be somewhat modular for easier adjustments in the future.

If the query returns at least one row, it is processed in the first Python script in the group similarly to the previous solutions but without pre-sorting. This means that the raw response of Orbital is taken as the input and only a table of open connection is the output.

The second Python script takes a global (string type) variable: Remote Services Connections JSON. It updates the variable to match the real open connections and also provides a report of open connections and the time for which the connections have been active. A third action saves the Python scripts' work, then a fourth action transforms the last script's output to an array, preparing it to be looped over in the next For Each block.

It is important to note that the Connections JSON is only updated for the current item of the loop, therefore the global variable gets updated multiple times in a row. This could only pose a problem is the workflow is run too frequently that the last running version is still modifying the global variable. If this happens, the connections report could result in false information for the duration of the two intersecting runs.

To counter this issue, the Orbital query is set to time out in one minute, which accounts for a minute per computers in the AMP Group queried. This alone is not enough if the workflows are launched fast enough in succession, which is solved by creating a schedule that runs every N+1 minutes where N is the number of computers. The extra minute is needed for the non-query awaiting actions. Of course, it is good practice to leave some headroom in case a new device is introduced to the group without any notice. My example uses 4 computers and runs every 60 minutes, which is more than enough in provision, but in case of a 100-computer environment, at least 101 minutes intervals would be needed to operate fully safely.

The For Each block contains a single Condition block with two branches. The condition for choosing a branch is whether the connection time of the live iteration of the For Each block is more than the threshold configurable in the workflow's local variables.

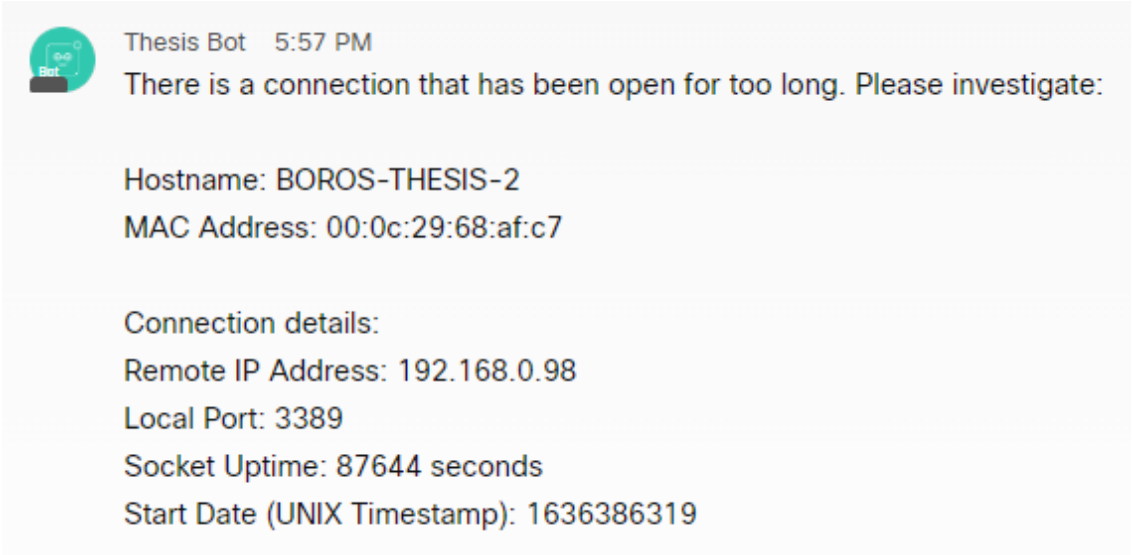

Figure 4.3.2. Webex Teams chat bot notifying an operator of a very long RDP session

If it exceeds the set threshold, it notifies an operator via a Webex Teams message containing detailed information as visible in Figure 4.3.2. As this is detection only, I reckon a request for manual appraisal reasonable.

If the connection has not been open for long enough, the workflow will simply wait for the next execution cycle. It is set to execute on a schedule that runs hourly. The default threshold that I set for the workflow is one day, which means that the latest a too long open session is discovered would be open for 25 hours at the point of detection.

5. TESTING DETECTION

5.1. TESTING LATERAL TOOL TRANSFER (T1570)

The passive testing method for Lateral Tool Transfer is simply duplicating files by hand. These are highly controlled synthetic situations, but just like Test Driven Development, we have our conditions arranged based on MITRE ATT&CK:

- Overpopulation of duplicate files on a single host should not count.
- Files with the same name but different contents should be separate.
- Files with the same content but different names should be treated as equals.
- Using any combination of all above statements, a detection must still be possible.

For a semi-automated test, a self-copying program written in C# was used. I wrote it because any programming language can simply perform Lateral Tool Transfer given the right conditions. It has an advantage of knowing the exact network topology and the available shares. It does not execute on its own but opening it on one end device should "infect" the others by copying itself to them. It relies on stored credentials or users having writing permissions in others' shared folders. Here is the code:

```
string from = @"C:\company-shared\c-sharp-target.exe";
for (int i = 1; i <= 4; i++)
{
    string to = @"\\BOROS-THESIS-"
        + i
        + @"\shared\c-sharp-target.exe";
    File.Copy(from, to, true);
}
```

The tests have been carried out with success; each test passed. A unique flaw in the system was discovered which is that the workflow only works if all computers in the AMP group have at least two shared folders. Devices with only one shared folder are ignored. This is caused by an error in the pre-sorting model, which can be fixed by merging the whole response analysis into the Python script, like in the T1091 Remote Services workflow.

5.2. TESTING REPLICATION THROUGH REMOVABLE MEDIA (T1091)

Since Replication Through Removable Media is a simple Windows specific threat, it allowed me to create my own AutoRun files to run programs. I settled on running these tests:

- Connecting a USB storage device with a known malware in its AutoRun.
- Connecting a USB storage device with a known safe file in its AutoRun.
- Connecting a USB storage device with a missing file where AutoRun points.
- Repeating the steps above with AutoPlay set to execute-on-connect AutoRun.

All tests passed. There had to be adjustments in the AMP Group policy: all programs that were about to start would be scanned by Cisco Secure Endpoint, hence failing the later tests due to Secure Endpoint intervention. I temporarily set the Conviction Modes in the policy to Audit instead of Quarantine/Block/Protect.

The known malware executed on the system was rensenWare, a ransomware that asks for game competition opposed to payment. The system was isolated before the program finished encrypting the documents on the infected machine, but rensenWare does not try to spread on its own, it is not indicative of direct success.

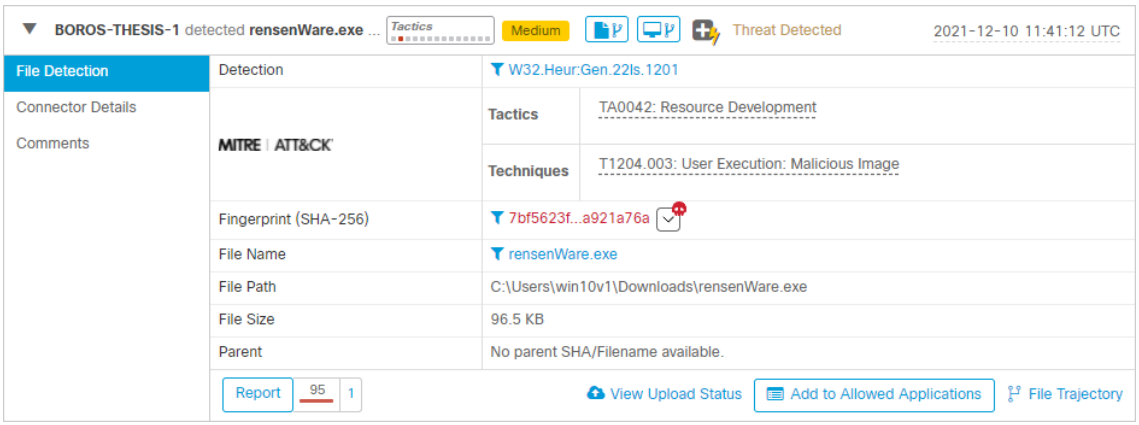

Figure 5.2. Cisco Secure Endpoint: rensenWare detection

It is worth noting that when the policy was set to Quarantine mode, the file hash was still submitted to SecureX before Cisco Secure Endpoint could remove the problematic program. This means that the Python script tied to the device initialization event runs very early, but late enough to see the file before it is executed by AutoRun therefore being deleted by Secure Endpoint.

The safe file was more complex than random bytes in a text file. I decided to run the test with the Google Chrome Installer. I believe it can be flagged as malicious if only the raw behavior is examined: it tries to elevate its privileges to an administrator level, then it extracts files from itself to the system's program files and application data folders. The workflow ended with an approval request that can be seen in Figure 4.2.4. This means that the threat intel services used could not confirm its malicious intent, or in other words, it was deemed as safe.

5.3. TESTING REMOTE SERVICES (T1021)

The case of our detection deployed for T1021 Remote Services is pretty straightforward:

- An RDP connection being kept open long enough should notify the operator.
- While it is open, it should notify the operator about the connection every time.
- If the session is closed even for just one run, the counter should reset.

The first result can be seen in Figure 4.3.2. then the same message (with increased uptime counter) was repeated two more times before closing the RDP session. Once closed, the messages stopped. Reopening the RDP session did not cause the messages to reappear instantly, it started sending messages after the configured threshold. This concludes all tests, all of which passed.

5.4. TESTING WHITELISTING

Both Lateral Tool Transfer and Replication Through Removable Media feature a whitelist to offer a sensible option of excluding known harmless files. First, rensenWare was added to the list, and neither workflow continued the warnings on finding too many copies or finding an AutoRun pointing to it. Using the task from Figure 4.2.4., the Google Chrome Installer was also whitelisted. After that, no new approval tasks were generated when an AutoRun was pointing to the Chrome Installer executable. Copying it to each and every shared folder raised no warnings, which means everything works as intended.

5.5. DETECTION MAP

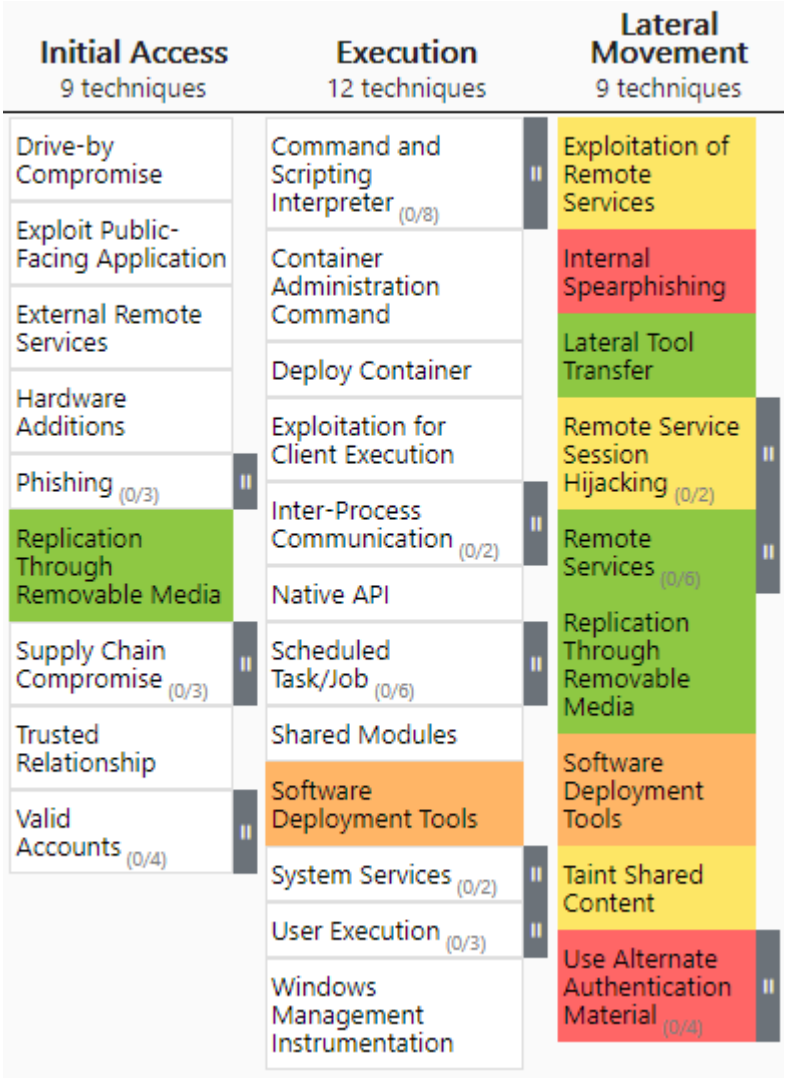

Figure 5.2. The detection map made using the ATT&CK Navigator[6]

Drawing the conclusion of the testing phase, we can see that our coverage is not perfect. The Lateral Movement column was the goal of this thesis, which has two red (uncovered) cells in Figure 5.2.

Focusing on what we have, it is a promising result because we have visibility over the other techniques (colored yellow, orange, and green), and some of these techniques appear in other tactics, such as Replication Through Removable Media that is a well-known method of securing Initial Access.

We can conclude that the system already has prevention in place as well. Whilst testing the system, our workflows have “learned” and stored the threats, and for those specific malicious attacks, our endpoints are protected. Luckily, the knowledge of the threats is shared between our devices with the common anchor point being SecureX.

6. FUTURE PLANS

Thinking ahead, we could improve this current set of workflows in three main ways. We could expand vertically, meaning that the current tactic, TA0008 Lateral Movement is to be progressed, starting with detection in all techniques, then progressing to prevention where possible without major drawbacks.

Another way is horizontal expansion, meaning that we cover multiple techniques that may be in other tactics to broaden our detection map. This approach sounds like the easiest next step because the ATT&CK matrix has plenty of techniques that have relatively easy countermeasures.

Finally, we could make the integration stronger and tighter with SecureX. One example would be to use a “Simple Block List” to deal with threats without using custom lists or tables. This would be a great step forward because having a unified “list of blocked applications” could simplify the work and no more custom solutions would be needed in regular cases. I reckon the “block list” is a reoccurring pattern in many file-related attacks.

Of course, the tighter integration is not only a one-way improvement. We could also make use of the flairs that indicate what ATT&CK technique fits the best a specific incident. Towards the end of writing this thesis, SecureX added their own auto-categorization of ATT&CK techniques to their native events. Using this resource could hasten building visibility, detection, and prevention.

To put it simply, the next action to take would be to convert the workflows of this thesis to use simple block lists, then incorporate the SecureX ATT&CK categorizations. Following that, easy techniques could be dealt with to open up to new tactics.

7. CONCLUSION

All in all, the detection is set up for network related techniques against tactic TA0008 Lateral Movement. The plan was to only use Cisco SecureX's built-in detections and extend them but SecureX components like Secure Endpoint or Umbrella do not always give enough information to start figuring out what the cause was. With an update to SecureX during my work, it became possible to create solid interlaced workflows that rely on information from SecureX mostly. Sadly, the detections were already in place by the time of the update, so I continued to progress my workflow to prevention without starting over with the new information.

Lateral Tool Transfer and Remote Services are the two real network-related techniques in the Lateral Movement tactic that I think should always involve systems like SecureX. Using SecureX to serve as a single point to give a yes/no answer to a file hash based on multiple services' intel is sensible in my opinion, but maybe the real evaluation should take place locally and SecureX should just provide the unified authentication details for the different services, where the threat intel is gathered from.

The original goal for T1021 Remote Services was to serve as a counterexample, however it seems less intensive for SecureX then I imagined. I still believe that device hardening is still an important passive defense, but once in a while this thesis's solution could serve as an extra layer for spotting anomalies.

As it proved itself during the testing, Cisco Secure Endpoint already has defense against executed malware, which invalidates the use of my solution T1091 Replication Through Removable Media in most cases as a good countermeasure. I think it still works well as a logging tool of the AutoRun files inserted into a controlled group of computers.

I would say the system of workflows and the client-side scripts work well if the limitation of the current implementation is considered.

My suggestions for future plans include continuing to implement detection for the Lateral Movement, deploying prevention, and moving to other tactics to detect and prevent.

Based on this work, I suggest involving even more systems. SecureX is great for centralizing the workflows but this thesis does not deal with on-the-machine frameworks. Only my own code was used locally on endpoints, which could be an anti-pattern. As we start covering the whole ATT&CK matrix, I suspect that the local solutions would become unmanageable quickly.

My SecureX workflows are open-source on GitHub.[42]

REFERENCES

[1] "Cisco SecureX – A Simplified Security Experience," *Cisco*. https://www.cisco.com/c/en/us/products/security/securex/index.html (accessed May 11, 2021).

[2] "Welcome to Python.org," *Python.org*. https://www.python.org/ (accessed May 12, 2021).

[3] "MITRE ATT&CK®." https://attack.mitre.org/ (accessed May 12, 2021).

[4] "Welcome to Cisco Orbital." https://orbital.amp.cisco.com/help/ (accessed May 12, 2021).

[5] K. Nickels, "How to Be a Savvy ATT&CK Consumer," *Medium*, Dec. 13, 2019. https://medium.com/mitre-attack/how-to-be-a-savvy-attack-consumer-63e45b8e94c9 (accessed May 11, 2021).

[6] "ATT&CK® Navigator." https://mitre-attack.github.io/attack-navigator/ (accessed May 11, 2021).

[7] B. E. Strom, A. Applebaum, D. P. Miller, K. C. Nickels, A. G. Pennington, and C. B. Thomas, "Mitre att&ck: Design and philosophy," *Tech. Rep.*, 2018.

[8] O. Alexander, M. Belisle, and J. Steele, "MITRE ATT&CK® for industrial control systems: Design and philosophy," 2020.

[9] "Cisco Secure Product Integrations with SecureX," *Cisco*. https://www.cisco.com/c/en/us/products/security/securex/securex-integrations.html (accessed May 11, 2021).

[10] "Cisco SecureX Threat Response - Security That Works Together," *Cisco*. https://www.cisco.com/c/en/us/products/security/threat-response.html (accessed May 12, 2021).

[11] "Cisco Talos Intelligence Group - Comprehensive Threat Intelligence." https://talosintelligence.com/ (accessed May 12, 2021).

[12] "CTR-Whitepaper.pdf." Accessed: May 12, 2021. [Online]. Available: https://enocg.com/wp-content/uploads/2020/10/CTR-Whitepaper.pdf

[13] "Take SecureX wherever you go – introducing the new ribbon browser extension," *Cisco Blogs*, Dec. 11, 2020. https://blogs.cisco.com/security/take-securex-wherever-you-go-introducing-the-new-ribbon-browser-extension (accessed May 13, 2021).

[14] E. Brumaghin, "Lemon Duck spreads its wings: Actors target Microsoft Exchange servers, incorporate new TTPs." http://blog.talosintelligence.com/2021/05/lemon-duck-spreads-wings.html (accessed May 12, 2021).

[15] "SecureX Orchestration: Automate Security Workflows," *Cisco*. https://www.cisco.com/c/en/us/products/security/securex/securex-orchestration.html (accessed May 11, 2021).

[16] "Why SOAR Is a Compelling Proposition for Your IT Security," *Cisco Blogs*, Sep. 22, 2020. https://blogs.cisco.com/customerexperience/why-soar-is-the-future-of-your-it-security (accessed May 12, 2021).

[17] "Home - SecureX Orchestration," *SecureX Orchestration*. https://ciscosecurity.github.io/sxo-05-security-workflows/ (accessed May 12, 2021).

[18] "Cisco SecureX Integration Workflows — Cisco SecureX Integration Workflows 0.1 documentation." https://ciscosecurity-sx-00-integration-workflows.readthedocs-hosted.com/en/latest/index.html (accessed May 12, 2021).

[19] “Threat Response for Developers - Cisco DevNet.” https://developer.cisco.com/threat-response/ (accessed May 12, 2021).
[20] “Cisco Secure Endpoint (Formerly AMP for Endpoints),” *Cisco*. https://www.cisco.com/c/en/us/products/security/amp-for-endpoints/index.html (accessed May 13, 2021).
[21] “ATT&CK in APT Reports | Kaspersky.” https://www.kaspersky.com/enterprise-security/mitre/tip (accessed May 12, 2021).
[22] F. Mottini, *teoseller/osquery-attck*. 2021. Accessed: May 12, 2021. [Online]. Available: https://github.com/teoseller/osquery-attck
[23] “teoseller - Overview,” *GitHub*. https://github.com/teoseller (accessed May 12, 2021).
[24] “BlueTeamLabs.io.” https://blueteamlabs.io/ (accessed May 12, 2021).
[25] *BlueTeamLabs/sentinel-attack*. BlueTeamLabs, 2021. Accessed: May 12, 2021. [Online]. Available: https://github.com/BlueTeamLabs/sentinel-attack
[26] Karl-Bridge-Microsoft, “System Monitor - Win32 apps.” https://docs.microsoft.com/en-us/windows/win32/sysmon/system-monitor-portal (accessed May 12, 2021).
[27] “Azure Sentinel – Cloud-native SIEM Solution | Microsoft Azure.” https://azure.microsoft.com/en-us/services/azure-sentinel/ (accessed May 12, 2021).
[28] orspod, “Getting started with Kusto.” https://docs.microsoft.com/en-us/azure/data-explorer/kusto/concepts/ (accessed May 13, 2021).
[29] “NavigatorLayerFileFormatv4_1.pdf.” Accessed: May 12, 2021. [Online]. Available: https://mitre-attack.github.io/attack-navigator/assets/NavigatorLayerFileFormatv4_1.pdf
[30] *3CORESec/S2AN*. 3CORESec, 2021. Accessed: May 12, 2021. [Online]. Available: https://github.com/3CORESec/S2AN
[31] *rabobank-cdc/DeTTECT*. Rabobank - Cyber Defence Centre, 2021. Accessed: May 13, 2021. [Online]. Available: https://github.com/rabobank-cdc/DeTTECT
[32] “The Official YAML Web Site.” https://yaml.org/ (accessed May 13, 2021).
[33] “Empowering App Development for Developers | Docker.” https://www.docker.com/ (accessed May 13, 2021).
[34] “DeTT&CT Editor.” https://rabobank-cdc.github.io/dettect-editor/#/home (accessed May 13, 2021).
[35] A. Pennington, A. Applebaum, K. Nickels, T. Schulz, B. Strom, and J. Wunder, “Getting Started With ATT and CK,” MITRE CORP MCLEAN VA MCLEAN, 2019.
[36] “Lateral Movement, Tactic TA0008 - Enterprise | MITRE ATT&CK®.” https://attack.mitre.org/tactics/TA0008/ (accessed May 13, 2021).
[37] Deland-Han, “How to detect, enable and disable SMBv1, SMBv2, and SMBv3 in Windows.” https://docs.microsoft.com/en-us/windows-server/storage/file-server/troubleshoot/detect-enable-and-disable-smbv1-v2-v3 (accessed May 13, 2021).
[38] “Custom Table Types,” *SecureX Orchestration*. https://ciscosecurity.github.io/sxo-05-security-workflows/variables/custom-types (accessed May 13, 2021).
[39] *Jayway JsonPath*. json-path, 2021. Accessed: Oct. 26, 2021. [Online]. Available: https://github.com/json-path/JsonPath

[40] “VirusTotal – Learning resources.” https://www.virustotal.com/getstarted/ (accessed Nov. 09, 2021).
[41] “Exploitation of Remote Services, Technique T1210 - Enterprise | MITRE ATT&CK®.” https://attack.mitre.org/techniques/T1210/ (accessed Nov. 10, 2021).
[42] Thayol, *securex-orch*. 2021. Accessed: May 14, 2022. [Online]. Available: https://github.com/thayol/securex-orch