Capability Maturity Measurement of a Security Operations Center through Analysis Detection

Matthijs Vos
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
m.vos@student.utwente.nl

ABSTRACT

Security Operations Centers (SOC) are vital in securing computer networks by detecting and responding to potential threats. In order to test the effectiveness of the SOC, Red Team tests are used. The goal of a Red Team is to simulate a type of adversary to test the company's defences against this adversary. During this engagement, the goal of the Red Team is to stay undetected by the SOC. To be aware of the analysis the SOC is performing, Red Team monitoring methods have been developed in previous research, for which we will introduce and use the term Analysis Detection. Those methods by themselves do not give insight in the capabilities of the SOC. This paper extends those methods and determines the capabilities of the opposing SOC. This helps the Red Team better understand their opponent and improve their campaign. Finally, we show how existing Analysis Detection methods can be used as input for SOC capability indicators to measure the SOC maturity level.

Keywords

Red Teaming, Security Operations Center, Analysis Detection

1. INTRODUCTION

The current world is becoming more and more digital, bringing new cybersecurity challenges. Every company will face a cyberattack someday and thus needs to invest in securing their infrastructure. One of the common ways to achieve this is by having a *Security Operations Center* (SOC).

The SOC is responsible for monitoring the IT infrastructure. To do this effectively and detect adversaries on time, they need adequate visibility in the network. The industry uses different kinds of *Intrusion Detection Systems* (IDS) to monitor the network and generate an alarm when potential suspicious activity occurs. A SOC analyst triages the alarms and takes the appropriate measures.

It is essential to do regular security testing to gain insight into the effectiveness of the security measures. Companies use different methods of ethical hacking for this, but the most realistic and advanced manner is a *Red Team engagement*. A Red Team engagement simulates an adversary to test the technical and organizational security measures, which test the technical protections and the effectiveness of the monitoring and response.

The main goal of the SOC is to understand the current threats in the network, while the main goal of the Red Team is to gain access to the network, while preventing the SOC from detecting their presence. Being aware that the SOC has spotted them helps them adapt and make more progress in their campaign. This field is relatively young and lacks some terminology at the moment. For that reason, we introduce and use the term *Analysis Detection* for the Red Team detecting the SOC analysis. The first paper in this field is from Lahaye[8] which dates to 2018, which shows that it is a young field of research. This paper discovered the first methods for detecting the SOC analysis from a Red Team perspective. Security Risk Advisors[11] did contribute to the terminology in this field by introducing the term *Indicator of Analysis* (IoA). These are the technical indicators that indicate a SOC might be analysing an attack of the Red Team. The Red Team can monitor these indicators to reveal the investigation of the SOC.

The MITRE ATT&CK [10] framework is widely used in the security industry. It consists of techniques attackers can use to get in, through and out of a target network. All these techniques are structured within 14 tactics or stages the attacker can go through. It is used to describe the activity of adversaries and define the monitoring coverage of Security Operations Centers. More about the MITRE ATT&CK framework will be described in Section 2.3.

The existing IoAs are primarily researched by the works of Lahaye[8] and Security Risk Advisors[11]. They both developed a list of potential IoAs, and how a Red Team could monitor for them. However, quite some of these Indicators of Analysis are related to each other, since they use the same kind of indicator. For example, there is an indicator that checks if the requested URL is similar to the URL used by the beacon, but just slightly different, since that might indicate a SOC analyst investigating the URL. Another indicator does the same but for the User-Agent. We group these two, and other, into a single category called *External Communication*. We do this for all the indicators and see that they all belong to four different categories, which we show in Section 5.1. Furthermore, there is also no indication of the number of MITRE ATT&CK paths covered by these IoAs. We investigate how many MITRE ATT&CK techniques can potentially generate an IoA, which gives more insight into the coverage of the MITRE ATT&CK framework of the currently existing IoAs.

Although the technical methods are known, that does not give us information about the strength of SOC. It is helpful to gain insight into the capabilities of the opposing SOC by

using these IoAs. There are different methods to determine the capability of a SOC, such as the SOC-CMM[15], which is a capability maturity framework developed in academia and used in the industry to measure the strength and weaknesses of the SOC in five different areas.

We use this knowledge on SOC capability and SOC analysis detection to determine the capability of a SOC from the Red Team perspective. First we extract the existing Indicators of Analysis from the related work. We structure them into different categories, to gain more oversight. Based on these Indicators of Analysis we determine which MITRE ATT&CK techniques can potentially generate such an IoA. This helps us to understand the current coverage of analysis detection, and which attack methods should be performed to perform an analysis detection.

Based on the SOC-CMM, we determine the capability indicators of a Security Operating Center that can be measured from the perspective of the Red Team using Indicators of Analysis. Together with the Indicators of Analysis we use these two parts to develop a method to measure the capability of a SOC. This is done by mapping an observed IoA onto a capability indicator via the MITRE ATT&CK framework. These three parts together form the SOC capability model.

Finally, we show that our capability model can be used in practice by applying it to a small scale attack. In this Proof of Concept, we execute an initial access attack. We simulate an analysis of this attack by six different methods a SOC could use while we monitor for this analysis. This shows different capability levels depending on the analysis method chosen by the SOC.

Our paper shows our contributions, which in short are:

- An overview of the existing Indicators of Analysis, and their corresponding MITRE ATT&CK techniques.
- An model to assess the capability of a Security Operations Center, based on the Red Teams perspective
- An Proof of Concept to show the working of our model, including extensions of the coverage of an existing Red Team monitoring tool

In the following sections, we present the existing work and propose a new framework for capability measurement from an external perspective. First, we present the current state-of-the-art and background knowledge in Section 2. We use this information to determine the gap in the current literature in Section 3, and introduce the research questions based on that. Based on this motivation we explain the Methodology in Section 4. We structure the existing analysis detection methods and identify the potential MITRE ATT&CK techniques in Section 5. Then we use the knowledge of the SOC maturity models to identify usable maturity levels and indicators in Section 6. In Section 7 we present the newly developed SOC capability maturity framework. Finally, we create a Proof of Concept to apply our framework and validate that it works in Section 8

2. BACKGROUND AND RELATED WORK

In this section we show the existing literature and domain knowledge. We start with the concept of Security Operations Centers. In the second part, we describe the concept of Red Teaming. Finally, we present the existing methodologies of SOC maturity classification.

2.1 Security Operations Center

There are two essential parts to keeping a network secure: protection and visibility. Protection is about configuring the devices and used products safely. However, every protection can still be vulnerable to unknown security bugs. For this reason it is vital to have visibility of what is going on in the network and respond to potential threats. Intrusion Detection Systems detect potential intrusions but do not give a global view of the network. This global view is the primary goal of a SOC by aggregating the information from those systems and taking action[3, 2].

The definition of a SOC as described by *McAfee*[9] showcases ten different tasks of a SOC, ranging from compliance certifications to monitoring and threat response. As depicted by Sundaramurthy et al.[14], every SOC is different, which also means that not all companies consider all these tasks a SOC responsibility. In the context of this paper, only three of the tasks are essential. The first is *Continuous Proactive Monitoring*, which is the 24/7 scanning and monitoring of the network. When a tool issues an alert, the SOC is responsible for the triage of the potential threat, which is part of the *Alert Ranking and Management* task. If the issue is a true positive, a response is needed, such as isolating the system or terminating a specific process, this is called the *Threat Response* task. The other tasks such as preparation, root cause investigation and compliance are also noteworthy for the organization. However, these can also be part of different teams (such as a *Computer Emergency Response Team*) and are thus not considered a SOC responsibility in the context of this paper.

The *Security Information and Event Management* (SIEM)[14] is the main component of the needed SOC tooling. This system collects and correlates data from multiple sources about potential threats. This data can be collected from a variety of systems, for example *Endpoint Detection and Response* (EDR), *Network Intrusion Detection* (IDS), *Threat Intelligence Platforms* (TIP) or *User and Entity Behavior Analytics* (UEBA) systems[9]. All these systems together give the SOC insight into the network and will help them to keep the environment safe.

2.2 Red Teaming

When an organization has security measures in place, it is vital to test their effectiveness. There are three types of security testing: vulnerability assessments, penetration testing and red teaming[16]. The community often interchanges these terms in practice, which is fine when discussing security testing in general, but each has a fundamentally different purpose.

There are three main goals in security testing. The first goal is to identify the current state of information security, how much is the organization in control. The second goal is to identify weaknesses in the infrastructure that real adversaries could misuse. The last goal is to train the security

team and let them improve their detection and response capabilities[17].

A vulnerability assessment or penetration test is helpful to reach the first two goals. With a vulnerability assessment, the goal is to find as many vulnerabilities in a system and prioritize them by risk. During a penetration test, the goal is to identify what a real adversary could do with these vulnerabilities. The tester exploits the vulnerabilities to determine the risk for the company.

Only testing the systems within a technical scope is not enough. As stated before, unknown vulnerabilities can always exist. For this reason, the SOC exists, but its member need to be trained and tested as well. That is where Red Teaming is valuable. Kovačević and Groš defines Red Teaming as follows *"Red teaming is a methodology through which a red team, with authorization, in accordance to the organization's policies and defined scope, conducts planned attack exercises to train the organization's defenders and employees."*[7]. Especially this goal of a red team engagement makes it different from the other types of security testing. It is not just testing the application in all possible ways, but the interaction with the organization's defenders is essential. Red Teaming is also known as adversary simulation, simulating a real adversary to give the organization insight into the efficiency of its defences against certain relevant adversaries.

2.3 MITRE ATT&CK

The MITRE *Adverserial Tactics, Techniques & Common Knowledge* (ATT&CK) is a knowledge base and model for adversarial behaviour. It covers several different ecosystems, such as Enterprise, Mobile and ICS. These are different ecosystems that an adversary could use to reach its objectives. Within the context of this paper we mainly focus on the Enterprise domain.

It is often visualized as a matrix, with techniques as rows and the tactics or stages as columns. Tactics are the main objectives for the adversary, such as gaining initial foothold which is represented by the initial access tactic. The techniques are the methods that the adversary use to reach this objective, such as sending phising emails. In total there are 14 tactics, and 218 techniques in the current version.

2.4 SOC maturity

To determine the maturity of a SOC, we first need to know how to define maturity. In the context of this paper there are two necessary aspects to know, the existing maturity models such as *SOC-CMM*[15], and the detection capabilities. This SOC-CMM gives us a framework to scale the maturity of a SOC, while the detection capabilities of the SOC provides insight into the amount of visibility they have.

2.4.1 SOC-CMM

The SOC-CMM framework categorizes the maturity of a process into five levels. These are the same categories as the CMMI[4], which is the general capability maturity framework used by the SOC-CMM. The first level is 'initial' when the process is unpredictable and has a reactive approach. Level two is 'managed'; the process is formalized for repeated quality. The next level is 'defined'; this is reached when it has been fully documented and formalized. The maturity level is 'quantitatively managed' when the aspect is being measured to optimize it further. The last level is 'optimizing'; the process is measured for optimization on an organizational level.

To create the SOC-CMM Van Os[15] surveyed security monitoring organizations to determine what makes a SOC mature. One of the results is a list of 17 capability indicators that every mature SOC should have. All these capabilities are measured on a five-level scale, starting at no capability up to fully capable. There is no formal definition in the research when to use which capability score.

2.4.2 Detection Capabilities

Products such as MITRE DEFEND [6] help SOCs to determine their visibility in the network and the types of threats they are not monitoring yet. This model categorizes different detection methods in different tactics and maps them to the corresponding MITRE ATT&CK [10] tactic. Using MITRE DEFEND gives insight into which types of adversaries and attacks are detected and which are not.

2.5 Analysis detection

When the SOC analyzes a threat, they might leave certain tracks called Indicators of Analysis (IoA)[11]. For example, the SOC might analyze an incoming email, which contains a link. In order to assess the link they could visit the website using a sandbox, or use curl. This will generate a trace for the Red Team in the access logs. This could give them insight into the fact that someone is trying to understand their campaign. We will use the term *Analysis Detection* for this detection of the SOC analysis by the Red Team.

Lahaye's study, called 'How to spot the Blue Team'[8], is the first in this field. He introduces the methodology to append the infrastructure of the Red Team with measures to detect the analysis of the Blue Team. There are five Indicators of Analysis proposed in this paper, mainly based on *Command and Control* (C2) communication. The first one is the detection of incorrect communication paths, for example, a subpath of the legit C2 path but not a path that a random network scanner can guess. Another option is looking at the user agent; the malware will use a specific user agent. When another user agent is detected, someone is investigating the C2 server. It could also be an indicator when the geographical location of the request and the company do not match. Another detection method is monitoring lookups of DNS records. If a subdomain that is not in use is looked up, this might indicate that the correct subdomain is found, and the investigators are investigating the domain. The last proposed IoA is not based on the C2 traffic but uses public intel. Different online virus databases (such as VirusTotal, IBM X-Force and Hybrid Analysis) can tell which files are being uploaded. If custom malware is uploaded to this kind of database, it could be detected and used as an IoA.

The second contribution in this field is done by the industry, by the company called Security Risk Advisors[11]. They proposed a method to have a Red Team SIEM and introduced the term IoA. A new method proposed in this study is using the request's origin. Specific IP addresses are never expected to request the malware payload, for example, a TOR exit node. Another proposed method is to detect when malware is executed in a sandbox, for example in cloud sandboxes

as any.run. The sandbox has other characteristics than a regular machine, which can be detected. Based on all these ideas, a Red Team SIEM[1] has been created, which makes it possible to detect this kind of behaviour.

Quite some of the aforementioned detection methods are based on behaviour that the SOC could prevent. If they emulated a real user realistically when analyzing the C2 server, this could stay hidden for the Red Team. Crichlow[1] did research into Blue Team operational security (OPSEC) failures. It identifies actions that could compromise the analyst's investigation, like uploading the malware samples to online services. These can be used for detection by the Red Team. However, some failures pose a risk for the analyst itself or even for the customer of the SOC. This could happen if the analyst provides real credentials to a phishing attempt by mistake or discloses the customer's name to the public. These risks are less interesting for this research since they do not reveal the investigation.

2.5.1 RedElk

The research of Lahaye[8] later on led to the development of a monitoring tool for Red Teams called *RedElk*[12]. This allows Red Teams to monitor their infrastructure and implement alerts based on IoAs. It gives the possibility to gain oversight of the operation. Often multiple servers are used during a Red Team engagement, which makes it cumbersome to find the correct logs or screenshots. The RedElk tool centralizes all of this, which helps during reporting.

RedElk is a collection of multiple open source products, combined with some custom scripts. The main components are *Elasticsearch*, *Logstash* and *Kibana*. All Red Team servers will be extended with Logstash to process the logging and push this to the central Elasticsearch database. Kibana is then used to visualize the logging and make it easy to search through all the logging.

RedElk features alarms that can be used to inform the Red Team about potential SOC analysis, which is useful for our research. Some of the Indicators of Analysis are already implemented, while some of the others are still missing. More about this is explained in Section 8.1.1

3. MOTIVATION

Now that we established the state-of-the-art, we identify gaps in the current research. This allows us to do more research and build on the existing work. We show those gaps and create our research questions using them in the next section.

3.1 Problem statement

The idea of Analysis Detection is a relatively new concept but has much potential. The current research identified different IoAs, as already outlined in Background and Related Work. However, there is currently no insight into which Red Team attacks could allow the SOC to create such an IoA. This insight is of interest since this allows us to see how powerful the current state of analysis detection is.

On the other hand, there is a lack of interpretation of these IoAs. Previous work has researched the technical methods, but that does not give us information about the kind of SOC. It would be beneficial for the Red Team to have insight into the maturity of the SOC. This insight will help in two ways; foremost, it allows the Red Team to adapt their campaign. If the opposing SOC has a low maturity, it makes no sense to perform a highly sophisticated attack since that will not help to train them. Secondly, it will help the Red Team advise the SOC on where they could improve. They could give them insight into how to grow in their OPSEC maturity and which attacks they did not detect.

[1]https://github.com/SecurityRiskAdvisors/RedTeamSIEM

3.2 Research question

Given these two lacking factors in the field, the missing coverage indication and the lack of IoA interpretation, we identify the objective for this research; develop a framework that can assess the SOC maturity from a Red Teams perspective.

Based on this goal, we define the main research question: **How to classify the capabilities of a SOC based on Analysis Detection methods?**

We define the following three sub-questions to give more structure to the research:

1. What are the current analysis detection methods, and how well do they cover the MITRE ATT&CK Framework?
2. What are the different maturity levels for a SOC? And how can they be classified?
3. How could analysis detection methods be used to assess the SOC capability maturity level?

4. METHODOLOGY

In order to answer these three research questions and the main question we break our research into four parts. We structured our research related to the process of Analysis Detection, which is visualised in Figure 1. The following paragraphs explain all those steps, and how we conduct our research.

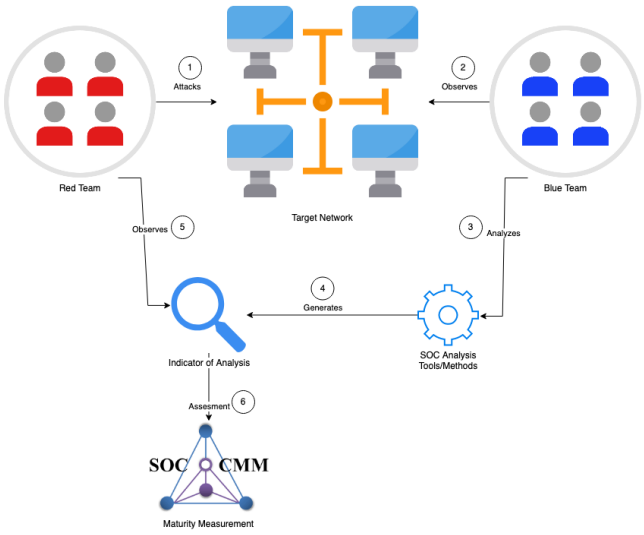

Figure 1: Complete Maturity Measurement process[2]

Red Team side On the Red Team side of the process we need to be able to attack the target (step one), and observe the Indicator of Analysis that might be generated (step five). In order to observe these Indicators of Analysis, we first want to know which IoAs exist and which attack methods we can use in step one to potentially let the SOC generate the IoA. In the first step of this research, we summarise and aggregate all analysis detection methods. We organize them into different categories in Section 5. We develop a method to determine the coverage of MITRE ATT&CK and potential attack paths using all these detection methods. We do this by developing criteria to apply to the MITRE ATT&CK techniques to determine if they can potentially lead to an IoA. This is presented and applied in Section 5.2.

SOC side On the SOC side there are multiple steps; they need to observe (step two) the threat, analyse it (step three) and, by accident create an IoA (step 4). Although these steps are needed in the processes, we do not research them in-depth. The relevant part is how we can classify this process of the SOC in terms of capability. We already identified the SOC-CMM as the main capability maturity framework in the Background and Related Work section. We investigate which capability indicators can be measured from a Red Team perspective, as we need these in our final capability framework. To this end, we extract the capability indicators that the SOC-CMM defines and divide them into two distinct groups. We show that one of the groups can be measured from the Red Team perspective, while the Red Team cannot observe the other.

Capability Measurement The last step in the process (step six) is assessing the capability of the SOC. For this we use the observed IoAs and the SOC capability indicators. We develop a method to grade the different IoAs with a capability level. Then we link every IoA to a SOC capability indicator to give a final overview of the SOC capability. In Section 7 we present an outline of how to determine the SOC capability maturity using these Indicators of Analysis.

Proof of Concept Finally, we develop a Proof of Concept to apply our developed framework. We execute an attack against a target machine and simulate different analysis options that a SOC could do. This PoC shows the possibility to detect the analysis and the corresponding capability of this simulated SOC.

Since we use a simulated SOC in our Proof of Concept, we cannot validate the accuracy of the resulted capability maturity level. We consider this out of scope because of the limited time and existing work, which makes it infeasible to develop and validate the model within the same time frame. However we propose a method to validate the framework as part of the future work (see Section 9).

5. ANALYSIS DETECTION

In this chapter we determine the existing analysis methods, structure them, and determine which attack paths could potentially lead to an IoA.

[2]Icons via: `http://www.flaticon.com`

5.1 Existing Indicators of Analysis

We described the current IoAs in the Related Work section. In order to give more oversight in them we categorize them in four different main categories; *External infrastructure*, *Public Intel*, *Run environment* and *miscellaneous*. These methods are all extracted from the works mentioned in the Related Work from the authors Security Risk Advisors[11], Lahaye[8] and Crichlow[1]. An overview of all these methods can be seen in Figure 2.

All IoAs in the External infrastructure category are based on external network traffic that the analyst could generate. During analysis, the analyst might try to access the adversary's infrastructure to analyze its behaviour. The analyst might not be able to mimic the malware completely, alerting the adversary. An example of this is using another user agent or accessing the incorrect path.

To do a quick initial investigation, public services are valuable. These services like *VirusTotal* or *urlscan.io* allow inspecting for malware in files or websites quickly and easily. From an OPSEC perspective, the disadvantage is that these services allow everyone to see if a specific file or URL was analyzed (without actually uploading the file). The Red Team can misuse this behaviour to check if their file or URL has been analyzed.

The SOC analyst might run the malware in a sandbox in a more in-depth analysis. This allows the Red Team to be informed about the sandbox execution of the malware. The malware could analyze the environment in which it is being executed. If this environment has known sandbox behaviour, it could alert the Red Team by calling back to their infrastructure.

The last category contains everything that could not be organized within the other categories. These are the IoAs providing fake credentials on a phishing website or creating a highly privileged account on a machine, suggesting an investigation.

External communication	Public intel	Run Environment[11	Miscellaneous
Communication Paths[8]	File hashes[8, 1]	Automated sandbox	Fake Credentials[1]
User-Agent[8, 1]	URL scanners[1]	Commercial sandbox	Analysis accounts[1]
Geolocation[8]	Targeted Google Ads[1]	Self built sandbox	SOC-Customer communication via compromised system[1]
DNS lookups[8]			
Request origin[11]			

Figure 2: Existing Indicators of Analysis

5.2 Mapping IoA to MITRE ATT&CK

We want to understand which kind of attacks can potentially be used to trigger an IoA. In order to get an IoA, three steps need to be taken. First, the Red Team needs to execute some attack step that leaves an artefact that the SOC can examine. The second step is the SOC detecting this action and finding the artefact. The last step is the SOC analyzing the artefact so that they create an IoA.

Not all attacks could trigger this chain of events since not every attack is observable by the SOC, or the SOC could do the complete analysis offline. We use the MITRE ATT&CK Enterprise framework to identify which techniques could potentially create an IoA. In order to select suitable attack techniques, there are a few criteria applied for each technique.

The first factor to check is if there are any detection methods for the offensive technique. If there are no detection methods, it is unlikely that the SOC will find the artefact. We used the MITRE DEFEND framework, which specifies many different defend techniques mapped to attack techniques.

We have a potential attack method that the SOC analyst could pick up if these conditions are met. However, there also needs to be useful information in the artefact, which allows the analyst to research the artefact.

In order to trigger one of the analysis detection methods in the External communication tactic, the analyst needs information about the infrastructure. This is information like an IP address or a hostname, such that the analyst can contact it. The hostname or IP address should be included in the artefact. Secondly, the infrastructure needs to be controlled by the Red Team. Otherwise, it is impossible to monitor the infrastructure's requests.

Another analysis detection method is monitoring public services, like VirusTotal and urlscan.io. In order to do this, a file or URL that can be used on these services should be left on the system. The same is true for the sandboxes; this analysis detection method cannot be triggered when no file can be executed in a sandbox.

The misc IoA 'fake credentials' can only be triggered when the analyst can enter information like credentials. This is mainly possible in attack types like a phishing page, so the possibility to enter credentials is one of the criteria.

The SOC-Customer communication and analysis account IoAs are not included in the criteria. These IoAs are not linked to a specific attack. Communication with the customer and creating a new high privileged analysis account can always happen, regardless of the attack. These are excluded from the criteria since that would result in the entire MITRE ATT&CK framework.

We summarized all these criteria in Figure 3. The technique is included when it is usable (indicated by the green box); otherwise, it is excluded. This results in a framework that includes all suitable techniques for the Red Team.

5.3 MITRE ATT&CK techniques

We reduce the MITRE ATT&CK to only include techniques that could potentially create an IoA. This reduced framework contains all techniques that satisfy the questions introduced in Figure 3 and thus can be used for analysis detection. All these techniques are shown in Figure 4.

As described in Section 2.3 there are a total of 218 techniques in MITRE ATT&CK . There are only 89 techniques left after applying the mapping between MITRE DEFEND and MITRE ATT&CK , 41% of all techniques. Of these 89 techniques, 28 unique techniques can potentially create an IoA and are thus usable. There are 38 attack options but only 28 unique techniques since some techniques are present in multiple stages. Those are the techniques that are listed in Figure 4.

When we analyze these results, we see techniques that can also be expected. Most techniques are persistence, privilege escalation and command and control techniques. Those are expected since these are the tactics in which tools and external infrastructure is used. This can create artefacts on the system that the analyst can analyze.

6. SOC CAPABILITY MATURITY

The existing literature describes different options to define SOC maturity, but the industry standard is the *SOC-CMM* [15]. The SOC-CMM is a tool to assess the maturity and capabilities of a SOC. It assesses the SOC on five different aspects, *business, people, process, technology and services.* The idea of this capability maturity assessment is based upon the *Capability Maturity Model Integration for Services* (CMMI-SVC)[5], which is a capability and maturity framework for software and system improvements.

6.1 Monitoring capabilities

The SOC-CMM has a list list of 17 different capabilities that a mature SOC should have, as already described in Section 2.4. All these capabilities are listed in Table 1. We want to select capabilities that could be measured from the Red Team perspective, by using the earlier introduced Indicators of Analysis.

When we analyze the capability indicators we see some differences in them. Some of them seem to be better observable by the Red Team using Indicators of Analysis than others. For example, User Monitoring is a capability that the Red Team cannot detect. For this they should have insight in the technologies used by the SOC. However, Early detection is something that can be detected by the Red Team. If something from an early attack stage generates an IoA they can detect this Early detection of the SOC.

This difference in Capability indicators can be seen as the methods and the goals of the SOC. Performing early detection is the goal of the SOC, while using user monitoring is one of the methods that allow the SOC to perform this. For that reason we divided the capability indicators into two groups, the 'goals' and 'methods'. The Red Team can observe the goals, while the methods are not observable from this perspective.

We see that six of them are 'goal' capabilities, and 11 of them are methodology capabilities. Things like detecting

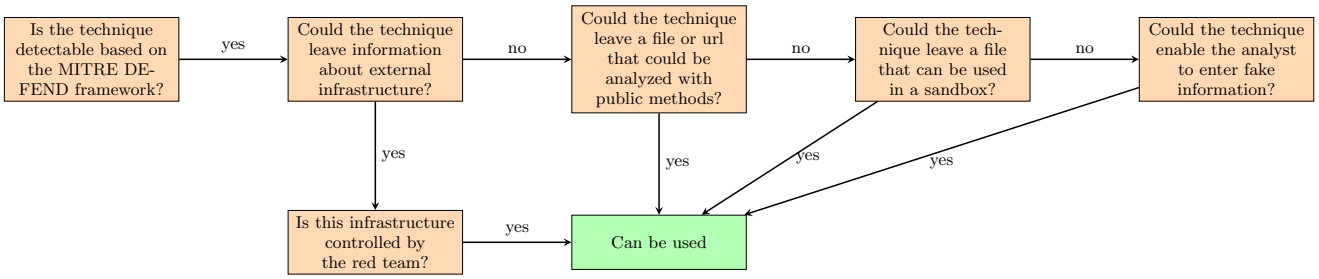

Figure 3: MITRE ATT&CK technique selection for analysis detection

in an early attack stage (*Early detection*) and noticing the use of malware (*Malware detection*) are describing goals that the SOC would like to achieve. In contrast, capabilities such as *Host monitoring* and the use of a *Hunting Team* are the technical capabilities to achieve these, which the Red Team can not monitor.

We use these goals to measure the SOC capabilities maturity. Since we can observe them from the Red Team perspective, we can link an IoA to them. In the next chapter, we couple these goals to the different IoAs and show how to determine the capability.

Capability	**Goal / Method**
Early detection	Goal
Intrusion detection	Goal
Exfiltration detection	Goal
Malware detection	Goal
Anomaly detection	Goal
Real-time detection	Goal
Alerting & notification	Method
Status monitoring	Method
Perimeter monitoring	Method
Host monitoring	Method
Network & traffic monitoring	Method
Access & usage monitoring	Method
User monitoring	Method
Application & service monitoring	Method
Hunting team	Method
Use cases	Method
Acceptable use policy	Method

Table 1: Security monitoring capabilities

7. SOC CAPABILITY MEASUREMENT

Now that we have created a list of SOC capabilities in Table 1, a list of Indicator of Analysis in Figure 2 and a list of attack techniques to allow the creation of these IoAs in Figure 4. We can combine this information to to answer the last research question: *How could analysis detection methods be used to assess the SOC capability maturity level?*.

To measure the SOC capability, we need to identify the capability level of a certain IoA and couple them with the SOC capabilities.

7.1 IoA capability

The generation of every IoA indicates different levels of maturity. Using the VirusTotal database is a relatively trivial and easy to execute step during an analysis. Creating and using a custom sandbox environment that imitates the client is much more complex and thus indicates a higher capability.

To make it easier to validate this model, we want to make it possible to map to the SOC-CMM. For this reason, we decided to use the same capability scale as the SOC-CMM, with a five-level scale, starting at no capability till fully capable. Each IoA gets a fixed capability level, indicating how capable the SOC is when they create that IoA.

As explained in Section 2.4, SOC-CMM uses a five-level scale for each capability. We use the same scale in our model since that should make the models interoperable. We use Capability Level zero (CL0) to indicate that there is no IoA detected within a particular category, so that level is not used in the capability list of the IoAs itself. So we scale every Indicator of Analysis within CL1 to CL4. To make it a bit easier to do the scaling, we provide some guidance which we used for the scale.

CL1 is used when the indicator is triggered without a lot of security knowledge or simple tools. If the IoA is triggered by something that has some basic Operational Security, we use CL2 for the capability. When the SOC did an average job on their OPSEC, we use CL3. The last capability level, CL4, is used when there is good OPSEC but still the possibility to detect the SOC.

These requirements are applied to all the IoAs that we identified in an earlier stage. The overview of this is listed in Table 2.

7.2 Total capability

We now use these capabilities from the different IoAs to gain insight into the opposing SOC's total capability maturity. The IoA is created based on an artefact dropped on the system in a specific MITRE ATT&CK stage. Using this stage, we couple the capability level of the IoA onto one of the six SOC-CMM goals. We use a radar chart to visualize this in figure Figure 5.

Every goal will get the level of the lowest IoA observed. The SOC does not want to inform the attacker, so the attacker only needs one OPSEC failure from the SOC to know this. For this reason, we decided to use the lowest IoA as the maturity of one of the goals. The zero grade is used when

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion
Drive-by Compromise	Command and Scripting Interpreter	Boot or Logon Autostart Execution	Boot or Logon Autostart Execution	Hijack Execution Flow
Phishing	Scheduled Task/Job	Boot or Logon Initialization Scripts	Boot or Logon Initialization Scripts	Modify Authentication Process
	User Execution	Create or Modify System Process	Create or Modify System Process	Rootkit
		Event Triggered Execution	Event Triggered Execution	
		Hijack Execution Flow	Hijack Execution Flow	
		Implant Internal Image	Scheduled Task/Job	
		Modify Authentication Process		
		Office Application Startup		
		Scheduled Task/Job		

Credential Access	Lateral Movement	Command And Control	Exfiltration
Modify Authentication Process	Internal Spearphishing	Application Layer Protocol	Automated Exfiltration
		Dynamic Resolution	Exfiltration Over Alternative Protocol
		Fallback Channels	Exfiltration Over C2 Channel
		Ingress Tool Transfer	Exfiltration Over Web Service
		Multi-Stage Channels	
		Non-Application Layer Protocol	
		Proxy	
		Remote Access Software	
		Web Service	

Figure 4: Eligible MITRE ATT&CK Enterprise methods

IoA	Capability level
Communication Paths	CL2
User-Agent	
Curl	CL1
Custom but incorrect	CL2
Geolocations	
Incorrect country	CL2
Correct country, unknown origin	CL3
DNS Lookups	CL1
Request origin	
SOC-Network	CL1
TOR	CL3
File hashes	CL1
URL scanners	CL1
Targeted Google Ads	CL3
Automated sandbox	CL1
Commercial sandbox	CL2
Self built sandbox	CL4
Fake credentials	CL1
Analysis accounts	CL3
SOC-Customer communication via compromised system	CL1

Table 2: Analysis detection methods capabilities

no IoA is observed. This could mean two things, we do not have a SOC on the opposing side, or they are good at their OPSEC. So this state is one to be careful with, but it is not possible to distinguish between these two states within the methodology.

We link the IoAs to the goals in the following way. The maturity of Early detection is determined by IoAs generated from attacks in the Initial Access, Execution and Persistence stages of MITRE ATT&CK . The maturity of the Exfiltration detection is based on the Exfiltration stage. Malware detection is not based on specific MITRE ATT&CK techniques but when malware has been used to trigger the IoA.

Real-Time detection can be measured based on speed. However, this paper does not cover that, so future research should investigate that. That also counts for Anomaly detection since that is quite a broad term that the current IoAs do not cover.

The Intrusion detection goal is a broad term since everything that a Red Team does is an intrusion. For that reason, we decide to score everything that is not in one of the other goals under the intrusion detection goal.

This means in practice that a Phishing URL used during an Initial Access attack which later on created an urlscan.io hit, will give CL1 for the Early Detection goal. While a malware sample that has been used during Lateral Movement that generates a custom sandbox IoA will result in CL4 for the Malware Detection goal.

8. PROOF OF CONCEPT

To test the just introduced capability framework, we create a Proof of Concept. We executed a small adversary simulation in a test environment and simulated the steps that a SOC could have performed. For this we use tools which are

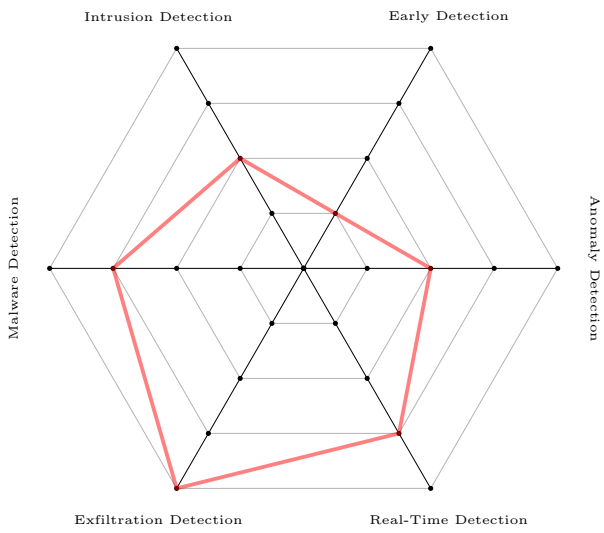

Figure 5: Total capability maturity radar chart

commonly used by the security community, like VirusTotal. To spot these steps of the SOC, we use *RedElk*[13], which allows us to get insight into the traffic to the C2 servers.

8.1 Setup

We create a test setup that conists of two networks, an adversary network and a target network. The adversary network is used to simulate the infrastrutucre a Red Team would use during an engagement, while the target network simulates the target of the Red Team engagement. The PoC network is visualized in Figure 6.

The C2 server is the server responsible for delivering all the malware payloads. When the malware executes on the target machine, it will try to reach the C2 server to ask for further instructions. This operator then uses this C2 to instruct the infected machines to perform specific tasks, allowing them to navigate the system and potentially the network.

Redirectors obfuscate the actual location of the C2 server. These are servers that forward all incoming traffic to the C2 server. The Red Team uses them to filter C2 traffic and redirect all other traffic to a decoy website, making it harder to detect the domain as malicious.

RedElk is a system that allows us to collect logs from different services, like the C2 and redirector servers. We use this to oversee the adversary simulation and monitor the traffic on suspicious behaviour. This monitoring allows us to identify the SOC analysis at the network level.

For the sake of simplicity we build a small target network with one Windows 10 machine. We used a Windows 10 machine since this is a supported windows version that could be used by a real target. It is not that important what the exact operating system is, since Red Teams could be executed against a variety of operating systems. We used Microsoft Defender for Endpoint as our Antivirus and EDR product, since this is a well known product and gives clear insight in the alarms it

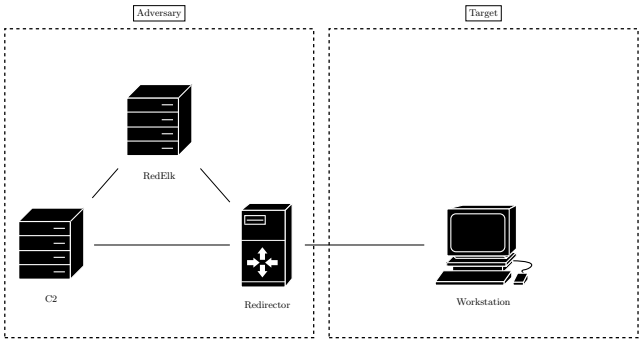

Figure 6: Network setup PoC

8.1.1 *RedElk*

RedElk mainly monitors for network traffic, so it is mainly usable for the External Communication type of IoAs. The current public version of RedElk contains three usable alarms for this research. During our research we appended this with three new alarms to improve the amount of IoAs that can be monitored. This contribution will also be added to the public version of RedElk. The newly developed alarms are marked with an asterisk in the list below.

- Alarm Filehash
- Alarm Geolocation*
- Alarm HTTP Traffic
- Alarm User-Agent
- Alarm TOR*
- Alarm Suspicious Beacon*

Filehash The file hash alarm checks known public malware sources on known hashes. This matches with the 'File hashes' IoA. RedElk allows us to register the hashes of the malware used during a campaign, and then this alarm will check VirusTotal, IBM X-Force and Hybrid Analysis to see if that malware has been uploaded.

Geolocation RedElk does not have an alarm for geolocations. Since this is one of the IoAs we want to monitor, we implemented this alarm in RedElk. This new alarm compares the geolocation of each request with a list of allowed geolocations. This matches with the 'Geolocation' IoA.

HTTP Traffic The HTTP Traffic alarm raises the alarm each time a new IP contacts the C2 server. This alarm generates a lot of false positives since each legitimate beacon will also cause an alarm. For this reason, we created a new suspicious beacon alarm. This HTTP Traffic alarm corresponds to the 'Request origin' IoA.

User-Agent The User-Agent alarm raises for well-known script language and chat application user agents. These are user agents like curl, python, and user agents from applications like WhatsApp or Slack. The normal C2 traffic will not contain these user agents, since they make use of standard browser-like user agents for obfuscation reasons. This means

that observing these user agents indicates an investigation, and maps to the 'User-Agent' IoA.

TOR The TOR alarm raises an alarm each time the TOR network makes a request. This alarm corresponds to the 'Request origin' IoA. Requests originating from TOR are suspicious since normal users will not use it to reach the C2 server.

Suspicious Beacon The newly built Suspicious Beacon alarm combines a few IoAs, namely the user-agent, communication paths and the request origin. It checks for all requests made from the target's IP addresses and determines the user-agent/path combinations. Then it checks if these combinations are also made from other IP addresses. If the same user-agent/path combination is found, the malware is executed outside the target network, which might be a SOC analyst running it in a (cloud) sandbox.

All these alarms together give us a good insight into the network traffic and the SOC operations. Based on what kind of traffic we observe, we can gain insight into the SOC's strengths.

8.2 Scenario

The main goal of our Proof of Concept is to show that we can apply our model in a simulated Red Team engagement. The goal is not to do an extensive test or validation of our model, since we leave that for future work which is described in Section 9. For that reason we decided to limit our PoC to a single attack, which is an initial access attack. An commonly used technique in a Red Team engagement and real world attacks are Word Macros. By embedding a bit of malicious code into a Word Macro the adversary can install a beacon on the target machine by making the user open the Word document. By attaching this file to a phising mail the user could open the document and thus unintentionally give the Red Team access to their machine.

The payload we use is a detectable Cobalt Strike payload. The Microsoft Defender instance running on the machine instantly finds it when the Word document is opened. In practice, this does not happen since more sophisticated payloads are used. However, we want to simulate and measure the SOC analysis, and not how advanced of a Red Team we can execute. In order to simulate the SOC analysis, we need MDE to generate an alert. So this scenario simulates the case when the EDR does detect the malware, which does happen during Red Team engagements.

When the SOC receives the alarm, it starts the analysis. For this, we simulate a few different options that the SOC analyst has:

- Upload the malware to VirusTotal
- Upload the malware to any.run
- Investigate the malware with Microsoft Deep Inspect
- Investigate the malware communication using curl
- Investigate the malware communication via TOR

8.3 Results

The analyst uploads the malware to VirusTotal, which will automatically upload the malware to multiple cloud sandboxes to analyze it. We uploaded our malware sample to VirusTotal, and within a few minutes, an alarm was raised by RedELK. This analysis resulted in one maturity for the Early detection goal.

The better option is using any.run; this allows to analyze the malware but without making it public that the file has been investigated. However, it is possible to detect this by using the Suspicious Beacon alarm. The any.run sandbox will still make the request to the real C2 server and from IP addresses outside the customer network and from an incorrect country. This analysis resulted in two maturity for the Early detection goal.

Related to this is using Microsoft Deep Inspect. It will analyze the malware within the sandbox environment of Microsoft. This will execute the malware, and create the connections to the C2 server, just like any.run did. So this results in the same maturity as any.run, a two on Early detection.

During an investigation an analyst might find the C2 communication URLs or IP addresses in the binary. The analyst can use curl or paste this C2 link into Slack. This results in a hit on the C2 server with a User-Agent that RedElk can match with curl/slack. The User-Agent alarm checks this and informs us. This analysis resulted in one maturity for the Early detection goal.

If the SOC wants to obfuscate that they are investigating the malware, they could use TOR to prevent their IP from showing up in the logging. However, most IPs of TOR exit nodes are known[3], so this is detected. This analysis resulted in three maturity for the Early detection goal.

9. VALIDATION

Further research could validate the currently defined capability levels for each IoA. They are based on the described criteria but not validated against an actual SOC. We propose applying this mechanism during Red Team engagements in which the opposing SOC has done the SOC-CMM maturity measurement. Future research could compare the result of the Analysis Detection Capability with the SOC-CMM Capability, and the Analysis Detection Capability could be calibrated using that data. It is impossible to do such extensive data gathering within the time available for this research, and thus we leave this for further research.

The different Indicators of Analysis can be tracked and monitored during the engagement using RedELK. At the end of the engagement, the capability overview could be created using the methodology of this paper. Note that this might take a lot of time since the duration of a Red Team engagement can be multiple months.

After this data gathering, multiple overviews of the capabilities of different SOCs have been gathered. All those SOCs could also do the internal SOC-CMM capability audit. Since we only use a part of the technology part of the SOC-CMM,

[3]https://check.torproject.org/torbulkexitlist

it is also possible to only execute that part of the SOC-CMM audit.

Using thse two datasets, the difference between both capabilities can be calculated. If there are significant differences, it would be possible to track this back to an individual IoA, of which the capability level could be adjusted. When this is done, the model is calibrated. If needed, this experiment could be repeated for validation reasons.

10. DISCUSSION

In this paper we have created a basic model to identify the capability maturity of a SOC. More work can be done in this field since this is the first paper combining Analysis Detection with Maturity measurements.

First, the validation of the model should be done, as described in the previous chapter. This paper proved that it is possible to identify the capability. However, we did not validate that the outcome of the capability is valid. Due to the constrained time of this research, there was no time to execute such a validation.

Currently we only used a small part of the SOC-CMM. This part of the SOC-CMM can be measured from the Red Team perspective, as shown in this paper. However in the future it could be beneficial to develop a model more technical oriented. The SOC-CMM is only partly focusing on the technical parts, and mainly focused on the processes itself. By researching a more technical SOC capability model it would be easier to scale the technical capabilities of the SOC.

Another improvement is to use combined IoAs. Every IoA is treated individually; however, some IoAs could be interlinked. Using a combination of an incorrect location with a correct User-Agent is an example. We already used this part in the alarm implementation for RedELK but did not investigate this further.

When combining different IoAs, it might be possible to detect even more. We identified that it is possible to spot the C2 communication from beacons running in a (cloud) sandbox during the Proof of Concept. It would benefit the Red Team to know which kind of sandboxes are investigating their C2 infrastructure since specific sandboxes are linked to their corresponding EDR products, like Microsoft Deep Inspect and Microsoft Defender for Endpoint. Detecting Microsoft Deep Inspect gives the Red Team the information that the SOC uses Microsoft Defender for Endpoint. This allows the Red Team to discover the used products by the target and thus evade them afterwards. Future research could determine the distinguishable characteristics of the different cloud sandboxes and detect them from their network connections.

During the mapping of the IoAs onto the capability indicators we found that two of the indicators can currently not be measured. This applies for the Anomaly Detection and Real-Time detection indicators. Further research should investigate if it would be possible to measure them, since they should be observable from the Red Team perspective.

Last of all it would be beneficial to research more Indicators of Analysis. The existing Indicators of Analysis can only be triggered by a limited set of attack techniques. There are also two capability indicators in our model that currently do not By studying more Indicators of Analysis the Red Team would improve their Analysis detection capabilities. Having more Indicators of Analysis also enhances the capability measurement since more SOC analysis can be detected.

11. CONCLUSION

The goal of this research was to develop a framework that can assess the SOC maturity from a Red Teams perspective. We showed, using the three research questions, that to some extend it is possible to assess the SOC maturity using Indicators of Analysis.

In Section 5 we answered the question *What are the current analysis detection methods, and how well do they cover the MITRE ATT&CK Framework?*. We showed that the existing literature already developed 13 Indicators of Analysis, which can be used for Analysis Detection. We structured these 13 indicators into four different groups, to give more organization to them and group the related indicators together. By applying criteria to the MITRE ATT&CK Framework matching with these Indicators of Analysis we found that only a small subset of the MITRE ATT&CK techniques can be used for analysis detection. The SOC is not able to detect all attack techniques. If they are able to detect the technique they will not always generate an IoA. It is only possible to detect the SOC if they generate an Indicator of Analysis, and thus the Red Team will not always be able to detect the SOC.

In the second part of our paper we answered RQ2: *What are the different maturity levels for a SOC? And how can they be classified?*. In order to answer this we researched the maturity levels from the SOC-CMM and the corresponding capability indicators. We found that there are six different capability indicators of a SOC that can be assessed from a Red Team perspective. These capabilities can be scaled onto a five point scale, starting at no maturity, to fully mature.

We then used this information to build our final framework and answer the last question: *How could analysis detection methods be used to assess the SOC capability maturity level?*. We used this five point scale to assess every Indicator of Analysis with a capability level. Then we used the MITRE ATT&CK tactics to map the IoA to one of the six capability indicators. By doing this we found that for two of these indicators further research is required, since they can currently not be mapped with the existing Indicators of Analysis.

Finally we applied our model in a simulated attack, to show that it can be used during a Red Team engagement. To achieve this we improved the detection capabilities of the Red Team within RedELK, since the current detection methods in RedElk were limited. We added a three new detection methods that help the Red Team to protect their campaign and spot the Security Operations Center.

References

[1] Matthias Caretta Crichlow. *A study on Blue Team's OPSEC failures*. info:eu-repo/semantics/masterThesis. Publisher: University of Twente. Oct. 2020. URL: https://essay.utwente.nl/84945/ (visited on 09/10/2021).

[2] Abdoul Karim Ganame et al. "A global security architecture for intrusion detection on computer networks". In: *Computers & Security* 27.1 (Mar. 2008), pp. 30–47. ISSN: 0167-4048. DOI: 10.1016/j.cose.2008.03.004. (Visited on 10/18/2021).

[3] Abdoul Karim Ganame et al. "A High Performance System for Intrusion Detection and Reaction Management". In: 3 (2006), p. 12. (Visited on 10/18/2021).

[4] Software Engineering Institute. *CMMI for Services, Version 1.3*. Jan. 2010. URL: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9665 (visited on 10/04/2021).

[5] Software Engineering Institute. "CMMI for Services, Version 1.3". In: (Jan. 2010), p. 520. DOI: 10.1184/R1/6572375.v1. (Visited on 10/04/2021).

[6] Peter E Kaloroumakis and Michael J Smith. "Toward a Knowledge Graph of Cybersecurity Countermeasures". In: (), p. 11.

[7] Ivan Kovačević and Stjepan Groš. "Red Teams - Pentesters, APTs, or Neither". In: *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*. ISSN: 2623-8764. Sept. 2020, pp. 1242–1249. DOI: 10.23919/MIPRO48935.2020.9245370.

[8] R A H Lahaye. "How to Spot the Blue Team". In: (Mar. 2018), p. 20.

[9] McAfee. *What Is a Security Operations Center (SOC)?* URL: https://www.mcafee.com/enterprise/en-us/security-awareness/operations/what-is-soc.html (visited on 10/14/2021).

[10] MITRE. *ATT&CK®*. URL: https://attack.mitre.org/ (visited on 09/23/2021).

[11] Security Risk Advisors. *BSides PGH 2018 – Heavy Machinery and Burly Lumberjacks and Logging! Oh My!* Dan Astor Evan Perotti. June 2018. URL: https://sra.io/blog/bsides-pgh-2018-heavy-machinery-and-burly-lumberjacks-and-logging-oh-my/ (visited on 09/13/2021).

[12] Marc Smeets. *Introducing RedELK – Part 1: why we need it | Outflank Blog*. Feb. 2019. URL: http://www.outflank.nl/publications (visited on 10/07/2021).

[13] Marc Smeets, Mark Bergman, and Lorenzo Bernardi. *Red Elk*. original-date: 2018-10-03T15:55:05Z. Sept. 2021. URL: https://github.com/outflanknl/RedELK (visited on 09/24/2021).

[14] Sathya Chandran Sundaramurthy et al. "A Tale of Three Security Operation Centers". In: *Proceedings of the 2014 ACM Workshop on Security Information Workers - SIW '14*. Scottsdale, Arizona, USA: ACM Press, 2014, pp. 43–50. ISBN: 978-1-4503-3152-4. DOI: 10.1145/2663887.2663904. (Visited on 10/11/2021).

[15] Rob Van Os. *SOC-CMM: Designing and Evaluating a Tool for Measurement of Capability Maturity in Security Operations Centers*. 2016. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:ltu:diva-59591 (visited on 09/20/2021).

[16] Joe Vest and James Tubberville. *Red team development and operations: A practical guide*. Independently Published, 2020. ISBN: 9798601431828.

[17] Overheidsbrede Cyberoefening & Webinars. *Red teaming in de praktijk*. Tech. rep. Oct. 2021, p. 38.