A SOC-MSSP guide

From reporting malicious activity to catching it as it happens

Gaston MARTIN

https://www.linkedin.com/in/gaston-m-7b742118a/

Summary

1. Introduction

The idea for this guide came out of the deep frustration of knowing what a SOC could do while witnessing what SOCs and MSSPs actually do, seeing what they claim to do - internally or to their customers, respectively - and hearing the ideas for improvement actually approved by the management.

When its needs are met and all the aspects of the SOC are correctly built, the SOC is an incredibly powerful defensive entity that can identify and stop - directly or indirectly - internal and external threats from damaging the environment.

However, if these needs are not met or if the SOC is poorly built in some way, then it can appear to be very ineffective and a waste of money from an external - executive, for example - point of view.

From an internal point of view, this leads to high frustration for the staff because they know that and how they could do better, but they are limited doing unattractive work. Most of the time they would try hard until they give up and go for a more attractive job, like in a CERT or CSIRT - where if someone talks to them, something actually happens, so they feel valued.

Many SOCs and MSSPs as they are currently built focus only on the time constraint for detection and response. This is indeed the most important one, because the more time passes as an attack goes undetected, the more damage will come to the company. However, these SOCs were built based on other SOCs from a time with fewer security needs, or at least concerns, and to stay competitive in a market with an economic model from that time. Times have changed, technologies and people as well, and now these SOCs and MSSPs struggle to stay relevant from a security point of view. Even more concerning, they have a hard time hiring people, and an even harder time keeping skilled analysts, as the way they are built creates unattractive work. Some of them try to distract analysts with task rotation, or some interesting project in parallel to their main missions. This isn't helping the analysts, who can only stand it for so long, and isn't helping these SOCs or MSSPs either, because they are stuck in a model belonging to the past.

Most of the current guides for building SOCs explain what to do and what not to do to achieve a SOC or MSSP as they exist today and while they usually cover a lot of the aspects - especially for the bigger guides out there - they fail to address all of them with sufficient details to actually build something that will have a strong enough base to go all the way.

One of the main goals of this guide is to explain in a simple - i.e. not deeply technical - way to executives what a SOC actually does and how to improve it so that all - executives, management and SOC staff - can benefit from it.

In the faint hope of general improvement and at least for another reference to which Blue Team members could point, this guide was born.

This guide explains the following:

- The differences between a SOC and a MSSP across human, financial, operational and technical aspects
- The goals, needs, strengths and weaknesses of the current models of MSSP
- What is needed to build a SOC or MSSP that can scale up in time
- What to improve to maximize a SOC efficiency to detect and capacity to respond to threats

1.1. Definitions

As the security landscape is constantly evolving, new teams, jobs, services and tools are created and the definitions change, sometimes to the point where differences become almost philosophical.

Therefore, writing some definitions at the beginning of this document will make it easier to understand as it won't be anchored in time.

1.1.1. SOC

A Security Operations Center (SOC) is a combination of people, skills, tools and processes that monitors an environment for cybersecurity threats.

Its main goal is to detect anomalies occurring in the environment in order to respond to them in the proper way. The response comprises:

- an investigation to determine whether the anomaly was a threat and its extent if it were
- a remediation, if there were indeed a threat and if this task had been attributed to the SOC

The SOC needs to know and understand the environment it monitors well in order to correctly detect anomalies and to respond to them.

This knowledge and understanding includes everything from the buildings in which a device is connected to the environment, to the people in IT teams, and to the actual business of the company.

In order to detect anomalies, the SOC uses artifacts produced by the different devices in the environment it monitors. The majority of these artifacts are logs created every time an event happens by the applications running on the devices.

Every source generating artifacts useful for the SOC is called a sensor. Therefore, a sensor can represent a piece of hardware, such as a physical firewall filtering network traffic, or a piece of software, such as a web server hosting websites.

In order to speed up and facilitate the detection of anomalies, the logs created by the sensors should be gathered and centralized into a SOC's most powerful tool: a SIEM. The SIEM is, in a way, a database storing the logs, on which specific queries are run to single out anomalies.

When an anomaly is detected, the SIEM creates an alert, and the SOC's response starts. An analyst investigates the anomaly to determine whether there is a security incident, and if the actions are malicious. The analyst logs all the findings in the SIRP for traceability.

The SIEM and the SIRP are the two main tools of a SOC, and have been for quite some time. Once these tools are in place, the next step is to use a DevOps platform, and then a SOAR. The former focuses on infrastructure and detection improvements, while the latter enables automated response.

The tools and the inner workings of a SOC are discussed at length later in the document.

1.1.2. MSSP

A Managed Security Services Provider (MSSP) is an entity that sells different security services to its customers - as per Gartner's definition - usually a SOC is included in these services.

Now, Gartner's definition of MSS uses the customer's point of view, whereas this document presents the MSSP from the provider's point of view - while of course taking into account the customers' needs.

Focusing on the SOC service - which is the biggest part in all aspects - of an MSSP, the goals, needs, strengths and weaknesses of the MSSP are more or less the same as those of a SOC, multiplied by the number of customers to which the MSSP provides services.

The main goal of the MSSP is - like a SOC - to detect anomalies occurring in the environment, but most of the time the MSSP will not have any means of remediation in its customers' environments.

Therefore its response will be limited to communicating to its customers an incident ticket that comprises:

- a context - i.e. how and why the anomaly was detected in the first place
- an investigation with facts, analysis and conclusion
- remediation recommendations - where an internal SOC would be able to remedy on its own
- impacts explanations - to help the customer understand the risks and prioritize their actions

The needs of an MSSP are the same as those of a SOC in terms of understanding the environments it monitors but to minimize cost and maximize efficiency, the MSSP tries to mutualize as much as it can across its customers (people, hardware, software, templates, etc).

Of course, this is quite a lot harder to achieve as an MSSP because a strong knowledge management is mandatory. Indeed, poor knowledge management will result in an amount of time loss growing at a larger rate than the number of customers - i.e. the difficulty to find a specific information grows at a larger rate than the volume of data if the data is unstructured.

This time loss will, in turn, impact the efficiency of the analysts and this will result either in a loss of productivity or quality of the work done, or even both - not to mention the mental impacts on these analysts.

Finally, this will show up in the finances one way or the other: less margins because the ratio of analyst-to-customer grows faster than the number of customers for the same quantity/quality level, customer loss due to dissatisfactions on quantity, quality or price paid for the level of service, etc.

The "bad news" on the needs of an MSSP are met by equally - arguably greater - "good news" on its strengths. Where a correctly built SOC is a powerful defensive entity, a correctly built MSSP can simply be the best defensive entity, period.

From a detection perspective, having multiple, different environments with different sensor types and editors is a gold mine for detection use cases ideas and implementation: when built correctly, any upgrade on detection made by the MSSP can instantly be applied to all its customers, therefore mutualizing all the R&D efforts - which leads to better skilled analysts, reduction in overall R&D costs, customers satisfaction, etc.

From an investigation perspective, a higher number of environments is also beneficial. The R&D mentioned above will translate into better tuned detection rules which will fire fewer alerts that end up being False

Positives (FP, i.e. alerts that shouldn't have been fired in the first place). This directly lessens the time spent on "pointless" investigations, so the analysts work on more pertinent ones, increasing their skills and motivation- this also means that the analyst-to-customer ratio goes down as there are fewer alerts per customer.

If these needs are not met or if the MSSP is poorly built, then, the cost for the MSSP would be higher and the attractiveness for its analysts lower, to the point where it would be more cost efficient for the MSSP to maintain one SOC per customer and just be, in fine, a juxtaposition of SOCs. This juxtaposition of SOCs is the complete opposite of the mutualization wanted by an MSSP.

In this worst case scenario, the MSSP would appear ineffective and a waste of money internally, but also for its customers. Its skilled analysts, like in a SOC, would eventually give up to find a more attractive job in a CERT or CSIRT.

1.2. SOC/MSSP vs CERT/CSIRT

While today's SOC and MSSP are mostly focused on detection and investigation - with sometimes means of remediation - the incident response is the main job of a Computer Emergency Response Team (CERT) or a Computer Security Incident Response Team (CSIRT).

Because of the struggles of today's SOC and MSSP - briefly mentioned before - the attractiveness of a SOC analyst job is less that than of a CERT analyst: from a technical perspective, it is both more challenging and rewarding to work on an actual incident caused by a malicious actor than to chase False Positive alerts from misconfigured detection rules.

Moreover, CERT analysts have a real sense of purpose in their job as, again, they work on actual, tangible issues where SOC analysts perform repetitive tasks without much hope for improvement.

Adding up all of this to the fact that the mindset and skills needed for a SOC or CERT analyst are very similar results in the creation of a vicious circle:

- A CERT is more attractive to analysts than a SOC job-wise, so higher skilled analysts tend to end up in a CERT.
- Higher skilled analysts leave SOCs, therefore lowering the overall SOCs skills, hence their overall capacity to properly perform their tasks.
- SOCs do not perform as well as executives had hoped, so a highly skilled CERT is needed as "last resort" when - and not if - the SOC does not detect and react in time and a crisis arises.
- Security funds are put into the tools, training/certifications and salaries of CERT analysts to make sure they are up to the task when it comes.
- Since the funds allocated to security are finite and prioritized towards the CERT, the SOC budget is that much smaller, therefore the tools, training/certifications and/or salaries of its analysts are lower than what they could be.
- A lower-budget SOC means a less mature SOC, directly translating into less attractive tasks for its analysts, which, at last, deepens the attractiveness gap between SOC and CERT.

From this observation, the - somewhat controversial - statement that good CERTs exist and are needed only because bad SOCs and MSSPs exist could be made.

It could therefore be argued - and more or less is in this guide - that a good SOC or MSSP would not only reduce the need of a good CERT, but also minimize the consequences of actual incidents as the earlier they are caught and dealt with, the less they directly and indirectly damage the environment, hence the company.

In order to avoid any misunderstandings: there always will be a need for security detection and for incident response, but it would be better for everyone if the entities tasked with defensive security could be more proactive and not only post-mortem reactive.

It has long been clear that it is better for everyone to detect, catch and manage the fires as they start rather than waiting until they have burned everything to the ground to investigate the cause and rebuild everything.

Why not do the same with security incidents?

2. Human aspects

This chapter tries and explains as simply and clearly as possible the tasks of SOC analysts, the skills required - both hard and soft, their needs and the most common mindset this particular population has.

In order for the SOC to consistently attain the goals set by the executives and since the actual intelligence and effectiveness of the SOC is defined by its crew, understanding the humans behind the job titles or employee numbers is of utmost importance.

Doing so at all times, whether it be when recruiting, managing or planning mid/long term strategies is the only reliable way to ensure the best efficiency possible.

2.1. Managing a SOC

Every company, private or public, has goals to achieve and whatever the motivations or how often or pertinent the goals are, they mark the desired state and time available to reach it from the current state.

Once this is clear, one or more strategies are imagined and one of them is adopted. Each strategy requires actions to be done in a timely fashion and those actions and timetable can be broken down into resources. The human resources each have a set of skills and a workload capacity that will, according to the strategy, enable the company to reach its desired state in time, therefore fulfilling its goals.

This is all very high level because it is not the point of this document but the important thing is that **company goals ultimately translate into human resources** (and other material needs).

From a high level - executive/HR - perspective, a SOC (or MSSP) is, like any other department, viewed in terms of goals and resources.

However, as explained in the following subchapters, the hard skills required to do most of the tasks in a SOC, the soft skills needed to work efficiently as a team and the mindset to keep calm under pressure combined with the current (most of the time) amount of "busy work" compared to "interesting work" and the lack of trained professionals make it so that one SOC analyst is not equal to another one and they have plenty of available opportunities that literally present themselves to them on a daily basis.

This means that it may be easy-ish to attract SOC analysts from other companies because they may not be happy with their current situation, but it is actually really hard to keep them.

The amount of turnover is crucial in a SOC or MSSP because they both need strong teamplay in order to perform well - or at least to the level expected by executives - so the overall skill of a SOC is far from being the mere sum of the skills of its members.

Therefore, to maintain a strong cohesion, there must be a constant clear majority of members that form a stable core which can absorb newcomers and remain as-is even when it loses people.

In other words, if you replace one player of a sports team without the team having time to train as a whole - even if you take the best player, a well trained team's performance may not vary much. Now, if you replace 20-30% (the more, the worst) of the players over a short period, then it is foolish to expect the team to stay at the same level.

This is why good management is needed. In this context, the human component has to be taken into account, as simply managing human resources will not cut it.

Each member of the SOC - analyst or otherwise - has to be understood or at least heard and compromises must be made so that both parties win - that is to say the company and the employee.

In the best case scenario, the SOC managers are actually leaders, the difference being that a leader leads by showing the example and inspiring his/her subordinates where the manager leads by distributing resources wherever needed to attain the objectives set. These leaders would naturally take into account the human component in their job and come to HR/upper management with solutions in the form of compromises.

Most likely, the SOC managers are "regular" managers and they would inevitably be facing issues by not hearing or acting on their teams' needs and complaints.

In both cases, HR has to hear either the compromises or the issues and help the SOC managers find and settle compromises that **would benefit everyone** - this means that the employees also need to see an actual gain; it does not have to necessarily be a big one, but they need to see and know that the company is willing to move a step in their direction.

All in all, the most important thing to remember is that upper management and HR have to clearly identify the type of managers the SOC has and help them build and maintain a strong cohesion by listening to and hearing the people to find compromises.

This may be a bit of a catchy phrase but it completely applies in this case: it’s time to put the “H” back in “HR”.

2.2. SOC analysts’ jobs

Depending on the SOC or MSSP size and the missions it is expected to perform, the exact jobs and tasks may vary. The table below shows an example of the most common tasks, the skills they require and jobs that can exist by assembling these tasks.

SOC job	Job tasks	Tasks details	Hard skills	Soft skills
Project manager	Project management	Manage customers' SOC related projects (new customers, migrations, new sensors...)	SOC working knowledge General IT and security controls knowledge Cybersecurity standards knowledge	Good listener Organizational skills Diplomatic
	Customer follow-up (project)	As the customer's project focal point, centralize and handle all demands pertaining to the project		
	Customer regular committees (project)	Prepare, attend (and lead) the meetings related to the project		
Service Delivery Manager (SDM)	Client follow-up (run)	As the client's usual focal point, centralize and handle all demands pertaining to the services subscribed	SOC working knowledge General IT and security controls knowledge Cybersecurity standards knowledge	Good listener Organizational skills Diplomatic
	Client regular committees (run)	Prepare, attend and lead the meetings related to the services subscribed		
Operational analyst	SIEM rules configuration (initial)	Deploy, configure and check new detection rules onto the SIEM	SIEM(s) technology(ies) advanced user SIRP(s) technology(ies) advanced user SOAR(s) technology(ies) advanced user CI/CD/CD tool(s) user SOC working knowledge Digital investigations	Collaborative skills Rigorous
	SIRP reporting configuration (initial)	Configure and check that the SIRP operates as expected (workflow, templates, SLAs…)		
	SIEM rules and configuration MRO	Perform updates and fixes on the SIEM's detection rules and front-end configuration		
	SIRP reporting and configuration MRO	Perform updates and fixes on the SIRP's front-end configuration		
	SIEM dashboards development	Create, maintain and improve dashboards used in the SIEM		

	SIRP dashboards development	Create, maintain and improve dashboards used in the SIRP		
	SOAR dashboards development	Create, maintain and improve dashboards used in the SOAR		
SOC manager	Financial management	Manage expenses so they fit into the budget (regular SOC) or so the margin meets the objective (MSSP)	Budgeting Strategic planning People management Knowledge of all SOC's jobs and purpose Knowledge of SOC needs and limits General IT and security controls knowledge Cybersecurity standards knowledge	Decisive Good listener Organizational skills Prioritization skills Diplomatic Rigorous Critical thinker Can work under pressure Can delegate
	Human management	Manage the people working in the SOC (career evolution, performance review, recruitment...)		
	Customers and partners management	Discuss SOC strategic issues and opportunities with (internal and external) customers and partners		
	SOC governance	Steer the SOC in the direction decided with upper management by planning and overseeing projects		
Team leader	Operational management	Manage day to day activities so that the team delivers the expected quantity at the expected time	Operational planning Skills of the team (quality control) Knowledge of other SOC teams' jobs	Decisive Good listener Organizational skills Prioritization skills Diplomatic Rigorous Can work under pressure
	Quality Control	Ensure that the team, as a whole, delivers to the expected quality standards		
System administrator	SIEM infrastructure setup	Deploy and check the SIEM infrastructure – i.e. software and devices that are part of the SIEM	OS (depending on env) administrator SIEM(s) technology(ies) administrator SIRP(s) technology(ies) administrator SOAR(s) technology(ies) administrator CI/CD/CD tool(s) advanced user	Collaborative skills Rigorous Critical thinker Prioritization skills Can work under pressure
	SOAR infrastructure setup	Deploy and check the SOAR infrastructure – i.e. software and devices that are part of the SOAR		
	SIRP infrastructure setup	Deploy and check the SIRP infrastructure – i.e. software and devices that are part of the SIRP		
	SIEM software configuration (initial)	Once the SIEM infrastructure is deployed, configure the software so it matches the projects specifications and is ready for use		
	SOAR software configuration (initial)	Once the SOAR infrastructure is deployed, configure the software so it matches the projects specifications and is ready for use		
	SIRP software configuration (initial)	Once the SIRP infrastructure is deployed, configure the software so it matches the projects specifications and is ready for use		

	SIEM software MRO	Perform updates and fixes on the SIEM's software and back-end configuration		
	SOAR software MRO	Perform updates and fixes on the SOAR's software and back-end configuration		
	SIRP software MRO	Perform updates and fixes on the SIRP's software and back-end configuration		
	Security solutions MRO	Perform updates and fixes on other security solutions managed by the SOC/MSSP		
	SOC-used servers MRO	Perform updates and fixes on other servers and solutions used by the SOC		
SecDevOps	SOC tools development	Create, maintain and improve the custom tools used by the SOC	Software design Proficiency in the language(s) used SOAR(s) technology(ies) administrator CI/CD/CD tool(s) administrator	Collaborative skills Rigorous Critical thinker Can work under pressure
	SOC automation development	Create, maintain and improve specific scripts and other automation solutions used by the SOC		
	SOAR playbooks configuration (initial)	Deploy, configure and check SOAR playbooks on a new instance		
	SOAR playbooks and configuration MRO	Create, maintain and improve the SOAR "atomic"/"building blocks" playbooks – i.e. atomic actions that can later be assembled		
Security architect	SIEM infrastructure design	Create the SIEM infrastructure specifications upon which deployment will be based, in light of the customer's use case	Networks working and hardening knowledge Systems working and hardening knowledge SOC working knowledge Cybersecurity standards knowledge SIEM(s) technology(ies) administrator SIRP(s) technology(ies) administrator SOAR(s) technology(ies) administrator	Collaborative skills Rigorous Critical thinker Constructive
	SOAR infrastructure design	Create the SOAR infrastructure specifications upon which deployment will be based, in light of the customer's use case		
	SIRP infrastructure design	Create the SIRP infrastructure specifications upon which deployment will be based, in light of the customer's use case		
Detection specialist	SIEM detection rules R&D	Create, maintain and improve SIEM detection rules	Digital incident response Adversary tactics and techniques knowledge SIEM(s) technology(ies) advanced user CI/CD/CD tool(s) user Security solutions knowledge	Collaborative skills Rigorous Critical thinker Can work under pressure
	New sensor study and integration	Study the features and limitations of the sensor to map potential coverage and create or update SIEM detection rules		
	Security solutions configuration	Create, maintain and improve precise guidelines for security solutions configuration – i.e. prevention, detection and logging		

Response specialist	SOC alerts triage	Investigate to determine whether the alert is a False Positive or a True Positive	Digital incident response Digital forensics Adversary tactics and techniques knowledge	Decisive Collaborative skills Rigorous Critical thinker Can work under pressure
	SOC incidents investigation	Investigate True Positive alerts (i.e. incidents) to find the full extent in time and space		
	SOC incidents remediation	Remediate the incident – either alone or by working with the proper system/network/application teams		
	SOC internal trainings	Perform internal SOC response trainings for other response specialists		
	Response procedures development	Create, maintain and improve SOC response processes and procedures – i.e. those used by human, translatable into playbooks		
	SOAR response playbooks development	Create, maintain and improve the SOAR response playbooks – i.e. assembling "building blocks" to achieve the desired response		
Threat hunter	Threat hunting	Hunt for known and unknown threats across customers' environments – this IS a full-time job	Digital investigation Adversary tactics and techniques knowledge	Collaborative skills Rigorous Critical thinker Can work under pressure

The jobs and tasks detailed in the table are merely examples of those that can exist in a SOC or MSSP. The tasks that add up to a job are always based on the needs, which depend on the context - and so does the number of people needed per job.

The hard skills shown in the table are not exhaustive, they are limited to the "security" aspects which are the essence of a SOC. Keep in mind that skills such as "Digital investigation" require both the knowledge and know-how of a digital investigation and the knowledge of how the underlying environment actually works. For example, to determine if a behavior is legitimate on a Windows system, one must know how Windows works to know what to look for and if such a chain of events would more likely be the result of malicious or legitimate intent, once the proper artifacts are gathered.

Ultimately, the jobs and tasks have to be tailored in a realistic way, meaning that there is actually a team member that can take up these tasks and has sufficient time to complete all their assignments. This is especially true for the technical skills needed in the long run: a "does it all" position will result in mediocre performances for the person holding it and/or a high amount of time (and money) spent training.

2.3. SOC analysts' needs

The tasks and jobs presented above require diverse skills, including highly technical ones.

This is due to the fact that in order to find anomalies, qualify incidents and perform remediation actions, a SOC (or CERT) analyst has to have a deep understanding of "normal operations" to single out the "abnormal operations" and cut out the malicious activity, preferably without interrupting production.

This is true for every asset monitored by the SOC, hardware or software, systems, networks, Cloud based activity, IoT, OT…

Just to be clear: it is nowadays obvious for everyone that each item of the previous list would need a different person (or team) with special skills to deploy or maintain, but everybody expects *the SOC* to monitor all of these and promptly and correctly respond to any and all malicious activity.

Well, if it is not expected for one person (or team) to manage multiple scopes because each one has its own specificities, then how can it be expected for one person (or team) to **understand** - meaning knowing why this or that was done this way and how it should **normally** behave - multiple scopes **and** be able to restore of fix whatever may have been damaged or misconfigured?

The simple answer is that it is not humanly possible for one person to have, at all times, an understanding deep enough of all those things to perform the job expected of a SOC.

This is why a SOC is mainly teamwork and why a SOC analyst needs to spend most of his/her time reading through documentation, understanding *what* they are looking at and comparing the behavior - mainly a sequence of events - of what they are investigating to a baseline of a "normal" behavior.

For the bigger SOC (MSSP) out there, there can be people specialized in one specific field who are called whenever needed, but for the majority of SOC and MSSPs, the analysts have to be able to do most of their investigations and remediations by themselves with the occasional support of other colleagues.

Therefore, the SOC analysts have some needs in order for them to perform at the expected level:

- A well organized, up-to-date, very accessible knowledge repository enables all SOC personnel to minimize the time and effort put into open source research. For this to work, every analyst needs to contribute - this would only happen if the repository is well structured and accessible and if the SOC managers keep showing and reminding everyone that it actually helps a lot. For bigger SOCs/MSSPs, it is strongly advised to use the services of a knowledge manager whose actual job is to make information easier to access and update.
- Regular and recurrent security training - internal or external - is very important to maintain a high level of awareness for analysts, build up cohesion and individual skills. The more the tasks given to the analysts are repetitive and "basic" (not technically advanced), the more regular training is needed.
- Security training is fine, but as stated before, the security part is only the tip of the iceberg. The SOC analysts also need training for every scope they are tasked to monitor in order to understand them and more importantly differentiate normal from abnormal. All SOC analysts do not need to follow all training, but there have to be enough analysts trained for **every scope at all times** so that the knowledge is there - both up-to-date in the knowledge repository and in the minds of some analysts that are currently working. This ensures that the SOC responds correctly and in a timely fashion, whatever the security issue may be.
- Finally, there need to be regular crisis simulations - both advertised and not advertised as an exercise - so that every analyst has experienced the pressure and the atmosphere of a crisis and that everyone knows their place and tasks in "crisis mode". This is the only way to make sure that when an actual crisis happens, there is only a cybersecurity issue and not also a simultaneous "oh my god what are we supposed to do"/"headless chickens running around" issue.

Usually, the further down the list, the less likely it is that the point is being well addressed. Therefore it is important for upper management, HR and SOC managers to understand and keep in mind that the SOC analysts need all of this to correctly do their jobs.

2.4. Understanding the profiles

The SOC managers have to know the profiles or their analysts in order to make the best of their teams.

Knowing what to aim for for the SOC as a whole, what the needs for the analysts are, what tasks can be given, what skills they require and how they can be added up to make coherent jobs is good - arguably better than what exists in some places today - but it is not enough.

To truly maximize the efficiency of the team members and, in the end, that of the SOC itself, there has to be a job and a place for everyone in the team and everyone in the team has to have a job and a place.

In other words, the company and the employee both need to be happy with the tasks assigned and the job done by the employee - this often means compromise.

A good way to reach an acceptable compromise is for the SOC managers and HR to understand the profile of the analyst - i.e. what makes them tick, explain to the analyst what the SOC needs and figure out together how to join both ends. If there is actually no common ground between what the analyst wants and what the SOC needs then it is maybe time to reassign the analyst elsewhere.

This may come across as crude, but motivation is key for the analyst to keep their skills at the expected level and for them to work well with the rest of the team. A SOC simply cannot afford internal conflicts or people that let themselves get carried by the flow, as the former simply destroys the cohesion and the latter is just dead weight that needs to be carried by the rest.

There are mostly two types of technical profiles for which a SOC or MSSP should look and one management profile. This of course covers only the SOC analysts and managers; other profiles may be needed for other jobs in the SOC.

Before talking about the management profile, here is a quick reminder that **a great technician** - i.e. person with high technical skills - **rarely makes for a great manager** or even a mediocre one, for that matter. Indeed, this cannot be said nor emphasized enough: **management is not, never has been nor ever will be, an evolution of technical expertise**.

A good SOC manager is typically someone curious, willing to try and understand technical subjects - at least enough to get the issues and possible solutions, with a good strategic vision and capacity for compromise. Basically, someone who is able to find a working compromise between the objectives set by the executives and the technical issues that the SOC is facing so that the SOC moves forward towards the objectives, even if the pace is slowed by technical issues.

In other words, a good SOC manager is a “Yes, but” person - meaning that everything could be achieved, but at some cost.

There are two main types of technical profiles that can be of use in a SOC.

The first one is the motivated and curious type: they most likely went into cybersecurity out of passion or interest for learning. The people with this profile are always asking “why” and “how” questions about every subject so they can truly understand what is happening. These make for good and great analysts because when they are responding to an incident, they will not rest until they are satisfied that they have understood exactly what happened and how so they can properly remedy the situation. Also, they are the type of people a SOC needs to better itself: because they like to learn and keep learning, they are often people that easily lose interest in repetitive actions in the sense that for them, any and every action they have to repeat without any added value is superfluous and therefore can and should be automated. In other words, they will try and optimize everything around them so that they can focus on what they want: learning stuff.

The downside is that they need to be kept in check and reminded that there are objectives and production priorities. They indeed tend to deep dive into subjects and sometimes the dive is too deep, taking too much of their time compared to what would have been acceptable from a production cost perspective.

The second type of technical profile is disciplined and hardworking in the sense that they don’t mind - or in some cases they even enjoy - repetitive tasks in a structured context with procedures to follow to the letter. For bigger SOCs and MSSPs, they can make the bulk of the teams tasked with investigations and remediation or even MRO because the job they do and the results they produce are very stable and reliable; that is exactly how a SOC needs to be. Also, if they are assigned tasks that suit them, they tend to be quiet and generally happy. That’s why it is extremely important to listen to and hear them whenever they make remarks or propose some kind of improvement about these tasks.

The people matching this profile often have less advanced technical skills than the other profile because they tend to dislike change, either in the tasks they are assigned or if they have to learn new things. It usually takes time and energy to make them pick up new skills or change their routine.

Both technical profiles are very important for a stable, reliable SOC that keeps on improving both the quality of its detection and response and the quantity of incidents per unit of time it can handle.

Of course and again, good management is key to put the right profiles on the right jobs, have all the profiles synergize well within the SOC and accurately convey the impacts of technical issues up the chain while explaining the “bigger picture” to the analysts.

2.5. Conclusion

A SOC or MSSP is a complex entity that requires a variety of people with specific profiles to perform how it is expected. All the profiles have to be correctly identified and positioned on the correct job, from the SOC analysts to the SOC managers. Therefore, it is important that HR helps the SOC managers manage their human resources and that both the upper management and HR pick the right kind of profile for the SOC managers together.

As a reminder: there is such a thing as technical expertise and it needs to be acknowledged - both in the missions and in the salary - and said technical expertise is the only way to improve technically.

Some of the current SOCs and MSSPs struggles are not having the right people in the right places, putting technical experts in management positions, leaving the MSSP managers battling the customers alone and/or having internal battles between the executives, HR and the SOC or MSSP managers instead of the formers helping the latter.

This leads to frustration, SOC managers leaving for another position with less pressure or a higher pay for the same amount of stress, SOC analysts corresponding to the first profile leaving for offensive or CERT/CSIRT positions and finally SOC analysts of the second profile leaving because as they stayed the longest, they are asked to do tasks they don’t want nor like.

3. Financial aspects

This chapter covers what may well be the hardest part of managing a SOC: making good decisions on where to spend the money. Both SOC and MSSP have limited budgets, in one case it is limited by whatever has been allocated to it and in the other case by its sales revenue and the margin objectives that have been set.

The hard part isn't to manage to stay within the budget - although it can be challenging at times - the hard part is to carefully plan the spending so that the SOC is actually relevant in its missions long term, mainly when it comes to detecting attacks early to prevent damage. This is hard because it takes a combination of skills and knowledge that are rarely held by one individual:

- A strong understanding of the needs and expectations of the executives and/or customers and good communication to explain to them the choices made.
- A detailed knowledge of how the SOC in its current state operates, both internally and with other entities - especially its strengths and weaknesses.
- A clear vision of what skills and levels each job needs and the actual skills and levels the people holding these jobs have.
- A precise knowledge of what profile each member of the SOC has, what they can and can't do, what they could or couldn't do and what they would or wouldn't do.
- A deep technical understanding of the mid/long term impact of any decision made, whether that decision regards people, tools, training, or interaction with other entities

In practice, most of the time, the decisions are made by executives with the information at their disposal, which is mostly hearsay about what others do and/or what the customers want. Oftentimes the decisions end up being about buying/using a new tool or adding some junior staff (or firing senior staff in hard times) and that seems logical, because when you spend your time handling money and projecting costs and revenues, the solutions you come up with are money related.

However, these decisions rarely solve anything and often add to the issues: even a very expensive tool will never be used to its full potential/worth its cost with its default configuration, and very good junior staff still misses the experience of senior staff, because experience isn't something you can learn in books, it's something you gain with practice.

As seen in the previous chapter, a SOC or MSSP is a complex entity because of its highly technical environment and the human relations it requires to operate properly. Therefore, its actual worth resides within its crew and only by exploiting this wealth - and/or working to grow it - can it be and stay relevant long term. One way to do just that is through industrialization by focusing the knowledge and experience of the senior staff into normalization, standardization and automatization of the processes and tools used by junior staff.

3.1. Balance

In order to establish the best course of action to meet the short term objectives while building for the long term, one can start by thoroughly examining the balance of expenses vs revenue for the SOC or MSSP.

For this exercise to be meaningful and yield actionable results, the different items must be meticulously detailed, especially when it comes to time spent on activities and missions because the goal is to determine what could and should be industrialized and with what priority.

3.1.1. Expenses

The tables below show an example of expenses that are often overlooked in an MSSP environment, but end up costing more the bigger the MSSP is. The list of items gives an idea of what to look for, but is not exhaustive in any way, and although it has been made with an MSSP environment in mind, many items are still relevant for a SOC.

To populate the "average cost of one unit" column, it is preferable to include the workforce cost, the cost of the infrastructure needed to perform the item and the cost of any tools and other licenses needed. The goal here is to get a representation of the actual cost of production for one unit of that item.

It could be obvious, but in case it isn't: it is good to have approximations for each column of each item, but it is better to have actual metrics and it is best to get those metrics per month to be able to monitor and project evolutions.

Keep in mind that this is not a theoretical exercise and that it has to be done carefully as having a precise view of expenses leads to pinpointing the wastes and minimizing the wastes is a crucial step in a long term plan.

The first table represents immediate and recurrent expenses depending directly on how the SOC/MSSP is structured, the choices of tools and technologies made, the workforce and the contracts with its customers.

Item	Average number of units per month	Average cost of one unit	Average monthly cost
Infrastructure deployment and configuration	h/month	€/h	€/month
Penalties on late deployment deliveries	length of delay / month	€/period of delay	€/month
SOC tools and servers MRO	h/month	€/h	€/month
Health check for SOC tools and servers	h/month	€/h	€/month
Detection rule engineering (first deployment)	h/month	€/h	€/month
Detection rule engineering (subsequent deployments)	h/month	€/h	€/month
Detection rule fixing (per environment)	h/month	€/h	€/month
Pointless/useless investigations (e.g. recurrent False Positives alerts)	h/month	€/h	€/month
Penalties on SLAs breach	nb of breaches / level/month	€/level of breach	€/month
Training on internal tools and methodologies (i.e. learning environment specific items mandatory to perform the job)	h/month	€/h	€/month

The second table lists delayed or indirect costs that should never happen in a perfect world. These are harder to measure in terms of actual cost, but they tend to be way higher when they happen.

This is especially true long term with items like repercussions on the SOC (and company) reputation due to customer dissatisfaction or even more so due to error, incompetence or negligence.

Item	Average number of units per month	Average cost of one unit	Average monthly cost
Late or no detection (False Negative) of an attack	nb/month	€	€/month
Late or no detection during a certification or audit	nb/month	€	€/month
Handover latency in incident response to CERT or other (which has negative consequences on the actual impact of the attack)	nb/month	€	€/month
Potential new businesses loss due to resources unavailability (i.e. lack of people in the SOC)	nb/month	€	€/month
Potential new businesses loss due to poor service quality (i.e. bad reputation of the SOC or company)	nb/month	€	€/month
Customer loss due to resources unavailability (i.e. lack of customer service)	nb/month	€	€/month
Customer loss due to poor service quality (e.g. recurrent bad detection or response)	nb/month	€	€/month

The tables listing expenses shown above are not exhaustive and are merely pointers to how one should list expenses of one's SOC or MSSP. Ideally, every task performed by every person in the SOC should have its own entry and be measured as it is the best way to lead to questions like "Why are we doing this?", "Why is this task taking so much time?" but also "Why are we not doing that?".

These questions and their corresponding answers should by themselves go a long way towards prioritizing the resources allocated to internal projects and give a clear and precise view of what and where the issues are.

3.1.2. Revenue

The list of revenues for a SOC is often limited to the budget it is allocated by the company.

However, the list for an MSSP can be a bit longer, although its customers' service subscriptions should make up the bulk of it. Let's explore a few ideas that could add to the revenue.

The most obvious one depends on the type of subscription or contract the MSSP has with its customers. One-time projects such as adding a new sensor to monitor, creating new detection rules / use cases or upgrading the already existing detection rules and use cases could be sold on top of the subscription. These could also lead to higher subscription revenue because of the extended perimeter monitored. Again, depending on the contracts, things like a complete response (and not just the investigation part) or advanced investigation (e.g. with forensics) could be in the service catalog.

Other ideas that could expand revenue is leasing a dedicated resource to a customer for an agreed upon amount of time, performing external (to the SOC/MSSP or even the company) training or any other creative way to directly sell the skills of the SOC personnel and bettering the SOC or company reputation.

Since most of the revenue for a SOC or MSSP is not controlled by it, in the sense that the SOC isn't the entity that negotiates the prices with the customers, it is very hard to optimize anything in terms of revenue beyond suggesting new kinds of services like those mentioned above.

However, expenses optimization is a real option to consider in order to create breathing room in the budget.

3.2. Hidden costs and optimizations

Like in most entities, there are costs in a SOC or MSSP that are somewhat hidden and there are some ways to optimize these expenses.

The following points are not specific to a SOC and can be applied to other entities. They do not constitute an exhaustive list either and such items should be carefully considered by establishing a strategy for each, because they have a long term impact on production costs and quality.

- Evaluate precisely the cost of training a new recruit for each position in the SOC. This way when someone is asking for a training, a raise or wants another position, it will be very easy to balance the potential loss of this person - individual skills, performances, mindset, internal working knowledge on top of which the cost of recruiting someone new and training him or her - versus the cost of meeting the demands.

  In many cases, it is actually far cheaper to negotiate and lean towards the demands than it is to take the loss and hire someone else. This is work, so it boils down to the math of the position's worth, both for the company and the employee and there shouldn't be any emotions, hurt feelings or letting things get personal here.

  This circles back to the previous chapter about knowing the people working in the SOC and their profiles, with an added cost/benefit approach that could very well be explained to the person at risk: "From what you have been doing over the past *N* months, we calculated that your work is worth *X* to us, therefore we can go up to that point, but not over to meet the *Y* for which you asked".
- Analyze the costs of security solutions and their actual added benefit as-is - i.e. with the current skills and tasks assignments of the SOC analysts - and compare that with what could be achieved towards the SOC priorities if the budget for the overpriced/underused solutions were redirected on other (internal) investments. Of course, the Return On Investment (ROI) should be taken into account for this comparison, both for the security solution and the other (internal) project.

  The goal here is to stop paying for things that are barely used or with little added value, if that budget item could be better allocated to some other project that would enhance the SOC production, either by increasing its quantity or quality or by reducing wasted expenses.
- Compare the cost of having teams (mainly those in charge of response and MRO) 24/7 available in shifts versus that of having some members of these teams doing on-call duty. There are a lot of variables to take into account here and not just flat costs: the employees may be more productive,

happier and on a better mindset if they are working, even in rotations, than if they get called regularly at night during their on-call duty and then have to work their regular shift the next day.

A department that is available 24/7 would also be able to answer in case a customer has made an on-call duty subscription - and even faster than with regular on-call duty - and they also could work on any alert or incident that happened outside of regular working hours, therefore lightening the workload for the regular day shift. Of course in that case, if the customer hasn't subscribed to a 24/7 monitoring, the notifications would be (automatically) postponed to the start of the day shift.

- Build a table that lists every task of every job in the SOC, the number of units to produce every week (or day or month), the average time it takes to complete, the total time needed, the cost per hour and total cost all depending on experience (junior, full or senior) as per the example below:

Task	Number of units per week	Average time spent			Total time needed			Cost per hour			Total cost		
		senior	full	junior	senior	full	junior	s	f	j	s	f	j
Triage 1 alert	100	5min	10min	20min	8.33h	16.67h	33.33h	40€	20€	10€	…	…	…

This is the baseline on which to build simulations of staff needed by level of experience. Some tasks will be limited to full and senior or senior only and some others would rather have junior or full handle them as the cost would be lower. Therefore, it will help simulate how many people with what level of experience would be the best for a particular team/job in the SOC to optimize cost efficiency (adding other columns like FTE, revenue generated by task, etc would come in handy for that matter).

However make no mistake, these are actual people behind the numbers and those people have preferences in jobs and tasks, wishes in terms of career evolution, life goals, etc.Therefore, the ratios (FTEs per experience level) simulated will always be unattainable goals and should be treated as such: a direction rather than an objective. This circles back to a previous point comparing the cost of losing a person and recruiting a new one versus that of meeting the demands.

All of these tables and graphs take a lot of work to build, especially if the data doesn't exist at first and has to be created. Also that work may be tedious, but it is well worth it.

Firstly, it paints an accurate picture of the state of the SOC and gives a lot of vision, if not control, to the SOC manager about the health of his or her SOC.

Secondly, it translates technical and operational feelings of what works well and what does not work in the SOC into actual, factual numbers and costs that are very well understood as-is by executives. Instead of expecting executives to spare the time to learn and understand the technical and operational aspects of the SOC, it is up to the SOC manager to speak their language in order to be heard and understood. This is one way to do just that.

Since a SOC has limited options to generate more revenues, most of the effort regarding financial optimization should be on reducing expenses. Therefore, the first step is to understand where the highest expenditures for the SOC or MSSP are, and this is precisely where all the suggested tables and graphs come in.

Usually, the two top items are the salaries of the personnel and the tools (licenses) used by the SOC.

The licenses for the tools could be negotiated with the vendors, and a regular review of tools' usage should take place to ascertain that they are the correct tools to answer the needs, and that there isn't another (cheaper) tool that could be more suited. There can be some gain there, but ultimately, people need proper tools to do a proper job.

On the other hand, from a purely economic point of view, it would be great to have the experience and know-how of senior staff at the price of a junior one. Obviously, this isn't happening, but what can happen is using the time of senior staff more wisely to enable junior staff to approach the work they do in terms of quality and quantity. In other words, senior staff can be tasked to create documentation, processes and procedures, all kinds of task automatization, and finally monitor the work done by other less experienced personnel. This would set standards and enable everyone to achieve them by using the knowledge and methods produced by experienced people. This way it requires fewer people with less experience to achieve the same quality and quantity of work. Therefore, depending on the situation - SOC or MSSP, expected growth or not - fewer people are needed for the same perimeter to monitor or the same people could monitor a larger perimeter.

3.3. Conclusion

The best way for a SOC and especially an MSSP to improve production stability, resiliency, quality and efficiency is to invest in industrialization. This mainly means identifying the key people who are able to do that job and making sure to keep them until the goal in level of industrialization is reached.

The tools are important, but having the right people on the right jobs with the right level of skills matters more: in case of attacks or other cybersecurity crises, skilled analysts with lesser tools will manage way better than poorly trained and inexperienced analysts with top of the shelf tools.

All in all, the best decisions in terms of spending are most of the time about enabling the SOC to accomplish its missions better, faster and more reliably by investing in its production capabilities rather than acquiring an overpriced underused tool or adding new members to the overflowing pool of juniors.

4. Operational aspects

This chapter goes over the main operational points to address in a SOC or MSSP. Some of them are easily overlooked and sometimes pushed back again and again until "there is time, because we are too busy at the moment".

The truth is that as long as the time available for people who could work to improve the SOC is spent on quick fixing bugs, getting quick wins or rushing urgent - or more likely already late - projects to production, there will never be any improvements made. It is even worse in the case of an MSSP because the current customers will likely continue to expand their monitored environment and new customers will come in, resulting in even less available time for improvement and automation and more bug fixing, quick wins and late projects.

Therefore, the time for improving operations has to be made available to the right people to get improvements going. This will hurt operations for a bit, because it will take some time to improve and to see the return on investment come in, but it is the only realistic way - without a budget increase that could translate into recruitments - to transition from a vicious circle to a virtuous circle.

The subchapters in this chapter are not ordered by any kind of to-do priority: they should all be addressed at one point, but the priority depends on the current state of the SOC and from what would give the maximum return on investment.

4.1. Knowledge management

Knowledge management is an actual job that requires specific skills and no amount of good intention or advice can replace that. It is strongly recommended to hire someone for this job either for the whole company or just the SOC, depending on the size and needs.

However, as this tragically falls into the "what is this job, did you make it up?" / "why do we need to pay someone just for this, can't you handle it?" category, it was important to say a few words about what can be done in this chapter, and a few more about how it can be done in the next.

As a disclaimer, please note that the pointers about knowledge management given in this document do not come from a professional knowledge manager and should not be regarded as best practices but rather as a small buoy on which you can cling desperately to try and not drown into the infinite ocean of knowledge.

That was a bit dramatic indeed but sadly not that much of an overstatement. As previously mentioned, a SOC is a complex entity and the mass of purely job related knowledge needed is too much on its own, but to that is added the contextual knowledge of the environment to monitor, multiplied by the number of customers when it comes to an MSSP.

There is simply, humanly no way that any one person can know everything he or she needs to work in a SOC by heart, without looking up or checking documentation multiple times per hour. If someone pretends otherwise, they're either lying outright to try and brag or a complete idiot that actually does not understand a thing they are doing. In both cases, it would be advisable to avoid having such people in a SOC or MSSP.

There are two main points to address when it comes to knowledge management and operations: creating and maintaining the knowledge and having and keeping the knowledge base actionable.

4.1.1. Knowledge creation

Knowledge, may it be documentation, processes, procedures, reports, etc. is key in many fields so that things can evolve and move forward instead of repeating themselves. That is true for a SOC and even more so for an MSSP.

Knowledge is perhaps the most worthy investment for the future, but writing documentation, for example, can be very tedious and therefore those who actually enjoy working on it are few, while most do it because it's just "that part of the job that you have to do" and some try to avoid it at all cost. However, it is imperative that everything is fully and accurately documented, because everyone should be expendable in the way that there is no Single Point Of Failure (SPOF). If anyone were to suddenly disappear, there should be some time to recover, but the recovery must be full without any loss of knowledge.

Therefore, since this is a crucial task and no one really wants to do it, everything must be done to encourage, enable and facilitate knowledge creation. For example, some of the following could be done:

- Explain and then demand that processes and procedures be followed and enforce it.
- If there is an issue with a process or procedure, quickly react to correct it and/or propose that everyone contributes - the person pointing out that there is an issue proposes a correction.
- In every task and mission, plan and allocate time for knowledge creation and insist that the task is not complete until knowledge has been created.
- Educate everyone to always refer to the knowledge base for everything.

There are many other creative ways to make everyone participate in knowledge creation but the best advice may be to try and show everyone that the knowledge they're creating has value. There is always value in knowledge but it shows only when it is used and showing that knowledge creation is meaningful and has value is oftentimes enough to convince people to participate.

Unfortunately, creating knowledge isn't the end of it, because it's alive. Granted there are some things that are immutable but the vast majority needs to be updated to stay relevant. Therefore, each piece of knowledge should be in one of the following states:

- Up to date and used
- Being created or updated
- Outdated and being archived (optionally before a later deletion)

This is an ever ongoing effort that must persist and be encouraged to prevent any loss of information over the long run. The cost of losing knowledge will always be, at best, the time and resources used to create it in the first place and at worst a net loss plus whatever it costs to find another way to achieve the same goal.

4.1.2. Knowledge organization

For the knowledge to have a real value, it must be used and in order to be used, it has to be organized, accessible and easily searchable.

This is the hard part when there is no one tasked with knowledge management, because there can either be multiple authorities or none at all when it comes to organizing. If no one has been designated by upper management, then at least someone should be designated within the SOC as a decision maker when it comes to knowledge and it should be made clear to everyone inside (and outside) the SOC that this person and only this person can make decisions about organizing knowledge within the SOC. This would ideally be someone that understands the needs of every team and has some experience - as a witness if nothing else - with successes and failures in knowledge management.

At this point it would be best to limit as much as possible the number of solutions hosting knowledge because it would most likely be far easier to search one source of knowledge than many sources. If there has to be more than one source of knowledge then it should be very explicit and clear to everyone what knowledge can be found where - preferably have each source host specific types or perimeters so that there is no overlap and all the knowledge on a single subject is in a single place.

In any case, the following should be observed:

- There are regular communications about what knowledge can be found where - or one document explaining that and regular reminders about its existence - so that newcomers get the correct information right away and other staff members are kept updated.
- Everyone must be able to access every solution hosting knowledge described in the previous point.
- There must exist an explanation on what knowledge to expect and more importantly on how to use every solution hosting knowledge. That explanation should of course be the landing page when accessing the solution.
- Processes for creation, modification and deletion of knowledge must exist and be enforced. These should include a validation step from another entity (either a person or a bot).
- Other quality assurance items, such as taxonomy, nomenclature, templates, etc. must also exist and be enforced.

Knowledge organization is indeed hard because once everyone understands the value of knowledge and keeping it up to date and starts creating and modifying things all around, it can very quickly become a huge mess.

Since it's not possible for everyone to know by heart how the knowledge is sorted, there must be rules, validations and templates to guide them through the process and make it easier. Keep in mind that creating knowledge is not a sexy task in the first place and now that people start doing it seriously, they should not be slowed or spooked… too much.

That balance is the hardest thing to grasp: how little is not enough and how much is too much? Not enough management and there will thrive duplicates, outdated knowledge or even misinformation; too much and people will give up, not wanting to have to jump through hoops to do a task they didn't want to do in the first place.

4.2. Infrastructure lifecycle

In a SOC and especially an MSSP, there are three main lifecycles that should be defined, enforced and monitored to ensure the continuity of operations.

The first one is about the infrastructure used by the SOC, because it is the foundation on which detection and response are built and if it fails, there is not much the SOC can do.

Every SOC and MSSP needs to define its own lifecycle because it will depend on a number of things such as choices made, services available, SOC jobs definition, SOC teams composition… The lifecycle may or may not vary depending on the hardware and software but if it does, every possibility needs to be defined, enforced and monitored.

In order to help with this, there are a few pointers to come, for the two main phases of said lifecycle. As stated before, these pointers will not be tied to a specific editor or even tool and could be applied for a SIEM, a SOAR…

4.2.1. Infrastructure architecture and deployment

The first phase is about the architecture and deployment of the infrastructure: its "birth".

The actual architecture design will of course vary with the goals, the tool and the editor, but there are some things that might be regarded as obvious or even dumb that should be double-checked to avoid surprises and complications down the line:

- If possible, try to use the same specs of hardware/OS across tools and customers as it will limit surprises and make MRO easier - this could then be templatized to automate deployment.
- Carefully design and test different "standard architectures" so that when a new customer comes in, there is no or only little time spent on the design step - only for some specific needs and customizations. This allows for a simpler deployment as it would mostly be copy/pasting something that already works elsewhere.
- Always test - if possible with production load and data - before going into production, especially if there is a need for collocation of multiple tools or roles on the same server.
- Make sure secrets are unique, strong enough and… stay secret - especially for anything with some admin privileges.
- Use secure network protocols everywhere, meaning both for the tool's internal and external traffic.

The last two points sure do seem obviously dumb in a document about SOC and MSSP, but they very sadly need to be mentioned.

Keep in mind that without its tools, a SOC is just a bunch of people taking guesses and rolling dice to figure out what may have happened or is still happening - and more and more attackers are figuring this out as well.

As much as possible, use best practices for that tool/editor both for the back end and front end configuration and when you don't, make sure that there is an actual reason and document it. This will always make maintenance and future evolution easier as it makes the configuration standard and recommended by the editor.

The best practices also often help a lot in resource optimization, which is easily overlooked when building up a new infrastructure because there is no need for it. However, resource optimization becomes key for a long term exploitation and it is much easier (and cheaper) if it's part of the design and documented than if some or all the work has to be done again later on when freeing up resources is a must have.

For this to be possible, the people working on the architecture and deployment must be sufficiently trained to know best practices in the first place and when not to apply them might yield better results. Moreover, people that are already trained tend to be less eager to "use that project to test stuff" because they already know the results or know how to build a lab environment when they are unsure, thanks to said training. There is simply no way to overstate how bad it is when the production environment ends up working somewhat but "it was built years ago by that guy who liked to fiddle, but isn't there anymore".

4.2.2. Maintenance, Repair and Operations

The second phase is about the main part of the life of the infrastructure, after it first comes online for production purposes until it is decommissioned.

The single most important point here is that the only way to guarantee the availability of all the tools used by the SOC is to have people dedicated to MRO duties. "Dedicated" means that these people should **always** have

at least priority for MRO and especially fixing whatever is down as without these tools, the SOC analysts find themselves unable to do their job.

The organizational particulars as in which team the people dedicated to MRO belong do not matter much in fine as long as these MRO people are dedicated to that job and are accountable to the SOC with some sort of SLAs. Of course, this does not have to be that formal, especially if they are part of a SOC team but there absolutely cannot, ever, be a time when something is down and it's either no one's job in particular to fix or it's someone's job but that person cannot do it because he or she has other priorities.

Of course MRO tasks do not add up only to fixing downed servers or services and the most time consuming part is often keeping everything up to date. In order to achieve this in a graceful manner, there are a few attention points:

- Having a precise inventory of what to maintain (hardware and software) and where (if multiple environments in case of an MSSP for example).
- Robust processes and procedures to ensure that there is very little room to fail even if the person assigned is missing part of the context or is light on some skill.
- A well established planning that is shared with other SOC teams (and the customers) to limit production issues and avoid panic. This point implies that scenarios and decisions must be established on how to go about upgrading and updating (i.e. by tool? by editor? by customer? …) and continued communication with every stakeholder exists. Here it is especially important to keep in mind that the tools are what make the SOC operate and although upgrades and updates have to be done, they should be done in the least impactful manner possible for the SOC.
- Skills and knowledge of the person assigned to the task versus what needs to be done. It may or may not be preferable to wait for the right person if it means postponing the job, but this kind of decision is better made in advance as a plan B scenario.

Planned MRO activities as described above are implied to be manual as it is often the case. However, it can be very hard to do properly in larger environments or multiple environments (e.g. MSSPs), especially if the MRO team is undersized.

That's where a DevOps approach should be considered to automate as much as possible and save a lot of time. DevOps always requires an investment (resources and time) but is simply a must have for MSSPs as manual upgrades and updates are too time consuming and prone to errors, which require troubleshooting, which consumes some more time… This is made easier if the infrastructures deployed in the first place are standardized or even templatized and follow best practices.

Again with DevOps, it does not matter to which team these people belong but they absolutely need a good understanding and close working relations with all of the SOC teams to correctly understand their needs and implement them.

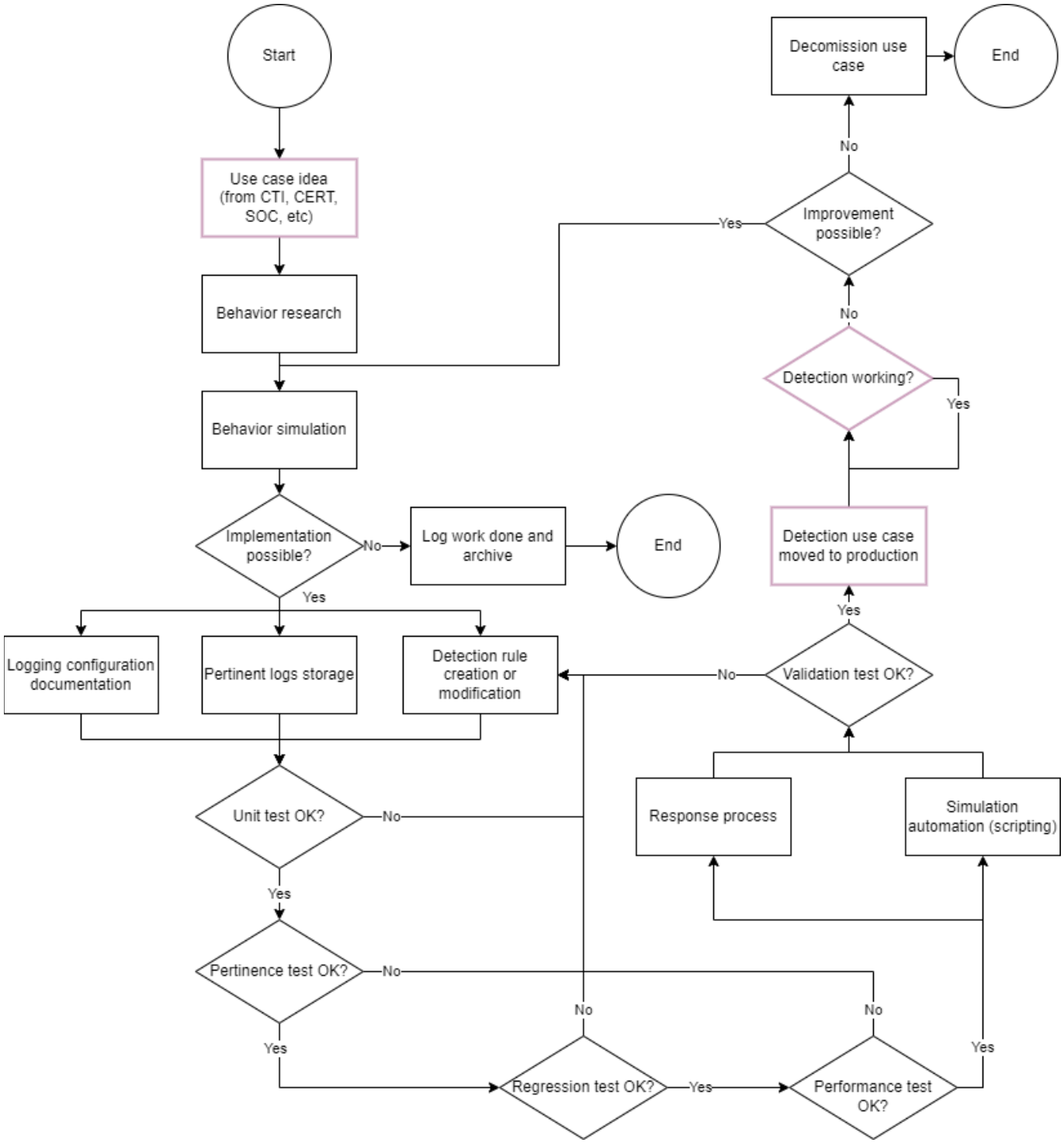
4.3. Detection lifecycle

The second main lifecycle is the detection lifecycle and it is probably the most important since the goal of a SOC is to detect anomalies.

This lifecycle defines the steps through which to go to build and improve detection use cases and the teams involved at each step. At any time, the state of any detection use case should be clear to everyone in the SOC or MSSP and who is responsible for the current step.

4.3.1. Detection engineering workflow

The engineering workflow will of course vary from one SOC or MSSP to another because it depends on a variety of factors, but the steps should more or less always follow the same pattern as shown in the diagram below.



The light purple steps represented in the detection engineering workflow diagram are not actual detection engineering steps because they deal with the initial idea or interacting with a production SIEM for example.

Every other step can and it is **strongly** recommended that they should be performed by the same team/people: the "detection specialists" mentioned earlier who could work together in a "purple team".

The details of the contents of a detection use case will be discussed later on but the workflow for its engineering should always start with an idea followed by research and simulation and end with a permanent monitoring of whether it works.

If and when it doesn't anymore, either improve it if possible or retire it, because it is worse to have a malfunctioning detection rule in production than none at all: with the latter it is clear to everyone that there is a miscoverage.

4.3.2. Workflow tests

There are 6 tests represented in the detection engineering workflow, all of which are mandatory to ensure that the detection use case works as intended.

It is strongly recommended that these tests be automated and the results only reviewed by a human in case of failure. More on the tools, including automation, in the next chapter.

Unit test

The unit tests imply that the detection engineering is done using a DevOps platform to help manage versions, automate testing, collaborate, etc. - this will be discussed in more detail later on.

Using a DevOps platform, unit tests are tests that are run with each commit - a commit is any file modification pushed to the server. These tests aim to check all modifications to control configuration integrity such as common mistakes and incoherence between files, ensure minimum requirements are met and ultimately prevent the deployment of any misconfiguration.

Pertinence test

A pertinence test checks that the detection rule is working as intended, i.e. it triggers when expected and only then and the alert it creates contains useful data.

This can be achieved by (re)playing a scenario, through scripts in a lab environment to have the devices create logs and submit them to the SIEM, through generated events made with an event generator or by reusing the logs stored from the initial behavior simulation.

Then, the detection rule should correctly detect every anomaly it was meant to detect and ignore the rest of the events. At this point, it is recommended to test the detection rule against more logs than only those which contain the anomaly to detect to ensure that there are no False Positives.

Regression test

Regression test aims to confirm that the new version of the detection rule does not introduce any breaking change. A breaking change is a major modification which breaks compatibility with older versions. In the case of a detection rule, this translates into the newer version of the rule not detecting anomalies that it previously detected, effectively breaking (part of) the detection.

Sometimes a breaking change is wanted and expected after a complete overhaul of a malfunctioning detection rule for example. In that case the regression test would fail but the failure would be converted into a pass once the reason has been reviewed.

Since reducing the number of False Negatives is a higher priority for the SOC than reducing the number of False Positives, the regression test will guarantee that the newer version of detection rule detects as much as older versions. There are two checks to perform to conduct a regression test:

- Run the detection rule using at least the same set of events or logs from the previous versions of the detection rule - meaning that there can be more logs if the rule now detects other anomalies, but the same set as before must be present. The same number of alerts must be triggered over the same logs set, proving that the newer version detects at least as much as the previous one.
- The alerts triggered by the newer version must contain at least the same data in nature and labeling as older versions.

If either check fails then the regression test fails and it must be reviewed to understand the reason. If it was expected, then the failure is converted into a pass, otherwise there is some fixing to do.

Performance test

This test's goal is to determine the resource usage of the detection rule as the SIEM resources are limited and the impact of deploying the rule must be known in advance to determine whether it is worth it in a resource constrained environment.

The relevant indicators to measure vary depending on the SIEM editor but the impact of the rule on resources must be known and a threshold should exist for this test which, if breached, fails the test. In that case, a review must be done to either optimize the resources used by the detection rule or document why the high resource consumption of the rule is acceptable.

Validation test

The validation test is the last step before declaring a detection use case ready for production. This test consists in simulating in the environment the behavior the detection rule should detect and checking in the SIEM that an alert has actually been fired by that detection rule proving that the logging, the collection, the parsing and the detection rule are all correctly configured.

This final test determines whether the detection rule delivered lives up to expectations.

Working detection test

The so-called working detection test pays in quality assurance what it lacks in originality. It is basically the same thing as the validation test but repeated (randomly) over time to ensure the system keeps working as expected.

This can be achieved by automating the previous validation test, which would be the better way, but it can also be done by injecting logs into the SIEM and checking that an alert is fired by the detection rule. The former is better because it also tests logging and collection but is not always possible; the latter "only" tests parsing and detection, but is easier to build.

Of course, if any working detection test fails, the matter must be addressed at once, because it means that that detection rule is out of order.

4.3.3. Purple team

Since the goal of a SOC is to detect anomalies in the environment in order to respond to them afterwards, detection is the key for a SOC or MSSP to be relevant.

There have been and are more and more instances where purple teams are assembled to simulate attacks and improve detection (and prevention), which is the only way to be sure that the detection rules in place are not theoretical, but actually work. These purple teams are usually temporary, lasting the predefined amount of time assigned to the mission. Moreover, they are made by plucking some people with offensive skills to simulate attacks, some others with detection know how to create and improve detection rules and maybe a few more with incident response experience to make sure that the response process for the use case is correct.

The purple teams described above are simply wrong on so many levels:

- Detection is the most important part of a SOC or MSSP, the only way to ensure that the detection rules are not theoretical is to simulate the behavior and yet the purple team is temporary. The team in charge of quality assurance for the most important aspect of the MSSP is not a permanent team.
- The people participating in the temporary team are yanked from their jobs with little to no training on the skills and context they miss (i.e. offensive, detection, incident response and even how a SOC works) and have to learn to communicate to the others and to produce something that is usable by the others as they go. The members of the team working on the most important part of the MSSP have issues understanding each other.

  This is sadly not an understatement: the SOC is a complex environment with a very specific workflow and that workflow is actually so different from other teams (i.e. pentest teams or CERT) that it is really hard to communicate properly without previous experience of the other jobs. To sum up this particular, crucial issue, one could say that pentest teams and CERT mostly work with a micro view of operational security - i.e. they work directly with systems, from system to system on a perimeter delimited by the engagement rules for the former and the attack for the latter - whereas a SOC works with a macro view, having to cover the whole environment. Because of this, SOC and MSSPs had to adapt to scale up, having to find workarounds to perform detection and investigation at scale that would be straightforward on a single system.
- It gets worse as this temporary team composed of people having a hard time communicating to each other is not homogenous: each person has one particular set of skills and cannot be replaced or helped in their tasks. Depending on the size of the team, each role or each member of the most important team is in itself a Single Point Of Failure (SPOF).
- The worst part is that after all is said and done and the mission finishes, if someone in the SOC realizes that a mistake has been made on a detection rule improvement and the documentation, timeline, report or whatever is incomplete on that matter, there is no way for the SOC to fix it properly on its own. The only thing left is to hope that the person from whichever other team remembers or wrote down something that didn't go into the report - let's not even mention that there could be a do-over of the purple team for that issue because this is not happening, ever. Once the most important job for the SOC is done, the team splits and the SOC can only pray that everything is alright.
- Finally, these purple teams often lack proper hierarchical attachment to any department. The direct consequence of this is that the product of the team, the detection use cases, may or may not be used in the production environment by the SOC or MSSP in the end and they surely escape any kind of controls or detection engineering workflow. The most important deliverables for the SOC may stay unused or be discarded entirely.

All of this could be comical if it were made up, alas, it's not.

Knowing where classic purple teams fail, it is easy to imagine one that could be beneficial for the MSSP and lives up to its name. **A purple team worth its money should**:

- Be permanent.
- Have its members be willing to work with both a blueteam and a redteam approach and/or have some kind of experience on both sides.
- Be made up of SOC/CERT skills and pentest/redteam skills.
- Be part of the SOC or MSSP as everything it produces is for the SOC.
- Work for the SOC or MSSP to improve the overall detection capabilities.
- Have **each of its members have the skills' "mix" blue/red to maximize efficiency and synergy**.

The purple team must have access to a controlled environment, such as a lab, that it can start up at any time, in order to properly perform its R&D without limitations or risks for production. Ideally, to validate the findings, the scenarios should be played in the company's or MSSP's own pre-production or production environment with the same rules of engagement, and precautions, as a pentest or red team exercise.

The purple team can and should be given all missions pertaining to detection and some new missions to get the most benefits possible from the offensive skills:

- Detection rules conception - from ideas to research and behavior simulation.
- Detection rules implementation – from a bunch of logs and artifacts gathered from the simulation to an actual detection rule.
- Detection rules testing – pertinence, regression, performance and validation tests either manually with offensive tooling or preferably with some automation.
- Detection rules tuning, updates and upgrades – with regard to the evolution of detection and offensive capabilities.

- Logging configuration best practices can be added here, since the purple team had to go through that to make the best detection rule possible.
- Subjects related to the SOC data model (see next chapter) and even logs parsing.
- Live detection and response testing – simulate attacks to test the SOC's response capability and reliability and give actionable feedback to response specialists.

However, in order to stay relevant, the purple team needs to know precisely how the SOC response specialists (the job described in a previous table) team and the redteam or pentest team work:

- It stays updated on offensive TTPs by interacting with the redteam as this guarantees the pertinence of the simulated offensive behavior and therefore the detection use cases.
- It takes into account all feedback from the SOC response specialists to maximize detection efficiency and optimize response procedures with every new or improved detection use case.

Another, complementary way to achieve this last point is to have core, permanent positions in the purple team, and some temporary positions. These temporary positions would be filled with people from other teams like CERT, redteam, SOC response specialists, etc, who would join the purple team for a few months. That way, the core members of the purple team would have their knowledge refreshed, and the team as a whole would stay up to date, with an interesting, beneficial mix of ideas and perspectives.

4.4. Incident lifecycle

The third main lifecycle that absolutely needs to be defined in a SOC or MSSP is the incident lifecycle. It is key especially in an MSSP to have any chance at having a stable homogeneous response across customers and time.

The incident lifecycle defines the response phases used by the SOC, the statuses an incident can have, the possible qualifications for an incident along with the conditions to meet to get there and the stakeholders involved at each step.

4.4.1. Detection result

There is a crucial point to discuss about incident qualification: **the qualification always refers to the result of the detection**.

Yes, this was purposefully put in bold and underlined because it is the key to understanding incident qualification by a SOC and it is very, **very** commonly misinterpreted.

The reason for misinterpretation is that when presented with an incident once the investigation is done or even later once the remediation is done, it is only logical to want to qualify it with regard to the consequences: there was a malicious activity with some impact so it must be a True Positive or on the contrary the actions were those of an administrator so it is of course a False Positive. It could not be more wrong in both cases.

Represented below is a confusion matrix applied to a SOC. The columns represent the results of detection, i.e. if an alert is fired then it is a "Positive" and if not, it is a "Negative". The rows represent the expected result of detection, i.e. if there is an anomalous event or events it is "Positive" or "Negative" otherwise.

		Actual result of detection	
		Positive	Negative
Expected result of detection	Positive	True Positive (TP) "hit"	False Negative (FN) "miss"
	Negative	False Positive (FP) "false alarm"	True Negative (TN) "correct rejection"

There are 4 cells in the table, one for each of the possible results:

- A True Positive (TP) is the qualification given to detection if an alert was created when there was indeed an anomaly to detect. This is what SOCs and MSSPs want to maximize because there is a security issue awaiting response.

- A False Negative (FN) is the qualification given to detection that did not fire any alert even though there was an anomaly to detect. This is what SOCs and MSSPs fear the most and want to avoid at (almost) all cost, as it means there is a security issue awaiting response going undetected.
- A False Positive (FP) is the qualification given to detection that fired an alert although there was no anomaly in the environment. This is not a real issue in itself but it generates a waste of time for the analyst performing the therefore needless response.

  However, if there are numerous FPs either in absolute or relative numbers to the TPs, serious issues can arise.

  High absolute numbers of FPs means that there are a lot of resources wasted, translating in a direct cost for the SOC or MSSP - that is if there are enough analysts staffed, otherwise they are also overworked.

  High relative numbers of FPs can be even worse because it cannot be solved by throwing money at the issue: it creates alert fatigue for the analysts which translates into increased frustration and lowered morale for them, as they spend lots of time chasing ghosts. The worst part of alert fatigue is that it greatly increases the chance of even a senior, experienced and skilled analyst missing an actual anomaly during response, because he or she simply is used to not ever finding any anomaly.
- A True Negative (TN) is the qualification given to detection that did not produce an alert when there was no anomaly. This would be the most common qualification if it were used.

In practice, a SOC or MSSP can hardly ever qualify detection as TN or FN. Although FNs would become widespread incidents or even a crisis that ends up in the news, the FN qualification can only be given after the incident response when realizing that some detection rule should have detected such or such an event and didn't. At that time, if the SOC still exists, nobody actually bothers to create an incident in the SIRP and qualify it as an FN, if that qualification even exists at all.

The SOC detects anomalies which can take the form of an alert from a SIEM and the triage step of the response phase is an investigation that determines the result of detection: if the detection worked as expected, then the alert is a TP and can be transformed into an incident, otherwise the detection did not work as expected and so the alert is an FP. This is precisely why **the qualification always refers to the result of the detection**.

To circle back to the previous examples of misinterpretation:

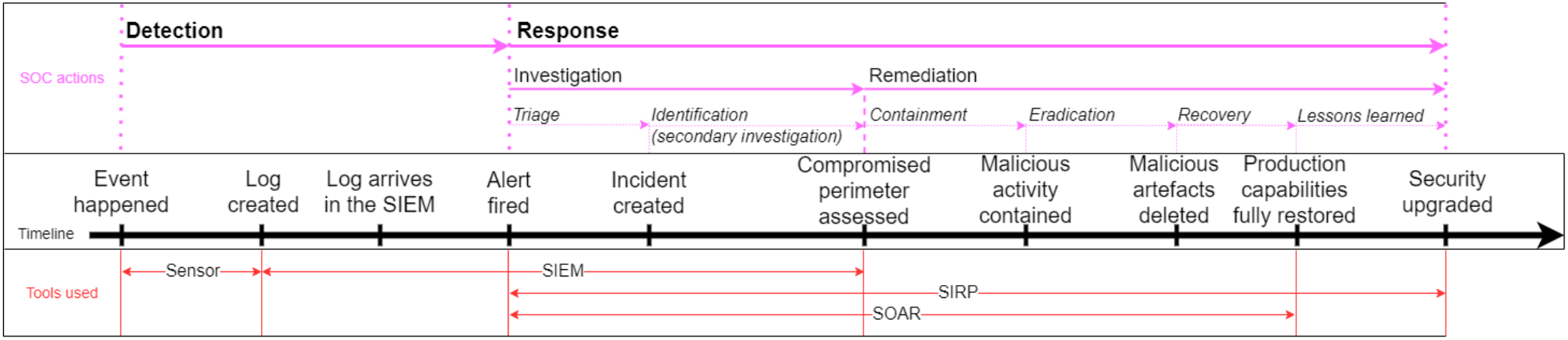
- "There was a malicious activity with some impact so it must be a True Positive", without knowledge of whether a detection picked up the anomaly in the first place, this scenario could as easily be an FN. Of course someone or something ended up alerting the SOC or CERT to respond but a call from an employee saying "Hello, all the computers on the floor went dark and none will boot up anymore" is not a detection but a mere acknowledgement of catastrophic failure.
- "The actions were those of an administrator so it is of course a False Positive" is a statement that is wrong in all but a few very rare and specific cases. Either the detection did not fire any alert for this anomaly so it cannot be a "Positive", period, or the detection did fire an alert because there was an anomaly so it worked and is therefore a TP. The fact that after investigation it comes to light that it was a legitimate action from an administrator does not change anything to the result of detection.

When explained, it becomes more logical to view qualifications this way: they are very often used to measure SOC or MSSP detection efficiency as they should. Now imagine that the qualifications were referring to the result of the investigation and the SOC were monitoring an environment with very few or no malicious activity, it could only ever have "bad grades" regarding qualifications.

Since the environment monitored should not impact the detection efficiency of the SOC, the qualifications must always refer to the result of detection for these metrics to be effective.

4.4.2. Incident evolution

The timeline below represents an example of the phases that could be defined and applied for a SOC or MSSP. The most important point is that it exists and is used as reference whenever needed.



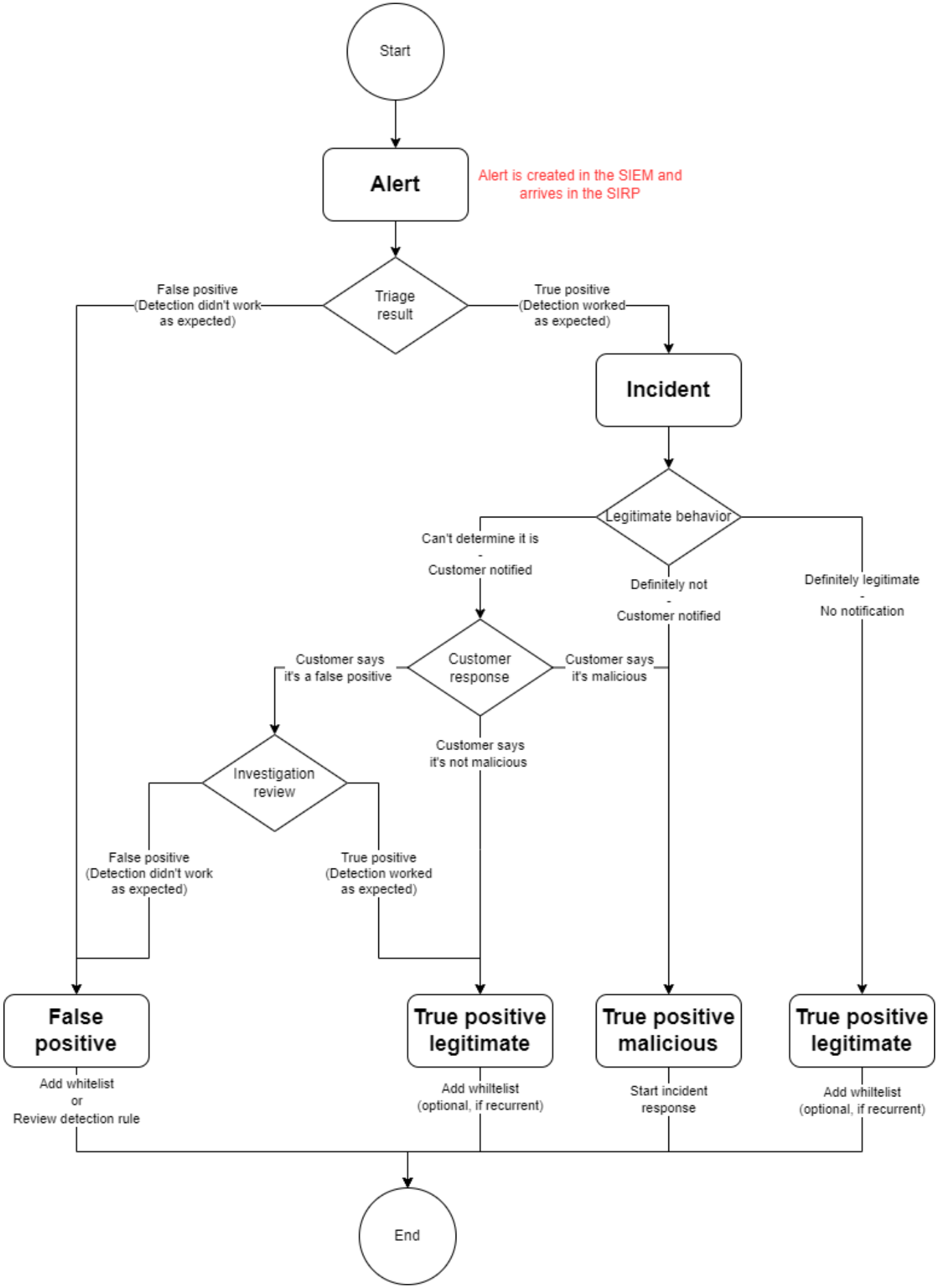
In this example, the timeline in the middle marks when the key events occurred while the top part shows the "SOC actions" for a lack of better expression, which are the phases mentioned earlier, and the bottom part is a reminder of when some generic tools used by SOCs and MSSPs are relevant.

A few remarks on the timeline:

- The first (and main) point addressed by a SOC is detection, which lasts from the time an event or anomaly happens until some tool (here a SIEM) fires an alert for it.
- The response phase represented is a modified version of the SANS Incident Response Cycle that divide the response between two stages:
  - Investigation, comprised of a Triage step in which the analyst performs a primary investigation to determine whether the detection worked and is therefore a True Positive, in which case the alert becomes an incident, or didn't work and is a False Positive, and an Identification (as per the SANS Incident Response Cycle terminology) or secondary investigation step to assess the compromised perimeter. This stage could be fully achieved without any active action in the environment if the investigation is performed with passively collected artifacts.
  - Remediation, regrouping the four steps left in the SANS Incident Response Cycle. At this stage, active actions are performed in the environment to remedy the incident.
- The sensor's main use from a SOC point of view is to generate logs for the events it observes.
- The SIEM is shown to be relevant from "Log created" because it is implied that the SIEM's infrastructure is also used to collect and centralize the logs, but depending on the technological choices made, there could be other tool(s) involved.

4.4.3. Ticket status

In this subchapter, "ticket" is used to refer to "alert" or "incident" independently from the qualification.



The previous diagram is an example of a ticket lifecycle with the different statuses it can have. It is specifically intended for an MSSP as it mentions an interaction with the customer.

The detection is either qualified as a False Positive or a True Positive:

- If it is an FP then something must be improved by adding a WhiteList (WL) entry or reviewing the detection rule in order to not get another unwanted detection such as this one.
- If it is a TP then two things can happen:
    - After investigation, there was actually nothing malicious about the activity. It was a legitimate action and so the status could be updated to "True Positive Legitimate" (TPL) which basically means that this time there was nothing wrong and there may or may not be a way to determine it at detection time. If there is, a WL entry could be added but this should be done with caution as it can be hard to know for sure, without behavior analysis, that it is actually legitimate.
    - After investigation, malicious activity has been confirmed. The status could then be updated to "True Positive Malicious" (TPM) before moving on to incident response, starting with the Identification or secondary investigation step. This update only aims to make statuses clearer at a glance because TP / TPL only could create some doubts as to "what is a TP".

The example diagram is a bit less straightforward than it could be because it takes into account misinterpretation of "TP" by the customer and analyst error.

4.5. Training and monitoring

It is important to keep SOC or MSSP teams trained as a whole and ready for the worst because when the worst comes, SOC members will not have time to think and elaborate big plans. There will be panic and confusion: this is extremely important and it cannot be overstated, there is nothing short of experience that can really prepare someone for this.

However, having trained teams in which everyone knows their jobs and what they are supposed to do when, and not if, trouble comes around is the best that can be achieved to enable fast, accurate reactions.

Also on a brighter note, there are ways to monitor the SOC environment and the user experiences associated with it in order to preemptively affect resources to stabilizing or fixing instabilities before breakdown occurs or, on the contrary, if every indicator is green, to divert resources to expanding or developing new features.

4.5.1. Incident simulation

An incident here is defined as a severe security issue that requires immediate response from the SOC or MSSP, e.g. an ongoing attack caught early in which the adversary has yet to spread too much.

Typically an incident could involve a few response specialists and their team leader, depending on the severity.

Each analyst needs to be individually trained so they are skilled enough to perform their jobs correctly but more importantly in this case they also need to know how to work together as a team, which is a whole other story.

Even with technical training, processes, procedures, diagrams, a good knowledge base and what not, without proper training specifically on coordination, a bunch of analysts put together won't amount to a proper teamwork. This is in fact much like sports teams.

Teamwork is therefore also a skill to train properly and this can be achieved by simulating incidents. The majority of simulations should be prepared and advertised in advance with a clear objective to improve teamwork in an emergency situation. However, once the team performs well on these exercises, it is advisable to throw in an unexpected simulation to check how the team reacts in what is for them a real scenario.

The "only" difference between both is psychological but this is what will actually matter when it comes down to it. If they think it is real then their reaction will be genuine and the simulation can create an experience for them, although it was but another exercise.

These simulations are of course technical but should also include any communication that would occur in a real scenario and especially any panic and/or pressure added by the customer (in case of an MSSP) or the management (for a SOC), if possible.

4.5.2. Crisis simulation

A crisis is defined here as a critical, widespread security incident already impacting production. For an MSSP, it could be either one customer victim of such an incident, multiple customers having (for the moment) a less impactful incident such as what can happen when a 0 day on a common product is exploited and there is no patch, or that the company of the MSSP itself has been compromised.

What is described above is where the real fun starts, isn't it?

This can clearly be a do or die situation for many companies, as reported too often in the news. There will be panic everywhere, the pressure on any stakeholder involved in the resolution (the SOC being one) will suddenly peak and stay at that level until the crisis ends, one way or the other.

If a SOC or MSSP goes through this without previous preparation or training, well… Let's say the good news is that there are plenty of open jobs for SOC analysts so they won't stay jobless for long.

The incident described before involved a few response specialists and their team leader when a crisis such as this will involve the whole SOC - every single member - and then more, especially in the upper management.

The incident is simulated at the analysts' level with some help of SOC or MSSP management and from a technical perspective, a crisis is actually almost only different in what's at stake. Therefore, the simulation for the technical incident part could be more or less the same, with higher stakes.

The real difference is in the panic of the management, customers or other employees and in this, especially the former, must be trained. The most important part is that upper management understands their jobs in a situation like this: how and what to communicate to whom, who will be their (only) contact on the response side that will report to them the state of things and what they can do to enable the best response possible.

At upper management levels, it is not realistic to simulate an unexpected crisis, therefore advertised ones will have to do. It is especially important to identify backups for such or such key people in the management chain in case they are not available when the time comes.

4.5.3. Site Reliability Engineering

Site Reliability Engineering (SRE) is a concept that originated at Google in 2003 and has kept evolving since. The reference and most of the best resources can be found on the website https://sre.google/.

SRE aims to maintain critical operations running whatever happens by relying on targeted prevention to minimize impacts, therefore reducing the overall direct and indirect costs.

The logic is that everything that is built is made for direct or indirect human usage and any trouble or discontent the human component has in using it will result in added cost in some way for the builder. Therefore, it is more cost-efficient to measure and monitor the level of contentment and take corrective actions when it falls down to a certain point than to wait for an SLA breach to try and fix things as there may be indirect cost of losing the user or the customer on top of the direct cost of the breach.

To achieve this, Service Level Indicators (SLIs) need to be put in place: these indicators are a quantifiable way to measure the reliability. There is real work to be done here, as most of the time, the reliability of a service is more of a qualitative feeling for its users than directly quantifiable and usable data.

Once the SLIs are defined, the Service Level Objectives (SLOs) can be set: if the SLIs fall down to the level of the SLOs, then corrective actions must be taken.

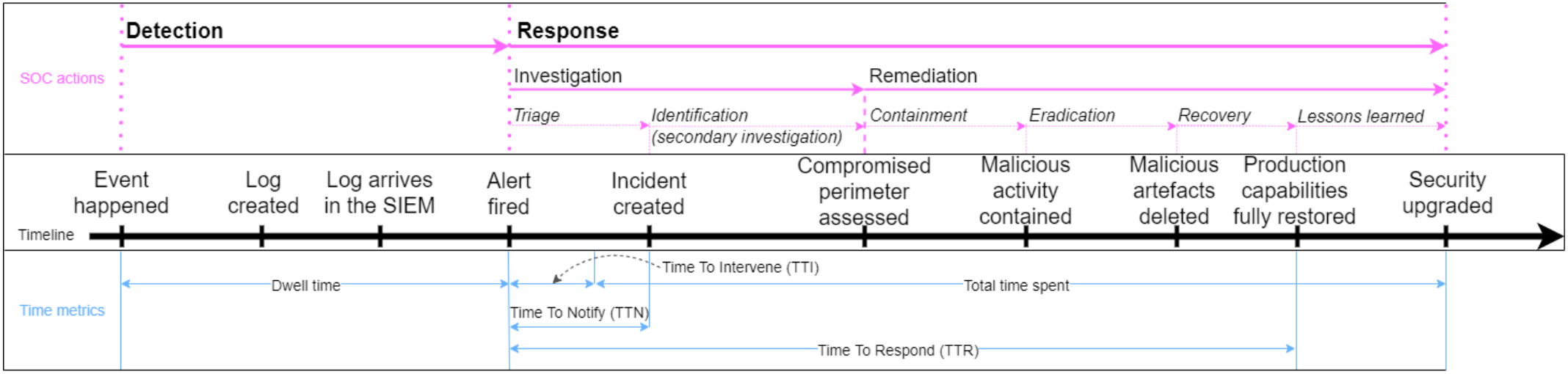
Finally, the monitoring of the SLIs with regard to the SLOs enables active and optimized resource and budget affectation: more development and new features if the SLIs are OK or more stability and fixing if the SLIs are dropping.

Applying the SRE concept to a SOC or even an MSSP can be very complex but is well worth it in the end, especially for an MSSP. That being said, the SOC or MSSP must already be very mature for an implementation to be possible and for it to make sense.

As this is very technical, more of a case by case basis and explained infinitely better on the website mentioned above than it ever could be here, it won't be discussed any further in this document.

4.6. Customer service

Customer service is an important subject for an MSSP but also for a SOC. Customer service is defined here as all relations a SOC has with external parties to which it answers. These parties, may they be customers (in the MSSP sense) or upper management of the company owning the SOC, will always fund the SOC. Therefore it is for the least advisable to work with them to understand their needs and to make them understand the SOC's constraints and issues.



The diagram above is a modified version of the one used to talk about incident evolution; this one shows the most important time metrics for a SOC or MSSP and its customers. The total time spent represents the time spent by SOC analysts on the response and is actually a SOC internal metric.

However, the dwell time, TTI, TTN and TTR are important indicators for the SOC to measure its own performance but also for its customers - some of which are usually under SLAs. These indicators should always be (graphically) explained to the customer, what they represent and how they are calculated to make for easier communication.

There should be communication templates for emails, incidents and other regular committees - each of which needs to be presented and explained in detail to the customer to make sure everyone is on the same page and that there are no misunderstandings.

All templates should be standardized so that they are as much the same as possible from one customer to another for an MSSP. They can then be automatically generated for the most part, e.g. Key Performance Indicators (KPIs), performance graphics and other task lists. The time spent by people working on these would then be on the SOC or MSSP added value, such as governance and security advice.

In any case, communication with customers about what the SOC is doing is important, especially if "nothing is happening", because a SOC is a complex environment almost opaque from the outside. Also, by nature the SOC is reactive so without active communication on the work being done, it is quite easy to conclude, when everything is fine, that there is no actual added value in a SOC or MSSP when it actually costs a lot - in other words that it is a bad investment.

4.7. Conclusion

There are many items that should be implemented, monitored and that can be improved for a SOC and even more for an MSSP. Their importance will vary depending on the current context of the SOC and its planned evolution, as will the priority of implementation.

However, they are all necessary to have and keep a pertinent, stable, efficient, reliable and robust SOC, which is what every SOC should try to be to consistently detect and/or thwart attacks.

Please do not underestimate the importance of items that are not directly related to the SOC's missions and objectives, such as knowledge management or customer service: the former, acting as the SOC's memory, participates in ensuring pertinence and the latter is the one that will keep the budget from getting lower.

5. Technical aspects

This chapter is the most technical of this document and although it tries to stay high level to be read and understood by everyone as this is the goal, there are some dives into finer points as some crucial details can influence the whole architecture.

It drills down into the tools used by a SOC and MSSP, what they aim to do, their strengths and weaknesses and gives pointers on how to use them properly. It also addresses the oh so frequent mistake of fiddling with a tool to make it do something it wasn't built to do, i.e. don't use a hammer when you need a screwdriver.

That being said, this chapter should be as editor-agnostic as possible because choosing and using the correct tool is far more important than the actual brand of the tool used, although there are of course differences between editors that need to be considered.

Consequently, the terminology used shouldn't be tied to one editor or another, but mistakes can happen.

5.1. Knowledge management

This (tiny) subchapter is mainly meant to underline the importance of knowledge management as previously discussed but also to give a few bullet points on what knowledge to create and how to create it.

Ideally, everything that needs to be remembered must be documented and that means everything:

- Processes and procedures - these are two different things and simply put, a process describes what to do and a procedure is the breakdown of the process to describe how to do it.
- Any piece of code and configuration file.
- Tools guides, usage and best practices.
- Detection use cases and incidents.
- Customer and partner information (context, points of contact…).
- Successful and failed Proof of Concept (PoC) conducted.
- For an MSSP, the services available, marketing strategy, contracts, etc.

This list is clearly not exhaustive but the keywords are "ideally, everything".

Also, the knowledge must have a target when it is created so that it is clear to any reader *why* it exists. That target is defined by a purpose and an intended audience; consequently, the level of (technical) details will vary accordingly.

For example, a document explaining the context of a customer will not contain the same information if it is intended for the SOC analyst that needs to respond to an alert on that environment or for the sales team representative who handles payment or contract renewal.

5.2. SOC/MSSP environment

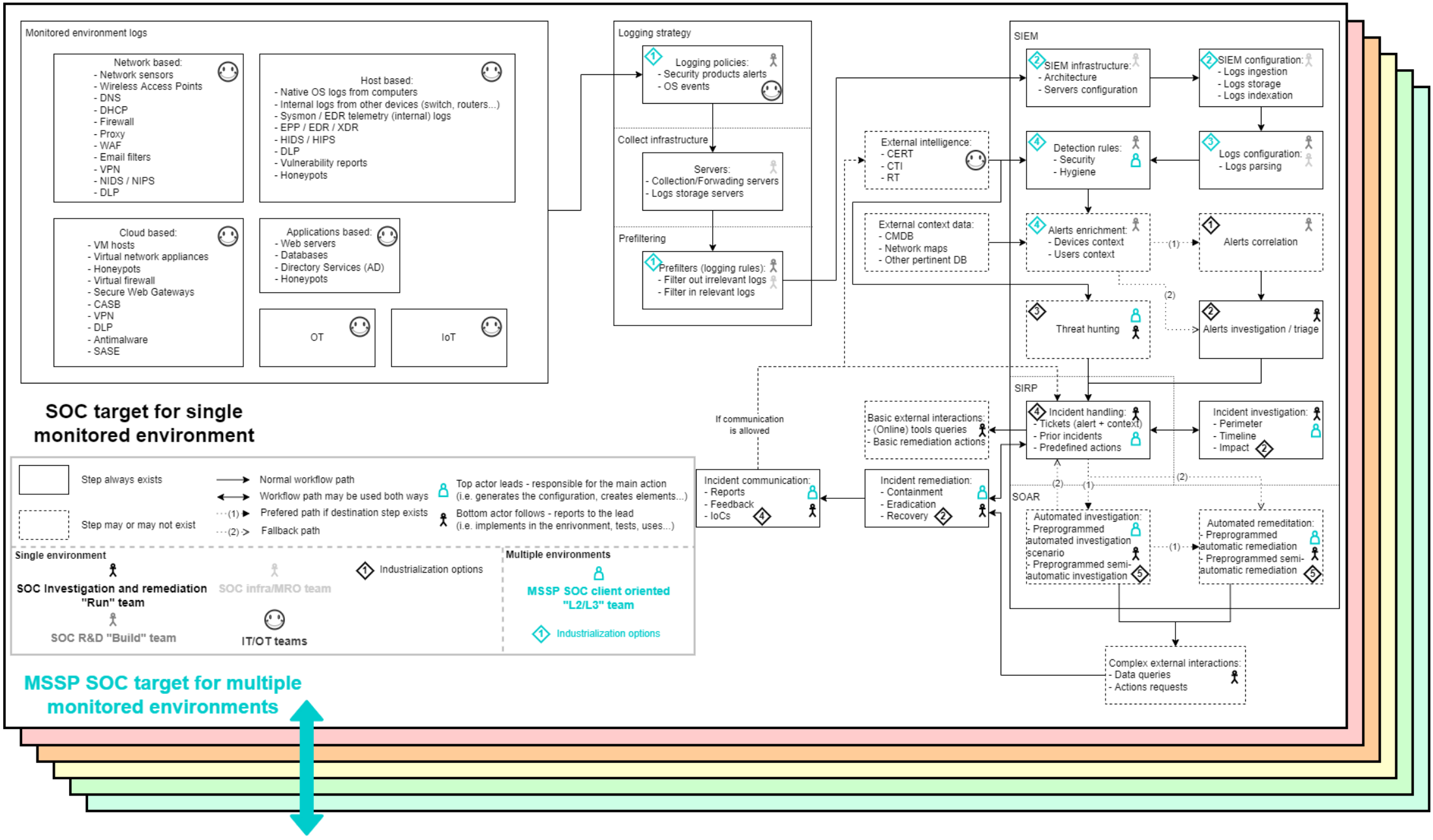
The environment in which a SOC operates is complex, as basically anything that uses electricity and integrates some sort of "intelligence" (i.e. decision making) will have security needs for prevention, detection and response.

As if this isn't vast and diverse enough, an MSSP has to deal with that complex environment multiplied by the number of customers it has and of course every environment is unique so there's no simple copy-pasting possible on the technical level.

Fortunately, some similarities exist between environments and these have to be correctly understood and built upon in order to properly automate and industrialize whatever tasks and jobs a robot can perform faster and better, i.e. with a constant and predictable output, than a human ever could.

The first place to start is with an overview of the workflow of the data - mostly logs and other relevant artifacts for a SOC - from its creation all the way to the security incident report and improvements made from detecting an anomaly in that data.

5.2.1. Overview



The diagram above shows a generic example of the different steps the data (here environment logs) go through to be processed and used.

The shades of gray represent the (single) environment within which a SOC operates and the colors add the depth of an MSSP.

The steps may be grouped together to illustrate in which tool they belong and the border between some tools may be dotted instead of full to represent the fact that, depending on the technology chosen, these tools might all be seamlessly merged into that particular technology.

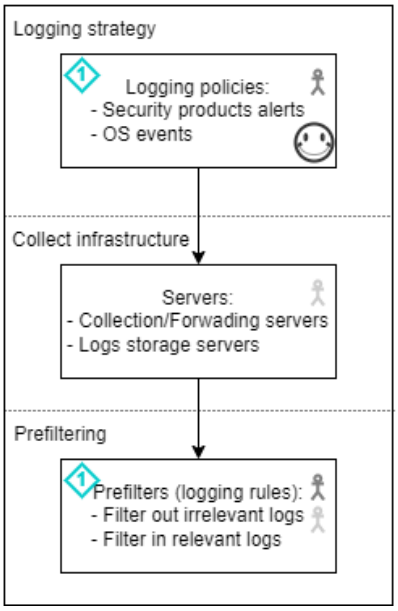
A team is responsible for each step and some steps can have two teams working on it, where one leads and the other one either helps or performs the required technical actions alone. The number and/or missions of these teams have been simplified to lighten the diagram but actual teams should represent the tasks and jobs that are needed, as discussed before.

Some steps have industrialization options which are improvements that can be made to free up humans from tasks for which a robot is more suited.

Details of the suggested industrialization options follow, where each option has been broken down in a series of steps.

Some of these steps will be discussed further in this chapter as although the idea may seem obvious and easy to understand, the technical implementation is a lot harder. The rest should be self-explanatory or a bit too technical to describe in this guide.

5.2.2. Logging industrialization

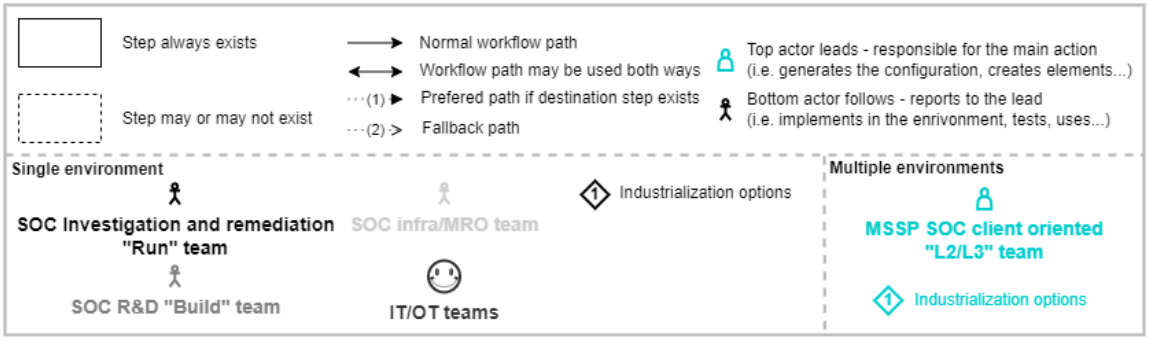


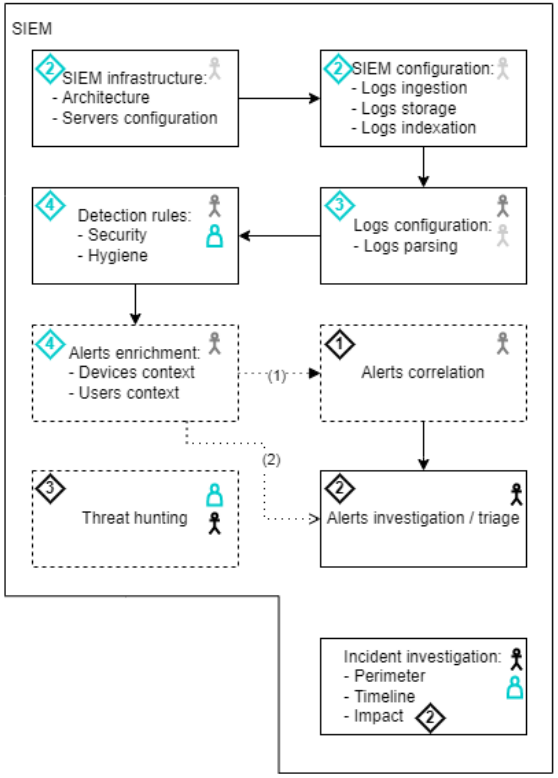
1. Centralized knowledge
2. Per product (OS/EPP/EDR...) logging policy, including volume estimations
3. Ready-to-use environment based policy adapations
4. Per product prefilter prioritized rules in case "ingest all in SIEM" is not an option
5. Automated deployment solution (API based, GPO...)

The industrialization options are numbered so that they can be easily referenced later on but these numbers are not sorted and most if not all options can be implemented simultaneously.

This industrialization option has multiple steps - it is of course recommended that these steps be implemented in that order as they are sorted by added value for detection and response and by return on time and resource investment while also taking into account prerequisites for the steps.

5.2.3. SIEM industrialization





1. Alerts correlation based on overall same-data fields/labels
2. Alerts correlation based on one-to-one same-data with respect to time and killchain incrementation

1. Investigation procedures referencing alerts to which they apply
2. Remediation procedures referenced in investigation procedures
3. Standardized procedures - i.e. templated format and information
4. Diagram-based procedures (e.g. PUML) where each action is a single atomic step (up to a point)

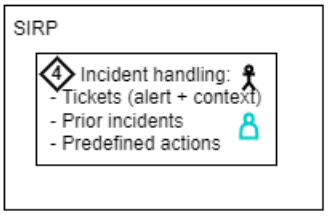
1. Use of a threat-hunting method/framework to maximize long term efficiency
2. Full-time workforce focused on Threat Hunting without doing anything else

1. Centralized SIEM knowledge
2. Per SIEM software (i.e. the SIEM itself) configuration with ready-to-use environment based adaptations
3. Per SIEM hardware (i.e. the servers hosting the SIEM) configuration with ready-to-use environment based adaptations
4. Automated deployment solution (Ansible...)

1. Datamodel suited to SOC usage - i.e. one type of data corresponds to one label and vice-versa for all data used by the SOC (detection and response)
2. Centralized sensors knowledge
3. Per data source (sensor type/sensor brand/sensor version/logs template) parsing and mapping (i.e. to the DM) configuration
4. Automated deployment solution onto the SIEMs (API based...)

1. Centralized detection rules repository with per rule additional context data needed for investigation
2. Generic (SIEM-abstract) rules implementation
3. Specific SIEM-ready rules implementation using only the SOC Datamodel defined labels
4. Automated deployment solution onto the SIEMs (API based...)
5. Automated solution for 5 (out of 6) test phases of the detection rule lifecycle
6. Inclusion of enrichment data (if available in the SIEM) in the alerts created by the detection rules
7. Full-time workforce focused on R&D by using an offensive approach to generate actual tangible logs on which detection will be based
8. Automated translation (as accurately as possible) of a generic rule into SIEM-specific rules

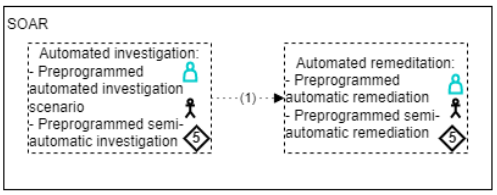
5.2.4. SIRP industrialization



4

1. Centralized knowledge base for templates
2. SIRP incident tickets detailed templates clearly stating all the investigation steps
3. Incidents communication templates (emails...)
4. Prefilled communication templates with data from the incident ticket
5. Prefilled incident tickets with data from the alerts
6. Regular tickets review (e.g. random selection) to control and enforce format and details quality
7. Incident report (major incident) template
8. Prefilled incident report with data from the incident ticket(s)

5.2.5. SOAR industrialization



5

1. Centralized knowledge base for playbooks
2. Generic (SOAR-abstract) playbooks implementation for atomic steps (automated from the diagram-based procedures)
3. Generic (SOAR-abstract) playbooks implementation for whole procedures (using atomic steps)
4. Specific SOAR-ready implementation of atomic steps
5. Specific SOAR-ready implementation of whole procedures
6. Automated deployment solution onto the SOARs

5.3. Logs and logging

Logs are the input of detection rules, without logs there is no detection, it's that simple. Also, the detection can only be as good as the data - logs - available, if these are not enough, do not contain relevant data, the data is partial or otherwise inconsistent, the detection will inevitably suffer the same fate.

To build a robust and consistent detection the nature and format of the data ingested must also be consistent and adequate. This is achieved in roughly four steps by having a consistent logging policy, using a data model suited for the SOC, checking that log parsing correctly with regard to the data model and monitoring the raw and processed data to ensure the input for detention stays consistent.

5.3.1. Logging policy

The logging policy is part of the industrialization options for logging. The goal is very straightforward but the implementation much more complex, directly depending on the variety of sensors monitored by the SOC or MSSP.

A logging policy is a per sensor configuration that enables said sensor to produce the logs - both in contents and format - expected. Also a recommended retention period should be associated with each log type but this may be a configuration for the tool storing the logs after collection and not the sensor itself.

This can be very quickly very hard to have for complex sensors and/or to maintain across sensor updates but it is the only way to ensure some consistency and have the default be that of the SOC and not that of the sensor.

Furthermore, the policy might change depending on the role of the sensor: e.g. the interests of the SOC in Windows event logs may not be the same on a workstation, on a filer and say on a Domain Controller.

Finally, the verbosity of each log and the quantity of logs of that nature should be recorded to estimate an average per day volume of data to process with such or such configuration. Most of the time, there will be a cost issue with the amount of data to process (for the SIEM) one way or the other. At that point, it is better to be prepared by knowing what the priorities for the SOC for each type of log on that sensor are and how much it would impact the infrastructure to create, process and store these.

5.3.2. Data model

The logs are now consistent and both the nature and format of data produced by all sensors is known in advance, which is a big step.

However, there are dozens and hundreds of sensors that could be monitored by the SOC and while it is tedious for a detection specialist to find the default labeling for such or such data needed for the detection rule he or she is building, it is simply impossible for a response specialist to remember by heart the labels of the data they need for their investigation.

This is an even bigger issue for MSSPs as they have multiple customers, each with a unique environment consisting of multiple sensors. Of course some sensors will be similar from one customer to another but it is nevertheless unrealistic to have analysts chase after labels at scale.

The solution to this problem, mentioned in the industrialization option number 3 for an MSSP in the previous diagram is for the SOC to implement a data model (DM) suited for its usage.

The actual definition of data model can easily be found on the web but simply put, for a SOC, the data model is a mapping of a label to a type of data. **There can only be one label for one type of data and there can only be one type of data corresponding to a specific label**.

For example, the SOC could decide to associate the data "*Full process's executable file path, name and command line used for execution*" to the label "*process*". Then, whatever the sensor, whatever the context, if there exists a label "*process*" then the data associated with it will be "*Full process's executable file path, name and command line used for execution*".

The application of the data model is maybe the most powerful industrialization option for a SOC and especially an MSSP as it creates an abstraction layer between the sensors and the usage of their logs:

- The detection specialist needs only to search the anomalous behavior they want to detect by referring to the label. Since the label always corresponds to the same type of data and the logs are consistent (thanks to the logging policy), then the detection rule becomes generic, in the sense that it can be

applied to multiple sensors of the same nature. Because that detection rule is generic, it can be run against multiple sensors across multiple customers without any change to its code.
- The response specialist doesn't need to know the editor and version of a firewall, for example, to use its logs for their investigation. They can search whatever label contains the data they are after and they will get the results they need.
- The threat hunter can just copy and paste their queries from one customer to the next and trust that it will work as intended.

The hard part, of course, is to create a data model that is suited for the SOC's usage. There are many SIEM or XDR editors, for example, who propose their own data model, and if any matches the needs of the SOC or MSSP then go for it. Otherwise, it is always possible to take the closest there is to the needs and build on it.

There are a few recommended points to follow in order to achieve a good data model for a SOC:

- Establish a strong hierarchical nomenclature for the labels, i.e. there surely will be "sublabels" relating to data that is to be grouped under one parent label, then maybe something like *<root><boundary_character><level_one><boundary_character><level_two>*... can do.
- By definition **there can only be one label for one type of data and there can only be one type of data corresponding to a specific label**, make sure that is and stays true - check and enforce it.
- Test it against multiple sensors of different natures to make sure it is viable before using it in production because it will be painful to modify it - that is, modify an existing label or data definition, adding is OK - afterwards.

5.3.3. Parsing

The definition of parsing here is the mapping of data contained in logs to labels. Parsing is therefore the application of the SOC's data model onto logs whose content and format are expected as per the logging policy. Parsing is also the link between the raw logs coming from the sensors and the tool in which they are used, e.g. a SIEM.

Configuration of parsing will then depend on the format of the raw log - depending on the sensor type, editor and sometimes version - and the tool that will use it.

Another point in the industrialization option number 3 for an MSSP in the previous diagram is for the SOC to store parsing configuration for each variation of sensor/editor/version/target tool.

This is yet another step which is hard and tedious to achieve, but at this point chances are that the SOC logging policy will not be the default one for the sensors and the SOC data model will not match existing default parsers in the SIEMs used… Consequently some parsing configuration must be done before the data is usable, hence it is best not to fight the inevitable, but capitalize on it by storing the configuration.

This way, parsing is done exactly once and is then reusable for every customer.

By the way, the last point in the industrialization option number 3 for an MSSP is to automate the deployment of parsing configuration to the tools… At that point it is only a matter of copying and pasting configuration, so why let an error prone and slow human do it when a robot could do it all faster and better - and let the human use his time to create parsing configuration, which is a task that a brainless robot could not do.

5.3.4. Monitoring

The consistent logs of expected nature and format are now correctly parsed according to the SOC data model. It's a paradise on earth for all SOC analysts and now they won't leave for any other SOC because they don't want to go back to wasting their time on idiotic issues.

The final step is to keep things that way and to do that, close monitoring of the data is required.

Here is a checklist of things to monitor to ensure the detection input, which is the logs parsed according to the SOC DM:

- Logs overall per time unit volume - the volume for a specific sensor with a specific logging policy is expected to remain more or less the same.
- Raw logs consistency - once the policy is applied, the raw logs should always have the same format.
- Parsed logs consistency - once the parsing configuration is applied, the parsed logs should always be broken up into the same labels.
- SOC data model application - each label should refer to one type of data only across all sensors and all customers, a misconfiguration in the parsing can happen and will result in partial detection.
- Logs sources and volume - a source suddenly sending any, more, fewer or no logs anymore unexpectedly is abnormal.

- Logs retention periods - make sure logs are not deleted before the end of their retention period because of storage issues, for example.
- Logs collection tools health - logs should be collected in almost real time for immediate use by the tools (e.g. SIEM), logs arriving in the tool hours or days after they have been created by the sensor is an abnormal event.

Depending on the infrastructure and technological choices, there may be other points to monitor but the gist is to make sure that logs' production, collection and ingestion runs as smoothly as when it was first put into production.

There is nothing worse than to believe that the SOC detection is working and the coverage is in a certain state when the reality is far from it, all because there was a change in some firewall rule and all logs forwarded to the SIEM since have been dropped and irrevocably lost.

5.4. Main SOC Tools

There are a few main tools used by any SOC or MSSP and it is best to have an overview of recommendations and common mistakes in order to get the most out of each.

One recommendation can be applied to all of those tools and is really a general best practice. Be sure to perform continuous automated, if possible, health checks for every and all tools used. These are meant to ensure that the tools are operational, running as usual, that the SOC is running smoothly and that the analysts can work under normal conditions.

5.4.1. SIEM

The Security Information and Event Management (SIEM) is the main and most important tool for many SOC and MSSPs.

Its goal is to centralize artifacts, such as sensors' event logs, in order to be able to run detection rules against them and fire alerts if anomalies have been detected. Also, the analysts can use it for their investigations by querying it for the artifacts it contains.

5.4.1.1. Generic detection rules

A detection rule is a SIEM (or other tool) query that runs on a schedule and aims to detect the anomalies for which it was created in event logs or other artifacts.

The detection rule mentions the artifacts it queries, the labels referring to the data it needs and can include a list of patterns to match, a list of patterns it should not match (called WhiteList or WL) and other thresholds used to determine whether there is an anomaly. Therefore the query for the detection rule has "static parts" (i.e. invariable for the same tool with regard to the environment) such as the detection logic and "dynamic parts" (i.e. variable with regard to the environment) like the labels, the patterns, etc.

Under normal circumstances, because a detection rule has dynamic parts, it must be rewritten or at the very least modified and adapted for each new deployment on another tool - SIEM of the same editor but in a different environment. This is very inefficient because there are a number of different versions of the same detection rule (logic) deployed in multiple environments, meaning that with every update to the logic, there must be adaptation to all deployments. This is very hard to track to keep every rule up to date and very time consuming to do.

Fortunately, as pointed out by the industrialization option number 4 for an MSSP in the previous diagram, there are options to build generic detection rules that can exist in one centralized place and be (automatically) deployed or updated to every environment at once.

The prerequisites for this to work are as follows:

- One centralized place to store all generic rules - preferably a DevOps platform
- Writing the rules' queries such as the dynamic parts are stored somewhere else and queried at run time. The solution for this strongly depends on the SIEM used but for example, instead of having a hardcoded WL in the query, there could be a subquery of sorts for a CSV file stored elsewhere containing the WL. Then the maintenance of the WL is deported onto that file and the dynamic patterns list becomes a static subquery. The point is that the dynamic parts must not be embarked in static queries.

Writing detection rules such as this may be a huge change in how things are done, but the work is done only once and reused everywhere, instead of being redone every time and adapted for each environment.

If there is a SOC data model and it has been correctly enforced, then things can be taken one step further. Since the detection rules can rely on the DM and do not have to match the parsing and content of such or such sensor, they can be applied to all sensors that produce logs containing the data used by them.

In other words, where there was one detection rule per firewall editor for the same detection logic, there can be one detection rule that applies to all editors.

Even better, where there were three detection rules looking for IP addresses flagged as Indicators of Compromise (IoCs), one covering WAF logs, another one web server logs and the third one firewall logs, there can be one detection rule that applies to all three, because the data will be under the same label across all sensors. That one rule will, of course, work against all editors of these sensor types across all environments of all customers.

To put things in numbers, let's say there are 5 customers each with 1 sensor of each WAF, web server and firewall. Between them they have 3 WAF, 4 web servers and 2 firewalls in common, the rest being from different editors. This means that there are 3 different WAF technologies, 2 web server technologies and 4 firewall technologies:

- With "classic" detection rules, there would be 5 * (1 + 1 + 1) = 15 rules to maintain.
- With generic detection rules, there would be 3 (WAF) + 2 (web servers) + 4 (firewalls) = 9 rules to maintain.
- With generic detection rules + an enforced SOC DM, there is… 1 rule to maintain.

5.4.1.2. SIEM-abstract detection rules

Detection rules build can be taken one ultimate step further, as advertised in industrialization option number 4 for an MSSP in the previous diagram.

An MSSP has to deal with a number of different SIEM editors, depending on the commercial choices made by the company.

In that scenario, each detection rule mentioned earlier also has to be written and maintained across SIEM technology and query language.

The issue is of lesser importance because it is easier to maintain 5 flavors of the same detection rule, each flavor corresponding to a different SIEM query language, than it is to maintain 15 rules for the same SIEM. Nevertheless the nature of the issue is the same.

A best practice would then be to store each detection rule's logic in a generic format, having nothing to do with a particular SIEM query language. This will not always be possible, depending on the generic format and the complexity of the queries, but it makes for easier updates because the translation is from the generic format to specific SIEM query languages and not from one query language to others.

Finally, maybe the hardest feature to achieve would be to automate the translation of the generic format rules to each SIEM specific query language. There are many such "translators" that exist, but they rarely correctly optimize the queries, when they even correctly translate the logic at all. Furthermore, none implements the more important points discussed before, such as subqueries for dynamic parts and use of the SOC DM.

5.4.1.3. Pre-production instance

The "integration", "development", "validation", "pre-production" or other such SIEM instance is used to properly test everything before applying it to production.

This instance is a must have to allow enough tests on deliveries, as described in the detection lifecycle. It is used to test all releases before deploying them to customers' production instances.

It is also used for rules design and development as all logs resulting from the tests are stored there and everyone working on rules development has the same data.

This instance is better the closer it mirrors to the production instance, especially for performance tests that aim to measure resource usage.

5.4.2. SIRP

The Security Incident Response Platform (SIRP) is the memory of the SOC's detection and response.

It hosts every ticket that has been created as a result of an alert, whatever its source, along with the detection qualification, status and notes.

This tool is central for the response specialists, as it is the one they will monitor to know on what they should work (alerts and priorities) and it is also the one on which they will report all the work they have done during the response phase.

It is imperative that this tool is as ergonomic as possible for the analysts because it will directly impact the time they spend handling tickets. For example, if the tool has a low responsiveness, is hard to search, doesn't display the data needed properly and clearly, or is simply bugged, then the analysts will have to compensate by spending more time and energy to understand the work they have been assigned and report their actions.

Nobody likes a laggy, buggy tool that disables people instead of enabling them. Well, SOC analysts are people, too.

Like any other commercial product, vendors tend to bait clients into buying their tools by advertising that it does everything better than the others, and then some. The current trend is about automation, therefore vendors push hard on this aspect, and include some SIRP features in their product to be able to say that yes, their product is also an SIRP. However, the SIRP is the cornerstone of the SOC and needs to be considered as such, and not just as a feature of something else.

If you are reading this and are in charge of making the decision of what tool to use or buy, **please, please, please ask your SOC analysts for their opinions and strongly consider their position**, as this will be their main tool. There are of course multiple criteria to consider, like the cost, how well the tool integrates in the existing environment, the KPIs it can produce, etc, and it is very likely that no tool will be perfect, and that some tweaks will be required. It boils down to a matter of cost, but the hidden and indirect costs must be **carefully** considered in the equation, as the overhead for analysts can rapidly add up in time spent, and the negative impact on working conditions will have a long term human cost.

It is very rare that tools correctly perform multiple very different tasks well, especially when it comes to SIRPs. On the other hand, it is sadly too frequent that once the existing top of the line expensive as possible do-it-all new tool is here, it becomes clear that it doesn't perform close to what it should with regard to SIRP features. So countless developers are tasked with fiddling with the tool to make a mediocre SIRP because "it costs so much it must perform well".

5.4.3. SOAR

Speaking of fiddling with tools to make them do what they are not meant for, the Security Orchestration, Automation and Response (SOAR) platform is basically a toolbox that holds scripts used together to orchestrate and automate **response**.

The key takeaways here are that a **SOAR is not an SIRP** and that the SOA**R** is used for **Response**.

When it comes to SOARs, there are a few DOs and a few DON'Ts. The DON'Ts have been slightly hinted at above:

- **DO NOT** try to use a SOAR as an SIRP - unless the consensus amongst SOC analysts, after they could test it, is that it is good enough as an SIRP.
- **DO NOT** use a SOAR for something else than response. In particular, as shown in the incident lifecycle earlier, the detection phase comes before the response phase. Therefore the SOAR must wait its time in the response phase to be put to use **and not** be tasked with things such as recurrent artifacts collection or running SIEM queries for detection purposes.

Most of the DOs depend on the technology chosen but some are more generic:

- **DO** break up playbooks in small unitary actions so that they can easily be assembled like blocks to build bigger and more complex playbooks. This also makes for easier maintenance if some action needs to be updated or fixed.
- **DO** divide the creation of playbooks between developers and response specialists. The response specialists should express their needs of unitary actions to developers who will script them accordingly. Then the response specialists can build all the playbooks they like using the set of blocks they have been given.

Always remember that a SOAR is "just" a platform for storing and executing pipelines of scripts at the right time with correct parameters. Everything else is some sort of wrapper.

5.4.4. DevOps platform

As with other tools, there are many DevOps platforms that are available for use (free or otherwise), with a few that are a bit more popular.

To put it simply: there is no realistic way around a DevOps platform to implement the industrialization options related to automation. If there is no automation then there is no real scaling, including for advanced detection capabilities.

The DevOps platform enables CI/CD/CD (Continuous Improvement/Continuous Delivery/Continuous Deployment) which can be summed up by automations that let the platform test, package and send to production any and/or every bit of modification made, depending on the configuration.

Also, modifications are versioned and stored so that it is easy to find who modified what when and to perform a rollback to a previous version, if needed.

A good "IT guy" (m/f) is a lazy IT guy, the lazier the better. It is just preposterous to work with or around computers and not use them for their greatest strength, which is the automation of tasks that do not require a human brain.

It is not acceptable to have people copy and paste whatever, over and over again. So lazy up and think very hard whether you can perform the task you need to do just once and have the computer take over for the rest of time. If it's possible, then automate it and if not then it must require some thinking, which is the greatest strength of humans.

5.4.4.1. Repositories

The DevOps platform hosts repositories which hold the files uploaded onto it, very much like folders.

The structure of the repositories must be well thought out and correctly implemented from the start as it is often quite a pain to refactor already working repositories if the architecture is flawed. This structure is important as it will help ease the development, configuration and access control for authorized users.

The architecture will vary depending on the needs, constraints and possible mid to long term evolutions, but a good way to approach it is to think in terms of groups of features. These groups are composed of features that are close enough in terms of need covered and people using them.

An architecture for an MSSP could look as follows:

- **Tools:** Contains all the scripts that do not achieve any feature by themselves but are called and used by scripts from the other repositories to build the features.
- **Infrastructure:** Contains all scripts and configuration files that deal with the automation of infrastructure deployment, either for initial deployment or for MRO/upgrades.
- **Datamodel:** Contains the SOC's data model and the parsing configuration for all sensors on all tools used by the SOC.
- **Production:** Contains all scripts and configuration files that deal with the usage of the SOC/MSSP tools and their deployment to its customers. This repository tends to the "front end" (parsing, detection rules configuration, WL, etc) while the *Infrastructure* one tends to the back end.
- **Detection:** Contains all generic detection use cases the SOC has - more details on this soon.
- **Packages:** Depending on the tools used by the SOC, such as SIEMs, there may be some package, bundle or other wrapper needed to hold all configuration that will be pushed onto the tool. This repository contains these product and editor specific packages used by the SOC.

Each of these repositories also contain the necessary scripts and configuration required to perform all the tests needed to validate modifications. The user privileges would be set, as always, on a need to have basis: every read, write and execute right must be given only to those who need it to perform their jobs.
Also, DevOps best practices are advisable, as always, for every repository.

5.4.4.2. Pipelines

Pipelines are basically a series of automations launched one after another with configuration options such as dependencies of the downstream jobs on the upstream ones. These automations are scripts given some arguments to perform some actions.

Pipelines are therefore a key component of the DevOps platform and can have different purposes:

- Building pipelines which generate files, packages and what not with each new modification of the source files.
- Testing pipelines for modified files and built components. These will perform, for example, the "unit tests" mentioned in the detection lifecycle.
- Deploying pipelines for built components. These will automatically with every modification, on a schedule or semi-automatically with a push of a button deploy the components to the target(s).

It is recommended that only the jobs needed with regard to the modifications processed be displayed to the user and launched by the server in order to avoid confusion, misclicks and resource waste.

5.5. SOC Use Case

SOC Use Cases exist to answer a specific need of detection and protection of the intelligent systems – IT, OT, cloud... - of an entity. They form the core of the SOC and as such they are its most critical assets. They absolutely need to be well thought, defined, tested and maintained.

A SOC Use Case has multiple components but depending on choices made, these components may not be all stored in the same place. The important thing is that the components reference one another so that it is (easily) possible to get the full picture of the use case. Any missing component will prevent someone at some point from doing their job properly.

These components are:

- Risks covered
- Knowledge
- Detection rule
- Response procedure
- Control logs - stored logs from simulations or generated logs containing anomalies that the detection rule should and should not detect.
- Simulation scenario - either in plain text for manual simulation or a script to automate it.

5.5.1. Risks

There is no way for a SOC to defend against all possible threats for different reasons such as technical limitations, human limitations, knowledge limitations, etc.

Therefore, the defense must be oriented using specific criteria and one way to achieve this is to use a risk based approach.

The risk that needs to be addressed for the entity the SOC protects is a business risk such as "Factory production stops" or "Intellectual property theft". The first step is to clearly identify these risks.

Then, the business risk needs to be translated into a cyber-security risk which could be for example, respectively for the cases mentioned before "(Distributed) Denial of Service on factory devices" and "Data exfiltration from critical projects servers".

Finally, the cyber-security risks must be decomposed into attack scenarios that can be, in turn, decomposed in unitary steps the attacker would take. The detection rules will be based on these steps.

Depending on whether it is a SOC having only one environment to monitor or an MSSP with multiple customers and environments, the risks that can be associated with the SOC Use Case will vary. In the case of a SOC, all the risks up to the business risk can be associated with the SOC Use Case right away. In the case of an MSSP, only the unitary steps or the attack scenarios could be linked directly in the SOC Use Case, while the rest of the risk mapping should be done on a per customer/per environment basis.

It is strongly recommended that the categorization and description of the attack scenarios and unitary steps are based on already existing and widely known frameworks, such as the MITRE ATT&CK matrices for example.

Always keep in mind that frameworks such as these **only reference known, already observed attack patterns** and are therefore in **no way an exhaustive list of possibilities**.

Also, multiple entities have published lots of recommendations for prevention and protection. These recommendations can be used both for risk and attack scenarios definition.

5.5.2. Knowledge

The following is a list of items that must be thoroughly documented in order to be able to maintain and upgrade the SOC Use Case so that everyone using it can correctly do their job.

- Names of the people involved in the ordering, creation and upgrade of the Use Case – these should answer the question "Who has created the Use Case?"
- Limitations and blind spots of the Use Case and references to all relevant materials used to create it – "What is the technical scope of the Use Case?"
- The virtual scope of coverage of the Use Case in terms of type of logs, devices, users, etc – "Where is the Use Case applicable?"
- Timestamped statuses of all tests performed on the Use Case – "When has this Use Case been cleared for production?"
- Detection rule, response procedure, control logs and simulation scenario - "How should the Use Case work and be used?"
- Business risk, cyber-security risk and the attack scenarios mitigated – "Why does the Use Case exist?"

5.5.3. Detection

The detection rule is the most obvious part of the SOC Use Case as the main goal of creating one is to detect anomalies in the environment.

There are however a few attention points to follow in order to ensure that the detection rule is pertinent, usable and maintainable.

The first one is about the inputs for the detection: the data, in the form of logs. It is crucial that the logs are complete and correctly parsed. In the end, this is about trust: the customer (internal for a SOC or external for an MSSP) trusts its SOC for protection, the SOC trusts its Use Cases for detection and therefore the whole trust chain relies on the availability and integrity of the logs fed to the Use Cases through the SIEM.

The second attention point concerns the dynamic nature of the detection rule. The rule must be coded in a way that enables dynamic – i.e. not hardcoded – tuning.

It is highly recommended that every detection rule embarks one WhiteList – per rule – for tuning. In any case it is mandatory that the WL, pattern list and threshold(s), if applicable, can be edited directly in the tool and that those modifications have an immediate impact on the detection rule.

Although the normal workflow could be that the changes in WhiteList, pattern list and thresholds be done through CI/CD, a technical issue or delay in the CI/CD chain must be anticipated. Therefore the possibility of dynamic editing must exist to ensure the best effectiveness and resilience possible of the tool and the SOC analysts in charge of the response. As this should be rare and an emergency, a break-glass account could be just the solution - of course, the use of this account would be monitored.

The third attention point deals with the actual code of the rule, the one that will be interpreted by the tool. The code can either be editor specific, meaning that the detection rule is written specifically for such or such SIEM for example or it could be an editor generic rule that is translated into specific code at some point in the CI/CD pipeline.

Whatever the case, the actual editor specific code needs to follow the best practices for that tool and be as optimized as possible in terms of resources that the tool will spend on that rule. The goal of the detection rule is to detect and the first action after an alert is created is to triage it; this means that the rule must create an alert that has just the right amount of information – one that is too verbose or not enough will negatively impact the analysis.

The last attention point is about the building of detection rule logic. The detection rule goal – what it is meant to detect – is based on the work done with the risks and attack scenarios definition.

Hence each detection rule must be tested in multiple ways as described in the detection lifecycle to ensure that the rule actually works and detects what it is supposed to detect.

5.5.4. Response procedure

The response comprises the investigation of the alert created by the detection rule and the remediation, if need be.

As the investigation is based on an alert created by a well-defined detection rule, it can – more or less easily – be standardized and broken down into unitary steps. Also, since the majority of the alerts raised by the tool (e.g. SIEM) do not end up being actual widespread attacks, most of the remediation can be standardized as well. It is recommended that these be standardized as it brings many benefits: shorter training time for the SOC analysts, better understanding of the scope of the SOC for the analysts, increase in efficiency and reliability of the response specialists team as a whole.

It is best that these standardized procedures take the form of simple diagrams which are easily accessible by the SOC analysts. Of course, depending on the maturity of the SOC, the investigation and remediation can be partially or completely automated, for example with the use of a SOAR.

In order to maximize efficiency and maintainability, it is best to build generic, basic "diagram blocks" and then assemble them to create the diagram that will be used by the SOC analysts in the end. This way, if part of an investigation or remediation changes for whatever reason, it is quite simple to update all the diagrams at once by updating the corresponding block.

The diagrams and the basic blocks must be as basic as they can be – the goal being that there is no misunderstanding and the actions done by the analysts are always the same, improving the overall reliability of the SOC. The other advantage is that if and when a SOAR comes into play, it will require basic, step by step instruction to be able to automate once manual work – the translation from a basic step by step human readable diagram into instructions for the SOAR should be an easy one.

5.5.5. Alerts correlation

Oftentimes, there are a number of detection rules that tend to detect compliance issues, misbehavior from users or specific usage of an account for example. These rules can be seen as "hygiene" rules from a security perspective.

Hygiene rules, together with some basic detection rules which are investigated unitarily, tend to yield a high percentage of False Positives and/or True Positives that are legitimate actions. Moreover, investigations on these alerts are more often than not quite boring for SOC analysts, especially if they make the better part of all the alerts to handle.

Although they are necessary because they can provide context and a nice trail of clues as to what the attacker did in case of an actual incident, it is highly unlikely that advanced actors are detected by the SOC using these rules in a unitary fashion.

In order to both detect advanced threat actors and limit the number of pointless – from a cybersecurity point of view – investigations for the analysts, it is recommended to create and use "advanced detection" or "alerts correlation" rules. These rules are based on the alerts generated by all the classic detection rules and correlate these anomalies over an extended period of time and across the attack killchain.

In doing so, not only the unitary alerts from hygiene rules can be ignored but these rules become quite pertinent as they can bring more pieces to the "attack puzzle". It is then possible and advisable to have more detection rules that detect less pertinent unitary anomalies and only use them for alerts correlation.

More importantly, this correlation enables the SOC or MSSP and its analysts to take a step back from all the alerts generated and actually see the big picture of past or even ongoing attacks.

All in all, alerts correlation is a gigantic step towards improving the SOC capabilities to detect complex attacks but they require many prerequisites - all of which have been discussed before.

5.6. Conclusion

The main takeaway from this chapter is that tools have to be configured and customized to an extent in order to extract their full potential. If they are used with default settings and the users - here SOC or MSSP personnel - do not understand their purpose, how they work and how to use them then they are just a waste of time and money for the SOC as an entity and for every user as well.

On the other hand, customization should be limited to that: customization. Using a commercial or open source tool and then sinking countless days of development in it to make it do what you wanted is not the way to go in most cases - it would be a lot better to switch to another tool that corresponds to the needs or use an additional one in complement.
If you buy a car, you may want to change the paint, add electric windows or get heating seats but you wouldn't go and try to add wings to it in the hopes it would fly. It's not because an application is more abstract than a physical object that your thought process should be any different.

6. Conclusion

A lot of current SOCs and MSSPs do not seem to perform as well as their customers - internal or external - would expect and can be seen as an overpriced check in a todo list from an external point of view. Internally, the jobs can be short term exciting - especially for the building phase and for the management - but do not end up being fulfilling or even interesting long term for most of the analysts population.

This state of things is due to multiple factors:

- The majority of the people attracted by these kinds of jobs may not be understood as well as they should be, especially when it comes to their motivations, needs and expectations.
- The priorities for the money spent may not be the right ones: money might be wasted on short term temporary fixes when it should be invested in mid to long term evolution.
- The tools and methodologies may not be as well thought out, implemented or industrialized as they could be.
- The internal structure and processes of the SOC/MSSP teams may not be as optimized as they could be.
- Maybe the most neglected, but nevertheless crucial point of external (to the SOC or MSSP) communication may need some improvement.

They are of course possible solutions to all these issues:

- Understanding that people are the key element to a SOC or MSSP and modeling the work environment for the type of profiles that are recruited.
- Including technical people in financial discussions, explaining to them the financial objectives and listening to the compromises they propose to balance short and long term gains within the financial objectives.
- Reviewing the tools and methodologies and having the courage to make a change even if it means going through a short term transitional hit for the mid and long term benefits.
- Questioning the structure and processes and aligning them with the goals set for the SOC/MSSP to achieve.
- Explaining to every member of the SOC/MSSP (management included) that explicitly communicating on the work they are doing is key and helping them find the correct metrics that reflect the actual work they do and that can be understood by their customer(s) and the executives.

When the issues described above are resolved and the SOC or MSSP is correctly industrialized (in all its aspects), its actual potential shows by itself, even to non technical people.

Then and only then the journey that will shift SOC and MSSP paradigm from "reporting past malicious activity" to "catching malicious activity as it happens" can truly start and safely continue until it ends.

This means that the majority of the SOC's/MSSP's resources can be used for actual improvement in detection and in response leading to more detections, increased pertinence, quicker detection and response time, existence of and better remediation…

At last, this will achieve the goal of reducing or even eliminating the overall frustration around SOCs and MSSPs, hopefully for good.

7. Acknowledgements

I am always cautious of putting any Personally Identifiable Information (PII) publicly accessible on the internet and even more so on the web, so I will not name anyone here.

However, I sincerely thank the people who agreed and took the time to read, discuss, advise and proofread my work, because I strongly think that the present version is incomparably better than the first one I showed them, thinking it was ready.