Internship: SOC Detection Engineer

Reflection

Bachelor's degree in IT
Field: Cloud and Cybersecurity

Name

Jarne Willems

Academic year 2024-2025

Campus Geel, Kleinhoefstraat 4, BE-2440 Geel

Contents

1 INTRODUCTION

In this report, I look back on my internship at Cegeka, where I had the opportunity to be part of the Detection Engineering Team for several months. My internship assignment focused on analyzing and harmonizing detection rules within two SIEM platforms: Splunk and Microsoft Sentinel. The goal of the assignment was to optimize internal processes by identifying and resolving inconsistencies in detection logic between the two systems wherever possible.

In addition to the main assignment, I also had the chance to contribute to extra tasks such as threat hunts, use-case validations, and the migration of detection rules. These experiences gave me a broader understanding of how a Security Operations Center (SOC) operates and introduced me to the collaboration between different teams within the cybersecurity domain.

This report begins with a content-focused reflection on my internship assignment and the approach I followed. I then reflect on my personal development throughout this period, both technically and in terms of communication. Finally, I conclude with a general summary of what I've learned and how I plan to carry this experience forward in my professional growth.

2 REFLECTION ON THE INTERNSHIP ASSIGNMENT

For this first section I'd like to give you an honest reflection on my internship assignment.

My assignment focused on detection rules in two SIEM environments: Splunk and Sentinel. Each of these products has its own set of detection rules. Some are unique to one environment, others are similar rules present in both but produce different results, and some are identical across both and yield the same outcome.

Cegeka maintains a use-case library that includes all rules used in both products. However, the issue I encountered at the beginning of my internship was that it wasn't entirely clear which rules existed in both environments, or how their detection logic differed. This led to some challenges, for example, analysts could be confused by inconsistent results, or it was difficult to determine which product offered the most effective detection logic.

The goal of the assignment was to optimize internal business processes. By researching the detection coverage of both Splunk and Sentinel, I was able to clearly identify the differences between the two. With guidance, I then systematically worked to address these differences. All of this contributes to better service for clients, who can now expect an optimized detection service, ultimately improving their protection against cyber threats.

2.1 My approach

I split the internship project into three phases:

Phase 1: Inventory of all detection rules
During this phase, I created an overview and marked the differences between the two products. I looked at which rules were unique to each and which ones overlapped. I documented all observed differences.

Phase 2: Comparison of overlapping detection rules
Whenever I found overlapping rules, I compared the version in Splunk with the one in Sentinel. If there were differences in detection logic, I marked them accordingly.

Phase 3: Harmonization of overlapping detection rules
In the final phase, I resolved the identified differences. The overlapping rules were aligned to ensure equal detection coverage across both products.

This turned out to be a successful approach. My goal was to work as efficiently as possible, and breaking the work into phases proved to be a smart move. It helped me break the overall goal into smaller steps, which added structure and kept me from getting lost in the bigger picture.

2.2 What have I realized?

Cegeka now has a complete inventory of all detection rules, with clear documentation of the differences between the two systems. In addition, together with my internship mentor, I created a repeatable procedure that I followed myself to compare and harmonize overlapping rules.

The development of this procedure was not very straightforward and came with its own set of challenges. One major issue was identifying consistent criteria for comparing all pairs of overlapping detection rules, not just one individual pair. This proved to be a significant obstacle, as each rule had its own nuances that made standardization difficult.

Additionally, while the procedure made sense to me, it wasn't immediately clear to others. This lack of clarity led to some confusion. However, with time and feedback, the necessary adjustments were made to improve its structure and make it easier to follow for everyone involved.

In total, I identified 26 overlapping rules (13 pairs). All 26 were compared, and I successfully aligned 25 of them across both environments. I also made a recommendation to add two new detection rules.

This was the point where the assignment was supposed to end, but since I had some time left in my internship period, I discussed with my mentor Lander and team leader Steven and we decided to actually implement the rules into client environments.

Nine of the twenty-five rules passed testing and were successfully deployed to clients.

The number of deployed rules may seem low, but this is mainly because rules require thorough testing before deployment. Testing requires relevant data or logs, and because many of these rules detect very specific activities, there often weren't suitable logs available.

Simulating events or creating test data could have been an option, but this was only possible in Sentinel -not in Splunk- so my mentor and I decided against it.

Of course, I would have liked to deploy all twenty-five rules, but one important lesson I’ve learned is that things don’t always go as planned in a business environment, just like some projects get postponed due to more urgent priorities.

That said, I’m glad I got to experience the deployment process firsthand. It gave me a complete picture of what it means to work as a detection engineer, and I’m very grateful for that.

2.3 Additional assignments

While working on my main assignment, I also handled a few side tasks for Cegeka, which I explain in more detail in my realization document:

- **Use-case validation for a client**
- **Two threat hunts**
- **Migration of use-cases from Splunk to Sentinel**

For the use-case validations, I reverse-engineered detection rules to identify logic flaws. This indirectly helped with my main assignment as well, since it gave me insight into which criteria I could use to compare overlapping rules. I developed a deeper understanding of how detection rules are structured, which made it easier to define comparison criteria.

During the threat hunts, CSIRT investigated two specific vulnerabilities. Working together with the detection engineers, we created detection rules for these cases. With support from Kris, a member of the DET team, and help from CSIRT, we tested the rules and successfully deployed two of them to client environments.

This task really highlighted for me the importance of collaboration between teams within the Security Operations Center. I also had the opportunity to observe CSIRT's testing process, an eye-opening experience for which I'm very grateful to Cegeka.

Finally, I worked with Maarten, a DET team member, on migrating detection rules from Splunk to Sentinel. This essentially meant copying existing rules from Splunk to Sentinel, but it gave me the chance to build detection rules myself.

These rules needed to be tested too. Using Sentinel's test environment, I simulated scenarios to trigger the rules. This involved a bit of red teaming: mimicking suspicious behavior to see if the rules would activate. I was able to build and test seven detection rules during this task, which are now set to be deployed to the customer's Sentinel environment in the near future.

This task significantly deepened my understanding of both Splunk and Sentinel. To translate the rules effectively, I had to grasp how Splunk's Processing Language (SPL) worked before rewriting them in Sentinel's Kusto Query Language (KQL).

The more rules I created, the easier the process became. Simulating events helped me grasp the full picture of what needs to happen for a detection rule to trigger.

All in all, I found this one of the most enjoyable assignments during my internship, and I'm especially thankful to Maarten for trusting me to assist with his work.

3 PERSONAL REFLECTION

On a personal level, I experienced a lot of growth during my time at Cegeka. From the very start, it was made clear that I would need to work independently. As an intern, that's not always what you want to hear. You're already unfamiliar with the company, and now you're expected to solve tasks largely on your own.

Although that initially sounded a bit intimidating, it turned out to be less daunting than I had expected. There was always enough guidance available, but I wasn't handheld. The situations that pushed me outside my comfort zone often taught me the most, provided I kept the right mindset.

3.1 Hard Skills

At the start of my internship, I had no prior knowledge of the technologies I would be working with. This made for a slow start, but the learning curve became clear as I progressed through my assignment.

SPL (Splunk Processing Language) and KQL (Kusto Query Language used in Sentinel) are two very different query languages, each with its own syntax, strengths, and weaknesses. Understanding the difference between them was difficult at first, you could compare it to learning French and Spanish. There are similarities, but certain things are just not the same.

After a while, I began to recognize patterns between the two: the structure of rules is mostly the same, and although syntax elements may have different names, they often serve the same purpose.

I got the hang of both languages relatively quickly, but the real challenge was understanding what a detection rule actually *does*. Sure, you can look it up or ask our AI assistants, but I wanted to *truly* understand why a rule existed, not just what it did.

This is where most of my time went, and it really opened my eyes to the importance of experience in this field. As a cybersecurity professional, you need in-depth knowledge not only of query languages like SPL and KQL, but also of systems, networks, operating systems, applications, and much more.

One example that stood out to me is Active Directory (AD) and its associated policies. I didn't have much prior knowledge about this, but it's hugely important, think of Group Policies, audit policies, etc. This is definitely an area I plan to improve on moving forward, because my lack of knowledge in this area caused me some issues during the internship.

Of course, it's an internship and the whole point is to learn as much as possible, but it gave me a clear view of where the gaps in my knowledge are and what topics I need to master to move forward, like AD, for instance.

3.2 Soft Skills

When it comes to soft skills, there are a few things I became more aware of during my internship.

One thing that became clear early on is that I sometimes struggle with communication, especially when trying to explain what I've been working on. I know exactly what I'm doing and how I'm doing it, but I often find it difficult to put that into words for others.

This is something I actively tried to improve during the internship, but it's still a work in progress. Both my team leader Steven and my mentor Lander told me that this gets better with experience, but I think it's good to already be working on it consciously.

Another thing I came to realize (thanks to feedback) is that I didn't always take the rest of the SOC team into account.

For example, when creating a detection rule, it's not just about whether the rule works, but also whether it's clear and actionable for SOC analysts. The output of a rule should be useful enough for analysts to investigate further or draw conclusions from it.

Another example is the procedure I created. While it made perfect sense to me, it wasn't immediately clear to others.

I learned that whenever I create something, be it a rule or a procedure, I need to pause and ask myself: *Would this be clear to someone else?* I started putting myself in someone else's shoes and reviewing my work from a third-person perspective.

That was tough at first, but over time I started recognizing patterns in my own communication and clarity issues. I then made a conscious effort to systematically address those.

Not everything was a challenge though, I also made progress in asking more questions and reaching out to others when needed. I spoke to pretty much everyone on the SOC team at least once, usually on my own initiative.

I also worked closely with other members of the DET team on various tasks. These collaborations required close teamwork, which gave me the opportunity to improve my teamworking skills.

At the same time, I was able to further develop my independence. I worked on my tasks autonomously, while regularly checking in for feedback with my mentor and team leader. My mentor encouraged me to take the initiative in planning these meetings, so I took responsibility for scheduling and preparing them myself.

4 CONCLUSION

During my internship at Cegeka, I gained valuable hands-on experience in detection engineering within two major SIEM environments: Splunk and Microsoft Sentinel. My main project focused on aligning and optimizing detection rules between these platforms to improve internal processes and ensure consistent coverage across both systems. The impact of this work went beyond technical implementation, it directly contributed to a more robust and reliable detection strategy for Cegeka and its clients.

One part of the assignment that stood out to me was the opportunity to **simulate attack behaviors to test and validate detection rules**. This process required me to think like an attacker, identifying how suspicious actions manifest in logs, crafting queries to catch them, and then safely triggering those scenarios in a test environment. While my role was that of a detection engineer, this task introduced me to **threat hunting techniques**: simulating adversary behavior and proactively validating whether the rules would actually detect real threats. I found this aspect of the internship especially engaging, and it helped me understand the deeper connection between detection engineering and threat intelligence.

On the technical side, I developed skills in SPL and KQL, deepened my understanding of how detection logic functions, and identified key areas for improvement, particularly in foundational topics like Active Directory and logging policies. This experience gave me a clear view of where I stand and what I need to focus on next to continue growing in the field.

Beyond the technical, I made real progress in soft skills, especially in communication and collaboration. I learned to explain my work more clearly, take other teams' needs into account, and ensure the outputs I created were usable by analysts in a real-world SOC setting. I also became more independent, took initiative in planning my work and check-ins, and built strong working relationships with team members across different roles.

Overall, this internship confirmed that I want to continue pursuing a career in cybersecurity. Whether I grow into a SOC analyst, detection engineer, or eventually a threat hunter, I’m confident this is the right direction for me.

I’m deeply grateful to Cegeka for the trust, responsibility, and support they offered throughout this internship. The experience gave me not just technical knowledge, but real insight into the mindset and skills required to succeed in cybersecurity.