The Intelligent Process Lifecycle of Active Cyber Defenders

DESIREE SACHER-BOLDEWIN, Finanz Informatik GmbH and Co KG
EIREANN LEVERETT, Waratah Analytics

“There are a thousand hacking at the branches of evil to one who is striking at the root.”
—Henry David Thoreau

Successful information security requires careful application of tools to goals. The tools must be well used and well configured and require skillful interpretation of results. If problems occur in daily digital forensic workflows, then they must be documented in a systematic manner that aids process improvement.

This article is the third in a series of three articles, where the connection between all the articles is displayed in the “Intelligent Process Lifecycle of Active Cyber Defenders” poster [1]. The poster showcases the main cyber defence disciplines and how, if false-positive events are classified in a systematic manner for each service, this information can afterwards be used to identify improvement areas. It focuses on false positives, error states, and common operation problems that occur when identifying and assessing vulnerabilities, as well as problems that happen when the security operations centre (SOC) tries to add a log source for continuous monitoring. For all these problems it also includes suggestions on how to solve or document them if they cannot be solved. We also demonstrate how to map these practical SOC outcome failures into clear risk management categories. That mapping can be used to optimize workflows, improve strategic cyber risk management, as well as deliver valuable metrics.

CCS Concepts: • Security and privacy ➔ Security requirements; Formal security models; Logic and verification;

Additional Key Words and Phrases: Vulnerability management, threat assessment, SOC, CDC, security operation centre, vulnerability, quality, detection rule

ACM Reference format:
Desiree Sacher-Boldewin and Eireann Leverett. 2022. The Intelligent Process Lifecycle of Active Cyber Defenders. Digit. Threat.: Res. Pract. 3, 3, Article 22 (October 2022), 17 pages.
https://doi.org/10.1145/3499427

1 INTRODUCTION
To design optimal defence, it is essential not only to detect suspicious actions but also to successfully prevent them [2]. Ideally, this should be achieved on the first try, and continually. To make informed decisions, the security and risk managers must have a deep understanding of issues faced by the operating team. Often this

This research was created as a follow-up to “Fingerpointing False Positives: How to Better Integrate Continuous Improvement into Security Monitoring” [3] to better illustrate further challenges often seen when building a SOC. It was presented at FIRST 2020 and the slides [21] and poster [1] were published at that time as well.

Authors’ addresses: D. Sacher-Boldewin, Finanz Informatik GmbH and Co KG; email: desiree.sacher-boldewin@f-i.de; E. Leverett, Waratah Analytics; email: e.leverett@waratah.io.

This work is licensed under a Creative Commons Attribution International 4.0 License.
© 2022 Copyright held by the owner/author(s).
2576-5337/2022/10-ART22
https://doi.org/10.1145/3499427

Digital Threats: Research and Practice, Vol. 3, No. 3, Article 22. Publication date: October 2022.
communication is impaired as the parties involved seem to speak different languages. By documenting the problems faced in a way that can be understood by both sides, this article suggests a way this challenge can be overcome. It is a systematic presentation of issues seen in almost every company running a security operations centre (SOC) or security program. The motivation for this work is based on years of observing miscommunication between operational IT and security or risk teams, as well as too-little clear guidance on how to make security actionable.

The approach chosen in this article is to categorize possible failure states systematically, so they can be added to the existing processes as a feedback option. Regular reports on the vulnerability mitigation and threat assessment activities, including the key performance indicators suggested in this article, will give deep insights into deficiencies of security processes. This is usually only achieved with well-supervised maturity assessments. This article is a successor of the release of the false-positive taxonomy for “security monitoring” [3] and “integrity and security compliance monitoring” false-positives taxonomy [4], both of which already emphasized categorizing the source of the error.

2 MATERIALS AND METHODS

To establish an optimal measuring and monitoring capability, it is important to document what is going wrong in each part of the process. This will address problems at the right part of the process and allow for continual improvement. As cyber defence has become highly critical to a company’s value, security operations processes must be directly connected to risk quantification so that effective and efficient risk management can be accomplished. Risk managers need to have a reflective map on what areas in the company need attention and focus most urgently; therefore, only focusing on risks coming from outside and being blind to inside risk is a critical mistake. Often this can be driven by companies trying to sell their newest products, but it is always recommended to focus on the basics as suggested by experts [5] and most well-known frameworks [6].

The connection between the here-provided taxonomies and the previously published ones is illustrated in the Intelligent Process Lifecycle of Active Cyber Defenders poster [1], shown in Figure 1. We recommend to directly build the possible failure states into the security orchestration workflows and existing tool landscape as a closing note or as a delay reason. It is recommended to as well add a short explanation for each option. Examples can be found in the GitHub with the poster.

In the following pages, we divided vulnerability management [7] into three parts. We enter vulnerability management at the state where a vulnerability was detected and rated as relevant. After this step, usually the question to be asked is “can the prevention measure be activated on time?” Usually, there is a policy dictating how long engineers must install a patch after it was released. If the security measure cannot be installed on time, you want to know the reason (this is where the second set of categories comes into place) and know if you can at least try to detect exploitation if you at least cannot prevent it. This is where the third set of categories come into play.

For identifying how many resolution categories are possible, we calculate the relevant “dimensions of failing states possible” multiplied by “dimensions of involved parties.” For the dimensions related to business processes, this article relies on the four aspects referenced in Information Technology Infrastructure Library (ITIL) v3 as the “4Ps” (people, processes, products, and partners) or in ITIL v4 “the four dimensions of service management” [8] (organisations and people, information and technology, partners and suppliers, and value streams and processes). In recent research, mapping of security incidents has been done, for example, the Center for Internet Security (CIS) Controls [9]. This is an interesting aspect but not helpful to illustrate company internal structures that are in conflict or power games that are disrupting efficiency in the company. Identifying what process or product is causing problems can also be done by correlating failing incident and vulnerability handling with the attribute that are anyway documented (log source, use case, affected software, responsible team). Which aspect is causing the failing state would still not be documented then.

To identify the correct resolution category and illustrate the thinking process that is activated in some way, the decision matrix in Figure 2 can be of help.
3 VULNERABILITY RELEVANCE DETECTION ERROR

A vulnerability becomes a relevant threat to the business when the presence of the vulnerable asset has been established and there is an absence of other mitigating factors. If the responsible engineers reply that the vulnerability is not relevant after all, then you most likely have one of the detection errors described in the following chapters. This is a very critical point, as exploitability has to drive the urgency in which a patch is rolled out. Research has shown that when vulnerabilities were not design choices, successful exploitation still requires that many aspects need to come together to be successful [10]. The decision process for that stage is reflected in Figure 3. For this subsection, we only work with one “dimension of involved parties,” which usually is the system engineers receiving the automatically identified vulnerabilities to rate the local potential impact before patch processes are initiated. Also, as we assume automatic assignment is performed (no manual assignment was done and the responsible people were correctly identified), the only features that can be wrong are products (information and technology), as in misconfigured products, as well as process problems (value streams and processes). If your process used manual assignments or has faulty contact information registered with your assets, then such categories should be added here as well.

3.1 Faulty Vulnerability Verification Setup

3.1.1 Example.

– The vulnerability, automatically identified by a scanner, does not actually exist on the system (classic false positive in the scanner).
– A local system configuration on the system prevents exploitation (special deployment or configuration option exist that prevent or mitigate exploitation).
Fig. 2. Decision matrix for identifying best error or problem state.
3.1.2 **Description/Significance.** This category shows that the identification of vulnerabilities by your vendor/supplier is faulty or should be optimized. If your vulnerabilities are closed several times by the responsible system engineers with this category, then you either want to find a better vendor or improve automated detection by using host patch management verification solutions or combining the alerting with technical security compliance verification or you want to verify your engineer’s judgement capabilities in risk/system assessment. It is always important to track the reasons a vulnerability was judged as “not relevant”; sometimes this might need to happen in a direct conversation. This way you can trust in your process and tools are set up correctly if a cumulation of this error category occurs.

3.1.3 **Benefits.** This category can assist strategic product decisions or illustrate engineer capabilities. If this is often selected, then a reviewing party of the vulnerability management process, most likely security management, should be considering auditing the work flow of the respective engineering team to better identify the actual cause of the error.

3.2 **Context of Exploitability Has Not Been Given**

3.2.1 **Example.**

—The vulnerability is protected by other security measures that were not detected by the vulnerability scanner, such as a web application firewall.
Table 1. Vulnerability Relevance Detection Error Category Perspectives and Follow-up Action

| Case                          | C-Level Perspective | SOC Perspective | MSSP Account Manager Perspective | Follow-up Action                                      |
|-------------------------------|---------------------|-----------------|-----------------------------------|-------------------------------------------------------|
| **Key driver**                |                     |                 |                                   |                                                       |
| Faulty Vulnerability verification setup | No – false alarm     | No – need improvement | No – need improvement            | Verify your tool capabilities                         |
| Context of exploitability has not been given | No – false alarm     | No – need improvement | No – need improvement            | Verify your security documentation of specific context for threat protection measures |

3.2.2 Description/Significance. If vulnerabilities are often graded down by other security measures, the identification process and the technical setup should be verified. Even though all vulnerabilities should eventually be patched, this category can point to risks that were accepted at one stage to avoid further assessments in a previous version. That risk acceptance has not been recomunicated to the scanning team. Do not scan what you never intend to remediate, and document everything you never intend to remediate in a risk entry.

3.2.3 Benefits. This category can directly illustrate technical setup improvement potential and occupied resources to handle misconfigurations. Misconfigurations can lead to boredom and alert fatigue with system engineers, as they are less likely to take patching and reacting to threats seriously. Often system engineer’s highest priority is the stability of their systems, installing patches is an “out of order” action that can endanger their current setup. Trying to enforce changes on them that are not as critical, as they have created security measures that are not visible in your scanning action, will lower their willingness to cooperate with security responsible in actual emergency cases. This category can also be a very valuable key performance indicators (KPI), as a high number can point toward a bad technical threat identification process or setup, or document lack of awareness in the engineering team about actual threats.

3.3 Results
These categories can help illustrate systematic problems in the vulnerability assessment process or setup that should be addressed to improve overall quality of vulnerability handling. They can help the SOC in being a valuable partner to the engineers by not burdening them with unnecessary extra work, only triggering work where needed. Table 1 includes an overview of the identified possible error states, as well as the interpretation from the different perspectives, where C-Level is chief executive officer or chief security officer, SOC perspective is the view of the SOC team lead, and Manager Security Service Provider (MSSP) reflects the view of a party offering SOC services to other companies and is therefore following defined Service Level Agreements (SLA).

4 VULNERABILITY REMEDIATION DELAY REASON
If a vulnerability cannot be remediated on time, then it is usually caused by different types of problems that should also be tracked as “security measure implementation failure.” The amount of time assigned for fixing a vulnerability is usually dictated by the risk appetite of the company and how critical exploitability of a vulnerability is estimated. This depends on your company’s risk profile, risk capacity, what priorities are set for corporate values, and what stakeholders are also willing to support [11]. Such values might not always be transparent to the system engineers, as their work is usually dictated by service level agreements and business service criticality. The difference between those delay states and why each of them has its validity is listed in the following. Figure 4 enumerates the different states possible as presented in Reference [1].
Ideally, all vulnerabilities in assets under protection should be fixed. In practice, we would settle for only patching what will eventually be exploited. Of course, since we do not know which vulnerabilities or assets these will be, we often have to settle for patching everything. If this cannot be done in the requested time, then operational teams usually have reason to avoid directly resolving the case. Reasons for such delays should be tracked systematically to illustrate where company incentives can be optimized.

For this subsection, we only work with one “dimension of involved parties,” which usually are the system engineers receiving the automatically identified vulnerabilities to rate the local affectedness before patch processes are initiated. We have one category for each “dimension of failing states possible”: a “resource problem,” reflecting people (or organisations and people), a “compatibility problem,” referencing product problems (or information and technology); “bad service level agreement (SLA),” relating to process problems (or value streams and processes); and “support problems,” illustrating issues with assistance by partners (or partners and suppliers).

4.1 Resource Problem

4.1.1 Example.
— High workload in the team or people shortages will not allow prevention on time.

4.1.2 Description/Significance. Low staffing, too many projects, or different priorities communicated to the team can lead to too few engineers being able to properly test and roll out the needed updates on time.

4.1.3 Benefits. This category can illustrate the outcome of too-low staffing that might not become obvious in other areas. Resource problems can also illustrate challenging change windows as extra staffing at night will lead to fewer people available during business hours to meet health regulations. As installing updates for many products has become a monthly activity, not being able to schedule this into the task list can seriously question the resilience of the team. This should be treated as an urgent warning sign and illustrates the need for better governance. If this is documented over several teams in your company, then your management decisions are most likely challenging the overall company resilience and sturdiness. This can be a deliberate decision but should be
alerting risk managers and stakeholders. A valuable KPI can be “number of delays due to resource problems” or “average number of days delays due to resource problems,” as this might have impact on your existing contracts with customers.

4.2 Compatibility Problem

4.2.1 Example.

—Installing the patch will break a current product setup that depends on the software that needs to be updated.

4.2.2 Description/Significance. Business products or solutions can rely on fixed dependencies or tight setups that break when installing an update. This can, for example, happen when a company relies on a product that has long stopped being supported by the vendor or skills to advance your product have left the company.

4.2.3 Benefits. This is helpful for risk management to identify where product lifecycle management should have identified follow-up solutions and have not become active yet. If the product has been officially decided to be taken out of service, then it may be tolerated but should always be documented in the risk register.

4.3 Bad SLA

4.3.1 Example.

—Installation of updates is only allowed a few times a year, as the devices or the services depending on them are facing such a high demand or are sold as such.

4.3.2 Description/Significance. High SLA KPIs for products, bad service design, bad service management monitoring, or combinations of these elements can lead to teams not being allowed to install patches on time. This can appear in change window planning with a team only receiving change windows once a month or less but having patching times of 21 days or less.

4.3.3 Benefits. This challenge can easily happen with services with many dependencies, like, for example, network and security devices. Often this is not made aware to decision authorities unless directly questioned by the team leadership. If this challenge is often preventing patches from being installed on time, then projects should be initiated to revise the distribution of change windows, high availability design layouts of the affected products, or the solutions for measuring compliance with SLAs. A valuable KPI can be “Number of delays due to unreasonable SLA,” evaluated in correlation to application. This should be used to influence your service level design.

4.4 Support Problem

4.4.1 Example.

—The vendor does not supply a patch for the vulnerability of the needed software version or adaption.

4.4.2 Description/Significance. The specific product version installed relies on a software component of another product (for example, open source) that has been fixed in the original version but not yet been updated by vendor you use your product version from. Documenting incurred risks due to bad vendor support should influence strategic choices for partnerships, even if they cannot be fixed short term.

4.4.3 Benefits. This occurrence should always remain close to zero. Within information security, supply chain problems have become a more important topic. It is important to not only focus such discussions on the secure setup of the supplier but also take their capability to deliver security fixes on time into examination.
Table 2. Vulnerability Remediation Delay Reason Category Perspectives and Follow-up Action

| Case         | C-Level Perspective                                                                 | SOC Perspective                                      | MSSP Account Manager Perspective                      | Follow-up Action                                      |
|--------------|-----------------------------------------------------------------------------------|------------------------------------------------------|--------------------------------------------------------|-------------------------------------------------------|
| Key driver   | Does this alert inform me about an actual threat to the company?                   | Are prevention capabilities working correctly?        | Were the MSSP service systems configured correctly to prevent a threat to my company/customer? | What lesson can be learned from this event?            |
| Resource problem | Yes – Governance problem                                                            | No – patch not installed                             | No – risk was not mitigated                            | Review and if reasonable report risk for staffing/budget priorities |
| Compatibility problem | Yes – Governance problem                                                            | No – patch not installed                             | No – risk was not mitigated                            | Review and if reasonable report risk for your product dependencies |
| Bad SLA      | Yes – Governance problem                                                            | No – patch not installed                             | No – risk was not mitigated                            | Review and if reasonable update SLA/SLA dependencies or report risk |
| Support problem | Yes – Governance problem                                                            | No – patch not installed                             | No – risk was not mitigated                            | Review and if reasonable verify partner dependencies  |

4.5 Results

The described values describe everyday challenges operating teams face when covenanting “providing a secure service” with “high availability.” All these values are important for strategic risk and governing decisions and should not be taken lightly by team managers. For good process management in your company, tracking these problems systematically will help in creating a culture of support and understanding for security measures.

Table 2 includes an overview of the identified possible error states, as well as the interpretation from the different perspectives, where C-Level is chief executive officer or chief security officer, SOC perspective is the view of the SOC team lead, and Manager Security Service Provider (MSSP) reflects the view of a party offering SOC services to other companies and is therefore following defined Service Level Agreements (SLA). It shows that from top management the outcome looks the same for each type of alert, but the treatment, reflected in the lesson learned, is different.

5 DETECTION FAILURE

When a vulnerability cannot be closed, the minimal fallback strategy is to try to detect exploitation. As detection is not always possible, also caused by several different kind of problems, the detection failure categories are listed and explained in the following. The state “Blind spot identified” is reached as also illustrated in Figure 5. The “dimension of failing states possible” here are (a) SIEM use cases cannot be created technically, (b) SIEM use cases cannot be created at this time, (c) log events are either not created or cannot be delivered, and (d) no governance stakeholder was identified for this kind of threat. The “dimensions of involved parties” are the SOC as the team trying to identify what use cases should be created and the “other” team, most likely engineers, trying to create the logs or security architects, evaluating risk and demand. If you are an MSSP and want to track this dimension more granularly, then it can make sense to add further categories based on your need. The project DeTT&CT [12] has made a great effort in mapping log source availability to the MITRE ATT&CK [13] framework. It focuses on illustrating overall visibility into the logs that are there and how trustworthy the log source is in general. If a log cannot be connected, then it does not illustrate though why this is the case. If the question is what kind of logs you want to collect at all, then other publications have already gone into great detail and provided suggestions [14]. These categories therefore do not help you in identifying what the most critical log types are for your infrastructure—they only help in illustrating why you are having the blind spots you find in your infrastructure so you can make informed decisions about correcting those blind spots. This is currently one of the biggest challenges SOCs face [15], and a methodology to efficiently deal with error states is still sought-after.
Figure 5 displays the decisions to be taken when implementation of detection rules is considered. On the left side, the actionable steps are listed, where your company can decide what the best treatment for this case is depending on the threat. On the right side, possible reasons for why a detection cannot be created are listed. Any of those failure states are concluded as a "Blind spot identified."

5.1 Events Cannot Be Logged

5.1.1 Example.

—It is technically not possible to log the exploitation of a vulnerability.

5.1.2 Description/Significance. It can happen that some vulnerabilities are hard to log directly. This can, for example, occur when information about a backdoor gets published and the usage of such a backdoor is done in a fashion that is hardly distinguishable from normal usage. In this case, it is a good idea to find a close-enough approximation by identifying what might be visible before or after exploitation occurred. This can trigger more false positives; implementation therefore should be based on risk levels and decisions and assumptions reached documented in a technical risk register.
5.1.3 Benefits. This can help the SOC in prioritizing what rules are more critical to implement. Often management is not aware of how much effort goes into baselining a new SIEM use case. Considering the amount of false positives that can be expected from a rule should be therefore documented as early in an implementation process as possible.

5.2 Detection Pattern Unclear

5.2.1 Example.

—It is not clear at the moment of evaluation how a SIEM use case should be implemented.

5.2.2 Description/Significance. It can happen in an early stage of publication that it is not well documented how a vulnerability is being exploited. This can especially happen with responsible disclosure where a patch was created by the supplier but cannot be installed due to one of the variety of reasons in the previous chapter. It can mean that for creating such a rule, contact to the supplier needs to be initiated in hope of receiving more information. It is possible that the information needed will not be forwarded for legal reasons. This situation can be very tricky, and ways to resolve this challenge need to be evaluated individually. Usually, the risk of the delayed patch installation needs to be raised and implementation should be prioritized if somehow possible. If this is still not possible, then projects to initiate architectural changes to the surrounding setup should be considered. Another cause of this category is analysts being unable to produce detection rules from blog posts about the vulnerability. This encompasses many subcategories ranging from lacking foundational skills in the analyst or inaccuracies in the documentation of the vulnerability and its exploitation. There are many resources available with ready-made detection rules, but understanding the implemented detection rules and, if needed, adjusting them for the individual usage is key. If this is often the cause of a rule not being implemented, then investing in analysts’ skills and sending them to trainings will be essential.

5.2.3 Benefits. Interpretation of this category should always be done by the SOC manager, who most likely can judge the skills of the analysts best. Depending on if lack of skills or lack of information published is identified as a source, priorities should be set toward creating training possibilities or creating a risk entry. A risk entry could cover failure of detection capabilities for exploitation of known and “fixed by vendor” vulnerabilities.

5.3 Log/Event Ratio Unreasonable

5.3.1 Example.

—The amount of logs created to detect suspicious event is not economical.

5.3.2 Description/Significance. For some types of events, it is often argued that to create and store such logs the costs are too high. SOC needs are here valued below operational needs and should be tracked and recorded as such. This is often argued for Domain Name System (DNS), NetFlow logs, or logs created and stored on virtualized endpoints for analysis. These decisions should be documented for each use case, and a risk entry should be created for documenting “failure of detection capabilities as log/event ratio not economical.”

5.3.3 Benefits. Documenting how many use cases failed to be implemented based on this category can help when consolidated and re-evaluated after some time. Eventually, the risk of all the events not detected, related to vulnerabilities by products affected, can be summarised to be considered in further maintaining the specific vulnerable setup.

5.4 Resource Problem Detection

5.4.1 Example.
The SOC does not have enough resources to create the needed detection rule.

5.4.2 Description/Significance. This can happen when the team is understaffed (chronic problem) or the workload is particularly high temporarily (acute problem). Creation of new rules often has less priority in a SOC, as handling daily incidents and vulnerabilities is more critical for the short-term perception by the outside. As having an updated set of use cases is critical to not only identify current threats to the company but also employee motivation, this should never be down prioritized for long periods.

5.4.3 Benefits. This should be an alarm bell for SOC managers who cannot manage their team closely for various reasons. If this happens for several months in a row, then distribution of roles should be revised, and if nothing can be optimized, then headcount should be increased.

5.5 Logs Can Not be Delivered

5.5.1 Example.

—Logs need to be created on a customer site and transmission is technically not possible.

5.5.2 Description/Significance. This information is usually communicated by the respective engineering team. It can happen when logs are not created on site or when the network architecture of your company does not allow the delivery of the needed logs to the SIEM. This problem cannot be resolved by the SOC; rather, it should be verified by network architects and documented with a risk entry until the delivery situation can be resolved.

5.5.3 Benefits. This category can be an early warning sign for governance teams and IT security or network architects to identify possible potential for improvement. If this happens for many use cases, then either reconstruction should be prioritized or judgement by the respective engineering team should be reviewed.

5.6 Resource Problem Delivery

5.6.1 Example.

—The logs could be created by the respective engineering team, but they don’t have resources to make the adjustments.

5.6.2 Description/Significance. Specific events require specific log settings to be able to detect right actions. If the respective engineering team does not have the resources or different priorities to create the adjustment, then it is hard for the SOC to enforce the change. This type of category can point to either heavy workload of the engineering team or lack of official support for capabilities the SOC needs to do its job. This category therefore is a valuable metric for documenting the dependency of the SOC on the other teams, and, combined with the information where this occurs most often, measures should be taken to improve the situation. Until the situation is resolved, the incapability to create detection rules based on this dependency should be documented in a risk entry.

5.6.3 Benefits. Documenting this category separately is key in illustrating the effectiveness of the SOC. In all the areas where the needed events cannot even be configured on the end systems, the best use cases in the SIEM will never be of any use. This is a major risk and often forgotten by people who first build up an SOC. Resource problems are always a sign of different priorities, based on the resources available. It therefore can be a container category for all kind of problems related to non-technical problems when configuration adjustments of other parties are needed. This category should be understood as such and especially in politically challenging environments, assumptions in the SOC on why a requested change cannot be implemented should be avoided to not endanger further collaboration.

Digital Threats: Research and Practice, Vol. 3, No. 3, Article 22. Publication date: October 2022.
5.7 No Tool for Logging

5.7.1 Example.

— A tool for documenting the needed events is currently not existing.

5.7.2 Description/Significance. This is most often a problem for newly built SOCs. This can happen if they for, example, would rely on process creation events, as usually created by Sysmon or an EDR solution, but such tools have not been enforced yet by security management or accepted be the respective engineering team. The SOC heavily relies on support by the surrounding teams and lack of detection logs should result in a documented risk entry for acceptance on invisibility on the respective systems. Security management and security architects should evaluate the risk, and if re-evaluation results in needed changes, then support for extending the log coverage should be enforced.

5.7.3 Benefits. This category illustrates again a dependency of the SOC on the already existing infrastructure. The more responsibility on creating such tooling is set in the SOC, the more resources and capabilities need to be considered in the build-up process. It can also happen that lack of the availability of specific capability is an accepted risk for economic reasons; documenting the number of rules not being created systematically can therefore allow a more fact-based risk assessment.

5.8 No Process/Demand/Responsible for Detection

5.8.1 Example.

— The vulnerability scan has found a vulnerable system but no responsible person or team can be identified for it, and therefore no patch can be installed and no party to support detection of an active threat is available.
— The ATT&CK technique is evaluated as critical based on relevant APT groups but no responsible team for creation of logs can be identified.

5.8.2 Description/Significance. This can happen if systems or applications are not built based on official policies, inventory is incomplete, or a loose governance is implemented throughout the company. This is a challenge SOCs often face and can result in excessive amount of time used in trying to identify responsible people. In vulnerability management, where the focus should be set to avoid a critical event or lower the risk of entering into one, it is easy to detect documentation gaps and hard to react to them, because no risk-based vulnerability assessment can actually be performed. For risk-based vulnerability management to be successful, information for every system should be kept on not only what type of system it is, what is running on it, and who is responsible for it, but the whole surrounding setup has to be considered to create an actual risk assessment. To produce this regularly at scale, information about what protection measures are in place to support any software or system inquired, as well as what attack vectors these protection measure consider and where the existing measures currently have their gaps, has to be documented and be made available during the use case assessment process. The gaps identified by the SOC should be put into fix process, and information about missing stakeholders should be forwarded for clarifying the actual need of the implementation of such a detection rule. If no responsible can be found and the system still cannot be removed from the network, due to political reasons (for example), then a risk entry should at least document knowledge of such a device.

5.8.3 Benefits. Creation of a separate category can assist in implementing automated processes in the SOAR platform to direct unresolvable threat or vulnerability assessments to a role outside of the SOC, so SOC employees can focus on implementing and improving the existing and processible tasks at hand.
Table 3. Detection Failure Reason Category Perspectives and Follow-up Action

| Case                          | C-Level Perspective | SOC Perspective | MSSP Account Manager Perspective | Follow-up Action                                      |
|-------------------------------|---------------------|----------------|----------------------------------|------------------------------------------------------|
| **Key driver**                |                     |                |                                  |                                                      |
| Events cannot be logged       | Yes – blind spot identified | No – blind spot identified | No – blind spot identified | Report risk, find a close enough approximation |
| Detection pattern unclear     | Yes – blind spot identified | No – blind spot identified | No – blind spot identified | Report risk, schedule education for SOC engineers |
| Log/event ratio unreasonable  | Yes – blind spot identified | No – blind spot identified | No – blind spot identified | Report risk, verify economic alternatives |
| Resource problem detection    | Yes – needs attention | No – improvement needed | No – SLA warning | Report risk, verify priorities and staffing within SOC |
| Logs cannot be delivered      | Yes – blind spot identified | No – blind spot identified | No – blind spot identified | Report risk, review architecture for possible log transmission |
| Resource problem delivery     | Yes – needs attention | No – out of SOC control | No – obligation to cooperate neglected | Report risk, verify priorities and staffing with security engineers |
| No tool for logging           | No – accepted risk | No – out of SOC control | No – obligation to cooperate neglected | Report risk, security management/IT governance verification |
| No process/ demand/ responsible for detection | No – accepted risk | No – out of SOC control | No – obligation to cooperate neglected | Report risk, security management/IT governance verification |

5.9 Results

The here described categories can help in identifying follow-up actions when a detection rule cannot be created. Sources of failure in rule creation can differ greatly and are often not to blame on the SOC capabilities. Creating a supporting infrastructure in a company is key for a successful SOC.

Table 3 includes an overview of the identified possible error states, as well as the interpretation from the the different perspectives, where C-Level is chief executive officer or chief security officer, SOC perspective is the view of the SOC team lead, and Manager Security Service Provider (MSSP) reflects the view of a party offering SOC services to other companies and is therefore following defined Service Level Agreements (SLA).

6 RESULTS AND IMPACT FOR RISK MANAGEMENT

The successful implementation of SOC services heavily relies on contributing teams by using existing configuration systems and good-quality inventory and asset management. Too often the related risks are not communicated outside of the SOC, because the surrounding teams are not ready to support the need of dealing with all-time new threats. Processes and relationships therefore need to be established to handle such gaps by prioritizing needed adjustments on the right level and with correct capabilities. An efficient transportation of identified gaps will make risk management more agile and more aware of technical threats to the company, not necessarily covered by other detection tools.

For SOC teams just beginning the journey toward mapping outcomes to risk management, it is worth outlining some old and trusted principles of risk management. Essentially, risk management comes down to four strategies: avoid, accept, mitigate, or transfer [16]. Of course, you can also employ more than one of these strategies, and they all apply both before and after a risk is actualised [17]. Some of them are only possible in part or in theory, depending on the risk in question [18]. So a risk manager must carefully layer these strategies to achieve their goals [19]. What does that mean to the practising SOC analyst?
First, you have had the security and patching risk transferred to your team. Even an internal SOC needs to be careful in how much risk they accept when comes to them. For example, they can accept the risk of failure to detect a vulnerability, because they are in control of that process. However, they should be careful not to accept the risk of not patching their systems. That belongs to the team who owns that asset, and both teams need to document the transfer or acceptance of risk to make that clear.

Mitigation is an approach that is common in infosec circles, and it is usually focused on preventative actions such as patching the vulnerability. However, it is also possible to do other types of mitigation, such as reductive and restorative—in other words, let it get hacked but keep it in a sandbox or let it get hacked and replace the VM to restore previous state and break persistence. Another useful organising principle is deciding if the risk is within your control (as a team or as an organisation) or to outwit your control. This refers to the risk itself and not the mitigations, so, for example, a misconfiguration in a vulnerability scanner might be endogenous, but a cloud outage is exogenous.

With these basic concepts of risk management, it is possible to classify each outcome of the issue review to see how it arrived in your team and how you can manage it as a risk. So even for outcome categories we have not identified here, you can continue to map tactical results to strategic risk management: avoid, accept, mitigate, or transfer (exogenous or endogenous? reductive or restorative?).

With just this simple approach, it should be possible to not only improve the quality of setup and configurations in place but to also transport the SOC outcomes into manageable cyber risks any board can discuss. Thus, we hope that the language barrier of operational cyber risk management and the SOC teams is reduced.

7 DISCUSSION
Vulnerability and threat assessment literature is often written with the assumption of success. Reasons for failure are usually not discussed and especially not systematically documented. By focusing only on how things should be done, little attention is put on what should be done when the expected path fails. As recent research shows [20], failure is more common than it is officially talked about. Documenting states of failure systematically can help in bringing visibility into hidden risks and therefore should influence risk management strategic decisions. Vulnerability relevance, detection errors, and vulnerability remediation delays should be documented by the system engineers. Detection failure states should be documented by the SOC when evaluating a relevant use case for implementation. The processes for these activities do not need to be connected directly; for best documentation of a risk-based coverage it is, however, a recommended activity.

The goal of such rigorous categorisation is to map the mitigation outcome results to business risk management practices. Some risks are endogenous, like when staff are poorly trained and make configuration errors of vulnerability scanners. Other risks are exogenous, such as when patch documentation does not reflect real and crucial changes to the operating environment, or when the SLA agreed by the sales team is unachievable. This can easily happen in an externally facing SOC when the sales team agree to patching rates without securing resources from the team who must apply the patches. Thus, contract risk is related to cyber risk and in turn becomes a transitive risk.

The management of cyber risk is itself a risky business, and one where no mistake can be wasted. For if a cyber risk management business cannot learn from its strategic mistakes, then why should any customer trust them to manage their cyber risk?

The biggest game in cyber risk management is liability shifting. It happens both within and between organisations. The sales team blames the IT team for not patching according to their SLA, and the IT team blames the operations team for not scheduling maintenance. The customer blames the vendor for slow patches, and the vendor blames the researcher for finding the vulnerability. In particular, in that last case, the vendor feels liable when informed by the security researcher but shifts the liability to the customer by producing the patch, regardless of whether that patch is installed. In turn, that customer may shift the liability to an external SOC,
and the management may shift that liability down to the SOC team unless a careful process of documentation and risk management can be performed.

Instead, our organisations and teams should seek to share risks and rewards of cyber risk management. If they can, then the work will become lighter and more efficient, and the risks can be reduced effectively and in a timely manner.

8 CONCLUSION

Within this article, we showed not only how daily SOC problems can be resolved but also that resolving is an inter-party responsibility. Communicating those problems transparently toward risk management also creates more effective SOC operations. The cyber risk problems usually tracked with security incidents and not-installed patches only scratch the surface and merely show the tip of the iceberg of suboptimal processes and general communication problems. They are usually produced by staffing, resourcing, scheduling, as well as prioritization issues or additionally by poorly considered contracts and counter party risks. Mapping from a cyber risk back to an organizational root cause allows risk management teams less familiar with cyber to manage parts of it effectively. It engages the board and transforms deadlocked arguments into actionable items.

A collection of KPIs is actively maintained in the project GitHub [22]. The KPIs not only include target values and reference the owner of the risk but also link to business impact and include a supporting case study. Finding the owner of a risk is a frequently cited cyber risk problem, but inter-party shared ownership is often achieved by mapping the impact to the business.

Traditional risk management treatments and techniques are relevant to cyber, and, in this article, applying them to a SOC has exhibited how effective dialogue can be produced between technical cyber teams and technical risk teams. Aligning the reports by measuring values translated to the other party will further enhance communication.

In times where it has become easy to buy indicators of compromises from threat intelligence providers, it is predominant to look for risks on the outside. This external intelligence though can only be efficiently applied if internal processes and tools are working in an optimal way, in short, if the security organisation has reached a mature efficiency or risk identification, communication, and then mitigation.

Continued process improvements as described here will help reaching the required maturity without needing additional tools or more external services first by creating internal intelligence. Essentially, we must learn to explain the cyber risks and business impacts that occur from non-cyber decisions. If we do not, then we will continue to be under-resourced, over scheduled, poorly prioritised, and getting blamed for contracts that were made without understanding the resourcing problems at play. To paraphrase Thoreau: We want to stop hacking at the branches and strike instead at the root cause.

ACKNOWLEDGMENTS

Thank you to Felix "FX" Lindner for reviewing this article. Thank you as well to FIRST for sponsoring the publication of this article and the peer-review feedback by ACM DTRAP.

REFERENCES

[1] Desiree Sacher. 2020. The intelligent process lifecycle of active cyber defenders poster. Retrieved from https://github.com/d3sre/IntelligentProcessLifecycle/blob/main/Poster_IntelligentProcessLifecycle_V1.1.pdf.

[2] Eric M. Hutchins, Michael J. Cloppert, and Rohan M. Amin. 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. https://www.lockheedmartin.com/content/dam/lockheed-martin/rms/documents/cyber/LM-White-Paper-Intel-Driven-Defense.pdf.

[3] Desiree Sacher. 2020. Fingerpointing false positives: How to better integrate continuous improvement into security monitoring. Digit. Threats: Res. Pract. 1, 1, Article 7 (March 2020), 7 pages. DOI: https://doi.org/10.1145/3370084
[4] Desiree Sacher. 2020. How to better integrate continuous improvement in security configuration compliance. Retrieved from https://github.com/d3sre/IntelligentProcessLifecycle/blob/main/ComplianceIntegrityTaxonomyPaper-v1.0.pdf.

[5] Florian Roth. 2017. The distorted risk landscape. Retrieved from https://cyb3rops.medium.com/the-distorted-risk-landscape-e83041eeea2d.

[6] National Institute of Standards and Technology. 2018. Framework for Improving Critical Infrastructure Cybersecurity. Retrieved from https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf.

[7] Murugiah Souppaya and Karen Scarfone. 2013. Guide to enterprise patch management technologies. Retrieved from https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-40r3.pdf.

[8] Martin Stewart. 2019. What are the four dimensions of ITIL 4? Retrieved from https://info.axiosystems.com/blog/what-are-the-four-dimensions-of-itil-4.

[9] L. de Castro, A. W. Lo, T. Reynolds, F. Susan, V. Vaikuntanathan, D. Weitzner, and N. Zhang. 2020. SCRAM: A platform for securely measuring cyber risk. Harv. Data Sci. Rev. https://doi.org/10.1162/99608f92

[10] T. Dullien. 2020. Weird machines, exploitability, and provable unexploitability. IEEE Trans. Emerg. Top. Comput. 8, 2 (1 April-June 2020), 391–403. DOI: 10.1109/TETC.2017.2785299. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8226852

[11] PriceWaterhouseCoopers Board oversight of risk. 2014. Defining risk appetite in plain english. Retrieved from https://www.pwc.com/us/en/corporate-governance/publications/assets/pwc-risk-appetite-management.pdf.

[12] Marcus Bakker and Ruben Bouman. 2019. Detect Tactics, Techniques & Combat Threats. Retrieved from https://github.com/rabobank-cdc/DeTTECT.

[13] MITRE ATT&CK. Retrieved from https://attack.mitre.org/.

[14] Anton Chuvakin. 2020. Detection Coverage and Detection-in-depth. Retrieved from https://medium.com/anton-on-security/detection-coverage-and-detection-in-depth-16137e6c203b.

[15] Anton Chuvakin. 2021. SOC Threat Coverage Analysis—Why/How? Retrieved from https://medium.com/anton-on-security/soc-threat-coverage-analysis-why-how-24055307528f.

[16] Douglas W. Hubbard. 2012. The Failure of Risk Management: Why It’s Broken and How to Fix It. Retrieved from https://onlinelibrary.wiley.com/doi/book/10.1002/9781119198536.

[17] Andrew Coburn, Eireann Leverett, and Gordon Woo. 2018. Solving cyber risk: Protecting your company and society. Retrieved from https://www.wiley.com/en-us/Solving+Cyber+Risk%3A+Protecting+Your+Company+and+Society-p-9781119490920.

[18] M.-E. Cornell, Marshall Kuypers, Matthew Smith, and Philip Keller. 2018. Cyber risk management for critical infrastructure: A risk analysis model and three case studies. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/risa.12844.

[19] Keyun Ruan. 2017. Introducing cybernomics: A unifying economic framework for measuring cyber risk. Comput. Secur. 65 (2017) 77–89 0167–4048, https://doi.org/10.1016/j.cose.2016.10.009.

[20] Yair Manor. 2021. Quantifying the Gap between Perceived Security and Comprehensive MitRE ATT&CK Coverage. Retrieved from https://github.com/3dsre/IntelligentProcessLifecycle/blob/main/MITRE ATT&CK Coverage report.

[21] Desiree Sacher and Eireann Leverett. 2020. The intelligent process lifecycle of active cyber defenders presentation. Retrieved from https://github.com/d3sre/IntelligentProcessLifecycle/blob/main/FirstIntelligentProcessLifecycle-FINAL.pdf.

[22] Desiree Sacher and Eireann Leverett. 2021. The intelligent process lifecycle of active cyber defenders continuous improvement metrics. Retrieved from https://github.com/d3sre/IntelligentProcessLifecycle/blob/main/README.md.

Received 21 February 2021; revised 24 July 2021; accepted 11 November 2021