How to Develop and Maintain Security Monitoring Use Cases

Published 9 April 2020 - ID G00464715 - 63 min read

By Analysts Augusto Barros

Initiatives:Security Operations for Technical Professionals

Use cases are a key part of security monitoring activity. A structured process to identify and implement use cases helps security and risk management technical professionals align monitoring efforts to security strategy, choose best-fit solutions and maximize the value of security monitoring tools.

Overview

Key Findings

- Use cases can be created from three different vectors: threat detection, control and asset-oriented.
- Monitoring use cases are generally implemented as security information and event management (SIEM) content, but can also be implemented with other technologies, including network traffic analysis (NTA) and endpoint detection and response (EDR).
- Some organizations create too much process overhead around use cases — agility and predictability are required. Processes must not be too complex because security monitoring requires fast and constant changes to align with evolving threats.
- Using Vendor-provided use case content is a workable starting point to monitoring, but ongoing tuning is required to ensure the relevancy and effectiveness of those use cases.

Recommendations

Security and risk management technical professionals focused on security operations should:

- Make use case development similar to agile software development by being able to quickly implement or modify a use case to adapt to changing threat and business conditions.

- Prioritize use cases based not only on their importance according to risk, but also on implementation feasibility.
- Select use cases from sources beyond those provided by tools vendors. They are an important starting point, but not the only source of potential use cases.
- Implement a process to frequently review, tune and eventually retire use cases to adjust to changes to the IT environment, business and threat landscape.

Problem Statement

Organizations perform security monitoring with many different tools, such as SIEM, NTA, EDR and data loss prevention (DLP). One of the important characteristics of broad-scope security monitoring technologies, such as SIEM, is a distinction between the tool itself and the “content.” In the case of a SIEM, the term “content,” or “use case content,” is typically used as shorthand for all reports, alerts, correlation and baselining rules and other data inserted into the tool after the tool itself has been developed. Content can be extended to the required settings to properly generate and collect the required data to be analyzed. It can even include nontechnology components, such as security operations center (SOC) playbooks describing how to react to the alerts generated by the technology.

The content deployed on security monitoring tools is driven by use cases. For this document, the term “use case” is used as a specific set of conditions or events, usually related to a specific threat, to be detected or reported by the security tool. An example of that would be, “Identify account compromise by tracking concurrent authentication events.” The need for a use case development process often starts from a startling realization: “We have all this data! What can we do with it?” Lack of clear use cases leaves organizations with a lot of data and no visibility.

> **The efficiency and effectiveness of security monitoring are directly related to the appropriate implementation and optimization of the right use cases on the right security monitoring tools.**

Organizations performing security monitoring need to implement the right processes to adequately identify, prioritize, implement, tune and eventually expire security monitoring use cases. Those processes need to be constantly measured to provide feedback on what needs to be changed or added to the existing technologies, procedures and teams involved in security monitoring.

The Gartner Approach

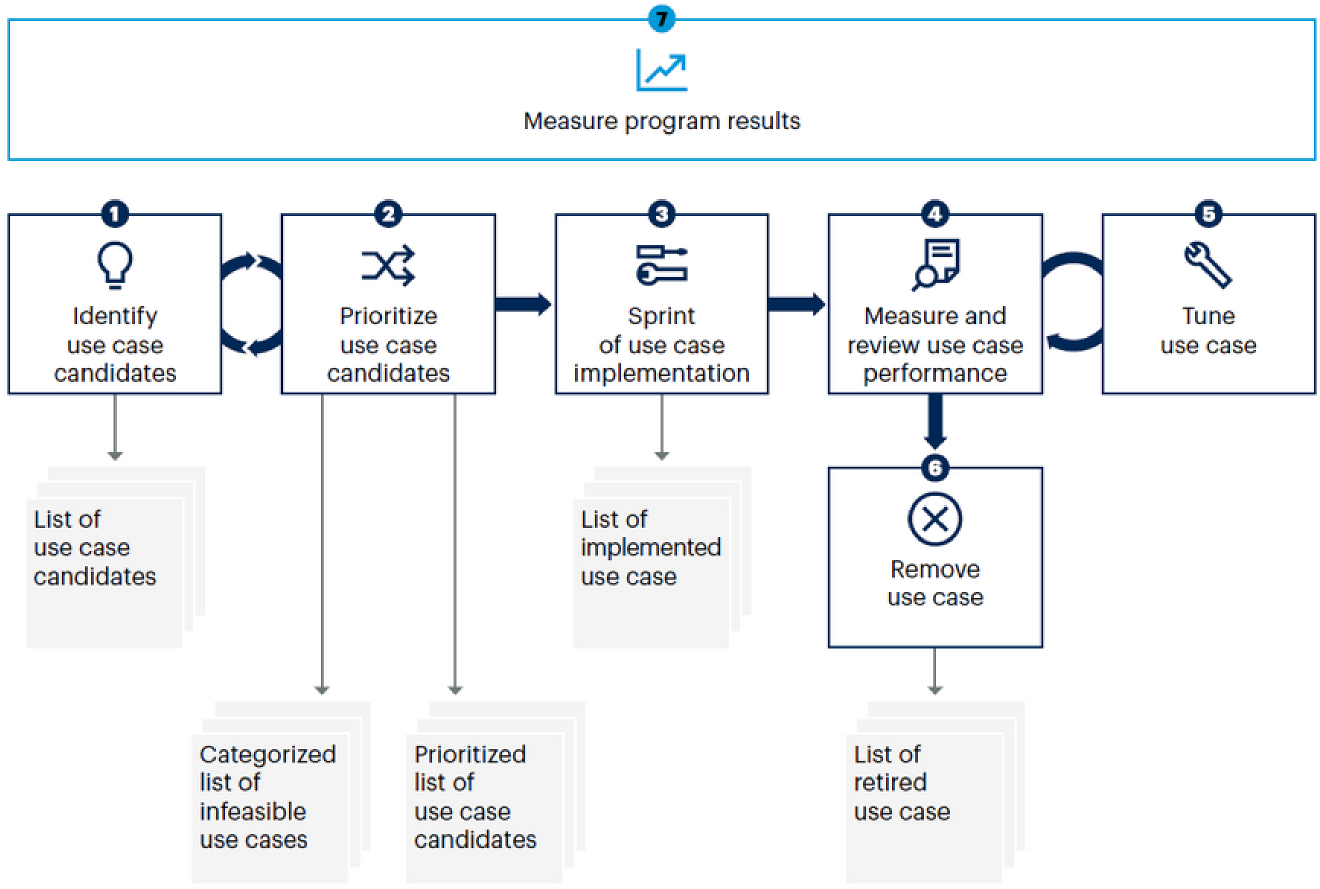
An organization deploying security monitoring use cases should be aware of the critical role content plays in security operations success. It should also be prepared to define and refine its own use case content, such as custom rules, reports and dashboards. Custom content significantly increases the value of security monitoring tools. Even organizations relying on managed security services (MSS) should ensure the service provider has a process to develop and maintain the use cases operated on their behalf (see "How to Work With an MSSP to Improve Security"). The approach described in this research aims to ensure that all monitoring activities and tools are viewed as parts of a single coordinated approach, evolving according to the needs of the organization.

The Guidance Framework

The security use case management framework summarized in Figure 1 is an iterative process, inspired by agile development practices, that cycles through activities to identify, prioritize, implement and manage security monitoring use cases. The basic concept behind the framework is that security teams are constantly identifying security monitoring requirements that drive the implementation of new, or the modification of existing, monitoring use cases. Each use case addresses a specific security monitoring need, has its own life cycle and requires constant review and tuning. Use cases that are no longer valuable should be removed. All of this activity is constantly being measured by established metrics and by maintaining lists of use cases at different points of the life cycle, thus feeding into the strategic and roadmap plans for security monitoring capabilities.

Figure 1. The Security Use Case Management Framework

The Security Use Case Management Framework


7
Measure program results
1
Identify use case candidates
2
Prioritize use case candidates
3
Sprint of use case implementation
4
Measure and review use case performance
5
Tune use case
6
Remove use case
List of use case candidates
Categorized list of infeasible use cases
Prioritized list of use case candidates
List of implemented use case
List of retired use case

Source: Gartner

464715_C

Prework

There is no point in discussing the implementation and management of security monitoring use cases when tools or resources for that activity are not available. Under ideal conditions, the implementation of monitoring tools, teams and processes would occur together with the implementation of this use case management framework. However, it is expected that most organizations will already have monitoring tools and processes in place before deciding to implement the processes described in this framework.

This framework does not require specific tools to be implemented for developing and maintaining use cases. Most organizations use common tools such as spreadsheets and general-purpose content management systems. Organizations looking for more control over the process and documentation can use development workflow tools such as Pivotal Tracker or Atlassian's Jira Software, and collaboration tools such as Jupyter Notebooks. They can also use tool-specific content management features; some SIEM tools, for example, include a content authoring environment. A few vendors are attempting to build security-specific content management tools as well.

Identify Use Case Candidates

Identification of the right use cases is the initial part of the use case development process. "Identify" is used here to reinforce that, at this step, you only need to identify the need for a specific use case.

The identification stage helps:

- Find the problems that are best solved by means of different security monitoring tools such as SIEM; use risk and threat assessments, as well as business unit requirements, as guidance.
- Identify externally mandated monitoring requirements that are most relevant to the organization, such as those from compliance and audit mandates.
- Start the process of converting vaguely defined business and security problems into SIEM content, UEBA models and algorithms, DLP policies and so on.
- Identify the prerequisites for implementing the use cases.

Gathering the problems to be solved is easier than identifying the most relevant ones. Popular starter use cases and control frameworks can be applied. ISO 27002, the Payment Card Industry Data Security Standard (PCI DSS), the Center for Internet Security (CIS) Critical Security Controls, the National Institute of Standards and Technology (NIST) Cybersecurity Framework (CSF) and other frameworks can be mined for ideas. Login monitoring, data access monitoring, attack detection, change detection and privileged abuse are among the many pervasive elements of security control frameworks, and many tools can be used to address them.

Gartner research shows that use cases can be classified in three categories: threat-, control- and asset-oriented use cases. You should expect to have use cases from all of them, but you don’t need to achieve a balanced number of use cases across these categories. Many organizations will have a strong focus on threat detection, for example, naturally making threat-oriented use cases more common than the others.

Threat-Oriented Use Cases

These use cases are implemented to detect a specific threat. With these use cases, you try to find activities related to specific sources and destinations or specific activities related to tactics, techniques and procedures (TTPs). Some items to look for are often provided by threat intelligence services. Examples of threat-related content include network events that indicate possible command and control (C&C) activity, executables running from user profile folders, and dynamic-link library (DLL) injection attempts.

Many sources should be used for threat-oriented use cases, including threat and risk assessment results, industry reports and strategic threat intelligence resources. If the organization performs red team exercises, the results of those exercises should also be used as input for use case requirements, focusing on red team activity not detected by the blue team. The same logic applies to threat-hunting exercises: They usually identify incidents that were initially missed by the monitoring systems. These situations are also usually opportunities for developing new use cases.

The MITRE ATT&CK model has emerged as the main resource to describe threat activity. [1] It is an important resource in this phase, as it provides a comprehensive taxonomy and structure to the knowledge about threats obtained from the multiple sources mentioned above. It can also be used to identify blind spots and neglected threats that should also be covered by monitoring use cases. However, it should not be treated as a single definitive source of use case candidates. Many techniques from the framework may need multiple use cases to be covered, and some techniques may not be applicable or even important in the context of the organization. Some types of threat activity, such as those oriented toward web or mobile applications, are not covered in depth. MITRE provides other matrices for some of that activity, such as Mobile and PRE-ATT&CK. Other sources, such as the Open Web Application Security Project (OWASP), should also be considered for expanded threat activity coverage when required.

Control-Oriented Use Cases

Control-oriented use cases are those implemented with the intent to monitor corporate policies, frameworks or regulatory documents, such as PCI DSS, ISO 27001, NIST SP 800-53, the Sarbanes-Oxley Act (SOX) or the CIS Critical Security Controls. These use cases can also act as controls themselves. For detective controls, the use case is often part of the control specification itself, such as “investigate all unauthorized access attempts.” For preventive controls, the use case can be a way to demonstrate the control presence or its effectiveness — for example, monitoring for denied events and antivirus signature update events.

Asset-Oriented Use Cases

There is plenty of malicious activity to detect, but ideally, you also want to know about any activities touching specific assets, such as payment card data, intellectual property, personally identifiable information (PII), critical Operational Technology (OT) systems and business applications. These are the use cases typically related to events from DLP systems, file integrity or activity monitoring, and even from business applications.

Use the inventory of sensitive data or applications, or data classification results to identify the data that needs monitoring; risk assessment results may also be useful. Note that asset-oriented use cases are not limited to regulated data. They can originate from compliance needs pointing to specific, regulated data types, such as payment card numbers, or risks, such as critical business data and key intellectual property.

Identifying Candidates

Some organizations start the use case identification process with hundreds of candidates, while others start with just a dozen. At this point, there should be no reason to restrict the number of candidates. Identifying more use case candidates will help the organization to find the best set of use cases for each implementation sprint. It will also give visibility into what the organization wants to monitor versus what it is able to monitor, or is currently monitoring, at each moment. In addition, the identification of use case candidates should be constant; there is no reason to run this activity only once. It should be an ongoing process, updating security monitoring requirements as business, technology and threat conditions change.

> **Do not worry about identifying too many use cases. These are candidates only for now and will pass through a prioritization process before implementation.**

Your process for identifying use case candidates should look something like this:

- Gather existing security monitoring requirements, aligning them to the three use case vectors: threat-, control- and asset-oriented. The organization should mine compliance documents, threat and risk assessment results, threat intelligence sources and asset lists to identify those requirements.
- Review and consider out-of-the-box content from existing security monitoring tools. Many monitoring needs are common across organizations and have been previously solved by the vendors. Vendors have also been mapping their content against the MITRE ATT&CK framework, making it easier to identify use cases covering certain techniques from that model. Also, vendors are constantly updating the content they provide, so revisit the provided use cases often to check for potential new candidates.

> **Do not simply enable everything that comes with the tools. A considerable part of that content may not be aligned with the organization's priorities, or may not be applicable to its environment.**

- Quickly determine the relevance of the potential use cases identified. This assessment should only check whether the potential candidate is relevant for the organization. It is simply a validation step due to the brainstorming nature of the identification phase. Some use cases, for example, may be about threats to a technology that does not exist in the organization. A deeper prioritization exercise will be performed later in the use case management process.
- Fill in the Use Case Design Template (see Table 1) to further define specific use cases to implement.

For example, the organization may have payment data, run an e-commerce website, and like many other organizations, collect sensitive and personal employee data. In this case, PCI DSS compliance, sensitive data and common threats to data popular among the attackers would likely be used to source security monitoring use cases.

Security content development communities, such as Emerging Threats, and commercial content marketplaces, such as the SOC Prime Threat Detection Marketplace, are also good sources of use cases. These should be used in the same way as out-of-the-box content from tools vendors: not as a list to simply download and deploy, but as a menu of potential use cases to be used by the organization. They are good options when you are just starting out and still trying to understand what type of content is usually deployed by organizations with a similar toolset to yours.

Information on a proposed use case can be gathered via a Use Case Design Template (see Table 1).

Table 1: Use Case Design Template Example

Use Case Design Template	
Use Case Name	PCI DSS data in an e-commerce database accessed from outside the cardholder data environment (CDE)
Use Case Description	Database of payment data accessed by hosts outside the CDE
Use Case Category	Unauthorized sensitive data access
Use Case Driver	PCI DSS compliance
Required Data	Access logs from the payment data database
Required Context Data	Information about CDE Internet Protocol (IP) ranges and any whitelists related to database administrator (DBA) activity (jump servers)
Time View	Real-time alerts and monthly reports
Candidate Tool for Implementation	SIEM
Processes Affected	SOC playbook and DBA activities
Teams Affected	SOC and DBAs

Source: Gartner (April 2020)

The template includes a "Use Case Category" field. Although not necessary, additional classification and grouping are helpful when the number of use cases identified and implemented grows. Categories should be used during the measurement and prioritization stages. Common ways to categorize use cases include:

- Broad threat categories, such as “malware” or “insider threat”
- Control groups, such as “endpoint security controls” or “access control”
- Attack chain stages or MITRE ATT&CK tactics, such as “lateral movement” or “exfiltration”

The right abstraction level of use case definitions varies for each organization. Useful use case definitions usually include “what” should be detected or reported, but also some measure of “how” it should happen.

For example, a description like “detect compromised accounts” is essentially “what” only because there are many ways to detect a situation where an account has been compromised. A description like this is better-suited to a category label. On the other hand, “Detect compromised accounts by identifying simultaneous logins from different geographical locations” includes some of the “how” and helps to provide the appropriate level of uniqueness for the use case.

Many organizations have asked Gartner to provide a list of popular use cases. Given the process recommended in this guidance, creating a suitable list is difficult; the correct answer will always depend on your risk exposure and, therefore, on the results of your risk assessment. Table 2 provides a list of popular starter use cases for common security monitoring technologies that makes no assumptions about any organization-specific risks.

Popular Starter Use Cases

Table 2 shows a set of common use cases, with minimal level of detail.

Table 2: Popular Starter Use Cases

Type	Use Case Name	Use Case Description	Tools	Required Data
Compliance	Monitoring endpoint protection platform (EPP) logs for problems with protection (PCI DSS Requirement 5.2)	Generates daily report with any hosts generating Active Directory (AD) authentication events without EPP signature update and daily scan events within the same day	SIEM	Anti-malware/EPP logs, AD logs

Type	Use Case Name	Use Case Description	Tools	Required Data
Compliance	Reviewing all login attempts to health records management application	Generates a daily report detailing all successful and failed login attempts to the health records management application, as required by HIPAA Section 164.308	SIEM or UEBA	Health records management application logs
Threat	Detecting compromised accounts by tracking simultaneous authentication events	Detects account takeovers via authentication tracking; user authenticates from multiple locations simultaneously or with impossible travel time	SIEM or UEBA	Authentication logs and user identity data
Threat	Monitoring for suspicious outbound connectivity	Monitors for suspicious (volume, frequency, destination and ports) outbound connectivity and data transfers by using firewall logs, web proxy logs and network flows; detects exfiltration and other suspicious external connectivity	SIEM, UEBA or NTA	Firewall and secure web gateway (SWG) logs; Cisco IOS NetFlow data
Threat	Detecting network traffic from compromised and infected systems	Tracks compromised and infected systems, including malware detection, by using outbound firewall logs, network intrusion prevention system (NIPS) alerts and web proxy logs, as well as internal connectivity logs, network flows and so on. Data is correlated with threat intelligence containing IP addresses and domain names	SIEM, NTA or UEBA	Firewall, NIPS and SWG logs; NetFlow data

Type	Use Case Name	Use Case Description	Tools	Required Data
Threat	Detecting malware from abnormal endpoint activity	Tracks detailed endpoint activity, including process and script execution searching for typical malware activity, such as abnormal process association and use of Microsoft Windows PowerShell	EDR or SIEM	Endpoint detailed logs, such as sysmon, or EDR logs
Threat	Validating intrusion detection system/intrusion prevention system (IDS/IPS) alerts	Validates IDPS alerts by using vulnerability data and other context data about the assets collected in the SIEM. Although some say this is obsolete, this use case is still here in its modern form of using SIEM to “context-enable” various alerts	SIEM	Vulnerability assessment results and IDS/IPS logs
Threat	Tracking system changes and other administrative actions	Tracks system changes and other administrative actions across internal systems and matches them to allowed policy; detects violations of various internal policies and so on	SIEM	Server and other infrastructure system logs (such as identity repositories and directories); File Integrity Monitoring (FIM) logs; context data (list of administrators, preauthorized source systems and so on)

Type ↓	Use Case Name ↓	Use Case Description ↓	Tools ↓	Required Data ↓
Threat	Tracking web application attacks	Tracks web application attacks and their consequences by using web server, web application firewall (WAF) and application server logs; detects attempts to compromise and abuse web applications by combining logs from different components	SIEM, WAF	WAF, web server logs, and application and database logs
Threat	Detecting abnormal internet access by users	Identifies user behavior anomalies on internet access (such as volume of data, frequency of access, number of destinations)	UEBA or NTA	Proxy logs, firewall logs and data captured on internet egress points
Threat	Matching threat intelligence content to logs for threat detection	Generates alerts for successful access to or from external IPs identified as malicious in existing threat feeds	SIEM	Firewall logs, threat intelligence feeds
Threat	Identifying lateral movement in the internal network	Identifies unusual workstation-to-workstation traffic or abnormal internal resource access patterns by internal devices	UEBA or SIEM	Internal network device logs, internal NetFlow data, server and workstation authentication logs
Threat	Detecting successful phishing attempts	Identifies successful phishing attempts by correlating email gateway and web proxy logs	SIEM	Email gateway and web proxy logs

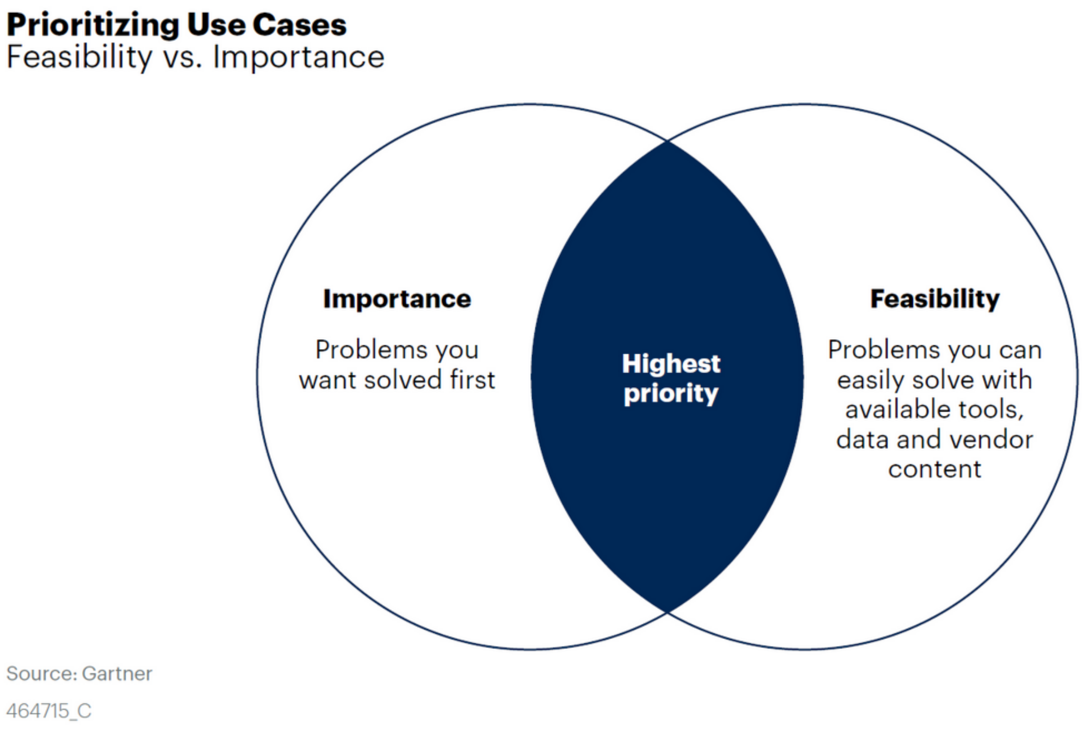
Type ↓	Use Case Name ↓	Use Case Description ↓	Tools ↓	Required Data ↓
Threat	Detecting data exfiltration by potentially malicious insiders	Correlates DLP events with list of contractors at the end of their contract terms and employees flagged as “suspicious” by HR	SIEM or DLP	DLP events and list of suspicious users
Threat	Detecting access to deception artifacts	Identifies access to honeypot systems or honeytoken accounts	SIEM	Firewall logs and honeypot-generated logs, authentication logs
Threat	Detecting distributed denial of service (DDoS) attacks	Identifies volume-based network attacks based on volume of events from firewalls, routers and internet-facing web servers	SIEM	Firewall logs, router logs and web server logs
Assets/data	Monitoring for sensitive data usage across networks	Monitors access to sensitive data usage across the network by monitoring access to data sources and data that crosses the perimeter	DLP or database audit and protection (DAP)	DLP and DAP events
Assets/data	Detecting abuse of privileged access	Identifies excessive access of sensitive data locations by users via utilization of privileged-access credentials	UEBA	File access logs, authentication logs and list of privileged accounts

Source: Gartner (April 2020)

Prioritize Use Case Candidates

Given a long list of monitoring use cases, which ones should the organization implement first? For example, some security architects claim that SIEM use cases must always be selected by order of importance, but that is a big mistake. Gartner research indicates that organizations should not undertake a complex, hard-to-develop use case as a first phase unless absolutely necessary and unless all precautions, such as moving in small steps, are taken. On the other hand, “doing only what is easy” will not yield the desired results either. A much better approach is to balance importance with “feasibility” — that is, ease of implementation (see Figure 2).

Figure 2. Prioritizing Use Cases


Prioritizing Use Cases
Feasibility vs. Importance
Importance
Problems you want solved first
Highest priority
Feasibility
Problems you can easily solve with available tools, data and vendor content

Source: Gartner

464715_C

Table 3 provides some questions that help when measuring the feasibility of a use case and determining its relative importance.

Table 3: Questions for Measuring Feasibility and Importance

Feasibility ↓	Importance ↓

Feasibility	Importance
■ Is there a tool available to address the use case? ■ Is the necessary data available? ■ Is the data being collected inside the candidate tool? ■ Is there any content available to jump-start the use case? ■ Is the necessary people/expertise available to develop content? ■ Is the security monitoring function mature/experienced enough to handle this use case? ■ What will be the performance impact of the volume of required data on the selected tool? ■ Are there legal/privacy/cultural roadblocks to implementation? ■ What are the required processes and personnel to handle the use case output?	■ Was it identified as a top risk on a risk assessment? ■ Is it related to any of the organization’s threat assessment results? ■ Is it a top audit finding deficiency? ■ Is there an imminent compliance audit requirement for the use case? ■ Is there any past incident history at the organization related to what this use case is supposed to detect/monitor? ■ Are there any recent intrusions at similar companies? ■ Is this related to a TTP used by multiple threats? ■ Is this ongoing or relevant activity according to recent threat intelligence reports? ■ Was this issue a cause in a recent public data breach or other intrusion?

Source: Gartner (April 2020)

The process of prioritizing use cases for implementation should follow the steps below:

- Assess the importance of each use case by reviewing the relevance of the related threats, controls and assets to your business, mission and so on. Consider the existing monitoring technologies and use cases already in production to determine relevance and to identify existing coverage gaps. Use tools such as the MITRE ATT&CK framework to map existing content and identify gaps related to threats.

- Analyze the prerequisites for actually implementing the use cases and getting value from them. Gather the details needed to make the determination based on the information from the Use Case Design Template. Do not spend too much time on conducting a detailed assessment at this point; this should be reviewed as part of the use case implementation steps.

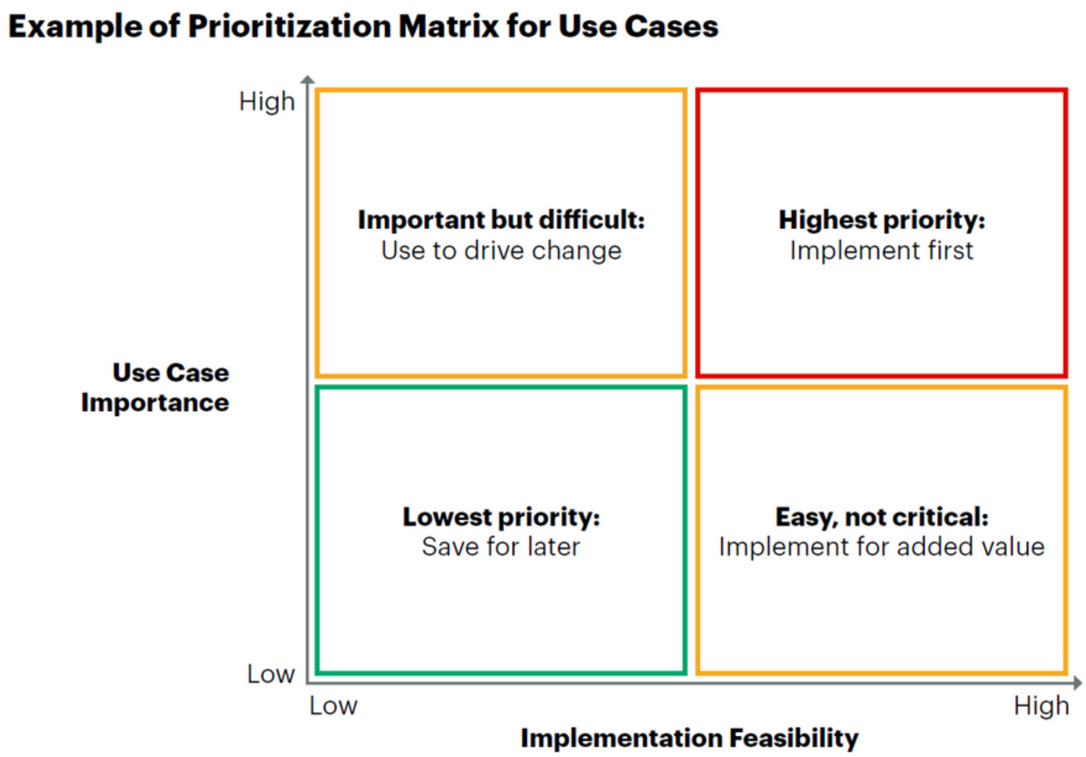
- Prioritize the use cases based on importance and feasibility. A matrix model, as illustrated by Figure 3, can be used at this stage. This balanced approach should help identify the most valuable use cases — those with high priority that are easier to implement.

- Select the top use cases for the next implementation sprint. You may also bundle together use cases when the same requirements, such as tools and required data, are expected to be addressed during the implementation.

- Maintain a list of important but infeasible use cases. This will set the direction to evolve monitoring capabilities. The list should include the reason why each use case cannot be implemented and, ideally, a timeline for a repeated review of feasibility.

Figure 3 provides an example matrix.

Figure 3. Example of Prioritization Matrix for Use Cases



Source: Gartner

464715_C

To give an example, a company focused on PCI DSS compliance and security threats to their website may list PCI DSS-derived use cases for card data access and intrusion detection system (IDS) monitoring, together with vendor-sourced top 10 use cases for common threats. The company may also list asset-oriented use cases to monitor access to employee data, card data and key web assets. The company decides that, because it has cardholder data access records collected in its log management tool and the assessment is coming, PCI DSS use cases should go first. Those use cases have both need and feasibility maximized. Based on the company's risk assessment, and because a competitor was recently hacked, its top risk is a hacked website, so it should focus on monitoring the external web hosting environment next.

Selecting use cases for prioritization is trivial when there are items in the top right quadrant from Figure 3. But what can organizations do to prioritize between those in the top left and bottom right quadrants? The best approach is to direct efforts to improve implementation feasibility for the remaining important use cases, while using use available time and resources to deploy other use cases with less value but which are easier deploy.

For example, an organization has use cases in the top left waiting for a required CASB to be deployed, which is expected to be within two months. Until then, analysts are available to deploy other use cases from the bottom right using different tools and without any impact to the CASB implementation schedule. Those use cases can be implemented during those two months, optimizing the resources available.

Some use cases are less important than others, but that doesn't mean they have no value. The identification phase includes a step to check use cases for relevance, so at the prioritization stage all use cases assessed should have the ability to deliver value to the organization. They may be placed at the bottom of the queue, but with the intent that they will eventually be implemented.

> **Organizations should use prioritization results to continuously improve their security monitoring capabilities. This must be done with the intent to move important use cases out of the "not feasible" zone to "ready for implementation." You should account for skill and technology development in your planning.**

The identification and prioritization of use cases are continuous activities. An organization may decide to conduct those activities periodically in sequence, but if possible they should be done continuously, with new requests or ideas for use cases going through the two mentioned steps as they appear. This approach ensures the list of use cases for implementation is always up to date and allows for rapid implementation of opportunistic use cases, such as those created to detect an ongoing attack.

Sprint of Use Case Implementation

The process of identifying and prioritizing use cases generates an ordered list of use cases to be implemented. Use cases are usually implemented in small batches, or "sprints," and extracted from the top of the prioritized list, similar to the scrum backlog in an agile development process. Start with a small number of use cases so that lessons learned from it can be rapidly integrated, in an agile fashion, in all future use case implementation efforts.

Use case implementation varies according to the selected technology. As expected, most use cases are implemented in SIEM tools, although other technologies can be used for security monitoring and selected for use case implementation. In fact, some organizations have been building their monitoring environment without an SIEM tool, relying on other detection tools such as IDPS, DLP, WAF, EDR and NTA. Even in those cases, the processes described in this framework still apply.

The use case implementation phase is similar to the development and deployment phases of a software development life cycle (SDLC). In the same way as the SDLC turns requirements and design into code, use case implementation transforms the actions and requirements that describe a monitoring use case into tool settings, rules, context data creation, log generation settings and others. Anything usually referred as “content” by some methodologies and practices can be generated as part of use case implementation.

The average organization beginning its journey into security monitoring and use case development should start implementing use cases one by one. This helps build the experience to improve the processes and put together the basic technology components that will form the core of the security monitoring infrastructure. As the process matures, in a “walk, then run” way, the organization should move to implement batches of use cases as part of sprints. This is especially important when there are dependencies between use cases or when they share similarities on chosen technology, data sources and objectives.

The implementation of a use case may include some or all of the steps described in this section. The initial steps refine the requirements listed when the use case was initially identified. During these steps, previously unseen challenges may affect the initial assessment of feasibility of the use case, causing it to be sent back to the prioritization step. A use case that was originally identified as “easy to implement” could later be found to have major obstacles. An assessment of existing logs may indicate that a critical data field for the use case is missing, for example. If the issue cannot be fixed as part of the implementation effort, it may move the use case down the list of prioritized use cases, losing its place in the queue for implementation.

Confirm Monitoring Tool for Use Case Implementation

The brief use case description created during the identification phase includes the tools selected for implementation, such as SIEM, UEBA, DLP, EDR, NTA or others. It should also include basic information about the use case output — for example, does it generate alerts, or perhaps a weekly report?

An organization that has already successfully deployed a SIEM will often use that as its main technology when implementing use cases. A properly working SIEM is one of the most appropriate tools to implement monitoring use cases due to its intrinsic ability to ingest and analyze data from a broad range of data sources.

However, there are many use cases for which other types of security monitoring tools are more appropriate. Use cases that try to identify abnormal user behavior, for example, are more suited to user and entity behavior analytics tools. Also, some organizations opt to build their security monitoring capabilities without an SIEM. For them, there is no option to implement use cases using an SIEM, forcing them to rely on other technologies. There are also those cases where more than one tool is needed — for example, a log management system to collect data and a UEBA tool for deeper analysis.

The decision of which tool to use for each use case depends mostly on the tools’ capabilities and available data. The timing of the use case also affects the tool selection. Some tools excel at alert generation and are more suitable for use cases with real-time requirements, while others may be better options for use cases applied to historical data or optimized to produce reports. The examples below include some of the common questions that help when selecting a tool and illustrate what organizations would typically go through during this step.

Example 1: Alert When Potentially Successful Attacks Against Internet-Facing Servers Are Identified

The security architect identified that existing IDSs can be used to identify attacks against the internet-facing servers. However, to avoid generating alerts against any attack attempt, the use case also requires that the system generate alerts only when the attack is against an unpatched vulnerability on the target server. That information is available as context data on the SIEM, which periodically imports the scan results from the existing vulnerability assessment tool. As the IDS also sends alerts to the SIEM, the SIEM is selected as the tool to use. It can see alerts about the attacks, and has the data required to filter the attack attempts, as well as the correlation capability to implement that filter.

Example 2: Alert When PII Is Being Sent to Systems Outside a Segregated Zone, and Provide Details of the Incident Only to a Specific Group Tasked With Data Breach Investigations

As part of the organization's strategy to maintain compliance with privacy regulations, the security architect wants to generate an alert any time PII is moved to systems outside a certain segregated zone. The only connection between that zone and other networks is currently monitored by a DLP systems network sensor.

This use case requires the ability to properly identify PII. This is a common capability among DLP systems, which makes the DLP tool a good option. The organization has SIEM for other security monitoring needs, but one of the use case requirements is to keep investigations of data breaches separate from other security incidents. Due to that requirement, the organization implements the use case directly on the DLP system instead of using the SIEM to generate alerts.

Example 3: Identify Cases of Potentially Compromised User Accounts Based on Anomalous User Authentication Behavior

Many organizations are moving to find breaches by identifying users behaving in a way that differs from their previous behavior. In this example, even if an SIEM is in place, it may not have the capabilities to dynamically learn what the "expected" behavior for the users is and when their behavior is deviating in a meaningful way. After that assessment, the security architect discovers the organization's stand-alone UEBA system would be the most appropriate system to implement the use case.

With the appropriate tool selected, the security architect checks if out-of-the-box content that matches the use case requirements is available in that tool. If it is, the use case implementation can be accelerated by leveraging the available content. The organization should still go through the next steps to ensure that the required data sources are available. It should also ensure that any required changes to processes, documentation, logging policies and other aspects, including capacity planning for the monitoring tools, are properly identified, tested, documented and implemented.

Determine Data Source Requirements

After the appropriate tools are confirmed, the next step is to specify in detail the required input data sources, including the technology component, specific instances of those components and any related configuration changes required to generate the data.

The data source requirements are primarily about log or event sources. Log and event sources are initially identified as the technology platform generating the events, such as Windows servers, UNIX servers, firewalls or even a business application. In many cases, that definition would be enough because the events to be monitored would come from all existing instances of that technology. However, for large organizations, it is common to have just a subset of those systems selected for log collection. Some use cases would also restrict the systems that would have events monitored by the selected tool. A use case to identify attacks against web servers may not need logs from other types of servers, while more generic use cases, such as authentication failure reports, would benefit from getting logs from all available server instances. In summary, the process of selecting a log source will often include answering not only “what,” but also “which” questions.

After identifying the log sources, confirm that they will generate the required events. Many technologies have flexible logging policies, where the organization can define what type of events will be generated. As such, part of this activity requires an assessment of the logging policies of the selected log sources to verify if the required events are being generated and, if not, to identify the required changes to enable that to happen. This is one of the steps in use case implementation and planning that would often identify major roadblocks, such as performance impact or software version limitations, that would prevent the required data from being generated by the log sources. For use cases that depend on logs from custom applications, additional development may also be necessary. These situations can affect the feasibility assessment of the use case and eventually prevent the implementation from proceeding.

Other data types may be required by some use cases, such as network traffic. In these cases the requirements can also include which network segments or chokepoints should be monitored in order to obtain the required data.

Use case implementation on SIEM may require steps to implement data collection. Just like changes to logging policies, the implementation of new log collections frequently affects the feasibility of the use case. The reasons for this can range from performance impact on sources and SIEM to excessive bandwidth use on sensitive network connections. Some of the decisions are more related to SIEM architecture and are not actually use-case-specific, but they must be made nevertheless to enable the collection of the required data. Typical decisions at this point include the selection of agent or agentless log collection for Windows servers, file transfer mechanisms for log files or even the introduction of new technologies, such as CSPM or CASB tools, to generate the necessary events. There are also cases where additional steps are required to properly ingest the source logs, especially when the source is not natively supported by the

SIEM product. This usually requires the development of connectors and parsers to properly extract and normalize the data fields from the events.

As the coverage of the security monitoring tools evolves, more data will be readily available for new use cases, reducing the need to work on new data source integration. Still, the effort to add and maintain data sources on tools such as SIEM will always be necessary, and the appropriate resources for that need to be available. For additional details, see "How to Architect and Deploy a SIEM Solution."

Determine Use Case Context Data Requirements

The integration of context data into security tools, such as SIEM, can happen via integration with other systems, such as AD. It can also come from a vulnerability assessment tool, or via the manual creation of lists and databases to be used as references by the use case. Context data usually required by security monitoring use cases include:

- User information, usually from AD or Identity and Access Management (IAM) systems
- Information about technology assets, including servers and applications, usually obtained from a CMDB
- IP reputation lists from threat intelligence feeds
- Watchlists for specific users, such as departing employees
- Vulnerability data obtained from vulnerability assessment tools
- Business context data, such as lists of "crown jewel" assets

As security monitoring tools evolve and each deployment expands in coverage and importance, the chances increase that context data will already be available in those tools. However, there will be situations where the necessary context data is not available in the selected tool, especially when the first use cases are being implemented. For those cases, you need to identify and follow the steps required to integrate that data into the tool. The organization must ensure that processes, roles and responsibilities to ensure the data will always be available and current are properly defined and assigned. Typical questions to be answered at this point are:

- Who is responsible for supporting the integration between the context data provider system and the security tool? Who is supposed to monitor that connection, and who should be contacted if it breaks? For example, if the connection between AD domain controllers and the SIEM fails, who is responsible for fixing it?
- Who is responsible for keeping the lists and tables in the security tool up to date? How frequently should they be updated, and what are the steps to do it? For example, if the use case uses a list of "executives' email addresses" to identify high-risk phishing attempts, who is responsible for keeping that list up to date? How frequently is the list refreshed, and what are the authoritative sources of that information?

Identify New or Affected Processes and Operational Procedures

If the use case will generate alerts, you should identify the processes related to handling those alerts. If reports will be generated, specify the recipients of the reports and what should be done with the content of the reports. Identify any new processes or procedures required by the use case and include them as part of the implementation. You may also need to adjust your incident response process, such as by adding new parties to be notified.

Many organizations keep “playbooks” for the SOC with detailed procedures on how to react and respond to every type of alert provided by the security monitoring tools. Changes and additions to the playbook would be identified and developed at this point. Changes to playbooks become even more important with the adoption of security orchestration, automation and response (SOAR) tools. Some organizations use orchestration and automation to extend the detection activity to after the initial alert is generated. In those cases, the playbooks and integration settings with investigation tools and services are just like any other detection content.

More complex or heavily distributed organizations may also have alerts and reports distributed to different groups of people or departments. They should also be involved in ensuring their processes are prepared to handle the output of the use case.

In addition to identifying or creating processes that handle the output of the use cases, identify any requirements related to use case input and maintenance. As described in the previous step, you should implement processes to keep context data up to date.

Develop, Test and Promote Content to Production

After all the preparation work for the use case implementation is complete, you can now begin the development phase, where the content and related settings required for the use case are developed on the selected tools. The steps for this phase will heavily depend on how the organization handles these activities for its security tools. Some large organizations adopt approaches similar to software development, creating nonproduction environments for the development and testing of their security tools. Smaller or more agile organizations tend to perform those activities using the same tools that are used in production. Reducing the chances of development and testing activities affecting production is always a good thing, but budget constraints will usually limit the availability and completeness of nonproduction environments for that end.

Data availability is also a potential limitation of nonproduction development and testing environments. Even a full SIEM environment may not be useful for those activities if the same data sources that exist in production are not available there too. More conservative organizations can go as far as leveraging nonproduction systems equivalent to the real data sources or moving data from production systems to the test environment. However, keeping that infrastructure up and running might be an excessive burden for security teams. In the end, the resources and effort applied to maintain nonproduction environments for security monitoring tools may not be justifiable when compared with the risks related to potential production issues caused by use case development and testing activities. Leaving the organization “blind” to threats for a couple of hours is not equivalent to stopping business production systems, no matter how important security is to the organization.

The key differential to avoid production impact and ensure adequate levels of quality and control is to apply consistent change management practices, no matter how segregated the systems and environments used for development and testing may be. A badly written SIEM correlation rule can sometimes disrupt the entire monitoring capability of the organization, confuse the analysts, fill storage and cause other problems as well.

For use case change management, Gartner recommends building a robust process to allow safe migration of new or modified content into production while not hampering the process with overly complicated rules and procedures. At a minimum, the process in place should ensure that:

- New or modified content is properly documented in a change tracking system. Documentation should include all content produced during the initial review phases and findings related to the use case performance during the testing phase, such as detection limitations or conditions where the use case will not perform effectively.
- Rollback procedures and backups are included in any change package to ensure prompt recovery of the production environment to the previous functioning state.
- People who will be affected are properly notified about the changed or new content deployed.
- Basic postimplementation checks are performed before considering the change successful and implemented. A full review of the use case should also be done after the implementation so that the performance and effectiveness of the use case are properly assessed according to the original expectations from the identification phase.
- New versions of documents with updated processes and procedures are posted on the appropriate repositories.
- New content, including scripts and config files are properly stored.

As mentioned, some of the implementation can be shortened when leveraging prebuilt content from tool vendors. Even in those cases, you must follow appropriate change management procedures to ensure consistency with anything developed from scratch. Doing so will also ensure that the changes required to adapt the prebuilt content to the context of the organization will be made as part of the implementation.

Finally, development and testing steps vary according to the tools being used for the use case. You may want to document those steps separately for each tool when the differences are enough to justify it.

Examples of use case development processes can be found below.

SIEM Use Case Development:

- Enable the collection of the required log sources and context data on the development environment.
- If a correlation rule is the chosen piece of content to be created, analyze what sequence of logged events needs to be tracked and how these events are represented in an SIEM. Using normalized events and taxonomy categories is highly recommended because they help structure the rules and make them easier to modify, maintain and apply to additional log sources. Alternatively, this step and the next may be replaced by the identification of vendor-provided rules to be introduced.
- Implement the correlation rule using the SIEM rule interface. Some products enable a person to click on events and define a rule straight from the observed sequence of events.
- Review the output, such as alerts, scores and dashboards, to check if the intended conditions are being correctly identified:
    - If the tool has functionality to test the rule or algorithm on historical data, execute this functionality to determine how often this rule would have fired in the past had it been enabled.
    - Check for cases of false positives and false negatives, and change the created rules accordingly.
- Review any processes that this rule will trigger, and set up alerts to go to the people who know how to triage them:
    - Update playbooks and inform involved people.
- Record content developed/refined in the appropriate change tracking systems.
- Prepare the change package, backup existing configuration, devise the rollback procedures, and notify SOC shift staff and other operations personnel about new content being deployed.
- Promote the package of content to the production system and enable the rule in the production environment.

DLP Use Case Development:

- Enable any data collection or required scanning settings in the development environment.
- If matching rules or signatures such as regular expressions are being used, analyze the data that needs to be detected and develop the required content to match the data:
    - Some DLP tools provide prebuilt expressions that can be directly used or adapted to the organization's needs. The tools normally provide user interfaces to help the development of

those signatures and expressions, showing in real time if specific data samples would match the developed rules.

- If dynamic profiling of data is being used, set up the data locations and required credentials. Run the profiling on test data to confirm the correct data is being scanned.
- Review the output, such as alerts and reports, to check if the intended conditions are being correctly identified:
  - If the tool has functionality to scan data at rest, execute this functionality to determine if the signature is triggering the expected alerts.
  - Check for cases of false positives and false negatives, and change the created signatures accordingly.
- Review any processes that this rule will trigger and set up alerts to go to the people who know how to triage them:
  - Update workflows, masking requirements and data access permissions.
  - Update playbooks and inform involved people.
- Record content developed and/or refined in the appropriate change tracking systems.
- Prepare the change package, backup existing configuration, devise the rollback procedures, and notify SOC shift staff and other operations personnel about new content being deployed.
- Import the package of content to the production system, and enable data source scanning jobs and policies to run in real-time data flow in the production environment.

Other tools can benefit from similar processes that will ensure that new use case content is produced and deployed both rapidly and reliably.

Implementation Example

Considering all aspects of a use case implementation, the following example illustrates it for a sample use case:

Use Case: Detect Account Takeover (SIEM):

- The use case implementation starts with a review of the initial information about the use case to confirm that the selected tool (SIEM) is the most appropriate and that all feasibility aspects previously assessed are correct. The organization plans to deploy a UEBA tool in the future, but it is not yet available. An SIEM is in place, making it the natural choice because it has the capabilities required.

- Once the tool is confirmed, the security architect verifies if the required authentication events and context, in the format of user information from AD, is available. The organization finds out that all log sources are already integrated, but the connection to AD for context data will have to be established as part of the implementation.
- The content provided by the SIEM vendor is also assessed, and it is determined that some of the canned use cases fit the organization's requirements. The required changes to the rules and new context data are identified and documented into the implementation steps.
- The SIEM team enables the LDAP connection to AD in the SIEM test environment. The required changes to the SIEM tool, including settings such as credentials and digital certificates, are documented so that they can be replicated in the production environment.
- The predefined rules from the vendor are customized as required and enabled in the test environment. The test reveals that some of the standard IT processes of the organization would trigger the rules as false positives. The SIEM team identifies the required changes to avoid those alerts and changes the rules accordingly.
- The SOAR solution used by the SOC is identified as the recipient for the new alerts to be generated. The playbook describing how the SOAR tool should react to the new alerts is written, reviewed and signed off on by all teams involved in the new procedures.
- All changes and new content are documented in a change tracking system. Required change tickets on all change management systems involved are created and scheduled for the accorded date and time for promotion to production.
- Changes to production systems are performed according to the written plans. Postimplementation review is performed to ensure that no negative impacts result from the changes and that the rules are behaving as expected.

Measure and Review Use Case Performance

The implementation of a use case is not the final step in its life cycle. You may need to modify use cases due to changes in the source data, nature of threats, business needs and technology environment. The performance of the use case, including aspects such as effectiveness and efficacy, could also be a reason for changes to a use case in production.

Use case metrics support the efforts to review and tune use cases. Table 4 shows some of the typical metrics related to use cases.

Table 4: Typical Use Case Metrics

Metric ↓	Objective ↓
Number of alerts generated by the use case per period of time	Verify the relevance and value of the use case and its impact to security operations.

Metric ↓	Objective ↓
False positive and false negative rates of the use case	Verify the effectiveness and efficiency of the use case.
Incident response investigations triggered by the use case	Verify the value of the use case.

Source: Gartner (April 2020)

Some conditions identified by these metrics can indicate the need to review a use case, while others would be identified during a periodic review.

Use cases are often reviewed:

- **Following a recent implementation:** You won't know the real results of a new use case until after it is exposed to full production conditions. A postimplementation review of the use case helps you identify issues not spotted during the implementation phase. Also, performance-related measurements may be possible only after the use case is exposed to production data. This should happen soon enough after deployment to minimize impact from a problematic use case, but waiting a few days or weeks so the appropriate measurements can be taken is acceptable.

- **After changes to the IT environment, business, or threat landscape:** Use case changes may be required for new conditions in the IT, business and threat environments. A merger or acquisition may require new context data to be added to some use cases. A new threat intelligence report on threat tactics, techniques and procedures (TTPs) could indicate that an existing use case needs to be modified in order to detect that new behavior. Organizations with mature change management and security intelligence processes can build triggers to start a use case review when necessary. These triggers can only be implemented when the appropriate metadata about the use case is available. This includes things like which business areas are affected by the use case, which threat or actors are monitored, or simply which technology components are used as data sources.

- **When thresholds are reached:** The most common performance expectations of a use case are related to the false positive ratio, the number of alerts being generated and false negatives identified, and the performance impact on the monitoring tool and on the data sources. Those thresholds may be defined individually by use case or follow a generic standard, such as "no use case can exceed a 10% ratio in false positives per day."

- **When detection failure, or the occurrence of a "false negative," is identified:** Incident response, threat hunting or even red team exercises may identify situations where an event should have been detected, but wasn't. In those cases, the use cases that should have generated an alert for those situations must be reviewed.

- **During periodic review:** Reviews should also occur in periodic cycles, either when all existing use cases are reviewed, or based on an individual use case schedule where each one has its own review date. This date should be set based on when the use case was originally implemented or last reviewed. The individual schedule approach requires more work to maintain the review schedule, but it also avoids accumulating too much review work on a single task. Periodic reviews should be conducted at least once per year, but ideally, more frequently than that.

- **During periodic refresh of context data:** Many use cases depend on context data to perform adequately. Stale lists used as context data are a common source of false-positive and false-negative cases, directly affecting use case performance. Context data refresh, when not carried out automatically, must be part of other IT or security operations processes. Periodic reviews should only be used as an assurance that ongoing review is being performed appropriately.

When a use case is reviewed, its main characteristics, settings, inputs and outputs are checked to ensure the use case is still filling its purpose and falls within acceptable conditions of performance and efficacy. Any issues found are addressed by the two remaining steps of the use case process: tuning or removing the use case.

Tune Use Case

The review step will frequently identify use cases that need to be changed in order to remain useful, or even to avoid negatively affecting the monitoring environment. Use cases are often modified to optimize effectiveness or improve the tool's performance. In addition, changes to the environment sometimes require changes to the use case just to keep it operational, such as when technology changes affect the generation of logs used by a SIEM use case. Changes to use cases usually involve:

- Changing correlation rules or detection signatures.

- Changing the event generation settings, usually suppressing the generation of certain events at the source or at a collection point.

- Changing output information in the generated alert or report.

- Adding or modifying context data, such as blacklists, whitelists, exceptions and watchlists.

- Changing operational processes, such as an initial alert triage procedure.

- Changing the use case implementation technology. This is usually a big change and could also be seen as removing the use case and implementing a new one.

> **Use case effectiveness optimization is, together with performance tuning, the most common situation of use case tuning. This is where false**

positives and false negatives are studied and minimized by changing the use case implementation characteristics.

Use case tuning can be seen as troubleshooting and adjusting the use case for effectiveness or performance. The specific steps are dependent on the technology being used. Tuning an SIEM use case usually includes changing correlation rules or the events being generated and collected from the sources, while working on tools such as UEBA may require steps to retrain a machine learning algorithm. In any case, the goal of the process would usually be related to:

- **Reducing false positives:** The use case is generating “false alarms,” alerts or reporting a potential threat or violation that is not confirmed after an investigation.
- **Reducing false negatives:** The use case is not generating alerts or reports that include situations that should be identified according to the use case objective.
- **Reducing number of alerts or report sizes:** The use case is consistently generating too many alerts for the security team to triage, or the reports are too big to be properly handled by the operations teams.

Most security teams adopt the strategy of building use cases in such a way as to minimize false negatives first, in order to maximize detection sensitivity. They will monitor the generated events in the postimplementation review to check if the false positive rate remains on an acceptable level. If too many false positives are generated, changes to the use case are made to keep it at an acceptable level, which usually affects the detection and false negative rate as well.

One of the main challenges of this process is to find out what the acceptable level is. Many organizations will start by defining simple rules like “no more than 1% of false positives” or “no more than five false positives per day,” but they quickly find that the requirements are more complex than that. Some use cases are designed to discover more “critical” events than others, with different levels of expectations for false positives and false negatives. A use case designed to monitor the “crown jewels” should be designed in a way to minimize the false negatives, even if it generates more false alerts. Use cases looking for less critical events would be designed to avoid generating unnecessary alerts, even if a few real incidents might be missed. The right balance will depend on the value of the alerts and the cost of investigating them.

An excessive volume of alerts or report entries is an additional challenge that may require tuning. A use case might be performing with negligible false positive and false negative rates, but the monitored conditions are so common that a high number of alerts or a big report is generated. No tuning related to false positives and false negatives would happen in this case because the prevalence of the conditions is too high. In these cases, you should reassess the value of the use case: Does it make sense to generate, and consequently investigate and respond to all these alerts? If not, what additional conditions could be added to the definition of the use case to reduce the number of alerts generated?

This problem occurs frequently in the security monitoring world, and experienced teams learn how to define use cases that would be less prone to these conditions. However, there will always be cases where executives or auditors will request something like “all attacks against our website must be investigated.” For those, it’s imperative to demonstrate the cost of these approaches — such as performance impact on the systems and resources used to investigate all generated alerts — and the obtained value. What is the organization planning to do with the alerts? Considering other use cases that have been identified and are waiting for implementation, are the alerts more or less valuable in terms of risk reduction? This evaluation can demonstrate the low value of the use case and persuade the requester to reprioritize it or to add conditions to reduce the number of alerts generated. In this example, it could be done by adding a condition, “all attacks against our website *with chances of success, determined by correlation with vulnerability data.”*

Table 5 includes examples of typical changes to use cases according to each tuning objective.

Table 5: Examples of Changes to Use Cases

	Reduce False Positives	Reduce False Negatives	Reduce Alert Frequency or Report Size	Reduce Performance Impact on Monitoring Tools

	Reduce False Positives	Reduce False Negatives	Reduce Alert Frequency or Report Size	Reduce Performance Impact on Monitoring Tools
Examples of possible changes	- Increase thresholds to trigger an alert. For example, increase the number of matches in DLP signatures, such as “alert if more than five SSNs are found.” - Expand rules to correlate events with context data such as whitelists, threat intelligence and vulnerability data. - Expand behavior learning time window in the UEBA tool	- Reduce filtering of events on log collection. - Identify additional events or event sources to match in the available input data. - Reduce thresholds to trigger an alert. For example, reduce the number of matches in DLP signatures, such as “alert if more than five SSNs are found.” - Expand correlation rules to include blacklists. - Reduce the behavior learning time window in the UEBA tool	- Expand rules to correlate events with context data such as threat intelligence and vulnerability data. - Reduce scope of the use case to only a certain group of assets, such as servers or networks.	- Reduce the logging policy on the event sources. - Reduce scope of the use case to only a certain group of assets, such as servers or networks. - Reduce the time window for correlating events to reduce memory utilization. For example, if a certain combination of events is seen happening in less than X minutes.

Source: Gartner (April 2020)

Removing a Use Case

There are many reasons why a use case would be removed. Some situations can render a use case no longer viable. Tuning is not capable of keeping false negatives and false positives below an acceptable level, or cannot keep the use case from using too many of the resources on the monitoring systems. Or perhaps the data required for the use case to function is no longer available due to changes in data sources.

In addition, a use case may not be providing enough value. The conditions being assessed may be so rare that it doesn't justify wasting resources looking for them, or perhaps the alerts generated are not useful for incident response or investigation purposes. In these cases, removing the use case is not something done to stop it from causing problems, but to keep the content on monitoring systems relevant.

There are also situations in the business environment that could affect the reason why the use case was implemented in the first place. Control-oriented use cases, for example, might become obsolete if the business is suddenly no longer required to be compliant with regulatory requirements. For example, the business may outsource payment processing to avoid PCI DSS requirements.

Finally, a use case may be valuable, but a better way to address it becomes available. For example, the use case content in an SIEM can be removed and reimplemented in a UEBA tool. This may be the case with some insider tracking examples or compromised account detection.

When a use case is marked to be removed, a series of steps must be followed to ensure all links and unnecessary processes and procedures are also removed. The most important step in this phase is to confirm the removal of the use case with the original requester or owner of the use case. Some use cases may have dependency issues or be part of a broader set of controls, usually related to compliance requirements. If the removal is related to use case implementation or tuning challenges, the requester may want to suggest changes to the use case in order to keep it viable instead of removing it. In any case, obtaining the agreement from the group or role that originally requested the use case is an important measure. It ensures all those involved are aware of the current state of the monitoring processes and what is currently implemented and running.

The remaining steps of the removal phase are similar to the implementation process. This includes updating documentation and playbooks, informing involved people and stakeholders, as well as making the actual changes to the monitoring systems to remove the content related to the use case. The quality of the documentation of the existing content is vital to avoid affecting other existing use cases that may be sharing content with the use case that is being removed. The risk is usually small when only a small number of use cases exists, but in large environments with many use cases in production, cases of shared content are more common and can be affected by use case removal activities. Content typically removed as part of a use case removal include rules, signatures, scanning policies, context data, log collection settings and report definitions.

As a final step, it is useful to maintain a list of use cases that have been implemented and removed, including the reason for removal. This list should be useful during the use case identification and prioritization phases to avoid reimplementation of content that previously failed to produce the expected results in production. The list can also be revisited periodically to identify use cases that had to be removed due to now resolved technology limitations and that now could be reimplemented.

Measure Program Performance

As with any continuous cycle, you must properly measure the use case process to allow appropriate management and to support strategic and tactical decisions. Metrics related to the use case management cycle will support decisions related not only to this process, but also to planning activities for security operations and monitoring infrastructure. Use case management metrics can also be integrated and incorporated into a broader set of security operations metrics. For more details, see "Developing Metrics for Security Operational Performance."

Table 6 lists the most common metrics related to use case management.

Table 6: Common Use Case Management Metrics

Metric ↓	Objective ↓
Use cases in production versus use cases waiting for implementation	Support resource planning by providing a view of how many of the intended use cases are currently in production.
Number of use cases reviewed per time period	Support resource planning by providing a view of the effort related to use case review.
Number of use cases tuned/changed per time period, including reasons for changes	Provide a view of how much the monitoring environment is changing and the reasons behind the changes.
Number of use cases removed per time period, including reasons for removal	Provide a view of how much the monitoring environment is changing and the reasons behind the changes.
Number of use cases implemented per monitoring tool	Measure value and utilization of the monitoring tools in place.
Number of use cases not implemented due to technology limitations	Provide support for technology decisions and planning by identifying bottlenecks.
Use Cases coverage of the MITRE ATT&CK Framework	Provide a view of how much of the most typical threat tactics and techniques are currently covered by implemented use cases.

Source: Gartner (April 2020)

In addition to the metrics listed, do not forget to maintain the lists of use cases in the many different states described throughout this research: Identified and prioritized, not feasible to implement, in production, being tuned, or removed. They provide valuable information to many aspects of security operations — for example, having to quickly assess if a certain threat can be detected with existing use cases.

> **Metrics will help you mature security monitoring capabilities by selecting and improving tools that can enable the implementation of use cases that have been identified but deemed unfeasible to implement.**

The key reason for having a measurement phase as part of this framework is to guide the efforts on evolving the security monitoring practices. They will also improve the capacity management of your security operations teams. The assessment of the implemented use cases, considering the threat landscape and control deficiencies, will also drive your decisions for the next use case prioritization cycle, ensuring appropriate coverage and optimal resource utilization.

Risks and Pitfalls

The process of defining and refining monitoring use cases is difficult to run well, and organizations have suffered the following risks and pitfalls:

- **Exclusive reliance on canned or vendor-imposed use cases:** One size fits all will not perfectly fit your organization. It will work, but it is most likely that the value won't be maximized for you. Vendor content is developed to address problems common to many organizations, leaving a gap related to the problems particular to your environment. To overcome this, look for the problems your stakeholders need solved, and adapt or create the use case content accordingly.

- **Lack of a consistent mechanism for "converting" vague problems into precise use cases:** Basically, a broken use case discovery process means you will solve only easy and specific problems, such as "see if anybody connects to our payment card database at night." To overcome this, use the guidance in this research to create your process and leverage additional resources such as the MITRE ATT&CK framework and control frameworks to obtain more prescriptive use case candidates.

- **Driving a use case from available data alone:** An input-driven approach to use case development leads to dangerous blind spots in detection capabilities and inefficient prioritization of use cases development. An output, or use-case-driven approach is the most effective manner to implement security monitoring tools, such as SIEM. [2] Start with the use case requirements, which will drive the data collection requirements.

- **Ignoring the data collection needs and planning to rely on data that cannot be collected:** Governance or political challenges sink many security monitoring projects. Some data, such as workstation logs or cloud provider logs, is exceedingly hard to collect at some organizations. To overcome this, include the prioritization framework for importance and feasibility in the planning and ensure the results are being properly communicated to all stakeholders.

- **No formal role assigned for content management:** Many organizations do not see the need to continuously create new and maintain existing use cases. They end up leaving this activity as just one more task for SOC analysts or security engineers to perform, but with no focus or priority. When this happens use case development and tuning tends to take the back seat to other day to day operational functions and incident response tasks. To overcome this, create a process for turning needs into content and solid monitoring deliverables and ensure it has enough resources formally assigned to it.

- **Lack of understanding of the MSSP role:** MSSPs are an increasingly common part of organizations' security operations. Many service providers do not do any use case development aside from enabling content provided by tools vendors. Organizations must clearly state their requirements for content development when writing MSS RFPs and when establishing the joint operating model with the selected provider.
- **Overly burdensome use case process:** Some organizations adopt a "SIEM content as code" model to the extreme, applying excessive governance and change management to the process. To overcome this, balance the agility needed to counter threats fast with a repeatable process that can be consistently used. Sentient attackers who are a threat to your organization can adapt fast, and so should your security monitoring practice. They won't just give up and leave, so your process should be threat-evolution-proof.
- **Lack of measurement and tuning efforts:** Use case optimization efforts are often limited to only one dimension, such as false-positive rate. This approach is not enough to obtain the expected value from security monitoring use cases. To overcome this, first accept the dynamic nature of security monitoring content and use cases, due to changing threats and changing environment, and implement the process for deciding what works and how well it works.
- **Not understanding or identifying use case limitations:** In the same way this process requires the precise definition of what each use case does, it should also help organizations understand what the existing use cases are not doing. This helps the effort to improve monitoring tools and practices and prevents an incorrect understanding of the current monitoring capabilities.
- **Not considering changes and evolution of monitoring tools:** Many organizations could benefit from new technologies and tools that better fit certain implemented use cases. Many use cases implemented on SIEM could be moved to other tools with more efficient implementations. Data sources can also change due to technology evolution, such as moving from NetFlow data collection and analysis to NTA tools, or from workstations' native logs to EDR events. Organizations should include the assessment of the changes in the monitoring tools landscape when reviewing implemented use cases and the list of infeasible use cases.

Related Guidance

These related guidance documents can be used together with this research:

"How to Operate and Evolve a SIEM Solution"

"How to Architect and Deploy a SIEM Solution"

Evidence

[1] B.E. Strom. "Adversarial Tactics, Techniques and Common Knowledge — ATT&CK™ — Presentation." MITRE. September 2015.

[2] "How to Architect and Deploy a SIEM Solution"

Recommended by the Author

Solution Path for Implementing Threat Detection and Incident Response

Applying Network-Centric Approaches for Threat Detection and Response

How to Start Your Threat Detection and Response Practice

Developing Metrics for Security Operational Performance

Recommended For You

Decision Point for Postmodern Security Zones

SOAR: Assessing Readiness Through Use-Case Analysis

2020 Planning Guide for Security and Risk Management

How to Operate and Evolve a SIEM Solution

Assessing the Impact of Machine Learning on Security