PICUS

5 Technical Mistakes That Break SIEM Detection Rules

Security Information and Event Management (SIEM) systems are designed to help organizations monitor and respond to security threats in real-time. By collecting, analyzing, and correlating data from multiple sources, SIEM systems can provide a comprehensive view of an organization's security posture and help identify potential security threats.

If you're using a SIEM system, you likely have a set of rules in place to help you detect and respond to potential threats. However, despite the benefits of SIEM systems, many organizations struggle to implement effective SIEM rules. If those rules aren't working properly, your entire network could be at risk.

Technical mistakes are a common problem that can cause your SIEM rules to stop working and disrupt an organization's security monitoring and incident response efforts. These mistakes can range from simple syntax errors to more serious issues, such as mismatched data models and event types. In this blog post, we will explore some common technical mistakes that can break SIEM rules and how to avoid them.

Assess Your Detection Rules with
PICUS Detection Rule Validation (DRV)

1. Regex Problems

One of the most common technical mistakes that can lead to broken SIEM rules is using regular expressions (regex). A regex is a powerful tool for defining patterns and matching specific strings of data. However, regex can be complex and challenging to understand, and even a tiny mistake or typo can lead to broken SIEM rules.

Here are some common regex mistakes that can lead to broken SIEM rules:

Incorrect syntax: Regex has a specific syntax and structure; even minor errors or typos can cause the regex to fail. For example, forgetting to escape special characters or using the wrong delimiters can lead to broken SIEM rules.

For example, suppose a SIEM rule uses a regular expression to match specific events in the log data. The regular expression includes a quotation mark (") as a special character to match a specific string in the log data. In order to properly use the quotation mark in the regular expression, it must be escaped with a backslash. If the backslash is not correctly placed before the quotation mark, the regular expression may not be able to match the string in the log data precisely, and the rule may not trigger as expected.

To avoid this problem, security teams should carefully review and test their regex to ensure that it uses the correct syntax and structure. This includes consulting online resources and documentation, as well as testing the regex against sample data to ensure that it is functioning properly.

Overly complex regex: Regex can be powerful and flexible, but it can also become challenging to understand and maintain as the regex pattern becomes more complicated. This can lead to broken SIEM rules, as small mistakes or typos can be difficult to spot and correct.

Security teams should strive to create simple and readable regex patterns to avoid this problem. This includes using clear and concise regex syntax and breaking the regex pattern into smaller, more manageable components.

Incorrectly matching data: Regex can be used to match specific strings of data, but it can also match data that was not intended to be matched. This can lead to broken SIEM rules, as the regex may match the incorrect data and generate false positives or negatives.

For example, consider a rule that is designed to detect instances of network scanning. The rule may use a regular expression to match IP addresses in the log data, with a backslash used to escape the dots in the IP address:

```
\\d+\\.\\d+\\.\\d+\\.\\d+
```

2. Case Insensitivity Misuse

Another common issue that can lead to broken SIEM detection rules is failing to account for case sensitivity. For example, let's say you have a rule that is supposed to trigger whenever a user logs in with the username "admin". You might create a rule that looks like this:

```
IF user logs in with username "admin" THEN trigger alert
```

This rule will only trigger if the username is entered exactly as "admin". If the user accidentally types "Admin" or "ADMIN", the rule will not trigger. This can be a serious problem if your system is being attacked by someone intentionally trying to evade detection by using a different case for the username.

3. Logical Errors From Operator Misuse

Operator misuse occurs when a detection engineer misunderstands, misinterprets, or misuses the meaning of a logical operator, such as AND, OR, or NOT. These operators combine or negate different conditions in a detection rule, and they play a crucial role in determining the behavior of the rule. Using the wrong operator can lead to broken detection rules, resulting in false positives or negatives. In other words, you might end up missing important security events or generating unnecessary alerts.

As a basic example, let's say you have a rule that triggers an alert if the source IP address is "192.168.1.1" and the destination port is "443". But if the "and" operator is misused and changed to an "or" operator, the rule will now trigger an alert if the source IP is "192.168.1.1" or the destination port is "443". This means that any traffic coming from a different IP address but going to port 443 will also trigger an alert, leading to false positives.

To avoid these logical errors, it's important to carefully review and test your detection rules to ensure that the operators are being used correctly. This means double-checking the logic and testing it against different scenarios to ensure that the rules accurately identify potential threats.

4. Mismatched Data Models

One common mistake that can break SIEM detection rules is the use of mismatched data models. A data model is a framework that defines the structure, relationships, and semantics of the data that a SIEM system processes. It specifies the various fields, data types, and formatting rules that must be followed to interpret and analyze the data accurately.

If a SIEM rule is based on a data model different from the data being analyzed, it can lead to incorrect results. For example, suppose a rule is looking for a specific string in a field that is not present in the data. In that case, it will always return a false negative, regardless of whether the string actually exists in the data.

5. Mismatched Event Types

An event type is a classification that is used to categorize security events based on their characteristics and behavior. It specifies the types of data collected and analyzed, as well as the actions that are taken in response to specific events.

If a SIEM rule is based on an event type different from the data being analyzed, it can also lead to incorrect results. For example, if a rule is looking for a specific type of network traffic, but the data being analyzed is from a different network or protocol, it will return false negatives or false positives.

Conclusion

In conclusion, SIEM systems are crucial for businesses looking to protect their data and systems and mitigate potential security threats. However, creating and managing effective SIEM rules can be challenging, and technical mistakes like regex typos can lead to broken SIEM rules.

To avoid this problem, organizations should carefully review and test each new rule before implementing it and regularly review and test existing ones. If you don't adequately test your rules, you run the risk of implementing ones that are ineffective or cause false positives or false negatives, which can cause your entire SIEM system to break down.

Picus Detection Rule Validation (DRV) product identifies not only broken detection rules but also inefficient rules. It provides security teams insights on improving their threat detection and response capabilities. Moreover, Picus DRV validates the effectiveness of existing and new rules based on log coverage, alert frequency, and performance metrics. Therefore, Picus DRV saves valuable time and resources by automating the manual detection engineering processes and improving the detection effectiveness and efficiency of SIEM systems.

Assess Your Detection Rules with PICUS Detection Rule Validation (DRV)