Fingerpointing False Positives: How to better integrate Continuous Improvement in Security Configuration Compliance

Desiree Sacher

Twitter: d3sre

Verifying secure configuration of IT setups is an everlasting challenge and becomes even more important when part of the infrastructure is moved to the cloud. To verify if the current defined baseline configuration or default specification is trustworthy and functional, bad alerts should be as well reviewed as security incidents. This paper presents a suggestion for analysing alerts caused by configuration compliance monitoring (also called technical security compliance) and integrity monitoring alerts.

I. Introduction

After publication of the last paper [1] and its peer reviewed version [2] I was asked if this categorisation of false positives could as well be applied to file integrity monitoring and configuration compliance monitoring, as those alerts become highly important with monitoring cloud infrastructure. This paper explains this adoption in detail and includes key performance metrics that can be derived from such a categorisation of alerts. The structural method of the categorisation is based on the four Ps of ITIL[3] (people, process, products, partners) whereas people were purposely excluded to avoid blaming culture and as this is hard to judge from outside. It is also based on a systems thinking approach by separating follow-up actions in tasks that can be directly resolved in the SOC and tasks that need actions from other functions but interpreting every alert in context of a bigger system. For this proposal to work it is expected to have IT processes in place that document changes to the infrastructure, authorized by a process where affected customers or application owners review changes that affect their systems.

II. Materials and Methods

The established structural method was applied to these new services as displayed in Table 1: Overview resolution categories and again illustrated with examples. The description/significance section explains what this category in numbers can tell the reader and why it is of importance. The benefit section is a quick outlook on what positive changes can be introduced when interpreting the numbers of such a statistic.

The resolution categories, listed in Table 1: Overview resolution categories, were defined for file integrity monitoring baselines and compliance configuration monitoring policies, usually configured in a technical security compliance and file integrity monitoring tool. Alerts created from such configurations often are managed by the SOC but can also be automated to be directly assigned to the system engineering team for immediate feedback. This value is not intended to replace other categorisation but should be instead used as an additional field to improve security configuration quality by focusing on what caused the wrong alert instead of just updating the baseline. Figure 1: Decision Matrix for Applicability Categorization can assist in finding the right categorization. The categories should be applied after successful analysis of the alert where identification of root cause for the event has been finished, along with detailed explanation for the alert.

	ENGLISH VERSION	DEUTSCHE VERSION
A)	Legitimate violation authorized by change	Rechtmäßiger Verstoß erlaubt durch dokumentierten „Change“
B)	Legitimate violation with missed change documentation	Rechtmäßiger Verstoß mit fehlendem dokumentierten „Change“
C)	Temporary configuration aberration	Temporäre Konfigurationsabweichung
D)	Configuration error in baseline	Konfigurationsfehler in “Baseline”
E)	Limitation in verification product	Einschränkung in Überprüfungswerkzeug
F)	Test alert	Test Alarm
G)	Unauthorized change without legitimate cause	Unerlaubter “Change” ohne rechtmäßige Ursache
H)	Activity with no change documentation required	Aktivität ohne erforderlichen “Change”

Table 1: Overview resolution categories

a) Legitimate violation authorized by change

Examples:

- Configuration file changed with last official change

Description/significance:

This type of alert is caused by changes in monitored files (on premises or in the cloud) but the SOC did not have a direct suppression associated. The update of the baseline configuration file was either not included in the process or was not possible beforehand. This can point to configuration limitation (such as needed updates that cannot be timed or activated time controlled) or process improvements where SOC should be better included. This category requires adjustment to the baseline configuration as a follow-up task to make sure no further false alarm is triggered by the old configuration.

Benefits:

Direct visibility into security operational process gaps.

b) Legitimate violation with missed change documentation

Example:

- Configuration file changed but opening a change for tracking the alteration was forgotten or skipped

Description/significance:

This category of alerts creates statistical values to illustrate when security and IT processes are not being correctly followed (often caused by human error). These activities are hard to detect otherwise and therefore important to track, report or otherwise communicate to system responsible teams or security/risk management authorities. This category requires adjustment to the baseline configuration as a follow up task.

Benefits:

Illustrate gaps in IT process compliance.

c) Temporary configuration aberration

Example:

- An emergency setup requires configuration adjustments that will be reversed after resolution

Description/significance:

This type of alert is usually unpredictable and important to track as long as the temporary setup is in place. Depending on the amount of time the emergency setup is in place, either the baseline should be adjusted or can be kept. For short amount of times it can make sense to configure a suppression. For longer adjustments, a risk entry could be reasonable.

Benefits:

This value can illustrate the number of probably handcrafted emergency solutions required in a company. High amounts of this number can illustrate less structured companies and therefore become valuable to rate a company’s IT maturity.

d) Configuration error in baseline

Example:

- The baseline setting derived from official configuration compliance policies was not reflecting current knowledge or best practice
- The integrity policy included a wrong configuration, e.g. value for test system was configured for production system

Description/significance:

This category reflects misconfiguration problems based on bad quality information delivered by the system engineering teams. This information needs to be imported from outside and is usually hard to verify by SOC

personnel. It should therefore not be included in SOC KPI values and rather be used to illustrate misconceptions, missing security awareness or missing security knowledge throughout the company.

Benefits:

Separate reporting of this value can be used to illustrate the company's overall security maturity and the maturity of security processes. This can be an important value to address structural problems strategically.

e) Limitation in verification product

Example:

- The value cannot be correctly configured in the monitoring product and is therefore causes bad alerts

Description/significance:

The current product in use for configuration compliance or integrity monitoring is limited in its configuration possibilities and therefore causes bad alerts. This can only be improved by changing the product or applying extensive tricks or workarounds to the setup.

Benefits:

This value can help illustrate the need for a different product, to take more efficient use of SOC analyst's time and skills.

f) Test Alert

Example:

- A new configuration is tested

Description/significance:

This alert reflects alerts created for testing purposes. This value is important for files or systems that hardly create alerts to proof operational reliability. It should be excluded from reports.

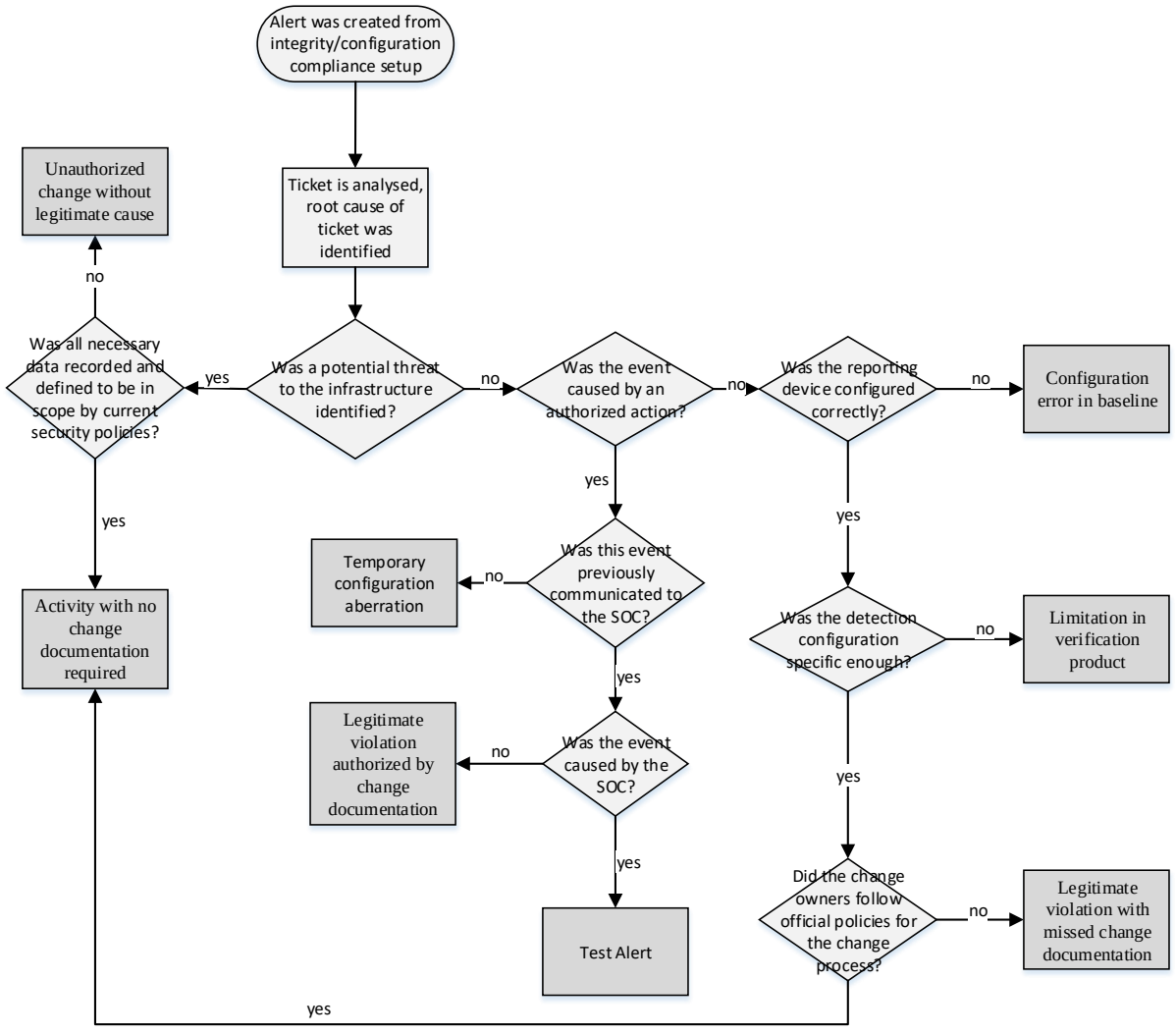

Figure 1: Decision Matrix for Applicability Categorization

Benefits:

Direct separation from alerts not relevant to judge the quality of security verification setups

g) Unauthorized change without legitimate cause

Example:

- Compliance monitoring or integrity alert cannot be tracked back to any official activity

Description/significance:

This type of resolution usually causes a security incident analysis. It is of important value for threat and risk estimations. The baseline configuration is not adjusted.

Benefits:

This is usually the true positive type of alert that Compliance Configuration monitoring as well as Integrity monitoring was created for. A separate display of this value shows how often these services have fulfilled the expected value.

h) Activity with no change documentation required

Example:

- Documentation for this type of system change is not required and this specific alert cannot be resolved
- Pentest activity with changes on systems that would not have to be documented in change documentation

Description/significance:

This category documents alerts that cannot be resolved due to missing documentation and no requirement to document changes on the system or this file type. It can also have resolved alerts, but extended analysis was needed as security policies do not regard these changes as critical. This category comprises of accepted risks, usually regulated by manageability. It can also include changes performed on systems during a penetration test that did not have to be documented by system engineers on a regular basis and therefore could not be otherwise verified. Such documentation gaps need to be verified by security management and possible influence security policy scope.

Benefits:

These kinds of alerts illustrate what risks are generally accepted and no requirement for documentation of activities is enforced according to the current security management guidelines.

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 detection capabilities working correctly?*	*Were the MSSP service systems configured correctly to detect a threat to my company/customer?*	*What lesson can be learned from this event?*
Legitimate violation authorized by change	No – False Positive	Yes – True Positive	No– False Positive	Adjust baseline configuration, Update information process to involve SOC
Legitimate violation with missed change documentation	No – False Positive	Yes – True Positive	Yes – True Positive	Adjust baseline configuration, document change
Temporary configuration aberration	No – False Positive	No – False Positive	No – False Positive	Save or keep old baseline, possibly add temporary suppression
Configuration error in baseline	No – False Positive	No – False Positive	No – False Positive	Security configuration/baseline needs engineer review
Limitation in verification product	No – False Positive	No – False Positive	No – False Positive	Tool provider should be verified
Test alert	No – False Positive	Yes – True Positive	Yes – True Positive	Should be excluded from reporting
Unauthorized change without legitimate cause	Yes – True Positive	Yes – True Positive	Yes – True Positive	Security incident should be opened, and alert followed up on

Case	**C-Level Perspective**	**SOC Perspective**	**MSSP Account Manager Perspective**	**Follow Up Action**
Activity with no change documentation required	No – False Positive	Yes – True Positive	No – False Positive	Alerts should be forwarded to security management/policy responsible to confirm scope

Table 2: False Positive - True Positive Comparison by Perspective

III. Results

These new categories again shift the focus from a two-sided to an eight-sided perspective. The approach is solution oriented and allows for more granular reporting possibilities, where the improvement steps can be better included in the process. Compliance configuration monitoring, as well as integrity verification is heavily driven by the knowledge and understanding of system engineers. Therefore, it is even more important to document where in processes and setups changes need to be included and to shift focus on the aspects a SOC can actually influence. Often it makes sense to directly assign such alerts to system engineers for an initial review. The traditional categorisation method (false positive or true positive) is also for this service too limited in specifying the perspective such a statement is made from. Table 2: False Positive - True Positive Comparison by Perspective therefore does not only show what category is a True or False positive from what perspective, but also suggested further steps to take after the alert was processed.

Suggested KPIs for a SOC are, besides reporting on what your actual coverage of the infrastructure is, for example "Number of legitimate violations authorized by change" or "Number of unauthorized changes without legitimate cause". Important KPIs for security management or tooling decision makers can be "Number of Limitation in verification products found" and "Number of activities with no change required". To determine the maturity of the secure configuration of systems and its attached compliance configuration setup, a KPI "Number of configuration errors in baseline" can be useful, associated to the causing system or product.

IV. Discussion

This categorisation is heavily based on the last paper published, referenced in the endnotes 1 and 2. The challenge is in teaching the function reviewing the responses from engineering teams (possibly SOC analysts) to detect what category is applicable, as well as teaching the engineering teams to add useful comments while responding to the alerts, so improvement steps can be derived from them. They should be therefore well introduced to be of use. The usage of such categorisation will become critical in an environment where monitoring process activities is only possible in a limited way (e.g. cloud environments) and security relies on secure configuration and zero trust setups.

This paper was created along with the poster "The Intelligent Process Lifecycle of Active Cyber Defenders"[4] showcasing categories of often seen as false alerts and illustration improvement actions that can be initiated from them.

V. Acknowledgment

Thank you to Claus Houmann for the original inspiration to extend the concept to more areas, as well as Gregor Bransky to create the opportunity to "outsource" some of the work that was put into creating the poster tying it all together. Thank you as well to Corsin Camichel, Florian Roth and Robert Schenk for reviewing this paper before publishing.

[1] https://github.com/d3sre/Use_Case_Applicability/blob/master/UseCaseApplicability-Paper.pdf

[2] https://dl.acm.org/doi/10.1145/3370084

[3] https://www.axelos.com/itil-4

[4] https://github.com/d3sre/IntelligentProcessLifecycle