OPEN ACCESS

Saudi Journal of Engineering and Technology
Abbreviated Key Title: Saudi J Eng Technol
ISSN 2415-6272 (Print) | ISSN 2415-6264 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com

Original Research Article

Alert Prioritization Techniques in Security Monitoring: A Focus on Severity Averaging and Alert Entities

Christian Bassey[1*], Samson Idowu[1], Courage Ojo[2]

[1]Department of Security and Network Engineering, Innopolis University, Russia
[2]Department of Computing, East Tennessee State University, U.S.A

DOI: 10.36348/sjet.2024.v09i07.008 | **Received:** 18.06.2024 | **Accepted:** 22.07.2024 | **Published:** 24.07.2024

***Corresponding author:** Christian Bassey
Department of Security and Network Engineering, Innopolis University, Russia

Abstract

Security monitoring is a crucial aspect of cybersecurity and a prong of organizational cybersecurity policies. It is achieved primarily using SIEM tools supported by logs ingested from intrusion detection tools and other security solutions. SIEM tools generate alerts of varying severities when detection rules identify anomalies or possible security incidents after analysis of ingested logs. These alerts need to be investigated, but due to the volume of alerts generated and the limited monitoring manhours, it is important to prioritize which security alerts are investigated first. This paper presents a sliding window technique for prioritizing security events by computing a priority value using the severity of previous alerts, alert entities, and criticality ratings. Findings from the experiment show that this approach improves the prioritization of security alerts with severe and medium alerts affecting critical systems prioritized over low, high, and critical alerts affecting non-critical systems. This work can potentially streamline and enhance the efficiency of security monitoring operations.

Keywords: SIEM, security monitoring, alert prioritization, critical systems.

1. INTRODUCTION

Security monitoring involves analyzing information and data to detect abnormal or malicious activity. These activities are termed incidents and need to be resolved. Security monitoring is a key strategy that supports threat detection forensics and verification of security controls (Gantz and Philpott, 2013). Monitoring focuses on detecting and responding to attacks and security incidents. There are various methods and tooling available for the detection of malicious activity. These detection methods typically involve ingesting system and network events, parsing and normalizing them, and then analyzing and correlating the events for malicious activity. The analysis and correlation of security events are typically done based on a set of detection rules, which may be heuristics, anomaly, or machine learning based in security tooling.

Fuentes-Garcia identified Intrusion Detection Systems (IDSs) / Intrusion Prevention Systems (IPSs), Security Event Management systems (SEMs) / Security Information and Event Management systems (SIEMs) systems, and Universal Threat Management systems (UTMs) as tooling used for network security monitoring (Fuentes-García *et al*., 2021). These tools all implement several modules that allow security monitoring, such as log parsing, normalization, event correlation, and malicious activity detection. Security engineers deploy and configure these security tools. In contrast, security analysts are responsible for monitoring the alerts generated by the tools, triaging them, validating their criticality, and escalating alerts where necessary (Vielberth *et al*., 2020). These security tools are often integrated with a SIEM tool to consolidate their logs and alerts, enhancing the SIEM's correlation and detection capabilities. Security analysts monitor the SIEM tool. The integration and centralization of security events from these tools introduce the challenges of high volumes of alerts being generated in a short period that must be triaged in equally short periods, leading to alert fatigue. Due to the volume of logs ingested by a SIEM, and the number of alerts generated with varying severities and affecting systems with various degrees of criticality, it is important to prioritize which alerts are investigated first (CREST, 2015).

Security monitoring is part of the core functions of a security operations center (SOC). As such, security monitoring is part of the roles and responsibilities of the SOC. Vielberth *et al*., derived three roles in a SOC - Tier 1 (triage specialist), Tier 2 (incident responder), and Tier 3 (threat hunter) analysts (Vielberth *et al*., 2020). Based on these roles, the Tier 1 analysts would review security alerts, validate their authenticity, determine if they are false positives, adjust the criticality, and then prioritize and escalate the events to the Tier 2 analysts who would proceed to analyze the incidents in detail and initiate incident response. The initial work of prioritizing alerts falls on the tier 1 analysts. So, alerts must be prioritized properly for handling before they are escalated. Current prioritization techniques are based on the criticality of the affected systems or entities, the severity of the security alert, and the adaptation of ML models for prioritization (Gelman *et al*., 2015).

By leveraging artificial intelligence, Ndichu *et al*., analyzed security alert data using imbalance learning methods for improved triaging and reduced false positives. The authors identified features in the security alerts, labeled them, and applied machine learning techniques to filter out noisy alerts and false positives (Ndichu, S *et al*., 2023).

It is important to prioritize alerts because not all high severity security alerts impact organizational security. Similarly, alerts of medium or low severity but occurring on an asset tagged as critical may have a devastating impact. Alert prioritization ensures that impactful security alerts are investigated first, especially in shared SOC monitoring contexts where information technology (IT) and operational technology (OT) assets are monitored side by side. This paper proposes an algorithm for prioritizing security alerts in security monitoring tools. Based on the alerts' severity and the affected systems' criticality, a sliding window computation of an alert priority is generated. This work enhances the triaging processes in security monitoring and optimizing incident handling processes.

2. MATERIALS AND METHODS

Currently, security alerts are dealt with using First in First Out (FIFO), Last in First Out (LIFO), or severity-based algorithms. The detection rule and parameters typically set the alert's severity. This implies that the priority in which an alert is treated is typically based on when the incident happened or the preconfigured detection rule. The criticality values of the entities in the alert generated and the impact of the detected incident on the associated entities' confidentiality, availability, or integrity are not evaluated in the initial event prioritization. What this means is that if an incident that may potentially impact service availability happens on an operational technology (OT) system - at 18:05 and the same incident happened on a user workstation at 18:04 of the same day, if the incidents have the same severity, then the incident that happened on the user workstation which is less impactful will get processed first. As a result, the more impactful incident will have more time to spawn and cause damage.

In this section, we propose a solution to prioritize the handling of security alerts based on the criticality rating of entities in the alert, the severity of preexisting open alerts associated with the affected entities, and the default severity of the alert just generated. Based on this information, a priority value is computed and assigned to the new alert. To perform this computation, it is necessary to define the variables that will be used in the mathematical computation of the priority and their numeric values:

2.1 Priority values

An alert's priority is rated from 1 (low) to 4 (critical). It is not preset and is the outcome of the algorithm.

Table 1: Priority values table

Priority	Numeric value
Critical	4
High	3
Medium	2
Low	1

2.2 Severity values

The severity of an incident is extracted from the alert generated in the SIEM. Different SIEM and security monitoring tools have different severity ratings. Severity is the rating of a security alert's impact on business processes. This research uses a benchmark range from 1 (low) to 4 (critical). The detection rules and SIEM configurations determine this value. If the SIEM system uses some other numerical rating that is not in the range of 1-4, it can be normalized using the formula below:

$$S_{research} = 1 + \frac{3 * S_{siem(present)}}{S_{siem(max)}}$$

Where:

$S_{research}$ is the severity level on a scale of 1-4

$S_{siem(present)}$ is the severity level of the present alert in the SIEM

$S_{siem(max)}$ is the maximum severity level in the SIEM

Table 2: Severity rating table

Severity	**Numeric value**
Critical	4
High	3
Medium	2
Low	1

2.3 Alert Entities

Alert entities are the identifiers in the alert data that can be leveraged to identify which assets have been impacted by the security alert (Fulfer, 2023). Examples of these entities include usernames, hostnames, IP addresses, locations, etc.

2.4 Criticality Values

The criticality rating is a value assigned to the entity by a system administrator. The system administrator computes this based on the impact a security incident affecting that entity will have on business processes, with a focus on confidentiality, integrity, and availability. This value is predetermined in advance.

Table 3: Criticality rating table

Criticality	**Numeric value**
Critical	4
High	3
Medium	2
Low	1
Total	10

2.5 Algorithm for Determining Alert Priority

We propose the following algorithm to determine the priority of a received alert and, thus, which alert should be handled first when competing concerns exist.

1. When an alert is generated, check for entities that can be used to compute the priority.
2. If there are no entities in the alert, set the criticality:
   $C_{unf} = 0.$
3. If there are entities in the alert
   a. If the entities do not have a criticality rating, set the criticality:
      $C_{unf} = 0.$
   b. If the entities have a criticality rating, compute the unified criticality for that alert as the average of all entities criticality ratings. i.e

$$C_{unf} = \frac{1}{n}\sum_{i=1}^{n} EC_i$$

Where:

EC_i is the individual impact value assigned to an entity by a system administrator.

C_{unf} is the unified criticality rating for that alert.

4. Check if there are pre-existing alerts related to the entities.
   a. If there are no pre-existing alerts compute the priority as the sum of the alert severity and the impact divided by 2 :

$$P = \frac{S_{now} + C_{unf}}{2}$$

Where:

P is the priority.

S_{now} is the severity of the present alert.

   b. If there are pre-existing open alerts, compute the priority as the arithmetic mean of the previous alerts' severities and the present alert severity:

$$P = \frac{\frac{1}{n+1}\left(\sum_{i=1}^{n} S_i + S_{now}\right) + C_{unf}}{2}$$

Where:

S_i is the severity of previous alerts with entities that are the same as the present alert.

5. Set the Alert priority based on the result of P.

Figure 1 below shows the flowchart for the algorithm.

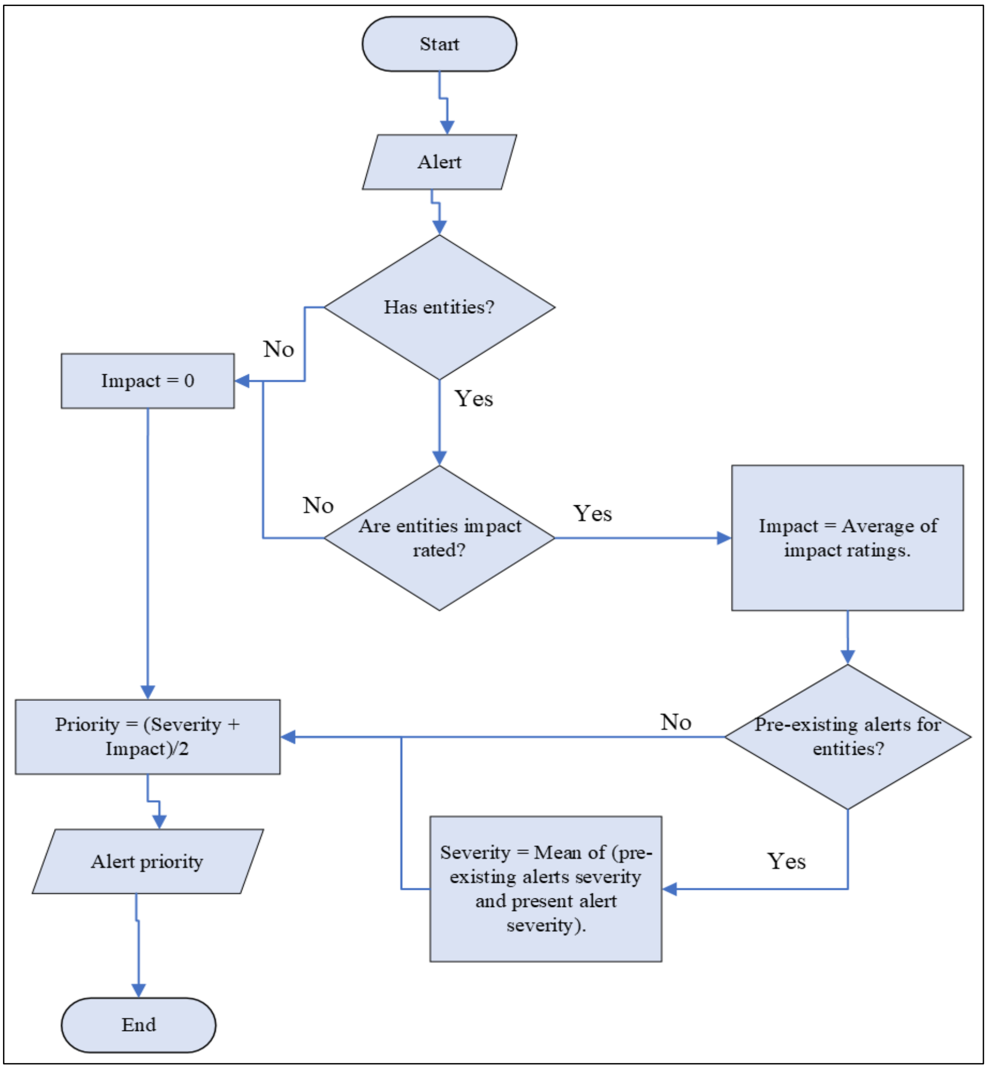
Start
Alert
Has entities?
No
Yes
Impact = 0
Are entities impact rated?
No
Yes
Impact = Average of impact ratings.
Pre-existing alerts for entities?
No
Yes
Priority = (Severity + Impact)/2
Severity = Mean of (pre-existing alerts severity and present alert severity).
Alert priority
End

Fig 1: Alert prioritization algorithm flowchart

2.6 Testing the Algorithm

To test the algorithm, a data set of security alerts was exported from a SIEM in a table and imported into Excel. The sections below describe the data set used for testing.

2.6.1 Test data set

The test data set contained nine alerts of varying severities generated within a 10-minute spread and three host entities. Each alert had one entity and its associated preset criticality ratings. Table 4 below shows the initial data set in the order they were exported from the SIEM

Table 4: Test data set

Timestamp	Alert	Severity	Host	Host entity criticality rating
13-04-2024 13:33	A1	Medium	H1	Critical
13-04-2024 13:34	A2	Low	H1	Critical
13-04-2024 13:35	A3	High	H1	Critical
13-04-2024 13:36	A4	Critical	H1	Critical
13-04-2024 13:37	A5	High	H2	Medium
13-04-2024 13:38	A6	Critical	H2	Medium
13-04-2024 13:39	A7	Medium	H2	Medium
13-04-2024 13:40	A8	Low	H2	Medium
13-04-2024 13:41	A9	Critical	H3	High

2.6.2 Sorted by severity and time

The original data set is sorted in chronological order, which means the alerts are analyzed in the order they were generated. However, it is important to note that analysts and SOCs have different monitoring, triaging, and operating principles. Hence, the data set could also be sorted by chronological order and event severity.

Table 5 shows the data set sorted by alert severity and time.

Table 5: Security alerts data set sorted by alert severity and time

Timestamp	Alert	Severity	Numeric severity	Host	Host entity criticality rating	Numeric criticality rating
13-04-2024 13:36	A4	Critical	4	H1	Critical	4
13-04-2024 13:38	A6	Critical	4	H2	Medium	2
13-04-2024 13:41	A9	Critical	4	H3	High	3
13-04-2024 13:35	A3	High	3	H1	Critical	4
13-04-2024 13:37	A5	High	3	H2	Medium	2
13-04-2024 13:33	A1	Medium	2	H1	Critical	4
13-04-2024 13:39	A7	Medium	2	H2	Medium	2
13-04-2024 13:34	A2	Low	1	H1	Critical	4
13-04-2024 13:40	A8	Low	1	H2	Medium	2

2.7 Implementing the algorithm

The algorithm was implemented in Excel. The steps taken are as follows:

1. Import the alert data into Excel.
2. Sort the data in chronological order using the timestamp column.
3. Add a column for the numeric severity value.
4. Add a column for the numeric criticality rating value.
5. Add a column for the priority rating value.
6. For each event, compute the priority based on the proposed algorithm outlined in section 2.5.

3. RESULTS AND DISCUSSION

The results of the computation of the alert priority values are shown in Table 6 below.

Table 6: Security alerts and priority values

Timestamp	Alert	Severity	Host	Host entity rating	Criticality rating	Numeric severity	Priority
13-04-2024 13:41	A9	Critical	H3	High	3	4	3.5
13-04-2024 13:36	A4	Critical	H1	Critical	4	4	3.25
13-04-2024 13:33	A1	Medium	H1	Critical	4	2	3
13-04-2024 13:35	A3	High	H1	Critical	4	3	3
13-04-2024 13:34	A2	Low	H1	Critical	4	1	2.75
13-04-2024 13:38	A6	Critical	H2	Medium	2	4	2.75
13-04-2024 13:37	A5	High	H2	Medium	2	3	2.5
13-04-2024 13:39	A7	Medium	H2	Medium	2	2	2.5
13-04-2024 13:40	A8	Low	H2	Medium	2	1	2.25

The results show that Alert A9 with a critical severity with an entity H3 with a high criticality rating has been given the highest priority for investigation. This is partly based on the fact that the computation for its priority is a direct averaging of its severity and criticality values due to the absence of preceding alerts. Note that A9 is investigated third in severity-based triaging and last in chronological-based triaging.

A4 is the next alert in order of computed priority. The priority considers the severity ratings of alerts A1, A3, and A2 to compute the priority value of 3.25 which is an approximate high priority. A9 is followed by A1, A3, and A2 alerts respectively. The associated entity in these alerts is the host H1. It should be noted that the entity criticality rating for that host is critical.

The last set of alerts in the order of priority are A6, A5, A7 and A8. Host H2 is the associated entity in this set of alerts. H2 has a criticality rating of medium. In the initial alert data, A5 and A6 are triaged fifth and sixth respectively; however, after applying the algorithm, they are triaged seventh and sixth. A6 ties with A2 in order of priority. In such a situation, recourse can be made to old methods of investigating based on chronological order or severity. In the results we have fallen back to chronological order. Alert A8 is the last in order of priority based on it being a low-severity alert on a medium criticality device.

4. CONCLUSION

In this work we have demonstrated how security alerts in SIEMs and other security monitoring solutions can be prioritized for investigation and triaging based on the entities in the alerts, the criticality ratings of those entities, and the severity of preceding alerts with the same entities. This algorithm considers that the impact of a breach on all systems is not the same and where there are a large amount of security alerts, critical system alerts must be prioritized first for investigation. It is worthy of note that after the computation of the priority, low alerts on critical systems even with preceding alerts were given the same priority as the critical alerts on medium-rated systems. This points to the tactical balancing inherent in the algorithm.

For future work, open-source security monitoring systems can be extended to include this priority computation algorithm for real-time monitoring and benchmarking to determine how well it streamlines alert prioritization and operational SOC.

Data availability: The data is available on reasonable request.

Funding: Not applicable.

Author Contributions: All authors contributed equally to the experiments, writing and reviewing the manuscript.

REFERENCES

- Gantz, S. D., & Philpott, D. R. (2013). Chapter 14 - Continuous Monitoring. In S. D. Gantz & D. R. Philpott (Eds.), *FISMA and the Risk Management Framework* (pp. 367-401). Syngress. https://doi.org/10.1016/B978-1-59-749641-4.00014-X
- Fuentes-García, N. M., Camacho, J., & Maciá-Fernández, G. (2021). Present and Future of Network Security Monitoring. *IEEE Access*, 9, 112744-112760. https://doi.org/10.1109/ACCESS.2021.3067106
- Vielberth, M., Bohm, F., Fichtinger, I., & Pernul, G. (2020). Security Operations Center: A Systematic Study and Open Challenges. *IEEE Access*, 8, 227756–227779. https://doi.org/10.1109/access.2020.3045514
- Council of Registered Ethical Security Testers. (2015). Cyber Security Monitoring and Logging Guide. CREST. https://www.crest-approved.org/wp-content/uploads/2022/04/Cyber-Security-Monitoring-Guide.pdf
- Gelman, B., Taoufiq, S., Vörös, T., & Berlin, K. (2023). That Escalated Quickly: An ML Framework for Alert Prioritization. ArXiv. /abs/2302.06648
- Ndichu, S., Ban, T., Takahashi, T., & Inoue, D. (2023). AI-Assisted Security Alert Data Analysis with Imbalanced Learning Methods. *Applied Sciences*, *13*(3), 1977. https://doi.org/10.3390/app13031977
- National Cybersecurity Authority, Kingdom of Saudi Arabia. (2019). Critical Systems Cybersecurity Controls. Retrieved July 14, 2024, from https://nca.gov.sa/ar/cscc-en.pdf
- Fulfer, T. (2023). Cybersecurity breaches: Understanding the different severity levels. Aldridge. Retrieved July 14, 2024, from https://aldridge.com/understanding-the-different-severity-levels/
- Computer Security Resource Center. (n.d.). Entity - glossary: CSRC. National Institute of Standards and Technology. Retrieved from https://csrc.nist.gov/glossary/term/entity
- Sheeraz, M., Paracha, M. A., Haque, M. U., Durad, M. H., Mohsin, S. M., Band, S. S., & Mosavi, A. (2023). Effective Security Monitoring Using Efficient SIEM Architecture. *Human-Centric Computing and Information Sciences*, 13, 1-18.
- Ban, T., Takahashi, T., Ndichu, S., & Inoue, D. (2023). Breaking Alert Fatigue: AI-Assisted SIEM Framework for Effective Incident Response. *Applied Sciences*, *13*(11), 6610. https://doi.org/10.3390/app13116610