# Security Monitoring & SIEM Fundamentals

## SIEM & SOC Fundamentals

### SIEM Definition & Fundamentals

What Is SIEM?

- Crucial within the realm of computer protection, Security Information and Event Management (SIEM) encompasses the utilization of software offerings and solutions that merge the management of security data with the supervision of security events.
  - These instruments facilitate real-time evaluations of alerts related to security, which are produced by network hardware and applications.
- SIEM tools possess an extensive range of core functionalities, such as the collection and administration of log events, the capacity to examine log events and supplementary data from various sources, as well as operational features like incident handling, visual summaries, and documentation.
- Employing SIEM innovations, IT personnel can detect cyberattacks at the time of or even prior to their occurrence, thereby enhancing the speed of their response during incident resolution.
  - Consequently, SIEM plays an indispensable role in the effectiveness and ongoing supervision of a company's information security framework.
  - It serves as the bedrock of an organization's security tactics, offering a holistic method for identifying and managing potential threats.

The Evolution Of SIEM Technology

- The acronym "SIEM" emerged from the collaboration of two Gartner analysts who suggested a novel security information framework that integrated two preceding technologies: Security Information Management (SIM) and Security Event Management (SEM).
  - This proposition appeared in a 2005 Gartner paper titled "Enhance IT Security through Vulnerability Management."
- First-generation SIM technology was developed upon conventional log collection management systems, allowing for extended storage, examination, and reporting of log data while incorporating logs with threat intelligence.
  - Conversely, the second-generation SEM technology tackled security events by delivering consolidation, correlation, and notification of events from a range of

security apparatuses, such as antivirus software, firewalls, Intrusion Detection Systems (IDS), in addition to events disclosed directly by authentication, SNMP traps, servers, and databases.

- In the years that followed, vendors amalgamated the capabilities of SIM and SEM to devise the SIEM, leading to a fresh definition as per Gartner's investigation.
    - This nascent technology gained widespread acceptance as it offered a comprehensive methodology for detecting and managing threats, including the ability to amass, preserve, and scrutinize logs and security events from various origins.

How Does A SIEM Solution Work?

- SIEM systems function by gathering data from a variety of sources, including PCs, network devices, servers, and more.
    - This data is then standardized and consolidated to facilitate ease of analysis.
- SIEM platforms employ security experts who scrutinize the data in order to identify and detect potential threats.
    - This procedure allows businesses to locate security breaches and examine alerts, offering crucial insights into the organization's security standing.
- Alerts notify Security Operations/Monitoring personnel that they must look into a (possible) security event or incident.
    - These notifications are usually concise and inform staff of a specific attack targeting the organization's information systems.
    - Alerts can be conveyed through multiple channels, such as emails, console pop-up messages, text messages, or phone calls to smartphones.
- SIEM systems generate a vast number of alerts owing to the substantial volume of events produced for each monitored platform.
    - It is not unusual for an hourly log of events to range from hundreds to thousands.
    - As a result, fine-tuning the SIEM for detecting and alerting on high-risk events is crucial.
- The capacity to accurately pinpoint high-risk events is what distinguishes SIEM from other network monitoring and detection tools, such as Intrusion Prevention Systems (IPS) or Intrusion Detection Systems (IDS).
    - SIEM does not supplant the logging capabilities of these devices; rather, it operates in conjunction with them by processing and amalgamating their log data to recognize events that could potentially lead to system exploitation.
    - By integrating data from numerous sources, SIEM solutions deliver a holistic strategy for threat detection and management.

# SIEM Business Requirements & Use Cases

## Log Aggregation & Normalization

- The importance of threat visibility through log consolidation offered by SIEM systems cannot be overstated.
  - In its absence, an organization's cybersecurity holds as much value as a mere paperweight.
  - Log consolidation entails gathering terabytes of security information from vital firewalls, confidential databases, and essential applications.
  - This process empowers the SOC team to examine the data and discern connections, significantly improving threat visibility.
- Utilizing SIEM log consolidation, the SOC team can identify and scrutinize security incidents and events throughout the organization's IT infrastructure.
  - By centralizing and correlating information from various sources, SIEM delivers a holistic strategy for threat detection and handling.
  - This approach allows organizations to recognize patterns, tendencies, and irregularities that could suggest potential security hazards.
  - Consequently, SOC teams can react promptly and efficiently to security incidents, reducing the repercussions on the organization.

## Threat Alerting

- Having a SIEM solution that can identify and notify IT security teams about possible threats within the vast volume of collected security event data is essential.
  - This feature is critical as it allows the IT security team to carry out swifter, more targeted investigations and respond to potential security incidents in a timely and efficient manner.
- Advanced analytics and threat intelligence are employed by SIEM solutions to recognize potential threats and generate real-time alerts.
  - When a threat is detected, the system forwards alerts to the IT security team, equipping them with the necessary details to effectively investigate and mitigate the risk.
  - By alerting IT security teams promptly, SIEM solutions aid in minimizing the potential impact of security incidents and safeguarding the organization's vital assets.

## Contextualization & Response

- It is important to understand that merely generating alerts is not enough.
    - If a SIEM solution sends alerts for every possible security event, the IT security team will soon be overwhelmed by the sheer volume of alerts, and false positives may become a frequent issue, particularly in older solutions.
        - As a result, threat contextualization is crucial for sorting through alerts, determining the actors involved in the security event, the affected parts of the network, and the timing.
- Contextualization enables IT security teams to identify genuine potential threats and act swiftly.
    - Automated configuration processes can filter some contextualized threats, reducing the number of alerts received by the team.
- An ideal SIEM solution should allow an enterprise to directly manage threats, often by stopping operations while investigations take place.
    - This approach helps to minimize the potential impact of security incidents and protect the organization's critical assets.
    - SIEM solutions provide context and automate threat filtering, allowing IT security teams to concentrate on genuine threats, reducing alert fatigue, and enhancing the efficiency and effectiveness of incident response.

Compliance

- SIEM solutions play a significant role in compliance by assisting organizations in meeting regulatory requirements through a comprehensive approach to threat detection and management.
- Regulations like PCI DSS, HIPAA, and GDPR mandate organizations to implement robust security measures, including real-time monitoring and analysis of network traffic.
    - SIEM solutions can help organizations fulfill these requirements, enabling SOC teams to detect and respond to security incidents promptly.
- Automated reporting and auditing capabilities are also provided by SIEM solutions, which are essential for compliance.
    - These features allow organizations to produce compliance reports swiftly and accurately, ensuring that they satisfy regulatory requirements and can demonstrate compliance to auditors and regulators.

Data Flows Within A SIEM

- Let us now briefly see how data travel within a SIEM, until they are ready for analysis.
    1. SIEM solutions ingest logs from various data sources.

    1. Each SIEM tool possesses unique capabilities for collecting logs from different sources. This process is known as data ingestion or data collection.
2. The gathered data is processed and normalized to be understood by the SIEM correlation engine.
    - The raw data must be written or read in a format that can be comprehended by the SIEM and converted into a common format from various types of datasets.
    - This process is called data normalization and data aggregation.
3. Finally, the most crucial part of SIEM, where SOC teams utilize the normalized data collected by the SIEM to create various detection rules, dashboards, visualizations, alerts, and incidents.
    - This enables the SOC team to identify potential security risks and respond swiftly to security incidents.

What Are The Benefits Of Using A SIEM Solution

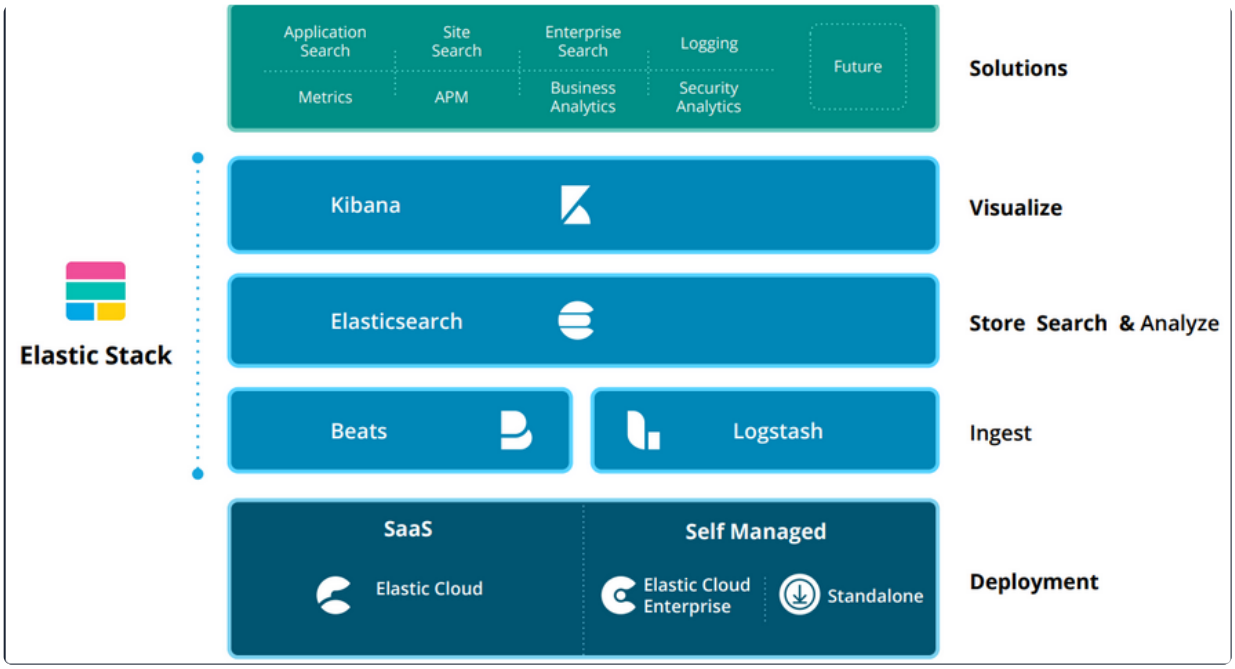
- It is evident that the advantages of deploying a Security Information and Event Management (SIEM) system significantly outweigh the potential risks associated with not having one, assuming that the security control is safeguarding something of higher importance.
- In the absence of a SIEM, IT personnel would not have a centralized perspective on all logs and events, which could result in overlooking crucial events and accumulating a large number of events awaiting investigation.
    - Conversely, a properly calibrated SIEM bolsters the incident response process, improving efficiency and offering a centralized dashboard for notifications based on predetermined categories and event thresholds.
- For instance, if a firewall records five successive incorrect login attempts, resulting in the admin account being locked, a centralized logging system that correlates all logs is necessary for monitoring the situation.
    - Similarly, a web filtering software that logs a computer connecting to a malicious website 100 times in an hour can be viewed and acted upon within a single interface using a SIEM.
- Contemporary SIEMs often include built-in intelligence capable of detecting configurable threshold limits and events within specific timeframes, as well as providing summaries and customizable reports.
    - More sophisticated SIEMs are now integrating AI to notify based on behavioral and pattern analysis.
- The reporting and notification capabilities of a SIEM empower IT staff to swiftly react and respond to potential incidents, emphasizing its ability to identify malicious attacks

before they occur.
    - This intelligence can lower the expenses associated with a full-scale security breach, sparing organizations significant financial and reputational harm.
- Numerous regulated organizations, such as those in Banking, Finance, Insurance, and Healthcare, are mandated to have a managed SIEM either on-premise or in the cloud.
    - SIEM systems offer evidence that systems are being monitored and logged, reviewed, and adhere to log retention policies, fulfilling compliance standards like ISO and HIPAA.

### Introduction To The Elastic Stack

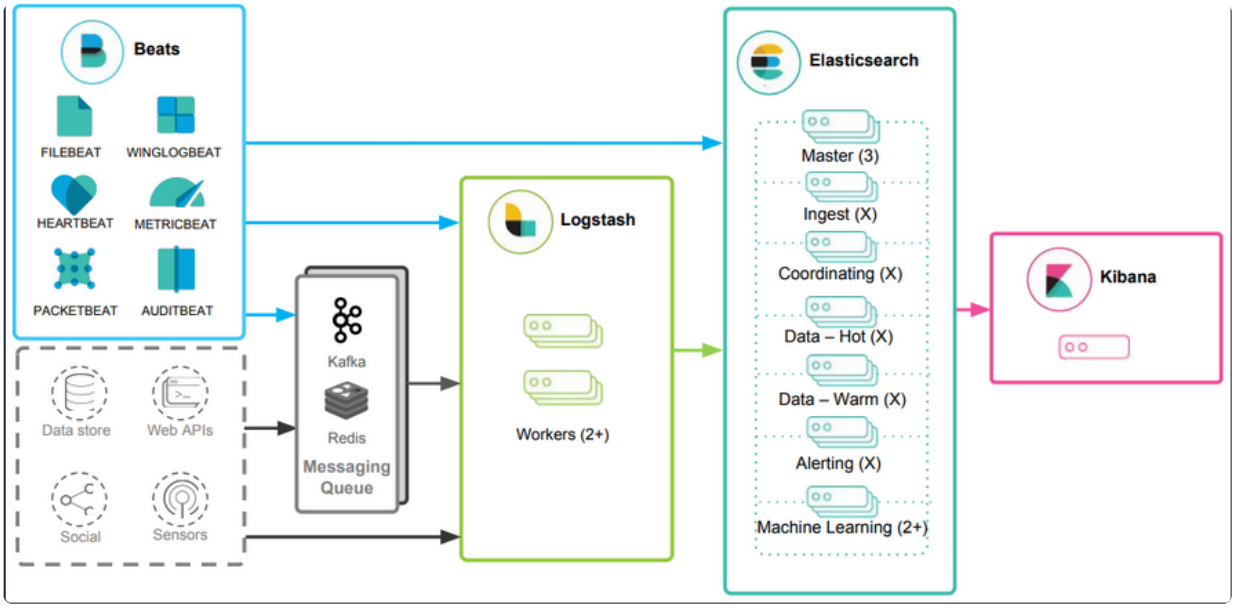
What Is The Elastic Stack?

- The Elastic stack, created by Elastic, is an open-source collection of mainly three applications (Elasticsearch, Logstash, and Kibana) that work in harmony to offer users comprehensive search and visualization capabilities for real-time analysis and exploration of log file sources.




- The high-level architecture of the Elastic stack can be enhanced in resource-intensive environments with the addition of Kafka, RabbitMQ, and Redis for buffering and

resiliency, and nginx for security.

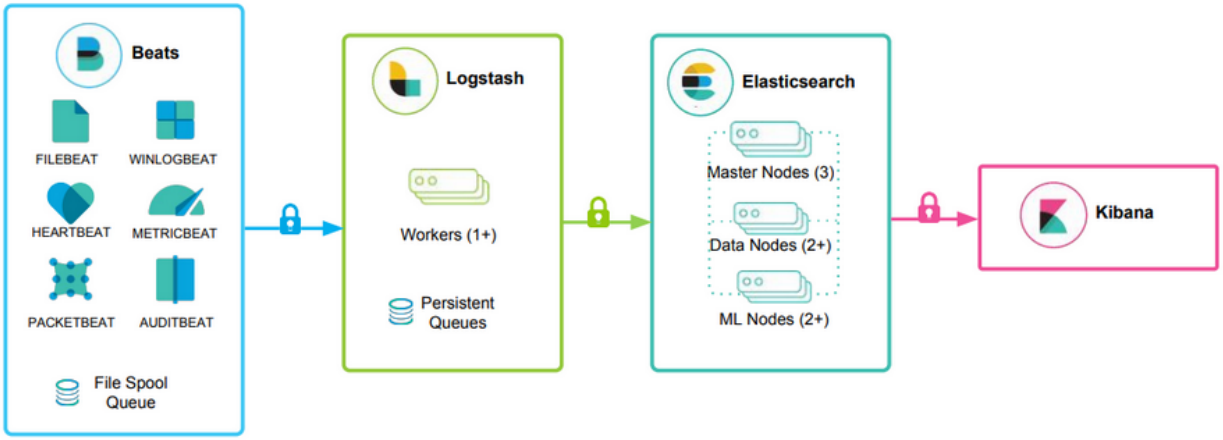



- Let's delve into each component of the Elastic stack.
- `Elasticsearch` is a distributed and JSON-based search engine, designed with RESTful APIs.
  - As the core component of the Elastic stack, it handles indexing, storing, and querying.
  - Elasticsearch empowers users to conduct sophisticated queries and perform analytics operations on the log file records processed by Logstash.
- `Logstash` is responsible for collecting, transforming, and transporting log file records.
  - Its strength lies in its ability to consolidate data from various sources and normalize them. Logstash operates in three main areas:
  1. `Process input`: Logstash ingests log file records from remote locations, converting them into a format that machines can understand.
     - It can receive records through different input methods, such as reading from a flat file, a TCP socket, or directly from syslog messages. After processing the input, Logstash proceeds to the next function.
  2. `Transform and enrich log records`: Logstash offers numerous ways to modify a log record's format and even content.
     - Specifically, filter plugins can perform intermediary processing on an event, often based on a predefined condition.
     - Once a log record is transformed, Logstash processes it further.
  3. `Send log records to Elasticsearch`: Logstash utilizes output plugins to transmit log records to Elasticsearch.
- `Kibana` serves as the visualization tool for Elasticsearch documents.
  - Users can view the data stored in Elasticsearch and execute queries through Kibana.

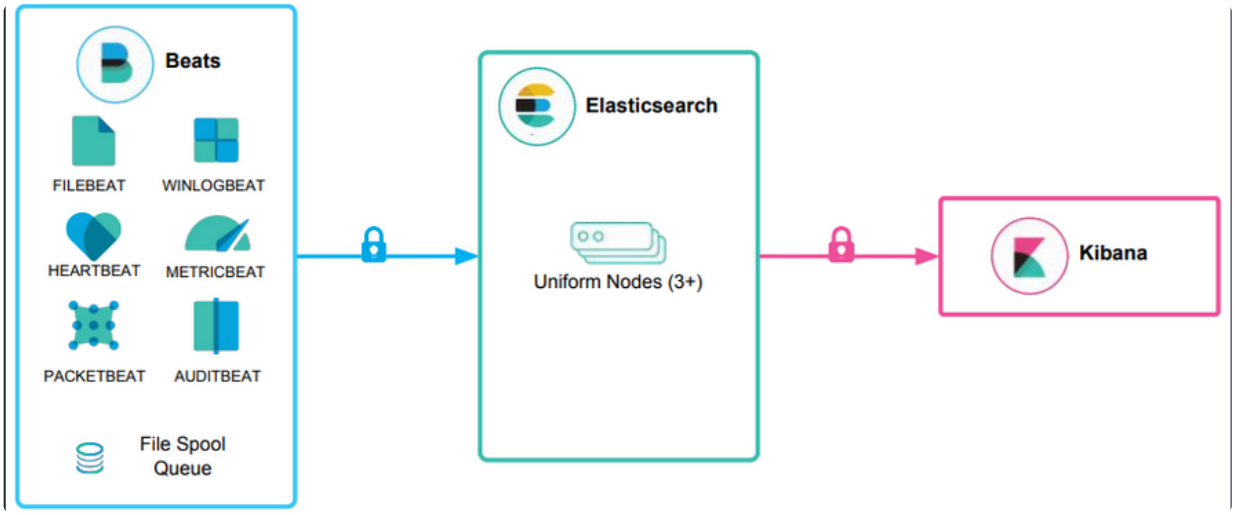
- Additionally, Kibana simplifies the comprehension of query results using tables, charts, and custom dashboards.

-> Note: `Beats` is an additional component of the Elastic stack.
-> These lightweight, single-purpose data shippers are designed to be installed on remote machines to forward logs and metrics to either Logstash or Elasticsearch directly.
-> Beats simplify the process of collecting data from various sources and ensure that the Elastic Stack receives the necessary information for analysis and visualization.

- `Beats` -> `Logstash` -> `Elasticsearch` -> `Kibana`




- `Beats` -> `Elasticsearch` -> `Kibana`



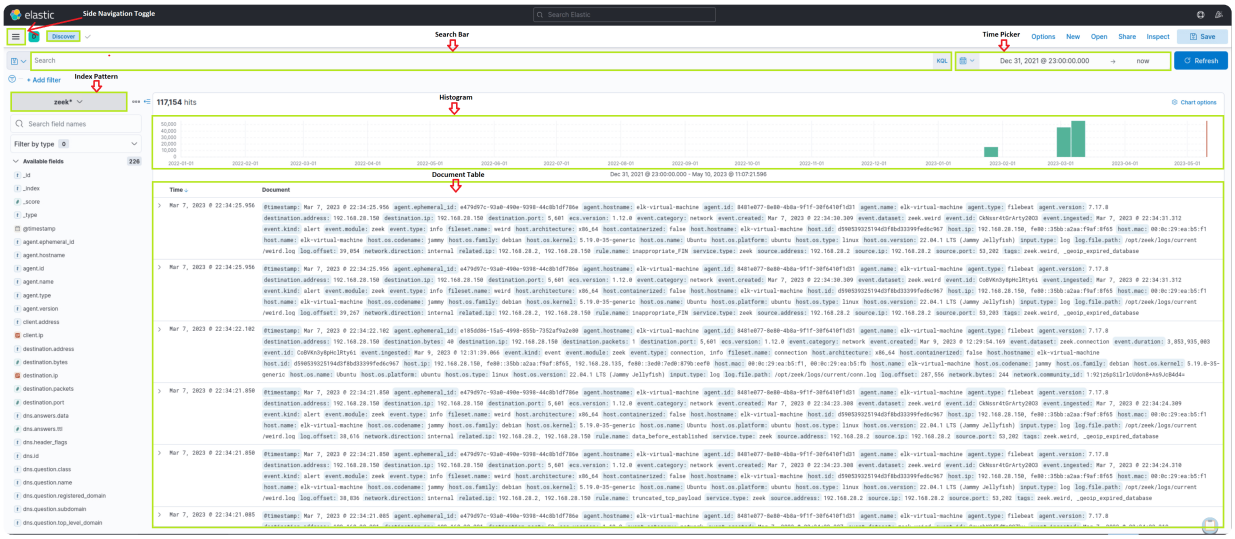

The Elastic Stack As A SIEM Solution

- The Elastic stack can be used as a Security Information and Event Management (SIEM) solution to collect, store, analyze, and visualize security-related data from various sources.
- To implement the Elastic stack as a SIEM solution, security-related data from various sources such as firewalls, IDS/IPS, and endpoints should be ingested into the Elastic

stack using Logstash.

  - Elasticsearch should be configured to store and index the security data, and Kibana should be used to create custom dashboards and visualizations to provide insights into security-related events.

- To detect security-related incidents, Elasticsearch can be used to perform searches and correlations on the collected security data.
- As Security Operations Center (SOC) analysts, we are likely to extensively use Kibana as our primary interface when working with the Elastic stack.
  - Therefore, it is essential to become proficient with its functionalities and features.




- Kibana Query Language (KQL) is a powerful and user-friendly query language designed specifically for searching and analyzing data in Kibana.
  - It simplifies the process of extracting insights from your indexed Elasticsearch data, offering a more intuitive approach than Elasticsearch's Query DSL.
  - Let's explore the technical aspects and key components of the KQL language.
    - `Basic Structure`: KQL queries are composed of `field:value` pairs, with the field representing the data's attribute and the value representing the data you're searching for. For example:

```
event.code:4625
```

- The KQL query `event.code:4625` filters data in Kibana to show events that have the Windows event code 4625.
  - This Windows event code is associated with failed login attempts in a Windows operating system.
- By using this query, SOC analysts can identify failed login attempts on Windows machines within the Elasticsearch index, and investigate the source of the attempts and potential security threats.

    - This type of query can help identify brute force attacks, password guessing, and other suspicious activities related to login attempts on Windows systems.
- By further refining the query with additional conditions, such as the source IP address, username, or time range, SOC analysts can gain more specific insights and effectively investigate potential security incidents.
- `Free Text Search`: KQL supports free text search, allowing you to search for a specific term across multiple fields without specifying a field name.
    - For instance:

```
"svc-sql1"
```

- This query returns records containing the string "svc-sql1" in any indexed field.
- `Logical Operators`: KQL supports logical operators AND, OR, and NOT for constructing more complex queries.
    - Parentheses can be used to group expressions and control the order of evaluation. For example:

```
event.code:4625 AND winlog.event_data.SubStatus:0xC0000072
```

- The KQL query `event.code:4625 AND winlog.event_data.SubStatus:0xC0000072` filters data in Kibana to show events that have the Windows event code 4625 (failed login attempts) and the SubStatus value of 0xC0000072.
- In Windows, the SubStatus value indicates the reason for a login failure.
    - A SubStatus value of 0xC0000072 indicates that the account is currently disabled.
- By using this query, SOC analysts can identify failed login attempts against disabled accounts.
    - Such a behavior requires further investigation, as the disabled account's credentials may have been identified somehow by an attacker.
- `Comparison Operators`: KQL supports various comparison operators such as `:`, `:>`, `:>=`, `:<`, `:<=`, and `:!`.
    - These operators enable you to define precise conditions for matching field values. For instance:

```
event.code:4625 AND winlog.event_data.SubStatus:0xC0000072 AND
@timestamp >= "2023-03-03T00:00:00.000Z" AND @timestamp <= "2023-03-
06T23:59:59.999Z"
```

- By using this query, SOC analysts can identify failed login attempts against disabled accounts that took place between March 3rd 2023 and March 6th 2023
- `Wildcards and Regular Expressions`: KQL supports wildcards and regular expressions to search for patterns in field values.
    - For example:

```
event.code:4625 AND user.name: admin*
```

- The Kibana KQL query `event.code:4625 AND user.name: admin*` filters data in Kibana to show events that have the Windows event code 4625 (failed login attempts) and where the username starts with "admin", such as "admin", "administrator", "admin123", etc.
- This query (if extended) can be useful in identifying potentially malicious login attempts targeted at administrator accounts.

How To Identify The Available Data

- "How can I identify the available fields and values?", you may ask.
    - Let's see how we could have identified the available fields and values that we used in this section.

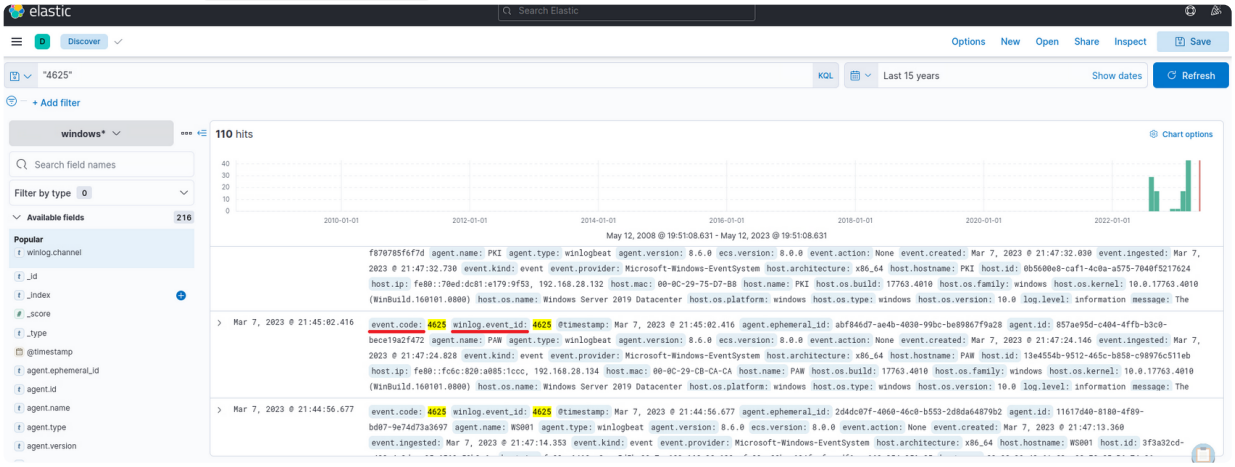
**Example**: Identify failed login attempts against disabled accounts that took place between March 3rd 2023 and March 6th 2023 KQL:

```
event.code:4625 AND winlog.event_data.SubStatus:0xC0000072 AND
@timestamp >= "2023-03-03T00:00:00.000Z" AND @timestamp <= "2023-03-
06T23:59:59.999Z"
```

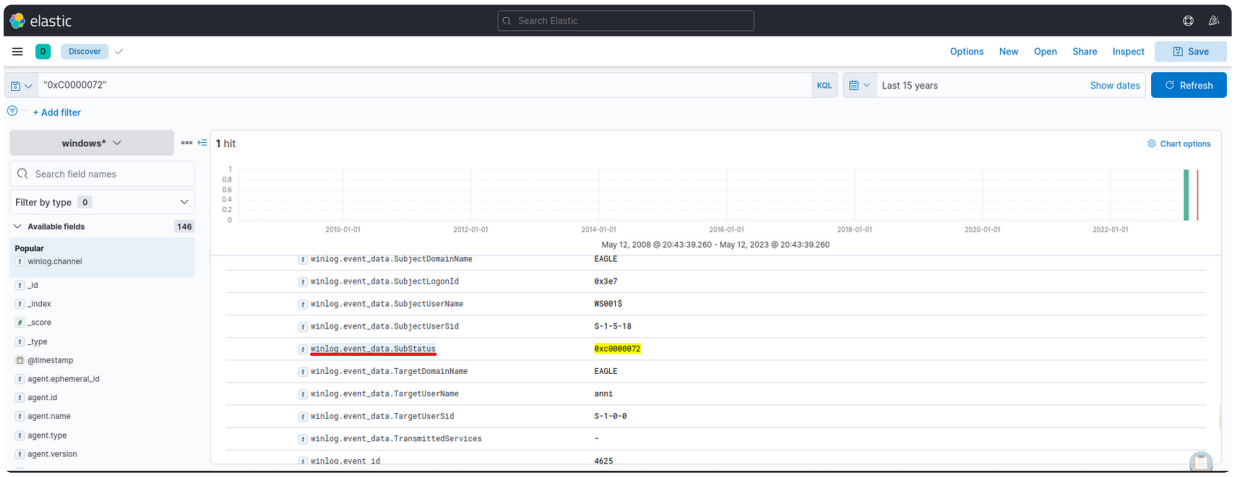
**Data and field identification approach 1: Leverage KQL's free text search**

- Using the Discover feature, we can effortlessly explore and sift through the available data, as well as gain insights into the architecture of the available fields, before we start constructing KQL queries.
- By using a search engine for the Windows event logs that are associated with failed login attempts, we will come across resources such as https://www.ultimatewindowssecurity.com/securitylog/encyclopedia/event.aspx?eventid=4625

- Using KQL's free text search we can search for `"4625"`. In the returned records we notice `event.code:4625`, `winlog.event_id:4625`, and `@timestamp`
  - `event.code` is related to the Elastic Common Schema (ECS)
  - `winlog.event_id` is related to Winlogbeat
  - If the organization we work for is using the Elastic stack across all offices and security departments, it is preferred that we use the ECS fields in our queries for reasons that we will cover at the end of this section.
  - `@timestamp` typically contains the time extracted from the original event and it is different from `event.created`




- when it comes to disabled accounts, the aforementioned resource informs us that a SubStatus value of 0xC0000072 inside a 4625 Windows event log indicates that the account is currently disabled.
  - Again using KQL's free text search we can search for `"0xC0000072"`. By expanding the returned record we notice `winlog.event_data.SubStatus` that is related to Winlogbeat




**Data and field identification approach 2: Leverage Elastic's documentation**

- It could be a good idea to first familiarize ourselves with Elastic's comprehensive documentation before delving into the "Discover" feature.
  - The documentation provides a wealth of information on the different types of fields we may encounter.

- Some good resources to start with are:
  - Elastic Common Schema (ECS)
  - Elastic Common Schema (ECS) event fields
  - Winlogbeat fields
  - Winlogbeat ECS fields
  - Winlogbeat security module fields
  - Filebeat fields
  - Filebeat ECS fields

The Elastic Common Schema (ECS)

- Elastic Common Schema (ECS) is a shared and extensible vocabulary for events and logs across the Elastic Stack, which ensures consistent field formats across different data sources.
  - When it comes to Kibana Query Language (KQL) searches within the Elastic Stack, using ECS fields presents several key advantages:
- `Unified Data View`: ECS enforces a structured and consistent approach to data, allowing for unified views across multiple data sources.
  - For instance, data originating from Windows logs, network traffic, endpoint events, or cloud-based data sources can all be searched and correlated using the same field names.
- `Improved Search Efficiency`: By standardizing the field names across different data types, ECS simplifies the process of writing queries in KQL.
  - This means that analysts can efficiently construct queries without needing to remember specific field names for each data source.
- `Enhanced Correlation`: ECS allows for easier correlation of events across different sources, which is pivotal in cybersecurity investigations.
  - For example, you can correlate an IP address involved in a security incident with network traffic logs, firewall logs, and endpoint data to gain a more comprehensive understanding of the incident.
- `Better Visualizations`: Consistent field naming conventions improve the efficacy of visualizations in Kibana.
  - As all data sources adhere to the same schema, creating dashboards and visualizations becomes easier and more intuitive.
  - This can help in spotting trends, identifying anomalies, and visualizing security incidents.
- `Interoperability with Elastic Solutions`: Using ECS fields ensures full compatibility with advanced Elastic Stack features and solutions, such as Elastic

Security, Elastic Observability, and Elastic Machine Learning.
    - This allows for advanced threat hunting, anomaly detection, and performance monitoring.
- `Future-proofing`: As ECS is the foundational schema across the Elastic Stack, adopting ECS ensures future compatibility with enhancements and new features that are introduced into the Elastic ecosystem.

### SOC Definition & Fundamentals

What Is A SOC?

- A Security Operations Center (SOC) is an essential facility that houses a team of information security experts responsible for continuously monitoring and evaluating an organization's security status.
    - The main objective of a SOC team is to identify, examine, and address cybersecurity incidents by employing a mix of technology solutions and a comprehensive set of procedures.
- The SOC team usually consists of proficient security analysts, engineers, and managers overseeing security operations.
    - They collaborate closely with organizational incident response teams to guarantee security concerns are promptly detected and resolved.
- Various technology solutions, such as Security Information and Event Management (SIEM) systems, Intrusion Detection and Prevention Systems (IDS/IPS), and Endpoint Detection and Response (EDR) tools, are utilized by the SOC team to monitor and identify security threats.
    - They also make use of threat intelligence and engage in threat hunting initiatives to proactively detect potential threats and vulnerabilities.
- Besides employing technology solutions, the SOC team follows a series of well-defined processes for addressing security incidents.
    - These processes encompass incident triage, containment, elimination, and recovery.
    - The SOC team cooperates closely with the incident response team to ensure proper handling of security incidents, safeguarding the organization's security stance.
- In summary, a SOC is a vital element of an organization's cybersecurity approach.
    - It offers continuous monitoring and response capabilities, enabling organizations to promptly detect and address security incidents, minimizing the consequences of a security breach and decreasing the likelihood of future attacks.

How Does A SOC Work?

- The primary function of the SOC team is to manage the ongoing operational aspect of enterprise information security rather than concentrating on the development of security strategies, designing security architecture, or implementing protective measures.
- The SOC team mainly consists of security analysts who work collectively to detect, assess, respond to, report on, and prevent cybersecurity incidents.
- Besides the primary responsibilities of a SOC team, some SOCs may possess advanced capabilities like forensic analysis and malware analysis.
  - These abilities enable the SOC team to conduct in-depth investigations of security incidents and examine the root cause of the incident to avert future attacks.
- As previously mentioned, the SOC team also collaborates closely with the incident response team to guarantee proper handling of security incidents and the preservation of the organization's security posture.

Roles Within A SOC

- A SOC team consists of diverse roles responsible for handling the continuous, operational aspect of enterprise information security.
  - These roles may encompass:
  - `SOC Director`: Responsible for overall management and strategic planning of the SOC, including budgeting, staffing, and alignment with organizational security objectives.
  - `SOC Manager`: Oversees day-to-day operations, manages the team, coordinates incident response efforts, and ensures smooth collaboration with other departments.
  - `Tier 1 Analyst`: Monitors security alerts and events, triages potential incidents, and escalates them to higher tiers for further investigation.
  - `Tier 2 Analyst`: Performs in-depth analysis of escalated incidents, identifies patterns and trends, and develops mitigation strategies to address security threats.
  - `Tier 3 Analyst`: Provides advanced expertise in handling complex security incidents, conducts threat hunting activities, and collaborates with other teams to improve the organization's security posture.
  - `Detection Engineer`: A Detection Engineer is responsible for developing, implementing, and maintaining detection rules and signatures for security monitoring tools, such as SIEM, IDS/IPS, and EDR solutions.

    - They work closely with security analysts to identify gaps in detection coverage and continuously improve the organization's ability to detect and respond to threats.
- `Incident Responder`: Takes charge of active security incidents, carries out in-depth digital forensics and containment and remediation efforts, and collaborates with other teams to restore affected systems and prevent future occurrences.
- `Threat Intelligence Analyst`: Gathers, analyzes, and disseminates threat intelligence data to help SOC team members better understand the threat landscape and proactively defend against emerging risks.
- `Security Engineer`: Develops, deploys, and maintains security tools, technologies, and infrastructure, and provides technical expertise to the SOC team.
- `Compliance and Governance Specialist`: Ensures that the organization's security practices and processes adhere to relevant industry standards, regulations, and best practices, and assists with audit and reporting requirements.
- `Security Awareness and Training Coordinator`: Develops and implements security training and awareness programs to educate employees about cybersecurity best practices and promote a culture of security within the organization.
- It is important to note that the specific roles and responsibilities within each tier can vary depending on the organization's size, industry, and specific security requirements.
- In general, the tiered structure can be described as follows:
    - `Tier 1 Analysts`: Also known as "first responders," these analysts monitor security events and alerts, perform initial triage, and escalate potential incidents to higher tiers for further investigation.
        - Their main goal is to quickly identify and prioritize security incidents.
    - `Tier 2 Analysts`: These analysts are more experienced and perform deeper analysis of escalated incidents.
        - They identify patterns and trends, develop mitigation strategies, and sometimes assist in incident response efforts.
        - They may also be responsible for tuning security monitoring tools to reduce false positives and improve detection capabilities.
    - `Tier 3 Analysts`: Often considered the most experienced and knowledgeable analysts on the team, Tier 3 analysts handle the most complex and high-profile security incidents.
        - They may also engage in proactive threat hunting, develop advanced detection and prevention strategies, and collaborate with other teams to improve the organization's overall security posture.

SOC Stages

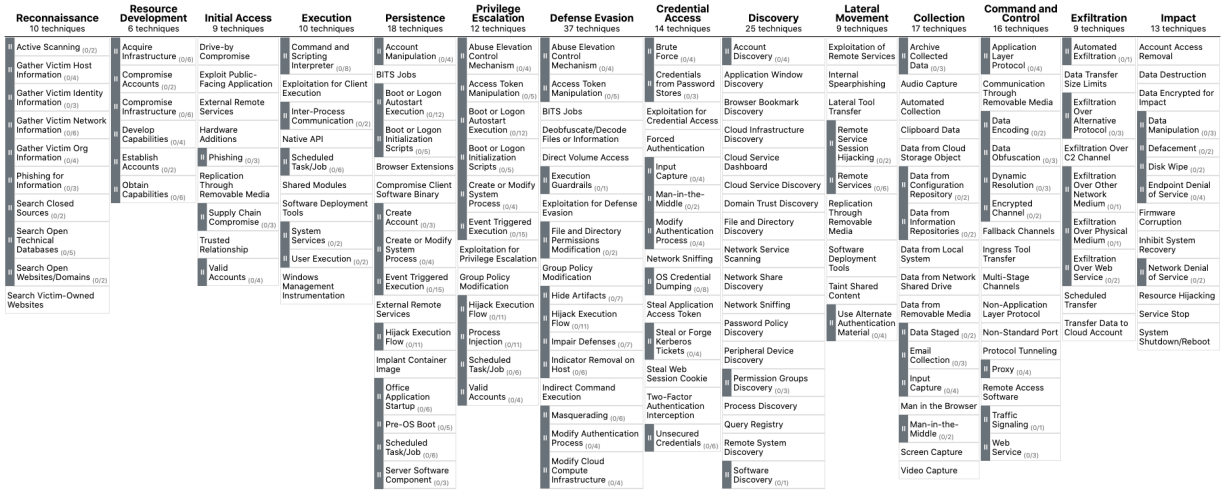
- Security Operations Centers (SOCs) have evolved significantly from their early days as Network Operation Centers focused primarily on network security.
  - In the first generation, known as SOC 1.0, organizations invested in certain security layers such as security intelligence platforms or identity management systems.
  - However, the lack of proper integration led to uncorrelated alerts and a buildup of tasks across multiple platforms.
  - This stage was characterized by an emphasis on network and perimeter security, even as threats began exploiting other vectors.
  - Surprisingly, some organizations continue to rely on this outdated approach, seemingly waiting for a major breach to occur.
- The emergence of sophisticated threats, including multi-vector, persistent, and asynchronous attacks with concealed indicators of compromise, has spurred the transition to SOC 2.0.
  - Malware, including mobile variants, and botnets serve as the primary delivery methods for these attacks.
  - The longevity, evolving behavior, and growth of botnets over time have become focal points for threat intelligence. SOC 2.0 is built on intelligence, integrating security telemetry, threat intelligence, network flow analysis, and other anomaly detection techniques.
  - Additionally, layer-7 analysis is employed at this stage to identify low and slow attacks and other hidden threats.
  - A forward-looking approach to threat research and collaboration between SOCs, either within sectors or at the national level, is crucial for SOC 2.0's success.
  - Emphasis is placed on complete situational awareness, pre-event preparedness through vulnerability management, configuration management, and dynamic risk management, as well as post-event analysis and learning through incident response and in-depth forensics.
  - Refining security intelligence rules and deploying countermeasures are also vital in this stage.
- The cognitive SOC, or next-generation SOC, seeks to address the remaining shortcomings of SOC 2.0.
  - While SOC 2.0 has all the essential subsystems, it often lacks operational experience and effective collaboration between business and security teams to create rules that detect threats specific to business processes and systems.
  - Moreover, many organizations still lack standardized incident response and recovery procedures.

- Cognitive SOCs aim to resolve these issues by incorporating learning systems that compensate for experience gaps in security decision-making.
    - While the success rate of this approach may not be perfect in every instance, it is expected to improve over time.

## MITRE ATT&CK & Security Operations

What Is MITRE ATT&CK?

- The MITRE ATT&CK (Adversarial Tactics, Techniques, and Common Knowledge) framework serves as an extensive, regularly updated resource outlining the tactics, techniques, and procedures (TTPs) employed by cyber threat actors.
    - This structured methodology assists cybersecurity experts in comprehending, identifying, and reacting to threats more proactively and knowledgeably.
- The ATT&CK framework comprises matrices tailored to various computing contexts, such as enterprise, mobile, or cloud systems.
  - Each matrix links the tactics (the goals attackers aim to achieve) and techniques (the methods used to accomplish their objectives) to distinct TTPs.
  - This linkage allows security teams to methodically examine and predict attacker activities.




MITRE ATT&CK Use Cases In Security Operations

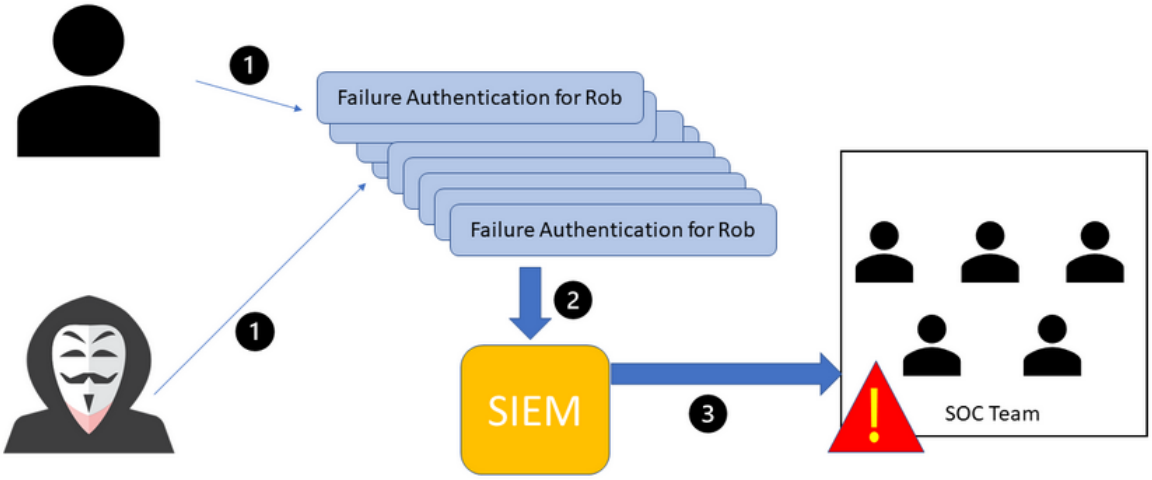
- The MITRE ATT&CK framework not only serves as a comprehensive resource for understanding adversarial tactics, techniques, and procedures (TTPs), but it also plays a crucial role in several aspects of Security Operations. These include:
    - `Detection and Response`: The framework supports SOCs in devising detection and response plans based on recognized attacker TTPs, empowering security teams to pinpoint potential dangers and develop proactive countermeasures.

    - `Security Evaluation and Gap Analysis`: Organizations can leverage the ATT&CK framework to identify the strengths and weaknesses of their security posture, subsequently prioritizing security control investments to effectively defend against relevant threats.
    - `SOC Maturity Assessment`: The ATT&CK framework enables organizations to assess their Security Operations Center (SOC) maturity by measuring their ability to detect, respond to, and mitigate various TTPs.
        - This assessment assists in identifying areas for improvement and prioritizing resources to strengthen the overall security posture.
    - `Threat Intelligence`: The framework offers a unified language and format to describe adversarial actions, enabling organizations to bolster their threat intelligence and improve collaboration among internal teams or with external stakeholders.
    - `Cyber Threat Intelligence Enrichment`: Leveraging the ATT&CK framework can help organizations enrich their cyber threat intelligence by providing context on attacker TTPs, as well as insights into potential targets and indicators of compromise (IOCs).
        - This enrichment allows for more informed decision-making and effective threat mitigation strategies.
    - `Behavioral Analytics Development`: By mapping the TTPs outlined in the ATT&CK framework to specific user and system behaviors, organizations can develop behavioral analytics models to identify anomalous activities indicative of potential threats.
        - This approach enhances detection capabilities and helps security teams proactively mitigate risks.
    - `Red Teaming and Penetration Testing`: The ATT&CK framework presents a systematic way to replicate genuine attacker techniques during red teaming exercises and penetration tests, ultimately assessing an organization's defensive capabilities.
    - `Training and Education`: The comprehensive and well-organized nature of the ATT&CK framework makes it an exceptional resource for training and educating security professionals on the latest adversarial tactics and methods.
- In conclusion, the MITRE ATT&CK framework is an indispensable asset for security operations, offering a shared language and structure for describing and understanding adversarial behavior.
    - It is vital for enhancing various aspects of security operations, from threat intelligence and behavioral analytics to SOC maturity assessment and cyber threat intelligence enrichment.

**SIEM Use Case Development**

What Is A SIEM Use Case?

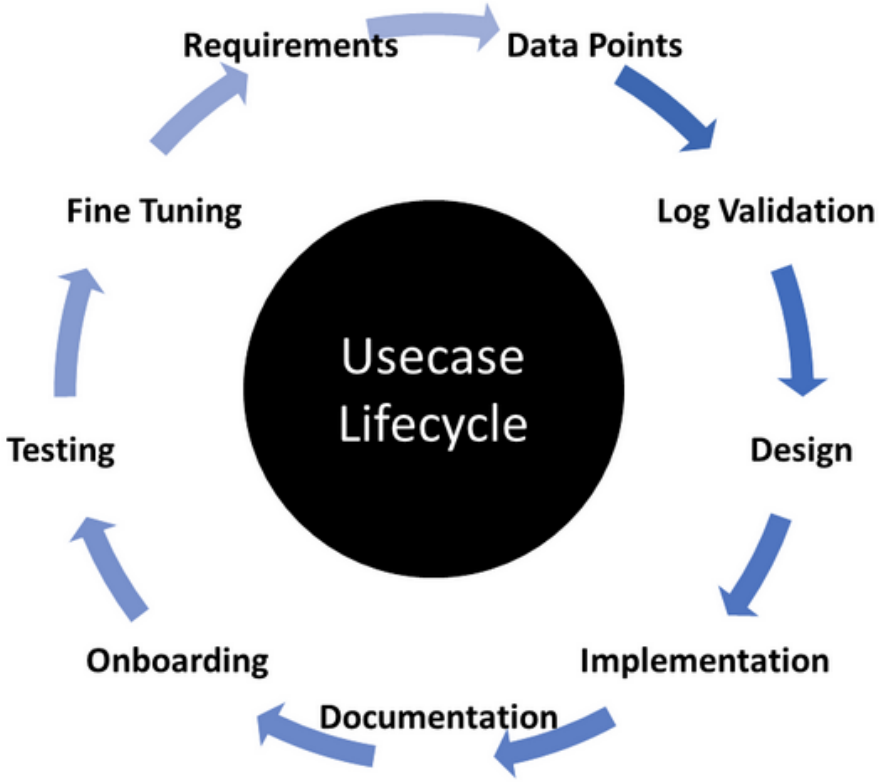
- Utilizing SIEM use cases is a fundamental aspect of crafting a robust cybersecurity strategy, as they enable the effective identification and detection of potential security incidents.
  - Use cases are designed to illustrate specific situations where a product or service can be applied, and they can range from general scenarios, such as failed login attempts, to more complex ones like detecting a ransomware outbreak.




- For instance, consider a situation where a user named Rob experiences 10 consecutive failed authentication attempts.
  - These events could originate from the actual user who forgot their credentials or from a malicious actor trying to brute force their way into the account.
  - In either case, these 10 events are sent to the SIEM system, which then correlates them into a single event and triggers an alert to the SOC team under the "brute force" use case category.
- Based on the log data generated within the SIEM, the SOC team is then responsible for taking appropriate action.
  - This example demonstrates just one of the many possible use cases that can be developed, ranging from straightforward to more intricate situations.

SIEM Use Case Development Lifecycle

- The following critical stages must be considered when developing any use cases:




1. `Requirements`: Comprehend the purpose or necessity of the use case, pinpointing the specific scenario for which an alert or notification is needed. Requirements can be proposed by customers, analysts, or employees.
    - For instance, the goal might be to design a detection use case for a brute force attack that triggers an alert after 10 consecutive login failures within 4 minutes.
2. `Data Points`: Identify all data points within the network where a user account can be used to log in.
    - Gather information about the data sources that generate logs for unauthorized access attempts or login failures.
    - For example, data might come from Windows machines, Linux machines, endpoints, servers, or applications.
    - Ensure logs capture essential details like user, timestamp, source, destination, etc.
3. `Log Validation`: Verify and validate the logs, ensuring they contain all crucial information such as user, timestamp, source, destination, machine name, and application name.
    - Confirm all logs are received during various user authentication events for critical data points, including local, web-based, application, VPN, and OWA (Outlook) authentication.

4. `Design and Implementation`: After identifying and verifying all logs with different data points and sources, begin designing the use case by defining the conditions under which an alert should be triggered.
    - Consider three primary parameters: Condition, Aggregation, and Priority.
    - For example, in a brute force attack use case, create an alert for 10 login failures in 4 minutes while considering aggregation to avoid false positives and setting alert priority based on the targeted user's privileges.
5. `Documentation`: Standard Operating Procedures (SOP) detail the standard processes analysts must follow when working on alerts.
    - This includes conditions, aggregations, priorities, and information about other teams to which analysts need to report activities.
    - The SOP also contains the escalation matrix.
6. `Onboarding`: Start with the development stage before moving the alert directly into the production environment.
    - Identify and address any gaps to reduce false positives, then proceed to production.
7. `Periodic Update/Fine-tuning`: Obtain regular feedback from analysts and maintain up-to-date correlation rules by whitelisting.
    - Continually refine and optimize the use case to ensure its effectiveness and accuracy.

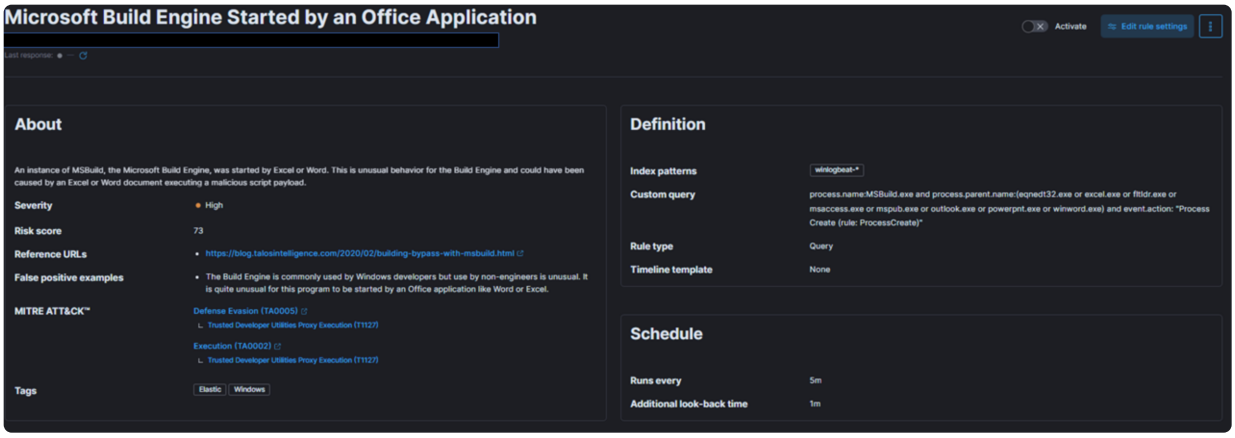
How To Build SIEM Use Cases

- Comprehend your needs, risks, and establish alerts for monitoring all necessary systems accordingly.
- Determine the priority and impact, then map the alert to the kill chain or MITRE framework.
- Establish the Time to Detection (TTD) and Time to Response (TTR) for the alert to assess the SIEM's effectiveness and analysts' performance.
- Create a Standard Operating Procedure (SOP) for managing alerts.
- Outline the process for refining alerts based on SIEM monitoring.
- Develop an Incident Response Plan (IRP) to address true positive incidents.
- Set Service Level Agreements (SLAs) and Operational Level Agreements (OLAs) between teams for handling alerts and following the IRP.
- Implement and maintain an audit process for managing alerts and incident reporting by analysts.
- Create documentation to review the logging status of machines or systems, the basis for creating alerts, and their triggering frequency.

- Establish a knowledge base document for essential information and updates to case management tools.

Example 1 (Microsoft Build Engine Started By An Office Application)

- Now, let's explore a practical example using the Elastic stack as a SIEM solution to help understand how to map each of the above points.



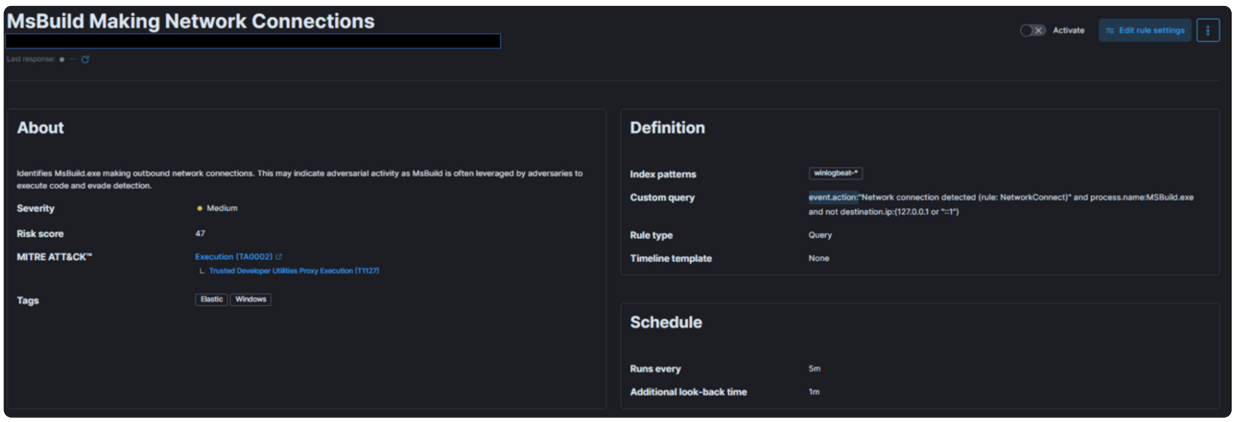

- In the provided snapshot (detection use case), we need to determine our risk and the target of our monitoring efforts.
- MSBuild, part of the Microsoft Build Engine, is a software build system that assembles applications according to its XML input file.
  - Typically, Microsoft Visual Studio generates the input file, but the .NET framework and other compilers can also compile applications without it.
  - Attackers exploit MSBuild's ability to include malicious source code within its configuration or project file.
- When monitoring process execution command-line arguments, it is crucial to investigate instances where a web browser or Microsoft Office executable initiates MSBuild.
  - This suspicious behavior suggests a potential breach. Once a baseline is established, unusual MSBuild calls should be easily identifiable and relatively rare, avoiding increased workload for the team.
- To address this risk, we create a detection use case in our SIEM solution that monitors instances of MSBuild initiated by Excel or Word, as this behavior could indicate a malicious script payload execution.
- Next, let's define priority, impact, and map the alert to the kill chain or MITRE framework.
- Given the above risk and threat intelligence, this technique, known as Living-off-the-land binaries (LoLBins), poses a significant threat if detected, making it a high global risk category.

    - Consequently, we assign it HIGH severity, though this may vary depending on your organization's specific context and landscape.
- Regarding MITRE mapping, this use case involves bypassing detection techniques via LoLBins usage, falling under the Defense Evasion (TA0005) tactic, the Trusted Developer Utilities Proxy Execution (T1127) technique, and the Trusted Developer Utilities Proxy Execution: MSBuild (T1127.001) sub-technique.
    - Additionally, executing the MSBuild binary on the endpoint also falls under the Execution (TA0002) tactic.
- To define TTD and TTR, we need to focus on the rule's execution interval and the data ingestion pipeline discussed earlier.
    - For this example, we set the rule to run every five minutes, monitoring all incoming logs.
- When creating an SOP and documenting alert handling, consider the following:
    - process.name
    - process.parent.name
    - event.action
    - machine where the alert was detected
    - user associated with the machine
    - user activity within +/- 2 days of the alert's generation
    - After gathering this information, defenders should engage with the user and examine the user's machine to analyze system logs, antivirus logs, and proxy logs from the SIEM for full visibility.
- The SOC team should document all the above points, along with the Incident Response Plan, so that Incident Handlers can reference them during analysis.
- For rule fine-tuning, it is essential to understand the conditions that may trigger false positives.
    - For example, while the Build Engine is common among Windows developers, its use by non-engineers is unusual.
    - Excluding legitimate parent process names from the rule helps avoid false positives.
    - Further details on fine-tuning SIEM rules will be given later on.
- The SOC team should document all the above points, along with the Incident Response Plan, so that Incident Handlers can reference them during analysis.
- For rule fine-tuning, it is essential to understand the conditions that may trigger false positives.
    - For example, while the Build Engine is common among Windows developers, its use by non-engineers is unusual.

- Excluding legitimate parent process names from the rule helps avoid false positives.
- Further details on fine-tuning SIEM rules will be given later on.

Example 2 (MSBuild Making Network Connections)

- Example 1 discussed a high-severity detection use case and rule. Now, let's examine a medium-severity use case using a SIEM solution to better understand how each pointer contributes to the effectiveness of use cases.




- In the given snapshot, we need to determine our risk and what we are trying to monitor.
- Like in Example 1, we are again focusing on the MsBuild.exe binary.
  - However, this time, we consider the scenario in which a machine attempts outbound communication with a remote or potentially malicious IP address, and the process behind that connection is MsBuild.exe.
  - This would raise an alarm, as it may indicate adversarial activity. MsBuild is often exploited by adversaries to execute code and evade detection.
- To address this risk, we need a monitoring solution capable of detecting instances where MsBuild is responsible for malicious outbound connections.
  - We create a detection use case in our SIEM solution for this purpose.
- Next, let's define priority, impact, and map the alert to the kill chain or MITRE framework.
- Unlike the previous example, this situation could occur whenever MsBuild.exe establishes an outbound connection.
  - It's also possible for this process to connect to a legitimate IP address, such as a Microsoft IP for updates.
  - Therefore, we might encounter more false positives unless we implement a robust threat intelligence process.
  - Consequently, we should assign this detection rule a MEDIUM severity instead of HIGH.

- As in Example 1, pulling off this particular threat requires attackers to execute the MsBuild binary on the endpoint, which falls under the Execution (TA0002) tactic.
- Most of the other pointers remain the same, but the SOP and Incident Response
    - Plan will differ when handling this specific type of alert. Defenders will need to focus on event.action, IP address, and the reputation of the IP, among other factors.

## SIEM Visualization Development

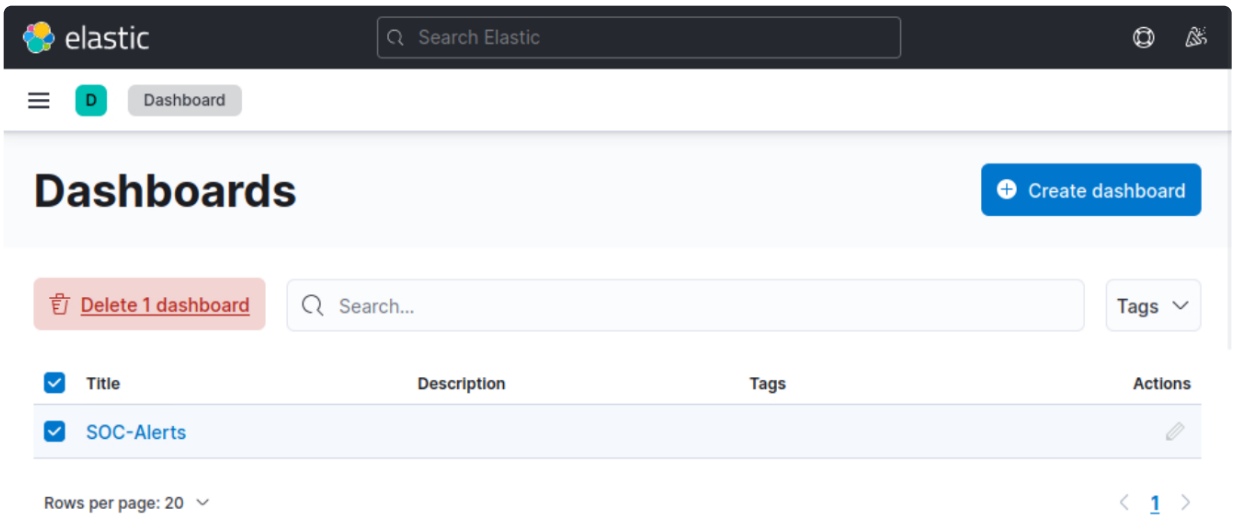
### SIEM Visualization Example 1: Failed Logon Attempts (All Users)

Overview

- Dashboards in SIEM solutions serve as containers for multiple visualizations, allowing us to organize and display data in a meaningful way.
- In this and the following sections, we will create a dashboard and some visualizations from scratch.

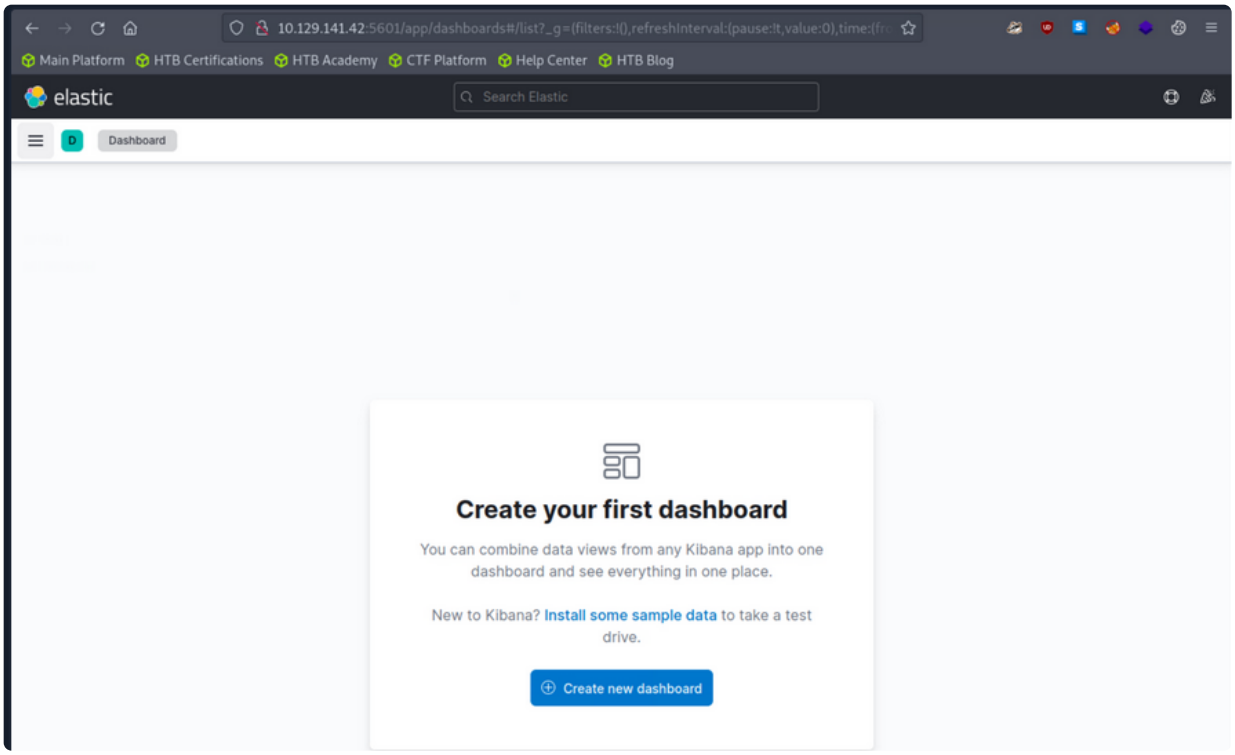
Developing Our First Dashboard & Visualization

- First spawn the target
- Now, navigate to `http://[Target IP]:5601`, click on the side navigation toggle, and click on "Dashboard".
- Delete the existing "SOC-Alerts" dashboard as follows.

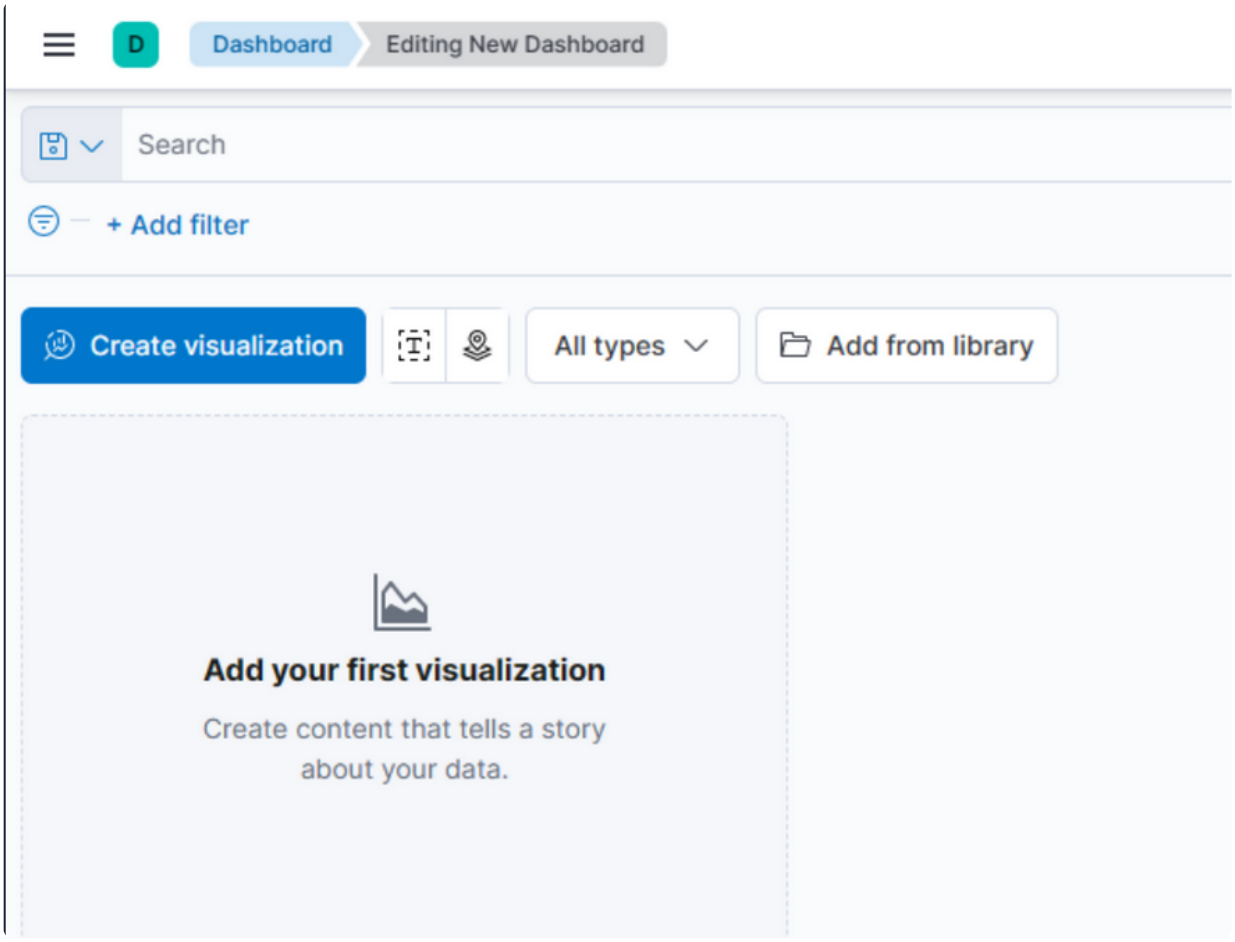



- When visiting the Dashboard page again we will be presented with a message indicating that no dashboards currently exist.
  - Additionally, there will be an option available to create a new Dashboard and its first visualization.

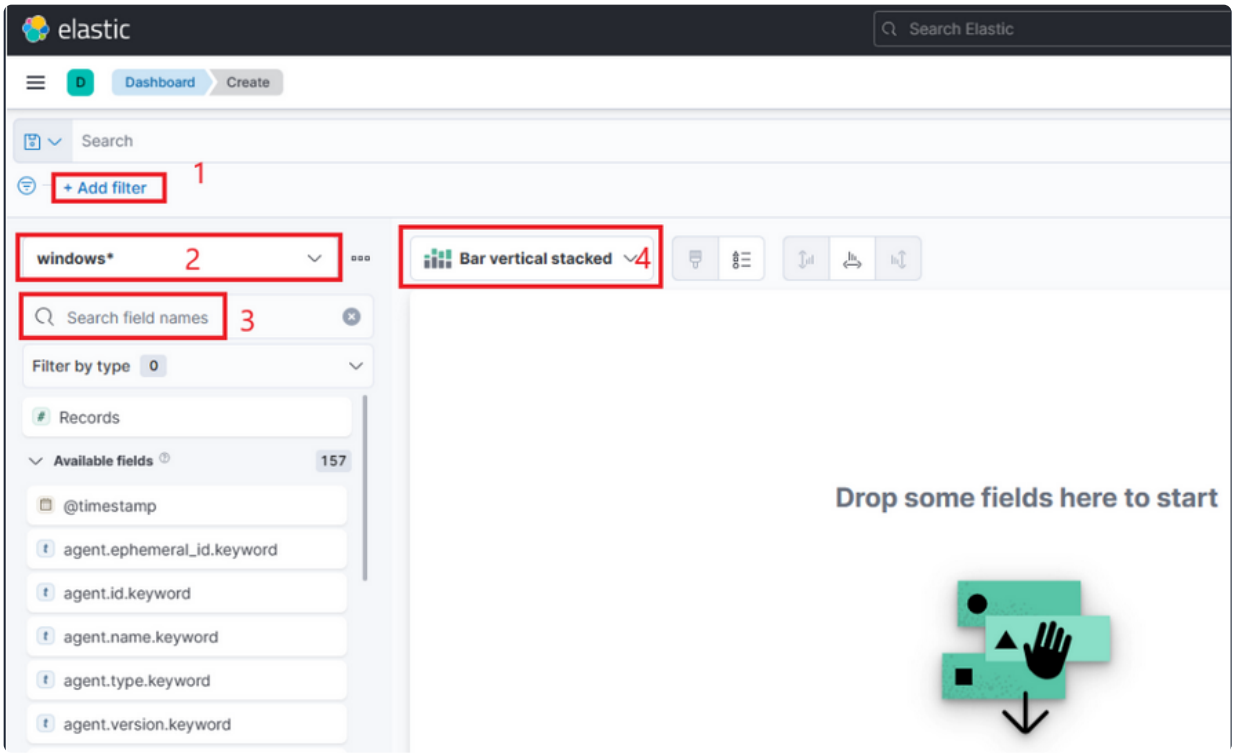
- To initiate the creation of our first dashboard, we simply have to click on the "Create new dashboard" button.




- Now, to initiate the creation of our first visualization, we simply have to click on the "Create visualization" button.



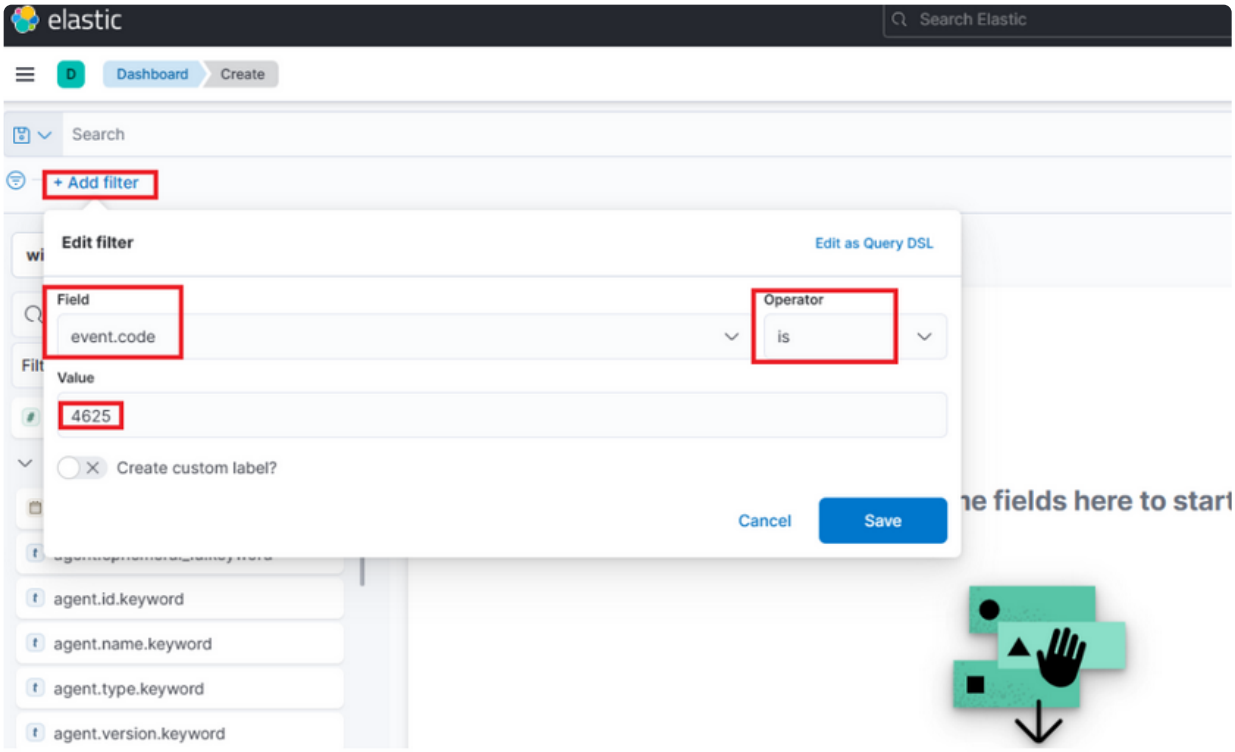
- Upon initiating the creation of our first visualization, the following new window will appear with various options and settings.
- Before proceeding with any configuration, it is important for us to first click on the calendar icon to open the time picker.
  - Then, we need to specify the date range as "last 15 years". Finally, we can click on the "Apply" button to apply the specified date range to the data.




- There are four things for us to notice on this window:

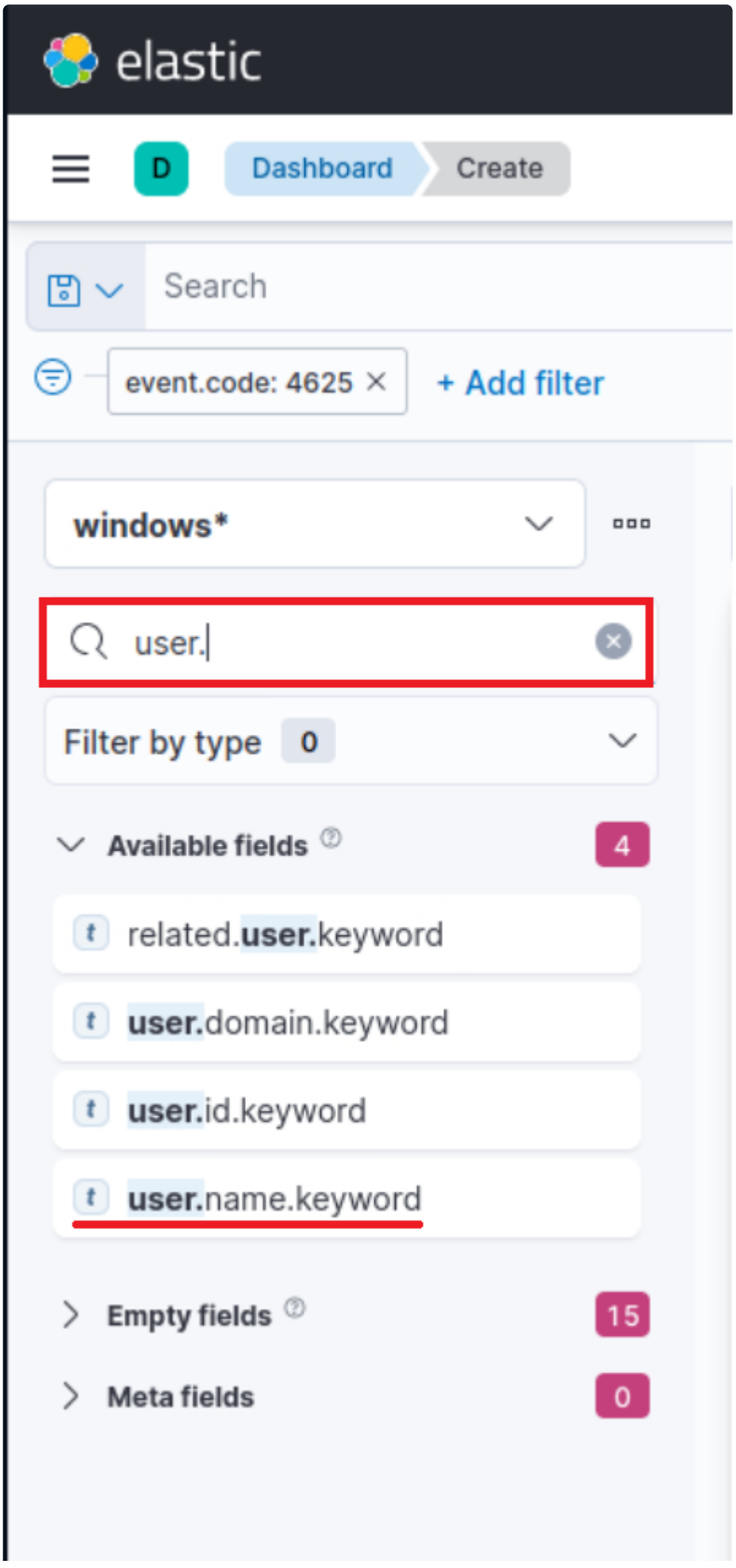
1. A filter option that allows us to filter the data before creating a graph.
   - For example, if our goal is to display failed logon attempts, we can use a filter to only consider event IDs that match `4625 – Failed logon attempt on a Windows system`.

- The following image demonstrates how we can specify such a filter.

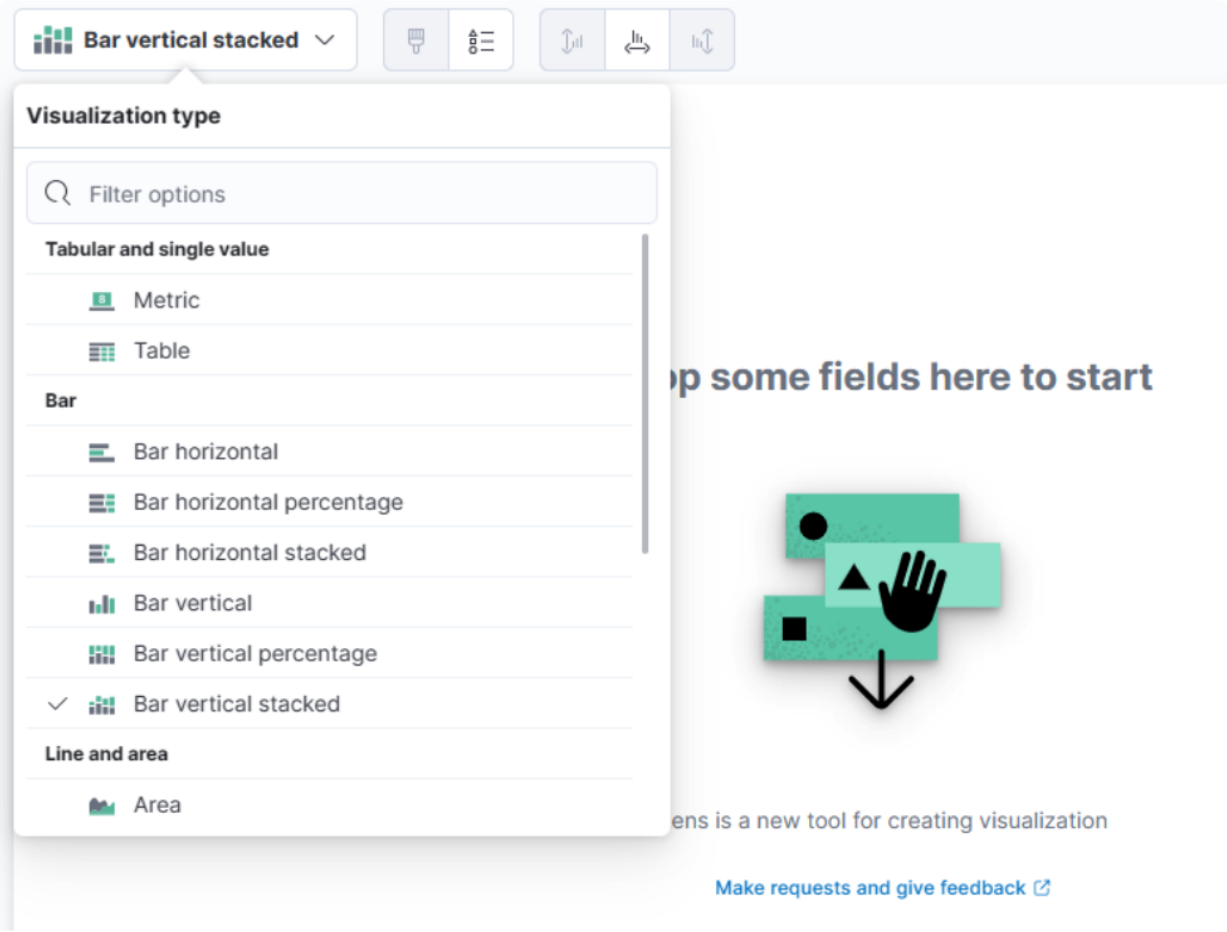



2. This field indicates the data set (index) that we are going to use.
    - It is common for data from various infrastructure sources to be separated into different indices, such as network, Windows, Linux, etc.
    - In this particular example, we will specify `windows*` in the "Index pattern".
3. This search bar provides us with the ability to double-check the existence of a specific field within our data set, serving as another way to ensure that we are looking at the correct data.
   - For example, let's say we are interested in the `user.name.keyword` field. We can use the search bar to quickly perform a search and verify if this field is present and discovered within our selected data set.
   - This allows us to confirm that we are accessing the desired field and working with

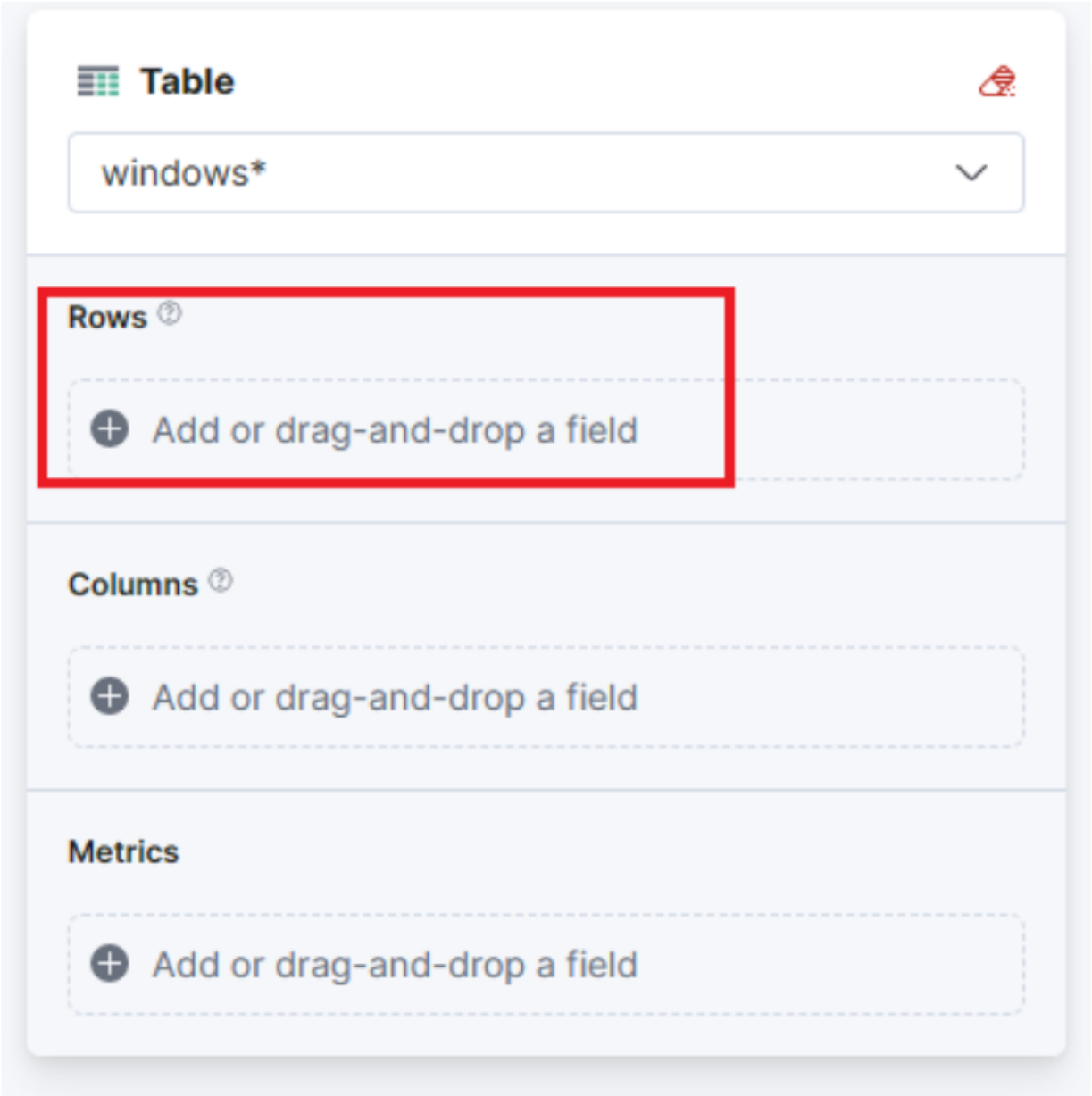
accurate data.


elastic
D
Dashboard
Create
Search
event.code: 4625
+ Add filter
windows*
user.
Filter by type 0
Available fields 4
related.user.keyword
user.domain.keyword
user.id.keyword
user.name.keyword
Empty fields 15
Meta fields 0

- "Why `user.name.keyword` and not `user.name`?", you may ask.
  - We should use the `.keyword` field when it comes to aggregations.
  - Please refer to this stackoverflow question for a more elaborate answer.
- Lastly, this drop-down menu enables us to select the type of visualization we want to create.
  - The default option displayed in the earlier image is "Bar vertical stacked".
  - If we click on that button, it will reveal additional available options (image redacted as not all options fit on the screen).
  - From this expanded list, we can choose the desired visualization type that best suits our requirements and data presentation needs.




- For this visualization, let's select the "Table" option.
  - After selecting the "Table", we can proceed to click on the "Rows" option.
  - This will allow us to choose the specific data elements that we want to include in the table view.




-> **Note**: You will notice `Rank by Alphabetical` and not `Rank by Count of records` like in the screenshot above.

- This is OK. By the time you perform the next configuration below, `Count of records` will become available.
- Moving forward, let's close the "Rows" window and proceed to enter the "Metrics" configuration.

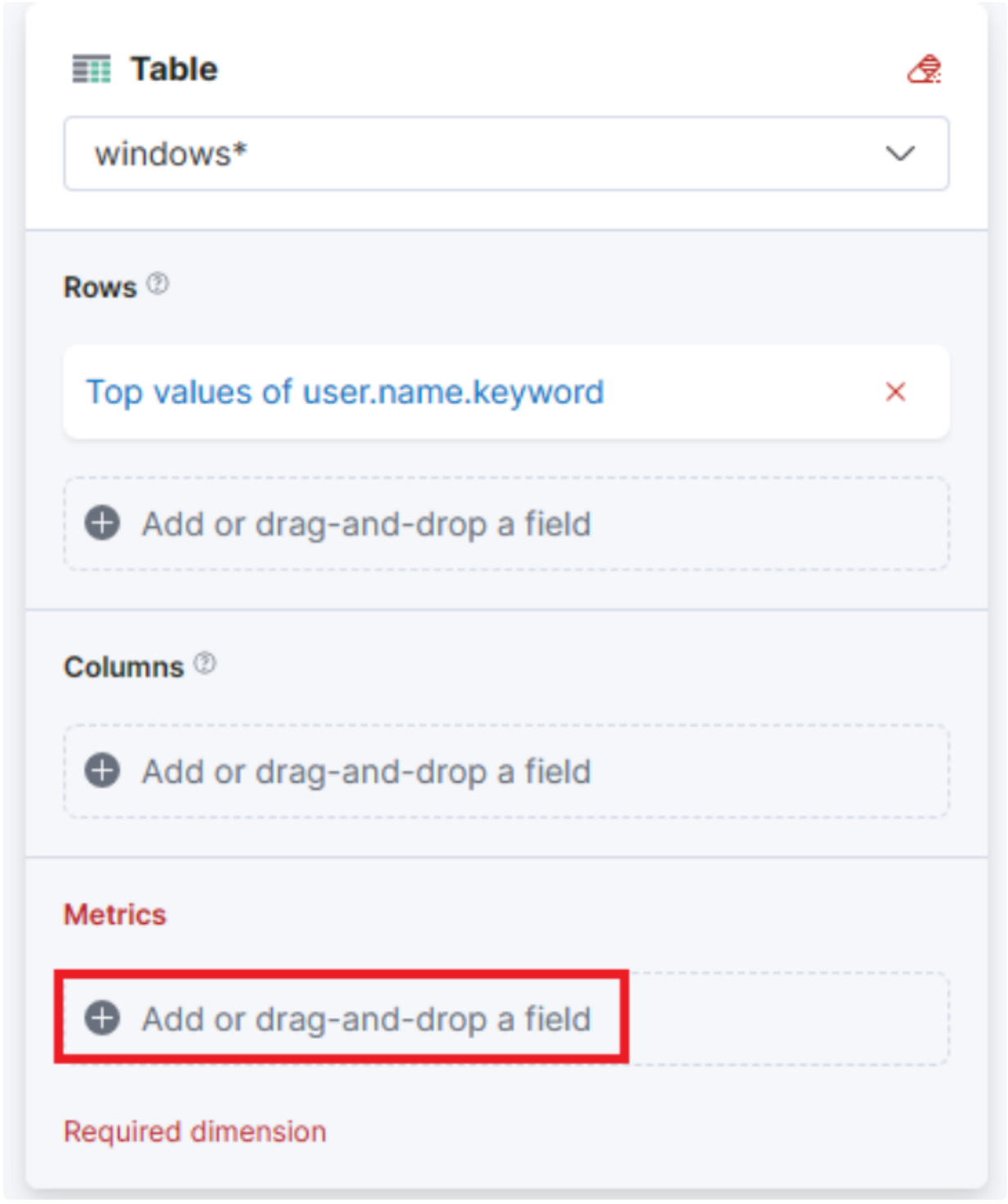
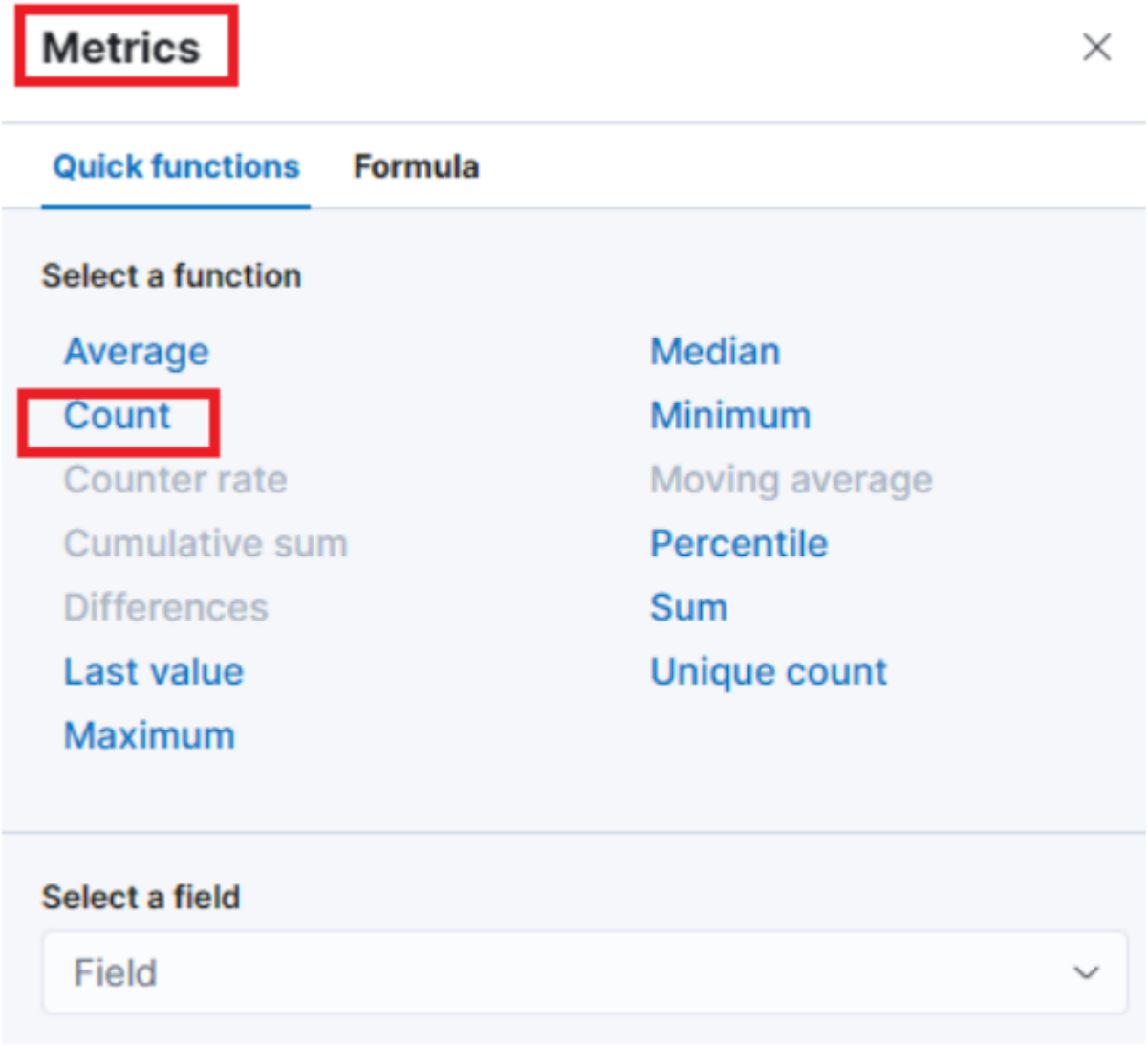


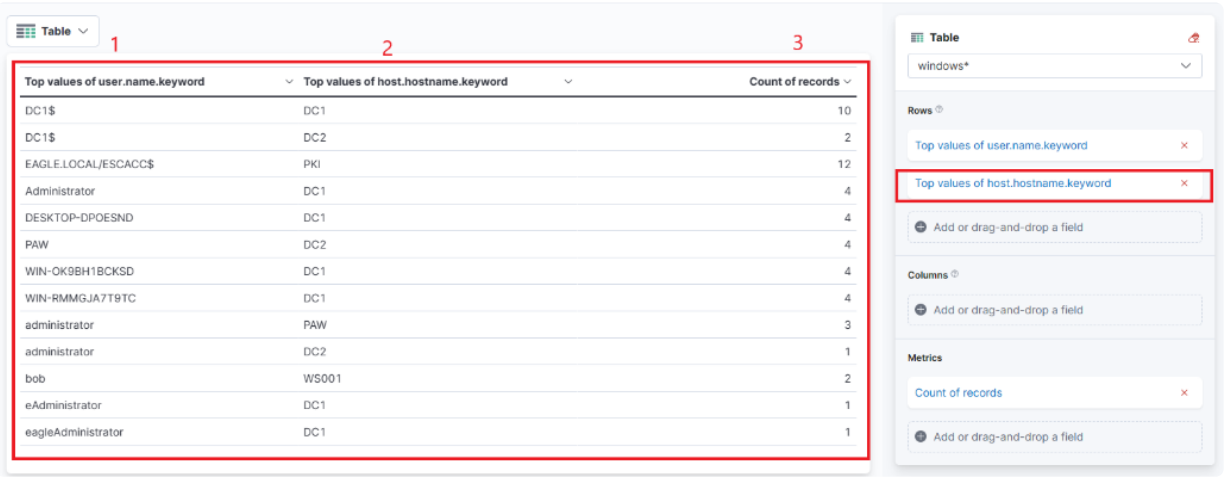
Table
windows*
Rows
Top values of user.name.keyword
Add or drag-and-drop a field
Columns
Add or drag-and-drop a field
Metrics
Add or drag-and-drop a field
Required dimension

- In the "Metrics" window, let's select "count" as the desired metric.

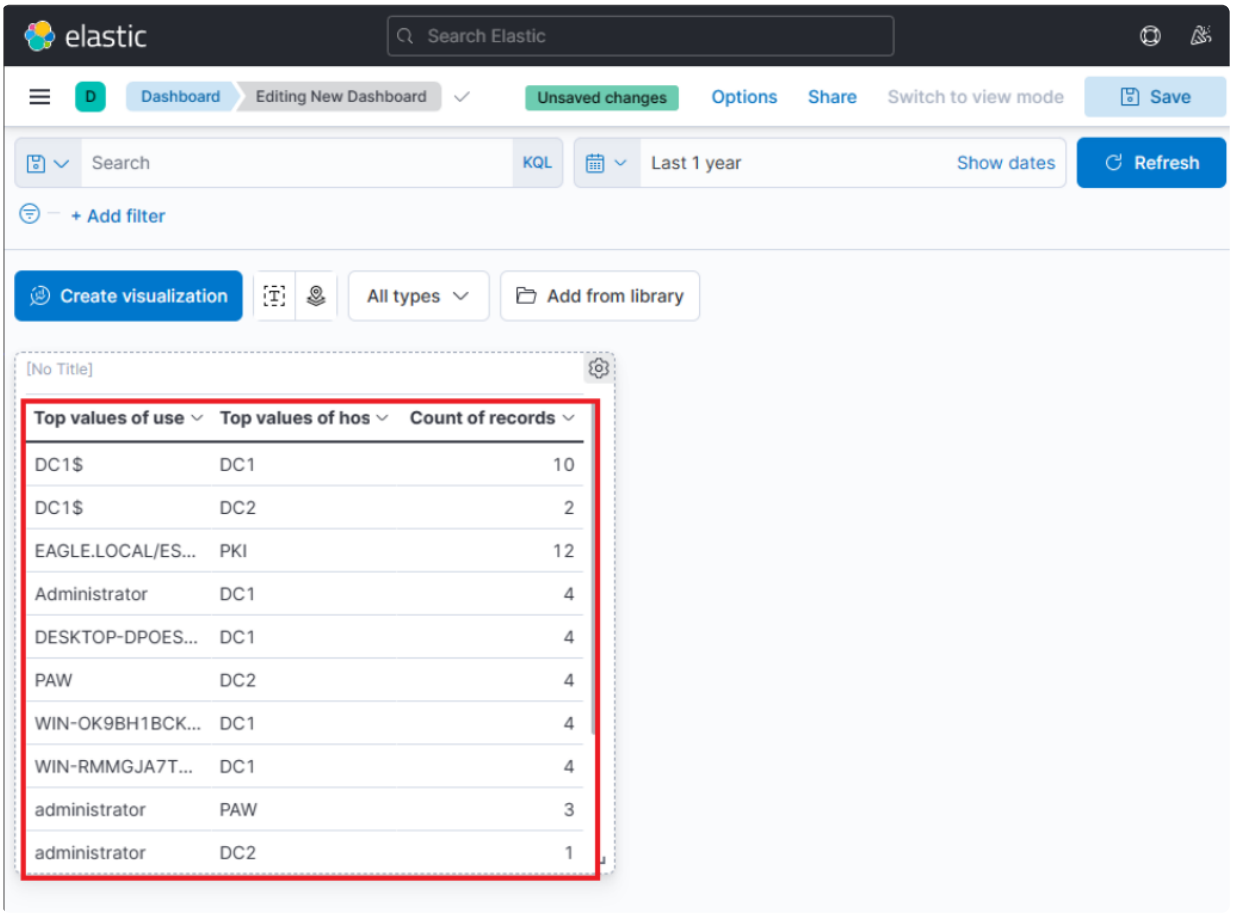



- As soon as we select "Count" as the metric, we will observe that the table gets populated with data (assuming that there are events present in the selected data set)
- One final addition to the table is to include another "Rows" setting to show the machine where the failed logon attempt occurred.
  - To do this, we will select the `host.hostname.keyword` field, which represents the computer reporting the failed logon attempt.
  - This will allow us to display the hostname or machine name alongside the count of

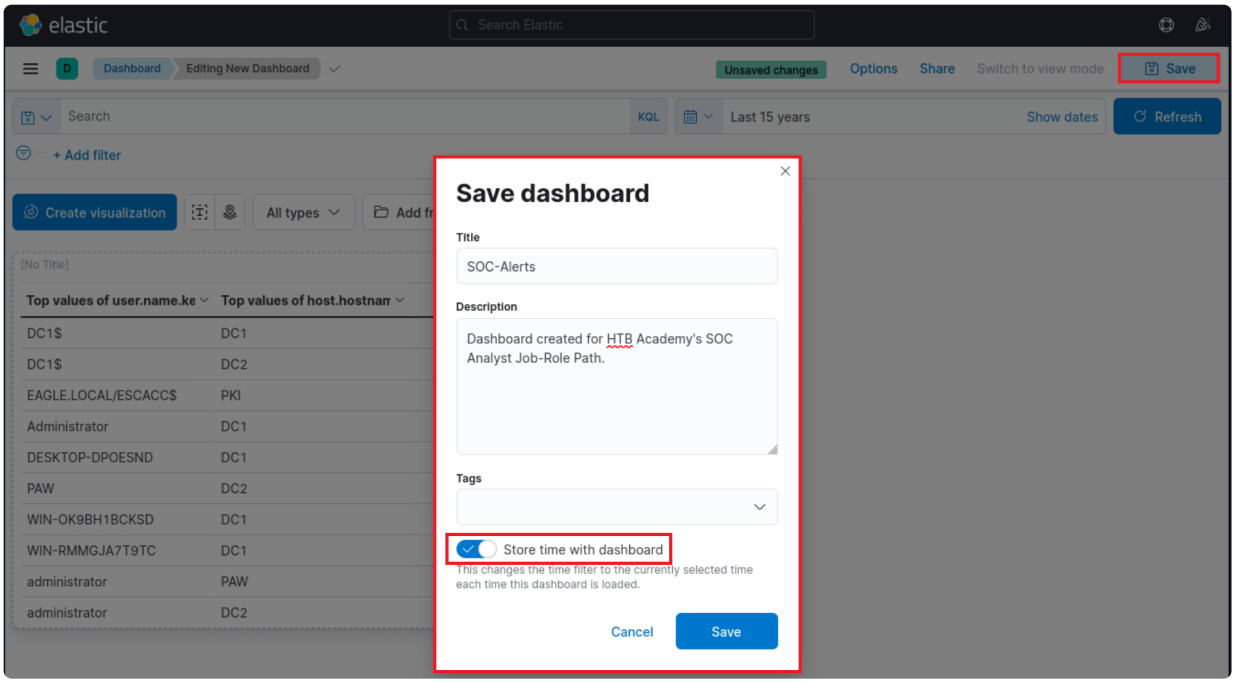
failed logon attempts, as shown in the image.




- Now we can see three columns in the table, which contain the following information:
  1. The username of the individuals logging in (Note: It currently displays both users and computers. Ideally, a filter should be implemented to exclude computer devices and only display users).
  2. The machine on which the logon attempt occurred.
  3. The number of times the event has occurred (based on the specified time frame or the entire data set, depending on the settings).
- Finally, click on "Save and return", and you will observe that the new visualization is added to the dashboard, appearing as shown in the following image.

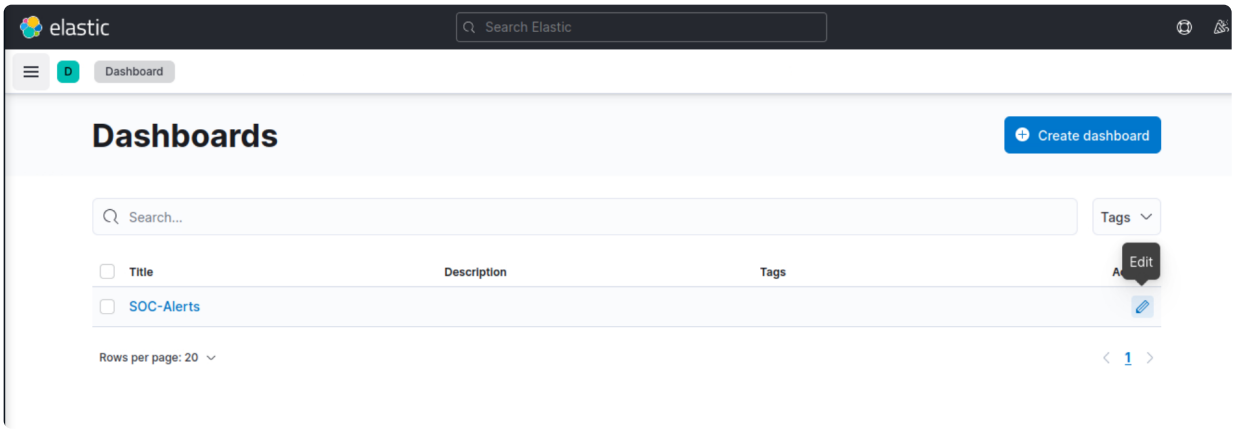


- Let's not forget to save the dashboard as well.
  - We can do so by simply clicking on the "Save" button.

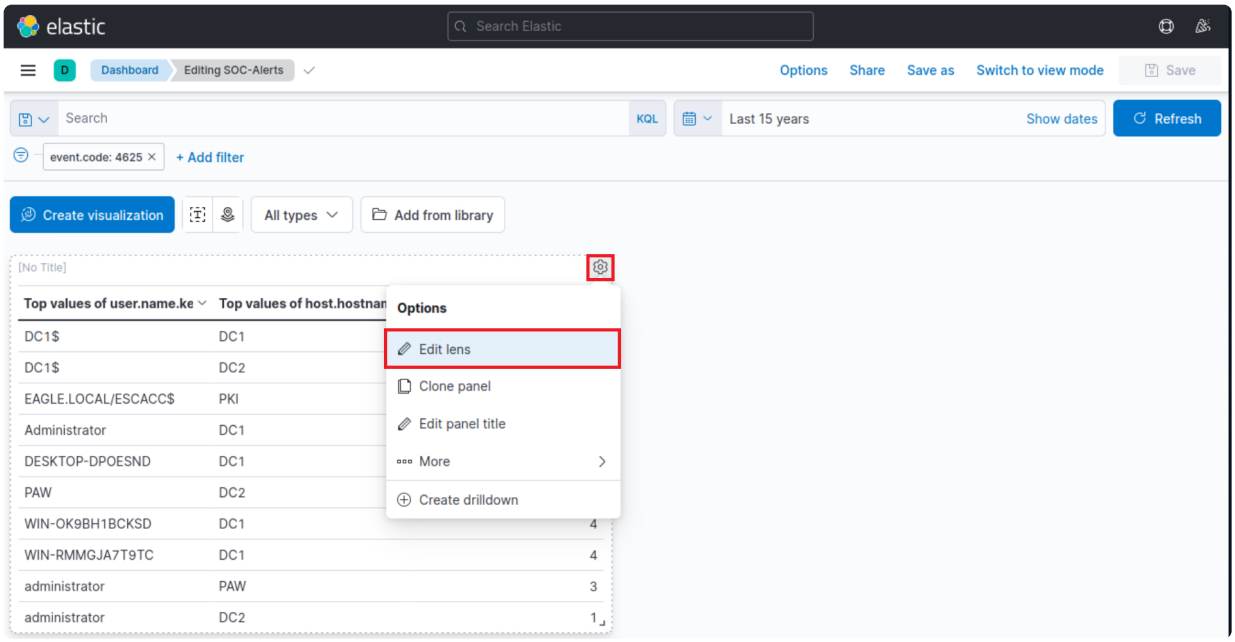



Refining The Visualization

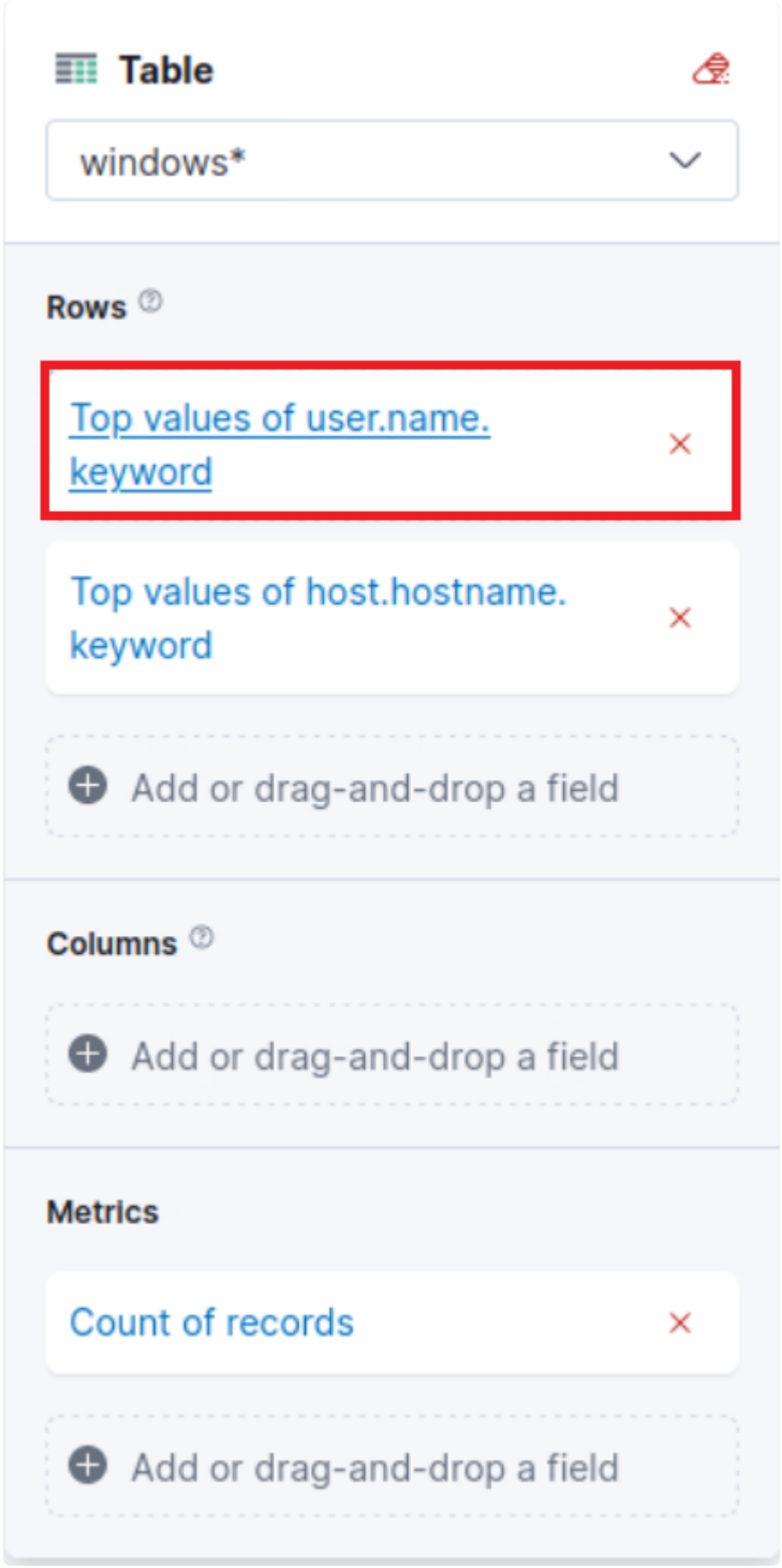
- Suppose the SOC Manager suggested the following refinements:
  - Clearer column names should be specified in the visualization
  - The Logon Type should be included in the visualization
  - The results in the visualization should be sorted
  - The DESKTOP-DPOESND, WIN-OK9BH1BCKSD, and WIN-RMMGJA7T9TC usernames should not be monitored
  - Computer accounts should not be monitored (not a good practice)
- Let's refine the visualization we created, so that it fulfills the suggestions above.
- Navigate to `http://[Target IP]:5601`, click on the side navigation toggle, and click on "Dashboard".
- The dashboard we previously created should be visible.
  - Let's click on the "pencil"/edit icon.



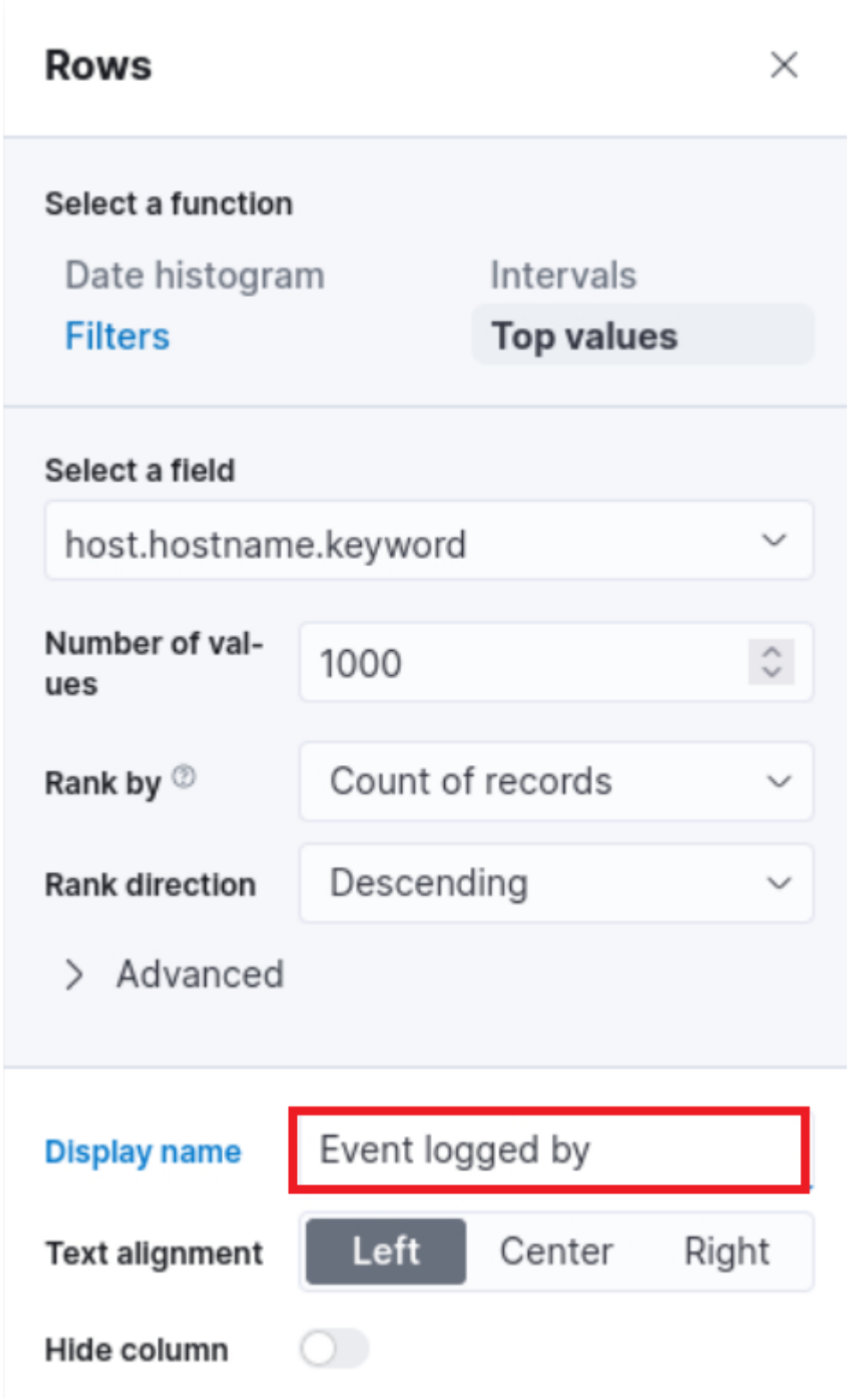

- Let's now click on the "gear" button at the upper-right corner of our visualization, and then click on "Edit lens".



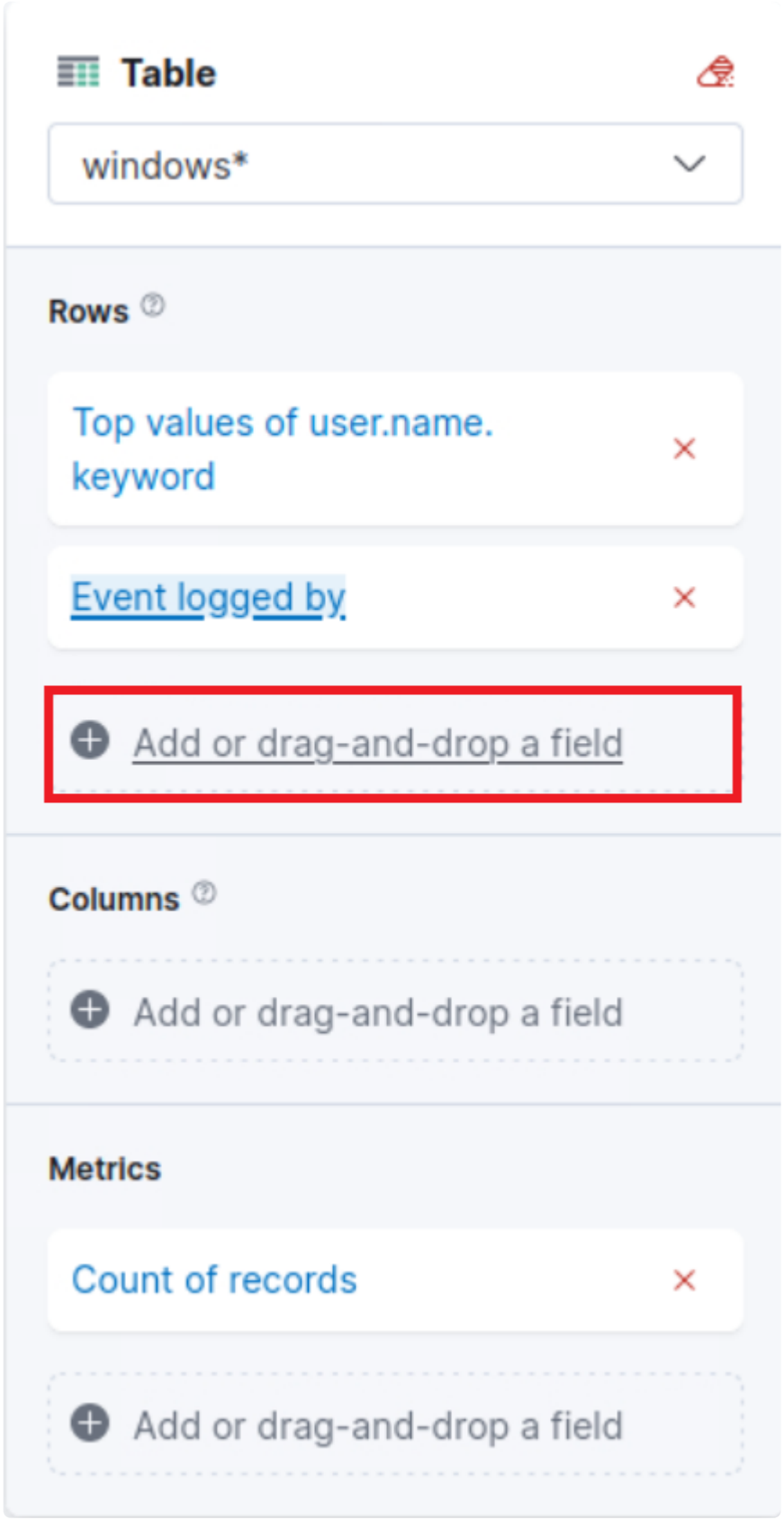
- "Top values of user.name.keyword" should be changed as follows.



- "Top values of host.hostname.keyword" should be changed as follows.




- The "Logon Type" can be added as follows (we will use the `winlog.logon.type.keyword` field).




- The "Logon Type" can be added as follows (we will use the `winlog.logon.type.keyword` field).

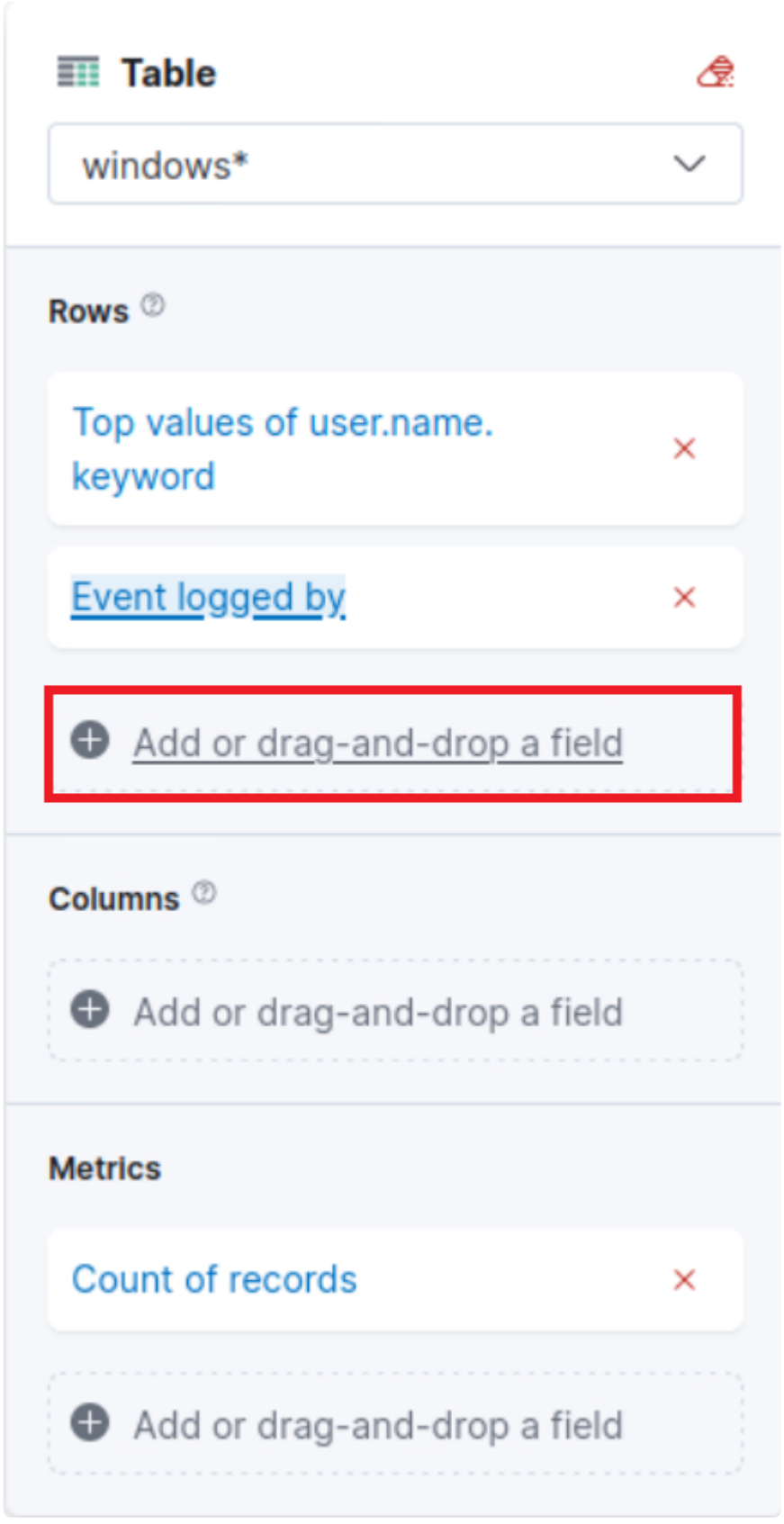
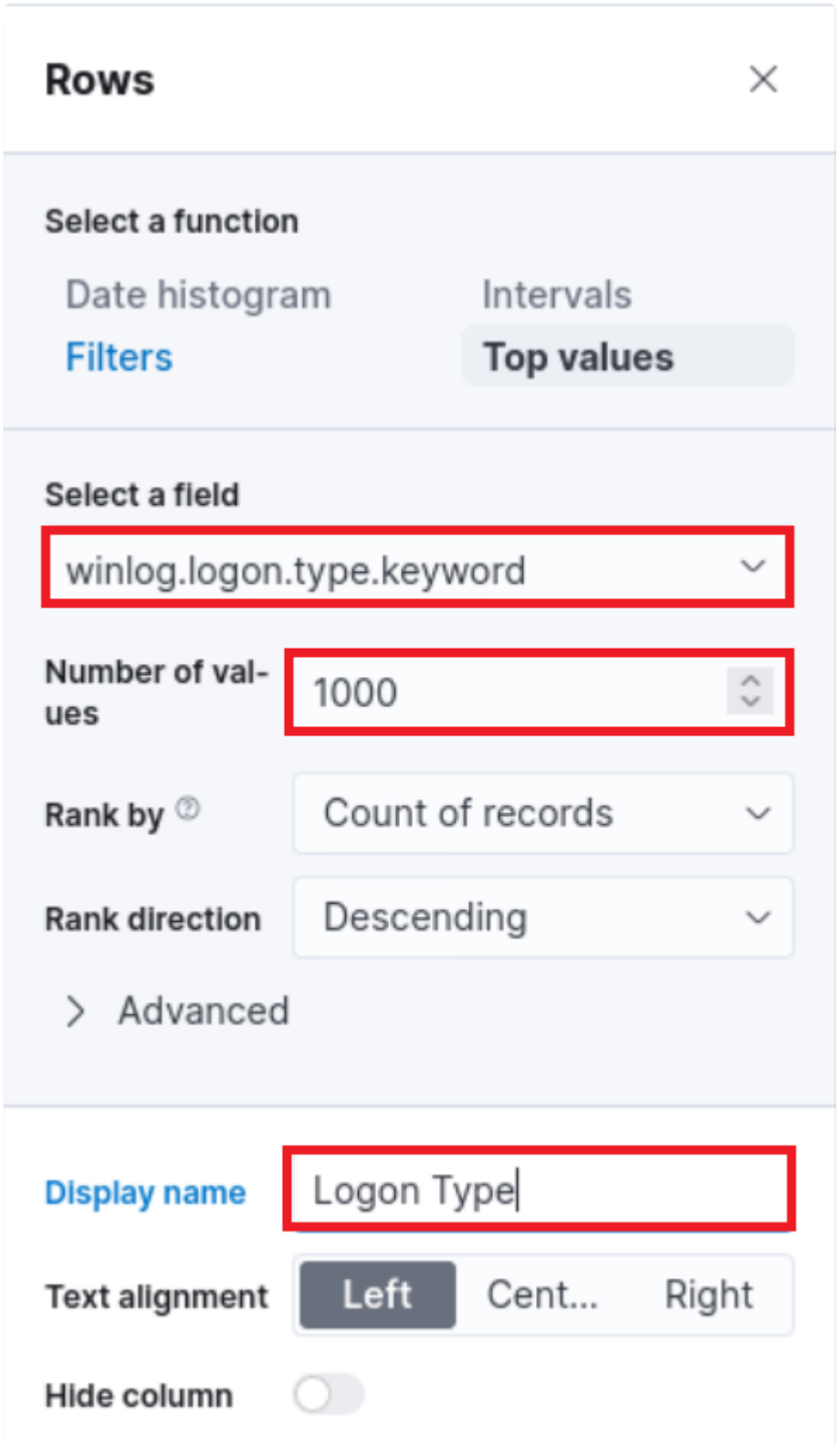
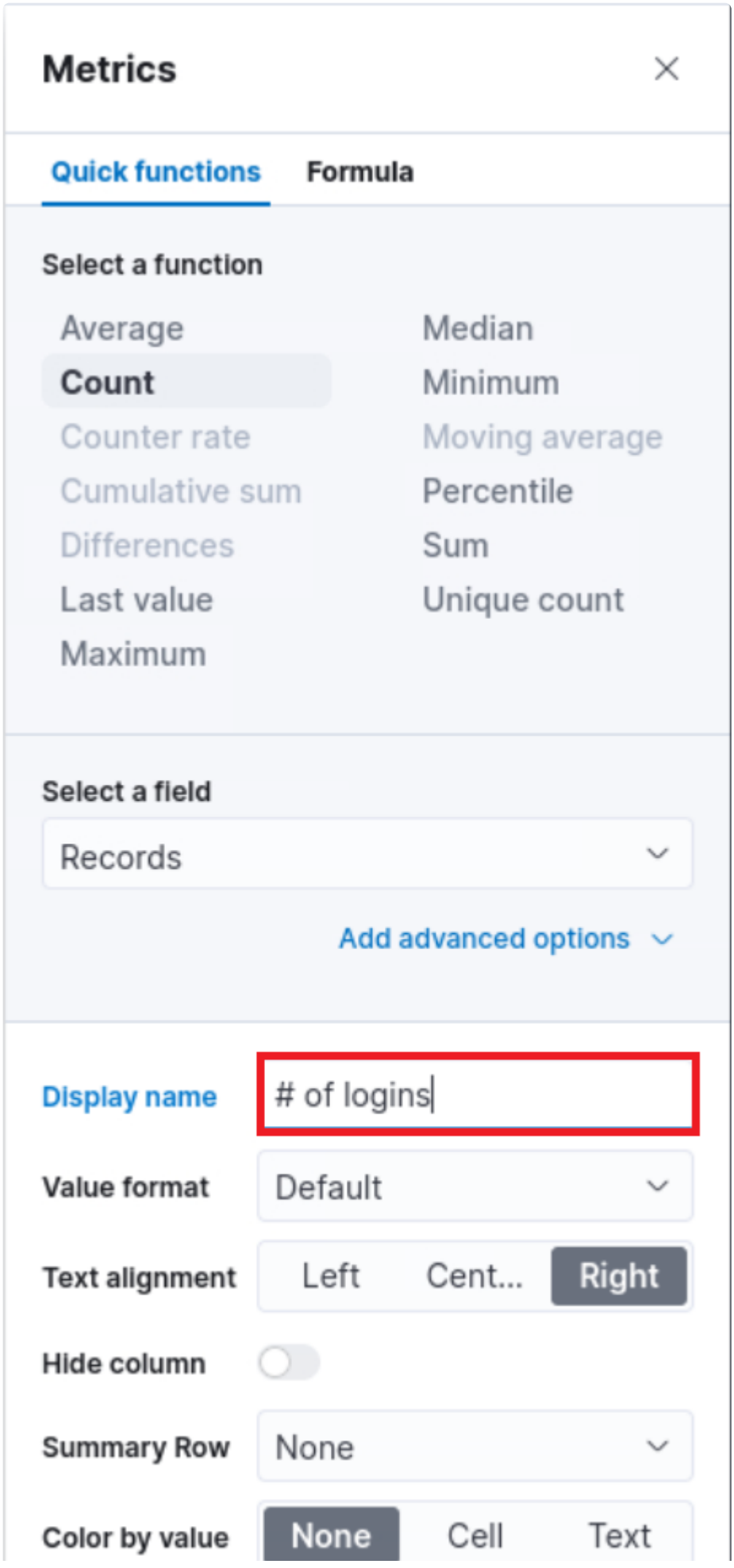


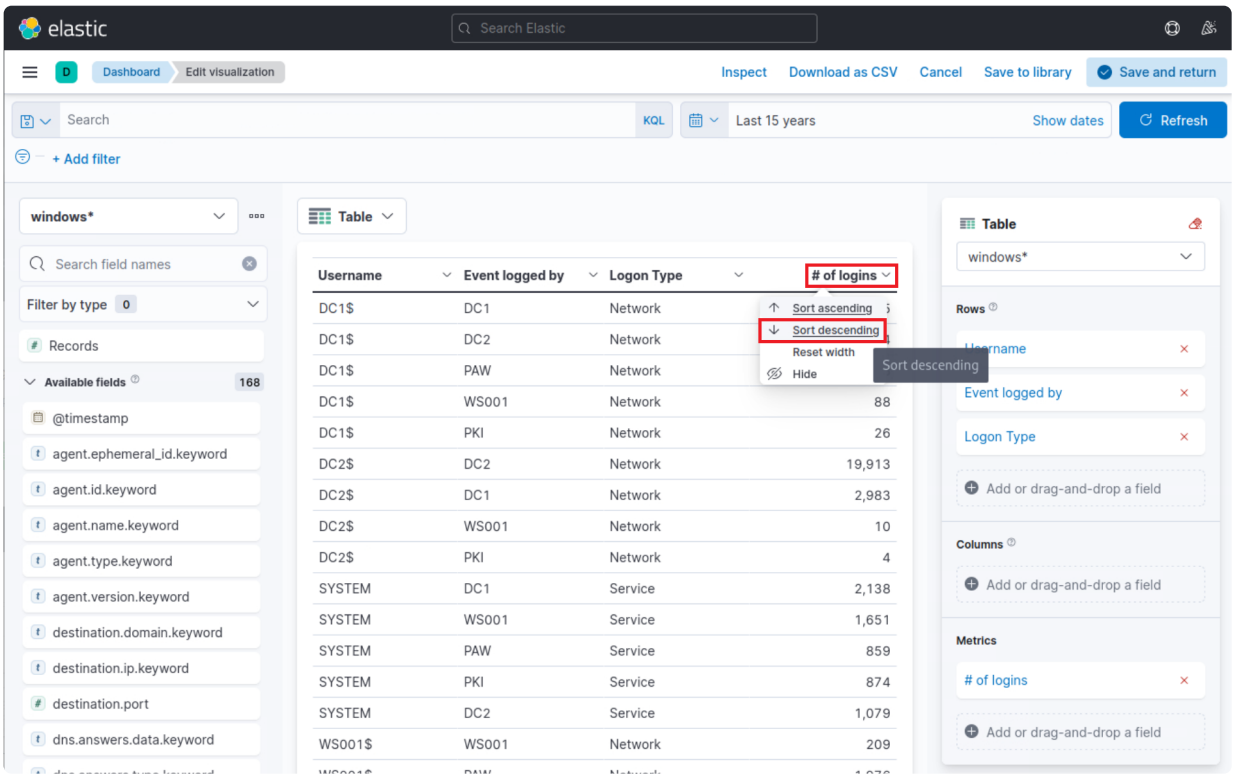
Table
windows*
Rows
Top values of user.name.
keyword
Event logged by
Add or drag-and-drop a field
Columns
Add or drag-and-drop a field
Metrics
Count of records
Add or drag-and-drop a field


Rows
Select a function
Date histogram
Intervals
Filters
Top values
Select a field
winlog.logon.type.keyword
Number of val-ues
1000
Rank by
Count of records
Rank direction
Descending
Advanced
Display name
Logon Type
Text alignment
Left
Cent...
Right
Hide column

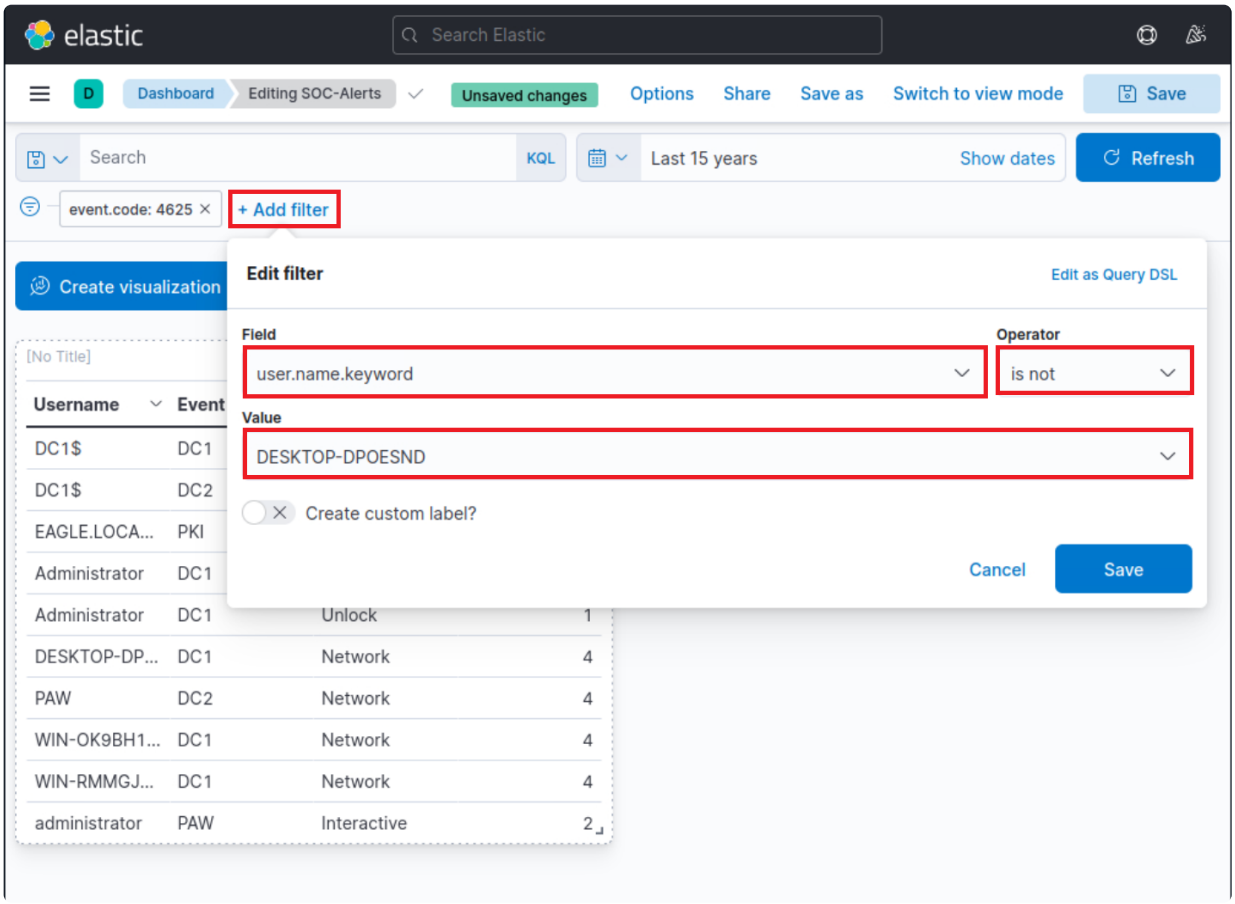
- "Count of records" should be changed as follows.


Metrics
Quick functions
Formula
Select a function
Average
Count
Counter rate
Cumulative sum
Differences
Last value
Maximum
Median
Minimum
Moving average
Percentile
Sum
Unique count
Select a field
Records
Add advanced options
Display name
# of logins
Value format
Default
Text alignment
Left
Cent...
Right
Hide column
Summary Row
None
Color by value
None
Cell
Text

- We can introduce result sorting as follows.




- All we have to do now is click on "Save and return".
- The DESKTOP-DPOESND, WIN-OK9BH1BCKSD, and WIN-RMMGJA7T9TC usernames can be excluded by specifying additional filters as follows.



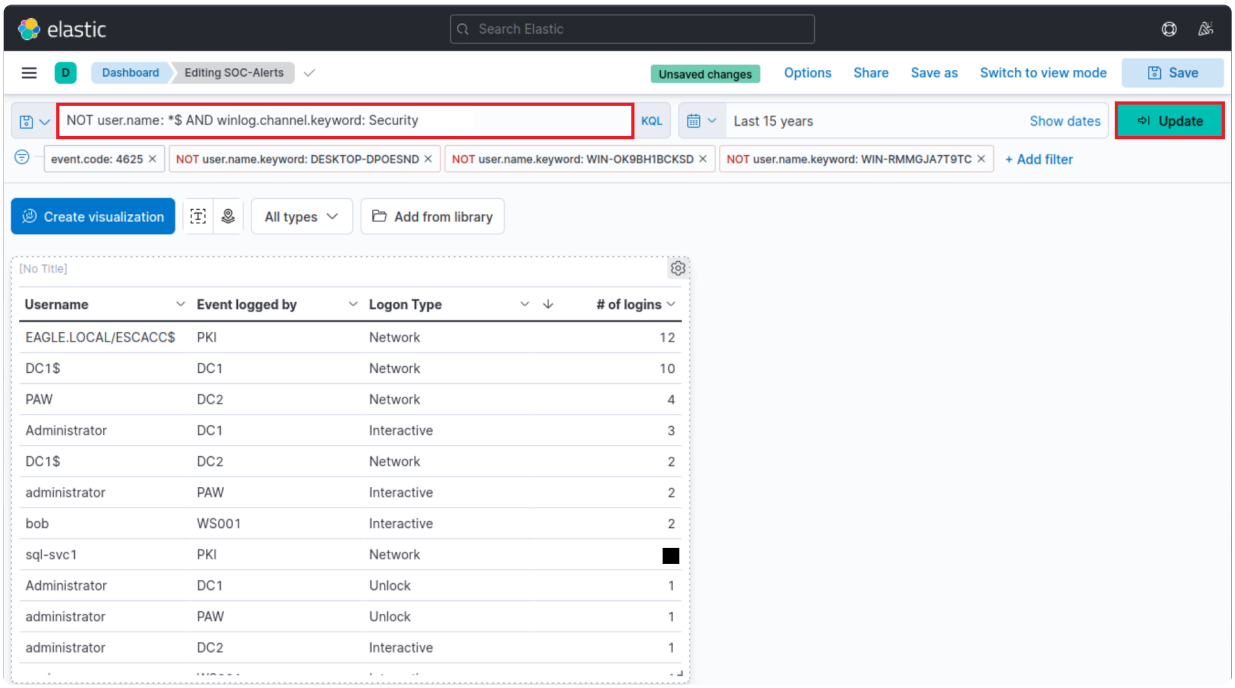

- Computer accounts can be excluded by specifying the following KQL query and clicking on the "Update" button.

```
NOT user.name: *$ AND winlog.channel.keyword: Security
```

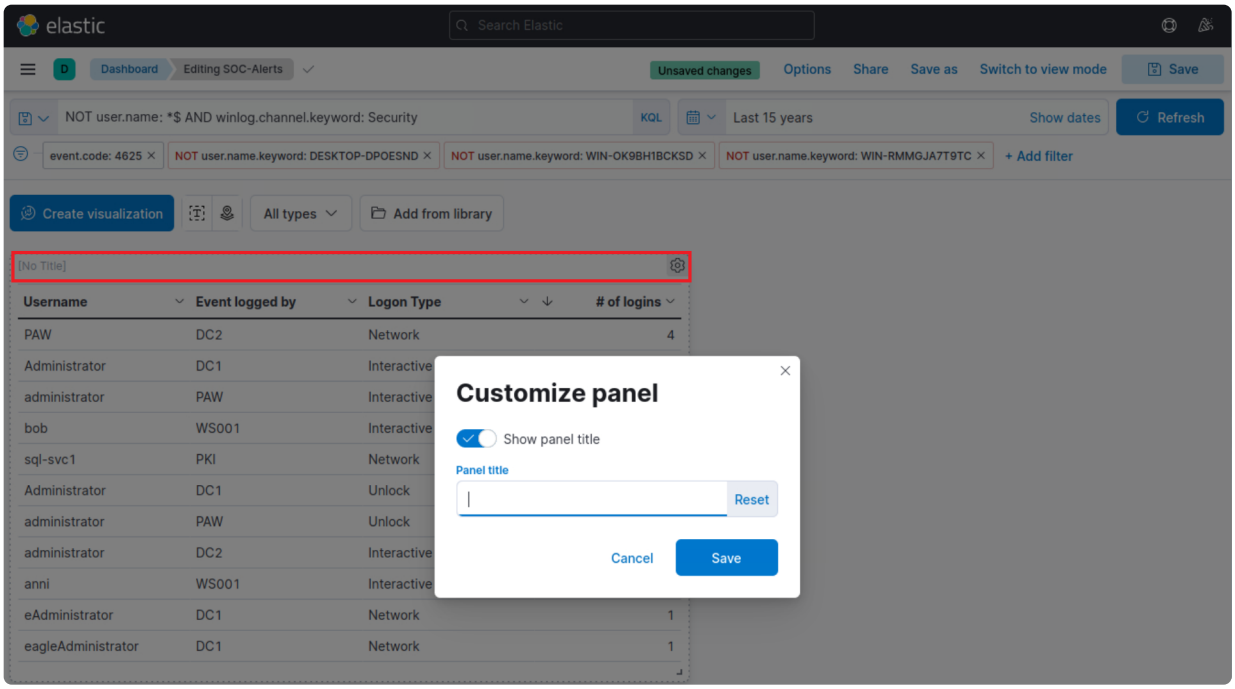

- Computer accounts can be excluded by specifying the following KQL query and clicking on the "Update" button.

```
NOT user.name: *$ AND winlog.channel.keyword: Security
```


- The `AND winlog.channel.keyword: Security` part is to ensure that no unrelated logs are accounted for.




- Finally, let's give our visualization a title by clicking on "No Title".

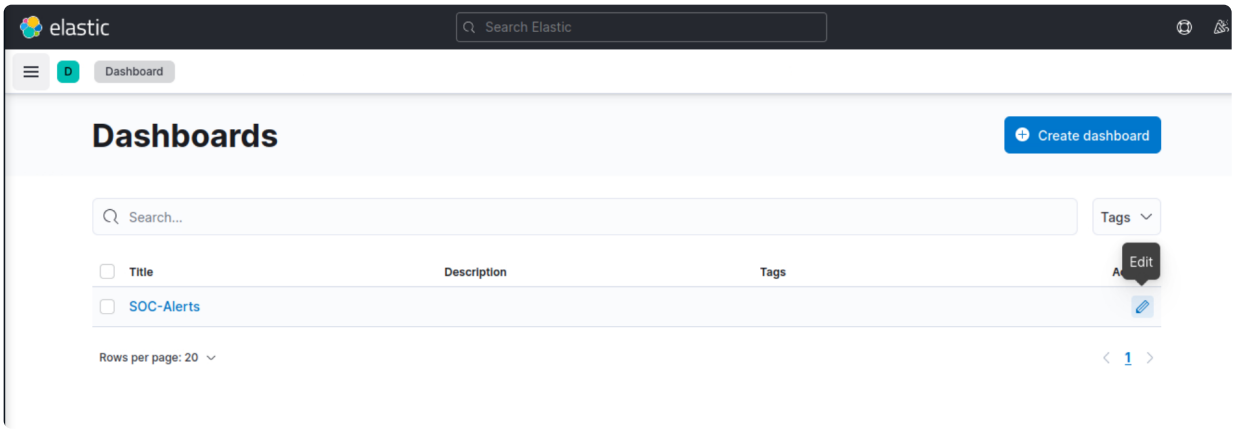



- Don't forget to click on the "Save" button (the one on the upper-right hand side of the window).

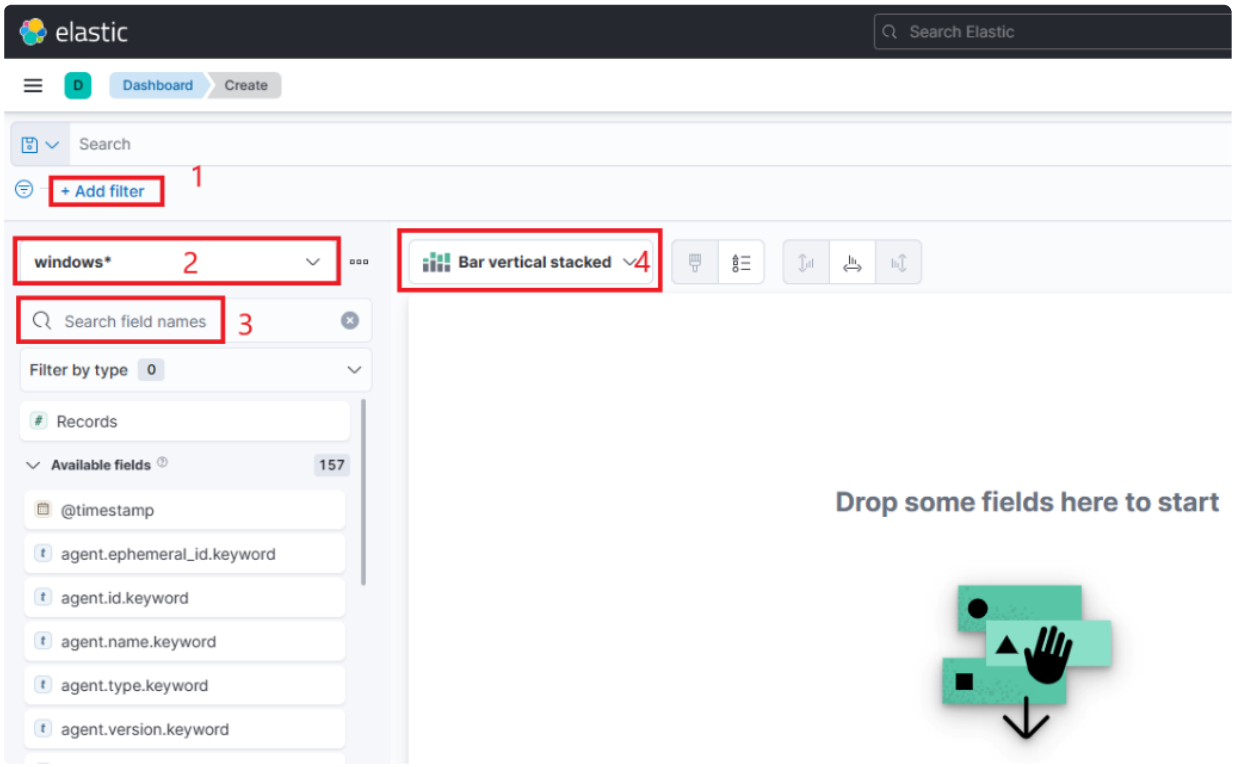
### SIEM Visualization Example 2: Failed Logon Attempts (Disabled Users)

- In this SIEM visualization example we want to create visualization to monitor failed login attempts against disabled users.
- We mention "failed" because it is not possible to log in with a disabled user, so it will never be successful even if the correct credentials are provided.

    - In a scenario where the correct credentials are provided, the Windows logs will contain an additional SubStatus value of 0xC0000072, that indicates the reason of the failure.
- Navigate to `http://[Target IP]:5601`, click on the side navigation toggle, and click on "Dashboard".
- A prebaked dashboard should be visible.
  - Let's click on the "pencil"/edit icon.

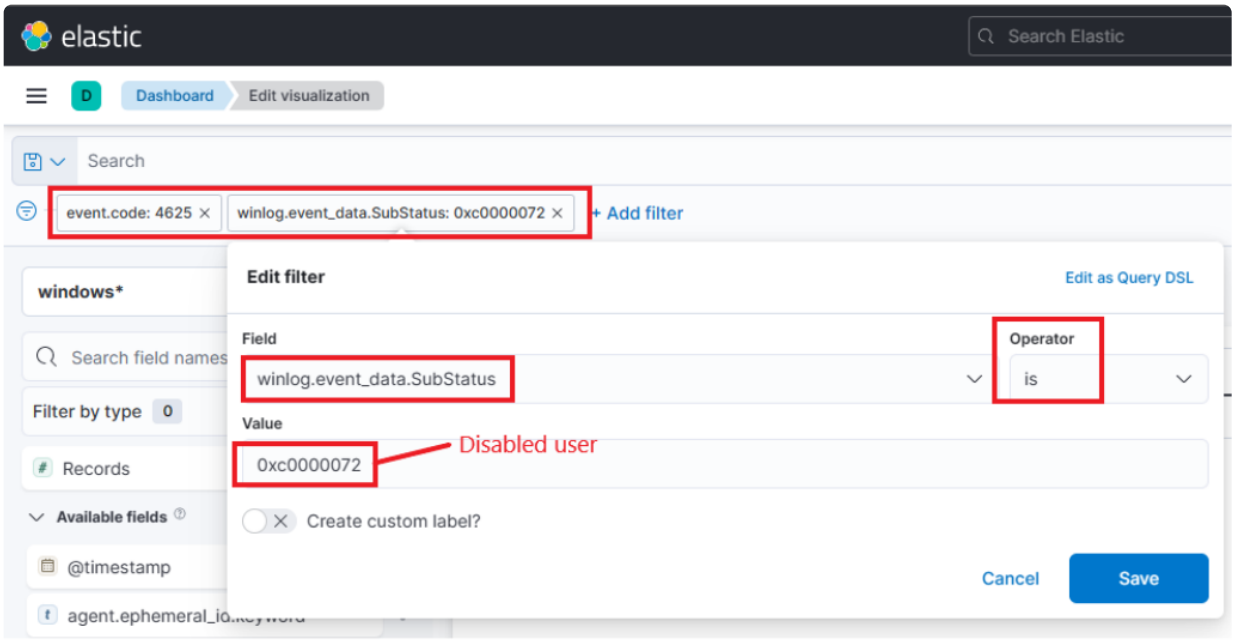



- Now, to initiate the creation of our first visualization, we simply have to click on the "Create visualization" button.
- Upon initiating the creation of our first visualization, the following new window will appear with various options and settings.

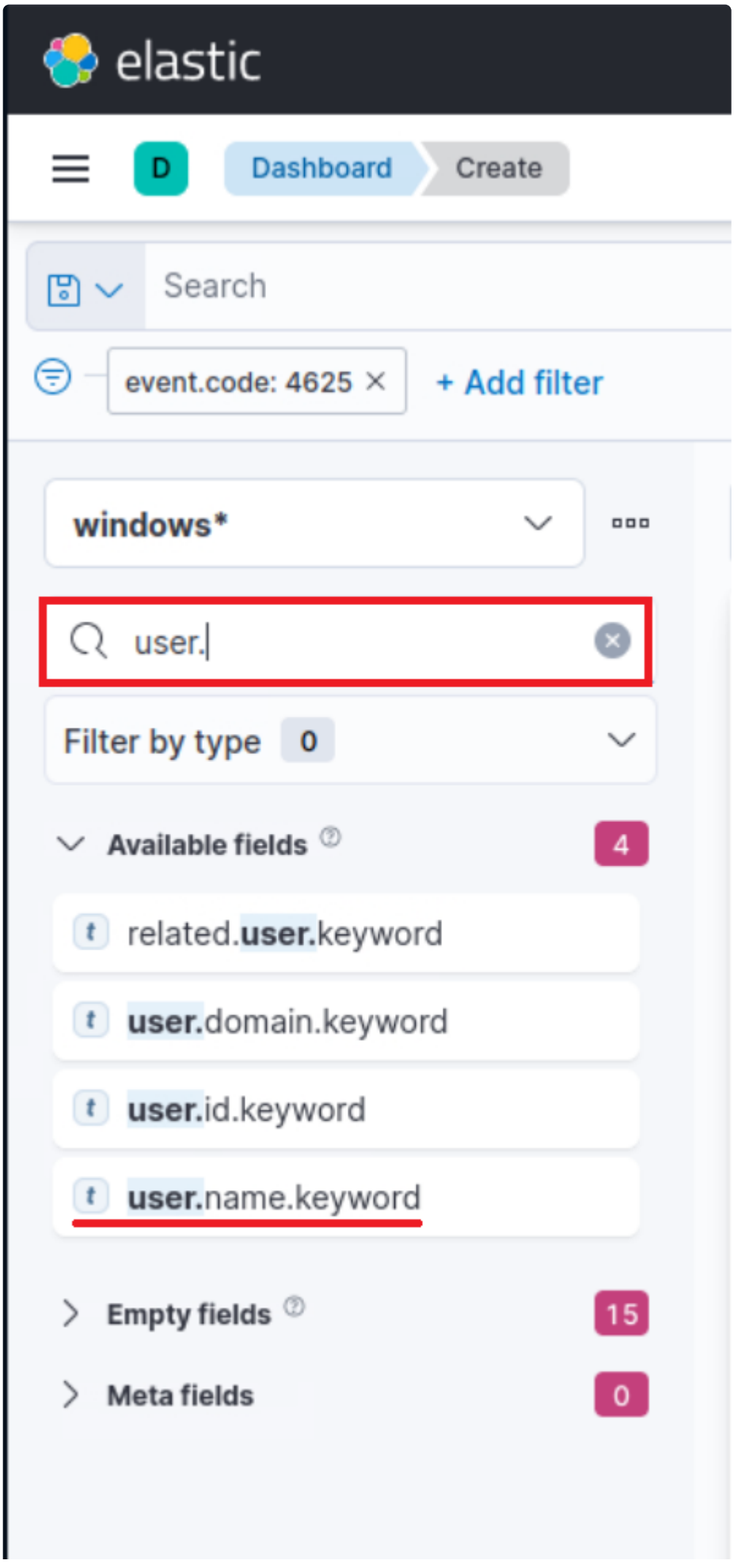



- There are four things for us to notice on this window:

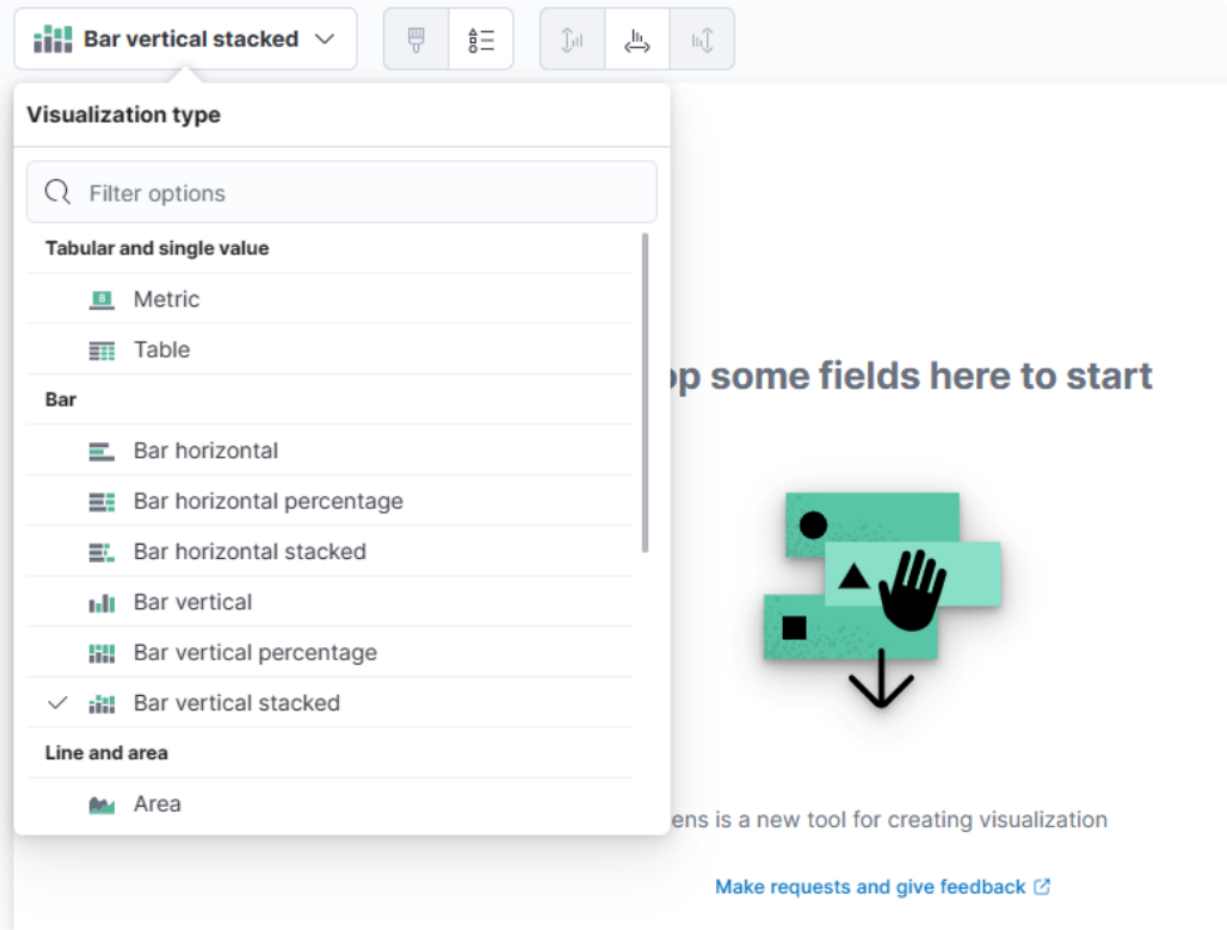
1. A filter option that allows us to filter the data before creating a graph. In this case our goal is to display failed logon attempts against disabled users only.
   - We can use a filter to only consider event IDs that match `4625 – Failed logon attempt on a Windows system`, like we did in the previous visualization example. In this case though, we should also take into account the SubStatus (`winlog.event_data.SubStatus` field) that indicates, when set to 0xC0000072, that the failure is due to a logon with disabled user.
   - The following image demonstrates how we can specify such a filter.




2. This field indicates the data set (index) that we are going to use.
   - It is common for data from various infrastructure sources to be separated into different indices, such as network, Windows, Linux, etc.
   - In this particular example, we will specify `windows*` in the "Index pattern".
3. This search bar provides us with the ability to double-check the existence of a specific field within our data set, serving as another way to ensure that we are looking at the correct data.
   - Like in the previous visualization, we are interested in the `user.name.keyword` field. We can use the search bar to quickly perform a search and verify if this field is present and discovered within our selected data set.
   - This allows us to confirm that we are accessing the desired field and working with accurate data.


elastic
D
Dashboard
Create
Search
event.code: 4625
+ Add filter
windows*
user.
Filter by type
0
Available fields
4
related.user.keyword
user.domain.keyword
user.id.keyword
user.name.keyword
Empty fields
15
Meta fields
0

4. Lastly, this drop-down menu enables us to select the type of visualization we want to create.
   - The default option displayed in the earlier image is "Bar vertical stacked". If we click on that button, it will reveal additional available options (image redacted as not all options fit on the screen).
   - From this expanded list, we can choose the desired visualization type that best suits our requirements and data presentation needs.




- For this visualization, let's select the "Table" option. After selecting the "Table", we can proceed to click on the "Rows" option.
  - This will allow us to choose the specific data elements that we want to include in the

table view.

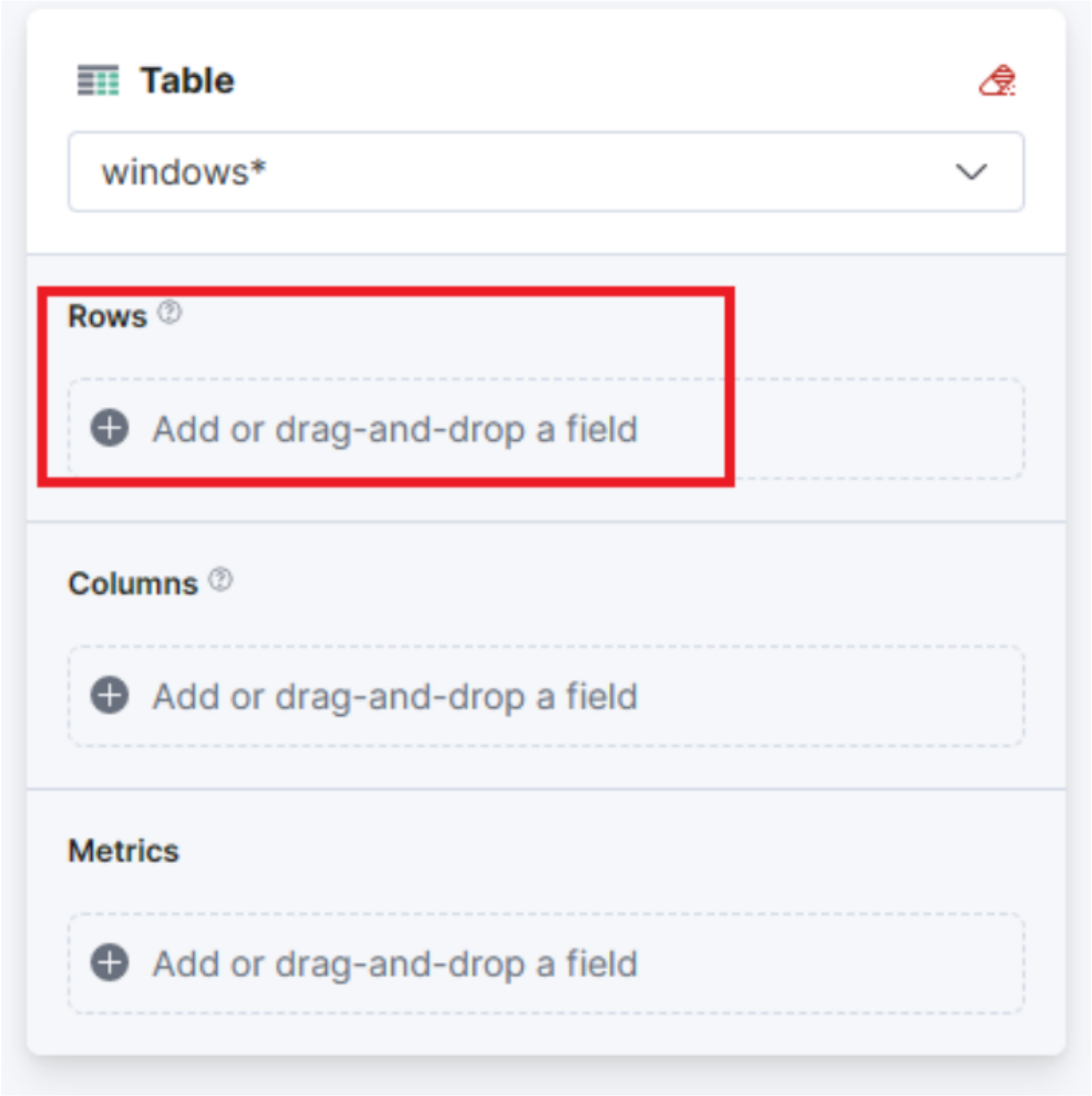
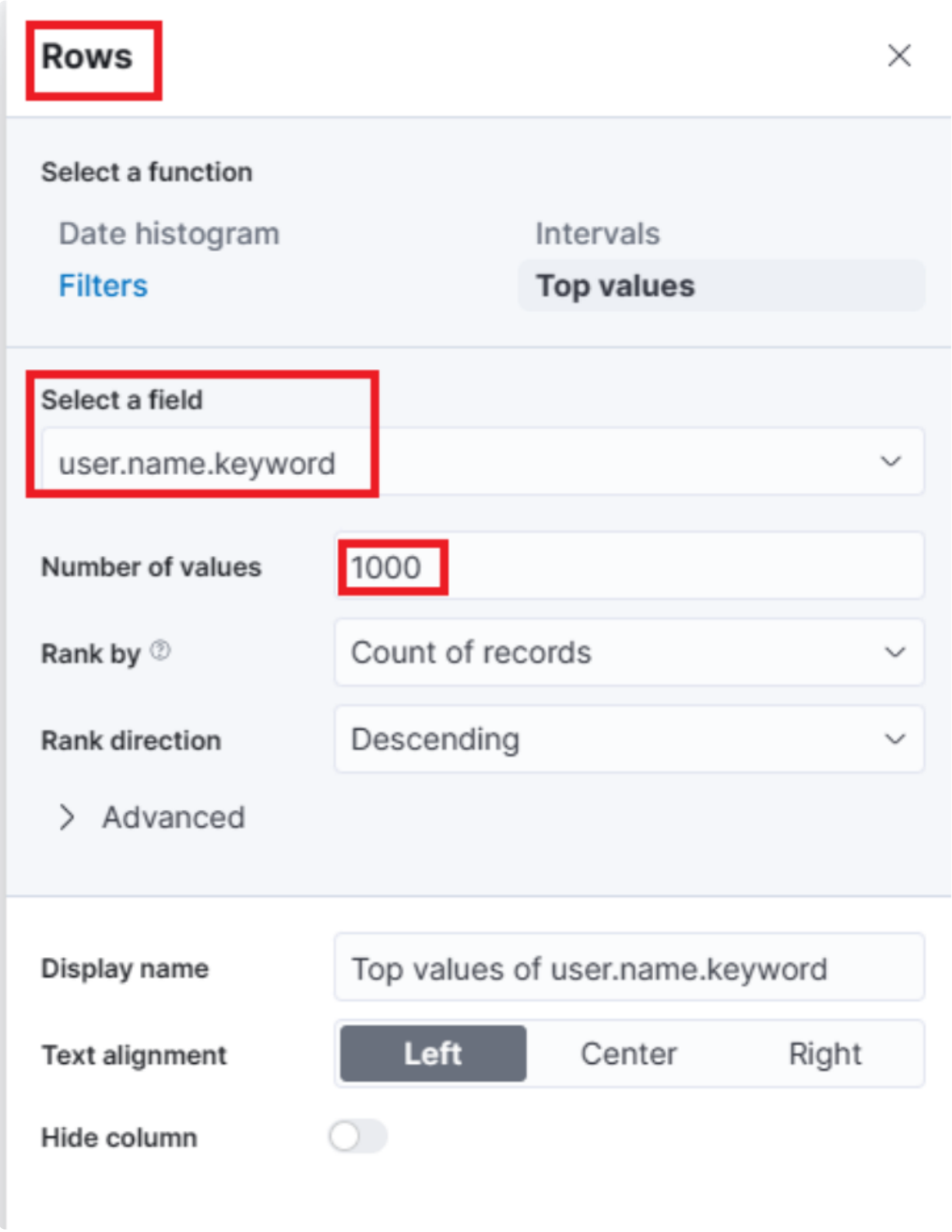


Table
windows*
Rows
Add or drag-and-drop a field
Columns
Add or drag-and-drop a field
Metrics
Add or drag-and-drop a field

- Let's configure the "Rows" settings as follows.




- Moving forward, let's close the "Rows" window and proceed to enter the "Metrics" configuration.

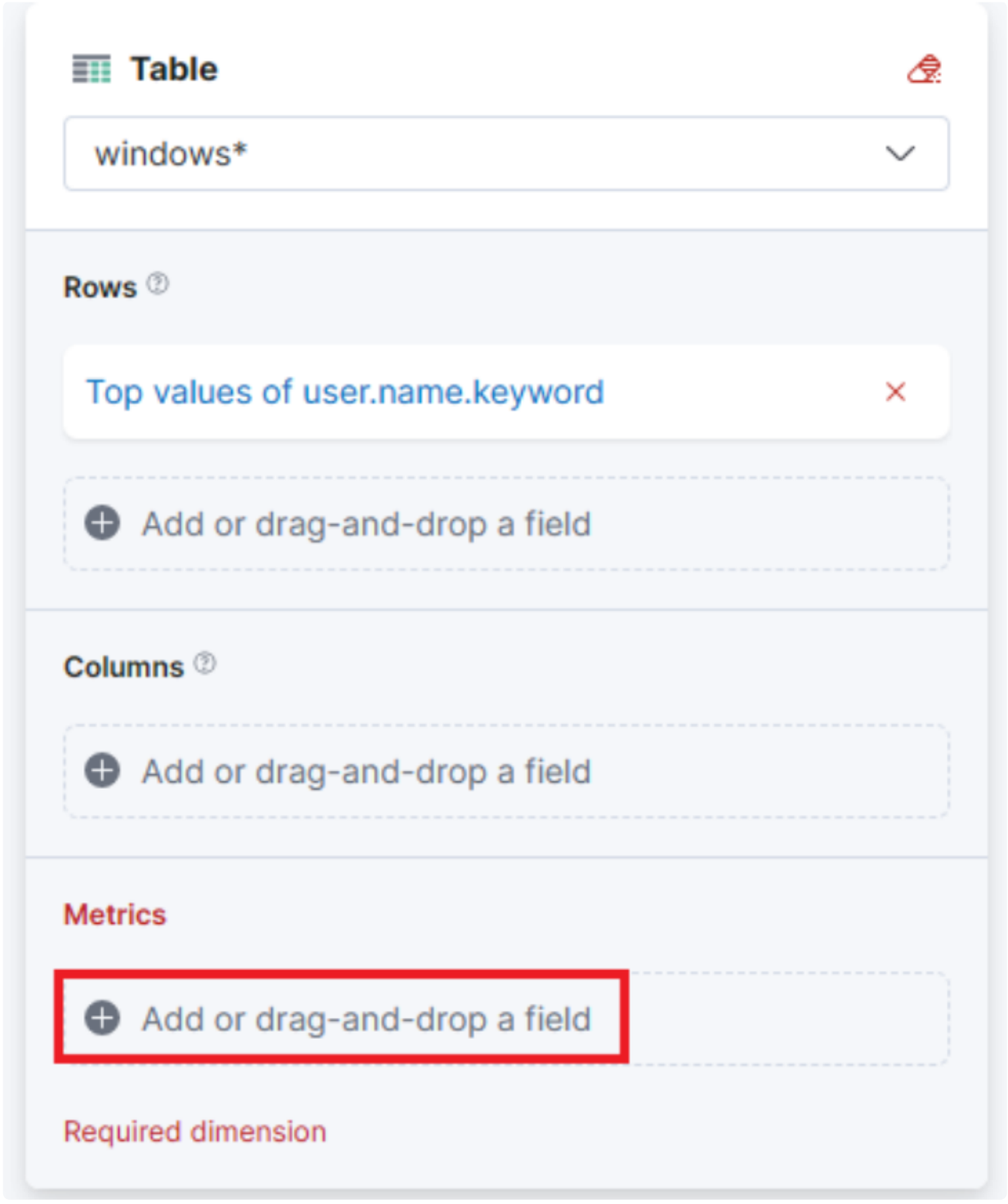
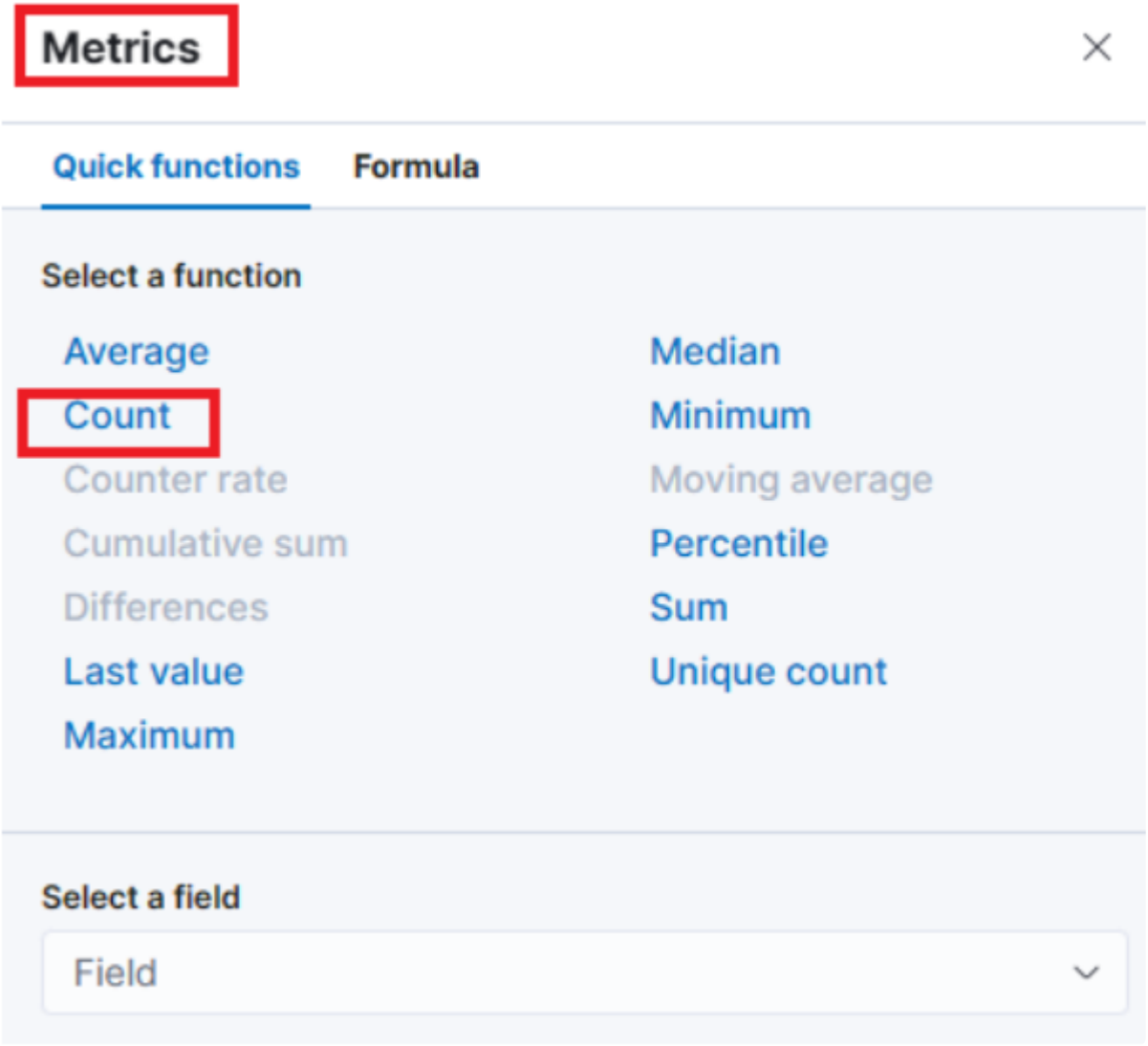
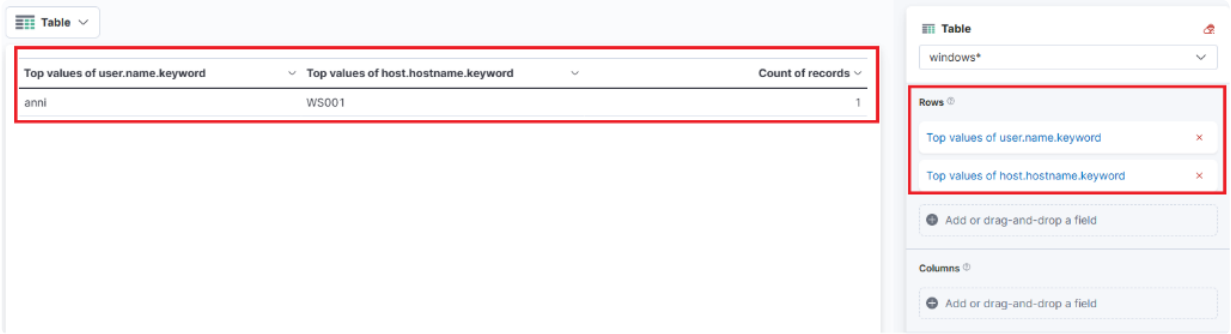


Table
windows*
Rows
Top values of user.name.keyword
Add or drag-and-drop a field
Columns
Add or drag-and-drop a field
Metrics
Add or drag-and-drop a field
Required dimension

- In the "Metrics" window, let's select "count" as the desired metric.




- One final addition to the table is to include another "Rows" setting to show the machine where the failed logon attempt occurred.
  - To do this, we will select the `host.hostname.keyword` field, which represents the computer reporting the failed logon attempt.
  - This will allow us to display the hostname or machine name alongside the count of failed logon attempts, as shown in the image.



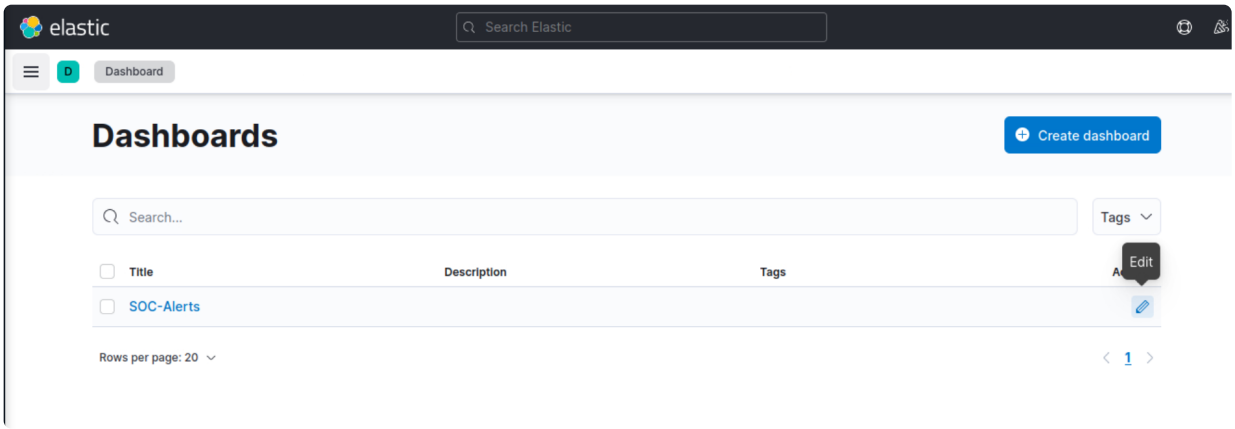

- Now we can see three columns in the table, which contain the following information:
  1. The disabled user whose credentials generated the failed logon attempt event.

2. The machine on which the logon attempt occurred.
3. The number of times the event has occurred (based on the specified time frame or the entire data set, depending on the settings).

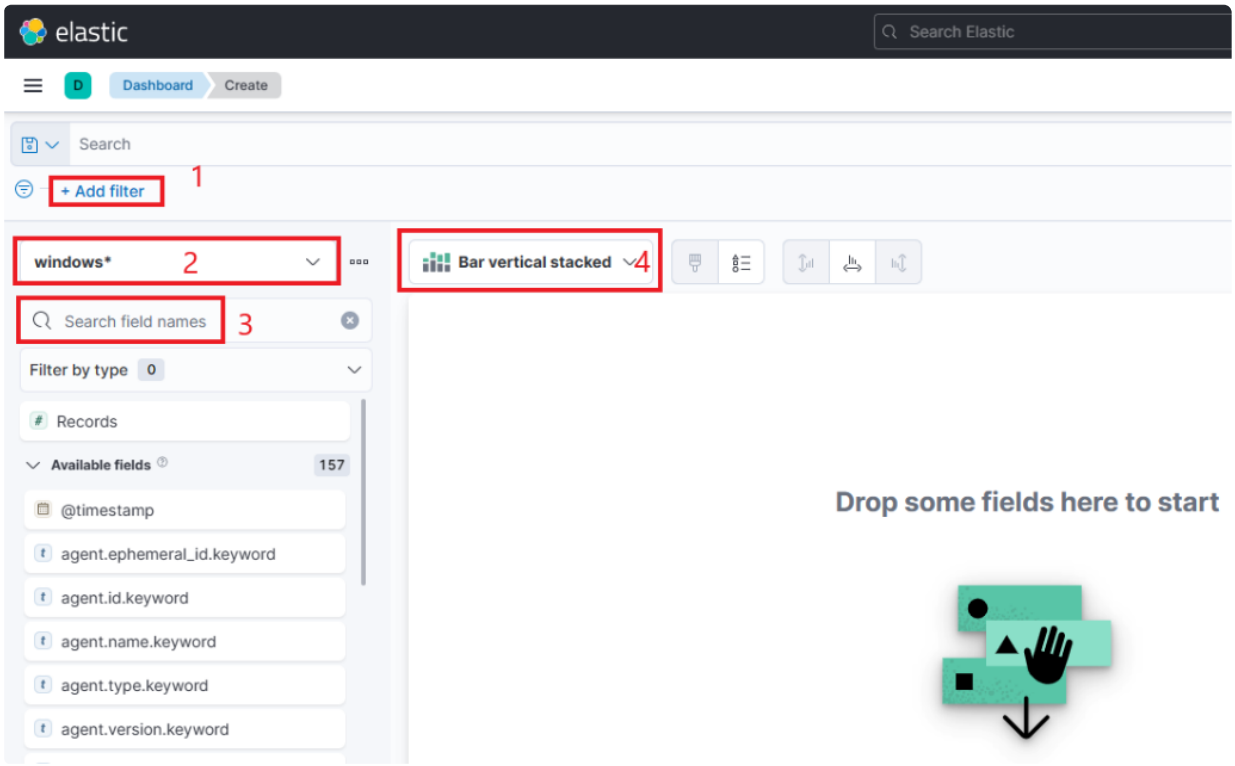
- Finally, click on "Save and return", and you will observe that the new visualization is added to the dashboard.

### SIEM Visualization Example 3: Successful RDP Logon Related To Service Accounts

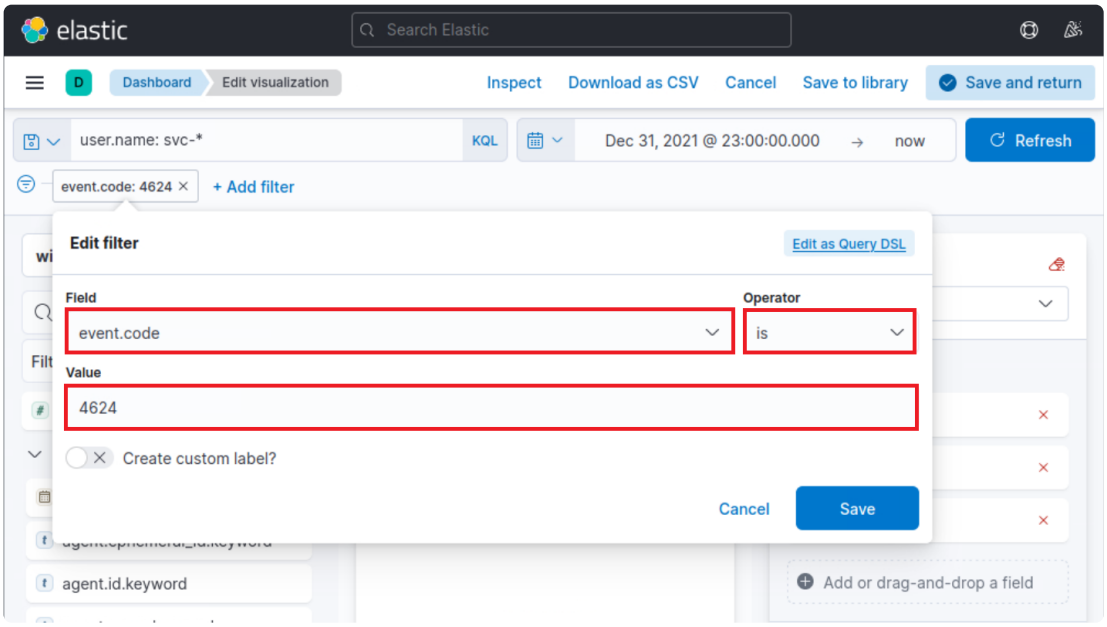
- In this SIEM visualization example, we aim to create a visualization to monitor successful RDP logons specifically related to service accounts.
  - Service account credentials are never used for RDP logons in corporate/real-world environments.
  - We have been informed by the IT Operations department that all service accounts on the environment start with `svc-`.
- The motivation for this visualization stems from the fact that service accounts often possess exceptionally high privileges.
  - We need to keep a close eye on how service accounts are used.
- Our visualization will be based on the following Windows event log.
  - 4624: An account was successfully logged on
- Navigate to the bottom of this section and click on `Click here to spawn the target system!`.
- Navigate to `http://[Target IP]:5601`, click on the side navigation toggle, and click on "Dashboard".
- A prebaked dashboard should be visible. Let's click on the "pencil"/edit icon.

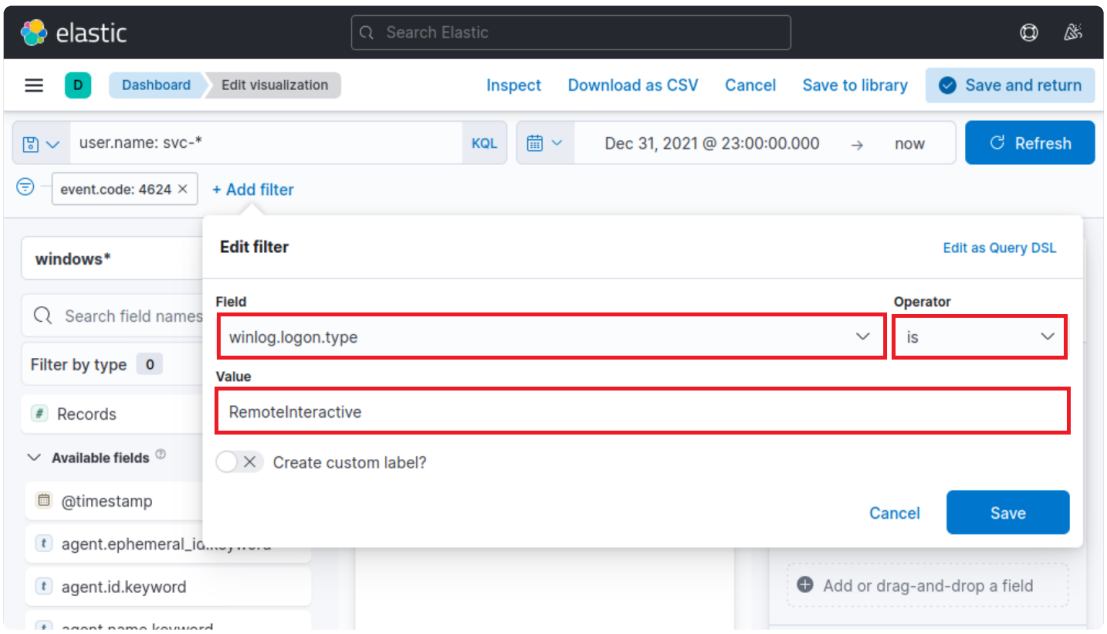



- Now, to initiate the creation of our first visualization, we simply have to click on the "Create visualization" button.
- Upon initiating the creation of our first visualization, the following new window will appear with various options and settings

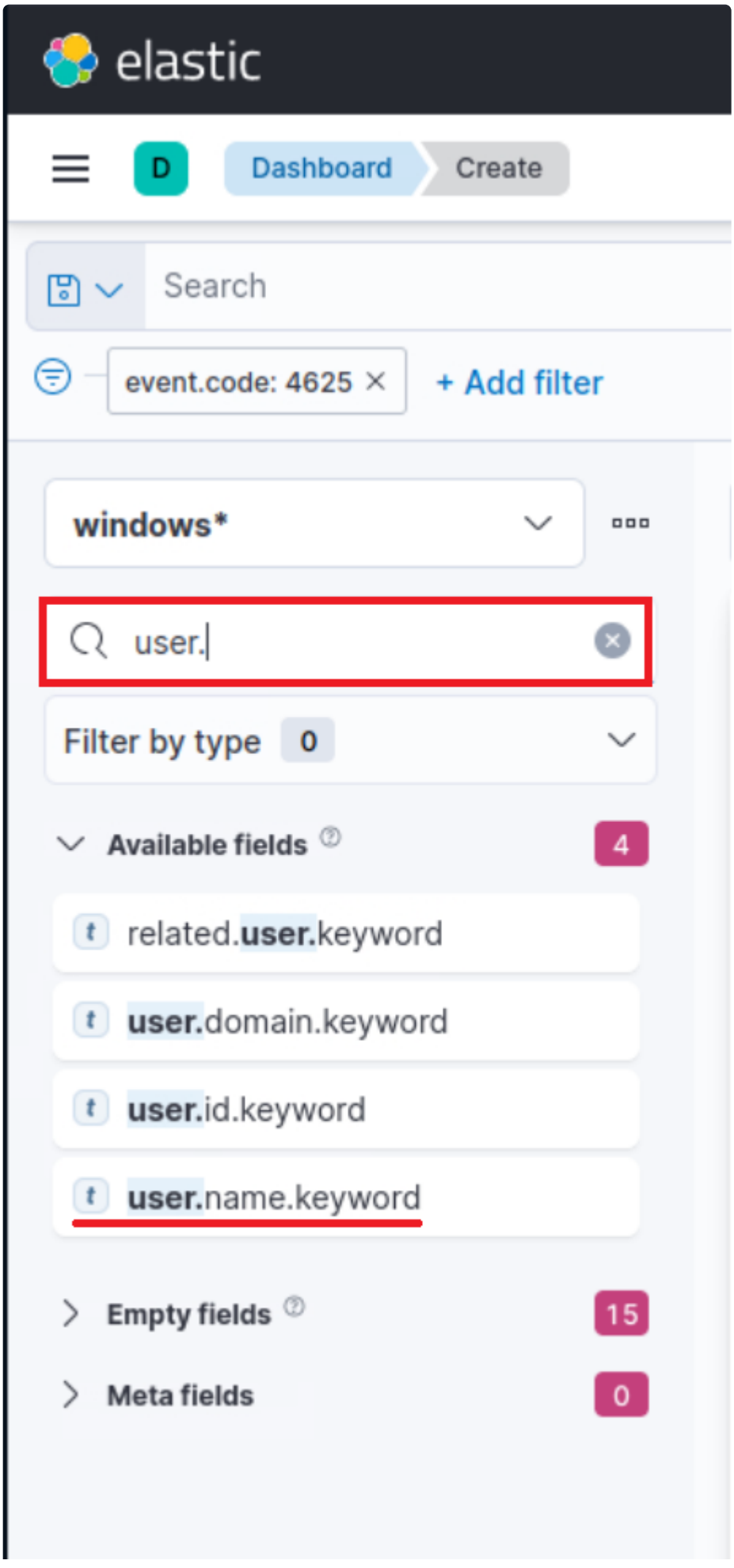



- There are five things for us to notice on this window:
  1. A filter option that allows us to filter the data before creating a graph. In this case our goal is to display successful RDP logons specifically related to service accounts.
     - We can use a filter to only consider event IDs that match `4624 – An account was successfully logged on`.
     - In this case though, we should also take into account the logon type which should be `RemoteInteractive` (`winlog.logon.type` field).
     - The following images demonstrates how we can specify such filters.

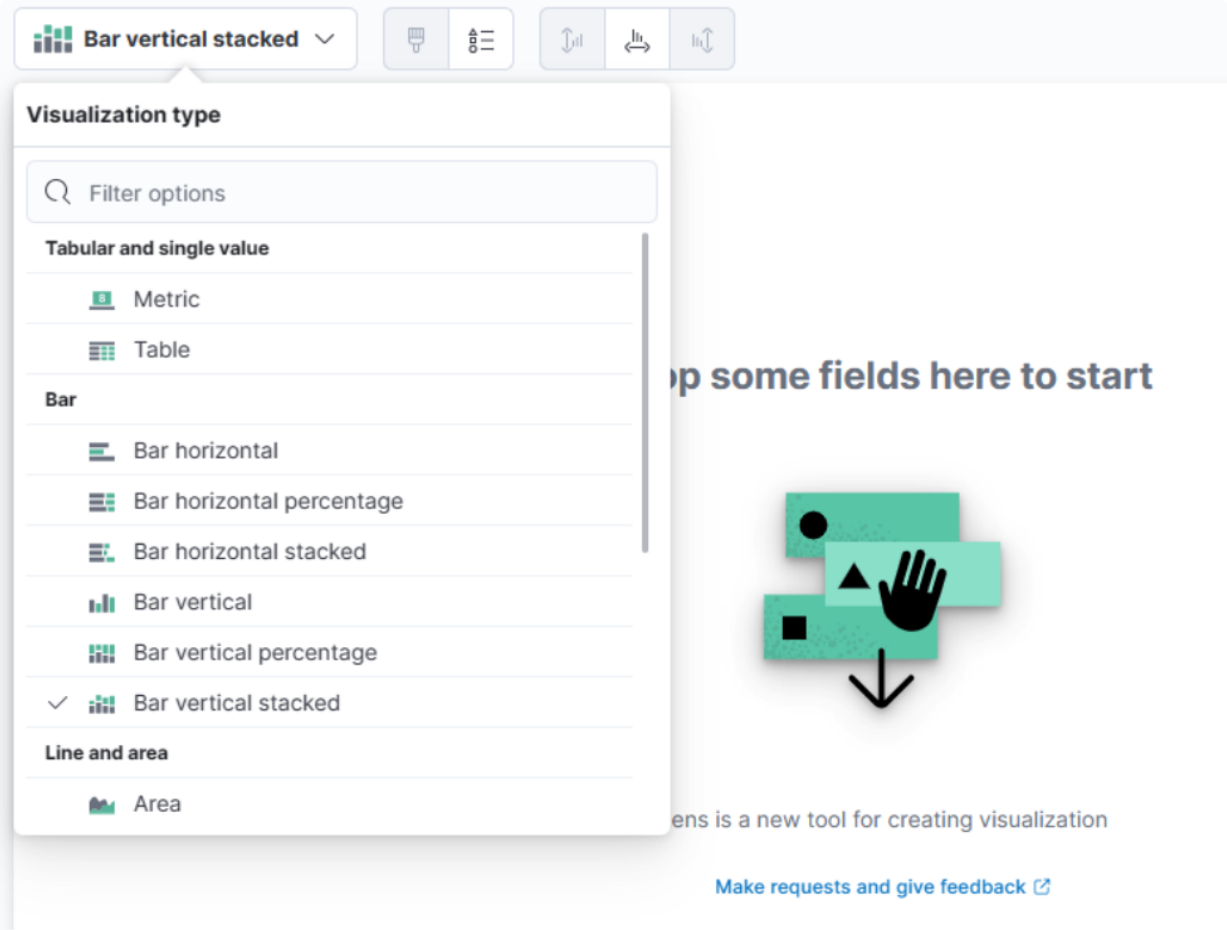





2. This field indicates the data set (index) that we are going to use. It is common for data from various infrastructure sources to be separated into different indices, such as network, Windows, Linux, etc.
    - In this particular example, we will specify `windows*` in the "Index pattern".
3. This search bar provides us with the ability to double-check the existence of a specific field within our data set, serving as another way to ensure that we are looking at the correct data.
    - We are interested in the `user.name.keyword` field.
    - We can use the search bar to quickly perform a search and verify if this field is present and discovered within our selected data set.
    - This allows us to confirm that we are accessing the desired field and working with accurate data.


elastic
D
Dashboard
Create
Search
event.code: 4625
+ Add filter
windows*
user.
Filter by type
0
Available fields
4
related.user.keyword
user.domain.keyword
user.id.keyword
user.name.keyword
Empty fields
15
Meta fields
0

4. Lastly, this drop-down menu enables us to select the type of visualization we want to create.
   - The default option displayed in the earlier image is "Bar vertical stacked".
   - If we click on that button, it will reveal additional available options (image redacted as not all options fit on the screen).
   - From this expanded list, we can choose the desired visualization type that best suits our requirements and data presentation needs.




- For this visualization, let's select the "Table" option. After selecting the "Table", we can proceed to click on the "Rows" option.
  - This will allow us to choose the specific data elements that we want to include in the

table view.

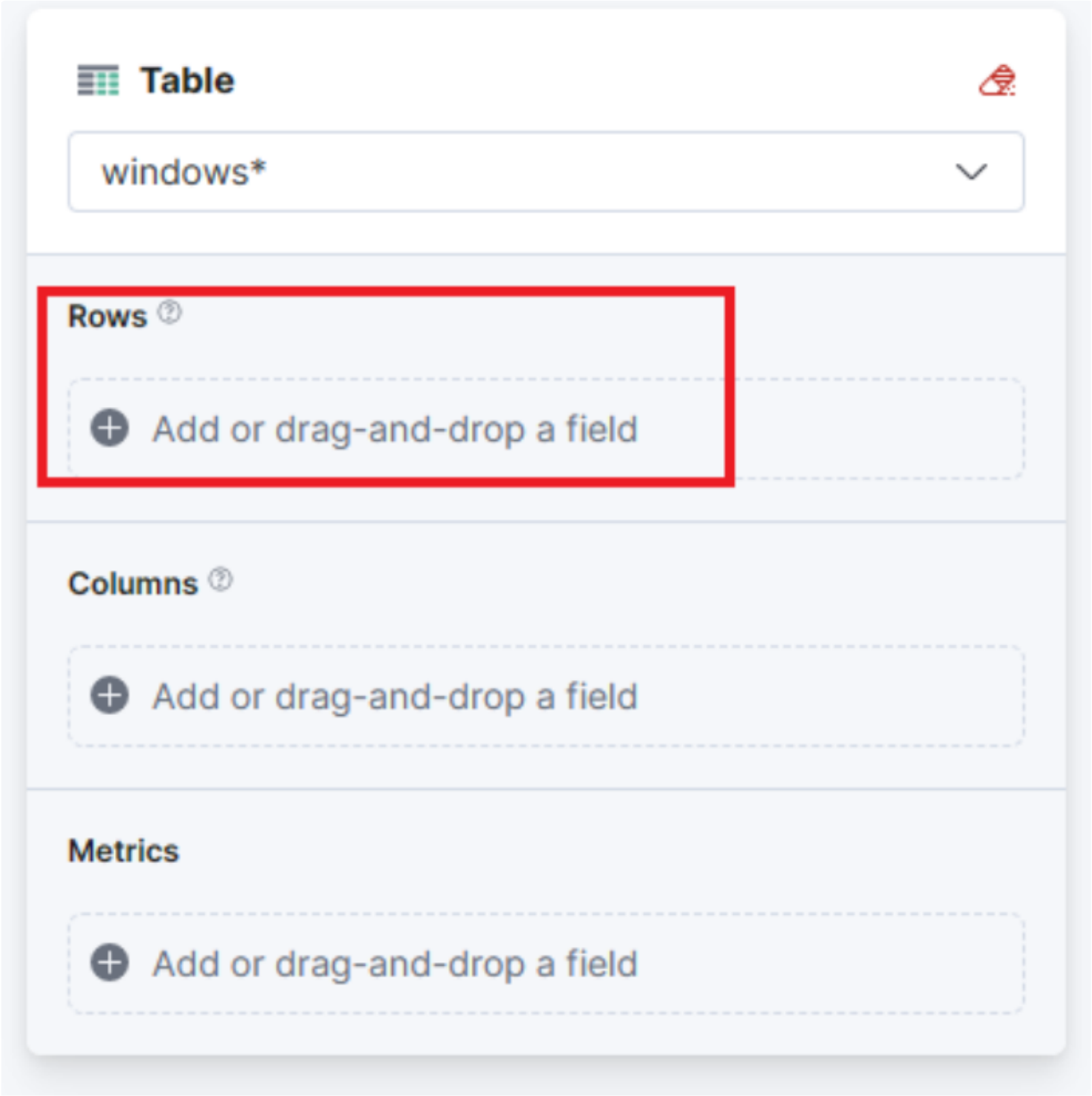
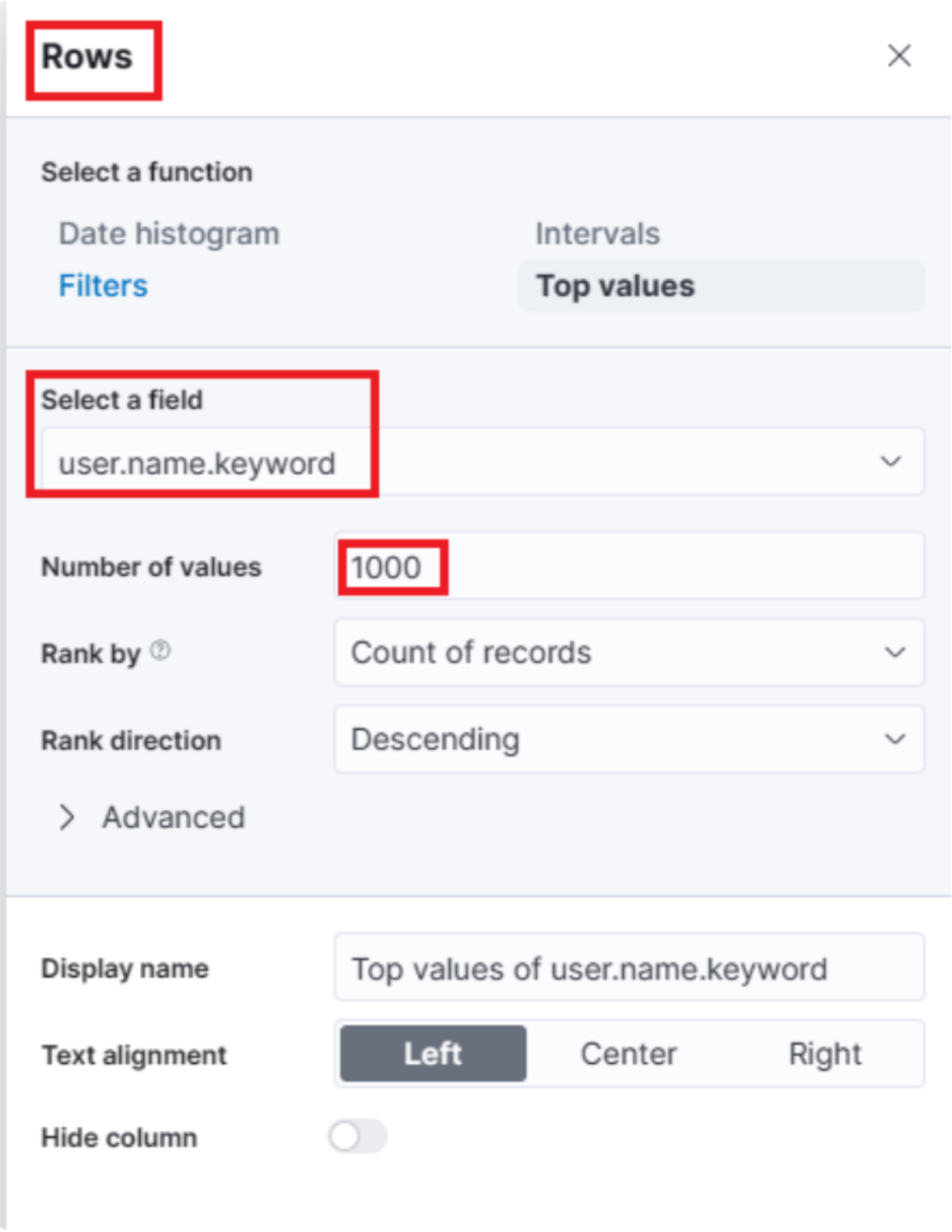


Table
windows*
Rows
Add or drag-and-drop a field
Columns
Add or drag-and-drop a field
Metrics
Add or drag-and-drop a field

- Let's configure the "Rows" settings as follows.




- Moving forward, let's close the "Rows" window and proceed to enter the "Metrics" configuration.

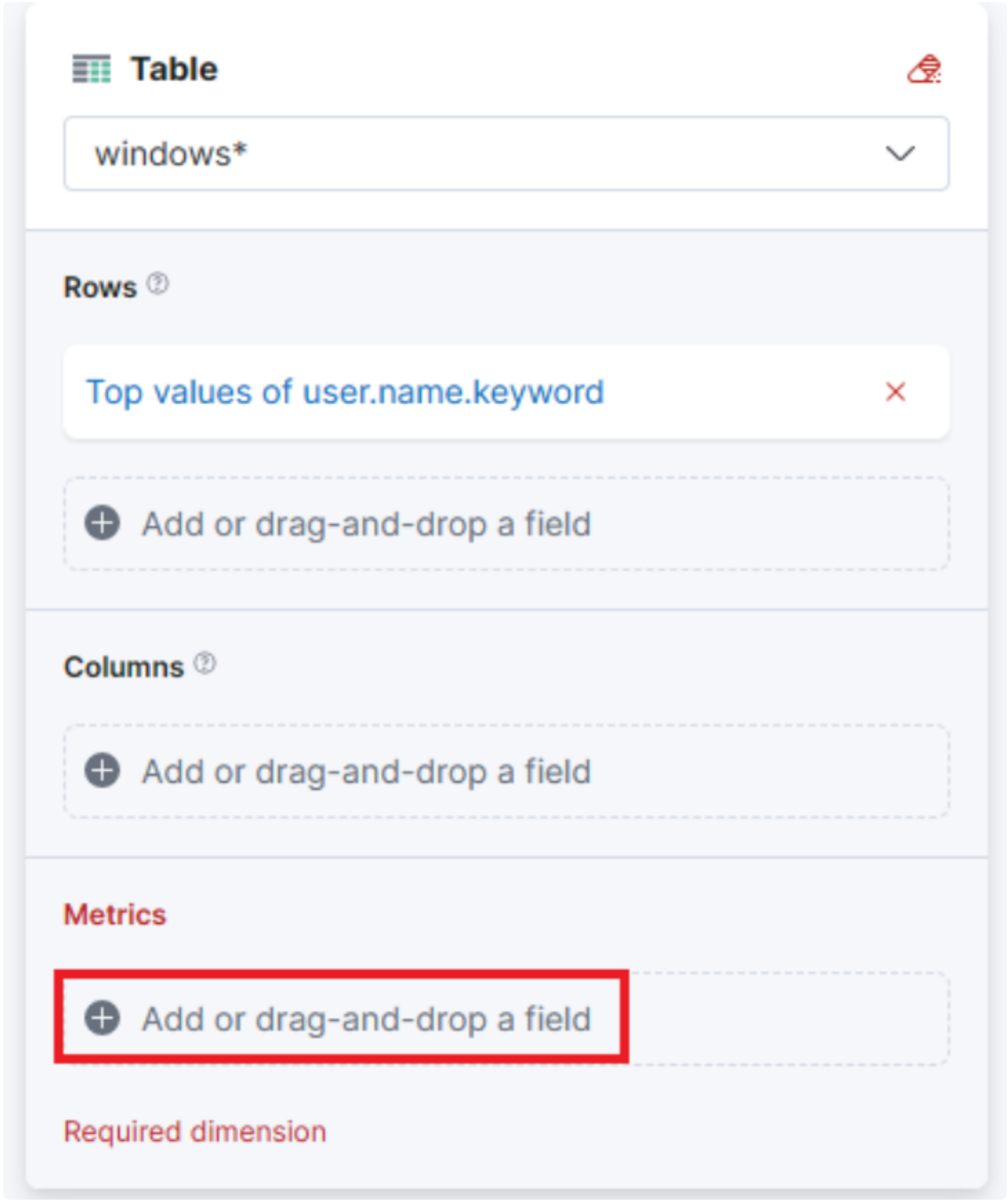
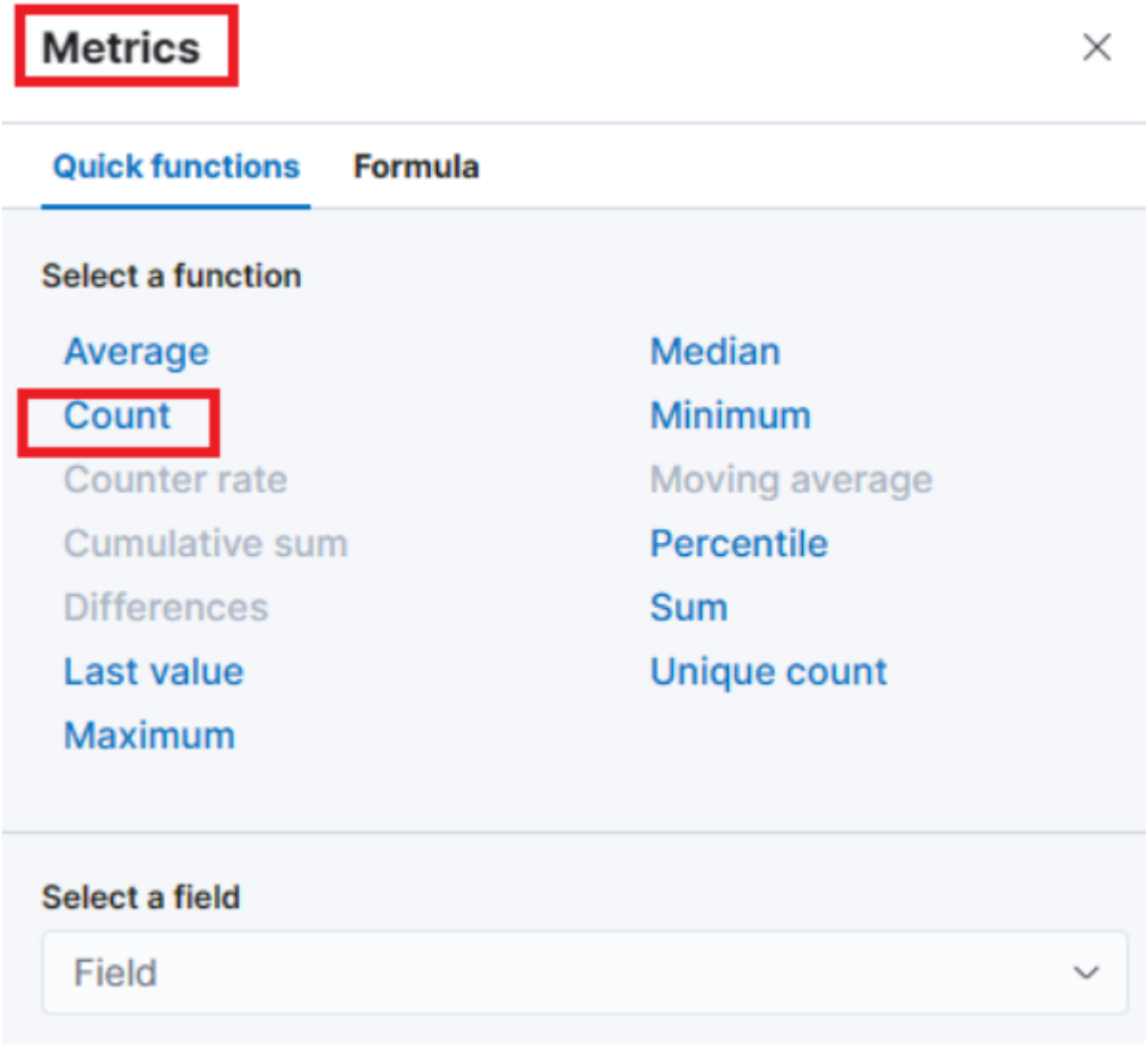
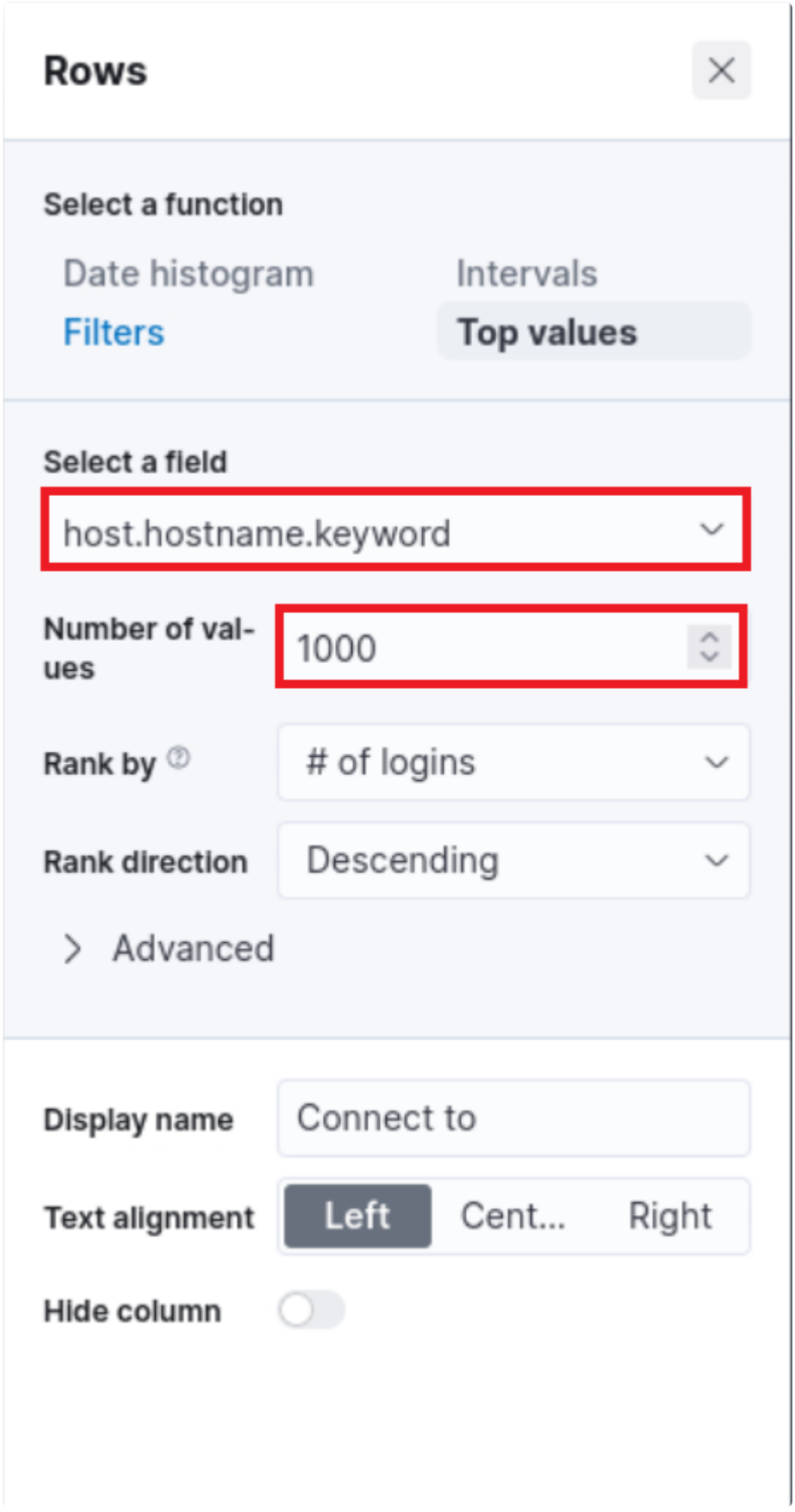


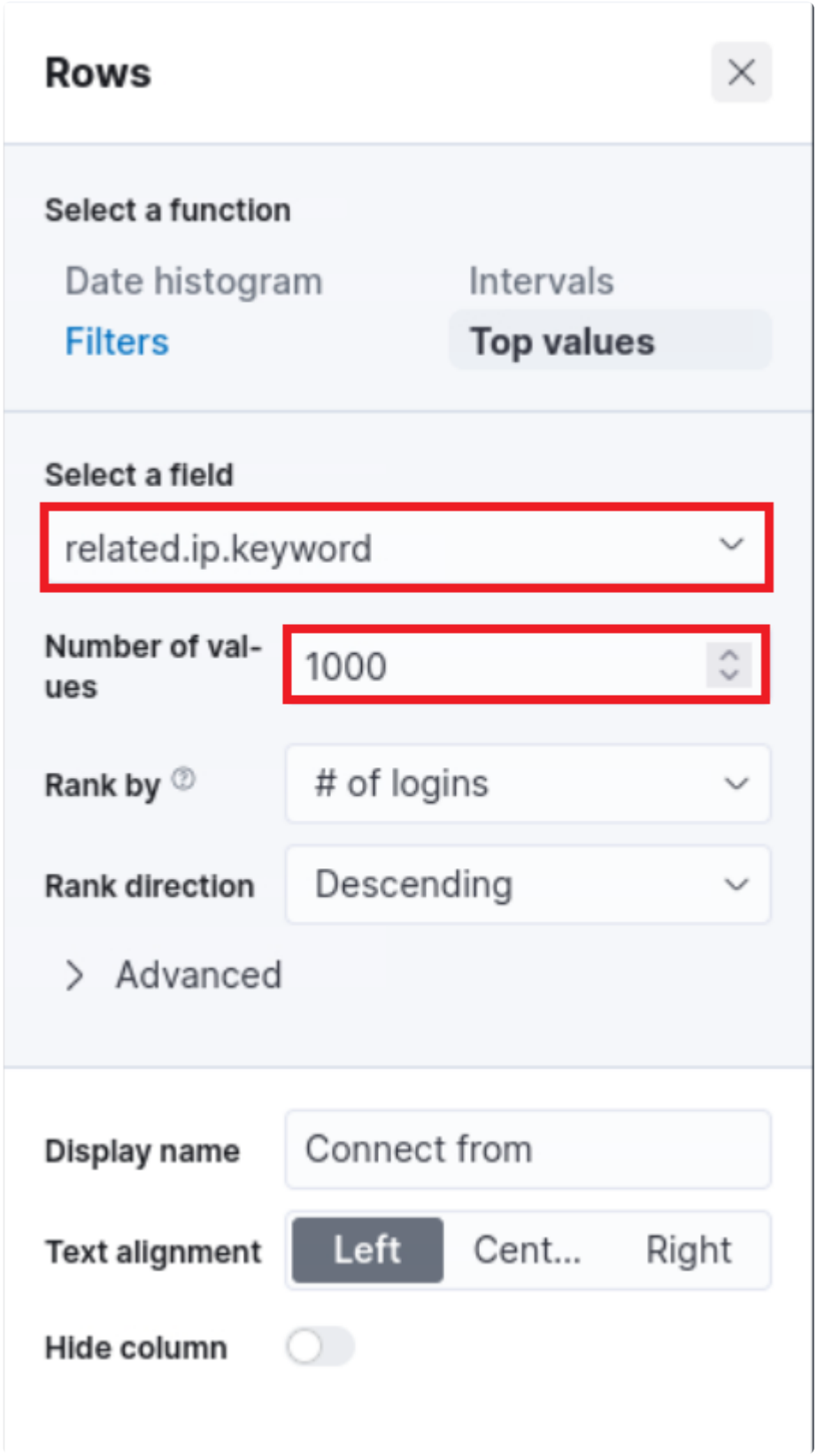
Table
windows*
Rows
Top values of user.name.keyword
Add or drag-and-drop a field
Columns
Add or drag-and-drop a field
Metrics
Add or drag-and-drop a field
Required dimension

- In the "Metrics" window, let's select "count" as the desired metric.




- One final addition to the table is to include two more "Rows" settings to show the machine where the successful RDP logon attempt occurred and the machine that initiated the successful RDP logon attempt.
  - To do this, we will select the `host.hostname.keyword` field that represents the computer reporting the successful RDP logon attempt and the `related.ip.keyword` field that represents the IP of the computer initiating the succsessful RDP logon attempt.
  - This will allow us to display the involved machines alongside the count of successful logon attempts, as shown in the image.

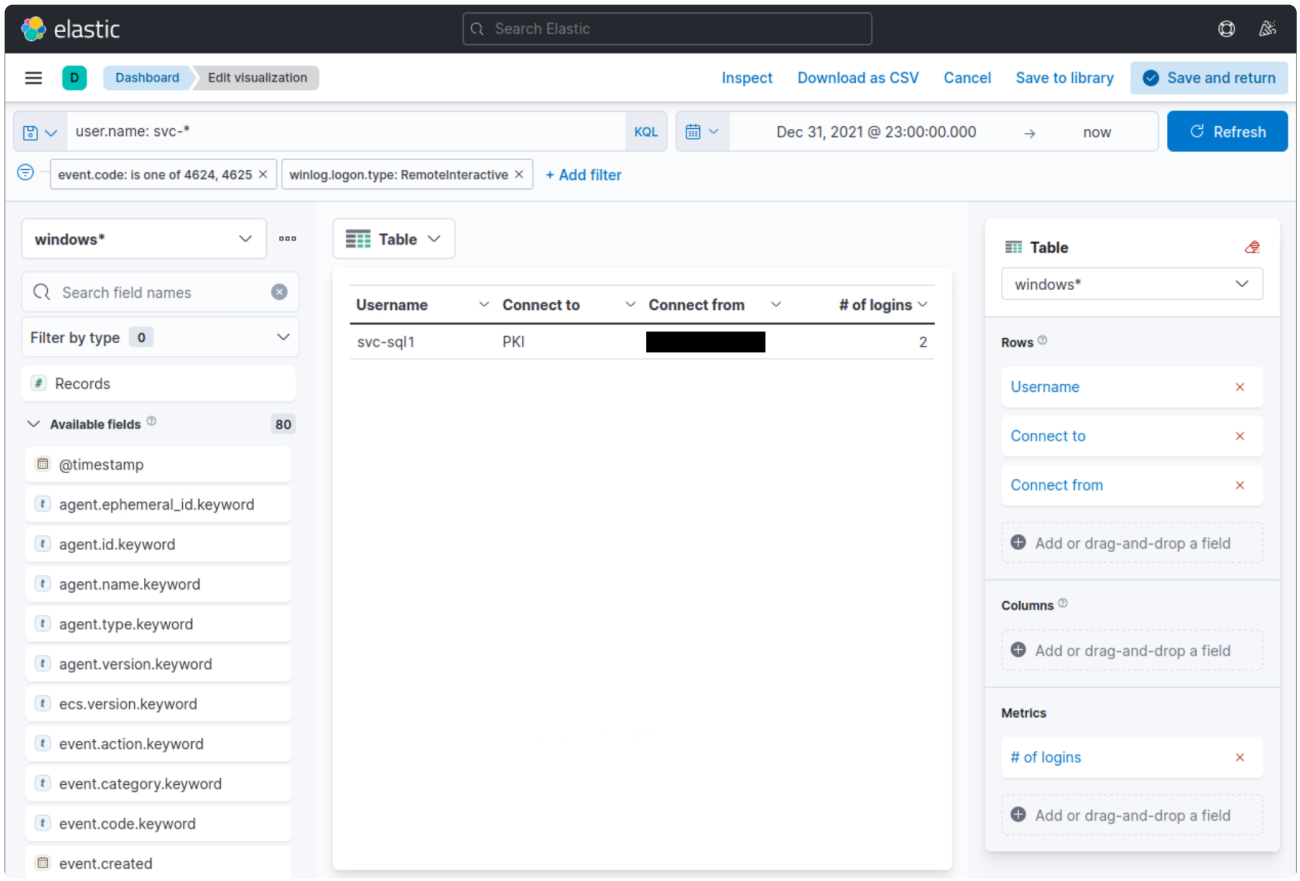

Rows
Select a function
Date histogram
Intervals
Filters
Top values
Select a field
host.hostname.keyword
Number of val-ues
1000
Rank by
# of logins
Rank direction
Descending
Advanced
Display name
Connect to
Text alignment
Left
Cent...
Right
Hide column




- As discussed, we want to monitor successful RDP logons specifically related to service accounts, knowing for a fact that all service accounts of the environment start with `svc-`.
    - So, to conclude our visualization we need to specify the following KQL query.

```
user.name: svc-*
```

-> **Note**: As you can see we don't use the `.keyword` field in KQL queries.

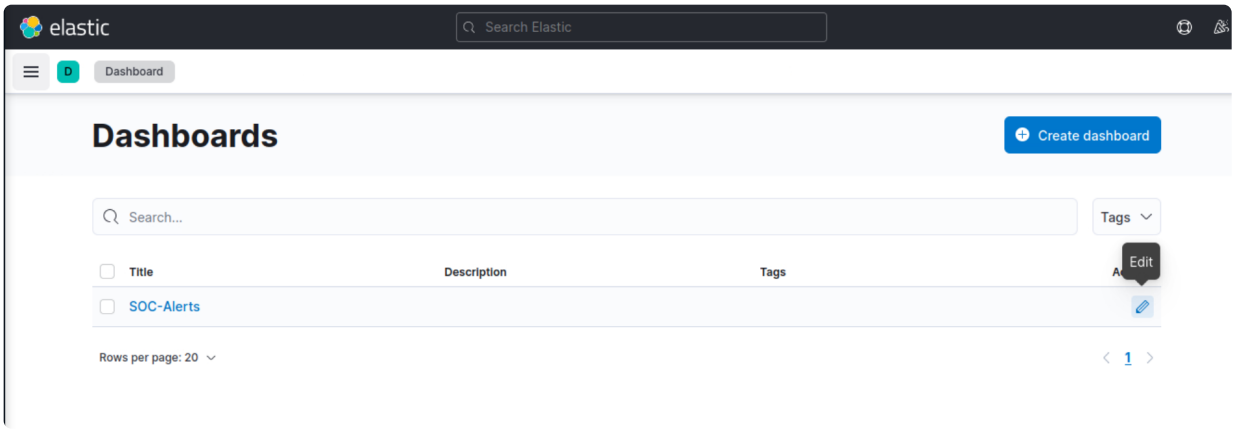



- Now we can see four columns in the table, which contain the following information:
    1. The service account whose credentials generated the successful RDP logon attempt event.
    2. The machine on which the logon attempt occurred.
    3. The IP of the machine that initiated the logon attempt.
    4. The number of times the event has occurred (based on the specified time frame or the entire data set, depending on the settings).
- Finally, click on "Save and return", and you will observe that the new visualization is added to the dashboard.

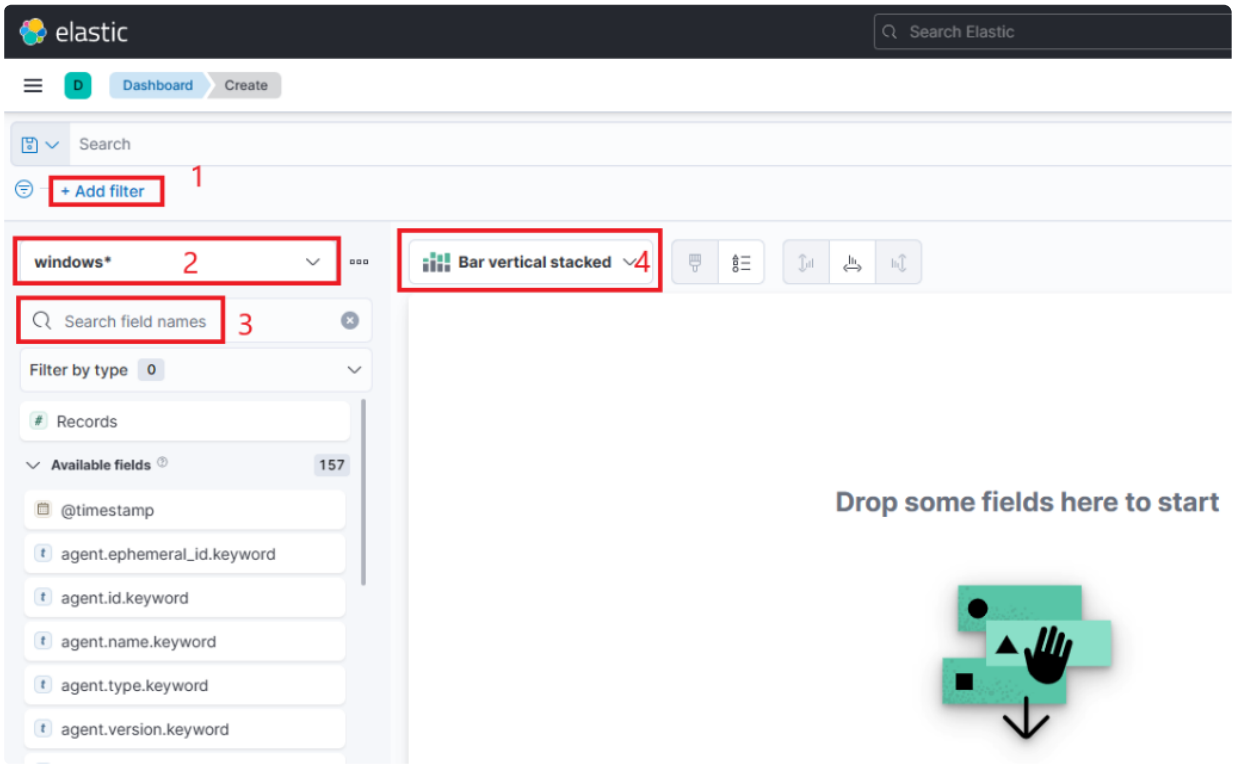
### SIEM Visualization Example 4: Users Added Or Removed From A Local Group (Within A Specific Timeframe)

Overview

- In this SIEM visualization example, we aim to create a visualization to monitor user additions or removals from the local "Administrators" group from March 5th 2023 to date.
- Our visualization will be based on the following Windows event logs.
  - 4732: A member was added to a security-enabled local group
  - 4733: A member was removed from a security-enabled local group
- Navigate to the bottom of this section and click on `Click here to spawn the target system!`.
- Navigate to `http://[Target IP]:5601`, click on the side navigation toggle, and click on "Dashboard".
- A prebaked dashboard should be visible. Let's click on the "pencil"/edit icon.

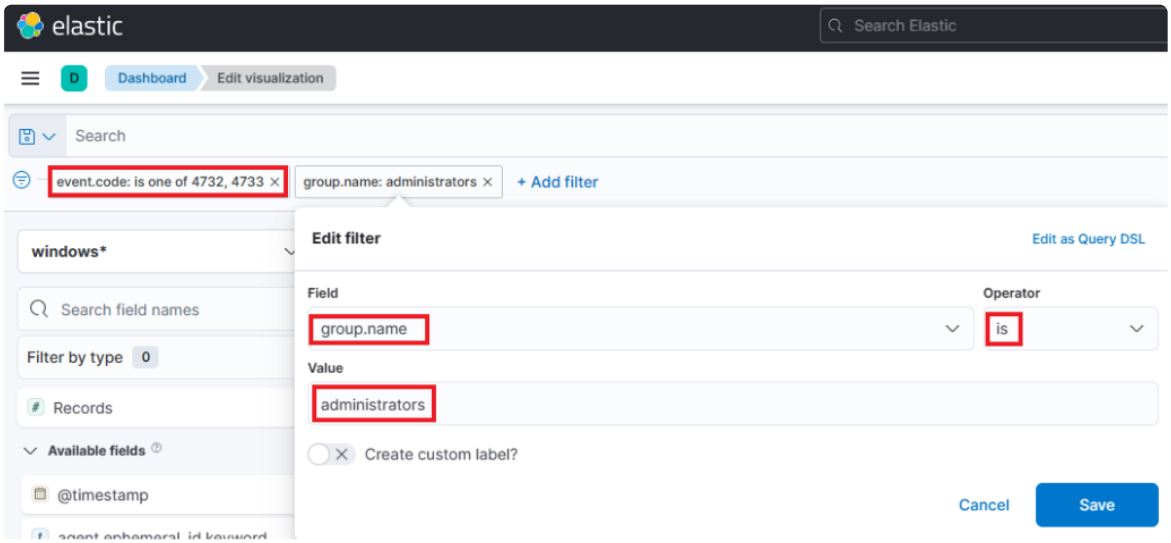



- Now, to initiate the creation of our first visualization, we simply have to click on the "Create visualization" button.
- Upon initiating the creation of our first visualization, the following new window will appear with various options and settings.




There are five things for us to notice on this window:

1. A filter option that allows us to filter the data before creating a graph. In this case our goal is to display user additions or removals from the local "Administrators" group.
   - We can use a filter to only consider event IDs that match `4732 – A member was added to a security-enabled local group` and `4733 – A member was removed from a security-enabled local group`.
   - We can also use a filter to only consider 4732 and 4733 events where the local group is the "Administrators" one.

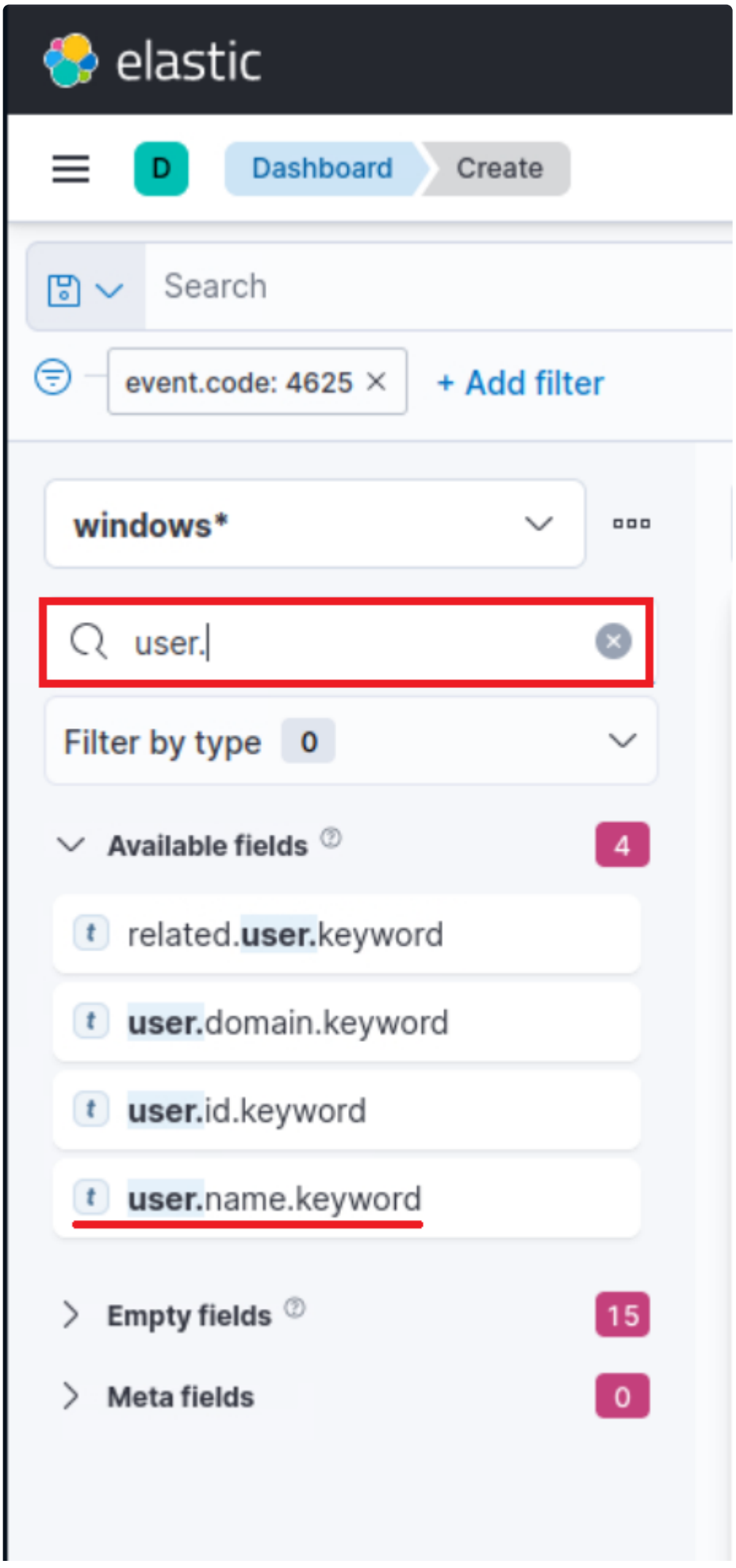



2. This field indicates the data set (index) that we are going to use.
   - It is common for data from various infrastructure sources to be separated into different indices, such as network, Windows, Linux, etc.

- In this particular example, we will specify `windows*` in the "Index pattern".

3. This search bar provides us with the ability to double-check the existence of a specific field within our data set, serving as another way to ensure that we are looking at the correct data.
   - We are interested in the `user.name.keyword` field. We can use the search bar to quickly perform a search and verify if this field is present and discovered within our selected data set.
   - This allows us to confirm that we are accessing the desired field and working with

accurate data.


elastic
D
Dashboard
Create
Search
event.code: 4625
+ Add filter
windows*
user.
Filter by type 0
Available fields 4
related.user.keyword
user.domain.keyword
user.id.keyword
user.name.keyword
Empty fields 15
Meta fields 0

4. Lastly, this drop-down menu enables us to select the type of visualization we want to create.
    - The default option displayed in the earlier image is "Bar vertical stacked".
    - If we click on that button, it will reveal additional available options (image redacted as not all options fit on the screen).
    - From this expanded list, we can choose the desired visualization type that best suits our requirements and data presentation needs.

- For this visualization, let's select the "Table" option. After selecting the "Table", we can proceed to click on the "Rows" option.
  - This will allow us to choose the specific data elements that we want to include in the

table view.

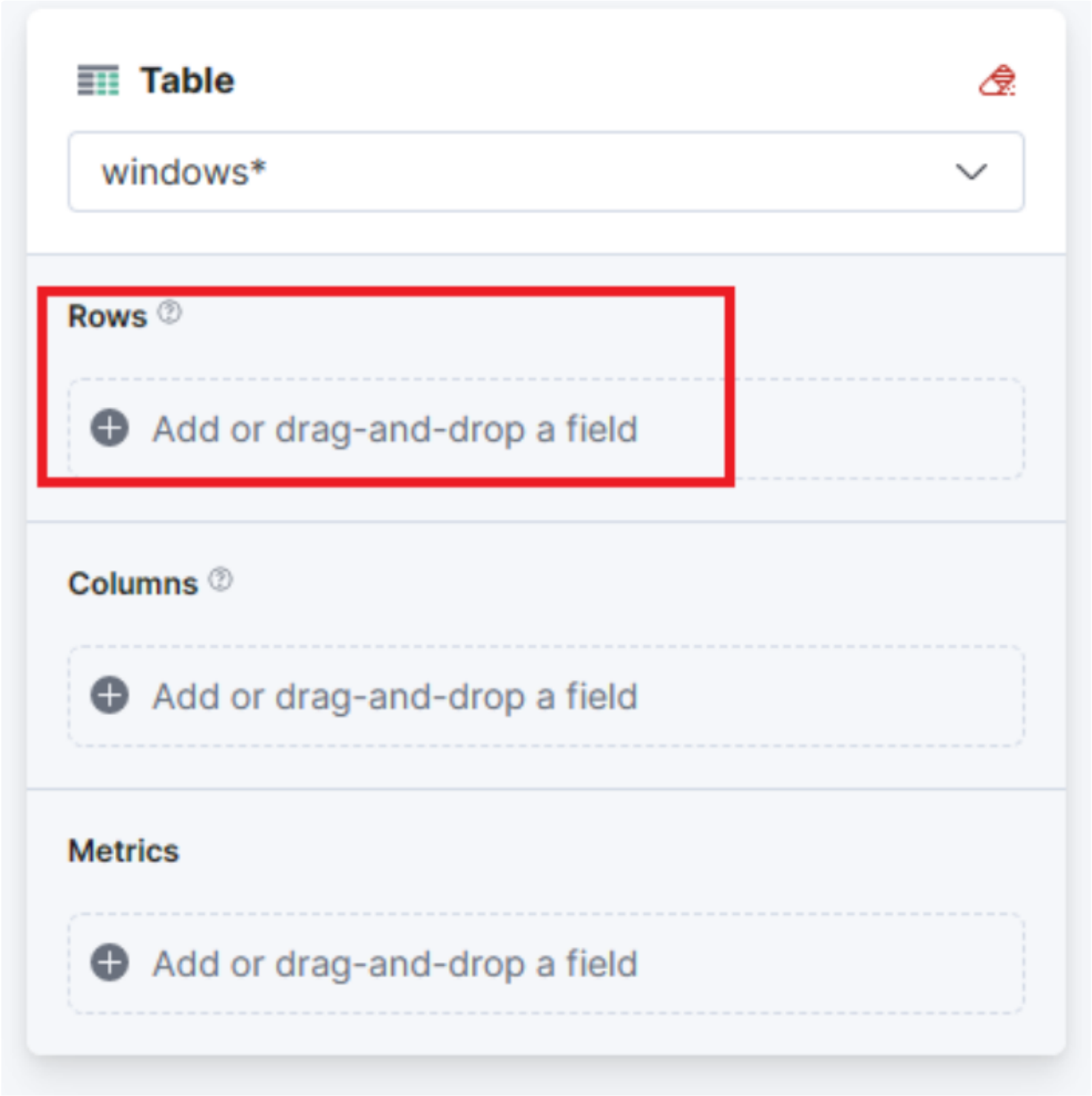
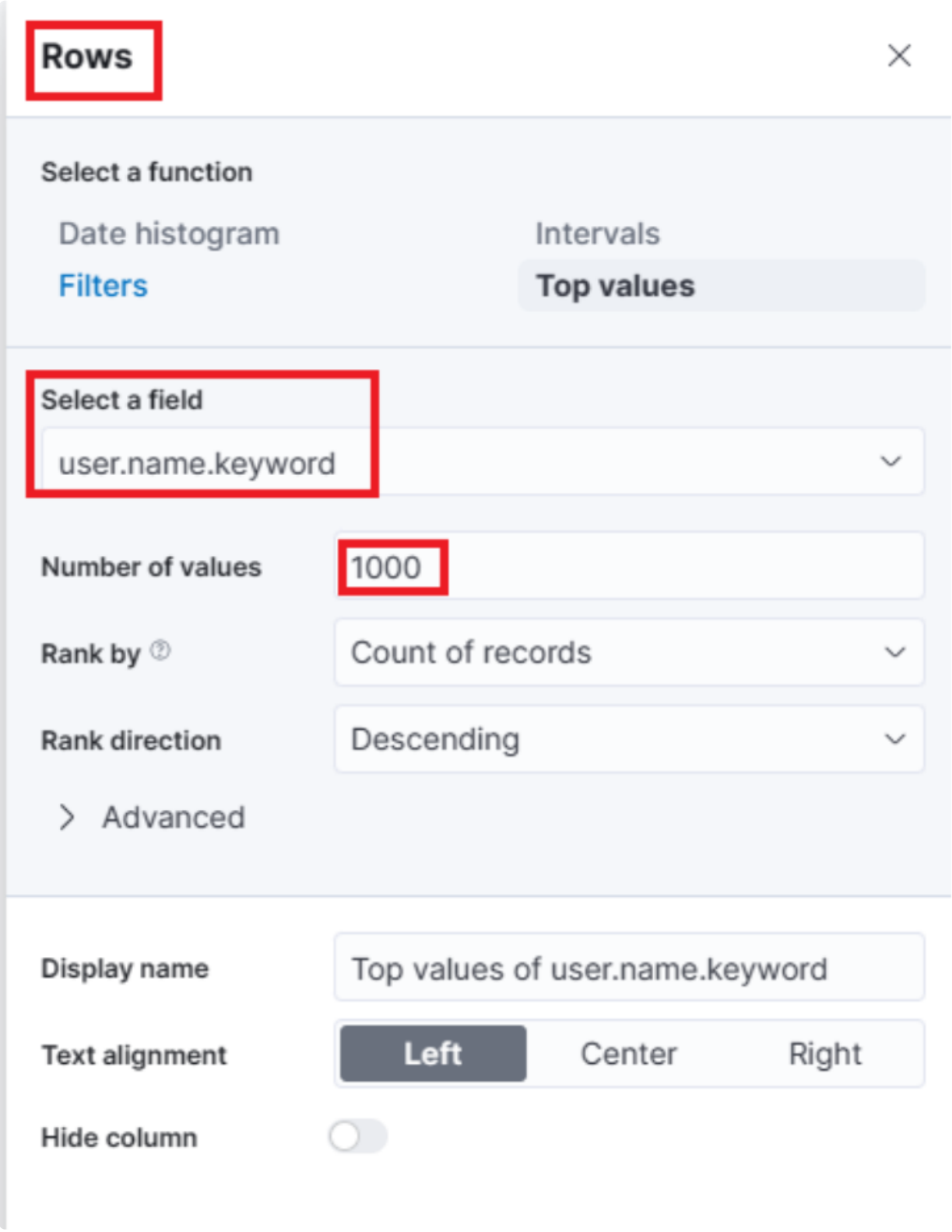


Table
windows*
Rows
Add or drag-and-drop a field
Columns
Add or drag-and-drop a field
Metrics
Add or drag-and-drop a field

- Let's configure the "Rows" settings as follows.




- Moving forward, let's close the "Rows" window and proceed to enter the "Metrics" configuration.

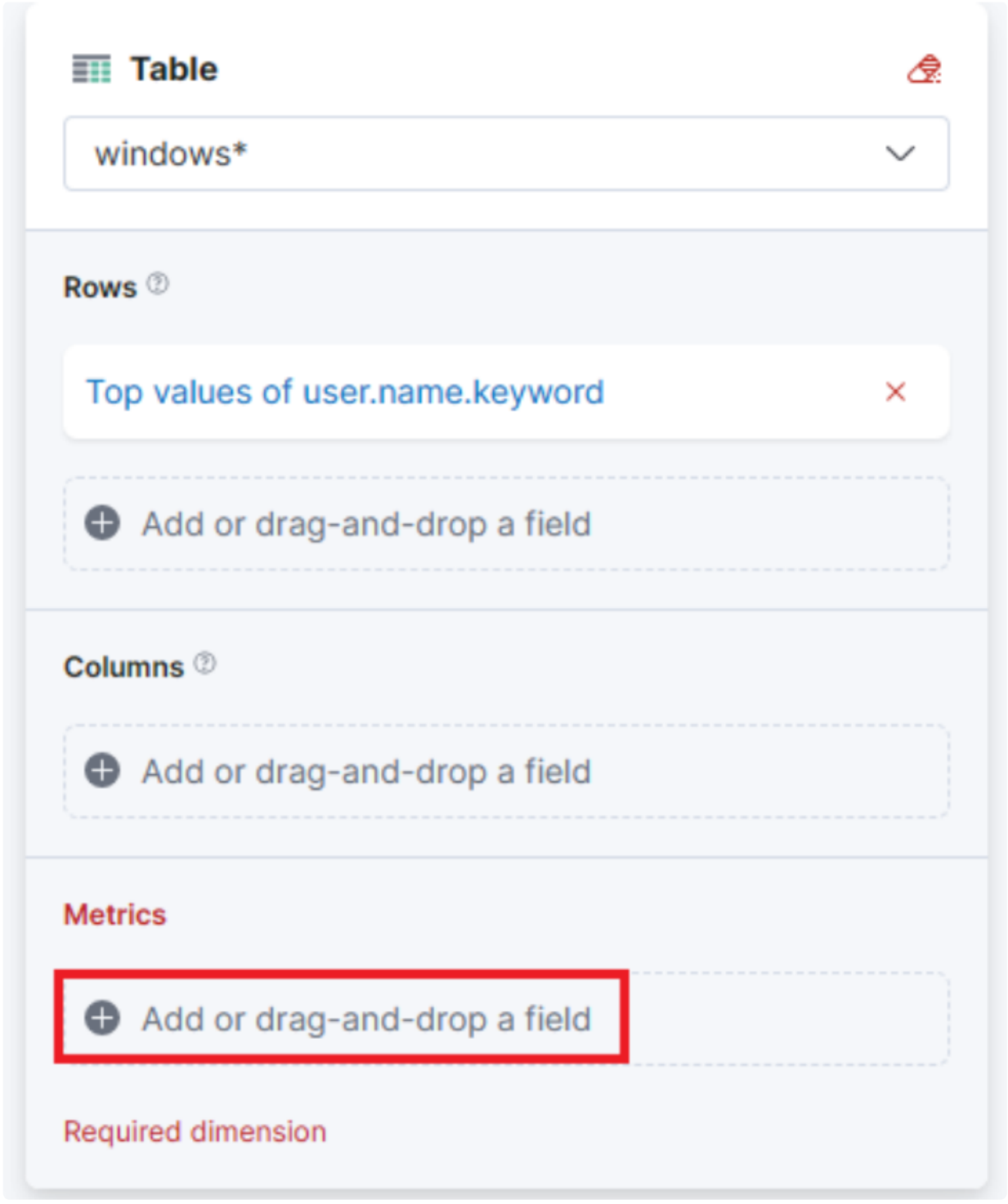
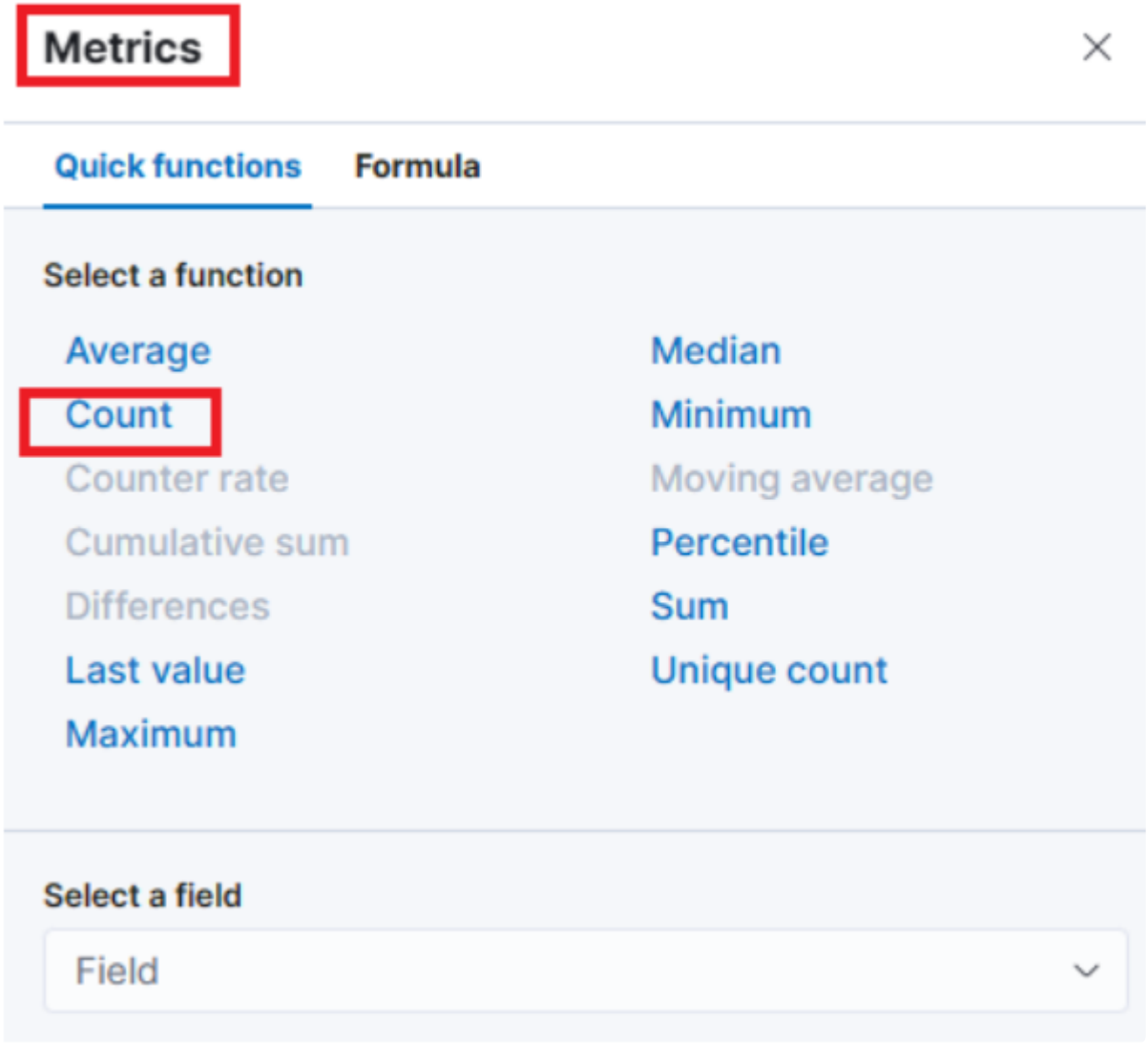


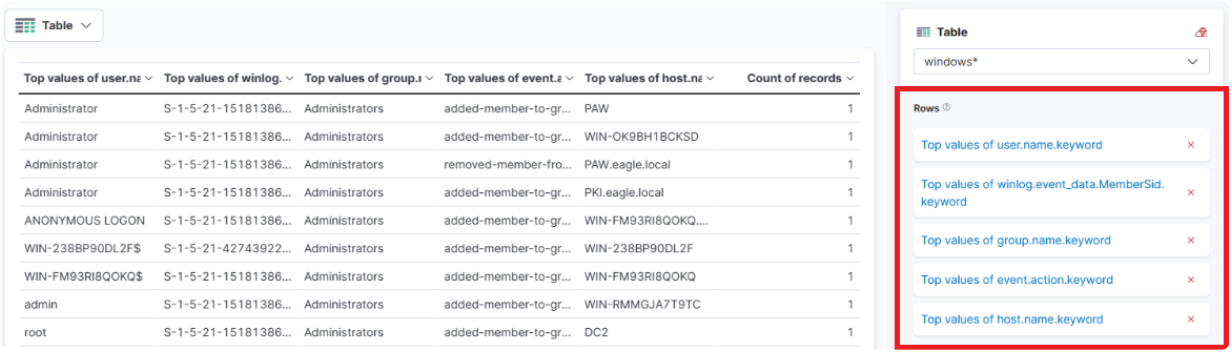
Table
windows*
Rows
Top values of user.name.keyword
Add or drag-and-drop a field
Columns
Add or drag-and-drop a field
Metrics
Add or drag-and-drop a field
Required dimension

- In the "Metrics" window, let's select "count" as the desired metric.

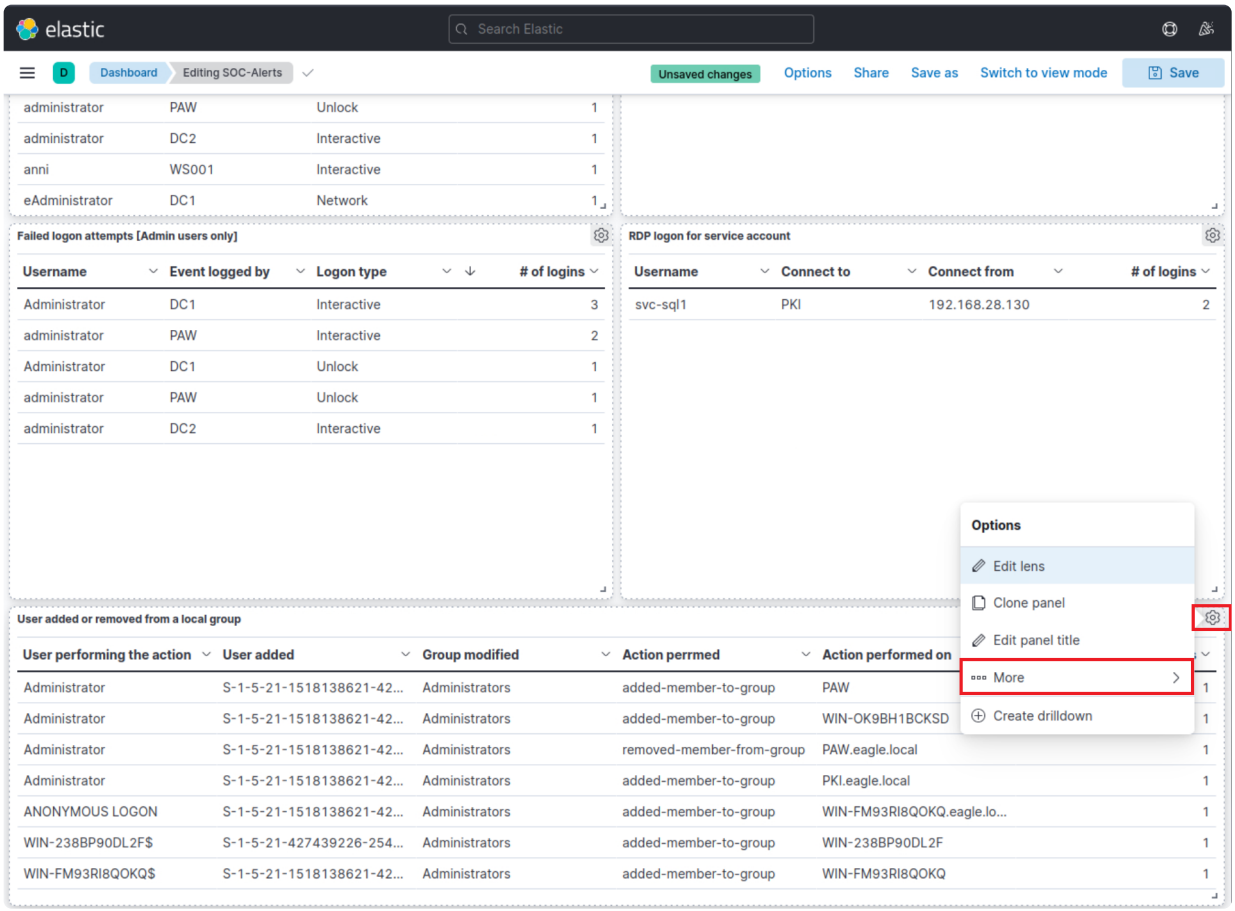


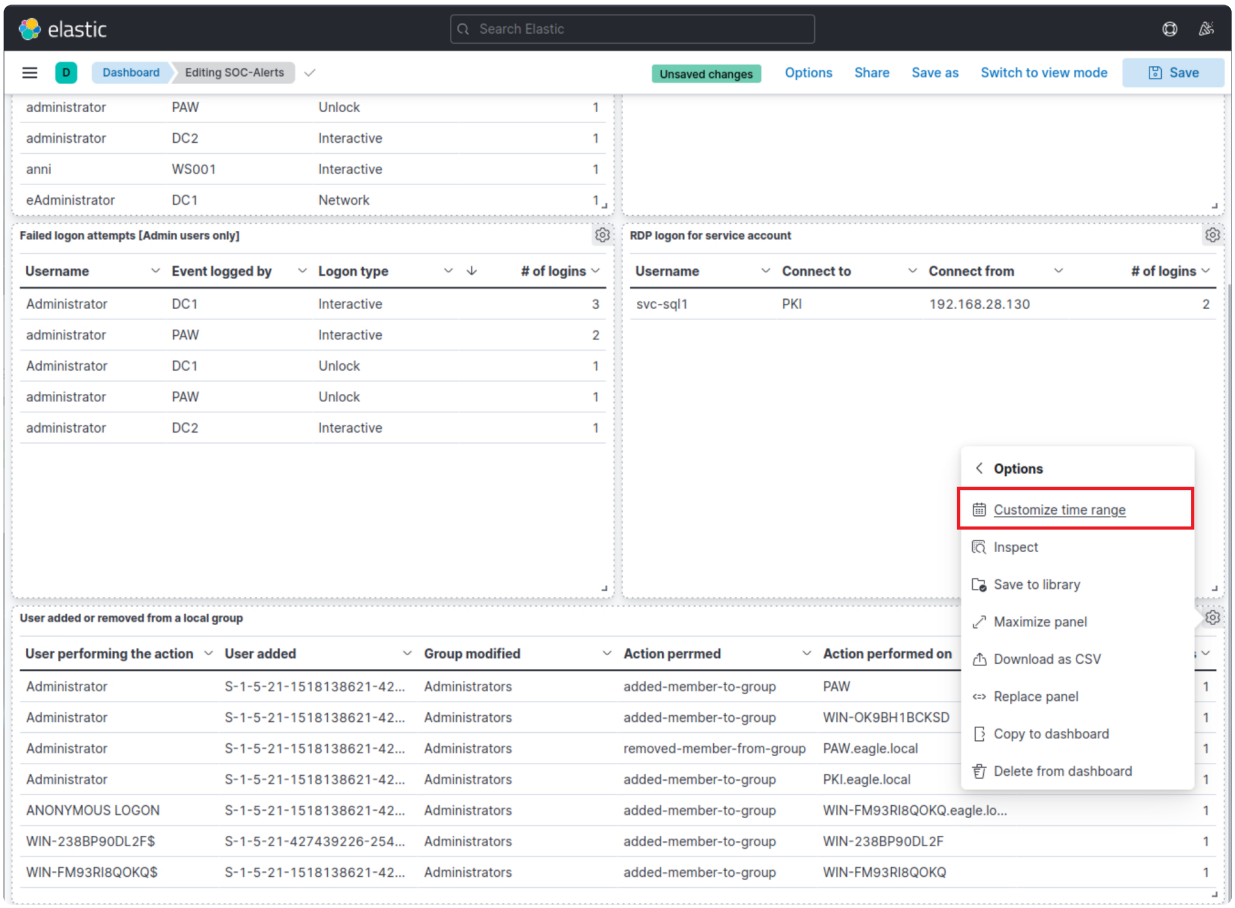

- One final addition to the table is to include some more "Rows" settings to enhance our understanding.

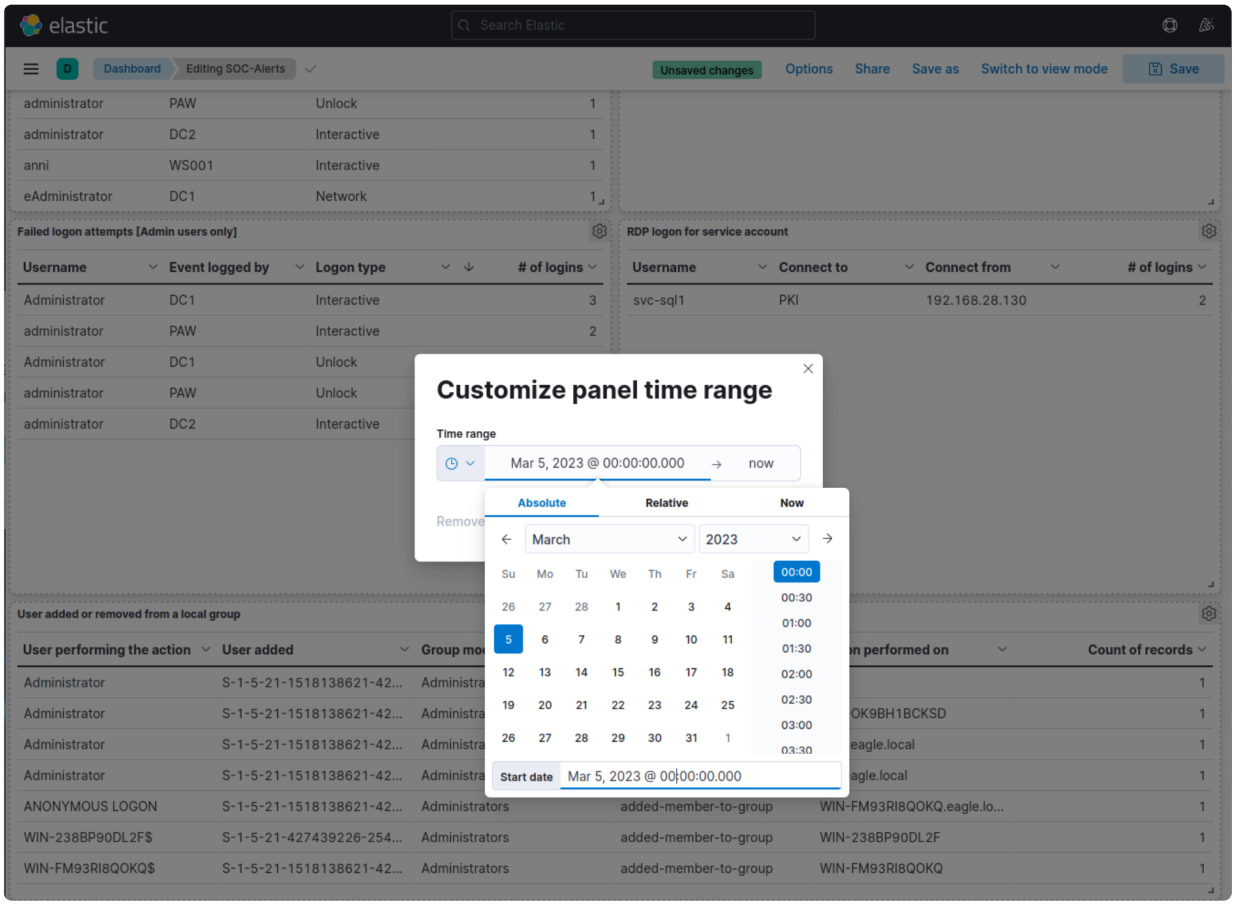
```
- Which user was added to or removed from the group?
(`winlog.event_data.MemberSid.keyword` field)

- To which group was the addition or the removal performed? (double-
checking that it is the "Administrators" one) (`group.name.keyword`
field)

- Was the user added to or removed from the group?
(`event.action.keyword` field)

- On which machine did the action occur? (`host.name.keyword` field)
```




- Click on "Save and return", and you will observe that the new visualization is added to the dashboard.
- As discussed, we want to monitor user additions or removals from the local "Administrators" group *within a specific timeframe (March 5th 2023 to date)*.
- We can narrow the scope of our visualization as follows.






## Alert Triaging

**The Triaging Process**

What Is Alert Triaging?

- `Alert triaging`, performed by a Security Operations Center (SOC) analyst, is the process of evaluating and prioritizing security alerts generated by various monitoring and detection systems to determine their level of threat and potential impact on an organization's systems and data.
  - It involves systematically reviewing and categorizing alerts to effectively allocate resources and respond to security incidents.
- `Escalation` is an important aspect of alert triaging in a SOC environment.
  - The escalation process typically involves notifying supervisors, incident response teams, or designated individuals within the organization who have the authority to make decisions and coordinate the response effort.
  - The SOC analyst provides detailed information about the alert, including its severity, potential impact, and any relevant findings from the initial investigation.
  - This allows the decision-makers to assess the situation and determine the appropriate course of action, such as involving specialized teams, initiating broader incident response procedures, or engaging external resources if necessary.
- Escalation ensures that critical alerts receive prompt attention and facilitates effective coordination among different stakeholders, enabling a timely and efficient response to potential security incidents.
  - It helps to leverage the expertise and decision-making capabilities of individuals who are responsible for managing and mitigating higher-level threats or incidents within the organization.

What Is The Ideal Triaging Process?

1. `Initial Alert Review`:
   - Thoroughly review the initial alert, including metadata, timestamp, source IP, destination IP, affected systems, and triggering rule/signature.
   - Analyze associated logs (network traffic, system, application) to understand the alert's context.
2. `Alert Classification`:
   - Classify the alert based on severity, impact, and urgency using the organization's predefined classification system.
3. `Alert Correlation`:

- Cross-reference the alert with related alerts, events, or incidents to identify patterns, similarities, or potential indicators of compromise (IOCs).
- Query the SIEM or log management system to gather relevant log data.
- Leverage threat intelligence feeds to check for known attack patterns or malware signatures.

4. `Enrichment of Alert Data`:
   - Gather additional information to enrich the alert data and gain context:
     - Collect network packet captures, memory dumps, or file samples associated with the alert.
     - Utilize external threat intelligence sources, open-source tools, or sandboxes to analyze suspicious files, URLs, or IP addresses.
     - Conduct reconnaissance of affected systems for anomalies (network connections, processes, file modifications).
5. `Risk Assessment`:
   - Evaluate the potential risk and impact to critical assets, data, or infrastructure:
     - Consider the value of affected systems, sensitivity of data, compliance requirements, and regulatory implications.
     - Determine likelihood of a successful attack or potential lateral movement.
6. `Contextual Analysis`:
   - The analyst considers the context surrounding the alert, including the affected assets, their criticality, and the sensitivity of the data they handle.
   - They evaluate the security controls in place, such as firewalls, intrusion detection/prevention systems, and endpoint protection solutions, to determine if the alert indicates a potential control failure or evasion technique.
   - The analyst assesses the relevant compliance requirements, industry regulations, and contractual obligations to understand the implications of the alert on the organization's legal and regulatory compliance posture.
7. `Incident Response Planning`:
   - Initiate an incident response plan if the alert is significant:
     - Document alert details, affected systems, observed behaviors, potential IOCs, and enrichment data.
     - Assign incident response team members with defined roles and responsibilities.
     - Coordinate with other teams (network operations, system administrators, vendors) as necessary.
8. `Consultation with IT Operations`:
   - Assess the need for additional context or missing information by consulting with IT operations or relevant departments:

- Engage in discussions or meetings to gather insights on the affected systems, recent changes, or ongoing maintenance activities.
- Collaborate to understand any known issues, misconfigurations, or network changes that could potentially generate false-positive alerts.
- Gain a holistic understanding of the environment and any non-malicious activities that might have triggered the alert.
- Document the insights and information obtained during the consultation.

9. `Response Execution`:
   - Based on the alert review, risk assessment, and consultation, determine the appropriate response actions.
   - If the additional context resolves the alert or identifies it as a non-malicious event, take necessary actions without escalation.
   - If the alert still indicates potential security concerns or requires further investigation, proceed with the incident response actions.
10. `Escalation`:
    - Identify triggers for escalation based on organization's policies and alert severity:
      - Triggers may include compromise of critical systems/assets, ongoing attacks, unfamiliar/sophisticated techniques, widespread impact, or insider threats.
    - Assess the alert against escalation triggers, considering potential consequences if not escalated.
    - Follow internal escalation process, notifying higher-level teams/management responsible for incident response.
    - Provide comprehensive alert summary, severity, potential impact, enrichment data, and risk assessment.
    - Document all communication related to escalation.
    - In some cases, escalate to external entities (law enforcement, incident response providers, CERTs) based on legal/regulatory requirements.
11. `Continuous Monitoring`:
    - Continuously monitor the situation and incident response progress.
    - Maintain open communication with escalated teams, providing updates on developments, findings, or changes in severity/impact.
    - Collaborate closely with escalated teams for a coordinated response.
12. `De-escalation`:
    - Evaluate the need for de-escalation as the incident response progresses and the situation is under control.
    - De-escalate when the risk is mitigated, incident is contained, and further escalation is unnecessary.

    - Notify relevant parties, providing a summary of actions taken, outcomes, and lessons learned

- Regularly review and update the process, aligning it with organizational policies, procedures, and guidelines.
    - Adapt the process to address emerging threats and evolving needs.