SIEM Solution

Musawer Hamad Khan
BSCS-2021-40

Contents

SIEM

Security and Event Management is a security and auditing tool comprised of different monitoring and analysis tools. Due to the huge rise in cyber-attacks and new techniques that are being used in these attacks, it has become crucial for the organization to adopt SEAM solution.

Why do we need SIEM?

According Symantec's 2018 Internet Security Threat report points out, it's not just the amount of attacks that is on the rise but also the avenues and methodologies used:

"From the sudden spread of WannaCry and Petya/NotPetya, to the swift growth in coinminers, 2017 provided us with another reminder that digital security threats can come from new and unexpected sources. With each passing year, not only has the sheer volume of threats increased, but the threat landscape has become more diverse, with attackers working harder to discover new avenues of attack and cover their tracks while doing so."

Traditionally, attacks and threats were blocked before they entered a network. The tools that were used in preventing these attacks would rely on predefined rules and signatures of known threats. But they had to struggle with the attacks and threats that haven't been identified yet or whose signature is unknown such as zero-day threats.

However, SIEM focuses on analyzing data from different sources across the network to identify suspicious activity and potential threat. It helps organizations to detect, analyze and respond to security threats before they harm business operation.

As implied above, SIEM combines several security disciplines and tools under one comprehensive umbrella: ("What is SIEM? (Security Information and Event Management) - Logz.io")

- **"Log management (LMS)** – tools used for traditional log collection and storage." ("What is SIEM? (Security Information and Event Management) - Logz.io")

- **Security Information Management (SIM)** – tools or systems focusing on collecting and managing security-related data from multiple data sources. These data sources could be, for example, firewalls, DNS servers, routers, antivirus apps.

- **Security Event Management (SEM**) – systems that are based on proactive monitoring and analysis, including data visualization, event correlation and alerting.

SIEM is combines all above process into single layer which knowns how to collect and analyze information from different sources and store it at one centralize location and corelate between different event to produce alerts and reports based on this information.

Components of SIEM

As mentioned above SIEM is not a single rather it is set of different tools and techniques and there are no predefined principles and methodologies for SEAM, but it will have some of the component described below.

- Aggregation
- Processing and Normalization
- Correlation
- Presentation
- Mitigation and Remediation

Aggregation

In SIEM solution data is gathered from different sources like firewall logs, database logs, host logs. A SIEM solution can collect this data and store it at one centralized location. This collection is performed by the agents or applications deployed on the system that is being monitored.

Processing and Normalization

Once data is collected and stored in one centralized location then data must be normalized to perform useful analysis. As we know data has been collected from different sources like firewall, servers, host etc. So, data logs that are generated by these sources have different formats.

For efficient data interpretation and event correlations SIEM systems can normalize the logs. During the normalization process, the logs are processed into an organized and understandable format, key information is extracted, and the various fields they include are mapped.

Correlation

When normalization is done then data patterns are traced within the normalized data to correlate different events that could provide information about a security threat. Moreover, correlation can provide help to focus on the data that is more useful and meaningful. "This correlation work is based on rules that are either provided by various SIEM tools, predefined for different attack scenarios, or created and fine-tuned by the analyst." ("What is SIEM? (Security Information and Event Management) - Logz.io")

Presentation

For better understanding of data, it is necessary to visualize data. Dashboard containing different visualization can help analyst to identify different trends, anomalies to maintain secure environment.

Mitigation and Remediation

Once everything is done the last step is how the SIEM system will identify and handle the incidents. Most of the SEAM solutions automatically identify and handle based on the correlation rules.

SIEM Solution

OSSEC

Open-Source HID Security is an open-source host-based intrusion detection system with capabilities of log analysis, integrity checking, window registry monitoring, rootkit detection, time-based alerting, and active response. It works with different platforms like windows, macOS, Linux, Solaris etc. OSSEC has centralized and cross-platform architecture which means it stores logs data against various agents, analyzes it, makes policies on one centralized server and OSSEC agent can be installed on different operating systems.

OSSEC lets you configure the alert based on your choice. Integration with smtp, SMS and syslog allows you to be on top of alerts by sending them to e-mail enabled devices. Active response options to block an attack immediately are also available. ("Getting started with OSSEC — OSSEC")

OSSEC Architecture

OSSEC architecture is as follow:

- **Manager**
  The manager is the main component of the deployment of OSSEC. It holds the logs, events, and system auditing entries in addition to the file integrity checking databases. The manager houses all of the rules, decoders, and important configuration choices in one central location, making it simple to handle even a big number of agents.
  The server is reached via agents via port 1514/udp. Agents cannot communicate with the server unless communication to this port is permitted.
- **Agents**
  Agent is a small program, or set of programs, that is installed on the systems being monitored. The agent will gather information and provide it to the manager for analysis and correlation. Some data is acquired in real time, while others are collected at regular intervals. It has a very minimal memory and CPU footprint by default, which does not affect the system's utilization.

- **Agentless**
  In OSSEC, agentless mode is a functionality that allows you to monitor systems for security threats without installing a dedicated OSSEC agent on them. Agentless mode can be used to monitor firewalls, routers, and even Unix systems. (“OSSEC - Wikipedia”)

OSSEC Features

- **Log based Intrusion Detection (LID):** Actively monitors and analyzes data from multiple log data points in real-time. (“OSSEC - Wikipedia”)
- **Rootkit and Malware Detection:** Process and file level analysis to detect malicious applications and rootkits.
- **Active Response:** Respond to attacks and changes on the system in real time through multiple mechanisms including firewall policies, integration with 3rd parties such as CDN’s and support portals, as well as self-healing actions.
- **Compliance Auditing:** Application and system level auditing for compliance with many common standards such as PCI-DSS, and CIS benchmarks.
- **File Integrity Monitoring(FIM):** For both files and windows registry settings in real time not only detects changes to the system, but it also maintains a forensic copy of the data as it changes over time. (“OSSEC - Open Source HIDS - FIM, Rootkit Detection, Malware Detection”)

OSSEC Drawbacks

OSSEC has primitive log storage engine which mean It does not support advance querying. By default it does not retain host log messages. Once analyzed messages are deleted however if the <logall> option is included in the OSSEC manager’s ossec.conf file. "If this option is enabled, OSSEC stores the incoming logs from agents in a text file that is rotated daily." (“10 Leading Open Source SIEM Tools - 2023 Update | Logz.io”)

OSSEC does not support visualization. However, we can use third party tools.

Gray log

Graylog's effectiveness in delivering an LMS stem from the fact that it was designed from the ground up for log management. To store and analyse log data efficiently, software must adhere to a specified design. It is more than just a database or a full-text search engine because it handles both text and metrics data on a temporal scale. Searches are always limited to a time window (relative or absolute), and they can only travel back in time because future log data has yet to be written. A general purpose database or full text search engine that can also store and index your online platform's private messages for search purposes will never be able to efficiently manage your log data.A general-purpose database or full-text search engine that can also store and index your online platform's private communications for search will never be able to manage your log data effectively.

Streams

Streams operate as a form of tagging for incoming messages. Streams route messages into categories in real time, and team rules instruct Graylog to route messages into the appropriate stream. Streams are used to route data for storage into an index. They are also used to control access to data, and route messages for parsing, enrichment, and other modification. Streams then determine which messages to archive. (“What is Graylog”)

Graylog Search Page

The Graylog Search page is the interface used to search logs directly. Graylog uses a simplified syntax, very similar to Lucene. Relative or absolute time ranges are configurable from drop down menus. Searches may be saved or visualized as dashboard widgets that may be added directly to dashboards from within the search screen. (“Explore Graylog — Graylog 3.2.0 documentation”) "Users may configure their own views and may choose to see either a summary or complete data from event messages." (“What is Graylog”)

Graylog Dashboards

Graylog Dashboards are visualizations or summaries of information contained in log events. Each dashboard is populated by one or more widgets. Widgets visualize or summarize event log data with data derived from field values such as counts, averages, or totals. Users can create indicators, charts, graphs, and maps to visualize the data. Dashboard widgets and dashboard layouts are configurable. Graylog's role-based access controls dashboard access. Users can import and export dashboards via content packs.

Alerts

Alerts are created using Event Definitions that consist of Conditions. When a given condition is met it will be stored as an Event and can be used to trigger a notification.

Content Packs

Content packs accelerate the set-up process for a specific data source. A content pack can include inputs/extractors, streams, dashboards, alerts, and pipeline processors. For example, users can create custom inputs, streams, dashboards, and alerts to support a security use case. Users can then export the content pack and import it on a newly installed Graylog instance to save configuration time and effort. Users may download content packs which are created, shared and supported by other users via the Graylog Marketplace. (“What is Graylog”)

Index Sets

An Index is the basic unit of storage for data in OpenSearch and Elasticsearch. Index sets provide configuration for retention, sharding, and replication of the stored data. Values, like retention and rotation strategy, are set on a per-index basis, so different data may be subjected to different handling rules.

Graylog Sidecar

Graylog Sidecar is an agent to manage fleets of log shippers, like Beats or NXLog. These log shippers are used to collect OS logs from Linux and Windows servers. Log shippers read logs written locally to a flat file, and then send them to a centralized log management solution. Graylog supports management of any log shipper as a backend.

Processing Pipelines

Graylog’s Processing Pipelines enable the user to run a rule, or a series of rules, against a specific type of event. Tied to streams, pipelines allow routing, denylisting, modification, and enrichment of messages as they flow through Graylog.

Onion Security

Security Onion is a free and open platform built by defenders for defenders. It includes network visibility, host visibility, intrusion detection honeypots, log management, and case management. Security Onion has been downloaded over 2 million times and is being used by security teams around the world to monitor and defend their enterprises.

Security Onion Features

Security Onion is a free and open-source Linux distribution pre-configured with a powerful suite of security monitoring tools. Here's a breakdown of its key features:

- **Log Management:** Security Onion includes tools like Logstash and Elasticsearch for centralizing log collection, parsing, and storage from various network devices and security applications.
- **Network Security Monitoring (NSM):** Security Onion integrates tools like Snort and Suricata for real-time network traffic analysis, intrusion detection, and identification of potential threats.
- **Security Information and Event Management (SIEM):** Security Onion leverages tools like Wazuh to collect, analyze, and correlate security events from various sources, providing centralized management and threat detection capabilities.
- **Packet Capture and Analysis:** Tools like Wireshark are included for capturing and analyzing network traffic in detail, aiding in forensic investigations and troubleshooting network issues.
- **Vulnerability Scanning:** Security Onion might integrate vulnerability scanners like OpenVAS to identify potential vulnerabilities in your systems and applications.
- **Security Threat Intelligence (STIX/TAXII):** Security Onion can connect to threat intelligence feeds to gain insights into emerging threats and improve its detection capabilities.
- **Centralized Management:** Security Onion provides a unified web interface for managing and monitoring all its integrated security tools, simplifying security operations.
- **Open-source and Free:** Being free and open-source, Security Onion offers a cost-effective and customizable solution for security monitoring.
- **Community Supported:** A large and active community supports Security Onion, providing resources, documentation, and ongoing development.

Part-2

Implementing Wazuh

Software and Hardware Requirements

Following requirements were given in the official documentation of Wazuh.

Agents	CPU	RAM	Storage (90 days)
1–25	4 vCPU	8 GiB	50 GB
25–50	8 vCPU	8 GiB	100 GB
50–100	8 vCPU	8 GiB	200 GB

Hardware req 1

Wazuh central components can be installed on a 64-bit Linux operating system. Wazuh recommends any of the following operating system versions:

Amazon Linux 2	CentOS 7, 8
Red Hat Enterprise Linux 7, 8, 9	Ubuntu 16.04, 18.04, 20.04, 22.04

Wazuh dashboard supports the following web browsers:

- Chrome 95 or later
- Firefox 93 or later
- Safari 13.7 or later

Other Chromium-based browsers might also work. Internet Explorer 11 is not supported.

Wazuh

Wazuh is an open-source security platform, or we can say all in one SIEM solution. It provides unified XDR and SIEM protection for endpoints and cloud workload. This SIEM solution consists of four major components.

1. Wazuh Agent
2. Wazuh Server
3. Wazuh Indexer
4. Wazuh Dashboard

Wazuh Server

Wazuh Server receives data from agent and perform analysis and generate alerts in case threat or anomalies are detected. It can also be used to manage and configure the agents remotely.

Wazuh server uses threat intelligence along with MITRE ATTACK framework to enhance its detection capabilities. Additionally, it can integrate with instant message platforms like Slack for notifying the analyst and to take quick actions.

Server components

The Wazuh server comprises several components listed below that have different functions, such as enrolling new agents, validating each agent identity, and encrypting the communications between the Wazuh agent and the Wazuh server.

- **Agent enrollment service:** It is used to enroll new agents. This service provides and distributes unique authentication keys to each agent. The process runs as a network service and supports authentication via TLS/SSL certificates or by providing a fixed password.
- **Agent connection service:** This service receives data from the agents. It uses the keys shared by the enrollment service to validate each agent identity and encrypt the communications between the Wazuh agent and the Wazuh server. Additionally, this service provides centralized configuration management, enabling you to push new agent settings remotely.
- **Analysis engine:** This is the server component that performs the data analysis. It uses decoders to identify the type of information being processed (Windows events, SSH logs, web server logs, and others). These decoders also extract relevant data elements from the log messages, such as source IP address, event ID, or username. Then, by using rules, the engine identifies specific patterns in the decoded events that could trigger alerts and possibly even call for automated countermeasures (e.g., banning an IP address, stopping a running process, or removing a malware artifact).
- **Wazuh RESTful API:** This service provides an interface to interact with the Wazuh infrastructure. It is used to manage configuration settings of agents and servers, monitor the infrastructure status and overall health, manage and edit Wazuh decoders and rules, and query about the state of the monitored endpoints. The Wazuh dashboard also uses it.
- **Wazuh cluster daemon:** This service is used to scale Wazuh servers horizontally, deploying them as a cluster. This kind of configuration, combined with a network load balancer, provides high availability and load balancing. The Wazuh cluster daemon is what Wazuh servers use to communicate with each other and to keep synchronized.
- **Filebeat:** It is used to send events and alerts to the Wazuh indexer. It reads the output of the Wazuh analysis engine and ships events in real time. It also provides load balancing when connected to a multi-node Wazuh indexer cluster.

Agent

The Wazuh agent runs on Linux, Windows, macOS, Solaris, AIX, and other operating systems. It can be deployed to laptops, desktops, servers, cloud instances, containers, or virtual machines. The agent helps to protect your system by providing threat prevention, detection, and response capabilities. It is also used to collect different types of system and application data that it forwards to the Wazuh server through an encrypted and authenticated channel.

Agent modules

All agent modules are configurable and perform different security tasks. This modular architecture allows you to enable or disable each component according to your security needs. Below you can learn about the different purposes of all the agent modules.

- **Log collector:** This agent component can read flat log files and Windows events, collecting operating system and application log messages. It supports XPath filters for Windows events and recognizes multi-line formats like Linux Audit logs. It can also enrich JSON events with additional metadata.
- **Command execution:** Agents run authorized commands periodically, collecting their output and reporting it back to the Wazuh server for further analysis. You can use this module for different purposes, such as monitoring hard disk space left or getting a list of the last logged-in users.
- **File integrity monitoring (FIM):** This module monitors the file system, reporting when files are created, deleted, or modified. It keeps track of changes in file attributes, permissions, ownership, and content. When an event occurs, it captures who, what, and when details in real time. Additionally, the FIM module builds and maintains a database with the state of the monitored files, allowing queries to be run remotely.
- **Security configuration assessment (SCA):** This component provides continuous configuration assessment, utilizing out-of-the-box checks based on the Center of Internet Security (CIS) benchmarks. Users can also create their own SCA checks to monitor and enforce their security policies.
- **System inventory:** This agent module periodically runs scans, collecting inventory data such as operating system version, network interfaces, running processes, installed applications, and a list of open ports. Scan results are stored in local SQLite databases that can be queried remotely.
- **Malware detection:** Using a non-signature-based approach, this component is capable of detecting anomalies and the possible presence of rootkits. Also, it looks for hidden processes, hidden files, and hidden ports while monitoring system calls.
- **Active response:** This module runs automatic actions when threats are detected, triggering responses to block a network connection, stop a running process, or delete a malicious file. Users can also create custom responses when necessary and customize, for example, responses for running a binary in a sandbox, capturing network traffic, and scanning a file with an antivirus.
- **Container security monitoring:** This agent module is integrated with the Docker Engine API to monitor changes in a containerized environment. For example, it detects changes to container images, network configuration, or data volumes. Besides, it alerts about

containers running in privileged mode and about users executing commands in a running container.
- **Cloud security monitoring:** This component monitors cloud providers such as Amazon AWS, Microsoft Azure, or Google GCP. It natively communicates with their APIs. It is capable of detecting changes to the cloud infrastructure (e.g., a new user is created, a security group is modified, a cloud instance is stopped, etc.) and collecting cloud services log data (e.g., AWS Cloudtrail, AWS Macie, AWS GuardDuty, Microsoft Entra ID, etc.)

Wazuh Indexer

It is an advanced full text-search and analytic engine. We can also say that it is kind of log manager which receive the data from the serve, index the data and store it. Additionally, it records and retrieve data almost close real time as the latency from the time a document is indexed until it becomes searchable is very short, typically one second.

Wazuh indexer stores data as JSON documents. An index is a collection of documents that are related to each other. The documents stored in the Wazuh indexer are distributed across different containers known as shards. By distributing the documents across multiple shards, and distributing those shards across multiple nodes, the Wazuh indexer can ensure redundancy. This protects your system against hardware failures and increases query capacity as nodes are added to a cluster.

Wazuh store data in following four indices:

- **Wazuh-alerts:**
  Stores alerts generated by wazuh server.
- **Wazuh-archives:**
  Stores all events (archive data) received by the Wazuh server, whether or not they trip a rule.
- **Wazuh-monitoring:**
  Stores data related to the Wazuh agent status over time. It is used by the web interface to represent when individual agents are or have been Active, Disconnected,
  or Never connected.
- **Wazuh-Statistics:**
  Stores data related to the Wazuh server performance. It is used by the web interface to represent the performance statistics.

Moreover, it can be configured as single-node or multi-node cluster which provides scalability, high availability, and redundancy.

Log Management

Wazuh uses Logcollector module to record all the log from monitored endpoints. The wazuh serever analyze these logs with the help of decoder and rules in real-time. When logs are analyzed then analysis module generates alert which are stored in a file named alerts.logs and alert.json.

Similarly, wazuh also stores all the logs in dedicated archive log file. This file captures all kinds of logs whether they have generated alert or they haven't generated alerts.

There are devices such as fire wall, switches, router and other devices which don't support wazuh agents so wazuh server receive syslog messages ensuring seamless integration and coverage across your entire network.

Event channel is a log format that supports Windows Vista and recent versions. It captures Applications and Services logs, as well as basic Windows logs, including Application, Security, and System logs. This log format also supports the use of queries to monitor specific Windows events. By default, the Wazuh agent monitors the System, Application, and Security Windows event channels. You can configure the Wazuh agent to monitor other Windows event channels of interest.

The table below shows the channels and providers that the Wazuh agent supports.

Source	Channel name	Provider name	Description
Application	Application	Any	This channel collects events related to system application management and is one of the main Windows administrative channels along with Security, and System.
Security	Security	Any	This channel gathers information related to user and group creation, login, logoff, and audit policy modifications.
System	System	Any	The System channel collects events associated with

Source	Channel name	Provider name	Description
			kernel and service control.
Sysmon	Microsoft-Windows-Sysmon/Operational	Microsoft-Windows-Sysmon	"Sysmon monitors system activity such as process creation and termination, network connections, and file changes." (“Configuring log collection for different operating systems”)
Windows Defender	Microsoft-Windows-Windows Defender/Operational	Microsoft-Windows-Windows Defender	The Windows Defender log shows information about the scans passed, malware detection, and actions taken against them.
McAfee	Application	McLogEvent	This source shows McAfee scan results, virus detection, and actions taken against them.
EventLog	System	Eventlog	This source retrieves information about audit and Windows logs.

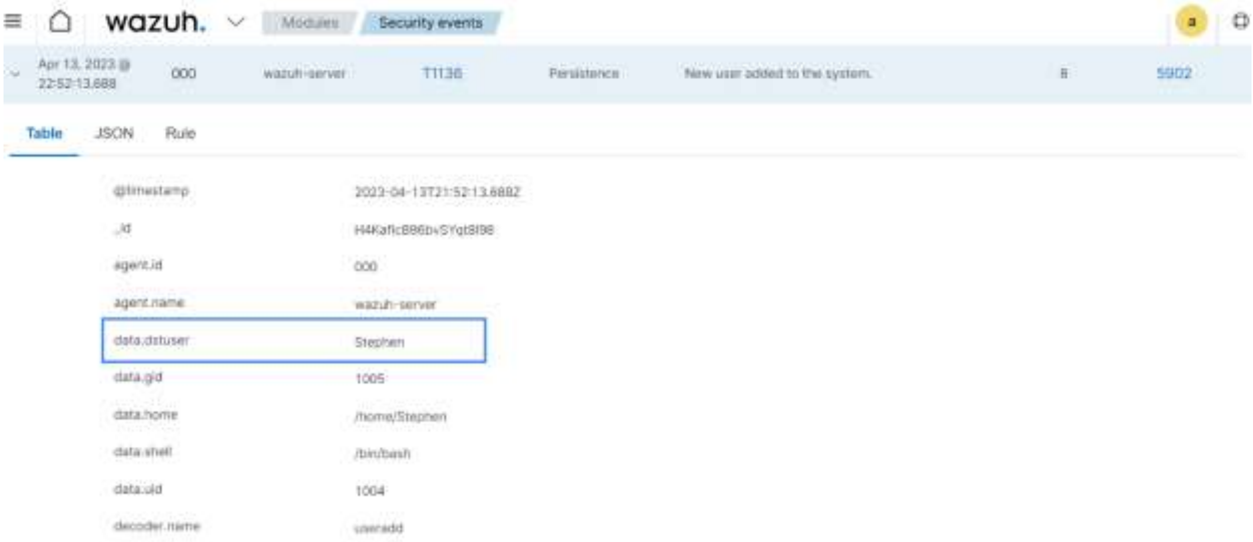
Source	Channel name	Provider name	Description
Microsoft Security Essentials	System	Microsoft Antimalware	This source gives information about real-time protection for the system, malware detection scans, and changes in antivirus settings.
Remote Access	File Replication Service	Any	Other channels (they are grouped in a generic Windows rule file).
Terminal Services	Microsoft-Windows-TerminalServices-RemoteConnectionManager		
Powershell	Microsoft-Windows-PowerShell/Operational	Microsoft-Windows-PowerShell	This channel collects and audits PowerShell activity.

Forwarding Linux logs using rsyslog in Wazuh involves configuring the rsyslog service on your Linux machines to send logs to the Wazuh server for centralized collection and analysis.

Forwarding Linux logs using rsyslog

In this use case, we configure a CentOS 7 endpoint to forward logs using rsyslog to the Wazuh server for analysis. On the CentOS 7 endpoint, we create and delete the user account Stephen. Wazuh has default rules that generate alerts for the creation and deletion of user accounts.

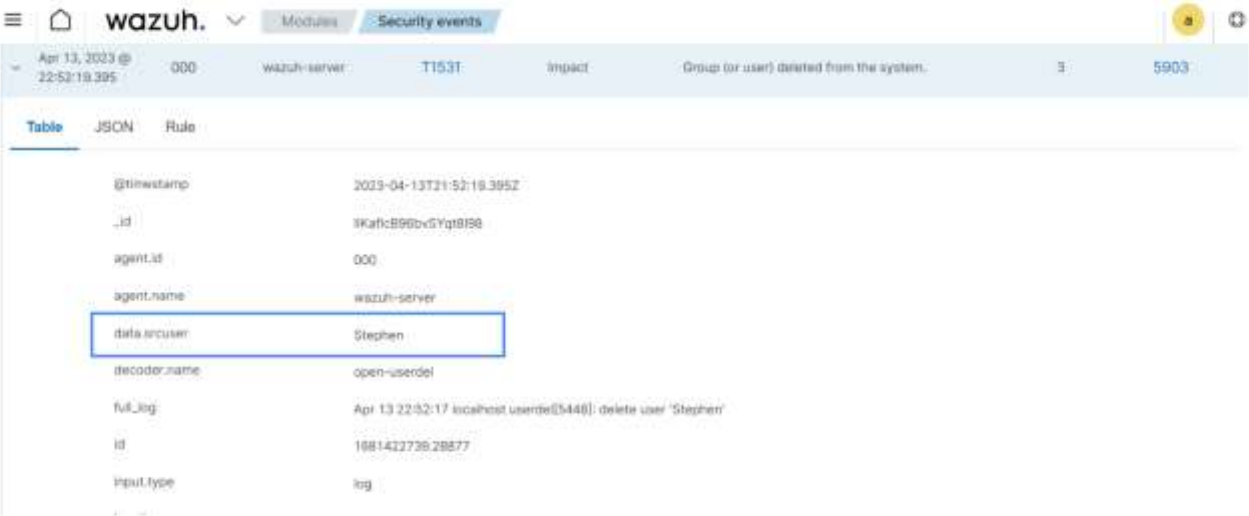
When rsyslog is configured in linux then we get alerts on creation and deletion of account.



Apr 13, 2023 @ 22:52:13.688	000	wazuh-server	T1136	Persistence	New user added to the system.	8	5902

@timestamp	2023-04-13T21:52:13.688Z
_id	H4KaflcB86bvSYqt8I98
agent.id	000
agent.name	wazuh-server
data.dstuser	Stephen
data.gid	1005
data.home	/home/Stephen
data.shell	/bin/bash
data.uid	1004
decoder.name	useradd

Account Creation Alert 1



Apr 13, 2023 @ 22:52:19.395	000	wazuh-server	T1531	Impact	Group (or user) deleted from the system.	3	5903

@timestamp	2023-04-13T21:52:19.395Z
_id	IIKaflcB96bvSYqt8I98
agent.id	000
agent.name	wazuh-server
data.srcuser	Stephen
decoder.name	open-userdel
full_log	Apr 13 22:52:17 localhost userdel[5448]: delete user 'Stephen'
id	1681422739.28877
input.type	log

Account Deletion Alert 1

Threat Detection and Rules

Traditionally threat was detected on the basis of signature but signature-based detection had a lot of limitations. Signature based detection struggled with unknown threat like zero-day attack, polymorphic malware and other threats. As a result, organizations are at risk of undetected breaches and data exfiltration. Wazuh empowers organizations to detect and respond to sophisticated and evasive threats effectively. Wazuh encompasses different modules that identify malware properties, activities, network connections, and more.

Wazuh's threat detection rules enable **behavior-based malware detection.** Rather than relying exclusively on predetermined signatures, Wazuh monitors and analyses malware's anomalous behaviour. This enables Wazuh to detect both known and previously unknown threats. Wazuh offers a proactive and adaptable defence against cyber threats. Wazuh includes out-of-the-box rulesets that are specifically designed to generate alerts for recognised malware patterns, allowing for swift reaction to possible security incidents.

File Integrity Monitoring (FIM) is a valuable component in malware detection. Wazuh provides FIM capabilities to monitor and detect changes to files and directories on monitored endpoints. These changes include creation, modification, or deletion. While FIM provides essential insights, combining it with other capabilities and integrations further enhances its effectiveness for malware detection. Wazuh allows security teams to create custom rules based on FIM events, enabling targeted malware detection. These customizable rules correlate FIM events with specific indicators of compromises such as suspicious file extensions, code snippets, or known malware signatures.

Malware commonly targets the Windows Registry in order to achieve destructive objectives such as persistence and other malicious operations. The Wazuh File Integrity Monitoring (FIM) module provides Windows Registry monitoring, which detects changes to regularly used registry paths. When changes occur, the FIM module sends out real-time notifications, allowing security teams to quickly detect and respond to suspicious registry key manipulation.

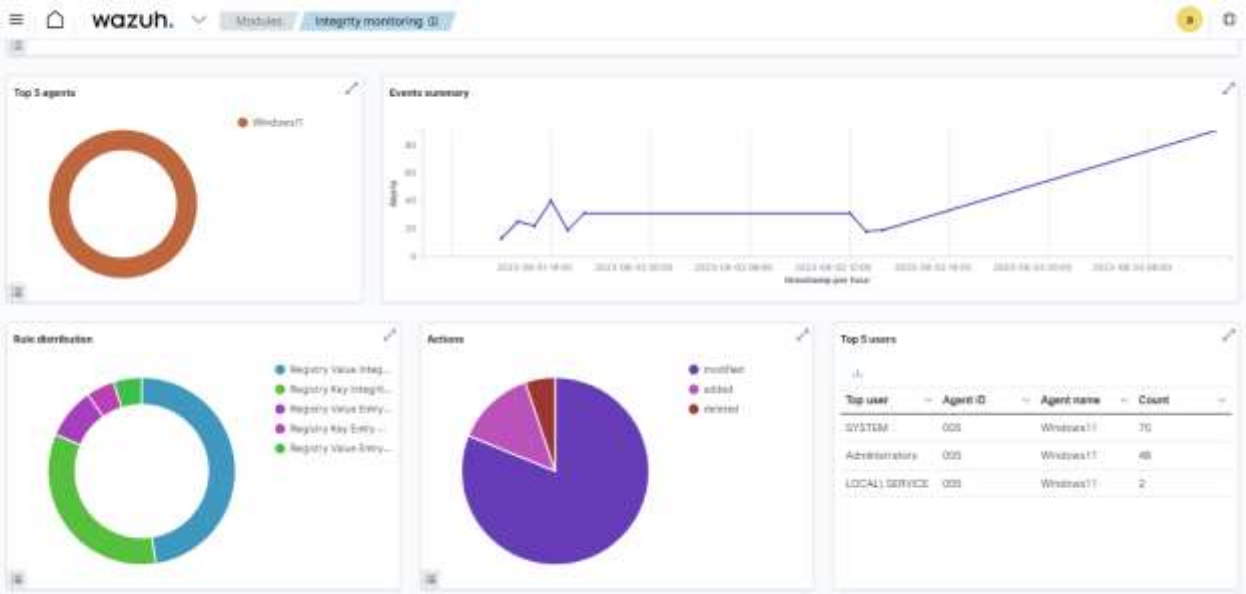
Users can improve their malware detection skills by incorporating **threat intelligence sources.** These intelligence feeds supplement the Wazuh knowledge base by providing up-to-date information on known malicious IP addresses, domains, URLs, and other indicators of compromise.

Rootkits are malicious software that manipulates operating system operations including system calls and kernel data structures to hide the existence of malware on an endpoint. Wazuh has a Rootcheck module, which examines the monitored endpoint on a regular basis for rootkits in both the kernel and user space. The rootcheck detects and alerts on probable rootkit activity. Wazuh discovers rootkit-related patterns quickly by analysing system behaviour and comparing it to known rootkit patterns, and signals for further research are generated.

Wazuh monitors system calls from Linux endpoints to improve malware detection and anomaly identification. Wazuh uses the Linux Audit system to monitor system calls. (“Monitoring system calls - Capabilities · Wazuh documentation”)
System call monitoring, together with Wazuh File Integrity Monitoring (FIM) and threat intelligence integration, improves malware detection. It records security-related events such as file access, command execution, and privilege escalation, providing real-time information on potential security problems. This holistic approach increases organisations' cybersecurity resilience. The graphic below depicts the notifications for privilege abuse on the Wazuh dashboard for Ubuntu Linux 22.04.

Security Dashboard

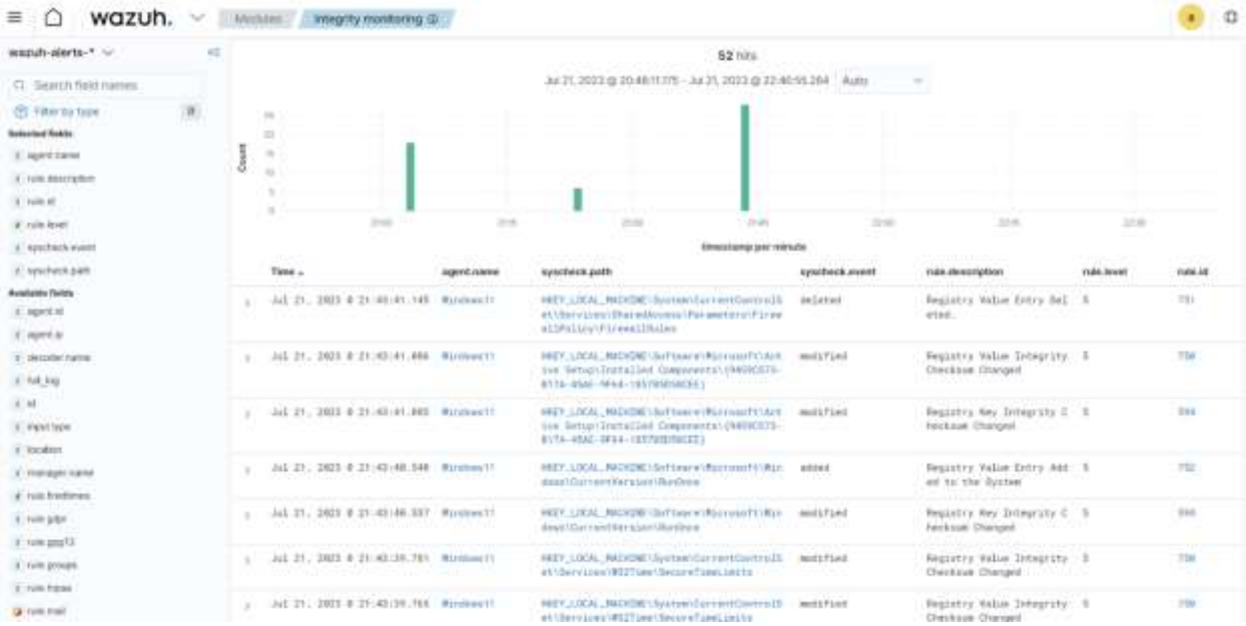
The Wazuh dashboard is a versatile and user-friendly online interface for collecting, analysing, and visualising security event and alarm data. It is also used to manage and monitor the Wazuh platform. It also has capabilities for role-based access control (RBAC) and single sign-on (SSO).



Top 5 users

Top user	Agent ID	Agent name	Count
SYSTEM	005	Windows11	70
Administrators	005	Windows11	48
LOCAL SERVICE	005	Windows11	2

Dashboard 1



52 hits

Jul 21, 2023 @ 20:48:11.775 - Jul 21, 2023 @ 22:40:55.264

Time	agent.name	syscheck.path	syscheck.event	rule.description	rule.level	rule.id
Jul 21, 2023 @ 21:40:41.145	Windows11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\SharedAccess\Parameters\FirewallPolicy\FirewallRules	deleted	Registry Value Entry Deleted.	5	751
Jul 21, 2023 @ 21:43:41.466	Windows11	HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components\{[illegible]}	modified	Registry Value Integrity Checksum Changed	5	750
Jul 21, 2023 @ 21:43:41.465	Windows11	HKEY_LOCAL_MACHINE\Software\Microsoft\Active Setup\Installed Components\{[illegible]}	modified	Registry Key Integrity Checksum Changed	5	594
Jul 21, 2023 @ 21:43:40.548	Windows11	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce	added	Registry Value Entry Added to the System	5	752
Jul 21, 2023 @ 21:43:40.337	Windows11	HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce	modified	Registry Key Integrity Checksum Changed	5	594
Jul 21, 2023 @ 21:43:39.781	Windows11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits	modified	Registry Value Integrity Checksum Changed	5	750
Jul 21, 2023 @ 21:43:39.765	Windows11	HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\W32Time\SecureTimeLimits	modified	Registry Value Integrity Checksum Changed	5	750

Dashboard 2

The web interface allows users to navigate through the many types of data gathered by the Wazuh agent, as well as security warnings issued by the Wazuh server. Users may also build reports, bespoke visualizations, and dashboards.

The Wazuh dashboard allows users to **control agent configurations** and track their status. For example, for each monitored endpoint, users may specify which agent modules will be enabled, which log files will be read, which files will be watched for integrity changes, and which configuration checks will be done.

The Wazuh dashboard is a user interface designed to help you **manage your Wazuh deployment**. This involves tracking the status, logs, and statistics of the various Wazuh components. It also involves setting up the Wazuh server and developing custom rules and decoders for log analysis and threat detection.

The Wazuh dashboard offers a **Ruleset Test tool** that may evaluate log messages to see how they are decoded and if they fit a threat detection rule or not. This functionality is especially handy when the user has written custom decoders and rules and wishes to test them.
The Wazuh dashboard also contains an API Console, which allows users to interact with the Wazuh API. This may be used to manage the Wazuh deployment (for example, changing server or agent configurations, monitoring status and log messages, adding or deleting agents, and so on).

Alerting and Reporting

The Wazuh agent regularly collects and communicates software inventory data from a monitored endpoint to the Wazuh server. The Wazuh Vulnerability Detector module uses software inventory data and vulnerability feeds to identify insecure software on a monitored endpoint. Wazuh analyses insecure programmes and generates risk reports using data gathered from several operating system manufacturers and vulnerability databases. The Vulnerability Detector module makes use of a database of Common Vulnerabilities and Exposures (CVEs) generated automatically by analysing data from several sources, including Wazuh feeds.

Wazuh assists in the implementation of compliance standards in order to support and increase regulatory visibility. This is achieved through automation, enhanced security controls, log analysis, and incident response.
The default Wazuh ruleset supports the PCI DSS, HIPAA, NIST 800-53, TSC, and GDPR frameworks and standards. Wazuh rules and decoders identify attacks, system problems, security misconfigurations, and policy breaches.

Integration with open-source tool

After going through documentation of **Wazuh** I have realized that it powerful tool which provides nitty gritty details of your entire networks but one thing that where I felt that wazuh was struggling is the network devices like firewall, switches, routers etc as these devices don’t support wazuh agent.

But wazuh server is still able to receive log generated by these devices through syslog server.

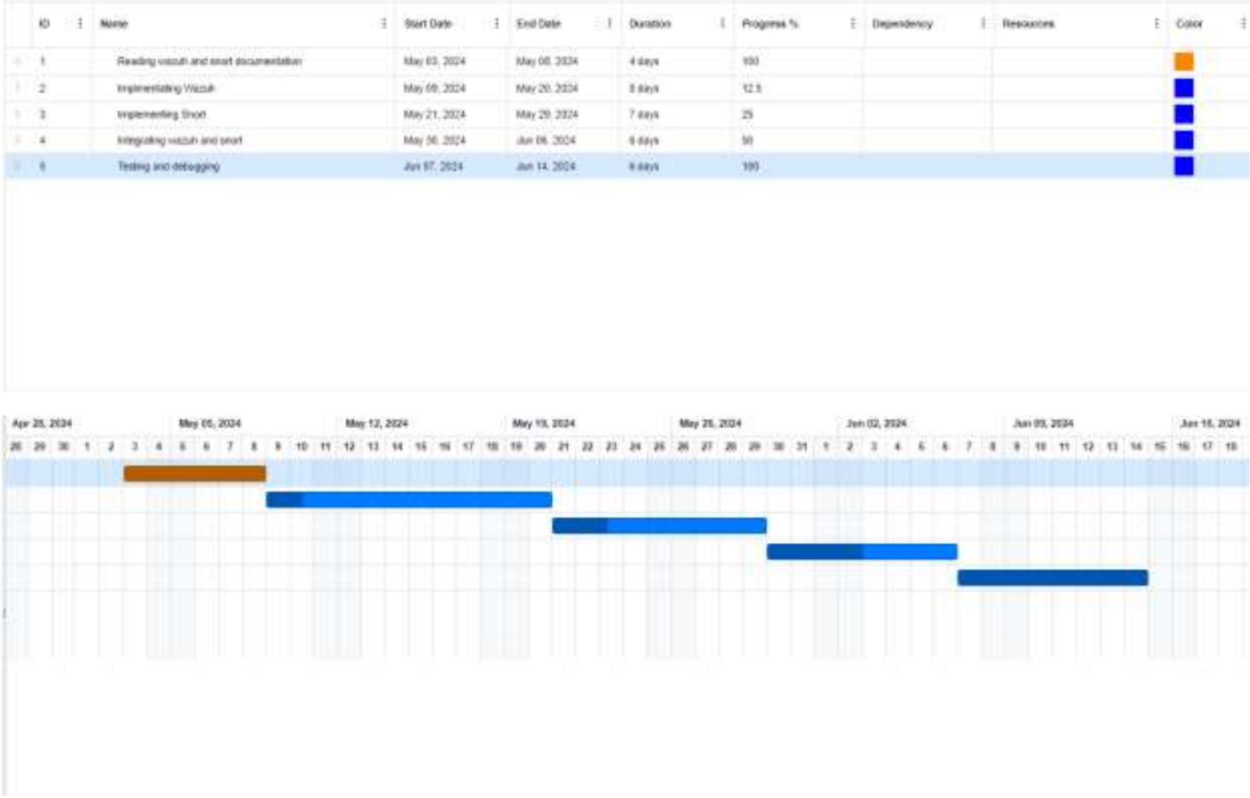
However, I think we can integrate **Snort** would be best option to integrate with wazuh for getting more detailed information that isn’t captures by the network devices like firewall, switches, routeres etc.

The reason behind suggesting snort is because being a packet sniffer it will ‘sniff’ out security threats to networks. It detects and reports attack methods, thereby sending an alert to syslog or through another channel. It conducts real-time traffic analysis along with logs. (“10 Leading Open

Source SIEM Tools - 2023 Update | Logz.io") It is designed to detect a long list of different attack vectors that includes OS fingerprinting, DDOS, CGI, SMB probes, buffer overflows and stealth port scans. It uses OpenAppID to detect applications.

Now snort will report to syslog and from syslog wazuh can easily retrieve the data. Moreover, Snort can easily be integrated with wazuh.

Integration plan



ID	Name	Start Date	End Date	Duration	Progress %	Dependency	Resources	Color
1	Reading wazuh and snort documentation	May 03, 2024	May 06, 2024	4 days	100			🟧
2	Implementating Wazuh	May 09, 2024	May 20, 2024	8 days	12.5			🟦
3	Implementing Snort	May 21, 2024	May 29, 2024	7 days	25			🟦
4	Integrating wazuh and snort	May 30, 2024	Jun 06, 2024	6 days	50			🟦
5	Testing and debugging	Jun 07, 2024	Jun 14, 2024	6 days	100			🟦

Bibliography

[1] Wazuh, "Wazuh documentation," [Online]. Available: https://documentation.wazuh.com/current/getting-started/components/wazuh-agent.html. [Accessed 19 4 2024].

[2] CHatgpt. [Online]. Available: https://chat.openai.com/c/1ffbe741-ab85-4ded-b170-aec4e1c6dff4.

[3] Gemini. [Online]. Available: https://gemini.google.com/app/125f249c986c200e.

[4] G. documentation. [Online]. Available: https://go2docs.graylog.org/5-2/what_is_graylog/what_is_graylog.htm. [Accessed 19 4 2024].

[5] Logz.io. [Online]. Available: https://logz.io/blog/open-source-siem-tools/. [Accessed 19 4 2024].