ENERGY
LOGSERVER

SIEM **FOR BEGINNERS**

ALL YOU WANTED TO KNOW ABOUT LOG MANAGEMENT
BUT DIDN'T DARE TO ASK

www.energylogserver.com

SIM, SEM and SIEM Evolution

ENERGY LOGSERVER

SIEM is a class of solutions that is often underestimated despite its recognition. **SIEM** tasks include centralized collection and management of security events from different systems. It was preceded by many solutions that sought to simplify and automate the control of security tools.

Two common types of such systems are:

- SIM – Security Information Management
- SEM – Security Event Management

Although the definitions look similar, there is a huge difference between them:

1. This is more an evolution of one idea for collecting events than two options for its implementation.
2. SIM automates the collection of logs, while SEM – any security events in general. Thus, the sources of obtaining information differ: SIM has software and hardware solutions, while SEM has Oracle and MySQL databases.
3. SIM allows you to simplify the search for logs through automatic collection and sorting. SEM has the functions of contextual analysis and correlation of the collected data – it looks for events with a common denominator and can conclude a threat or a problem from them.

SIEM (Security Information and Event Management) combines the functions of these two systems, allowing you to manage information and events generated by other security solutions centrally.

It collects and analyzes the log entries that your assets generate. So everything that is recorded by your antiviruses, firewalls, data loss prevention systems, endpoint protection, privileged access management, and others is now collected in one single solution – **SIEM**.

What Is the Reason for the SIEM Emergence and the Need for Solutions of This Class?

An increase in the amount of information that specialists must analyze. In the era of digitalization, companies have realized that typical tasks are constantly becoming more complex. For each of them, now you need to understand and connect so many events that it is impossible to compare them correctly and draw a conclusion on the go.

Attacks become more complex, adding the number of markers that issue them. Modern threats include several vectors at once – from banal phishing to the introduction of intelligent malware, and the attackers themselves qualitatively mask their actions and bypass protection systems. Thus, no single security solution will give you complete information that something bad is happening.

Still, each particular solution has a narrow specification and provides you with information out of context. Tools analyze facts without reference to the situation in which they occur. It is not enough because not a single attack occurs in some isolated framework. An attack is not an event but a combination of them.

Compare: an employee on vacation briefly checks corporate mail – a hacker steals his data and also enters the mail. For us, these are two completely different situations in terms of importance and essence, let alone the consequences and reactions.

For access control systems, these are just two logins to the account from different IP and MAC addresses. By scaling the situation to dozens of employees and hundreds of processes, we get security full of unnecessary and often useless routines. To check all such suspicious cases, you need to understand further user actions and changes in the system: check file access requests, suspicious software downloads, network scans and attempts to increase an access level. All of this is in different solutions; thus, you need to check each.

**Your solutions cannot detect an attack independently, but they provide all the needed information.
The trick is to find it.**

Investigating the Incident with SIEM

Let's analyze in practice what processes take place inside the solution and how they lead to the result. Below you can find a step-by-step demonstration of how to solve an incident investigation task.

We recreated an actual attack on a separate host in a test infrastructure. As planned, it all started with a notification about a suspicious incident from the attacked computer. Our task is to investigate this incident as an information security specialist. First, let's determine if this event is an attack. If yes, we will drill into it to the smallest detail. If not, we will find out what happened and how to avoid such worries.

Another critical condition: we will assume that there are no gaps in the protection system and that **SIEM** is correctly integrated into all solutions. In other words, it is possible to solve the problem; the system has information about it, and all you have to do is to find it.

The conditions accepted. Now moving on to **SIEM**.

Logs Collection and Information from the System

ENERGY LOGSERVER

Log Management is at the heart of any **SIEM** – a centralized, automated collection of log records in one place. We have already discovered that we can detect an attack by a combination of events, which means that at the first stage, we need data from the logs of each decision.

Information Security Tools

- firewalls (VPN concentrators, web filters)
- IDS/IPS
- endpoint protection (antivirus, EDR, etc.)
- DLP
- PAM, WAF, MFA

Asset data:

- configuration
- location
- users
- ports and protocols
- vulnerability reports
- software inventory

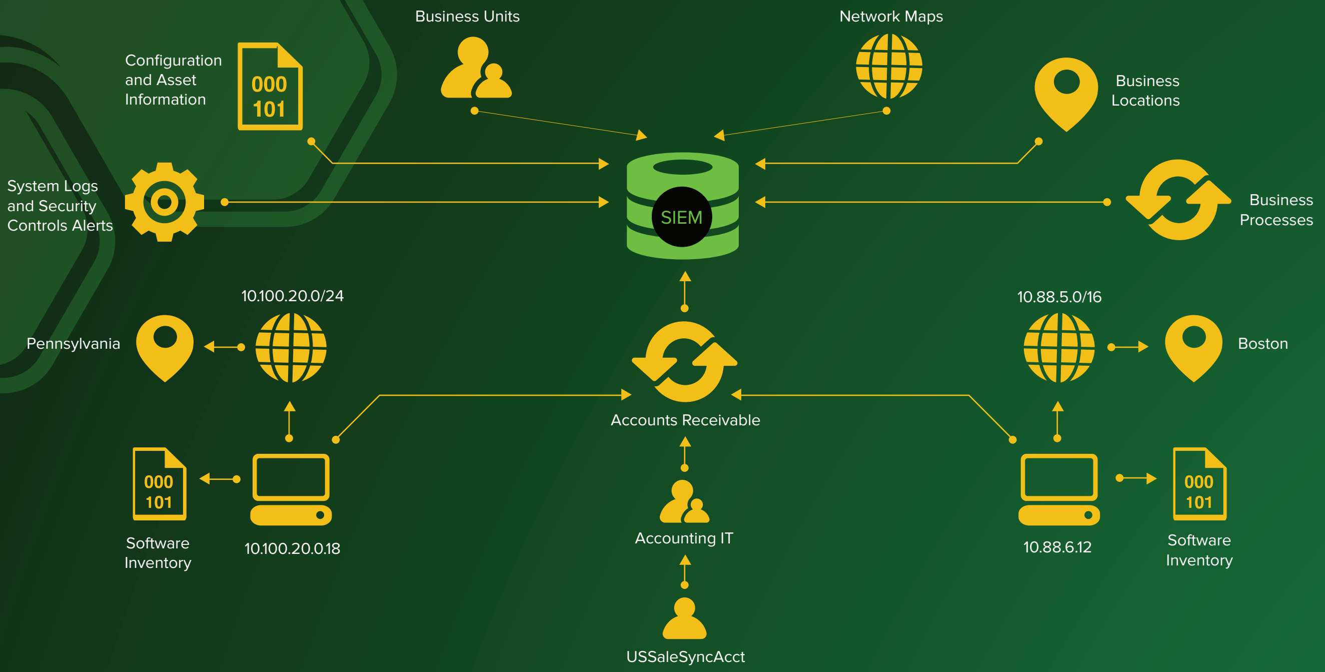
Infrastructure:

- routers, switches, access points
- domain controllers
- application servers, corporate portals and services
- database
- OS and cloud service logs

Logs Generating on Your Network

ENERGY LOGSERVER

Each source listed above creates tens and hundreds of records daily; we cannot change this. How logs are generated on your network – see the diagram below.



10.100.20.18 initiated a database copy using USSalesSyncAcct credentials on a remote host 10.88.6.12 - Status Code 0x44F8

Now imagine that similar processes are repeated a thousand times daily, creating a whole array of information, including valuable data. Without **SIEM**, all this will be just "dead weight", which you cannot sort out manually.

How Does It Work?

With the help of log management, we can get from the storage and view the log of this event, which is generated on the attacked computer.

It looks like that:

```
5ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯
b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(1
5ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49▯b(15ssh:nottyroot192.168.202.49ý¹b(15ssh:nottyroot192.168.202.49ý¹b(1¸5s
sh:nottyroot192.168.202.49ÿ¹b(1·5ssh:nottyroot192.168.202.49ÿ¹b(1½5ssh:nottyroot192.168.202.49°b(1¿5ssh:nottyroot192.168.202.49°b(1¸5ssh:nottyroot192.168.202.49°b(1·5ssh:nottyroot192.168.202.49°b(1 ¿5ssh:nottyroot192.168.202.49°b(1½5ssh:nottyroot192.168.202.49°
b(1¸5ssh:nottyroot192.168.202.49°b(1·5ssh:nottyroot192.168.202.49°b(1¿5ssh:nottyroot192.168.202.49°b(1½5ssh:nottyroot192.168.202.49 °b(1¸5ssh:nottyroot192.168.202.49°b(1·5ssh:nottyroot192.168.202.49°b(1¿5ssh:nottyroot192.168.202.49°b(1½5ssh:nottyroot192.168.202.
49°b(1¸5ssh:nottyroot192.168.202.49°b(1·5ssh:nottyroot192.168.202.49°b(1¿5ssh:nottyroot192.168.202.4°b(1½5ssh:nottyroot192.168.202.4°b(1¸5ssh:nottyroot192.168.202.4°b(1·5ssh:nottyroot192.168.202.4°b(1¿5ssh:nottyroot192.168.202.49 °b(1½5ssh:nottyroot192.168.202.49
°b(15ssh:nottyroot192.168.202.49(°b(15ssh:nottyroot192.168.202.49(°b(15ssh:nottyroot192.168.202.49*°b(15ssh:nottyroot192.168.202.49*°b(15ssh:nottyroot192.168.202.49*°b(15ssh:nottyroot192.168.202.49*°b(15ssh:nottyroot192.168.202.49,°b(15ssh:nottyroot192.168.202.49
,°b(15ssh:nottyroot192.168.202.49,°b(15ssh:nottyroot192.168.202.49,°b(15ssh:nottyroot192.168.202.49.°b(15ssh:nottyroot192.168.202.49.°b(15ssh:nottyroot192.168.202.49.°b(15ssh:nottyroot192.168.202.49.°b(15ssh:nottyroot192.168.202.490°b(15ssh:nottyroot192.168.202.4
90°b(15ssh:nottyroot192.168.202.490°b(15ssh:nottyroot192.168.202.490°b(15ssh:nottyroot192.168.202.491°b(15ssh:nottyroot192.168.202.491°b(15ssh:nottyroot192.168.202.492°b(15ssh:nottyroot192.168.202.492°b(15ssh:nottyroot192.168.202.494°b(15ssh:nottyroot192.168
.202.494°b(16ssh:nottyroot192.168.202.49R°b(16ssh:nottyroot192.168.202.49R°b(16ssh:nottyroot192.168.202.49S°b(16ssh:nottyroot192.168.202.49S°b(16ssh:nottyroot192.168.202.49T°b(1 6ssh:nottyroot192.168.202.49T°b(16ssh:nottyroot192.168.202.49V°b(1 6ssh:nottyroo
t192.168.202.49V°b(16ssh:nottyroot192.168.202.49V°b(16ssh:nottyroot192.168.202.49V°b(16ssh:nottyroot192.168.202.49W°b(1 6ssh:nottyroot192.168.202.49W°b(16ssh:nottyroot192.168.202.49X°b(16ssh:nottyroot192.168.202.49X°b(1 6ssh:nottyroot192.168.202.49Y°b(1 6ss
h:nottyroot192.168.202.49Y°b(16ssh:nottyroot192.168.202.49Z°b(16ssh:nottyroot192.168.202.49Z°b(16ssh:nottyroot192.168.202.49[°b(1 6ssh:nottyroot192.168.202.49[°b(16ssh:nottyroot192.168.202.49\°b(16ssh:nottyroot192.168.202.49\°b(16ssh:nottyroot192.168.202.49\°b(1
6ssh:nottyroot192.168.202.49\°b(1>6ssh:nottyroot192.168.202.49|°b(1?6ssh:nottyroot192.168.202.49|°b(1D6ssh:nottyroot192.168.202.49}°b(1F6ssh:nottyroot192.168.202.49}°b(1>6ssh:nottyroot192.168.202.49}°b(1?6ssh:nottyroot192.168.202.49}°b(1D6ssh:nottyroot192.168.20
2.4°b(1F6ssh:nottyroot192.168.202.4°b(1>6ssh:nottyroot192.168.202.4°b(1?6ssh:nottyroot192.168.202.4°b(1D6ssh:nottyroot192.168.202.49°b(1F6ssh:nottyroot192.168.202.49°b(1>6ssh:nottyroot192.168.202.49°b?6ssh:nottyroot192.168.202.49°b(1¼Hssh:nottyroot192.168.20
2.49¿ýb(1½Hssh:nottyroot192.168.202.49¿ýb(1»Hssh:nottyroot192.168.202.49¿ýb(1¾Hssh:nottyroot192.168.202.49¿ýb(1¼Hssh:nottyroot192.168.202.49ëýb(1½Hssh:nottyroot192.168.202.49ëýb(1»Hssh:nottyroot192.168.202.49ëýb(1¾Hssh:nottyroot192.168.202.49ëýb(1 ¼Hss
h:nottyroot192.168.202.49ëýb(1½Hssh:nottyroot192.168.202.49ëýb(1»Hssh:nottyroot192.168.202.49ëýb(1¾Hssh:nottyroot192.168.202.49ëýb(1¼Hssh:nottyroot192.168.202.49ëýb(1½Hssh:nottyroot192.168.202.49ëýb(1»Hssh:nottyroot192.168.202.49ëýb(1¾Hssh:nottyroot192
.168.202.49ëýb(1¼Hssh:nottyroot192.168.202.49ëýb(1»Hssh:nottyroot192.168.202.49ëýb(1½Hssh:nottyroot192.168.202.49ëýb(1¾Hssh:nottyroot192.168.202.49ëýb(1¼Hssh:nottyroot192.168.202.49ëýb(1»Hssh:nottyroot192.168.202.49ëýb(1½Hssh:nottyroot192.168.202.49ëý
b(1¾Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.16
8.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyr
oot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1Hssh:nottyroot192.168.202.49ëýb(1H
```

From this example, it is evident that analyzing the logs on your own is useless. Moreover, in a real company, there will be dozens of such logs, and if you can cope with one, in theory, you won't be able to work at this pace all the time.

Conclusion: Log Management helps us find old records, but more is needed to get some valuable information. Fortunately, **SIEM** is just beginning with this.

Preparing Data for Analysis

All log sources usually record events in their way. These records are supposed to be seen by humans, not machines, so different log formats are not generally considered a problem. But in our case, it is crucial to be able to process records with **SIEM** tools, which means that the solution must first bring all available logs to a certain common denominator.

E.g., in the log

```
“User Broberts Successfully Authenticated to 10.100.52.105 from client 10.10.8.22”
```

it is necessary to select elements common to all logs

```
“User [USERNAME] [STATUS] Authenticated to [DESTIP] from client [SOURCEIP]”
```

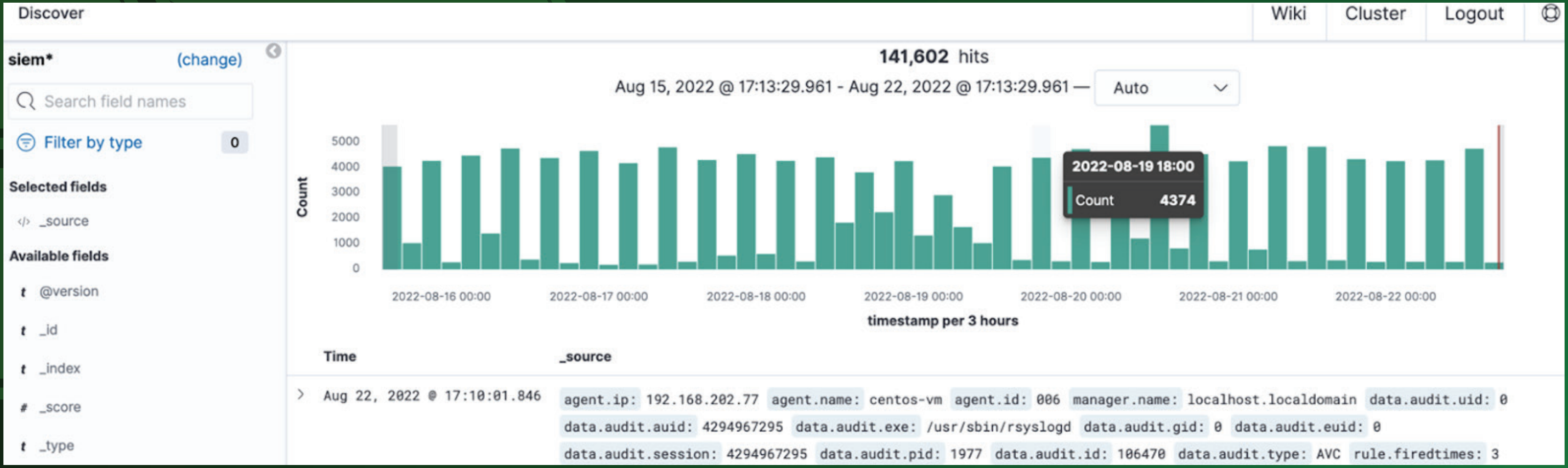
This process is called normalization. Thanks to it, **SIEM** processes tons of records at the level of mathematical models and statistical methods. In other words, it filters and sorts logs. The difference between log management with and without normalization is that without normalization, we can only process logs from one source. Therefore, we will not be able to filter the logs of different sources by any common criterion.

This is what we need for subsequent actions because the essence of **SIEM** is to supplement information from one sources with data from others.

How Does It Work?

Having the opportunity to add some statistics and filter the data, it becomes possible to structure the log and see something.

Here's what it looks like in the console:



```
Aug 22, 2022 @ 17:10:01.846   agent.ip: 192.168.202.77 agent.name: centos-vm agent.id: 006 manager.name: localhost.localdomain data.audit.uid: 0
                              data.audit.auid: 4294967295 data.audit.exe: /usr/sbin/rsyslogd data.audit.gid: 0 data.audit.euid: 0
                              data.audit.session: 4294967295 data.audit.pid: 1977 data.audit.id: 106470 data.audit.type: AVC rule.firedtimes: 3
```

We see account login attempts that fail due to incorrectly entered credentials. **SIEM** recorded the time, IP addresses and other characteristics of requests. We see that they were repeated many times, we see the passwords used, and we also note the difference in milliseconds between requests.

This is a sign of machine password guessing, but why should we conclude on our own if we have **SIEM**? Let's keep on investigating.

The Essence of the Correlation Process

Correlation allows you to find the relationship between different events, link them into one chain and compare them with the state of the norm. As a result, we get classified events according to the criteria we have adopted (risk management, alerts, prioritization, etc.). And if we add integration with bases of indicators of compromise, we will get the possibility of more effective detection of threats using **SIEM**, including zero-day attacks.

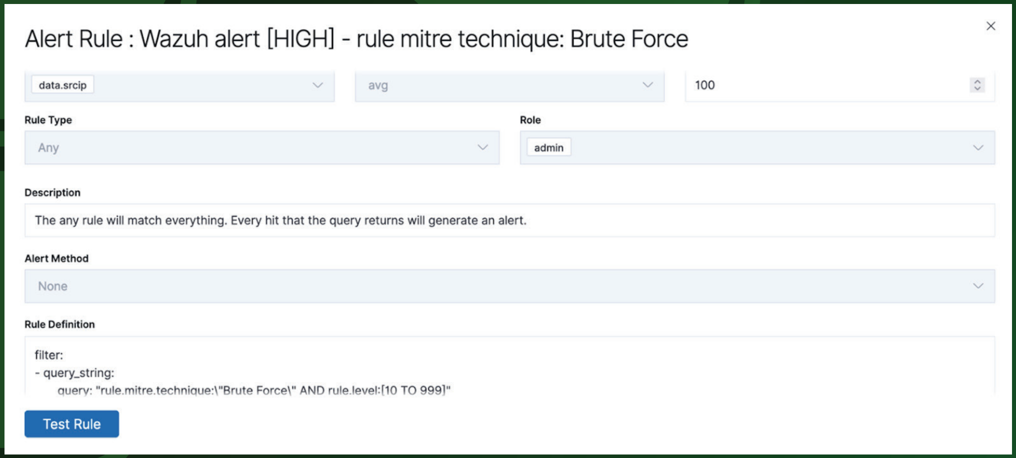
Detection rules integrated with the notification and response system allow you to configure reaction scenarios for events that should not occur in your system. A properly configured system will be able to isolate marker events from the entire variety of logs and notify you of a potential threat.

Suppose one event from a particular solution can be both wholly harmless and indicate a threat. However, several such pieces of evidence from different sources already show the fact of an attack. The presence of information about the normal state of the system creates a context for the analyzed events – the system does not evaluate events according to the principle “is it good or bad?” but finds the answer to the question “Should this be happening under **current conditions?”**

How Does It Work?

Normalization and correlation allow us to do whatever we want with our as-yet incomprehensible log. So let's do something useful!

Enable notification for the type of Brute Force attacks



Alert Rule : Wazuh alert [HIGH] - rule mitre technique: Brute Force

data.srcip | avg | 100

Rule Type: Any

Role: admin

Description

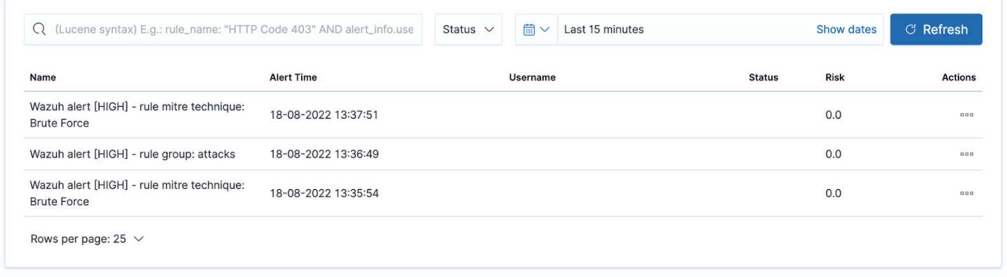
The any rule will match everything. Every hit that the query returns will generate an alert.

Alert Method: None

Rule Definition

```
filter:
- query_string:
    query: "rule.mitre.technique:\"Brute Force\" AND rule.level:[10 TO 999]"
```

This immediately gives us a list of rule operation (with the ability to delve into each incident):



(Lucene syntax) E.g.: rule_name: "HTTP Code 403" AND alert_info.use
Status
Last 15 minutes
Show dates
Refresh

Name	Alert Time	Username	Status	Risk	Actions
Wazuh alert [HIGH] - rule mitre technique: Brute Force	18-08-2022 13:37:51			0.0	•••
Wazuh alert [HIGH] - rule group: attacks	18-08-2022 13:36:49			0.0	•••
Wazuh alert [HIGH] - rule mitre technique: Brute Force	18-08-2022 13:35:54			0.0	•••

Rows per page: 25

— It ‘s getting easier, isn’t it?

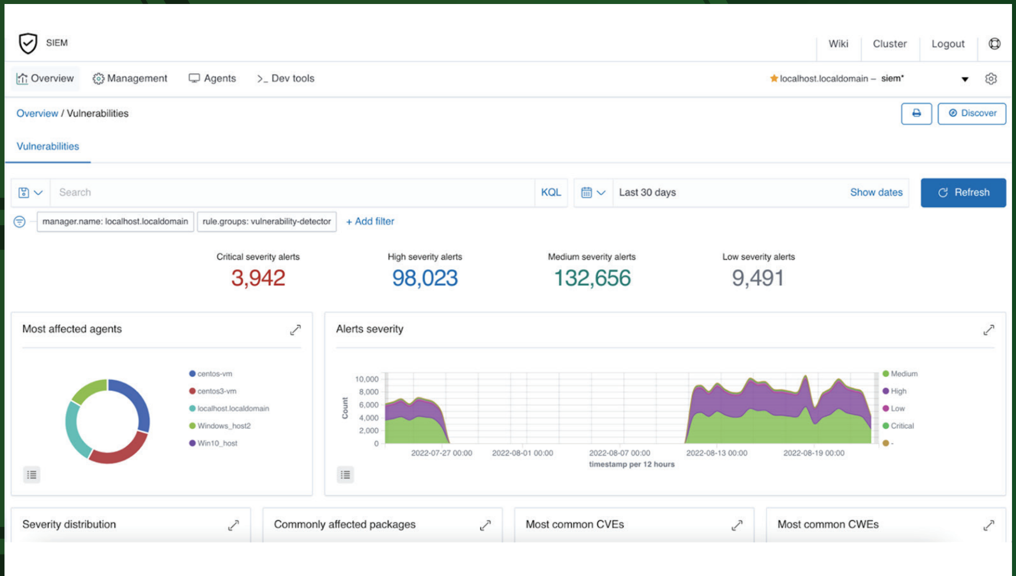
But that's not all – we have already found and highlighted the event, and we can surely see that we had an attack like Brute Force.

But this is not enough for us. So let's go back to the big picture for a second, see how things are going, and dive back into our incident.

— This time, deeper.

How Does It Work?

General dashboard for all events: a glance is enough to assess the situation and get essential metrics. Of course, all this can be configured and customized to suit your needs.



Last 30 days

manager.name: localhost.localdomain

rule.groups: vulnerability-detector

Critical severity alerts	High severity alerts	Medium severity alerts	Low severity alerts
3,942	98,023	132,656	9,491

Most affected agents

Alerts severity

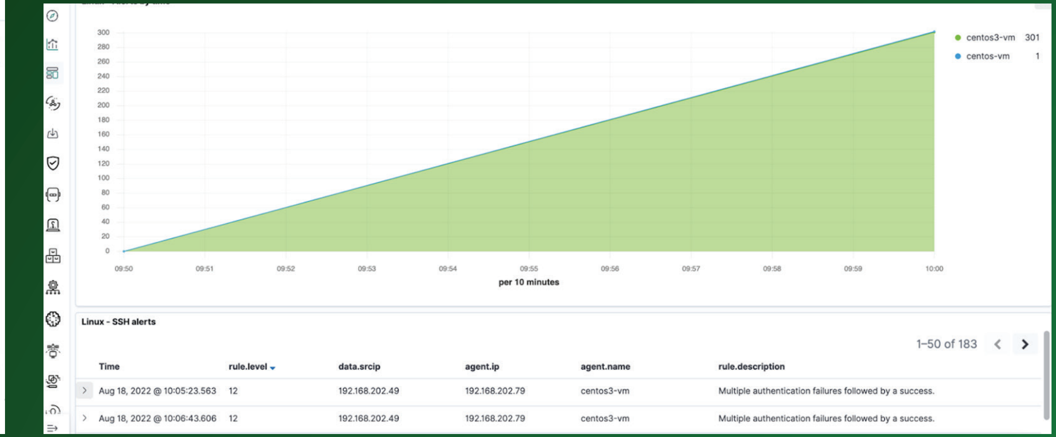
Severity distribution

Commonly affected packages

Most common CVEs

Most common CWEs

We highlight attacks on a specific host:



Linux - SSH alerts

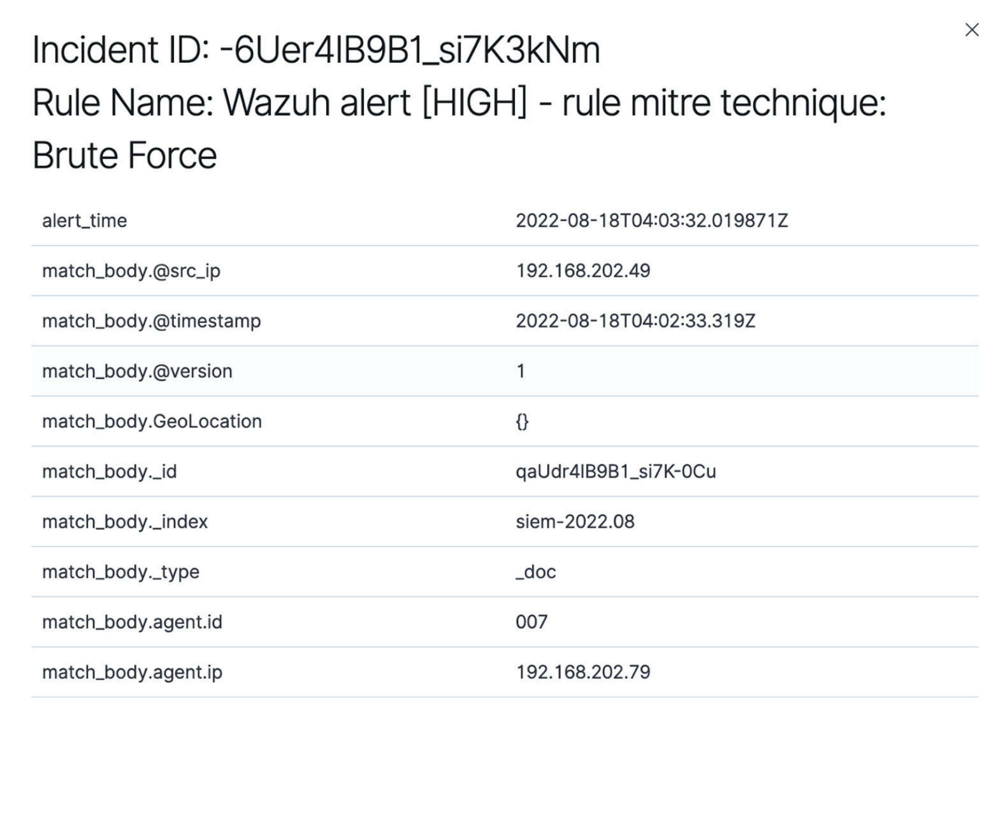
1–50 of 183

Time	rule.level	data.srcip	agent.ip	agent.name	rule.description
Aug 18, 2022 @ 10:05:23.563	12	192.168.202.49	192.168.202.79	centos3-vm	Multiple authentication failures followed by a success.
Aug 18, 2022 @ 10:06:43.606	12	192.168.202.49	192.168.202.79	centos3-vm	Multiple authentication failures followed by a success.

How Does It Work?

Next, click on our event and study the details.

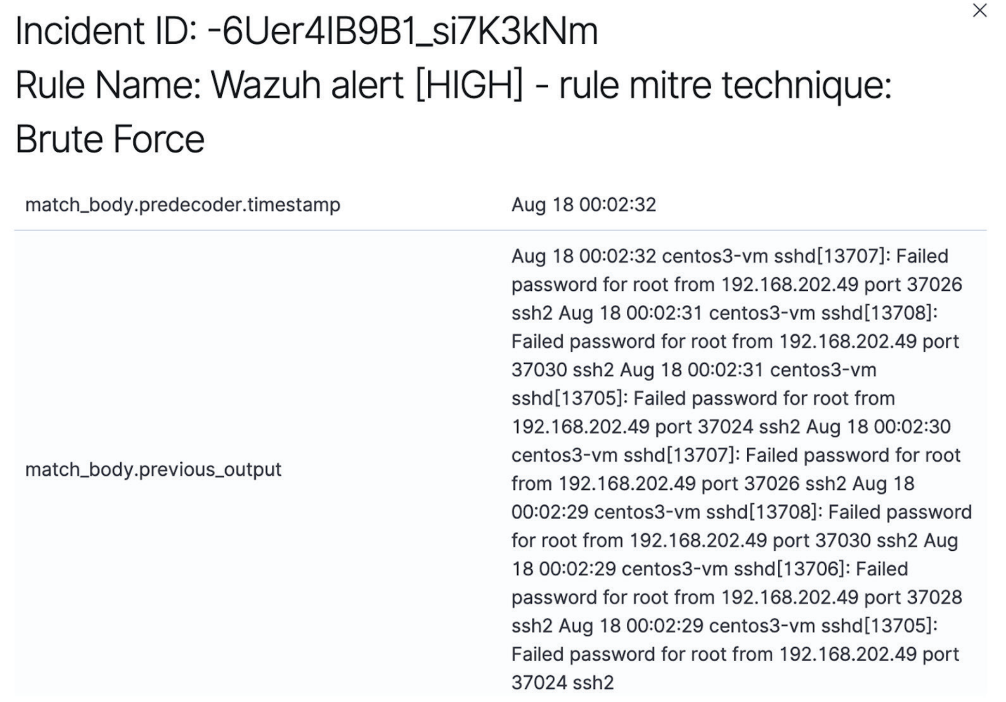
For example, which source from which was attacked ...



Incident ID: -6Uer4IB9B1_si7K3kNm
Rule Name: Wazuh alert [HIGH] - rule mitre technique: Brute Force

alert_time	2022-08-18T04:03:32.019871Z
match_body.@src_ip	192.168.202.49
match_body.@timestamp	2022-08-18T04:02:33.319Z
match_body.@version	1
match_body.GeoLocation	{}
match_body._id	qaUdr4IB9B1_si7K-0Cu
match_body._index	siem-2022.08
match_body._type	_doc
match_body.agent.id	007
match_body.agent.ip	192.168.202.79

...here – information that the passwords were wrong, that is, the attack failed...



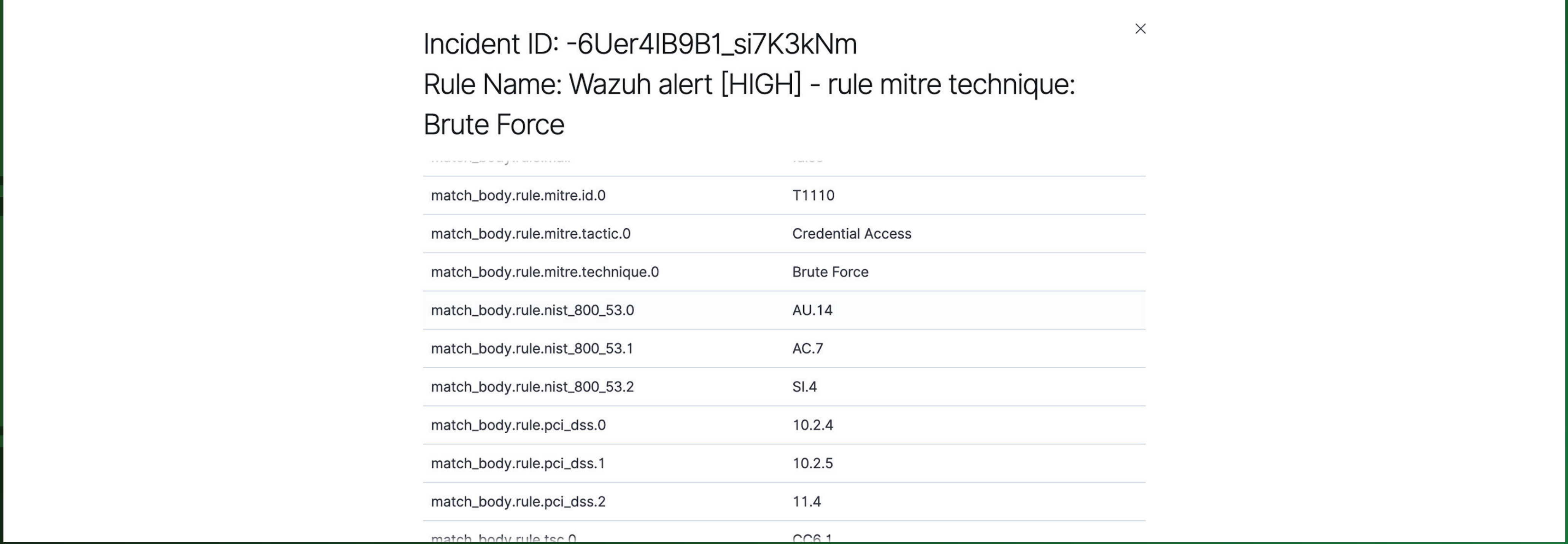
Incident ID: -6Uer4IB9B1_si7K3kNm
Rule Name: Wazuh alert [HIGH] - rule mitre technique: Brute Force

match_body.predecoder.timestamp	Aug 18 00:02:32
match_body.previous_output	Aug 18 00:02:32 centos3-vm sshd[13707]: Failed password for root from 192.168.202.49 port 37026 ssh2 Aug 18 00:02:31 centos3-vm sshd[13708]: Failed password for root from 192.168.202.49 port 37030 ssh2 Aug 18 00:02:31 centos3-vm sshd[13705]: Failed password for root from 192.168.202.49 port 37024 ssh2 Aug 18 00:02:30 centos3-vm sshd[13707]: Failed password for root from 192.168.202.49 port 37026 ssh2 Aug 18 00:02:29 centos3-vm sshd[13708]: Failed password for root from 192.168.202.49 port 37030 ssh2 Aug 18 00:02:29 centos3-vm sshd[13706]: Failed password for root from 192.168.202.49 port 37028 ssh2 Aug 18 00:02:29 centos3-vm sshd[13705]: Failed password for root from 192.168.202.49 port 37024 ssh2

How Does It Work?

ENERGY LOGSERVER

...and here is what attack method was used:



Incident ID: -6Uer4IB9B1_si7K3kNm
Rule Name: Wazuh alert [HIGH] - rule mitre technique: Brute Force

match_body.rule.mitre.id.0	T1110
match_body.rule.mitre.tactic.0	Credential Access
match_body.rule.mitre.technique.0	Brute Force
match_body.rule.nist_800_53.0	AU.14
match_body.rule.nist_800_53.1	AC.7
match_body.rule.nist_800_53.2	SI.4
match_body.rule.pci_dss.0	10.2.4
match_body.rule.pci_dss.1	10.2.5
match_body.rule.pci_dss.2	11.4
match_body.rule.tsc.0	CC6.1

In a genuinely incomprehensible set of characters we received from the host, we discovered and examined the attack to the smallest detail.

Now ask yourself – would it be easy to achieve the same result without **SIEM**?

What’s the Use?

Imagine that your company generates hundreds of GB of text logs daily. All this array hides precisely the information that will indicate an attack or a critical vulnerability. But now you see that processing even one log is a challenging task. Even if you hire a whole army of specialists whose only mission is to read logs, the human factor will sooner or later intervene in the process.

By the way, the expansion of security departments sometimes tries to compensate for the lack of opportunities for analyzing existing events. More people will do more work – that's a fact. But considering the number of such tasks, as in our example, such use of human resources still remains inefficient.

A **SIEM** system is a tool that will save specialists from routine and free up time for strategy, analysis and response. **SIEM** processes in seconds thousands of cases from our example. Thus, the team gets the time and energy to optimize the infrastructure, respond to attacks, and implement improvements. The alternative is to sort through a bunch of events and false positives for days, which only leads to maintaining the current norm.

As a result, **SIEM** will make the work of all solutions more efficient since essential events and other helpful information detected by them will not pass by specialists. At the same time, the specialists themselves will be more productive because now they have time to deal with everything that truly needs their intervention.

Energy Logserver — a New Generation SIEM

ENERGY LOGSERVER

Energy Logserver's **SIEM** has erased the upper limit of possibilities. The solution is available in three basic configurations, each with a specific set of functions for solving tasks of different levels. This framework can be scaled and extended as you wish without any limits.

Log Management — all about managing event logs

- Role access model
- Integration with LDAP, AD, Radius, SSO
- Scalable architecture and clustering
- Hundreds of ready-made parsers, flexible constructor
- Multidimensional notification and reporting system
- Multilevel system of data archiving
- Flexible dashboard constructor and ready templates

SIEM — extended security management

- Behavioural analysis of users and devices
- Dynamic integration with IoC, TTP, Threat Intelligence, MITRE ATT&CK databases
- GDPR, NIST, CIS, PCI DSS, HIPAA compliance
- File Integrity Monitoring (FIM)
- Vulnerability scanner
- 1000+ detection and correlation rules
- AI-based assistance in detecting suspicious behaviour
- Playbooks
- Applications and services monitoring
- Integrated risk management system
- Incident management system

Network Probe — network analysis

- Detailed analysis of network traffic
- Netflow analysis (v5, v9, IPFIX, sflow. iflow. NetStrean)
- Productivity from 10 Gbps
- From 100 000 FPS
- Traffic visualisation in levels L2-L7
- Correlation of logs and traffic data
- Traffic checks according to reputation databases and IoC
- Detection of zero-day attacks
- Analysis of user network activity
- Monitoring of SRT, RTT, Delay, Jitter, network anomalies

Contacts

ENERGY
LOGSERVER©

Energy Logserver is a tool that doesn't force you to compromise and reduce requirements. On the contrary, it expands your possibilities and adds control in the most unbelievable areas.

To get the Energy Logserver demo, contact us

sales@energylogserver.com.

www.energylogserver.com