Stop Touching SIEM Until You Know These Fundamentals

📌 INTRODUCTION — READ THIS BEFORE SIEM

Most people learn SIEM **the wrong way**.

They jump straight to:
alerts, rules, and dashboards.

That's why they fail.

❌ What SIEM is NOT

- Not a detection engine
- Not an alert machine
- Not cybersecurity itself

Alerts don't equal truth.

✅ What SIEM actually is

SIEM (Security Information and Event Management) is a **decision-support system** for SOC analysts.

It collects activity from many systems, structures it, connects related behavior, and presents evidence so a **human** can decide:

Normal, suspicious, or incident?

🧠 The core problem SIEM solves

Modern systems generate **massive noise**.
Raw logs are messy, inconsistent, and misleading.

SIEM exists to **organize chaos**, not to declare attacks.

🔍 How SIEM creates clarity

SIEM:

- parses logs into fields
- normalizes meaning across platforms
- aggregates volume
- correlates behavior over time

So analysts can see **who, what, where, and in what order**.

⚠️ Why fundamentals matter

If you don’t understand:
logs vs events,
alert vs incident,
correlation vs aggregation,
time windows and severity,

the SIEM will confidently show you **wrong conclusions**.

🧭 Correct mindset

SIEM is not a judge.

Logs = evidence
Alerts = claims
Analyst = decision-maker

1️⃣ WHAT IS SIEM (CORE DEFINITION)

SIEM = Security Information and Event Management

Simple definition:
A SIEM collects logs from many systems, converts them into structured events, correlates related activity, and helps analysts detect and investigate security incidents.

Real-world example:

- **Firewall logs → blocked connections**
- **Windows logs → login failures**
- **Linux logs → sudo usage**

SIEM brings all of these into **one timeline**, so you can answer:

- Who did what?
- From where?
- On which system?
- In what order?

SIEM vs SEM

- **SIM** → Storage + visibility (logs, history)
- **SEM** → Real-time analysis + alerting
- **SIEM** → Both combined

2️⃣ LOGS

Definition:
Logs are **raw records of activity** generated by systems, applications, or devices.

Hard truth:
Logs are **not security events**. They are just text until processed.

Example:

```
Accepted password for root from 10.0.0.5 port 54321 ssh2
```

That's a log.
It means nothing *yet*.

3️⃣ EVENTS

Definition:
An event is a **structured, meaningful representation of a log** after parsing and normalization.

Why events matter:
SOC conclusions are defended using **fields**, not raw text.

Example event fields:

- **user = root**
- **src_ip = 10.0.0.5**
- **action = success**
- **service = ssh**

Logs → Events = chaos → evidence

4️⃣ NORMALIZATION

Definition:
Normalization converts different log formats into a **common schema**.

Why this is critical:
Different systems say the same thing differently.

Example:

- **Windows: EventID=4625**
- **Linux: Failed password**
- **Firewall: deny**

Normalization turns all of them into:

action=failed_authentication

Without normalization:

- correlation fails
- rules break

- detections lie

5️⃣ CORRELATION

Definition:
Correlation links **multiple related events** to identify suspicious behavior.

Important:
One event ≠ attack
Pattern over time = signal

Example:

- **10 failed logins**
- **followed by 1 success**
- **from the same IP**
- **within 2 minutes**

That's correlation → **brute-force behavior**

6️⃣ AGGREGATION

Definition:
Aggregation groups **similar events** to reduce noise.

Example:
Instead of:

- **500 failed login events**

SIEM shows:

- **500 failed logins**
- **from 1 IP**
- **targeting 1 user**
- **in 5 minutes**

Aggregation makes volume understandable.

7️⃣ PARSING

Definition:
Parsing extracts **fields** from raw log text.

If parsing is wrong:

- fields are missing
- searches are unreliable
- detections are invalid

Example:
Raw log ❌

Failed password for admin from 10.0.0.8

Parsed fields ✅

- **user=admin**
- **src_ip=10.0.0.8**
- **action=failed**

8️⃣ ALERT

Definition:
An alert is a **rule-triggered notification** based on conditions.

Hard truth:
Alerts are **claims**, not facts.

Example:
"Multiple failed logins detected"

That's an alert.
It still needs validation.

9️⃣ INCIDENT

Definition:
An incident is a **confirmed security issue** that requires response.

Key difference:

- Alert → suspicion
- Incident → validated threat

Most alerts should **never** become incidents.

🔟 ALERT → INCIDENT WORKFLOW

How transition actually happens:

1. Alert triggers
2. Analyst reviews context
3. Checks logs, fields, timeline
4. Confirms malicious behavior
5. Escalates as incident
6. Response begins

👉 **SIEM does not create incidents. Analysts do.**

1️⃣1️⃣ RULE / DETECTION RULE

Definition:
A rule is a **logic condition** that looks for suspicious patterns.

Example:

- **If failed_logins > 5**
- **AND time_window = 2 minutes**
- **THEN generate alert**

Bad rules = alert fatigue
Good rules = signal

1️⃣2️⃣ TIME WINDOW

Definition:
The time range in which events are evaluated.

Why it matters:
Same events, different time windows = different conclusions.

Example:

- **5 failures in 5 seconds → suspicious**
- **5 failures in 5 days → normal**

1️⃣3️⃣ SEVERITY

Definition:
Severity indicates **impact + urgency**, not fear.

Typical levels:

- **Low → informational**
- **Medium → suspicious**
- **High → likely malicious**
- **Critical → confirmed threat**

Severity is **prioritization**, not truth.

🔄 SIEM END-TO-END FLOW (REALITY, NOT TOOL UI)

I'll use **one continuous real-world example** throughout, so you will understand fast

An attacker performs a brute-force attack on an SSH server and finally logs in successfully.

1️⃣ Activity Happens (User / System / Attacker)

What this really means
Something **executes in reality**, outside SIEM.

This can be:

- A user typing a password
- A service starting
- An attacker scanning or authenticating
- A system enforcing a rule

👉 **SIEM sees nothing at this stage.**

Example

- **Attacker at IP 203.0.113.10**
- **Tries SSH login to Linux server**
- **Enters wrong password multiple times**
- **Eventually succeeds**

Key truth:
If no activity happens, **no log will ever exist**.

2️⃣ Logs Are Generated

Definition
The system **records the activity** in its local log files.

Logs are:

- Raw
- Unstructured or semi-structured
- System-specific

Example (Linux `/var/log/auth.log`)

```
Failed password for root from 203.0.113.10 port 51122 ssh2
Failed password for root from 203.0.113.10 port 51123 ssh2
Accepted password for root from 203.0.113.10 port 51130 ssh2
```

Important reality check

- Logs are **not security events**
- Logs are **just text**
- Logs can be noisy, inconsistent, and misleading

At this stage, SIEM still understands **nothing**.

3️⃣ Logs Are Collected

Definition
Logs are **transported** from the source system to the SIEM.

How this usually happens

- Agent (Splunk Forwarder, AMA, Beats)
- Syslog
- API
- Cloud-native connectors

Text diagram

```
Linux Server
   |
   | auth.log
   v
Log Forwarder
   |
   v
SIEM Platform
```

Critical failure point

- If logs are not collected:
  - No visibility
  - No detection
  - No investigation

Hard truth:
Most "SIEM issues" are actually **log collection failures**.

4️⃣ Parsing Extracts Fields

Definition
Parsing breaks raw log text into **key=value fields**.

Before parsing (raw text only ❌)

Failed password for root from 203.0.113.10 port 51122 ssh2

After parsing (structured fields ✅)

```
user=root
src_ip=203.0.113.10
action=failed
service=ssh
port=51122
```

Why parsing is foundational

- Searches depend on fields
- Rules depend on fields
- Correlation depends on fields

If parsing fails

- user is missing
- IP is buried in text
- Rules silently fail

👉 **Bad parsing = fake SIEM competence**

5️⃣ Normalization Standardizes Data

Definition
Normalization converts **different log formats** into a **common language**.

Problem
Different systems describe the same action differently.

Examples

- **Linux:** Failed password
- **Windows:** EventID 4625
- **Firewall:** action=deny

After normalization

```
event_type=authentication
action=failure
```

Text diagram

```
Linux log    → action=failure
Windows log → action=failure
Firewall log → action=failure
```

Why this matters

- Correlation across systems
- Single detection logic
- Consistent investigations

Without normalization:

- Rules explode
- Logic becomes unmaintainable
- SOC misses attacks

6️⃣ Events Are Created

Definition
An event is a **clean, structured, normalized record** stored in SIEM.

Think of it as:

"This is what actually happened, expressed clearly."

Example event

```
timestamp=10:02:15
host=linux-prod-01
user=root
src_ip=203.0.113.10
event_type=authentication
action=failure
service=ssh
```

Important distinction

- Logs → raw evidence
- Events → usable evidence

SOC analysts **investigate events**, not logs.

7️⃣ Aggregation Reduces Noise

Definition
Aggregation groups **similar events** into summaries.

Without aggregation ❌

- 300 failed login events
- Analyst overwhelmed
- Signal buried in noise

With aggregation ✅

```
300 authentication failures
from src_ip=203.0.113.10
targeting user=root
within 2 minutes
```

Text diagram

```
300 events
   ↓
1 aggregated record
```

Key insight
Aggregation:

- Reduces alert fatigue
- Improves readability
- Preserves context

It does **not** decide maliciousness.

8️⃣ Correlation Finds Patterns

Definition
Correlation connects **multiple related events over time**.

This is where detection logic starts.

Example correlation

- Multiple failures
- Same IP
- Same user
- Short time window
- Followed by success

Text diagram

```
Fail → Fail → Fail → Success
        |
        └── same IP + user
```

This pattern strongly suggests:

Brute-force followed by compromise

Key rule

- Single event ≠ attack
- Pattern + context = signal

9️⃣ Alerts Are Generated

Definition
An alert is a **rule-based signal** that says:

"This pattern matches something suspicious."

Example alert

Alert Name: SSH Brute Force Success
Severity: High
Reason: Multiple failures followed by success

Hard truth

- Alerts are **hypotheses**
- Alerts can be wrong
- Alerts are not incidents

If you trust alerts blindly, you're not an analyst — you're a notification viewer.

🔟 Analyst Validates

Definition
A human analyst **tests whether the alert is true or false**.

What validation includes

- Check event timeline
- Verify IP reputation
- Confirm user legitimacy
- Compare with baseline behavior

Example questions

- Is this IP internal or external?
- Does this user normally SSH?
- Is this time normal?
- Any other suspicious activity?

This is where **SOC skill actually matters**.

1️⃣1️⃣ Incident Is Created (If Confirmed)

Definition
An incident is a **confirmed security issue requiring response**.

When incident is created

- Evidence supports malicious activity
- Impact is possible or real
- Action is required

Example

- **Root account compromised**
- **External attacker IP**
- **Confirmed brute-force**

Text diagram

```
Alert
  ↓
Validation
  ↓
Incident
  ↓
Containment / Response
```

Final truth

SIEM does NOT create incidents.
Analysts do.

🧠 FINAL MENTAL MODEL (MEMORIZE THIS)

Reality → Logs → Fields → Meaning → Pattern → Signal → Decision

If you don't understand **each transformation**,
you don't understand SIEM — you're just clicking dashboards.

🎯 FINAL TAKEAWAY (LAST PAGE IDEA)

If you don’t understand:

- logs vs events
- parsing vs normalization
- alert vs incident
- correlation vs aggregation

Then **SIEM will lie to you**.

Learn the fundamentals.
Then touch the tool.
