zurück

# Monitoring – Grundlagen (SNMP, Metrics, Logs, Alerts, Dashboards)

Monitoring überwacht Systeme, Netzwerke und Anwendungen, um Fehler frühzeitig zu erkennen, Performance zu analysieren und Ausfälle zu vermeiden.

Moderne Monitoring-Lösungen bestehen aus:

- Metriken (CPU, RAM, Netzwerk)
- Logs (Syslog, Anwendungslogs)
- Events / Alerts
- Dashboards (Grafana)
- Health-Checks
- Status-Überwachung (Up/Down)

Monitoring ist zentraler Bestandteil jeder IT-Infrastruktur.

---

# 1. Was wird überwacht?

### Hardware

- CPU-Auslastung
- RAM
- Festplattenplatz
- Temperatur
- Lüfter
- Netzwerkkarten

### Netzwerk

- Bandbreite (Ingress/Egress)
- Paketverlust
- Switch-Status
- Port-Status (Up/Down)
- Ping-Latenz
- SNMP-Daten

### Dienste

- Webserver (HTTP 200 OK)

- Mailserver (SMTP/IMAP)
- DNS-Server
- Datenbanken (MySQL/MariaDB)
- Container / Docker-Stacks

### Sicherheit

- IDS/IPS Alerts
- Login-Versuche
- Firewall-Events
- CrowdSec-Signale

### Anwendungen

- Fehler in Logs
- Antwortzeiten
- API-Performance
- Benutzerzugriffe

---

# 2. Arten des Monitorings

### a) Host-Monitoring

Überwachung eines einzelnen Systems:

- CPU, RAM, Disk, Netzwerk
- Dienste
- Prozesse

Beispiele:

- Node Exporter (Prometheus)
- Netdata
- Zabbix Agent

### b) Netzwerk-Monitoring

Überwachung über das Netzwerk:

- SNMP-Abfragen
- ICMP (Ping)
- Port-Checks

Beispiele:

- LibreNMS
- PRTG
- Checkmk

### c) Log-Monitoring

Auswertung von Logdateien:

- Syslog
- Docker-Logs
- Suricata eve.json
- Application Logs

Beispiele:

- Loki + Promtail
- Elasticsearch / Logstash / Kibana (ELK)
- Graylog

### d) Anwendungs-Monitoring

- HTTP-Checks
- Response Time
- API-Fehler
- Business-Metriken (z. B. Bestellungen)

---

# 3. SNMP – Simple Network Management Protocol

SNMP nutzt Netzwerkabfragen, um Geräte zu überwachen.

Ports:

- UDP 161 (Abfragen)
- UDP 162 (Traps/Eventmeldungen)

Man erhält Daten wie:

- Port-Status
- Traffic-Counter
- Temperaturen
- Lüfter
- Geräteinformationen

SNMP Versionen:

- **v1/v2c** → Community-Strings (weniger sicher)
- **v3** → voll verschlüsselt (empfohlen)

Beispiel Abfrage:

```
snmpwalk -v2c -c public 192.168.1.1
```

---

# 4. Prometheus & Exporter

Prometheus sammelt **Metriken**, z. B.:

- CPU
- RAM
- Netzwerk
- Containerstatus
- HTTP-Latenzen

Jeder Dienst liefert Daten über einen „Exporter“.

Beispiele:

- Node Exporter (Server)
- Blackbox Exporter (HTTP/TCP Tests)
- SNMP Exporter (Switches)
- MySQL Exporter

Prometheus arbeitet mit einer Pull-Architektur:

```
Prometheus → fragt Exporter regelmäßig ab
```

---

# 5. Grafana – Dashboards

Grafana ist ein Visualisierungstool für Monitoring-Daten.

Man erstellt damit Dashboards für:

- Serverstatus
- Suricata Alerts
- Docker-Statistiken
- Mailserver-Metriken
- Netzwerkverkehr
- CPU/RAM

ASCII:

```
Prometheus / Loki / MySQL →
              Grafana →
            Dashboards
```

---

# 6. Logging – zentrale Logverwaltung

Logs sind essenziell, um Fehler zu finden und sicherheitsrelevante Events zu erkennen.

Quellen:

- Syslog (Switches, Server, Firewalls)
- Docker-Container
- Suricata eve.json
- Auth-Logs (Loginversuche)
- Mailserver-Logs

Moderne Systeme:

- **Loki + Promtail** (leicht, schnell)
- **ELK Stack** (Elasticsearch, Logstash, Kibana)
- **Graylog**

---

# 7. Alerts & Benachrichtigungen

Alerting-Systeme informieren dich sofort bei Problemen.

Typische Trigger:

- CPU > 90%
- Datenträger voll

- Server nicht erreichbar
- viele IDS-Alarme
- Mailqueue wächst
- Container abgestürzt

Benachrichtigung über:

- E-Mail
- Teams/Slack
- Matrix/Element
- Grafana Alerting
- SMS (optional)
- Webhooks

---

# 8. Health Checks

Healthchecks prüfen Dienste automatisch:

- HTTP-Status 200
- Datenbank erreichbar
- DNS antwortet
- SMTP Check
- Container status

Viele Load Balancer nutzen Healthchecks.

---

# 9. Blackbox Monitoring

Der „Blackbox Exporter“ testet externe Dienste:

```
GET https://example.com → 200 OK?
SMTP Port 587 erreichbar?
Ping erfolgreich?
```

Super zur Überwachung:

- Domains
- APIs
- Mailserver

- externe Dienste

---

# 10. Beispiel moderner Monitoring-Stack

ASCII:

```
+------------------------+
|      Prometheus        | ← Metrics
+------------------------+
|      Loki + Promtail   | ← Logs
+------------------------+
|         Grafana        | ← Dashboards + Alerts
+------------------------+
|      SNMP Exporter     |
+------------------------+
```

Dieser Stack läuft perfekt auch in Docker.

---

# 11. Best Practices

- zentrale Loggingstelle einrichten
    - SNMPv3 statt v2c verwenden
      * Alerts nicht zu empfindlich, aber sinnvoll scharf
      * Dashboards für CPU/RAM/Disk/Netz
      * IDS/IPS-Events grafisch auswerten
      * Healthchecks für alle Dienste
      * Monitoring in eigene VLANs trennen

—

# Zusammenfassung

- Monitoring überwacht Systeme, Netzwerke, Sicherheit und Dienste
    - SNMP sammelt Netzwerkdaten
      * Prometheus sammelt Metriken
      * Grafana visualisiert alles
      * Logs → via Loki oder ELK
      * Alerts warnen vor Problemen
      * Healthchecks sichern Verfügbarkeit
      * Monitoring ist Schlüssel für stabile IT-Infrastrukturen