Kibana — projekt alertów

Zakres: Filebeat, Winlogbeat, Metricbeat, Packetbeat, Heartbeat, APM (.NET), RUM (Angular)  
*Data: 2025-08-21 07:57*

# 1) Cel i zakres

* Jak definiować, grupować i wysyłać alerty w Kibanie dla wszystkich źródeł danych.
* Gotowe szablony reguł (KQL/Observability), progi, „Group by”, okna czasowe, treści powiadomień.
* Minimalizowanie szumu: deduplikacja, suppression, okna utrzymania, monitoring „no data”.

# 2) Z czego korzystamy w Kibanie

* Typy reguł (rule types): Elasticsearch query (KQL), Logs/Metric threshold, Uptime (Heartbeat), APM (duration/error rate/anomaly).
* Akcje (connectors): e-mail / Slack / Teams / webhook; dodaj Recovery notification.
* Grupowanie (Group by): tworzy osobne instancje alertu dla service.name / host.name / monitor.id / url.path itd.

# 3) Nazewnictwo, tagi, harmonogramy

* Nazwa: <DOMAIN>\_<WHAT>\_<THRESHOLD>\_<WINDOW> np. APP\_ERRORS\_SPIKE\_5m.
* Tagi: env:prod, team:platform, source:filebeat — ułatwiają filtrowanie alertów.
* Schedule: co 1–5 min (logs/uptime), 5–10 min (metrics). Lookback window ≥ 2× interval.

# 4) Treści powiadomień (szablon)

[{{rule.name}}] status={{context.alertState}} env={{rule.tags}}  
WHEN: {{date}}  
GROUP: {{state.groupBy}} = {{context.group}}  
COUNT: {{state.matches}}  
QUERY: {{rule.params.searchConfiguration.query.query}}  
Link: {{context.viewInAppUrl}}  
Runbook: https://intra/runbooks/{{rule.name}}

W Teams/Slack dodaj emoji (🛑/⚠️/ℹ️) i krótki „co robić” (runbook).

# 5) „Starter pack” reguł — gotowe szablony

## 5.1 Filebeat (logi aplikacyjne + IIS)

* APP\_ERRORS\_SPIKE — KQL: log.level:(error or critical) [+ app.name:"myapp"]; Okno 5 min; Group by: app.name, host.name; Warunek: > 0 lub próg liczby/min.
* HTTP\_5XX\_SPIKE (IIS) — KQL: event.dataset:"iis.access" and http.response.status\_code >= 500; Group by: url.path, host.name; Warunek: ratio 5xx > X%.
* IIS\_SLOW — KQL: event.dataset:"iis.access" and iis.access.time\_taken >= 3000; Group by: url.path, host.name; Warunek: > N w 5 min.
* APP\_LOGS\_SILENCE — Typ: Less than 1 match; KQL: app.name:"myapp"; Okno 10–15 min; Group by: host.name.

## 5.2 Winlogbeat (Security/PowerShell/Sysmon)

* SEC\_BRUTEFORCE\_4625 — KQL: winlog.channel:"Security" and winlog.event\_id:4625; Group by: user.name, source.ip; Warunek: > N/5–10 min.
* SEC\_AUDIT\_CLEARED\_1102 — KQL: winlog.channel:"Security" and winlog.event\_id:1102; Group by: host.name; Warunek: > 0 (natychmiast).
* SEC\_NEW\_SERVICE — KQL: (winlog.event\_id:4697) or (winlog.event\_id:7045 and winlog.provider\_name:"Service Control Manager"); Group by: host.name, ServiceName.
* SEC\_GROUP\_CHANGE — KQL: winlog.event\_id:(4728 or 4732); Group by: TargetUserName, host.name.
* PS\_SCRIPT\_INDICATORS — KQL: winlog.channel:"Microsoft-Windows-PowerShell/Operational" and winlog.event\_id:4104 and message:("Invoke-WebRequest" or "FromBase64String" or "IEX"); Group by: host.name.
* HOST\_SECURITY\_SILENCE — Less than 1 match dla winlog.channel:"Security" w 10–15 min (per host.name).

## 5.3 Metricbeat (hosty, usługi Windows)

* CPU\_HIGH\_90 — KQL: event.dataset:"system.cpu" and system.cpu.total.norm.pct >= 0.9; Group by: host.name; Okno 5 min.
* MEM\_HIGH\_90 — KQL: event.dataset:"system.memory" and system.memory.actual.used.pct >= 0.9; Group by: host.name.
* DISK\_FULL\_90 — KQL: event.dataset:"system.filesystem" and system.filesystem.used.pct >= 0.9; Group by: host.name, mount\_point.
* WIN\_SERVICE\_STOPPED — KQL: event.dataset:"windows.service" and windows.service.state:"stopped"; Group by: host.name, windows.service.name.
* HOST\_SILENCE — Less than 1 match; KQL: host.name:\* w 5–10 min (per host).

## 5.4 Packetbeat (HTTP/DNS/TLS/flows)

* HTTP\_5XX\_SPIKE — KQL: event.dataset:"http" and http.response.status\_code >= 500; Group by: url.path, url.domain.
* HTTP\_LATENCY\_HIGH — KQL: event.dataset:"http" and event.duration >= 2000000 (≥ 2 s, μs); Group by: url.path.
* DNS\_NXDOMAIN\_SURGE — KQL: event.dataset:"dns" and dns.response\_code:"NXDOMAIN"; Group by: dns.question.name.
* TLS\_HYGIENE — KQL: event.dataset:"tls" and tls.version:("1.0" or "1.1"); Expiry: event.dataset:"tls" and tls.server.x509.not\_after <= now+30d; Group by: CN/domain.
* HEAVY\_FLOWS — KQL: event.category:"network" and event.type:"connection" and network.bytes >= 100000000; Group by: source.ip / destination.ip.
* PACKETBEAT\_SILENCE — Less than 1 match dla event.category:"network" w 10–15 min.

## 5.5 Heartbeat (Uptime)

* UPTIME\_DOWN — Rule type: Monitor status; monitor.status:"down"; Group by: monitor.id/monitor.name; Okno 1–5 min.
* UPTIME\_SLOW — KQL: monitor.type:"http" and monitor.duration.us >= 1000000; Group by: monitor.name/url.full.
* TLS\_EXPIRY\_30D — Rule type: TLS; warunek: do wygaśnięcia ≤ 30 dni; harmonogram 1×/dzień.
* UPTIME\_AVAILABILITY\_SLO — availability = 100\*UP/(UP+DOWN) (Lens + Threshold) lub „More than N DOWN” w 7/30d.
* UPTIME\_CONSECUTIVE\_FAILURES — „More than N matches” w 5–10 min; Group by: monitor.name, observer.geo.name.

## 5.6 APM (.NET Core — paczka w kodzie)

* APM\_LATENCY\_P95\_THRESHOLD — Rule: APM Transaction duration; service.name:"my-api-dotnet"; p95 > 1000 ms / 5 min; Group by: transaction.name.
* APM\_ERROR\_RATE — Rule: APM Error rate; (errors/transactions) > 5% / 5–10 min; Group by: service.name, transaction.name.
* APM\_DB\_SPAN\_SLOW — KQL: processor.event:"span" and span.type:"db" and service.name:"my-api-dotnet" and span.duration.us >= 500000; Group by: span.subtype, destination.service.name.
* APM\_NO\_DATA — Less than 1 match: processor.event:"transaction" and service.name:"my-api-dotnet" w 5–10 min.

## 5.7 RUM JavaScript (Angular)

* RUM\_LATENCY\_THRESHOLD — KQL: processor.event:"transaction" and transaction.type:("page-load" or "route-change") and service.name:"my-spa" and transaction.duration.us >= 3000000; Group by: transaction.name.
* RUM\_JS\_ERRORS\_SPIKE — KQL: processor.event:"error" and service.name:"my-spa"; Group by: url.path/transaction.name; Warunek: > N / 5–10 min.
* RUM\_XHR\_SLOW — KQL: processor.event:"span" and span.type:"external" and span.subtype:"http" and span.duration.us >= 2000000 and service.name:"my-spa"; Group by: destination.service.resource/url.domain.
* RUM\_LCP\_THRESHOLD — KQL: experience.lcp >= 4000 (jeśli dostępne); Group by: url.path.
* RUM\_NO\_DATA — Less than 1 match: service.name:"my-spa" and processor.event:"transaction" w 10–15 min.

# 6) Anti-noise i jakość alertów

* Group by po kluczach domenowych (service.name, host.name, monitor.id, url.path).
* Alert suppression / throttle — ogranicz ponowne wysyłki (np. co 30–60 min).
* Recovery notification — informuj o powrocie do normy.
* Maintenance windows — nie wysyłaj alertów w trakcie releasu.
* Runbook link — w treści akcji dodaj stały link „co robić”.

# 7) Operacje: przegląd i obserwowalność samych alertów

* Alert as Data — dashboard stanu reguł (czas wykonania, liczba instancji, błędy akcji).
* Health check łańcucha notyfikacji — syntetyczny alert dzienny, który wysyła ping na kanały.

# 8) Procedura tworzenia (UI)

* Stack Management → Rules → Create rule → wybierz Rule type (KQL/APM/Uptime/Logs/Metric).
* Skonfiguruj KQL/wskaźnik, Schedule, Lookback window, Group by, Threshold.
* Dodaj Actions (connector, message — szablon), Recovery, Tags, Throttle/Notify when.
* Zapisz i przetestuj (sprowokuj zdarzenie w bezpieczny sposób).

# 9) Checklist wdrożenia

* Connectory (e-mail/Slack/Teams/webhook) i uprawnienia.
* Spójne nazwy + tagi (env, team, source).
* Group by ustawione (unikaj jednego mega-alertu).
* Recovery + throttle + suppression.
* Okna utrzymania + „no data”.
* Runbooki podlinkowane.
* Dashboard „Alerts overview” (Alert as Data) z filtrami po source:\*.

# 10) Co dalej?

Mogę przygotować pakiet eksportu (Saved Objects) z kompletem powyższych reguł (wraz z przykładowymi akcjami i dashboardem „Alerts overview”) pod Twoje nazwy (service.name, app.name, monitor.id) i progi (prod/stage).