**gotowy elastic-agent.yml dla Windows Server (offline, standalone)**, który zastępuje:

* **Winlogbeat** → dzienniki Windows,
* **Heartbeat** → Uptime (HTTP + ICMP),
* **Metricbeat** → metryki hosta,
* **Packetbeat** → przechwytywanie ruchu (flows + DNS).

Uwaga: to jest tryb **standalone** (bez Fleet/EPR). Nie uruchamiaj równocześnie Agenta zarządzanego przez Fleet i tego standalone na tej samej maszynie.

**elastic-agent.yml (wklej, podmień host/loginy/CA)**

# === Połączenie z Elasticsearch ===

outputs:

default:

type: elasticsearch

hosts: ["https://<ES\_HOST>:9200"]

username: "<USER>"

password: "<PASS>"

ssl:

verification\_mode: certificate

# Jeśli masz własne CA, odkomentuj:

# certificate\_authorities: ["C:/certs/rootCA.crt"]

agent:

monitoring:

enabled: true

use\_output: default

# === Inputs: Winlogbeat / Heartbeat / Metricbeat / Packetbeat ===

inputs:

# --- Winlogbeat (Windows Event Logs) ---

- id: winlogs

type: winlog

use\_output: default

data\_stream:

type: logs

streams:

- data\_stream: { dataset: windows.eventlog }

event\_logs:

- name: Application

- name: System

- name: Security

# (opcjonalnie) Sysmon:

# - data\_stream: { dataset: windows.sysmon\_operational }

# event\_logs: [{ name: Microsoft-Windows-Sysmon/Operational }]

# (opcjonalnie) PowerShell Operational:

# - data\_stream: { dataset: windows.powershell\_operational }

# event\_logs: [{ name: Microsoft-Windows-PowerShell/Operational }]

# --- Heartbeat (Uptime) — HTTP monitor ---

- id: uptime-http

type: synthetics/http

use\_output: default

data\_stream:

type: synthetics

streams:

- data\_stream: { dataset: http }

urls: ["http://onet.pl"] # dodaj kolejne URL-e w tej liście, jeśli chcesz

schedule: "@every 1m"

timeout: "16s"

max\_redirects: 10

check:

request: { method: GET }

response: { status: [200, 301, 302, 308] }

# --- Heartbeat (Uptime) — ICMP (ping) ---

- id: uptime-icmp

type: synthetics/icmp

use\_output: default

data\_stream:

type: synthetics

streams:

- data\_stream: { dataset: icmp }

hosts: ["onet.pl"]

schedule: "@every 1m"

timeout: "5s"

# --- Metricbeat (host metrics) ---

- id: system-metrics

type: system/metrics

use\_output: default

data\_stream:

type: metrics

streams:

- data\_stream: { dataset: system.cpu }

period: 1m

cpu.metrics: ["percentages","normalized\_percentages"]

- data\_stream: { dataset: system.memory }

period: 1m

- data\_stream: { dataset: system.network }

period: 1m

- data\_stream: { dataset: system.diskio }

period: 1m

- data\_stream: { dataset: system.filesystem }

period: 5m

# --- Packetbeat (network packet capture) ---

# WYMAGA zainstalowanego Npcap (https://nmap.org/npcap/) i uprawnień administratora.

# Jeśli nie masz Npcap, ZOSTAW ten blok zakomentowany na start.

- id: packet-flows

type: packet

use\_output: default

data\_stream:

type: logs

streams:

# Przepływy (flows)

- data\_stream: { dataset: network\_traffic.flow }

# Interfejs przechwytywania:

# "any" = wszystkie; zamiast "any" możesz podać indeks (np. "0") lub nazwę urządzenia Npcap.

interfaces: ["any"]

period: 30s

timeout: 30s

# (opcjonalnie) BPF, aby nie łapać ruchu do ES/Kibany/Fleet:

# bpf\_filter: "not host <ES\_HOST> and not port 8220"

# DNS (metadane zapytań)

- data\_stream: { dataset: network\_traffic.dns }

interfaces: ["any"]

ports: [53]

include\_authorities: true

include\_additionals: true

**Instalacja na Windows (PowerShell jako Administrator)**

# 1) Jeśli masz uruchomionego Agenta z Fleet – zatrzymaj:

# Stop-Service Elastic-Agent

# 2) Test w konsoli (zobaczysz log startu i ewentualne błędy configu):

.\elastic-agent.exe run -c .\elastic-agent.yml

# 3) Instalacja jako usługa:

.\elastic-agent.exe install -c .\elastic-agent.yml

Start-Service Elastic-Agent

**ICMP (ping) na Windows** wymaga, by usługa działała z uprawnieniami admina (LocalSystem jest OK).  
**Packet capture** wymaga **Npcap** (z opcją WinPcap Compatible Mode) i często restartu usługi po instalacji Npcap.  
Jeśli Agent nie startuje przez sekcję packet, skomentuj ją, uruchom usługę, doinstaluj Npcap i odkomentuj ponownie.

**Szybka weryfikacja w Kibanie (klikami)**

* **Discover → KQL**
  + Windows logs (Winlogbeat):
  + \_index: ("logs-windows.\*" OR ".ds-logs-windows.\*")
  + Uptime (Heartbeat):
  + \_index: (".ds-synthetics-\*" OR "synthetics-http-\*" OR "synthetics-icmp-\*")
  + Host metrics (Metricbeat):
  + \_index: ("metrics-system.\*" OR ".ds-metrics-system.\*")
  + Packet capture (flows/DNS):
  + \_index: ("logs-network\_traffic.\*" OR ".ds-logs-network\_traffic.\*")
* **Observability → Uptime**: powinieneś zobaczyć monitor(y) HTTP/ICMP.
* **Observability → Metrics → Hosts**: host pojawi się z metrykami CPU/RAM/Disk/Net.

**Najczęstsze pułapki (i szybkie fixy)**

* **401/403 do ES** → sprawdź username/password.
* **Błędy TLS** → podaj plik CA w certificate\_authorities lub na POC zostaw verification\_mode: certificate.
* **Brak danych Uptime** → serwer nie ma DNS/HTTP do celu; sprawdź curl -I http://onet.pl.
* **Packet input** nie startuje → brak Npcap / zły interfejs; zamień interfaces: ["any"] na konkretny (np. "0"), uruchom usługę ponownie.

## Kroki (Windows Server)

1. **Pobierz ZIP** tej samej wersji agenta (8.13.2) i rozpakuj do: C:\Elastic\agent-standalone\ (inny katalog niż Fleet Server).
2. Utwórz C:\Elastic\agent-standalone\elastic-agent.yml (podmień <ES\_HOST>, dane logowania, ewent. CA):

outputs:

default:

type: elasticsearch

hosts: ["https://<ES\_HOST>:9200"]

username: "<USER>"

password: "<PASS>"

ssl:

verification\_mode: certificate

# certificate\_authorities: ["C:/certs/rootCA.crt"]

agent.monitoring.enabled: true

inputs:

# Winlogbeat-equiv

- id: winlogs

type: winlog

use\_output: default

data\_stream: { type: logs }

streams:

- data\_stream: { dataset: windows.eventlog }

event\_logs: [{name: Application}, {name: System}, {name: Security}]

# (opcjonalnie) Sysmon / PowerShell patrz wcześniejsze wiadomości

# Metricbeat-equiv (host)

- id: system-metrics

type: system/metrics

use\_output: default

data\_stream: { type: metrics }

streams:

- data\_stream: { dataset: system.cpu } , period: 1m

- data\_stream: { dataset: system.memory } , period: 1m

- data\_stream: { dataset: system.network } , period: 1m

- data\_stream: { dataset: system.diskio } , period: 1m

- data\_stream: { dataset: system.filesystem }, period: 5m

# Heartbeat-equiv (HTTP)

- id: uptime-http

type: synthetics/http

use\_output: default

data\_stream: { type: synthetics }

streams:

- data\_stream: { dataset: http }

urls: ["http://onet.pl"]

schedule: "@every 1m"

timeout: "16s"

max\_redirects: 10

check:

request: { method: GET }

response: { status: [200,301,302,308] }

# Heartbeat-equiv (ICMP)

- id: uptime-icmp

type: synthetics/icmp

use\_output: default

data\_stream: { type: synthetics }

streams:

- data\_stream: { dataset: icmp }

hosts: ["onet.pl"]

schedule: "@every 1m"

timeout: "5s"

# Packetbeat-equiv (flows + DNS) — wymaga zainstalowanego Npcap

- id: packet-flows

type: packet

use\_output: default

data\_stream: { type: logs }

streams:

- data\_stream: { dataset: network\_traffic.flow }

interfaces: ["any"]

period: 30s

timeout: 30s

- data\_stream: { dataset: network\_traffic.dns }

interfaces: ["any"]

ports: [53]

include\_authorities: true

include\_additionals: true

1. **Test w konsoli** (PowerShell jako Administrator, w C:\Elastic\agent-standalone\):

.\elastic-agent.exe run -c .\elastic-agent.yml

Jeśli startuje bez błędów → przerwij Ctrl+C.

1. **Autostart bez instalowania drugiej usługi** (Harmonogram zadań):

* Otwórz **Task Scheduler** → **Create Task…**
  + **Run whether user is logged on or not**, **Run with highest privileges**.
  + **Trigger:** At startup.
  + **Action:**
    - Program: C:\Elastic\agent-standalone\elastic-agent.exe
    - Arguments: run -c C:\Elastic\agent-standalone\elastic-agent.yml
  + **Conditions:** odznacz „Start the task only if the computer is on AC power” (jeśli dotyczy).
  + Zapisz (podaj hasło, aby mógł startować jako SYSTEM/administrator).

**ICMP** wymaga kontekstu z uprawnieniami admina (LocalSystem działa).  
**Packet capture**: zainstaluj **Npcap** (WinPcap Compatible Mode), zrestartuj zadanie; jeśli nie startuje, tymczasowo zakomentuj blok packet.

**Jak sprawdzić (klikami w Kibanie)**

* **Uptime → Monitors**: powinien być UP/DOWN dla http://onet.pl.
* **Discover (KQL):**
  + Windows logs: \_index: ("logs-windows.\*" OR ".ds-logs-windows.\*")
  + Metrics: \_index: ("metrics-system.\*" OR ".ds-metrics-system.\*")
  + Synthetics: \_index: (".ds-synthetics-\*" OR "synthetics-http-\*" OR "synthetics-icmp-\*")
  + Packet: \_index: ("logs-network\_traffic.\*" OR ".ds-logs-network\_traffic.\*")
  + (dodaj AND host.name:"<NAZWA\_TEGO\_HOSTA>")