Agent ochrony na serwerach Linux: instalacja i weryfikacja

Poziom: średnio

Dotyczy: agenci EDR na Rocky Linux 9, na przykładzie Microsoft Defender for Endpoint na Linuksie

Czas: ~30 min

Serwer linuksowy dostaje agenta ochrony stacji dokładnie tak samo jak stacja robocza, tylko innym pakietem i innym poleceniem weryfikacji. Instalujemy repozytorium producenta, sam pakiet, uruchamiamy skrypt onboardingu i sprawdzamy stan agenta, zanim uznamy wdrożenie za zakończone.

Skróty użyte na tej stronie (6)

Skrót	Co znaczy
CLI	Command Line Interface: praca przez wpisywanie komend w terminalu
DNS	Domain Name System: książka telefoniczna internetu, zamienia nazwy na adresy IP
eBPF	extended Berkeley Packet Filter: mechanizm jądra Linux pozwalający programowi obserwować zdarzenia systemowe (pliki, procesy, sieć) bez ładowania własnego modułu jądra
EDR	Endpoint Detection and Response: ochrona stacji, która obserwuje zachowanie programów, nie tylko sygnatury
EICAR	European Institute for Computer Antivirus Research: instytut, od którego nazwy pochodzi nazwa bezpiecznego pliku testowego rozpoznawanego przez każdy silnik antywirusowy jako zagrożenie testowe
WMS	Warehouse Management System: system zarządzania magazynem

Fiszka

Komendy wykonujemy w CLI bash jako użytkownik z uprawnieniami sudo albo root, na serwerze z rozwiązywaniem nazw DNS i ruchem wychodzącym do internetu.

ZMIENIA SYSTEM

Dodanie repozytorium pakietów Microsoftu na Rocky Linux 9

```
sudo dnf config-manager --add-repo
https://packages.microsoft.com/config/rocky/9/prod.repo
```

Rocky Linux ma własną ścieżkę repozytorium, inną niż ogólna ścieżka `rhel`, mimo że pakiety są identyczne. Kanał `prod` to wydania produkcyjne, w odróżnieniu od kanałów `insiders-fast` i `insiders-slow` przeznaczonych do wcześniejszego testowania nowych wersji.

dnf = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Zaimportowanie klucza GPG producenta

```
sudo rpm --import https://packages.microsoft.com/keys/microsoft.asc
```

Bez zaimportowanego klucza `dnf` odrzuci instalację pakietu jako niepodpisanego.

GPG = GNU Privacy Guard: program do podpisywania i szyfrowania plików oraz sprawdzania podpisów paczek · **rpm** = RPM Package Manager: format i narzędzie paczek RHEL

ZMIENIA SYSTEM

Instalacja pakietu agenta: dnf install mdatp

```
sudo dnf install mdatp
```

Pakiet instaluje binaria agenta w `/opt/microsoft/mdatp/`, usługę systemd `mdatp` oraz narzędzie wiersza poleceń `mdatp`, ale sam agent zostaje nieaktywny do czasu onboardingu.

dnf = Dandified YUM: menedżer pakietów RHEL/Rocky

TYLKO ODCZYT

Uruchomienie skryptu onboardingu

```
unzip WindowsDefenderATPOnboardingPackage.zip
sudo python3 MicrosoftDefenderATPOnboardingLinuxServer.py
```

Plik `WindowsDefenderATPOnboardingPackage.zip` pobieramy z portalu Microsoft Defender, z sekcji Ustawienia, Punkty końcowe, Onboarding, wybierając system Linux Server. Skrypt zapisuje w konfiguracji agenta identyfikator naszej organizacji, dzięki czemu agent wie, do którego portalu wysyłać dane.

python3 = uruchom interpreter Pythona w wersji 3 albo wskazany skrypt

TYLKO ODCZYT

Sprawdzenie usługi agenta: systemctl status mdatp

```
sudo systemctl status mdatp
```

Przykład:

```
● mdatp.service - Microsoft Defender for Endpoint
   Loaded: loaded (/lib/systemd/system/mdatp.service; enabled)
   Active: active (running) since Mon 2026-09-28 06:14:02 CEST
 Main PID: 21044 (wdavdaemon)
   CGroup: /system.slice/mdatp.service
           └─21044 /opt/microsoft/mdatp/sbin/wdavdaemon
```

systemctl = system control: sterowanie usługami systemd

TYLKO ODCZYT

Sprawdzenie zdrowia agenta: mdatp health

```
sudo mdatp health
```

Przykład (skrócony) z `ora01`:

```
healthy: true
health_issues: []
licensed: true
org_id: 8f1c2e4a-91b3-4d7c-a2e5-3c0b1f9a6d21
real_time_protection_enabled: true
```

```
definitions_status: up_to_date
definitions_updated: 2026-09-28 05:02:11
engine_version: 1.1.24080.2
```

Pole `healthy` na `true` jest pierwszym, najogólniejszym potwierdzeniem, że agent działa i komunikuje się z portalem, a dopiero potem sprawdzamy szczegóły.

mdatp = CLI: Microsoft Defender Advanced Threat Protection, narzędzie wiersza poleceń agenta Microsoft Defender for Endpoint na Linuksie: sprawdza stan agenta, zarządza wykluczeniami i uruchamia diagnostykę

TYLKO ODCZYT

Sprawdzenie pojedynczego pola stanu: mdatp health --field

```
mdatp health --field real_time_protection_enabled
```

Przydatne w skrypcie, który sprawdza flotę serwerów automatem: jedno pole zamiast parsowania całego wyniku.

mdatp = CLI: Microsoft Defender Advanced Threat Protection, narzędzie wiersza poleceń agenta Microsoft Defender for Endpoint na Linuksie: sprawdza stan agenta, zarządza wykluczeniami i uruchamia diagnostykę

TYLKO ODCZYT

Dodanie wykluczenia katalogu: mdatp exclusion folder add

```
sudo mdatp exclusion folder add --path "SciezkaKatalogu"
```

> (gdzie `SciezkaKatalogu` to katalog danych silnika bazy albo katalog kontenerów, bez cudzysłowu, jeśli ścieżka nie zawiera spacji)

Przykład z `docker01`:

```
sudo mdatp exclusion folder add --path /docker/
```

Domyślny zasięg wykluczenia to `epp`, czyli ochrona w czasie rzeczywistym, skanowanie na żądanie i monitorowanie zachowania procesu naraz. Wykluczenie w zasięgu `global` (od wersji agenta 101.24092.0001) dodajemy flagą `--scope global`, gdy ma dotyczyć też innych

modułów.

mdatp = CLI: Microsoft Defender Advanced Threat Protection, narzędzie wiersza poleceń agenta Microsoft Defender for Endpoint na Linuksie: sprawdza stan agenta, zarządza wykluczeniami i uruchamia diagnostykę

TYLKO ODCZYT

Sprawdzenie listy wykluczeń: mdatp exclusion list

```
mdatp exclusion list
```

Sprawdzamy tym poleceniem, czy wykluczenie faktycznie zostało zapisane, zanim uznamy problem wydajnościowy za rozwiązany.

mdatp = CLI: Microsoft Defender Advanced Threat Protection, narzędzie wiersza poleceń agenta Microsoft Defender for Endpoint na Linuksie: sprawdza stan agenta, zarządza wykluczeniami i uruchamia diagnostykę

TYLKO ODCZYT

Wskazanie procesów generujących najwięcej skanowań: mdatp diagnostic real-time-protection-statistics

```
sudo mdatp diagnostic real-time-protection-statistics --sort --top 5
```

Wypisuje procesy, których pliki agent skanował najczęściej w danym oknie czasu, posortowane malejąco. Wynik tego polecenia jest podstawą do decyzji, co dokładnie wykluczyć, zamiast zgadywania.

mdatp = CLI: Microsoft Defender Advanced Threat Protection, narzędzie wiersza poleceń agenta Microsoft Defender for Endpoint na Linuksie: sprawdza stan agenta, zarządza wykluczeniami i uruchamia diagnostykę

TYLKO ODCZYT

Pobranie bezpiecznego pliku testowego EICAR

```
curl -o /tmp/eicar.com.txt https://secure.eicar.org/eicar.com.txt
```

> **Pojęcie: plik EICAR.** Tekstowy plik testowy ustandaryzowany przez European Institute for Computer Antivirus Research, rozpoznawany przez praktycznie każdy silnik antywirusowy jako zagrożenie testowe, mimo że nie zawiera żadnego faktycznego złośliwego kodu. Służy wyłącznie do sprawdzenia, czy ochrona w czasie rzeczywistym w ogóle reaguje, bez ryzyka uruchomienia prawdziwego złośliwego oprogramowania na produkcyjnym serwerze.

EICAR = European Institute for Computer Antivirus Research: instytut, od którego nazwy pochodzi nazwa bezpiecznego pliku testowego rozpoznawanego przez każdy silnik antywirusowy jako zagrożenie testowe · **curl** = client URL: pobieranie i wysyłanie przez HTTP z terminala

TYLKO ODCZYT

Sprawdzenie wykrytych zagrożeń: mdatp threat list

```
mdatp threat list
```

Po pobraniu pliku EICAR powinniśmy zobaczyć tu jeden wpis z wykryciem, potwierdzający, że agent faktycznie skanuje pliki na dysku w czasie rzeczywistym, a nie tylko melduje się jako zainstalowany.

EICAR = European Institute for Computer Antivirus Research: instytut, od którego nazwy pochodzi nazwa bezpiecznego pliku testowego rozpoznawanego przez każdy silnik antywirusowy jako zagrożenie testowe · **mdatp** = CLI: Microsoft Defender Advanced Threat Protection, narzędzie wiersza poleceń agenta Microsoft Defender for Endpoint na Linuksie: sprawdza stan agenta, zarządza wykluczeniami i uruchamia diagnostykę

Z życia wzięte

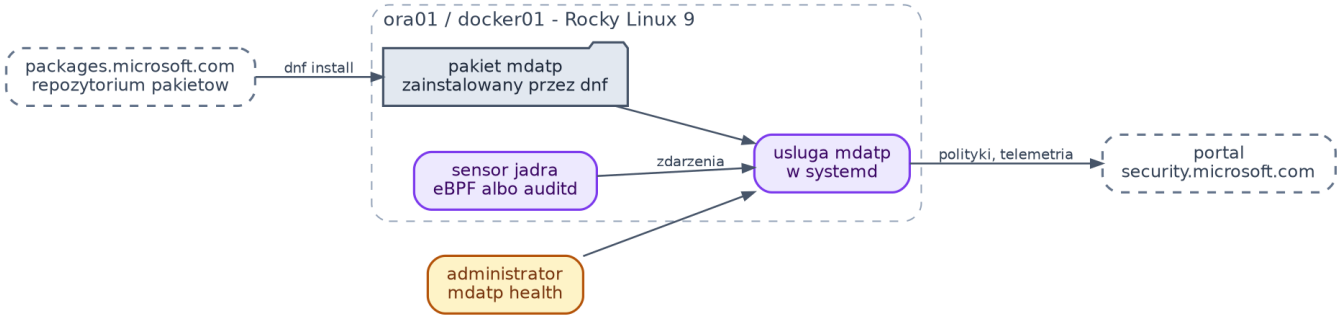
Z ŻYCIA WZIĘTE

Poniedziałek, 5:58 w Swarzędzu. Tomek dzień wcześniej zainstalował Microsoft Defender for Endpoint na `ora01` według checklisty, ale pominął krok z wykluczeniami katalogu danych Oracle. O 6:00 loguje się naraz czterdzieści skanerów magazynowych, WMS zaczyna odpowiadać z kilkusekundowym opóźnieniem na każdy skan kodu kreskowego, a Marek dzwoni z pytaniem, czy coś się zepsuło w nocy. `mdatp diagnostic real-time-protection-statistics --sort --top 5` pokazuje proces silnika Oracle na szczycie listy, a dodanie wykluczenia katalogu danych bazy rozwiązuje problem przed kolejną zmianą.

Jak to działa

Rozwiń: jak to działa pod spodem

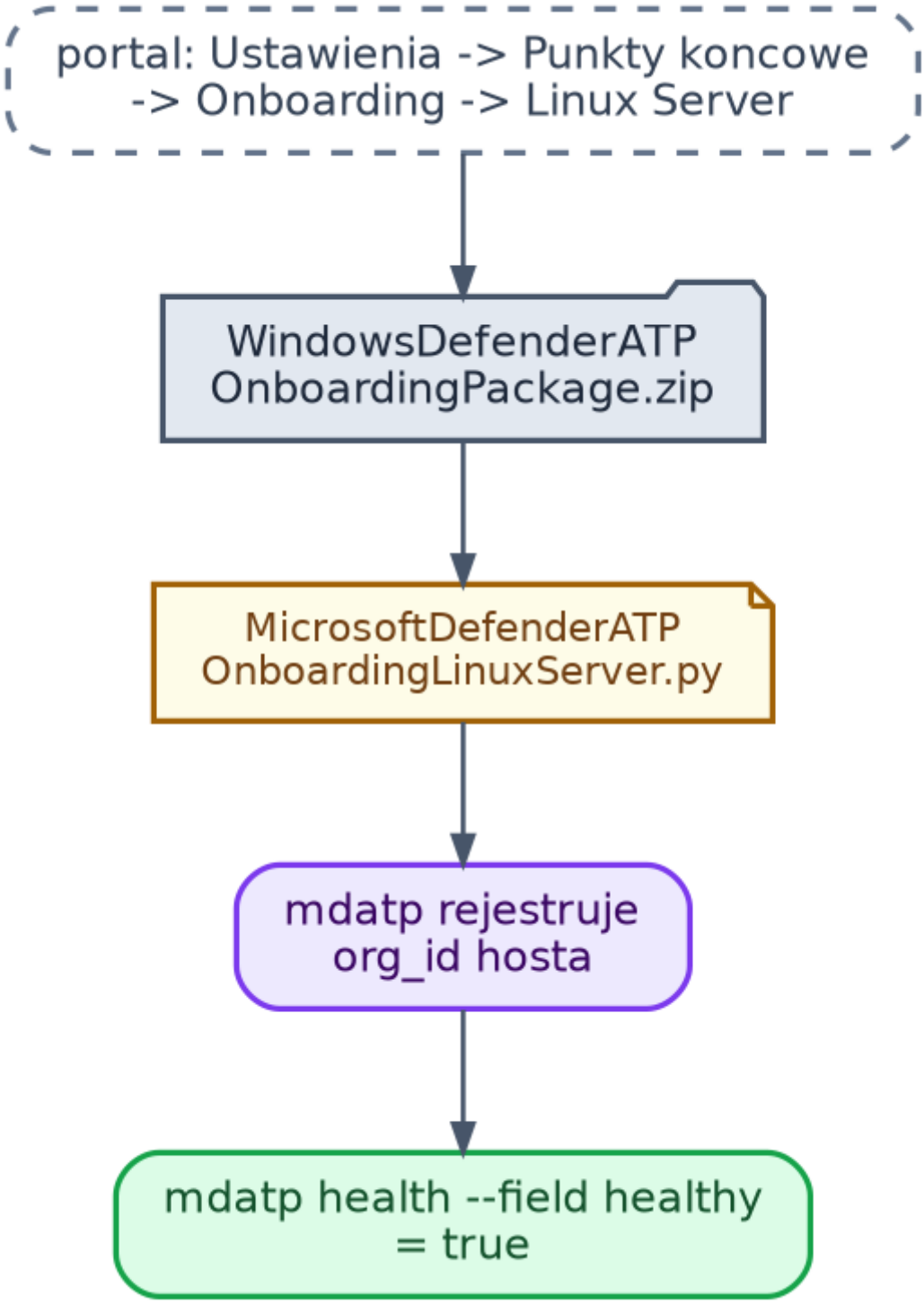
> **Pojęcie: agent EDR na Linuksie.** Tak samo jak na Windowsie, agent to proces działający lokalnie, tu jako usługa systemd, który skanuje pliki, obserwuje zachowanie procesów i wysyła zdarzenia do konsoli w chmurze. Różnica leży w mechanizmie, którym agent w ogóle widzi, co się dzieje w systemie plików i wśród procesów.



Architektura agenta Microsoft Defender for Endpoint na Rocky Linux 9: repozytorium, pakiet, sensor jądra i portal (kliknięcie otwiera pełny rozmiar)

> **Pojęcie: sensor jądra (eBPF).** Mechanizm jądra Linux (extended Berkeley Packet Filter) pozwalający agentowi obserwować operacje na plikach i procesach bez modyfikowania samego jądra ani ładowania własnego modułu. Na starszych jądrach, gdzie eBPF nie jest dostępny albo nie jest wspierany, agent korzysta zamiast tego z podsystemu `auditd`, wolniejszego i bardziej inwazyjnego w konfigurację systemu.

Instalacja przebiega w dwóch niezależnych krokach, które łatwo pomylić: sama instalacja pakietu `mdatp` z repozytorium tylko kładzie binaria na dysku, a dopiero uruchomienie skryptu onboardingu wiąże agenta z konkretną organizacją w portalu.



Przebieg onboardingu agenta na Linuksie: od pobrania pakietu z portalu do potwierdzenia mdatp health (kliknięcie otwiera pełny rozmiar)

Wykluczenia dla katalogów baz danych i katalogu /docker

Zasada, dlaczego w ogóle wykluczamy katalogi danych silnika bazy i katalog `/docker`, jest ogólna dla każdego produktu ochrony stacji i opisana na stronie Antywirus, EDR, XDR, konsola zarządzania: cztery pojęcia. Dla Microsoft Defender for Endpoint na Linuksie dochodzi jeden dodatkowy powód: sensor eBPF przechwytuje zdarzenia na poziomie jądra dla każdej operacji na pliku, więc intensywny ruch zapisu i odczytu w katalogach danych PostgreSQL, Oracle czy MariaDB generuje wielokrotnie więcej zdarzeń do przeanalizowania niż zwykły katalog użytkownika.

Katalog `/docker` na `docker01` wykluczamy z podobnego powodu: demon Dockera rozpakowuje i kasuje warstwy obrazów, a kontenery piszą dane aplikacji intensywnie i często tymczasowo. Ryzyko tego konkretnego wykluczenia ograniczamy tym, że obrazy w Zielonej pochodzą z kontrolowanych źródeł (Gitea Actions albo oficjalne rejestry z przypiętą wersją), a nie z przypadkowych, niezweryfikowanych źródeł w internecie.

> ⚠ Wykluczenie katalogu nie jest decyzją jednorazową. Za każdym razem, gdy na serwerze pojawia się nowa rola albo nowa aplikacja, sprawdzamy `mdatp diagnostic real-time-protection-statistics`, zanim uznamy, że wydajność jest w porządku.

Wpływ na wydajność

Ochrona w czasie rzeczywistym przechwytuje każdą operację odczytu i zapisu pliku, zanim ta dojdzie do faktycznego wykonania, więc obciążenie procesora rośnie proporcjonalnie do intensywności operacji dyskowych na serwerze. Na serwerach aplikacyjnych i bazodanowych o dużym ruchu I/O ten narzut bywa zauważalny bez odpowiednich wykluczeń, a na serwerach o niskim ruchu (np. `ipa01` albo `mon01`) praktycznie niezauważalny. Diagnozę zawsze zaczynamy od `mdatp diagnostic real-time-protection-statistics`, nie od zgadywania, który katalog wykluczyć.

Częste błędy

Objaw	Przyczyna	Co robimy

`dnf install mdatp` kończy się błędem podpisu pakietu	brak zaimportowanego klucza GPG Microsoftu	uruchamiamy `rpm --import` na kluczu przed ponowną instalacją
`mdatp health` pokazuje `healthy: false` zaraz po instalacji	skrypt onboardingu nie został jeszcze uruchomiony	uruchamiamy skrypt pobrany z portalu i sprawdzamy `mdatp health --field org_id`
Wysokie obciążenie procesora na serwerze bazodanowym po wdrożeniu agenta	brak wykluczenia katalogu danych silnika bazy	sprawdzamy `mdatp diagnostic real-time-protection-statistics --sort --top 5` i dodajemy wykluczenie wskazanego katalogu
Agent zainstalowany, ale nie widać zdarzeń plikowych	starsze jądro bez wsparcia dla eBPF, przełączenie na `auditd` nie zaszło poprawnie	sprawdzamy wspierane wersje jądra w dokumentacji producenta i status sensora w `mdatp health --details edr`
Wykluczenie dodane komendą znika po jakimś czasie	wykluczenie ustawione lokalnie zostało nadpisane przez politykę zarządzaną centralnie z portalu	trwałe wykluczenie dodajemy w polityce w portalu, komendę lokalną traktujemy jako test przed wprowadzeniem zmiany

□□ Ćwiczenia

1. Dlaczego wykluczamy katalog `/docker` na `docker01`, mimo że to jeden z najważniejszych katalogów w firmie, trzymający Giteę, NetBox i Grayloga naraz?

Odpowiedź

Demon Dockera intensywnie zapisuje i kasuje warstwy obrazów oraz dane aplikacji, co przy pełnym skanowaniu w czasie rzeczywistym generuje nieproporcjonalnie duże obciążenie w stosunku do realnego ryzyka: same pliki danych kontenerów nie są typowym wektorem infekcji, w odróżnieniu od np. katalogu pobranych załączników pocztowych. Ryzyko tego wykluczenia ograniczamy tym, że obrazy pochodzą z kontrolowanych źródeł, a nie z przypadkowych rejestrów w internecie, tak jak opisano na stronie Antywirus, EDR, XDR, konsola zarządzania: cztery pojęcia.

1. `mdatp health` pokazuje `healthy: true` na serwerze, ale w portalu od kilku dni nie widać żadnych nowych zdarzeń z tego hosta. Co sprawdzamy?

Odpowiedź

Skoro agent lokalnie zgłasza się jako zdrowy, sprawdzamy łączność sieciową serwera do adresów chmury Microsoft Defender (firewall, DNS), pole `definitions_updated` w `mdatp health` pod kątem tego, czy aktualizacje w ogóle dochodzą, oraz czy identyfikator organizacji `org_id` zgadza się z oczekiwanym: `healthy: true` potwierdza tylko, że sam proces agenta działa

poprawnie lokalnie, nie że dane faktycznie docierają do portalu.

Powiązane: Antywirus, EDR, XDR, konsola zarządzania: cztery pojęcia · ePolicy Orchestrator: konsola zarządzania · Wdrożenie ochrony na stacje Windows: GPO, Intune albo RMM · Utrzymanie i typowe problemy

Źródła: Microsoft Learn, „Deploy Microsoft Defender for Endpoint on Linux manually" · Microsoft Learn, „Linux exclusions for Microsoft Defender for Endpoint" · Microsoft Learn, „Troubleshoot performance issues for Microsoft Defender for Endpoint on Linux" · Microsoft Learn, „Investigate agent health issues" · `mdatp health --help`

Revision #21
Created 2026-09-28 17:32:46 UTC by Tomasz Konieczny
Updated 2026-10-26 14:14:08 UTC by Tomasz Konieczny