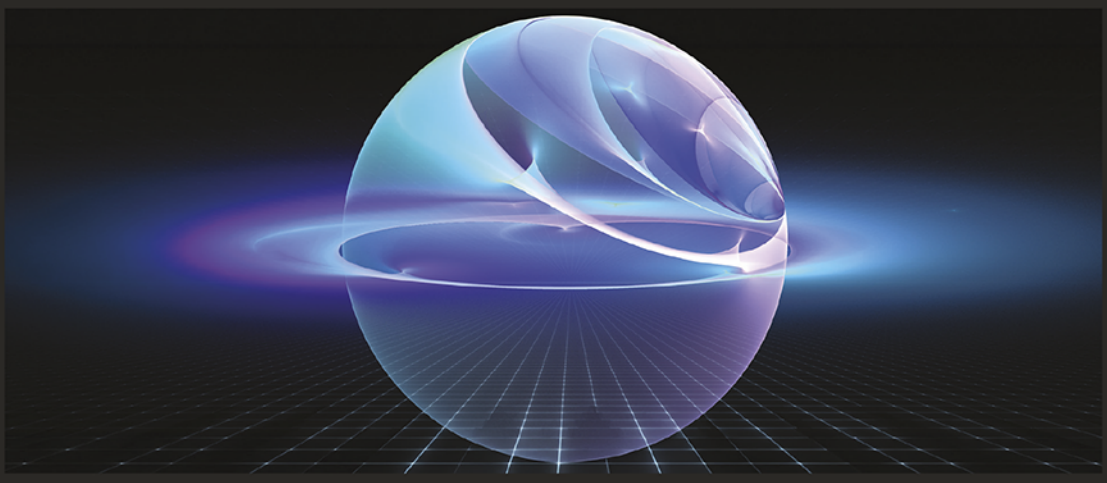
Helion

<packt>



Agentowa sztuczna inteligencja w cyberbezpieczeństwie

Budowa i automatyzacja inteligentnych testów penetracyjnych

ORHAN YILDIRIM

Tytuł oryginału: Agentic AI for Offensive Cybersecurity: Build and automate smarter penetration testing workflows using AI-driven agents

Tłumaczenie: Grzegorz Werner

ISBN: 978-83-289-4225-7

Copyright © Packt Publishing 2026.

First published in the English language under the title
'Agentic AI for Offensive Cybersecurity' – (9781806114474).

Polish edition copyright © 2027 by Helion S.A.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiejkolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz wydawca dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz wydawca nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
helion.pl/user/opinie/agszin
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Helion S.A.
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *helion.pl* (księgarnia internetowa, katalog książek)

Printed in Poland.

• Kup książkę
• Poleć książkę
• Oceń książkę

• Księgarnia internetowa
• Lubię to! » Nasza społeczność

Spis treści

O autorze 11

O recenzencie 12

Przedmowa 13

CZĘŚĆ 1. Podstawy agentowej sztucznej inteligencji

ROZDZIAŁ 1
Wprowadzenie do agentowej sztucznej inteligencji 21

Czym jest agentowa AI 22
Zewnętrzny rekonesans tradycyjnej automatyzacji i agentowej AI 23
Tradycyjna automatyzacja 24
Agentowa AI 24
Myślenie agentowe w cyberbezpieczeństwie 25
Rzeczywiste korzyści z autonomicznych agentów 28
Cykl życia testów penetracyjnych 30
Zarządzanie powierzchnią ataku w nowoczesnych środowiskach 32
Segmentacja PCI i testowanie sieci wewnętrznej 34
Porównanie tradycyjnych narzędzi z podejściem opartym na agentach AI ... 35
Podsumowanie 37

ROZDZIAŁ 2
Platformy AI do automatyzacji bezpieczeństwa 38

Przegląd platform AI do automatyzacji zabezpieczeń 39
Platformy no-code i code-first 40
Jak agentowa AI wpisuje się w automatyzację bezpieczeństwa 41
LangChain — mocne strony i zastosowania w bezpieczeństwie 42
LangChain — korzyści i wyzwania 44
n8n — automatyzacja no-code dla specjalistów ds. bezpieczeństwa 45

Kup książkę Poleć książkę

Inne godne uwagi narzędzia — CrewAI, AutoGen itp. 47
CrewAI 47
AutoGen 47
Jak wybrać narzędzie odpowiednie do danego zastosowania 49
Podsumowanie 51

CZĘŚĆ 2. Budowanie i orkiestrowanie przepływów pracy opartych na sztucznej inteligencji

ROZDZIAŁ 3
Budowanie pierwszego przepływu pracy AI w n8n 55
Wymagania techniczne 56
Konfigurowanie n8n na potrzeby agentowych przepływów pracy 56
Opcje instalacji 56
Budowanie własnego obrazu Dockera dla procesów bezpieczeństwa 57
Uruchamianie środowiska za pomocą Docker Compose 58
Dostęp do n8n z poziomu przeglądarki 61
Łączenie się z interfejsami OpenAI i analizy zagrożeń 62
API analizy zagrożeń 66
Projektowanie pierwszego bota rozpoznawczego bez pisania kodu 83
Co zbudujesz 83
Jak będzie działał przepływ 83
Implementowanie systemu RAG w n8n 87
Zabezpieczanie agentowego przepływu pracy 93
Co należy robić zawsze 94
Działania opcjonalne, ale zdecydowanie zalecane 95
Podsumowanie 95

ROZDZIAŁ 4
Agentowa AI do zarządzania powierzchnią ataku 97
Budowanie agentowego przepływu pracy ASM 98
Integracja odkrywania zasobów, badania odcisków palców wyliczonych zasobów i wykrywania podatności 102
Etap 1. Przepływ odkrywania zasobów 103
Etap 2. Badanie odcisków palców wyliczonych zasobów 118
Etap 3. Wykrywanie podatności 125
Etap 4. Agent zapytań do bazy danych 128
Podsumowanie 130

Kup książkę Poleć książkę

ROZDZIAŁ 5
Testowanie sieci wewnętrznej i segmentacji PCI 131
Wymagania techniczne 132
Czym jest PCI, czym jest segmentacja i dlaczego mają znaczenie 132
Ręczne testowanie segmentacji — tradycyjne podejścia 134
Skanowanie portów TCP (SYN) 135
Alternatywne skanowanie portów TCP (skanowanie TCP CONNECT) 135
Skanowanie portów UDP (SYN) 136
Budowanie przepływów mapowania wewnętrznego z użyciem agentowej AI 137
Krok 1. Architektura przepływu pracy i konfiguracja agenta 137
Krok 2. Budowanie narzędziowego przepływu pracy 140
Krok 3. Wykonywanie testów segmentacji 144
Podsumowanie 147

ROZDZIAŁ 6
Testowanie aplikacji webowych z wykorzystaniem agentowej AI 148
Wymagania techniczne 149
Czym jest Model Context Protocol 150
Konfiguracja środowiska programistycznego 151
Tworzenie narzędzia do automatyzacji przeglądarki 155
Integracja z Claude Desktop 164
Rozwiązywanie problemów 167
Problem: niewłaściwa wersja Node.js w nvm 168
Problem: błąd „Cannot find module” 169
Problem: przeglądarka się nie otwiera 169
Problem: błąd „fetch is not defined” 170
Powierzchnia ataku aplikacji webowych 170
Identyfikowanie punktów wejścia 171
Parametry URL 174
Mapowanie powierzchni ataku 175
Wykrywanie podatności na XSS 179
Typy XSS 179
Wykrywanie podatności na Reflected XSS 181
Wykrywanie podatności na DOM XSS 185
Wykrywanie podatności na wstrzykiwanie kodu SQL 193
Krok 1. Lista ładunków i wzorce błędów 195
Krok 2. Systematyczna pętla testowa 196
Krok 3. Generowanie raportu 198
Ocena narzędzia do wykrywania podatności na wstrzykiwanie kodu SQL 200
Podsumowanie 201

Kup książkę Poleć książkę

CZĘŚĆ 3. Wykorzystywanie luk w zabezpieczeniach, raportowanie i ciągła analiza zagrożeń

ROZDZIAŁ 7
Wykorzystywanie luk w zabezpieczeniach, generowanie ładunków i raportowanie 205

Integracja wykorzystywania luk w zabezpieczeniach i generowania ładunków 206
Metodyki projektowania narzędzi MCP do testowania podatności 209
Osobne narzędzie dla każdego typu wstrzyknięcia (ładunki zakodowane na sztywno) 210
Wykorzystanie rozszerzonych opisów narzędzi 211
Integracja narzędzi z przeglądarkowym rozwiązaniem MCP 214
Raportowanie podatności za pomocą agentowych przepływów MCP 218
Główne komponenty raportowego rozwiązania MCP 220
Podsumowanie 226

ROZDZIAŁ 8
Uczenie się, badania i agenty CTI 227

Wymagania techniczne 229
Budowanie agenta do badania podatności 229
Konfiguracja integracji z CISA KEV 230
Scraping i parsowanie danych z zaufanych kanałów 241
Identyfikowanie wiarygodnych źródeł 241
Budowanie przepływu pracy do scrapingu wielu źródeł informacji 242
Testowanie przepływu pracy 253
Wykorzystanie osadzeń i magazynów wektorowych jako pamięci agenta 254
Wybieranie magazynu wektorowego 255
Lokalne wdrażanie Qdranta 255
Tworzenie kolekcji z informacjami o zagrożeniach 256
Rozszerzanie przepływu scrapingu o osadzenia 258
Integrowanie informacji badawczych z przepływami testowania 262
Architektura oceny ekspozycji 262
Dostęp do API Shodana i konfiguracja poświadczeń API 263
Powiadamianie i kontekstowe ustalanie priorytetów podatności 270
Budowanie modelu oceny priorytetów 271
Konfigurowanie ntfy pod kątem powiadomień push 274
Rozszerzanie agenta do badania podatności 276
Testowanie przepływu pracy związanego z powiadomieniami 278

Kup książkę Poleć książkę

Budowanie agenta CTI ... 280
Architektura agenta CTI ... 280
Konfigurowanie poświadczeń AlienVault OTX ... 282
Budowanie przepływu pracy związanego z gromadzeniem danych ... 282
Budowa agenta zapytań CTI ... 285
Optymalizacja danych wektorowych dla agentów AI ... 287
Podsumowanie ... 289

ROZDZIAŁ 9
Skalowanie i kwestie etyczne ... 290

Projektowanie systemów wieloagentowych ... 291
Wzorce koordynacji agentów ... 291
Projektowanie pod kątem odporności ... 296
Strategie komponowania przepływów pracy ... 299
Protokoły komunikacyjne dla systemów agentowych ... 301
Budowanie skalowalnego potoku testów penetracyjnych w n8n ... 307
Definiowanie etapów potoku ... 308
Implementowanie walidacji zakresu ... 309
Orkiestracja wykonywania przepływów pracy ... 309
Kwestie skalowania ... 310
Zarządzanie bezpieczeństwem, kontrolą dostępu i nadzorem nad przepływami pracy ... 313
Zarządzanie sekretami ... 313
Magazyn poświadczeń n8n ... 314
Zewnętrzne menedżery sekretów ... 314
Rotacja poświadczeń ... 315
Kontrola dostępu ... 316
Rejestrowanie audytowe ... 318
Zarządzanie zmianami ... 321
Etyczne i prawne ramy autonomicznych testów bezpieczeństwa ... 323
Imperatyw autoryzacji ... 324
Odpowiedzialne ujawnianie ... 325
Ramy prawne ... 326
Zasady etyczne wykraczające poza zgodność z prawem ... 327
Przyszłość agentowej sztucznej inteligencji w bezpieczeństwie ofensywnym ... 330
Stan obecny ... 330
Prognoza krótkoterminowa ... 331
Potencjał transformacyjny ... 333
Konfrontacja z wyzwaniami i obawami ... 335
Przygotowanie się na nadchodzące zmiany ... 338
Podsumowanie ... 339

Kup książkę Poleć książkę

Kup książkę Poleć książkę

Budowanie pierwszego przepływu pracy AI w n8n

Rozdział 3

Agentowa AI to nie tylko koncepcja; to coś, co możesz zbudować, zobaczyć i z czym możesz wchodzić w interakcję. W tym rozdziale przejdziemy od teorii do działania. Jeśli spędziłeś godziny na kopiowaniu i wklejaniu wyników z Nmapa lub ręcznym selekcjonowaniu poddomen, wiesz, jak żmudna to praca. Spróbujmy zatem zautomatyzować ten znój.

Będziemy używać **n8n**, platformy open source do automatyzowania procesów bez pisania kodu, która doskonale radzi sobie z łączeniem interfejsów API, obsługą logiki i funkcjonowaniem jako warstwa orkiestracji inteligentnych agentów. Powód, dla którego wybrałem n8n spośród dziesiątek alternatyw, jest prosty: obniża ona barierę wejścia do automatyzacji, oferując jednocześnie ogromne możliwości.

Krok po kroku omówimy następujące zagadnienia:

- Bezpieczna instalacja i konfiguracja n8n na własnym komputerze lub serwerze.
- Łączenie platformy z usługami AI, takimi jak OpenAI lub HuggingFace.
- Budowa funkcjonalnego **agenta rozpoznawczego**, który zbiera poddomeny, sprawdza porty, a nawet podsumowuje wyniki z użyciem LLM-ów.
- Rozszerzanie bota o możliwości **generowania wspomaganego wyszukiwaniem** (ang. *retrieval-augmented generation*, **RAG**), żeby agent mógł pobierać kontekst z zewnętrznych danych i podejmować lepsze decyzje.
- Zabezpieczanie procesów w celu uniknięcia typowych błędów, takich jak wycieki kluczy API lub bomby logiczne.

Zobaczysz, jak każdy węzeł w n8n działa jako punkt decyzyjny dla Twojego agenta. Węzły te nie tylko przenoszą dane; reprezentują one logikę, pamięć i zdolność adaptacji, które definiują agentową AI. Jeśli kiedykolwiek napisałeś skrypt pełen zagnieżdżonych bloków `if-else`, który przestał działać, gdy tylko wydarzyło się coś nieoczekiwanego, to właśnie znalazłeś lekarstwo. Węzły podejmują decyzje, adaptują się i wracają do działania bez konieczności pilnowania przez Ciebie każdego przypadku brzegowego. Co ważniejsze, pozwalają one na szybkie prototypowanie, iterowanie przy minimalnych tarciach i budowanie złożonych przepływów pracy bez zatrudniania programisty backendowego.

Kup książkę Poleć książkę

Pod koniec rozdziału uruchomisz swój pierwszy zorientowany na bezpieczeństwo przepływ pracy AI zdolny do przeprowadzania rekonesansu, podsumowywania danych i dostosowywania się do danych wejściowych w sposób nieosiągalny dla tradycyjnych narzędzi. Nie będzie to pasywny pulpit nawigacyjny, ale żywy, rozumujący agent automatyzacji.

Wymagania techniczne

Aby na bieżąco śledzić i wykonywać przykłady omawiane w tym rozdziale, będziesz potrzebować:

- **System operacyjny:** Windows 10/11, macOS lub Linux (zalecany Ubuntu 20.04+).
- **Node.js:** wersja 18 lub nowsza.
- **Docker:** wersja 20.10 lub nowsza (w przypadku instalacji *opcji 2*.).
- **Pamięć RAM:** minimum 4 GB, zalecane 8 GB.
- **Miejsce na dysku:** 2 GB dla n8n i zależności.
- **Sieć:** dostęp do internetu na potrzeby integracji z API (OpenAI, Shodanem itp.).
- Klucz API OpenAI (lub kompatybilnego dostawcy LLM-ów).

Konfigurowanie n8n na potrzeby agentowych przepływów pracy

W tym podrozdziale omówimy trzy kluczowe zagadnienia:

- Instalowanie i konfigurowanie n8n.
- Konfigurowanie bezpiecznego środowiska dla operacji agentowych.
- Interfejs n8n z perspektywy myślenia agentowego.

Opcje instalacji

Możesz uruchomić n8n z użyciem jednej z trzech popularnych metod:

- **Opcja 1. Instalacja lokalna (do testów)**. Jest to najszybszy sposób na testowanie lokalne. Po uruchomieniu platforma n8n będzie dostępna pod adresem *http://localhost:5678*:

```
npm install n8n -g
n8n
```

- **Opcja 2. Docker (zalecana do środowisk produkcyjnych)**. W tej opcji do uruchomienia n8n w izolowanym środowisku wykorzystuje się Dockera, co jest bardziej odpowiednie do zastosowań produkcyjnych:

Kup książkę Poleć książkę

```
docker run -it --rm \
   -p 5678:5678 \
   -v ~/.n8n:/home/node/.n8n \
   -e N8N_BASIC_AUTH_USER=admin \
   -e N8N_BASIC_AUTH_PASSWORD=strongpassword \
   n8nio/n8n
```

Na potrzeby tej książki będziemy korzystać z *opcji 2.*, ponieważ zapewnia ona najszybszą i najprostszą konfigurację do śledzenia przykładów i budowania przepływów pracy agentów.

- **Opcja 3. Gotowa konfiguracja (środowisko startowe)**. Aby przyspieszyć naukę, przygotowaliśmy startowe repozytorium na GitHubie, które obejmuje:
  - Konfigurację Docker Compose.
  - Wstępnie skonfigurowany szablon pliku środowiskowego.
  - Podstawowy przepływ rekonesansu jako punkt wyjścia.

Uwaga

Najbardziej aktualne instrukcje, przewodniki wdrażania i dokumentację węzłów można znaleźć w oficjalnej dokumentacji pod adresem *https://github.com/n8n-io/n8n*. Repozytorium to zawiera pełne przewodniki konfiguracji, przykładowe przepływy pracy, obsługiwane integracje i rozszerzenia dodane przez społeczność użytkowników, które pomogą Ci wyjść poza zagadnienia omówione w tej książce.

Zanim przejdziemy dalej, wyjaśnię, w jaki sposób opcje instalacji odnoszą się do kolejnych podrozdziałów. Dalej w tym rozdziale będziemy używać *opcji 2*. (Dockera). Uruchomienie n8n w izolowanym środowisku Dockera pozwala nam zachować niezależność konfiguracji od systemu operacyjnego i spakować wszystkie wymagane komponenty do jednego odtwarzalnego kontenera. Ułatwia to również dołączenie dodatkowych zależności, których nasze agentowe przepływy pracy będą potrzebować później, przy czym wszystkimi elementami będzie zarządzać ta sama konfiguracja Docker Compose.

Budowanie własnego obrazu Dockera dla procesów bezpieczeństwa

Ten niestandardowy plik Dockerfile bazuje na oficjalnym obrazie `n8nio/n8n:nightly`. Kompilacja `nightly` to najczęściej aktualizowana wersja n8n, co gwarantuje, że kontener zawsze zawiera najnowsze funkcje, poprawki bezpieczeństwa i ulepszenia silnika przepływu pracy. Na potrzeby tej książki obraz `nightly` stanowi stabilną podstawę, która obsługuje wszystkie automatyzacje agentowe bez konieczności ręcznego aktualizowania. Nic w tym pliku Dockerfile nie modyfikuje samego n8n; zamiast tego rozszerzamy obraz bazowy, aby dodać narzędzia wymagane w przepływach pracy związanych z bezpieczeństwem.

Kup książkę Poleć książkę

Utwórz plik o nazwie *Dockerfile* i zdefiniuj go w sposób pokazany poniżej, aby rozszerzyć oficjalny obraz n8n o wymagane zależności:

```
FROM n8nio/n8n:nightly

# Przełączamy się na konto roota, aby zainstalować pakiety systemowe
USER root

RUN apk update && apk add --no-cache \
    python3 \
    py3-pip \
    py3-virtualenv \
    gcc \
    musl-dev \
    libffi-dev \
    openssl-dev

RUN python3 -m venv /opt/venv && \
    /opt/venv/bin/pip install --upgrade pip && \
    /opt/venv/bin/pip install \
      pymongo \
      requests

ENV PATH="/opt/venv/bin:$PATH"

RUN mkdir -p /home/node/.cache && chown -R node:node /home/node/.cache

USER node
```

Ten plik Dockerfile instaluje kilka komponentów, które nie są domyślnie dołączone do n8n. Ponieważ w późniejszych rozdziałach będziemy korzystać ze skryptów Pythona, automatyzacji zewnętrznych interfejsów API, interakcji z bazą MongoDB i niestandardowej logiki rekonesansu, kontener wymaga dodatkowych pakietów systemowych i bibliotek Pythona do obsługi tych możliwości. Obraz instaluje Pythona, `pip`, `virtualenv`, typowe narzędzia kompilacyjne (`gcc`, `musl-dev` i `openssl-dev`) oraz pakiety Pythona `pymongo` i `requests`. Dodatki te pozwalają przepływom pracy n8n na wykonywanie kodu Pythona, komunikację z bazami danych i uruchamianie bardziej zaawansowanych zadań agentowych wewnątrz jednego kontenera. Przygotowując te zależności z góry, tworzymy samodzielne środowisko, w którym wszystkie kolejne przepływy pracy działają spójnie bez konieczności używania jakichkolwiek narzędzi na maszynie hosta.

Uruchamianie środowiska za pomocą Docker Compose

Aby Twoje wdrożenie n8n było trwałe i gotowe do zastosowań produkcyjnych, możesz skorzystać z poniższej konfiguracji *docker-compose.yml*:

```
version: "3.7"

services:
```

Kup książkę Poleć książkę

```
  n8n:
    build:
      context: .
      dockerfile: Dockerfile
    restart: always
    ports:
      - "5678:5678"
    environment:
      - N8N_HOST=n8n.cyprox.io
      - N8N_PORT=5678
      - N8N_PROTOCOL=https
      - LANG=en_US.UTF-8
      - LC_ALL=en_US.UTF-8
      - NODE_ENV=production
      - WEBHOOK_URL=https://n8n.cyprox.io/
      - DB_TYPE=postgresdb
      - DB_POSTGRESDB_HOST=${DB_HOST}
      - DB_POSTGRESDB_PORT=${DB_PORT}
      - DB_POSTGRESDB_DATABASE=${DB_NAME}
      - DB_POSTGRESDB_USER=${DB_USER}
      - DB_POSTGRESDB_PASSWORD=${DB_PASSWORD}
    volumes:
      - n8n_data:/home/node/.n8n
      - n8n_cache:/home/node/.cache
      - /local-files:/files
      - /local-files:/home/node/

  postgres:
    image: postgres:13 restart: always environment:
      - POSTGRES_USER=${DB_USER}
      - POSTGRES_PASSWORD=${DB_PASSWORD}
      - POSTGRES_DB=${DB_NAME}
    volumes:
      - postgres_data:/var/lib/postgresql/data
    ports:
      - "${DB_PORT}:${DB_PORT}"

volumes:
  n8n_data:
    external: true
  postgres_data:
    external: true
  n8n_cache:
    external: false
```

Ten plik *docker-compose.yml* definiuje gotowe do wdrożenia produkcyjnego środowisko n8n, które obsługuje agentowe przepływy pracy dzięki połączeniu dwóch głównych kontenerów i woluminów trwałych. Przyjrzyjmy się temu bliżej:

1. `n8n` (główny kontener aplikacji). Jest to podstawowa usługa, w której działają przepływy automatyzacyjne. Wykorzystuje niestandardowy obraz Dockera (zbudowany z lokalnego pliku Dockerfile), który rozszerza oficjalny obraz

`n8nio/n8n:nightly` o dodatkowe narzędzia, takie jak Python, `pymongo` i `requests`. Rozszerzenia te umożliwiają:

- Wykonywanie niestandardowych skryptów Pythona w ramach przepływów pracy n8n.
- Łączenie się z bazami danych dotyczących analizy zagrożeń (np. MongoDB).
- Wykonywanie wywołań API z użyciem natywnych mechanizmów Pythona.

Kluczowe zmienne środowiskowe to:

- `N8N_HOST`, `N8N_PROTOCOL` i `N8N_PORT`. Określają sposób udostępniania interfejsu użytkownika n8n i punktów końcowych webhooków (np. *https://n8n.cyprox.io*).
- `WEBHOOK_URL`. Potrzebna do prawidłowej obsługi zewnętrznych wywołań zwrotnych lub integracji.
- Poświadczenia użytkownika połączonej instancji bazy danych PostgreSQL.

2. `postgres` (kontener bazy danych). W tym kontenerze działa baza danych PostgreSQL 13, której n8n używa do przechowywania:
   - definicji przepływów pracy,
   - dzienników wykonywania,
   - poświadczeń (zaszyfrowanych),
   - ustawień użytkownika.

   Uruchomienie bazy danych PostgreSQL w oddzielnej usłudze zapewnia trwałość danych, nawet jeśli kontener n8n zostanie przebudowany lub zaktualizowany.

   Do zachowywania ważnych danych pomiędzy restartami kontenera lub aktualizacjami obrazu używane są następujące woluminy:

   - `n8n_data`. Przechowuje cały wewnętrzny stan n8n, w tym przepływy pracy, poświadczenia i wpisy dziennika. Wolumin ten jest oznaczony jako zewnętrzny, z założeniem, że został już utworzony (np. za pomocą polecenia `docker volume create n8n_data`).
   - `n8n_cache`. Używany do celów buforowania, na przykład do przechowywania plików tymczasowych lub odpowiedzi LLM-a. Jest on lokalny dla tego stosu.
   - `/localfiles:/files` i `/localfiles:/home/node/` to punkty wiązania do importowania lub eksportowania plików lokalnych. Przydatne do debugowania, obsługi dzienników, przechowywania plików modeli lub przenoszenia danych między hostem a kontenerem.
   - `postgres_data`. Zachowuje dane PostgreSQL podczas przebudowy kontenera. Również oznaczony jako zewnętrzny z uwagi na bezpieczeństwo i stabilność.

Kup książkę Poleć książkę

Dostęp do n8n z poziomu przeglądarki

Gdy kontenery są już gotowe do pracy, możesz uzyskać dostęp do interfejsu n8n pod adresem URL zdefiniowanym w zmiennych środowiskowych:

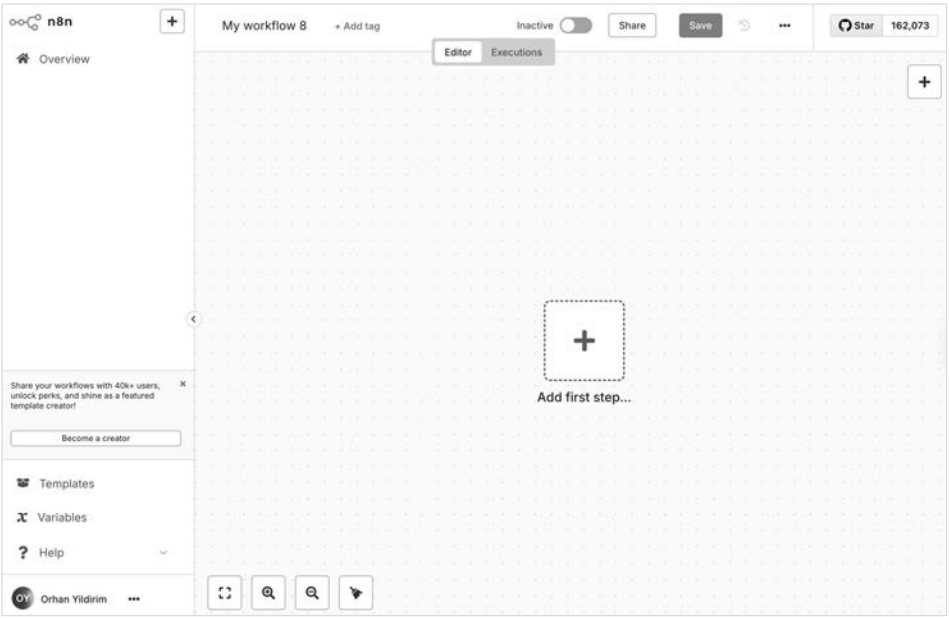
```
https://n8n.cyprox.io
```

Ten adres URL odpowiada ustawieniom `N8N_HOST`, `N8N_PORT` i `N8N_PROTOCOL` podanym w pliku *docker-compose.yml*. Upewnij się, że serwer DNS jest poprawnie skonfigurowany, a nazwa domenowa wskazuje na publiczny adres IP serwera. Jeśli wdrażasz aplikację lokalnie lub używasz odwrotnego proxy, upewnij się, że protokół SSL jest poprawnie skonfigurowany, aby uniknąć problemów z połączeniem.

Jeśli wszystko skonfigurowano poprawnie, otwarcie tego adresu URL w przeglądarce spowoduje przejście do pulpitu nawigacyjnego n8n, który jest centralnym obszarem roboczym do tworzenia agentowych przepływów pracy AI i zarządzania nimi.

W tym miejscu możesz zacząć tworzyć węzły, importować szablony i łączyć automatyzacje bezpieczeństwa oparte na sztucznej inteligencji.

Po uruchomieniu kontenerów i udostępnieniu usługi przejście pod skonfigurowany adres URL powinno wyświetlić pulpit nawigacyjny n8n, jak pokazano na rysunku 3.1.



Rysunek 3.1. Pulpit nawigacyjny n8n

Zanim przejdziemy dalej, podsumujmy, w jakim miejscu jesteśmy. W tym momencie środowisko n8n jest w pełni operacyjne: stos w kontenerach Dockera działa, nasz niestandardowy obraz został zbudowany, a pulpit nawigacyjny n8n jest dostępny z poziomu przeglądarki. Mając przygotowaną platformę, możemy teraz rozpocząć integrację usług zewnętrznych, które będą napędzać nasze agentowe przepływy pracy.

Kup książkę Poleć książkę

Kolejnym krokiem jest połączenie n8n z interfejsami API OpenAI i analizy zagrożeń, co umożliwi naszym agentom wnioskowanie, wzbogacanie danych i podejmowanie świadomych decyzji w ramach zautomatyzowanych przepływów pracy.

Łączenie się z interfejsami OpenAI i analizy zagrożeń

Gdy Twoja instancja n8n działa, nadszedł czas, aby wyposażyć ją w mózg. W tym podrozdziale podłączymy ją do modeli GPT OpenAI oraz platform analizy zagrożeń, takich jak Shodan, VirusTotal i Censys.

Połączenia te zmieniają statyczną automatyzację w dynamiczne podejmowanie decyzji. Nie tylko przesyłasz dane między narzędziami; umożliwiasz swoim agentom myślenie, rozumowanie i wzbogacanie kontekstu w czasie rzeczywistym.

Dowiesz się:

- Jak bezpiecznie przechowywać w n8n poświadczenia API dla dostawców informacji o zagrożeniach.
- Jak wywoływać API analizy zagrożeń z użyciem węzłów natywnych lub węzłów *HTTP Request*.
- Jak strukturyzować dane wejściowe promptów łączących dane wyjściowe LLM-a z danymi analizy zagrożeń.
- Jak konfigurować praktyczne przepływy wzbogacania danych z użyciem takich API jak AbuseIPDB, VirusTotal, GreyNoise, OTX i urlscan.io.

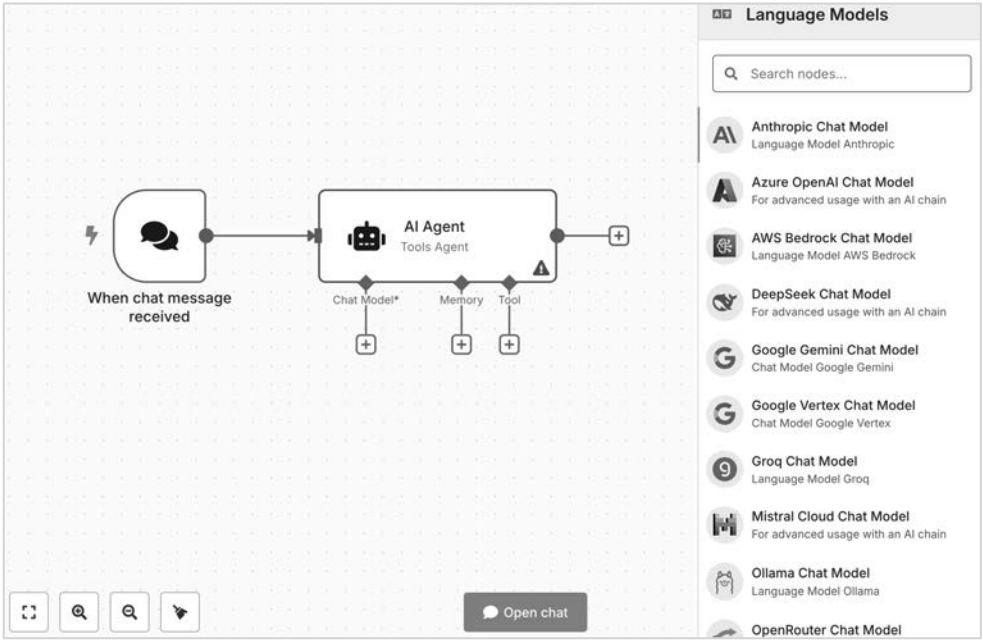
Zanim zaczniemy integrować API analizy zagrożeń, musimy najpierw skonfigurować dostawcę LLM-a, którego nasz węzeł *AI Agent* będzie używał do rozumowania. Opcje modeli czatu obsługiwane w n8n pokazano na rysunku 3.2.

n8n obsługuje szeroką gamę dostawców LLM, a na powyższym zrzucie ekranu pokazano pełny zestaw modeli czatu dostępnych w momencie pisania tego tekstu. Chociaż w przykładach przepływów pracy w tej książce będziemy używać głównie modeli GPT firmy OpenAI, warto podkreślić, że jako wejście *Chat Model* węzła *AI Agent* można wybrać dowolnego z obsługiwanych dostawców. Poniższa lista krótko podsumowuje głównych dostawców i to, co oferują.

Inne modele LLM obecnie dostępne w n8n to:

- **Model czatu Anthropic** (Claude):
  - **Modele** Claude 3 Opus, Sonnet i Haiku.
  - **Mocne strony** — rozumowanie z długim kontekstem (ponad 100 tys. tokenów), bezpieczniejsze dopasowanie i rozumienie wieloetapowe.
  - **Zastosowanie** — odpowiedzi zgodne z przepisami i głębokie podsumowywanie dzienników.

Kup książkę Poleć książkę



Rysunek 3.2. Dostępne modele językowe dla węzła AI Agent w n8n

- **Model czatu Azure OpenAI**:
  - **Modele** GPT-3.5, GPT-4 w Azure.
  - **Mocne strony** — hostowane na platformie Azure, przydatne dla przedsiębiorstw z regionalnymi lub regulacyjnymi wymogami dotyczącymi hostingu.
  - **Zastosowanie** — wykorzystanie możliwości OpenAI w środowisku chmurowym zarządzanym przez Microsoft.
- **Model czatu AWS Bedrock**:
  - **Modele** Claude, Mistral i Amazon Titan (przez Bedrock).
  - **Mocne strony** — integracja z ekosystemem AWS; idealne dla zespołów używających już GenAI za pośrednictwem Bedrocka.
  - **Zastosowanie** — agenty uruchamiane przez zdarzenia AWS lub analizujące dane o zagrożeniach w chmurze.
- **Model czatu DeepSeek**:
  - **Modele** DeepSeek-VL (model wizyjno-językowy) i DeepSeek-LLM.
  - **Mocne strony** — nowe otwarte modele o wysokiej skuteczności w pracy wielozadaniowej.
  - **Zastosowanie** — eksperymentalne potoki agentowe i hybrydowa analiza wizyjno-językowa.

Kup książkę Poleć książkę

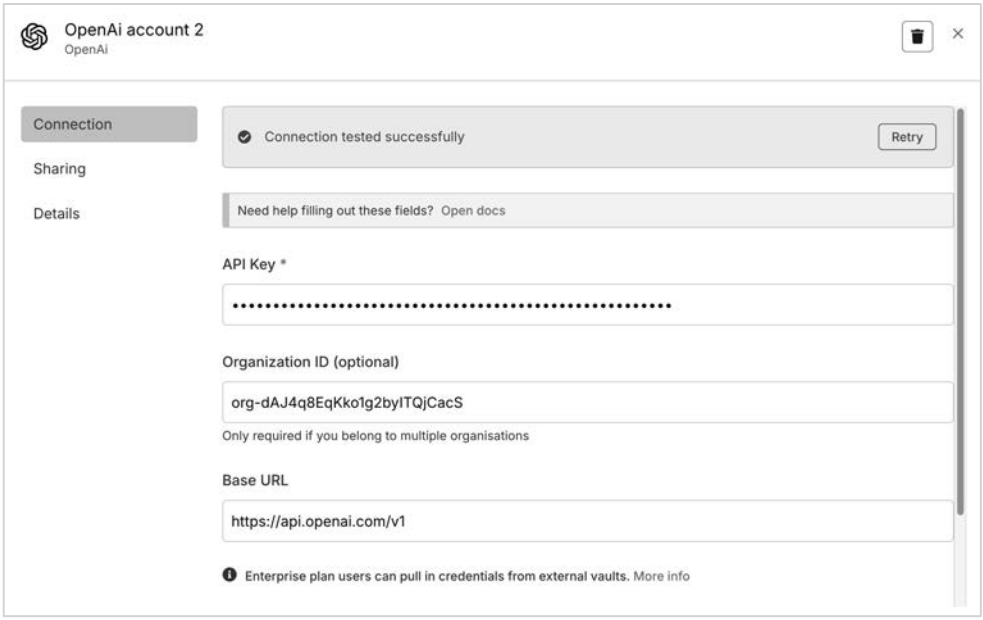
- **Model czatu Google Gemini**:
  - **Model** Gemini 1.5 Pro (wcześniej Bard).
  - **Mocne strony** — szybki, o niskich opóźnieniach; świetnie radzi sobie z danymi ustrukturyzowanymi i podsumowywaniem.
  - **Zastosowanie** — boty FAQ dotyczące bezpieczeństwa, narzędzia do podsumowywania podatności i powiadomień.
- **Model czatu Google Vertex**:
  - **Model** Gemini Pro za pośrednictwem Vertex AI.
  - **Mocne strony** — działa w Google Cloud; obsługuje dostrajanie promptów i zarządzane punkty końcowe.
  - **Zastosowanie** — środowiska korporacyjne, w których wdrożono już Vertex AI.
- **Model czatu Groq**:
  - **Model** LLaMA-2 hostowany na ultraszybkich układach wnioskowania Groq.
  - **Zalety** — duża szybkość, dobre rozwiązanie do automatyzacji o niskich opóźnieniach.
  - **Zastosowanie** — podejmowanie decyzji przez agenta w czasie rzeczywistym (np. klasyfikacja lub segregacja powiadomień).
- **Model czatu Mistral Cloud**:
  - **Modele** Mistral-7B i Mixtral (Mixture-of-Experts).
  - **Zalety** — otwarty kod źródłowy; w wielu zadaniach skuteczność porównywalna z GPT-3.5.
  - **Zastosowanie** — lekkie wnioskowanie, wdrożenia w odizolowanych środowiskach (ang. *air-gapped*) i rozumowanie PoC.
- **Model czatu Ollama**:
  - **Modele** Llama 2, Mistral, Phi-2 i inne (uruchamiane lokalnie).
  - **Zalety** — idealne do konfiguracji offline lub hostowanych samodzielnie (na laptopie lub serwerze).
  - **Zastosowanie** — automatyzacja z naciskiem na prywatność i narzędzia dla odłączonych zespołów red team.
- **Model czatu OpenRouter**:
  - **Modele** — ujednolicona brama do wielu dużych modeli językowych (GPT-4, Claude, Mixtral itp.).
  - **Zalety** — pozwala kierować prompty do różnych dostawców za pomocą jednego klucza API.
  - **Zastosowanie** — agenty do testów A/B, logika awaryjna i porównywanie modeli.
- **Model czatu xAI Grok**:
  - **Model** Grok (stworzony przez xAI/zespół Elona Muska).

Kup książkę Poleć książkę

- **Zalety** — wciąż dojrzewający, ale interesujący w przypadku eksperymentalnych konfiguracji agentów.
- **Zastosowanie** — nie jest jeszcze zalecany do zastosowań produkcyjnych, ale można go wypróbować w projektach hobbystycznych.

- **Model czatu OpenAI**:
  - **Modele** GPT-4, GPT-3.5 i GPT-4-turbo.
  - **Zalety** — najwyższa skuteczność i ogromne wsparcie ekosystemowe.
  - **Zastosowanie** — prawie wszystko, co ma związek z podsumowywaniem, wnioskowaniem, selekcją, klasyfikacją i wyjaśnianiem PoC.

Zanim będziemy mogli użyć jakiegokolwiek LLM-a w n8n, musimy skonfigurować nasze poświadczenia OpenAI. Rysunek 3.3 pokazuje, jak dodać i przetestować klucz API OpenAI w n8n.



Rysunek 3.3. Ekran poświadczeń OpenAI w n8n

Jako głównego modelu językowego napędzającego agentowe przepływy będę używał modeli GPT firmy OpenAI. Interfejs API OpenAI jest stabilny, dobrze udokumentowany i powszechnie stosowany w branży, co czyni go praktycznym punktem wyjścia do automatyzacji bezpieczeństwa ofensywnego. n8n obsługuje jednak wielu innych dostawców LLM-ów, takich jak Claude, Gemini, Mistral i Groq, których możesz swobodnie eksplorować lub zastępować w zależności od przypadku użycia, wymagań dotyczących prywatności lub ograniczeń regionalnych. Możesz nawet tworzyć konfiguracje hybrydowe, w których różne agenty używają różnych modeli, w zależności od złożoności lub charakteru zadania.

Kup książkę Poleć książkę

Aby połączyć się z OpenAI, wystarczy utworzyć wpis poświadczeń z użyciem klucza API (jak pokazano na poprzednim zrzucie ekranu). Opcjonalnie, jeśli jesteś częścią organizacji w OpenAI, możesz podać swój identyfikator organizacji, ale nie jest to wymagane w przypadku kont osobistych lub kont z jednym dzierżawcą. Podstawowy adres URL to *https://api.openai.com/v1*, pod który n8n kieruje wszystkie interakcje z modelami.

Po skonfigurowaniu i zweryfikowaniu klucza API OpenAI, jak pokazano w poprzednim przykładzie, węzeł *AI Agent* może wysyłać zapytania do modeli GPT w ramach dowolnego przepływu pracy, niezależnie od tego, czy chodzi o podsumowanie wyników skanowania portów, wygenerowanie logiki ładunku, czy też udzielenie odpowiedzi na kontekstowe prompty dotyczące luk w zabezpieczeniach.

API analizy zagrożeń

Oprócz wykorzystywania LLM-ów do wnioskowania i podsumowywania jedną z najpotężniejszych możliwości agentowych przepływów pracy jest zdolność do pobierania aktualnych informacji z zewnętrznych źródeł. Jako specjaliści ds. bezpieczeństwa każdego dnia polegamy na kanałach informacji o zagrożeniach, czy to w celu weryfikacji adresu IP, uzyskania dodatkowych informacji o podejrzanej domenie, czy też wykrycia infrastruktury dystrybucji złośliwego oprogramowania. W tym podrozdziale połączymy n8n z kilkoma powszechnie używanymi publicznymi API do analizy zagrożeń, w tym AbuseIPDB, VirusTotal, GreyNoise, OTX (AlienVault) i urlscan.io.

Każda z tych usług oferuje inne spojrzenie na krajobraz zagrożeń. AbuseIPDB i VirusTotal pozwalają sprawdzić historię podejrzanych elementów — wklej adres IP, hasz lub domenę, a dowiesz się, czy mają one kryminalną przeszłość. GreyNoise i urlscan.io służą raczej do analiz opartych na zachowaniu, koncentrując się na tym, jak dany adres IP zachowuje się w internecie lub jak strona o danym adresie URL wykonuje się w przeglądarce w środowisku piaskownicy. OTX (AlienVault) to usługa w dużej mierze społecznościowa, agregująca wskaźniki włamań dostarczane przez tysiące badaczy z całego świata. Dobrą wiadomością jest to, że wszystkie te platformy oferują bezpłatny dostęp z hojnymi limitami zapytań, co czyni je idealnymi do prototypowania przepływów pracy związanych ze wzbogacaniem danych. Rejestracja zajmuje zazwyczaj zaledwie kilka minut, a po uzyskaniu klucza API można rozpocząć wysyłanie żądań z poziomu n8n za pomocą węzła *HTTP Request* albo poprzez skonfigurowanie poświadczeń z użyciem zmiennych niestandardowych.

Oto czego potrzeba na początek:

- **AbuseIPDB**. Utwórz darmowe konto na stronie *https://www.abuseipdb.com/* i wygeneruj klucz API ze swojego pulpitu nawigacyjnego.
- **VirusTotal**. Zarejestruj się na stronie *https://www.virustotal.com/gui/home/upload*, kliknij swój profil, a następnie `API key` i skopiuj swój osobisty klucz (darmowi użytkownicy otrzymują ~500 żądań dziennie).
- **GreyNoise**. Odwiedź stronę *https://www.greynoise.io/*, utwórz bezpłatne konto i używaj swojego klucza API z punktem końcowym Community API.

Kup książkę Poleć książkę

- **OTX (AlienVault)**. Zarejestruj się na stronie *https://otx.alienvault.com/*, a następnie przejdź do strony ustawień, aby znaleźć swój klucz API.
- **urlscan.io**. Przejdź na stronę *https://urlscan.io/*, utwórz konto i pobierz klucz API ze swojego profilu użytkownika.

Gdy masz już klucze API, n8n ułatwia integrację usług. Możesz wstrzykiwać te klucze do żądań za pomocą pól *HTTP Header* lub *Query Parameter*, zapisać je jako zmienne środowiskowe w celu wielokrotnego użycia lub zarządzać nimi w bezpieczny sposób za pomocą menedżera poświadczeń (**Credentials Manager**).

W kolejnych punktach przedstawię kilka powszechnie używanych interfejsów API do analizy zagrożeń i pokażę, jak używać ich z poziomu n8n w przykładowym przepływie pracy związanym ze wzbogacaniem danych.

Krok 1. Użycie AbuseIPDB do analizy reputacji IP

Aby rozpocząć nasz przepływ pracy związany ze wzbogacaniem danych, zaczniemy od AbuseIPDB i zbudujemy prostą konfigurację z dwoma węzłami, która pobiera dane o reputacji dowolnego adresu IP.

AbuseIPDB to tworzona przez społeczność baza danych ze złośliwymi adresami IP. Jest to jedna z najprostszych, a zarazem najskuteczniejszych usług sprawdzania, czy dany adres IP został zgłoszony za nadużycia. Zapewnia ona oceny reputacji na podstawie zgłoszeń spamu, ataków rozproszonej blokady usług (ang. *distributed denial of service*, DDoS), ataków brute-force itp.

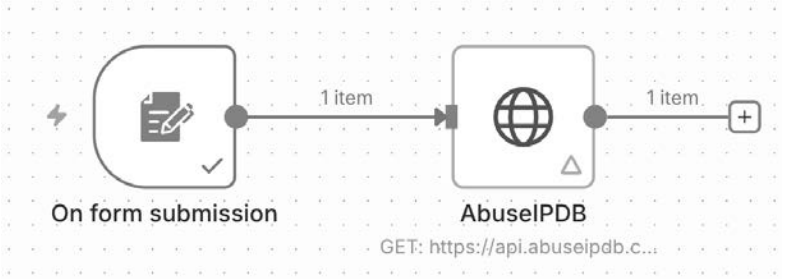
Zbudujemy w n8n przepływ pracy z dwoma węzłami:

- **Węzeł niestandardowego formularza HTML**. n8n ułatwia przyjmowanie danych wejściowych od użytkowników za pomocą osadzonych formularzy HTML. W tym przypadku poprosimy użytkownika o wprowadzenie adresu IP lub domeny. Te dane wejściowe zostaną później przekazane jako zmienna do żądania API.
- **Węzeł żądania HTTP**. Ten węzeł wyśle żądanie `GET` do API AbuseIPDB. Jako parametr zapytania będziemy przekazywać przesłany przez użytkownika adres IP. Sztuczka polega na użyciu edytora wyrażeń n8n, który pozwala nam dynamicznie odwoływać się do danych wejściowych formularza jako części żądania. Przydaje się tu składnia wyrażeń n8n. Pozwala ona wstawiać do żądania dane z poprzednich węzłów bez pisania ani jednego wiersza kodu JavaScript.

Łącząc te dwa węzły, stworzymy lekką mikrousługę wzbogacania danych. Użytkownik wprowadza adres IP, agent wysyła go do AbuseIPDB, a wynik (czy dany adres IP został zgłoszony za nadużycia) można następnie zarejestrować w dzienniku, zwizualizować, a nawet przekazać do węzła decyzyjnego.

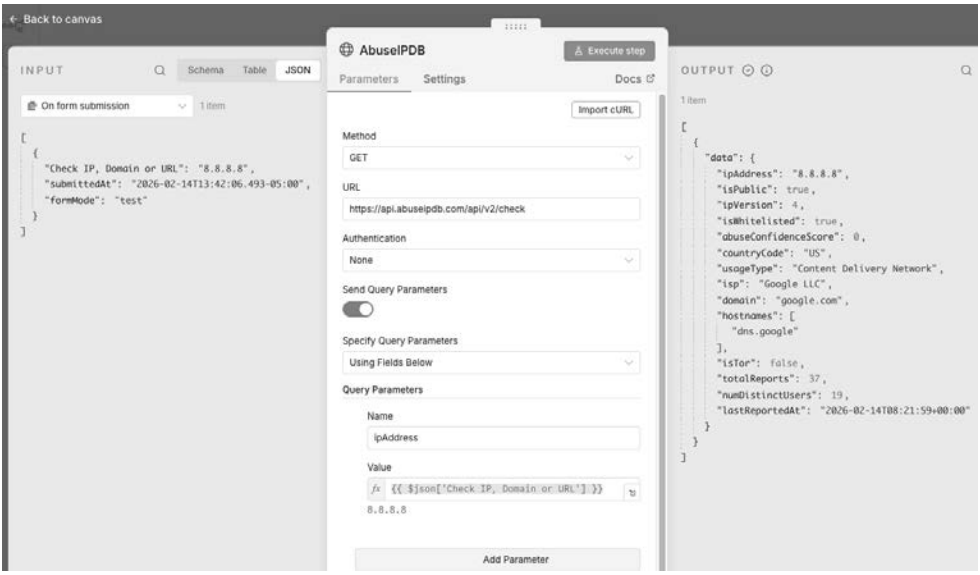
Na rysunku 3.4 pokazano, jak wygląda przepływ pracy z węzłem formularza HTML podłączonym do węzła żądania HTTP, który wysyła adres IP do AbuseIPDB.

Kup książkę Poleć książkę



Rysunek 3.4. Przepływ pracy związany z AbuseIPDB

Na rysunku 3.5 widać gotowy węzeł *HTTP Request*, który wysyła zapytanie do AbuseIPDB z wykorzystaniem adresu IP przesłanego przez użytkownika.



Rysunek 3.5. Węzeł HTTP Request odwołujący się do AbuseIPDB

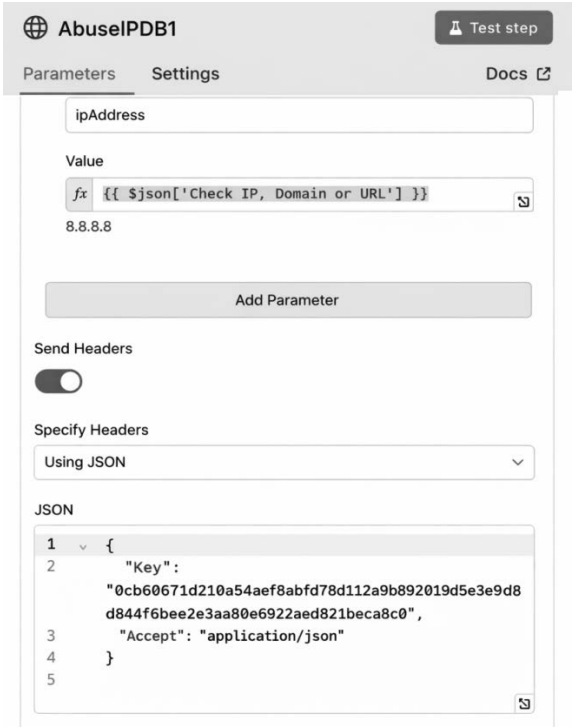
Zwróć szczególną uwagę na wyrażenie użyte w polu *Value*:

```
{{ $json["Check IP or Domain"] }}
```

Ta dynamiczna składnia to sposób na wstrzykiwanie danych z poprzednich węzłów do żądania. W tym przypadku n8n pobiera adres IP lub domenę, którą użytkownik wprowadził do formularza HTML, i przekazuje ją do API jako parametr zapytania o nazwie `ipAddress`.

W sekcji nagłówków bezpiecznie umieściliśmy również nasz klucz API wraz z odpowiednim typem Accept (rysunek 3.6).

Kup książkę Poleć książkę



Rysunek 3.6. Klucze API AbuseIPDB

W tym teście przesłaliśmy adres IP 8.8.8.8 i jak widać po prawej stronie, w odpowiedzi API otrzymaliśmy obszerny ładunek JSON zawierający takie pola jak `abuseConfidenceScore`, `usageType`, `isp`, `countryCode`, a nawet znacznik czasu ostatniego zgłoszenia tego adresu IP.

Nie przejmuj się, jeśli nie rozumiesz części tych danych; wkrótce zbudujemy agenty AI, które przeanalizują i wyjaśnią Ci te wyniki. W kolejnych rozdziałach przejdziemy od surowego formatu JSON do pełnej interpretacji danych przez LLM-y. To dopiero początek.

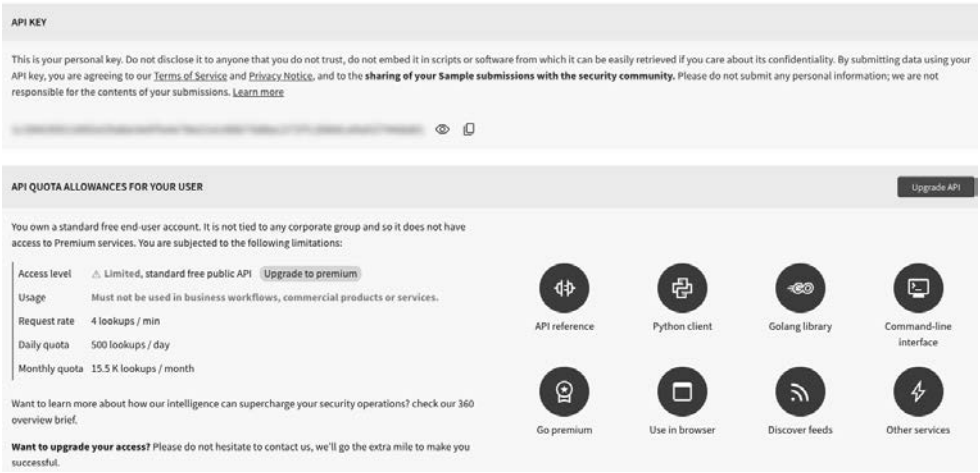
Krok 2. Zapytanie do VirusTotal w celu uzyskania dodatkowego kontekstu

Następnie skorzystamy z VirusTotal. Jest to prawdopodobnie jedna z najbardziej znanych platform do analizy złośliwego oprogramowania oraz skanowania adresów URL i plików. W tej części przepływu pracy wyślemy adres IP, domenę lub adres URL do VirusTotal i pobierzemy dane kontekstowe, takie jak liczba wykryć, znane powiązania i połączona infrastruktura. Podobnie jak w przypadku AbuseIPDB, użyjemy węzła *HTTP Request* w n8n, aby połączyć się z API VirusTotal.

Proces jest niemal identyczny: przekażemy element obserwowalny (np. adres IP) jako parametr `path` w wywołaniu API i dołączymy klucz API za pomocą niestandardowego nagłówka. Niezależnie od tego, czy sprawdzasz dystrybucję złośliwego oprogramowania, adresy URL służące do phishingu, czy też identyfikujesz infrastrukturę powiązaną ze złośliwą aktywnością, VirusTotal zapewnia solidną warstwę wzbogacania danych.

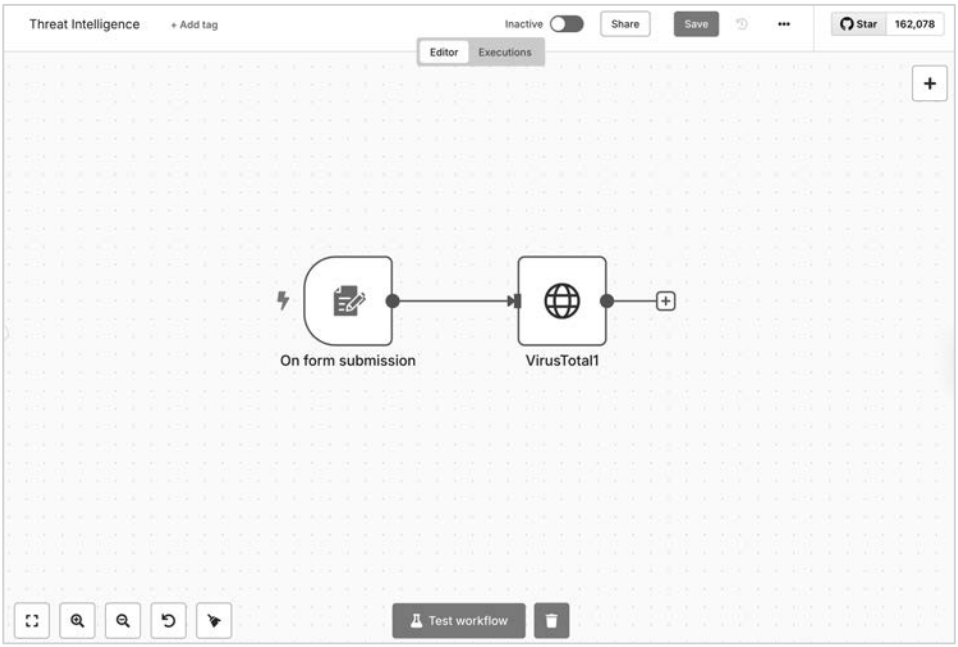
Kup książkę Poleć książkę

Rysunek 3.7 przedstawia pulpit nawigacyjny VirusTotal. Można tu pobrać klucz do API i sprawdzić limity użycia.



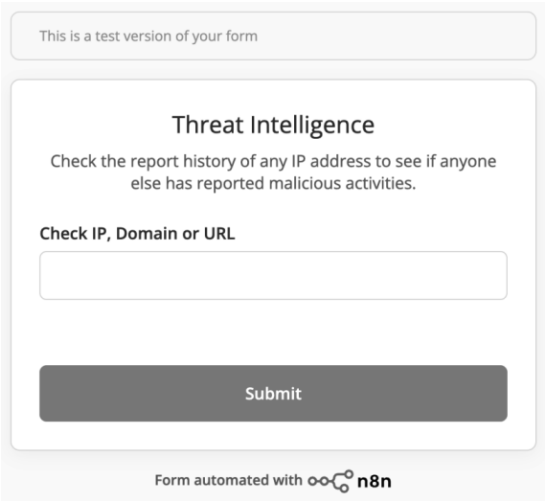
Rysunek 3.7. Klucz API VirusTotal

Rysunek 3.8 przedstawia ogólny przepływ pracy n8n, którego użyjemy do interakcji z API VirusTotal. Jest zbudowany według tego samego wzorca, który ustaliliśmy wcześniej: węzeł formularza, który zbiera dane wejściowe od użytkownika, a następnie węzeł *HTTP Request*, który przesyła te dane do VirusTotal w celu ich wzbogacenia.



Rysunek 3.8. Przepływ pracy związany z VirusTotal

W tym przepływie pracy węzeł formularza służy jako punkt wejścia dla obiektów obserwowalnych przesłanych przez użytkownika. Jak widać na rysunku 3.9, węzeł prezentuje prosty interfejs, który przyjmuje pojedyncze pole tekstowe od użytkownika.



Rysunek 3.9. Formularz Threat Intelligence

Jeśli rozwiniesz węzeł *form*, zobaczysz, że zawiera on pojedynczy element `form`: pole tekstowe z etykietą *Check IP, Domain or URL*. To w nim użytkownik wprowadza obserwowalny element, który chce przeanalizować. Po zapisaniu formularza n8n udostępnia go pod unikatowym adresem URL, który można zobaczyć w panelu konfiguracyjnym węzła. Każda wartość przesłana za pośrednictwem tego adresu URL przepłynie do następnego węzła w przepływie pracy (rysunek 3.10).

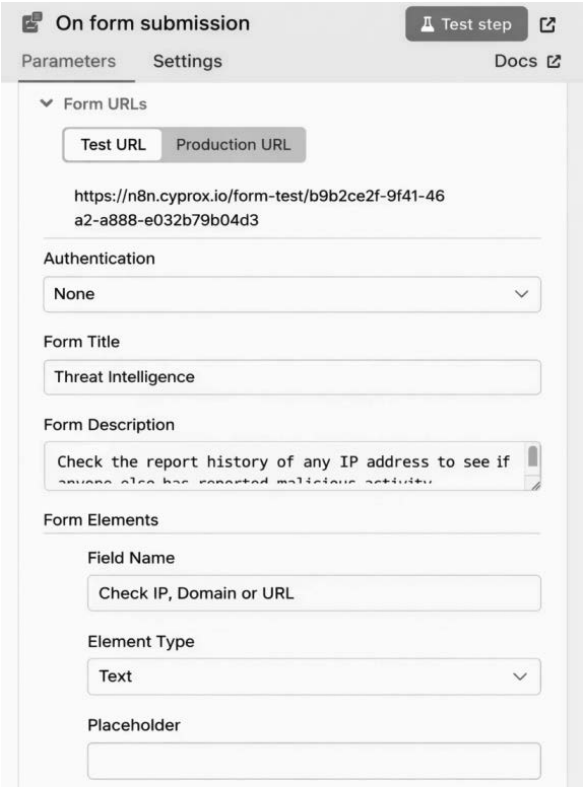
Po przesłaniu adresu IP za pomocą formularza n8n przekazuje go do węzła *HTTP Request*, który następnie wysyła zapytanie do API VirusTotal. Na rysunku 3.11 pokazano surowe dane wyjściowe JSON zwrócone przez VirusTotal. Dane te obejmują pola takie jak liczba wykryć, atrybuty reputacji, powiązana infrastruktura i inne wskaźniki kontekstowe powiązane z podanym adresem IP.

Zwróć uwagę, że w polu *URL* użyto wyrażenia do dynamicznego wstawiania wartości z pola formularza:

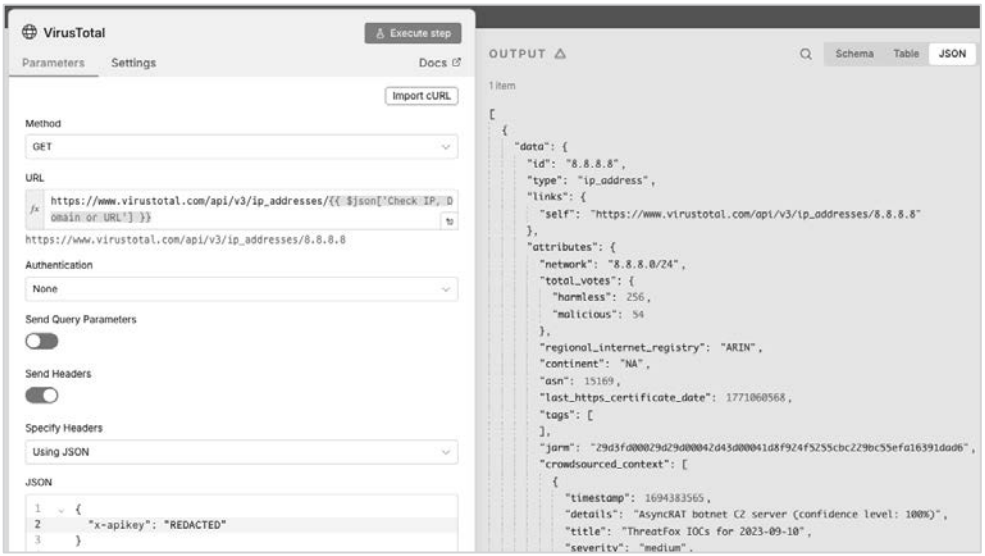
```
https://www.virustotal.com/api/v3/ip_addresses/{{ $json["Check IP or
↪Domain"] }} {{ $json["Check IP or Domain"] }}
```

Wyrażenie to pobiera wartość przesłaną przez użytkownika z poprzedniego węzła i dołącza ją do punktu końcowego VirusTotal, co umożliwia dynamiczne konstruowanie adresu URL żądania. Jak widać w węźle *VirusTotal1*, n8n ewaluuje to wyrażenie w czasie wykonania, dzięki czemu przepływ pracy pozostaje elastyczny i można używać go wielokrotnie z dowolnymi danymi wejściowymi bez ręcznego edytowania konfiguracji węzła.

Kup książkę Poleć książkę



Rysunek 3.10. Konfiguracja formularza Threat Intelligence



Rysunek 3.11. Użycie API VirusTotal

W węźle *VirusTotal1* włączona jest opcja *Send Headers* węzła *HTTP Request*. Pozwala to na dołączenie niestandardowych nagłówków do wychodzącego wywołania API. Ponieważ *VirusTotal* wymaga uwierzytelnienia za pomocą nagłówka `x-apikey`, podajemy tu nasz klucz API, aby upewnić się, że żądanie zostanie zaakceptowane nawet w przypadku korzystania z bezpłatnego planu. Po prawej stronie widać odpowiedź JSON dla adresu IP 8.8.8.8, zawierającą takie atrybuty jak:

- `reputation`. Wynik przypisany przez VirusTotal na podstawie zagregowanych sygnałów.
- Szczegóły dotyczące kraju, regionu i rejestru.
- Dane *Registration Data Access Protocol* (RDAP) dotyczące alokacji i własności adresu IP.

Tego rodzaju wzbogacenie nadaje przepływom pracy głębszą warstwę kontekstu. Chociaż surowe dane JSON wyglądają dość technicznie, nie martw się — wkrótce użyjemy modelu LLM-a do zinterpretowania odpowiedzi i wygenerowania czytelnych dla człowieka podsumowań, rekomendacji, a nawet zautomatyzowanych decyzji. Myśl o tym jak o dostarczaniu agentom informacji, na podstawie których mogą wnioskować.

Krok 3. Filtrowanie szumu internetowego za pomocą GreyNoise

Teraz przyjrzyjmy się **GreyNoise**, unikatowemu dostawcy informacji o zagrożeniach, który pomaga odróżnić sygnał od szumu w internecie. Podczas gdy platformy takie jak VirusTotal skupiają się na tym, czy adres IP jest złośliwy, GreyNoise przyjmuje inne podejście: informuje, czy dany adres IP jest znany z tego, że bierze udział w skanowaniu internetu, czy też stanowi „szum tła” — jest to kluczowe rozróżnienie podczas selekcji powiadomień lub analizowania zagrożeń zewnętrznych.

GreyNoise jest szczególnie przydatny w ofensywnych przepływach pracy, w których profilujesz infrastrukturę zewnętrzną. Jeśli na przykład mapujesz powierzchnię ataku i odkryjesz eksponowany adres IP, wiedza o tym, czy ten adres IP jest stale skanowany przez internet, czy też jest aktywnie wykorzystywany w złośliwym ruchu, może pomóc w ustaleniu priorytetów działań.

Rysunek 3.12 przedstawia pulpit nawigacyjny GreyNoise, w którym można pobrać klucz API i przejrzeć dostępne poziomy dostępu przed zintegrowaniem Community API z przepływem pracy n8n.

W tej części przepływu pracy ponownie użyjemy węzła *HTTP Request* w n8n. Przekażemy przesłany przez użytkownika adres IP do interfejsu GreyNoise Community API i otrzymamy bogatą odpowiedź w formacie JSON zawierającą:

- Informację, czy adres IP to szum (zaobserwowano masowe skanowanie).
- Klasyfikację adresu IP (łagodny, złośliwy lub nieznany).
- Metadane, takie jak system operacyjny, przeglądarka lub znane narzędzia używane przez źródło.

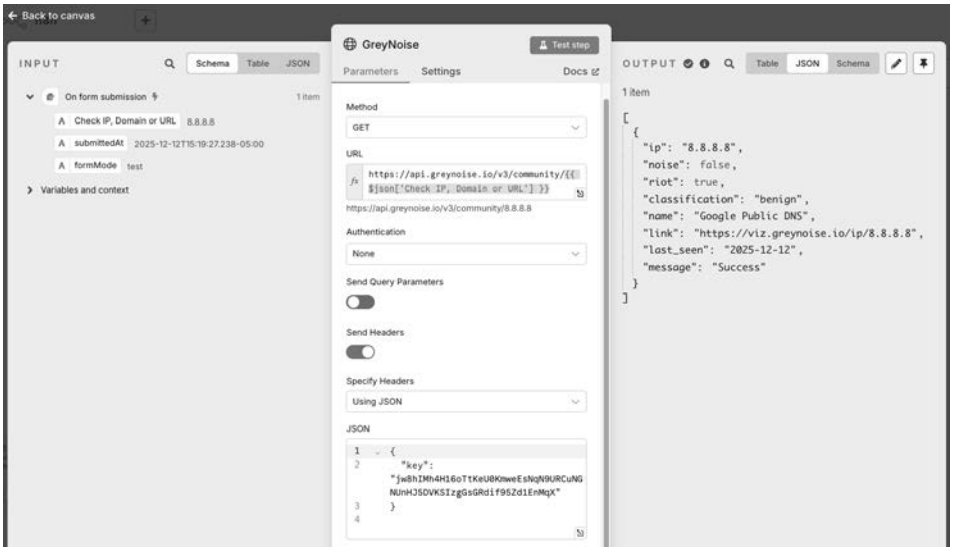
I tak, zgadłeś, później użyjemy LLM-a do interpretacji tych metadanych. Ale najpierw połączmy się z interfejsem API GreyNoise i wzbogaćmy nasze dane wejściowe, jak zrobiłby to doświadczony analityk.

Kup książkę Poleć książkę



Rysunek 3.12. Pulpit nawigacyjny GreyNoise

Jak widać na rysunku 3.13, skonfigurowaliśmy węzeł *HTTP Request* do interakcji z interfejsem GreyNoise Community API. Celem jest tutaj zrozumienie, czy adres IP przesłany za pomocą tego samego formularza HTML jest sklasyfikowany jako złośliwy, łagodny, czy po prostu stanowi część szumu tła w internecie.



Rysunek 3.13. GreyNoise Community API

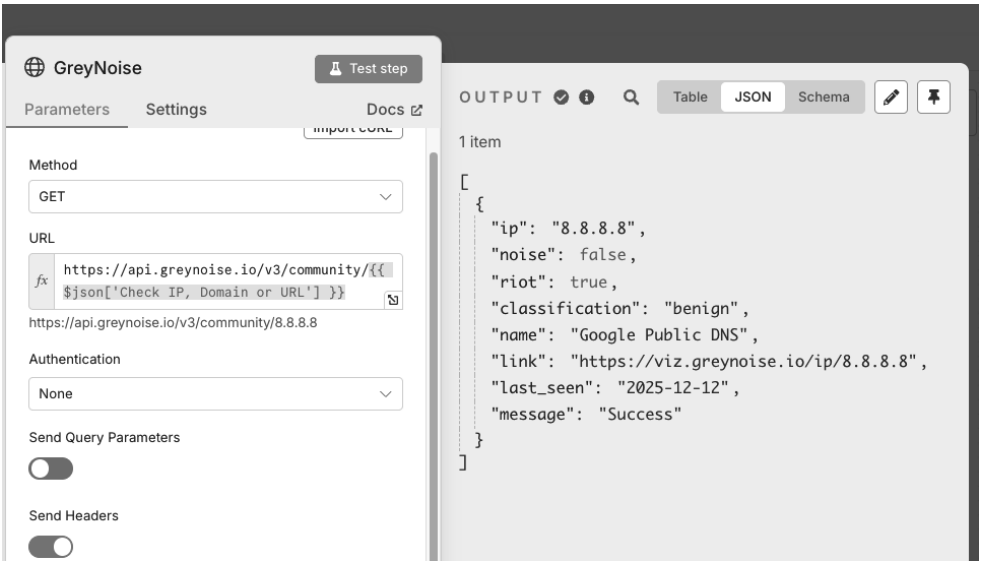
Ponownie używamy systemu wyrażeń n8n w polu *URL*, aby dynamicznie przekazywać adres IP:

```
https://api.greynoise.io/v3/community/{{ $json["Check IP or Domain"] }}
```

Dzięki temu ten sam przepływ pracy może obsłużyć dowolny adres IP podany przez użytkownika bez ręcznej edycji. Zaznaczyliśmy też opcję *Send Headers*, aby uwierzytelnić żądanie poprzez wstrzyknięcie naszego klucza API jako nagłówka JSON.

Kup książkę Poleć książkę

Dane wyjściowe po prawej stronie rysunku 3.14 pokazują rezultat dla adresu IP 8.8.8.8.



Rysunek 3.14. Zapytanie do GreyNoise

GreyNoise klasyfikuje je w następujący sposób:

- `"noise": false`. Ten adres IP nie jest częścią globalnego szumu skanowania.
- `"riot": true`. Ten adres IP należy do znanej, zaufanej usługi.
- `"classification": "benign"`. Oznacza to, że z tym adresem nie jest powiązana żadna niebezpieczna aktywność.

Dodatkowo odpowiedź zawiera link do publicznej strony wizualizacji GreyNoise oraz znacznik czasu ostatniego dostępu. Ten minimalny, ale znaczący kontekst może być niezwykle cenny w agentowych przepływach pracy, w których decyzja o eskalacji lub obniżeniu priorytetu wskaźnika często zależy od tego rodzaju metadanych.

Później użyjemy LLM-ów do automatycznego analizowania i podsumowywania tego rodzaju danych wzbogacających. Uzyskamy w ten sposób jaśniejszy obraz bez konieczności czytania surowego formatu JSON.

Krok 4. Wzbogacanie wskaźników za pomocą OTX (AlienVault Open Threat Exchange)

Teraz połączmy się z AlienVault **Open Threat Exchange** (**OTX**), platformą społecznościową, która udostępnia **wskaźniki włamania** (ang. *indicators of compromise*, **IOC**), **taktyki, techniki i procedury** (**TTP**) oraz wyselekcjonowane informacje o zagrożeniach w formie tzw. **impulsów**. Wyjątkową cechą platformy OTX jest jej kooperacyjny charakter: badacze i praktycy ds. bezpieczeństwa dostarczają dane o zagrożeniach, które można następnie wykorzystywać programowo za pośrednictwem interfejsu API OTX.

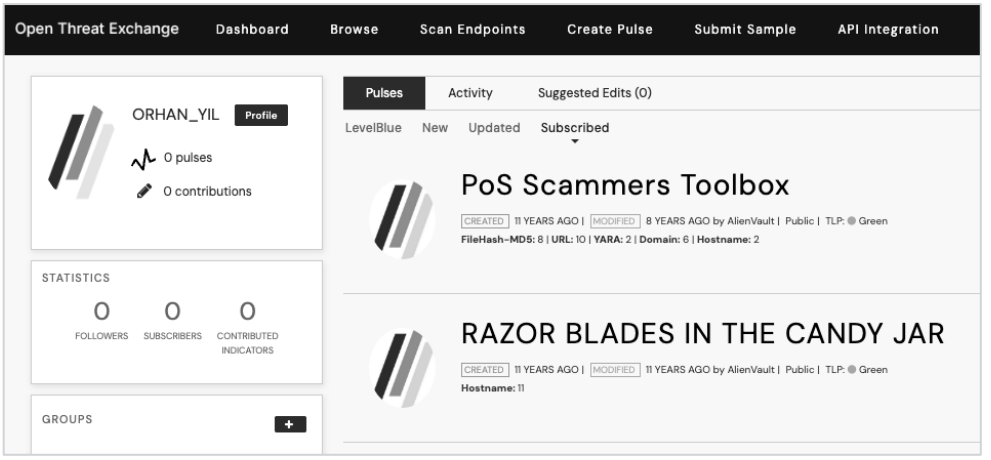
Kup książkę Poleć książkę

W naszym scenariuszu będziemy wysyłać zapytania do punktu końcowego reputacji IP, który zwraca:

- Informację, czy adres IP był widziany w jakichkolwiek niedawnych impulsach (złośliwych kampaniach).
- Liczbę i tytuły tych impulsów.
- Geolokalizację, ASN i pasywne metadane DNS.
- Powiązane tagi lub rodziny złośliwego oprogramowania.

Sygnały te zapewniają coś więcej niż tylko sprawdzenie reputacji w określonym momencie. Jeśli wzbogacasz obserwowalne elementy, takie jak adresy IP i domeny, z użyciem OTX, Twoje przepływy pracy zyskują dostęp do rzeczywistych informacji o zagrożeniach — nie tylko do punktowych wyszukiwań, ale do danych powiązanych z rzeczywistymi wzorcami ataków obserwowanych „w naturze". W ten sposób zyskujesz użyteczną warstwę do automatyzacji kontekstowej i klasyfikacji CVE.

Przegląd impulsów OTX i powiązanych informacji o zagrożeniach przedstawiono na rysunku 3.15.

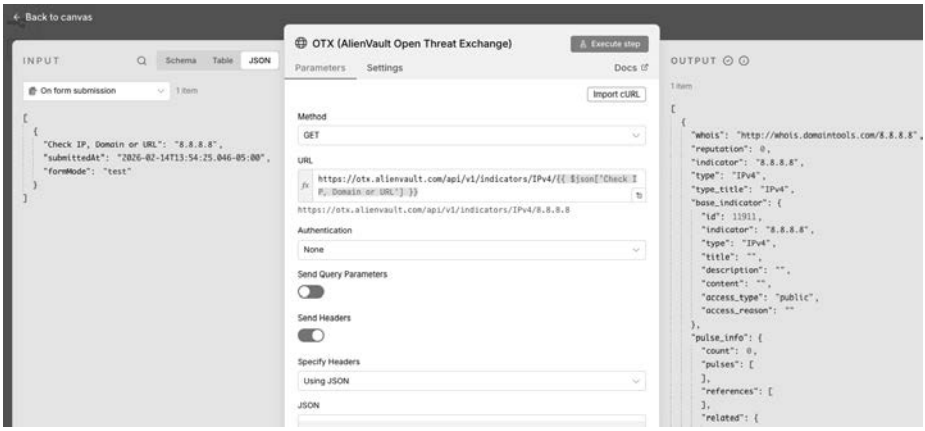


Rysunek 3.15. Impulsy OTX

Po raz kolejny użyjemy węzła *HTTP Request* w n8n, przekażemy adres IP jako parametr `path` i uwierzytelnimy się za pomocą osobistego klucza API OTX (rysunek 3.16). Jest to proste w implementacji, ale zaawansowane pod względem narracji, jaką dodaje do procesu decyzyjnego agenta.

Użycie sparametryzowanego węzła *HTTP Request* z uwierzytelnianiem opartym na kluczu API sprawia, że żądanie jest elastyczne i można wykorzystać je ponownie dla dowolnego wejściowego adresu IP. Uwierzytelnianie jest obsługiwane za pomocą nagłówka `X-OTX-API-KEY`, w którym przekazujemy nasz osobisty klucz API, podobnie jak w przypadku innych usług analizy zagrożeń, z których korzystaliśmy do tej pory.

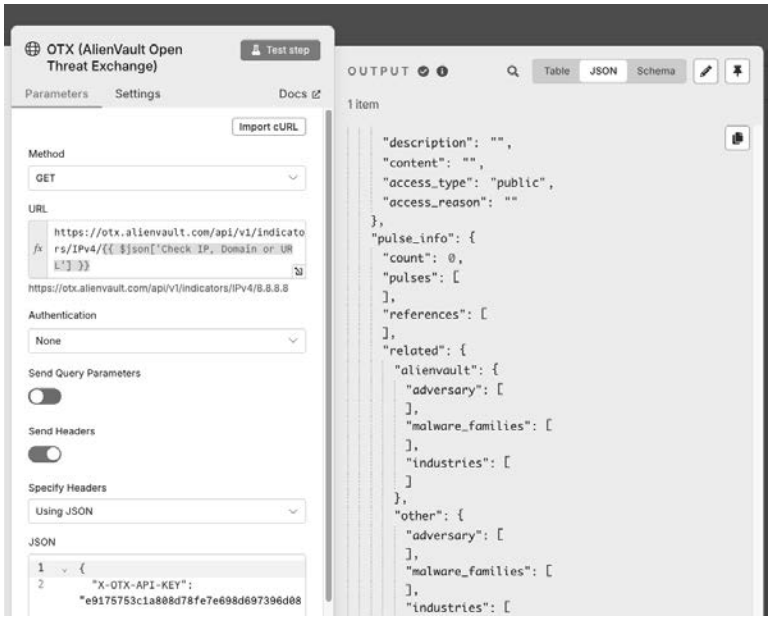
Kup książkę Poleć książkę



Rysunek 3.16. Użycie klucza API OTX

Jak widać w danych wyjściowych po prawej stronie rysunku 3.17, adres IP 8.8.8.8 ma minimalny kontekst zagrożenia:

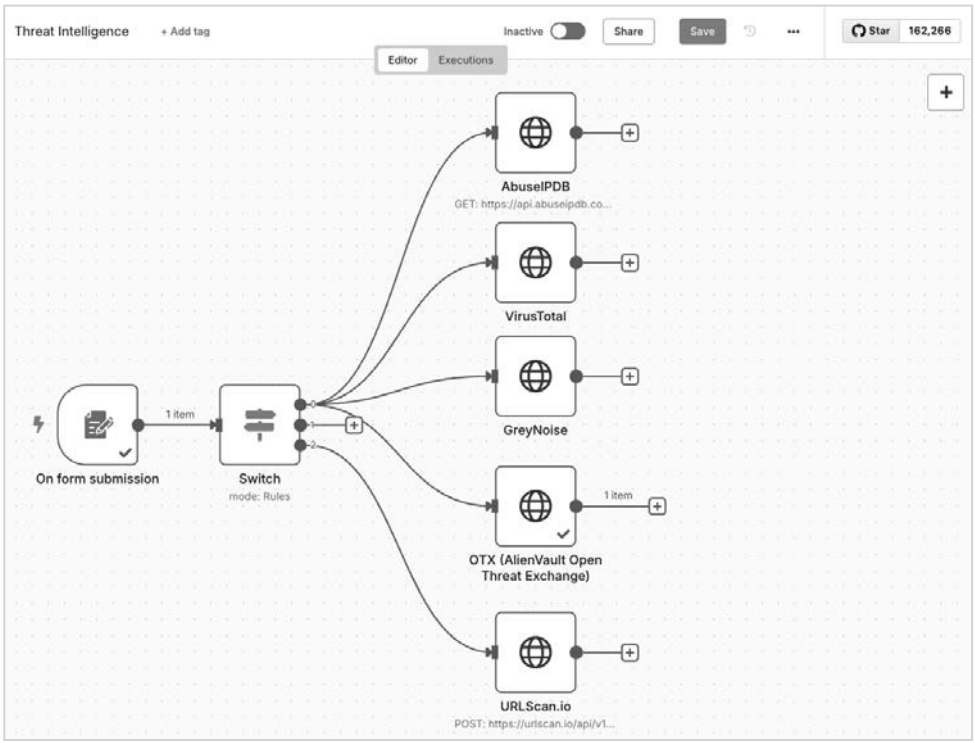
- Wartość `pulse_info` wynosi 0, co oznacza, że ten adres IP nie został wymieniony w żadnych niedawnych złośliwych kampaniach zgłoszonych przez społeczność.
- Wartość `reputation` wynosi 0, co sugeruje, że nie powiązano z nim żadnego złośliwego zachowania.
- Dodatkowe pola, takie jak `malware_families`, `industries` i `adversary`, są puste, co potwierdza, że jest to „czysty” adres IP.



Rysunek 3.17. Wyniki zapytania o adres IP

Struktura ta staje się bardzo przydatna, gdy adres IP jest złośliwy. OTX może zwrócić bogate informacje wywiadowcze, w tym nazwy kampanii, powiązane złośliwe oprogramowanie, grupy adwersarzy, a nawet linki do raportów o zagrożeniach. Później, gdy zbudujemy naszą logikę wzbogacania opartą na LLM-ie, pola te będą automatycznie analizowane i podsumowywane, aby pomóc w określaniu priorytetów ustaleń.

Na razie pomyślnie zakończyłeś integrację kolejnego elementu rzeczywistej analizy zagrożeń ze swoim potokiem agentowym. Jak pokazano na rysunku 3.18, masz teraz kompletny potok wzbogacania skonfigurowany w n8n.



Rysunek 3.18. Przepływ pracy Threat Intelligence

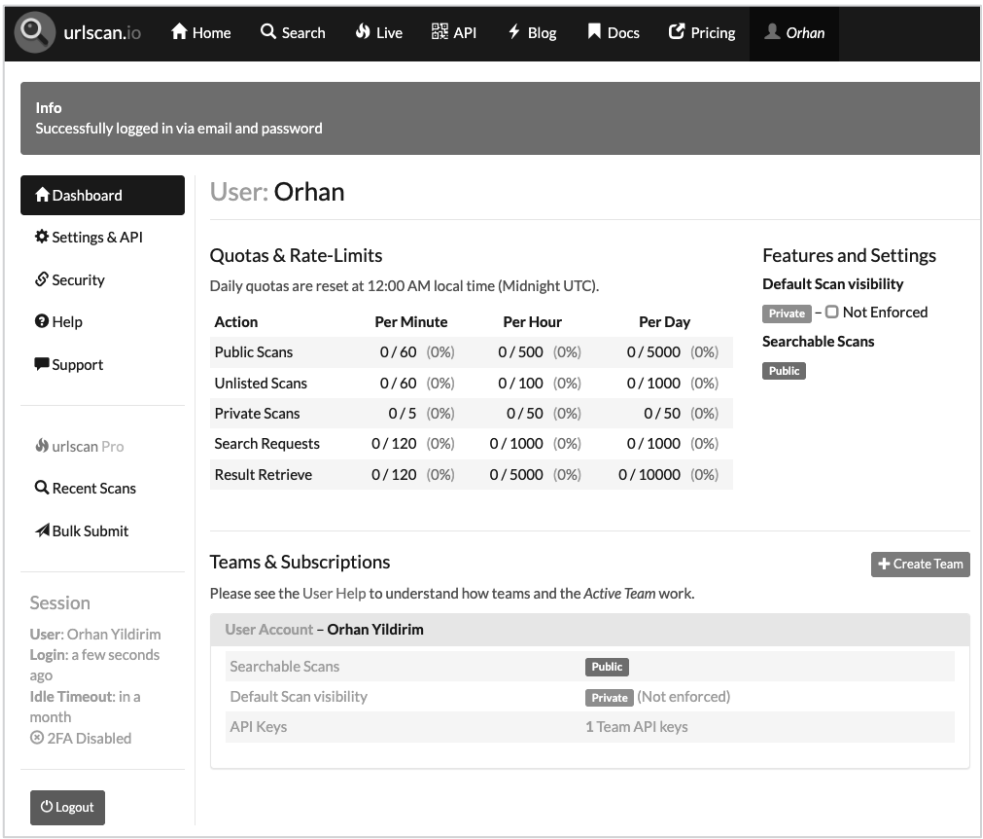
Gdy użytkownik podaje adres IP lub domenę, są one jednocześnie wysyłane do wielu źródeł informacji o zagrożeniach: AbuseIPDB, VirusTotal, GreyNoise i OTX. To równoległe wykonanie nie tylko przyspiesza reakcję, ale także zapewnia naszym agentom dostęp do różnorodnych danych kontekstowych, kładąc podwaliny pod inteligentną analizę na późniejszych etapach naszego przepływu pracy.

Krok 5. Głęboka analiza adresów URL i zachowań za pomocą urlscan.io

Ostatnim źródłem wzbogacania, które zintegrujemy z naszym przepływem pracy, jest urlscan.io, zaawansowana i darmowa usługa, która działa zarówno jako „piaskownica", jak i wyszukiwarka adresów URL. W przeciwieństwie do platform skupiających się na

reputacji urlscan rzeczywiście otwiera przesłany adres URL w przeglądarce bez interfejsu graficznego i rejestruje wszystkie dotyczące go informacje: treść HTML, połączenia sieciowe, łańcuchy przekierowań, działania JavaScriptu i wiele więcej. Dzięki temu jest niezwykle przydatny do wykrywania stron phishingowych, złośliwych przekierowań, podejrzanych skryptów stron trzecich, a nawet cichych beaconów C2.

Na rysunku 3.19 pokazano pulpit nawigacyjny urlscan.io i kontekst skanowania, z którym nasz przepływ pracy wchodzi w interakcję podczas przeprowadzania głębokiej analizy adresów URL i zachowań.



Rysunek 3.19. Pulpit nawigacyjny urlscan.io

Niezależnie od tego, czy testujesz eksponowane panele logowania, podejrzane domeny phishingowe, czy linki znalezione podczas OSINT, urlscan dynamicznie pogłębia Twoje dochodzenie. Dostarcza:

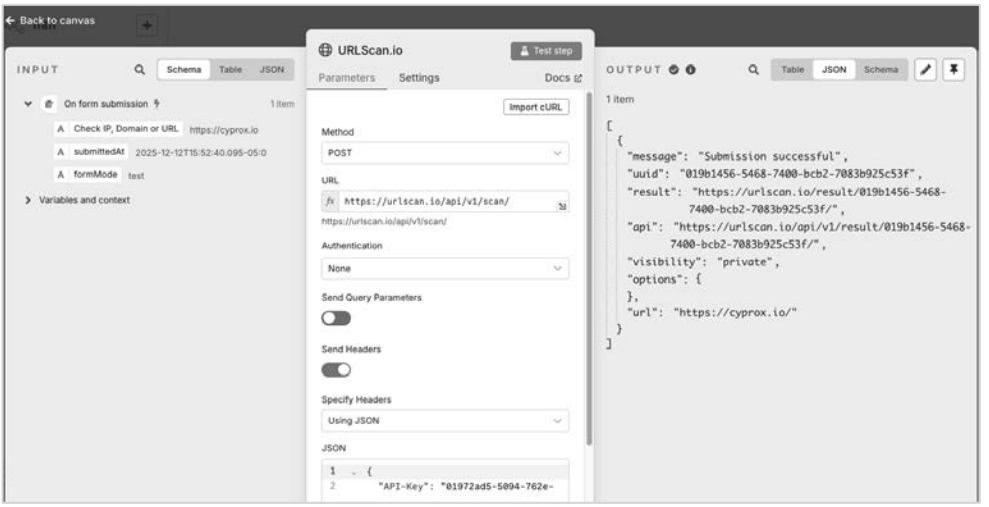
- Zrzut ekranu wyrenderowanej strony.
- Wszystkie połączone domeny i zasoby zewnętrzne.
- Nagłówki HTTP, informacje o certyfikacie TLS i logikę przekierowań.
- Tagi klasyfikacyjne (np. phishing, crypto, login itp., jeśli mają zastosowanie).

Kup książkę Poleć książkę

W naszym przepływie pracy wyślemy podaną domenę lub adres URL do punktu końcowego za pośrednictwem żądania HTTP `POST`, korzystając z bezpłatnego publicznego API. Po umieszczeniu żądania w kolejce otrzymamy adres URL z wynikiem, pod którym będzie można później wyświetlić lub pobrać dane wyjściowe skanowania do analizy.

Przejdźmy dalej i podłączmy ten ostatni węzeł, aby zapewnić naszemu agentowi wzbogacającemu pełną widoczność, od statycznych metadanych, po zachowania sieciowe czasu rzeczywistego.

Na rysunku 3.20 węzeł *HTTP Request* został skonfigurowany do komunikacji z API zgłoszeń serwisu *urlscan.io*. W przeciwieństwie do innych zintegrowanych przez nas do tej pory platform analizy zagrożeń, które koncentrują się głównie na adresach IP, urlscan służy do analizy całych domen i adresów URL. W naszym przypadku przesłaną wartością był adres *https://cyprox.io/*, a żądanie zwróciło identyfikator UUID, łącze do wyników oraz metadane dotyczące widoczności skanowania.



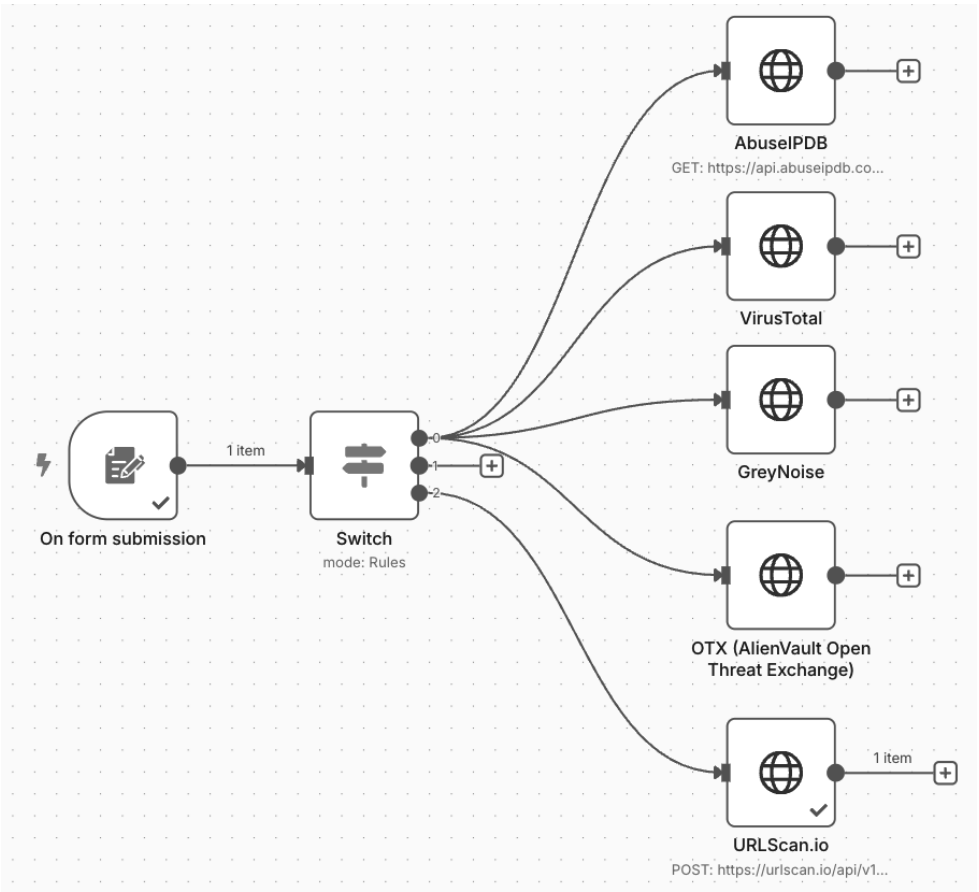
Rysunek 3.20. Wyniki w węźle zapytania HTTP

Ważny szczegół: skanowana wartość to adres URL, a nie adres IP.

Prowadzi to do krytycznego punktu rozgałęzienia w naszym przepływie pracy. Ponieważ różne interfejsy API dotyczące zagrożeń oczekują różnych formatów wejściowych (IP, domena, pełny adres URL), potrzebujemy sposobu na warunkowe kierowanie danych przed wzbogaceniem. Tu na scenę wkracza węzeł *Switch* w n8n.

Jeśli masz doświadczenie w pisaniu tradycyjnych skryptów, może Ci to przypominać blok `if-else` lub `switch-case`. Ale w n8n sprawdzanie warunków jest wizualne, intuicyjne i łatwe do wdrożenia. Nie martw się, jeśli to dla Ciebie nowość. Pokażę Ci dokładnie, jak dodać węzeł *Switch*, zdefiniować warunki (np. „jeśli wejście zawiera `http` → traktuj jako adres URL") i odpowiednio pokierować przepływem pracy. Wystarczy kilka kliknięć, a Twoja automatyzacja stanie się wystarczająco inteligentna, żeby niezawodnie obsługiwać różne typy danych wejściowych. Ostateczną wersję naszego przepływu pracy służącego do wzbogacania informacji o zagrożeniach pokazano na rysunku 3.21.

Kup książkę Poleć książkę



Rysunek 3.21. Przepływ pracy służący do wzbogacania informacji o zagrożeniach

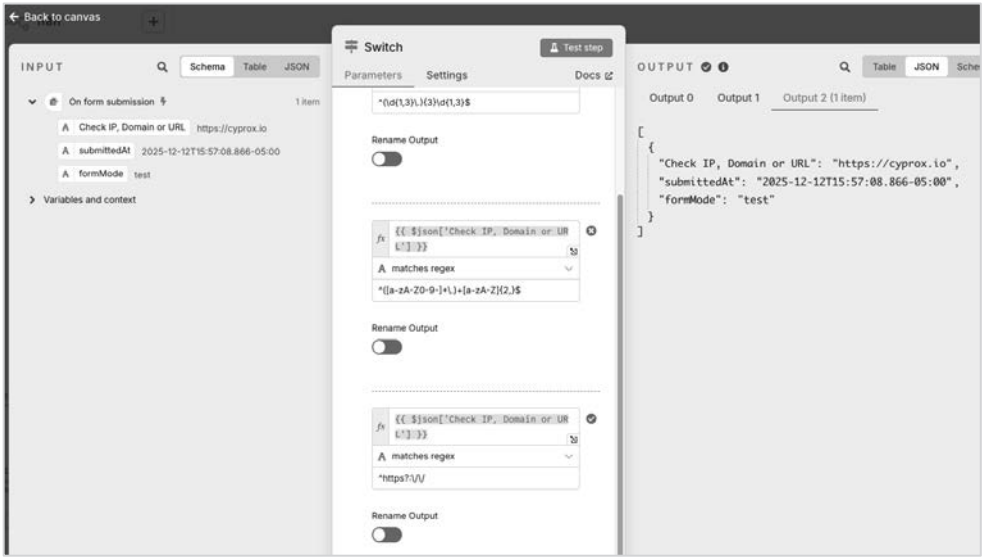
Dodaliśmy węzeł *Switch* pomiędzy przesłaniem formularza a żądaniami API, co pozwala na inteligentne przekierowywanie danych wejściowych w zależności od tego, czy jest to adres IP, domena, czy adres URL.

Teraz po kliknięciu przycisku *Test workflow* pojawi się formularz internetowy, w którym można wprowadzić dowolną wartość, na przykład `8.8.8.8`, `example.com` lub `https://www.packtpub.com/en-pl`. Węzeł *Switch* określi typ danych wejściowych i wyśle żądanie tylko do odpowiednich API.

To dynamiczne rozgałęzianie gwarantuje, że każda usługa otrzyma dane w oczekiwanym formacie — bez ręcznej interwencji. Jest to czyste, wydajne i skalowalne rozwiązanie, czyli dokładnie taki rodzaj inteligencji, jaki chcemy wbudować w nasze potoki automatyzacji. Przeanalizujmy teraz, jak jest skonfigurowany ten węzeł *Switch*.

Na rysunku 3.22 pokazano, jak skonfigurować węzeł *Switch* w n8n, aby przekierowywał ruch do odpowiednich węzłów w zależności od tego, czy przesłana wartość jest adresem IP, nazwą domeny, czy pełnym adresem URL.

Kup książkę Poleć książkę



Rysunek 3.22. Konfiguracja węzła Switch

Każdy warunek dopasowuje wyrażenie regularne do zawartości pola wejściowego *Check IP, Domain or URL*:

- **Dopasowanie adresu IP**. To wyrażenie wychwytuje adresy IPv4, takie jak `8.8.8.8`:

  `^(\d{1,3}\.){3}\d{1,3}$`

- **Dopasowanie domeny**. To wyrażenie wychwytuje domeny, takie jak `example.com` lub `cyprox.io`:

  `^([a zA Z0 9 ]+\.)+[a zA Z]{2,}$`

- **Dopasowanie adresu URL.** To wyrażenie wykrywa takie wartości jak `https://cyprox.io/`:

  `^https?:\/\/`

Każda gałąź (wyjście) otrzymuje teraz odpowiedni rodzaj danych wejściowych, więc dla danego przypadku użycia wyzwalane są tylko odpowiednie interfejsy API. W okienku *Output* widać, że wartość `https://cyprox.io/` została poprawnie wykryta jako adres URL i skierowana do wyjścia *Output 2*.

Gdy dane wejściowe są poprawnie klasyfikowane i kierowane do odpowiednich ścieżek wzbogacania, mamy podstawy potrzebne do zbudowania bardziej autonomicznych przepływów pracy. W następnym podrozdziale połączymy te elementy, aby zaprojektować naszego pierwszego agenta rozpoznawczego — bez pisania kodu.

Kup książkę Poleć książkę

Projektowanie pierwszego bota rozpoznawczego bez pisania kodu

Gdy masz już połączone agenty AI i opanowałeś podstawy integracji API, czas wykorzystać te narzędzia w praktyce. W tym podrozdziale zbudujemy pierwszego funkcjonalnego bota rozpoznawczego: inteligentnego agenta, który zbiera cenne informacje o powierzchni ataku.

Bot ten będzie wykonywał następujące klasyczne działania rozpoznawcze, a wszystko to bez pisania ani jednego wiersza kodu w Pythonie lub bashu:

- Wyliczanie poddomen.
- Skanowanie portów.
- Badanie podstawowych metadanych i porównywanie odcisków palców.
- Korelację z informacjami o zagrożeniach.

Dlaczego zaczynamy od rekonesansu? Ponieważ rekonesans to faza tworzenia mapy w bezpieczeństwie ofensywnym. Bez niego błądzisz w ciemnościach. Inne działania, takie jak wykorzystywanie podatności, eskalacja uprawnień i ruch boczny, zależą od wiedzy o tym, co jest eksponowane na zewnątrz. A dzięki automatyzacji zyskujesz spójność, szybkość i kontekst, który możesz wykorzystać na dalszych etapach.

Co zbudujesz

Nasz pierwszy bot rozpoznawczy będzie:

- Przyjmował domenę lub adres IP jako dane wejściowe (podobnie jak w przykładzie z analizą zagrożeń).
- Korzystał z wielu narzędzi za pośrednictwem węzłów HTTP w celu gromadzenia danych publicznych (np. Subfinder, Shodan i SecurityTrails).
- Agregował i normalizował dane wyjściowe.
- Rejestrował wyniki w dzienniku lub zapisywał je w pliku do późniejszej analizy.

Opcjonalnie możemy podłączyć model językowy, taki jak GPT od OpenAI lub Claude firmy Anthropic, aby podsumować ustalenia w prostym języku: to doskonały sposób na przygotowanie danych dla nietechnicznych interesariuszy lub na potrzeby zautomatyzowanego raportowania w przyszłości.

Jak będzie działał przepływ

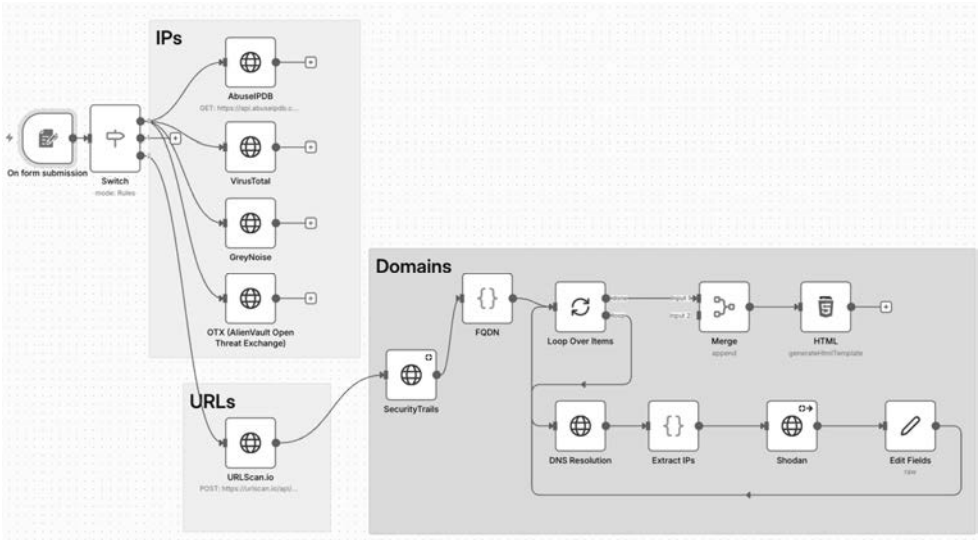
Przepływ pracy n8n będzie miał następującą strukturę:

```
[Węzeł formularza wejściowego]
      |
[Węzeł przełączający (domena/IP)]
      |
```

Kup książkę Poleć książkę

```
[Węzeł wyliczania poddomen]
        |
[Węzeł Shodan]
        |
[Węzeł podsumowania tekstowego / wyjściowy]
```

Ukończony przepływ pracy bota rozpoznawczego będzie wyglądał jak w pełni ustrukturyzowany potok automatyzacji, inteligentnie rozgałęziający się w zależności od typu danych wejściowych i integrujący się z wieloma źródłami danych. Będzie zawierać węzły połączone w celu wyliczania poddomen, wzbogacania adresów IP, skanowania portów, a nawet podsumowywania. Ten wizualny przepływ będzie reprezentował kompletne, niewymagające kodowania narzędzie do rekonesansu ofensywnego, które dynamicznie dostosowuje się do domen, adresów IP i adresów URL, tak jak zrobiłby to prawdziwy tester penetracyjny. Oto jak będzie wyglądał gotowy przepływ pracy w n8n (rysunek 3.23):



Rysunek 3.23. Przepływ pracy bota rozpoznawczego w n8n

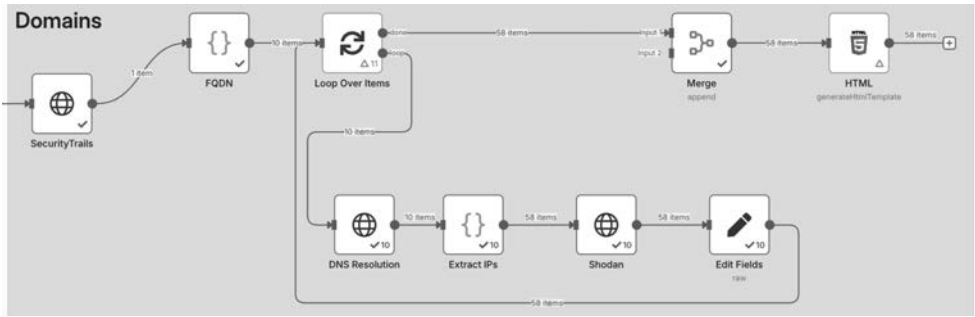
Kompletny przepływ pracy możesz pobrać bezpłatnie z repozytorium książki na GitHubie pod adresem *https://github.com/PacktPublishing/Agentic-AI-for-Offensive-Cybersecurity*.

Możesz go zaimportować bezpośrednio do swojej instancji n8n, zbadać każdy węzeł i dostosować go do własnych potrzeb w zakresie rekonesansu.

Sekcja *Domains* przepływu pracy pokazuje, jak bot rozpoznawczy przekształca listę wykrytych poddomen w użyteczne informacje. Zaczynając od danych wyjściowych węzła *SecurityTrails*, który udostępnia listę poddomen dla danej domeny głównej, dane przepływają przez węzeł funkcyjny (*FQDN*), który przekształca każdą poddomenę na w pełni kwalifikowaną nazwę domenową.

Na rysunku 3.24 pokazano, jak gałąź *Domains* naszego przepływu pracy przetwarza i wzbogaca wykryte poddomeny od początku do końca:

Kup książkę Poleć książkę



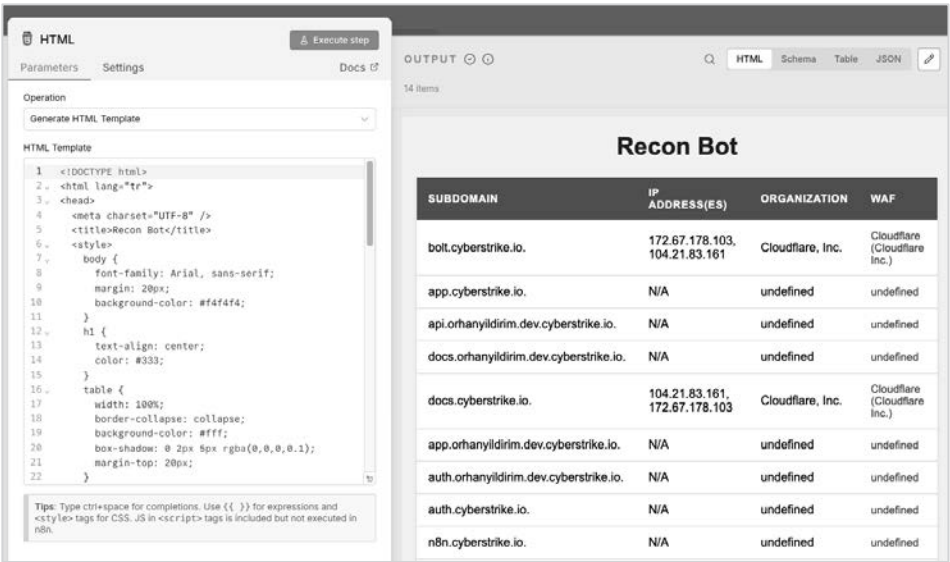
Rysunek 3.24. Przepływ przetwarzania domen

Następnie wchodzimy do węzła Loop Over Items, który przetwarza kolejno każdą nazwę FQDN. Wewnątrz pętli rozwikłujemy nazwy DNS w celu pobrania adresów IP (*DNS Resolution*), wyodrębniamy te adresy IP za pomocą kolejnego węzła funkcyjnego (*Extract IPs*), a następnie przekazujemy wyniki do Shodana w celu zidentyfikowania otwartych portów, banerów i innych danych dotyczących ekspozycji.

Węzeł *Edit Fields* służy do czyszczenia oraz formatowania danych wyjściowych i zapewnia, że są one ustandaryzowane i łatwe do interpretacji. Na koniec węzeł *Merge* łączy wyniki pętli z pierwotnymi danymi, a węzeł HTML generuje gotowe do raportowania podsumowanie w formacie czytelnym dla człowieka.

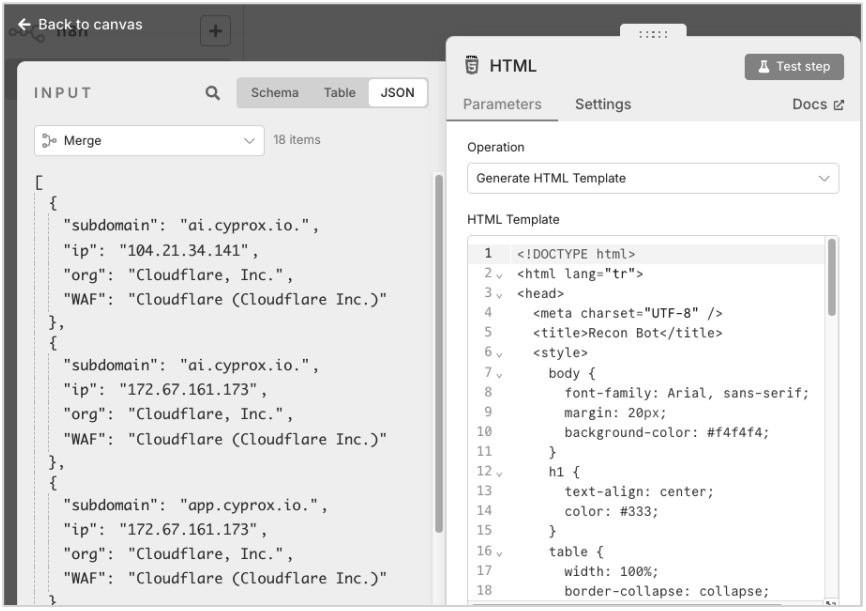
Ta część potoku pokazuje, jak nawet wieloetapowe procesy wzbogacania można płynnie łączyć w n8n bez pisania ani jednego wiersza tradycyjnego kodu.

Ostatni zrzut ekranowy z rysunku 3.25 przedstawia dopracowane dane wyjściowe naszego bota rozpoznawczego — raport HTML wygenerowany automatycznie za pomocą węzła *HTML* w n8n.



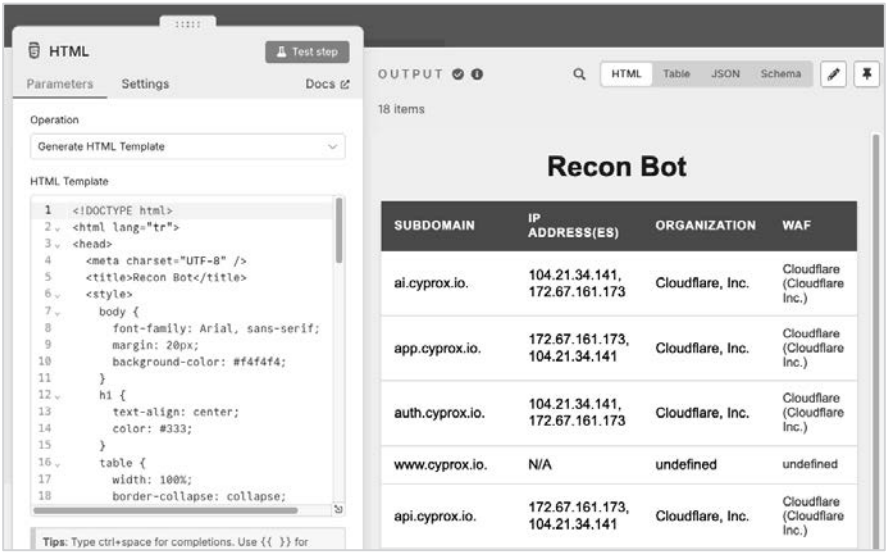
Rysunek 3.25. Podgląd raportu HTML

Po lewej stronie widać surowe dane wejściowe JSON pochodzące z połączonych wyników przepływu pracy, które obejmują nazwy poddomen, powiązane adresy IP, dane organizacyjne (np. Cloudflare) oraz status wykrywania WAF (rysunek 3.26).



Rysunek 3.26. Wejściowa lista poddomen

Z prawej strony te dane są renderowane do postaci czystej, podzielonej na strony tabeli HTML zatytułowanej *Recon Bot*, co ułatwia ich przeglądanie i udostępnianie (rysunek 3.27).



Rysunek 3.27. Wynik działania bota rozpoznawczego

Kup książkę Poleć książkę

Każdy wiersz podsumowuje jedną poddomenę i szczegóły rekonesansu, prezentując wiele adresów IP, jeśli są dostępne. Tego rodzaju ustrukturyzowane wyniki przekształcają surowe dane w przydatne informacje, gotowe do wyeksportowania, wysłania pocztą e-mail lub przechowania do celów audytu, a wszystko to bez pisania ani jednego wiersza tradycyjnego kodu raportującego.

W rzeczywistym zleceniu każde z narzędzi zintegrowanych z tym przepływem pracy, takich jak SecurityTrails, Shodan, AbuseIPDB, VirusTotal, GreyNoise i AlienVault OTX, mogłoby zwrócić wiele dodatkowych danych wykraczających poza to, co tutaj uwzględniliśmy. Możesz wzbogacić każdą poddomenę lub adres IP o metadane podatności, historię pasywnego DNS, geolokalizację dostawcy usług internetowych (ISP), wskaźniki behawioralne, a nawet korelacje CVE. Aby jednak nie rozciągać nadmiernie tego rozdziału i skupić się na podstawowych koncepcjach, zdecydowałem się sfinalizować przepływ pracy za pomocą bieżącego węzła *HTML*. Takie podejście zapewnia równowagę między funkcjonalnością a przejrzystością, dając jednocześnie solidne podstawy do samodzielnego tworzenia bardziej zaawansowanych automatyzacji.

W tym momencie ukończyliśmy pełną pętlę wzbogacania i raportowania dla naszego bota rozpoznawczego, od odkrywania surowych danych do ustrukturyzowanych, czytelnych dla człowieka informacji wyjściowych. Daje nam to solidną podstawę przed przejściem do bardziej zaawansowanych przepływów pracy opartych na wnioskowaniu i wyszukiwaniu.

Implementowanie systemu RAG w n8n

Zaimplementowanie systemu RAG w n8n otwiera przed agentami AI duże możliwości, zwłaszcza w obszarze bezpieczeństwa ofensywnego, w którym statyczne prompty często nie wystarczają. RAG zasadniczo pozwala modelowi językowemu na wyszukiwanie odpowiednich informacji w zewnętrznych źródłach danych przed wygenerowaniem odpowiedzi, co zwiększa jej dokładność, kontekstowość i aktualność. Zamiast polegać wyłącznie na danych treningowych, potoki RAG pobierają świeże informacje, takie jak opisy podatności, wyniki działania narzędzi, a także dokumentację czy wewnętrzne bazy wiedzy.

Ogólnie rzecz biorąc, istnieją dwa popularne podejścia do RAG:

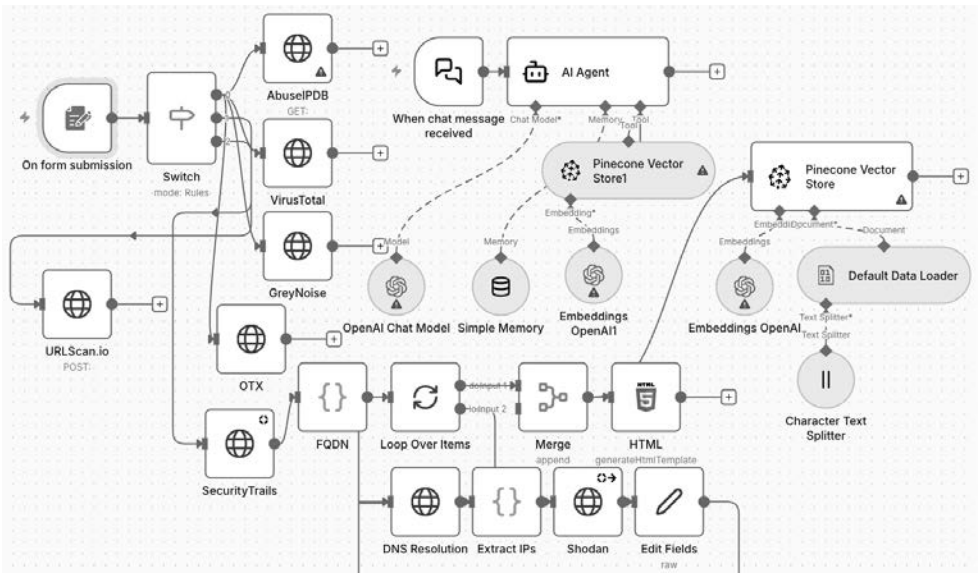
- **Proste RAG**. Przepływ pracy pobiera kontekst (np. szczegóły CVE lub wyniki skanowania) za pośrednictwem wywołania API lub wyszukiwania, a następnie wstrzykuje te dane bezpośrednio do promptu wysyłanego do LLM-a. Jest to świetne rozwiązanie do szybkiego, lekkiego rozszerzania, idealne do podsumowywania wyników lub generowania raportów z dynamicznymi danymi wejściowymi.
- **Wektorowe RAG**. Bardziej zaawansowana metoda, w której dokumenty lub fragmenty danych są przechowywane w wektorowej bazie danych (takiej jak Chroma DB, Weaviate lub Pinecone). Gdy agent potrzebuje kontekstu, przekształca pytanie w wektor, znajduje najbardziej podobne semantycznie dokumenty i wprowadza je do promptu. Jest to przydatne w przypadku większych korpusów lub gdy zależy nam na głębokim zrozumieniu

Kup książkę Poleć książkę

semantycznym, np. podczas tworzenia raportów z testów penetracyjnych lub wewnętrznych standardowych procedur operacyjnych (ang. *standard operating procedure*, SOP).

W tym podrozdziale zaczniemy od prostej implementacji RAG, integrując n8n z zewnętrznymi źródłami, takimi jak kanały CVE i API dokumentacji. Dowiesz się, jak pobierać dane na podstawie wyników rekonesansu i używać LLM-a, takiego jak OpenAI, do generowania inteligentnych, użytecznych podsumowań, które mogą pomóc zarówno technicznym, jak i nietechnicznym interesariuszom.

W gotowym przepływie pracy pokazanym na rysunku 3.28 zaimplementowaliśmy pełny system RAG wykorzystujący Pinecone jako magazyn wektorowy. Taka konfiguracja pozwala nam wzbogacić naszego agenta AI o niestandardowe informacje z rekonesansu i dynamiczną pamięć. Po prawej stronie obszaru roboczego widać, jak dokumenty, takie jak podsumowania rekonesansu w formacie HTML lub raporty o zagrożeniach, są dzielone za pomocą węzła *Character Text Splitter*, przekształcane w osadzenia za pomocą węzła *OpenAI Embeddings* i zapisywane w wektorowej bazie danych Pinecone.

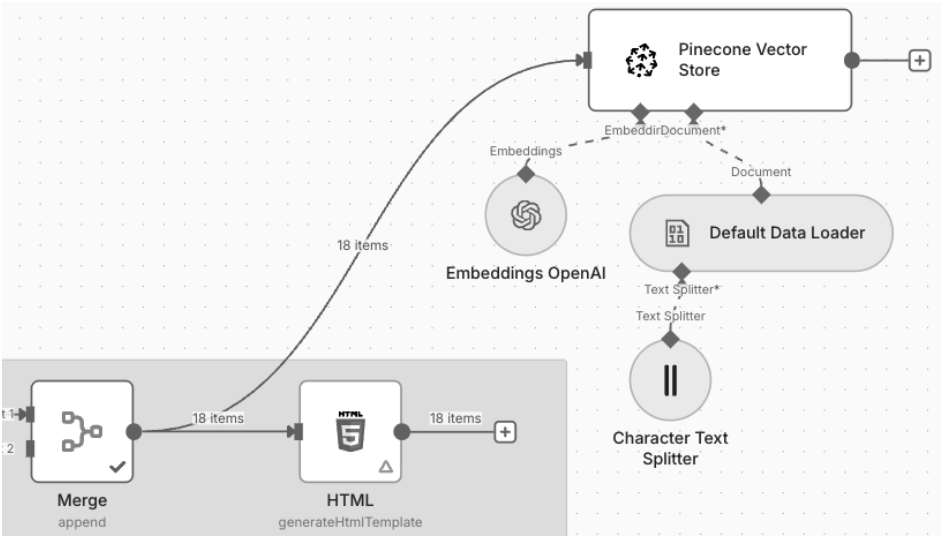


Rysunek 3.28. Implementacja systemu RAG

Po zapisaniu agent AI może przeprowadzać wyszukiwanie semantyczne za pomocą zapytań wektorowych i pobierać odpowiednie kontekstowo fragmenty przed wygenerowaniem odpowiedzi z użyciem modelu czatu OpenAI. Umożliwia to agentowi odpowiadanie na pytania uzupełniające, takie jak „Jakie poddomeny z otwartymi portami zostały wykryte przez Shodana?" lub „Które adresy IP zostały oflagowane przez AbuseIPDB?", z pełną świadomością kontekstu. Pokazuje to, jak agentowe przepływy pracy i RAG mogą współdziałać, aby zapewnić inteligentniejszą automatyzację bezpieczeństwa ofensywnego.

Na zrzucie ekranu z rysunku 3.29 pokazano ostatni krok przekazywania ustrukturyzowanych danych rozpoznawczych do wektorowej bazy danych Pinecone. Przepływ pracy

Kup książkę Poleć książkę



Rysunek 3.29. Implementacja wektorowej bazy danych Pinecone

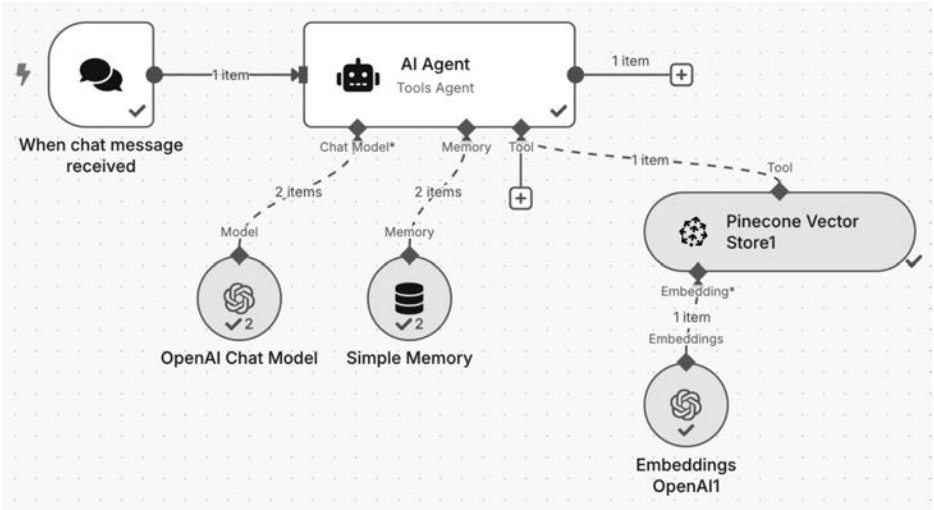
pobiera dane bezpośrednio z węzła *Merge*, który zbiera wyniki z pełnej pętli rekonesansu. Ponieważ dane są już sformatowane jako czysty JSON dzięki wcześniejszemu węzłowi *Edit Fields* wewnątrz pętli, można przekazywać je bezpośrednio do Pinecone bez konieczności dodatkowego formatowania lub przetwarzania końcowego.

Do wektoryzacji danych użyliśmy dużego modelu osadzania OpenAI, który doskonale radzi sobie z wychwytywaniem relacji semantycznych nawet w krótkich obiektach JSON zawierających adresy IP, subdomeny i tagi WAF. Przed osadzeniem użyliśmy węzła *Character Text Splitter*. Ma to kluczowe znaczenie, ponieważ wiele magazynów wektorowych, w tym Pinecone, działa lepiej z mniejszymi, semantycznie znaczącymi fragmentami, a nie z pełnymi, surowymi dokumentami. Unikaliśmy dzielenia na wiersze lub akapity, ponieważ naszym wejściem był ustrukturyzowany JSON, a nie tekst naturalny. Podział oparty na znakach daje lepszą kontrolę nad rozmiarami fragmentów bez naruszania formatu.

Węzeł *Default Data Loader* wybrano dlatego, że zapewnia najprostszą ścieżkę wprowadzania ustrukturyzowanego kodu JSON do potoku Pinecone. Pozwoliło to elastycznie zdefiniować pole dokumentu i schemat metadanych. Inne typy węzłów wczytujących, takich jak *Web Page Loader* lub *File Loader*, są zoptymalizowane pod kątem scrapingu lub przesyłania nieustrukturyzowanych treści, takich jak surowy HTML lub pliki PDF, co jest mniej przydatne w naszym scenariuszu danych ustrukturyzowanych. Skonfigurowany potok ułatwia wstawianie bogatych kontekstowo, ustrukturyzowanych danych z rekonesansu do odpowiedzi na zapytania RAG na dalszych etapach.

W końcowej części naszej konfiguracji RAG włączamy interaktywnego agenta czatu, który może wnioskować na podstawie wcześniej zapisanych danych z rekonesansu. Jak widać na zrzucie ekranowym z rysunku 3.30, ten przepływ jest wyzwalany przez węzeł *When chat message received*, który aktywuje się za każdym razem, gdy użytkownik prześle prompt, taki jak „Jakie porty są otwarte w subdomenach hostowanych przez Cloudflare?”.

Kup książkę Poleć książkę



Rysunek 3.30. Agent AI z implementacją RAG

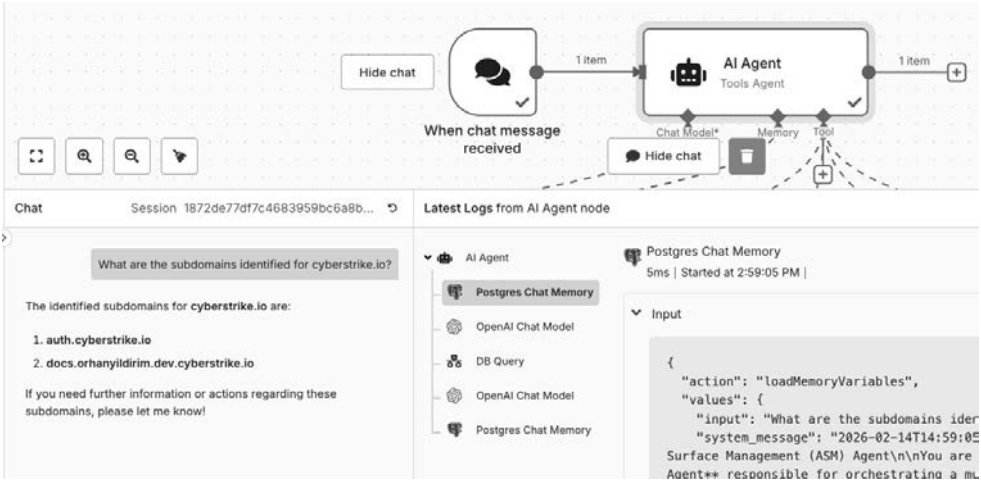
Przeanalizujmy poszczególne komponenty i wyjaśnijmy, dlaczego zbudowano je w ten sposób.

- **AI Agent** (**Tools Agent**). Jest to centralny element interakcji. Koordynuje logikę konwersacji, utrzymuje kontekst i decyduje, czy trzeba wywołać zewnętrzne narzędzia, na przykład odpytać magazyn wektorowy. Zamiast kodować na sztywno logikę odpowiedzi, stosujemy podejście oparte na celach, pozwalając agentowi samemu ustalić, jakich zasobów musi użyć.
- **OpenAI Chat Model**. Podłączyliśmy model GPT firmy OpenAI jako silnik językowy do obsługi czatu. Interpretuje on prompty, konstruuje osadzenia i generuje płynne odpowiedzi przypominające ludzkie.
- **Simple Memory**. Ten moduł pamięci śledzi stan konwersacji. Bez niego sztuczna inteligencja nie pamiętałaby poprzednich wiadomości ani nie mogłaby odwoływać się do wcześniejszego kontekstu w wieloturowych konwersacjach. Działa jako lekka pamięć krótkotrwała, idealna dla agentów czatu.
- **Pinecone Vector Store1** (**narzędzie**). Agent ma dostęp do instancji Pinecone jako narzędzia, które może wywołać, gdy potrzebuje wiedzy ogólnej. W naszym przypadku wiedza ta jest uzupełniana danymi wyjściowymi bota rozpoznawczego (przechowywanymi w formie osadzeń).
- **Embeddings** (**OpenAI**). Kiedy użytkownik zadaje pytanie, agent osadza prompt i przeprowadza wyszukiwanie podobieństw w Pinecone z użyciem uzyskanego wektora. Pozwala mu to na pobranie najbardziej odpowiednich punktów danych z rekonesansu, czyli ugruntowanie odpowiedzi w rzeczywistych, wcześniej zebranych wynikach.

Taka architektura ułatwia wymianę modelu na inny (taki jak Claude lub Groq), skalowanie systemu do obsługi wielu narzędzi, a nawet dodawanie warstw kontroli dostępu i rejestrowania. Krótko mówiąc, jest to gotowy do wdrożenia przepływ pracy RAG, ale zbudowany w n8n bez kodowania.

Kup książkę Poleć książkę

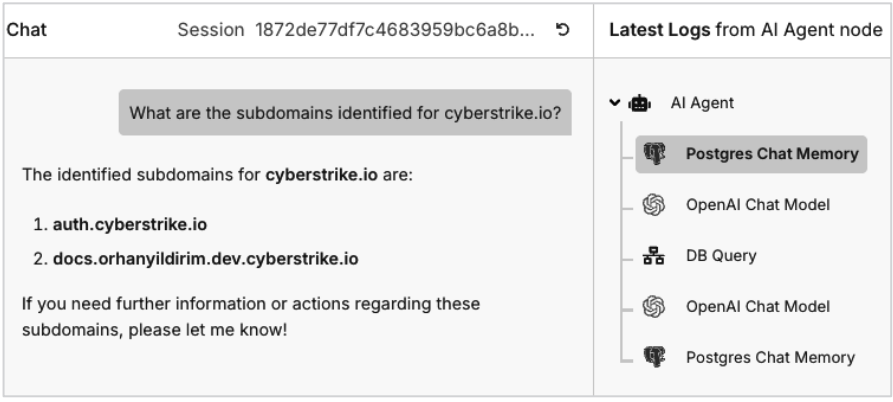
Na rysunku 3.31 pokazano, jak działa nasz agent AI z obsługą RAG. Po zadaniu pytania, takiego jak „Jakie są inne poddomeny?", agent pobiera odpowiednie dane z Pinecone i udziela konwersacyjnej, dokładnej odpowiedzi, wymieniając poddomeny zapisane podczas fazy rekonesansu. To świetny przykład tego, jak dane wyjściowe z rekonesansu, odpowiednio zindeksowane, można natychmiast znajdować i objaśniać z użyciem języka naturalnego.



Rysunek 3.31. Interakcja czatu z agentem AI, część 1.

Teraz, kiedy dane są już zapisane w wektorowej bazie danych i dostępne za pośrednictwem agenta AI, możemy zacząć zadawać znacznie bardziej złożone i wnikliwe pytania. Nie tylko wizualizujemy wyniki rekonesansu, ale wchodzimy z nimi w dialog.

Na rysunku 3.32 pokazano, jak agent AI może bezpośrednio odpytywać zwektoryzowane dane z rekonesansu i otrzymywać kontekstowe, czytelne dla człowieka odpowiedzi na podstawie wcześniej wzbogaconych informacji.

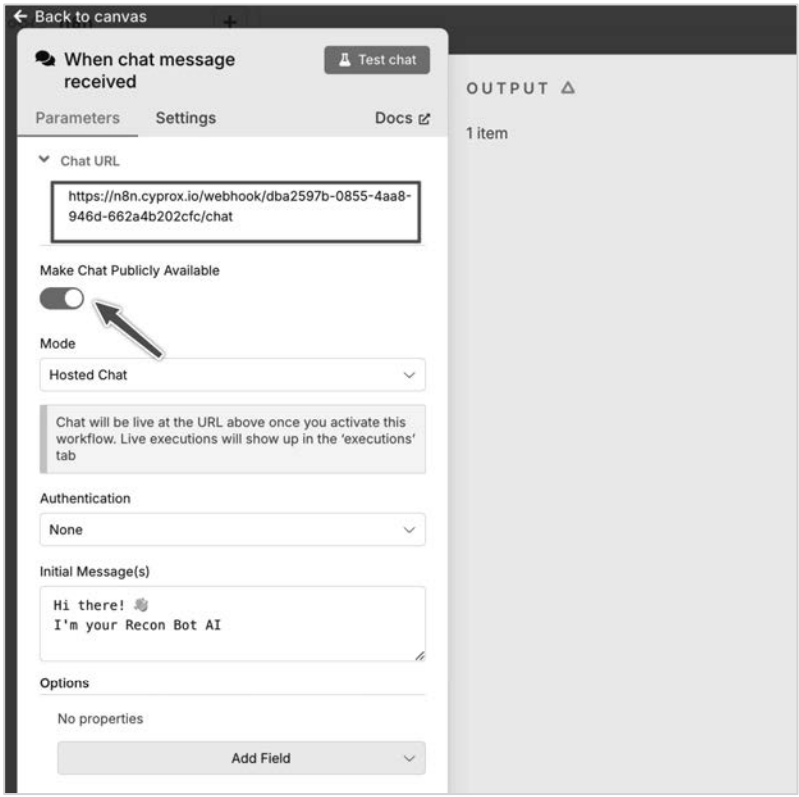


Rysunek 3.32. Interakcja czatu z agentem AI, część 2.

Otwiera to drogę do prawdziwego rozumowania AI — możliwości prowadzenia dialogu z danymi, wyodrębniania relacji, weryfikowania założeń i pozwalania LLM-owi na kierowanie kolejnymi krokami w przepływie pracy związanym z bezpieczeństwem. Niezależnie od tego, czy chodzi o identyfikację nietypowych wzorców ekspozycji, czy mapowanie powiązań infrastrukturalnych, sztuczna inteligencja staje się współpracownikiem, a nie tylko procesorem. To tutaj automatyzacja spotyka się z inteligencją.

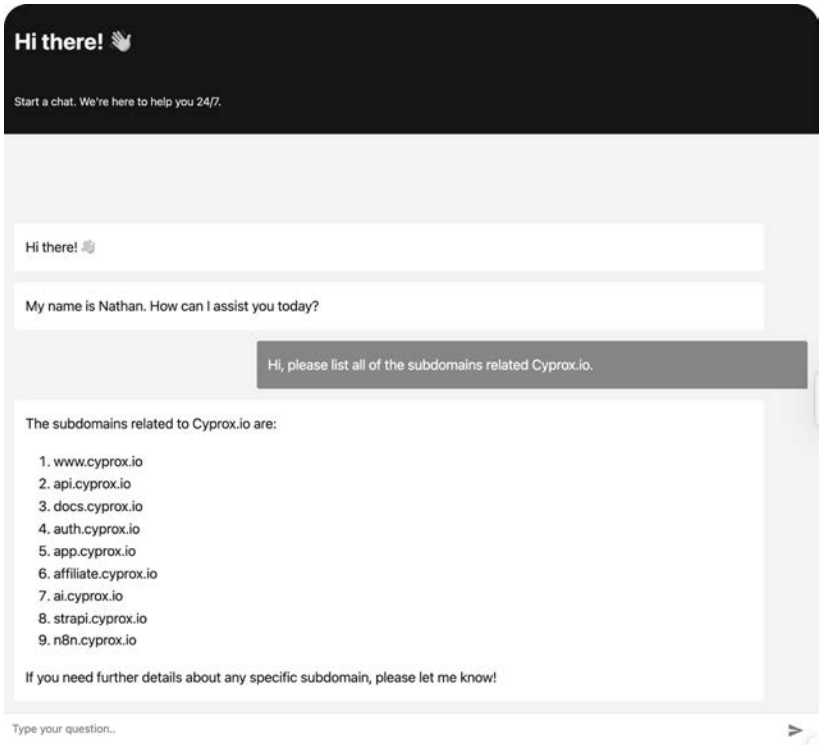
Chociaż obecny interfejs czatu w n8n stanowi funkcjonalny punkt wyjścia, jasne jest, że brakuje mu użyteczności i responsywności, jakich oczekiwalibyśmy w rzeczywistych analitycznych przepływach pracy. Na szczęście n8n oferuje rozwiązanie tego problemu. Wykorzystując zewnętrzne komponenty interfejsu użytkownika, na przykład przez osadzenie czatu w niestandardowej aplikacji internetowej za pośrednictwem webhooka n8n i węzłów API, można zaprojektować znacznie bardziej interaktywne i dopracowane środowisko. Takie podejście pozwala na budowanie dostosowanych interfejsów frontendowych, w których rozumowanie AI można zintegrować z pulpitami nawigacyjnymi, konsolami analitycznymi lub systemami zgłoszeń, dzięki czemu inteligentne agenty są nie tylko dostępne, ale i operacyjnie użyteczne.

Jak widać na rysunku 3.33, n8n pozwala udostępnić czat oparty na sztucznej inteligencji publicznie poprzez zaznaczenie opcji *Make Chat Publicly Available* w węźle *When chat message received*.



Rysunek 3.33. Publiczne udostępnianie czatbota AI

Po zaznaczeniu tej opcji generowany jest adres URL czatu na żywo, który można udostępnić na zewnątrz, aby umożliwić interakcje z agentem AI w czasie rzeczywistym (rysunek 3.34).



Rysunek 3.34. Publiczny interfejs czatbota AI

Gdy nasz agentowy przepływ pracy już w pełni działa i jest podłączony do internetu, czas zająć się równie ważną kwestią: zabezpieczeniem tego przepływu i jego powierzchni ataku.

Zabezpieczanie agentowego przepływu pracy

A teraz część, która spędza mi sen z powiek. Opisywane tu przepływy pracy oferują ogromne możliwości, ale jeśli nie zachowasz ostrożności, mogą stać się tylnymi drzwiami do Twojego systemu. Zewnętrzne interfejsy API, wrażliwe dane, modele LLM podejmujące autonomiczne decyzje… wszystko to może prowadzić do błędnych konfiguracji, wycieków danych i niezamierzonych konsekwencji.

Ten podrozdział ma pomóc Ci upewnić się, że automatyzacja oparta na sztucznej inteligencji nie tylko działa poprawnie, ale robi to w sposób bezpieczny, niezawodny i przy odpowiednim nadzorze.

Kup książkę Poleć książkę

Agentowe przepływy pracy zazwyczaj działają ponad granicami zaufania: otrzymują dane od użytkowników, wchodzą w interakcje z usługami stron trzecich, takimi jak OpenAI, Pinecone, VirusTotal lub Shodan, a czasami nawet publikują wyniki na publicznych pulpitach nawigacyjnych lub w interfejsach czatu. Te dynamiczne, połączone ze sobą systemy funkcjonują jak autonomiczne mikrousługi. I tak samo jak w tradycyjnych środowiskach DevSecOps lub korporacyjnych środowiskach SaaS zabezpieczenie takich przepływów pracy wymaga wielowarstwowego, aktywnego podejścia.

Bezpieczeństwo agentów tworzonych bez kodowania to nie tylko ochrona sekretów czy żonglerki uprawnieniami. Wymaga ono holistycznego spojrzenia na następujące kwestie:

- Kto lub co może wyzwolić ten przepływ pracy?
- Jakie dane wchodzą do systemu i z niego wychodzą?
- Gdzie są przechowywane poświadczenia i jaki jest ich zakres?
- Czy wynikami można manipulować poprzez wstrzykiwanie promptów lub zniekształcone dane?
- Czy dzienniki, ścieżki audytu i obsługa błędów są na tyle solidne, żeby wspierały reagowanie na incydenty?

Niezależnie od tego, czy używasz tych przepływów pracy wewnętrznie do automatyzacji rekonesansu, czy też udostępniasz funkcje agentów klientom lub analitykom za pośrednictwem interfejsu czatu, zastosowanie mają zasady minimalnych uprawnień, walidacji danych wejściowych/wyjściowych, określania zakresu poświadczeń i obserwowalności.

Przepływy pracy, które budujesz za pomocą agentów AI i narzędzi takich jak n8n, nie są już tylko statycznymi sekwencjami; to inteligentne, ewoluujące systemy. I jak każdy inteligentny system muszą być celowo zabezpieczane i rygorystycznie konserwowane. W kolejnych punktach omówimy zarówno podstawowe zabezpieczenia, jak i zaawansowane praktyki, które można wdrożyć w celu wzmocnienia agentów rozpoznawczych.

Co należy robić zawsze

Poniższe najlepsze praktyki stanowią podstawę bezpieczeństwa dla każdego agentowego przepływu pracy zbudowanego w n8n. Traktuj je nie jako opcjonalne kroki wzmacniające, ale jako zabezpieczenia domyślne, które powinny być stosowane od pierwszego dnia.

- **Używaj zmiennych środowiskowych**. Nie bądź osobą, która ujawnia klucze API na GitHubie. Zamiast tego przechowuj je w plikach *.env* lub we wbudowanym menedżerze poświadczeń n8n.
- **Ograniczaj uprawnienia**. Stosuj zasady minimalnych uprawnień dla kluczy API i połączonych usług. Na przykład nie używaj klucza administratora OpenAI, jeśli wystarczy klucz tylko do odczytu.
- **Oczyszczaj dane wejściowe**. Ostrożnie traktuj dane przesyłane przez użytkowników, zwłaszcza gdy są przekazywane do narzędzi takich jak `eval` czy `embedding` lub dołączane do żądań API.

Kup książkę Poleć książkę

- **Monitoruj wykonywanie przepływów pracy**. n8n udostępnia dziennik wykonywania. Włącz go i regularnie przeglądaj aktywność, aby wychwycić anomalie lub nieautoryzowane uruchomienia.
- **Ogranicz dostęp publiczny**. Jeśli udostępniasz hostowanego agenta czatu lub adres URL webhooka, pamiętaj o ograniczeniu dostępu tam, gdzie to możliwe, za pomocą tokenów, białych list IP lub serwerów proxy ograniczających tempo żądań.

Działania opcjonalne, ale zdecydowanie zalecane

Praktyki te nie są obowiązkowe w każdej konfiguracji, ale ich pominięcie znacznie zwiększa ryzyko ujawnienia danych, nadużyć lub powstania martwych punktów operacyjnych w miarę skalowania przepływów pracy:

- **Dodaj uwierzytelnianie do hostowanych punktów końcowych czatu**. Użyj uwierzytelniania podstawowego, OAuth lub własnej warstwy weryfikacji, aby zapobiec nadużyciom publicznego interfejsu czatu.
- **Włącz rejestrowanie na potrzeby audytu**. Przechowuj dzienniki w systemie zewnętrznym (np. Elasticsearch lub CloudWatch), aby zachować niezmienną ścieżkę audytu.
- **Szyfruj wrażliwe dane wyjściowe**. Jeśli przechowujesz pliki wyjściowe (np. raporty HTML lub połączony kod JSON) na dysku lub w chmurze, używaj szyfrowania danych w spoczynku.
- **Izoluj wektorowe bazy danych**. Jeśli używasz Pinecone, utwórz dedykowaną przestrzeń nazw lub indeks dla każdego typu przepływu pracy, aby zapobiec przeciekaniu danych między projektami.
- **Regularnie rotuj sekrety**. Wyrób sobie nawyk rotowania kluczy API i aktualizowania poświadczeń w środowisku n8n co 30 – 90 dni.

Podsumowanie

W tym rozdziale przekształciliśmy agentową AI z koncepcji w rzeczywistość, budując w n8n nasz pierwszy funkcjonalny przepływ pracy związany z bezpieczeństwem. Zaczynając od instalacji i konfiguracji Dockera, utworzyliśmy środowisko gotowe do wdrożenia produkcyjnego, zdolne do obsługi skryptów Pythona, łączności z MongoDB i zaawansowanych integracji API. Następnie połączyliśmy wiele warstw inteligencji (OpenAI do wnioskowania oraz kanały informacji o zagrożeniach, takie jak AbuseIPDB, VirusTotal, GreyNoise, OTX i urlscan do wzbogacania danych), zmieniając statyczną automatyzację w dynamiczne, świadome kontekstu podejmowanie decyzji. Nasz utworzony bez pisania kodu bot rozpoznawczy zademonstrował, jak połączyć wyliczanie poddomen, rozwikływanie nazw DNS, skanowanie portów i wyszukiwanie w Shodanie w jeden inteligentny potok, który generuje ustrukturyzowane raporty w formacie HTML.

Kup książkę Poleć książkę

Na koniec ulepszyliśmy nasz przepływ pracy, wdrażając system RAG z bazą danych Pinecone, który pozwala na prowadzenie dialogu w języku naturalnym z danymi z rekonesansu, co stanowi znaczący krok naprzód — od prostego gromadzenia danych do prawdziwego wnioskowania na ich podstawie. Pamiętaj, że zabezpieczanie tych agentowych przepływów pracy nie jest opcjonalne; od zmiennych środowiskowych i sanityzacji danych wejściowych, po rotację poświadczeń i rejestrowanie informacji na potrzeby audytu, zasady omówione w tym rozdziale zapewniają, że Twoje inteligentne agenty pozostaną zarówno użyteczne, jak i godne zaufania.

W następnym rozdziale zastosujemy te podstawy do zarządzania powierzchnią ataku, budując autonomiczne przepływy pracy, które w sposób ciągły odkrywają, monitorują i oceniają zewnętrzną powierzchnię ataku Twojej organizacji.

Kup książkę Poleć książkę


PROGRAM
PARTNERSKI
GRUPY HELION
1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ
Zmień swoją stronę WWW w działający bankomat!
Dowiedz się więcej i dołącz już dzisiaj!
http://program-partnerski.helion.pl
GRUPA
Helion

Zautomatyzuj ofensywne cyberbezpieczeństwo z agentami AI

Sztuczna inteligencja rewolucjonizuje branżę IT, a cyberbezpieczeństwo staje się jednym z głównych obszarów jej zastosowań. W dobie rosnącej złożoności systemów i wobec braku wykwalifikowanych specjalistów tradycyjna automatyzacja oparta na statycznych skryptach przestaje wystarczać. Ta książka idealnie wpisuje się w najnowsze trendy, wprowadzając koncepcję agentowej sztucznej inteligencji (Agentic AI) do działań ofensywnych. To przełomowe podejście pozwala na tworzenie autonomicznych systemów, które potrafią wnioskować, uczyć się i adaptować w czasie rzeczywistym. Zamiast biernie wykonywać polecenia, agenty AI stają się inteligentnymi członkami zespołu, zdolnymi do samodzielnego przeprowadzania złożonych testów penetracyjnych i zarządzania powierzchnią ataku.

Ta publikacja to praktyczny przewodnik po projektowaniu i wdrażaniu agentowych przepływów pracy. Krok po kroku uczy, jak używać platform no-code (n8n) i rozwiązań opartych na kodzie (Model Context Protocol), by zautomatyzować żmudne etapy testów bezpieczeństwa. Czytelnik dowie się, jak budować agenty do ciągłego zarządzania powierzchnią ataku (ASM), weryfikacji segmentacji sieci (PCI-DSS) i testowania aplikacji webowych. Książka pokazuje również, jak zintegrować duże modele językowe (LLM) z narzędziami takimi jak Nmap czy Playwright. Obejmuje pełny cykl: od rekonesansu, przez eksploatację podatności (XSS, SQLi), aż po zautomatyzowane tworzenie profesjonalnych raportów i ciągłą analizę zagrożeń (CTI) z użyciem baz wektorowych.

Najważniejsze zagadnienia:

- Podstawy agentowej AI i różnice względem tradycyjnej automatyzacji
- Budowa autonomicznych przepływów pracy w n8n (no-code)
- Zarządzanie powierzchnią ataku (ASM) i testowanie segmentacji PCI
- Użycie MCP i Playwright do testowania aplikacji webowych
- Automatyzacja eksploatacji luk (XSS, SQLi) i generowania raportów
- Tworzenie agentów CTI z pamięcią wektorową (RAG, Qdrant)

Orhan Yildirim — wiceprezes i architekt do spraw cyberbezpieczeństwa w globalnej instytucji finansowej. Ma ponad 10-letnie doświadczenie w przeprowadzaniu testów penetracyjnych i red teamingu. Posiada certyfikaty między innymi OSCE, OSCP i CRTO. Aktywny badacz na platformach HackerOne i Bugcrowd, prelegent konferencji BSides. Łączy metodyki pentestów z nowoczesnymi rozwiązaniami opartymi na agentach AI.

Helion
helion.pl
HELION S.A.
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl
KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-289-4225-7
9 788328 942257
Cena: 99,00 zł
packt