**ScamShield**

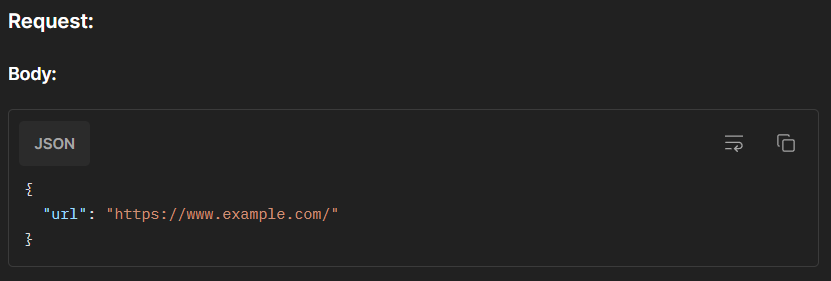
**Dokumentacja**

Dokumentacja została stworzona na potrzeby użytkownika, który chce wykorzystać API ScamShield

1. [Używanie EndPointów](#_1.Używanie_EndPointów)
2. [Moduły](#_2.Lista_Funkcji)
   1. [Detector.py](#_Detector.py)
      1. [\_recognize\_url](#__recognize_url)
      2. [estimate\_score](#_estimate_score)
   2. Scans.py
      1. [Scan\_protocol](#_scan_protocol)
      2. [Scan\_ssl](#_scan_ssl)
      3. [\_ssl\_check\_CA](#__ssl_check_CA)
      4. \_ssl\_check\_serial
      5. Scan\_html\_compare
      6. [Scan\_page\_age](#_scan_page_age)
      7. Scanjs
   3. [js\_anylzer.py](#_js_analyzer.py)
      1. [Analyze](#_analyze)
      2. \_analyze\_script
      3. \_get\_script

# 1.Endpoint

API ScamShield udostępnia jeden endpoint pod adresem ***/api/url*** . Przyjmuje tylko metody **HTTP Post** oraz **body** typu:



***url*** *–* adres domeny do sprawdzenia

**Response**

**Status Code: 200**

Obraz zawierający tekst

Opis wygenerowany automatycznie

**Response**

**Status Code: 400**

**Obraz zawierający tekst

Opis wygenerowany automatycznie**

***Domain –*** podana domena do sprawdzenia

***Phishing\_estimate –*** poziom zaufania do strony w przedziale <0, 100>, gdzie ‘0’ to w pełni wiarygodna sprawa, a ‘100’ to pewne zagrożenie.

# 2. Moduły:

## detector.py

## \_recognize\_url

**Opis werbalny**: Funkcja wykorzystywana jest do pobrania jak najwięcej danych z url oraz przetworzenia go w łatwiejsza formę do obsługi przez program

**Parametry funkcj**i: String w który zawiera url do strony

**Wartość zwracana**: Dictionary, który posiada podzielone dane pobrane ze strony

## estimate\_score

**Opis werbalny**: Funkcja jest clue programu wywołuję algorytmy sprawdzania oraz sumuje wyniki tych algorytmów

**Parametry funkcji**: String zawierający url do strony

**Wartość zwracana**: String który zawiera liczbę z zakresu 0-100 gdzie:

1. To strona uznana przez algorytm za całkowicie bezpieczną

100- To strona uznana przez algorytm za pewny phishing

## scans.py

## scan\_protocol

**Opis werbalny**: Funkcja pobiera z url protokół sieci www

**Parametry funkcji**: String zawierający url do strony

**Wartość zwracana**: Int z przedziału od 0-10

## scan\_ssl

**Opis werbalny**: Funkcja pobiera z danych o stronie certyfikat ssl

**Parametry funkcji**: String zawierający url do strony

**Wartość zwracana**: Int z przedziału od 0-100

## \_ssl\_check\_CA

**Opis werbalny**: Funkcja na podstawie nazwy organizacji która wystawiła certyfikat sprawdza czy widnieje w tabeli, jako popularny dostawca certyfikatów dla stron phsingowych. W tej funkcji wykorzystujemy darmowa stronę <https://sslbl.abuse.ch/statistics/> która udostępnia takie dane, licencja pozwala na wykorzystanie danych ze strony w projektach komercyjnych jak i niekomercyjnych za darmo.

**Parametry funkcji**: String zawierający nazwę organizacji wystawiającej certyfikat

**Wartość zwracana**: Int z przedziału od 0-100

## \_ssl\_check\_serial

**Opis werbalny**: Funkcja sprawdza na podstawie *serial key* certyfikatu czy znajduje się on na *blacklist* na stronie <https://sslbl.abuse.ch/blacklist/sslblacklist.csv> w przypadku wykrycia takiego klucza automatycznie zostaje ona uznana za stronę phishingowa

**Parametry funkcji**: String zawierający klucz certyfikatu ssl

**Wartość zwracana**: Int z przedziału 0-100

## scan\_html\_compare

**Opis werbalny**: Funkcja wyszukuje na podstawie tytułu strony wyszukuję ją w googlu i pobiera pierwszy wynik wyszukiwania, pobiera jego kod html hashuje go i porównuję go z hashowanym kodem html przekazanej do sprawdzenia strony, jeżeli różni choć jeden bit na stronie strona jest flagowana jako potencjalnie niebezpieczna

**Parametry funkcji**: Zestaw danych o stronie, domene strony do sprawdzenia

**Wartość zwracana**: Int z przedziału 0-50

## scan\_page\_age

**Opis werbalny** : Funkcja pobiera z danych strony datę utworzenia, następnie oblicza ilość dni które upłynęły od utworzenia strony.

**Parametry funkcji**: String zawierający url sprawdzanej strony

**Wartość zwracana**: Int z przedziału 0-10

## scan\_js

**Opis werbalny**: Funkcja wywołuję algorytmy (opisane poniżej) sprawdzenia kodu js na stronie

**Parametry funkcji**: Zestaw danych o stronie

**Wartość zwracana** Int z przedziału 0-15

## scan\_shops\_service

**Opis werbalny:** Funkcja wywołuje algorytmy sprawdzenia aukcji w popularnych serwisach aukcyjnych (OLX, Allegro, ebay, Vinted). Algorytmy te działają na tej samej zasadzie. Otrzymują one jako parametr link url do sprawdzanej aukcji szukają one numeru aukcji który jest unikalny dla każdego z serwisów i jeżeli aukcja nie istnieje z podanym id aukcji to strona jest oznaczana jako bardzo mocno podejrzana.

**Parametry funkcji:** String zawierający url sprawdzanej strony

**Wartość zwracana:** Int z przedziału 0-60

## js\_analyzer.py

## analyze

**Opis werbalny:** Funkcja wywołuje dalsze algorytmy sprawdzenia dzięki czemu wiemy ile jest skryptów oraz ile jest prób dekodowania rzeczy w tych skryptach i jeżeli jest duża ilość prób dekodowania uznajemy stronę za potencjalnie niebezpieczną

**Parametry funkcji**: Zestaw danych o stronie

**Wartość zwracana**: Int z przedziału 0-15

## \_analyze\_script

**Opis werbalny:** Funkcja sprawdza użycie metod dekodujących w skryptach, dzięki czemu wiemy czy osoba pisząca skrypt chciała ukryć coś przed nami

**Parametry funkcji:** Zestaw danych o stronie

**Wartość zwracana:** Int ilość prób dekodowania w skryptach

## \_get\_scripts

**Opis werbalny**: Funkcja skanuje kod html w poszukiwaniu scryptów umieszczonych na stronie, następnie rozpoczyna pobieranie skryptów

**Parametry funkcji**: Zestaw danych o stronie

**Wartość zwracana**: Tablica skryptów, ilość skryptów