dokuentacija.md

IP Melnā Saraksta Pārvaldības Sistēma - Projekta Struktūra

Grupas dalībnieku saraksts

Autors: Agris Pudāns, ap08426

Īss programmas apraksts

Šis projekts ir pārstrādāta versija IP Melnā saraksta pārvaldībai izmantojot Flask API. Univeritātes darba vajadzībām sistēma ir pieejama interneta vietnē https://python.pudans.lv/. Atvērtais Git repozitorijs: https://github.com/achenlv/ap08426-DatZB084.

Sistēma apkopo IP adreses no dažādiem avotiem un nodrošina kopēju sarakstu. Pieprasījumus var sūtīt ar manuāliem API pieprasījumiem vai speciāli sagatavotiem Python skriptiem hostu (host.py) un kontrolieru (controller.py) mašīnām, kas būs izstrādāti nākamājās sistēmas versijās.

Projekta Struktūra

```
blacklist/
│
├── app/                          # Galvenā API lietojumprogrammas
pakotne
│   ├── __init__.py               # API inicializācija
│   ├── routes/                   # API maršrutu apstrādātāji
│   │   ├── __init__.py
│   │   ├── export.py             # Melnā saraksta eksportēšanas
galapunkti
│   │   └── import_routes.py      # Importēšanas galapunkti
│   │
│   └── models/                   # Datu modeļi un shēmas
│       ├── __init__.py
│       └── schemas.py            # Pydantic modeļi datu validācijai
│
├── config/                       # Konfigurācija visiem komponentiem
│   ├── __init__.py
│   └── settings.py               # Iestatījumu klases API, Host,
Controller
│
├── db/                           # Datubāzes pārvaldība
│   ├── __init__.py
```

```
│   └── database.py              # SQLite3 savienojums un operācijas
│
├── utils/                       # Koplietojamās utilītas
│   ├── __init__.py
│   ├── logging.py               # Žurnālu veidošanas konfigurācija
│   └── validators.py            # Kopējās validācijas funkcijas
│
├── scripts/                     # Klienta skripti (TODO sarakstā)
│   ├── host.py                  # Host mašīnas skripts (IPFW
pārvaldība)
│   └── controller.py            # Kontroliera mašīnas skripts
(plānotie importi)
│
├── logs/                        # Žurnālfailu direktorija
│   ├── api.log
│   ├── host.log
│   └── controller.log
│
├── data/                        # Datu direktorija importēšanas
failiem
│   ├── elastiflow/              # Elastiflow CSV faili
│   └── fail2ban/                # Fail2ban datu faili
│
├── app.py                       # Galvenais API sākumpunkts
├── requirements.txt             # Projekta atkarības
├── .env.example                 # Piemērs vides mainīgajiem
└── README.md                    # Projekta dokumentācija
```

Galvenie Faili

- `app.py`: Galvenais Flask lietojumprogrammas sākumpunkts
- `requirements.txt`: Projekta atkarības
- `.env.example`: Vides iestatījumu veidne
- `README.md`: Projekta dokumentācija un uzstādīšanas instrukcijas
- `dokumentacija.md`: Projekta dokumentācija un uzstādīšanas instrukcijas

Vides Uzstādīšana

1. Virtuālās vides izveidošana

```
python -m venv venv
source venv/bin/activate  # Linux/Mac
venv\Scripts\activate.bat  # Windows
```

2. Moduļu instalēšana

```
pip install -r requirements.txt
```

3. Vides konfigurēšana

```
cp .env.example .env
```

Piemērs `.env` faila struktūrai:

```
# API Iestatījumi
API_HOST=0.0.0.0
API_PORT=5000
DATABASE_PATH=saraksts.db

# Host Iestatījumi
HOST_API_URL=http://127.0.0.1:5000
IPFW_TABLE=2
IPFW_RULE=1035

# Kontroliera Iestatījumi
CONTROLLER_API_URL=http://127.0.0.1:5000
SCAN_INTERVAL=300
```

4. Importējamo datu direoktroju izveidošana (Kontrolieris):

```
mkdir -p logs data/elastiflow data/fail2ban
```

5. API Servera darba uzsākšana:

```
python app.py
```

Komponentu Detalizācija

API Komponenti (app/)

- **routes/**

    - `export.py`: Apstrādā melnā saraksta eksportēšanas pieprasījumus
        - `/export/blacklist`: Eksportē melno sarakstu
        - `/export/whitelist`: Eksportē balto sarakstu
    - `import.py`: Pārvalda visus importēšanas darbus (vienreizējs IP, masveida, ārējie avoti)
        - `/import/single`: Importē vienu IP adresi melnajā vai baltajā sarakstā
        - `/import/bulk`: Masveida IP adrešu importēšana
        - `/imoprt/blocklist_de`: IP adrešu importēšana no blocklist.de vietnes
        - `/import/source`: Importē IP adreses no ārējiem avotiem, piemēram, fail2ban vai elastiflow (paredzēts izstrādāt nākošajās versijās)

- **models/**

    - `schemas.py`: Definē datu validācijas modeļus, izmantojot Pydantic
        - `IPEntry`
        - `BulkImportRequest`
        - `ExportFilter`

Konfigurācija (config/)

- `settings.py`: Satur atsevišķas konfigurācijas klases katram komponentam
    - `APISettings`: Flask API iestatījumi
    - `HostSettings`: Host skripta iestatījumi
    - `ControllerSettings`: Kontroliera skripta iestatījumi

Datubāze (db/)

- `database.py`: Pārvalda SQLite3 datubāzes operācijas
    - Savienojuma pārvaldība
    - Tabulu izveide un inicializācija
    - Kopējās datubāzes operācijas

Utilītas (utils/)

- `logging.py`: Centralizēta logu veidošanas konfigurācija
- `validators.py`: Kopējās validācijas funkcijas
    - IP adrešu validācija
    - Datu formāta validācija

Klienta Skripti (scripts/)

Funkcionalitāte būs izstrādāta nākamājās sistēmas versijās.

- `host.py`: Darbojas uz host mašīnām

    - Sinhronizējas ar API
    - Pārvalda vietējo melno sarakstu

- `controller.py`: Darbojas uz kontroliera mašīnas
    - Plāno periodiskos importus
    - Pārvalda datu vākšanu no avotiem

Datu Direktorija (data/)

Funkcionalitāte būs izstrādāta nākamājās sistēmas versijās.

- **elastiflow**/: Elastiflow CSV ziņojumu glabāšana
- **fail2ban**/: Fail2ban datu failu glabāšana

Logu Direktorija (logs/)

Glabā rotējošos logus katram komponentam:

- `api.log`: API darbības logi
- `host.log`: Host skripta darbības logi
- `controller.log`: Kontroliera skripta logi

API mezgli

Importēšana

1. Vienas IP adreses pievienošana

Blacklist piemērs

```
curl -X POST http://127.0.0.1:5000/import/single \
-H "Content-Type: application/json" \
-d '{
  "ip": "192.168.1.100",
  "type": "blacklist",
  "source": "manual",
  "comment": "Suspicious activity"
}'
```

Whitelist piemērs

```
curl -X POST http://127.0.0.1:5000/import/single \
-H "Content-Type: application/json" \
-d '{
  "ip": "95.216.168.7",
  "type": "whitelist",
```

```
  "source": "manual",
  "reason": "Trusted IP",
  "comment": "Added manually"
}'
```

Atbildes piemērs:

```
{
  "message": "IP 192.168.1.100 added to blacklist"
}
```

2. Vairāku IP adrešu pievienošana

```
curl -X POST http://127.0.0.1:5000/import/bulk \
-H "Content-Type: application/json" \
-d '{
  "ips": ["192.168.1.100", "192.168.1.101", "10.0.0.0/30"],
  "source": "bulk-import",
  "type": "whitelist",
  "comment": "Batch import"
}'
```

Atbildes piemērs:

```
{
  "message": "Added 256 IPs to blacklist"
}
```

3. Blocklist.de datu pievienošana

```
curl -X POST http://127.0.0.1:5000/import/blocklist_de
```

Atbildes piemērs:

```
{
  "message": "Added 1234 IPs from blocklist.de"
```

```
}
```

Eksportēšana

Melnā/Baltā saraksta eksports

Eksportēšana notiek CSV vai JSON formātā

```
# JSON eksports
curl "http://127.0.0.1:5000/export/blacklist"

# CSV eksports
curl "http://127.0.0.1:5000/export/blacklist?format=csv"

# Eksports ar filtrēšana
curl "http://127.0.0.1:5000/export/blacklist?source=manual"

# Baltā saraksta eksports JSON formātā
curl "http://127.0.0.1:5000/export/whitelist"

# Baltā saraksta eksports CSV formātā
curl "http://127.0.0.1:5000/export/whitelist?format=csv"
```

Atbildes piemērs

```
ip,source,added_at,reason,comment
192.168.1.101,manual,2025-01-12 11:59:56.617142,,Suspicious activity
192.168.1.100,bulk-import,2025-01-12 12:00:10.502383,,Batch import
10.0.0.1,bulk-import,2025-01-12 12:00:10.502463,,Batch import
10.0.0.2,bulk-import,2025-01-12 12:00:10.502464,,Batch import
```

Eksports iptbales formātā

Melnā saraksta IP adrešu eksports

```
curl "http://127.0.0.1:5000/export/iptables"
```

Abildes piemērs:

```
{
  "count": 2,
```

```
  "ips": ["192.168.1.100", "192.168.1.101"],
  "format": "plain"
}
```

Melnā saraksta IP adrešu eksports iptables formātā

```
curl "http://127.0.0.1:5000/export/iptables/rules"
```

Atbildes piemērs:

```
{
  "count": 2,
  "rules": [
    "iptables -A INPUT -s 192.168.1.100 -j DROP # manual",
    "iptables -A INPUT -s 192.168.1.101 -j DROP # fail2ban"
  ],
  "format": "iptables"
}
```

Direktoriju Atļaujas

Nepieciešamās atļaujas pareizai darbībai:

- `data/`: Lasīšanas atļaujas API
- `logs/`: Rakstīšanas atļaujas visiem komponentiem
- Datubāzes fails: Lasīšanas/Rakstīšanas atļaujas API

Izvietošanas Apsvērumi

Katru komponentu var izvietot uz dažādām mašīnām:

1. API Serveris: Hostē Flask lietojumprogrammu
2. Host Mašīnas: Darbojas host.py skripts
3. Kontroliera Mašīna: Darbojas controller.py skripts

Nepieciešamā tīkla savienojamība:

- Host mašīnas → API Serveris
- Kontroliera mašīna → API Serveris
- Visas mašīnas → Žurnālu direktorijas (vietējās)

TODO uzlabojumi nākošajās versijās

- Host mašīnas skripta atjaunošana
- Kontroliera mašinas skripta atjaunošana vai dzēšana
- Datu ielāde no Elastiflow atskaitēm