Huligany 1.5

`huligany1.5.py` **— это инструмент для анализа подозрительных IP-адресов, автоматической оценки их угрозы и подготовки рекомендаций по блокировке.**
Он агрегирует данные из 6+ открытых источников (AbuseIPDB, FireHOL, Feodo, blocklist.de и др.), определяет ASN и страну, группирует атаки по подсетям и автономным системам, а затем выдаёт:

- **CSV-отчёт** с деталями по каждому IP
- **HTML-отчёт** с цветовой маркировкой и фильтрацией
- **Готовый список для блокировки** (IP и /24)
  Скрипт помогает SOC-аналитикам, администраторам и специалистам по ИБ быстро принимать решения: **кого банить, а кого мониторить**.
  **При запуске скрипта у пользователя будут запрашиваться следующая информация:

1. **Конфигурация**
   - Найдены ли `.huligany1.5.config.json` файлы? → выбрать или продолжить без
   - Если конфигов нет → предложение создать или продолжить без
2. **API-ключи AbuseIPDB**
   - Введите первый токен
   - Будут ли ещё токены? (y/N) → повторять, пока не ответят «нет»
3. **API-ключи ipinfo.io**
   - Введите первый токен
   - Будут ли ещё токены? (y/N)
4. **Определение ASN/страны**
   - Требуется ли определение ASN, страны и имени сети? (Y/n)
5. **Проверка через FireHOL**
   - Выполнить проверку по спискам FireHOL? (y/N)
   - Если да → использовать текущую базу или обновить?
6. **Кастомные критерии блокировки**
   - Использовать только стандартные критерии? (Y/n)
   - Если «нет» → ввод кастомного условия
   - Будет ли ещё кастомный критерий? (y/N)
   - Учитывать стандартные критерии вместе с кастомными? (Y/n)
7. **Сохранение конфига**
   - Сохранить настройки в файл? (y/N) → если да, ввести имя конфига

На вход подается spisok.txt с перечнем IP-адресов и количеством сработок по этим адресам. Полный результат скрипта будет содержать следующую информацию:

Столбец	Описание	Источник
IP	Исходный IP-адрес из логов	—
Count	Количество сработок СЗИ с этого IP	—
/24	Сеть с маской /24 (например, 192.168.1.0)	Расчёт по IP
Count/24	Суммарное количество сработок в подсети /24	Агрегация
AS	Номер автономной системы (ASN)	https://ipinfo.io/developers/lite-api
AS Name	Название организации, владеющей ASN	https://ipinfo.io/developers/lite-api
Country	Код страны (например, US, RU)	https://ipinfo.io/developers/lite-api
Count/AS	Суммарное количество сработок по ASN	Агрегация
AbuseScore	Уровень доверия угрозы (0–100%)	https://www.abuseipdb.com/
AbuseReports	Число жалоб за последние 90 дней	https://www.abuseipdb.com/
BlocklistDE	1 если IP в списке атак на серверы blocklist.de	https://www.blocklist.de/ru/index.html
Feodo_Original	1 если IP — C&C-сервер ботнета (Emotet, TrickBot)	https://feodotracker.abuse.ch/
EmergingThreats	1 если IP скомпрометирован (бот, зомби)	https://rules.emergingthreats.net/
firehol_level1	1 если IP в основном списке угроз FireHOL	https://iplists.firehol.org/?ipset=firehol_level1
firehol_level2	Умеренный уровень угрозы (подтверждённые атаки)	https://iplists.firehol.org/?ipset=firehol_level2
firehol_level3	Высокая достоверность (минимум ложных срабатываний)	https://iplists.firehol.org/?ipset=firehol_level3
feodo	1 если IP в списке Feodo (через FireHOL)	https://iplists.firehol.org/?ipset=feodo_badips

Столбец	Описание	Источник
bruteforceblocker	1 если IP атакует SSH/RDP (≥3 жалобы)	https://iplists.firehol.org/?ipset=bruteforceblocker
Action	Предлагаемое действие с IP адресом	—
Reason	Обоснование поля Action	—
CustomReason	Обоснование блокировки, если блокировка произошла по кастомному критерию	—

Стандартные критерии для `Action = "Block"`:

Условие	Пояснение
`AbuseScore >= 80`	Очень высокая репутационная угроза (AbuseIPDB)
`Feodo_Original == 1`	IP является C&C-сервером ботнета (Emotet, TrickBot и др.)
`firehol_level1 == 1`	IP в основном списке FireHOL (подтверждённые угрозы)
`BlocklistDE == 1`	IP атаковал серверы blocklist.de (SSH/FTP брутфорс)
`bruteforceblocker == 1`	IP замечен в брутфорсе (≥3 жалобы от разных пользователей)
`AbuseScore >= 70 AND Count >= 5`	Постоянный агрессор: высокий рейтинг + частые сработки

В ходе работы скрипта будет предложение добавления кастомных критериев для удобства фильтрации полученной информации. Синтаксис для написания кастомных критериев:

Элемент	Значения
Параметры	`IP`, `Count`, `AbuseScore`, `Country`, `AS`, `Feodo_Original`, `firehol_level1`, и т.д. (все столбцы CSV)
Операторы сравнения	`==` (равно), `>=` (больше или равно), `<=` (меньше или равно), `!=` (не равно)
Логика	`and` (И), `or` (ИЛИ)
Значения	Числа (`80`, `1`) или строки в одинарных кавычках (`'RU'`, `'US'`)

Примеры кастомных критериев:
Блокировать, если страна — **Россия И** репутационный рейтинг **≥80**:

```
Country == 'RU' and AbuseScore >= 80
```

Блокировать, если страна **НЕ Россия и НЕ Беларусь И** количество сработок **≥3**

```
Country != 'RU' and Country != 'BY' and Count >= 3
```

Блокировать, если IP есть в **Feodo ИЛИ** в **FireHOL Level1**

```
Feodo_Original == 1 or firehol_level1 == 1
```

Блокировать, если **AbuseScore ≥70**, **Count ≥5**, **И страна НЕ Россия**

```
AbuseScore >= 70 and Count >= 5 and Country != 'RU'
```

Список стран с расшифровкой кода страны: 2letter-country.xlsx

> Перед началом работы нужно сгенерировать API токены систем **abuseIPDB** (можно сделать здесь, после регистрации на ресурсе) и **ipinfo** (можно сделать здесь, после регистрации на ресурсе). Токены будут запрошены в самом начале работы скрипта.

> Важный момент. У токена **abuseipdb** есть ограничение в виде 1000 запросов в день, поэтому если понадобится сделать больше 1000 запросов, то найти дополнительный токен и добавить его в скрипт. Скрипт предложит вам добавить дополнительные токены. При выполнении работы скрипт автоматически перейдет на следующий токен при израсходовании текущего токена. Количество выполненных запросов можно увидеть тут
> Схожая ситуация в **ipinfo.io**, но здесь ясности в ограничении. В документации указано неограниченное обращение в месяц по тарифному плану IPinfo Lite, который и используется в скрипте. Но на всякий случай в скрипте добавлена возможность добавления нескольких токенов. Тарифные планы для ознакомления тут. Статистику по запросам IPinfo можно увидеть тут

Запуск

```
python huligany1.5.py spisok_pidorasov.txt
```