**Назначение системы**

Система будет определять и классифицировать репозитории с небезопасным кодом на github (работа с github api)

**Принцип работы системы**

Программа получает код из репозиториев по ключевым словам основные это python, django, но возможно в будущем добавятся и другие java, c#, flask, …

После того как найдены репозитории каждый мы проверяем на наличие небезопасного кода. Если такой код найден, сохраняем информацию в следующем виде:

1. Ссылка на репозиторий
2. Ключевые слова
3. Ссылка на модуль с кодом в котором есть небезопасный код
4. Тип небезопасного кода
5. Статус

(Подсказка) На гитхабе так же есть возможность вести поиск не по репозиториям, а сразу по коду, возможно это будет удобнее

Также можно добавить дополнительную информацию например строчку с кодом и все что может быть полезно

Возможные статусы кода: 1) Потенциально опасен 2) Содержит уязвимость

Примерное описание по поиску небезопасного кода:

\* - отмечена усложненная аналитика

1. В любых проектах на python

* В коде есть функция eval (потенциально опасен)
* \* В коде есть функция eval на вход она принимает данные из стороннего источника (например input, из файла, по сети) (Содержит уязвимость)
* В коде есть sql инъекция. Определить можно следующим образом внутри f строки или строки format есть sql-запрос, например select (Потенциально опасен)
* \* Этот запрос после выполняется в базе данных (например есть import sqlite3, …) (Содержит уязвимость)
* В коде используется модуль pickle (Потенциально опасен)
* \* В функцию pickle.load передаются данные из стороннего источника (например input, чтение из файла, по сети) (Содержит уязвимость)
* \* в коде в явном виде указаны login-и и пароли (например от email или от какой либо системы) (Содержит уязвимость)

1. В проектах на django

* Явно указаны пароли от email в файле settings.py в константах EMAIL\_HOST\_USER, EMAIL\_HOST\_PASSWORD (Содержит уязвимость)
* Отключен csrf token. В settings.py в константе MIDDLEWARE\_CLASSES отстутствует 'django.middleware.csrf.CsrfViewMiddleware' или он закоменчен (Потенциально опасен)
* \* локально отключен csrf token. В коде проекта используется декоратор @csrf\_exempt (Потенциально опасен)

**Примеры небезопасного кода можно найти тут:** [**https://github.com/DanteOnline/unsafe-python-code/tree/master**](https://github.com/DanteOnline/unsafe-python-code/tree/master)

Все пункты делать не обязательно на зачет достаточно 1-го любого пункта, на максимально баллов 3-х пунктов. Также можно добавить какие то еще условия небезопасного кода

После того как данные получены сохранить их в файл в формате json. Пример итогового файла:

{

**'https://github.com/DanteOnline/nu-loto-resolve-dz'**: {

**'words'**: [**'python'**, **'django'**],

**'unsafe\_modules'**: [

{

**'name'**: **'main.py'**,

**'unsafe code type'**: **'В коде используется eval'**,

**'status'**: **'Потенциально опасен'**

},

{

**'name'**: **'other.py'**,

**'unsafe code type'**: **'явно указаны логин и пароль'**,

**'status'**: **'Содержит уязвимость'**

},

]

},

**'https://github.com/DanteOnline/nu-loto-resolve-111'**: {

**'words'**: [**'python'**],

**'unsafe\_modules'**: [

{

**'name'**: **'net.py'**,

**'unsafe code type'**: **'sql иньекция'**,

**'status'**: **'Потенциально опасен'**

},

]

}

}