Задание по дисциплине

Дисциплина	Разработка защищённых программных систем
Тема	Итоговое домашнее задание
Форма проверки	**Это итоговое задание и проверяется преподавателем**
Имя преподавателя	Владимир Телепов
Время выполнения	360 минут
Цели задания	Закрепить: • практический навык поиска ошибок при использовании инструментов статического и динамического анализа, • архитектурную декомпозицию приложения, • навыки построения модели угроз и оценки рисков, • навык ревью безопасности исходного кода, • навыки выявления уязвимостей в исходном коде и формулирования замечаний по их устранению
Инструменты для выполнения ДЗ	Для выполнения задания используйте: • Google Docs или Яндекс Документы, • OWASP Treat Dragon для моделирования угроз, • secure-code-review-checklist для ревью кода, • калькулятор по оценке рисков ФСТЭК, • Semgrep **или аналог** в качестве SAST, • OWASP ZAP в качестве DAST
Правила приёма работы	1. Для выполнения задания создайте Google- или Яндекс-документ и заполняйте в нём информацию по этапам. 2. Прикрепите ссылку на документ. **Важно:** убедитесь, что по ссылке есть доступ. Название файла должно содержать фамилию и имя студента и название ДЗ (итоговое домашнее задание по дисциплине «Разработка защищённых программных систем»)
Критерии оценки	**Задание считается выполненным, если:**

Дисциплина	Разработка защищённых программных систем
Тема	Итоговое домашнее задание
Форма проверки	**Это итоговое задание и проверяется преподавателем**
Имя преподавателя	Владимир Телепов
Время выполнения	360 минут
	• прикреплена ссылка на файл с выполненным заданием, • доступы к материалам открыты, • выполнены все остальные требования задания. **Задание не выполнено, если:** • файл с заданием не прикреплён или отсутствует доступ по ссылке, • не выполнены все остальные требования чек-листа

Этап 0

Изучите лекции и вебинары по пройденным темам и повторите самостоятельные домашние задания.

Этап 1

Клонируйте или загрузите на локальный диск проект уязвимого приложения по вариантам:

1. https://github.com/flaskbb/flaskbb
2. https://github.com/cookiecutter-flask/cookiecutter-flask
3. https://github.com/hack4impact/flask-base
4. https://github.com/anguzz/Flask-JWT-SoapUI
5. https://github.com/dharaneesh71/API-Secure-A-Flask-Based-Web-Application-Security-Simulator
6. https://github.com/sebst/pythonic-news
7. https://github.com/cpcdiu/hacksprint
8. https://github.com/Code-Institute-Community/ci-hackathon-app
9. https://github.com/hackers-friend/HackersFriend-NewsAggregator
10. https://github.com/Raahim2/Synergy_MatrixOut

Этап 2

В данном этапе вы можете выбрать один из вариантов или выполнить два варианта:

Вариант 1
В случае выбора этого варианта - Этап 3 не выполняете

1.Проанализируйте выбранный проект. Соберите и запустите приложение. Смоделируйте модель угроз.
2. Выполните декомпозицию приложения выбрав не более двух функций. Выделите возможные уязвимости и опишите их.

Вариант 2

1. Запустите сканирование проекта средством SAST, дождитесь его окончания и получите отчёт о сканировании (https://github.com/semgrep/semgrep).
2. Разверните и запустите проект по инструкции из его репозитория, запустите сканирование средством DAST, дождитесь его окончания и получите отчёт о сканировании (https://ultahost.com/knowledge-base/install-owasp-zap-in-kali-linux/).

Этап 3
Выполняется в случае выбора **Варианта 2**

Оформите документ в соответствии со следующими требованиями:

1. В документе должны быть перечислены все выявленные обоими средствами проблемы безопасности в формате, аналогичном формату домашнего задания по теме 11.
2. Для каждой проблемы **с критичесностью выше “низкой**” (то есть уровень угрозы выставленный инструментом должен быть: **средний, высокий, критический**) обязательно укажите, является ли она ложным срабатыванием и, если является, обоснуйте своё решение.

Примечание. Низкий и информационный уровень не рассматриваете, однако, если кроме эти двух уровней срабатываний нет – берите две любые уязвимости.

3. Для каждой проблемы **с критичностью “высокая” и/или “критическая”**, не являющейся ложным срабатыванием, обязательно сформулируйте рекомендации по её устранению в формате, аналогичном формату домашнего задания по прошлым темам.

В общем случае вы проверяете на ложноположительность **уязвимости от среднего до критического уровня** и пишете рекомендации по устранению для всех уязвимостей **высокого и критического** уровня.

В случае **если срабатывания среднего и выше уровня нет** в отчете, то необходимо проверить на ложноположительность все срабатывания до нахождения **двух любых уязвимостей**, а также написать **рекомендации по устранению** этих двух уязвимостей.

Этап 4

Разместите ссылку на документ в Google Doc или Яндекс Документах. Проверьте доступ.

Название файла должно содержать фамилию, имя студента и название ДЗ (итоговое домашнее задание по дисциплине «Разработка защищённых программных систем»).

Чек-лист самопроверки

Для проекта уязвимого приложения:

Вариант 1

- создана модель угроз;
- выполнена декомпозиция приложения, выделены и описаны возможные уязвимости;

Вариант 2:

- получен отчёт о сканировании с помощью SAST и DAST;
- оформлен итоговый документ по уязвимостям приложения.