Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет прикладной информатики (ФПИН)

Лабораторная работа №5

по дисциплине: Основы кибербезопасности

Автор: доцент практики, кандидат технических наук

Кравчук Алексей Владимирович

Санкт-Петербург
2025

Тема занятия: Исследование web-уязвимостей.

Цель работы:

- Ознакомиться с часто встречающимися уязвимостями в web-приложениях и «худшими практиками веб-разработки (OWASP Top 10 и уязвимое приложение Juicy Shop).
- Получить навыки работы со сканерами web-уязвимостей.

Задание:

По ходу выполнения работы вставлять в отчет скриншоты и краткие пояснения. В блоке 2.2 нужно попробовать заставить nuclei найти больше уязвимостей (чтобы нашел ещё что-то кроме Prometheus Metrics). В выводах по лабораторной работе привести сравнительный анализ результатов работы nuclei и Acunetix.

Краткие теоретические сведения

В этой лабораторной работе (далее - лаба) логично было бы использовать результаты, полученные в ходе выполнения лабы №3 (получили информацию об IT-инфраструктуре ИТМО: DNS-сервера, домены, поддомены, web-директории). Однако в таком случае мы бы нарушили закон, а наши действия квалифицировались бы как «Преступления в сфере компьютерной информации» (Глава 28 Уголовного Кодекса РФ).

Поэтому мы вынуждены тренировать наши навыки на тестовом полигоне, в качестве которого выбран **OWASP Juice Shop**. Это open-source-приложение, написанное на JavaScript с преднамеренно внесенными уязвимостями.

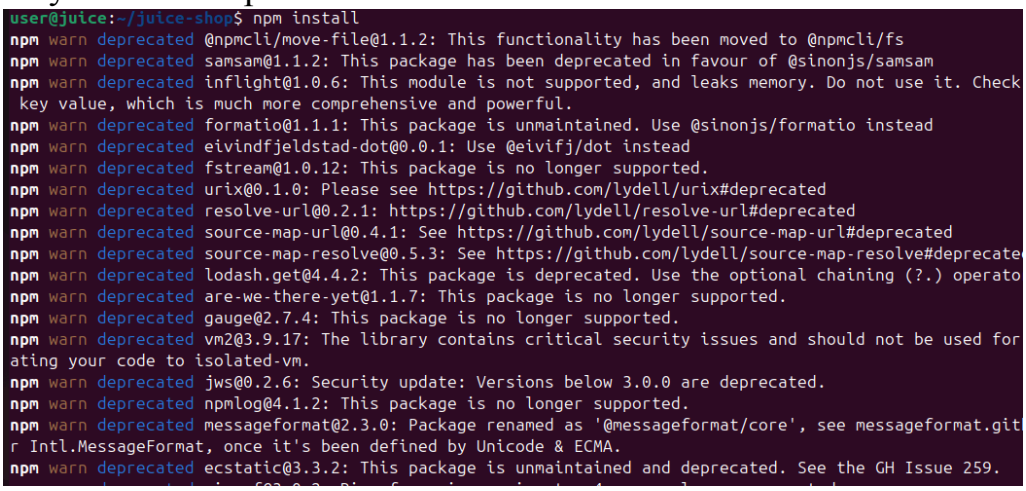
Можно думать нём как о «легко взламываемом интернет-магазине» - он ведёт себя как обычный сайт: вход на сайт, выбор товаров, помещение их в корзину и оформление заказа.

В **Juice Shop** встроена система учебных заданий (challenges). Каждое задание направлено на поиск конкретной уязвимости или скрытой возможности, которые можно найти и проэксплуатировать.

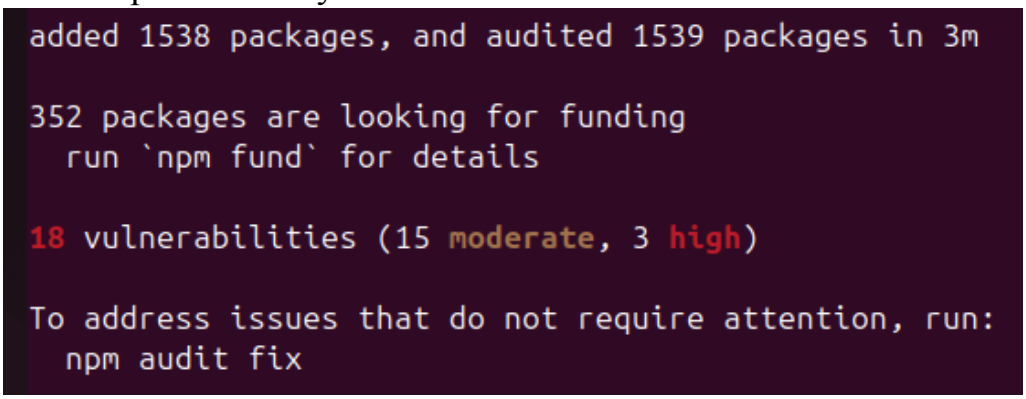
Подробное руководство, список задач (challenges), подсказки и решения приведены на официальном сайте «Pwning OWASP Juice Shop»: https://pwning.owasp-juice.shop/companion-guide/latest/.

Juice Shop включает уязвимости почти из всех пунктов OWASP Top-10 (https://owasp.org/www-project-top-ten/) и множества других реальных ошибок. **Juice Shop** это пример того, как делать **не** надо, «песочница» для безопасного практического обучения, с которой можно начинать изучение безопасности веб-приложений.

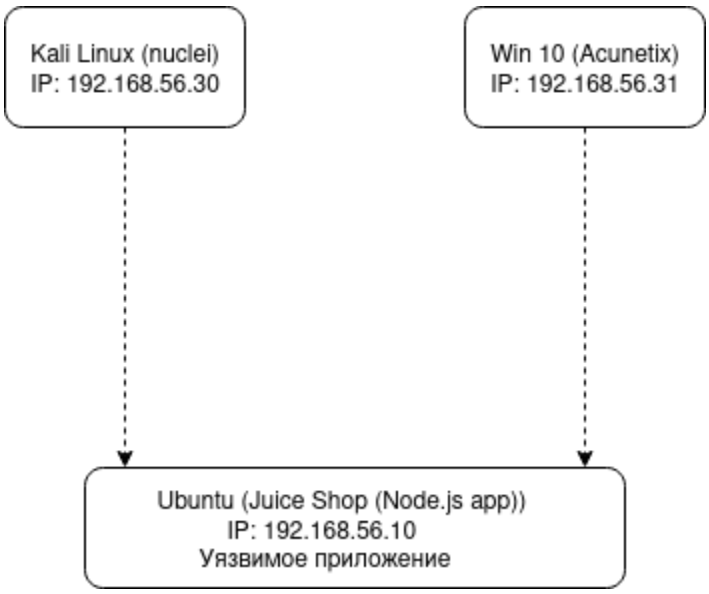
В процессе установки приложения можно увидеть такие предупреждения менеджера пакетов **npm (**Node.js) – это нормально, так и задумано, npm подтягивает уязвимые версии пакетов:



В конце установки можно увидеть отчет установщика с общим количеством используемых в приложении уязвимых зависимостей:



Состав стенда:



Практическая часть

Есть 2 варианта развертывания стенда:

1) Скачать сжатые архивы с виртуальными машинами (VM) **для VirtualBox** по ссылке на Yandex Disk (суммарный размер архивов 18+ ГБ):
   - Ubuntu 24.04 (Juice Shop): https://disk.yandex.ru/d/KRaifKXExf42ow;
   - Kali Linux (nuclei): https://disk.yandex.ru/d/f6b5l86yAGPKDg;
   - Windows 10 (Acunetix): https://disk.yandex.ru/d/n_zdAg_CiHCbhw.

После скачивания и открытия архивов (tar => 7z) нужно завести все 3 VM в одну сеть, например, 192.168.56.0/24, запустить "Juice Shop" и сделать снимок (snapshot) текущего состояния Ubuntu 24.04 (Juice Shop), чтобы потом после работы с уязвимостями и сканированиями иметь возможность возвратиться в исходное состояние.

2) Развернуть самостоятельно (все необходимые команды по установке приведены в приложении к лабе).

1. Ручной поиск уязвимостей (скрытых возможностей) web-приложения

Помимо автоматизированного поиска уязвимостей и скрытых возможностей приложения также используются ручные методы – первым заданием является поиск скрытой страницы (score-board).

Заходим в систему (**login**: user; **password**: user).

Запускаем терминал (псевдотерминал): ***Ctrl+Alt+T***

Переходим в каталог ~/juice-shop: ***cd ~/juice-shop/***

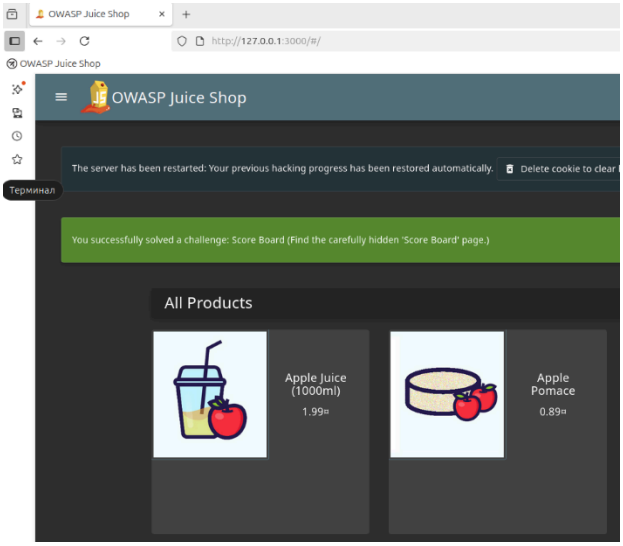
Запускаем наше уязвимое приложение в терминале:

npm start

В выводе должна появиться строка: *info: Port 3000 is available (OK).*

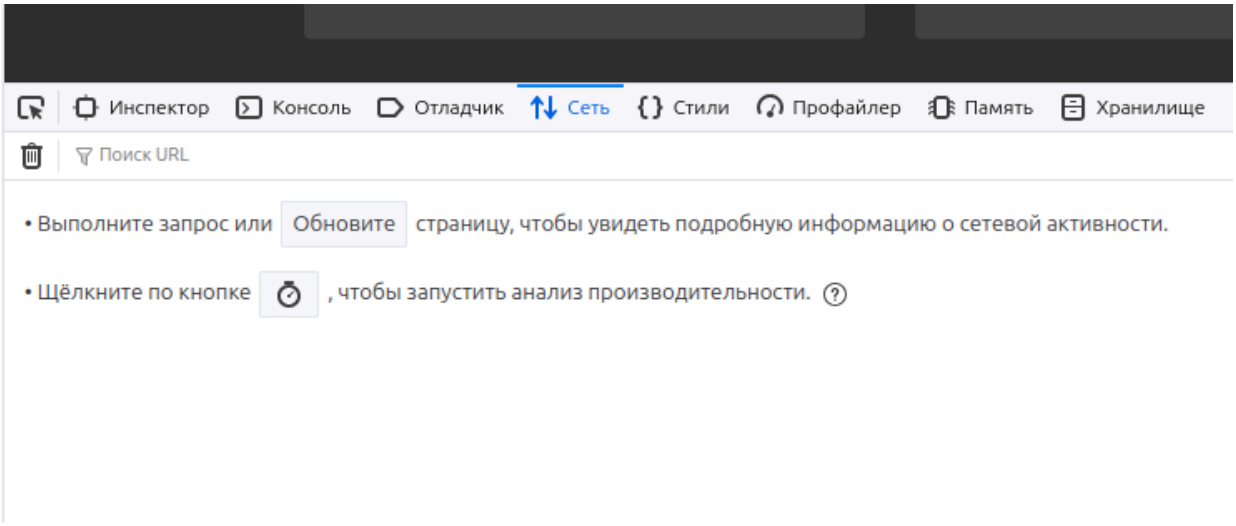
Запускаем браузер Firefox и в строке адреса вводим:

http://127.0.0.1:3000

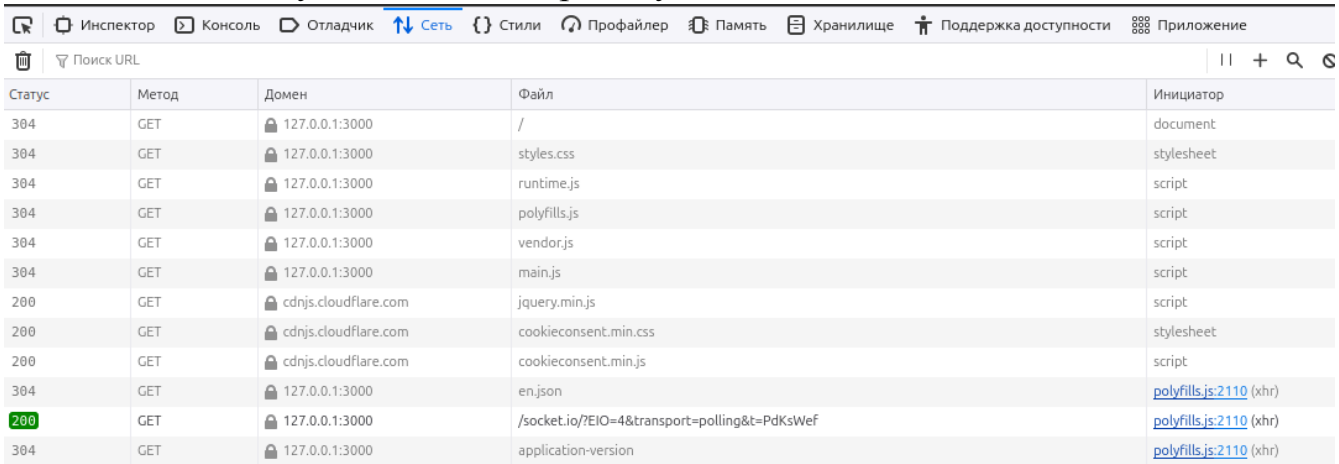


Появится главное окно web-приложения.

Открываем главную страницу приложения (http://127.0.0.1:3000), нажимаем F12 (откроются «Инструменты разработчика», DevTools), переходим во вкладку «Сеть»:



Жмем кнопку «Обновите» страницу:



И путем внимательного просмотра ищем в столбце «Файл» ищем всё, что имеет отношение к заданиям («Challenge»), обязательно **прикладываем скриншот** с процессом поиска.

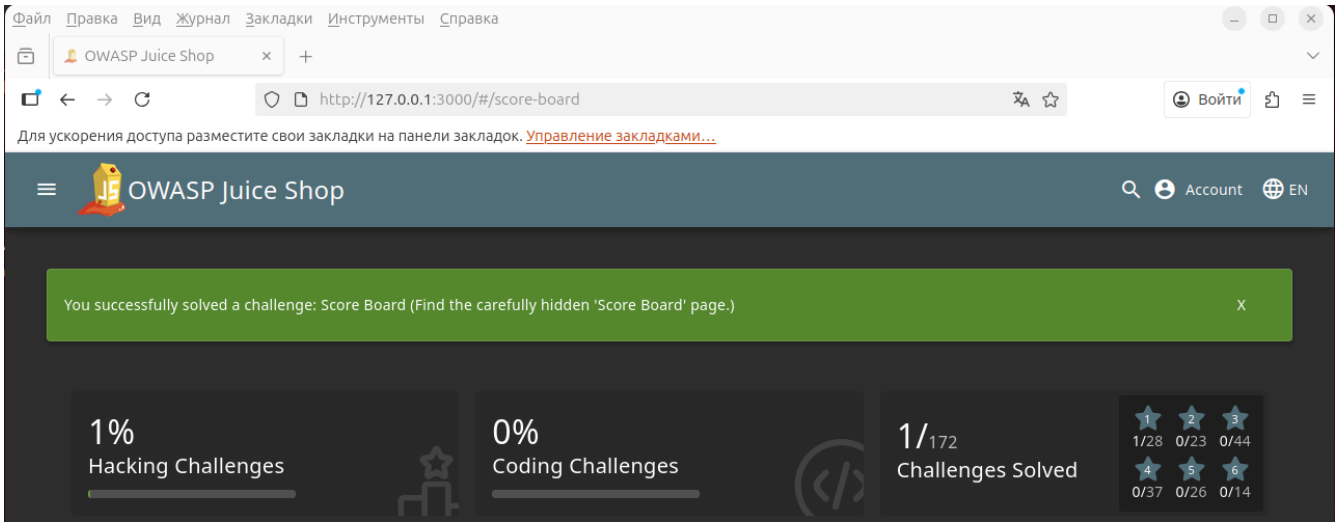
Далее подразумевается осознанный поиск в Отладчике по ключевым словам. Учитывая, что Juice Shop – это Angular-приложение, у которого маршруты имеют вид:

#/login
#/register
#/contact

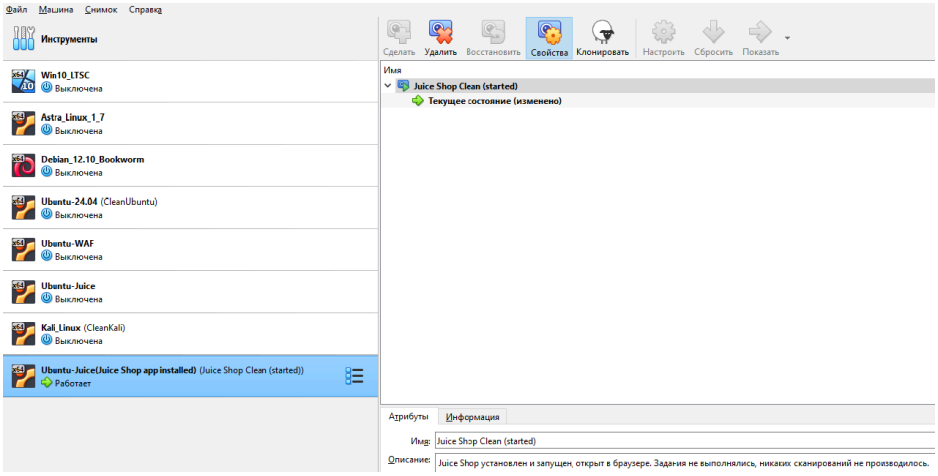
нужно догадаться, что скрытая страница располагается по следующему пути:

http://127.0.0.1:3000/#/score-board

Важное примечание: после выполнения очередного задания в Juice Shop или **после проведения сканирования** на странице score board будут появляться новые уведомления (зеленый прямоугольник с указанием решенного задания) о том, что вы нашли уязвимость. По количеству появившихся записей в score board можно косвенно судить о количестве уязвимостей, которые выявил анализатор.



Чтобы откатиться назад в базовое состояние нужно сделать snapshot (*для тех, кто разворачивает стенд самостоятельно*) в VirtualBox и откатываться на него после каждого выполнения сканирования (сканировать будем с помощью 2-х средств).


Файл Машина Снимок Справка
Инструменты
Win10_LTSC
Выключена
Astra_Linux_1_7
Выключена
Debian_12.10_Bookworm
Выключена
Ubuntu-24.04 (CleanUbuntu)
Выключена
Ubuntu-WAF
Выключена
Ubuntu-Juice
Выключена
Kali_Linux (CleanKali)
Выключена
Ubuntu-Juice(Juice Shop app installed) (Juice Shop Clean (started))
Работает
Сделать
Удалить
Восстановить
Свойства
Клонировать
Настроить
Сбросить
Показать
Имя
Juice Shop Clean (started)
Текущее состояние (изменено)
Атрибуты
Информация
Имя:
Juice Shop Clean (started)
Описание:
Juice Shop установлен и запущен, открыт в браузере. Задания не выполнялись, никаких сканирований не производилось.

2. Автоматизированный поиск уязвимостей web-приложения.

2.1. Работа со сканером уязвимостей nuclei

Nuclei – это open-source инструмент для автоматизированного сканирования уязвимостей, написанный на Go и разработанный проектом ProjectDiscovery.

Работает на основе yaml-шаблонов (templates), в которых описаны паттерны уязвимостей, запросы и условия срабатывания.

Используется в bug bounty, DevSecOps-пайплайнах и Red Team-тестах.

На рабочем столе Kali Linux есть файл **launch_nuclei.txt** – из него можете взять команду запуска nuclei:

export PROJECT=~/nuclei_lab

```
nuclei \
  -u http://192.168.56.10:3000 \
  -headless \
  -code \
  -severity low,medium,high,critical \
  -no-interactsh \
  -no-mhe \
  -retries 3 \
  -markdown-export $PROJECT/nuclei_results \
  -exclude-tags dos
```

С результатами работы сканера можно ознакомиться в каталоге с проектом.

2.2. Работа со сканером уязвимостей Acunetix.

В сканере уязвимостей Acunetix реализован классический механизм DAST (Dynamic Application Security Testing).

Таким образом он:

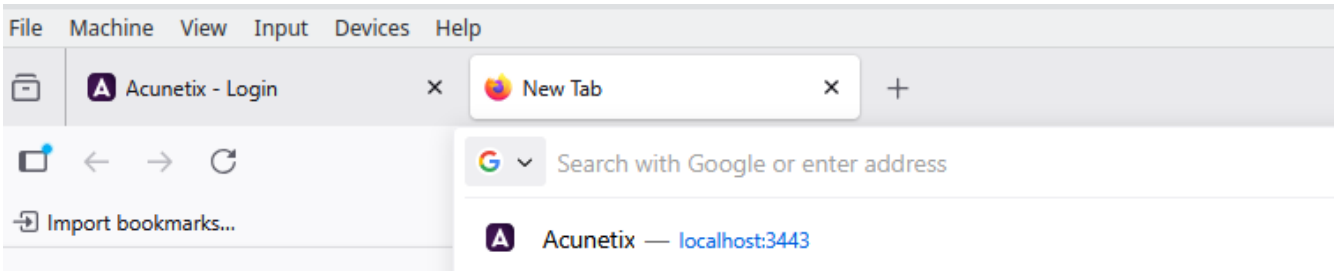
- анализирует поведение веб-приложения во время его работы;
- умеет выполнять «crawl & click»-сканирование, т.е. сканер не только посылает HTTP-запросы по списку URL, но и имитирует действия реального пользователя в приложении: кликает по ссылкам, заполняет и отправляет формы, нажимает кнопки, переходит по выпадающим меню, выполняет JavaScript-события, переходит по ссылкам, заполняет формы, пытается пройти аутентификацию.

То есть Acunetix моделирует поведение реального пользователя и злоумышленника, а не просто отправляет фиксированные запросы.

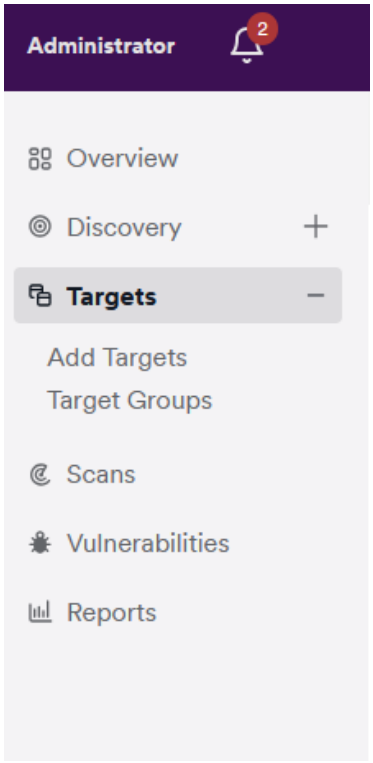
Запуск Acunetix:

Запустить виртуальную машину Win10-Base, убедиться, что она находится во внутренней сети VirtualBox, запустить сканер Acunetix (работает на порту 3443). Для этого нужно открыть браузер Mozilla Firefox и в строке адреса написать: https://localhost:3443/

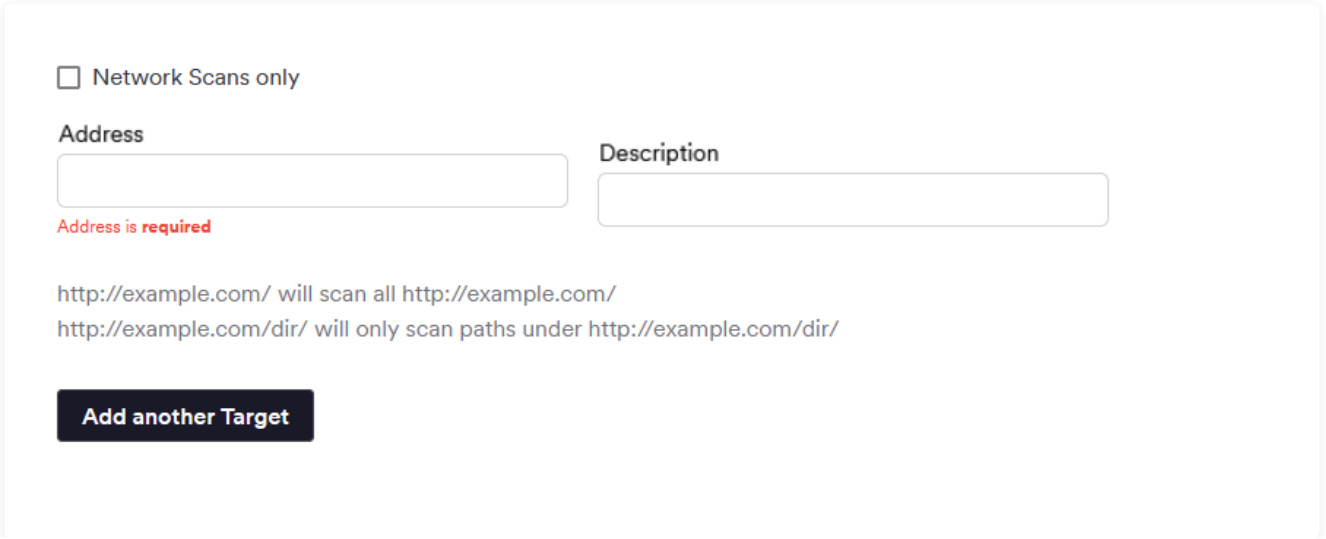
В появившемся окне ввести логин: logdir@proton.me и пароль: **#565656-ACUN**



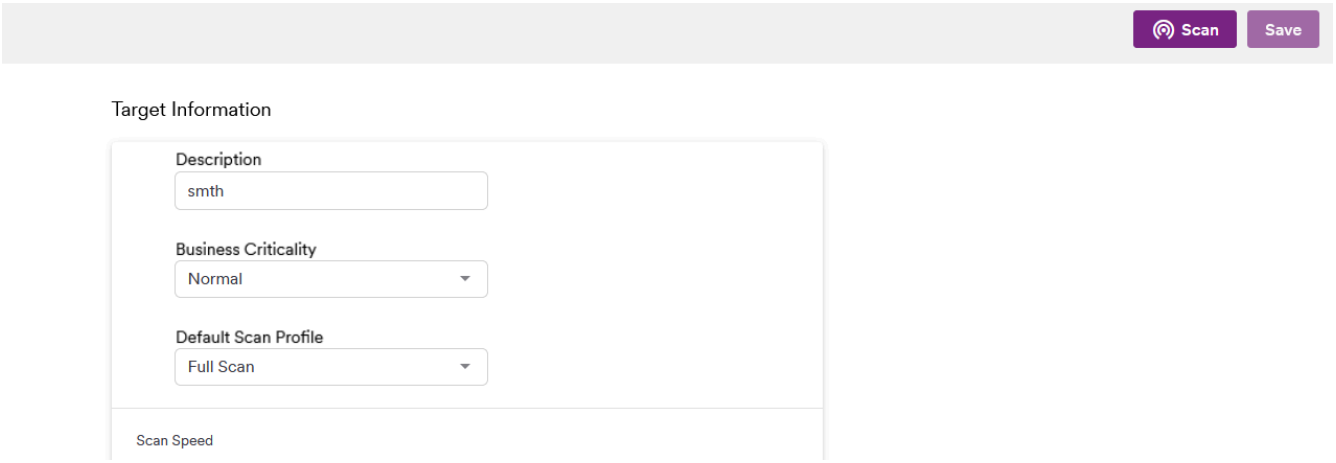
Далее необходимо выбрать цель (target). Для этого переходим в раздел Targets на левой панели приложения:



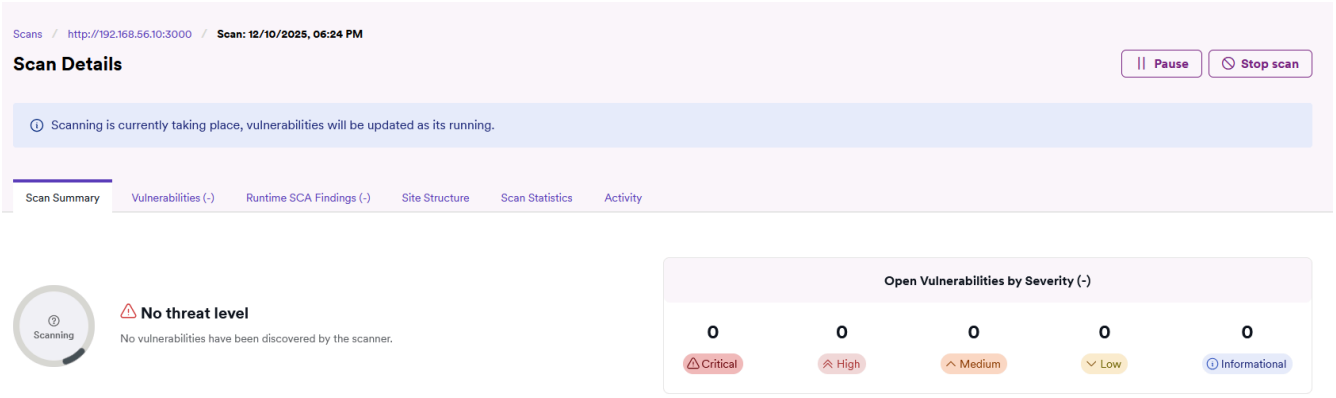
Выбираем Add Target:



В поле Address вводим http://192.168.56.10:3000/ в пояснении (Description) что-то типа JuiceShop-Lab (на своё усмотрение). Нажимаем кнопку Save, а затем Scan:

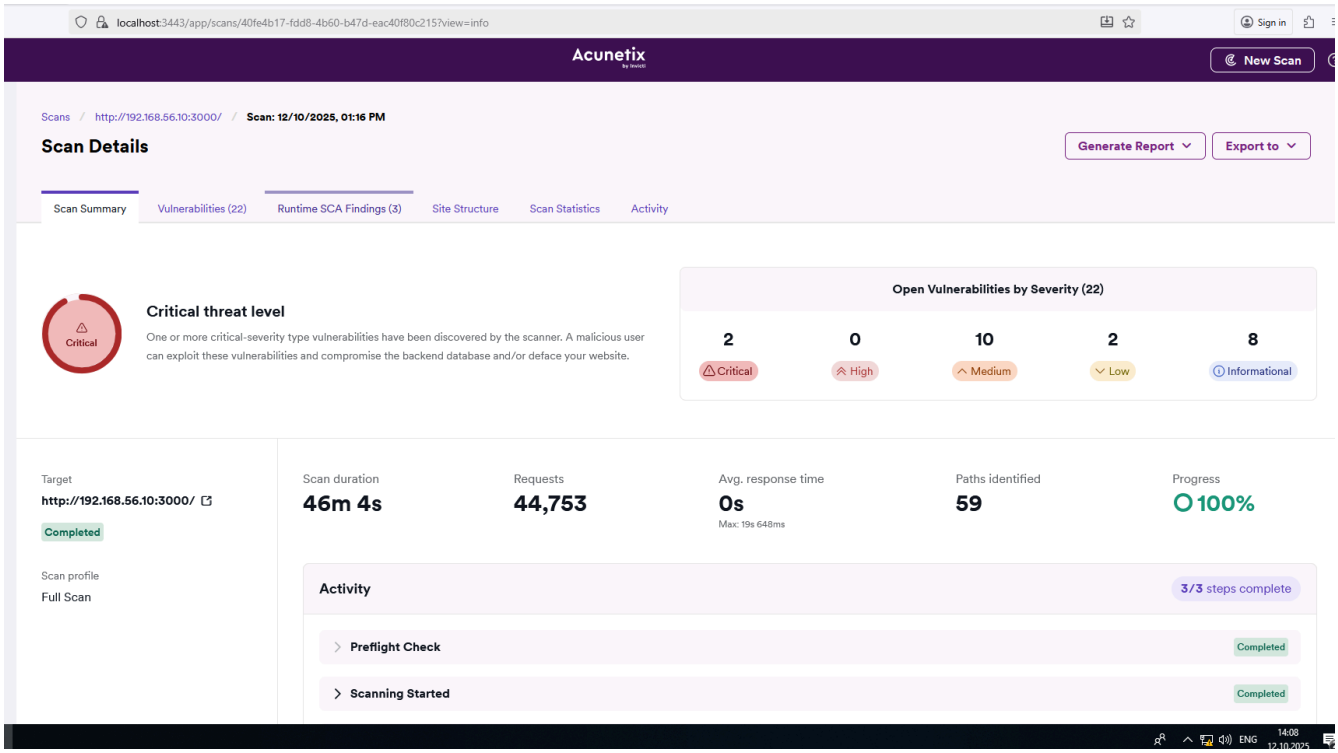


Сканирование запущено:



После запуска Acunetix смело отправляйтесь покушать и выпить чашечку чая/кофе (процесс занимает 35-45 минут).

После завершения сканирования необходимо сохранить отчет о сканировании (например, в формате pdf), проанализировать его, сделать краткие выводы и записать их в отчет о лабе.



Приложение к лабораторной работе № 5

1. Подготовка машины Ubuntu-Juice (IP: 192.168.56.10)

1.1 Установка Node 22 LTS (22.12.x)

```
# Забираем архив и распаковываем в /usr/local
cd /tmp
curl -fsSLO https://nodejs.org/dist/v22.12.0/node-v22.12.0-linux-x64.tar.xz
sudo tar -C /usr/local --strip-components=1 -xJf node-v22.12.0-linux-x64.tar.xz

# Обновим кеш путей в текущем шелле
hash -r

# Проверим, что всё в PATH
which node
which npm
node -v
npm -v
```

Если which npm ничего не выводит, добавляем /usr/local/bin в PATH и перезапускаем bash:

```
echo 'export PATH=/usr/local/bin:$PATH' >> ~/.bashrc
exec $SHELL -l
```

1.2 Сборка проекта (Juice Shop)

ПРИМЕЧАНИЕ: при сборке будет очень много warning'ов – так и должно быть.

```
# Клонируем проект и переходим в каталог:
sudo git clone https://github.com/juice-shop/juice-shop.git
cd juice-shop

# Проверяем содержимое (д.б. файлы server.js, package.json, frontend/ и т.д.).
ls ~/juice-shop

# Чистим кэш npm
npm cache clean --force


# Ставим зависимости
npm install --legacy-peer-deps

# Смотрим список опций по сборке проекта:
npm run
```

```
# Собираем проект:
npm run build:frontend   # собирает Angular во frontend/
npm run build:server     # компилирует сервер (tsc)
npm start                # запускает собранное приложение (node build/app)
```

Если всё сделали правильно, то должны получить сообщение «info: Server listening on port 3000».

2. Подготовка машины Kali-Linux (IP: 192.168.56.30)

Подготовка соответствует шагам, перечисленным в лабе № 3 (установить golang нужной версии, pdtm, а затем – nuclei).

Дополнительно к тому, что вы уже поставили, нужно поставить google chrome (https://go-rod.github.io/#/compatibility?id=os).

Также нужно перед первым запуском с тестированием Juice Shop дать nuclei выход в интернет (можно натравить его на эту цель: **http://testphp.vulnweb.com**)

```
##################################################################
# ####### тестируем nuclei на внешнем ресурсе #######
# устанавливаем переменную shell:
export PROJECT=~/nuclei_lab

# непосредственно запуск:
nuclei \
  -u http://testphp.vulnweb.com \
  -headless \
  -code \
  -severity low,medium,high,critical \
  -no-interactsh \
  -no-mhe \
  -retries 3 \
  -markdown-export $PROJECT/nuclei_results \
  -exclude-tags dos
##################################################################
```

3. Подготовка машины Win10-Acunetix (IP: 192.168.56.31)

Нужно развернуть виртуальную машину с Windows 10/11 и установить анализатор уязвимостей Acunetix. Он платный, в лабе будем использовать ломаную версию. Ссылка на Яндекс Диск для скачивания архива с Acunetix: https://disk.yandex.ru/d/eRnPxGjBOf0yaA .

После скачивания файла его нужно разархивировать (ПАРОЛЬ для распаковывания архива:**Pwn3rzs**). После чего внимательно ознакомиться с содержимым файла ReadMe и установить сканер в соответствии с ReadMe.