

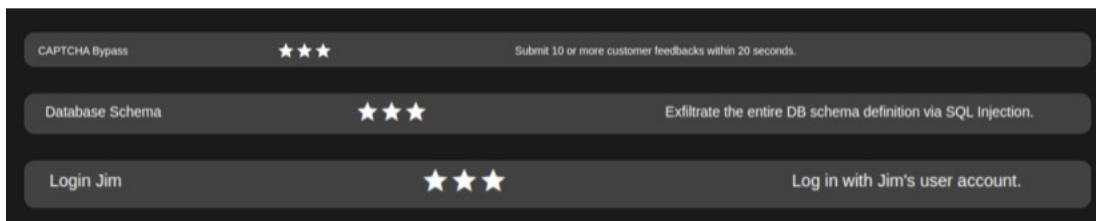
## Раздел: Анализ Защищенности ПО (на веб)

### Модуль 3: Средства автоматизированного поиска уязвимостей в веб-приложениях (HW)

Выполнил: Александр Ганицев

#### Задание.

Пройдите следующие лабораторные работы в Juice Shop:



Также

проанализируйте фрагменты кода на наличие уязвимостей. Используйте дополнительные правила (любые) для сканирования кода:

Фрагмент № 1.

Фрагмент № 2.

Фрагмент № 3.

#### Условия реализации.

В качестве отчёта предоставьте следующие скриншоты:

Скриншот выполненной работы CAPTCHA Bypass из таблицы Scoreboard Juice Shop.

Скриншот выполненной работы Database Schema из таблицы Scoreboard Juice Shop.

Скриншот выполненной работы Login Jim из таблицы Scoreboard Juice Shop

А также:

Команду для сканирования файлов на наличие уязвимостей.

Скриншот терминала с результатами сканирования.

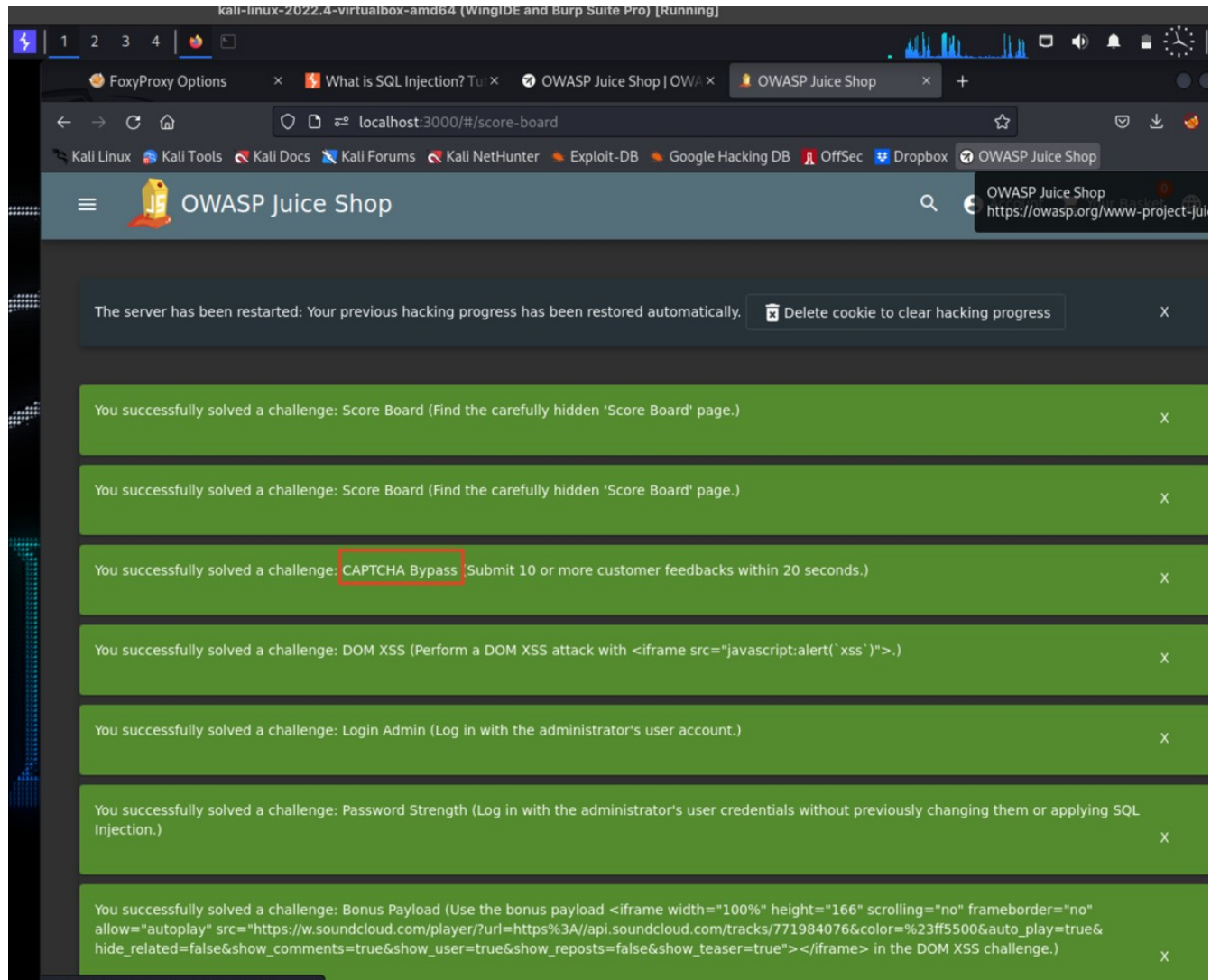
Уязвимости, содержащиеся в приведённых фрагментах кода, в формате: «название файла» — «уязвимость».

Отчёт создайте в формате DOCX или PDF и загрузите в свой гит. Ссылку на гит отправьте ментору.

## Выполнение задания.

### 1. Лабораторные работы Juicy-Shop.

#### 1.1. CAPTCHA Bypass:



#### 2.1. Login Jim и все остальные доступные задания:

Name	Difficulty	Description	Category	Status
Score Board	★	Find the carefully hidden 'Score Board' page.	Miscellaneous	<input checked="" type="checkbox"/>
DOM XSS	★	Perform a DOM XSS attack with <code>&lt;iframe src="javascript:alert('xss')"&gt;</code> .	XSS	<input checked="" type="checkbox"/>
Bonus Payload	★	Use the bonus payload <code>&lt;iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&amp;color=%23ff5500&amp;auto_play=true&amp;hide_related=false&amp;show_comments=true&amp;show_user=true&amp;show_reposts=false&amp;show_teaser=true"&gt;&lt;/iframe&gt;</code> in the DOM XSS challenge.	XSS	<input checked="" type="checkbox"/>
Privacy Policy	★	Read our privacy policy.	Miscellaneous	<input checked="" type="checkbox"/>
Login Admin	★★	Log in with the administrator's user account.	Injection	<input checked="" type="checkbox"/>
Password Strength	★★	Log in with the administrator's user credentials without previously changing them or applying SQL Broken Authentication Injection.		<input checked="" type="checkbox"/>
View Basket	★★	View another user's shopping basket.	Broken Access Control	<input checked="" type="checkbox"/>
Forged Feedback	★★★	Post some feedback in another user's name.	Broken Access Control	<input checked="" type="checkbox"/>
Login Jim	★★★	Log in with Jim's user account.	Injection	<input checked="" type="checkbox"/>
Login Bender	★★★	Log in with Bender's user account.	Injection	<input checked="" type="checkbox"/>

### 3.1. Database Schema.

Я не смог открыть это задание ни в установленном с Гита Docker-образе, ни с официального сайта:

OWASP Juice Shop | OWASP Juice Shop
404 Not Found

https://pwning.owasp-juice.shop/part1/challenges.html#potentially-dangerous-challenges

Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec Dropbox OWASP Juice Shop

## Not Found

The requested URL was not found on this server.

Injection	11	Christmas Special, Database Schema, Ephemeral Accountant, Login Admin, Login Bender, Login Jim, NoSQL DoS, NoSQL Exfiltration, NoSQL Manipulation, SSTi, User Credentials
-----------	----	---

## 2. Сканирование кода на уязвимости.

2.1. Склонировал мой GitHub main repository на Ubuntu 22.10 vm. Попытки сканирования провалились, по причине слишком длинных имён (я использую систему вложенных директорий с включением Русского языка).

```
skillfactory_lab@Ubuntu22:~/ProgrammingPython/MIFIIB$ semgrep scan --config auto

+-----+
| Scan Status |
+-----+
[Errno 36] File name too long: '"\\320\\220\\320\\275\\320\\260\\320\\273\\320\\270\\320\\267 \\320\\267\\320\\260\\321\\211\\320\\270\\321\\211\\320\\265\\320\\275\\320\\275\\320\\276\\321\\201\\321\\202\\320\\270 \\320\\270\\320\\275\\321\\204\\321\\200\\320\\260\\321\\201\\321\\202\\321\\200\\321\\203\\320\\272\\321\\202\\321\\203\\321\\200\\321\\213\\320\\234\\320\\276\\320\\264\\321\\203\\320\\273\\321\\214 1. \\320\\242\\320\\265\\320\\276\\321\\200\\320\\270\\321\\217 \\320\\270 \\320\\277\\321\\200\\320\\260\\320\\272\\321\\202\\320\\270\\320\\272\\320\\260 \\320\\277\\320\\276\\320\\270\\321\\201\\320\\272\\320\\260 \\321\\203\\321\\217\\320\\267\\320\\262\\320\\270\\320\\274\\320\\276\\321\\201\\321\\202\\320\\265\\320\\271 OWASP Top-10/Labs_Portswigger.net.txt"'
Traceback (most recent call last):
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/commands/wrapper.py", line 35, in wrapper
    func(*args, **kwargs)
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/commands/scan.py", line 865, in scan
    ) = semgrep.run_scan.run_scan(
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/run_scan.py", line 520, in run_scan
    ) = run_rules(
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/run_scan.py", line 29, in run_rules
    num_executed_rules = print_scan_status(filtered_rules, target_manager)
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/run_scan.py", line 153, in print_scan_status
    sast_plan = CoreRunner.plan_core_run(
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/core_runner.py", line 900, in plan_core_run
    lockfiles = target_manager.get_all_lockfiles()
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/target_manager.py", line 791, in get_all_lockfiles
    return {
```

Затем я перенес три фрагмента для сканирования в корень моего репозитория, выполнил git pull, но снова упёрся в ошибки.

```
▼ MIFIIB
> .github
> 1_Python_Course
> 2_GrayHatPython
> 3_CoolThings
> 4_BeyondBasicStuff
> Linux_OS
> module_1_9
> module_2_6
> module_3_and_4
> module_5
> module_6
> module_7
> venv
> Windows_OS
> Анализ защищенности инфраструктуры
> Жизненный цикл ИБ
> Целенаправленные атаки
📄 .gitignore 2023-06-15, 10:32 a.m., 355 B 2023-06-10, 1:23 p.m.
📄 find_vuln6.py 2023-09-05, 3:38 p.m., 303 B
📄 find_vuln7.js 2023-09-05, 3:38 p.m., 850 B
📄 find_vuln8.php 2023-09-05, 3:39 p.m., 389 B
📄 README.md 2023-06-10, 1:24 p.m., 747 B 36 minutes ago
> share
📄 .gitignore 2022-11-08, 11:09 a.m., 40 B 2023-04-04, 4:34 p.m.
📄 pyvenv.cfg 2023-02-08, 3:31 p.m., 350 B
📄 typescript 2022-11-08, 12:01 p.m., 592 B
```

После, я на локальной vm удалил всё, кроме этих трёх фрагментов для сканирования, но тем не менее, Semgrep упирается в ошибки конфигурации Python в системе.

```
skillfactory_lab@Ubuntu22: ~/ProgrammingPython

+-----+
| Scan Status |
+-----+
Scanning 4 files tracked by git with 1671 Code rules, 580 Pro rules:

  Language | Rules | Files | Origin | Rules
  +-----+ +-----+ +-----+ +-----+ +-----+
  <multilang> | 58 | 12 | Community | 1091
  python | 349 | 1 | Pro rules | 580
  js | 241 | 1 |
  php | 60 | 1 |

----- 100% 0:00:03

Received output encoding error, please set PYTHONIOENCODING=utf-8
Traceback (most recent call last):
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/output.py", line 380
, in output
    print(output)
UnicodeEncodeError: 'latin-1' codec can't encode character '\u2506' in position 937: ordinal no
t in range(256)

The above exception was the direct cause of the following exception:

Traceback (most recent call last):
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/commands/wrapper.py"
, line 35, in wrapper
    func(*args, **kwargs)
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/commands/scan.py", l
ine 900, in scan
    output_handler.output(
  File "/home/skillfactory_lab/.local/lib/python3.10/site-packages/semgrep/output.py", line 382
, in output
    raise Exception(
Exception: Received output encoding error, please set PYTHONIOENCODING=utf-8
```

По всей видимости это и являлось моей проблемой при попытках конфигурации Semgrep и добавления нового проекта (моего GitHub).

Token был создан успешно:

```
skillfactory_lab@Ubuntu22:~/ProgrammingPython$ semgrep login
API token already exists in /home/skillfactory_lab/.semgrep/settings.yml. To login with a diffe
rent token logout use `semgrep logout`
```

При этом, при добавлении новой задачи сканирования на мой репозиторий, у меня вылетает ошибка “Semgrep has crashed”, то есть он крутится на этапе соединения с GitHub:



# Semgrep has crashed

Sorry about this! We've been notified and will get to work on fixing the root cause. Logging out might fix it. Wiping your editor session will lose your work from the editor, but it can fix some crashes that keep coming back when reloading the page.

[Sign out](#) [Wipe editor session](#) [Dismiss error](#)

Settings | Semgrep

123ALEXG

Dashboard

Projects

Code76

Supply Chain


Rules>

Settings

Docs

Help

Updates



Add to GitHub Actions

Using the Semgrep Github app, you can automate scans and PR comments.

Requires admin permissions on repos

> Don't want to install the app?

1. Review permissions

2. Automatically set up CI jobs?

☐ Automatically configure and enable CI jobs for Github Repos.

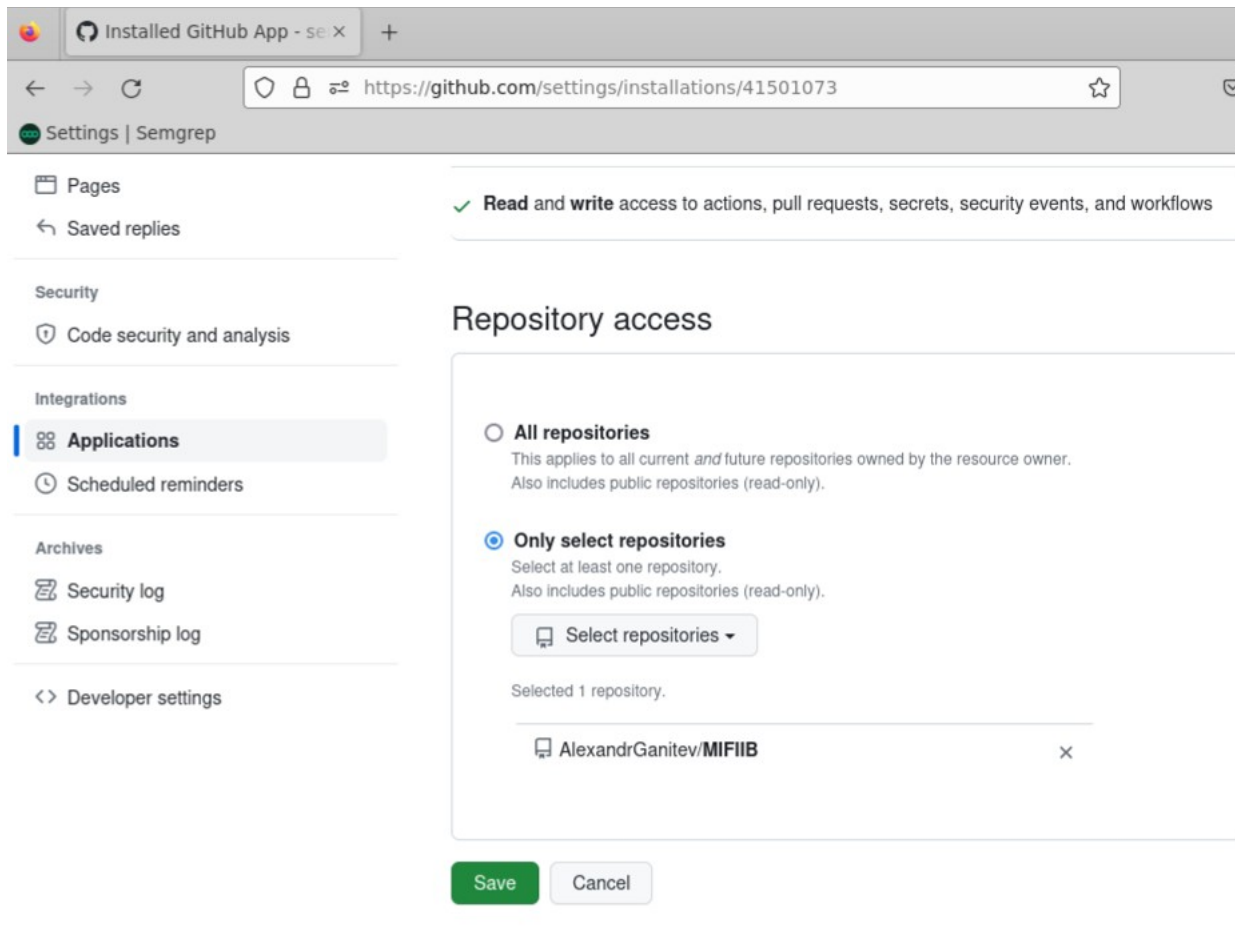
This will create create secrets and commit a semgrep.yml scan configuration file to each repo you select.

☒ Manually configure and enable CI jobs per repo

3. Install Github App

Connect to GitHub

Дальнейшие попытки добавления репозитория не приносят результата:





Alexandr Ganitev (AlexandrGanitev)

Your personal account

[Go to your personal profile](#)

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives



semgrep-app

Installed 6 minutes ago

Developed by [returntoorp](#)

<https://semgrep.dev>

**Semgrep** is a fast, open-source, static analysis tool for modern languages. With 1,500+ existing rules and simple-to-create custom ones, it finds the bugs that matter.

- Open source, works on 20+ languages
- Scan with 1,500+ community rules
- Write rules that look like your code
- Quickly get results in the terminal, editor, or CI/CD
- Flag issues moving forward, get results in pull requests, Slack, + more

This GitHub App allows you to get Semgrep results as PR comments, add Semgrep to your projects with one-click, and manage rules and results across multiple projects from one centralized place. Learn more at [semgrep.dev](https://semgrep.dev).

Semgrep is developed and supported by [r2c](#). It is an evolution of [pfff](#), which began at Facebook in 2009, which itself was an evolution of the Linux refactoring tool [Coccinelle](#).

## Permissions

- ✓ **Write** access to files located at `.github/workflows/semgrep.yml`, `.semgreignore`
- ✓ **Read** access to checks and metadata
- ✓ **Read and write** access to actions, pull requests, secrets, security events, and workflows

## Repository access

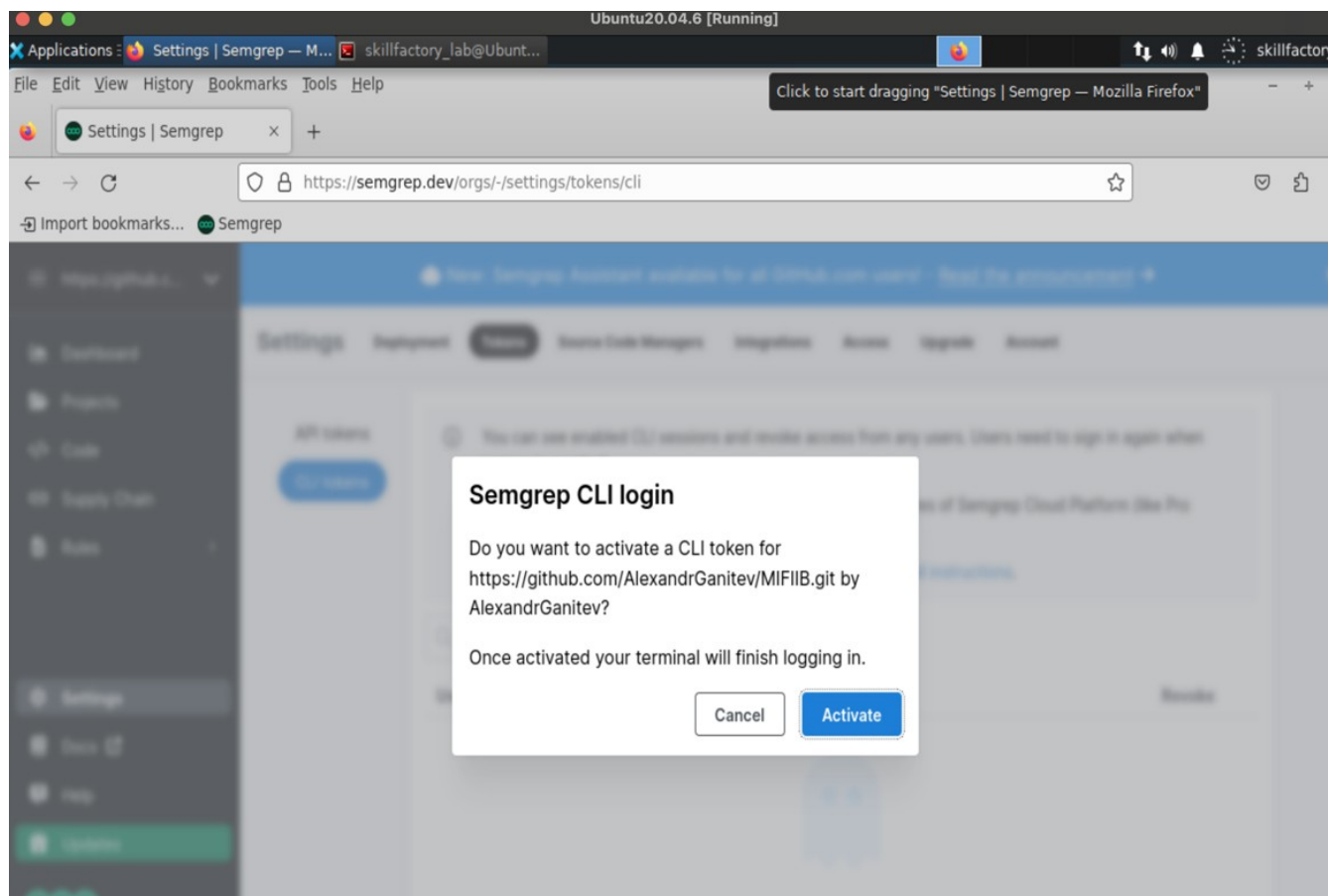
### ☐ All repositories

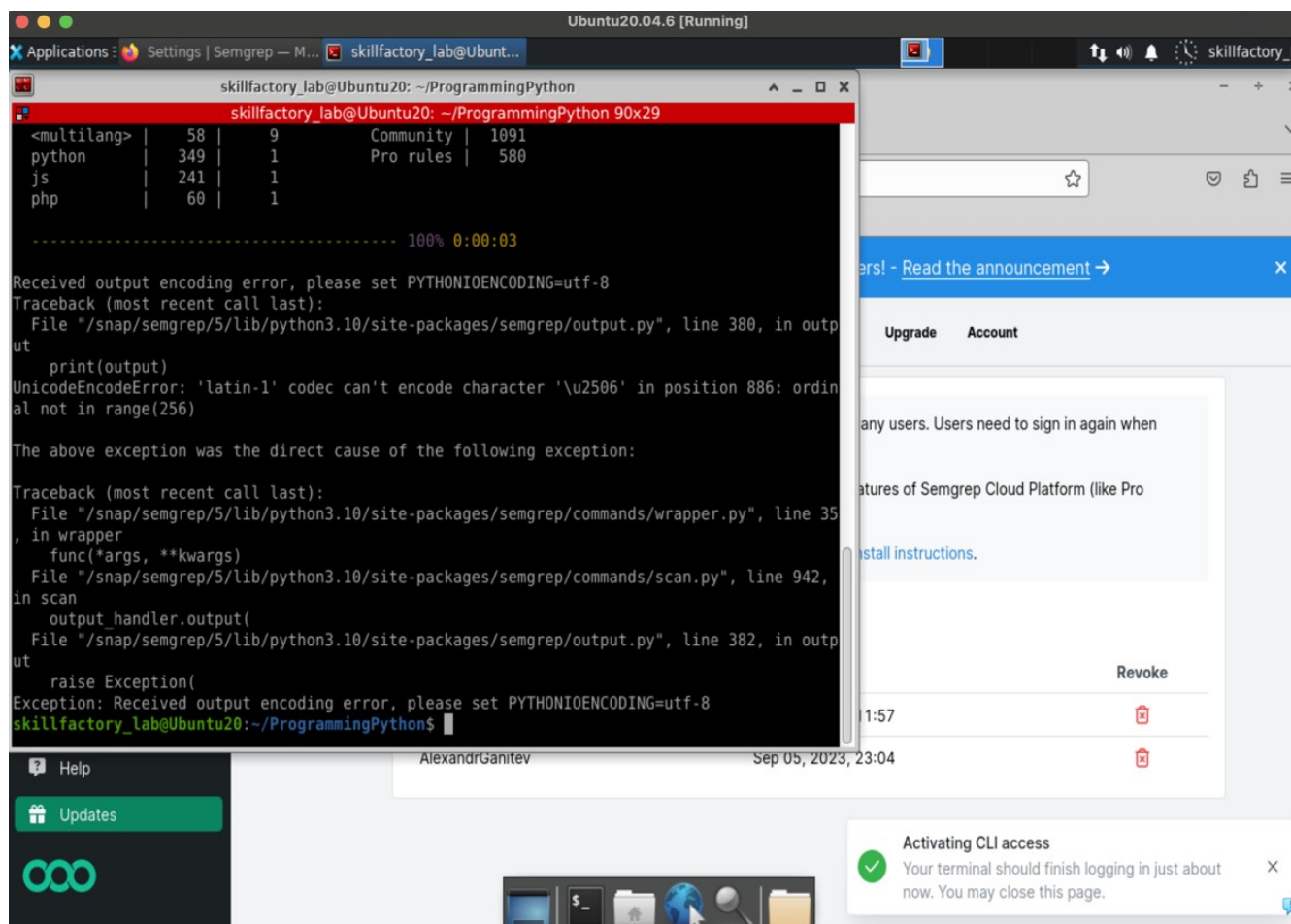
This applies to all future repositories owned by the resource owner.  
Also includes public repositories (read-only).

Также я попробовал те же действия на другой VM и пришёл к тем же результатам, поэтому не смог применить правила.

Ubuntu 20.04.6, здесь Semgrep устанавливался при помощи snap, в отличии от Ubuntu 22.10. Для ускорения процесса сканирования файлы были скопированы вручную:







**Вывод:** для выполнения 2 пункта данного задания необходимо выделить больше времени с объяснением возможных подводных камней и с чётким указанием системы и её версии, в которой будет работать данный анализатор кода.