Описание используемых средств:

1. Статический анализ кода (Static Application Security Testing, SAST) — это метод выявления уязвимостей и недочетов в исходном коде приложения на ранних этапах разработки.

Статический анализ кода использует встроенные правила для поиска недочетов, используя шаблоны уязвимостей, такие как небезопасное формирование SQL выражения в сочетании с пользовательскими данными (SQL инъекция). Это достигается с помощью анализа кода на уровне синтаксического дерева (AST), что позволяет выявлять потенциальные уязвимости на уровне структуры кода.

AST (Abstract Syntax Tree) — это способ представления программы в виде дерева, где каждый узел и ветвь соответствуют определенным элементам или конструкциям языка программирования. Например, операторы, выражения, вызовы функций и другие элементы исходного кода.

Самые популярные инструменты для SAST: SonarQube, Checkmarx, Semgrep, Veracode

2. Dynamic Application Security Testing (DAST) — это метод динамического анализа кода, который используется при тестировании веб-приложений в реальном времени.

В отличие от SAST, DAST работает с скомпилированным, собранным и запущенным приложением, анализируя его поведение и взаимодействие с клиентами. DAST тестирует как клиентскую, так и серверную части приложения. Это позволяет находить уязвимости, которые проявляются только в процессе выполнения приложения.

DAST-инструменты отправляют HTTP/HTTPS-запросы и анализируют ответы, имитируя действия злоумышленника. В отличие от SAST, DAST направлен на поиск реальных уязвимостей, которые могут возникнуть в момент развертывания приложения. Часто бывает так, что SAST обнаруживает уязвимый кусок кода, но во время запуска приложения пользовательский ввод не может дотянуться до этой уязвимости, и поэтому DAST не находит ее. В обратном случае, после статического анализа можно не обнаружить уязвимости, а во время динамического анализа они могут появиться, особенно если возникают проблемы в логике между микросервисами или API. Именно поэтому крайне важно применять комплексный подход к анализу кода.

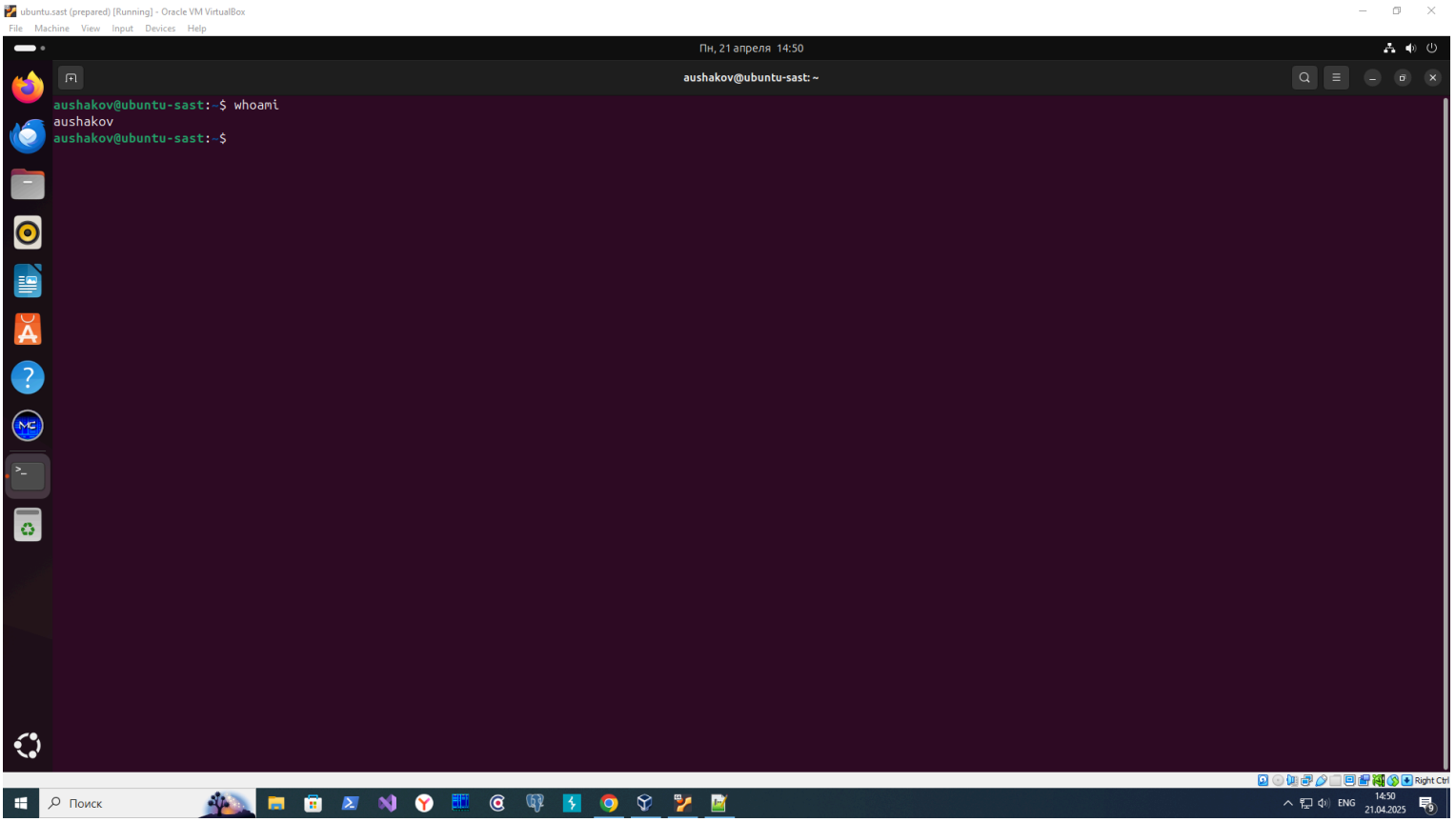
Самые популярные инструменты для DAST: OWASP ZAP, Burp Suite, Acunetix, Netsparker

3. Software Composition Analysis (SCA) — это процесс анализа и управления сторонними библиотеками и зависимостями, используемыми в проектах.

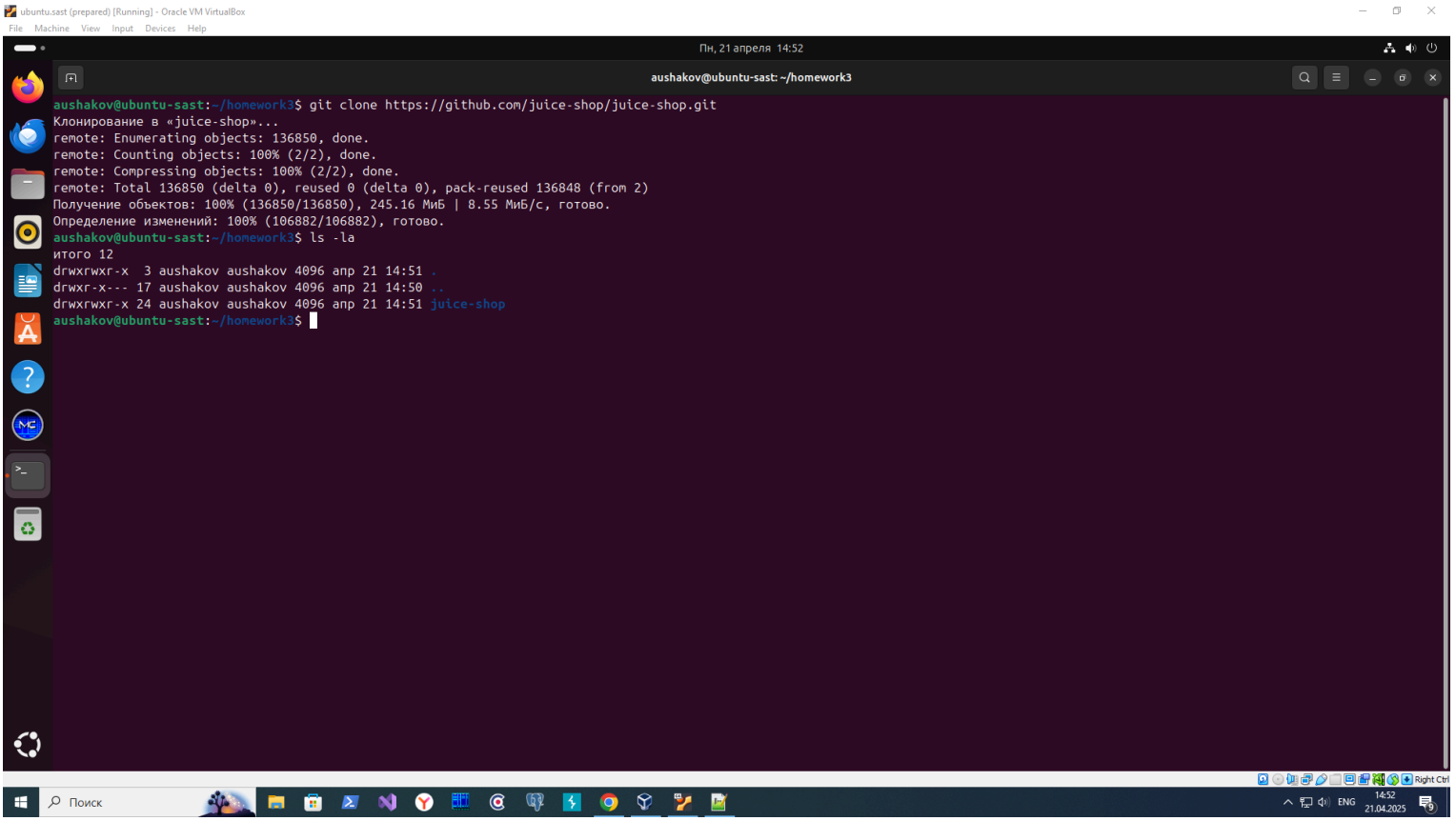
SCA помогает выявлять открытые уязвимости, несоответствие лицензий и потенциальные риски, связанные с использованием сторонних зависимостей, библиотек и кода. Современная разработка активно использует библиотеки, модули и фрагменты кода из открытых источников, что значительно сокращает время разработки. Однако использование таких компонентов связано с рисками наличия известных уязвимостей. SCA-инструменты эффективно справляются с задачей обнаружения подобных рисков, предоставляя рекомендации по их устранению и обеспечивая безопасность приложений.

Самые популярные инструменты для SCA: Dependency-Check, Dependabot, Snyk, WhiteSource (Whitesource Bolt), Black Duck by Synopsys, OSS Index

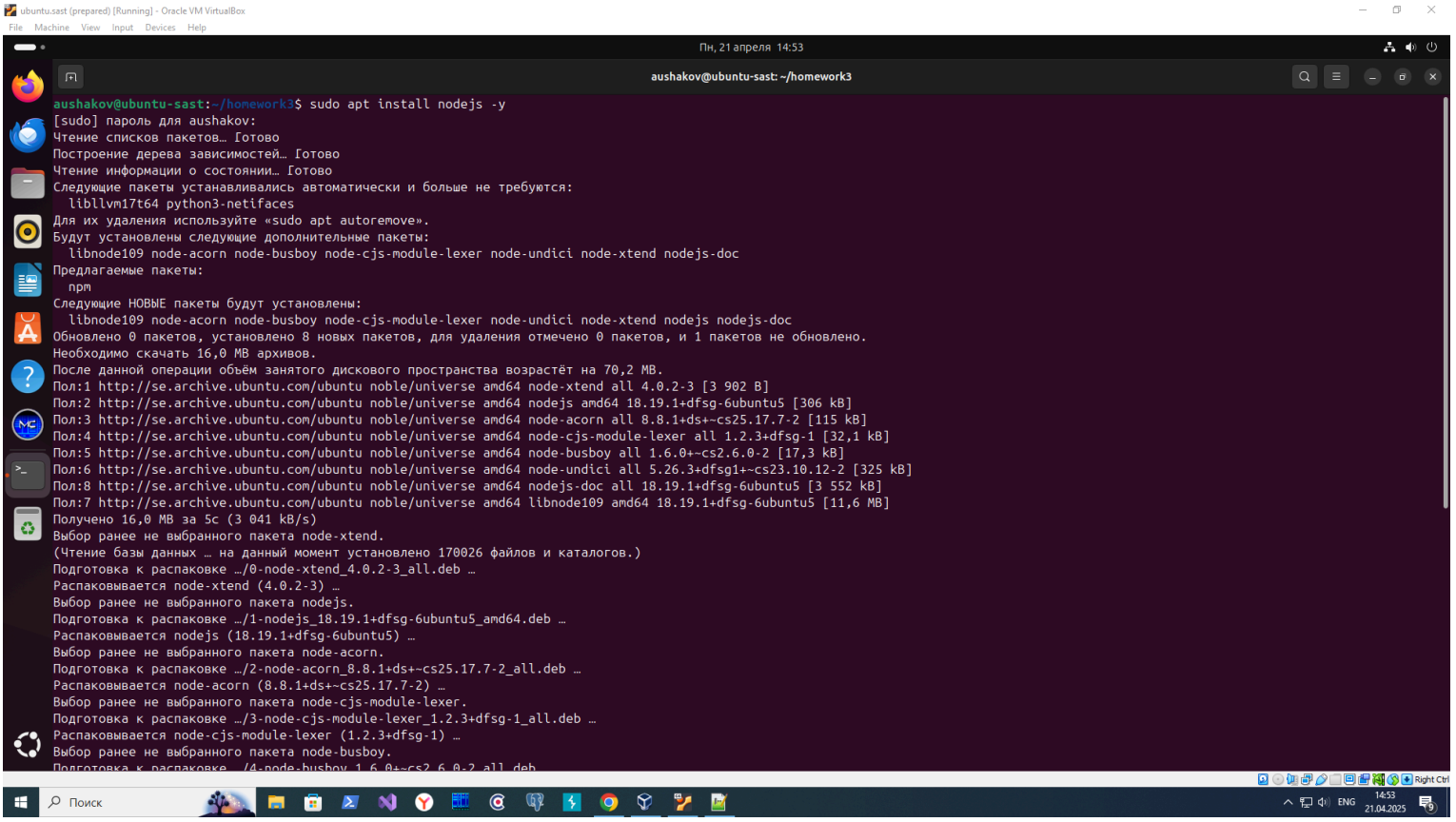
Создаю виртуальную машину для выполнения домашнего задания

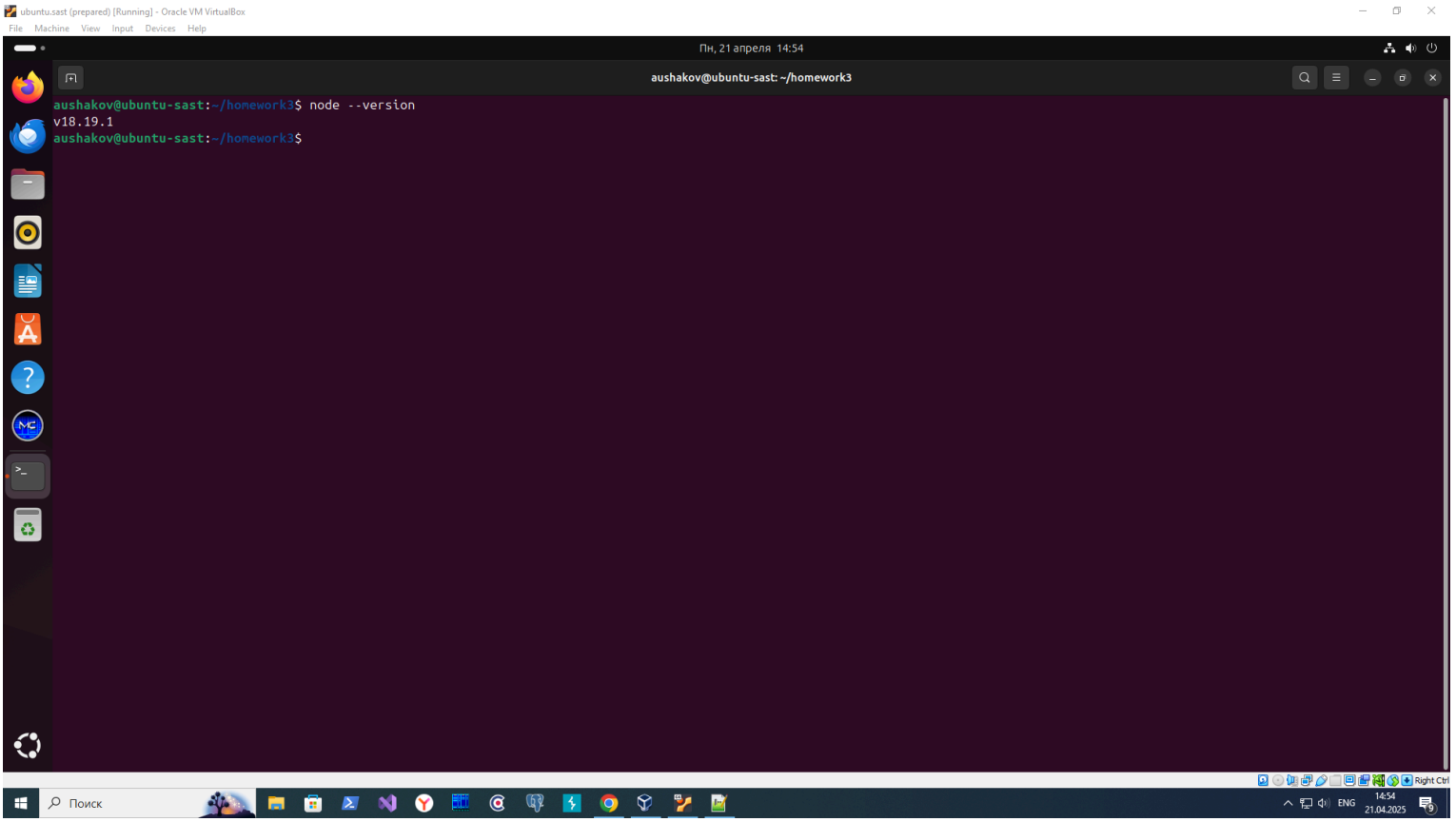


Клонирую репозиторий OWASP Juice Shop:

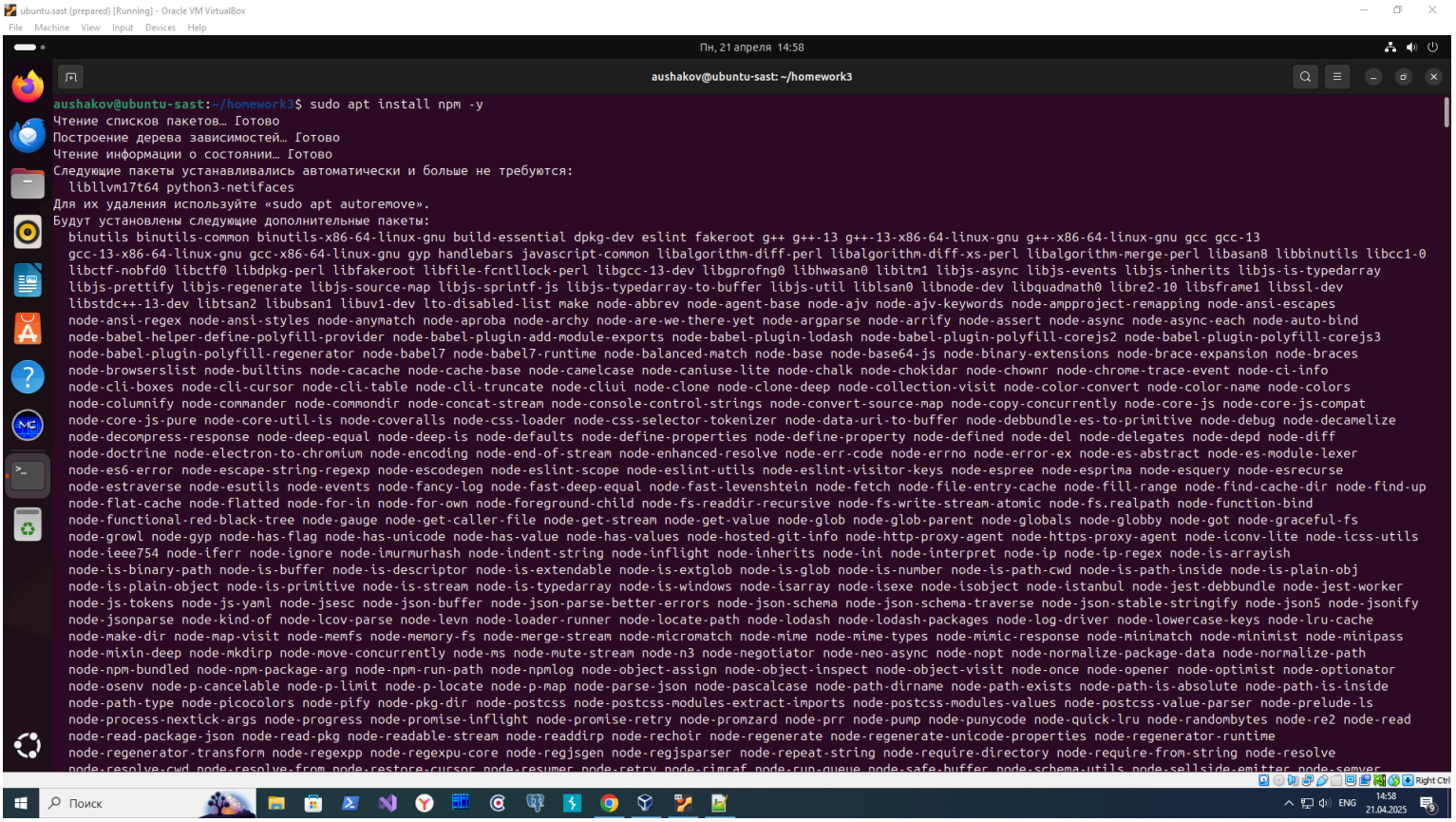


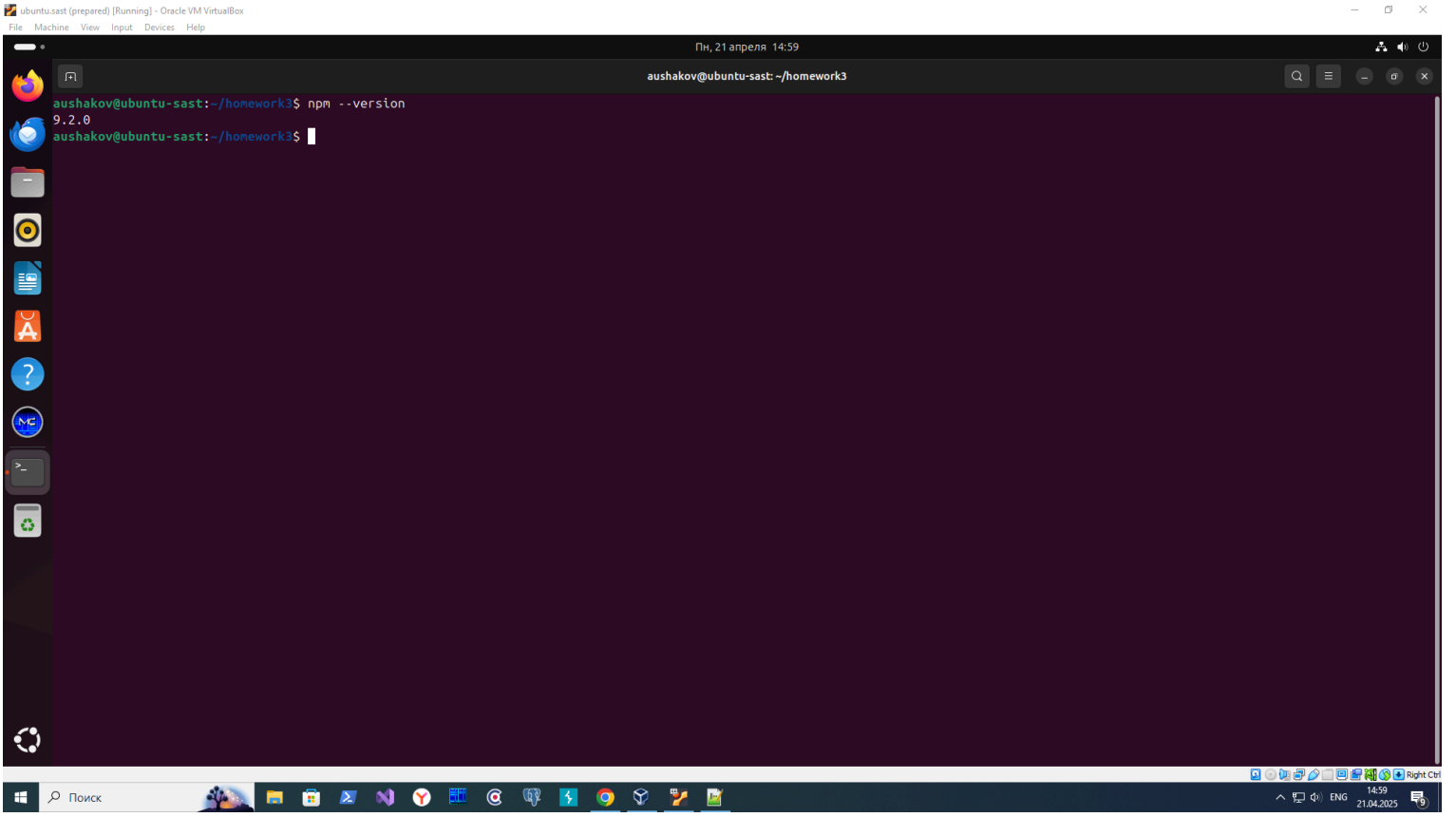
Устанавливаю Node.js



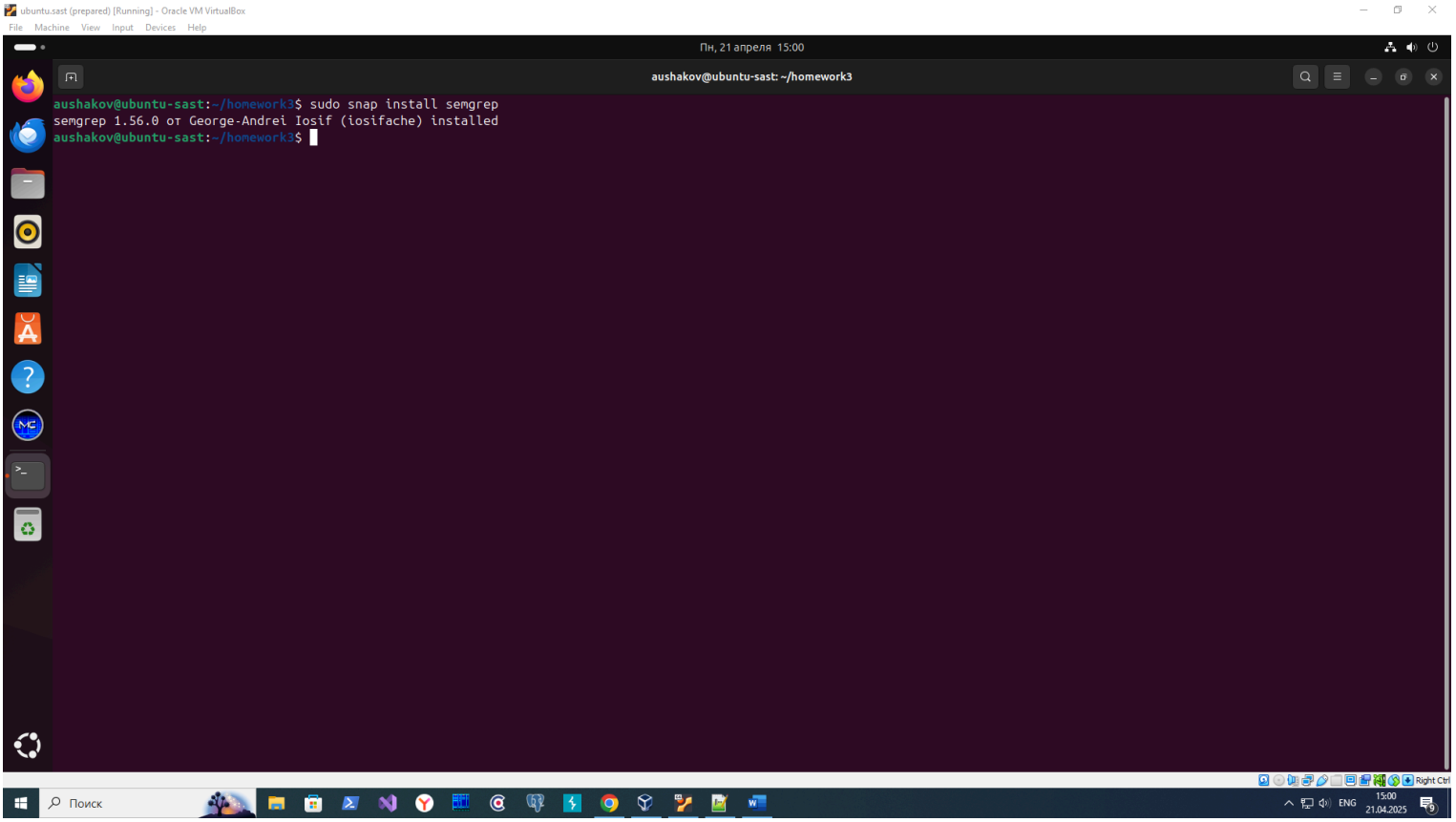


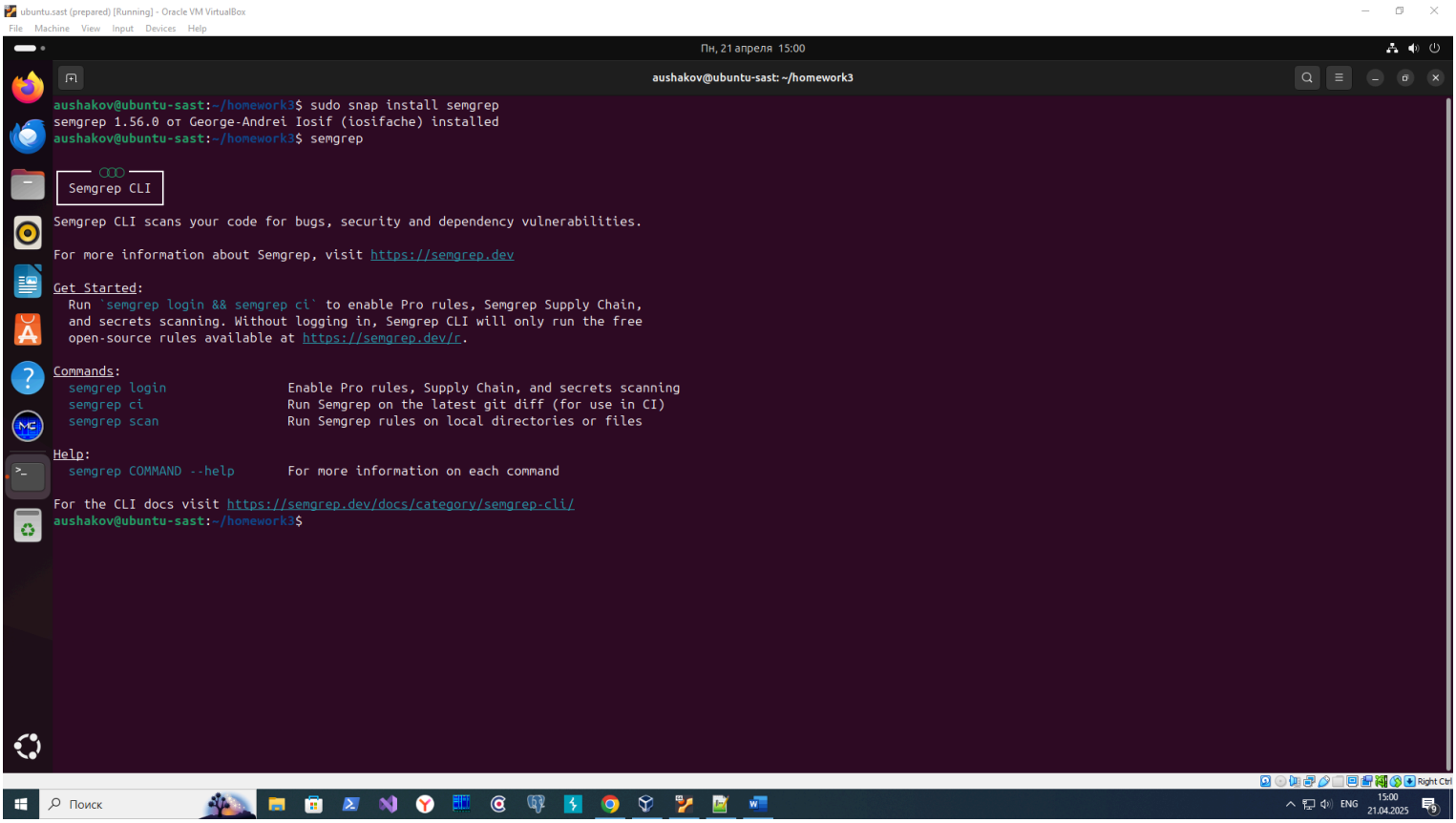
Устанавливаю npm



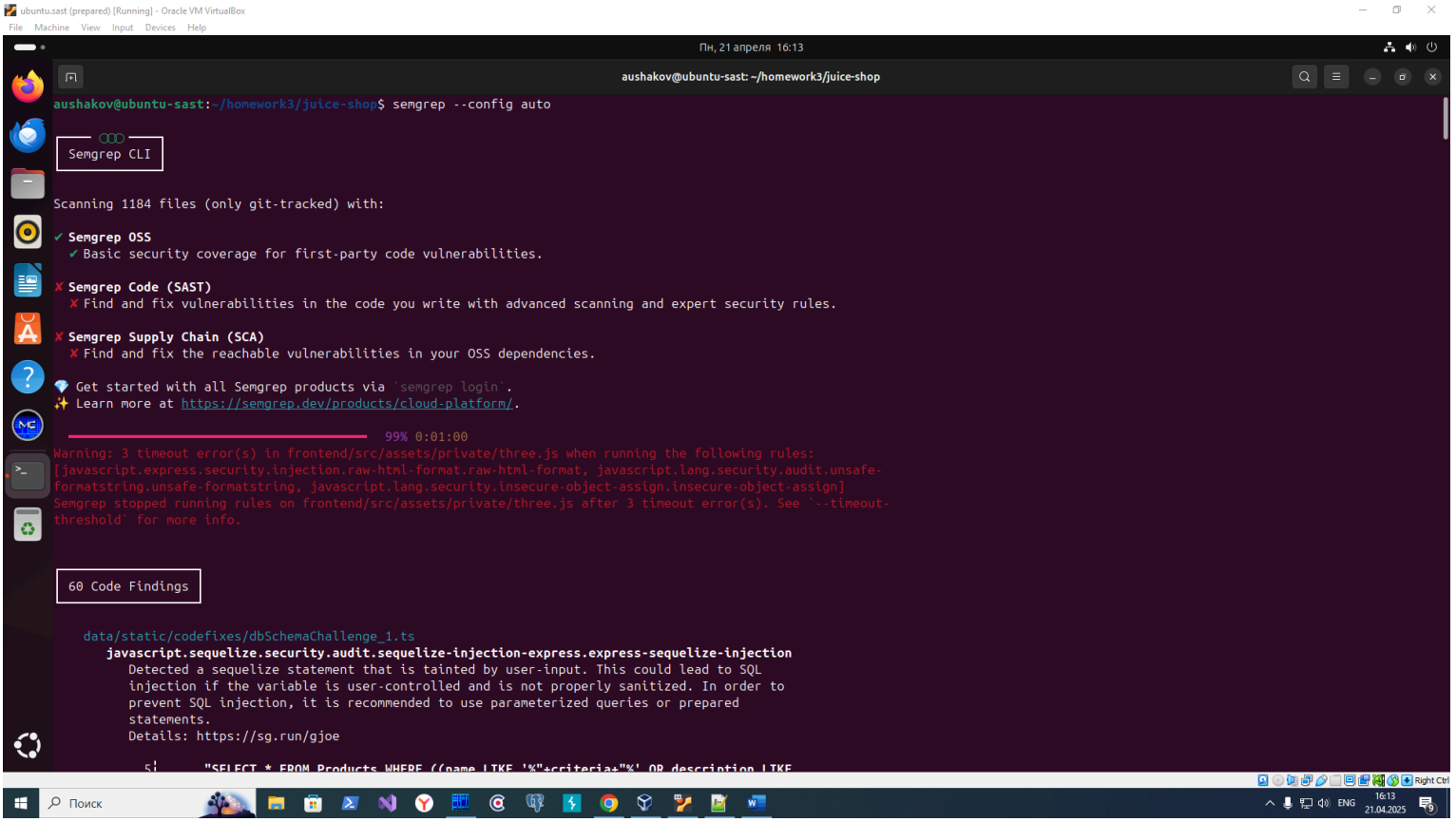


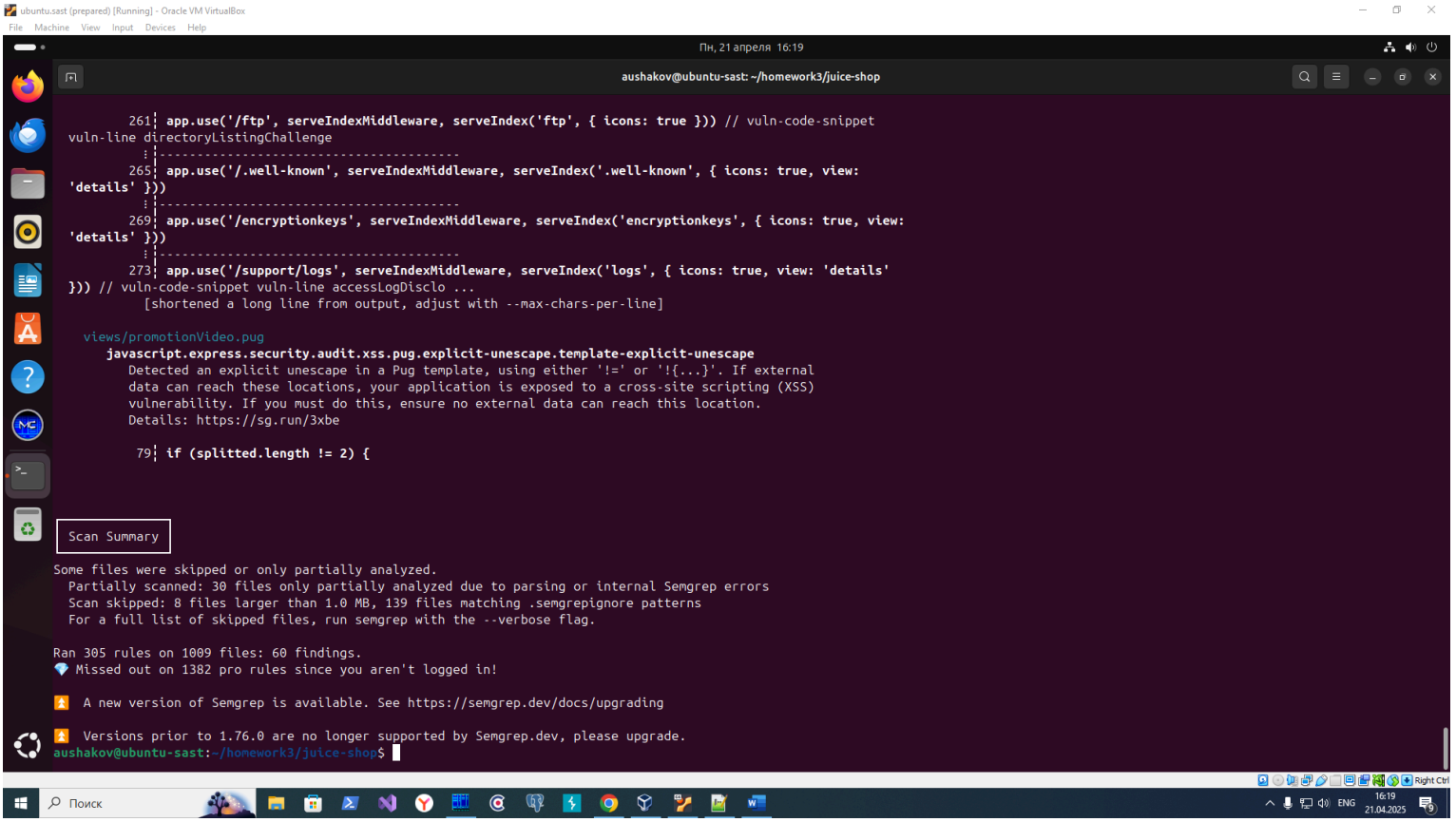
Устанавливаю Semgrep (через snap)



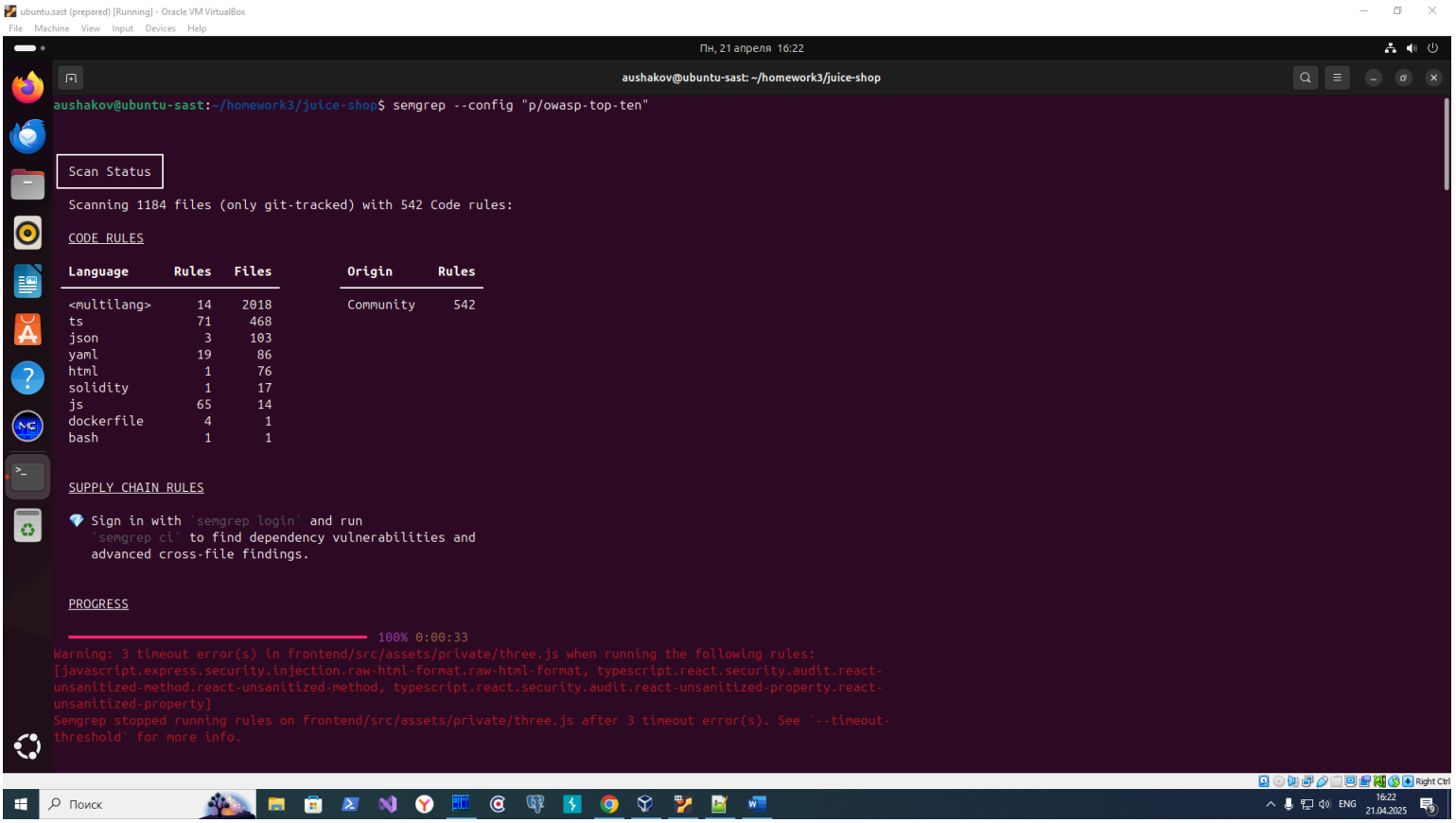


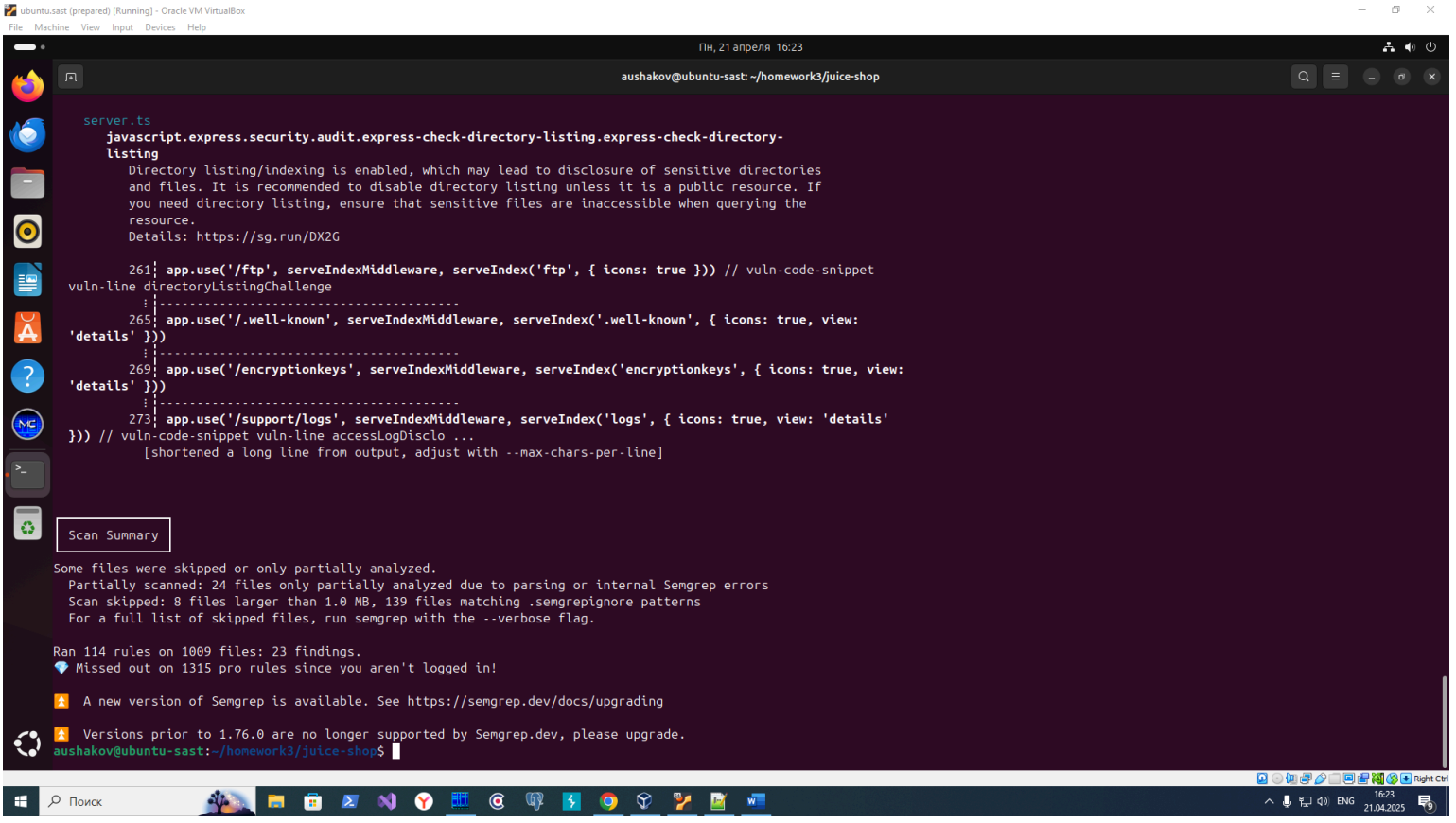
Запускаю Semgrep с конфигурацией по умолчанию: semgrep --config auto



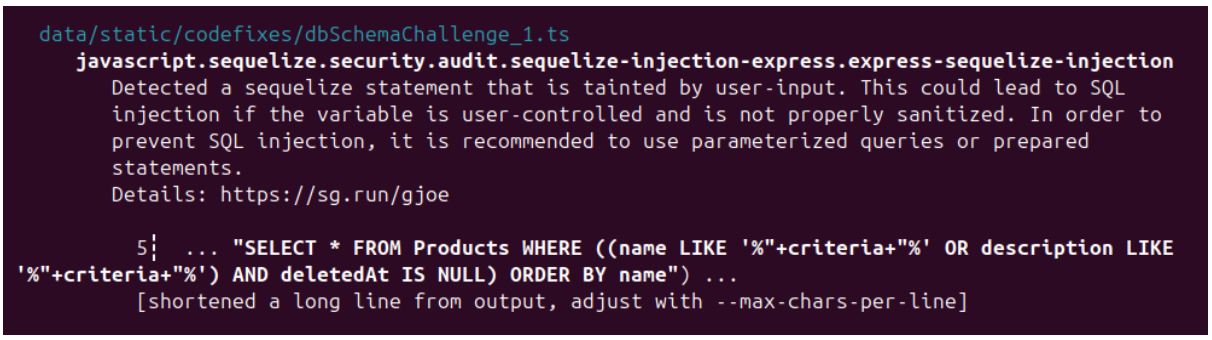


Запускаю Semgrep с конфигурацией проверки по OWASP Top 10: semgrep --config "p/owasp-top-ten"

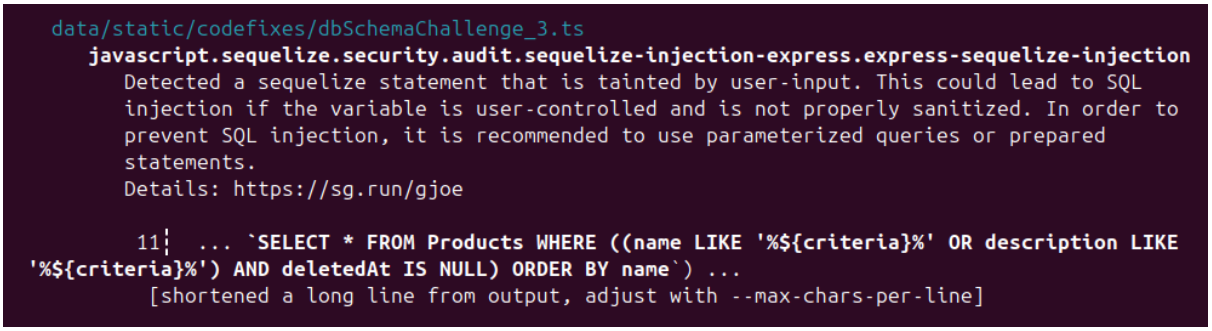




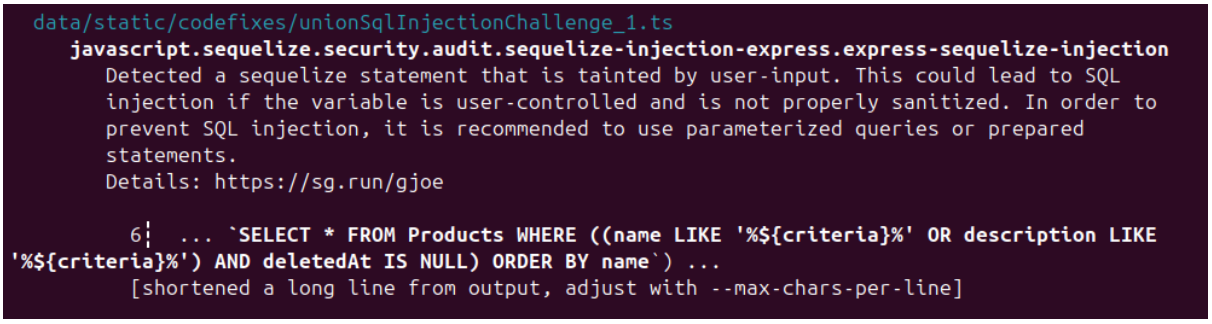
Давайте рассмотрим каждую из найденных уязвимостей:
1



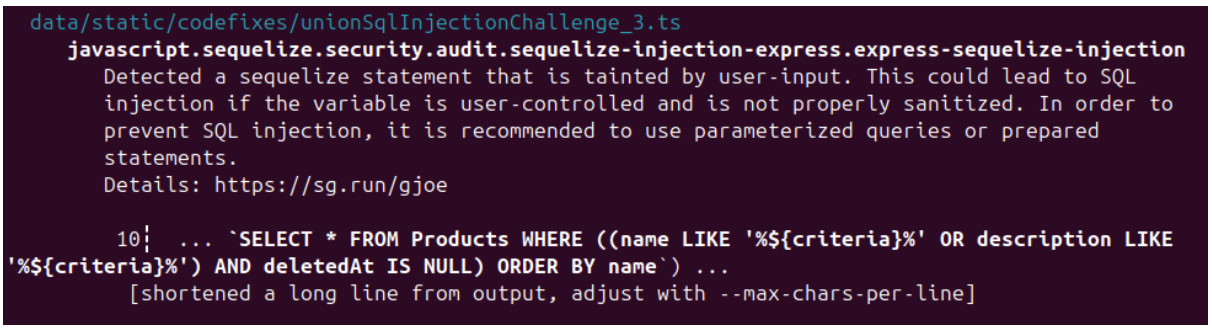
Файл: data/static/codefixes/dbSchemaChallenge_1.ts
Строка: 5
Тип уязвимости: SQL injection (sequelize-injection-express.express-sequelize-injection)
Описание: использование пользовательского ввода напрямую при формировании SQL запроса
Рекомендация по устранению: использование параметризованных запросов или prepared statements
2



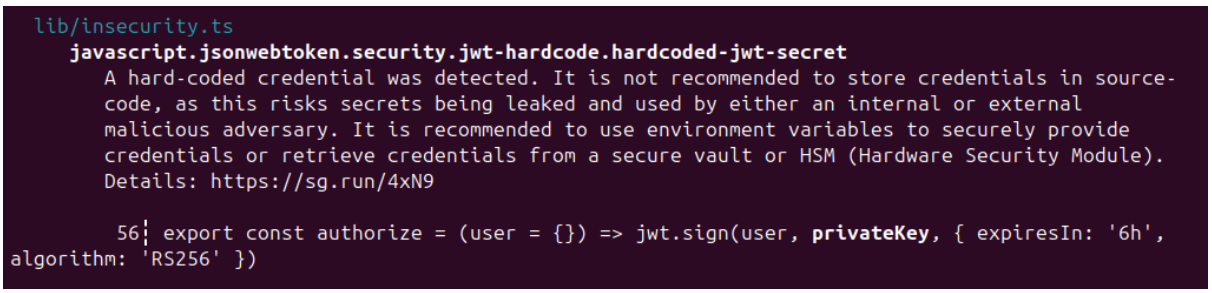
Файл: data/static/codefixes/dbSchemaChallenge_3.ts
Строка: 11
Тип уязвимости: SQL injection (sequelize-injection-express.express-sequelize-injection)
Описание: использование пользовательского ввода напрямую при формировании SQL запроса
Рекомендация по устранению: использование параметризованных запросов или prepared statements
3



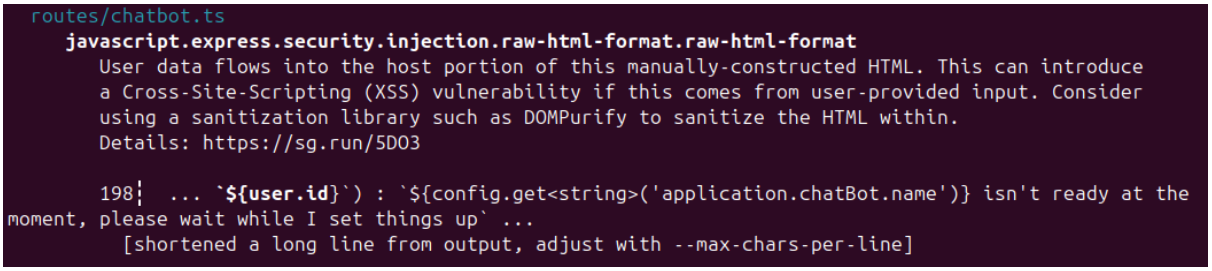
Файл: data/static/codefixes/unionSqlInjectionChallenge_1.ts
Строка: 6
Тип уязвимости: SQL injection (sequelize-injection-express.express-sequelize-injection)
Описание: использование пользовательского ввода напрямую при формировании SQL запроса
Рекомендация по устранению: использование параметризованных запросов или prepared statements
4



Файл: data/static/codefixes/unionSqlInjectionChallenge_3.ts
Строка: 10
Тип уязвимости: SQL injection (sequelize-injection-express.express-sequelize-injection)
Описание: использование пользовательского ввода напрямую при формировании SQL запроса
Рекомендация по устранению: использование параметризованных запросов или prepared statements
5

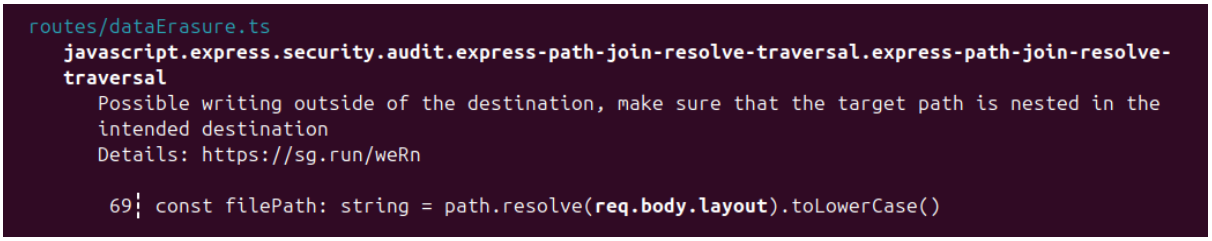


Файл: lib/insecurity.ts
Строка: 56
Тип уязвимости: использование учетных данных, прописанных в коде (jwt-hardcode.hardcoded-jwt-secret)
Описание: исходный код содержит учетные данные
Рекомендация по устранению: использование переменных среды или безопасного хранилища или HSM (Hardware Security Module) для безопасного получения учетных данных
6



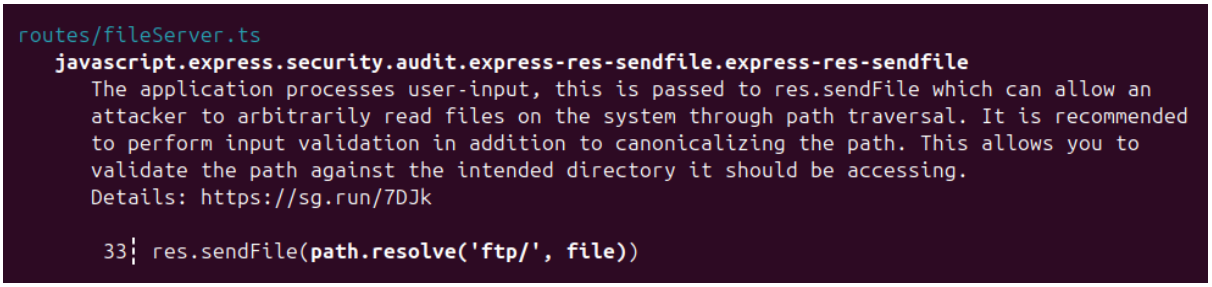
Файл: routes/chatbot.ts
Строка: 198
Тип уязвимости: XSS injection (injection.raw-html-format.raw-html-format)
Описание: использование пользовательского ввода напрямую при формировании HTML
Рекомендация по устранению: использование библиотек для обработки (sanitization) пользовательского ввода, например DOMPurify, перед использованием для формирования HTML

7

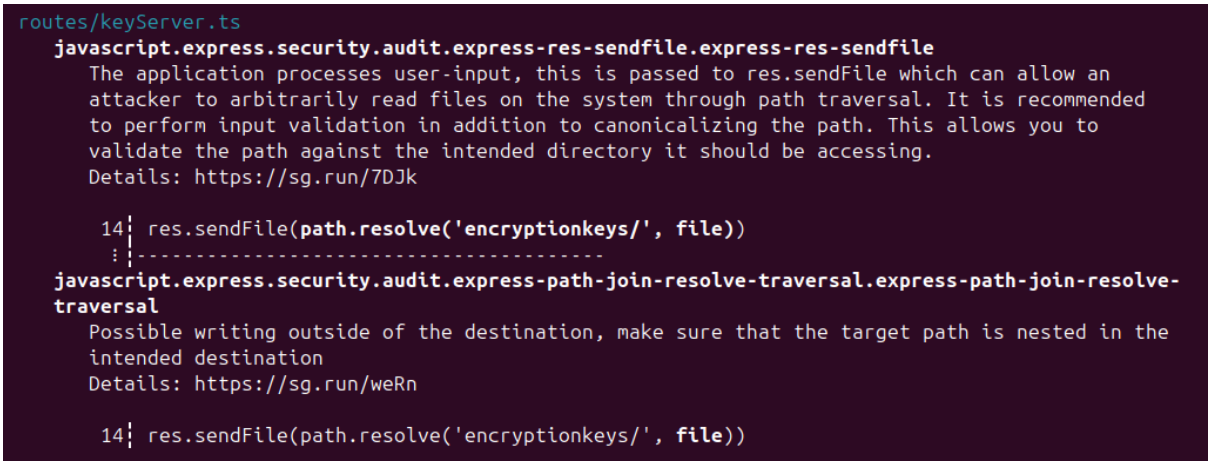


Файл: routes/dataErasure.ts
Строка: 69
Тип уязвимости: выход пути за границы (express-path-join-resolve-traversal.express-path-join-resolve-traversal)
Описание: путь потенциально может выйти за границы (за директорию), доступные приложению
Рекомендация по устранению: убедиться (например, с помощью проверок), что результирующий путь не выходит за границы (за директорию), доступные приложению

8



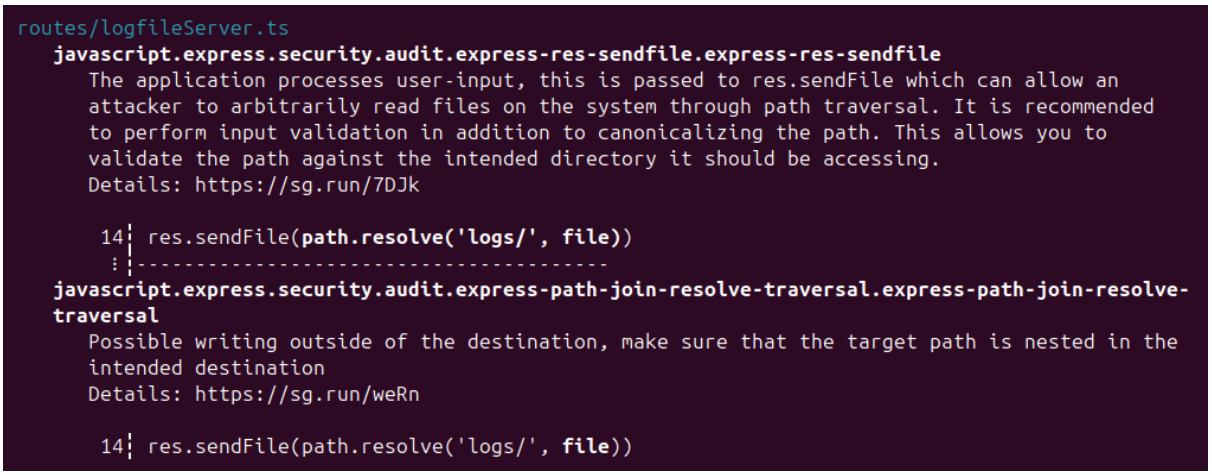
Файл: routes/fileServer.ts
Строка: 33
Тип уязвимости: использование ввода пользователя при формировании пути (express-res-sendfile.express-res-sendfile)
Описание: путь потенциально может выйти за границы (за директорию), доступные приложению, из-за использования ввода пользователя при его формировании
Рекомендация по устранению: использование проверки пользовательского ввода и приведения пути к каноническому виду с последующей валидацией сформированного пути
9



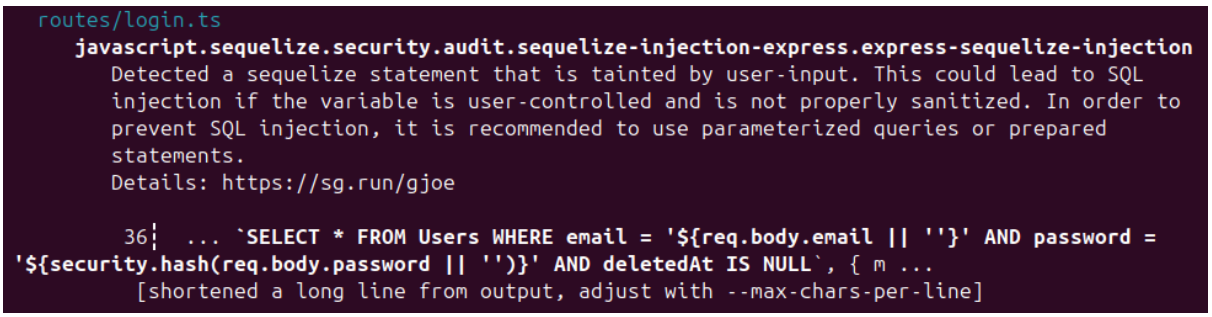
Файл: routes/keyServer.ts
Строка: 14
Тип уязвимости: использование ввода пользователя при формировании пути (express-res-sendfile.express-res-sendfile)

Описание: путь потенциально может выйти за границы (за директорию), доступные приложению, из-за использования ввода пользователя при его формировании
Рекомендация по устранению: использование проверки пользовательского ввода и приведения пути к каноническому виду с последующей валидацией сформированного пути
Строка: 14
Тип уязвимости: выход пути за границы (express-path-join-resolve-traversal.express-path-join-resolve-traversal)
Описание: путь потенциально может выйти за границы (за директорию), доступные приложению
Рекомендация по устранению: убедиться (например, с помощью проверок), что результирующий путь не выходит за границы (за директорию), доступные приложению

10



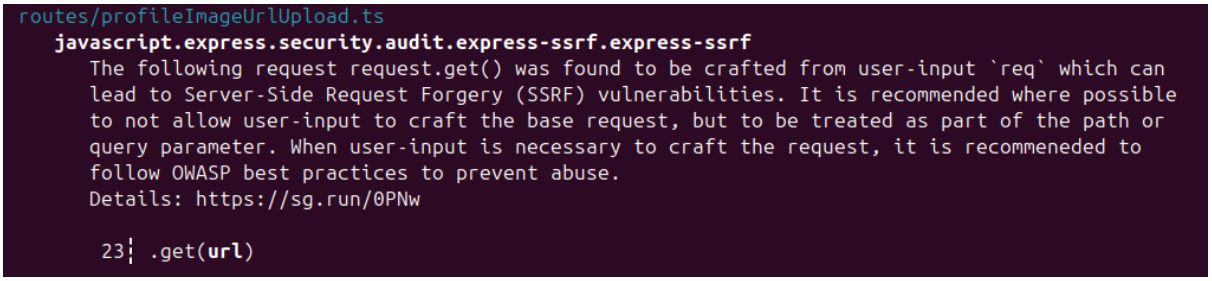
Файл: routes/logfileServer.ts
Строка: 14
Тип уязвимости: использование ввода пользователя при формировании пути (express-res-sendfile.express-res-sendfile)
Описание: путь потенциально может выйти за границы (за директорию), доступные приложению, из-за использования ввода пользователя при его формировании
Рекомендация по устранению: использование проверки пользовательского ввода и приведения пути к каноническому виду с последующей валидацией сформированного пути
Строка: 14
Тип уязвимости: выход пути за границы (express-path-join-resolve-traversal.express-path-join-resolve-traversal)
Описание: путь потенциально может выйти за границы (за директорию), доступные приложению
Рекомендация по устранению: убедиться (например, с помощью проверок), что результирующий путь не выходит за границы (за директорию), доступные приложению
11



Файл: routes/login.ts
Строка: 36

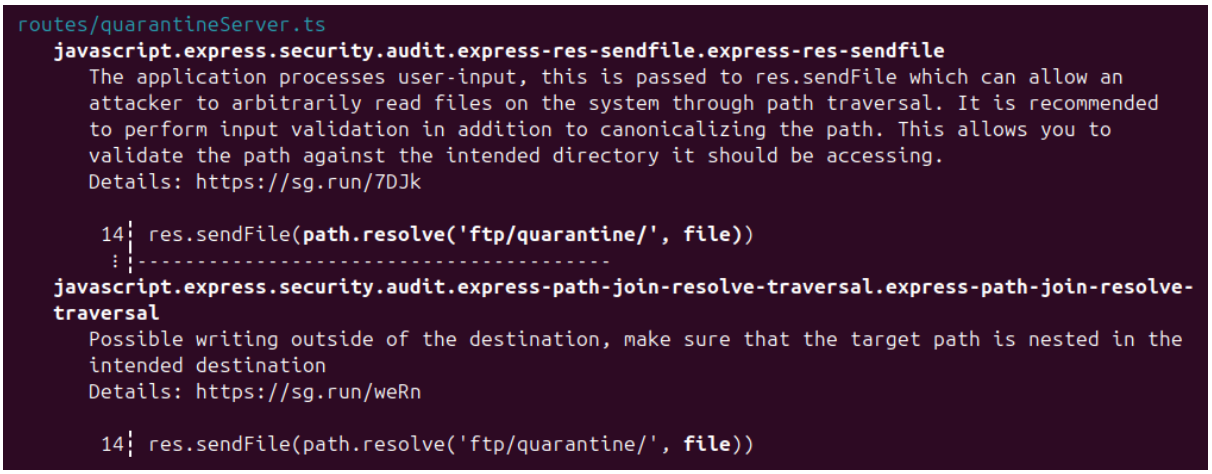
Тип уязвимости: SQL injection (sequelize-injection-express.express-sequelize-injection)
Описание: использование пользовательского ввода напрямую при формировании SQL запроса
Рекомендация по устранению: использование параметризованных запросов или prepared statements

12



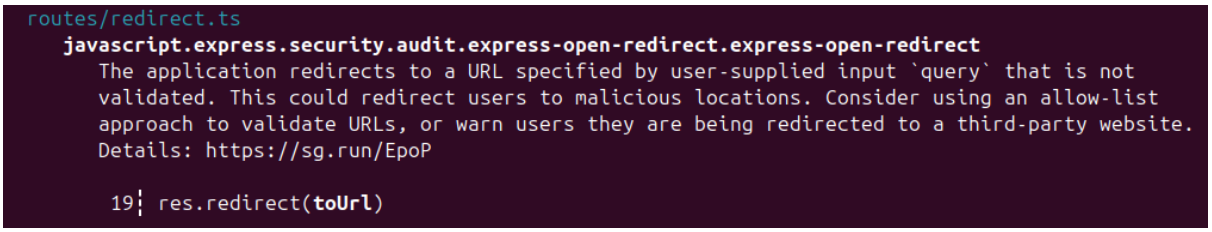
Файл: routes/profileImageUrlUpload.ts
Строка: 23
Тип уязвимости: Server-Side Request Forgery (SSRF) (express-ssrf.express-ssrf)
Описание: использование пользовательского ввода для формирования HTTP запроса, что может привести к Server-Side Request Forgery (SSRF)
Рекомендация по устранению: не использовать пользовательский ввод для формирования HTTP запроса целиком, а использовать его для формирования части относительного пути или параметров запроса

13

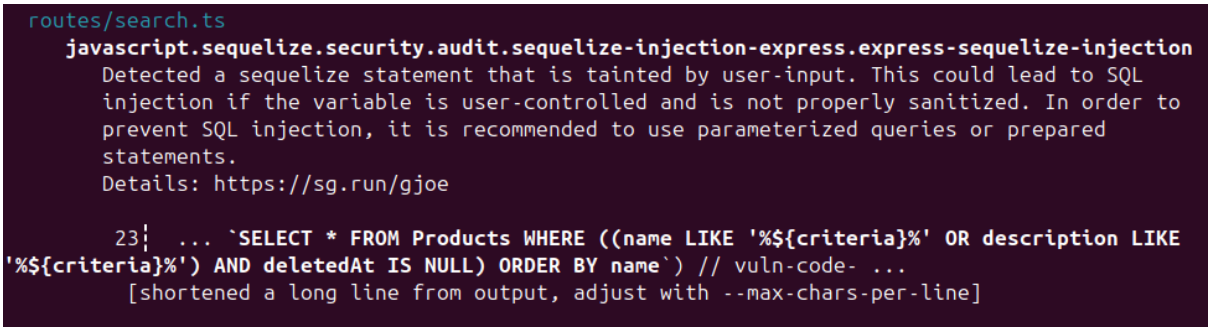


Файл: routes/quarantineServer.ts
Строка: 14
Тип уязвимости: использование ввода пользователя при формировании пути (express-res-sendfile.express-res-sendfile)
Описание: путь потенциально может выйти за границы (за директорию), доступные приложению, из-за использования ввода пользователя при его формировании
Рекомендация по устранению: использование проверки пользовательского ввода и приведения пути к каноническому виду с последующей валидацией сформированного пути
Строка: 14
Тип уязвимости: выход пути за границы (express-path-join-resolve-traversal.express-path-join-resolve-traversal)
Описание: путь потенциально может выйти за границы (за директорию), доступные приложению
Рекомендация по устранению: убедиться (например, с помощью проверок), что результирующий путь не выходит за границы (за директорию), доступные приложению

14

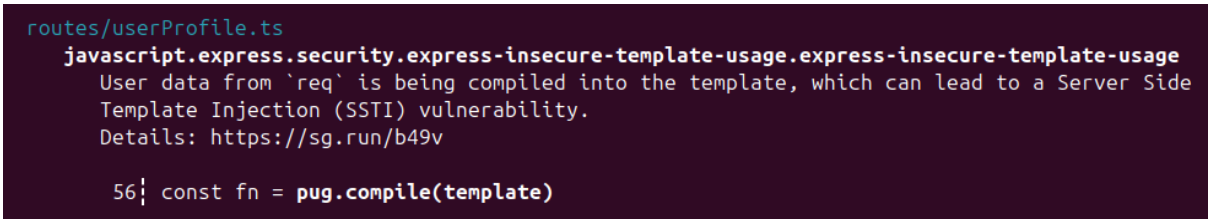


Файл: routes/redirect.ts
Строка: 19
Тип уязвимости: перенаправление на не валидируемый URL, задаваемый пользователем (express-open-redirect.express-open-redirect)
Описание: перенаправление на не валидируемый URL, задаваемый пользователем, что может привести к переходу по вредоносному URL
Рекомендация по устранению: использование белого списка URL или предупреждение пользователей при попытке перехода по URL третьей стороны
15

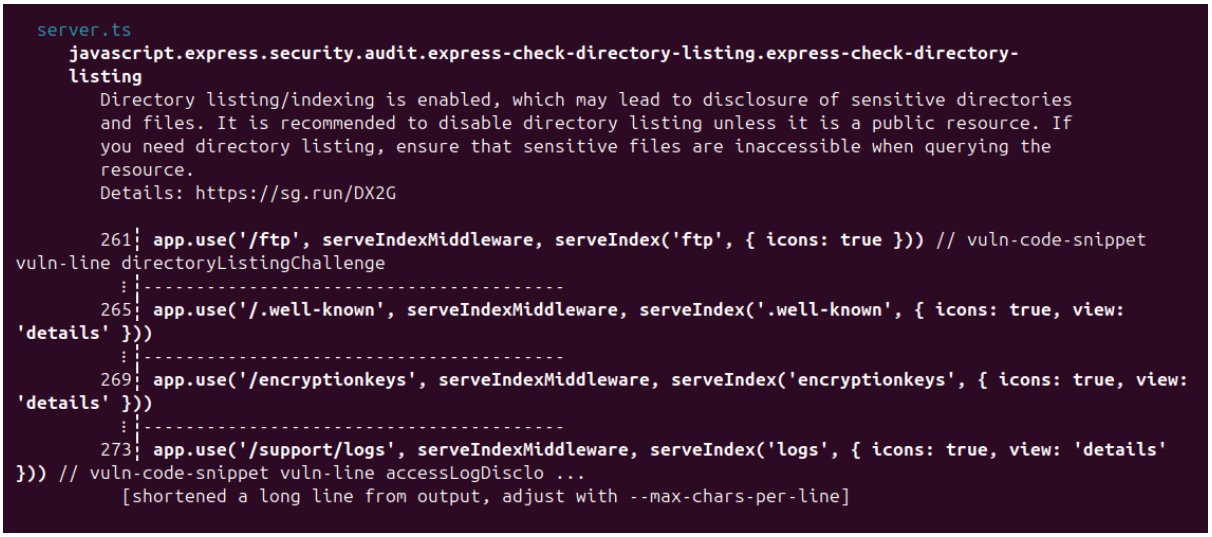


Файл: routes/search.ts
Строка: 23
Тип уязвимости: SQL injection (sequelize-injection-express.express-sequelize-injection)
Описание: использование пользовательского ввода напрямую при формировании SQL запроса
Рекомендация по устранению: использование параметризованных запросов или prepared statements

16



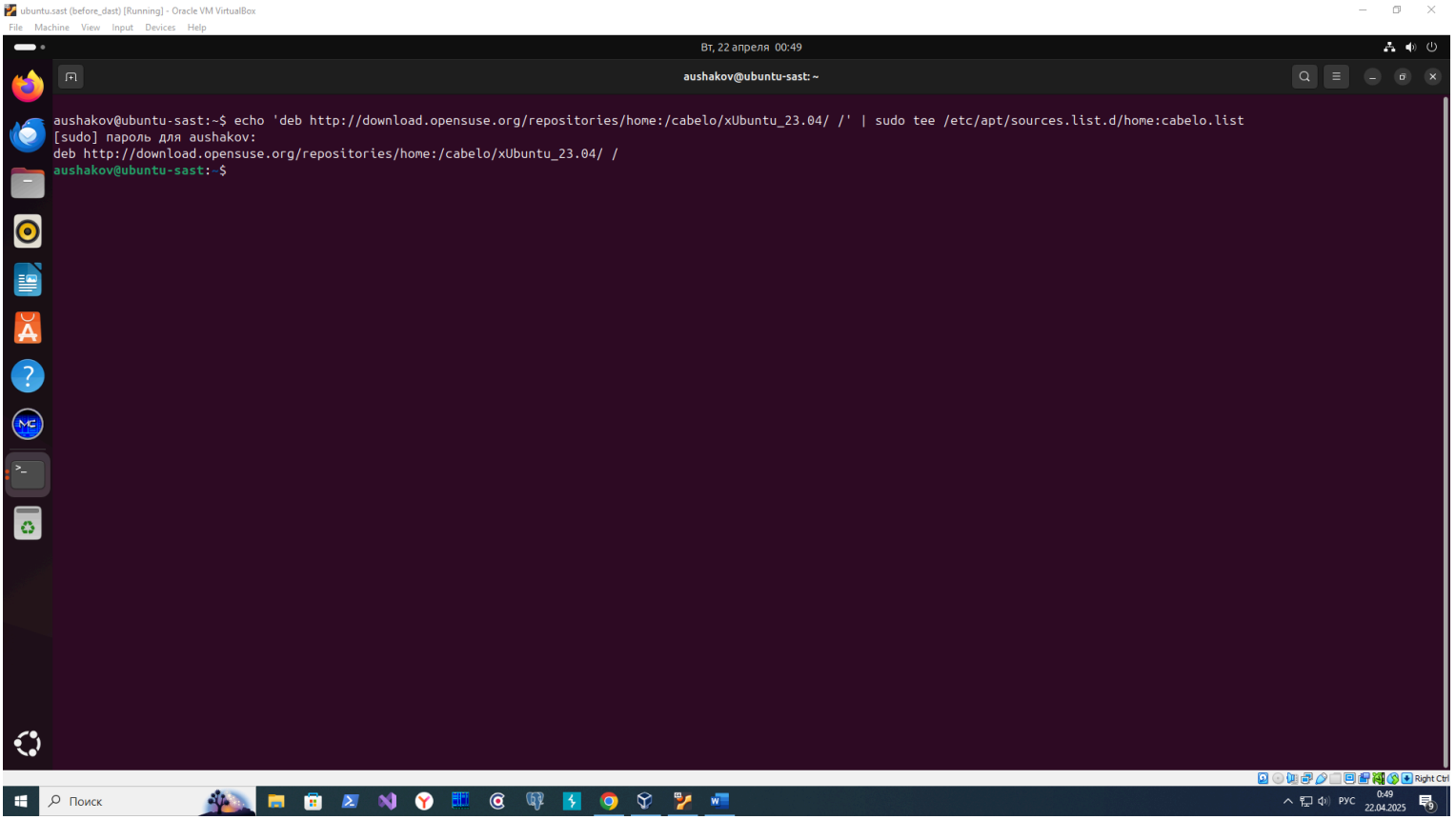
Файл: routes/userProfile.ts
Строка: 56
Тип уязвимости: Server Side Template Injection (SSTI) (express-insecure-template-usage.express-insecure-template-usage)
Описание: пользовательские данные компилируются в шаблон, что может привести к Server Side Template Injection (SSTI)
Рекомендация по устранению: валидация и экранирование пользовательских данных
17

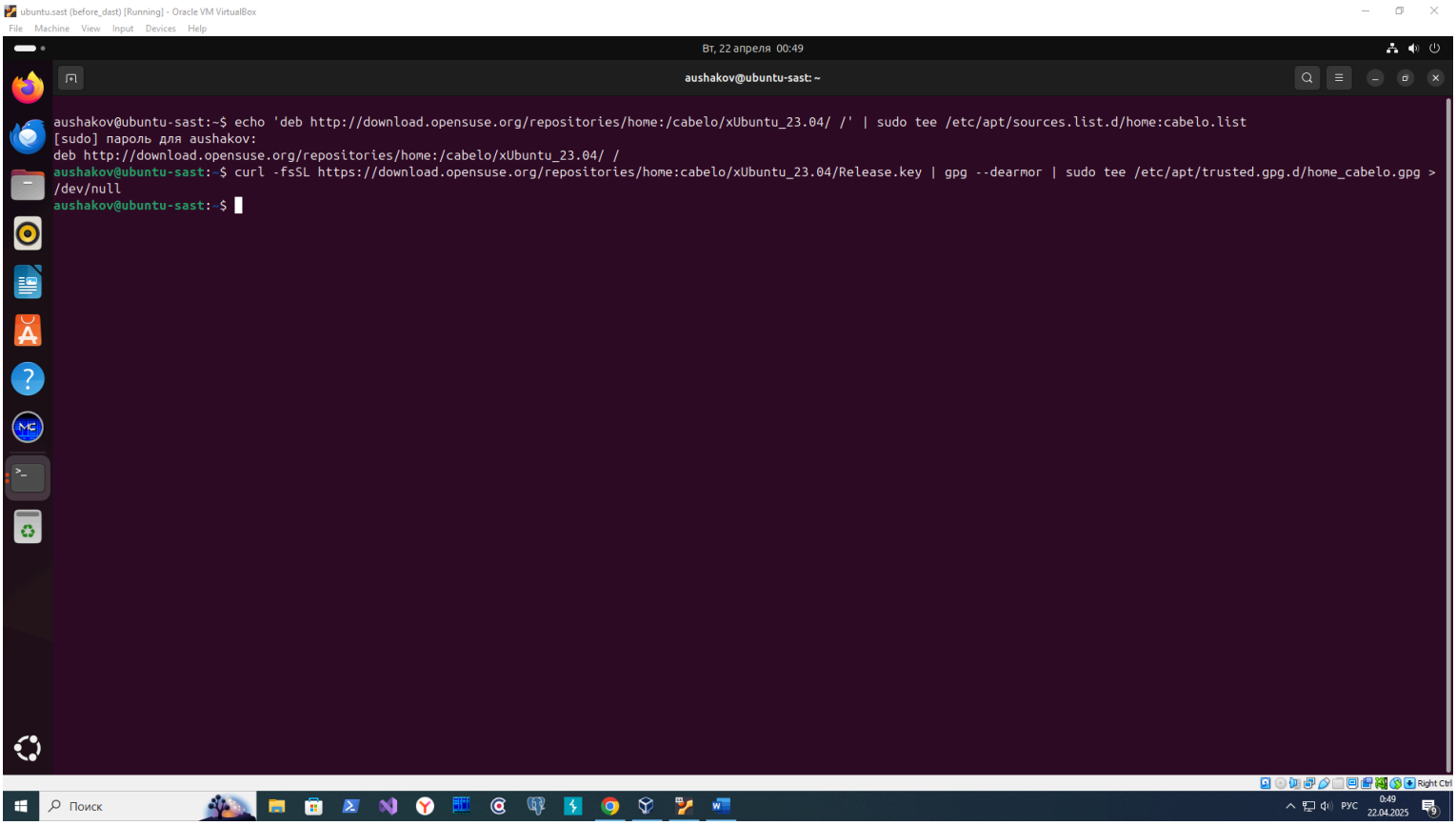


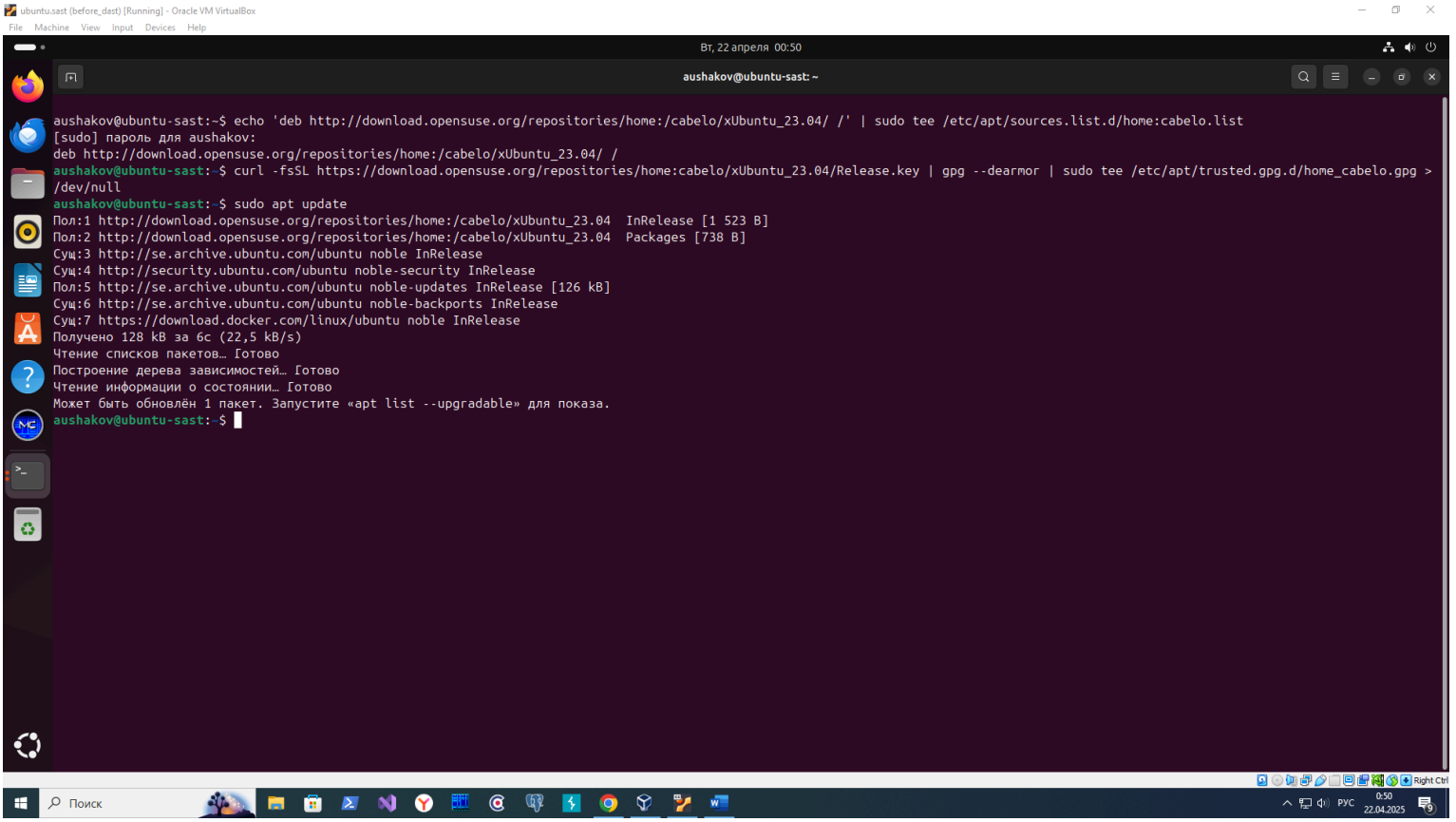
Файл: server.ts
Строки: 261, 265, 269, 273
Тип уязвимости: включено перечисление и индексирование каталогов (express-check-directory-listing.express-check-directory-listing)
Описание: включено перечисление и индексирование каталогов, что может привести к раскрытию конфиденциальных каталогов и файлов
Рекомендация по устранению: отключить перечисление и индексирование каталогов для непубличных ресурсов

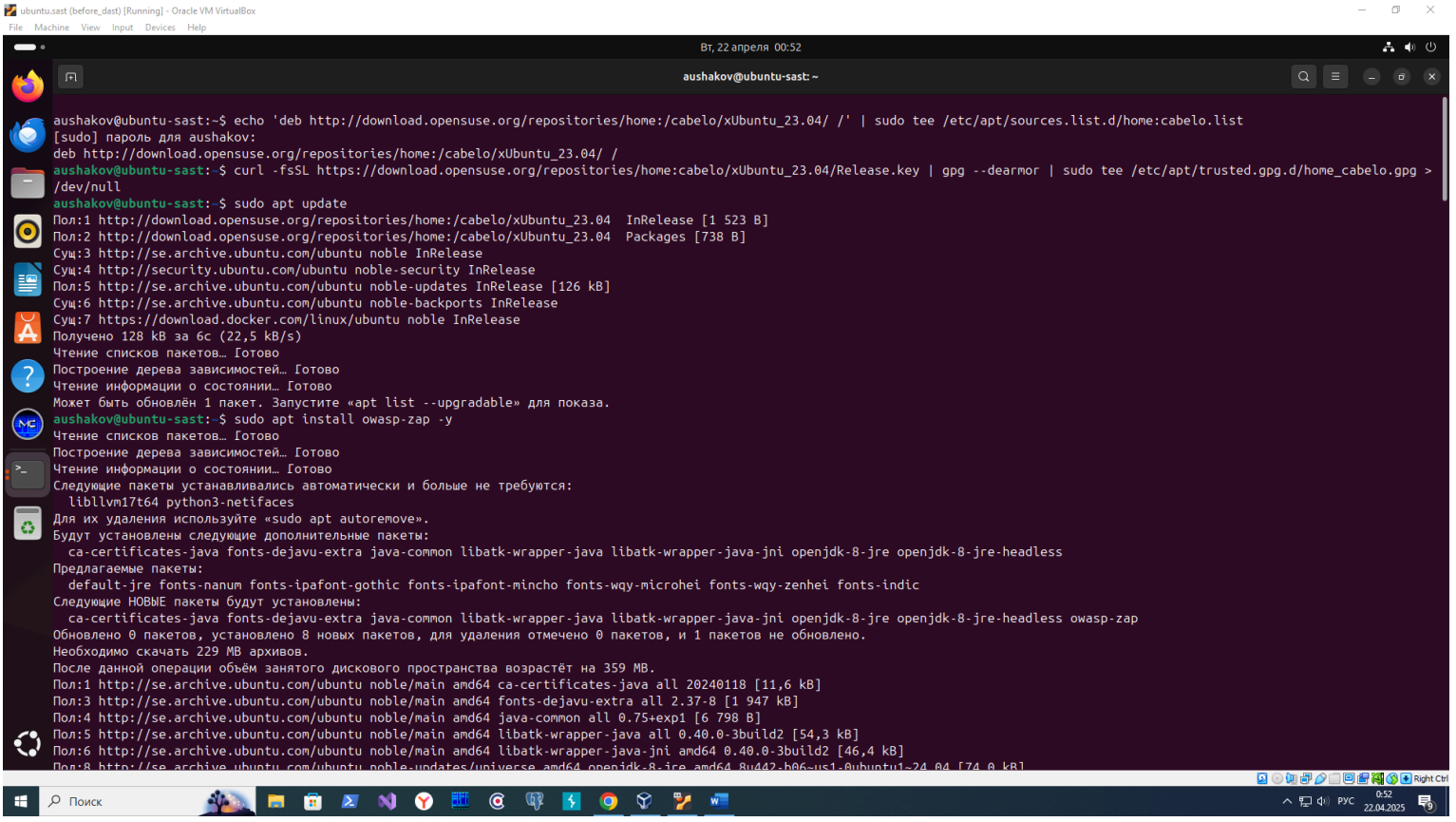
Выводы:
1. Практически все найденные уязвимости основаны на непроверяемых пользовательских данных; их все можно считать High/Critical. Особенно, стоит обратить внимание на наличие нескольких SQL инъекций (которые 100% можно считать Critical уязвимостями).
2. Некоторые из уязвимостей возможно являются false positive случаями, однако из результатов анализа Semgrep этого понять не возможно, т.к. нужно исследовать исходный код.

Устанавливаю OWASP ZAP

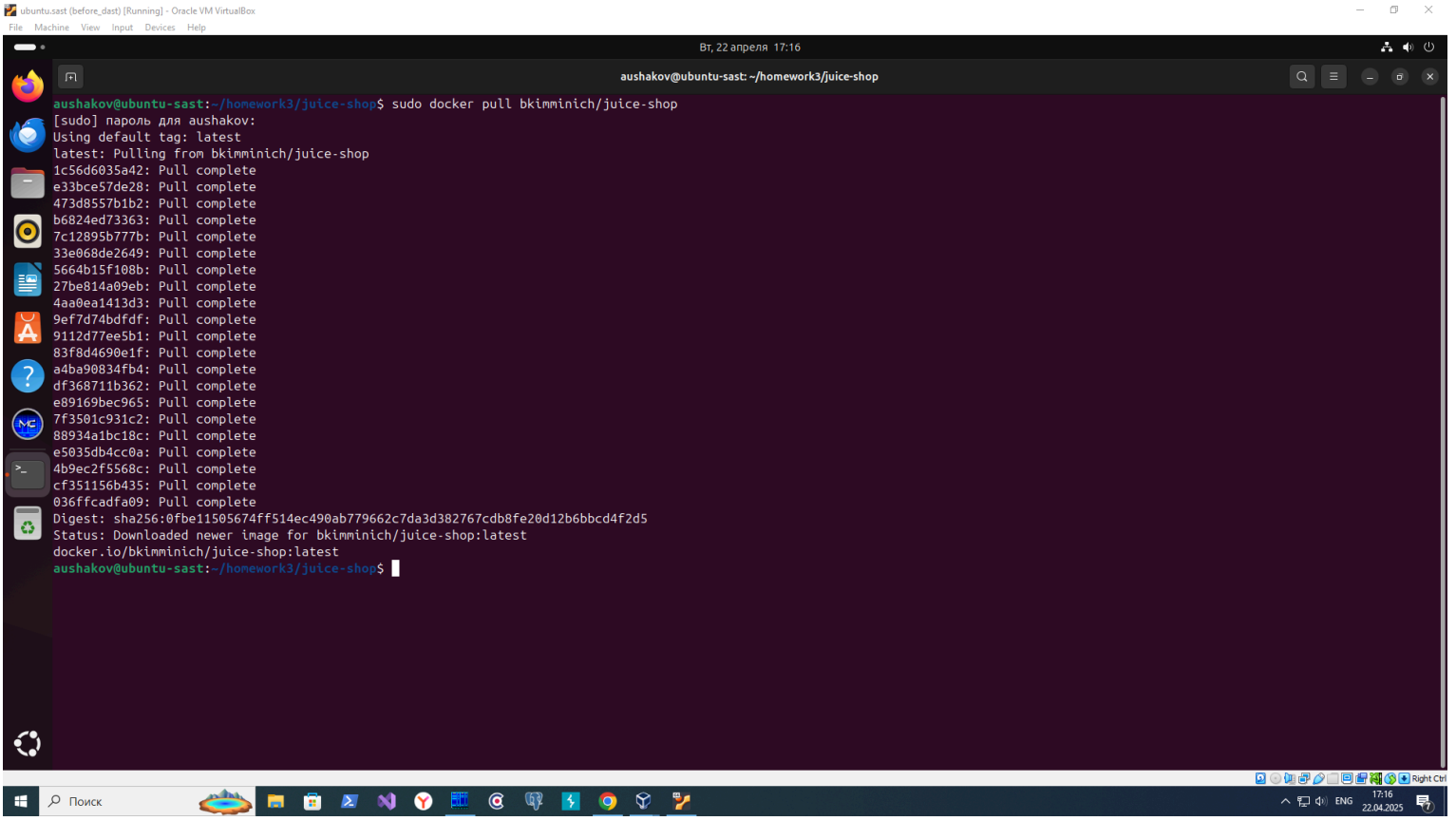

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Вт, 22 апреля 00:49
aushakov@ubuntu-sast: ~
aushakov@ubuntu-sast:~$ echo 'deb http://download.opensuse.org/repositories/home:/cabelo/xUbuntu_23.04/ /' | sudo tee /etc/apt/sources.list.d/home:cabelo.list
[sudo] пароль для aushakov:
deb http://download.opensuse.org/repositories/home:/cabelo/xUbuntu_23.04/ /
aushakov@ubuntu-sast:~$
Поиск
РУС
0:49
22.04.2025

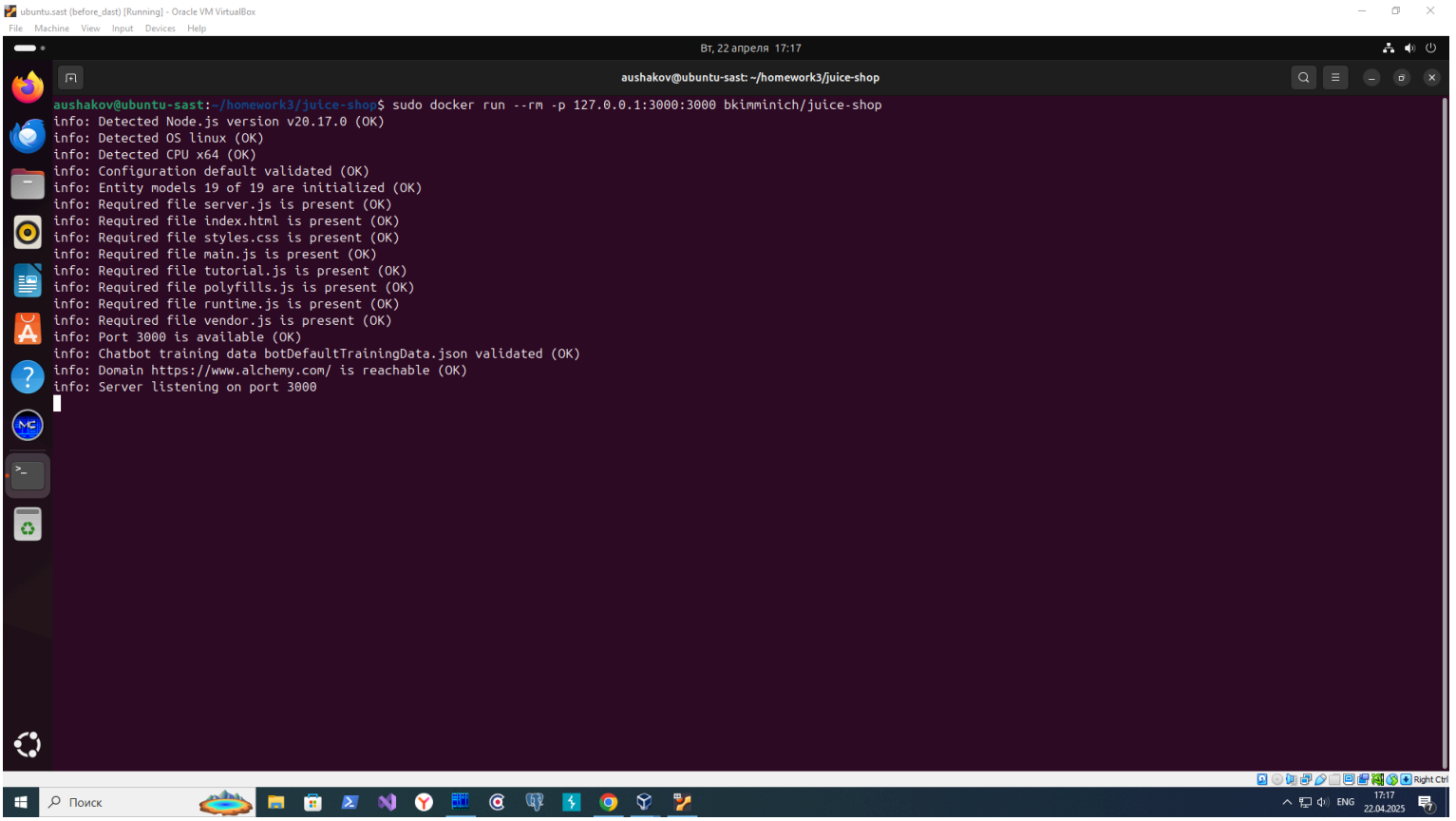

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Вт, 22 апреля 00:49
aushakov@ubuntu-sast: ~
aushakov@ubuntu-sast:~$ echo 'deb http://download.opensuse.org/repositories/home:/cabelo/xUbuntu_23.04/ /' | sudo tee /etc/apt/sources.list.d/home:cabelo.list
[sudo] пароль для aushakov:
deb http://download.opensuse.org/repositories/home:/cabelo/xUbuntu_23.04/ /
aushakov@ubuntu-sast:~$ curl -fsSL https://download.opensuse.org/repositories/home:cabelo/xUbuntu_23.04/Release.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/home_cabelo.gpg >
/dev/null
aushakov@ubuntu-sast:~$
Поиск
РУС
0:49
22.04.2025

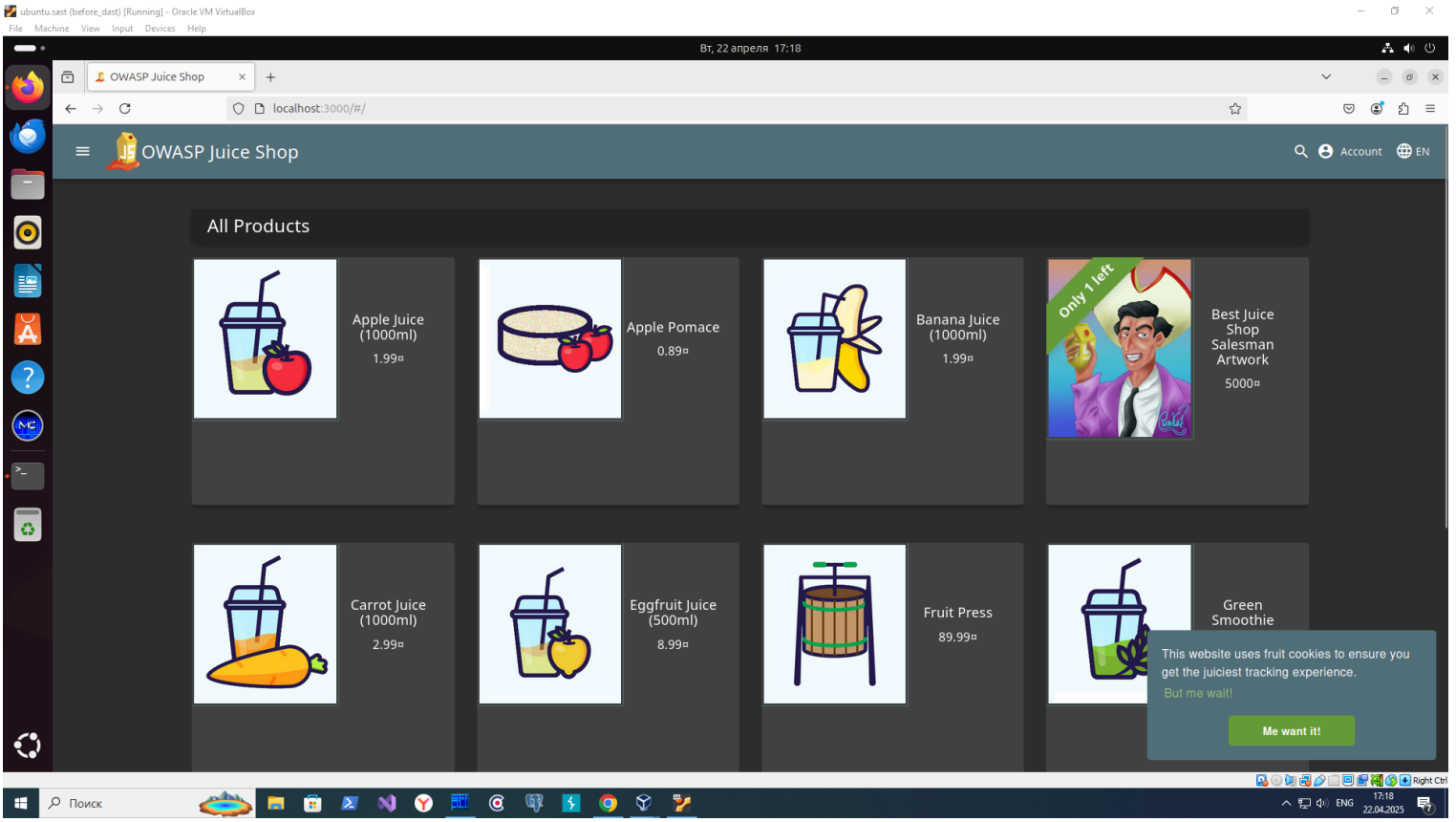




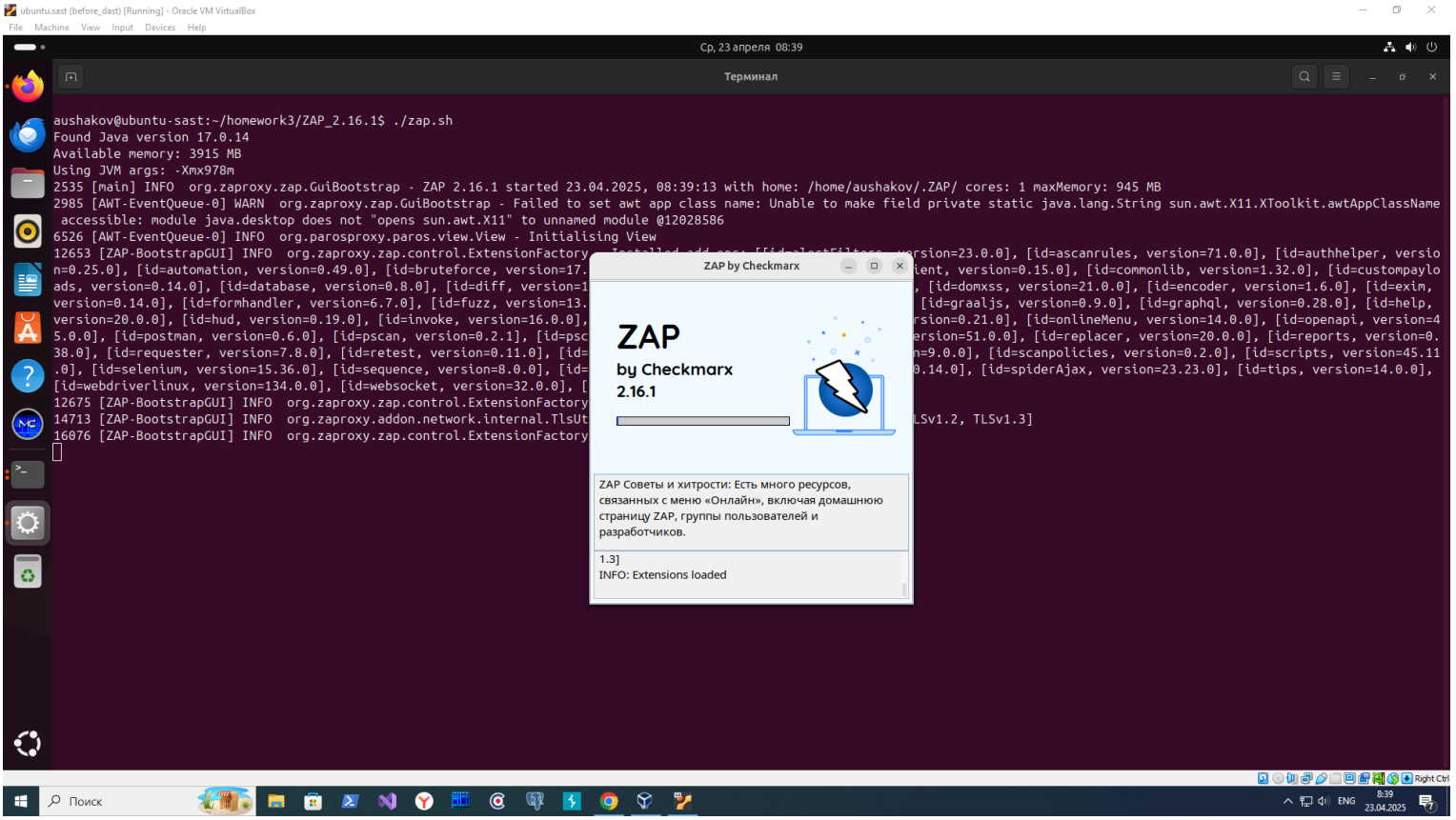
Запускаю OWASP Juice Shop (с помощью docker контейнера)

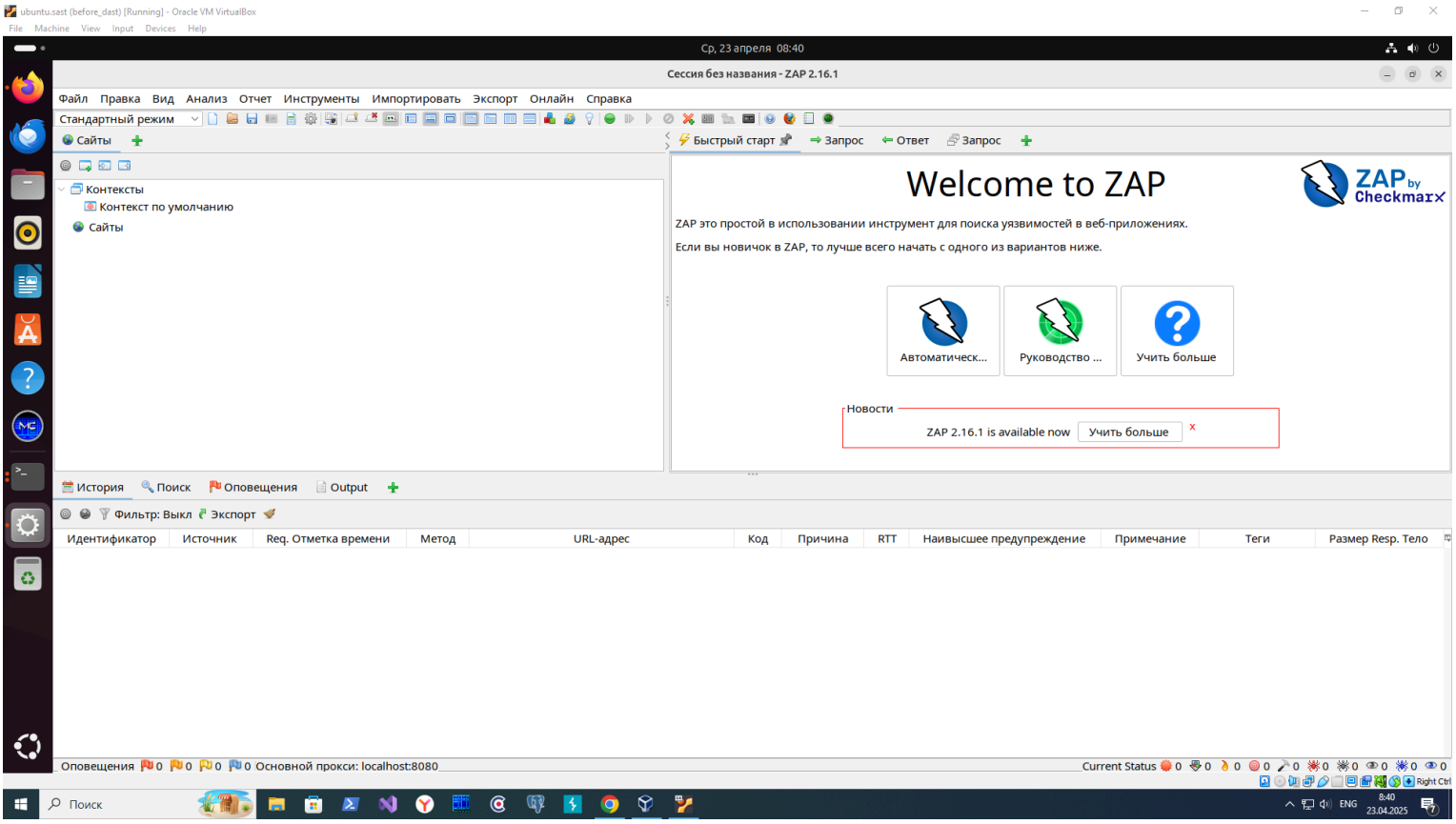

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Вт, 22 апреля 17:16
aushakov@ubuntu-sast: ~/homework3/juice-shop
aushakov@ubuntu-sast:~/homework3/juice-shop$ sudo docker pull bkimminich/juice-shop
[sudo] пароль для aushakov:
Using default tag: latest
latest: Pulling from bkimminich/juice-shop
1c56d6035a42: Pull complete
e33bce57de28: Pull complete
473d8557b1b2: Pull complete
b6824ed73363: Pull complete
7c12895b777b: Pull complete
33e068de2649: Pull complete
5664b15f108b: Pull complete
27be814a09eb: Pull complete
4aa0ea1413d3: Pull complete
9ef7d74bdfdf: Pull complete
9112d77ee5b1: Pull complete
83f8d4690e1f: Pull complete
a4ba90834fb4: Pull complete
df368711b362: Pull complete
e89169bec965: Pull complete
7f3501c931c2: Pull complete
88934a1bc18c: Pull complete
e5035db4cc0a: Pull complete
4b9ec2f5568c: Pull complete
cf351156b435: Pull complete
036ffcadfa09: Pull complete
Digest: sha256:0fbe11505674ff514ec490ab779662c7da3d382767cdb8fe20d12b6bbcd4f2d5
Status: Downloaded newer image for bkimminich/juice-shop:latest
docker.io/bkimminich/juice-shop:latest
aushakov@ubuntu-sast:~/homework3/juice-shop$
Поиск
ENG
17:16
22.04.2025


ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Вт, 22 апреля 17:17
aushakov@ubuntu-sast: ~/homework3/juice-shop
aushakov@ubuntu-sast:~/homework3/juice-shop$ sudo docker run --rm -p 127.0.0.1:3000:3000 bkimminich/juice-shop
info: Detected Node.js version v20.17.0 (OK)
info: Detected OS linux (OK)
info: Detected CPU x64 (OK)
info: Configuration default validated (OK)
info: Entity models 19 of 19 are initialized (OK)
info: Required file server.js is present (OK)
info: Required file index.html is present (OK)
info: Required file styles.css is present (OK)
info: Required file main.js is present (OK)
info: Required file tutorial.js is present (OK)
info: Required file polyfills.js is present (OK)
info: Required file runtime.js is present (OK)
info: Required file vendor.js is present (OK)
info: Port 3000 is available (OK)
info: Chatbot training data botDefaultTrainingData.json validated (OK)
info: Domain https://www.alchemy.com/ is reachable (OK)
info: Server listening on port 3000
Поиск
ENG
17:17
22.04.2025

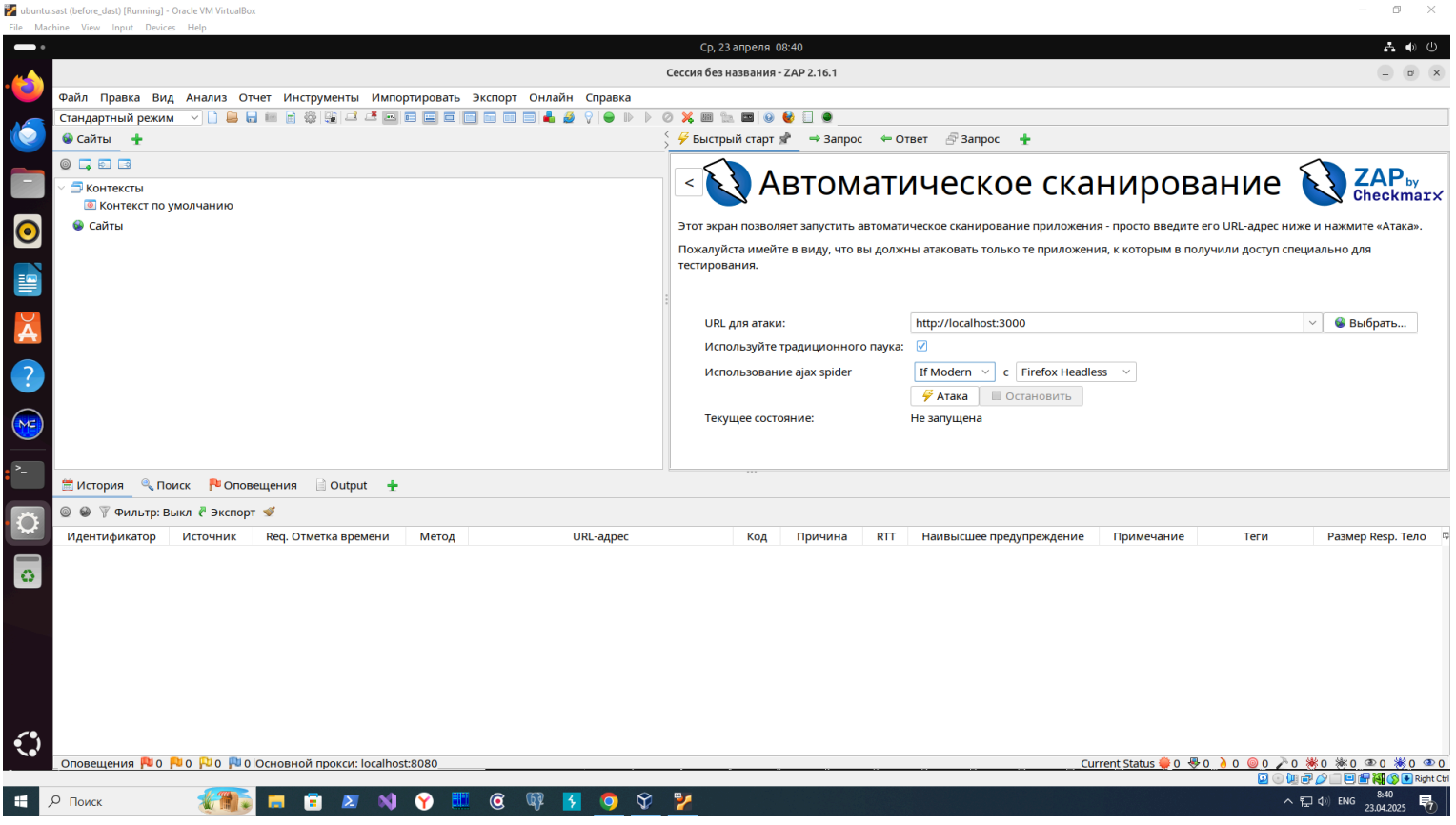


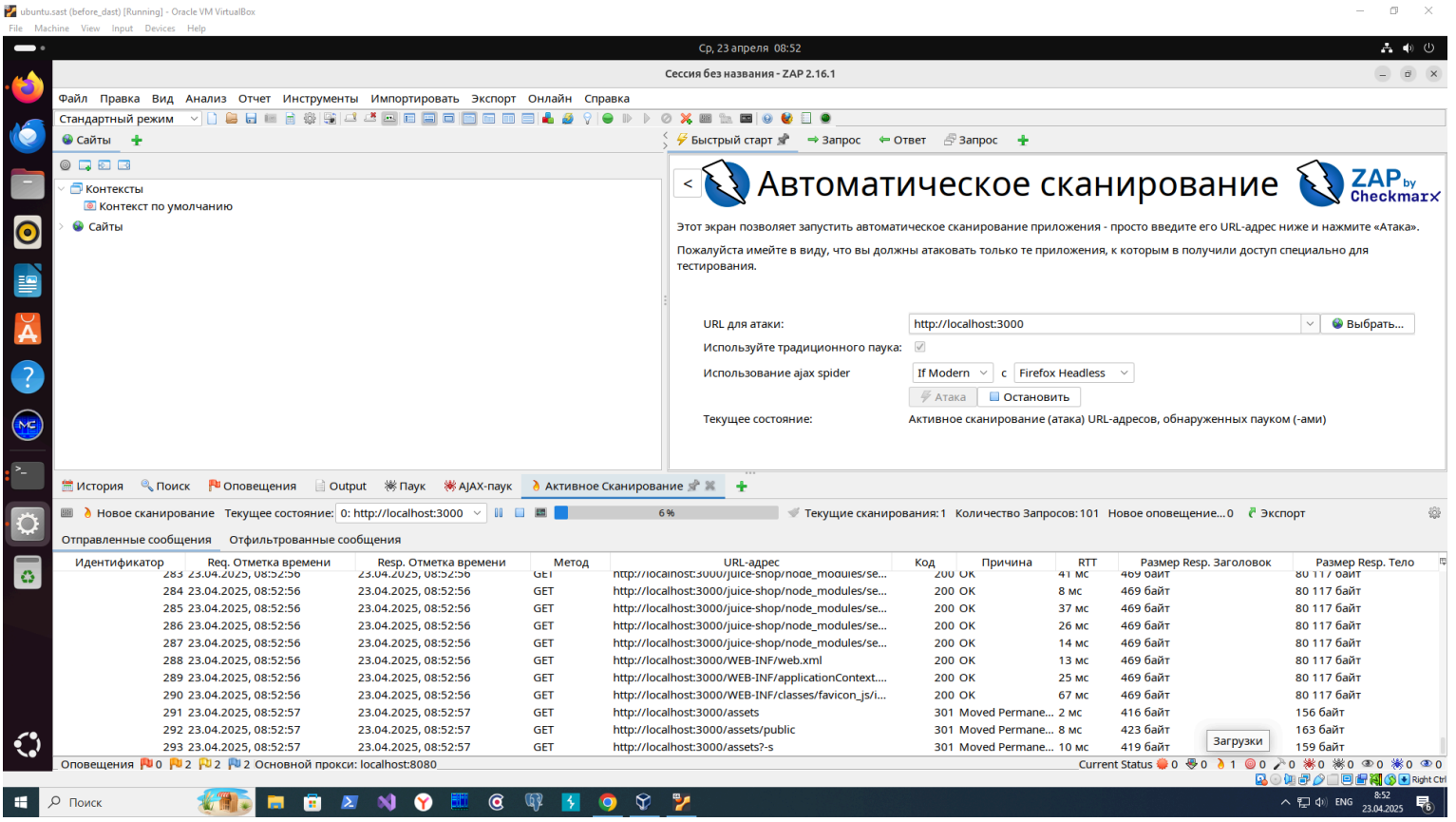
Запускаю OWASP ZAP

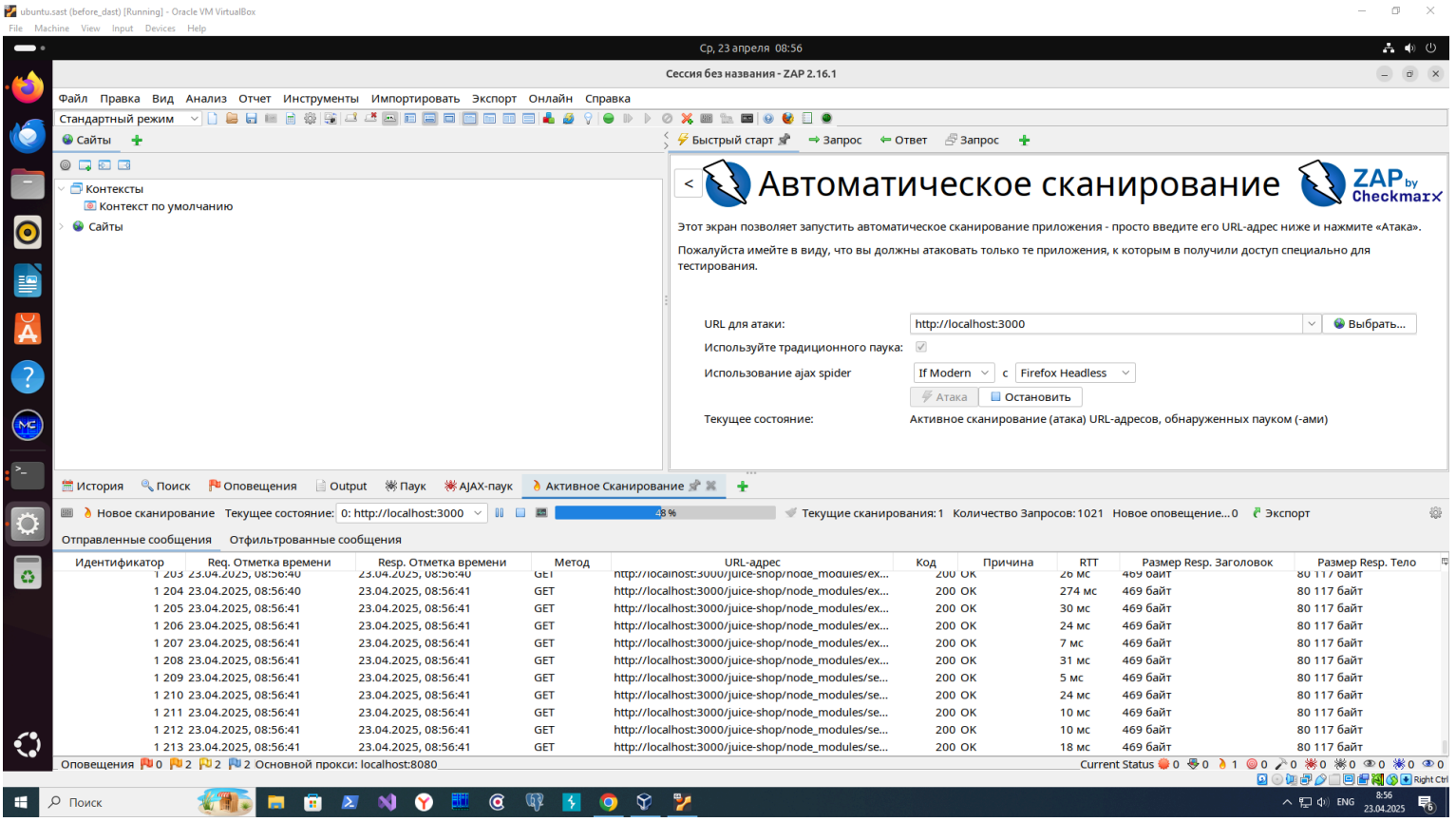


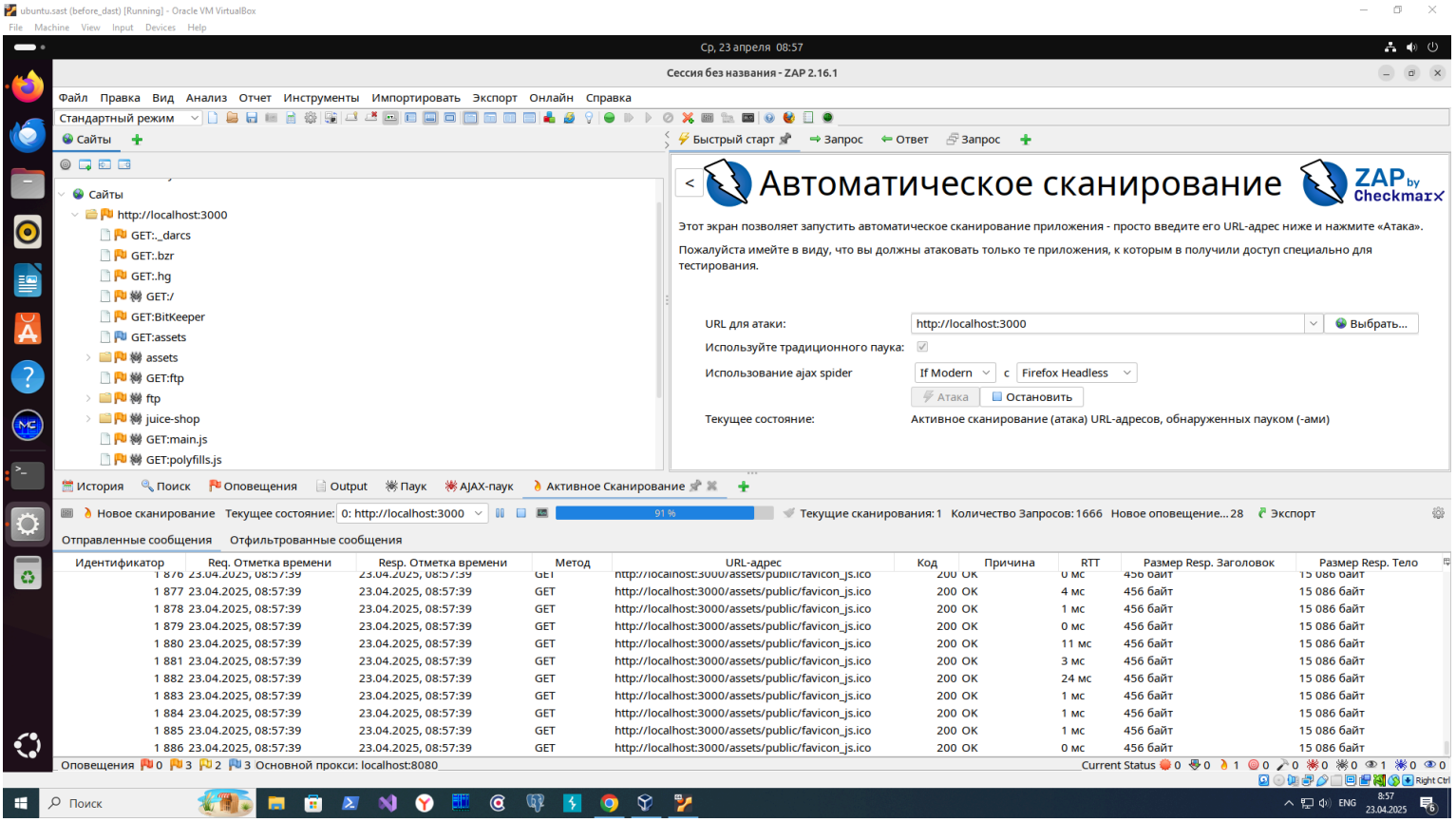


Запускаю сканирование OWASP Juice Shop

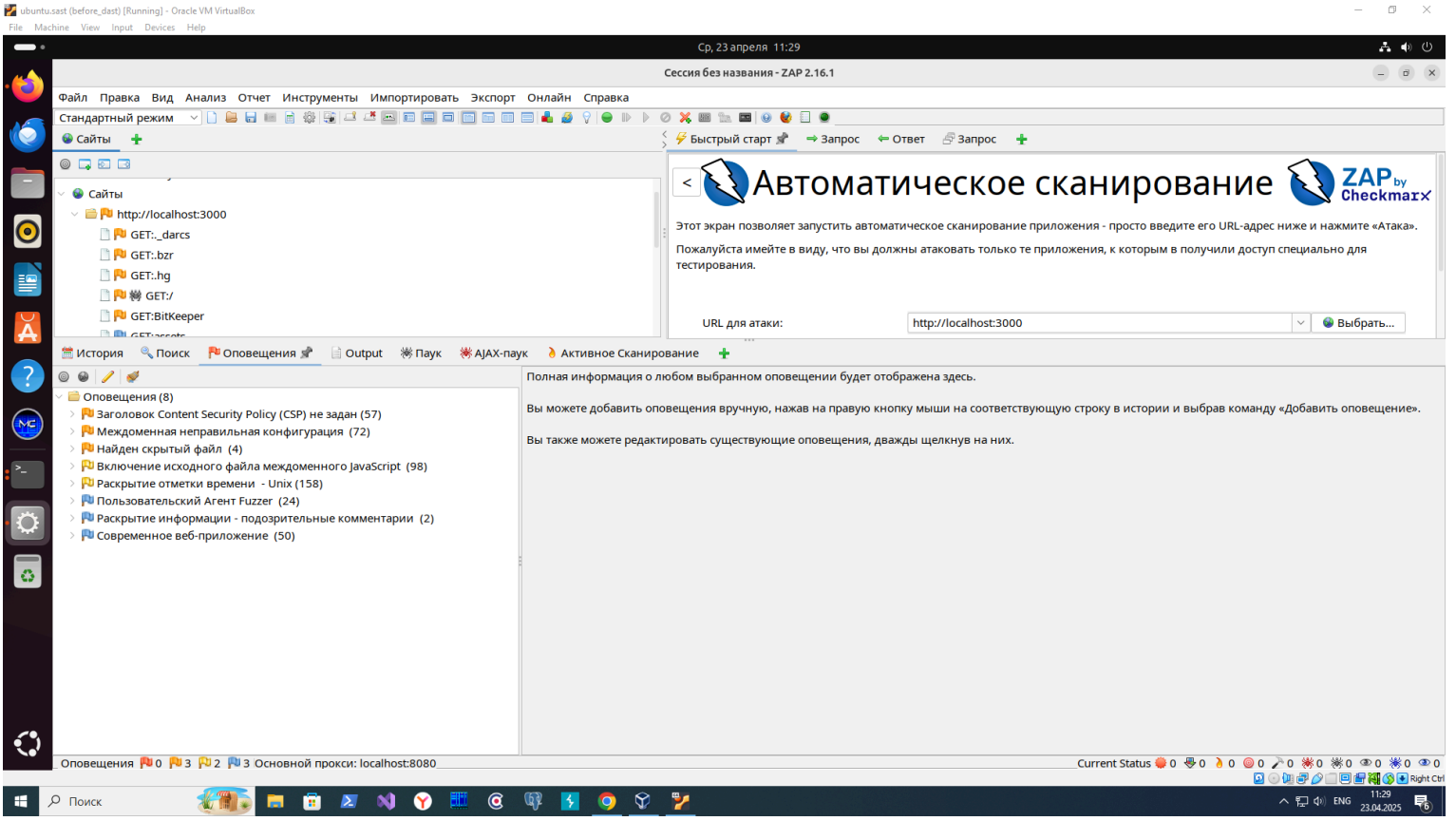


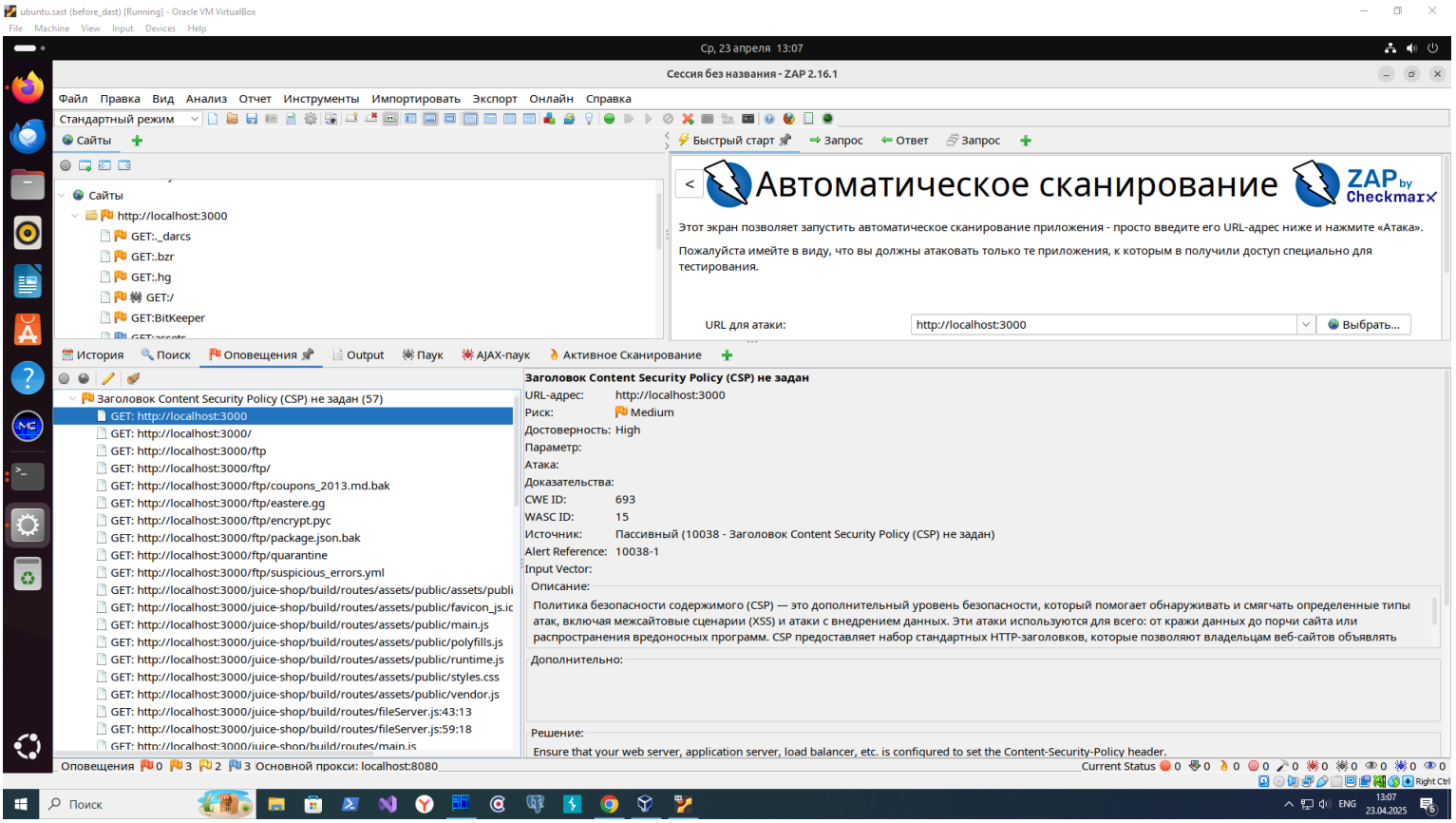

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
Ср, 23 апреля 08:52
Сессия без названия - ZAP 2.16.1
Файл Правка Вид Анализ Отчет Инструменты Импортировать Экспорт Онлайн Справка
Стандартный режим
Сайты
Быстрый старт
Запрос
Ответ
Запрос
Контексты
Контекст по умолчанию
Сайты
Автоматическое сканирование
ZAP by Checkmarx
Этот экран позволяет запустить автоматическое сканирование приложения - просто введите его URL-адрес ниже и нажмите «Атака».
Пожалуйста имейте в виду, что вы должны атаковать только те приложения, к которым в получили доступ специально для тестирования.
URL для атаки:
http://localhost:3000
Выбрать...
Используйте традиционного паука:
Использование ajax spider
If Modern
с
Firefox Headless
Атака
Остановить
Текущее состояние:
Активное сканирование (атака) URL-адресов, обнаруженных пауком (-ами)
История
Поиск
Оповещения
Output
Паук
AJAX-паук
Активное Сканирование
Новое сканирование
Текущее состояние:
0: http://localhost:3000
6 %
Текущие сканирования: 1
Количество Запросов: 101
Новое оповещение...0
Экспорт
Отправленные сообщения
Отфильтрованные сообщения
Идентификатор
Req. Отметка времени
Resp. Отметка времени
Метод
URL-адрес
Код
Причина
RTT
Размер Resp. Заголовок
Размер Resp. Тело
284 23.04.2025, 08:52:56 23.04.2025, 08:52:56 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 8 мс 469 байт 80 117 байт
285 23.04.2025, 08:52:56 23.04.2025, 08:52:56 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 37 мс 469 байт 80 117 байт
286 23.04.2025, 08:52:56 23.04.2025, 08:52:56 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 26 мс 469 байт 80 117 байт
287 23.04.2025, 08:52:56 23.04.2025, 08:52:56 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 14 мс 469 байт 80 117 байт
288 23.04.2025, 08:52:56 23.04.2025, 08:52:56 GET http://localhost:3000/WEB-INF/web.xml 200 OK 13 мс 469 байт 80 117 байт
289 23.04.2025, 08:52:56 23.04.2025, 08:52:56 GET http://localhost:3000/WEB-INF/applicationContext.... 200 OK 25 мс 469 байт 80 117 байт
290 23.04.2025, 08:52:56 23.04.2025, 08:52:56 GET http://localhost:3000/WEB-INF/classes/favicon_js/i... 200 OK 67 мс 469 байт 80 117 байт
291 23.04.2025, 08:52:57 23.04.2025, 08:52:57 GET http://localhost:3000/assets 301 Moved Permane... 2 мс 416 байт 156 байт
292 23.04.2025, 08:52:57 23.04.2025, 08:52:57 GET http://localhost:3000/assets/public 301 Moved Permane... 8 мс 423 байт 163 байт
293 23.04.2025, 08:52:57 23.04.2025, 08:52:57 GET http://localhost:3000/assets?-s 301 Moved Permane... 10 мс 419 байт 159 байт
Загрузки
Оповещения 0 2 2 2 Основной прокси: localhost:8080
Current Status 0 0 1 0 0 0 0 0 0 0 0
Right Ctrl
Поиск
ENG
8:52
23.04.2025

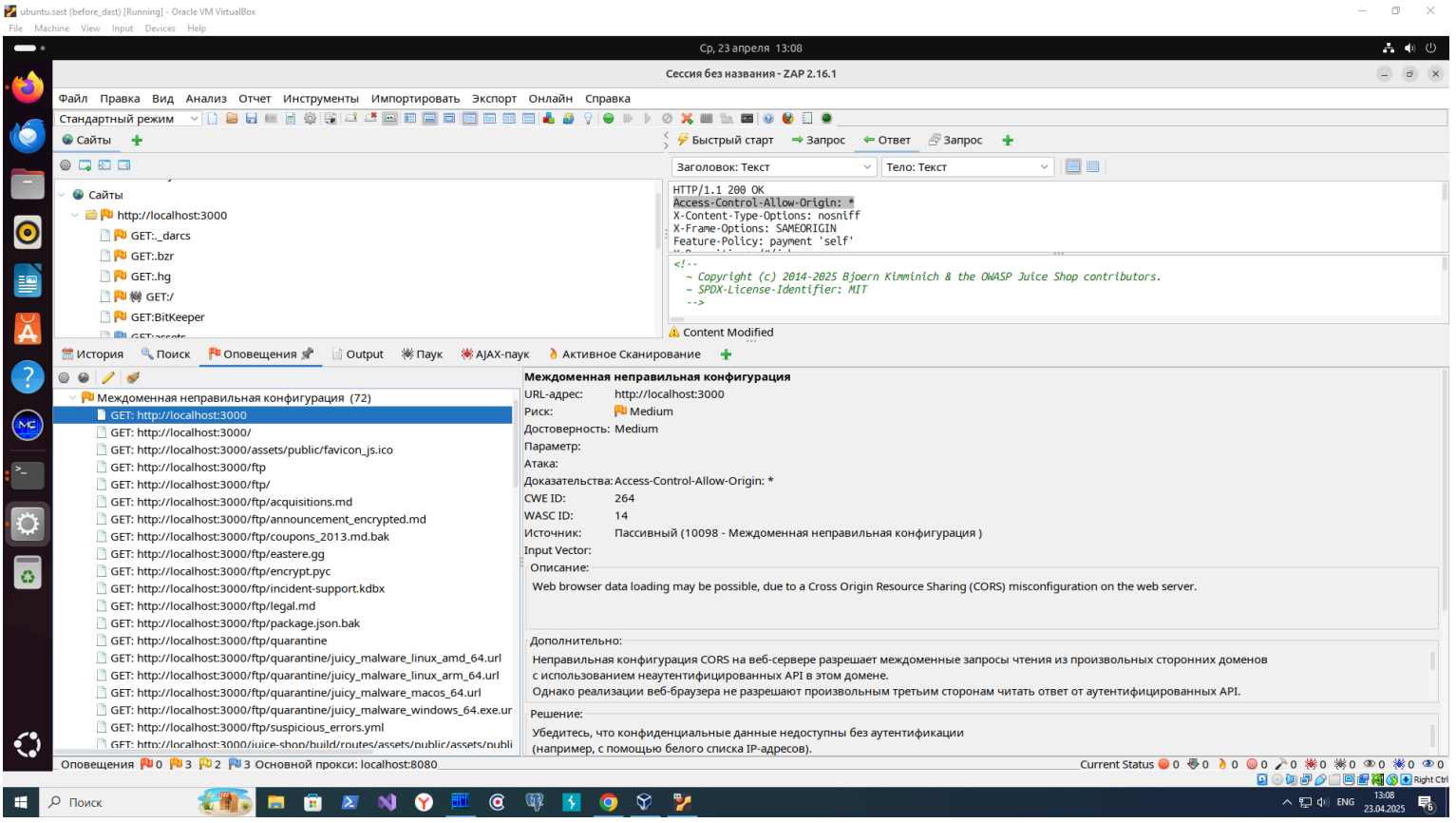

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
Ср, 23 апреля 08:56
Сессия без названия - ZAP 2.16.1
Файл Правка Вид Анализ Отчет Инструменты Импортировать Экспорт Онлайн Справка
Стандартный режим
Сайты
Быстрый старт
Запрос
Ответ
Запрос
Контексты
Контекст по умолчанию
Сайты
Автоматическое сканирование
ZAP by Checkmarx
Этот экран позволяет запустить автоматическое сканирование приложения - просто введите его URL-адрес ниже и нажмите «Атака».
Пожалуйста имейте в виду, что вы должны атаковать только те приложения, к которым в получили доступ специально для тестирования.
URL для атаки:
http://localhost:3000
Выбрать...
Используйте традиционного паука:
Использование ajax spider
If Modern
с
Firefox Headless
Атака
Остановить
Текущее состояние:
Активное сканирование (атака) URL-адресов, обнаруженных пауком (-ами)
История
Поиск
Оповещения
Output
Паук
AJAX-паук
Активное Сканирование
Новое сканирование
Текущее состояние:
0: http://localhost:3000
28 %
Текущие сканирования: 1
Количество Запросов: 1021
Новое оповещение...0
Экспорт
Отправленные сообщения
Отфильтрованные сообщения
Идентификатор
Req. Отметка времени
Resp. Отметка времени
Метод
URL-адрес
Код
Причина
RTT
Размер Resp. Заголовок
Размер Resp. Тело
1 204 23.04.2025, 08:56:40 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/ex... 200 OK 274 мс 469 байт 80 117 байт
1 205 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/ex... 200 OK 30 мс 469 байт 80 117 байт
1 206 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/ex... 200 OK 24 мс 469 байт 80 117 байт
1 207 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/ex... 200 OK 7 мс 469 байт 80 117 байт
1 208 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/ex... 200 OK 31 мс 469 байт 80 117 байт
1 209 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 5 мс 469 байт 80 117 байт
1 210 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 24 мс 469 байт 80 117 байт
1 211 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 10 мс 469 байт 80 117 байт
1 212 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 10 мс 469 байт 80 117 байт
1 213 23.04.2025, 08:56:41 23.04.2025, 08:56:41 GET http://localhost:3000/juice-shop/node_modules/se... 200 OK 18 мс 469 байт 80 117 байт
Оповещения 0 2 2 2 Основной прокси: localhost:8080
Current Status 0 0 1 0 0 0 0 0 0 0 0
Right Ctrl
Поиск
ENG
8:56
23.04.2025

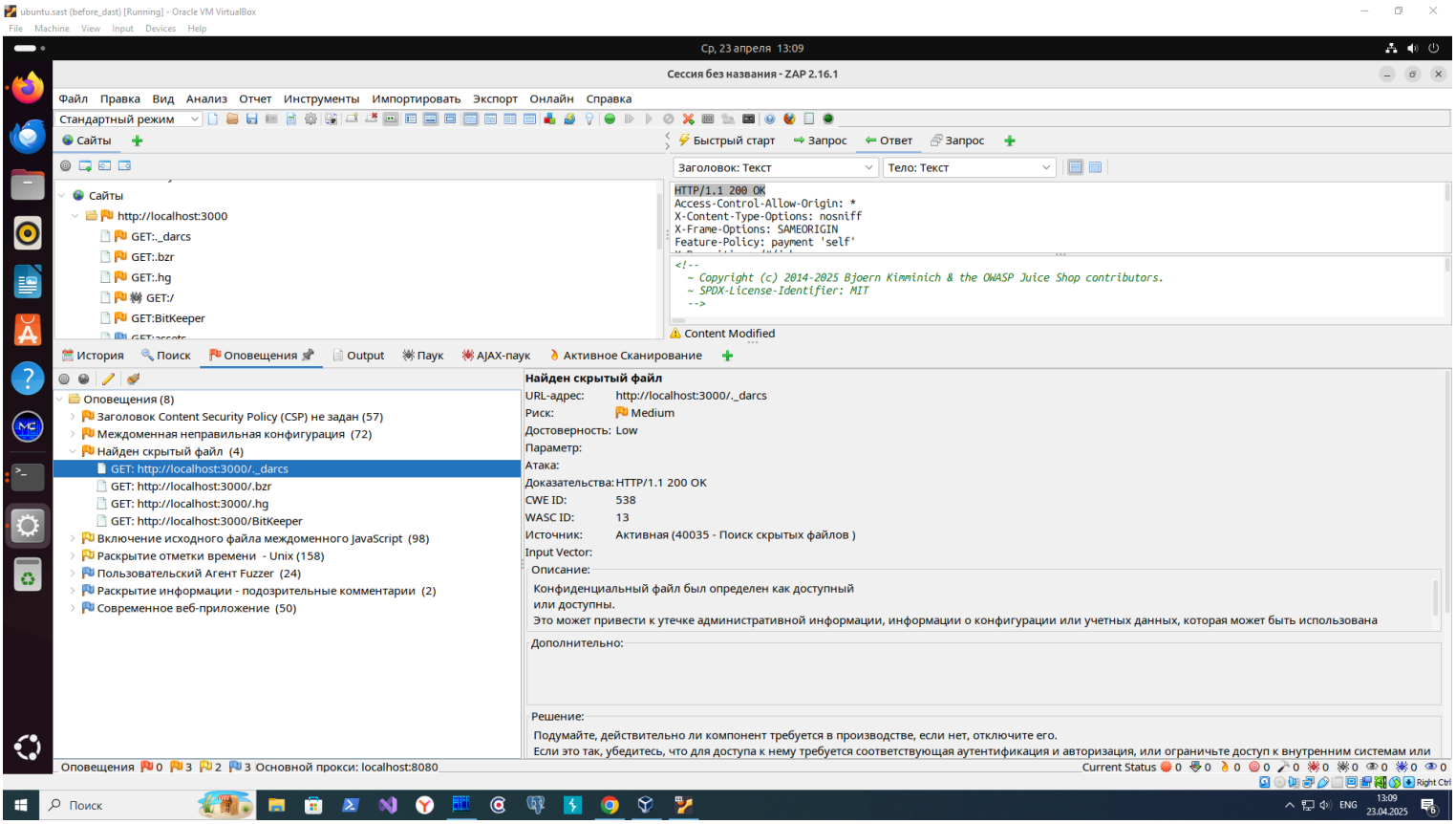


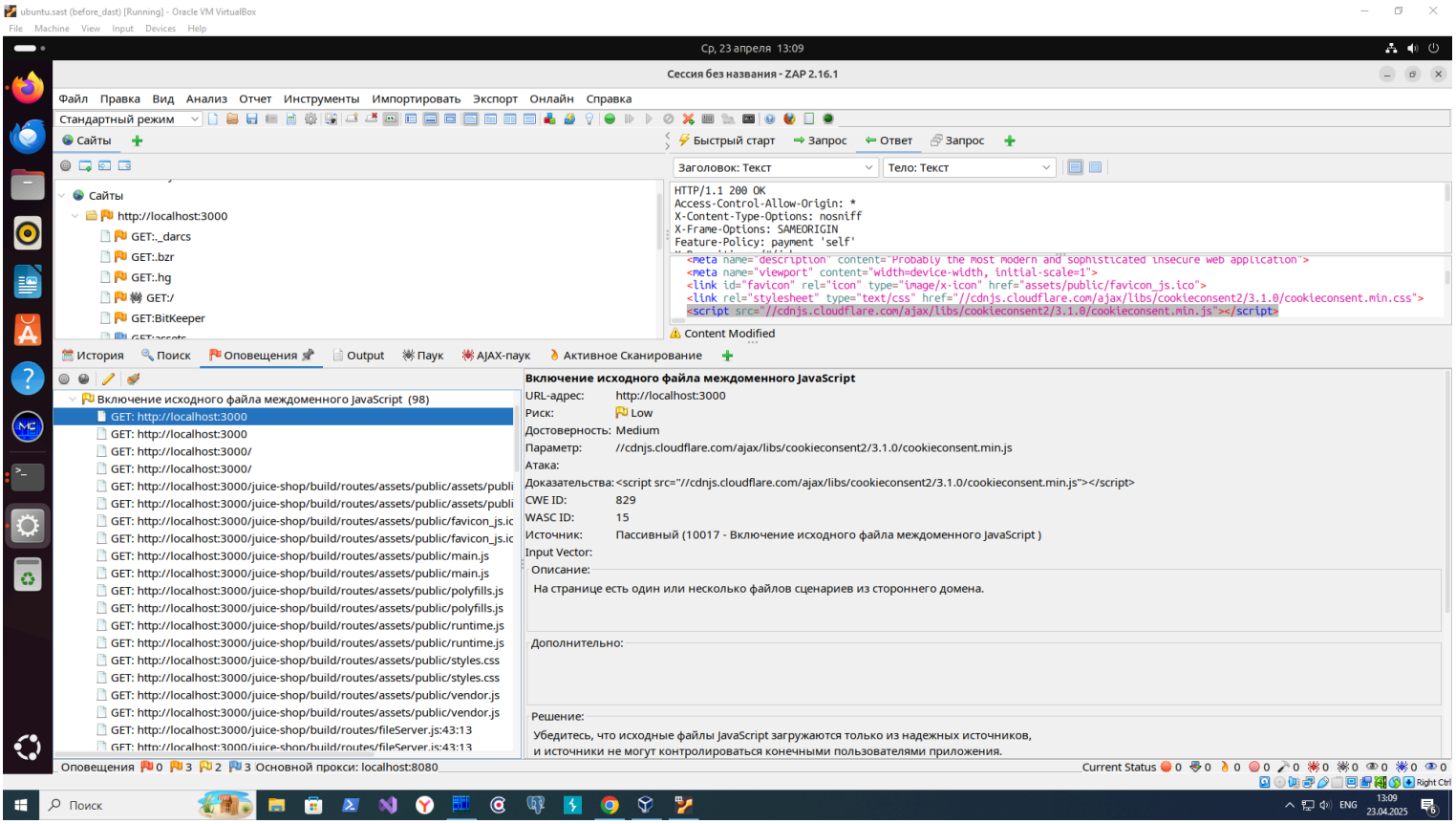
Результаты сканирования OWASP Juice Shop

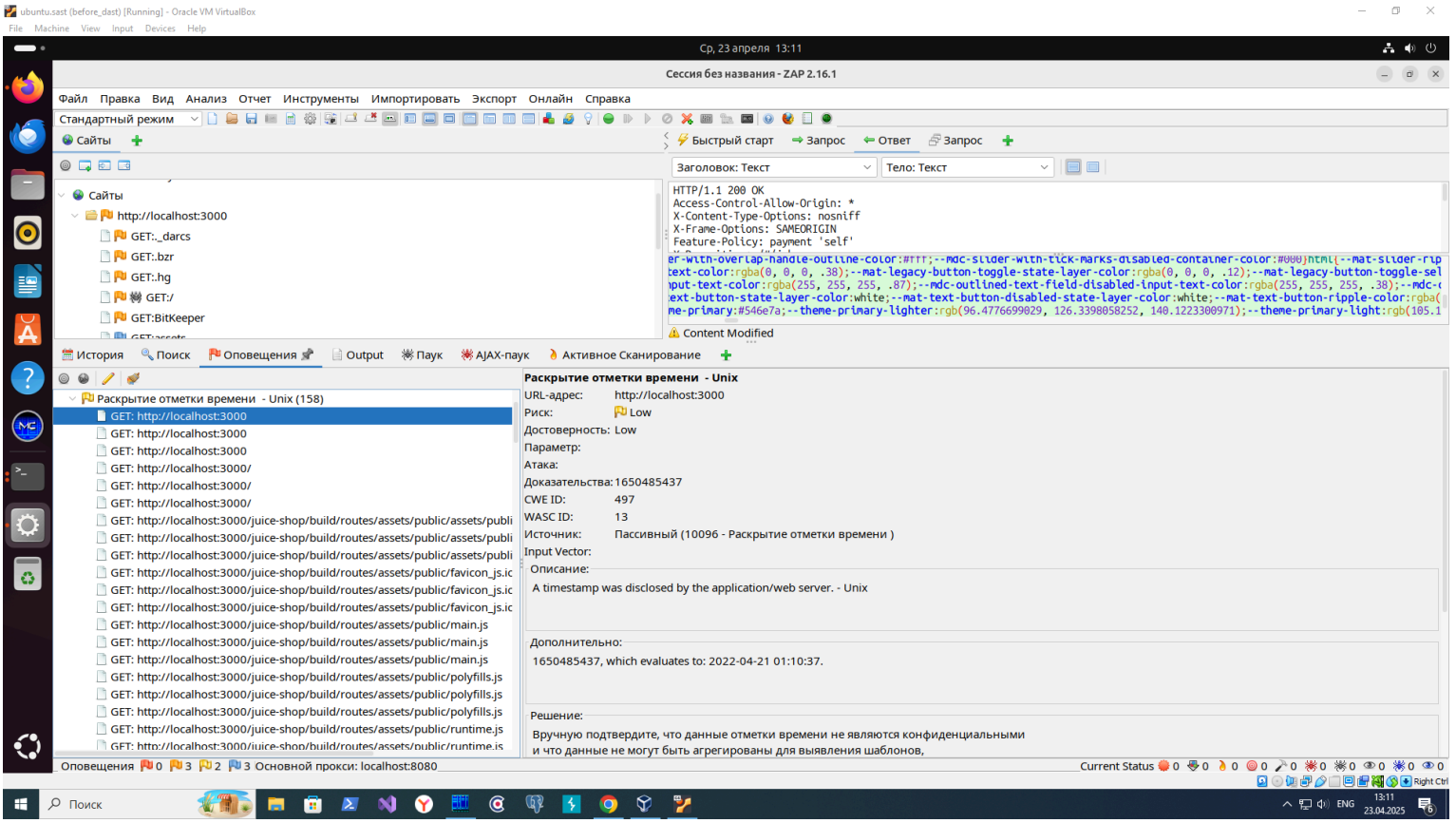


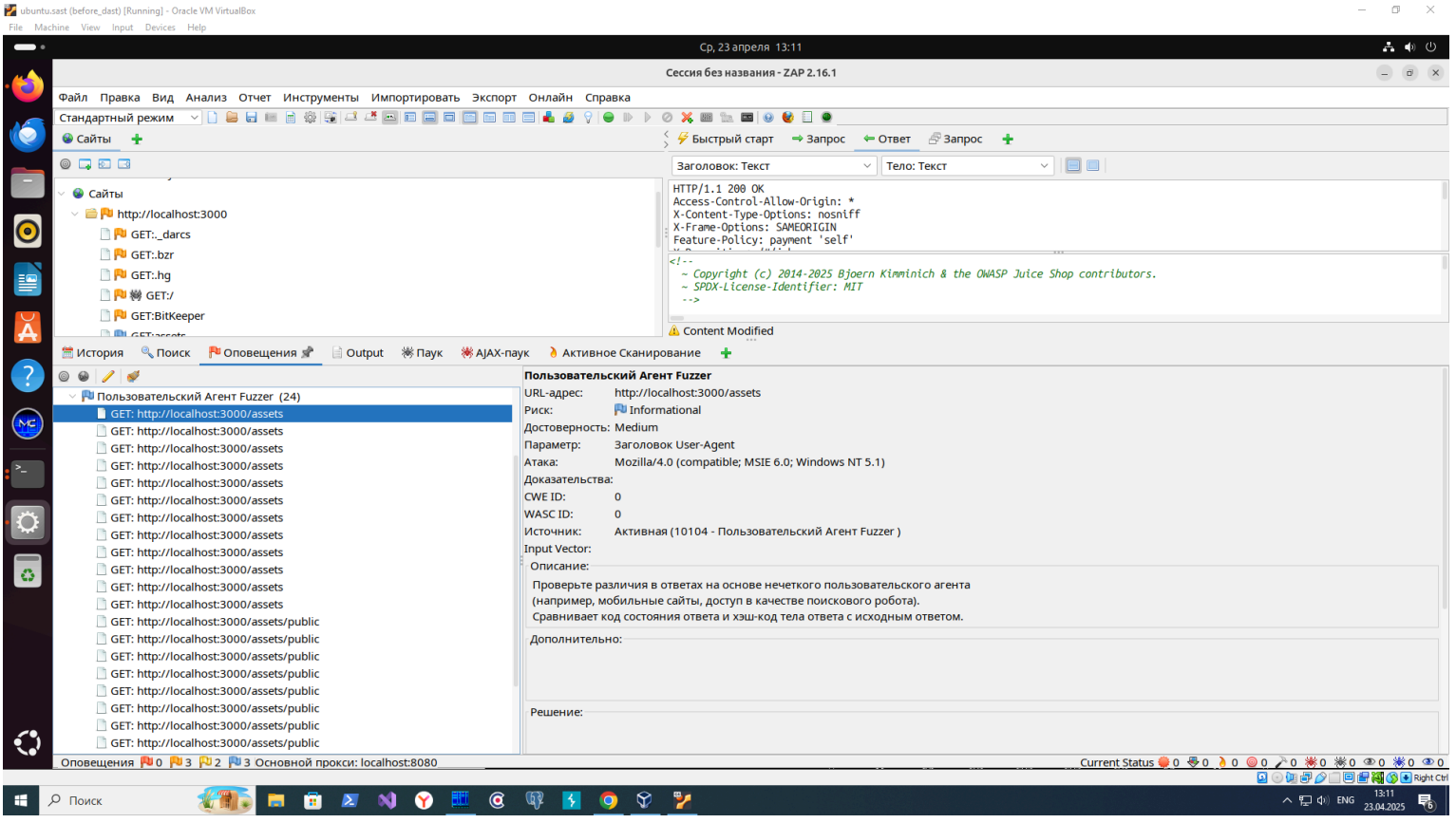

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
Ср, 23 апреля 13:07
Сессия без названия - ZAP 2.16.1
Файл Правка Вид Анализ Отчет Инструменты Импортировать Экспорт Онлайн Справка
Стандартный режим
Сайты
Быстрый старт
Запрос
Ответ
Запрос
Автоматическое сканирование
ZAP by Checkmarx
Этот экран позволяет запустить автоматическое сканирование приложения - просто введите его URL-адрес ниже и нажмите «Атака».
Пожалуйста имейте в виду, что вы должны атаковать только те приложения, к которым в получили доступ специально для тестирования.
URL для атаки:
http://localhost:3000
Выбрать...
http://localhost:3000
GET:._darcs
GET:.bzr
GET:.hg
GET:/
GET:BitKeeper
История
Поиск
Оповещения
Output
Паук
AJAX-паук
Активное Сканирование
Заголовок Content Security Policy (CSP) не задан (57)
GET: http://localhost:3000
GET: http://localhost:3000/
GET: http://localhost:3000/ftp
GET: http://localhost:3000/ftp/
GET: http://localhost:3000/ftp/coupons_2013.md.bak
GET: http://localhost:3000/ftp/eastere.gg
GET: http://localhost:3000/ftp/encrypt.pyc
GET: http://localhost:3000/ftp/package.json.bak
GET: http://localhost:3000/ftp/quarantine
GET: http://localhost:3000/ftp/suspicious_errors.yml
GET: http://localhost:3000/juice-shop/build/routes/assets/public/assets/publi
GET: http://localhost:3000/juice-shop/build/routes/assets/public/favicon_js.ic
GET: http://localhost:3000/juice-shop/build/routes/assets/public/main.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/polyfills.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/runtime.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/styles.css
GET: http://localhost:3000/juice-shop/build/routes/assets/public/vendor.js
GET: http://localhost:3000/juice-shop/build/routes/fileServer.js:43:13
GET: http://localhost:3000/juice-shop/build/routes/fileServer.js:59:18
GET: http://localhost:3000/juice-shop/build/routes/main.js
Заголовок Content Security Policy (CSP) не задан
URL-адрес: http://localhost:3000
Риск: Medium
Достоверность: High
Параметр:
Атака:
Доказательства:
CWE ID: 693
WASC ID: 15
Источник: Пассивный (10038 - Заголовок Content Security Policy (CSP) не задан)
Alert Reference: 10038-1
Input Vector:
Описание:
Политика безопасности содержимого (CSP) — это дополнительный уровень безопасности, который помогает обнаруживать и смягчать определенные типы атак, включая межсайтовые сценарии (XSS) и атаки с внедрением данных. Эти атаки используются для всего: от кражи данных до порчи сайта или распространения вредоносных программ. CSP предоставляет набор стандартных HTTP-заголовков, которые позволяют владельцам веб-сайтов объявлять
Дополнительно:
Решение:
Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
Оповещения 0 3 2 3 Основной прокси: localhost:8080
Current Status
Поиск
ENG
13:07
23.04.2025

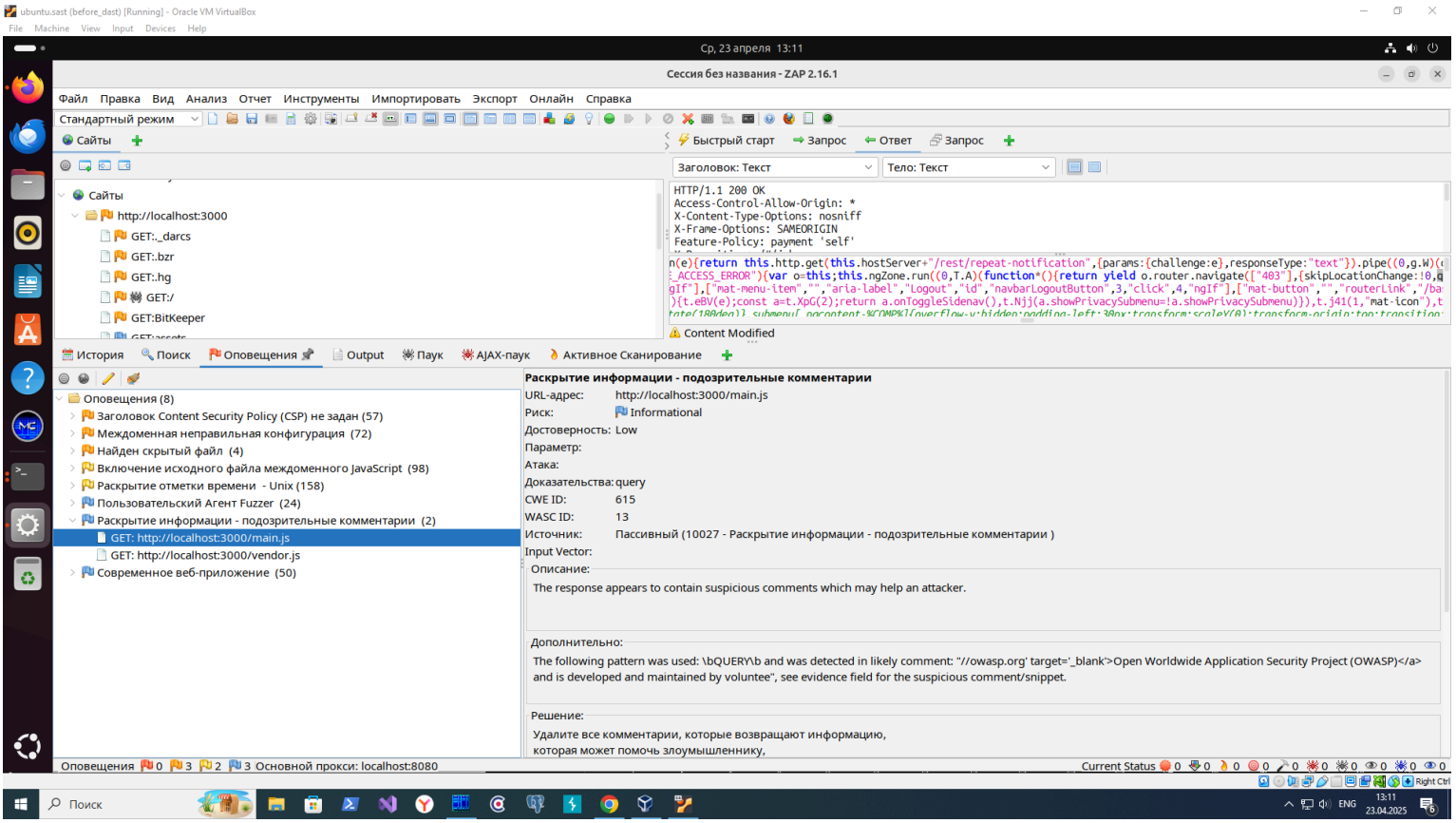

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
Ср, 23 апреля 13:08
Сессия без названия - ZAP 2.16.1
Файл Правка Вид Анализ Отчет Инструменты Импортировать Экспорт Онлайн Справка
Стандартный режим
Сайты
Быстрый старт
Запрос
Ответ
Запрос
Заголовок: Текст
Тело: Текст
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
<!--
~ Copyright (c) 2014-2025 Bjoern Kimminich & the OWASP Juice Shop contributors.
~ SPDX-License-Identifier: MIT
-->
Content Modified
http://localhost:3000
GET:._darcs
GET:.bzr
GET:.hg
GET:/
GET:BitKeeper
История
Поиск
Оповещения
Output
Паук
AJAX-паук
Активное Сканирование
Междоменная неправильная конфигурация (72)
GET: http://localhost:3000
GET: http://localhost:3000/
GET: http://localhost:3000/assets/public/favicon_js.ico
GET: http://localhost:3000/ftp
GET: http://localhost:3000/ftp/
GET: http://localhost:3000/ftp/acquisitions.md
GET: http://localhost:3000/ftp/announcement_encrypted.md
GET: http://localhost:3000/ftp/coupons_2013.md.bak
GET: http://localhost:3000/ftp/eastere.gg
GET: http://localhost:3000/ftp/encrypt.pyc
GET: http://localhost:3000/ftp/incident-support.kdbx
GET: http://localhost:3000/ftp/legal.md
GET: http://localhost:3000/ftp/package.json.bak
GET: http://localhost:3000/ftp/quarantine
GET: http://localhost:3000/ftp/quarantine/juicy_malware_linux_amd_64.url
GET: http://localhost:3000/ftp/quarantine/juicy_malware_linux_arm_64.url
GET: http://localhost:3000/ftp/quarantine/juicy_malware_macos_64.url
GET: http://localhost:3000/ftp/quarantine/juicy_malware_windows_64.exe.ur
GET: http://localhost:3000/ftp/suspicious_errors.yml
GET: http://localhost:3000/juice-shop/build/routes/assets/public/assets/publi
Междоменная неправильная конфигурация
URL-адрес: http://localhost:3000
Риск: Medium
Достоверность: Medium
Параметр:
Атака:
Доказательства: Access-Control-Allow-Origin: *
CWE ID: 264
WASC ID: 14
Источник: Пассивный (10098 - Междоменная неправильная конфигурация)
Input Vector:
Описание:
Web browser data loading may be possible, due to a Cross Origin Resource Sharing (CORS) misconfiguration on the web server.
Дополнительно:
Неправильная конфигурация CORS на веб-сервере разрешает междоменные запросы чтения из произвольных сторонних доменов с использованием неаутентифицированных API в этом домене.
Однако реализации веб-браузера не разрешают произвольным третьим сторонам читать ответ от аутентифицированных API.
Решение:
Убедитесь, что конфиденциальные данные недоступны без аутентификации (например, с помощью белого списка IP-адресов).
Оповещения 0 3 2 3 Основной прокси: localhost:8080
Current Status
Поиск
ENG
13:08
23.04.2025

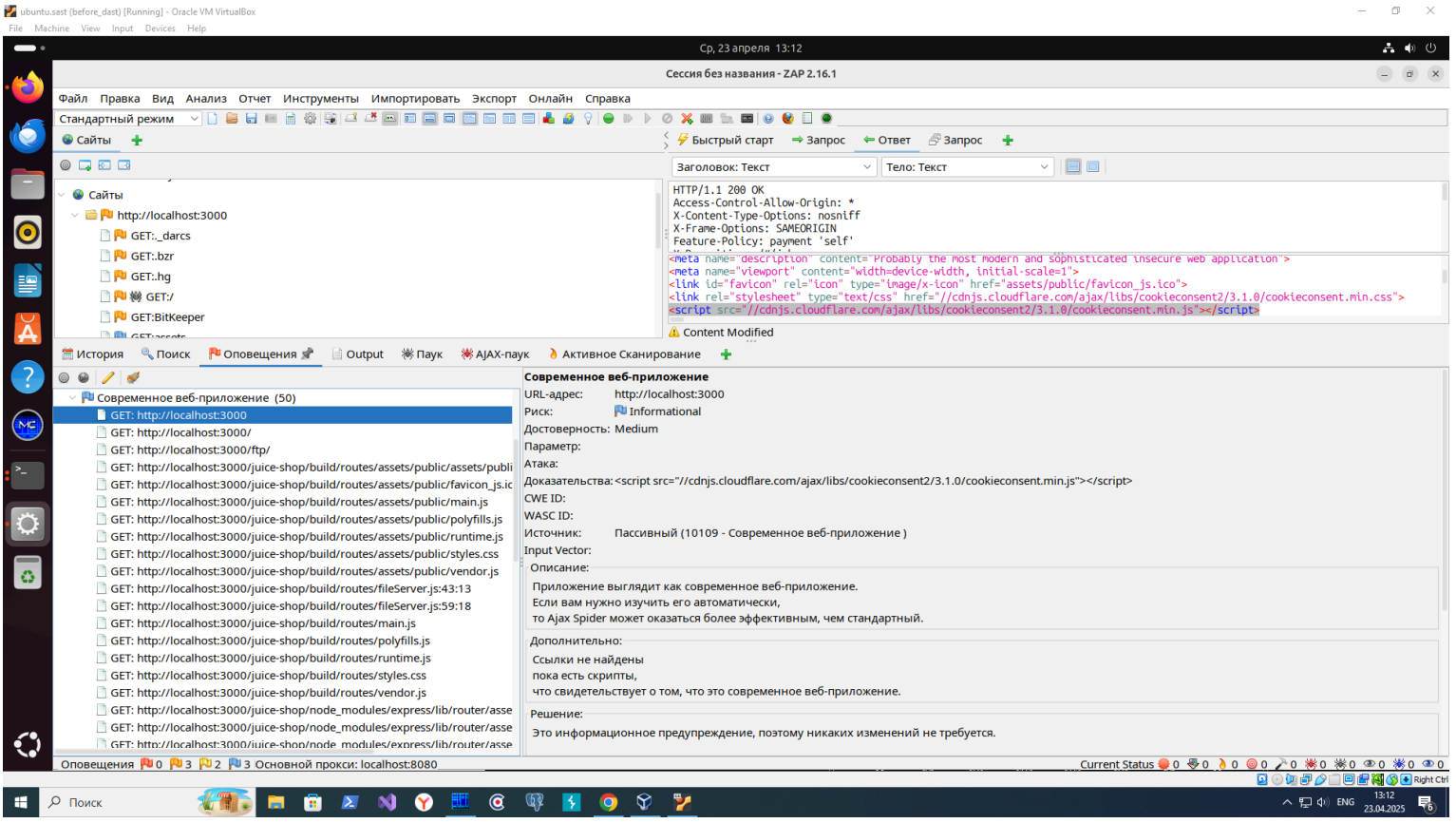

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
Ср, 23 апреля 13:09
Сессия без названия - ZAP 2.16.1
Файл Правка Вид Анализ Отчет Инструменты Импортировать Экспорт Онлайн Справка
Стандартный режим
Сайты
Быстрый старт Запрос Ответ Запрос
Заголовок: Текст Тело: Текст
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
<!--
~ Copyright (c) 2014-2025 Bjoern Kimminich & the OWASP Juice Shop contributors.
~ SPDX-License-Identifier: MIT
-->
Content Modified
http://localhost:3000
GET:._darcs
GET:.bzr
GET:.hg
GET:/
GET:BitKeeper
История Поиск Оповещения Output Паук AJAX-паук Активное Сканирование
Оповещения (8)
Заголовок Content Security Policy (CSP) не задан (57)
Междоменная неправильная конфигурация (72)
Найден скрытый файл (4)
GET: http://localhost:3000/._darcs
GET: http://localhost:3000/.bzr
GET: http://localhost:3000/.hg
GET: http://localhost:3000/BitKeeper
Включение исходного файла междоменного JavaScript (98)
Раскрытие отметки времени - Unix (158)
Пользовательский Агент Fuzzer (24)
Раскрытие информации - подозрительные комментарии (2)
Современное веб-приложение (50)
Найден скрытый файл
URL-адрес: http://localhost:3000/._darcs
Риск: Medium
Достоверность: Low
Параметр:
Атака:
Доказательства: HTTP/1.1 200 OK
CWE ID: 538
WASC ID: 13
Источник: Активная (40035 - Поиск скрытых файлов)
Input Vector:
Описание:
Конфиденциальный файл был определен как доступный
или доступны.
Это может привести к утечке административной информации, информации о конфигурации или учетных данных, которая может быть использована
Дополнительно:
Решение:
Подумайте, действительно ли компонент требуется в производстве, если нет, отключите его.
Если это так, убедитесь, что для доступа к нему требуется соответствующая аутентификация и авторизация, или ограничьте доступ к внутренним системам или
Оповещения 0 3 2 3 Основной прокси: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0 0
Поиск
ENG 13:09 23.04.2025


ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
Ср, 23 апреля 13:09
Сессия без названия - ZAP 2.16.1
Файл Правка Вид Анализ Отчет Инструменты Импортировать Экспорт Онлайн Справка
Стандартный режим
Сайты
Быстрый старт Запрос Ответ Запрос
Заголовок: Текст Тело: Текст
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
<meta name="description" content="Probably the most modern and sophisticated insecure web application">
<meta name="viewport" content="width=device-width, initial-scale=1">
<link id="favicon" rel="icon" type="image/x-icon" href="assets/public/favicon_js.ico">
<link rel="stylesheet" type="text/css" href="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.css">
<script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
Content Modified
http://localhost:3000
GET:._darcs
GET:.bzr
GET:.hg
GET:/
GET:BitKeeper
История Поиск Оповещения Output Паук AJAX-паук Активное Сканирование
Включение исходного файла междоменного JavaScript (98)
GET: http://localhost:3000
GET: http://localhost:3000
GET: http://localhost:3000/
GET: http://localhost:3000/
GET: http://localhost:3000/juice-shop/build/routes/assets/public/assets/publi
GET: http://localhost:3000/juice-shop/build/routes/assets/public/assets/publi
GET: http://localhost:3000/juice-shop/build/routes/assets/public/favicon_js.ic
GET: http://localhost:3000/juice-shop/build/routes/assets/public/favicon_js.ic
GET: http://localhost:3000/juice-shop/build/routes/assets/public/main.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/main.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/polyfills.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/polyfills.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/runtime.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/runtime.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/styles.css
GET: http://localhost:3000/juice-shop/build/routes/assets/public/styles.css
GET: http://localhost:3000/juice-shop/build/routes/assets/public/vendor.js
GET: http://localhost:3000/juice-shop/build/routes/assets/public/vendor.js
GET: http://localhost:3000/juice-shop/build/routes/fileServer.js:43:13
GET: http://localhost:3000/juice-shop/build/routes/fileServer.js:43:13
Включение исходного файла междоменного JavaScript
URL-адрес: http://localhost:3000
Риск: Low
Достоверность: Medium
Параметр: //cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js
Атака:
Доказательства: <script src="//cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js"></script>
CWE ID: 829
WASC ID: 15
Источник: Пассивный (10017 - Включение исходного файла междоменного JavaScript)
Input Vector:
Описание:
На странице есть один или несколько файлов сценариев из стороннего домена.
Дополнительно:
Решение:
Убедитесь, что исходные файлы JavaScript загружаются только из надежных источников,
и источники не могут контролироваться конечными пользователями приложения.
Оповещения 0 3 2 3 Основной прокси: localhost:8080
Current Status 0 0 0 0 0 0 0 0 0 0 0
Поиск
ENG 13:09 23.04.2025


ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
Ср, 23 апреля 13:11
Сессия без названия - ZAP 2.16.1
Файл Правка Вид Анализ Отчет Инструменты Импортировать Экспорт Онлайн Справка
Стандартный режим
Сайты
Быстрый старт
Запрос
Ответ
Запрос
Заголовок: Текст
Тело: Текст
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content Modified
http://localhost:3000
GET:._darcs
GET:.bzr
GET:.hg
GET:/
GET:BitKeeper
История
Поиск
Оповещения
Output
Паук
AJAX-паук
Активное Сканирование
Раскрытие отметки времени - Unix (158)
GET: http://localhost:3000
Раскрытие отметки времени - Unix
URL-адрес: http://localhost:3000
Риск: Low
Достоверность: Low
Параметр:
Атака:
Доказательства: 1650485437
CWE ID: 497
WASC ID: 13
Источник: Пассивный (10096 - Раскрытие отметки времени)
Input Vector:
Описание:
A timestamp was disclosed by the application/web server. - Unix
Дополнительно:
1650485437, which evaluates to: 2022-04-21 01:10:37.
Решение:
Вручную подтвердите, что данные отметки времени не являются конфиденциальными
и что данные не могут быть агрегированы для выявления шаблонов,
Оповещения 0 3 2 3 Основной прокси: localhost:8080
Current Status
Поиск
ENG
13:11
23.04.2025

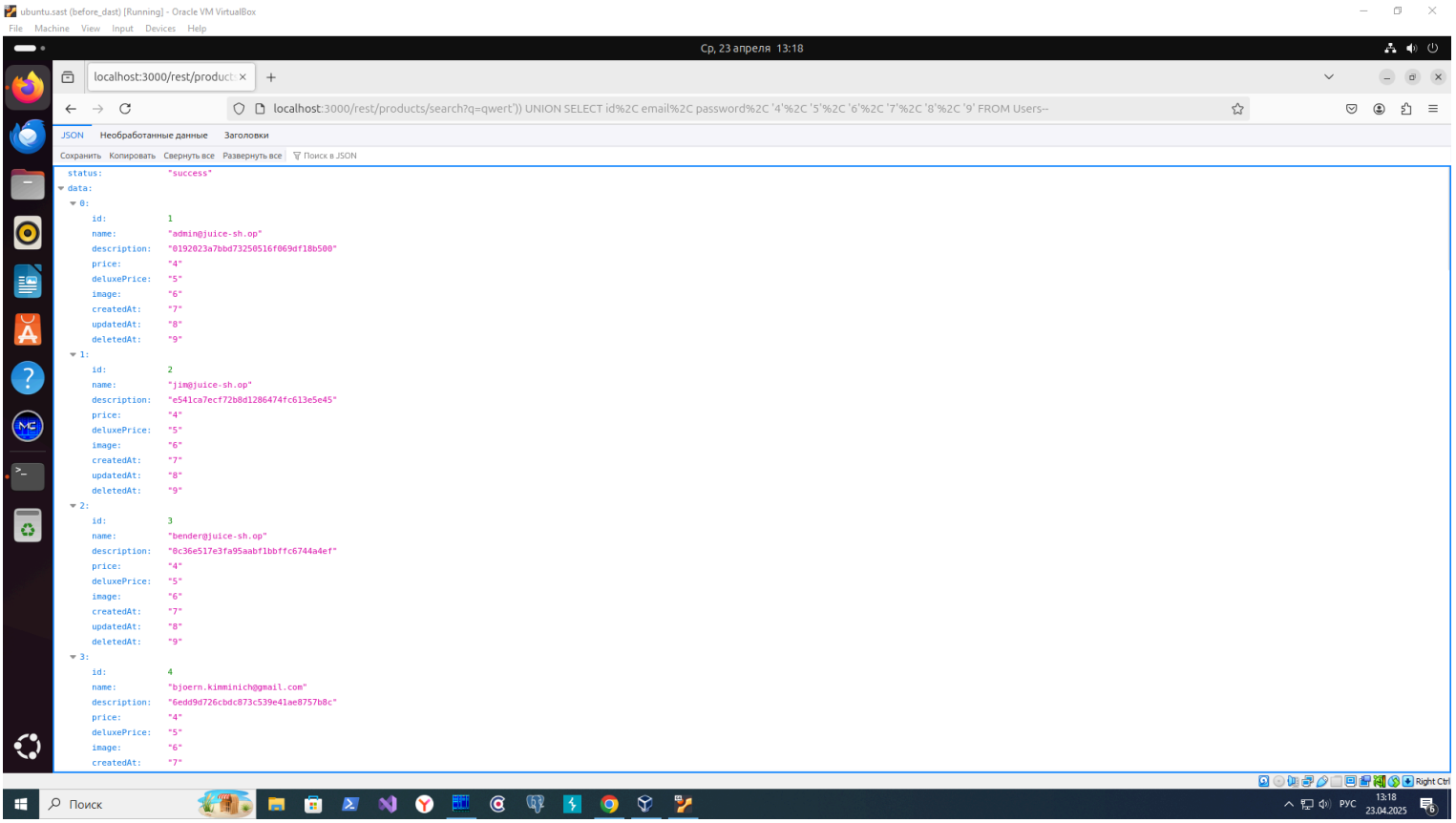

ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
Ср, 23 апреля 13:11
Сессия без названия - ZAP 2.16.1
Файл Правка Вид Анализ Отчет Инструменты Импортировать Экспорт Онлайн Справка
Стандартный режим
Сайты
Быстрый старт
Запрос
Ответ
Запрос
Заголовок: Текст
Тело: Текст
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
<!--
~ Copyright (c) 2014-2025 Bjoern Kimminich & the OWASP Juice Shop contributors.
~ SPDX-License-Identifier: MIT
-->
Content Modified
http://localhost:3000
GET:._darcs
GET:.bzr
GET:.hg
GET:/
GET:BitKeeper
История
Поиск
Оповещения
Output
Паук
AJAX-паук
Активное Сканирование
Пользовательский Агент Fuzzer (24)
GET: http://localhost:3000/assets
Пользовательский Агент Fuzzer
URL-адрес: http://localhost:3000/assets
Риск: Informational
Достоверность: Medium
Параметр: Заголовок User-Agent
Атака: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Доказательства:
CWE ID: 0
WASC ID: 0
Источник: Активная (10104 - Пользовательский Агент Fuzzer)
Input Vector:
Описание:
Проверьте различия в ответах на основе нечеткого пользовательского агента
(например, мобильные сайты, доступ в качестве поискового робота).
Сравнивает код состояния ответа и хэш-код тела ответа с исходным ответом.
Дополнительно:
Решение:
Оповещения 0 3 2 3 Основной прокси: localhost:8080
Current Status
Поиск
ENG
13:11
23.04.2025





Демонстрация наличия SQL инъекции в OWASP Juice Shop
Используем следующий относительный путь:

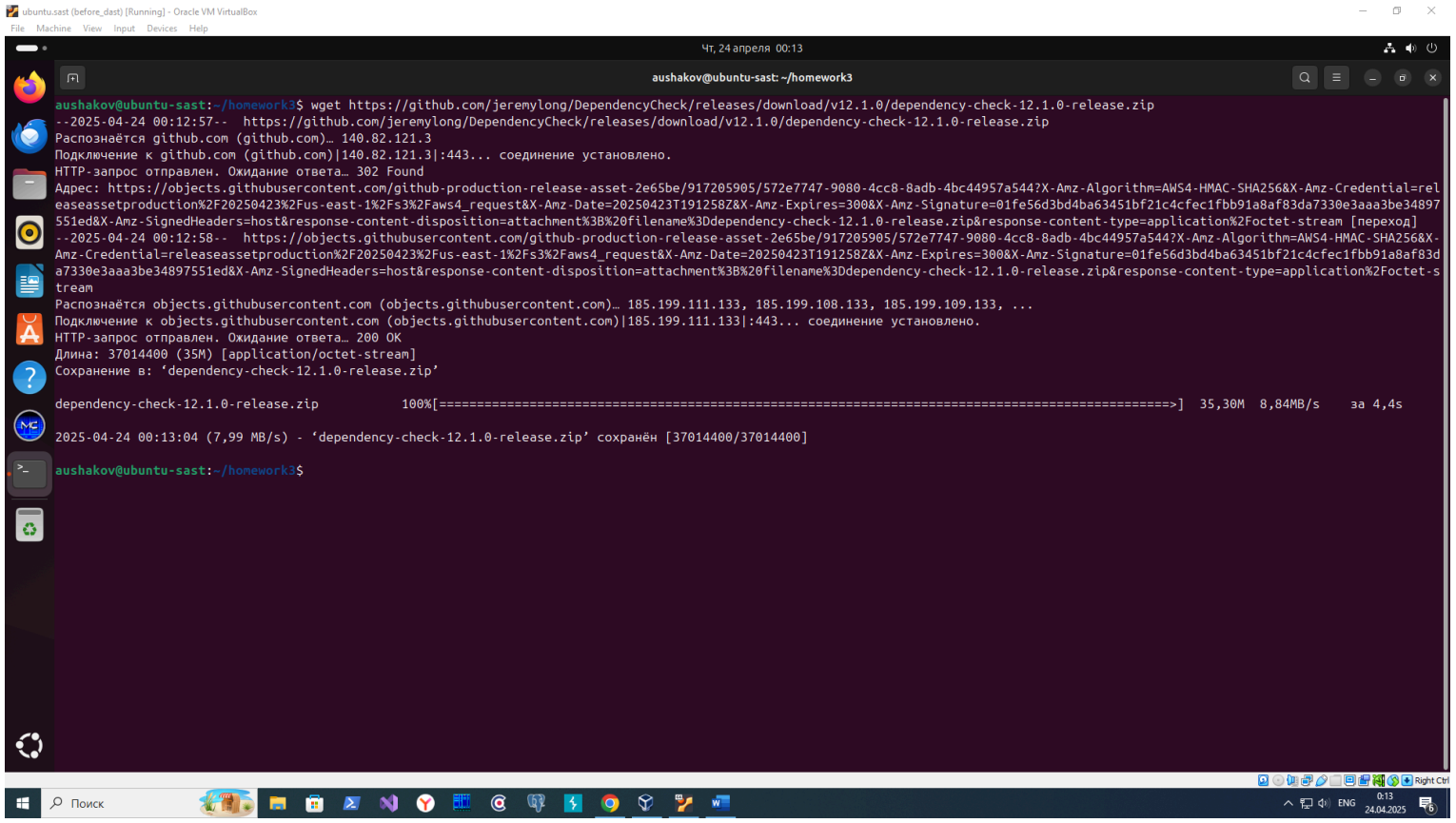
/rest/products/search?q=qwert%27%29%29%20UNION%20SELECT%20id%2C%20email%2C%20password%2C%20%274%27%2C%20%275%27%2C%20%276%27%2C%20%277%27%2C%20%278%27%2C%20%279%27%20FROM%20Users–

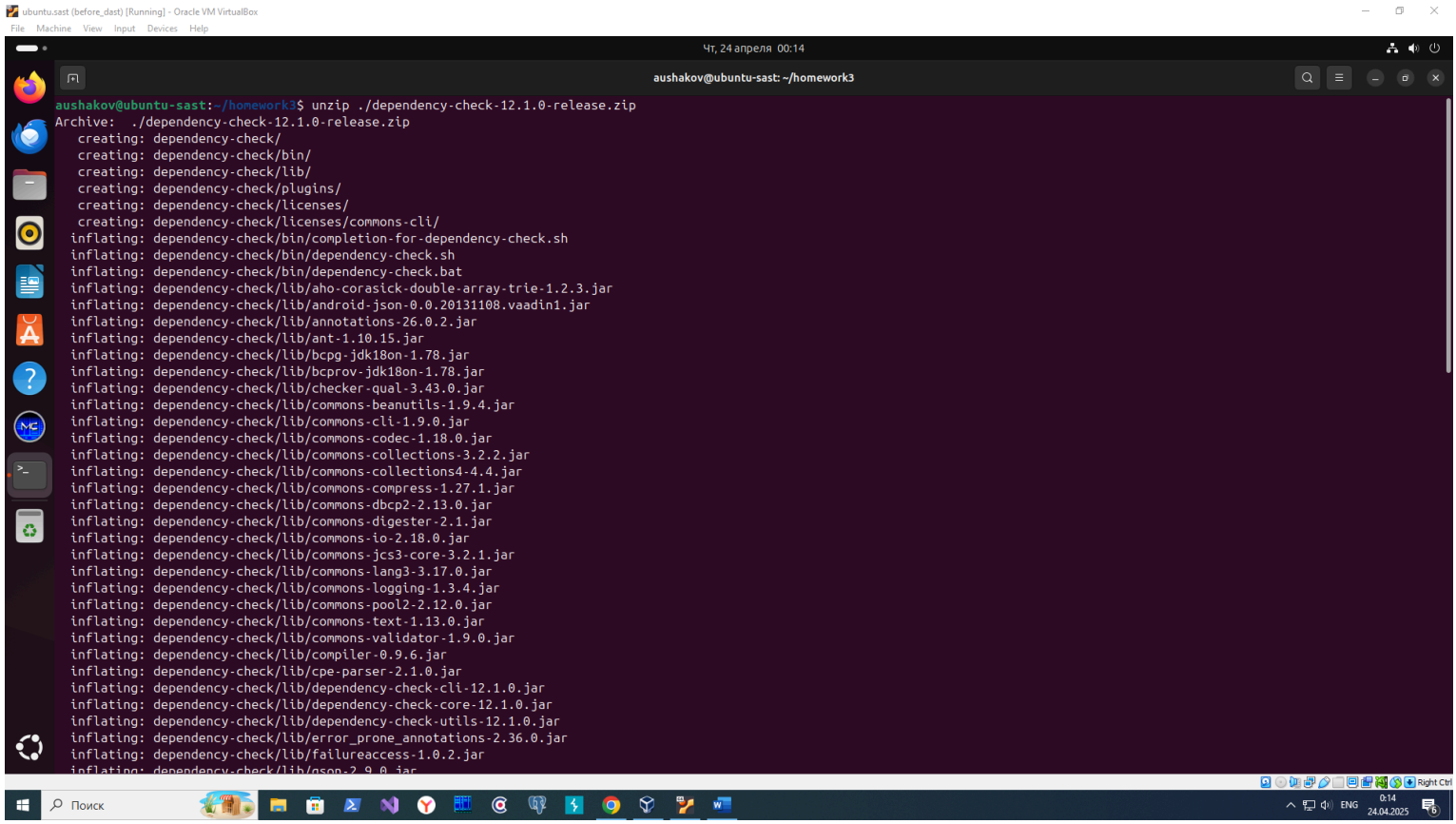


Как результат - мы получаем список всех пользователей в базе вместе с хешами их паролей.

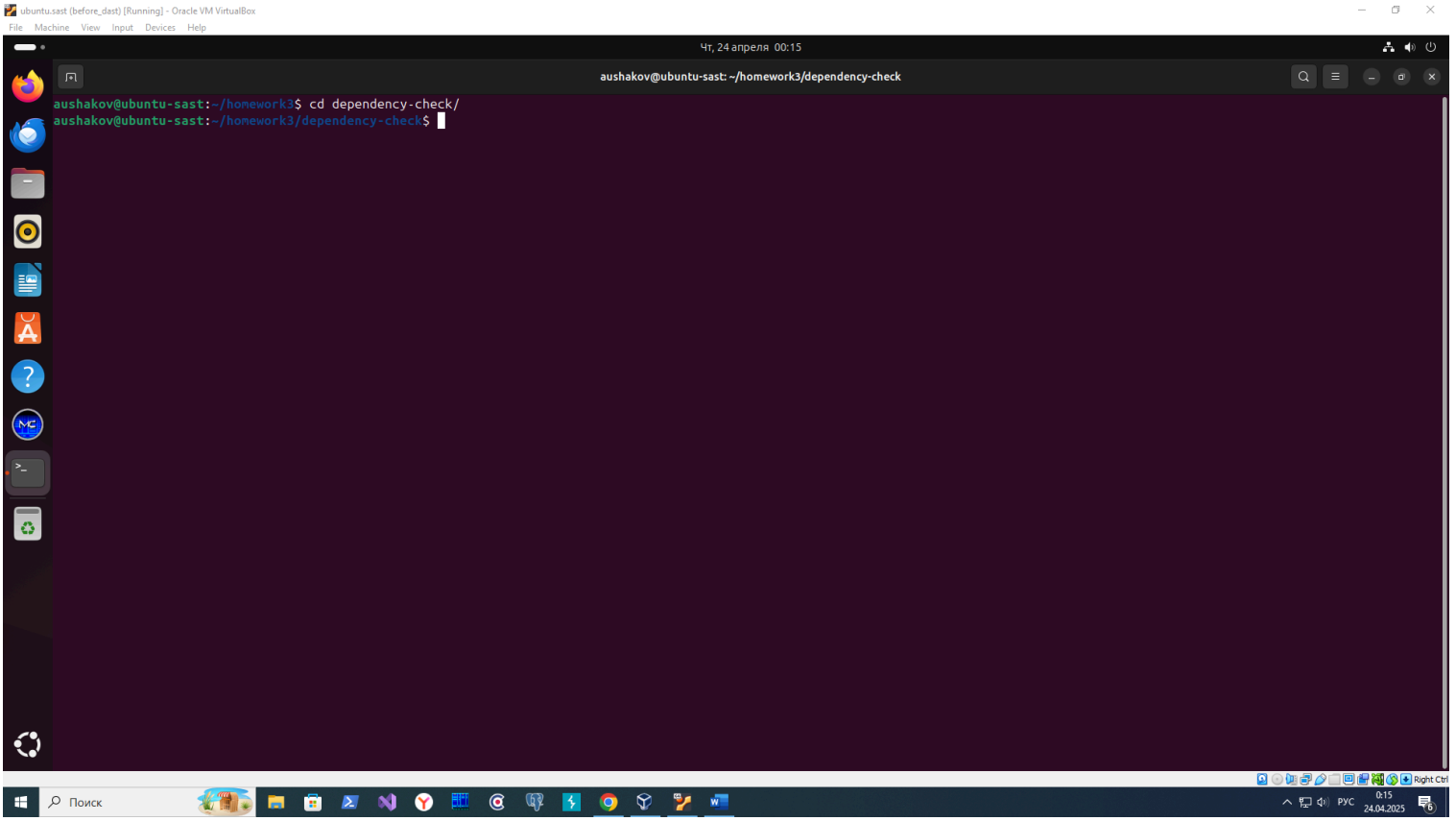
Выводы:
1. DAST средства (в нашем случае это OWASP ZAP) путем динамического анализа (путем тестирования) способны находить некоторые уязвимости (смотри отчет от OWASP ZAP)
2. DAST средства (в нашем случае это OWASP ZAP) не являются панацеей; так, например, в OWASP Juice Shop присутствует SQL инъекция (как я показал это на демонстрации), но у меня OWASP ZAP не нашел ни одной SQL инъекции в OWASP Juice Shop.

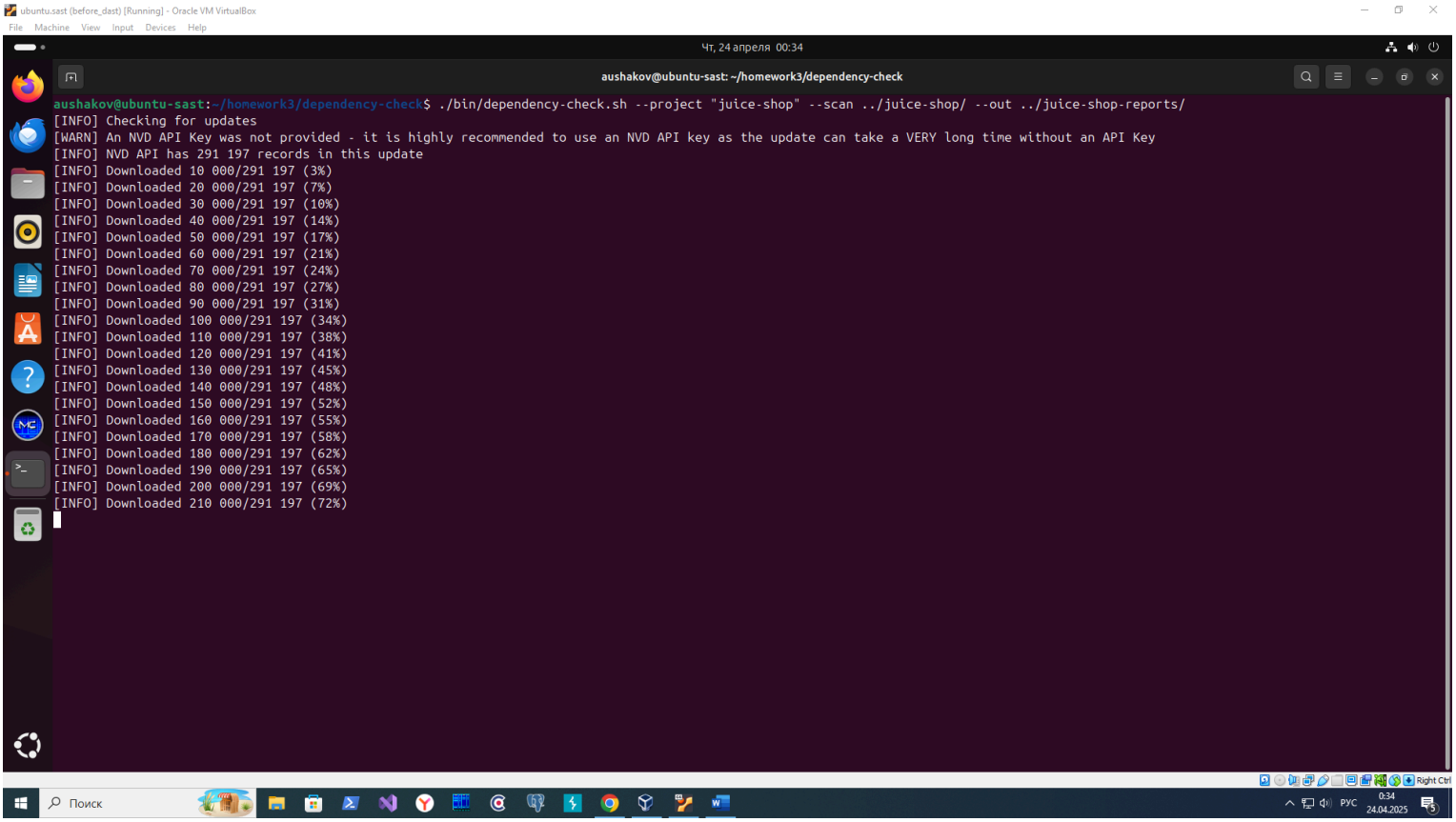
Устанавливаю Dependency-Check

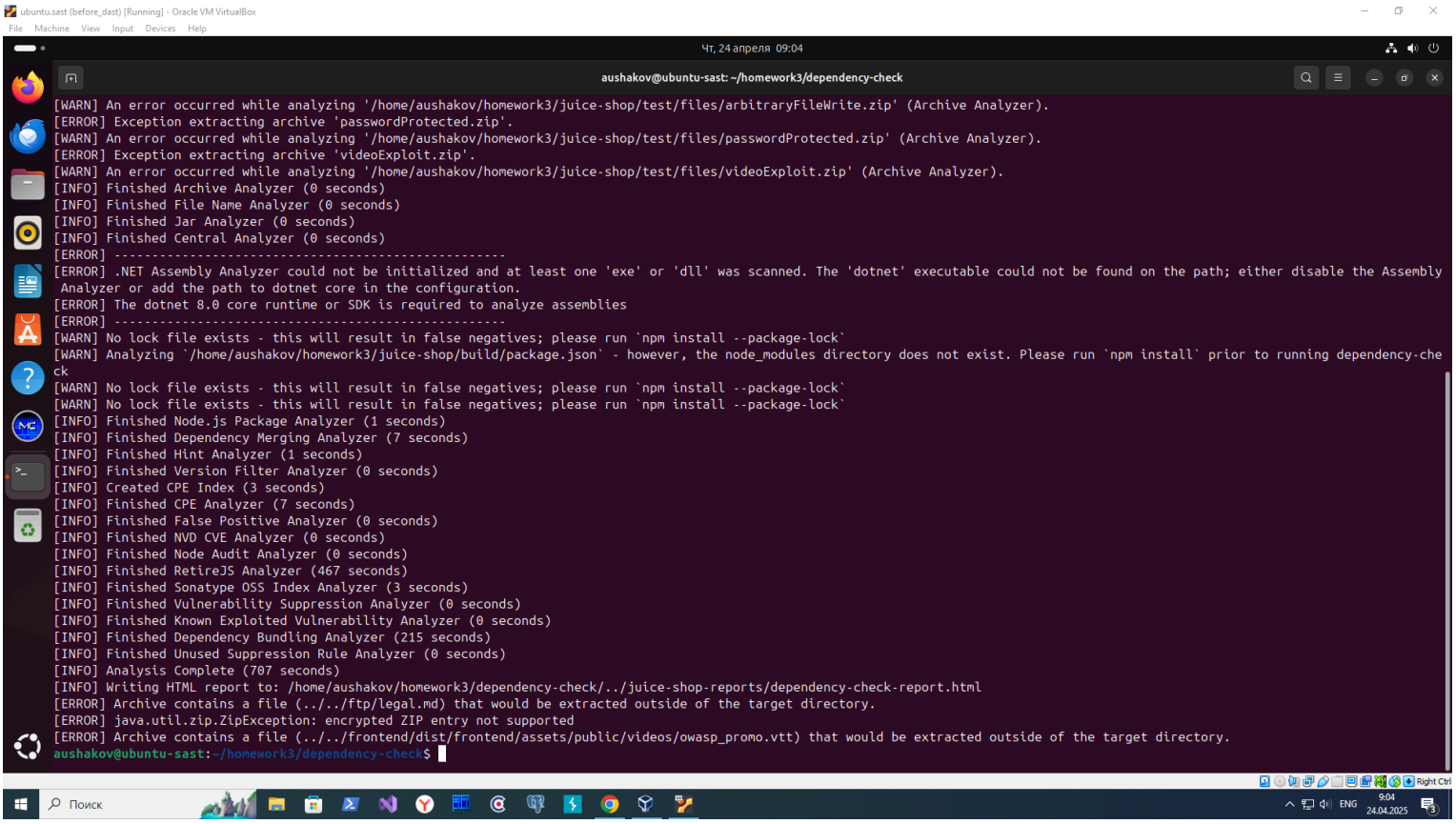




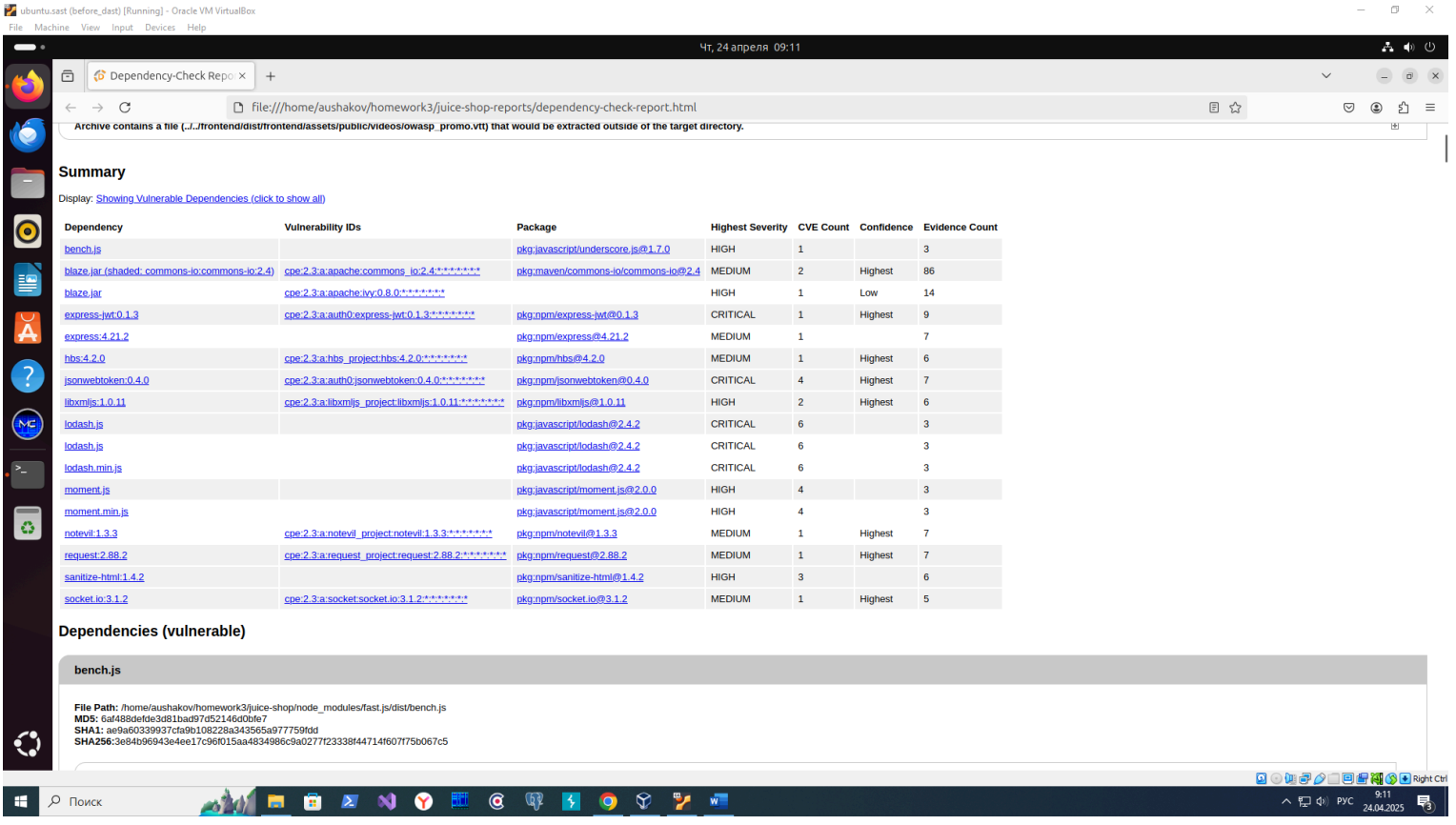
Запускаю Dependency-Check


ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Чт, 24 апреля 00:15
aushakov@ubuntu-sast: ~/homework3/dependency-check
aushakov@ubuntu-sast:~/homework3$ cd dependency-check/
aushakov@ubuntu-sast:~/homework3/dependency-check$
Поиск
РУС
0:15
24.04.2025


ubuntu.sast (before_dast) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Чт, 24 апреля 00:34
aushakov@ubuntu-sast: ~/homework3/dependency-check
aushakov@ubuntu-sast:~/homework3/dependency-check$ ./bin/dependency-check.sh --project "juice-shop" --scan ../juice-shop/ --out ../juice-shop-reports/
[INFO] Checking for updates
[WARN] An NVD API Key was not provided - it is highly recommended to use an NVD API key as the update can take a VERY long time without an API Key
[INFO] NVD API has 291 197 records in this update
[INFO] Downloaded 10 000/291 197 (3%)
[INFO] Downloaded 20 000/291 197 (7%)
[INFO] Downloaded 30 000/291 197 (10%)
[INFO] Downloaded 40 000/291 197 (14%)
[INFO] Downloaded 50 000/291 197 (17%)
[INFO] Downloaded 60 000/291 197 (21%)
[INFO] Downloaded 70 000/291 197 (24%)
[INFO] Downloaded 80 000/291 197 (27%)
[INFO] Downloaded 90 000/291 197 (31%)
[INFO] Downloaded 100 000/291 197 (34%)
[INFO] Downloaded 110 000/291 197 (38%)
[INFO] Downloaded 120 000/291 197 (41%)
[INFO] Downloaded 130 000/291 197 (45%)
[INFO] Downloaded 140 000/291 197 (48%)
[INFO] Downloaded 150 000/291 197 (52%)
[INFO] Downloaded 160 000/291 197 (55%)
[INFO] Downloaded 170 000/291 197 (58%)
[INFO] Downloaded 180 000/291 197 (62%)
[INFO] Downloaded 190 000/291 197 (65%)
[INFO] Downloaded 200 000/291 197 (69%)
[INFO] Downloaded 210 000/291 197 (72%)
Поиск
РУС
0:34
24.04.2025



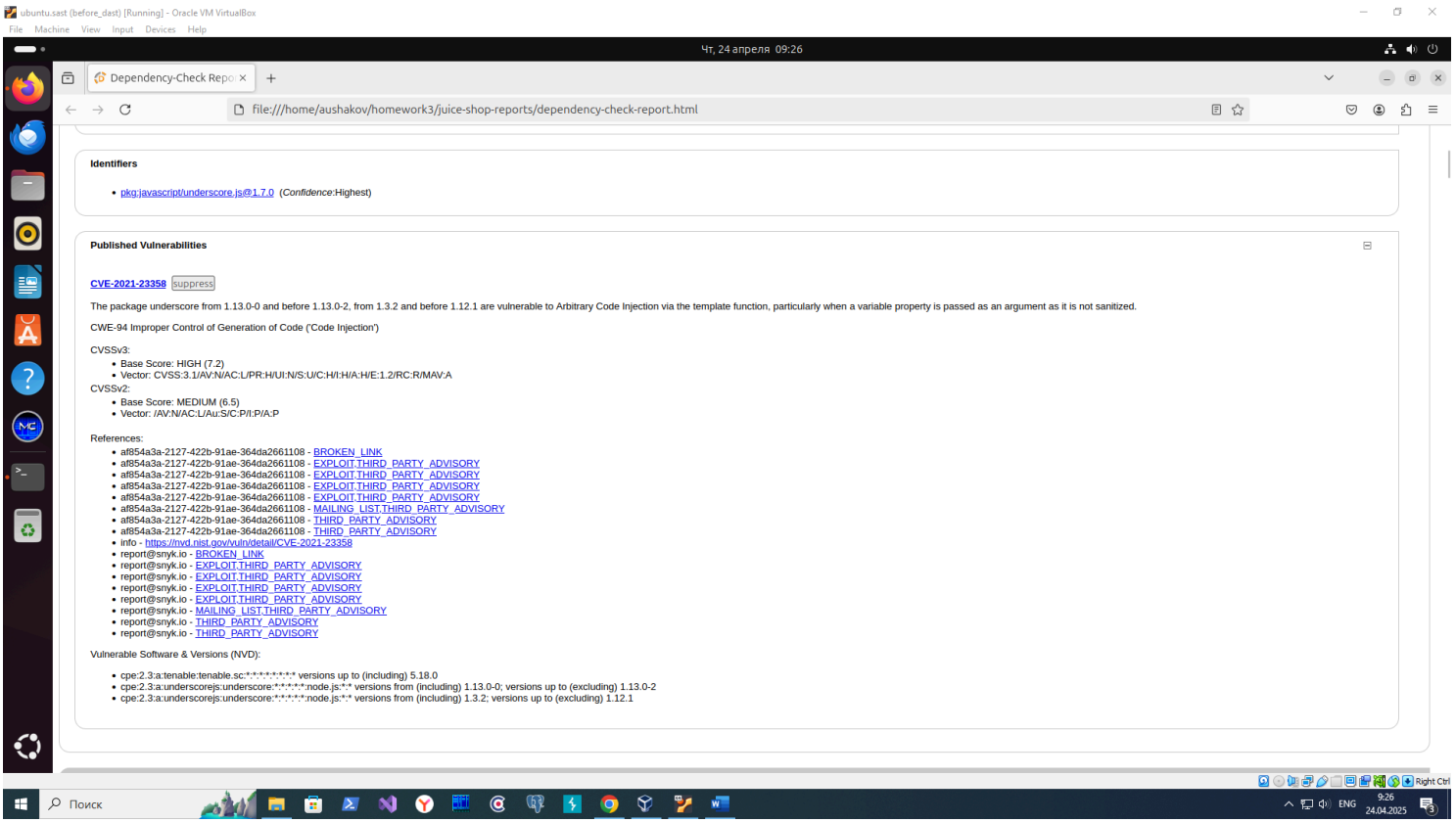
Результат сканирования Dependency-Check



Найденные уязвимости в bench.js

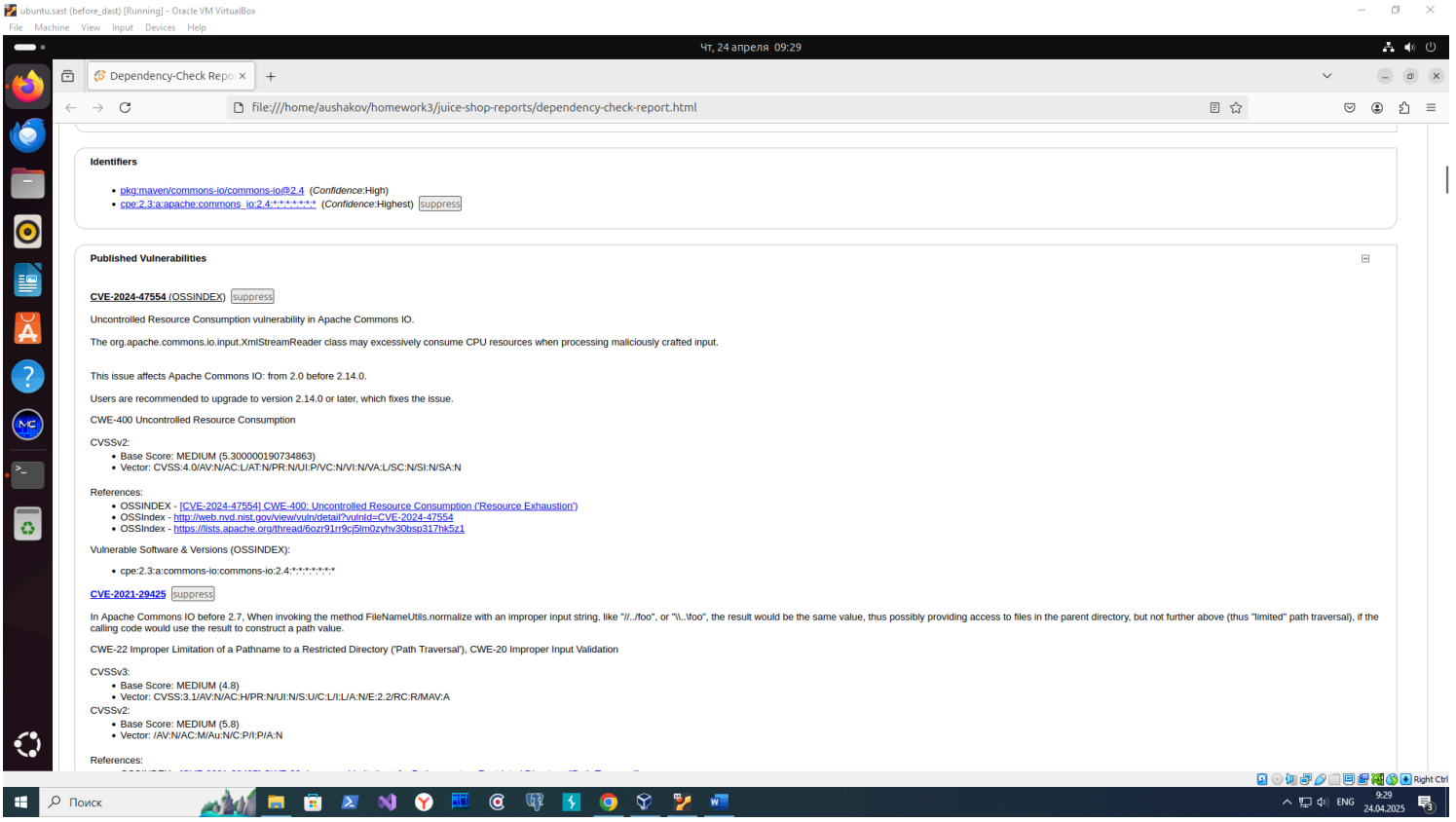
- CVE-2021-23358:

Base Score: HIGH (7.2)
Vector: CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H/E:1.2/RC:R/MAV:A

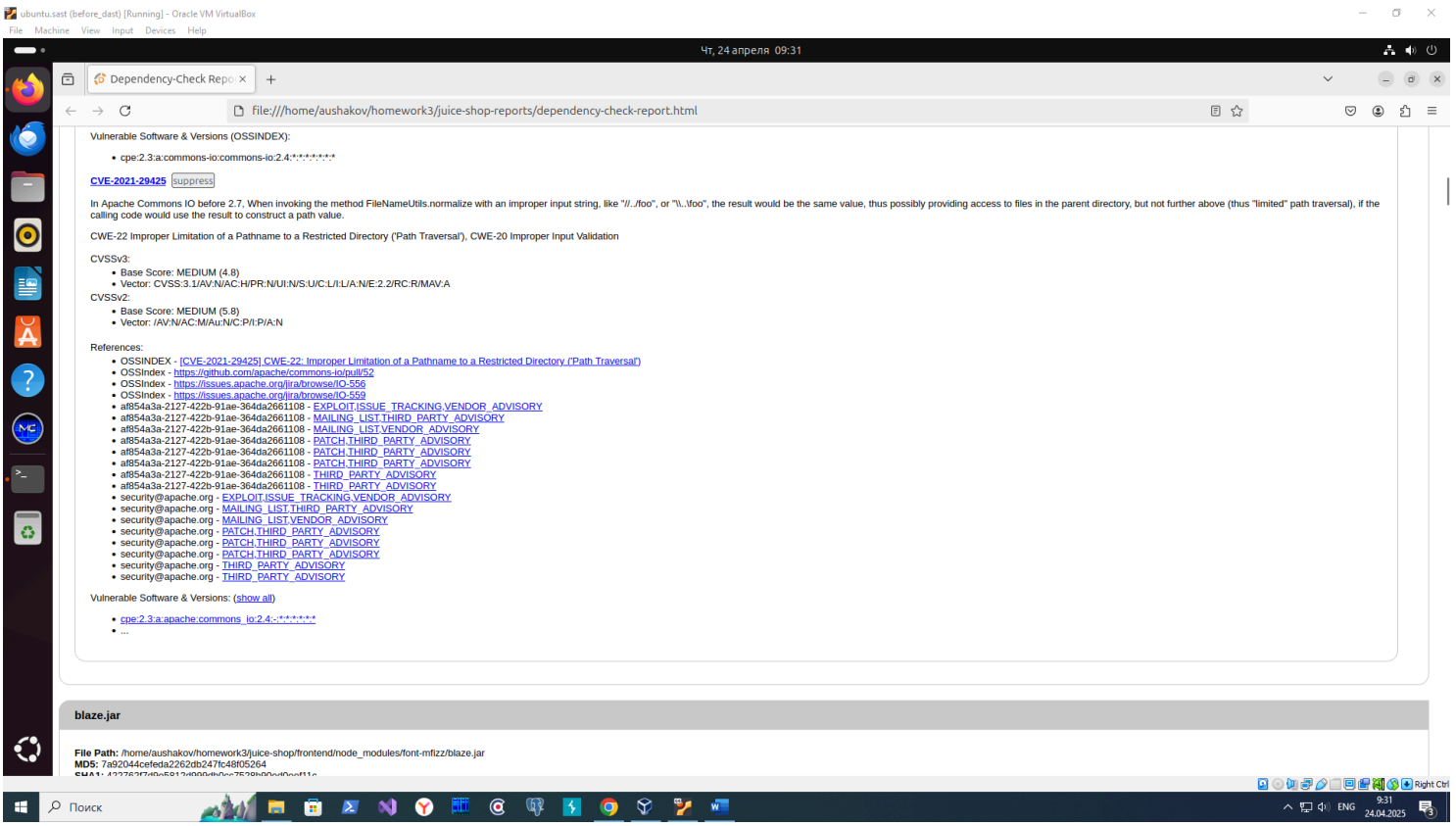


Найденные уязвимости в blaze.jar

- CVE-2024-47554:
  Base Score: MEDIUM (5.3)
  Vector: CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:P/VC:N/VI:N/VA:L/SC:N/SI:N/SA:N

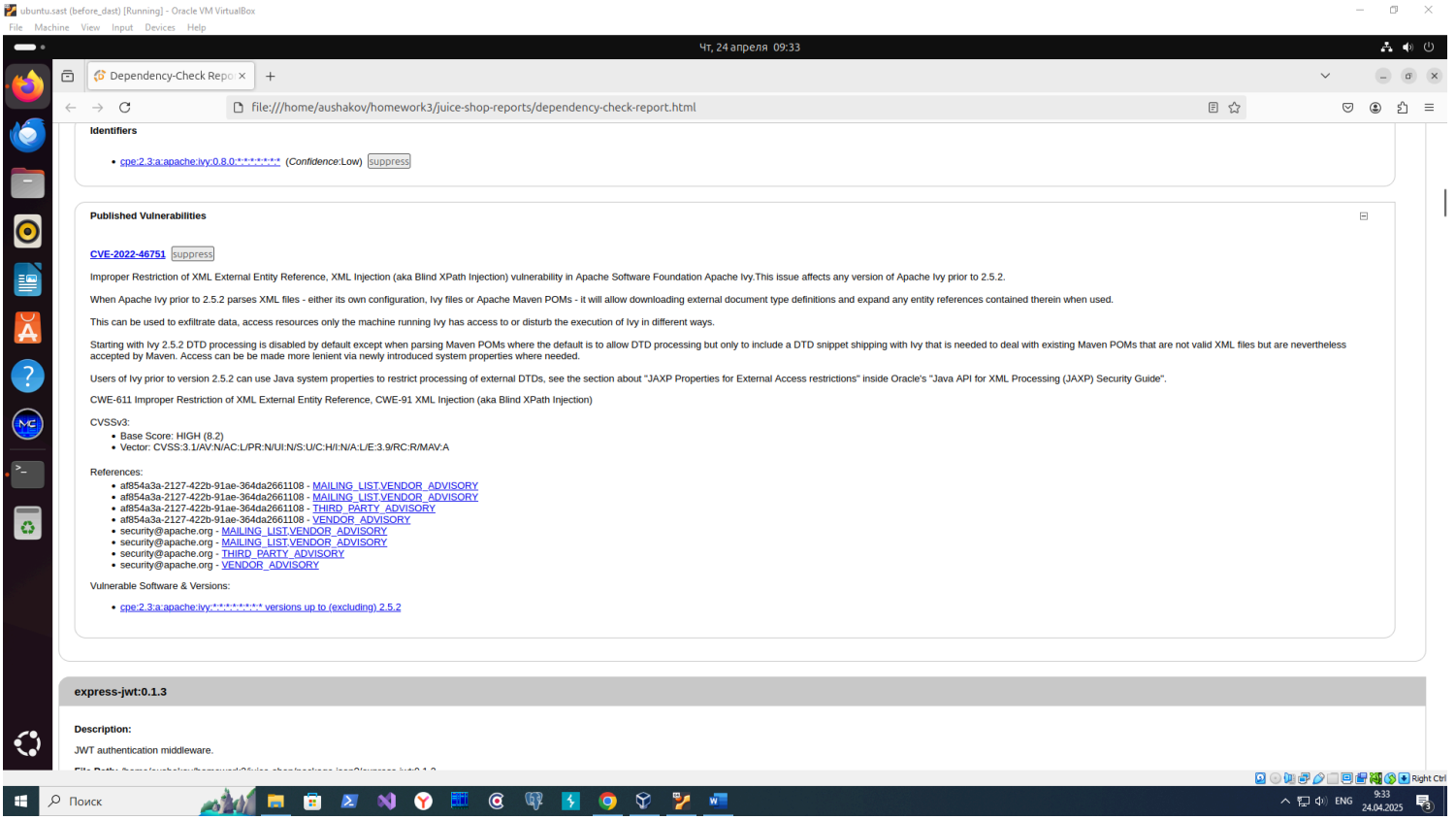


- CVE-2021-29425:
  Base Score: MEDIUM (4.8)
  Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:N/E:2.2/RC:R/MAV:A



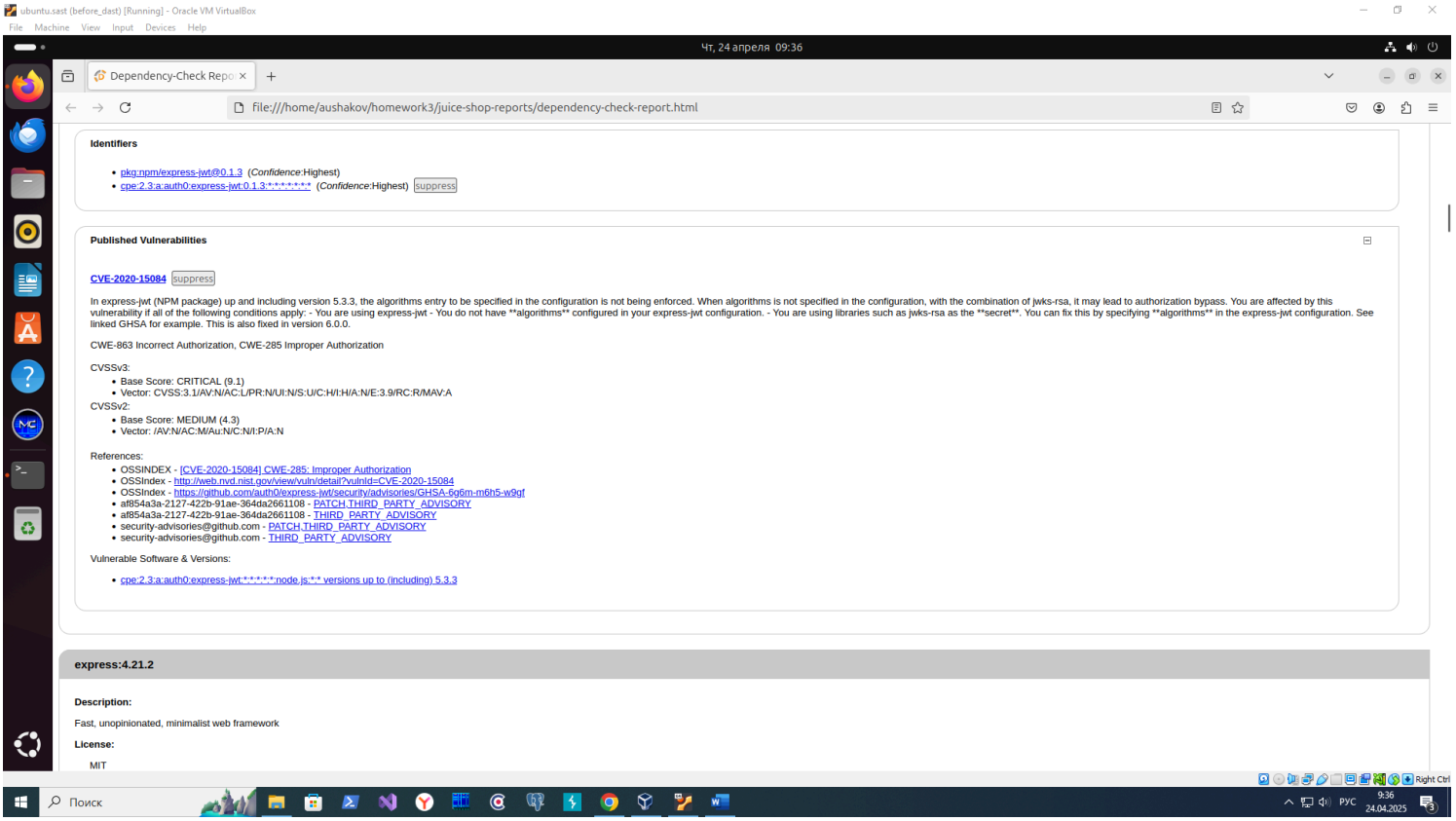
- CVE-2022-46751:

Base Score: HIGH (8.2)
Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:N/A:L/E:3.9/RC:R/MAV:A



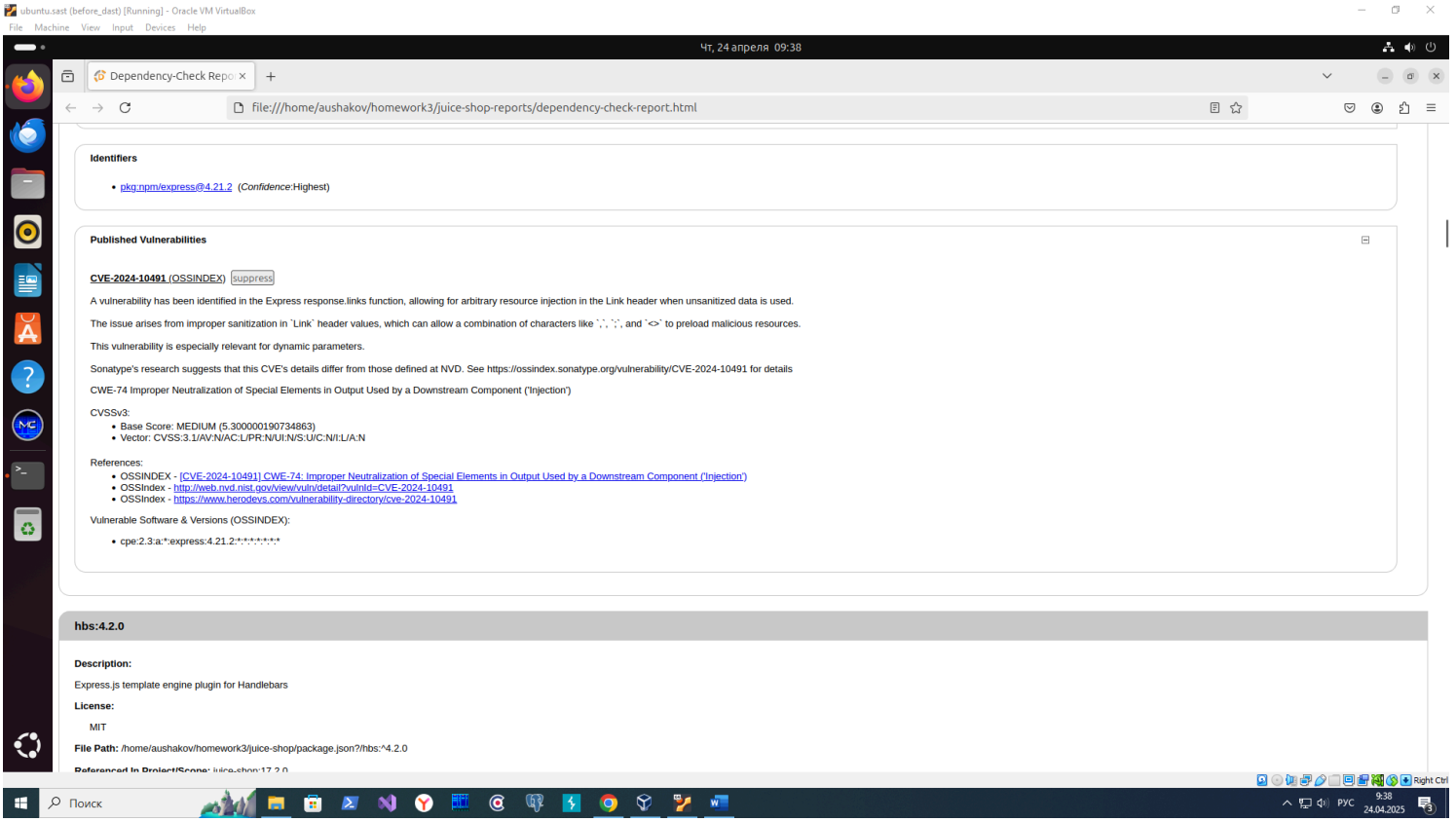
Найденные уязвимости в express-jwt:0.1.3

- CVE-2020-15084:
  Base Score: CRITICAL (9.1)
  Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:N/E:3.9/RC:R/MAV:A



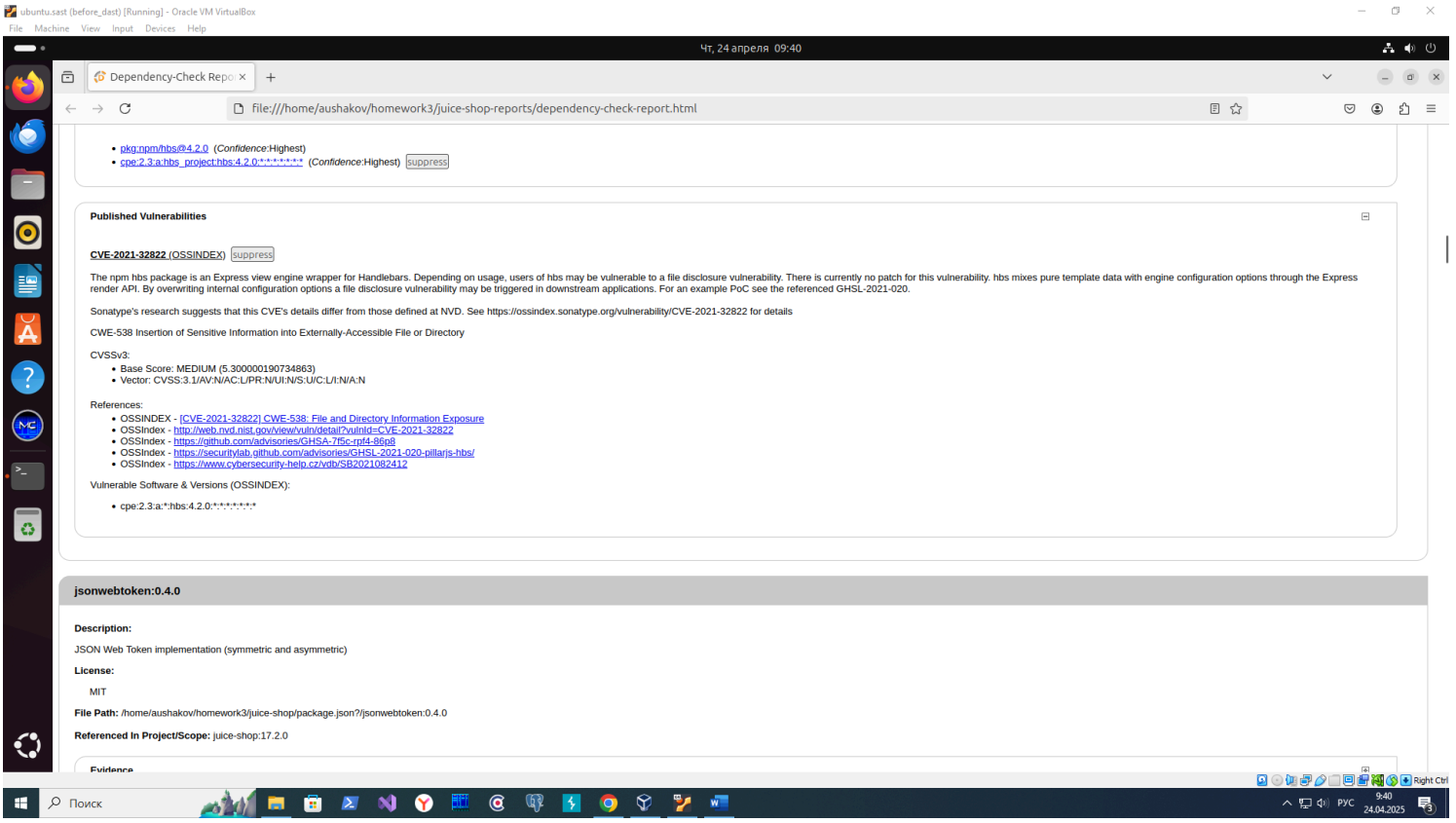
Найденные уязвимости в express:4.21.2

- CVE-2024-10491:
  Base Score: MEDIUM (5.3)
  Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N



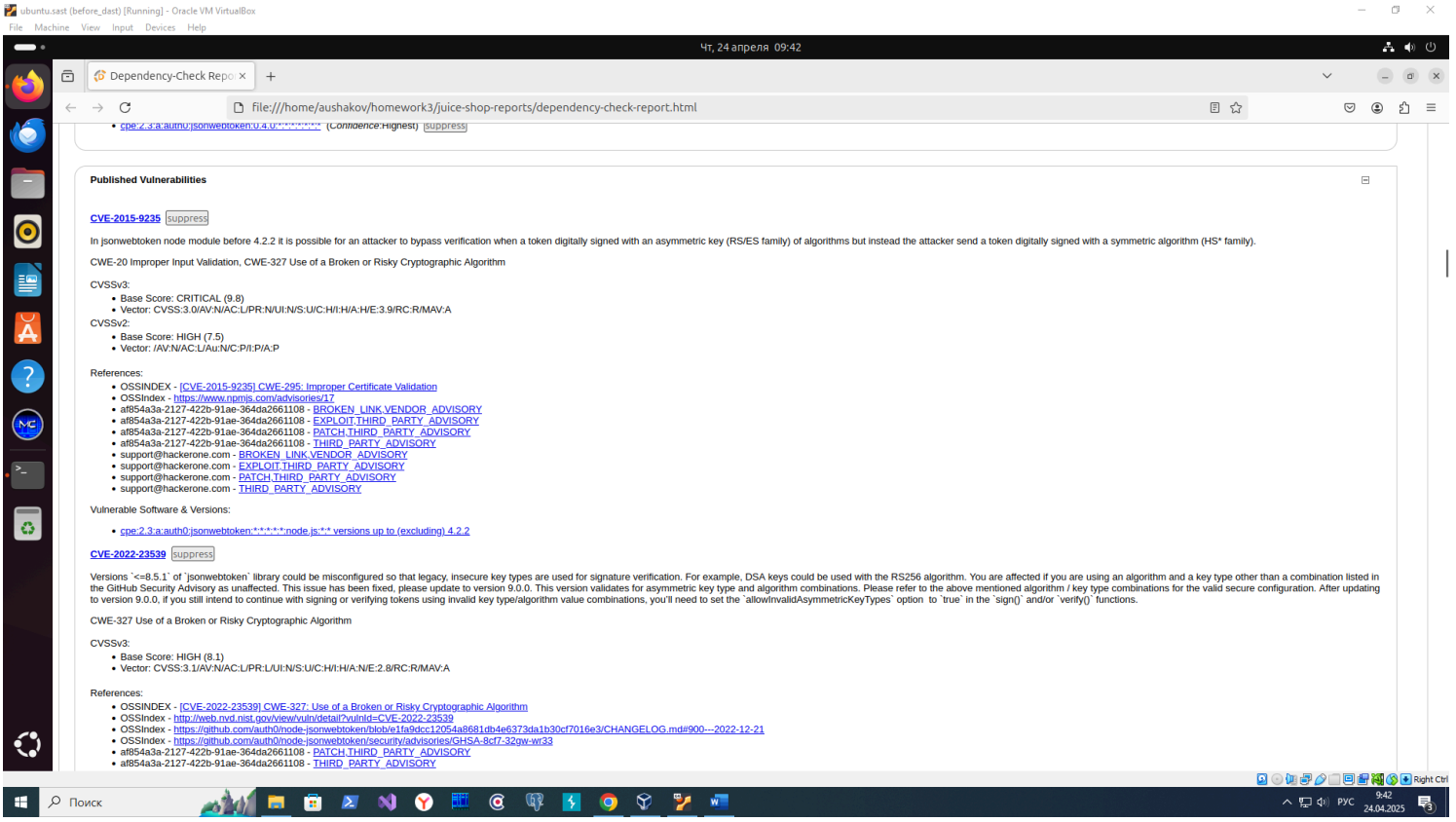
Найденные уязвимости в hbs:4.2.0

- CVE-2021-32822:
  Base Score: MEDIUM (5.3)
  Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N

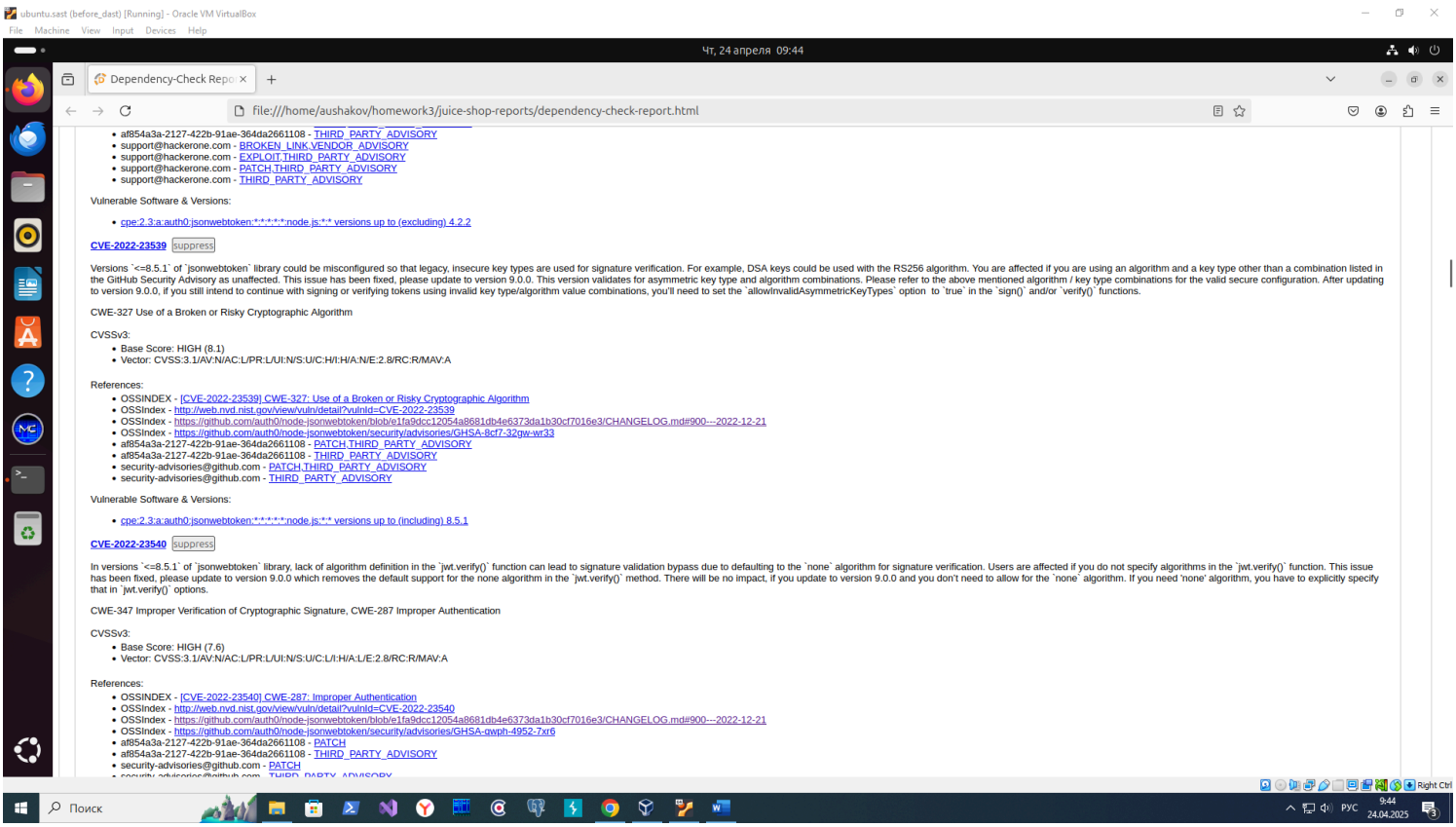


Найденные уязвимости в jsonwebtoken:0.4.0

- CVE-2015-9235:
  Base Score: CRITICAL (9.8)
  Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H/E:3.9/RC:R/MAV:A

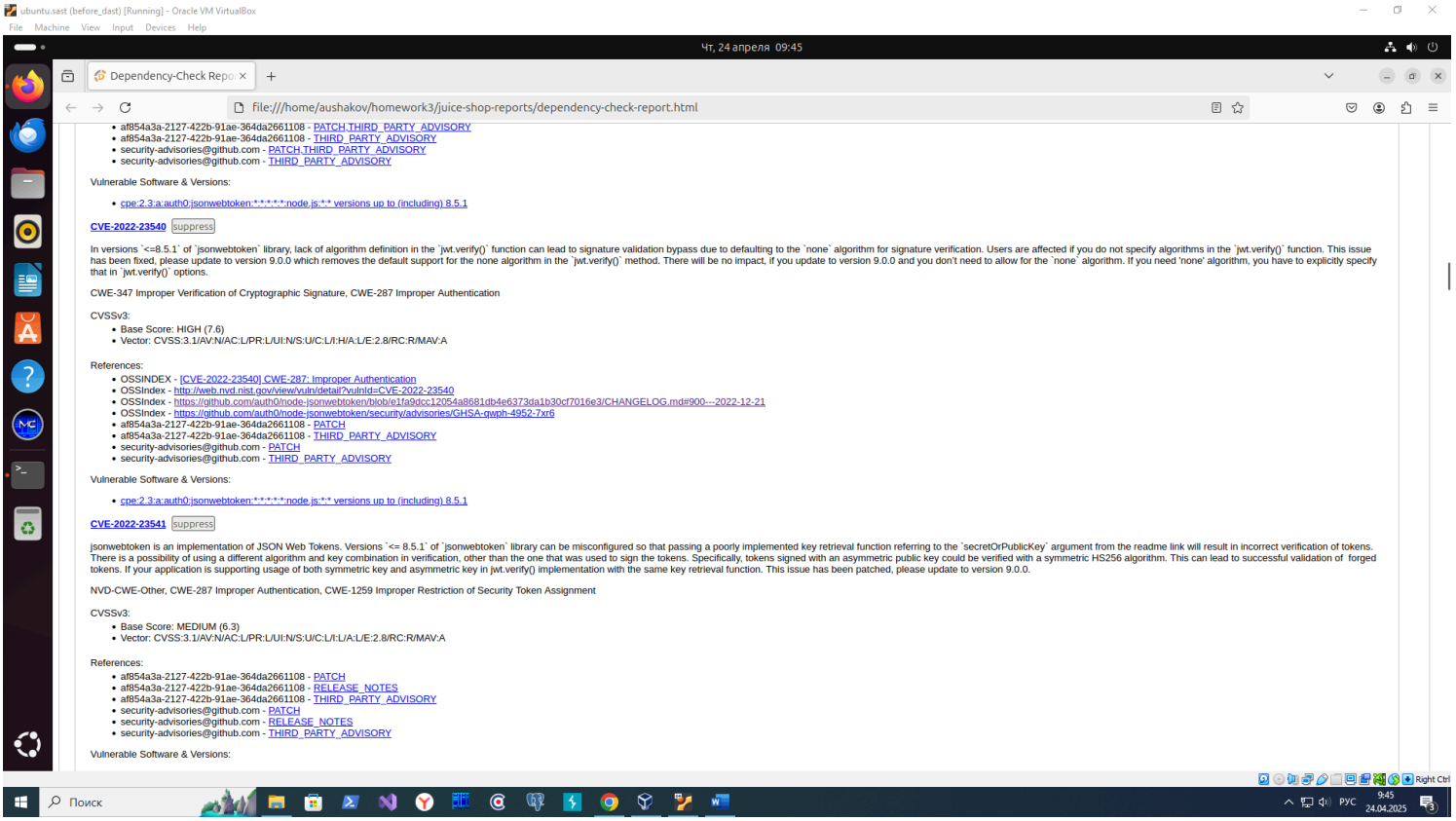


- CVE-2022-23539:
  Base Score: HIGH (8.1)
  Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:N/E:2.8/RC:R/MAV:A

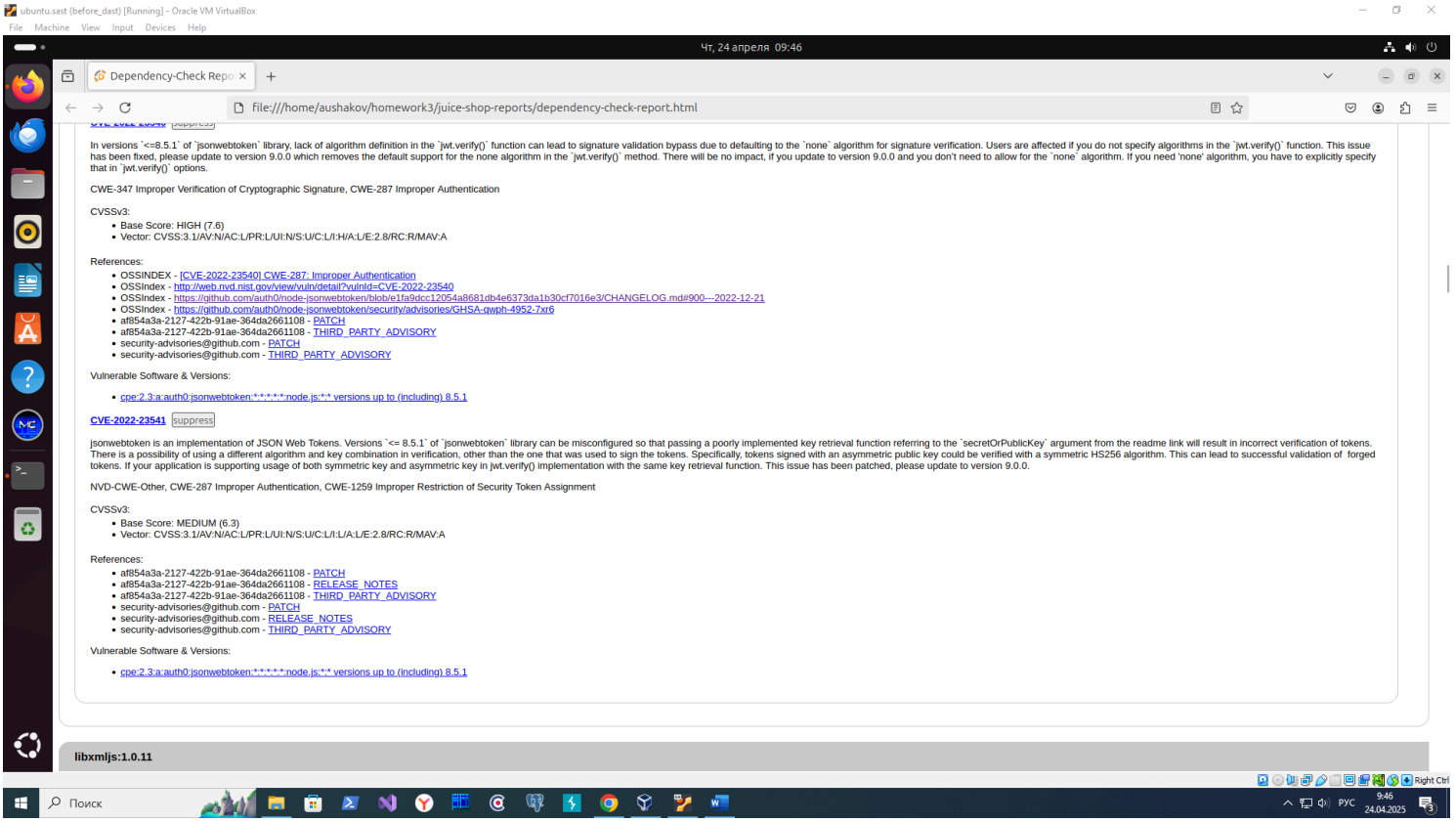


- CVE-2022-23540:

Base Score: HIGH (7.6)
Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:H/A:L/E:2.8/RC:R/MAV:A

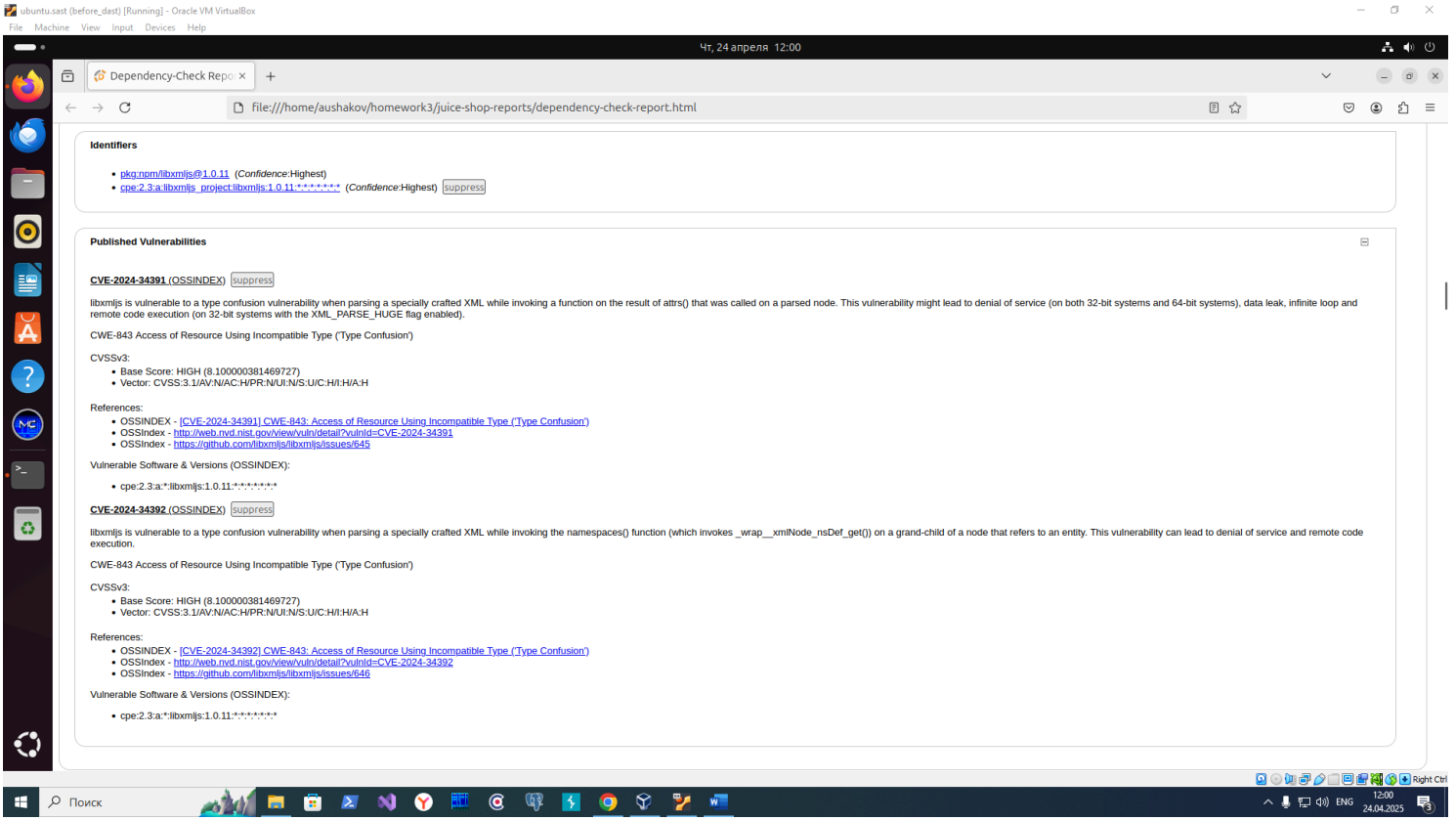


- CVE-2022-23541:
  Base Score: MEDIUM (6.3)
  Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:L/I:L/A:L/E:2.8/RC:R/MAV:A

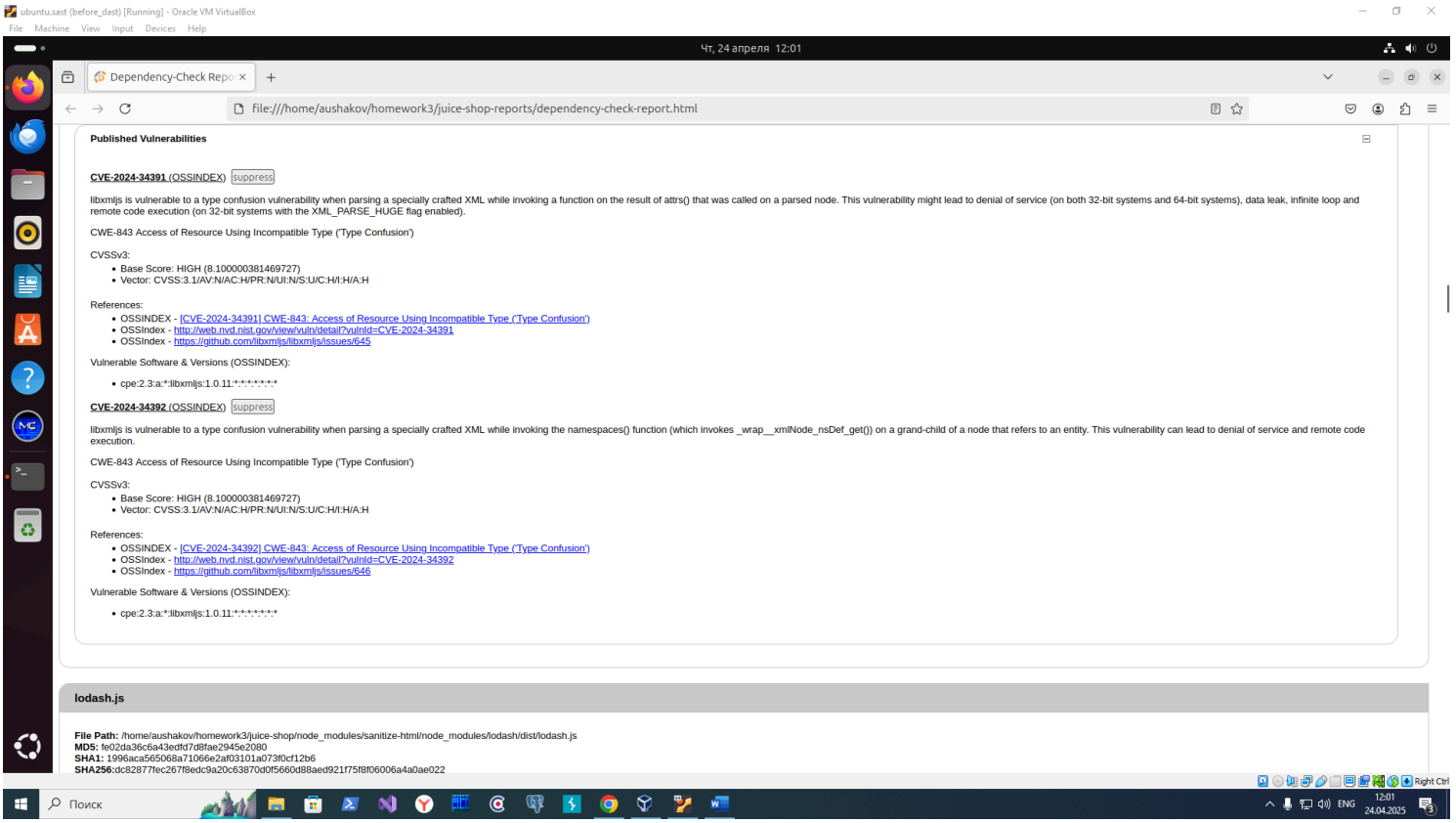


Найденные уязвимости в libxmljs:1.0.11

- CVE-2024-34391:
  Base Score: HIGH (8.1)
  Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

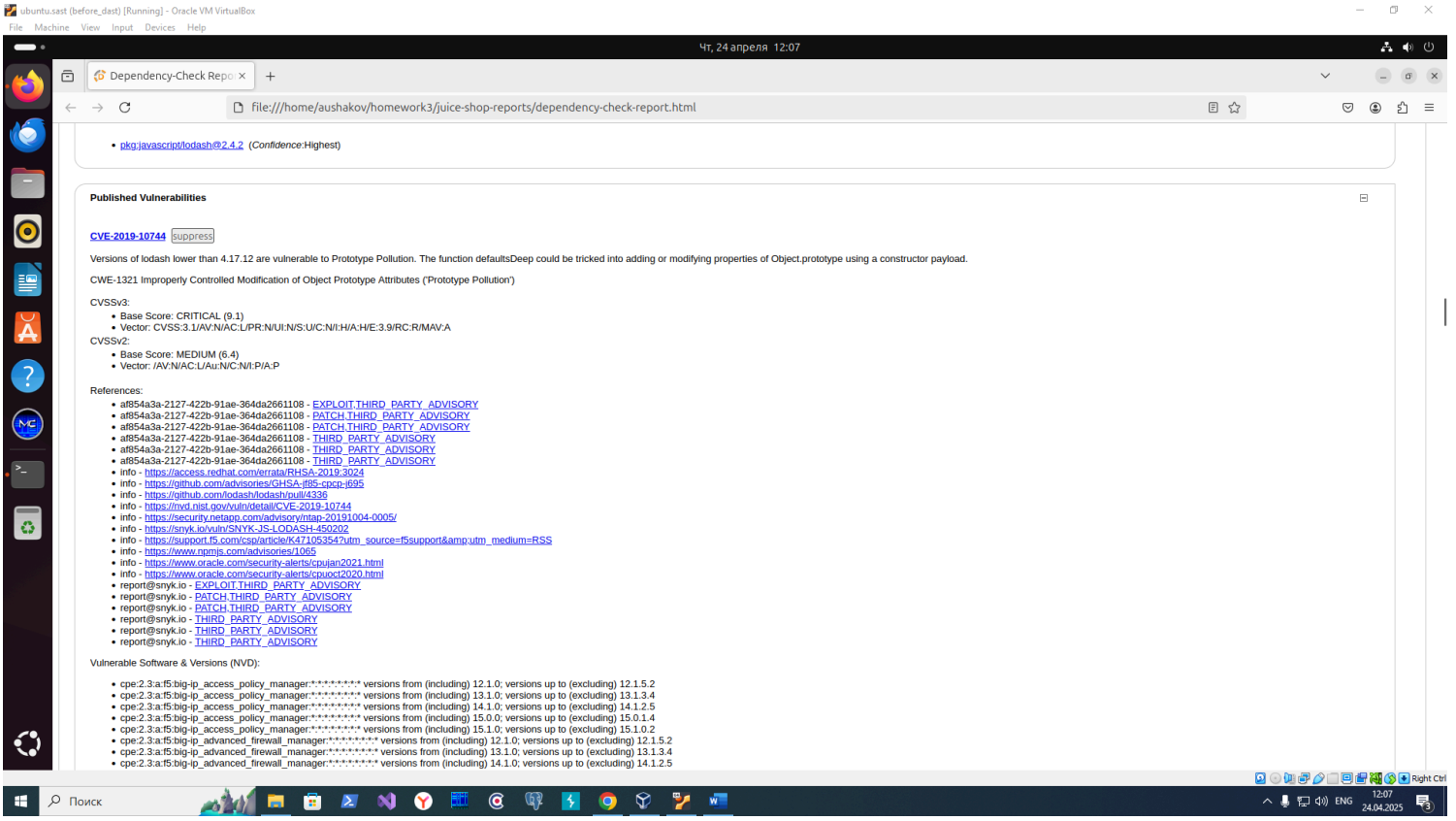


- CVE-2024-34392:
  Base Score: HIGH (8.1)
  Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:H/I:H/A:H

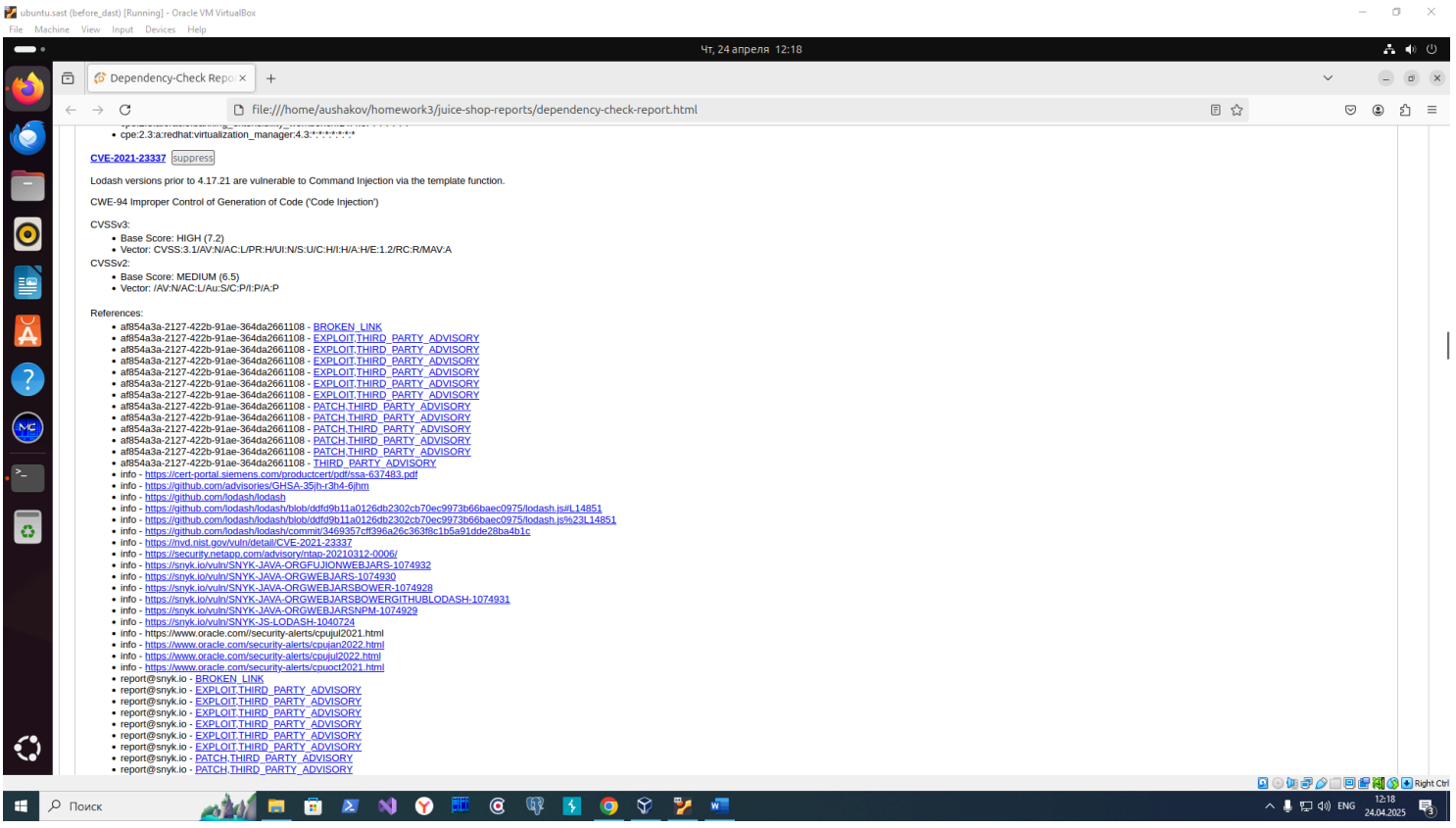


Найденные уязвимости в lodash.js

- CVE-2019-10744:
  Base Score: CRITICAL (9.1)
  Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:H/E:3.9/RC:R/MAV:A

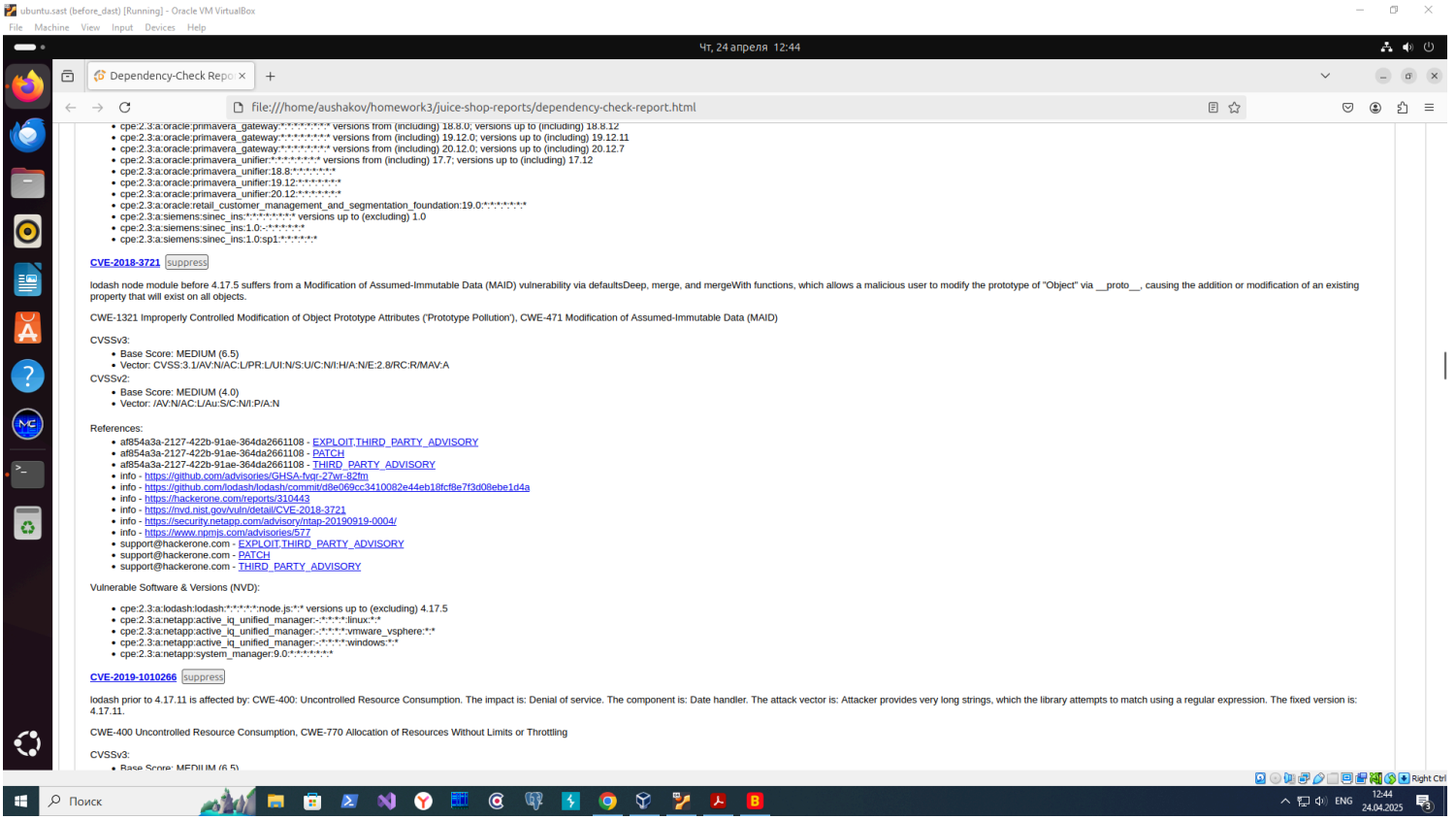


- CVE-2021-23337:
  Base Score: HIGH (7.2)
  Vector: CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:U/C:H/I:H/A:H/E:1.2/RC:R/MAV:A

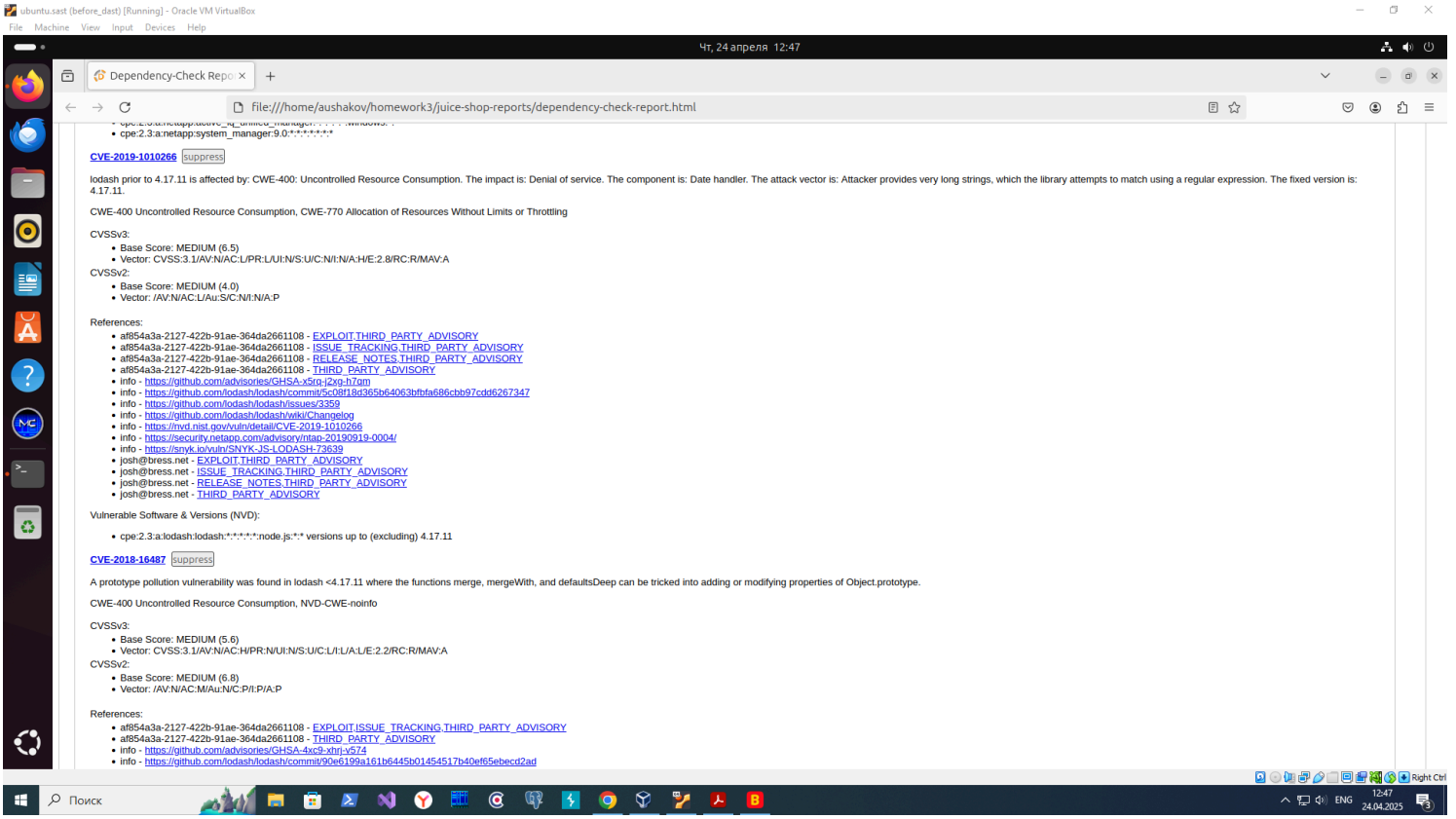


- CVE-2018-3721:

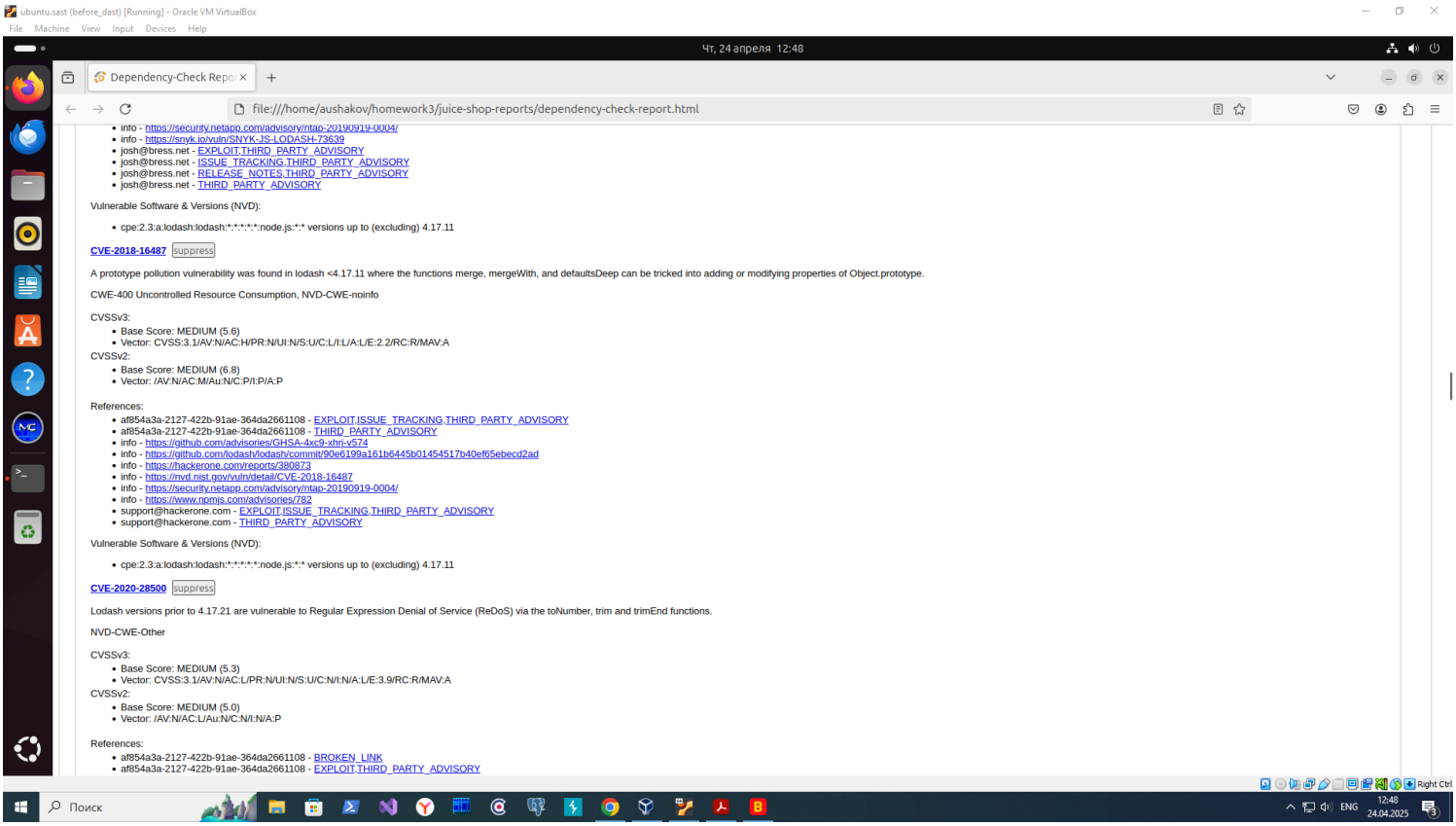
Base Score: MEDIUM (6.5)
Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:H/A:N/E:2.8/RC:R/MAV:A



- CVE-2019-1010266:
  Base Score: MEDIUM (6.5)
  Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H/E:2.8/RC:R/MAV:A

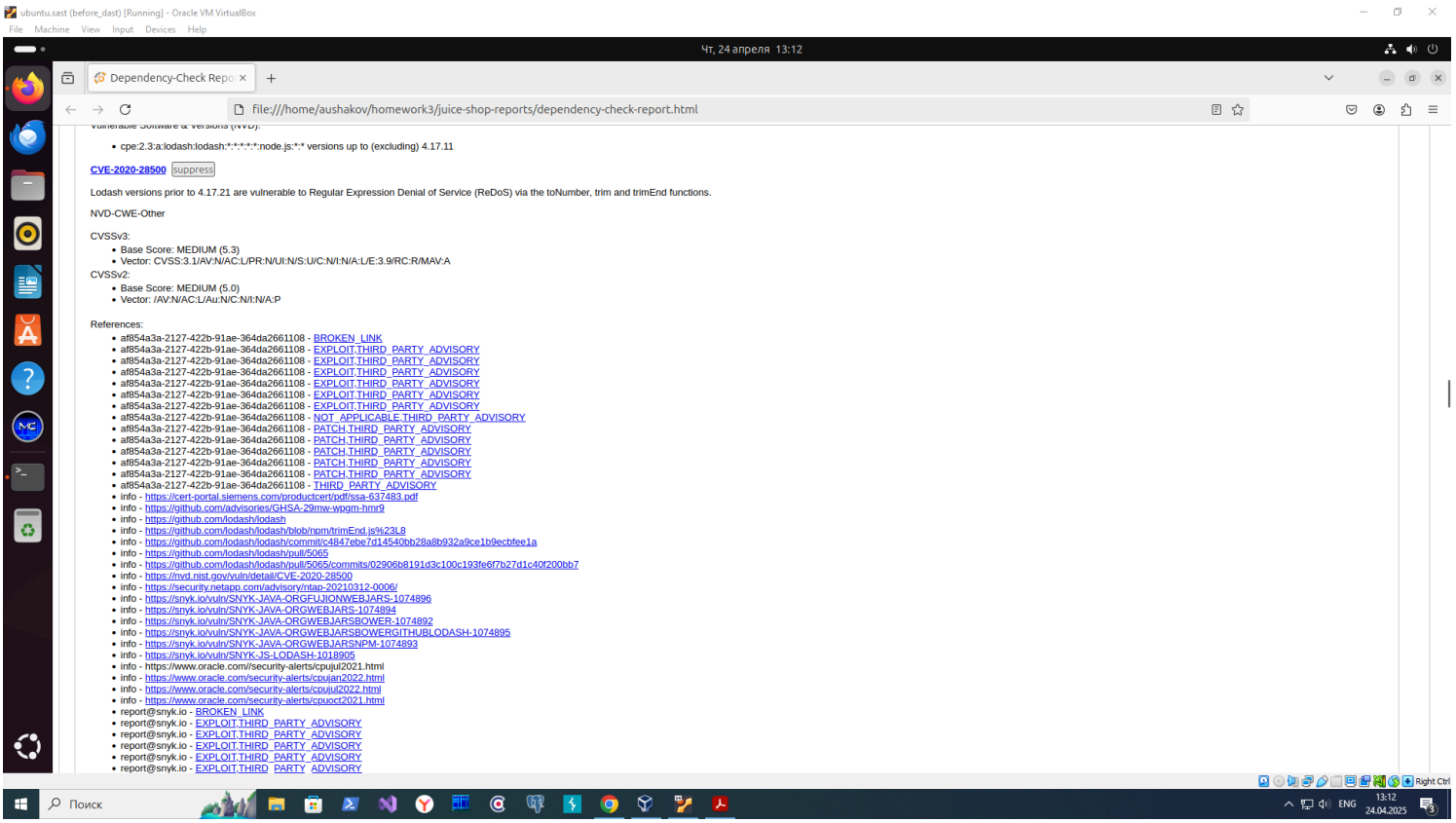


- CVE-2018-16487:
  Base Score: MEDIUM (5.6)
  Vector: CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:L/A:L/E:2.2/RC:R/MAV:A



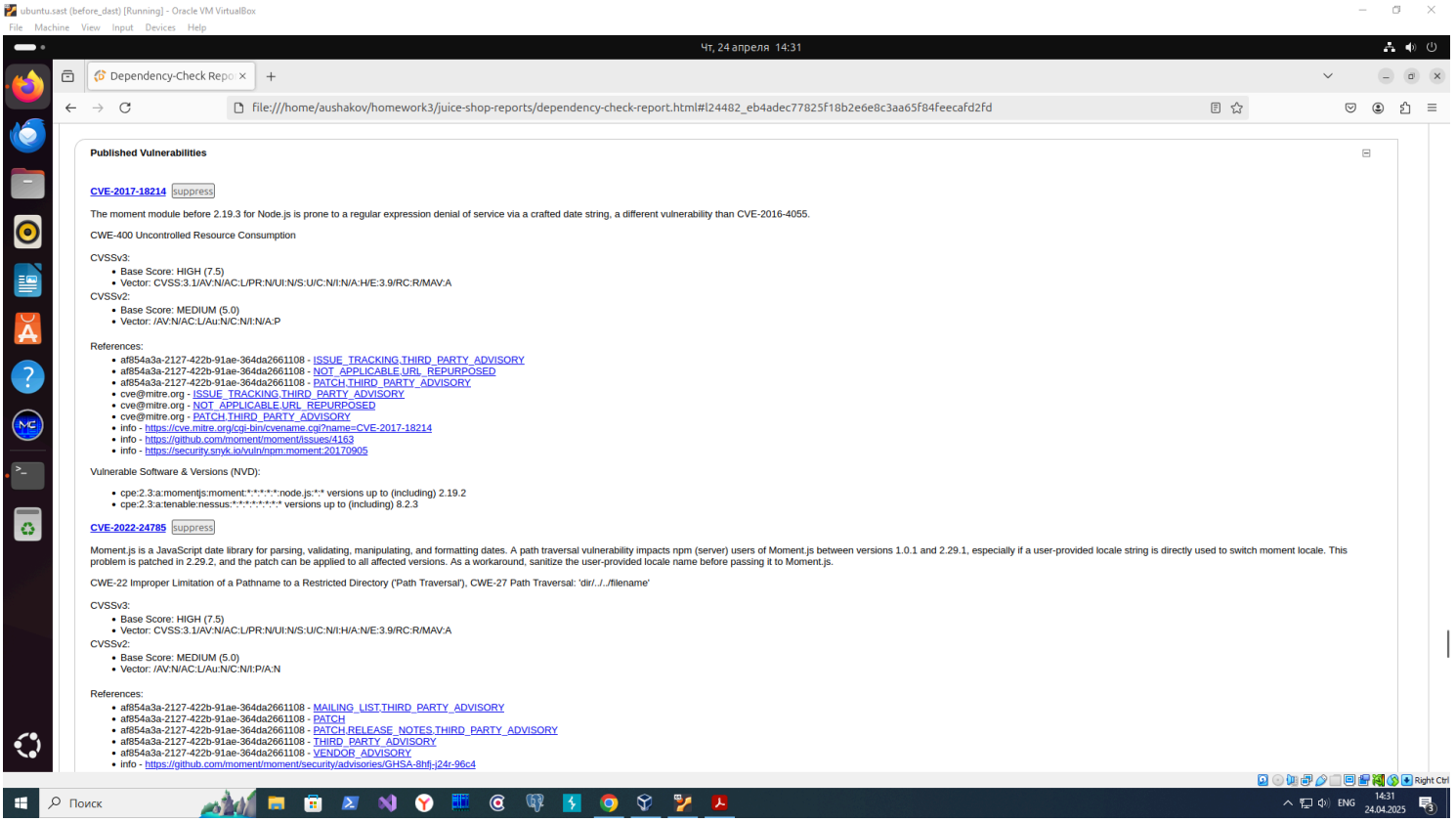
- CVE-2020-28500:

Base Score: MEDIUM (5.3)
Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:L/E:3.9/RC:R/MAV:A

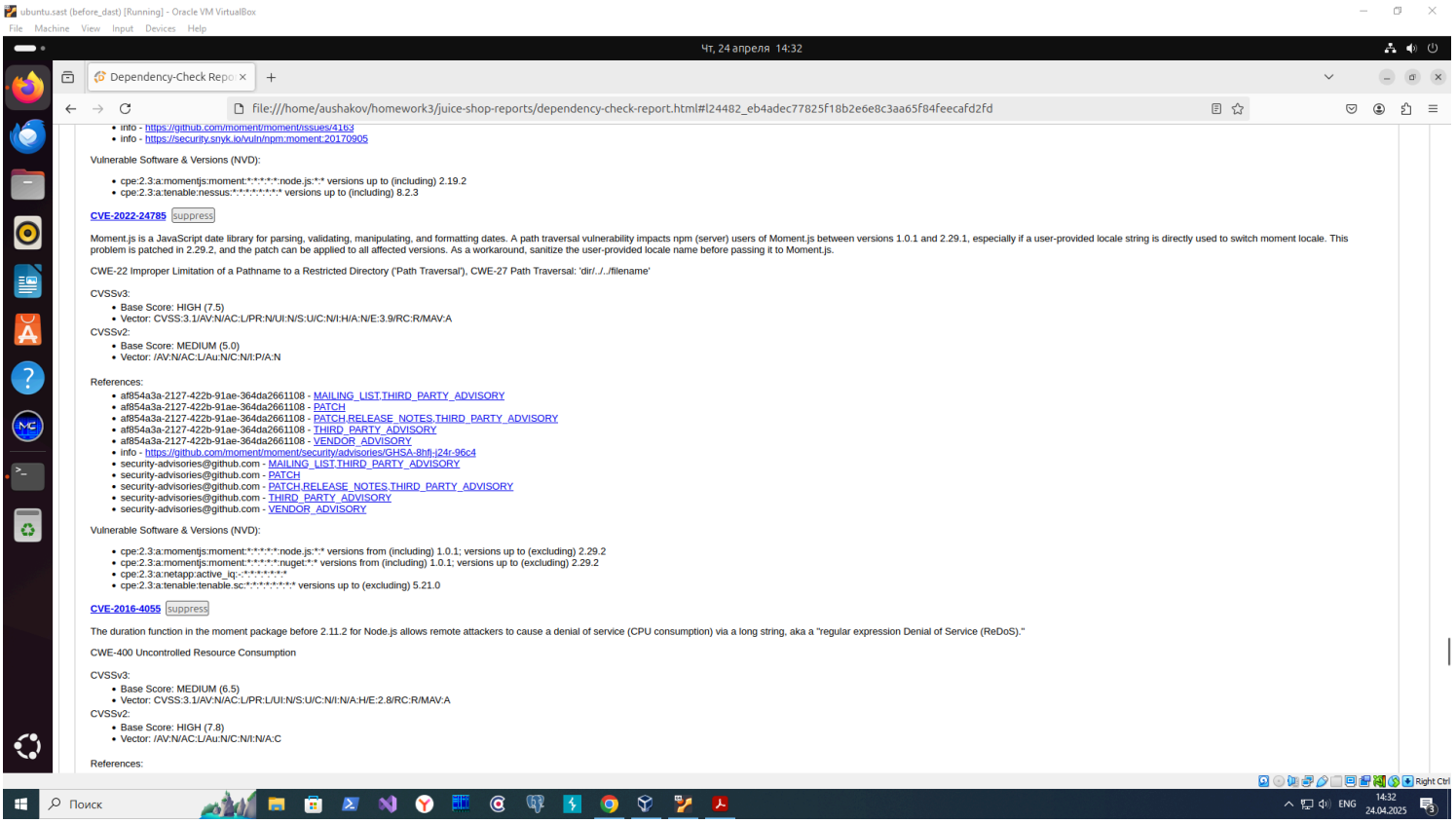


Найденные уязвимости в moment.js

- CVE-2017-18214:
  Base Score: HIGH (7.5)
  Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H/E:3.9/RC:R/MAV:A

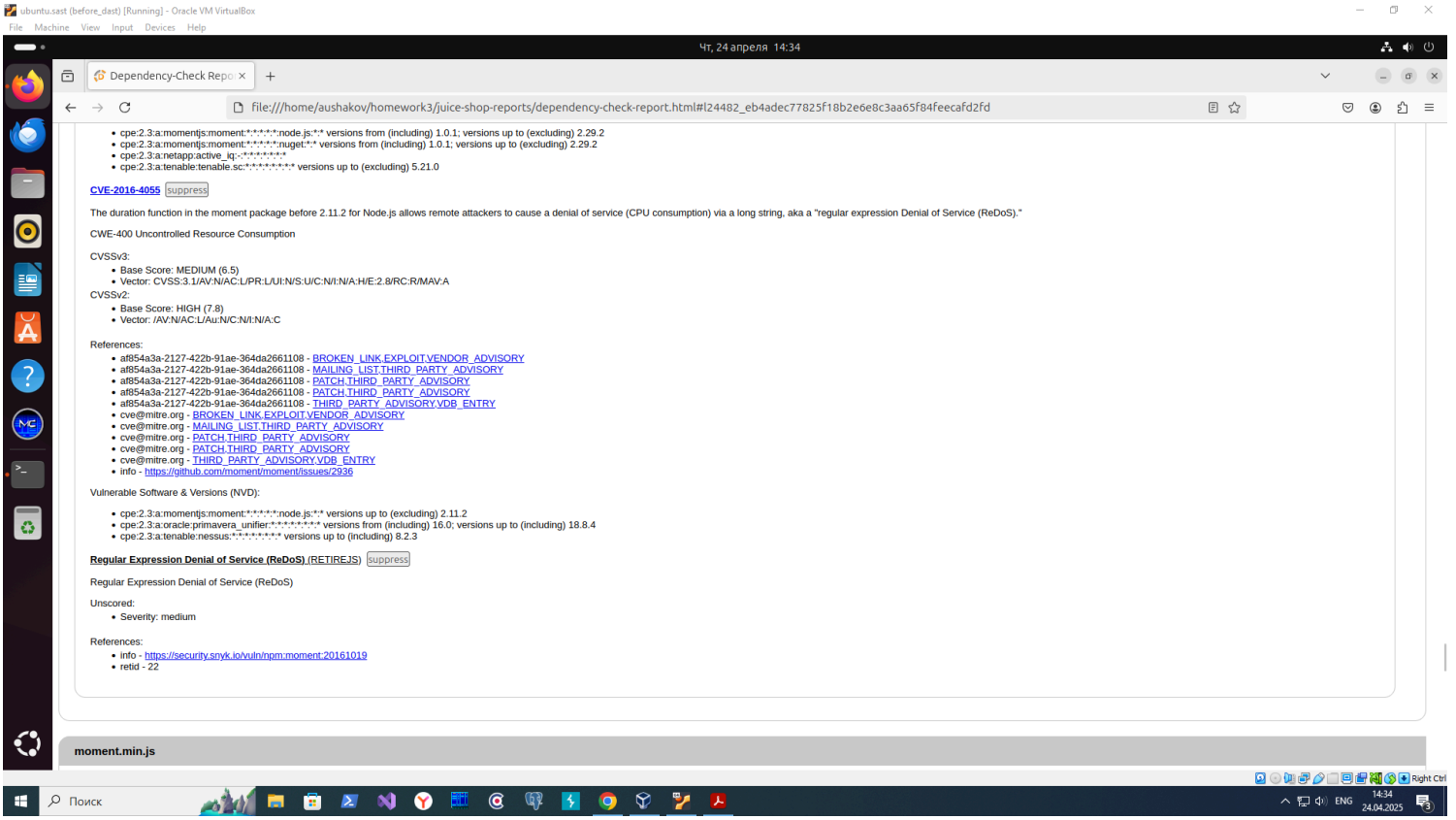


- CVE-2022-24785:
  Base Score: HIGH (7.5)
  Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:H/A:N/E:3.9/RC:R/MAV:A



- CVE-2016-4055:

Base Score: MEDIUM (6.5)
Vector: CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:N/A:H/E:2.8/RC:R/MAV:A



Выводы:

1. Найдено достаточно много уязвимостей в используемых библиотеках. Первый шаг, который можно сделать - это попробовать их обновить и запустить проверку с помощью Dependency-Check еще раз.
2. Вполне возможно, что обновление какой-либо библиотеки может сломать использующее ее приложение. Поэтому после обновления каждой библиотеки рекомендуется запускать автотесты (если они есть) и проводить процедуру ручного тестирования приложения, для обнаружения проблем, связанных с обновлением конкретно этой библиотеки.
3. Если после обновления какой-либо библиотеки до последней версии и последующей проверки с помощью Dependency-Check в ней были обнаружены уязвимости, которые критичны для приложения, то следует провести анализ: можем ли мы заменить данную библиотеку на другую, реализующую ту же самую функциональность или можем ли мы как-то с помощью написания дополнительного кода замаскировать эту уязвимость (с помощью паттерна проектирования “адаптер”, например).