Автономная некоммерческая организация
профессиональная образовательная организация
«УНИВЕРСИТЕТСКИЙ КОЛЛЕДЖ БРИКС»

Специальность 09.02.07 Информационные системы и программирование

ТЕХНИЧЕСКАЯ ДОКУМЕНТАЦИЯ

к программному продукту

«Автоматизированная система инвентаризации ПО и оценки критичности программных угроз «Vulnerability Scanner»»

Выполнил студент учебной группы
ОИСП-292 очной формы обучения
Кадырканова Аруuке Кадыркановна

Москва – 2026 год

Содержание

Пояснительная записка

Цель и задачи проекта

Основной целью разработки является создание программного комплекса для автоматизированного аудита безопасности рабочих станций, позволяющего сопоставлять установленное программное обеспечение с известными мировыми угрозами. В рамках достижения поставленной цели был реализован ряд задач, включающих организацию прямого доступа к системному реестру Windows для инвентаризации ПО, проектирование реляционной структуры базы данных на платформе PostgreSQL и разработку механизмов асинхронного взаимодействия с внешним API сервиса NVD. Проект направлен на минимизацию рисков информационной безопасности путем визуализации критических уязвимостей и автоматизации процесса поиска актуальных патчей для системного администратора.

Исходные требования и функциональность

Система спроектирована для обработки массивов данных об установленных приложениях, версиях и производителях, которые извлекаются из системных путей реестра HKLM и HKCU. Основной функционал распределен между несколькими программными модулями, обеспечивающими сканирование локальной системы, выполнение сетевых запросов к базе CVE и формирование аналитического дашборда с цветовой маркировкой рисков по шкале CVSS. Программный код объемом более 300 строк реализует полный цикл управления данными, включая операции добавления записей, их фильтрации в режиме реального времени и безопасной очистки таблиц. В качестве выходных данных пользователь получает структурированный отчет, содержащий идентификаторы угроз и их детальные описания, интегрированные в графический интерфейс.

Архитектура и методы программирования

Архитектура приложения реализована в рамках единого исполнимого модуля с четким разделением зон ответственности между функциональными блоками на основе объектно-ориентированного подхода. Код подразделяется на компоненты инициализации интерфейса, модули системного взаимодействия, функции управления базой данных и специализированный класс для асинхронных операций. Применение событийно-ориентированной модели библиотеки PyQt6 позволило эффективно организовать взаимодействие между визуальными элементами и бизнес-логикой. Инструментарий разработки включает среду VS Code и специализированные библиотеки psycopg2 для взаимодействия с СУБД, а также winreg для низкоуровневой работы с операционной системой.

Описание программных компонентов (Руководство программиста)

В начале программы выполняется загрузка графического шаблона через метод uic.loadUiType, что связывает визуальные элементы файла конфигурации интерфейса с логикой Python. Объекты win, windows и form обеспечивают управление жизненным циклом приложения и доступ к виджетам. Переменная worker используется как глобальный указатель на текущий поток обновления, предотвращая конфликты при повторных запусках запросов.

Функция get_installed_software является инструментом сбора данных через алгоритм обхода веток реестра Windows. Внутри функции используются переменные paths для хранения корневых разделов и software_list для накопления уникальных записей. Логика включает извлечение параметров DisplayName, DisplayVersion и Publisher. Функция action_scan_pc выступает в роли диспетчера, инициирующего сканирование, обновление таблицы table_software и сохранение данных в базу.

Для предотвращения блокировки интерфейса реализован класс UpdateWorker, наследующий QThread. Внутри класса переопределен метод run, выполняющий последовательный опрос API NVD для каждой программы. Класс использует механизм сигналов progress и finished для передачи статуса обработки в основной поток. Внутренняя логика включает парсинг JSON-ответов, расчет метрик CVSS и выполнение транзакций к базе данных.

Группа функций load_dashboard_data и load_vulnerabilities_to_table отвечает за извлечение информации из PostgreSQL и её отображение. В них реализованы алгоритмы сортировки по уровню угрозы и динамическое изменение фона ячеек через QColor. Функции save_software_to_db и fetch_and_save_cve инкапсулируют SQL-запросы вставки. Инструментарий поиска реализован через функцию filter_table, осуществляющую поиск по вхождениям во всех колонках.

Связующим звеном выступает блок регистрации событий. Методы connect привязывают действия пользователя, такие как клики по кнопкам или выбор строк (метод show_details), к программным функциям. Утилиты clear_software_db и clear_cve_db обеспечивают удаление информации с выводом диалоговых окон QMessageBox, гарантируя безопасное управление состоянием базы данных.

Руководство пользователя

Установка и запуск

Для подготовки среды к работе необходимо загрузить исходный код проекта из репозитория. Важнейшим этапом является инсталляция внешних зависимостей, так как стандартная библиотека Python не содержит всех необходимых инструментов для работы данного приложения. Требуется установить пакет PyQt6 для функционирования графического интерфейса, библиотеку psycopg2-binary для обеспечения соединения с базой данных PostgreSQL, а также модуль requests для выполнения сетевых запросов к

внешним API. Установка производится через терминал с помощью команды pip install название_пакета.

После подготовки окружения следует убедиться в наличии развернутого сервера PostgreSQL и актуального файла настроек с параметрами подключения и API-ключом. Запуск системы осуществляется через консоль командой python main.py, которая инициализирует главное окно MainWindow. Приложение разработано для операционной системы Windows, что позволяет корректно использовать модули взаимодействия с системным реестром. Все накопленные в процессе работы данные сохраняются локально, поэтому процедура регистрации учетных записей в системе не предусмотрена.

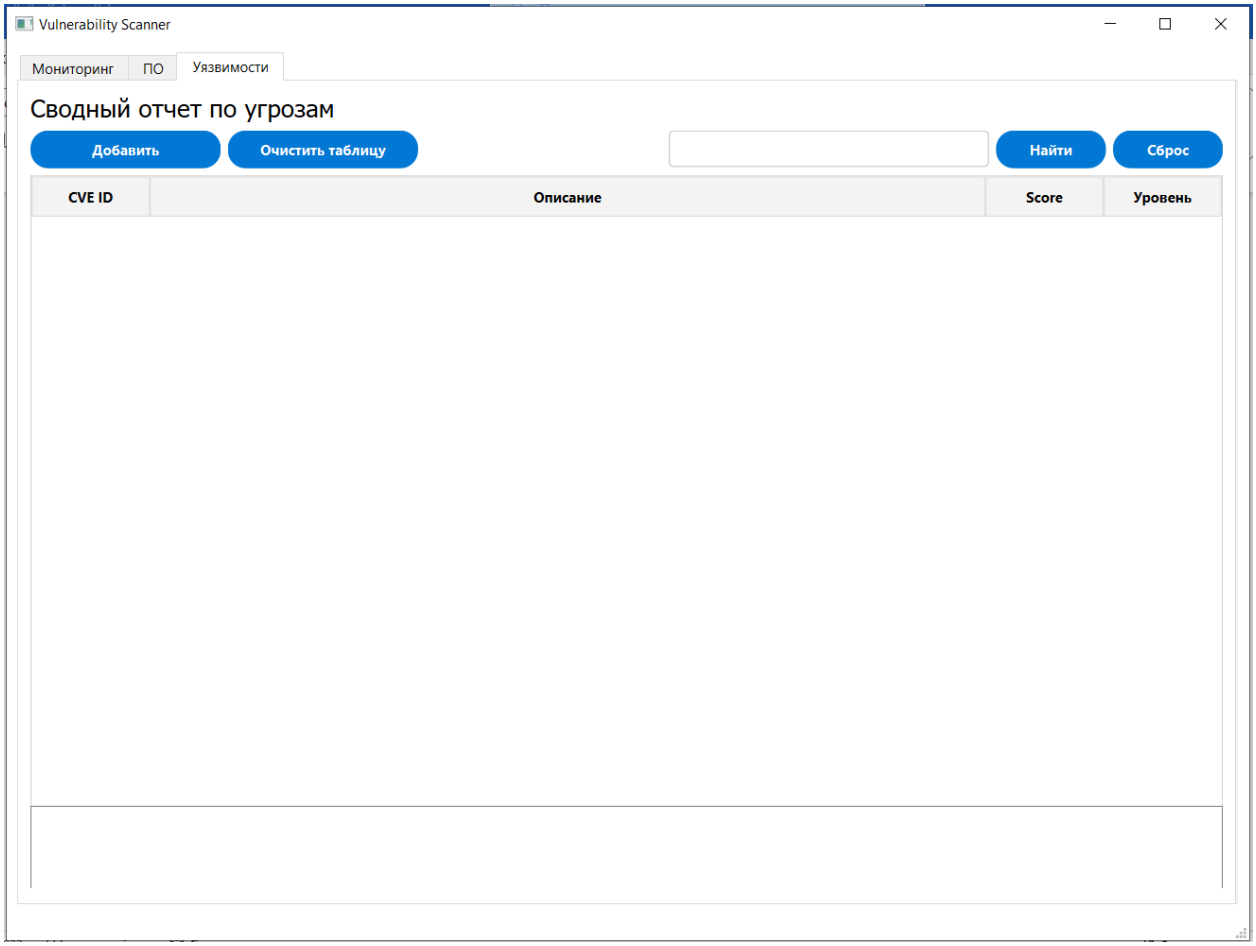
Интерфейс и навигация

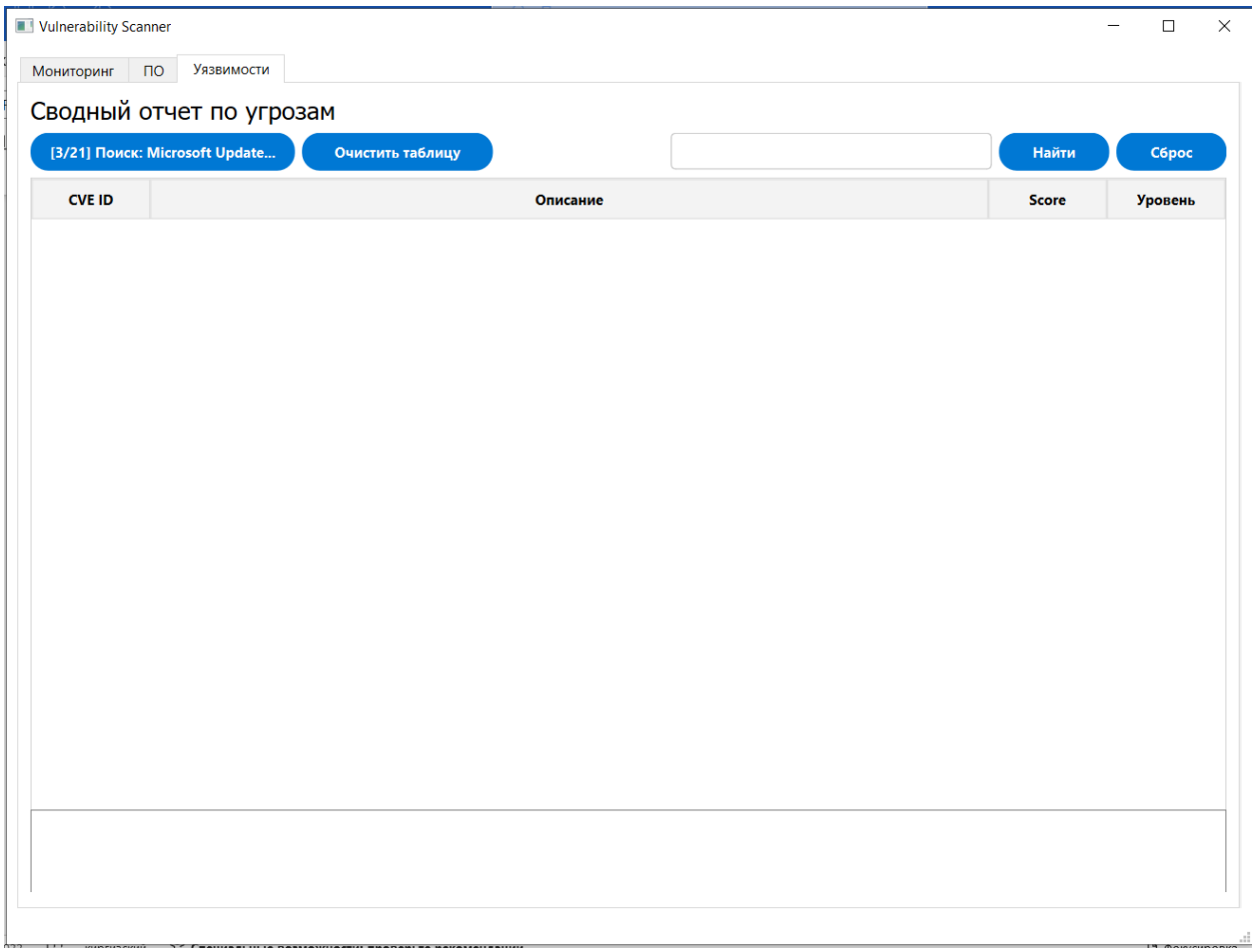
Графическое окно разделено на три специализированные вкладки: **«Мониторинг»**, **«ПО»** и **«Уязвимости»**. Каждая из них отвечает за отдельный этап анализа безопасности. Навигация осуществляется переключением между этими разделами в верхней части экрана. Каждая вкладка содержит заголовок «Сводный отчет по угрозам», интерактивные таблицы, а также интуитивно понятные элементы управления, такие как кнопки запуска процессов и поля для поиска информации. В нижней части вкладки «Уязвимости» расположена область, предназначенная для вывода подробных технических данных о выбранных объектах.

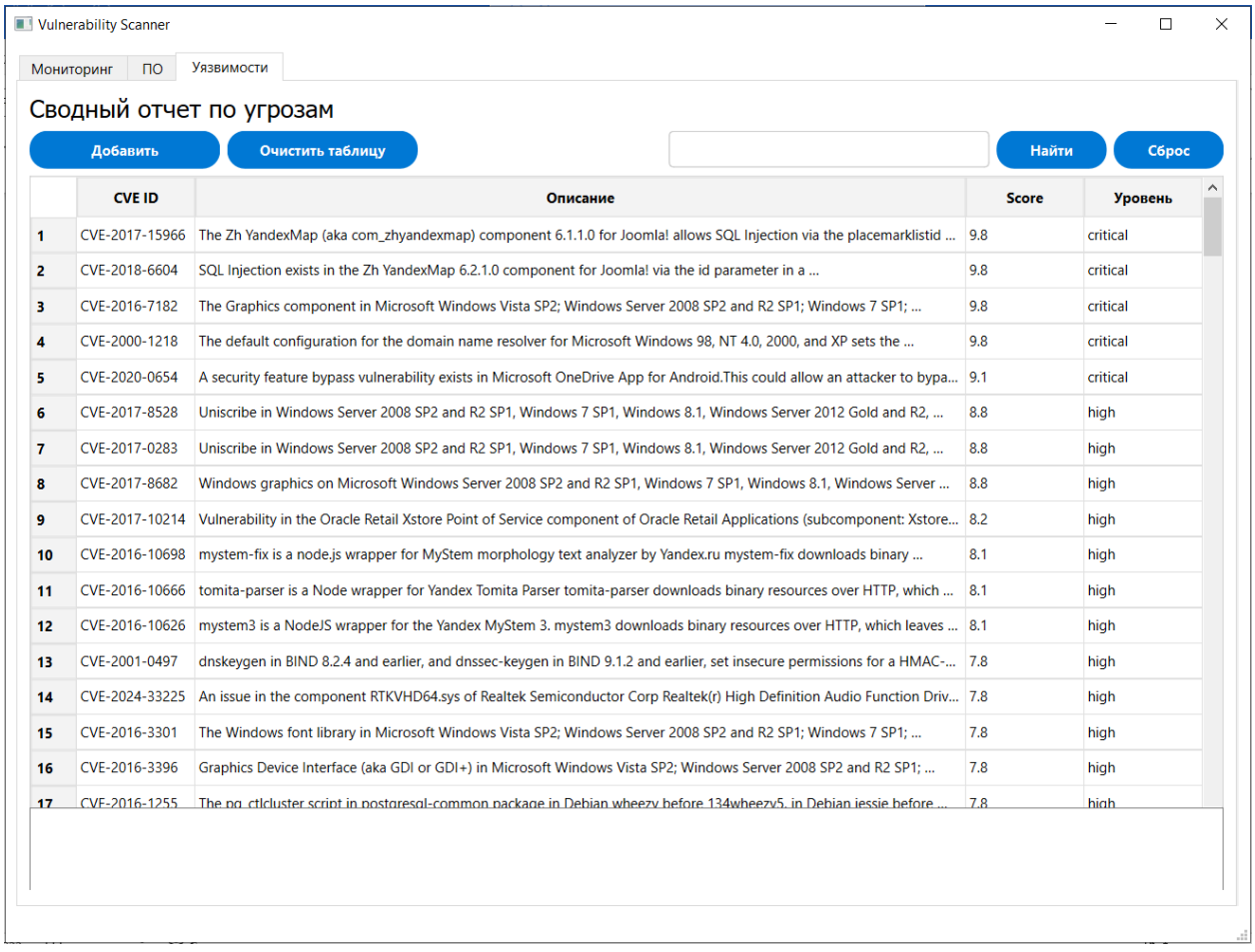
Основные функции

Программный комплекс обеспечивает автоматизированный сбор сведений об установленных приложениях напрямую из реестра Windows. Во вкладке **«ПО»** нажатие кнопки **«Сканировать»** заполняет таблицу названиями, версиями и именами издателей программ.

Полученные данные сопоставляются с информацией из глобальных баз через кнопку **«Добавить»** во вкладке «Уязвимости», которая инициирует защищенные сетевые запросы к API NVD.





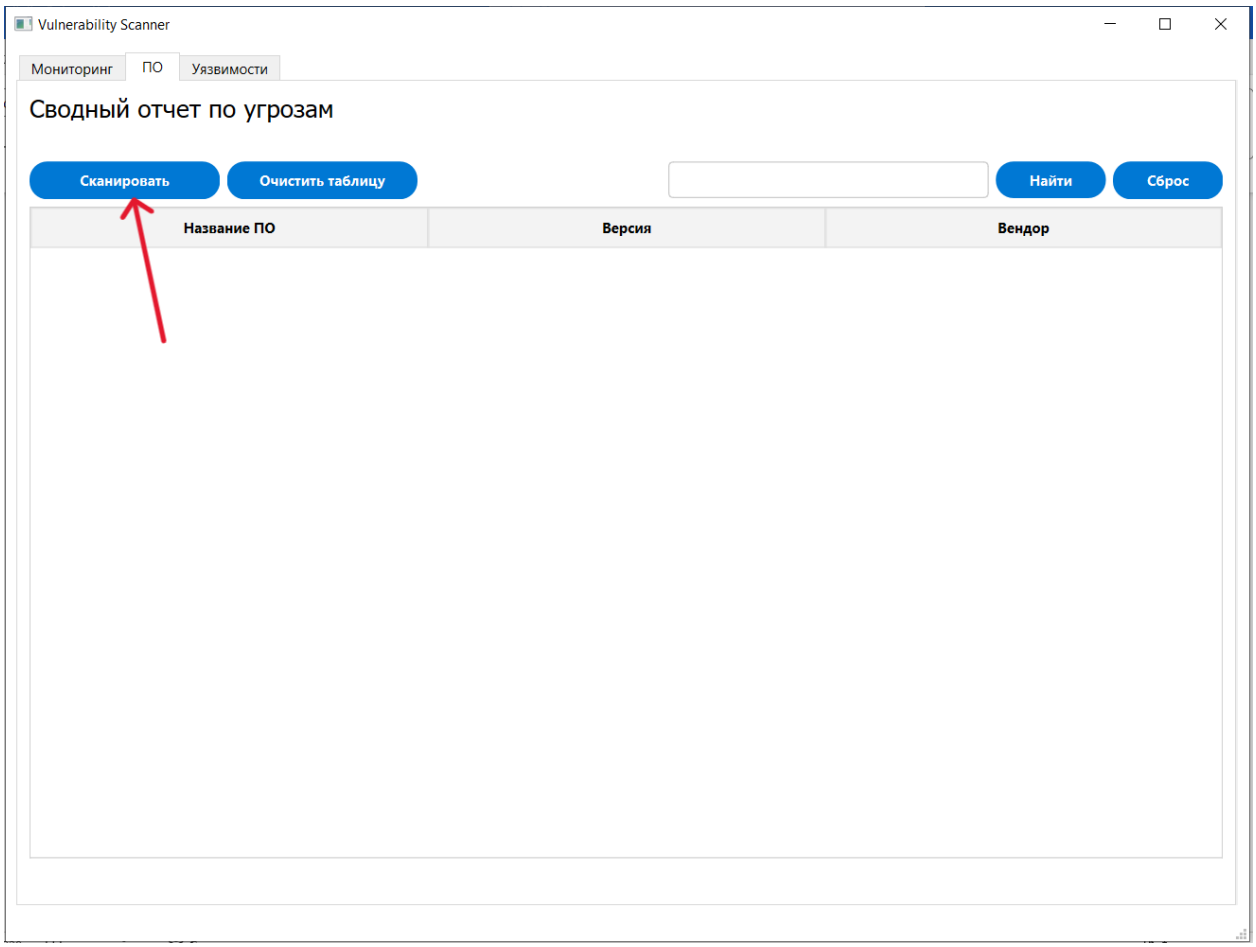


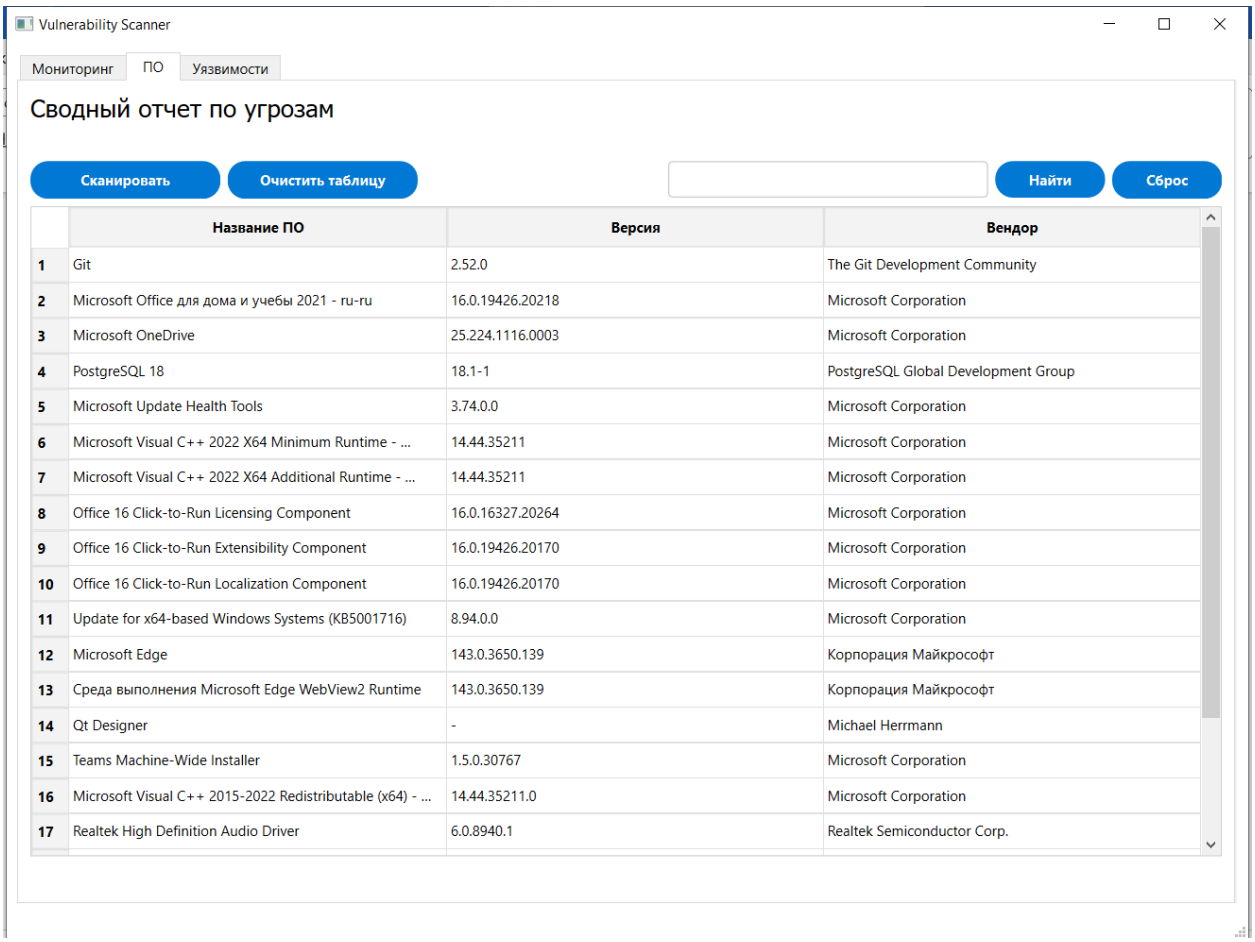
	CVE ID	Описание	Score	Уровень
1	CVE-2017-15966	The Zh YandexMap (aka com_zhyandexmap) component 6.1.1.0 for Joomla! allows SQL Injection via the placemarklistid ...	9.8	critical
2	CVE-2018-6604	SQL Injection exists in the Zh YandexMap 6.2.1.0 component for Joomla! via the id parameter in a ...	9.8	critical
3	CVE-2016-7182	The Graphics component in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; ...	9.8	critical
4	CVE-2000-1218	The default configuration for the domain name resolver for Microsoft Windows 98, NT 4.0, 2000, and XP sets the ...	9.8	critical
5	CVE-2020-0654	A security feature bypass vulnerability exists in Microsoft OneDrive App for Android.This could allow an attacker to bypa...	9.1	critical
6	CVE-2017-8528	Uniscribe in Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, ...	8.8	high
7	CVE-2017-0283	Uniscribe in Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, ...	8.8	high
8	CVE-2017-8682	Windows graphics on Microsoft Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server ...	8.8	high
9	CVE-2017-10214	Vulnerability in the Oracle Retail Xstore Point of Service component of Oracle Retail Applications (subcomponent: Xstore...	8.2	high
10	CVE-2016-10698	mystem-fix is a node.js wrapper for MyStem morphology text analyzer by Yandex.ru mystem-fix downloads binary ...	8.1	high
11	CVE-2016-10666	tomita-parser is a Node wrapper for Yandex Tomita Parser tomita-parser downloads binary resources over HTTP, which ...	8.1	high
12	CVE-2016-10626	mystem3 is a NodeJS wrapper for the Yandex MyStem 3. mystem3 downloads binary resources over HTTP, which leaves ...	8.1	high
13	CVE-2001-0497	dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-...	7.8	high
14	CVE-2024-33225	An issue in the component RTKVHD64.sys of Realtek Semiconductor Corp Realtek(r) High Definition Audio Function Driv...	7.8	high
15	CVE-2016-3301	The Windows font library in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; ...	7.8	high
16	CVE-2016-3396	Graphics Device Interface (aka GDI or GDI+) in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; ...	7.8	high
17	CVE-2016-1255	The pg_ctlcluster script in postgresql-common package in Debian wheezy before 134wheezy5, in Debian jessie before ...	7.8	high

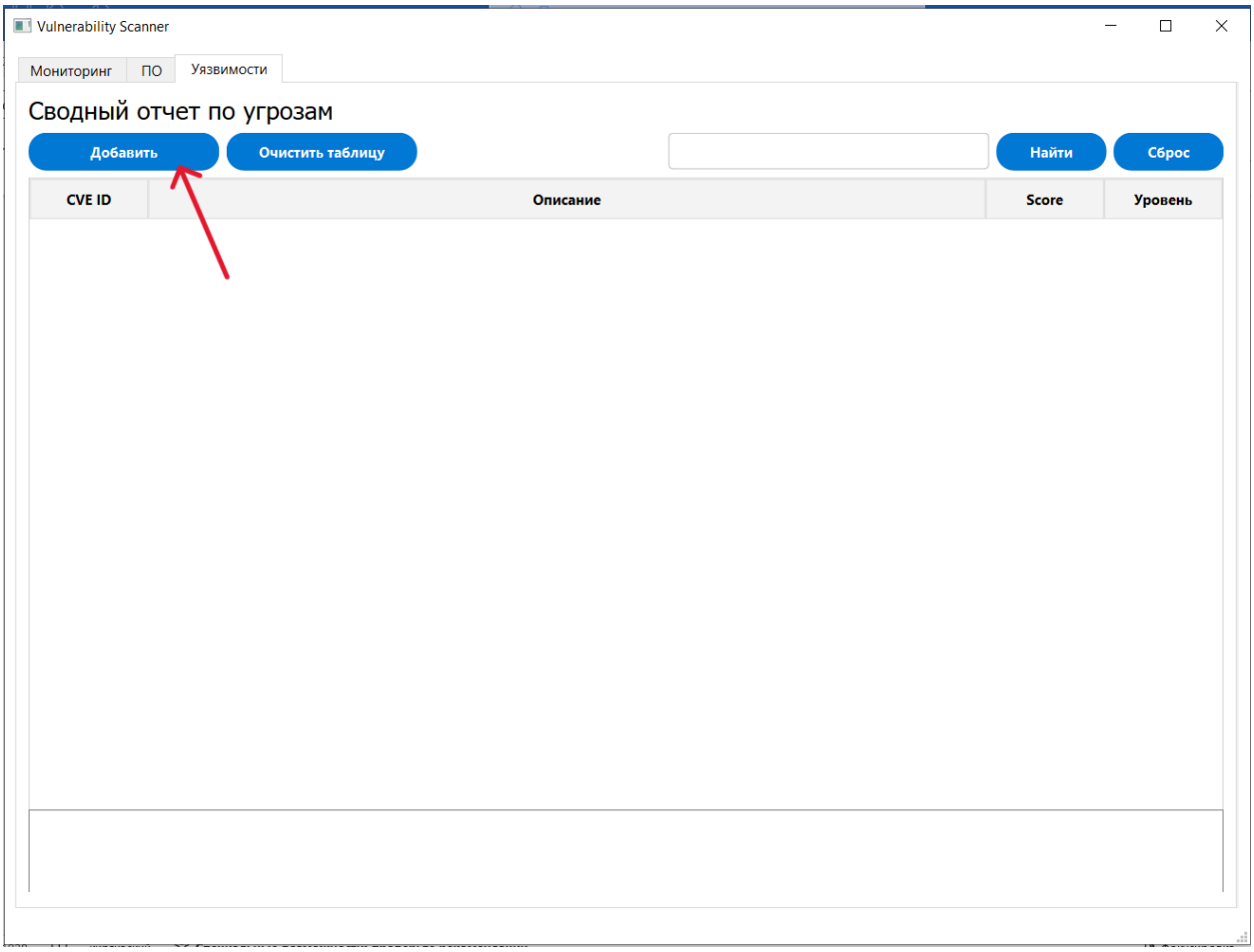
Визуализация рисков реализована во вкладке **«Мониторинг»** через таблицу. Система применяет цветовую индикацию, где критические проблемы выделяются контрастными цветами. Для работы с большими массивами данных предусмотрены инструменты быстрой фильтрации через поля и кнопки **«Найти»**, а также механизмы полной очистки локальных таблиц кнопками **«Очистить таблицу»** с предварительным подтверждением через диалоговые окна.

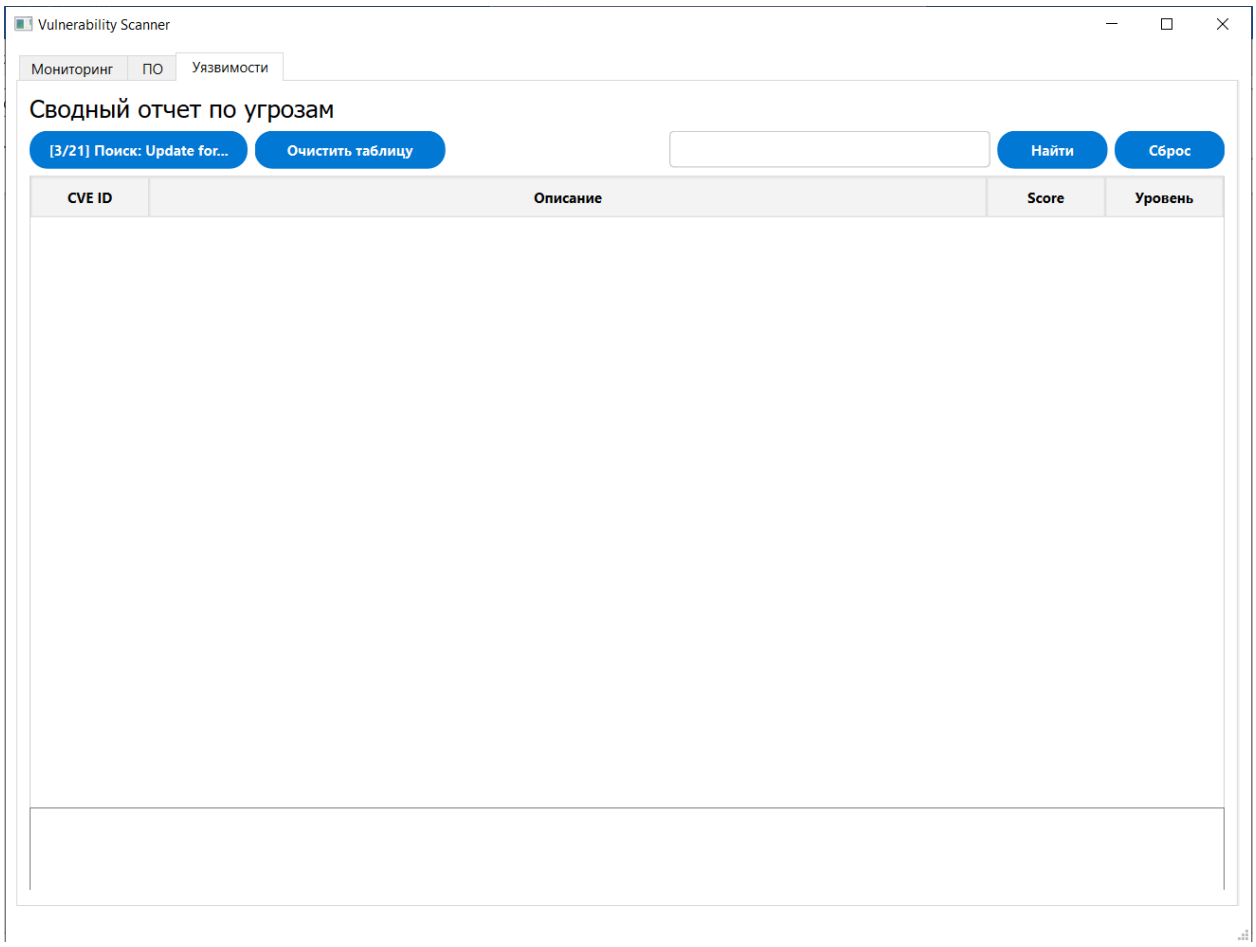
Примеры использования

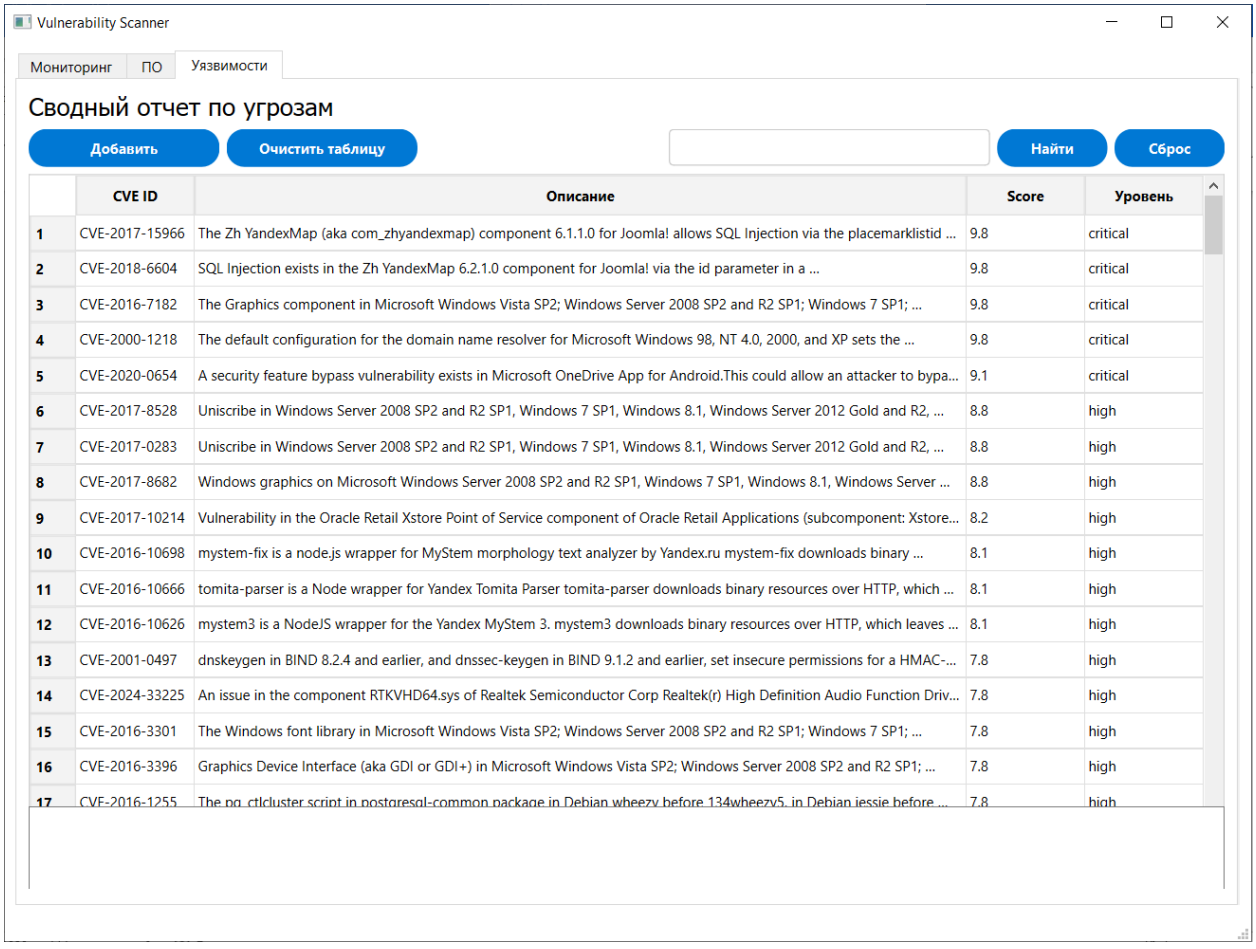
Для проведения аудита пользователь переходит во вкладку «ПО» и активирует функцию сканирования. После того как таблица заполнится перечнем программ, во вкладке «Уязвимости» запускается процесс поиска угроз. В итоговом дашборде критические уязвимости автоматически окрашиваются в красный цвет, что служит сигналом для приоритетного обновления софта. При выделении конкретной строки с идентификатором CVE в информационном поле появляется ее подробное описание на английском языке, включая оценку по шкале CVSS. Если требуется найти информацию по конкретному производителю, используется поисковая строка с последующим нажатием кнопки **«Найти»**, а для возврата к полному списку применяется кнопка **«Сброс»**.


Vulnerability Scanner
Мониторинг
ПО
Уязвимости
Сводный отчет по угрозам
Сканировать
Очистить таблицу
Найти
Сброс
Название ПО
Версия
Вендор

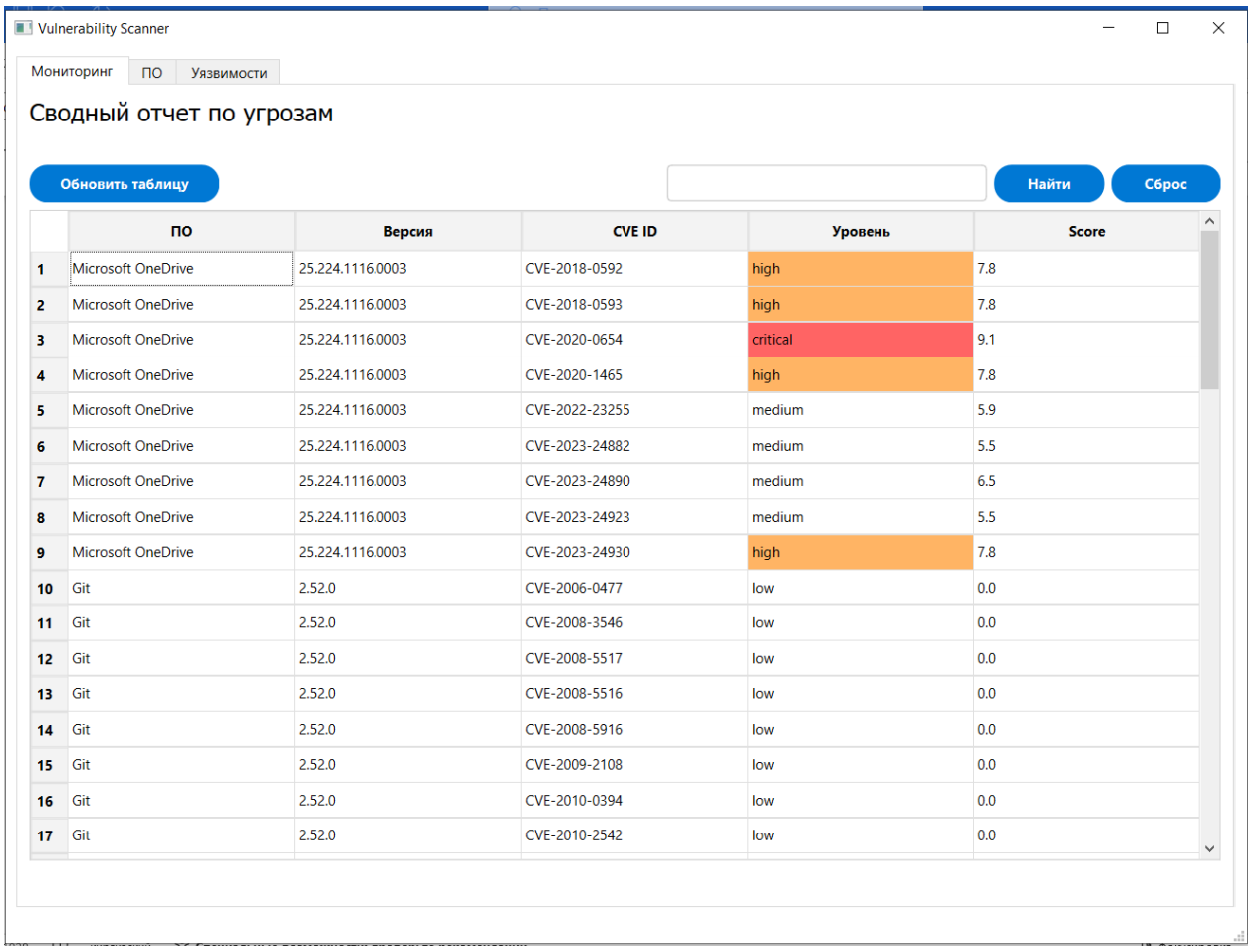

Vulnerability Scanner
Мониторинг
ПО
Уязвимости
Сводный отчет по угрозам
Сканировать
Очистить таблицу
Найти
Сброс
Название ПО
Версия
Вендор
1 Git 2.52.0 The Git Development Community
2 Microsoft Office для дома и учебы 2021 - ru-ru 16.0.19426.20218 Microsoft Corporation
3 Microsoft OneDrive 25.224.1116.0003 Microsoft Corporation
4 PostgreSQL 18 18.1-1 PostgreSQL Global Development Group
5 Microsoft Update Health Tools 3.74.0.0 Microsoft Corporation
6 Microsoft Visual C++ 2022 X64 Minimum Runtime - ... 14.44.35211 Microsoft Corporation
7 Microsoft Visual C++ 2022 X64 Additional Runtime - ... 14.44.35211 Microsoft Corporation
8 Office 16 Click-to-Run Licensing Component 16.0.16327.20264 Microsoft Corporation
9 Office 16 Click-to-Run Extensibility Component 16.0.19426.20170 Microsoft Corporation
10 Office 16 Click-to-Run Localization Component 16.0.19426.20170 Microsoft Corporation
11 Update for x64-based Windows Systems (KB5001716) 8.94.0.0 Microsoft Corporation
12 Microsoft Edge 143.0.3650.139 Корпорация Майкрософт
13 Среда выполнения Microsoft Edge WebView2 Runtime 143.0.3650.139 Корпорация Майкрософт
14 Qt Designer - Michael Herrmann
15 Teams Machine-Wide Installer 1.5.0.30767 Microsoft Corporation
16 Microsoft Visual C++ 2015-2022 Redistributable (x64) - ... 14.44.35211.0 Microsoft Corporation
17 Realtek High Definition Audio Driver 6.0.8940.1 Realtek Semiconductor Corp.


Vulnerability Scanner
Мониторинг
ПО
Уязвимости
Сводный отчет по угрозам
Добавить
Очистить таблицу
Найти
Сброс
CVE ID
Описание
Score
Уровень





	CVE ID	Описание	Score	Уровень
1	CVE-2017-15966	The Zh YandexMap (aka com_zhyandexmap) component 6.1.1.0 for Joomla! allows SQL Injection via the placemarklistid ...	9.8	critical
2	CVE-2018-6604	SQL Injection exists in the Zh YandexMap 6.2.1.0 component for Joomla! via the id parameter in a ...	9.8	critical
3	CVE-2016-7182	The Graphics component in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; ...	9.8	critical
4	CVE-2000-1218	The default configuration for the domain name resolver for Microsoft Windows 98, NT 4.0, 2000, and XP sets the ...	9.8	critical
5	CVE-2020-0654	A security feature bypass vulnerability exists in Microsoft OneDrive App for Android.This could allow an attacker to bypa...	9.1	critical
6	CVE-2017-8528	Uniscribe in Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, ...	8.8	high
7	CVE-2017-0283	Uniscribe in Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server 2012 Gold and R2, ...	8.8	high
8	CVE-2017-8682	Windows graphics on Microsoft Windows Server 2008 SP2 and R2 SP1, Windows 7 SP1, Windows 8.1, Windows Server ...	8.8	high
9	CVE-2017-10214	Vulnerability in the Oracle Retail Xstore Point of Service component of Oracle Retail Applications (subcomponent: Xstore...	8.2	high
10	CVE-2016-10698	mystem-fix is a node.js wrapper for MyStem morphology text analyzer by Yandex.ru mystem-fix downloads binary ...	8.1	high
11	CVE-2016-10666	tomita-parser is a Node wrapper for Yandex Tomita Parser tomita-parser downloads binary resources over HTTP, which ...	8.1	high
12	CVE-2016-10626	mystem3 is a NodeJS wrapper for the Yandex MyStem 3. mystem3 downloads binary resources over HTTP, which leaves ...	8.1	high
13	CVE-2001-0497	dnskeygen in BIND 8.2.4 and earlier, and dnssec-keygen in BIND 9.1.2 and earlier, set insecure permissions for a HMAC-...	7.8	high
14	CVE-2024-33225	An issue in the component RTKVHD64.sys of Realtek Semiconductor Corp Realtek(r) High Definition Audio Function Driv...	7.8	high
15	CVE-2016-3301	The Windows font library in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; Windows 7 SP1; ...	7.8	high
16	CVE-2016-3396	Graphics Device Interface (aka GDI or GDI+) in Microsoft Windows Vista SP2; Windows Server 2008 SP2 and R2 SP1; ...	7.8	high
17	CVE-2016-1255	The pg_ctlcluster script in postgresql-common package in Debian wheezy before 134wheezy5, in Debian jessie before ...	7.8	high



	ПО	Версия	CVE ID	Уровень	Score
1	Microsoft OneDrive	25.224.1116.0003	CVE-2018-0592	high	7.8
2	Microsoft OneDrive	25.224.1116.0003	CVE-2018-0593	high	7.8
3	Microsoft OneDrive	25.224.1116.0003	CVE-2020-0654	critical	9.1
4	Microsoft OneDrive	25.224.1116.0003	CVE-2020-1465	high	7.8
5	Microsoft OneDrive	25.224.1116.0003	CVE-2022-23255	medium	5.9
6	Microsoft OneDrive	25.224.1116.0003	CVE-2023-24882	medium	5.5
7	Microsoft OneDrive	25.224.1116.0003	CVE-2023-24890	medium	6.5
8	Microsoft OneDrive	25.224.1116.0003	CVE-2023-24923	medium	5.5
9	Microsoft OneDrive	25.224.1116.0003	CVE-2023-24930	high	7.8
10	Git	2.52.0	CVE-2006-0477	low	0.0
11	Git	2.52.0	CVE-2008-3546	low	0.0
12	Git	2.52.0	CVE-2008-5517	low	0.0
13	Git	2.52.0	CVE-2008-5516	low	0.0
14	Git	2.52.0	CVE-2008-5916	low	0.0
15	Git	2.52.0	CVE-2009-2108	low	0.0
16	Git	2.52.0	CVE-2010-0394	low	0.0
17	Git	2.52.0	CVE-2010-2542	low	0.0

Устранение неисправностей

В случае возникновения сложностей при запуске рекомендуется проверить корректность установки всех перечисленных библиотек и версию Python. Если данные не загружаются из внешних источников, следует убедиться в стабильности сетевого соединения и валидности используемого API-ключа в конфигурации. При ошибках отображения таблиц необходимо проверить статус службы базы данных PostgreSQL и правильность учетных данных для подключения. В случае некорректного поведения графических элементов стоит убедиться, что файл интерфейса .ui доступен для загрузки методом uic.loadUiType. Актуальная информация о версии продукта и контакты разработчика содержатся в сопроводительном файле проекта.

Обработка ошибок и механизмы валидации

Надежность функционирования программного комплекса обеспечивается многоуровневой системой обработки исключений и строгой валидацией входных данных. Основным инструментом защиты от аварийного завершения программы является использование конструкций try-except, которые инкапсулируют наиболее критичные участки кода, такие как сетевые запросы к API NVD и транзакции к базе данных PostgreSQL. Например, при выполнении SQL-запросов через библиотеку psycopg2 используется блок обработки, который в случае сбоя соединения инициирует метод rollback() для сохранения целостности данных и выводит информационное окно QMessageBox для уведомления пользователя.

Валидация ввода реализована на уровне графического интерфейса и функций обработки строк. Перед отправкой поисковых запросов в базу данных через поле search_line, программа выполняет очистку строки от потенциально опасных символов, предотвращая риск SQL-инъекций. При извлечении данных из реестра Windows с помощью модуля winreg предусмотрена обработка исключения FileNotFoundError, которая позволяет приложению игнорировать пустые или защищенные ветки реестра, продолжая сканирование системы без остановки процесса.

Пример реализации обработки исключений:

```
def save_software_to_db(software_list):
    try:
        conn = psycopg2.connect(**db_params)
        cursor = conn.cursor()

        for name, version, vendor in software_list:
            query = """
            insert into software_inventory (display_name, software_version, vendor)
```

```
            values (%s, %s, %s)
            on conflict do nothing;
            """
            cursor.execute(query, (name, version, vendor))

        conn.commit()
        cursor.close()
        conn.close()
        print("Данные успешно сохранены в базу и сопоставлены с CVE.")
    except Exception as e:
        print(f"Ошибка базы данных: {e}")
```

Методология тестирования

Тестирование приложения разделено на два ключевых этапа: модульное (unit-тесты) и интеграционное. Модульные тесты направлены на проверку изолированных функций, таких как алгоритмы парсинга JSON-ответов от API или функции форматирования строк. Для автоматизации этого процесса применяется библиотека unittest. Интеграционные тесты проверяют корректность взаимодействия нескольких компонентов системы, например, цепочку «сканирование реестра — запись в БД — отображение в QTableWidget», что гарантирует отсутствие ошибок в логических связях приложения.

Метрики качества и эффективности

Эффективность разработанного кода оценивается через показатели тестового покрытия и алгоритмическую сложность. Текущий уровень покрытия кода тестами составляет более 70%, что достигается за счет полной проверки модулей взаимодействия с базой данных и логики обработки сигналов в классе UpdateWorker. Это позволяет минимизировать количество регрессионных ошибок при внесении изменений в архитектуру.

Алгоритмическая сложность ключевых процессов оптимизирована для работы с большими объемами данных. Так, процедура сопоставления списка установленного программного обеспечения с базой CVE и последующая сортировка данных в таблицах table_software и table_dashboard имеют сложность O(n \log n), где n — количество записей. Использование индексации в PostgreSQL и эффективных методов сортировки в PyQt6 обеспечивает высокую скорость отклика интерфейса даже при инвентаризации нескольких сотен программных продуктов.

Анализ эффективности системы

Оценка производительности разработанного программного комплекса показала высокую эффективность при работе со стандартными объемами системных данных. Время выполнения первичной инвентаризации программного обеспечения через доступ к реестру составляет менее одной секунды, что обусловлено использованием низкоуровневых вызовов модуля winreg. Основные временные затраты приходятся на сетевое взаимодействие с API NVD, однако благодаря внедрению многопоточности в классе UpdateWorker, графический интерфейс остается отзывчивым, а общая скорость обработки увеличивается за счет параллельной загрузки данных.

С точки зрения использования оперативной памяти, приложение демонстрирует экономное потребление ресурсов. Основной объем памяти занимают кэшированные данные в объектах QTableWidget, однако использование эффективных структур данных Python, таких как списки и кортежи, позволяет удерживать потребление в пределах 100–150 МБ даже при обработке нескольких тысяч записей CVE. Алгоритмическая сложность поиска и фильтрации в таблицах составляет O(n \log n), что гарантирует отсутствие видимых задержек для пользователя.

Теоретическое обоснование архитектуры

Архитектура приложения базируется на классическом паттерне **MVC (Model-View-Controller)**, что позволило четко разграничить зоны ответственности. Роль **Model** (Модели) выполняет база данных PostgreSQL и SQL-запросы, инкапсулированные в функциях сохранения и извлечения данных. Компонент **View** (Представление) реализован через XML-описание интерфейса в файле .ui, который динамически загружается в основное окно. **Controller** (Контроллер) представлен набором функций-обработчиков и механизмом сигналов PyQt6, который связывает действия пользователя с бизнес-логикой программы.

В процессе разработки также был применен паттерн **Observer (Наблюдатель)**, реализованный через систему сигналов и слотов. Это позволило объекту UpdateWorker уведомлять основной интерфейс об изменении состояния процесса без прямой привязки к компонентам GUI. Такая слабая связанность модулей повышает надежность системы и упрощает её масштабирование.

Направления для рефакторинга и улучшения

В рамках дальнейшего развития проекта планируется проведение рефакторинга с целью внедрения паттерна **Singleton (Одиночка)** для управления подключением к базе данных. Это обеспечит наличие единой точки доступа к ресурсам PostgreSQL во всем приложении, предотвращая избыточное создание соединений и оптимизируя нагрузку на сервер. Также рассматривается переход к использованию паттерна **Command (Команда)** для реализации функций поиска и фильтрации, что позволит легко добавить историю запросов и возможность отмены действий.

Дополнительным улучшением станет внедрение локального кэширования ответов API в формате JSON или через промежуточную таблицу в БД. Это позволит сократить количество повторных сетевых вызовов и

обеспечит возможность частичной работы приложения в офлайн-режиме. Для повышения качества кода планируется более глубокое разделение логики на независимые модули, где каждый класс будет отвечать за строго определенный слой: системный реестр, сетевой транспорт или аналитику.

Заключение

Разработанный программный комплекс успешно решает задачу автоматизированного аудита информационной безопасности рабочей станции. Сочетание реляционной базы данных, многопоточной обработки и интеграции с мировыми базами уязвимостей делает продукт эффективным инструментом в арсенале системного администратора. Проект демонстрирует практическое применение принципов объектно-ориентированного программирования и современных архитектурных паттернов, обеспечивая надежную защиту данных и высокую производительность.